Modified Roland PG1000

Program Listings

Modified Roland PG1000

78C10 eForth Assembly

```
Page 60,132
;===============================================================
;
;   eForth 1.0 by Bill Muench and C. H. Ting, 1990
;
;   This is an implementation for the NEC 78C10 microcomputer by
;   John Talbert, 1994, Oberlin Conservatory.
;
;   Register Use:       Interpreter Pointer = DE
;                       Data Stack Pointer  = SP
;                       Return Stack Pointer  = HL

;                       Free to use: BC, EA, VA, Alternate Registers.
;
;   'doList'  is accessed as a subroutine through a CALT instruction
;            (Call to Jump Table).  This shows up as a 'DB 80H' line
;            in the $COLON and $USER Macros.  When executed the
;            processor jumps to an address vector located at 80H.  The
;            vectored 'doList' code is then located at 0F0H. The word
;            'call,' was changed to load 80H into the code area for a
;            doLST assembly.
;
;   A 9600 Baud serial I/O is provided.  PortB/bit0 is used for serial
;       output and PortC/bit3 (INT2) is used for the serial input.  The
;       serial input is interrupt driven with a vectored interrupt routine
;       located at 0A0H.  The code words ?RX, TX!, and !IO make up the
;       rest of the serial I/O code.  Three USER variables have been set
;       up for use by these serial I/O routines:  SERIN, which holds the
;       received character and a flag; HAFBIT, which adjusts the software
;       timing of the receiver to read in the middle of each bit frame
;       (set it for 1/2 the BITIME minus 5); and BITIME, which adjusts the
;       software for a specific baud rate (17H for 9600 baud assuming a 12Mhz
;       processor clock).
;
;   The 78C10 is an 8-bit micro, therefore cell aligning to even addresses
;       is unnecessary. The $ALIGN Macro was taken out along with the NOP's
;       used for cell alignment in the other Macros.  All occurrences of the
;       word ALGND were erased also. The word SEE no longer works because it
;       relies on cell alignment.
;
;   All of the system FORTH code is to be stored in ROM (up to 32K) starting
;       at address 0000H. Then there is 2K of RAM starting at address COOOH.
;       This memory setup required the following changes:
;               1) Return and Data stacks and TIB moved to RAM.
;                  (See the Memory allocation EQU assignments.)
;               2) The USER variables were moved to the micro's
;                  internal RAM at FF00H to FFFFH.
;               3) PAD word was changed to move the temporary buffer
;                  area to RAM space.
;               4) The vocabulary pointers found in the word FORTH were
;                  moved to RAM space by creating two new USER variables,
;                  FHEAD and FLINK and changing DOVOC to read:
;                  DW FHEAD,CNTXT,STORE,EXIT.
;               5) NTOP and CTOP were moved to RAM space to allow dictionary
;                  expansion into RAM space.
;
;   Several words were added to the ROM Dictionary.  The simple operators
;       1+,1-,2+,2-,2*,2/, were defined in machine code.  The words C, ,
;       CCOMPILE, CODE, and ENDCODE were created to enable the creation
;       of code definition.
;
;    The NEC78C10 offers the following advantages:
;               1) Ten 16-bit internal registers and a 16-bit ALU.
;                  Many 16-bit instructions for those FORTH stack operations.
;               2) Three 8-bit I/O ports.
;               3) Eight 8-bit Analog to Digital Converters.
;               4) Internal counters and programmable clock generators.
;               5) Internal hardware serial I/O.  (can be used for MIDI I/O).
;               6) 64K address space including 256 bytes of internal RAM.
;
;
;=====================================================================
;; Version control
VER             EQU     01H                     ;major release version
EXT             EQU     01H                     ;minor extension

;; Constants
COMPO           EQU     040H                    ;lexicon compile only bit
IMEDD           EQU     080H                    ;lexicon immediate bit
MASKK           EQU     07F1FH                  ;lexicon bit mask

CELLL           EQU     2                       ;size of a cell
BASEE           EQU     10                      ;default radix
VOCSS           EQU     6                       ;depth of vocabulary stack

BKSPP           EQU     8                       ;backspace
LF              EQU     10                      ;line feed
CRR             EQU     13                      ;carriage return
ERR             EQU     27                      ;error escape
TIC             EQU     39                      ;tick

CALLL           EQU     80H                     ;CALT opcodes

;; Memory allocation    0//code>--//--<name//up>--<sp//tib>--rp//em

COLDD           EQU     00100H                  ;cold start
```

```
RPP             EQU     0C2F0H                  ;start of return stack (RP0)
TIBB            EQU     0C200H                  ;terminal input buffer (TIB)
SPP             EQU     0C1F0H                  ;start of data stack (SP0)
UPP             EQU     0FF00H                  ;start of user area (UP0)
NAMEE           EQU     01FFDH                  ;name dictionary
CODEE           EQU     00300H                  ;code dictionary
CTOP            EQU     0C390H                  ;RAM code dict. expansion
NTOP            EQU     0C7FFH                  ;RAM name dict. expansion
PADD            EQU     0C300H                  ;PAD area

;; Initialize assembly variables

_LINK   = 0                                     ;force a null link
_NAME   = NAMEE                                 ;initialize name pointer
_CODE   = CODEE                                 ;initialize code pointer
_USER   = 4*CELLL                               ;first user variable offset

;; Define assembly macros

;       Compile a code definition header.

$CODE   MACRO   LEX,NAME,LABEL
LABEL:                                          ;;assembly label
     _CODE   = $                             ;;save code pointer
     _LEN    = (LEX AND 01FH)/CELLL          ;;string cell count, round down
     _NAME   = _NAME-((_LEN+3)*CELLL)        ;;new header on cell boundary
ORG     _NAME                                   ;;set name pointer
     DW      _CODE,_LINK                     ;;token pointer and link
     _LINK   = $                             ;;link points to a name string
     DB      LEX,NAME                        ;;name string
ORG     _CODE                                   ;;restore code pointer
     ENDM

;       Compile a colon definition header.

$COLON  MACRO   LEX,NAME,LABEL
     $CODE   LEX,NAME,LABEL
     DB 80H                                  ;;include CALT doLIST
     ENDM

;       Compile a user variable header.

$USER   MACRO   LEX,NAME,LABEL
     $CODE   LEX,NAME,LABEL
     DB 80H                                  ;;include CALT doLIST
     DW      DOUSE,_USER                     ;;followed by doUSER and offset
     _USER   = _USER+CELLL                   ;;update user area offset
     ENDM
```

```
;       Compile an inline string.

D$      MACRO   FUNCT,STRNG
     DW      FUNCT                              ;;function
     _LEN    = $                                ;;save address of count byte
     DB      0,STRNG                            ;;count byte and string
     _CODE   = $                                ;;save code pointer
ORG     _LEN                                       ;;point to count byte
     DB      _CODE-_LEN-1                       ;;set count
ORG     _CODE                                      ;;restore code pointer
     ENDM

;       Compile a stored string.

SD$     MACRO   STRNG
     DW DOLIT
     _LEN    = $ + 4                            ;;save address of count byte
     DW      _LEN,EXIT                          ;;save cnt address on stack
     DB      0,STRNG                            ;;count byte and string
     _CODE   = $                                ;;save code pointer
ORG     _LEN                                       ;;point to count byte
     DB      _CODE-_LEN-1                       ;;set count
ORG     _CODE                                      ;;restore code pointer
     ENDM

;       Assemble inline direct threaded code ending.

$NEXT   MACRO
     DB 48H,84H                                 ;;EA<(DE)++,next code address into
AX
     DB 48H,28H                                 ;;JMP EA,jump directly to code
address
     ENDM

;; Main entry points and COLD start data

MAIN    SEGMENT
ASSUME  CS:MAIN,DS:MAIN,ES:MAIN,SS:MAIN

ORG     0000H

ORIG:   DB 54H,00,01,00                            ;RESET vector, JMP 0100H
     DB 0AAH,62H,0,0                            ;NMI vector, EI RETI
     DB 8 DUP(0)                                ;INT T0/T1 vector
     DB 54H,0A0H,00H, 5 DUP(0)                  ;INT1/2 vector, JMP 00A0H
     DB 8 DUP(0)                                ;INT E1/E0 vector
     DB 54H,00,02, 5 DUP(0)                     ;INT EIN/AD vector, JMP 0200H
     DB 8 DUP(0)                                 ;INT SR/ST vector
     DB 48 DUP(0)                                ;FREE
     DB 32 DUP(0)                                ;SOFTI vector at 0060H

ORG     00A0H

; Vectored INT2 routine for Serial Input from Host Computer.
; Uses address FFF0 as a counter location - do not use elsewhere!
     DB 0B1H                ;PUSH BC
     DB 0B2H                ;PUSH DE
     DB 0B0H                ;PUSH VA
     DB 68H,0FFH            ;MVI, V<FF
     DB 71H,0F0H,07H        ;MVIW, (V/F0)<07, number of bits to receive.
     DB 70H,1FH,04CH,0FFH   ;LBCD, BC<(FF4C), wait for a half bit.
     DB 53H                 ;DCR, C<C-1 skip, LOOP1
     DB 0FEH                ;JR, Jump to loop1
     DB 52H                 ;DCR, B<B-1 skip
     DB 0FCH                ;JR, Jump to loop1
     DB 70H,1FH,4EH,0FFH    ;LBCD, BC<(FF4E),wait 1 bit time,  LOOP2
     DB 53H                 ;DCR, C<C-1 skip
     DB 0FEH                ;JR, Jump to loop2
     DB 52H                 ;DCR, B<B-1 skip
     DB 0FCH                ;JR, Jump to loop2
     DB 04CH,0C2H           ;MOV, A<PC, read serial input on pc3
     DB 48H,31H,48H,31H     ;Rotate PC3 bit into Cy
     DB 48H,31H,48H,31H     ;RLR, A rotate right 4xs
     DB 0CH                 ;MOV, A<D, D collects the bits
     DB 48H,31H             ;RLR, shift in next bit, CY to top of D
     DB 1CH                 ;MOV, D<A
     DB 30H,0F0H            ;DCRW, (V/F0)<(V/F0)-1 skip
     DB 0E7H                ;JR, Jump to loop2 for next bit.
     DB 70H,1FH,4EH,0FFH    ;LBCD, BC<(FF4E)
     DB 53H,0FEH,52H,0FCH   ;DCR JR DCR JR, stop bit loop time.
     DB 71H,04BH,0FFH       ;MVIW, (V/4B)<FF, load flag
     DB 0CH,63H,04AH        ;MOV STAW, A<D (V/4A)<A, load data
     DB 0A0H,0A2H,0A1H      ;POP, restore AV DE and BC
     DB 48H,44H,0           ;SKIT,NOP
     DB 0AAH,062H           ;EI RETI, enable interrupts and return

;; Kernel doLST routine.  Always accessed by the CALT instruction: 80H
;; which is a Call Subroutine to jump to address vector located at 0080H.

ORG     00F0H
     DB 33H,33H             ;HL<HL-2
     DB 0A6H                ;EA<DE
     DB 48H,93H             ;(HL)<EA
     DB 0A2H                ;POP DE previously pushed by CALT
     DB 48H,84H             ;EA<(DE)++, $NEXT
     DB 48H,28H              ;JMP EA
ORG     0080H
     DB 0F0H,0               ; set up vector to doLST

ORG     COLDD                   ;Beginning of Cold Boot
     DB 69H,0FH,4DH,0D0H     ;MM<0F, memory map (11-8)
     DB 69H,0FFH,4DH,0D2H    ;MA<FF, pa inputs (4-2)
     DB 69H,00H,4DH,0D3H     ;MB<00, pb outputs (4-6)
     DB 64H,01H,05H          ;PB<5
     DB 4DH,0D7H             ;MF<00, pf outputs (4-15)
     DB 69H,0AH,4DH,0D4H     ;MC<0A, pc1/3 inputs (4-9)
     DB 69H,0BH,4DH,0D1H     ;MCC<0B, pc mode (4-8)
     DB 64H,02H,04H          ;PC<04
     DB 64H,81H,06H          ;SMH<06, serial mode (7-7)
     DB 69H,4EH,4DH,0CAH     ;SML<4E, serial mode (7-9)
     DB 04H                  ;SP<SPP, stack pointer=data stack
     DB LOW SPP
     DB HIGH SPP
     DB 34H                  ;HL<RPP, HL=return stack pointer
     DB LOW RPP
     DB HIGH RPP
     DB 69H,00H,4DH,0E8H     ;ZCM<0, zero cross disabled (3-26)
     DB 68H,0FFH             ;V<FF
     DB 10H,68H,0FFH,69H,0   ;V'<FF, A"<0, V<FF, A<0

;; timer setups for Midi and LCD use
     DB 69H,64H,4DH,0DAH     ;TM0<64, timer0 (5-1)
     DB 69H,0FFH,4DH,0DBH    ;TM1<FF, timer1 (5-1)
     DB 64H,85H,0B3H         ;TMM<B3, timer mode (5-6)
     DB 44H,60H,0EAH,48H,0D3H        ;ETM1<EA = EA60 (6-2)
     DB 64H,83H,0CCH         ;EOM<CC, timer event mode (6-14)
     DB 69H,5CH,4DH,0CCH     ;ETMM<5C, timer event mode (6-11)

     DB 54H,00,03H           ;JMP to 0300, high level cold start
                      ;COLD WORD MOVED TO THE START OF CODE AREA.
                      ;ATTEMPTED TO AUTOMATE-JMP COLD-WITH $JUMP
                      ;BUT MACRO PRODUCES ERROR CODES.

; COLD start moves the following to USER variables.
; MUST BE IN SAME ORDER AS USER VARIABLES.

UZERO:          DW      4 DUP (0)               ;reserved
          DW      SPP                     ;SP0
          DW      RPP                     ;RP0
          DW      QRX                     ;'?KEY
          DW      TXSTO                   ;'EMIT
          DW      ACCEP                   ;'EXPECT
          DW      KTAP                    ;'TAP
```

```
        DW      TXSTO                       ;'ECHO
        DW      DOTOK                       ;'PROMPT
        DW      BASEE                       ;BASE
        DW      0                           ;tmp
        DW      0                           ;SPAN
        DW      0                           ;>IN
        DW      0                           ;#TIB
        DW      TIBB                        ;TIB
        DW      0                           ;CSP
        DW      INTER                       ;'EVAL
        DW      NUMBQ                       ;'NUMBER
        DW      0                           ;HLD
        DW      0                           ;HANDLER
        DW      0                           ;CONTEXT pointer
        DW      VOCSS DUP (0)               ;vocabulary stack
        DW      0                           ;CURRENT pointer
        DW      0                           ;vocabulary link pointer
        DW      0                           ;FORTH HEAD
        DW      0                           ;FORTH LINK
        DW      CTOP                        ;CP
        DW      NTOP                        ;NP
        DW      LASTN                       ;LAST
        DW      0                           ;SERIN host receive char & flag
        DW      06H                         ;HAFBIT time for serial host,
                                ; (1/2 BITIME - 5)
        DW      16H                         ;BITIME baud for serial host

ULAST:

ORG 0200H

;       Interrupt routine for Analog to Digital Converters

     DB 10H                 ;EXA
     DB 11H                 ;EXX
;  Load ADC Address and Counter into HL.  Uses FFF2 and FFF3.
     DB 68H,0FFH            ;V'<FF
     DB 01H,0F2H            ;A<(V/F2)
     DB 1EH                 ;H<A
     DB 01H,0F3H            ;A<(V/F3)
     DB 1FH                 ;L<A
;  Store ADC 0.
     DB 2BH                 ;A<(HL)
     DB 57H,80H             ;A AND 80, Skip if zero
     DB 0CDH                ;Jump to EXIT if slider is disabled.
     DB 1AH                 ;B<A
     DB 4CH,0E0H            ;A<CR0
     DB 48H,21H             ;A Shift right, Midi is 7 bits, throw LSB.
```

```
    DB 60H,6AH                ;B-A, Skip if not zero
    DB 0C5H                   ;Jump to EXIT if slider has not changed.
    DB 3DH                    ;(HL)<A, Store slider data, 0 in top bit.
    DB 69H,0FFH               ;A<FF
    DB 0BFH,38H               ;(HL+38)<A, Store slider change flag.
    DB 32H                    ;HL<HL+1, EXIT
;  Store ADC 1.
    DB 2BH                    ;A<(HL)
    DB 57H,80H                ;A AND 80, Skip if zero
    DB 0CDH                   ;Jump to EXIT if slider is disabled.
    DB 1AH                    ;B<A
    DB 4CH,0E1H               ;A<CR1
    DB 48H,21H                ;A Shift right, Midi is 7 bits, throw LSB.
    DB 60H,6AH                ;B-A, Skip if not zero
    DB 0C5H                   ;Jump to EXIT if slider has not changed.
    DB 3DH                    ;(HL)<A, Store slider data, 0 in top bit.
    DB 69H,0FFH               ;A<FF
    DB 0BFH,38H               ;(HL+38)<A, Store slider change flag.
    DB 32H                    ;HL<HL+1, EXIT
;  Store ADC 2.
    DB 2BH                    ;A<(HL)
    DB 57H,80H                ;A AND 80, Skip if zero
    DB 0CDH                   ;Jump to EXIT if slider is disabled.
    DB 1AH                    ;B<A
    DB 4CH,0E2H               ;A<CR2
    DB 48H,21H                ;A Shift right, Midi is 7 bits, throw LSB.
    DB 60H,6AH                ;B-A, Skip if not zero
    DB 0C5H                   ;Jump to EXIT if slider has not changed.
    DB 3DH                    ;(HL)<A, Store slider data, 0 in top bit.
    DB 69H,0FFH               ;A<FF
    DB 0BFH,38H               ;(HL+38)<A, Store slider change flag.
    DB 32H                    ;HL<HL+1, EXIT
;  Store ADC 3.
    DB 2BH                    ;A<(HL)
    DB 57H,80H                ;A AND 80, Skip if zero
    DB 0CDH                   ;Jump to EXIT if slider is disabled.
    DB 1AH                    ;B<A
    DB 4CH,0E3H               ;A<CR3
    DB 48H,21H                ;A Shift right, Midi is 7 bits, throw LSB.
    DB 60H,6AH                ;B-A, Skip if not zero
    DB 0C5H                   ;Jump to EXIT if slider has not changed.
    DB 3DH                    ;(HL)<A, Store slider data, 0 in top bit.
    DB 69H,0FFH               ;A<FF
    DB 0BFH,38H               ;(HL+38)<A, Store slider change flag.
    DB 32H                    ;HL<HL+1, EXIT
;  Update Counters
    DB 0FH                    ;A<L
    DB 37H,37H                ;A-37H, Skip if borrow
```

```
     DB 69H,0                 ;A<0, Reset counter after 56D sliders.
     DB 63H,0F3H              ;(V/F3)<A, Load counter.
     DB 48H,25H,48H,25H       ;A shift logical left 2xs.
     DB 1AH                   ;B<A
     DB 74H,0AH,0E0H          ;B<B AND E0
     DB 4CH,0C2H              ;A<Pc
     DB 07H,1FH               ;A<A AND 1F
     DB 60H,9AH               ;A<A OR B
     DB 4DH,0C2H              ;Pc<A, Load high 3 bits of slider select.
     DB 64H,90H,08H           ;Invert ANM bit and restart conversion.
;  Return from Interrupt.
     DB 10H                   ;EXA
     DB 11H                   ;EXX
     DB 0AAH                  ;EI
     DB 62H                   ;RETI

ORG     CODEE                                            ;start code dictionary

;   COLD         ( -- )
;                The hilevel cold start sequence.
          CCOLD = $
          $COLON  4,'COLD',COLD
COLD1:           DW      DOLIT,UZERO,DOLIT,UPP
          DW      DOLIT,ULAST-UZERO,CMOVE ;initialize user area
          DW      PRESE                   ;initialize stack and TIB
          DW      TBOOT,ATEXE             ;application boot
          DW      FORTH,CNTXT,AT,DUPP     ;initialize search order
          DW      CRRNT,DSTOR,OVERT
          DW      LCDINIT                 ;initialize LCD
          DW      QUIT                    ;start interpretation
          DW      BRAN,COLD1              ;just in case


;; Device dependent I/O

;   BYE          ( -- )
;                Exit eForth.
          $CODE   3,'BYE',BYE
          DB 54H,0,0                    ;JMP Reset Vector

;   ?RX          ( -- c T | F )
;                Return input character and true, or a false if no input.

          $CODE   3,'?RX',QRX
     DB 68H,0FFH              ;MVI, V<FF
     DB 01H,4BH               ;LDAW, A<(V/4B) read serial-in flag
     DB 47H,0FFH              ;ONI, A AND FF skip if flag not zero
     DB 0CAH                  ;JR, jump ahead1
```

```
    DB 71H,04BH,0             ;MVIW, (V/4B)<0, reset flag to zero
    DB 70H,1FH,4AH,0FFH       ;LBCD, BC<(FF4A), read serin data
    DB 0B1H                   ;PUSH BC, push serial input data to stack
    DB 69H,0FFH               ;A<FF
    DB 1BH                    ;C<A, AHEAD1
    DB 6AH,0                  ;B<0
    DB 0B1H                   ;PUSH BC, push serial input flag to stack
    $NEXT


;   TX!         ( c -- )
;               Send character c to the output device.

        $CODE   3,'TX!',TXSTO
    DB 0BAH                   ;Disable Interrupts
    DB 0A1H                   ;POP BC, pop char into C
    DB 0B2H                   ;PUSH DE, store interpreter pointer
    DB 0BH,1CH                ;A<C, D<A, char in A and D
    DB 68H,0FFH               ;V<FF
    DB 71H,0F0H,07H           ;(V/F0)<7
    DB 60H,91H                ;A<A EXOR A
    DB 6DH,01H                ;E<01
    DB 4DH,0C1H               ;PB<A
    DB 70H,1FH,04EH,0FFH      ;BC<(FF4E) set baud, LOOP1
    DB 53H,0FEH,52H,0FCH      ;C<C-1, JR, B<B-1, JR, jr to loop1
    DB 0CH                    ;A<D
    DB 07H,01H                ;A<A AND 01
    DB 4DH,0C1H               ;PB<A, send a bit
    DB 0CH                    ;A<D
    DB 48H,31H                ;A rotate logical right
    DB 1CH                    ;D<A
    DB 0,0,0,0                ;NOPs to make rec loop = transmit loop.
    DB 30H,0F0H               ;(V/F0)<(V/F0)-1 skip
    DB 0E8H                   ;JR, jump to loop1
    DB 0DH                    ;A<E
    DB 51H                    ;A<A-1 skip
    DB 0C6H                   ;JR, jump to loop2
    DB 0A2H                   ;POP DE, restore interpreter pointer
    DB 0AAH                   ;Enable Interrupts
    $NEXT                     ;End of routine

    DB 6CH,03H                ;D<03, LOOP2
    DB 1DH                    ;E<A
    DB 71H,0F0H,01            ;(V/F0)<01
    DB 4FH,0D7H               ;JRE, jump to loop1

;   !IO         ( -- )
;               Initialize the serial I/O devices.
```

```
          $CODE   3,'!IO',STOIO
     DB 69H,0EFH,4DH,0C7H    ;MKL<EF, enable int2 interrupt and
     DB 69H,0FFH,4DH,0C6H    ;MKH<FF, disable all others with mask
     DB 0AAH                 ;EI, enable interrupt
     $NEXT


;; The kernel

;   doLIT       ( -- w )
;               Push an inline literal.

          $CODE   COMPO+5,'doLIT',DOLIT
          DB 48H,84H                        ;EA<(DE)++
          DB 0B4H                           ;PUSH EA
          $NEXT

;   EXIT        ( -- )
;               Terminate a colon definition.

          $CODE   4,'EXIT',EXIT
          DB 48H,85H                        ;EA<(HL)++
          DB 0B6H                           ;DE<EA
          $NEXT

;   EXECUTE     ( ca -- )
;               Execute the word at ca.

          $CODE   7,'EXECUTE',EXECU
          DB 0A1H                           ;POP BC
          DB 21H                            ;JMP BC

;   next        ( -- )
;               Run time code for the single index loop.
;               : next ( -- ) \ hilevel model
;                 r> r> dup if 1 - >r @ >r exit then drop cell+ >r ;

          $CODE   COMPO+4,'next',DONXT
          DB 6AH,0                          ;B<00
          DB 6BH,1                          ;C<01
          DB 48H,83H                        ;EA<(HL)
          DB 74H,0B5H                       ;EA<EA-BC Skip if no borrow
          DB 0C9H                           ;JMP NEXT1
          DB 48H,93H                        ;(HL)<EA
          DB 48H,82H                        ;EA<(DE)
          DB 0B6H                           ;DE<EA
          $NEXT
```

```
NEXT1:          DB 22H,22H                          ;DE<DE+2
        DB 32H,32H                          ;HL<HL+2
        $NEXT

;   ?branch     ( f -- )
;               Branch if flag is zero.

        $CODE   COMPO+7,'?branch',QBRAN
        DB 6AH,0FFH                         ;B<FF
        DB 6BH,0FFH                         ;C<FF
        DB 0A4H                             ;POP EA
        DB 74H,0CDH                         ;EA AND BC Skip if not zero
        DB 0C6H                             ;JMP BRAN1
        DB 22H,22H                          ;DE<DE+2
        $NEXT
BRAN1:          DB 48H,82H                          ;EA<(DE)
        DB 0B6H                             ;DE<EA
        $NEXT

;   branch      ( -- )
;               Branch to an inline address.

        $CODE   COMPO+6,'branch',BRAN
        DB 48H,82H                          ;EA<(DE)
        DB 0B6H                             ;DE<EA
        $NEXT

;   !           ( w a -- )
;               Pop the data stack to memory.

        $CODE   1,'!',STORE
        DB 0A1H                             ;POP BC, address
        DB 0A4H                             ;POP EA, data
        DB 09H                              ;A<EAL
        DB 39H                              ;(BC)<A
        DB 12H                              ;BC<BC+1
        DB 08H                              ;A<EAH
        DB 39H                              ;(BC)<A
        $NEXT

;   @           ( a -- w )
;               Push data at memory location to the data stack.

        $CODE   1,'@',AT
        DB 0A1H                             ;POP BC
        DB 29H                              ;A<(BC)
        DB 19H                              ;EAL<A
        DB 12H                              ;BC<BC+1
        DB 29H                          ;A<(BC)
        DB 18H                          ;EAH<A
        DB 0B4H                         ;PUSH EA
        $NEXT

;   C!          ( c b -- )
;               Pop the data stack to byte memory.

        $CODE   2,'C!',CSTOR
        DB 0A1H                         ;POP BC address
        DB 0A4H                         ;POP AE data
        DB 09H                          ;A<EAL
        DB 39H                          ;(BC)<A
        $NEXT

;   C@          ( b -- c )
;               Push byte memory location to the data stack.

        $CODE   2,'C@',CAT
        DB 0A1H                         ;POP BC
        DB 29H                          ;A<(BC)
        DB 6AH,0                        ;B<00
        DB 1BH                          ;C<A
        DB 0B1H                         ;PUSH BC
        $NEXT

;   RP@         ( -- a )
;               Push the current RP to the data stack.

        $CODE   3,'RP@',RPAT
        DB 0B3H                         ;PUSH HL
        $NEXT

;   RP!         ( a -- )
;               Set the return stack pointer.

        $CODE   COMPO+3,'RP!',RPSTO
        DB 0A3H                         ;POP HL
        $NEXT

;   R>          ( -- w )
;               Pop the return stack to the data stack.

        $CODE   2,'R>',RFROM
        DB 48H,85H                      ;EA<(HL)++
        DB 0B4H                         ;PUSH EA
        $NEXT
```

```
;   R@          ( -- w )
;               Copy top of return stack to the data stack.

      $CODE   2,'R@',RAT
      DB 48H,83H                        ;EA<(HL)
      DB 0B4H                           ;PUSH EA
      $NEXT

;   >R          ( w -- )
;               Push the data stack to the return stack.

      $CODE   COMPO+2,'>R',TOR
      DB 33H,33H                        ;HL<HL-2
      DB 0A4H                           ;POP EA
      DB 48H,93H                        ;(HL)<EA
      $NEXT

;   SP@         ( -- a )
;               Push the current data stack pointer.

      $CODE   3,'SP@',SPAT
      DB 70H,0EH,0FEH,0FFH              ;(FFFE)<SP
      DB 70H,1FH,0FEH,0FFH              ;BC<(FFFE)
      DB 0B1H                           ;PUSH BC
      $NEXT

;   SP!         ( a -- )
;               Set the data stack pointer.

      $CODE   3,'SP!',SPSTO
      DB 0A1H                           ;POP BC
      DB 70H,1EH,0FEH,0FFH              ;(FFFE)<BC
      DB 70H,0FH,0FEH,0FFH              ;PC<(FFFE)
      $NEXT

;   DROP        ( w -- )
;               Discard top stack item.

      $CODE   4,'DROP',DROP
      DB 0A4H                           ;POP EA
      $NEXT

;   DUP         ( w -- w w )
;               Duplicate the top stack item.

      $CODE   3,'DUP',DUPP
      DB 0A4H                           ;POP EA
      DB 0B4H                           ;PUSH EA
```

```
        DB 0B4H                                 ;PUSH EA
        $NEXT

;    SWAP          ( w1 w2 -- w2 w1 )
;                  Exchange top two stack items.

        $CODE   4,'SWAP',SWAP
        DB 0A4H                                 ;POP EA
        DB 0A1H                                 ;POP BC
        DB 0B4H                                 ;PUSH EA
        DB 0B1H                                 ;PUSH BC
        $NEXT

;    OVER          ( w1 w2 -- w1 w2 w1 )
;                  Copy second stack item to top.

        $CODE   4,'OVER',OVER
        DB 0A4H                                 ;POP AE
        DB 0A1H                                 ;POP BC
        DB 0B1H                                 ;PUSH BC
        DB 0B4H                                 ;PUSH AE
        DB 0B1H                                 ;PUSH BC
        $NEXT

;    0<            ( n -- t )
;                  Return true if n is negative.

        $CODE   2,'0<',ZLESS
        DB 0A1H                                 ;POP BC
        DB 69H,0FFH                             ;A<FF
        DB 48H,06H                              ;B Shift Left, Skip if carry
        DB 69H,0                                ;A<00
        DB 1AH                                  ;B<A
        DB 1BH                                  ;C<A
        DB 0B1H                                 ;PUSH BC
        $NEXT

;    AND           ( w w -- w )
;                  Bitwise AND.

        $CODE   3,'AND',ANDD
        DB 0A1H                                 ;POP BC
        DB 0A4H                                 ;POP AE
        DB 74H,8DH                              ;EA<EA AND BC
        DB 0B4H                                 ;PUSH EA
        $NEXT

;    OR            ( w w -- w )
;               Bitwise inclusive OR.

        $CODE   2,'OR',ORR
        DB 0A1H                         ;POP BC
        DB 0A4H                         ;POP EA
        DB 74H,9DH                      ;EA<EA OR BC
        DB 0B4H                          ;PUSH EA
        $NEXT

;   XOR         ( w w -- w )
;               Bitwise exclusive OR.

        $CODE   3,'XOR',XORR
        DB 0A1H                         ;POP BC
        DB 0A4H                         ;POP EA
        DB 74H,95H                      ;EA<EA EX-OR BC
        DB 0B4H                         ;PUSH EA
        $NEXT

;   UM+         ( w w -- w cy )
;               Add two numbers, return the sum and carry flag.

        $CODE   3,'UM+',UPLUS
        DB 0A1H                         ;POP BC
        DB 0A4H                         ;POP EA
        DB 69H,0                        ;A<00
        DB 74H,0A5H                     ;EA<EA+BC Skip if no carry
        DB 41H                          ;A<A+1
        DB 1BH                          ;C<A
        DB 6AH,0                        ;B<00
        DB 0B4H                         ;PUSH EA
        DB 0B1H                         ;PUSH BC
        $NEXT

;; System and user variables

;   doVAR       ( -- a )
;               Run time routine for VARIABLE and CREATE.

        $COLON  COMPO+5,'doVAR',DOVAR
        DW      RFROM,EXIT

;   UP          ( -- a )
;               Pointer to the user area.

        $COLON  2,'UP',UP
        DW      DOVAR
        DW      UPP
```

```
;   doUSER      ( -- a )
;               Run time routine for user variables.

      $COLON  COMPO+6,'doUSER',DOUSE
      DW      RFROM,AT,UP,AT,PLUS,EXIT

;   SP0         ( -- a )
;               Pointer to bottom of the data stack.

      $USER   3,'SP0',SZERO

;   RP0         ( -- a )
;               Pointer to bottom of the return stack.

      $USER   3,'RP0',RZERO

;   '?KEY       ( -- a )
;               Execution vector of ?KEY.

      $USER   5,"'?KEY",TQKEY

;   'EMIT       ( -- a )
;               Execution vector of EMIT.

      $USER   5,"'EMIT",TEMIT

;   'EXPECT     ( -- a )
;               Execution vector of EXPECT.

      $USER   7,"'EXPECT",TEXPE

;   'TAP        ( -- a )
;               Execution vector of TAP.

      $USER   4,"'TAP",TTAP

;   'ECHO       ( -- a )
;               Execution vector of ECHO.

      $USER   5,"'ECHO",TECHO

;   'PROMPT     ( -- a )
;               Execution vector of PROMPT.

      $USER   7,"'PROMPT",TPROM

;   BASE        ( -- a )
;               Storage of the radix base for numeric I/O.

        $USER   4,'BASE',BASE

;   tmp         ( -- a )
;               A temporary storage location used in parse and find.

        $USER   COMPO+3,'tmp',TEMP

;   SPAN        ( -- a )
;               Hold character count received by EXPECT.

        $USER   4,'SPAN',SPAN

;   >IN         ( -- a )
;               Hold the character pointer while parsing input stream.

        $USER   3,'>IN',INN

;   #TIB        ( -- a )
;               Hold the current count and address of the terminal input buffer.

        $USER   4,'#TIB',NTIB
        _USER = _USER+CELLL

;   CSP         ( -- a )
;               Hold the stack pointer for error checking.

        $USER   3,'CSP',CSP

;   'EVAL       ( -- a )
;               Execution vector of EVAL.

        $USER   5,"'EVAL",TEVAL

;   'NUMBER     ( -- a )
;               Execution vector of NUMBER?.

        $USER   7,"'NUMBER",TNUMB

;   HLD         ( -- a )
;               Hold a pointer in building a numeric output string.

        $USER   3,'HLD',HLD

;   HANDLER     ( -- a )
;               Hold the return stack pointer for error handling.
```

```
        $USER   7,'HANDLER',HANDL

;   CONTEXT     ( -- a )
;               A area to specify vocabulary search order.

        $USER   7,'CONTEXT',CNTXT
        _USER = _USER+VOCSS*CELLL       ;vocabulary stack

;   CURRENT     ( -- a )
;               Point to the vocabulary to be extended.

        $USER   7,'CURRENT',CRRNT
        _USER = _USER+CELLL             ;vocabulary link pointer

;   FHEAD       ( -- a )
;               Point to the FORTH vocab head pointer.
        $USER   5,'FHEAD',FHEAD

;   FLINK       ( -- a )
;               Point to the FORTH vocab link pointer.
        $USER   5,'FLINK',FLINK

;   CP          ( -- a )
;               Point to the top of the code dictionary.

        $USER   2,'CP',CP

;   NP          ( -- a )
;               Point to the bottom of the name dictionary.

        $USER   2,'NP',NP

;   LAST        ( -- a )
;               Point to the last name in the name dictionary.

        $USER   4,'LAST',LAST

;   SERIN       ( -- a )
;               Point to host serial input. Flag in high, char in low byte.

        $USER   5,'SERIN',SERIN

;   HAFBIT      ( -- a )
;               Point to half bit time used by serial i/0 routines.

        $USER   6,'HAFBIT',HAFBIT

;   BITIME      ( -- a )
;               Point to bit time used to set serial i/o baud rate.

          $USER   6,'BITIME',BITIME

;; Common functions

;   doVOC       ( -- )
;               Run time action of VOCABULARY's.

          $COLON  COMPO+5,'doVOC',DOVOC
          DW      FHEAD,CNTXT,STORE,EXIT

;   FORTH       ( -- )
;               Make FORTH the context vocabulary.

          $COLON  5,'FORTH',FORTH
          DW      DOVOC,EXIT
          ; Head and Link pointers normally here were moved to User Ram.

;   ?DUP        ( w -- w w | 0 )
;               Dup tos if its is not zero.

          $CODE  4,'?DUP',QDUP
          DB 6AH,0FFH            ;B<FF
          DB 6BH,0FFH            ;C<FF
          DB 0A4H                ;POP EA
          DB 74H,0DDH            ;EA AND BC, Skip if zero
          DB 0B4H                ;PUSH EA
          DB 0B4H                ;PUSH EA
          $NEXT

;   ROT         ( w1 w2 w3 -- w2 w3 w1 )
;               Rot 3rd item to top.

          $COLON  3,'ROT',ROT
          DW      TOR,SWAP,RFROM,SWAP,EXIT

;   2DROP       ( w w -- )
;               Discard two items on stack.

          $CODE  5,'2DROP',DDROP
          DB 0A4H,0A4H           ;POP EA, POP EA
          $NEXT

;   2DUP        ( w1 w2 -- w1 w2 w1 w2 )
;               Duplicate top two items.

          $CODE  4,'2DUP',DDUP
```

```
        DB 0A4H,0A1H            ;POP EA, POP BC
        DB 0B1H,0B4H            ;PUSH BC, PUSH EA
        DB 0B1H,0B4H            ;PUSH BC, PUSH EA
        $NEXT

;   +           ( w w -- sum )
;               Add top two items.

        $CODE  1,'+',PLUS
        DB 0A1H,0A4H            ;POP BC, POP EA
        DB 74H,0A5H             ;EA<EA+BC, Skip
        DB 0                    ;NOP
        DB 0B4H                 ;PUSH EA
        $NEXT

;   D+          ( d d -- d )
;               Double addition, as an example using UM+.
;
;               $COLON  2,'D+',DPLUS
;               DW      TOR,SWAP,TOR,UPLUS
;               DW      RFROM,RFROM,PLUS,PLUS,EXIT

;   NOT         ( w -- w )
;               One's complement of tos.

        $CODE  3,'NOT',INVER
        DB 0A1H                 ;POP BC
        DB 69H,0FFH             ;A<FF
        DB 60H,12H              ;B<B EX-OR A
        DB 60H,13H              ;C<C EX-OR A
        DB 0B1H                 ;PUSH BC
        $NEXT

;   NEGATE      ( n -- -n )
;               Two's complement of tos.

        $CODE  6,'NEGATE',NEGAT
        DB 0A1H                 ;POP BC
        DB 69H,0FFH             ;A<FF
        DB 60H,12H              ;B<B EX-OR A
        DB 60H,13H              ;C<C EX-OR A
        DB 12H                  ;BC<BC+1
        DB 0B1H                 ;PUSH BC
        $NEXT


;   DNEGATE     ( d -- -d )
;               Two's complement of top double.
```

```
          $COLON  7,'DNEGATE',DNEGA
          DW      INVER,TOR,INVER
          DW      DOLIT,1,UPLUS
          DW      RFROM,PLUS,EXIT

;   -           ( n1 n2 -- n1-n2 )
;               Subtraction.

          $CODE  1,'-',SUBB
          DB 0A1H                ;POP BC
          DB 069H,0FFH           ;A<FF
          DB 060H,12H            ;B<B EX-OR A
          DB 060H,13H            ;C<C EX-OR A
          DB 12H                 ;BC<BC+1
          DB 0A4H                ;POP EA
          DB 74H,0A5H            ;EA<EA+BC Skip
          DB 0                   ;NOP
          DB 0B4H                ;PUSH EA
          $NEXT

;   ABS         ( n -- n )
;               Return the absolute value of n.

          $COLON  3,'ABS',ABSS
          DW      DUPP,ZLESS
          DW      QBRAN,ABS1
          DW      NEGAT
ABS1:           DW      EXIT

;   =           ( w w -- t )
;               Return true if top two are equal.

          $CODE  1,'=',EQUAL
          DB 0A4H,0A1H           ;POP EA, POP BC
          DB 69H,0FFH            ;A<FF
          DB 74H,0FDH            ;EA-BC, Skip if zero
          DB 69H,00H             ;A<00
          DB 1AH,1BH             ;B<A, C<A
          DB 0B1H                ;PUSH BC
          $NEXT

;   U<          ( u u -- t )
;               Unsigned compare of top two items.

          $COLON  2,'U<',ULESS
          DW      DDUP,XORR,ZLESS
          DW      QBRAN,ULES1
          DW      SWAP,DROP,ZLESS,EXIT
ULES1:          DW      SUBB,ZLESS,EXIT

;   <           ( n1 n2 -- t )
;               Signed compare of top two items.

          $COLON  1,'<',LESS
          DW      DDUP,XORR,ZLESS
          DW      QBRAN,LESS1
          DW      DROP,ZLESS,EXIT
LESS1:          DW      SUBB,ZLESS,EXIT

;   MAX         ( n n -- n )
;               Return the greater of two top stack items.

          $CODE  3,'MAX',MAX
          DB 0A4H,0A1H              ;POP EA, POP BC
          DB 74H,0BDH               ;EA-BC, Skip if borrow
          DB 0C2H                   ;Jump to Push EA
          DB 0B1H                   ;PUSH BC
          DB 0C1H                   ;Jump to next
          DB 0B4H                   ;PUSH EA
          $NEXT

;   MIN         ( n n -- n )
;               Return the smaller of top two stack items.

          $CODE  3,'MIN',MIN
          DB 0A4H,0A1H              ;POP EA, POP BC
          DB 74H,0BDH               ;EA-BC, Skip if borrow
          DB 0C2H                   ;Jump to Push EA
          DB 0B4H                   ;PUSH EA
          DB 0C1H                   ;Jump to next
          DB 0B1H                   ;PUSH BC
          $NEXT

;   WITHIN      ( u ul uh -- t )
;               Return true if u is within the range of ul and uh.

          $COLON  6,'WITHIN',WITHI
          DW      OVER,SUBB,TOR                 ;ul <= u < uh
          DW      SUBB,RFROM,ULESS,EXIT

;; Quick Operators

;
```

```
;   1+          ( n -- n+1 )
        $CODE 2,'1+',ONEP
        DB 0A1H                 ;POP BC
        DB 12H                  ;BC<BC+1
        DB 0B1H                 ;PUSH BC
        $NEXT

;   1-          ( n -- n-1 )
        $CODE 2,'1-',ONEM
        DB 0A1H                 ;POP BC
        DB 013H                 ;BC<BC-1
        DB 0B1H                 ;PUSH BC
        $NEXT

;   2+          ( n -- n+2 )
        $CODE 2,'2+',TWOP
        DB 0A1H                 ;POP BC
        DB 12H,12H              ;BC<BC+2
        DB 0B1H                 ;PUSH BC
        $NEXT

;   2-          ( n -- n-2 )
        $CODE 2,'2-',TWOM
        DB 0A1H                 ;POP BC
        DB 13H,13H              ;BC<BC-2
        DB 0B1H                 ;PUSH BC
        $NEXT

;   2*          ( n -- n*2 )
        $CODE 2,'2*',TWOSL
        DB 0A4H                 ;POP EA
        DB 48H,0A4H             ;EA Logical Shift Left
        DB 0B4H                 ;PUSH EA
        $NEXT

;   2/          ( n -- n/2 )
        $CODE 2,'2/',TWOSR
        DB 0A4H                 ;POP EA
        DB 48H,0A0H             ;EA Logical Shift Right
        DB 0B4H                 ;PUSH EA
        $NEXT

;; Divide

;   UM/MOD      ( udl udh u -- ur uq )
;               Unsigned divide of a double by a single. Return mod and
quotient.

          $COLON  6,'UM/MOD',UMMOD
          DW      DDUP,ULESS
          DW      QBRAN,UMM4
          DW      NEGAT,DOLIT,15,TOR
UMM1:           DW      TOR,DUPP,UPLUS
          DW      TOR,TOR,DUPP,UPLUS
          DW      RFROM,PLUS,DUPP
          DW      RFROM,RAT,SWAP,TOR
          DW      UPLUS,RFROM,ORR
          DW      QBRAN,UMM2
          DW      TOR,DROP,ONEP,RFROM
          DW      BRAN,UMM3
UMM2:           DW      DROP
UMM3:           DW      RFROM
          DW      DONXT,UMM1
          DW      DROP,SWAP,EXIT
UMM4:           DW      DROP,DDROP
          DW      DOLIT,-1,DUPP,EXIT      ;overflow, return max

;   M/MOD       ( d n -- r q )
;               Signed floored divide of double by single. Return mod and
quotient.

          $COLON  5,'M/MOD',MSMOD
          DW      DUPP,ZLESS,DUPP,TOR
          DW      QBRAN,MMOD1
          DW      NEGAT,TOR,DNEGA,RFROM
MMOD1:          DW      TOR,DUPP,ZLESS
          DW      QBRAN,MMOD2
          DW      RAT,PLUS
MMOD2:          DW      RFROM,UMMOD,RFROM
          DW      QBRAN,MMOD3
          DW      SWAP,NEGAT,SWAP
MMOD3:          DW      EXIT

;   /MOD        ( n n -- r q )
;               Signed divide. Return mod and quotient.

          $COLON  4,'/MOD',SLMOD
          DW      OVER,ZLESS,SWAP,MSMOD,EXIT

;   MOD         ( n n -- r )
;               Signed divide. Return mod only.

          $COLON  3,'MOD',MODD
          DW      SLMOD,DROP,EXIT

;   /           ( n n -- q )
;               Signed divide. Return quotient only.

          $COLON  1,'/',SLASH
          DW      SLMOD,SWAP,DROP,EXIT

;; Multiply

;   UM*         ( u u -- ud )
;               Unsigned multiply. Return double product.

          $COLON  3,'UM*',UMSTA
          DW      DOLIT,0,SWAP,DOLIT,15,TOR
UMST1:          DW      DUPP,UPLUS,TOR,TOR
          DW      DUPP,UPLUS,RFROM,PLUS,RFROM
          DW      QBRAN,UMST2
          DW      TOR,OVER,UPLUS,RFROM,PLUS
UMST2:          DW      DONXT,UMST1
          DW      ROT,DROP,EXIT

;   *           ( n n -- n )
;               Signed multiply. Return single product.

          $COLON  1,'*',STAR
          DW      UMSTA,DROP,EXIT

;   M*          ( n n -- d )
;               Signed multiply. Return double product.

          $COLON  2,'M*',MSTAR
          DW      DDUP,XORR,ZLESS,TOR
          DW      ABSS,SWAP,ABSS,UMSTA
          DW      RFROM
          DW      QBRAN,MSTA1
          DW      DNEGA
MSTA1:          DW      EXIT

;   */MOD       ( n1 n2 n3 -- r q )
;               Multiply n1 and n2, then divide by n3. Return mod and quotient.

          $COLON  5,'*/MOD',SSMOD
          DW      TOR,MSTAR,RFROM,MSMOD,EXIT

;   */          ( n1 n2 n3 -- q )
;               Multiply n1 by n2, then divide by n3. Return quotient only.
```

```
          $COLON  2,'*/',STASL
          DW      SSMOD,SWAP,DROP,EXIT

;; Miscellaneous

;   BL          ( -- 32 )
;               Return 32, the blank character.

          $COLON  2,'BL',BLANK
          DW      DOLIT,' ',EXIT

;   >CHAR       ( c -- c )
;               Filter non-printing characters.

          $COLON  5,'>CHAR',TCHAR
          DW      DOLIT,07FH,ANDD,DUPP    ;mask msb
          DW      DOLIT,127,BLANK,WITHI   ;check for printable
          DW      QBRAN,TCHA1
          DW      DROP,DOLIT,'_'          ;replace non-printables
TCHA1:          DW      EXIT

;   DEPTH       ( -- n )
;               Return the depth of the data stack.

          $COLON  5,'DEPTH',DEPTH
          DW      SPAT,SZERO,AT,SWAP,SUBB
          DW      DOLIT,CELLL,SLASH,EXIT

;   PICK        ( ... +n -- ... w )
;               Copy the nth stack item to tos.

          $COLON  4,'PICK',PICK
          DW      ONEP,TWOSL
          DW      SPAT,PLUS,AT,EXIT

;; Memory access

;   +!          ( n a -- )
;               Add n to the contents at address a.

          $COLON  2,'+!',PSTOR
          DW      SWAP,OVER,AT,PLUS
          DW      SWAP,STORE,EXIT

;   2!          ( d a -- )
;               Store the double integer to address a.

          $COLON  2,'2!',DSTOR
```

```
        DW      SWAP,OVER,STORE
        DW      TWOP,STORE,EXIT

;   2@          ( a -- d )
;               Fetch double integer from address a.

        $COLON  2,'2@',DAT
        DW      DUPP,TWOP,AT
        DW      SWAP,AT,EXIT

;   COUNT       ( b -- b +n )
;               Return count byte of a string and add 1 to byte address.

        $COLON  5,'COUNT',COUNT
        DW      DUPP,ONEP
        DW      SWAP,CAT,EXIT

;   HERE        ( -- a )
;               Return the top of the code dictionary.

        $COLON  4,'HERE',HERE
        DW      CP,AT,EXIT

;   PAD         ( -- a )
;               Return the address of a temporary buffer.

        $COLON  3,'PAD',PAD
        DW      DOLIT,PADD,EXIT

;   TIB         ( -- a )
;               Return the address of the terminal input buffer.

        $COLON  3,'TIB',TIB
        DW      NTIB,TWOP,AT,EXIT

;   @EXECUTE    ( a -- )
;               Execute vector stored in address a.

        $COLON  8,'@EXECUTE',ATEXE
        DW      AT,QDUP                 ;?address or zero
        DW      QBRAN,EXE1
        DW      EXECU                   ;execute if non-zero
EXE1:           DW      EXIT                    ;do nothing if zero

;   CMOVE       ( b1 b2 u -- )
;               Copy u bytes from b1 to b2.

        $COLON  5,'CMOVE',CMOVE
```

```
          DW      TOR
          DW      BRAN,CMOV2
CMOV1:          DW      TOR,DUPP,CAT
          DW      RAT,CSTOR
          DW      ONEP
          DW      RFROM,ONEP
CMOV2:          DW      DONXT,CMOV1
          DW      DDROP,EXIT

;   FILL        ( b u c -- )
;               Fill u bytes of character c to area beginning at b.

          $COLON  4,'FILL',FILL
          DW      SWAP,TOR,SWAP
          DW      BRAN,FILL2
FILL1:          DW      DDUP,CSTOR,ONEP
FILL2:          DW      DONXT,FILL1
          DW      DDROP,EXIT

;   -TRAILING   ( b u -- b u )
;               Adjust the count to eliminate trailing white space.

          $COLON  9,'-TRAILING',DTRAI
          DW      TOR
          DW      BRAN,DTRA2
DTRA1:          DW      BLANK,OVER,RAT,PLUS,CAT,LESS
          DW      QBRAN,DTRA2
          DW      RFROM,ONEP,EXIT           ;adjusted count
DTRA2:          DW      DONXT,DTRA1
          DW      DOLIT,0,EXIT            ;count=0

;   PACK$       ( b u a -- a )
;               Build a counted string with u characters from b. Null fill.

          $COLON  5,'PACK$',PACKS
          DW      DUPP,TOR          ;strings only on cell boundary
          DW      OVER,DUPP,DOLIT,0
          DW      DOLIT,CELLL,UMMOD,DROP  ;count mod cell
          DW      SUBB,OVER,PLUS
          DW      DOLIT,0,SWAP,STORE      ;null fill cell
          DW      DDUP,CSTOR,ONEP           ;save count
          DW      SWAP,CMOVE,RFROM,EXIT   ;move string

;; Numeric output, single precision

;   DIGIT       ( u -- c )
;               Convert digit u to a character.
```

```
          $COLON  5,'DIGIT',DIGIT
          DW      DOLIT,9,OVER,LESS
          DW      DOLIT,7,ANDD,PLUS
          DW      DOLIT,'0',PLUS,EXIT

;   EXTRACT     ( n base -- n c )
;               Extract the least significant digit from n.

          $COLON  7,'EXTRACT',EXTRC
          DW      DOLIT,0,SWAP,UMMOD
          DW      SWAP,DIGIT,EXIT

;   <#          ( -- )
;               Initiate the numeric output process.

          $COLON  2,'<#',BDIGS
          DW      PAD,HLD,STORE,EXIT

;   HOLD        ( c -- )
;               Insert a character into the numeric output string.

          $COLON  4,'HOLD',HOLD
          DW      HLD,AT,ONEM
          DW      DUPP,HLD,STORE,CSTOR,EXIT

;   #           ( u -- u )
;               Extract one digit from u and append the digit to output string.

          $COLON  1,'#',DIG
          DW      BASE,AT,EXTRC,HOLD,EXIT

;   #S          ( u -- 0 )
;               Convert u until all digits are added to the output string.

          $COLON  2,'#S',DIGS
DIGS1:          DW      DIG,DUPP
          DW      QBRAN,DIGS2
          DW      BRAN,DIGS1
DIGS2:          DW      EXIT

;   SIGN        ( n -- )
;               Add a minus sign to the numeric output string.

          $COLON  4,'SIGN',SIGN
          DW      ZLESS
          DW      QBRAN,SIGN1
          DW      DOLIT,'-',HOLD
SIGN1:          DW      EXIT
```

```
;   #>          ( w -- b u )
;               Prepare the output string to be TYPE'd.

          $COLON  2,'#>',EDIGS
          DW      DROP,HLD,AT
          DW      PAD,OVER,SUBB,EXIT

;   str         ( n -- b u )
;               Convert a signed integer to a numeric string.

          $COLON  3,'str',STR
          DW      DUPP,TOR,ABSS
          DW      BDIGS,DIGS,RFROM
          DW      SIGN,EDIGS,EXIT

;   HEX         ( -- )
;               Use radix 16 as base for numeric conversions.

          $COLON  3,'HEX',HEX
          DW      DOLIT,16,BASE,STORE,EXIT

;   DECIMAL     ( -- )
;               Use radix 10 as base for numeric conversions.

          $COLON  7,'DECIMAL',DECIM
          DW      DOLIT,10,BASE,STORE,EXIT

;; Numeric input, single precision

;   DIGIT?      ( c base -- u t )
;               Convert a character to its numeric value. A flag indicates
success.

          $COLON  6,'DIGIT?',DIGTQ
          DW      TOR,DOLIT,'0',SUBB
          DW      DOLIT,9,OVER,LESS
          DW      QBRAN,DGTQ1
          DW      DOLIT,7,SUBB
          DW      DUPP,DOLIT,10,LESS,ORR
DGTQ1:          DW      DUPP,RFROM,ULESS,EXIT

;   NUMBER?     ( a -- n T | a F )
;               Convert a number string to integer. Push a flag on tos.

          $COLON  7,'NUMBER?',NUMBQ
          DW      BASE,AT,TOR,DOLIT,0,OVER,COUNT
          DW      OVER,CAT,DOLIT,'$',EQUAL
```

```
        DW      QBRAN,NUMQ1
        DW      HEX,SWAP,ONEP
        DW      SWAP,ONEM
NUMQ1:          DW      OVER,CAT,DOLIT,'-',EQUAL,TOR
        DW      SWAP,RAT,SUBB,SWAP,RAT,PLUS,QDUP
        DW      QBRAN,NUMQ6
        DW      ONEM,TOR
NUMQ2:          DW      DUPP,TOR,CAT,BASE,AT,DIGTQ
        DW      QBRAN,NUMQ4
        DW      SWAP,BASE,AT,STAR,PLUS,RFROM
        DW      ONEP
        DW      DONXT,NUMQ2
        DW      RAT,SWAP,DROP
        DW      QBRAN,NUMQ3
        DW      NEGAT
NUMQ3:          DW      SWAP
        DW      BRAN,NUMQ5
NUMQ4:          DW      RFROM,RFROM,DDROP,DDROP,DOLIT,0
NUMQ5:          DW      DUPP
NUMQ6:          DW      RFROM,DDROP
        DW      RFROM,BASE,STORE,EXIT

;; Basic I/O

;   ?KEY        ( -- c T | F )
;               Return input character and true, or a false if no input.

        $COLON  4,'?KEY',QKEY
        DW      TQKEY,ATEXE,EXIT

;   KEY         ( -- c )
;               Wait for and return an input character.

        $COLON  3,'KEY',KEY
KEY1:           DW      QKEY
        DW      QBRAN,KEY1
        DW      EXIT

;   EMIT        ( c -- )
;               Send a character to the output device.

        $COLON  4,'EMIT',EMIT
        DW      TEMIT,ATEXE,EXIT

;   NUF?        ( -- t )
;               Return false if no input, else pause and if CR return true.

        $COLON  4,'NUF?',NUFQ
```

```
          DW      QKEY,DUPP
          DW      QBRAN,NUFQ1
          DW      DDROP,KEY,DOLIT,CRR,EQUAL
NUFQ1:          DW      EXIT

;   PACE        ( -- )
;               Send a pace character for the file downloading process.

          $COLON  4,'PACE',PACE
          DW      DOLIT,11,EMIT,EXIT

;   SPACE       ( -- )
;               Send the blank character to the output device.

          $COLON  5,'SPACE',SPACE
          DW      BLANK,EMIT,EXIT

;   SPACES      ( +n -- )
;               Send n spaces to the output device.

          $COLON  6,'SPACES',SPACS
          DW      DOLIT,0,MAX,TOR
          DW      BRAN,CHAR2
CHAR1:          DW      SPACE
CHAR2:          DW      DONXT,CHAR1
          DW      EXIT

;   TYPE        ( b u -- )
;               Output u characters from b.

          $COLON  4,'TYPE',TYPEE
          DW      TOR
          DW      BRAN,TYPE2
TYPE1:          DW      DUPP,CAT,EMIT
          DW      ONEP
TYPE2:          DW      DONXT,TYPE1
          DW      DROP,EXIT

;   CR          ( -- )
;               Output a carriage return and a line feed.

          $COLON  2,'CR',CR
          DW      DOLIT,CRR,EMIT
          DW      DOLIT,LF,EMIT,EXIT

;   do$         ( -- a )
;               Return the address of a compiled string.
```

```
          $COLON  COMPO+3,'do$',DOSTR
          DW      RFROM,RAT,RFROM,COUNT,PLUS
          DW      TOR,SWAP,TOR,EXIT

;   $"|         ( -- a )
;                Run time routine compiled by $". Return address of a compiled
string.

          $COLON  COMPO+3,'$"|',STRQP
          DW      DOSTR,EXIT                ;force a call to do$

;   ."|         ( -- )
;                Run time routine of ." . Output a compiled string.

          $COLON  COMPO+3,'."|',DOTQP
          DW      DOSTR,COUNT,TYPEE,EXIT

;   .R           ( n +n -- )
;                Display an integer in a field of n columns, right justified.

          $COLON  2,'.R',DOTR
          DW      TOR,STR,RFROM,OVER,SUBB
          DW      SPACS,TYPEE,EXIT

;   U.R          ( u +n -- )
;                Display an unsigned integer in n column, right justified.

          $COLON  3,'U.R',UDOTR
          DW      TOR,BDIGS,DIGS,EDIGS
          DW      RFROM,OVER,SUBB
          DW      SPACS,TYPEE,EXIT

;   U.           ( u -- )
;                Display an unsigned integer in free format.

          $COLON  2,'U.',UDOT
          DW      BDIGS,DIGS,EDIGS
          DW      SPACE,TYPEE,EXIT

;   .            ( w -- )
;                Display an integer in free format, preceeded by a space.

          $COLON  1,'.',DOT
          DW      BASE,AT,DOLIT,10,XORR     ;?decimal
          DW      QBRAN,DOT1
          DW      UDOT,EXIT                 ;no, display unsigned
DOT1:            DW      STR,SPACE,TYPEE,EXIT    ;yes, display signed
```

```
;   ?           ( a -- )
;               Display the contents in a memory cell.

          $COLON  1,'?',QUEST
          DW      AT,DOT,EXIT

;; Parsing

;   parse       ( b u c -- b u delta ; <string> )
;               Scan string delimited by c. Return found string and its offset.

          $COLON  5,'parse',PARS
          DW      TEMP,STORE,OVER,TOR,DUPP
          DW      QBRAN,PARS8
          DW      ONEM,TEMP,AT,BLANK,EQUAL
          DW      QBRAN,PARS3
          DW      TOR
PARS1:          DW      BLANK,OVER,CAT          ;skip leading blanks ONLY
          DW      SUBB,ZLESS,INVER
          DW      QBRAN,PARS2
          DW      ONEP
          DW      DONXT,PARS1
          DW      RFROM,DROP,DOLIT,0,DUPP,EXIT
PARS2:          DW      RFROM
PARS3:          DW      OVER,SWAP
          DW      TOR
PARS4:          DW      TEMP,AT,OVER,CAT,SUBB   ;scan for delimiter
          DW      TEMP,AT,BLANK,EQUAL
          DW      QBRAN,PARS5
          DW      ZLESS
PARS5:          DW      QBRAN,PARS6
          DW      ONEP
          DW      DONXT,PARS4
          DW      DUPP,TOR
          DW      BRAN,PARS7
PARS6:          DW      RFROM,DROP,DUPP
          DW      ONEP,TOR
PARS7:          DW      OVER,SUBB
          DW      RFROM,RFROM,SUBB,EXIT
PARS8:          DW      OVER,RFROM,SUBB,EXIT

;   PARSE       ( c -- b u ; <string> )
;               Scan input stream and return counted string delimited by c.

          $COLON  5,'PARSE',PARSE
          DW      TOR,TIB,INN,AT,PLUS     ;current input buffer pointer
          DW      NTIB,AT,INN,AT,SUBB     ;remaining count
          DW      RFROM,PARS,INN,PSTOR,EXIT
```

```
;   .(          ( -- )
;               Output following string up to next ) .

        $COLON  IMEDD+2,'.(',DOTPR
        DW      DOLIT,')',PARSE,TYPEE,EXIT

;   (           ( -- )
;               Ignore following string up to next ) . A comment.

        $COLON  IMEDD+1,'(',PAREN
        DW      DOLIT,')',PARSE,DDROP,EXIT

;   \           ( -- )
;               Ignore following text till the end of line.

        $COLON  IMEDD+1,'\',BKSLA
        DW      NTIB,AT,INN,STORE,EXIT

;   CHAR        ( -- c )
;               Parse next word and return its first character.

        $COLON  4,'CHAR',CHAR
        DW      BLANK,PARSE,DROP,CAT,EXIT

;   TOKEN       ( -- a ; <string> )
;               Parse a word from input stream and copy it to name dictionary.

        $COLON  5,'TOKEN',TOKEN
        DW      BLANK,PARSE,DOLIT,31,MIN
        DW      NP,AT,OVER,SUBB,TWOM
        DW      PACKS,EXIT

;   WORD        ( c -- a ; <string> )
;               Parse a word from input stream and copy it to code dictionary.

        $COLON  4,'WORD',WORDD
        DW      PARSE,HERE,PACKS,EXIT

;; Dictionary search

;   NAME>       ( na -- ca )
;               Return a code address given a name address.

        $COLON  5,'NAME>',NAMET
        DW      TWOM,TWOM,AT,EXIT

;   SAME?       ( a a u -- a a f \ -0+ )
;               Compare u cells in two strings. Return 0 if identical.

          $COLON  5,'SAME?',SAMEQ
          DW      TOR
          DW      BRAN,SAME2
SAME1:          DW      OVER,RAT,TWOSL,PLUS,AT
          DW      OVER,RAT,TWOSL,PLUS,AT
          DW      SUBB,QDUP
          DW      QBRAN,SAME2
          DW      RFROM,DROP,EXIT         ;strings not equal
SAME2:          DW      DONXT,SAME1
          DW      DOLIT,0,EXIT            ;strings equal

;   find        ( a va -- ca na | a F )
;               Search a vocabulary for a string. Return ca and na if succeeded.

          $COLON  4,'find',FIND
          DW      SWAP,DUPP,CAT
          DW      DOLIT,CELLL,SLASH,TEMP,STORE
          DW      DUPP,AT,TOR,TWOP,SWAP
FIND1:          DW      AT,DUPP
          DW      QBRAN,FIND6
          DW      DUPP,AT,DOLIT,MASKK,ANDD,RAT,XORR
          DW      QBRAN,FIND2
          DW      TWOP,DOLIT,-1          ;true flag
          DW      BRAN,FIND3
FIND2:          DW      TWOP,TEMP,AT,SAMEQ
FIND3:          DW      BRAN,FIND4
FIND6:          DW      RFROM,DROP
          DW      SWAP,TWOM,SWAP,EXIT
FIND4:          DW      QBRAN,FIND5
          DW      TWOM,TWOM
          DW      BRAN,FIND1
FIND5:          DW      RFROM,DROP,SWAP,DROP
          DW      TWOM
          DW      DUPP,NAMET,SWAP,EXIT

;   NAME?       ( a -- ca na | a F )
;               Search all context vocabularies for a string.

          $COLON  5,'NAME?',NAMEQ
          DW      CNTXT,DUPP,DAT,XORR     ;?context=also
          DW      QBRAN,NAMQ1
          DW      TWOM                   ;no, start with context
NAMQ1:          DW      TOR
NAMQ2:          DW      RFROM,TWOP,DUPP,TOR    ;next in search order
          DW      AT,QDUP
          DW      QBRAN,NAMQ3
          DW      FIND,QDUP               ;search vocabulary
          DW      QBRAN,NAMQ2
          DW      RFROM,DROP,EXIT         ;found name
NAMQ3:          DW      RFROM,DROP              ;name not found
          DW      DOLIT,0,EXIT            ;false flag

;; Terminal response

;   ^H          ( bot eot cur -- bot eot cur )
;               Backup the cursor by one character.

          $COLON  2,'^H',BKSP
          DW      TOR,OVER,RFROM,SWAP,OVER,XORR
          DW      QBRAN,BACK1
          DW      DOLIT,BKSPP,TECHO,ATEXE,ONEM
          DW      BLANK,TECHO,ATEXE
          DW      DOLIT,BKSPP,TECHO,ATEXE
BACK1:          DW      EXIT

;   TAP         ( bot eot cur c -- bot eot cur )
;               Accept and echo the key stroke and bump the cursor.

          $COLON  3,'TAP',TAP
          DW      DUPP,TECHO,ATEXE
          DW      OVER,CSTOR,ONEP,EXIT

;   kTAP        ( bot eot cur c -- bot eot cur )
;               Process a key stroke, CR or backspace.

          $COLON  4,'kTAP',KTAP
          DW      DUPP,DOLIT,CRR,XORR
          DW      QBRAN,KTAP2
          DW      DOLIT,BKSPP,XORR
          DW      QBRAN,KTAP1
          DW      BLANK,TAP,EXIT
KTAP1:          DW      BKSP,EXIT
KTAP2:          DW      DROP,SWAP,DROP,DUPP,EXIT

;   accept      ( b u -- b u )
;               Accept characters to input buffer. Return with actual count.

          $COLON  6,'accept',ACCEP
          DW      OVER,PLUS,OVER
ACCP1:          DW      DDUP,XORR
          DW      QBRAN,ACCP4
          DW      KEY,DUPP
;               DW      BLANK,SUBB,DOLIT,95,ULESS
          DW      BLANK,DOLIT,127,WITHI
```

```
        DW      QBRAN,ACCP2
        DW      TAP
        DW      BRAN,ACCP3
ACCP2:          DW      TTAP,ATEXE
ACCP3:          DW      BRAN,ACCP1
ACCP4:          DW      DROP,OVER,SUBB,EXIT

;   EXPECT      ( b u -- )
;               Accept input stream and store count in SPAN.

        $COLON  6,'EXPECT',EXPEC
        DW      TEXPE,ATEXE,SPAN,STORE,DROP,EXIT

;   QUERY       ( -- )
;               Accept input stream to terminal input buffer.

        $COLON  5,'QUERY',QUERY
        DW      TIB,DOLIT,80,TEXPE,ATEXE,NTIB,STORE
        DW      DROP,DOLIT,0,INN,STORE,EXIT

;; Error handling

;   CATCH       ( ca -- 0 | err# )
;               Execute word at ca and set up an error frame for it.

        $COLON  5,'CATCH',CATCH
        DW      SPAT,TOR,HANDL,AT,TOR   ;save error frame
        DW      RPAT,HANDL,STORE,EXECU  ;execute
        DW      RFROM,HANDL,STORE       ;restore error frame
        DW      RFROM,DROP,DOLIT,0,EXIT ;no error

;   THROW       ( err# -- err# )
;               Reset system to current local error frame an update error flag.

        $COLON  5,'THROW',THROW
        DW      HANDL,AT,RPSTO          ;restore return stack
        DW      RFROM,HANDL,STORE       ;restore handler frame
        DW      RFROM,SWAP,TOR,SPSTO    ;restore data stack
        DW      DROP,RFROM,EXIT

;   NULL$       ( -- a )
;               Return address of a null string with zero count.

        $COLON  5,'NULL$',NULLS
        DW      DOVAR                   ;emulate CREATE
        DW      0
        DB      99,111,121,111,116,101
```

```
;   ABORT       ( -- )
;               Reset data stack and jump to QUIT.

          $COLON  5,'ABORT',ABORT
          DW      NULLS,THROW

;   abort"     ( f -- )
;               Run time routine of ABORT" . Abort with a message.

          $COLON  COMPO+6,'abort"',ABORQ
          DW      QBRAN,ABOR1                 ;text flag
          DW      DOSTR,THROW                 ;pass error string
ABOR1:          DW      DOSTR,DROP,EXIT             ;drop error

;; The text interpreter

;   $INTERPRET  ( a -- )
;               Interpret a word. If failed, try to convert it to an integer.

          $COLON  10,'$INTERPRET',INTER
          DW      NAMEQ,QDUP                  ;?defined
          DW      QBRAN,INTE1
          DW      AT,DOLIT,COMPO,ANDD         ;?compile only lexicon bits
          D$      ABORQ,' compile only'
          DW      EXECU,EXIT                  ;execute defined word
INTE1:          DW      TNUMB,ATEXE                 ;convert a number
          DW      QBRAN,INTE2
          DW      EXIT
INTE2:          DW      THROW                       ;error

;   [           ( -- )
;               Start the text interpreter.

          $COLON  IMEDD+1,'[',LBRAC
          DW      DOLIT,INTER,TEVAL,STORE,EXIT

;   .OK         ( -- )
;               Display 'ok' only while interpreting.

          $COLON  3,'.OK',DOTOK
          DW      DOLIT,INTER,TEVAL,AT,EQUAL
          DW      QBRAN,DOTO1
          D$      DOTQP,' ok'
DOTO1:          DW      CR,EXIT

;   ?STACK      ( -- )
;               Abort if the data stack underflows.
```

```
          $COLON  6,'?STACK',QSTAC
          DW      DEPTH,ZLESS               ;check only for underflow
          D$      ABORQ,' underflow'
          DW      EXIT

;   EVAL         ( -- )
;                Interpret the input stream.

          $COLON  4,'EVAL',EVAL
EVAL1:           DW      TOKEN,DUPP,CAT           ;?input stream empty
          DW      QBRAN,EVAL2
          DW      TEVAL,ATEXE,QSTAC         ;evaluate input, check stack
          DW      BRAN,EVAL1
EVAL2:           DW      DROP,TPROM,ATEXE,EXIT   ;prompt

;; Shell

;   PRESET       ( -- )
;                Reset data stack pointer and the terminal input buffer.

          $COLON  6,'PRESET',PRESE
          DW      SZERO,AT,SPSTO
          DW      DOLIT,TIBB,NTIB,TWOP,STORE,EXIT

;   xio          ( a a a -- )
;                Reset the I/O vectors 'EXPECT, 'TAP, 'ECHO and 'PROMPT.

          $COLON  COMPO+3,'xio',XIO
          DW      DOLIT,ACCEP,TEXPE,DSTOR
          DW      TECHO,DSTOR,EXIT

;   FILE         ( -- )
;                Select I/O vectors for file download.

          $COLON  4,'FILE',FILE
          DW      DOLIT,PACE,DOLIT,DROP
          DW      DOLIT,KTAP,XIO,EXIT

;   HAND         ( -- )
;                Select I/O vectors for terminal interface.

          $COLON  4,'HAND',HAND
          DW      DOLIT,DOTOK,DOLIT,EMIT
          DW      DOLIT,KTAP,XIO,EXIT

;   I/O          ( -- a )
;                Array to store default I/O vectors.
```

```
          $COLON  3,'I/O',ISLO
          DW      DOVAR                       ;emulate CREATE
          DW      QRX,TXSTO                   ;default I/O vectors

;   CONSOLE     ( -- )
;               Initiate terminal interface.

          $COLON  7,'CONSOLE',CONSO
          DW      ISLO,DAT,TQKEY,DSTOR        ;restore default I/O device
          DW      HAND,EXIT                   ;keyboard input

;   QUIT        ( -- )
;               Reset return stack pointer and start text interpreter.

          $COLON  4,'QUIT',QUIT
          DW      RZERO,AT,RPSTO              ;reset return stack pointer
QUIT1:          DW      LBRAC                       ;start interpretation
QUIT2:          DW      QUERY                       ;get input
          DW      DOLIT,EVAL,CATCH,QDUP       ;evaluate input
          DW      QBRAN,QUIT2                 ;continue till error
          DW      TPROM,AT,SWAP               ;save input device
          DW      CONSO,NULLS,OVER,XORR       ;?display error message
          DW      QBRAN,QUIT3
          DW      SPACE,COUNT,TYPEE           ;error message
          D$      DOTQP,' ? '                 ;error prompt
QUIT3:          DW      DOLIT,DOTOK,XORR            ;?file input
          DW      QBRAN,QUIT4
          DW      DOLIT,ERR,EMIT              ;file error, tell host
QUIT4:          DW      PRESE                       ;some cleanup
          DW      BRAN,QUIT1

;; The compiler

;   '           ( -- ca )
;               Search context vocabularies for the next word in input stream.

          $COLON  1,"'",TICK
          DW      TOKEN,NAMEQ                 ;?defined
          DW      QBRAN,TICK1
          DW      EXIT                        ;yes, push code address
TICK1:          DW      THROW                       ;no, error

;   ALLOT       ( n -- )
;               Allocate n bytes to the code dictionary.

          $COLON  5,'ALLOT',ALLOT
          DW      CP,PSTOR,EXIT               ;adjust code pointer
```

```
;   ,           ( w -- )
;               Compile an integer into the code dictionary.

        $COLON  1,',',COMMA
        DW      HERE,DUPP,TWOP          ;cell boundary
        DW      CP,STORE,STORE,EXIT      ;adjust code pointer, compile

;   C,          ( b -- )
;               Compile a byte into the code dictionary

        $COLON  2,'C,',CCOMMA
        DW      HERE,DUPP,ONEP
        DW      CP,STORE,CSTOR,EXIT

;   [COMPILE]   ( -- ; <string> )
;               Compile the next immediate word into code dictionary.

        $COLON  IMEDD+9,'[COMPILE]',BCOMP
        DW      TICK,COMMA,EXIT

;   COMPILE     ( -- )
;               Compile the next address in colon list to code dictionary.

        $COLON  COMPO+7,'COMPILE',COMPI
        DW      RFROM,DUPP,AT,COMMA      ;compile address
        DW      TWOP,TOR,EXIT           ;adjust return address

;   LITERAL     ( w -- )
;               Compile tos to code dictionary as an integer literal.

        $COLON  IMEDD+7,'LITERAL',LITER
        DW      COMPI,DOLIT,COMMA,EXIT

;   $,"        ( -- )
;               Compile a literal string up to next " .

        $COLON  3,'$,"',STRCQ
        DW      DOLIT,'"',WORDD       ;move string to code dictionary
        DW      COUNT,PLUS        ;calculate aligned end of string
        DW      CP,STORE,EXIT            ;adjust the code pointer

;   RECURSE     ( -- )
;               Make the current word available for compilation.

        $COLON  IMEDD+7,'RECURSE',RECUR
        DW      LAST,AT,NAMET,COMMA,EXIT

;; Structures
```

```
;   FOR         ( -- a )
;               Start a FOR-NEXT loop structure in a colon definition.

      $COLON  IMEDD+3,'FOR',FOR
      DW        COMPI,TOR,HERE,EXIT

;   BEGIN       ( -- a )
;               Start an infinite or indefinite loop structure.

      $COLON  IMEDD+5,'BEGIN',BEGIN
      DW        HERE,EXIT

;   NEXT        ( a -- )
;               Terminate a FOR-NEXT loop structure.

      $COLON  IMEDD+4,'NEXT',NEXT
      DW        COMPI,DONXT,COMMA,EXIT

;   UNTIL       ( a -- )
;               Terminate a BEGIN-UNTIL indefinite loop structure.

      $COLON  IMEDD+5,'UNTIL',UNTIL
      DW        COMPI,QBRAN,COMMA,EXIT

;   AGAIN       ( a -- )
;               Terminate a BEGIN-AGAIN infinite loop structure.

      $COLON  IMEDD+5,'AGAIN',AGAIN
      DW        COMPI,BRAN,COMMA,EXIT

;   IF          ( -- A )
;               Begin a conditional branch structure.

      $COLON  IMEDD+2,'IF',IFF
      DW        COMPI,QBRAN,HERE
      DW        DOLIT,0,COMMA,EXIT

;   AHEAD       ( -- A )
;               Compile a forward branch instruction.

      $COLON  IMEDD+5,'AHEAD',AHEAD
      DW        COMPI,BRAN,HERE,DOLIT,0,COMMA,EXIT

;   REPEAT      ( A a -- )
;               Terminate a BEGIN-WHILE-REPEAT indefinite loop.

      $COLON  IMEDD+6,'REPEAT',REPEA
```

```
        DW      AGAIN,HERE,SWAP,STORE,EXIT

;   THEN        ( A -- )
;               Terminate a conditional branch structure.

        $COLON  IMEDD+4,'THEN',THENN
        DW      HERE,SWAP,STORE,EXIT

;   AFT         ( a -- a A )
;               Jump to THEN in a FOR-AFT-THEN-NEXT loop the first time through.

        $COLON  IMEDD+3,'AFT',AFT
        DW      DROP,AHEAD,BEGIN,SWAP,EXIT

;   ELSE        ( A -- A )
;               Start the false clause in an IF-ELSE-THEN structure.

        $COLON  IMEDD+4,'ELSE',ELSEE
        DW      AHEAD,SWAP,THENN,EXIT

;   WHILE       ( a -- A a )
;               Conditional branch out of a BEGIN-WHILE-REPEAT loop.

        $COLON  IMEDD+5,'WHILE',WHILE
        DW      IFF,SWAP,EXIT

;   ABORT"     ( -- ; <string> )
;               Conditional abort with an error message.

        $COLON  IMEDD+6,'ABORT"',ABRTQ
        DW      COMPI,ABORQ,STRCQ,EXIT

;   $"         ( -- ; <string> )
;               Compile an inline string literal.

        $COLON  IMEDD+2,'$"',STRQ
        DW      COMPI,STRQP,STRCQ,EXIT

;   ."         ( -- ; <string> )
;               Compile an inline string literal to be typed out at run time.

        $COLON  IMEDD+2,'."',DOTQ
        DW      COMPI,DOTQP,STRCQ,EXIT

;; Name compiler

;   ?UNIQUE     ( a -- a )
;               Display a warning message if the word already exists.
```

```
          $COLON  7,'?UNIQUE',UNIQU
          DW      DUPP,NAMEQ                  ;?name exists
          DW      QBRAN,UNIQ1                 ;redefinitions are OK
          D$      DOTQP,' reDef '             ;but warn the user
          DW      OVER,COUNT,TYPEE            ;just in case its not planned
UNIQ1:          DW      DROP,EXIT

;   $,n         ( na -- )
;               Build a new dictionary name using the string at na.

          $COLON  3,'$,n',SNAME
          DW      DUPP,CAT                    ;?null input
          DW      QBRAN,PNAM1
          DW      UNIQU                       ;?redefinition
          DW      DUPP,LAST,STORE             ;save na for vocabulary link
          DW      HERE,SWAP                   ;align code address
          DW      TWOM                       ;link address
          DW      CRRNT,AT,AT,OVER,STORE
          DW      TWOM,DUPP,NP,STORE         ;adjust name pointer
          DW      STORE,EXIT                  ;save code pointer
PNAM1:          D$      STRQP,' name'               ;null input
          DW      THROW

;; FORTH compiler

;   $COMPILE    ( a -- )
;               Compile next word to code dictionary as a token or literal.

          $COLON  8,'$COMPILE',SCOMP
          DW      NAMEQ,QDUP                  ;?defined
          DW      QBRAN,SCOM2
          DW      AT,DOLIT,IMEDD,ANDD         ;?immediate
          DW      QBRAN,SCOM1
          DW      EXECU,EXIT                  ;its immediate, execute
SCOM1:          DW      COMMA,EXIT                  ;its not immediate, compile
SCOM2:          DW      TNUMB,ATEXE                 ;try to convert to number
          DW      QBRAN,SCOM3
          DW      LITER,EXIT                  ;compile number as integer
SCOM3:          DW      THROW                       ;error

;   CCOMPILE    ( a -- )
;               Compile next byte to code dictionary as machine code.

          $COLON  8,'CCOMPILE',CCOMP
          DW      NAMEQ,QDUP                  ;?defined
          DW      QBRAN,CCOM2
          DW      AT,DOLIT,IMEDD,ANDD         ;?immediate
          DW      QBRAN,CCOM1
          DW      EXECU,EXIT               ;its immediate, execute
CCOM1:          DW      DROP,EXIT                ;its not immediate,drop
CCOM2:          DW      TNUMB,ATEXE              ;try to convert to number
          DW      QBRAN,CCOM3
          DW      CCOMMA,EXIT              ;compile as code byte
CCOM3:          DW      THROW                    ;error

;   OVERT       ( -- )
;               Link a new word into the current vocabulary.

          $COLON  5,'OVERT',OVERT
          DW      LAST,AT,CRRNT,AT,STORE,EXIT

;   ;           ( -- )
;               Terminate a colon definition.

          $COLON  IMEDD+COMPO+1,';',SEMIS
          DW      COMPI,EXIT,LBRAC,OVERT,EXIT

;   ]           ( -- )
;               Start compiling the words in the input stream.

          $COLON  1,']',RBRAC
          DW      DOLIT,SCOMP,TEVAL,STORE,EXIT

;   call,       ( ca -- )
;               Assemble a call instruction to doLST.

          $COLON  5,'call,',CALLC
          DW      DOLIT,CALLL,CCOMMA,EXIT  ;Direct Threaded Code

;   :           ( -- ; <string> )
;               Start a new colon definition using next word as its name.

          $COLON  1,':',COLON
          DW      TOKEN,SNAME
          DW      CALLC,RBRAC,EXIT

;   IMMEDIATE   ( -- )
;               Make the last compiled word an immediate word.

          $COLON  9,'IMMEDIATE',IMMED
          DW      DOLIT,IMEDD,LAST,AT,AT,ORR
          DW      LAST,AT,STORE,EXIT

;; Defining words
```

```
;   USER        ( u -- ; <string> )
;               Compile a new user variable.

          $COLON  4,'USER',USER
          DW      TOKEN,SNAME,OVERT,CALLC
          DW      COMPI,DOUSE,COMMA,EXIT

;   CREATE      ( -- ; <string> )
;               Compile a new array entry without allocating code space.

          $COLON  6,'CREATE',CREAT
          DW      TOKEN,SNAME,OVERT,CALLC
          DW      COMPI,DOVAR,EXIT

;   VARIABLE    ( -- ; <string> )
;               Compile a new variable initialized to 0.

          $COLON  8,'VARIABLE',VARIA
          DW      CREAT,DOLIT,0,COMMA,EXIT

;   CODE        ( -- )
;               Start a new code definition using next word as its name.

          $COLON  4,'CODE',CODE
          DW      TOKEN,SNAME
          DW      DOLIT,CCOMP,TEVAL,STORE,EXIT

;   ENDCODE     ( -- )
;               Terminate a code definition

          $COLON  IMEDD+COMPO+7,'ENDCODE',ENDCD
          DW      DOLIT,48H,CCOMMA,DOLIT,84H,CCOMMA       ;$NEXT
          DW      DOLIT,48H,CCOMMA,DOLIT,28H,CCOMMA
          DW      LBRAC,OVERT,EXIT

;; Tools

;   _TYPE       ( b u -- )
;               Display a string. Filter non-printing characters.

          $COLON  5,'_TYPE',UTYPE
          DW      TOR                     ;start count down loop
          DW      BRAN,UTYP2              ;skip first pass
UTYP1:          DW      DUPP,CAT,TCHAR,EMIT     ;display only printable
          DW      ONEP                      ;increment address
UTYP2:          DW      DONXT,UTYP1             ;loop till done
          DW      DROP,EXIT
```

```
;   dm+         ( a u -- a )
;               Dump u bytes from , leaving a+u on the stack.

          $COLON  3,'dm+',DMP
          DW      OVER,DOLIT,4,UDOTR       ;display address
          DW      SPACE,TOR                ;start count down loop
          DW      BRAN,PDUM2               ;skip first pass
PDUM1:          DW      DUPP,CAT,DOLIT,3,UDOTR  ;display numeric data
          DW      ONEP                       ;increment address
PDUM2:          DW      DONXT,PDUM1              ;loop till done
          DW      EXIT

;   DUMP        ( a u -- )
;               Dump u bytes from a, in a formatted manner.

          $COLON  4,'DUMP',DUMP
          DW      BASE,AT,TOR,HEX          ;save radix, set hex
          DW      DOLIT,16,SLASH           ;change count to lines
          DW      TOR                      ;start count down loop
DUMP1:          DW      CR,DOLIT,16,DDUP,DMP    ;display numeric
          DW      ROT,ROT
          DW      SPACE,SPACE,UTYPE        ;display printable characters
          DW      NUFQ,INVER               ;user control
          DW      QBRAN,DUMP2
          DW      DONXT,DUMP1              ;loop till done
          DW      BRAN,DUMP3
DUMP2:          DW      RFROM,DROP               ;cleanup loop stack, early exit
DUMP3:          DW      DROP,RFROM,BASE,STORE   ;restore radix
          DW      EXIT

;   .S          ( ... -- ... )
;               Display the contents of the data stack.

          $COLON  2,'.S',DOTS
          DW      CR,DEPTH                 ;stack depth
          DW      TOR                      ;start count down loop
          DW      BRAN,DOTS2               ;skip first pass
DOTS1:          DW      RAT,PICK,DOT             ;index stack, display contents
DOTS2:          DW      DONXT,DOTS1              ;loop till done
          D$      DOTQP,' <sp'
          DW      EXIT

;   !CSP        ( -- )
;               Save stack pointer in CSP for error checking.

          $COLON  4,'!CSP',STCSP
          DW      SPAT,CSP,STORE,EXIT      ;save pointer
```

```
;   ?CSP        ( -- )
;               Abort if stack pointer differs from that saved in CSP.

          $COLON  4,'?CSP',QCSP
          DW      SPAT,CSP,AT,XORR        ;compare pointers
          D$      ABORQ,'stacks'          ;abort if different
          DW      EXIT

;   >NAME       ( ca -- na | F )
;               Convert code address to a name address.

          $COLON  5,'>NAME',TNAME
          DW      CRRNT                   ;vocabulary link
TNAM1:          DW      TWOP,AT,QDUP            ;check all vocabularies
          DW      QBRAN,TNAM4
          DW      DDUP
TNAM2:          DW      AT,DUPP                 ;?last word in a vocabulary
          DW      QBRAN,TNAM3
          DW      DDUP,NAMET,XORR         ;compare
          DW      QBRAN,TNAM3
          DW      TWOM                   ;continue with next word
          DW      BRAN,TNAM2
TNAM3:          DW      SWAP,DROP,QDUP
          DW      QBRAN,TNAM1
          DW      SWAP,DROP,SWAP,DROP,EXIT
TNAM4:          DW      DROP,DOLIT,0,EXIT       ;false flag

;   .ID         ( na -- )
;               Display the name at address.

          $COLON  3,'.ID',DOTID
          DW      QDUP                    ;if zero no name
          DW      QBRAN,DOTI1
          DW      COUNT,DOLIT,01FH,ANDD   ;mask lexicon bits
          DW      UTYPE,EXIT              ;display name string
DOTI1:          D$      DOTQP,' {noName}'
          DW      EXIT

;   WORDS       ( -- )
;               Display the names in the context vocabulary.

          $COLON  5,'WORDS',WORDS
          DW      CR,CNTXT,AT             ;only in context
WORS1:          DW      AT,QDUP                 ;?at end of list
          DW      QBRAN,WORS2
          DW      DUPP,SPACE,DOTID        ;display a name
          DW      TWOM,NUFQ              ;user control
          DW      QBRAN,WORS1
          DW      DROP
WORS2:          DW      EXIT

;; Hardware reset

;   VER         ( -- n )
;               Return the version number of this implementation.

          $COLON  3,'VER',VERSN
          DW      DOLIT,VER*256+EXT,EXIT

;   hi          ( -- )
;               Display the sign-on message of eForth.

          $COLON  2,'hi',HI
          DW      STOIO,CR                ;initialize I/O
          D$      DOTQP,'eForth v'
          DW      BASE,AT,HEX
          DW      VERSN,BDIGS,DIG,DIG
          DW      DOLIT,'.',HOLD
          DW      DIGS,EDIGS,TYPEE
          DW      BASE,STORE,CR,EXIT

;   'BOOT       ( -- a )
;               The application startup vector.

          $COLON  5,"'BOOT",TBOOT
          DW      DOVAR
          DW      HI                      ;application to boot

;   SEE         ( --word-- )
;               Decompiles word.
          $COLON  3,'SEE',SEE
          DW      TICK
          DW      CR,ONEP
    SEE1:       DW      DUPP,DUPP,SPACE,DOT,AT,DUPP
          DW      QBRAN,SEE2
          DW      TNAME
    SEE2:       DW      QDUP
          DW      QBRAN,SEE3
          DW      DOTID
          DW      BRAN,SEE4
    SEE3:       DW      DUPP,AT,UDOT
    SEE4:       DW      TWOP,NUFQ
          DW      QBRAN,SEE1
          DW      DROP,EXIT
```

```
;   ADCINIT     ( -- )
;               Init routine for starting ADC Interrupts
        $CODE 7,'ADCINIT',ADCINIT
        DB 68H,0FFH             ;V<FF
        DB 69H,0C6H             ;A<C6
        DB 63H,0F2H             ;(V/F2)<A
        DB 69H,0                ;A<0
        DB 63H,0F3H             ;(V/F3)<A
        DB 4DH,0C8H             ;ANM <A
        DB 48H,48H              ;SKIT FAD, reset INTFAD
        DB 00                   ;NOP
        DB 64H,0EH,0FEH         ;ENABLE INTAD
        $NEXT

;   TMIDI       ( n -- )
;               Wait for last transmit, then send midi byte n.
        $CODE 5,'TMIDI',TMIDI
        DB 0A1H                 ;POP BC
        DB 0BH                  ;A<C
        DB 48H,4AH              ;SKIT FST, skip if interrupt
        DB 0FDH                 ;JMP TO SKIT
        DB 4DH,0D8H             ;MOV TXB,A
        $NEXT

;   DELAY        ( n -- )
;               Wait for n loops.
        $CODE 5,'DELAY',DELAY
        DB 0A1H                 ;POP BC
        DB 53H                  ;C<C-1, Skip if borrow
        DB 0FEH                 ;JMP
        DB 52H                  ;B<B-1, Skip if borrow
        DB 0FCH                 ;JMP
        $NEXT

;   LCD         ( n -- )
;               Load control n to LCD display.
        $CODE 4,'LCD',LCD
        DB 0A1H                 ;POP BC
        DB 0BH                  ;A<C
        DB 14H,0,0A0H           ;BC<A000
        DB 39H                  ;(BC)<A
        $NEXT

;  LLI          ( --- )
;               Sets RS=0 for LCD setup commands.
        $CODE 3,'LLI',LLI
        DB 64H,0AH,0EFH         ;Pc<Pc AND EF
        $NEXT
```

```
;  LLC           ( --- )
;                Sets RS=1 for LCD character loading
       $CODE 3,'LLC',LLC
       DB 64H,1AH,10H          ;Pc<Pc OR 10
       $NEXT

;  LI            ( n --- )
;                load LCD setup instruction n, exit ready for char loads
       $COLON 2,'LI',LI
       DW      LLI,LCD,LLC,EXIT

;   LCDINIT      ( -- )
;                Initialize LCD display.
       $COLON 7,'LCDINIT',LCDINIT
       DW      DOLIT,0D7AH,DELAY
       DW      DOLIT,038H,LI
       DW      DOLIT,047EH,DELAY
       DW      DOLIT,038H,LI
       DW      DOLIT,017H,DELAY
       DW      DOLIT,038H,LI
       DW      DOLIT,017H,DELAY
       DW      DOLIT,038H,LI
       DW      DOLIT,017H,DELAY
       DW      DOLIT,08H,LI
       DW      DOLIT,017H,DELAY
       DW      DOLIT,01H,LI
       DW      DOLIT,01CCH,DELAY
       DW      DOLIT,02H,LI
       DW      DOLIT,01CCH,DELAY
       DW      DOLIT,06H,LI
       DW      DOLIT,17H,DELAY
       DW      DOLIT,0EH,LI
       DW      DOLIT,17H,DELAY
       DW      EXIT

;  #DISP         ( n,p --- )
;                Display n as a 3-digit number at LCD position p.
       $COLON  5,'#DISP',NDISP
       DW      DUPP,LI,SWAP
       DW      BDIGS,DIG,DIG,DIG,EDIGS
       DW      DROP,DUPP,CAT,LCD,ONEP
       DW      DUPP,CAT,LCD,ONEP,CAT,LCD,LI,EXIT

;  DISP          ( a,p --- )
;                Display packed string at a to LCD position p.
       $COLON  4,'DISP',DISP
       DW      LI,DUPP,CAT,TWOM,TOR
```

```
   DISP1:        DW      ONEP
          DW      DUPP,CAT,LCD
          DW      DONXT,DISP1
          DW      DROP,EXIT

;  CASE          ( n --- )
;                Execute one of a list of words pointed to by n.
          $COLON  4,'CASE',CASE
          DW      RFROM,SWAP,TWOSL,PLUS
          DW      ATEXE,EXIT

;  INCR          ( n,nmax --- n+1 )
;                Increment n mod nmax.
          $COLON  4,'INCR',INCR
          DW      OVER,ONEP,LESS
          DW      QBRAN,INCR1
          DW      DROP,DOLIT,0
          DW      BRAN,INCR2
   INCR1:        DW      ONEP
   INCR2:        DW      EXIT

;  DECR          ( n,nmax --- n-1 )
;                Decrement n mod nmax.
          $COLON  4,'DECR',DECR
          DW      OVER,ONEM,ZLESS
          DW      QBRAN,DECR1
          DW      SWAP,DROP
          DW      BRAN,DECR2
   DECR1:        DW      DROP
          DW      ONEM
   DECR2:        DW      EXIT

;  SW@           ( --- n )
;                Read Roland switches as a byte.
          $CODE   3,'SW@',SWAT
          DB      4CH,0C0H        ;;A<PA
          DB      6AH,0           ;;B<0
          DB      1BH             ;;C<A
          DB      0B1H            ;;PUSH BC
          $NEXT

;  S@            ( --- n )
;                Return number of the lowest Roland switch on.
          $CODE   2,'S@',SAT
          DB      4CH,0C0H        ;;A<PA
          DB      6BH,0           ;;C<0
          DB      74H,11H,0FFH    ;;A<A EXOR FF
          DB      74H,49H,0FFH    ;;A AND FF, SKIP IF NO ZERO
```

```
         DB      0C4H            ;; JMP OUT
         DB      43H             ;; C<C+1, LOOP1
         DB      48H,1           ;; A SHIFT RIGHT, SKIP IF CARRY
         DB      0FCH            ;; JMP LOOP1
         DB      6AH,0           ;;B<0, OUT
         DB      0B1H            ;;PUSH BC
         $NEXT

;  LED!          ( n --- )
;                Turn on/off Roland LED's.
         $CODE   4,'LED!',LEDB
         DB      0A1H            ;;POP BC
         DB      0BH             ;;A<C
         DB      74H,9H,0FCH     ;;A<A AND FC
         DB      74H,19H,1       ;;A<A OR 1
         DB      4DH,0C1H        ;;PB<A
         $NEXT

;  eUPDAT        ( --- )
;                Move data from Slider Ram to Edit Buffer.
         $CODE   6,'eUPDAT',EUPDAT
         DB      68H,0FFH        ;;V<FF
         DB      6AH,0C6H        ;;B<C6
         DB      1,0F0H          ;;A<(V/F0)
         DB      1BH             ;;C<A
         DB      29H             ;;A<(BC)
         DB      63H,4           ;;(V/04)<A
         DB      69H,38H         ;;A<70
         DB      60H,43H         ;;C<C+A
         DB      29H             ;;A<(BC)
         DB      63H,6H          ;;(V/06)<A
         DB      69H,38H         ;;A<38
         DB      60H,43H         ;;C<C+A
         DB      29H             ;;A<(BC)
         DB      63H,7H          ;;(V/07)<A
         $NEXT

;  eLOAD         ( --- )
;                Load Edit Buffer data into Slider Memory.
         $CODE   5,'eLOAD',ELOAD
         DB      68H,0FFH        ;;V<FF
         DB      6AH,0C6H        ;;B<C6
         DB      1,0             ;;A<(V/00)
         DB      1BH             ;;C<A
         DB      1,4             ;;A<(V/04)
         DB      39H             ;;(BC)<A
         DB      69H,38H         ;;A<38
         DB      60H,43H         ;;C<C+A
         DB      49H,0           ;;(BC)<0
         DB      69H,70H         ;;A<38
         DB      60H,43H         ;;C<C+A
         DB      1,6             ;;A<(V/06)
         DB      39H             ;;(BC)<A
         DB      69H,38H         ;;A<38
         DB      60H,43H         ;;C<C+A
         DB      1,7             ;;A<(V/07)
         DB      39H             ;;(BC)<A
         $NEXT

;  esUPDAT       ( --- )
;                Update only the Slider data of the Edit Buffer.
         $CODE   7,'esUPDAT',ESUPDAT
         DB      68H,0FFH        ;;V<FF
         DB      6AH,0C6H        ;;B<C6
         DB      1,0F0H          ;;A<(V/F0)
         DB      1BH             ;;C<A
         DB      29H             ;;A<(BC)
         DB      63H,4           ;;(V/04)<A
         $NEXT

;  eSLD#         ( --- FF00 )
;                Edit Buffer Slider number.
         $COLON  5,'eSLD#',ESLDN
         DW      DOLIT,0FF00H,EXIT

;  eFLD          ( --- FF01 )
;                Edit Buffer LCD Field.
         $COLON  4,'eFLD',EFLD
         DW      DOLIT,0FF01H,EXIT

;  eBYTE1        ( --- FF07 )
;                Edit Buffer Midi Status/Chnl byte.
         $COLON  6,'eBYTE1',EBYTE1
         DW      DOLIT,0FF07H,EXIT

;  eBYTE2        ( --- FF06 )
;                Edit Buffer Midi Key#, Controller#, or Program# byte.
         $COLON  6,'eBYTE2',EBYTE2
         DW      DOLIT,0FF06H,EXIT

;  eBYTE3        ( --- FF04 )
;                Edit Buffer Slider value.
         $COLON  6,'eBYTE3',EBYTE3
         DW      DOLIT,0FF04H,EXIT

;  eFLAG         ( --- FF05 )
;               Edit Buffer Midi Flag byte.
          $COLON  5,'eFLAG',EFLAG
          DW      DOLIT,0FF05H,EXIT

;  FLD0           ( --- 80 )
;               LCD Field start.
          $COLON  4,'FLD0',FLD0
          DW      DOLIT,080H,EXIT

;  FLD1           ( --- 86 )
;               LCD Field start.
          $COLON  4,'FLD1',FLD01
          DW      DOLIT,086H,EXIT

;  FLD2           ( --- 8A )
;               LCD Field start.
          $COLON  4,'FLD2',FLD2
          DW      DOLIT,08AH,EXIT

;  FLD3           ( --- 8D )
;               LCD Field start.
          $COLON  4,'FLD3',FLD3
          DW      DOLIT,08DH,EXIT

;  FLD4           ( --- C0 )
;               LCD Field start.
          $COLON  4,'FLD4',FLD4
          DW      DOLIT,0C0H,EXIT

;  FLD5           ( --- C9 )
;               LCD Field start.
          $COLON  4,'FLD5',FLD5
          DW      DOLIT,0C9H,EXIT

;  FLD6           ( --- CD )
;               LCD Field start.
          $COLON  4,'FLD6',FLD6
          DW      DOLIT,0CDH,EXIT

;  L0             ( --- a )
;               Packed string. 'a' is addr of count byte.
          $COLON  2,'L0',L0
          SD$ 'Slider'

;  L1             ( --- a )
;               Packed string. 'a' is addr of count byte.
          $COLON  2,'L1',L1
          SD$ 'Setup#'
```

```
;  L2            ( --- a )
;                Packed string. 'a' is addr of count byte.
          $COLON  2,'L2',L2
          SD$ '* MIDI Running *'

;  L20           ( --- a )
;                Packed string. 'a' is addr of count byte.
          $COLON  3,'L20',L20
          SD$ 'Ch '

;  L21           ( --- a )
;                Packed string. 'a' is addr of count byte.
          $COLON  3,'L21',L21
          SD$ 'Off'

;  L40           ( --- a )
;                Packed string. 'a' is addr of count byte.
          $COLON  3,'L40',L40
          SD$ 'Key#    '

;  L41           ( --- a )
;                Packed string. 'a' is addr of count byte.
          $COLON  3,'L41',L41
          SD$ 'Key# A-T'

;  L42           ( --- a )
;                Packed string. 'a' is addr of count byte.
          $COLON  3,'L42',L42
          SD$ 'Control#'

;  L43           ( --- a )
;                Packed string. 'a' is addr of count byte.
          $COLON  3,'L43',L43
          SD$ 'Program#'

;  L44           ( --- a )
;                Packed string. 'a' is addr of count byte.
          $COLON  3,'L44',L44
          SD$ 'Ch Press'

;  L45           ( --- a )
;                Packed string. 'a' is addr of count byte.
          $COLON  3,'L45',L45
          SD$ 'Ptch Whl'

;  L4X           ( --- a )
;                Packed string. 'a' is addr of count byte.
```

```
        $COLON  3,'L4X',L4X
        SD$ '********'

;  L50          ( --- a )
;               Packed string. 'a' is addr of count byte.
        $COLON  3,'L50',L50
        SD$ '***'

;  LSTAT        ( n --- )
;               Choose a midi status label.
        $COLON  5,'LSTAT',LSTAT
        DW      CASE,L4X,L40,L41,L42,L43,L44,L45,L4X,EXIT

;  eDISP        ( --- )
;               Display the Edit buffer on the LCD
        $COLON  5,'eDISP',EDISP
        DW      L0,FLD0,DISP,ESLDN,CAT,FLD01,NDISP
        DW      EBYTE3,CAT,DOLIT,80H,ANDD
        DW      QBRAN,EDISP1
        DW      L21,FLD2,DISP
        DW      BRAN,EDISP2
   EDISP1:      DW      L20,FLD2,DISP
   EDISP2:      DW      EBYTE1,CAT,DUPP,DOLIT,0FH,ANDD,FLD3,NDISP
        DW      TWOSL,TWOSL,TWOSL,TWOSL,DOLIT,7H,ANDD
        DW      LSTAT,FLD4,DISP,EBYTE2,CAT,FLD5,NDISP
        DW      EBYTE3,CAT,DOLIT,7FH,ANDD,FLD6,NDISP
        DW      DOLIT,6H,EFLD,CSTOR,EXIT

;  BL/R         ( fld --- pos )
;               Translates LCD field number to a position number.
        $COLON  4,'BL/R',BLR
        DW      DUPP,EFLD,CAT,CASE
        DW      FLD0,FLD01,FLD2,FLD3,FLD4,FLD5,FLD6
        DW      EXIT

;  BLEFT        ( --- )
;               Moves the LCD cursor to next field. Loads eFLD.
        $COLON  5,'BLEFT',BLEFT
        DW      DOLIT,40H,LEDB
        DW      EFLD,CAT,DOLIT,6,DECR,BLR,LI,EXIT

;  BRIGHT       ( --- )
;               Moves the LCD cursor to next field. Loads eFLD.
        $COLON  6,'BRIGHT',BRIGHT
        DW      DOLIT,80H,LEDB
        DW      EFLD,CAT,DOLIT,6,INCR,BLR,LI,EXIT

;  BLOAD        ( --- )
;               Load Buffer data shown on LCD into Slider Memory.
        $COLON  5,'BLOAD',BLOAD
        DW      DOLIT,4,LEDB
        DW      ELOAD,EXIT

;  BMIDI        ( --- )
;               Start the Midi program.
        $COLON  5,'BMIDI',BMIDI
        DW      DOLIT,1,LI
        DW      L2,FLD0,DISP
        DW      EXIT

;  BUP          ( --- )
;               Increment value in LCD cursor field.
        $COLON  3,'BUP',BUP
        DW      DOLIT,10H,LEDB
        DW      DOLIT,1,EFLD,CAT
        DW      DOLIT,7,ANDD,CASE
        DW      UD0,UD1,UD2,UD3,UD4,UD5,UD6,UD7,EXIT

;  BDOWN        ( --- )
;               Decrement value in LCD cursor field.
        $COLON  3,'BDOWN',BDOWN
        DW      DOLIT,20H,LEDB
        DW      DOLIT,0,EFLD,CAT
        DW      DOLIT,7,ANDD,CASE
        DW      UD0,UD1,UD2,UD3,UD4,UD5,UD6,UD7,EXIT

;  U/D0         ( i/d --- )
;               Field increment/decrement routine.
        $COLON  4,'U/D0',UD0
        DW      DROP,FLD0,LI,EXIT

; U/D7          ( i/d --- )
;               Field increment/decrement routine. (bogus field)
        $COLON  4,'U/D7',UD7
        DW      DROP,EXIT

;  U/D6         ( i/d --- )
;               Field increment/decrement routine.
        $COLON  4,'U/D6',UD6
        DW      DROP,FLD6,LI,EXIT

;  U/D1         ( i/d --- )
;               Field increment/decrement routine.
        $COLON  4,'U/D1',UD1
        DW      ESLDN,CAT,DOLIT,37H,ROT
        DW      QBRAN,UD1A
```

```
        DW      INCR
        DW      BRAN,UD1B
UD1A:         DW      DECR
UD1B:         DW      CFLD1,EXIT

; CFLD1         ( sld# --- )
;               Change Slider# in field 1.  Update Edit buffer & LCD.
        $COLON  5,'CFLD1',CFLD1
        DW      ESLDN,CAT,EUPDAT,EDISP
        DW      DOLIT,1,EFLD,CSTOR,FLD01,LI,EXIT

; U/D2          ( i/d --- )
;               Field increment/decrement routine.
        $COLON  4,'U/D2',UD2
        DW      DROP,EBYTE3,CAT,DUPP,DOLIT,80H,ANDD
        DW      QBRAN,UD2A
        DW      DOLIT,7FH,ANDD,EBYTE3,CAT,L20,FLD2,DISP
        DW      BRAN,UD2B
UD2A:         DW      DOLIT,80H,ORR,EBYTE3,CSTOR,L21,FLD2,DISP
UD2B:         DW      FLD2,LI,EXIT

; U/D3          ( i/d --- )
;               Field increment/decrement routine.
        $COLON  4,'U/D3',UD3
        DW      EBYTE1,CAT,DOLIT,0FH,ANDD,DOLIT,0FH,ROT
        DW      QBRAN,UD3A
        DW      INCR
        DW      BRAN,UD3B
UD3A:         DW      DECR
UD3B:         DW      CFLD3,EXIT

; CFLD3         ( chnl --- )
;               Change midi channel in field 3.
        $COLON  5,'CFLD3',CFLD3
        DW      DUPP,EBYTE1,CAT,DOLIT,0F0H
        DW      ANDD,ORR,EBYTE1,CSTOR,FLD3,NDISP,EXIT

; U/D4          ( i/d --- )
;               Field increment/decrement routine.
        $COLON  4,'U/D4',UD4
        DW      EBYTE1,CAT,DOLIT,70H,ANDD
        DW      TWOSR,TWOSR,TWOSR,TWOSR,DOLIT,7,ROT
        DW      QBRAN,UD4A
        DW      INCR
        DW      BRAN,UD4B
UD4A:         DW      DECR
UD4B:         DW      CFLD4,EXIT
```

```
;  CFLD4         ( status --- )
;                Change Midi operation label in field 4.
          $COLON  5,'CFLD4',CFLD4
          DW      DUPP,TWOSL,TWOSL,TWOSL,TWOSL
          DW      DOLIT,80H,ORR,EBYTE1,CAT
          DW      DOLIT,0FH,ANDD,ORR,EBYTE1,CSTOR
          DW      LSTAT,FLD4,DISP,FLD4,LI,EXIT

;  U/D5          ( i/d --- )
;                Field increment/decrement routine.
          $COLON  4,'U/D5',UD5
          DW      DOLIT,0CFH,EBYTE1,CAT,DOLIT,0F0H,ANDD,LESS
          DW      QBRAN,UD5A
          DW      L50,FLD5,DISP,FLD5,LI,DROP
          DW      BRAN,UD5B
   UD5A:         DW      DUPP,EBYTE2,CSTOR,FLD5,NDISP
   UD5B:         DW      EXIT




;
;================================================================

LASTN           EQU     _NAME+4                 ;last name address

NTOPP           EQU     _NAME-0      ;next available memory in ROM name
dictionary
CTOPP           EQU     $+0          ;next available memory in ROM code
dictionary
ROMSPC          EQU     NTOPP-CTOPP  ;UNUSED DICTIONARY ROM SPACE

MAIN    ENDS
END     ORIG

;================================================================
```

Modified Roland PG1000

PG1000 Assembly Listing

```
                                Page 60,132
                                ;==============================================================
                                ;
                                ;   eForth 1.0 by Bill Muench and C. H. Ting, 1990
                                ;
                                ;   This is an implementation for the NEC 78C10 microcomputer by
                                ;   John Talbert, 1994, Oberlin Conservatory.
                                ;
                                ;   Register Use:       Interpreter Pointer = DE
                                ;                       Data Stack Pointer  = SP
                                ;                       Return Stack Pointer  = HL

                                ;                       Free to use: BC, EA, VA, Alternate Registers.
                                ;
                                ;   'doList'  is accessed as a subroutine through a CALT instruction
                                ;            (Call to Jump Table).  This shows up as a 'DB 80H' line
                                ;            in the $COLON and $USER Macros.  When executed the
                                ;            processor jumps to an address vector located at 80H.  The
                                ;            vectored 'doList' code is then located at 0F0H. The word
                                ;            'call,' was changed to load 80H into the code area for a
                                ;            doLST assembly.
                                ;
                                ;   A 9600 Baud serial I/O is provided.  PortB/bit0 is used for serial
                                ;       output and PortC/bit3 (INT2) is used for the serial input.  The
                                ;       serial input is interrupt driven with a vectored interrupt routine
                                ;       located at 0A0H.  The code words ?RX, TX!, and !IO make up the
                                ;       rest of the serial I/O code.  Three USER variables have been set
                                ;       up for use by these serial I/O routines:  SERIN, which holds the
                                ;       received character and a flag; HAFBIT, which adjusts the software
                                ;       timing of the receiver to read in the middle of each bit frame
                                ;       (set it for 1/2 the BITIME minus 5); and BITIME, which adjusts the
                                ;       software for a specific baud rate (17H for 9600 baud assuming a 12Mhz
                                ;       processor clock).
                                ;
                                ;   The 78C10 is an 8-bit micro, therefore cell aligning to even addresses
                                ;       is unnecessary. The $ALIGN Macro was taken out along with the NOP's
                                ;       used for cell alignment in the other Macros.  All occurrences of the
                                ;       word ALGND were erased also. The word SEE no longer works because it
                                ;       relies on cell alignment.
                                ;
                                ;   All of the system FORTH code is to be stored in ROM (up to 32K) starting
                                ;       at address 0000H. Then there is 2K of RAM starting at address COOOH.
                                ;       This memory setup required the following changes:
                                ;               1) Return and Data stacks and TIB moved to RAM.
                                ;                  (See the Memory allocation EQU assignments.)
                                ;               2) The USER variables were moved to the micro's
                                ;                  internal RAM at FF00H to FFFFH.
                                ;               3) PAD word was changed to move the temporary buffer
                                ;                  area to RAM space.
                                ;               4) The vocabulary pointers found in the word FORTH were
                                ;                  moved to RAM space by creating two new USER variables,
                                ;                  FHEAD and FLINK and changing DOVOC to read:
                                ;                  DW FHEAD,CNTXT,STORE,EXIT.
                                ;               5) NTOP and CTOP were moved to RAM space to allow dictionary
                                ;                  expansion into RAM space.
                                ;
```

```
                                 ;   Several words were added to the ROM Dictionary.  The simple operators
                                 ;       1+,1-,2+,2-,2*,2/, were defined in machine code.  The words C, ,
                                 ;       CCOMPILE, CODE, and ENDCODE were created to enable the creation
                                 ;       of code definition.
                                 ;
                                 ;    The NEC78C10 offers the following advantages:
                                 ;               1) Ten 16-bit internal registers and a 16-bit ALU.
                                 ;                  Many 16-bit instructions for those FORTH stack operations.
                                 ;               2) Three 8-bit I/O ports.
                                 ;               3) Eight 8-bit Analog to Digital Converters.
                                 ;               4) Internal counters and programmable clock generators.
                                 ;               5) Internal hardware serial I/O.  (can be used for MIDI I/O).
                                 ;               6) 64K address space including 256 bytes of internal RAM.
                                 ;
                                 ;
                                 ;===================================================================
                                 ;; Version control
 = 0001                                 VER             EQU     01H                     ;major release version
 = 0001                                 EXT             EQU     01H                     ;minor extension

                                 ;; Constants
 = 0040                                 COMPO           EQU     040H                    ;lexicon compile only bit
 = 0080                                 IMEDD           EQU     080H                    ;lexicon immediate bit
 = 7F1F                                 MASKK           EQU     07F1FH                  ;lexicon bit mask

 = 0002                                 CELLL           EQU     2                       ;size of a cell
 = 000A                                 BASEE           EQU     10                      ;default radix
 = 0006                                 VOCSS           EQU     6                       ;depth of vocabulary stack

 = 0008                                 BKSPP           EQU     8                       ;backspace
 = 000A                                 LF              EQU     10                      ;line feed
 = 000D                                 CRR             EQU     13                      ;carriage return
 = 001B                                 ERR             EQU     27                      ;error escape
 = 0027                                 TIC             EQU     39                      ;tick

 = 0080                                 CALLL           EQU     80H                     ;CALT opcodes

                                 ;; Memory allocation    0//code>--//--<name//up>--<sp//tib>--rp//em

 = 0100                                 COLDD           EQU     00100H                  ;cold start
 = C2F0                                 RPP             EQU     0C2F0H                  ;start of return stack
(RP0)
 = C200                                 TIBB            EQU     0C200H                  ;terminal input buffer
(TIB)
 = C1F0                                 SPP             EQU     0C1F0H                  ;start of data stack (SP0)
 = FF00                                 UPP             EQU     0FF00H                  ;start of user area (UP0)
 = 3FFD                                 NAMEE           EQU     03FFDH                  ;name dictionary
 = 0300                                 CODEE           EQU     00300H                  ;code dictionary
 = C390                                 CTOP            EQU     0C390H                  ;RAM code dict. expansion
 = C7FF                                 NTOP            EQU     0C7FFH                  ;RAM name dict. expansion
 = C300                                 PADD            EQU     0C300H                  ;PAD area

                                 ;; Initialize assembly variables

 = 0000                                 _LINK   = 0                                     ;force a null link
 = 3FFD                                 _NAME   = NAMEE                                 ;initialize name pointer
 = 0300                                 _CODE   = CODEE                                 ;initialize code pointer
 = 0008                                 _USER   = 4*CELLL                               ;first user variable offset
```

```

                        ;; Define assembly macros

                        ;       Compile a code definition header.

                        $CODE   MACRO   LEX,NAME,LABEL
                        LABEL:                                  ;;assembly label
                                _CODE   = $                     ;;save code pointer
                                _LEN    = (LEX AND 01FH)/CELLL  ;;string cell count, round down
                                _NAME   = _NAME-((_LEN+3)*CELLL) ;;new header on cell boundary
                        ORG     _NAME                           ;;set name pointer
                                DW      _CODE,_LINK             ;;token pointer and link
                                _LINK   = $                     ;;link points to a name string
                                DB      LEX,NAME                ;;name string
                        ORG     _CODE                           ;;restore code pointer
                                ENDM

                        ;       Compile a colon definition header.

                        $COLON  MACRO   LEX,NAME,LABEL
                                $CODE   LEX,NAME,LABEL
                                DB 80H                          ;;include CALT doLIST
                                ENDM

                        ;       Compile a user variable header.

                        $USER   MACRO   LEX,NAME,LABEL
                                $CODE   LEX,NAME,LABEL
                                DB 80H                          ;;include CALT doLIST
                                DW      DOUSE,_USER             ;;followed by doUSER and offset
                                _USER   = _USER+CELLL           ;;update user area offset
                                ENDM

                        ;       Compile an inline string.

                        D$      MACRO   FUNCT,STRNG
                                DW      FUNCT                   ;;function
                                _LEN    = $                     ;;save address of count byte
                                DB      0,STRNG                 ;;count byte and string
                                _CODE   = $                     ;;save code pointer
                        ORG     _LEN                            ;;point to count byte
                                DB      _CODE-_LEN-1            ;;set count
                        ORG     _CODE                           ;;restore code pointer
                                ENDM

                        ;       Compile a stored string.

                        SD$     MACRO   STRNG
                                DW DOLIT
                                _LEN    = $ + 4                 ;;save address of count byte
                                DW      _LEN,EXIT               ;;save cnt address on stack
                                DB      0,STRNG                 ;;count byte and string

                                _CODE   = $                     ;;save code pointer
                        ORG     _LEN                            ;;point to count byte
                                DB      _CODE-_LEN-1            ;;set count
                                ORG     _CODE                                   ;;restore code pointer
                                        ENDM

                                ;       Assemble inline direct threaded code ending.

                                $NEXT   MACRO
                                        DB 48H,84H                              ;;EA<(DE)++,next code address into
AX
                                        DB 48H,28H                              ;;JMP EA,jump directly to code
address
                                        ENDM

                                ;; Main entry points and COLD start data

 0000                           MAIN    SEGMENT
                                ASSUME  CS:MAIN,DS:MAIN,ES:MAIN,SS:MAIN

 0000                           ORG     0000H

 0000  54 00 01 00              ORIG:   DB 54H,00,01,00                         ;RESET vector, JMP 0100H
 0004  AA 62 00 00                      DB 0AAH,62H,0,0                         ;NMI vector, EI RETI
 0008  0008[                            DB 8 DUP(0)                             ;INT T0/T1 vector
          00
                         ]

 0010  54 A0 00                                 DB 54H,0A0H,00H, 5 DUP(0)               ;INT1/2 vector, JMP 00A0H
       0005[
          00
                         ]

 0018  0008[                            DB 8 DUP(0)                             ;INT E1/E0 vector
          00
                         ]

 0020  54 00 02                                 DB 54H,00,02, 5 DUP(0)                  ;INT EIN/AD vector, JMP
0200H
       0005[
          00
                         ]

 0028  0008[                            DB 8 DUP(0)                             ;INT SR/ST vector
          00
                         ]

 0030  0030[                            DB 48 DUP(0)                            ;FREE
          00
                         ]

 0060  0020[                            DB 32 DUP(0)                            ;SOFTI vector at 0060H
          00
                         ]


 00A0                           ORG     00A0H

                                ; Vectored INT2 routine for Serial Input from Host Computer.
                                ; Uses address FFF0 as a counter location - do not use elsewhere!
 00A0  B1                               DB 0B1H                 ;PUSH BC
 00A1  B2                               DB 0B2H                 ;PUSH DE
```

```
00A2  B0                            DB 0B0H                 ;PUSH VA
00A3  68 FF                         DB 68H,0FFH             ;MVI, V<FF
00A5  71 F0 07                              DB 71H,0F0H,07H        ;MVIW, (V/F0)<07, number of bits to
receive.
00A8  70 1F 4C FF                   DB 70H,1FH,04CH,0FFH    ;LBCD, BC<(FF4C), wait for a half bit.
00AC  53                            DB 53H                  ;DCR, C<C-1 skip, LOOP1
00AD  FE                            DB 0FEH                 ;JR, Jump to loop1
00AE  52                            DB 52H                  ;DCR, B<B-1 skip
00AF  FC                            DB 0FCH                 ;JR, Jump to loop1
00B0  70 1F 4E FF                   DB 70H,1FH,4EH,0FFH     ;LBCD, BC<(FF4E),wait 1 bit time,  LOOP2
00B4  53                            DB 53H                  ;DCR, C<C-1 skip
00B5  FE                            DB 0FEH                 ;JR, Jump to loop2
00B6  52                            DB 52H                  ;DCR, B<B-1 skip
00B7  FC                            DB 0FCH                 ;JR, Jump to loop2
00B8  4C C2                         DB 04CH,0C2H            ;MOV, A<PC, read serial input on pc3
00BA  48 31 48 31                   DB 48H,31H,48H,31H      ;Rotate PC3 bit into Cy
00BE  48 31 48 31                   DB 48H,31H,48H,31H      ;RLR, A rotate right 4xs
00C2  0C                            DB 0CH                  ;MOV, A<D, D collects the bits
00C3  48 31                         DB 48H,31H              ;RLR, shift in next bit, CY to top of D
00C5  1C                            DB 1CH                  ;MOV, D<A
00C6  30 F0                         DB 30H,0F0H             ;DCRW, (V/F0)<(V/F0)-1 skip
00C8  E7                            DB 0E7H                 ;JR, Jump to loop2 for next bit.
00C9  70 1F 4E FF                   DB 70H,1FH,4EH,0FFH     ;LBCD, BC<(FF4E)
00CD  53 FE 52 FC                   DB 53H,0FEH,52H,0FCH    ;DCR JR DCR JR, stop bit loop time.
00D1  71 4B FF                              DB 71H,04BH,0FFH       ;MVIW, (V/4B)<FF, load flag
00D4  0C 63 4A                              DB 0CH,63H,04AH        ;MOV STAW, A<D (V/4A)<A, load data
00D7  A0 A2 A1                              DB 0A0H,0A2H,0A1H      ;POP, restore AV DE and BC
00DA  48 44 00                              DB 48H,44H,0           ;SKIT,NOP
00DD  AA 62                         DB 0AAH,062H            ;EI RETI, enable interrupts and return

                            ;; Kernel doLST routine.  Always accessed by the CALT instruction: 80H
                            ;; which is a Call Subroutine to jump to address vector located at 0080H.

00F0                        ORG     00F0H
00F0  33 33                         DB 33H,33H              ;HL<HL-2
00F2  A6                            DB 0A6H                 ;EA<DE
00F3  48 93                         DB 48H,93H              ;(HL)<EA
00F5  A2                            DB 0A2H                 ;POP DE previously pushed by CALT
00F6  48 84                         DB 48H,84H              ;EA<(DE)++, $NEXT
00F8  48 28                         DB 48H,28H              ;JMP EA
0080                        ORG     0080H
0080  F0 00                         DB 0F0H,0               ; set up vector to doLST

0100                        ORG     COLDD                    ;Beginning of Cold Boot
0100  69 0F 4D D0                   DB 69H,0FH,4DH,0D0H     ;MM<0F, memory map (11-8)
0104  69 FF 4D D2                   DB 69H,0FFH,4DH,0D2H    ;MA<FF, pa inputs (4-2)
0108  69 00 4D D3                   DB 69H,00H,4DH,0D3H     ;MB<00, pb outputs (4-6)
010C  64 01 05                              DB 64H,01H,05H         ;PB<5
010F  4D D7                         DB 4DH,0D7H             ;MF<00, pf outputs (4-15)
0111  69 0A 4D D4                   DB 69H,0AH,4DH,0D4H     ;MC<0A, pc1/3 inputs (4-9)
0115  69 0B 4D D1                   DB 69H,0BH,4DH,0D1H     ;MCC<0B, pc mode (4-8)
0119  64 02 04                              DB 64H,02H,04H         ;PC<04
011C  64 81 06                              DB 64H,81H,06H         ;SMH<06, serial mode (7-7)
011F  69 4E 4D CA                   DB 69H,4EH,4DH,0CAH     ;SML<4E, serial mode (7-9)
0123  04                            DB 04H                  ;SP<SPP, stack pointer=data stack
0124  F0                            DB LOW SPP
0125  C1                            DB HIGH SPP
```

```
0126  34                            DB 34H                  ;HL<RPP, HL=return stack pointer
0127  F0                            DB LOW RPP
0128  C2                            DB HIGH RPP
0129  69 00 4D E8                   DB 69H,00H,4DH,0E8H     ;ZCM<0, zero cross disabled (3-26)
012D  68 FF                         DB 68H,0FFH             ;V<FF
012F  10 68 FF 69 00                DB 10H,68H,0FFH,69H,0   ;V'<FF, A"<0, V<FF, A<0

                            ;; timer setups for Midi and LCD use
0134  69 64 4D DA                   DB 69H,64H,4DH,0DAH     ;TM0<64, timer0 (5-1)
0138  69 FF 4D DB                   DB 69H,0FFH,4DH,0DBH    ;TM1<FF, timer1 (5-1)
013C  64 85 B3                             DB 64H,85H,0B3H         ;TMM<B3, timer mode (5-6)
013F  44 60 EA 48 D3                DB 44H,60H,0EAH,48H,0D3H        ;ETM1<EA = EA60 (6-2)
0144  64 83 CC                             DB 64H,83H,0CCH         ;EOM<CC, timer event mode (6-14)
0147  69 5C 4D CC                   DB 69H,5CH,4DH,0CCH     ;ETMM<5C, timer event mode (6-11)

014B  54 00 03                             DB 54H,00,03H           ;JMP to 0300, high level cold start
                                                          ;COLD WORD MOVED TO THE START OF CODE AREA.
                                                          ;ATTEMPTED TO AUTOMATE-JMP COLD-WITH $JUMP
                                                          ;BUT MACRO PRODUCES ERROR CODES.

                            ; COLD start moves the following to USER variables.
                            ; MUST BE IN SAME ORDER AS USER VARIABLES.

014E  0004[                 UZERO:          DW      4 DUP (0)               ;reserved
         0000
                      ]

0156  C1F0                                 DW      SPP                     ;SP0
0158  C2F0                                 DW      RPP                     ;RP0
015A  032E R                               DW      QRX                     ;'?KEY
015C  0347 R                               DW      TXSTO                   ;'EMIT
015E  0D34 R                               DW      ACCEP                   ;'EXPECT
0160  0D09 R                               DW      KTAP                    ;'TAP
0162  0347 R                               DW      TXSTO                   ;'ECHO
0164  0E2A R                               DW      DOTOK                   ;'PROMPT
0166  000A                                 DW      BASEE                   ;BASE
0168  0000                                 DW      0                       ;tmp
016A  0000                                 DW      0                       ;SPAN
016C  0000                                 DW      0                       ;>IN
016E  0000                                 DW      0                       ;#TIB
0170  C200                                 DW      TIBB                    ;TIB
0172  0000                                 DW      0                       ;CSP
0174  0DEE R                               DW      INTER                   ;'EVAL
0176  0963 R                               DW      NUMBQ                   ;'NUMBER
0178  0000                                 DW      0                       ;HLD
017A  0000                                 DW      0                       ;HANDLER
017C  0000                                 DW      0                       ;CONTEXT pointer
017E  0006[                                DW      VOCSS DUP (0)           ;vocabulary stack
         0000
                      ]

018A  0000                                 DW      0                       ;CURRENT pointer
018C  0000                                 DW      0                       ;vocabulary link pointer
018E  0000                                 DW      0                       ;FORTH HEAD
0190  0000                                 DW      0                       ;FORTH LINK
0192  C390                                 DW      CTOP                    ;CP
```

```
0194  C7FF                                 DW      NTOP                    ;NP
0196  3527                                 DW      LASTN                   ;LAST
0198  0000                                 DW      0                       ;SERIN host receive char & flag
019A  0006                                 DW      06H                     ;HAFBIT time for serial host,
                                                                         ; (1/2 BITIME - 5)
019C  0016                                 DW      16H                     ;BITIME baud for serial host

019E                        ULAST:

0200                        ORG 0200H

                            ;       Interrupt routine for Analog to Digital Converters

0200  10                            DB 10H                  ;EXA
0201  11                            DB 11H                  ;EXX
                            ;  Load ADC Address and Counter into HL.  Uses FFF2 and FFF3.
0202  68 FF                         DB 68H,0FFH             ;V'<FF
0204  01 F2                         DB 01H,0F2H             ;A<(V/F2)
0206  1E                            DB 1EH                  ;H<A
0207  01 F3                         DB 01H,0F3H             ;A<(V/F3)
0209  1F                            DB 1FH                  ;L<A
                            ;  Store ADC 0.
020A  2B                            DB 2BH                  ;A<(HL)
020B  57 80                         DB 57H,80H              ;A AND 80, Skip if zero
020D  CD                            DB 0CDH                 ;Jump to EXIT if slider is disabled.
020E  1A                            DB 1AH                  ;B<A
020F  4C E0                         DB 4CH,0E0H             ;A<CR0
0211  48 21                         DB 48H,21H              ;A Shift right, Midi is 7 bits, throw LSB.
0213  60 6A                         DB 60H,6AH              ;B-A, Skip if not zero
0215  C5                            DB 0C5H                 ;Jump to EXIT if slider has not changed.
0216  3D                            DB 3DH                  ;(HL)<A, Store slider data, 0 in top bit.
0217  69 FF                         DB 69H,0FFH             ;A<FF
0219  BF 38                         DB 0BFH,38H             ;(HL+38)<A, Store slider change flag.
021B  32                            DB 32H                  ;HL<HL+1, EXIT
                            ;  Store ADC 1.
021C  2B                            DB 2BH                  ;A<(HL)
021D  57 80                         DB 57H,80H              ;A AND 80, Skip if zero
021F  CD                            DB 0CDH                 ;Jump to EXIT if slider is disabled.
0220  1A                            DB 1AH                  ;B<A
0221  4C E1                         DB 4CH,0E1H             ;A<CR1
0223  48 21                         DB 48H,21H              ;A Shift right, Midi is 7 bits, throw LSB.
0225  60 6A                         DB 60H,6AH              ;B-A, Skip if not zero
0227  C5                            DB 0C5H                 ;Jump to EXIT if slider has not changed.
0228  3D                            DB 3DH                  ;(HL)<A, Store slider data, 0 in top bit.
0229  69 FF                         DB 69H,0FFH             ;A<FF
022B  BF 38                         DB 0BFH,38H             ;(HL+38)<A, Store slider change flag.
022D  32                            DB 32H                  ;HL<HL+1, EXIT
                            ;  Store ADC 2.
022E  2B                            DB 2BH                  ;A<(HL)
022F  57 80                         DB 57H,80H              ;A AND 80, Skip if zero
0231  CD                            DB 0CDH                 ;Jump to EXIT if slider is disabled.
0232  1A                            DB 1AH                  ;B<A
0233  4C E2                         DB 4CH,0E2H             ;A<CR2
0235  48 21                         DB 48H,21H              ;A Shift right, Midi is 7 bits, throw LSB.
0237  60 6A                         DB 60H,6AH              ;B-A, Skip if not zero
0239  C5                            DB 0C5H                 ;Jump to EXIT if slider has not changed.
```

```
023A  3D                            DB 3DH                   ;(HL)<A, Store slider data, 0 in top bit.
023B  69 FF                         DB 69H,0FFH              ;A<FF
023D  BF 38                         DB 0BFH,38H              ;(HL+38)<A, Store slider change flag.
023F  32                            DB 32H                   ;HL<HL+1, EXIT
                             ;  Store ADC 3.
0240  2B                            DB 2BH                   ;A<(HL)
0241  57 80                         DB 57H,80H               ;A AND 80, Skip if zero
0243  CD                            DB 0CDH                  ;Jump to EXIT if slider is disabled.
0244  1A                            DB 1AH                   ;B<A
0245  4C E3                         DB 4CH,0E3H              ;A<CR3
0247  48 21                         DB 48H,21H               ;A Shift right, Midi is 7 bits, throw LSB.
0249  60 6A                         DB 60H,6AH               ;B-A, Skip if not zero
024B  C5                            DB 0C5H                  ;Jump to EXIT if slider has not changed.
024C  3D                            DB 3DH                   ;(HL)<A, Store slider data, 0 in top bit.
024D  69 FF                         DB 69H,0FFH              ;A<FF
024F  BF 38                         DB 0BFH,38H              ;(HL+38)<A, Store slider change flag.
0251  32                            DB 32H                   ;HL<HL+1, EXIT
                             ;  Update Counters
0252  0F                            DB 0FH                   ;A<L
0253  37 37                         DB 37H,37H               ;A-37H, Skip if borrow
0255  69 00                         DB 69H,0                 ;A<0, Reset counter after 56D sliders.
0257  63 F3                         DB 63H,0F3H              ;(V/F3)<A, Load counter.
0259  48 25 48 25                   DB 48H,25H,48H,25H       ;A shift logical left 2xs.
025D  1A                            DB 1AH                   ;B<A
025E  74 0A E0                              DB 74H,0AH,0E0H         ;B<B AND E0
0261  4C C2                         DB 4CH,0C2H              ;A<Pc
0263  07 1F                         DB 07H,1FH               ;A<A AND 1F
0265  60 9A                         DB 60H,9AH               ;A<A OR B
0267  4D C2                         DB 4DH,0C2H              ;Pc<A, Load high 3 bits of slider select.
0269  64 90 08                              DB 64H,90H,08H          ;Invert ANM bit and restart conversion.
                             ;  Return from Interrupt.
026C  10                            DB 10H                   ;EXA
026D  11                            DB 11H                   ;EXX
026E  AA                            DB 0AAH                  ;EI
026F  62                            DB 62H                   ;RETI

0300                         ORG     CODEE                                   ;start code dictionary

                             ;   COLD        ( -- )
                             ;               The hilevel cold start sequence.
= 0300                                              CCOLD = $
                                            $COLON  4,'COLD',COLD
0300                        2       COLD:                                           ;
3FF3                        2       ORG     _NAME                                    ;
3FF3  0300 R 0000           2               DW       _CODE,_LINK                    ;
3FF7  04 43 4F 4C 44        2               DB      4,'COLD'                        ;
0300                        2       ORG     _CODE                                    ;
0300  80                    1               DB 80H                                  ;
0301  038D R 014E R 038D R   COLD1:          DW      DOLIT,UZERO,DOLIT,UPP
      FF00
0309  038D R 0050 0803 R                    DW      DOLIT,ULAST-UZERO,CMOVE ;initialize user area
030F  0E74 R                                DW      PRESE                   ;initialize stack and TIB
0311  12DC R 07F6 R                         DW      TBOOT,ATEXE             ;application boot
0315  0538 R 04FD R 03DE R                  DW      FORTH,CNTXT,AT,DUPP     ;initialize search order
      043B R
031D  0502 R 07BA R 10BF R                  DW      CRRNT,DSTOR,OVERT
```

```
0323  1363 R                                DW      LCDINIT                 ;initialize LCD
0325  0ECC R                                DW      QUIT                    ;start interpretation
0327  03CC R 0301 R                         DW      BRAN,COLD1              ;just in case


                            ;; Device dependent I/O

                            ;   BYE         ( -- )
                            ;               Exit eForth.
                                            $CODE   3,'BYE',BYE
032B                        1       BYE:                                         ;
3FEB                        1       ORG     _NAME                                 ;
3FEB  032B R 3FF7 R         1               DW       _CODE,_LINK                  ;
3FEF  03 42 59 45           1               DB      3,'BYE'                      ;
032B                        1       ORG     _CODE                                 ;
032B  54 00 00                                     DB 54H,0,0                  ;JMP Reset Vector

                            ;   ?RX         ( -- c T | F )
                            ;               Return input character and true, or a false if no input.

                                            $CODE   3,'?RX',QRX
032E                        1       QRX:                                         ;
3FE3                        1       ORG     _NAME                                 ;
3FE3  032E R 3FEF R         1               DW       _CODE,_LINK                  ;
3FE7  03 3F 52 58           1               DB      3,'?RX'                      ;
032E                        1       ORG     _CODE                                 ;
032E  68 FF                         DB 68H,0FFH             ;MVI, V<FF
0330  01 4B                         DB 01H,4BH              ;LDAW, A<(V/4B) read serial-in flag
0332  47 FF                         DB 47H,0FFH             ;ONI, A AND FF skip if flag not zero
0334  CA                            DB 0CAH                 ;JR, jump ahead1
0335  71 4B 00                              DB 71H,04BH,0           ;MVIW, (V/4B)<0, reset flag to zero
0338  70 1F 4A FF                   DB 70H,1FH,4AH,0FFH     ;LBCD, BC<(FF4A), read serin data
033C  B1                            DB 0B1H                 ;PUSH BC, push serial input data to stack
033D  69 FF                         DB 69H,0FFH             ;A<FF
033F  1B                            DB 1BH                  ;C<A, AHEAD1
0340  6A 00                         DB 6AH,0                ;B<0
0342  B1                            DB 0B1H                 ;PUSH BC, push serial input flag to stack
                                    $NEXT
0343  48 84                 1               DB 48H,84H                           ;
0345  48 28                 1               DB 48H,28H                           ;


                            ;   TX!         ( c -- )
                            ;               Send character c to the output device.

                                            $CODE   3,'TX!',TXSTO
0347                        1       TXSTO:                                        ;
3FDB                        1       ORG     _NAME                                 ;
3FDB  0347 R 3FE7 R         1               DW       _CODE,_LINK                  ;
3FDF  03 54 58 21           1               DB      3,'TX!'                      ;
0347                        1       ORG     _CODE                                 ;
0347  BA                            DB 0BAH                 ;Disable Interrupts
0348  A1                            DB 0A1H                 ;POP BC, pop char into C
0349  B2                            DB 0B2H                 ;PUSH DE, store interpreter pointer
034A  0B 1C                         DB 0BH,1CH              ;A<C, D<A, char in A and D
034C  68 FF                         DB 68H,0FFH             ;V<FF
```

```
034E  71 F0 07                              DB 71H,0F0H,07H         ;(V/F0)<7
0351  60 91                         DB 60H,91H              ;A<A EXOR A
0353  6D 01                         DB 6DH,01H              ;E<01
0355  4D C1                         DB 4DH,0C1H             ;PB<A
0357  70 1F 4E FF                   DB 70H,1FH,04EH,0FFH    ;BC<(FF4E) set baud, LOOP1
035B  53 FE 52 FC                   DB 53H,0FEH,52H,0FCH    ;C<C-1, JR, B<B-1, JR, jr to loop1
035F  0C                            DB 0CH                  ;A<D
0360  07 01                         DB 07H,01H              ;A<A AND 01
0362  4D C1                         DB 4DH,0C1H             ;PB<A, send a bit
0364  0C                            DB 0CH                  ;A<D
0365  48 31                         DB 48H,31H              ;A rotate logical right
0367  1C                            DB 1CH                  ;D<A
0368  00 00 00 00                   DB 0,0,0,0              ;NOPs to make rec loop = transmit loop.
036C  30 F0                         DB 30H,0F0H             ;(V/F0)<(V/F0)-1 skip
036E  E8                            DB 0E8H                 ;JR, jump to loop1
036F  0D                            DB 0DH                  ;A<E
0370  51                            DB 51H                  ;A<A-1 skip
0371  C6                            DB 0C6H                 ;JR, jump to loop2
0372  A2                            DB 0A2H                 ;POP DE, restore interpreter pointer
0373  AA                            DB 0AAH                 ;Enable Interrupts
                                    $NEXT                   ;End of routine
0374  48 84                 1               DB 48H,84H                              ;
0376  48 28                 1               DB 48H,28H                              ;

0378  6C 03                         DB 6CH,03H              ;D<03, LOOP2
037A  1D                            DB 1DH                  ;E<A
037B  71 F0 01                              DB 71H,0F0H,01          ;(V/F0)<01
037E  4F D7                         DB 4FH,0D7H             ;JRE, jump to loop1

                            ;   !IO          ( -- )
                            ;                Initialize the serial I/O devices.

                                            $CODE   3,'!IO',STOIO
0380                        1       STOIO:                                          ;
3FD3                        1       ORG     _NAME                                   ;
3FD3  0380 R 3FDF R         1               DW       _CODE,_LINK                    ;
3FD7  03 21 49 4F           1               DB      3,'!IO'                        ;
0380                        1       ORG     _CODE                                   ;
0380  69 EF 4D C7                   DB 69H,0EFH,4DH,0C7H    ;MKL<EF, enable int2 interrupt and
0384  69 FF 4D C6                   DB 69H,0FFH,4DH,0C6H    ;MKH<FF, disable all others with mask
0388  AA                            DB 0AAH                 ;EI, enable interrupt
                                    $NEXT
0389  48 84                 1               DB 48H,84H                              ;
038B  48 28                 1               DB 48H,28H                              ;


                            ;; The kernel

                            ;   doLIT        ( -- w )
                            ;                Push an inline literal.

                                            $CODE   COMPO+5,'doLIT',DOLIT
038D                        1       DOLIT:                                          ;
3FC9                        1       ORG     _NAME                                   ;
3FC9  038D R 3FD7 R         1               DW       _CODE,_LINK                    ;
3FCD  45 64 6F 4C 49 54     1               DB      COMPO+5,'doLIT'                        ;
```

```
038D                        1       ORG     _CODE                              ;
038D  48 84                                 DB 48H,84H                     ;EA<(DE)++
038F  B4                                    DB 0B4H                        ;PUSH EA
                                            $NEXT
0390  48 84                 1               DB 48H,84H                         ;
0392  48 28                 1               DB 48H,28H                         ;

                            ;   EXIT         ( -- )
                            ;                Terminate a colon definition.

                                            $CODE   4,'EXIT',EXIT
0394                        1       EXIT:                                      ;
3FBF                        1       ORG     _NAME                              ;
3FBF  0394 R 3FCD R         1               DW       _CODE,_LINK               ;
3FC3  04 45 58 49 54        1               DB      4,'EXIT'                   ;
0394                        1       ORG     _CODE                              ;
0394  48 85                                 DB 48H,85H                     ;EA<(HL)++
0396  B6                                    DB 0B6H                        ;DE<EA
                                            $NEXT
0397  48 84                 1               DB 48H,84H                         ;
0399  48 28                 1               DB 48H,28H                         ;

                            ;   EXECUTE      ( ca -- )
                            ;                Execute the word at ca.

                                            $CODE   7,'EXECUTE',EXECU
039B                        1       EXECU:                                     ;
3FB3                        1       ORG     _NAME                              ;
3FB3  039B R 3FC3 R         1               DW       _CODE,_LINK               ;
3FB7  07 45 58 45 43 55 54  1               DB      7,'EXECUTE'                   ;
039B                        1       ORG     _CODE                              ;
039B  A1                                    DB 0A1H                        ;POP BC
039C  21                                    DB 21H                         ;JMP BC

                            ;   next         ( -- )
                            ;                Run time code for the single index loop.
                            ;                : next ( -- ) \ hilevel model
                            ;                  r> r> dup if 1 - >r @ >r exit then drop cell+ >r ;

                                            $CODE   COMPO+4,'next',DONXT
039D                        1       DONXT:                                     ;
3FA9                        1       ORG     _NAME                              ;
3FA9  039D R 3FB7 R         1               DW       _CODE,_LINK               ;
3FAD  44 6E 65 78 74        1               DB      COMPO+4,'next'                ;
039D                        1       ORG     _CODE                              ;
039D  6A 00                                 DB 6AH,0                       ;B<00
039F  6B 01                                 DB 6BH,1                       ;C<01
03A1  48 83                                 DB 48H,83H                     ;EA<(HL)
03A3  74 B5                                 DB 74H,0B5H                    ;EA<EA-BC Skip if no borrow
03A5  C9                                    DB 0C9H                        ;JMP NEXT1
03A6  48 93                                 DB 48H,93H                     ;(HL)<EA
03A8  48 82                                 DB 48H,82H                     ;EA<(DE)
03AA  B6                                    DB 0B6H                        ;DE<EA
                                            $NEXT
03AB  48 84                 1               DB 48H,84H                         ;
03AD  48 28                 1               DB 48H,28H                         ;
```

```
03AF  22 22                 NEXT1:          DB 22H,22H                      ;DE<DE+2
03B1  32 32                                DB 32H,32H                      ;HL<HL+2
                                           $NEXT
03B3  48 84                 1              DB 48H,84H                              ;
03B5  48 28                 1              DB 48H,28H                              ;

                            ;   ?branch     ( f -- )
                            ;               Branch if flag is zero.

                                           $CODE   COMPO+7,'?branch',QBRAN
03B7                        1       QBRAN:                                         ;
3F9D                        1       ORG     _NAME                                  ;
3F9D  03B7 R 3FAD R         1              DW       _CODE,_LINK                   ;
3FA1  47 3F 62 72 61 6E 63  1              DB      COMPO+7,'?branch'                       ;
03B7                        1       ORG     _CODE                                  ;
03B7  6A FF                                DB 6AH,0FFH                     ;B<FF
03B9  6B FF                                DB 6BH,0FFH                     ;C<FF
03BB  A4                                   DB 0A4H                         ;POP EA
03BC  74 CD                                DB 74H,0CDH                     ;EA AND BC Skip if not zero
03BE  C6                                   DB 0C6H                         ;JMP BRAN1
03BF  22 22                                DB 22H,22H                      ;DE<DE+2
                                           $NEXT
03C1  48 84                 1              DB 48H,84H                              ;
03C3  48 28                 1              DB 48H,28H                              ;
03C5  48 82                 BRAN1:          DB 48H,82H                      ;EA<(DE)
03C7  B6                                   DB 0B6H                         ;DE<EA
                                           $NEXT
03C8  48 84                 1              DB 48H,84H                              ;
03CA  48 28                 1              DB 48H,28H                              ;

                            ;   branch      ( -- )
                            ;               Branch to an inline address.

                                           $CODE   COMPO+6,'branch',BRAN
03CC                        1       BRAN:                                         ;
3F91                        1       ORG     _NAME                                  ;
3F91  03CC R 3FA1 R         1              DW       _CODE,_LINK                   ;
3F95  46 62 72 61 6E 63 68  1              DB      COMPO+6,'branch'                       ;
03CC                        1       ORG     _CODE                                  ;
03CC  48 82                                DB 48H,82H                      ;EA<(DE)
03CE  B6                                   DB 0B6H                         ;DE<EA
                                           $NEXT
03CF  48 84                 1              DB 48H,84H                              ;
03D1  48 28                 1              DB 48H,28H                              ;

                            ;   !           ( w a -- )
                            ;               Pop the data stack to memory.

                                           $CODE   1,'!',STORE
03D3                        1       STORE:                                         ;
3F8B                        1       ORG     _NAME                                  ;
3F8B  03D3 R 3F95 R         1              DW       _CODE,_LINK                   ;
3F8F  01 21                 1              DB      1,'!'                       ;
03D3                        1       ORG     _CODE                                  ;
03D3  A1                                   DB 0A1H                         ;POP BC, address
03D4  A4                                   DB 0A4H                         ;POP EA, data
```

```
03D5  09                                    DB 09H                          ;A<EAL
03D6  39                                    DB 39H                          ;(BC)<A
03D7  12                                    DB 12H                          ;BC<BC+1
03D8  08                                    DB 08H                          ;A<EAH
03D9  39                                    DB 39H                          ;(BC)<A
                                            $NEXT
03DA  48 84                 1               DB 48H,84H                              ;
03DC  48 28                 1               DB 48H,28H                              ;

                            ;   @            ( a -- w )
                            ;                Push data at memory location to the data stack.

                                            $CODE   1,'@',AT
03DE                        1       AT:                                          ;
3F85                        1       ORG      _NAME                                  ;
3F85  03DE R 3F8F R         1               DW       _CODE,_LINK                    ;
3F89  01 40                 1               DB      1,'@'                       ;
03DE                        1       ORG      _CODE                                  ;
03DE  A1                                    DB 0A1H                         ;POP BC
03DF  29                                    DB 29H                          ;A<(BC)
03E0  19                                    DB 19H                          ;EAL<A
03E1  12                                    DB 12H                          ;BC<BC+1
03E2  29                                    DB 29H                          ;A<(BC)
03E3  18                                    DB 18H                          ;EAH<A
03E4  B4                                    DB 0B4H                         ;PUSH EA
                                            $NEXT
03E5  48 84                 1               DB 48H,84H                              ;
03E7  48 28                 1               DB 48H,28H                              ;

                            ;   C!           ( c b -- )
                            ;                Pop the data stack to byte memory.

                                            $CODE   2,'C!',CSTOR
03E9                        1       CSTOR:                                              ;
3F7D                        1       ORG      _NAME                                  ;
3F7D  03E9 R 3F89 R         1               DW       _CODE,_LINK                    ;
3F81  02 43 21              1               DB      2,'C!'                       ;
03E9                        1       ORG      _CODE                                  ;
03E9  A1                                    DB 0A1H                         ;POP BC address
03EA  A4                                    DB 0A4H                         ;POP AE data
03EB  09                                    DB 09H                          ;A<EAL
03EC  39                                    DB 39H                          ;(BC)<A
                                            $NEXT
03ED  48 84                 1               DB 48H,84H                              ;
03EF  48 28                 1               DB 48H,28H                              ;

                            ;   C@           ( b -- c )
                            ;                Push byte memory location to the data stack.

                                            $CODE   2,'C@',CAT
03F1                        1       CAT:                                          ;
3F75                        1       ORG      _NAME                                  ;
3F75  03F1 R 3F81 R         1               DW       _CODE,_LINK                    ;
3F79  02 43 40              1               DB      2,'C@'                       ;
03F1                        1       ORG      _CODE                                  ;
03F1  A1                                    DB 0A1H                         ;POP BC
```

```
03F2  29                                   DB 29H                           ;A<(BC)
03F3  6A 00                                DB 6AH,0                         ;B<00
03F5  1B                                   DB 1BH                           ;C<A
03F6  B1                                   DB 0B1H                          ;PUSH BC
                                           $NEXT
03F7  48 84                 1              DB 48H,84H                               ;
03F9  48 28                 1              DB 48H,28H                               ;

                            ;   RP@         ( -- a )
                            ;               Push the current RP to the data stack.

                                           $CODE   3,'RP@',RPAT
03FB                        1       RPAT:                                          ;
3F6D                        1       ORG     _NAME                                   ;
3F6D  03FB R 3F79 R         1              DW       _CODE,_LINK                    ;
3F71  03 52 50 40           1              DB      3,'RP@'                        ;
03FB                        1       ORG     _CODE                                   ;
03FB  B3                                   DB 0B3H                          ;PUSH HL
                                           $NEXT
03FC  48 84                 1              DB 48H,84H                               ;
03FE  48 28                 1              DB 48H,28H                               ;

                            ;   RP!         ( a -- )
                            ;               Set the return stack pointer.

                                           $CODE   COMPO+3,'RP!',RPSTO
0400                        1       RPSTO:                                          ;
3F65                        1       ORG     _NAME                                   ;
3F65  0400 R 3F71 R         1              DW       _CODE,_LINK                    ;
3F69  43 52 50 21           1              DB      COMPO+3,'RP!'                           ;
0400                        1       ORG     _CODE                                   ;
0400  A3                                   DB 0A3H                          ;POP HL
                                           $NEXT
0401  48 84                 1              DB 48H,84H                               ;
0403  48 28                 1              DB 48H,28H                               ;

                            ;   R>          ( -- w )
                            ;               Pop the return stack to the data stack.

                                           $CODE   2,'R>',RFROM
0405                        1       RFROM:                                          ;
3F5D                        1       ORG     _NAME                                   ;
3F5D  0405 R 3F69 R         1              DW       _CODE,_LINK                    ;
3F61  02 52 3E              1              DB      2,'R>'                        ;
0405                        1       ORG     _CODE                                   ;
0405  48 85                                DB 48H,85H                       ;EA<(HL)++
0407  B4                                   DB 0B4H                          ;PUSH EA
                                           $NEXT
0408  48 84                 1              DB 48H,84H                               ;
040A  48 28                 1              DB 48H,28H                               ;

                            ;   R@          ( -- w )
                            ;               Copy top of return stack to the data stack.

                                           $CODE   2,'R@',RAT
040C                        1       RAT:                                          ;
```

```
3F55                        1       ORG     _NAME                               ;
3F55  040C R 3F61 R         1               DW       _CODE,_LINK                ;
3F59  02 52 40              1               DB      2,'R@'                    ;
040C                        1       ORG     _CODE                               ;
040C  48 83                                 DB 48H,83H                  ;EA<(HL)
040E  B4                                    DB 0B4H                     ;PUSH EA
                                            $NEXT
040F  48 84                 1               DB 48H,84H                          ;
0411  48 28                 1               DB 48H,28H                          ;

                            ;   >R           ( w -- )
                            ;                Push the data stack to the return stack.

                                            $CODE   COMPO+2,'>R',TOR
0413                        1       TOR:                                      ;
3F4D                        1       ORG     _NAME                               ;
3F4D  0413 R 3F59 R         1               DW       _CODE,_LINK                ;
3F51  42 3E 52              1               DB      COMPO+2,'>R'                       ;
0413                        1       ORG     _CODE                               ;
0413  33 33                                 DB 33H,33H                  ;HL<HL-2
0415  A4                                    DB 0A4H                     ;POP EA
0416  48 93                                 DB 48H,93H                  ;(HL)<EA
                                            $NEXT
0418  48 84                 1               DB 48H,84H                          ;
041A  48 28                 1               DB 48H,28H                          ;

                            ;   SP@          ( -- a )
                            ;                Push the current data stack pointer.

                                            $CODE   3,'SP@',SPAT
041C                        1       SPAT:                                      ;
3F45                        1       ORG     _NAME                               ;
3F45  041C R 3F51 R         1               DW       _CODE,_LINK                ;
3F49  03 53 50 40           1               DB      3,'SP@'                    ;
041C                        1       ORG     _CODE                               ;
041C  70 0E FE FF                           DB 70H,0EH,0FEH,0FFH        ;(FFFE)<SP
0420  70 1F FE FF                           DB 70H,1FH,0FEH,0FFH        ;BC<(FFFE)
0424  B1                                    DB 0B1H                     ;PUSH BC
                                            $NEXT
0425  48 84                 1               DB 48H,84H                          ;
0427  48 28                 1               DB 48H,28H                          ;

                            ;   SP!          ( a -- )
                            ;                Set the data stack pointer.

                                            $CODE   3,'SP!',SPSTO
0429                        1       SPSTO:                                      ;
3F3D                        1       ORG     _NAME                               ;
3F3D  0429 R 3F49 R         1               DW       _CODE,_LINK                ;
3F41  03 53 50 21           1               DB      3,'SP!'                    ;
0429                        1       ORG     _CODE                               ;
0429  A1                                    DB 0A1H                     ;POP BC
042A  70 1E FE FF                           DB 70H,1EH,0FEH,0FFH        ;(FFFE)<BC
042E  70 0F FE FF                           DB 70H,0FH,0FEH,0FFH        ;PC<(FFFE)
                                            $NEXT
0432  48 84                 1               DB 48H,84H                          ;
```

```
0434  48 28                 1              DB 48H,28H                              ;

                            ;   DROP        ( w -- )
                            ;               Discard top stack item.

                                           $CODE   4,'DROP',DROP
0436                        1       DROP:                                           ;
3F33                        1       ORG     _NAME                                    ;
3F33  0436 R 3F41 R         1              DW       _CODE,_LINK                     ;
3F37  04 44 52 4F 50        1              DB      4,'DROP'                         ;
0436                        1       ORG     _CODE                                    ;
0436  A4                                   DB 0A4H                         ;POP EA
                                           $NEXT
0437  48 84                 1              DB 48H,84H                              ;
0439  48 28                 1              DB 48H,28H                              ;

                            ;   DUP         ( w -- w w )
                            ;               Duplicate the top stack item.

                                           $CODE   3,'DUP',DUPP
043B                        1       DUPP:                                           ;
3F2B                        1       ORG     _NAME                                    ;
3F2B  043B R 3F37 R         1              DW       _CODE,_LINK                     ;
3F2F  03 44 55 50           1              DB      3,'DUP'                         ;
043B                        1       ORG     _CODE                                    ;
043B  A4                                   DB 0A4H                         ;POP EA
043C  B4                                   DB 0B4H                         ;PUSH EA
043D  B4                                   DB 0B4H                         ;PUSH EA
                                           $NEXT
043E  48 84                 1              DB 48H,84H                              ;
0440  48 28                 1              DB 48H,28H                              ;

                            ;   SWAP        ( w1 w2 -- w2 w1 )
                            ;               Exchange top two stack items.

                                           $CODE   4,'SWAP',SWAP
0442                        1       SWAP:                                           ;
3F21                        1       ORG     _NAME                                    ;
3F21  0442 R 3F2F R         1              DW       _CODE,_LINK                     ;
3F25  04 53 57 41 50        1              DB      4,'SWAP'                         ;
0442                        1       ORG     _CODE                                    ;
0442  A4                                   DB 0A4H                         ;POP EA
0443  A1                                   DB 0A1H                         ;POP BC
0444  B4                                   DB 0B4H                         ;PUSH EA
0445  B1                                   DB 0B1H                         ;PUSH BC
                                           $NEXT
0446  48 84                 1              DB 48H,84H                              ;
0448  48 28                 1              DB 48H,28H                              ;

                            ;   OVER        ( w1 w2 -- w1 w2 w1 )
                            ;               Copy second stack item to top.

                                           $CODE   4,'OVER',OVER
044A                        1       OVER:                                           ;
3F17                        1       ORG     _NAME                                    ;
3F17  044A R 3F25 R         1              DW       _CODE,_LINK                     ;
```

```
3F1B  04 4F 56 45 52        1              DB      4,'OVER'                        ;
044A                        1       ORG     _CODE                                    ;
044A  A4                                   DB 0A4H                     ;POP AE
044B  A1                                   DB 0A1H                     ;POP BC
044C  B1                                   DB 0B1H                     ;PUSH BC
044D  B4                                   DB 0B4H                     ;PUSH AE
044E  B1                                   DB 0B1H                     ;PUSH BC
                                           $NEXT
044F  48 84                 1              DB 48H,84H                              ;
0451  48 28                 1              DB 48H,28H                              ;

                            ;   0<          ( n -- t )
                            ;               Return true if n is negative.

                                           $CODE   2,'0<',ZLESS
0453                        1       ZLESS:                                           ;
3F0F                        1       ORG     _NAME                                    ;
3F0F  0453 R 3F1B R         1              DW       _CODE,_LINK                    ;
3F13  02 30 3C              1              DB      2,'0<'                        ;
0453                        1       ORG     _CODE                                    ;
0453  A1                                   DB 0A1H                     ;POP BC
0454  69 FF                                DB 69H,0FFH                 ;A<FF
0456  48 06                                DB 48H,06H                  ;B Shift Left, Skip if carry
0458  69 00                                DB 69H,0                    ;A<00
045A  1A                                   DB 1AH                      ;B<A
045B  1B                                   DB 1BH                      ;C<A
045C  B1                                   DB 0B1H                     ;PUSH BC
                                           $NEXT
045D  48 84                 1              DB 48H,84H                              ;
045F  48 28                 1              DB 48H,28H                              ;

                            ;   AND         ( w w -- w )
                            ;               Bitwise AND.

                                           $CODE   3,'AND',ANDD
0461                        1       ANDD:                                           ;
3F07                        1       ORG     _NAME                                    ;
3F07  0461 R 3F13 R         1              DW       _CODE,_LINK                    ;
3F0B  03 41 4E 44           1              DB      3,'AND'                        ;
0461                        1       ORG     _CODE                                    ;
0461  A1                                   DB 0A1H                     ;POP BC
0462  A4                                   DB 0A4H                     ;POP AE
0463  74 8D                                DB 74H,8DH                  ;EA<EA AND BC
0465  B4                                   DB 0B4H                     ;PUSH EA
                                           $NEXT
0466  48 84                 1              DB 48H,84H                              ;
0468  48 28                 1              DB 48H,28H                              ;

                            ;   OR          ( w w -- w )
                            ;               Bitwise inclusive OR.

                                           $CODE   2,'OR',ORR
046A                        1       ORR:                                           ;
3EFF                        1       ORG     _NAME                                    ;
3EFF  046A R 3F0B R         1              DW       _CODE,_LINK                    ;
3F03  02 4F 52              1              DB      2,'OR'                        ;
```

```
046A                        1       ORG     _CODE                                    ;
046A  A1                                   DB 0A1H                          ;POP BC
046B  A4                                   DB 0A4H                          ;POP EA
046C  74 9D                                DB 74H,9DH                       ;EA<EA OR BC
046E  B4                                   DB 0B4H                           ;PUSH EA
                                           $NEXT
046F  48 84                 1              DB 48H,84H                              ;
0471  48 28                 1              DB 48H,28H                              ;

                            ;   XOR         ( w w -- w )
                            ;               Bitwise exclusive OR.

                                           $CODE   3,'XOR',XORR
0473                        1       XORR:                                         ;
3EF7                        1       ORG     _NAME                                  ;
3EF7  0473 R 3F03 R         1              DW       _CODE,_LINK                   ;
3EFB  03 58 4F 52           1              DB      3,'XOR'                       ;
0473                        1       ORG     _CODE                                  ;
0473  A1                                   DB 0A1H                          ;POP BC
0474  A4                                   DB 0A4H                          ;POP EA
0475  74 95                                DB 74H,95H                       ;EA<EA EX-OR BC
0477  B4                                   DB 0B4H                          ;PUSH EA
                                           $NEXT
0478  48 84                 1              DB 48H,84H                              ;
047A  48 28                 1              DB 48H,28H                              ;

                            ;   UM+         ( w w -- w cy )
                            ;               Add two numbers, return the sum and carry flag.

                                           $CODE   3,'UM+',UPLUS
047C                        1       UPLUS:                                         ;
3EEF                        1       ORG     _NAME                                  ;
3EEF  047C R 3EFB R         1              DW       _CODE,_LINK                   ;
3EF3  03 55 4D 2B           1              DB      3,'UM+'                       ;
047C                        1       ORG     _CODE                                  ;
047C  A1                                   DB 0A1H                          ;POP BC
047D  A4                                   DB 0A4H                          ;POP EA
047E  69 00                                DB 69H,0                         ;A<00
0480  74 A5                                DB 74H,0A5H                      ;EA<EA+BC Skip if no carry
0482  41                                   DB 41H                           ;A<A+1
0483  1B                                   DB 1BH                           ;C<A
0484  6A 00                                DB 6AH,0                         ;B<00
0486  B4                                   DB 0B4H                          ;PUSH EA
0487  B1                                   DB 0B1H                          ;PUSH BC
                                           $NEXT
0488  48 84                 1              DB 48H,84H                              ;
048A  48 28                 1              DB 48H,28H                              ;

                            ;; System and user variables

                            ;   doVAR       ( -- a )
                            ;               Run time routine for VARIABLE and CREATE.

                                           $COLON  COMPO+5,'doVAR',DOVAR
048C                        2       DOVAR:                                         ;
3EE5                        2       ORG     _NAME                                  ;
```

```
3EE5  048C R 3EF3 R         2              DW       _CODE,_LINK                    ;
3EE9  45 64 6F 56 41 52     2              DB      COMPO+5,'doVAR'                        ;
048C                        2       ORG     _CODE                                   ;
048C  80                    1              DB 80H                                  ;
048D  0405 R 0394 R                        DW      RFROM,EXIT

                            ;   UP          ( -- a )
                            ;               Pointer to the user area.

                                           $COLON  2,'UP',UP
0491                        2       UP:                                          ;
3EDD                        2       ORG     _NAME                                   ;
3EDD  0491 R 3EE9 R         2              DW       _CODE,_LINK                    ;
3EE1  02 55 50              2              DB      2,'UP'                        ;
0491                        2       ORG     _CODE                                   ;
0491  80                    1              DB 80H                                  ;
0492  048C R                               DW      DOVAR
0494  FF00                                 DW      UPP

                            ;   doUSER      ( -- a )
                            ;               Run time routine for user variables.

                                           $COLON  COMPO+6,'doUSER',DOUSE
0496                        2       DOUSE:                                          ;
3ED1                        2       ORG     _NAME                                   ;
3ED1  0496 R 3EE1 R         2              DW       _CODE,_LINK                    ;
3ED5  46 64 6F 55 53 45 52  2              DB      COMPO+6,'doUSER'                        ;
0496                        2       ORG     _CODE                                   ;
0496  80                    1              DB 80H                                  ;
0497  0405 R 03DE R 0491 R                 DW      RFROM,AT,UP,AT,PLUS,EXIT
      03DE R 0565 R 0394 R

                            ;   SP0         ( -- a )
                            ;               Pointer to bottom of the data stack.

                                           $USER   3,'SP0',SZERO
04A3                        2       SZERO:                                          ;
3EC9                        2       ORG     _NAME                                   ;
3EC9  04A3 R 3ED5 R         2              DW       _CODE,_LINK                    ;
3ECD  03 53 50 30           2              DB      3,'SP0'                        ;
04A3                        2       ORG     _CODE                                   ;
04A3  80                    1              DB 80H                                  ;
04A4  0496 R 0008           1              DW      DOUSE,_USER                     ;

                            ;   RP0         ( -- a )
                            ;               Pointer to bottom of the return stack.

                                           $USER   3,'RP0',RZERO
04A8                        2       RZERO:                                          ;
3EC1                        2       ORG     _NAME                                   ;
3EC1  04A8 R 3ECD R         2              DW       _CODE,_LINK                    ;
3EC5  03 52 50 30           2              DB      3,'RP0'                        ;
04A8                        2       ORG     _CODE                                   ;
04A8  80                    1              DB 80H                                  ;
04A9  0496 R 000A           1              DW      DOUSE,_USER                     ;

```

```
                            ;    '?KEY       ( -- a )
                            ;                Execution vector of ?KEY.

                                            $USER   5,"'?KEY",TQKEY
04AD                        2       TQKEY:                                          ;
3EB7                        2       ORG     _NAME                                   ;
3EB7  04AD R 3EC5 R         2               DW       _CODE,_LINK                    ;
3EBB  05 27 3F 4B 45 59     2               DB      5,"'?KEY"                        ;
04AD                        2       ORG     _CODE                                   ;
04AD  80                    1               DB 80H                                  ;
04AE  0496 R 000C           1               DW      DOUSE,_USER                     ;

                            ;    'EMIT       ( -- a )
                            ;                Execution vector of EMIT.

                                            $USER   5,"'EMIT",TEMIT
04B2                        2       TEMIT:                                          ;
3EAD                        2       ORG     _NAME                                   ;
3EAD  04B2 R 3EBB R         2               DW       _CODE,_LINK                    ;
3EB1  05 27 45 4D 49 54     2               DB      5,"'EMIT"                        ;
04B2                        2       ORG     _CODE                                   ;
04B2  80                    1               DB 80H                                  ;
04B3  0496 R 000E           1               DW      DOUSE,_USER                     ;

                            ;    'EXPECT     ( -- a )
                            ;                Execution vector of EXPECT.

                                            $USER   7,"'EXPECT",TEXPE
04B7                        2       TEXPE:                                          ;
3EA1                        2       ORG     _NAME                                   ;
3EA1  04B7 R 3EB1 R         2               DW       _CODE,_LINK                    ;
3EA5  07 27 45 58 50 45 43  2               DB      7,"'EXPECT"                        ;
04B7                        2       ORG     _CODE                                   ;
04B7  80                    1               DB 80H                                  ;
04B8  0496 R 0010           1               DW      DOUSE,_USER                     ;

                            ;    'TAP        ( -- a )
                            ;                Execution vector of TAP.

                                            $USER   4,"'TAP",TTAP
04BC                        2       TTAP:                                          ;
3E97                        2       ORG     _NAME                                   ;
3E97  04BC R 3EA5 R         2               DW       _CODE,_LINK                    ;
3E9B  04 27 54 41 50        2               DB      4,"'TAP"                        ;
04BC                        2       ORG     _CODE                                   ;
04BC  80                    1               DB 80H                                  ;
04BD  0496 R 0012           1               DW      DOUSE,_USER                     ;

                            ;    'ECHO       ( -- a )
                            ;                Execution vector of ECHO.

                                            $USER   5,"'ECHO",TECHO
04C1                        2       TECHO:                                          ;
3E8D                        2       ORG     _NAME                                   ;
3E8D  04C1 R 3E9B R         2               DW       _CODE,_LINK                    ;
3E91  05 27 45 43 48 4F     2               DB      5,"'ECHO"                        ;
```

```
04C1                         2       ORG     _CODE                                    ;
04C1  80                     1             DB 80H                                    ;
04C2  0496 R 0014            1             DW      DOUSE,_USER                       ;

                             ;   'PROMPT     ( -- a )
                             ;               Execution vector of PROMPT.

                                           $USER   7,"'PROMPT",TPROM
04C6                         2       TPROM:                                           ;
3E81                         2       ORG     _NAME                                    ;
3E81  04C6 R 3E91 R          2             DW       _CODE,_LINK                      ;
3E85  07 27 50 52 4F 4D 50   2             DB      7,"'PROMPT"                          ;
04C6                         2       ORG     _CODE                                    ;
04C6  80                     1             DB 80H                                    ;
04C7  0496 R 0016            1             DW      DOUSE,_USER                       ;

                             ;   BASE        ( -- a )
                             ;               Storage of the radix base for numeric I/O.

                                           $USER   4,'BASE',BASE
04CB                         2       BASE:                                           ;
3E77                         2       ORG     _NAME                                    ;
3E77  04CB R 3E85 R          2             DW       _CODE,_LINK                      ;
3E7B  04 42 41 53 45         2             DB      4,'BASE'                          ;
04CB                         2       ORG     _CODE                                    ;
04CB  80                     1             DB 80H                                    ;
04CC  0496 R 0018            1             DW      DOUSE,_USER                       ;

                             ;   tmp         ( -- a )
                             ;               A temporary storage location used in parse and find.

                                           $USER   COMPO+3,'tmp',TEMP
04D0                         2       TEMP:                                           ;
3E6F                         2       ORG     _NAME                                    ;
3E6F  04D0 R 3E7B R          2             DW       _CODE,_LINK                      ;
3E73  43 74 6D 70            2             DB      COMPO+3,'tmp'                          ;
04D0                         2       ORG     _CODE                                    ;
04D0  80                     1             DB 80H                                    ;
04D1  0496 R 001A            1             DW      DOUSE,_USER                       ;

                             ;   SPAN        ( -- a )
                             ;               Hold character count received by EXPECT.

                                           $USER   4,'SPAN',SPAN
04D5                         2       SPAN:                                           ;
3E65                         2       ORG     _NAME                                    ;
3E65  04D5 R 3E73 R          2             DW       _CODE,_LINK                      ;
3E69  04 53 50 41 4E         2             DB      4,'SPAN'                          ;
04D5                         2       ORG     _CODE                                    ;
04D5  80                     1             DB 80H                                    ;
04D6  0496 R 001C            1             DW      DOUSE,_USER                       ;

                             ;   >IN         ( -- a )
                             ;               Hold the character pointer while parsing input stream.

                                           $USER   3,'>IN',INN
```

```
04DA                        2       INN:                                        ;
3E5D                        2       ORG     _NAME                                ;
3E5D  04DA R 3E69 R         2               DW       _CODE,_LINK                 ;
3E61  03 3E 49 4E           2               DB      3,'>IN'                     ;
04DA                        2       ORG     _CODE                                ;
04DA  80                    1               DB 80H                              ;
04DB  0496 R 001E           1               DW      DOUSE,_USER                 ;

                            ;   #TIB        ( -- a )
                            ;                Hold the current count and address of the terminal input buffer.

                                            $USER   4,'#TIB',NTIB
04DF                        2       NTIB:                                        ;
3E53                        2       ORG     _NAME                                ;
3E53  04DF R 3E61 R         2               DW       _CODE,_LINK                 ;
3E57  04 23 54 49 42        2               DB      4,'#TIB'                    ;
04DF                        2       ORG     _CODE                                ;
04DF  80                    1               DB 80H                              ;
04E0  0496 R 0020           1               DW      DOUSE,_USER                 ;
= 0024                                              _USER = _USER+CELLL

                            ;   CSP         ( -- a )
                            ;                Hold the stack pointer for error checking.

                                            $USER   3,'CSP',CSP
04E4                        2       CSP:                                        ;
3E4B                        2       ORG     _NAME                                ;
3E4B  04E4 R 3E57 R         2               DW       _CODE,_LINK                 ;
3E4F  03 43 53 50           2               DB      3,'CSP'                     ;
04E4                        2       ORG     _CODE                                ;
04E4  80                    1               DB 80H                              ;
04E5  0496 R 0024           1               DW      DOUSE,_USER                 ;

                            ;   'EVAL       ( -- a )
                            ;                Execution vector of EVAL.

                                            $USER   5,"'EVAL",TEVAL
04E9                        2       TEVAL:                                       ;
3E41                        2       ORG     _NAME                                ;
3E41  04E9 R 3E4F R         2               DW       _CODE,_LINK                 ;
3E45  05 27 45 56 41 4C     2               DB      5,"'EVAL"                    ;
04E9                        2       ORG     _CODE                                ;
04E9  80                    1               DB 80H                              ;
04EA  0496 R 0026           1               DW      DOUSE,_USER                 ;

                            ;   'NUMBER     ( -- a )
                            ;                Execution vector of NUMBER?.

                                            $USER   7,"'NUMBER",TNUMB
04EE                        2       TNUMB:                                       ;
3E35                        2       ORG     _NAME                                ;
3E35  04EE R 3E45 R         2               DW       _CODE,_LINK                 ;
3E39  07 27 4E 55 4D 42 45  2               DB      7,"'NUMBER"                     ;
04EE                        2       ORG     _CODE                                ;
04EE  80                    1               DB 80H                              ;
04EF  0496 R 0028           1               DW      DOUSE,_USER                 ;
```

```

                                ;   HLD         ( -- a )
                                ;               Hold a pointer in building a numeric output string.

                                               $USER   3,'HLD',HLD
04F3                            2       HLD:                                             ;
3E2D                            2       ORG     _NAME                                    ;
3E2D  04F3 R 3E39 R             2               DW       _CODE,_LINK                    ;
3E31  03 48 4C 44               2               DB      3,'HLD'                        ;
04F3                            2       ORG     _CODE                                    ;
04F3  80                        1               DB 80H                                  ;
04F4  0496 R 002A               1               DW      DOUSE,_USER                     ;

                                ;   HANDLER     ( -- a )
                                ;               Hold the return stack pointer for error handling.

                                               $USER   7,'HANDLER',HANDL
04F8                            2       HANDL:                                           ;
3E21                            2       ORG     _NAME                                    ;
3E21  04F8 R 3E31 R             2               DW       _CODE,_LINK                    ;
3E25  07 48 41 4E 44 4C 45  2               DB      7,'HANDLER'                        ;
04F8                            2       ORG     _CODE                                    ;
04F8  80                        1               DB 80H                                  ;
04F9  0496 R 002C               1               DW      DOUSE,_USER                     ;

                                ;   CONTEXT     ( -- a )
                                ;               A area to specify vocabulary search order.

                                               $USER   7,'CONTEXT',CNTXT
04FD                            2       CNTXT:                                           ;
3E15                            2       ORG     _NAME                                    ;
3E15  04FD R 3E25 R             2               DW       _CODE,_LINK                    ;
3E19  07 43 4F 4E 54 45 58  2               DB      7,'CONTEXT'                        ;
04FD                            2       ORG     _CODE                                    ;
04FD  80                        1               DB 80H                                  ;
04FE  0496 R 002E               1               DW      DOUSE,_USER                     ;
= 003C                                                 _USER = _USER+VOCSS*CELLL       ;vocabulary stack

                                ;   CURRENT     ( -- a )
                                ;               Point to the vocabulary to be extended.

                                               $USER   7,'CURRENT',CRRNT
0502                            2       CRRNT:                                           ;
3E09                            2       ORG     _NAME                                    ;
3E09  0502 R 3E19 R             2               DW       _CODE,_LINK                    ;
3E0D  07 43 55 52 52 45 4E  2               DB      7,'CURRENT'                        ;
0502                            2       ORG     _CODE                                    ;
0502  80                        1               DB 80H                                  ;
0503  0496 R 003C               1               DW      DOUSE,_USER                     ;
= 0040                                                 _USER = _USER+CELLL             ;vocabulary link pointer

                                ;   FHEAD       ( -- a )
                                ;               Point to the FORTH vocab head pointer.
                                               $USER   5,'FHEAD',FHEAD
0507                            2       FHEAD:                                           ;
3DFF                            2       ORG     _NAME                                    ;
```

```
3DFF  0507 R 3E0D R         2               DW       _CODE,_LINK                    ;
3E03  05 46 48 45 41 44     2               DB      5,'FHEAD'                        ;
0507                        2       ORG     _CODE                                   ;
0507  80                    1               DB 80H                                  ;
0508  0496 R 0040           1               DW      DOUSE,_USER                     ;

                            ;   FLINK       ( -- a )
                            ;               Point to the FORTH vocab link pointer.
                                            $USER   5,'FLINK',FLINK
050C                        2       FLINK:                                          ;
3DF5                        2       ORG     _NAME                                   ;
3DF5  050C R 3E03 R         2               DW       _CODE,_LINK                    ;
3DF9  05 46 4C 49 4E 4B     2               DB      5,'FLINK'                        ;
050C                        2       ORG     _CODE                                   ;
050C  80                    1               DB 80H                                  ;
050D  0496 R 0042           1               DW      DOUSE,_USER                     ;

                            ;   CP          ( -- a )
                            ;               Point to the top of the code dictionary.

                                            $USER   2,'CP',CP
0511                        2       CP:                                         ;
3DED                        2       ORG     _NAME                                   ;
3DED  0511 R 3DF9 R         2               DW       _CODE,_LINK                    ;
3DF1  02 43 50              2               DB      2,'CP'                      ;
0511                        2       ORG     _CODE                                   ;
0511  80                    1               DB 80H                                  ;
0512  0496 R 0044           1               DW      DOUSE,_USER                     ;

                            ;   NP          ( -- a )
                            ;               Point to the bottom of the name dictionary.

                                            $USER   2,'NP',NP
0516                        2       NP:                                         ;
3DE5                        2       ORG     _NAME                                   ;
3DE5  0516 R 3DF1 R         2               DW       _CODE,_LINK                    ;
3DE9  02 4E 50              2               DB      2,'NP'                      ;
0516                        2       ORG     _CODE                                   ;
0516  80                    1               DB 80H                                  ;
0517  0496 R 0046           1               DW      DOUSE,_USER                     ;

                            ;   LAST        ( -- a )
                            ;               Point to the last name in the name dictionary.

                                            $USER   4,'LAST',LAST
051B                        2       LAST:                                           ;
3DDB                        2       ORG     _NAME                                   ;
3DDB  051B R 3DE9 R         2               DW       _CODE,_LINK                    ;
3DDF  04 4C 41 53 54        2               DB      4,'LAST'                        ;
051B                        2       ORG     _CODE                                   ;
051B  80                    1               DB 80H                                  ;
051C  0496 R 0048           1               DW      DOUSE,_USER                     ;

                            ;   SERIN       ( -- a )
                            ;               Point to host serial input. Flag in high, char in low byte.

```

```
                                            $USER   5,'SERIN',SERIN
0520                         2       SERIN:                                          ;
3DD1                         2       ORG     _NAME                                   ;
3DD1  0520 R 3DDF R          2              DW       _CODE,_LINK                    ;
3DD5  05 53 45 52 49 4E      2              DB      5,'SERIN'                        ;
0520                         2       ORG     _CODE                                   ;
0520  80                     1              DB 80H                                  ;
0521  0496 R 004A            1              DW      DOUSE,_USER                     ;

                             ;   HAFBIT      ( -- a )
                             ;               Point to half bit time used by serial i/0 routines.

                                            $USER   6,'HAFBIT',HAFBIT
0525                         2       HAFBIT:                                          ;
3DC5                         2       ORG     _NAME                                   ;
3DC5  0525 R 3DD5 R          2              DW       _CODE,_LINK                    ;
3DC9  06 48 41 46 42 49 54  2              DB      6,'HAFBIT'                         ;
0525                         2       ORG     _CODE                                   ;
0525  80                     1              DB 80H                                  ;
0526  0496 R 004C            1              DW      DOUSE,_USER                     ;

                             ;   BITIME      ( -- a )
                             ;               Point to bit time used to set serial i/o baud rate.

                                            $USER   6,'BITIME',BITIME
052A                         2       BITIME:                                          ;
3DB9                         2       ORG     _NAME                                   ;
3DB9  052A R 3DC9 R          2              DW       _CODE,_LINK                    ;
3DBD  06 42 49 54 49 4D 45  2              DB      6,'BITIME'                         ;
052A                         2       ORG     _CODE                                   ;
052A  80                     1              DB 80H                                  ;
052B  0496 R 004E            1              DW      DOUSE,_USER                     ;

                             ;; Common functions

                             ;   doVOC       ( -- )
                             ;               Run time action of VOCABULARY's.

                                            $COLON  COMPO+5,'doVOC',DOVOC
052F                         2       DOVOC:                                          ;
3DAF                         2       ORG     _NAME                                   ;
3DAF  052F R 3DBD R          2              DW       _CODE,_LINK                    ;
3DB3  45 64 6F 56 4F 43      2              DB      COMPO+5,'doVOC'                         ;
052F                         2       ORG     _CODE                                   ;
052F  80                     1              DB 80H                                  ;
0530  0507 R 04FD R 03D3 R                  DW      FHEAD,CNTXT,STORE,EXIT
      0394 R

                             ;   FORTH       ( -- )
                             ;               Make FORTH the context vocabulary.

                                            $COLON  5,'FORTH',FORTH
0538                         2       FORTH:                                          ;
3DA5                         2       ORG     _NAME                                   ;
3DA5  0538 R 3DB3 R          2              DW       _CODE,_LINK                    ;
3DA9  05 46 4F 52 54 48      2              DB      5,'FORTH'                        ;
```

```
0538                       2       ORG     _CODE                                        ;
0538  80                   1               DB 80H                                       ;
0539  052F R 0394 R                        DW      DOVOC,EXIT
                                           ; Head and Link pointers normally here were moved to User Ram.

                           ;   ?DUP        ( w -- w w | 0 )
                           ;               Dup tos if its is not zero.

                                           $CODE  4,'?DUP',QDUP
053D                       1       QDUP:                                                ;
3D9B                       1       ORG     _NAME                                        ;
3D9B  053D R 3DA9 R        1               DW       _CODE,_LINK                         ;
3D9F  04 3F 44 55 50       1               DB      4,'?DUP'                             ;
053D                       1       ORG     _CODE                                        ;
053D  6A FF                                DB 6AH,0FFH             ;B<FF
053F  6B FF                                DB 6BH,0FFH             ;C<FF
0541  A4                                   DB 0A4H                 ;POP EA
0542  74 DD                                DB 74H,0DDH             ;EA AND BC, Skip if zero
0544  B4                                   DB 0B4H                 ;PUSH EA
0545  B4                                   DB 0B4H                 ;PUSH EA
                                           $NEXT
0546  48 84                1               DB 48H,84H                                   ;
0548  48 28                1               DB 48H,28H                                   ;

                           ;   ROT         ( w1 w2 w3 -- w2 w3 w1 )
                           ;               Rot 3rd item to top.

                                           $COLON  3,'ROT',ROT
054A                       2       ROT:                                                 ;
3D93                       2       ORG     _NAME                                        ;
3D93  054A R 3D9F R        2               DW       _CODE,_LINK                         ;
3D97  03 52 4F 54          2               DB      3,'ROT'                              ;
054A                       2       ORG     _CODE                                        ;
054A  80                   1               DB 80H                                       ;
054B  0413 R 0442 R 0405 R                 DW      TOR,SWAP,RFROM,SWAP,EXIT
      0442 R 0394 R

                           ;   2DROP       ( w w -- )
                           ;               Discard two items on stack.

                                           $CODE  5,'2DROP',DDROP
0555                       1       DDROP:                                               ;
3D89                       1       ORG     _NAME                                        ;
3D89  0555 R 3D97 R        1               DW       _CODE,_LINK                         ;
3D8D  05 32 44 52 4F 50    1               DB      5,'2DROP'                            ;
0555                       1       ORG     _CODE                                        ;
0555  A4 A4                                DB 0A4H,0A4H            ;POP EA, POP EA
                                           $NEXT
0557  48 84                1               DB 48H,84H                                   ;
0559  48 28                1               DB 48H,28H                                   ;

                           ;   2DUP        ( w1 w2 -- w1 w2 w1 w2 )
                           ;               Duplicate top two items.

                                           $CODE  4,'2DUP',DDUP
055B                       1       DDUP:                                                ;
```

```
3D7F                          1       ORG     _NAME                                   ;
3D7F  055B R 3D8D R           1               DW       _CODE,_LINK                    ;
3D83  04 32 44 55 50          1               DB      4,'2DUP'                        ;
055B                          1       ORG     _CODE                                   ;
055B  A4 A1                                   DB 0A4H,0A1H            ;POP EA, POP BC
055D  B1 B4                                   DB 0B1H,0B4H            ;PUSH BC, PUSH EA
055F  B1 B4                                   DB 0B1H,0B4H            ;PUSH BC, PUSH EA
                                              $NEXT
0561  48 84                   1               DB 48H,84H                              ;
0563  48 28                   1               DB 48H,28H                              ;

                              ;   +            ( w w -- sum )
                              ;                Add top two items.

                                              $CODE  1,'+',PLUS
0565                          1       PLUS:                                           ;
3D79                          1       ORG     _NAME                                   ;
3D79  0565 R 3D83 R           1               DW       _CODE,_LINK                    ;
3D7D  01 2B                   1               DB      1,'+'                        ;
0565                          1       ORG     _CODE                                   ;
0565  A1 A4                                   DB 0A1H,0A4H            ;POP BC, POP EA
0567  74 A5                                   DB 74H,0A5H             ;EA<EA+BC, Skip
0569  00                                      DB 0                    ;NOP
056A  B4                                      DB 0B4H                 ;PUSH EA
                                              $NEXT
056B  48 84                   1               DB 48H,84H                              ;
056D  48 28                   1               DB 48H,28H                              ;

                              ;   D+           ( d d -- d )
                              ;                Double addition, as an example using UM+.
                              ;
                              ;                $COLON  2,'D+',DPLUS
                              ;                DW      TOR,SWAP,TOR,UPLUS
                              ;                DW      RFROM,RFROM,PLUS,PLUS,EXIT

                              ;   NOT          ( w -- w )
                              ;                One's complement of tos.

                                              $CODE  3,'NOT',INVER
056F                          1       INVER:                                          ;
3D71                          1       ORG     _NAME                                   ;
3D71  056F R 3D7D R           1               DW       _CODE,_LINK                    ;
3D75  03 4E 4F 54             1               DB      3,'NOT'                       ;
056F                          1       ORG     _CODE                                   ;
056F  A1                                      DB 0A1H                 ;POP BC
0570  69 FF                                   DB 69H,0FFH             ;A<FF
0572  60 12                                   DB 60H,12H              ;B<B EX-OR A
0574  60 13                                   DB 60H,13H              ;C<C EX-OR A
0576  B1                                      DB 0B1H                 ;PUSH BC
                                              $NEXT
0577  48 84                   1               DB 48H,84H                              ;
0579  48 28                   1               DB 48H,28H                              ;

                              ;   NEGATE       ( n -- -n )
                              ;                Two's complement of tos.

```

```
                                            $CODE  6,'NEGATE',NEGAT
057B                         1       NEGAT:                                            ;
3D65                         1       ORG     _NAME                                     ;
3D65  057B R 3D75 R          1               DW       _CODE,_LINK                     ;
3D69  06 4E 45 47 41 54 45   1               DB      6,'NEGATE'                         ;
057B                         1       ORG     _CODE                                     ;
057B  A1                                     DB 0A1H                  ;POP BC
057C  69 FF                                  DB 69H,0FFH              ;A<FF
057E  60 12                                  DB 60H,12H               ;B<B EX-OR A
0580  60 13                                  DB 60H,13H               ;C<C EX-OR A
0582  12                                     DB 12H                   ;BC<BC+1
0583  B1                                     DB 0B1H                  ;PUSH BC
                                             $NEXT
0584  48 84                  1               DB 48H,84H                               ;
0586  48 28                  1               DB 48H,28H                               ;


                             ;   DNEGATE      ( d -- -d )
                             ;                Two's complement of top double.

                                             $COLON  7,'DNEGATE',DNEGA
0588                         2       DNEGA:                                            ;
3D59                         2       ORG     _NAME                                     ;
3D59  0588 R 3D69 R          2               DW       _CODE,_LINK                     ;
3D5D  07 44 4E 45 47 41 54   2               DB      7,'DNEGATE'                          ;
0588                         2       ORG     _CODE                                     ;
0588  80                     1               DB 80H                                   ;
0589  056F R 0413 R 056F R                   DW      INVER,TOR,INVER
058F  038D R 0001 047C R                     DW      DOLIT,1,UPLUS
0595  0405 R 0565 R 0394 R                   DW      RFROM,PLUS,EXIT

                             ;   -            ( n1 n2 -- n1-n2 )
                             ;                Subtraction.

                                             $CODE  1,'-',SUBB
059B                         1       SUBB:                                            ;
3D53                         1       ORG     _NAME                                     ;
3D53  059B R 3D5D R          1               DW       _CODE,_LINK                     ;
3D57  01 2D                  1               DB      1,'-'                        ;
059B                         1       ORG     _CODE                                     ;
059B  A1                                     DB 0A1H                  ;POP BC
059C  69 FF                                  DB 069H,0FFH             ;A<FF
059E  60 12                                  DB 060H,12H              ;B<B EX-OR A
05A0  60 13                                  DB 060H,13H              ;C<C EX-OR A
05A2  12                                     DB 12H                   ;BC<BC+1
05A3  A4                                     DB 0A4H                  ;POP EA
05A4  74 A5                                  DB 74H,0A5H              ;EA<EA+BC Skip
05A6  00                                     DB 0                     ;NOP
05A7  B4                                     DB 0B4H                  ;PUSH EA
                                             $NEXT
05A8  48 84                  1               DB 48H,84H                               ;
05AA  48 28                  1               DB 48H,28H                               ;

                             ;   ABS          ( n -- n )
                             ;                Return the absolute value of n.

```

```
                                               $COLON  3,'ABS',ABSS
 05AC                         2       ABSS:                                             ;
 3D4B                         2       ORG     _NAME                                      ;
 3D4B  05AC R 3D57 R          2               DW       _CODE,_LINK                       ;
 3D4F  03 41 42 53            2               DB      3,'ABS'                           ;
 05AC                         2       ORG     _CODE                                      ;
 05AC  80                     1               DB 80H                                     ;
 05AD  043B R 0453 R                          DW      DUPP,ZLESS
 05B1  03B7 R 05B7 R                          DW      QBRAN,ABS1
 05B5  057B R                                 DW      NEGAT
 05B7  0394 R                ABS1:             DW      EXIT

                              ;   =            ( w w -- t )
                              ;                Return true if top two are equal.

                                              $CODE  1,'=',EQUAL
 05B9                         1       EQUAL:                                             ;
 3D45                         1       ORG     _NAME                                      ;
 3D45  05B9 R 3D4F R          1               DW       _CODE,_LINK                       ;
 3D49  01 3D                  1               DB      1,'='                           ;
 05B9                         1       ORG     _CODE                                      ;
 05B9  A4 A1                                  DB 0A4H,0A1H            ;POP EA, POP BC
 05BB  69 FF                                  DB 69H,0FFH             ;A<FF
 05BD  74 FD                                  DB 74H,0FDH             ;EA-BC, Skip if zero
 05BF  69 00                                  DB 69H,00H              ;A<00
 05C1  1A 1B                                  DB 1AH,1BH              ;B<A, C<A
 05C3  B1                                     DB 0B1H                 ;PUSH BC
                                              $NEXT
 05C4  48 84                  1               DB 48H,84H                                ;
 05C6  48 28                  1               DB 48H,28H                                ;

                              ;   U<           ( u u -- t )
                              ;                Unsigned compare of top two items.

                                              $COLON  2,'U<',ULESS
 05C8                         2       ULESS:                                             ;
 3D3D                         2       ORG     _NAME                                      ;
 3D3D  05C8 R 3D49 R          2               DW       _CODE,_LINK                       ;
 3D41  02 55 3C               2               DB      2,'U<'                          ;
 05C8                         2       ORG     _CODE                                      ;
 05C8  80                     1               DB 80H                                     ;
 05C9  055B R 0473 R 0453 R                   DW      DDUP,XORR,ZLESS
 05CF  03B7 R 05DB R                          DW      QBRAN,ULES1
 05D3  0442 R 0436 R 0453 R                   DW      SWAP,DROP,ZLESS,EXIT
       0394 R
 05DB  059B R 0453 R 0394 R  ULES1:            DW      SUBB,ZLESS,EXIT

                              ;   <            ( n1 n2 -- t )
                              ;                Signed compare of top two items.

                                              $COLON  1,'<',LESS
 05E1                         2       LESS:                                             ;
 3D37                         2       ORG     _NAME                                      ;
 3D37  05E1 R 3D41 R          2               DW       _CODE,_LINK                       ;
 3D3B  01 3C                  2               DB      1,'<'                           ;
 05E1                         2       ORG     _CODE                                      ;
```

```
05E1  80                     1               DB 80H                                  ;
05E2  055B R 0473 R 0453 R                   DW      DDUP,XORR,ZLESS
05E8  03B7 R 05F2 R                          DW      QBRAN,LESS1
05EC  0436 R 0453 R 0394 R                   DW      DROP,ZLESS,EXIT
05F2  059B R 0453 R 0394 R  LESS1:            DW      SUBB,ZLESS,EXIT

                            ;   MAX           ( n n -- n )
                            ;                 Return the greater of two top stack items.

                                             $CODE  3,'MAX',MAX
05F8                         1       MAX:                                            ;
3D2F                         1       ORG      _NAME                                   ;
3D2F  05F8 R 3D3B R          1               DW       _CODE,_LINK                     ;
3D33  03 4D 41 58            1               DB      3,'MAX'                         ;
05F8                         1       ORG      _CODE                                   ;
05F8  A4 A1                                  DB 0A4H,0A1H           ;POP EA, POP BC
05FA  74 BD                                  DB 74H,0BDH            ;EA-BC, Skip if borrow
05FC  C2                                     DB 0C2H                ;Jump to Push EA
05FD  B1                                     DB 0B1H                ;PUSH BC
05FE  C1                                     DB 0C1H                ;Jump to next
05FF  B4                                     DB 0B4H                ;PUSH EA
                                             $NEXT
0600  48 84                  1               DB 48H,84H                              ;
0602  48 28                  1               DB 48H,28H                              ;

                            ;   MIN           ( n n -- n )
                            ;                 Return the smaller of top two stack items.

                                             $CODE  3,'MIN',MIN
0604                         1       MIN:                                            ;
3D27                         1       ORG      _NAME                                   ;
3D27  0604 R 3D33 R          1               DW       _CODE,_LINK                     ;
3D2B  03 4D 49 4E            1               DB      3,'MIN'                         ;
0604                         1       ORG      _CODE                                   ;
0604  A4 A1                                  DB 0A4H,0A1H           ;POP EA, POP BC
0606  74 BD                                  DB 74H,0BDH            ;EA-BC, Skip if borrow
0608  C2                                     DB 0C2H                ;Jump to Push EA
0609  B4                                     DB 0B4H                ;PUSH EA
060A  C1                                     DB 0C1H                ;Jump to next
060B  B1                                     DB 0B1H                ;PUSH BC
                                             $NEXT
060C  48 84                  1               DB 48H,84H                              ;
060E  48 28                  1               DB 48H,28H                              ;

                            ;   WITHIN        ( u ul uh -- t )
                            ;                 Return true if u is within the range of ul and uh.

                                             $COLON  6,'WITHIN',WITHI
0610                         2       WITHI:                                           ;
3D1B                         2       ORG      _NAME                                   ;
3D1B  0610 R 3D2B R          2               DW       _CODE,_LINK                     ;
3D1F  06 57 49 54 48 49 4E  2               DB      6,'WITHIN'                        ;
0610                         2       ORG      _CODE                                   ;
0610  80                     1               DB 80H                                  ;
0611  044A R 059B R 0413 R                   DW      OVER,SUBB,TOR                   ;ul <= u < uh
0617  059B R 0405 R 05C8 R                   DW      SUBB,RFROM,ULESS,EXIT
```

```
      0394 R

                            ;; Quick Operators

                            ;




                            ;    1+          ( n -- n+1 )
                                           $CODE 2,'1+',ONEP
061F                        1       ONEP:                                          ;
3D13                        1       ORG     _NAME                                   ;
3D13  061F R 3D1F R         1              DW       _CODE,_LINK                    ;
3D17  02 31 2B              1              DB      2,'1+'                        ;
061F                        1       ORG     _CODE                                   ;
061F  A1                                   DB 0A1H                ;POP BC
0620  12                                   DB 12H                 ;BC<BC+1
0621  B1                                   DB 0B1H                ;PUSH BC
                                           $NEXT
0622  48 84                 1              DB 48H,84H                              ;
0624  48 28                 1              DB 48H,28H                              ;

                            ;    1-          ( n -- n-1 )
                                           $CODE 2,'1-',ONEM
0626                        1       ONEM:                                          ;
3D0B                        1       ORG     _NAME                                   ;
3D0B  0626 R 3D17 R         1              DW       _CODE,_LINK                    ;
3D0F  02 31 2D              1              DB      2,'1-'                        ;
0626                        1       ORG     _CODE                                   ;
0626  A1                                   DB 0A1H                ;POP BC
0627  13                                   DB 013H                ;BC<BC-1
0628  B1                                   DB 0B1H                ;PUSH BC
                                           $NEXT
0629  48 84                 1              DB 48H,84H                              ;
062B  48 28                 1              DB 48H,28H                              ;

                            ;    2+          ( n -- n+2 )
                                           $CODE 2,'2+',TWOP
062D                        1       TWOP:                                          ;
3D03                        1       ORG     _NAME                                   ;
3D03  062D R 3D0F R         1              DW       _CODE,_LINK                    ;
3D07  02 32 2B              1              DB      2,'2+'                        ;
062D                        1       ORG     _CODE                                   ;
062D  A1                                   DB 0A1H                ;POP BC
062E  12 12                                DB 12H,12H             ;BC<BC+2
0630  B1                                   DB 0B1H                ;PUSH BC
                                           $NEXT
0631  48 84                 1              DB 48H,84H                              ;
0633  48 28                 1              DB 48H,28H                              ;

                            ;    2-          ( n -- n-2 )
                                           $CODE 2,'2-',TWOM
0635                        1       TWOM:                                          ;
3CFB                        1       ORG     _NAME                                   ;
3CFB  0635 R 3D07 R         1              DW       _CODE,_LINK                    ;
```

```
3CFF  02 32 2D             1               DB      2,'2-'                          ;
0635                       1       ORG     _CODE                                      ;
0635  A1                                   DB 0A1H                 ;POP BC
0636  13 13                                DB 13H,13H              ;BC<BC-2
0638  B1                                   DB 0B1H                 ;PUSH BC
                                           $NEXT
0639  48 84                1               DB 48H,84H                                ;
063B  48 28                1               DB 48H,28H                                ;

                           ;   2*           ( n -- n*2 )
                                           $CODE 2,'2*',TWOSL
063D                       1       TWOSL:                                             ;
3CF3                       1       ORG     _NAME                                      ;
3CF3  063D R 3CFF R        1               DW       _CODE,_LINK                      ;
3CF7  02 32 2A             1               DB      2,'2*'                          ;
063D                       1       ORG     _CODE                                      ;
063D  A4                                   DB 0A4H                 ;POP EA
063E  48 A4                                DB 48H,0A4H             ;EA Logical Shift Left
0640  B4                                   DB 0B4H                 ;PUSH EA
                                           $NEXT
0641  48 84                1               DB 48H,84H                                ;
0643  48 28                1               DB 48H,28H                                ;

                           ;   2/           ( n -- n/2 )
                                           $CODE 2,'2/',TWOSR
0645                       1       TWOSR:                                             ;
3CEB                       1       ORG     _NAME                                      ;
3CEB  0645 R 3CF7 R        1               DW       _CODE,_LINK                      ;
3CEF  02 32 2F             1               DB      2,'2/'                          ;
0645                       1       ORG     _CODE                                      ;
0645  A4                                   DB 0A4H                 ;POP EA
0646  48 A0                                DB 48H,0A0H             ;EA Logical Shift Right
0648  B4                                   DB 0B4H                 ;PUSH EA
                                           $NEXT
0649  48 84                1               DB 48H,84H                                ;
064B  48 28                1               DB 48H,28H                                ;

                           ;; Divide

                           ;   UM/MOD       ( udl udh u -- ur uq )
                           ;                Unsigned divide of a double by a single. Return mod and quotient.

                                           $COLON  6,'UM/MOD',UMMOD
064D                       2       UMMOD:                                             ;
3CDF                       2       ORG     _NAME                                      ;
3CDF  064D R 3CEF R        2               DW       _CODE,_LINK                      ;
3CE3  06 55 4D 2F 4D 4F 44  2              DB      6,'UM/MOD'                          ;
064D                       2       ORG     _CODE                                      ;
064D  80                   1               DB 80H                                    ;
064E  055B R 05C8 R                        DW      DDUP,ULESS
0652  03B7 R 069E R                        DW      QBRAN,UMM4
0656  057B R 038D R 000F                   DW      NEGAT,DOLIT,15,TOR
      0413 R
065E  0413 R 043B R 047C R  UMM1:           DW      TOR,DUPP,UPLUS
0664  0413 R 0413 R 043B R                 DW      TOR,TOR,DUPP,UPLUS
      047C R
```

```
066C  0405 R 0565 R 043B R                  DW      RFROM,PLUS,DUPP
0672  0405 R 040C R 0442 R                  DW      RFROM,RAT,SWAP,TOR
      0413 R
067A  047C R 0405 R 046A R                  DW      UPLUS,RFROM,ORR
0680  03B7 R 0690 R                         DW      QBRAN,UMM2
0684  0413 R 0436 R 061F R                  DW      TOR,DROP,ONEP,RFROM
      0405 R
068C  03CC R 0692 R                         DW      BRAN,UMM3
0690  0436 R               UMM2:             DW      DROP
0692  0405 R               UMM3:             DW      RFROM
0694  039D R 065E R                         DW      DONXT,UMM1
0698  0436 R 0442 R 0394 R                  DW      DROP,SWAP,EXIT
069E  0436 R 0555 R        UMM4:             DW      DROP,DDROP
06A2  038D R FFFF 043B R                    DW      DOLIT,-1,DUPP,EXIT      ;overflow, return max
      0394 R

                           ;   M/MOD        ( d n -- r q )
                           ;                Signed floored divide of double by single. Return mod and
quotient.

                                            $COLON  5,'M/MOD',MSMOD
06AA                      2       MSMOD:                                              ;
3CD5                      2       ORG     _NAME                                       ;
3CD5  06AA R 3CE3 R       2               DW       _CODE,_LINK                       ;
3CD9  05 4D 2F 4D 4F 44   2               DB      5,'M/MOD'                           ;
06AA                      2       ORG     _CODE                                       ;
06AA  80                  1               DB 80H                                     ;
06AB  043B R 0453 R 043B R                  DW      DUPP,ZLESS,DUPP,TOR
      0413 R
06B3  03B7 R 06BF R                         DW      QBRAN,MMOD1
06B7  057B R 0413 R 0588 R                  DW      NEGAT,TOR,DNEGA,RFROM
      0405 R
06BF  0413 R 043B R 0453 R  MMOD1:           DW      TOR,DUPP,ZLESS
06C5  03B7 R 06CD R                         DW      QBRAN,MMOD2
06C9  040C R 0565 R                         DW      RAT,PLUS
06CD  0405 R 064D R 0405 R  MMOD2:           DW      RFROM,UMMOD,RFROM
06D3  03B7 R 06DD R                         DW      QBRAN,MMOD3
06D7  0442 R 057B R 0442 R                  DW      SWAP,NEGAT,SWAP
06DD  0394 R               MMOD3:            DW      EXIT

                           ;   /MOD         ( n n -- r q )
                           ;                Signed divide. Return mod and quotient.

                                            $COLON  4,'/MOD',SLMOD
06DF                      2       SLMOD:                                              ;
3CCB                      2       ORG     _NAME                                       ;
3CCB  06DF R 3CD9 R       2               DW       _CODE,_LINK                       ;
3CCF  04 2F 4D 4F 44      2               DB      4,'/MOD'                           ;
06DF                      2       ORG     _CODE                                       ;
06DF  80                  1               DB 80H                                     ;
06E0  044A R 0453 R 0442 R                  DW      OVER,ZLESS,SWAP,MSMOD,EXIT
      06AA R 0394 R

                           ;   MOD          ( n n -- r )
                           ;                Signed divide. Return mod only.

                                            $COLON  3,'MOD',MODD
```

```
06EA                         2       MODD:                                            ;
3CC3                         2       ORG     _NAME                                     ;
3CC3  06EA R 3CCF R          2               DW       _CODE,_LINK                     ;
3CC7  03 4D 4F 44            2               DB      3,'MOD'                        ;
06EA                         2       ORG     _CODE                                     ;
06EA  80                     1               DB 80H                                   ;
06EB  06DF R 0436 R 0394 R                   DW      SLMOD,DROP,EXIT

                             ;   /            ( n n -- q )
                             ;                Signed divide. Return quotient only.

                                             $COLON  1,'/',SLASH
06F1                         2       SLASH:                                            ;
3CBD                         2       ORG     _NAME                                     ;
3CBD  06F1 R 3CC7 R          2               DW       _CODE,_LINK                     ;
3CC1  01 2F                  2               DB      1,'/'                       ;
06F1                         2       ORG     _CODE                                     ;
06F1  80                     1               DB 80H                                   ;
06F2  06DF R 0442 R 0436 R                   DW      SLMOD,SWAP,DROP,EXIT
      0394 R

                             ;; Multiply

                             ;   UM*          ( u u -- ud )
                             ;                Unsigned multiply. Return double product.

                                             $COLON  3,'UM*',UMSTA
06FA                         2       UMSTA:                                           ;
3CB5                         2       ORG     _NAME                                     ;
3CB5  06FA R 3CC1 R          2               DW       _CODE,_LINK                     ;
3CB9  03 55 4D 2A            2               DB      3,'UM*'                        ;
06FA                         2       ORG     _CODE                                     ;
06FA  80                     1               DB 80H                                   ;
06FB  038D R 0000 0442 R                     DW      DOLIT,0,SWAP,DOLIT,15,TOR
      038D R 000F 0413 R
0707  043B R 047C R 0413 R   UMST1:           DW      DUPP,UPLUS,TOR,TOR
      0413 R
070F  043B R 047C R 0405 R                   DW      DUPP,UPLUS,RFROM,PLUS,RFROM
      0565 R 0405 R
0719  03B7 R 0727 R                          DW      QBRAN,UMST2
071D  0413 R 044A R 047C R                   DW      TOR,OVER,UPLUS,RFROM,PLUS
      0405 R 0565 R
0727  039D R 0707 R          UMST2:           DW      DONXT,UMST1
072B  054A R 0436 R 0394 R                   DW      ROT,DROP,EXIT

                             ;   *            ( n n -- n )
                             ;                Signed multiply. Return single product.

                                             $COLON  1,'*',STAR
0731                         2       STAR:                                            ;
3CAF                         2       ORG     _NAME                                     ;
3CAF  0731 R 3CB9 R          2               DW       _CODE,_LINK                     ;
3CB3  01 2A                  2               DB      1,'*'                       ;
0731                         2       ORG     _CODE                                     ;
0731  80                     1               DB 80H                                   ;
0732  06FA R 0436 R 0394 R                   DW      UMSTA,DROP,EXIT
```

```

                            ;   M*          ( n n -- d )
                            ;               Signed multiply. Return double product.

                                           $COLON  2,'M*',MSTAR
0738                        2      MSTAR:                                          ;
3CA7                        2      ORG     _NAME                                   ;
3CA7  0738 R 3CB3 R         2              DW       _CODE,_LINK                   ;
3CAB  02 4D 2A              2              DB      2,'M*'                       ;
0738                        2      ORG     _CODE                                   ;
0738  80                    1              DB 80H                                 ;
0739  055B R 0473 R 0453 R                 DW      DDUP,XORR,ZLESS,TOR
      0413 R
0741  05AC R 0442 R 05AC R                 DW      ABSS,SWAP,ABSS,UMSTA
      06FA R
0749  0405 R                               DW      RFROM
074B  03B7 R 0751 R                        DW      QBRAN,MSTA1
074F  0588 R                               DW      DNEGA
0751  0394 R                MSTA1:          DW      EXIT

                            ;   */MOD       ( n1 n2 n3 -- r q )
                            ;               Multiply n1 and n2, then divide by n3. Return mod and quotient.

                                           $COLON  5,'*/MOD',SSMOD
0753                        2      SSMOD:                                          ;
3C9D                        2      ORG     _NAME                                   ;
3C9D  0753 R 3CAB R         2              DW       _CODE,_LINK                   ;
3CA1  05 2A 2F 4D 4F 44     2              DB      5,'*/MOD'                       ;
0753                        2      ORG     _CODE                                   ;
0753  80                    1              DB 80H                                 ;
0754  0413 R 0738 R 0405 R                 DW      TOR,MSTAR,RFROM,MSMOD,EXIT
      06AA R 0394 R

                            ;   */          ( n1 n2 n3 -- q )
                            ;               Multiply n1 by n2, then divide by n3. Return quotient only.

                                           $COLON  2,'*/',STASL
075E                        2      STASL:                                          ;
3C95                        2      ORG     _NAME                                   ;
3C95  075E R 3CA1 R         2              DW       _CODE,_LINK                   ;
3C99  02 2A 2F              2              DB      2,'*/'                       ;
075E                        2      ORG     _CODE                                   ;
075E  80                    1              DB 80H                                 ;
075F  0753 R 0442 R 0436 R                 DW      SSMOD,SWAP,DROP,EXIT
      0394 R

                            ;; Miscellaneous

                            ;   BL          ( -- 32 )
                            ;               Return 32, the blank character.

                                           $COLON  2,'BL',BLANK
0767                        2      BLANK:                                          ;
3C8D                        2      ORG     _NAME                                   ;
3C8D  0767 R 3C99 R         2              DW       _CODE,_LINK                   ;
3C91  02 42 4C              2              DB      2,'BL'                       ;
```

```
0767                        2       ORG     _CODE                                   ;
0767  80                    1               DB 80H                                  ;
0768  038D R 0020 0394 R                    DW      DOLIT,' ',EXIT

                             ;   >CHAR       ( c -- c )
                             ;               Filter non-printing characters.

                                            $COLON  5,'>CHAR',TCHAR
076E                        2       TCHAR:                                          ;
3C83                        2       ORG     _NAME                                   ;
3C83  076E R 3C91 R         2               DW       _CODE,_LINK                    ;
3C87  05 3E 43 48 41 52     2               DB      5,'>CHAR'                        ;
076E                        2       ORG     _CODE                                   ;
076E  80                    1               DB 80H                                  ;
076F  038D R 007F 0461 R                    DW      DOLIT,07FH,ANDD,DUPP    ;mask msb
      043B R
0777  038D R 007F 0767 R                    DW      DOLIT,127,BLANK,WITHI   ;check for printable
      0610 R
077F  03B7 R 0789 R                         DW      QBRAN,TCHA1
0783  0436 R 038D R 005F                    DW      DROP,DOLIT,'_'          ;replace non-printables
0789  0394 R                TCHA1:           DW      EXIT

                             ;   DEPTH       ( -- n )
                             ;               Return the depth of the data stack.

                                            $COLON  5,'DEPTH',DEPTH
078B                        2       DEPTH:                                          ;
3C79                        2       ORG     _NAME                                   ;
3C79  078B R 3C87 R         2               DW       _CODE,_LINK                    ;
3C7D  05 44 45 50 54 48     2               DB      5,'DEPTH'                        ;
078B                        2       ORG     _CODE                                   ;
078B  80                    1               DB 80H                                  ;
078C  041C R 04A3 R 03DE R                  DW      SPAT,SZERO,AT,SWAP,SUBB
      0442 R 059B R
0796  038D R 0002 06F1 R                    DW      DOLIT,CELLL,SLASH,EXIT
      0394 R

                             ;   PICK        ( ... +n -- ... w )
                             ;               Copy the nth stack item to tos.

                                            $COLON  4,'PICK',PICK
079E                        2       PICK:                                          ;
3C6F                        2       ORG     _NAME                                   ;
3C6F  079E R 3C7D R         2               DW       _CODE,_LINK                    ;
3C73  04 50 49 43 4B        2               DB      4,'PICK'                        ;
079E                        2       ORG     _CODE                                   ;
079E  80                    1               DB 80H                                  ;
079F  061F R 063D R                         DW      ONEP,TWOSL
07A3  041C R 0565 R 03DE R                  DW      SPAT,PLUS,AT,EXIT
      0394 R

                             ;; Memory access

                             ;   +!          ( n a -- )
                             ;               Add n to the contents at address a.

```

```
                                                $COLON  2,'+!',PSTOR
07AB                        2       PSTOR:                                        ;
3C67                        2       ORG     _NAME                                 ;
3C67  07AB R 3C73 R         2               DW       _CODE,_LINK                  ;
3C6B  02 2B 21              2               DB      2,'+!'                      ;
07AB                        2       ORG     _CODE                                 ;
07AB  80                    1               DB 80H                                ;
07AC  0442 R 044A R 03DE R                  DW      SWAP,OVER,AT,PLUS
      0565 R
07B4  0442 R 03D3 R 0394 R                  DW      SWAP,STORE,EXIT

                            ;   2!           ( d a -- )
                            ;                Store the double integer to address a.

                                            $COLON  2,'2!',DSTOR
07BA                        2       DSTOR:                                        ;
3C5F                        2       ORG     _NAME                                 ;
3C5F  07BA R 3C6B R         2               DW       _CODE,_LINK                  ;
3C63  02 32 21              2               DB      2,'2!'                      ;
07BA                        2       ORG     _CODE                                 ;
07BA  80                    1               DB 80H                                ;
07BB  0442 R 044A R 03D3 R                  DW      SWAP,OVER,STORE
07C1  062D R 03D3 R 0394 R                  DW      TWOP,STORE,EXIT

                            ;   2@           ( a -- d )
                            ;                Fetch double integer from address a.

                                            $COLON  2,'2@',DAT
07C7                        2       DAT:                                        ;
3C57                        2       ORG     _NAME                                 ;
3C57  07C7 R 3C63 R         2               DW       _CODE,_LINK                  ;
3C5B  02 32 40              2               DB      2,'2@'                      ;
07C7                        2       ORG     _CODE                                 ;
07C7  80                    1               DB 80H                                ;
07C8  043B R 062D R 03DE R                  DW      DUPP,TWOP,AT
07CE  0442 R 03DE R 0394 R                  DW      SWAP,AT,EXIT

                            ;   COUNT        ( b -- b +n )
                            ;                Return count byte of a string and add 1 to byte address.

                                            $COLON  5,'COUNT',COUNT
07D4                        2       COUNT:                                        ;
3C4D                        2       ORG     _NAME                                 ;
3C4D  07D4 R 3C5B R         2               DW       _CODE,_LINK                  ;
3C51  05 43 4F 55 4E 54     2               DB      5,'COUNT'                      ;
07D4                        2       ORG     _CODE                                 ;
07D4  80                    1               DB 80H                                ;
07D5  043B R 061F R                         DW      DUPP,ONEP
07D9  0442 R 03F1 R 0394 R                  DW      SWAP,CAT,EXIT

                            ;   HERE         ( -- a )
                            ;                Return the top of the code dictionary.

                                            $COLON  4,'HERE',HERE
07DF                        2       HERE:                                        ;
3C43                        2       ORG     _NAME                                 ;
```

```
3C43  07DF R 3C51 R         2              DW       _CODE,_LINK                    ;
3C47  04 48 45 52 45        2              DB      4,'HERE'                        ;
07DF                        2       ORG     _CODE                                   ;
07DF  80                    1              DB 80H                                  ;
07E0  0511 R 03DE R 0394 R                 DW      CP,AT,EXIT

                            ;   PAD         ( -- a )
                            ;               Return the address of a temporary buffer.

                                           $COLON  3,'PAD',PAD
07E6                        2       PAD:                                          ;
3C3B                        2       ORG     _NAME                                   ;
3C3B  07E6 R 3C47 R         2              DW       _CODE,_LINK                    ;
3C3F  03 50 41 44           2              DB      3,'PAD'                        ;
07E6                        2       ORG     _CODE                                   ;
07E6  80                    1              DB 80H                                  ;
07E7  038D R C300 0394 R                   DW      DOLIT,PADD,EXIT

                            ;   TIB         ( -- a )
                            ;               Return the address of the terminal input buffer.

                                           $COLON  3,'TIB',TIB
07ED                        2       TIB:                                          ;
3C33                        2       ORG     _NAME                                   ;
3C33  07ED R 3C3F R         2              DW       _CODE,_LINK                    ;
3C37  03 54 49 42           2              DB      3,'TIB'                        ;
07ED                        2       ORG     _CODE                                   ;
07ED  80                    1              DB 80H                                  ;
07EE  04DF R 062D R 03DE R                 DW      NTIB,TWOP,AT,EXIT
      0394 R

                            ;   @EXECUTE    ( a -- )
                            ;               Execute vector stored in address a.

                                           $COLON  8,'@EXECUTE',ATEXE
07F6                        2       ATEXE:                                          ;
3C25                        2       ORG     _NAME                                   ;
3C25  07F6 R 3C37 R         2              DW       _CODE,_LINK                    ;
3C29  08 40 45 58 45 43 55  2              DB      8,'@EXECUTE'                        ;
07F6                        2       ORG     _CODE                                   ;
07F6  80                    1              DB 80H                                  ;
07F7  03DE R 053D R                        DW      AT,QDUP                 ;?address or zero
07FB  03B7 R 0801 R                        DW      QBRAN,EXE1
07FF  039B R                               DW      EXECU                   ;execute if non-zero
0801  0394 R                EXE1:           DW      EXIT                    ;do nothing if zero

                            ;   CMOVE       ( b1 b2 u -- )
                            ;               Copy u bytes from b1 to b2.

                                           $COLON  5,'CMOVE',CMOVE
0803                        2       CMOVE:                                          ;
3C1B                        2       ORG     _NAME                                   ;
3C1B  0803 R 3C29 R         2              DW       _CODE,_LINK                    ;
3C1F  05 43 4D 4F 56 45     2              DB      5,'CMOVE'                        ;
0803                        2       ORG     _CODE                                   ;
0803  80                    1              DB 80H                                  ;
```

```
0804  0413 R                                DW      TOR
0806  03CC R 081A R                         DW      BRAN,CMOV2
080A  0413 R 043B R 03F1 R  CMOV1:          DW      TOR,DUPP,CAT
0810  040C R 03E9 R                         DW      RAT,CSTOR
0814  061F R                                DW      ONEP
0816  0405 R 061F R                         DW      RFROM,ONEP
081A  039D R 080A R         CMOV2:          DW      DONXT,CMOV1
081E  0555 R 0394 R                         DW      DDROP,EXIT

                            ;   FILL        ( b u c -- )
                            ;               Fill u bytes of character c to area beginning at b.

                                            $COLON  4,'FILL',FILL
0822                        2       FILL:                                           ;
3C11                        2       ORG     _NAME                                   ;
3C11  0822 R 3C1F R         2               DW       _CODE,_LINK                    ;
3C15  04 46 49 4C 4C        2               DB      4,'FILL'                        ;
0822                        2       ORG     _CODE                                   ;
0822  80                    1               DB 80H                                  ;
0823  0442 R 0413 R 0442 R                  DW      SWAP,TOR,SWAP
0829  03CC R 0833 R                         DW      BRAN,FILL2
082D  055B R 03E9 R 061F R  FILL1:          DW      DDUP,CSTOR,ONEP
0833  039D R 082D R         FILL2:          DW      DONXT,FILL1
0837  0555 R 0394 R                         DW      DDROP,EXIT

                            ;   -TRAILING   ( b u -- b u )
                            ;               Adjust the count to eliminate trailing white space.

                                            $COLON  9,'-TRAILING',DTRAI
083B                        2       DTRAI:                                          ;
3C03                        2       ORG     _NAME                                   ;
3C03  083B R 3C15 R         2               DW       _CODE,_LINK                    ;
3C07  09 2D 54 52 41 49 4C  2               DB      9,'-TRAILING'                       ;
083B                        2       ORG     _CODE                                   ;
083B  80                    1               DB 80H                                  ;
083C  0413 R                                DW      TOR
083E  03CC R 0858 R                         DW      BRAN,DTRA2
0842  0767 R 044A R 040C R  DTRA1:          DW      BLANK,OVER,RAT,PLUS,CAT,LESS
      0565 R 03F1 R 05E1 R
084E  03B7 R 0858 R                         DW      QBRAN,DTRA2
0852  0405 R 061F R 0394 R                  DW      RFROM,ONEP,EXIT           ;adjusted count
0858  039D R 0842 R         DTRA2:          DW      DONXT,DTRA1
085C  038D R 0000 0394 R                    DW      DOLIT,0,EXIT            ;count=0

                            ;   PACK$       ( b u a -- a )
                            ;               Build a counted string with u characters from b. Null fill.

                                            $COLON  5,'PACK$',PACKS
0862                        2       PACKS:                                          ;
3BF9                        2       ORG     _NAME                                   ;
3BF9  0862 R 3C07 R         2               DW       _CODE,_LINK                    ;
3BFD  05 50 41 43 4B 24     2               DB      5,'PACK$'                        ;
0862                        2       ORG     _CODE                                   ;
0862  80                    1               DB 80H                                  ;
0863  043B R 0413 R                         DW      DUPP,TOR          ;strings only on cell boundary
0867  044A R 043B R 038D R                  DW      OVER,DUPP,DOLIT,0
```

```
      0000
086F  038D R 0002 064D R                    DW      DOLIT,CELLL,UMMOD,DROP  ;count mod cell
      0436 R
0877  059B R 044A R 0565 R                  DW      SUBB,OVER,PLUS
087D  038D R 0000 0442 R                    DW      DOLIT,0,SWAP,STORE      ;null fill cell
      03D3 R
0885  055B R 03E9 R 061F R                  DW      DDUP,CSTOR,ONEP           ;save count
088B  0442 R 0803 R 0405 R                  DW      SWAP,CMOVE,RFROM,EXIT   ;move string
      0394 R

                            ;; Numeric output, single precision

                            ;   DIGIT       ( u -- c )
                            ;               Convert digit u to a character.

                                            $COLON  5,'DIGIT',DIGIT
0893                        2       DIGIT:                                     ;
3BEF                        2       ORG     _NAME                              ;
3BEF  0893 R 3BFD R         2               DW       _CODE,_LINK              ;
3BF3  05 44 49 47 49 54     2               DB      5,'DIGIT'                   ;
0893                        2       ORG     _CODE                              ;
0893  80                    1               DB 80H                            ;
0894  038D R 0009 044A R                    DW      DOLIT,9,OVER,LESS
      05E1 R
089C  038D R 0007 0461 R                    DW      DOLIT,7,ANDD,PLUS
      0565 R
08A4  038D R 0030 0565 R                    DW      DOLIT,'0',PLUS,EXIT
      0394 R

                            ;   EXTRACT     ( n base -- n c )
                            ;               Extract the least significant digit from n.

                                            $COLON  7,'EXTRACT',EXTRC
08AC                        2       EXTRC:                                     ;
3BE3                        2       ORG     _NAME                              ;
3BE3  08AC R 3BF3 R         2               DW       _CODE,_LINK              ;
3BE7  07 45 58 54 52 41 43  2               DB      7,'EXTRACT'                   ;
08AC                        2       ORG     _CODE                              ;
08AC  80                    1               DB 80H                            ;
08AD  038D R 0000 0442 R                    DW      DOLIT,0,SWAP,UMMOD
      064D R
08B5  0442 R 0893 R 0394 R                  DW      SWAP,DIGIT,EXIT

                            ;   <#          ( -- )
                            ;               Initiate the numeric output process.

                                            $COLON  2,'<#',BDIGS
08BB                        2       BDIGS:                                     ;
3BDB                        2       ORG     _NAME                              ;
3BDB  08BB R 3BE7 R         2               DW       _CODE,_LINK              ;
3BDF  02 3C 23              2               DB      2,'<#'                  ;
08BB                        2       ORG     _CODE                              ;
08BB  80                    1               DB 80H                            ;
08BC  07E6 R 04F3 R 03D3 R                  DW      PAD,HLD,STORE,EXIT
      0394 R

```

```
                            ;   HOLD        ( c -- )
                            ;               Insert a character into the numeric output string.

                                           $COLON  4,'HOLD',HOLD
08C4                        2       HOLD:                                               ;
3BD1                        2       ORG    _NAME                                         ;
3BD1  08C4 R 3BDF R         2              DW       _CODE,_LINK                         ;
3BD5  04 48 4F 4C 44        2              DB      4,'HOLD'                             ;
08C4                        2       ORG    _CODE                                         ;
08C4  80                    1              DB 80H                                       ;
08C5  04F3 R 03DE R 0626 R                 DW      HLD,AT,ONEM
08CB  043B R 04F3 R 03D3 R                 DW      DUPP,HLD,STORE,CSTOR,EXIT
      03E9 R 0394 R

                            ;   #           ( u -- u )
                            ;               Extract one digit from u and append the digit to output string.

                                           $COLON  1,'#',DIG
08D5                        2       DIG:                                               ;
3BCB                        2       ORG    _NAME                                         ;
3BCB  08D5 R 3BD5 R         2              DW       _CODE,_LINK                         ;
3BCF  01 23                 2              DB      1,'#'                            ;
08D5                        2       ORG    _CODE                                         ;
08D5  80                    1              DB 80H                                       ;
08D6  04CB R 03DE R 08AC R                 DW      BASE,AT,EXTRC,HOLD,EXIT
      08C4 R 0394 R

                            ;   #S          ( u -- 0 )
                            ;               Convert u until all digits are added to the output string.

                                           $COLON  2,'#S',DIGS
08E0                        2       DIGS:                                               ;
3BC3                        2       ORG    _NAME                                         ;
3BC3  08E0 R 3BCF R         2              DW       _CODE,_LINK                         ;
3BC7  02 23 53              2              DB      2,'#S'                            ;
08E0                        2       ORG    _CODE                                         ;
08E0  80                    1              DB 80H                                       ;
08E1  08D5 R 043B R         DIGS1:          DW      DIG,DUPP
08E5  03B7 R 08ED R                        DW      QBRAN,DIGS2
08E9  03CC R 08E1 R                        DW      BRAN,DIGS1
08ED  0394 R                DIGS2:          DW      EXIT

                            ;   SIGN        ( n -- )
                            ;               Add a minus sign to the numeric output string.

                                           $COLON  4,'SIGN',SIGN
08EF                        2       SIGN:                                               ;
3BB9                        2       ORG    _NAME                                         ;
3BB9  08EF R 3BC7 R         2              DW       _CODE,_LINK                         ;
3BBD  04 53 49 47 4E        2              DB      4,'SIGN'                             ;
08EF                        2       ORG    _CODE                                         ;
08EF  80                    1              DB 80H                                       ;
08F0  0453 R                               DW      ZLESS
08F2  03B7 R 08FC R                        DW      QBRAN,SIGN1
08F6  038D R 002D 08C4 R                   DW      DOLIT,'-',HOLD
08FC  0394 R                SIGN1:          DW      EXIT
```

```

                            ;   #>          ( w -- b u )
                            ;               Prepare the output string to be TYPE'd.

                                           $COLON  2,'#>',EDIGS
08FE                        2       EDIGS:                                      ;
3BB1                        2       ORG     _NAME                               ;
3BB1  08FE R 3BBD R         2              DW       _CODE,_LINK                ;
3BB5  02 23 3E              2              DB      2,'#>'                    ;
08FE                        2       ORG     _CODE                               ;
08FE  80                    1              DB 80H                              ;
08FF  0436 R 04F3 R 03DE R                 DW      DROP,HLD,AT
0905  07E6 R 044A R 059B R                 DW      PAD,OVER,SUBB,EXIT
      0394 R

                            ;   str         ( n -- b u )
                            ;               Convert a signed integer to a numeric string.

                                           $COLON  3,'str',STR
090D                        2       STR:                                      ;
3BA9                        2       ORG     _NAME                               ;
3BA9  090D R 3BB5 R         2              DW       _CODE,_LINK                ;
3BAD  03 73 74 72           2              DB      3,'str'                   ;
090D                        2       ORG     _CODE                               ;
090D  80                    1              DB 80H                              ;
090E  043B R 0413 R 05AC R                 DW      DUPP,TOR,ABSS
0914  08BB R 08E0 R 0405 R                 DW      BDIGS,DIGS,RFROM
091A  08EF R 08FE R 0394 R                 DW      SIGN,EDIGS,EXIT

                            ;   HEX         ( -- )
                            ;               Use radix 16 as base for numeric conversions.

                                           $COLON  3,'HEX',HEX
0920                        2       HEX:                                      ;
3BA1                        2       ORG     _NAME                               ;
3BA1  0920 R 3BAD R         2              DW       _CODE,_LINK                ;
3BA5  03 48 45 58           2              DB      3,'HEX'                   ;
0920                        2       ORG     _CODE                               ;
0920  80                    1              DB 80H                              ;
0921  038D R 0010 04CB R                   DW      DOLIT,16,BASE,STORE,EXIT
      03D3 R 0394 R

                            ;   DECIMAL     ( -- )
                            ;               Use radix 10 as base for numeric conversions.

                                           $COLON  7,'DECIMAL',DECIM
092B                        2       DECIM:                                      ;
3B95                        2       ORG     _NAME                               ;
3B95  092B R 3BA5 R         2              DW       _CODE,_LINK                ;
3B99  07 44 45 43 49 4D 41  2              DB      7,'DECIMAL'                   ;
092B                        2       ORG     _CODE                               ;
092B  80                    1              DB 80H                              ;
092C  038D R 000A 04CB R                   DW      DOLIT,10,BASE,STORE,EXIT
      03D3 R 0394 R

                            ;; Numeric input, single precision

                            ;   DIGIT?      ( c base -- u t )
                            ;               Convert a character to its numeric value. A flag indicates 
success.

                                           $COLON  6,'DIGIT?',DIGTQ
0936                        2       DIGTQ:                                          ;
3B89                        2       ORG     _NAME                                   ;
3B89  0936 R 3B99 R         2              DW       _CODE,_LINK                    ;
3B8D  06 44 49 47 49 54 3F  2              DB      6,'DIGIT?'                        ;
0936                        2       ORG     _CODE                                   ;
0936  80                    1              DB 80H                                  ;
0937  0413 R 038D R 0030                   DW      TOR,DOLIT,'0',SUBB
      059B R
093F  038D R 0009 044A R                   DW      DOLIT,9,OVER,LESS
      05E1 R
0947  03B7 R 095B R                        DW      QBRAN,DGTQ1
094B  038D R 0007 059B R                   DW      DOLIT,7,SUBB
0951  043B R 038D R 000A                   DW      DUPP,DOLIT,10,LESS,ORR
      05E1 R 046A R
095B  043B R 0405 R 05C8 R  DGTQ1:          DW      DUPP,RFROM,ULESS,EXIT
      0394 R

                            ;   NUMBER?     ( a -- n T | a F )
                            ;               Convert a number string to integer. Push a flag on tos.

                                           $COLON  7,'NUMBER?',NUMBQ
0963                        2       NUMBQ:                                          ;
3B7D                        2       ORG     _NAME                                   ;
3B7D  0963 R 3B8D R         2              DW       _CODE,_LINK                    ;
3B81  07 4E 55 4D 42 45 52  2              DB      7,'NUMBER?'                        ;
0963                        2       ORG     _CODE                                   ;
0963  80                    1              DB 80H                                  ;
0964  04CB R 03DE R 0413 R                 DW      BASE,AT,TOR,DOLIT,0,OVER,COUNT
      038D R 0000 044A R
      07D4 R
0972  044A R 03F1 R 038D R                 DW      OVER,CAT,DOLIT,'$',EQUAL
      0024 05B9 R
097C  03B7 R 098A R                        DW      QBRAN,NUMQ1
0980  0920 R 0442 R 061F R                 DW      HEX,SWAP,ONEP
0986  0442 R 0626 R                        DW      SWAP,ONEM
098A  044A R 03F1 R 038D R  NUMQ1:          DW      OVER,CAT,DOLIT,'-',EQUAL,TOR
      002D 05B9 R 0413 R
0996  0442 R 040C R 059B R                 DW      SWAP,RAT,SUBB,SWAP,RAT,PLUS,QDUP
      0442 R 040C R 0565 R
      053D R
09A4  03B7 R 09EE R                        DW      QBRAN,NUMQ6
09A8  0626 R 0413 R                        DW      ONEM,TOR
09AC  043B R 0413 R 03F1 R  NUMQ2:          DW      DUPP,TOR,CAT,BASE,AT,DIGTQ
      04CB R 03DE R 0936 R
09B8  03B7 R 09E0 R                        DW      QBRAN,NUMQ4
09BC  0442 R 04CB R 03DE R                 DW      SWAP,BASE,AT,STAR,PLUS,RFROM
      0731 R 0565 R 0405 R
09C8  061F R                               DW      ONEP
09CA  039D R 09AC R                        DW      DONXT,NUMQ2
09CE  040C R 0442 R 0436 R                 DW      RAT,SWAP,DROP
09D4  03B7 R 09DA R                        DW      QBRAN,NUMQ3
```

```
09D8  057B R                                DW      NEGAT
09DA  0442 R                NUMQ3:           DW      SWAP
09DC  03CC R 09EC R                         DW      BRAN,NUMQ5
09E0  0405 R 0405 R 0555 R  NUMQ4:           DW      RFROM,RFROM,DDROP,DDROP,DOLIT,0
      0555 R 038D R 0000
09EC  043B R                NUMQ5:           DW      DUPP
09EE  0405 R 0555 R         NUMQ6:           DW      RFROM,DDROP
09F2  0405 R 04CB R 03D3 R                  DW      RFROM,BASE,STORE,EXIT
      0394 R

                            ;; Basic I/O

                            ;   ?KEY         ( -- c T | F )
                            ;                Return input character and true, or a false if no input.

                                            $COLON  4,'?KEY',QKEY
09FA                        2       QKEY:                                       ;
3B73                        2       ORG     _NAME                               ;
3B73  09FA R 3B81 R         2               DW       _CODE,_LINK                ;
3B77  04 3F 4B 45 59        2               DB      4,'?KEY'                    ;
09FA                        2       ORG     _CODE                               ;
09FA  80                    1               DB 80H                              ;
09FB  04AD R 07F6 R 0394 R                  DW      TQKEY,ATEXE,EXIT

                            ;   KEY          ( -- c )
                            ;                Wait for and return an input character.

                                            $COLON  3,'KEY',KEY
0A01                        2       KEY:                                        ;
3B6B                        2       ORG     _NAME                               ;
3B6B  0A01 R 3B77 R         2               DW       _CODE,_LINK                ;
3B6F  03 4B 45 59           2               DB      3,'KEY'                     ;
0A01                        2       ORG     _CODE                               ;
0A01  80                    1               DB 80H                              ;
0A02  09FA R                KEY1:            DW      QKEY
0A04  03B7 R 0A02 R                         DW      QBRAN,KEY1
0A08  0394 R                                DW      EXIT

                            ;   EMIT         ( c -- )
                            ;                Send a character to the output device.

                                            $COLON  4,'EMIT',EMIT
0A0A                        2       EMIT:                                       ;
3B61                        2       ORG     _NAME                               ;
3B61  0A0A R 3B6F R         2               DW       _CODE,_LINK                ;
3B65  04 45 4D 49 54        2               DB      4,'EMIT'                    ;
0A0A                        2       ORG     _CODE                               ;
0A0A  80                    1               DB 80H                              ;
0A0B  04B2 R 07F6 R 0394 R                  DW      TEMIT,ATEXE,EXIT

                            ;   NUF?         ( -- t )
                            ;                Return false if no input, else pause and if CR return true.

                                            $COLON  4,'NUF?',NUFQ
0A11                        2       NUFQ:                                       ;
3B57                        2       ORG     _NAME                               ;
```

```
3B57  0A11 R 3B65 R         2              DW       _CODE,_LINK                    ;
3B5B  04 4E 55 46 3F        2              DB      4,'NUF?'                        ;
0A11                        2       ORG     _CODE                                   ;
0A11  80                    1              DB 80H                                  ;
0A12  09FA R 043B R                        DW      QKEY,DUPP
0A16  03B7 R 0A24 R                        DW      QBRAN,NUFQ1
0A1A  0555 R 0A01 R 038D R                 DW      DDROP,KEY,DOLIT,CRR,EQUAL
      000D 05B9 R
0A24  0394 R                NUFQ1:          DW      EXIT

                            ;   PACE        ( -- )
                            ;               Send a pace character for the file downloading process.

                                           $COLON  4,'PACE',PACE
0A26                        2       PACE:                                          ;
3B4D                        2       ORG     _NAME                                   ;
3B4D  0A26 R 3B5B R         2              DW       _CODE,_LINK                    ;
3B51  04 50 41 43 45        2              DB      4,'PACE'                        ;
0A26                        2       ORG     _CODE                                   ;
0A26  80                    1              DB 80H                                  ;
0A27  038D R 000B 0A0A R                   DW      DOLIT,11,EMIT,EXIT
      0394 R

                            ;   SPACE       ( -- )
                            ;               Send the blank character to the output device.

                                           $COLON  5,'SPACE',SPACE
0A2F                        2       SPACE:                                          ;
3B43                        2       ORG     _NAME                                   ;
3B43  0A2F R 3B51 R         2              DW       _CODE,_LINK                    ;
3B47  05 53 50 41 43 45     2              DB      5,'SPACE'                        ;
0A2F                        2       ORG     _CODE                                   ;
0A2F  80                    1              DB 80H                                  ;
0A30  0767 R 0A0A R 0394 R                 DW      BLANK,EMIT,EXIT

                            ;   SPACES      ( +n -- )
                            ;               Send n spaces to the output device.

                                           $COLON  6,'SPACES',SPACS
0A36                        2       SPACS:                                          ;
3B37                        2       ORG     _NAME                                   ;
3B37  0A36 R 3B47 R         2              DW       _CODE,_LINK                    ;
3B3B  06 53 50 41 43 45 53  2              DB      6,'SPACES'                        ;
0A36                        2       ORG     _CODE                                   ;
0A36  80                    1              DB 80H                                  ;
0A37  038D R 0000 05F8 R                   DW      DOLIT,0,MAX,TOR
      0413 R
0A3F  03CC R 0A45 R                        DW      BRAN,CHAR2
0A43  0A2F R                CHAR1:          DW      SPACE
0A45  039D R 0A43 R         CHAR2:          DW      DONXT,CHAR1
0A49  0394 R                               DW      EXIT

                            ;   TYPE        ( b u -- )
                            ;               Output u characters from b.

                                           $COLON  4,'TYPE',TYPEE
```

```
0A4B                        2       TYPEE:                                         ;
3B2D                        2       ORG     _NAME                                  ;
3B2D  0A4B R 3B3B R         2               DW       _CODE,_LINK                   ;
3B31  04 54 59 50 45        2               DB      4,'TYPE'                       ;
0A4B                        2       ORG     _CODE                                  ;
0A4B  80                    1               DB 80H                                 ;
0A4C  0413 R                                DW      TOR
0A4E  03CC R 0A5A R                         DW      BRAN,TYPE2
0A52  043B R 03F1 R 0A0A R  TYPE1:           DW      DUPP,CAT,EMIT
0A58  061F R                                DW      ONEP
0A5A  039D R 0A52 R         TYPE2:           DW      DONXT,TYPE1
0A5E  0436 R 0394 R                         DW      DROP,EXIT

                            ;   CR           ( -- )
                            ;                Output a carriage return and a line feed.

                                            $COLON  2,'CR',CR
0A62                        2       CR:                                          ;
3B25                        2       ORG     _NAME                                  ;
3B25  0A62 R 3B31 R         2               DW       _CODE,_LINK                   ;
3B29  02 43 52              2               DB      2,'CR'                       ;
0A62                        2       ORG     _CODE                                  ;
0A62  80                    1               DB 80H                                 ;
0A63  038D R 000D 0A0A R                    DW      DOLIT,CRR,EMIT
0A69  038D R 000A 0A0A R                    DW      DOLIT,LF,EMIT,EXIT
      0394 R

                            ;   do$          ( -- a )
                            ;                Return the address of a compiled string.

                                            $COLON  COMPO+3,'do$',DOSTR
0A71                        2       DOSTR:                                         ;
3B1D                        2       ORG     _NAME                                  ;
3B1D  0A71 R 3B29 R         2               DW       _CODE,_LINK                   ;
3B21  43 64 6F 24           2               DB      COMPO+3,'do$'                      ;
0A71                        2       ORG     _CODE                                  ;
0A71  80                    1               DB 80H                                 ;
0A72  0405 R 040C R 0405 R                  DW      RFROM,RAT,RFROM,COUNT,PLUS
      07D4 R 0565 R
0A7C  0413 R 0442 R 0413 R                  DW      TOR,SWAP,TOR,EXIT
      0394 R

                            ;   $"|         ( -- a )
                            ;                Run time routine compiled by $". Return address of a compiled
string.

                                            $COLON  COMPO+3,'$"|',STRQP
0A84                        2       STRQP:                                         ;
3B15                        2       ORG     _NAME                                  ;
3B15  0A84 R 3B21 R         2               DW       _CODE,_LINK                   ;
3B19  43 24 22 7C           2               DB      COMPO+3,'$"|'                      ;
0A84                        2       ORG     _CODE                                  ;
0A84  80                    1               DB 80H                                 ;
0A85  0A71 R 0394 R                         DW      DOSTR,EXIT              ;force a call to do$

                            ;   ."|         ( -- )
                            ;                Run time routine of ." . Output a compiled string.
```

```

                                        $COLON  COMPO+3,'."|',DOTQP
0A89                         2       DOTQP:                                                ;
3B0D                         2       ORG     _NAME                                         ;
3B0D  0A89 R 3B19 R          2               DW       _CODE,_LINK                          ;
3B11  43 2E 22 7C            2               DB      COMPO+3,'."|'                              ;
0A89                         2       ORG     _CODE                                         ;
0A89  80                     1               DB 80H                                        ;
0A8A  0A71 R 07D4 R 0A4B R                   DW      DOSTR,COUNT,TYPEE,EXIT
      0394 R

                             ;   .R          ( n +n -- )
                             ;               Display an integer in a field of n columns, right justified.

                                        $COLON  2,'.R',DOTR
0A92                         2       DOTR:                                                ;
3B05                         2       ORG     _NAME                                         ;
3B05  0A92 R 3B11 R          2               DW       _CODE,_LINK                          ;
3B09  02 2E 52               2               DB      2,'.R'                              ;
0A92                         2       ORG     _CODE                                         ;
0A92  80                     1               DB 80H                                        ;
0A93  0413 R 090D R 0405 R                   DW      TOR,STR,RFROM,OVER,SUBB
      044A R 059B R
0A9D  0A36 R 0A4B R 0394 R                   DW      SPACS,TYPEE,EXIT

                             ;   U.R         ( u +n -- )
                             ;               Display an unsigned integer in n column, right justified.

                                        $COLON  3,'U.R',UDOTR
0AA3                         2       UDOTR:                                                ;
3AFD                         2       ORG     _NAME                                         ;
3AFD  0AA3 R 3B09 R          2               DW       _CODE,_LINK                          ;
3B01  03 55 2E 52            2               DB      3,'U.R'                             ;
0AA3                         2       ORG     _CODE                                         ;
0AA3  80                     1               DB 80H                                        ;
0AA4  0413 R 08BB R 08E0 R                   DW      TOR,BDIGS,DIGS,EDIGS
      08FE R
0AAC  0405 R 044A R 059B R                   DW      RFROM,OVER,SUBB
0AB2  0A36 R 0A4B R 0394 R                   DW      SPACS,TYPEE,EXIT

                             ;   U.          ( u -- )
                             ;               Display an unsigned integer in free format.

                                        $COLON  2,'U.',UDOT
0AB8                         2       UDOT:                                                ;
3AF5                         2       ORG     _NAME                                         ;
3AF5  0AB8 R 3B01 R          2               DW       _CODE,_LINK                          ;
3AF9  02 55 2E               2               DB      2,'U.'                              ;
0AB8                         2       ORG     _CODE                                         ;
0AB8  80                     1               DB 80H                                        ;
0AB9  08BB R 08E0 R 08FE R                   DW      BDIGS,DIGS,EDIGS
0ABF  0A2F R 0A4B R 0394 R                   DW      SPACE,TYPEE,EXIT

                             ;   .           ( w -- )
                             ;               Display an integer in free format, preceeded by a space.

```

```
                                             $COLON  1,'.',DOT
0AC5                          2       DOT:                                               ;
3AEF                          2       ORG     _NAME                                      ;
3AEF  0AC5 R 3AF9 R           2               DW       _CODE,_LINK                       ;
3AF3  01 2E                   2               DB      1,'.'                           ;
0AC5                          2       ORG     _CODE                                      ;
0AC5  80                      1               DB 80H                                     ;
0AC6  04CB R 03DE R 038D R                    DW      BASE,AT,DOLIT,10,XORR   ;?decimal
      000A 0473 R
0AD0  03B7 R 0AD8 R                           DW      QBRAN,DOT1
0AD4  0AB8 R 0394 R                           DW      UDOT,EXIT               ;no, display unsigned
0AD8  090D R 0A2F R 0A4B R   DOT1:             DW       STR,SPACE,TYPEE,EXIT    ;yes, display signed
      0394 R

                             ;   ?           ( a -- )
                             ;               Display the contents in a memory cell.

                                             $COLON  1,'?',QUEST
0AE0                          2       QUEST:                                             ;
3AE9                          2       ORG     _NAME                                      ;
3AE9  0AE0 R 3AF3 R           2               DW       _CODE,_LINK                       ;
3AED  01 3F                   2               DB      1,'?'                           ;
0AE0                          2       ORG     _CODE                                      ;
0AE0  80                      1               DB 80H                                     ;
0AE1  03DE R 0AC5 R 0394 R                    DW      AT,DOT,EXIT

                             ;; Parsing

                             ;   parse       ( b u c -- b u delta ; <string> )
                             ;               Scan string delimited by c. Return found string and its offset.

                                             $COLON  5,'parse',PARS
0AE7                          2       PARS:                                              ;
3ADF                          2       ORG     _NAME                                      ;
3ADF  0AE7 R 3AED R           2               DW       _CODE,_LINK                       ;
3AE3  05 70 61 72 73 65       2               DB      5,'parse'                          ;
0AE7                          2       ORG     _CODE                                      ;
0AE7  80                      1               DB 80H                                     ;
0AE8  04D0 R 03D3 R 044A R                    DW      TEMP,STORE,OVER,TOR,DUPP
      0413 R 043B R
0AF2  03B7 R 0B70 R                           DW      QBRAN,PARS8
0AF6  0626 R 04D0 R 03DE R                    DW      ONEM,TEMP,AT,BLANK,EQUAL
      0767 R 05B9 R
0B00  03B7 R 0B2A R                           DW      QBRAN,PARS3
0B04  0413 R                                  DW      TOR
0B06  0767 R 044A R 03F1 R   PARS1:            DW       BLANK,OVER,CAT          ;skip leading blanks ONLY
0B0C  059B R 0453 R 056F R                    DW      SUBB,ZLESS,INVER
0B12  03B7 R 0B28 R                           DW      QBRAN,PARS2
0B16  061F R                                  DW      ONEP
0B18  039D R 0B06 R                           DW      DONXT,PARS1
0B1C  0405 R 0436 R 038D R                    DW      RFROM,DROP,DOLIT,0,DUPP,EXIT
      0000 043B R 0394 R
0B28  0405 R                 PARS2:            DW       RFROM
0B2A  044A R 0442 R          PARS3:            DW       OVER,SWAP
0B2E  0413 R                                  DW      TOR
0B30  04D0 R 03DE R 044A R   PARS4:            DW       TEMP,AT,OVER,CAT,SUBB   ;scan for delimiter
```

```
      03F1 R 059B R
0B3A  04D0 R 03DE R 0767 R                  DW      TEMP,AT,BLANK,EQUAL
      05B9 R
0B42  03B7 R 0B48 R                         DW      QBRAN,PARS5
0B46  0453 R                                DW      ZLESS
0B48  03B7 R 0B5A R          PARS5:          DW      QBRAN,PARS6
0B4C  061F R                                DW      ONEP
0B4E  039D R 0B30 R                         DW      DONXT,PARS4
0B52  043B R 0413 R                         DW      DUPP,TOR
0B56  03CC R 0B64 R                         DW      BRAN,PARS7
0B5A  0405 R 0436 R 043B R   PARS6:          DW      RFROM,DROP,DUPP
0B60  061F R 0413 R                         DW      ONEP,TOR
0B64  044A R 059B R          PARS7:          DW      OVER,SUBB
0B68  0405 R 0405 R 059B R                  DW      RFROM,RFROM,SUBB,EXIT
      0394 R
0B70  044A R 0405 R 059B R   PARS8:          DW      OVER,RFROM,SUBB,EXIT
      0394 R

                             ;   PARSE       ( c -- b u ; <string> )
                             ;               Scan input stream and return counted string delimited by c.

                                            $COLON  5,'PARSE',PARSE
0B78                         2      PARSE:                                      ;
3AD5                         2      ORG     _NAME                               ;
3AD5  0B78 R 3AE3 R          2              DW       _CODE,_LINK               ;
3AD9  05 50 41 52 53 45      2              DB      5,'PARSE'                   ;
0B78                         2      ORG     _CODE                               ;
0B78  80                     1              DB 80H                              ;
0B79  0413 R 07ED R 04DA R                  DW      TOR,TIB,INN,AT,PLUS     ;current input buffer pointer
      03DE R 0565 R
0B83  04DF R 03DE R 04DA R                  DW      NTIB,AT,INN,AT,SUBB     ;remaining count
      03DE R 059B R
0B8D  0405 R 0AE7 R 04DA R                  DW      RFROM,PARS,INN,PSTOR,EXIT
      07AB R 0394 R

                             ;   .(          ( -- )
                             ;               Output following string up to next ) .

                                            $COLON  IMEDD+2,'.(',DOTPR
0B97                         2      DOTPR:                                      ;
3ACD                         2      ORG     _NAME                               ;
3ACD  0B97 R 3AD9 R          2              DW       _CODE,_LINK               ;
3AD1  82 2E 28               2              DB      IMEDD+2,'.('                    ;
0B97                         2      ORG     _CODE                               ;
0B97  80                     1              DB 80H                              ;
0B98  038D R 0029 0B78 R                    DW      DOLIT,')',PARSE,TYPEE,EXIT
      0A4B R 0394 R

                             ;   (           ( -- )
                             ;               Ignore following string up to next ) . A comment.

                                            $COLON  IMEDD+1,'(',PAREN
0BA2                         2      PAREN:                                      ;
3AC7                         2      ORG     _NAME                               ;
3AC7  0BA2 R 3AD1 R          2              DW       _CODE,_LINK               ;
3ACB  81 28                  2              DB      IMEDD+1,'('                    ;
0BA2                        2       ORG     _CODE                                  ;
0BA2  80                    1               DB 80H                                 ;
0BA3  038D R 0029 0B78 R                    DW      DOLIT,')',PARSE,DDROP,EXIT
      0555 R 0394 R

                             ;   \            ( -- )
                             ;                Ignore following text till the end of line.

                                            $COLON  IMEDD+1,'\',BKSLA
0BAD                        2       BKSLA:                                         ;
3AC1                        2       ORG     _NAME                                  ;
3AC1  0BAD R 3ACB R         2               DW       _CODE,_LINK                   ;
3AC5  81 5C                 2               DB      IMEDD+1,'\'                       ;
0BAD                        2       ORG     _CODE                                  ;
0BAD  80                    1               DB 80H                                 ;
0BAE  04DF R 03DE R 04DA R                  DW      NTIB,AT,INN,STORE,EXIT
      03D3 R 0394 R

                             ;   CHAR         ( -- c )
                             ;                Parse next word and return its first character.

                                            $COLON  4,'CHAR',CHAR
0BB8                        2       CHAR:                                         ;
3AB7                        2       ORG     _NAME                                  ;
3AB7  0BB8 R 3AC5 R         2               DW       _CODE,_LINK                   ;
3ABB  04 43 48 41 52        2               DB      4,'CHAR'                       ;
0BB8                        2       ORG     _CODE                                  ;
0BB8  80                    1               DB 80H                                 ;
0BB9  0767 R 0B78 R 0436 R                  DW      BLANK,PARSE,DROP,CAT,EXIT
      03F1 R 0394 R

                             ;   TOKEN        ( -- a ; <string> )
                             ;                Parse a word from input stream and copy it to name dictionary.

                                            $COLON  5,'TOKEN',TOKEN
0BC3                        2       TOKEN:                                         ;
3AAD                        2       ORG     _NAME                                  ;
3AAD  0BC3 R 3ABB R         2               DW       _CODE,_LINK                   ;
3AB1  05 54 4F 4B 45 4E     2               DB      5,'TOKEN'                       ;
0BC3                        2       ORG     _CODE                                  ;
0BC3  80                    1               DB 80H                                 ;
0BC4  0767 R 0B78 R 038D R                  DW      BLANK,PARSE,DOLIT,31,MIN
      001F 0604 R
0BCE  0516 R 03DE R 044A R                  DW      NP,AT,OVER,SUBB,TWOM
      059B R 0635 R
0BD8  0862 R 0394 R                         DW      PACKS,EXIT

                             ;   WORD         ( c -- a ; <string> )
                             ;                Parse a word from input stream and copy it to code dictionary.

                                            $COLON  4,'WORD',WORDD
0BDC                        2       WORDD:                                         ;
3AA3                        2       ORG     _NAME                                  ;
3AA3  0BDC R 3AB1 R         2               DW       _CODE,_LINK                   ;
3AA7  04 57 4F 52 44        2               DB      4,'WORD'                       ;
0BDC                        2       ORG     _CODE                                  ;
```

```
0BDC  80                    1              DB 80H                                  ;
0BDD  0B78 R 07DF R 0862 R                 DW      PARSE,HERE,PACKS,EXIT
      0394 R

                            ;; Dictionary search

                            ;   NAME>       ( na -- ca )
                            ;                Return a code address given a name address.

                                           $COLON  5,'NAME>',NAMET
0BE5                        2       NAMET:                                          ;
3A99                        2       ORG     _NAME                                   ;
3A99  0BE5 R 3AA7 R         2              DW       _CODE,_LINK                    ;
3A9D  05 4E 41 4D 45 3E     2              DB      5,'NAME>'                        ;
0BE5                        2       ORG     _CODE                                   ;
0BE5  80                    1              DB 80H                                  ;
0BE6  0635 R 0635 R 03DE R                 DW      TWOM,TWOM,AT,EXIT
      0394 R

                            ;   SAME?       ( a a u -- a a f \ -0+ )
                            ;                Compare u cells in two strings. Return 0 if identical.

                                           $COLON  5,'SAME?',SAMEQ
0BEE                        2       SAMEQ:                                          ;
3A8F                        2       ORG     _NAME                                   ;
3A8F  0BEE R 3A9D R         2              DW       _CODE,_LINK                    ;
3A93  05 53 41 4D 45 3F     2              DB      5,'SAME?'                        ;
0BEE                        2       ORG     _CODE                                   ;
0BEE  80                    1              DB 80H                                  ;
0BEF  0413 R                               DW      TOR
0BF1  03CC R 0C17 R                        DW      BRAN,SAME2
0BF5  044A R 040C R 063D R  SAME1:          DW      OVER,RAT,TWOSL,PLUS,AT
      0565 R 03DE R
0BFF  044A R 040C R 063D R                 DW      OVER,RAT,TWOSL,PLUS,AT
      0565 R 03DE R
0C09  059B R 053D R                        DW      SUBB,QDUP
0C0D  03B7 R 0C17 R                        DW      QBRAN,SAME2
0C11  0405 R 0436 R 0394 R                 DW      RFROM,DROP,EXIT         ;strings not equal
0C17  039D R 0BF5 R         SAME2:          DW      DONXT,SAME1
0C1B  038D R 0000 0394 R                   DW      DOLIT,0,EXIT            ;strings equal

                            ;   find        ( a va -- ca na | a F )
                            ;                Search a vocabulary for a string. Return ca and na if succeeded.

                                           $COLON  4,'find',FIND
0C21                        2       FIND:                                         ;
3A85                        2       ORG     _NAME                                   ;
3A85  0C21 R 3A93 R         2              DW       _CODE,_LINK                    ;
3A89  04 66 69 6E 64        2              DB      4,'find'                        ;
0C21                        2       ORG     _CODE                                   ;
0C21  80                    1              DB 80H                                  ;
0C22  0442 R 043B R 03F1 R                 DW      SWAP,DUPP,CAT
0C28  038D R 0002 06F1 R                   DW      DOLIT,CELLL,SLASH,TEMP,STORE
      04D0 R 03D3 R
0C32  043B R 03DE R 0413 R                 DW      DUPP,AT,TOR,TWOP,SWAP
      062D R 0442 R
```

```
0C3C  03DE R 043B R          FIND1:          DW      AT,DUPP
0C40  03B7 R 0C6C R                          DW      QBRAN,FIND6
0C44  043B R 03DE R 038D R                   DW      DUPP,AT,DOLIT,MASKK,ANDD,RAT,XORR
      7F1F 0461 R 040C R
      0473 R
0C52  03B7 R 0C60 R                          DW      QBRAN,FIND2
0C56  062D R 038D R FFFF                     DW      TWOP,DOLIT,-1           ;true flag
0C5C  03CC R 0C68 R                          DW      BRAN,FIND3
0C60  062D R 04D0 R 03DE R   FIND2:          DW      TWOP,TEMP,AT,SAMEQ
      0BEE R
0C68  03CC R 0C78 R          FIND3:          DW      BRAN,FIND4
0C6C  0405 R 0436 R          FIND6:          DW      RFROM,DROP
0C70  0442 R 0635 R 0442 R                   DW      SWAP,TWOM,SWAP,EXIT
      0394 R
0C78  03B7 R 0C84 R          FIND4:          DW      QBRAN,FIND5
0C7C  0635 R 0635 R                          DW      TWOM,TWOM
0C80  03CC R 0C3C R                          DW      BRAN,FIND1
0C84  0405 R 0436 R 0442 R   FIND5:          DW      RFROM,DROP,SWAP,DROP
      0436 R
0C8C  0635 R                                 DW      TWOM
0C8E  043B R 0BE5 R 0442 R                   DW      DUPP,NAMET,SWAP,EXIT
      0394 R

                             ;   NAME?       ( a -- ca na | a F )
                             ;               Search all context vocabularies for a string.

                                             $COLON  5,'NAME?',NAMEQ
0C96                         2       NAMEQ:                                          ;
3A7B                         2       ORG     _NAME                                   ;
3A7B  0C96 R 3A89 R          2               DW      _CODE,_LINK                     ;
3A7F  05 4E 41 4D 45 3F      2               DB      5,'NAME?'                       ;
0C96                         2       ORG     _CODE                                   ;
0C96  80                     1               DB 80H                                  ;
0C97  04FD R 043B R 07C7 R                   DW      CNTXT,DUPP,DAT,XORR     ;?context=also
      0473 R
0C9F  03B7 R 0CA5 R                          DW      QBRAN,NAMQ1
0CA3  0635 R                                 DW      TWOM                    ;no, start with context
0CA5  0413 R                 NAMQ1:          DW      TOR
0CA7  0405 R 062D R 043B R   NAMQ2:          DW      RFROM,TWOP,DUPP,TOR     ;next in search order
      0413 R
0CAF  03DE R 053D R                          DW      AT,QDUP
0CB3  03B7 R 0CC5 R                          DW      QBRAN,NAMQ3
0CB7  0C21 R 053D R                          DW      FIND,QDUP               ;search vocabulary
0CBB  03B7 R 0CA7 R                          DW      QBRAN,NAMQ2
0CBF  0405 R 0436 R 0394 R                   DW      RFROM,DROP,EXIT         ;found name
0CC5  0405 R 0436 R          NAMQ3:          DW      RFROM,DROP              ;name not found
0CC9  038D R 0000 0394 R                     DW      DOLIT,0,EXIT            ;false flag

                             ;; Terminal response

                             ;   ^H          ( bot eot cur -- bot eot cur )
                             ;               Backup the cursor by one character.

                                             $COLON  2,'^H',BKSP
0CCF                         2       BKSP:                                           ;
3A73                         2       ORG     _NAME                                   ;
```

```
3A73  0CCF R 3A7F R         2              DW       _CODE,_LINK                    ;
3A77  02 5E 48              2              DB      2,'^H'                        ;
0CCF                        2       ORG     _CODE                                  ;
0CCF  80                    1              DB 80H                                 ;
0CD0  0413 R 044A R 0405 R                 DW      TOR,OVER,RFROM,SWAP,OVER,XORR
      0442 R 044A R 0473 R
0CDC  03B7 R 0CF8 R                        DW      QBRAN,BACK1
0CE0  038D R 0008 04C1 R                   DW      DOLIT,BKSPP,TECHO,ATEXE,ONEM
      07F6 R 0626 R
0CEA  0767 R 04C1 R 07F6 R                 DW      BLANK,TECHO,ATEXE
0CF0  038D R 0008 04C1 R                   DW      DOLIT,BKSPP,TECHO,ATEXE
      07F6 R
0CF8  0394 R               BACK1:           DW      EXIT

                           ;   TAP          ( bot eot cur c -- bot eot cur )
                           ;                Accept and echo the key stroke and bump the cursor.

                                           $COLON  3,'TAP',TAP
0CFA                        2       TAP:                                          ;
3A6B                        2       ORG     _NAME                                  ;
3A6B  0CFA R 3A77 R         2              DW       _CODE,_LINK                   ;
3A6F  03 54 41 50           2              DB      3,'TAP'                        ;
0CFA                        2       ORG     _CODE                                  ;
0CFA  80                    1              DB 80H                                 ;
0CFB  043B R 04C1 R 07F6 R                 DW      DUPP,TECHO,ATEXE
0D01  044A R 03E9 R 061F R                 DW      OVER,CSTOR,ONEP,EXIT
      0394 R

                           ;   kTAP         ( bot eot cur c -- bot eot cur )
                           ;                Process a key stroke, CR or backspace.

                                           $COLON  4,'kTAP',KTAP
0D09                        2       KTAP:                                         ;
3A61                        2       ORG     _NAME                                  ;
3A61  0D09 R 3A6F R         2              DW       _CODE,_LINK                   ;
3A65  04 6B 54 41 50        2              DB      4,'kTAP'                       ;
0D09                        2       ORG     _CODE                                  ;
0D09  80                    1              DB 80H                                 ;
0D0A  043B R 038D R 000D                   DW      DUPP,DOLIT,CRR,XORR
      0473 R
0D12  03B7 R 0D2A R                        DW      QBRAN,KTAP2
0D16  038D R 0008 0473 R                   DW      DOLIT,BKSPP,XORR
0D1C  03B7 R 0D26 R                        DW      QBRAN,KTAP1
0D20  0767 R 0CFA R 0394 R                 DW      BLANK,TAP,EXIT
0D26  0CCF R 0394 R        KTAP1:           DW      BKSP,EXIT
0D2A  0436 R 0442 R 0436 R  KTAP2:          DW      DROP,SWAP,DROP,DUPP,EXIT
      043B R 0394 R

                           ;   accept       ( b u -- b u )
                           ;                Accept characters to input buffer. Return with actual count.

                                           $COLON  6,'accept',ACCEP
0D34                        2       ACCEP:                                         ;
3A55                        2       ORG     _NAME                                  ;
3A55  0D34 R 3A65 R         2              DW       _CODE,_LINK                   ;
3A59  06 61 63 63 65 70 74  2              DB      6,'accept'                       ;
```

```
0D34                         2       ORG     _CODE                                  ;
0D34  80                     1               DB 80H                                 ;
0D35  044A R 0565 R 044A R                   DW      OVER,PLUS,OVER
0D3B  055B R 0473 R           ACCP1:          DW      DDUP,XORR
0D3F  03B7 R 0D61 R                          DW      QBRAN,ACCP4
0D43  0A01 R 043B R                          DW      KEY,DUPP
                             ;                DW      BLANK,SUBB,DOLIT,95,ULESS
0D47  0767 R 038D R 007F                     DW      BLANK,DOLIT,127,WITHI
      0610 R
0D4F  03B7 R 0D59 R                          DW      QBRAN,ACCP2
0D53  0CFA R                                 DW      TAP
0D55  03CC R 0D5D R                          DW      BRAN,ACCP3
0D59  04BC R 07F6 R           ACCP2:          DW      TTAP,ATEXE
0D5D  03CC R 0D3B R           ACCP3:          DW      BRAN,ACCP1
0D61  0436 R 044A R 059B R    ACCP4:          DW      DROP,OVER,SUBB,EXIT
      0394 R

                             ;   EXPECT      ( b u -- )
                             ;               Accept input stream and store count in SPAN.

                                             $COLON  6,'EXPECT',EXPEC
0D69                         2       EXPEC:                                         ;
3A49                         2       ORG     _NAME                                  ;
3A49  0D69 R 3A59 R          2               DW       _CODE,_LINK                   ;
3A4D  06 45 58 50 45 43 54   2               DB      6,'EXPECT'                       ;
0D69                         2       ORG     _CODE                                  ;
0D69  80                     1               DB 80H                                 ;
0D6A  04B7 R 07F6 R 04D5 R                   DW      TEXPE,ATEXE,SPAN,STORE,DROP,EXIT
      03D3 R 0436 R 0394 R

                             ;   QUERY       ( -- )
                             ;               Accept input stream to terminal input buffer.

                                             $COLON  5,'QUERY',QUERY
0D76                         2       QUERY:                                         ;
3A3F                         2       ORG     _NAME                                  ;
3A3F  0D76 R 3A4D R          2               DW       _CODE,_LINK                   ;
3A43  05 51 55 45 52 59      2               DB      5,'QUERY'                       ;
0D76                         2       ORG     _CODE                                  ;
0D76  80                     1               DB 80H                                 ;
0D77  07ED R 038D R 0050                     DW      TIB,DOLIT,80,TEXPE,ATEXE,NTIB,STORE
      04B7 R 07F6 R 04DF R
      03D3 R
0D85  0436 R 038D R 0000                     DW      DROP,DOLIT,0,INN,STORE,EXIT
      04DA R 03D3 R 0394 R

                             ;; Error handling

                             ;   CATCH       ( ca -- 0 | err# )
                             ;               Execute word at ca and set up an error frame for it.

                                             $COLON  5,'CATCH',CATCH
0D91                         2       CATCH:                                         ;
3A35                         2       ORG     _NAME                                  ;
3A35  0D91 R 3A43 R          2               DW       _CODE,_LINK                   ;
3A39  05 43 41 54 43 48      2               DB      5,'CATCH'                       ;
```

```
0D91                        2       ORG     _CODE                                    ;
0D91  80                    1               DB 80H                                   ;
0D92  041C R 0413 R 04F8 R                  DW      SPAT,TOR,HANDL,AT,TOR   ;save error frame
      03DE R 0413 R
0D9C  03FB R 04F8 R 03D3 R                  DW      RPAT,HANDL,STORE,EXECU  ;execute
      039B R
0DA4  0405 R 04F8 R 03D3 R                  DW      RFROM,HANDL,STORE       ;restore error frame
0DAA  0405 R 0436 R 038D R                  DW      RFROM,DROP,DOLIT,0,EXIT ;no error
      0000 0394 R

                            ;   THROW       ( err# -- err# )
                            ;               Reset system to current local error frame an update error flag.

                                            $COLON  5,'THROW',THROW
0DB4                        2       THROW:                                           ;
3A2B                        2       ORG     _NAME                                    ;
3A2B  0DB4 R 3A39 R         2               DW       _CODE,_LINK                    ;
3A2F  05 54 48 52 4F 57     2               DB      5,'THROW'                        ;
0DB4                        2       ORG     _CODE                                    ;
0DB4  80                    1               DB 80H                                   ;
0DB5  04F8 R 03DE R 0400 R                  DW      HANDL,AT,RPSTO          ;restore return stack
0DBB  0405 R 04F8 R 03D3 R                  DW      RFROM,HANDL,STORE       ;restore handler frame
0DC1  0405 R 0442 R 0413 R                  DW      RFROM,SWAP,TOR,SPSTO    ;restore data stack
      0429 R
0DC9  0436 R 0405 R 0394 R                  DW      DROP,RFROM,EXIT

                            ;   NULL$       ( -- a )
                            ;               Return address of a null string with zero count.

                                            $COLON  5,'NULL$',NULLS
0DCF                        2       NULLS:                                           ;
3A21                        2       ORG     _NAME                                    ;
3A21  0DCF R 3A2F R         2               DW       _CODE,_LINK                    ;
3A25  05 4E 55 4C 4C 24     2               DB      5,'NULL$'                        ;
0DCF                        2       ORG     _CODE                                    ;
0DCF  80                    1               DB 80H                                   ;
0DD0  048C R                                DW      DOVAR                   ;emulate CREATE
0DD2  0000                                  DW      0
0DD4  63 6F 79 6F 74 65                     DB      99,111,121,111,116,101

                            ;   ABORT       ( -- )
                            ;               Reset data stack and jump to QUIT.

                                            $COLON  5,'ABORT',ABORT
0DDA                        2       ABORT:                                           ;
3A17                        2       ORG     _NAME                                    ;
3A17  0DDA R 3A25 R         2               DW       _CODE,_LINK                    ;
3A1B  05 41 42 4F 52 54     2               DB      5,'ABORT'                        ;
0DDA                        2       ORG     _CODE                                    ;
0DDA  80                    1               DB 80H                                   ;
0DDB  0DCF R 0DB4 R                         DW      NULLS,THROW

                            ;   abort"     ( f -- )
                            ;               Run time routine of ABORT" . Abort with a message.

                                            $COLON  COMPO+6,'abort"',ABORQ
```

```
0DDF                          2       ABORQ:                                          ;
3A0B                          2       ORG     _NAME                                   ;
3A0B  0DDF R 3A1B R           2               DW       _CODE,_LINK                   ;
3A0F  46 61 62 6F 72 74 22    2               DB      COMPO+6,'abort"'                        ;
0DDF                          2       ORG     _CODE                                   ;
0DDF  80                      1               DB 80H                                 ;
0DE0  03B7 R 0DE8 R                           DW      QBRAN,ABOR1             ;text flag
0DE4  0A71 R 0DB4 R                           DW      DOSTR,THROW             ;pass error string
0DE8  0A71 R 0436 R 0394 R   ABOR1:           DW      DOSTR,DROP,EXIT         ;drop error

                             ;; The text interpreter

                             ;   $INTERPRET  ( a -- )
                             ;               Interpret a word. If failed, try to convert it to an integer.

                                              $COLON  10,'$INTERPRET',INTER
0DEE                          2       INTER:                                          ;
39FB                          2       ORG     _NAME                                   ;
39FB  0DEE R 3A0F R           2               DW       _CODE,_LINK                   ;
39FF  0A 24 49 4E 54 45 52    2               DB      10,'$INTERPRET'                        ;
0DEE                          2       ORG     _CODE                                   ;
0DEE  80                      1               DB 80H                                 ;
0DEF  0C96 R 053D R                           DW      NAMEQ,QDUP              ;?defined
0DF3  03B7 R 0E13 R                           DW      QBRAN,INTE1
0DF7  03DE R 038D R 0040                      DW      AT,DOLIT,COMPO,ANDD     ;?compile only lexicon bits
      0461 R
                                              D$      ABORQ,' compile only'
0DFF  0DDF R                  1               DW      ABORQ                          ;
0E01  00 20 63 6F 6D 70 69    1               DB      0,' compile only'                        ;
0E01                          1       ORG     _LEN                                    ;
0E01  0D                      1               DB      _CODE-_LEN-1                   ;
0E0F                          1       ORG     _CODE                                   ;
0E0F  039B R 0394 R                           DW      EXECU,EXIT              ;execute defined word
0E13  04EE R 07F6 R          INTE1:           DW      TNUMB,ATEXE             ;convert a number
0E17  03B7 R 0E1D R                           DW      QBRAN,INTE2
0E1B  0394 R                                  DW      EXIT
0E1D  0DB4 R                 INTE2:           DW      THROW                   ;error

                             ;   [           ( -- )
                             ;               Start the text interpreter.

                                              $COLON  IMEDD+1,'[',LBRAC
0E1F                          2       LBRAC:                                          ;
39F5                          2       ORG     _NAME                                   ;
39F5  0E1F R 39FF R           2               DW       _CODE,_LINK                   ;
39F9  81 5B                   2               DB      IMEDD+1,'['                  ;
0E1F                          2       ORG     _CODE                                   ;
0E1F  80                      1               DB 80H                                 ;
0E20  038D R 0DEE R 04E9 R                    DW      DOLIT,INTER,TEVAL,STORE,EXIT
      03D3 R 0394 R

                             ;   .OK         ( -- )
                             ;               Display 'ok' only while interpreting.

                                              $COLON  3,'.OK',DOTOK
0E2A                          2       DOTOK:                                          ;
```

```
39ED                        2       ORG     _NAME                               ;
39ED  0E2A R 39F9 R         2               DW       _CODE,_LINK               ;
39F1  03 2E 4F 4B           2               DB      3,'.OK'                   ;
0E2A                        2       ORG     _CODE                               ;
0E2A  80                    1               DB 80H                             ;
0E2B  038D R 0DEE R 04E9 R                  DW      DOLIT,INTER,TEVAL,AT,EQUAL
      03DE R 05B9 R
0E35  03B7 R 0E3F R                         DW      QBRAN,DOTO1
                                            D$      DOTQP,' ok'
0E39  0A89 R                1               DW      DOTQP                      ;
0E3B  00 20 6F 6B           1               DB      0,' ok'                    ;
0E3B                        1       ORG     _LEN                                ;
0E3B  03                    1               DB      _CODE-_LEN-1               ;
0E3F                        1       ORG     _CODE                               ;
0E3F  0A62 R 0394 R         DOTO1:           DW      CR,EXIT

                            ;   ?STACK      ( -- )
                            ;                Abort if the data stack underflows.

                                            $COLON  6,'?STACK',QSTAC
0E43                        2       QSTAC:                                      ;
39E1                        2       ORG     _NAME                               ;
39E1  0E43 R 39F1 R         2               DW       _CODE,_LINK               ;
39E5  06 3F 53 54 41 43 4B  2               DB      6,'?STACK'                  ;
0E43                        2       ORG     _CODE                               ;
0E43  80                    1               DB 80H                             ;
0E44  078B R 0453 R                         DW      DEPTH,ZLESS             ;check only for underflow
                                            D$      ABORQ,' underflow'
0E48  0DDF R                1               DW      ABORQ                      ;
0E4A  00 20 75 6E 64 65 72  1               DB      0,' underflow'                      ;
0E4A                        1       ORG     _LEN                                ;
0E4A  0A                    1               DB      _CODE-_LEN-1               ;
0E55                        1       ORG     _CODE                               ;
0E55  0394 R                                DW      EXIT

                            ;   EVAL        ( -- )
                            ;                Interpret the input stream.

                                            $COLON  4,'EVAL',EVAL
0E57                        2       EVAL:                                      ;
39D7                        2       ORG     _NAME                               ;
39D7  0E57 R 39E5 R         2               DW       _CODE,_LINK               ;
39DB  04 45 56 41 4C        2               DB      4,'EVAL'                   ;
0E57                        2       ORG     _CODE                               ;
0E57  80                    1               DB 80H                             ;
0E58  0BC3 R 043B R 03F1 R  EVAL1:           DW      TOKEN,DUPP,CAT           ;?input stream empty
0E5E  03B7 R 0E6C R                         DW      QBRAN,EVAL2
0E62  04E9 R 07F6 R 0E43 R                  DW      TEVAL,ATEXE,QSTAC       ;evaluate input, check stack
0E68  03CC R 0E58 R                         DW      BRAN,EVAL1
0E6C  0436 R 04C6 R 07F6 R  EVAL2:           DW      DROP,TPROM,ATEXE,EXIT   ;prompt
      0394 R

                            ;; Shell

                            ;   PRESET      ( -- )
                            ;                Reset data stack pointer and the terminal input buffer.
```

```

                                             $COLON  6,'PRESET',PRESE
0E74                        2       PRESE:                                     ;
39CB                        2       ORG     _NAME                              ;
39CB  0E74 R 39DB R         2               DW       _CODE,_LINK              ;
39CF  06 50 52 45 53 45 54  2               DB      6,'PRESET'                  ;
0E74                        2       ORG     _CODE                              ;
0E74  80                    1               DB 80H                            ;
0E75  04A3 R 03DE R 0429 R                  DW      SZERO,AT,SPSTO
0E7B  038D R C200 04DF R                    DW      DOLIT,TIBB,NTIB,TWOP,STORE,EXIT
      062D R 03D3 R 0394 R

                            ;   xio         ( a a a -- )
                            ;               Reset the I/O vectors 'EXPECT, 'TAP, 'ECHO and 'PROMPT.

                                            $COLON  COMPO+3,'xio',XIO
0E87                        2       XIO:                                    ;
39C3                        2       ORG     _NAME                              ;
39C3  0E87 R 39CF R         2               DW       _CODE,_LINK              ;
39C7  43 78 69 6F           2               DB      COMPO+3,'xio'                   ;
0E87                        2       ORG     _CODE                              ;
0E87  80                    1               DB 80H                            ;
0E88  038D R 0D34 R 04B7 R                  DW      DOLIT,ACCEP,TEXPE,DSTOR
      07BA R
0E90  04C1 R 07BA R 0394 R                  DW      TECHO,DSTOR,EXIT

                            ;   FILE        ( -- )
                            ;               Select I/O vectors for file download.

                                            $COLON  4,'FILE',FILE
0E96                        2       FILE:                                    ;
39B9                        2       ORG     _NAME                              ;
39B9  0E96 R 39C7 R         2               DW       _CODE,_LINK              ;
39BD  04 46 49 4C 45        2               DB      4,'FILE'                  ;
0E96                        2       ORG     _CODE                              ;
0E96  80                    1               DB 80H                            ;
0E97  038D R 0A26 R 038D R                  DW      DOLIT,PACE,DOLIT,DROP
      0436 R
0E9F  038D R 0D09 R 0E87 R                  DW      DOLIT,KTAP,XIO,EXIT
      0394 R

                            ;   HAND        ( -- )
                            ;               Select I/O vectors for terminal interface.

                                            $COLON  4,'HAND',HAND
0EA7                        2       HAND:                                    ;
39AF                        2       ORG     _NAME                              ;
39AF  0EA7 R 39BD R         2               DW       _CODE,_LINK              ;
39B3  04 48 41 4E 44        2               DB      4,'HAND'                  ;
0EA7                        2       ORG     _CODE                              ;
0EA7  80                    1               DB 80H                            ;
0EA8  038D R 0E2A R 038D R                  DW      DOLIT,DOTOK,DOLIT,EMIT
      0A0A R
0EB0  038D R 0D09 R 0E87 R                  DW      DOLIT,KTAP,XIO,EXIT
      0394 R

```

```
                            ;   I/O         ( -- a )
                            ;               Array to store default I/O vectors.

                                           $COLON  3,'I/O',ISLO
0EB8                        2       ISLO:                                          ;
39A7                        2       ORG     _NAME                                   ;
39A7  0EB8 R 39B3 R         2               DW       _CODE,_LINK                    ;
39AB  03 49 2F 4F           2               DB      3,'I/O'                        ;
0EB8                        2       ORG     _CODE                                   ;
0EB8  80                    1               DB 80H                                  ;
0EB9  048C R                                DW      DOVAR                   ;emulate CREATE
0EBB  032E R 0347 R                         DW      QRX,TXSTO               ;default I/O vectors

                            ;   CONSOLE     ( -- )
                            ;               Initiate terminal interface.

                                           $COLON  7,'CONSOLE',CONSO
0EBF                        2       CONSO:                                          ;
399B                        2       ORG     _NAME                                   ;
399B  0EBF R 39AB R         2               DW       _CODE,_LINK                    ;
399F  07 43 4F 4E 53 4F 4C  2               DB      7,'CONSOLE'                        ;
0EBF                        2       ORG     _CODE                                   ;
0EBF  80                    1               DB 80H                                  ;
0EC0  0EB8 R 07C7 R 04AD R                  DW      ISLO,DAT,TQKEY,DSTOR    ;restore default I/O device
      07BA R
0EC8  0EA7 R 0394 R                         DW      HAND,EXIT               ;keyboard input

                            ;   QUIT        ( -- )
                            ;               Reset return stack pointer and start text interpreter.

                                           $COLON  4,'QUIT',QUIT
0ECC                        2       QUIT:                                          ;
3991                        2       ORG     _NAME                                   ;
3991  0ECC R 399F R         2               DW       _CODE,_LINK                    ;
3995  04 51 55 49 54        2               DB      4,'QUIT'                        ;
0ECC                        2       ORG     _CODE                                   ;
0ECC  80                    1               DB 80H                                  ;
0ECD  04A8 R 03DE R 0400 R                  DW      RZERO,AT,RPSTO          ;reset return stack pointer
0ED3  0E1F R                QUIT1:           DW      LBRAC                   ;start interpretation
0ED5  0D76 R                QUIT2:           DW      QUERY                   ;get input
0ED7  038D R 0E57 R 0D91 R                  DW      DOLIT,EVAL,CATCH,QDUP   ;evaluate input
      053D R
0EDF  03B7 R 0ED5 R                         DW      QBRAN,QUIT2             ;continue till error
0EE3  04C6 R 03DE R 0442 R                  DW      TPROM,AT,SWAP           ;save input device
0EE9  0EBF R 0DCF R 044A R                  DW      CONSO,NULLS,OVER,XORR   ;?display error message
      0473 R
0EF1  03B7 R 0F01 R                         DW      QBRAN,QUIT3
0EF5  0A2F R 07D4 R 0A4B R                  DW      SPACE,COUNT,TYPEE       ;error message
                                            D$      DOTQP,' ? '             ;error prompt
0EFB  0A89 R                1               DW      DOTQP                   ;
0EFD  00 20 3F 20           1               DB      0,' ? '                 ;
0EFD                        1       ORG     _LEN                                    ;
0EFD  03                    1               DB      _CODE-_LEN-1                    ;
0F01                        1       ORG     _CODE                                   ;
0F01  038D R 0E2A R 0473 R  QUIT3:           DW      DOLIT,DOTOK,XORR        ;?file input
0F07  03B7 R 0F11 R                         DW      QBRAN,QUIT4
```

```
0F0B  038D R 001B 0A0A R                        DW      DOLIT,ERR,EMIT          ;file error, tell host
0F11  0E74 R                QUIT4:              DW      PRESE                   ;some cleanup
0F13  03CC R 0ED3 R                             DW      BRAN,QUIT1

                            ;; The compiler

                            ;   '               ( -- ca )
                            ;                   Search context vocabularies for the next word in input stream.

                                                $COLON  1,"'",TICK
0F17                        2       TICK:                                           ;
398B                        2       ORG         _NAME                               ;
398B  0F17 R 3995 R         2                   DW       _CODE,_LINK                ;
398F  01 27                 2                   DB      1,"'"                    ;
0F17                        2       ORG         _CODE                               ;
0F17  80                    1                   DB 80H                              ;
0F18  0BC3 R 0C96 R                             DW      TOKEN,NAMEQ             ;?defined
0F1C  03B7 R 0F22 R                             DW      QBRAN,TICK1
0F20  0394 R                                    DW      EXIT                    ;yes, push code address
0F22  0DB4 R                TICK1:              DW      THROW                   ;no, error

                            ;   ALLOT           ( n -- )
                            ;                   Allocate n bytes to the code dictionary.

                                                $COLON  5,'ALLOT',ALLOT
0F24                        2       ALLOT:                                          ;
3981                        2       ORG         _NAME                               ;
3981  0F24 R 398F R         2                   DW       _CODE,_LINK                ;
3985  05 41 4C 4C 4F 54     2                   DB      5,'ALLOT'                    ;
0F24                        2       ORG         _CODE                               ;
0F24  80                    1                   DB 80H                              ;
0F25  0511 R 07AB R 0394 R                      DW      CP,PSTOR,EXIT           ;adjust code pointer

                            ;   ,               ( w -- )
                            ;                   Compile an integer into the code dictionary.

                                                $COLON  1,',',COMMA
0F2B                        2       COMMA:                                          ;
397B                        2       ORG         _NAME                               ;
397B  0F2B R 3985 R         2                   DW       _CODE,_LINK                ;
397F  01 2C                 2                   DB      1,','                    ;
0F2B                        2       ORG         _CODE                               ;
0F2B  80                    1                   DB 80H                              ;
0F2C  07DF R 043B R 062D R                      DW      HERE,DUPP,TWOP         ;cell boundary
0F32  0511 R 03D3 R 03D3 R                      DW      CP,STORE,STORE,EXIT     ;adjust code pointer, compile
      0394 R

                            ;   C,              ( b -- )
                            ;                   Compile a byte into the code dictionary

                                                $COLON  2,'C,',CCOMMA
0F3A                        2       CCOMMA:                                          ;
3973                        2       ORG         _NAME                               ;
3973  0F3A R 397F R         2                   DW       _CODE,_LINK                ;
3977  02 43 2C              2                   DB      2,'C,'                    ;
0F3A                        2       ORG         _CODE                               ;
```

```
0F3A  80                     1               DB 80H                                  ;
0F3B  07DF R 043B R 061F R                   DW      HERE,DUPP,ONEP
0F41  0511 R 03D3 R 03E9 R                   DW      CP,STORE,CSTOR,EXIT
      0394 R

                             ;   [COMPILE]   ( -- ; <string> )
                             ;               Compile the next immediate word into code dictionary.

                                             $COLON  IMEDD+9,'[COMPILE]',BCOMP
0F49                         2       BCOMP:                                           ;
3965                         2       ORG     _NAME                                    ;
3965  0F49 R 3977 R          2               DW       _CODE,_LINK                     ;
3969  89 5B 43 4F 4D 50 49   2               DB      IMEDD+9,'[COMPILE]'                        ;
0F49                         2       ORG     _CODE                                    ;
0F49  80                     1               DB 80H                                   ;
0F4A  0F17 R 0F2B R 0394 R                   DW      TICK,COMMA,EXIT

                             ;   COMPILE     ( -- )
                             ;               Compile the next address in colon list to code dictionary.

                                             $COLON  COMPO+7,'COMPILE',COMPI
0F50                         2       COMPI:                                           ;
3959                         2       ORG     _NAME                                    ;
3959  0F50 R 3969 R          2               DW       _CODE,_LINK                     ;
395D  47 43 4F 4D 50 49 4C   2               DB      COMPO+7,'COMPILE'                        ;
0F50                         2       ORG     _CODE                                    ;
0F50  80                     1               DB 80H                                   ;
0F51  0405 R 043B R 03DE R                   DW      RFROM,DUPP,AT,COMMA     ;compile address
      0F2B R
0F59  062D R 0413 R 0394 R                   DW      TWOP,TOR,EXIT          ;adjust return address

                             ;   LITERAL     ( w -- )
                             ;               Compile tos to code dictionary as an integer literal.

                                             $COLON  IMEDD+7,'LITERAL',LITER
0F5F                         2       LITER:                                           ;
394D                         2       ORG     _NAME                                    ;
394D  0F5F R 395D R          2               DW       _CODE,_LINK                     ;
3951  87 4C 49 54 45 52 41   2               DB      IMEDD+7,'LITERAL'                        ;
0F5F                         2       ORG     _CODE                                    ;
0F5F  80                     1               DB 80H                                   ;
0F60  0F50 R 038D R 0F2B R                   DW      COMPI,DOLIT,COMMA,EXIT
      0394 R

                             ;   $,"         ( -- )
                             ;               Compile a literal string up to next " .

                                             $COLON  3,'$,"',STRCQ
0F68                         2       STRCQ:                                           ;
3945                         2       ORG     _NAME                                    ;
3945  0F68 R 3951 R          2               DW       _CODE,_LINK                     ;
3949  03 24 2C 22            2               DB      3,'$,"'                         ;
0F68                         2       ORG     _CODE                                    ;
0F68  80                     1               DB 80H                                   ;
0F69  038D R 0022 0BDC R                     DW      DOLIT,'"',WORDD               ;move string to code
dictionary
0F6F  07D4 R 0565 R                          DW      COUNT,PLUS        ;calculate aligned end of string
```

```
0F73  0511 R 03D3 R 0394 R                 DW      CP,STORE,EXIT           ;adjust the code pointer

                            ;   RECURSE     ( -- )
                            ;               Make the current word available for compilation.

                                           $COLON  IMEDD+7,'RECURSE',RECUR
0F79                        2      RECUR:                                        ;
3939                        2      ORG     _NAME                                 ;
3939  0F79 R 3949 R         2              DW       _CODE,_LINK                 ;
393D  87 52 45 43 55 52 53  2              DB      IMEDD+7,'RECURSE'                     ;
0F79                        2      ORG     _CODE                                 ;
0F79  80                    1              DB 80H                               ;
0F7A  051B R 03DE R 0BE5 R                 DW      LAST,AT,NAMET,COMMA,EXIT
      0F2B R 0394 R

                            ;; Structures

                            ;   FOR         ( -- a )
                            ;               Start a FOR-NEXT loop structure in a colon definition.

                                           $COLON  IMEDD+3,'FOR',FOR
0F84                        2      FOR:                                        ;
3931                        2      ORG     _NAME                                 ;
3931  0F84 R 393D R         2              DW       _CODE,_LINK                 ;
3935  83 46 4F 52           2              DB      IMEDD+3,'FOR'                     ;
0F84                        2      ORG     _CODE                                 ;
0F84  80                    1              DB 80H                               ;
0F85  0F50 R 0413 R 07DF R                 DW      COMPI,TOR,HERE,EXIT
      0394 R

                            ;   BEGIN       ( -- a )
                            ;               Start an infinite or indefinite loop structure.

                                           $COLON  IMEDD+5,'BEGIN',BEGIN
0F8D                        2      BEGIN:                                        ;
3927                        2      ORG     _NAME                                 ;
3927  0F8D R 3935 R         2              DW       _CODE,_LINK                 ;
392B  85 42 45 47 49 4E     2              DB      IMEDD+5,'BEGIN'                     ;
0F8D                        2      ORG     _CODE                                 ;
0F8D  80                    1              DB 80H                               ;
0F8E  07DF R 0394 R                        DW      HERE,EXIT

                            ;   NEXT        ( a -- )
                            ;               Terminate a FOR-NEXT loop structure.

                                           $COLON  IMEDD+4,'NEXT',NEXT
0F92                        2      NEXT:                                        ;
391D                        2      ORG     _NAME                                 ;
391D  0F92 R 392B R         2              DW       _CODE,_LINK                 ;
3921  84 4E 45 58 54        2              DB      IMEDD+4,'NEXT'                     ;
0F92                        2      ORG     _CODE                                 ;
0F92  80                    1              DB 80H                               ;
0F93  0F50 R 039D R 0F2B R                 DW      COMPI,DONXT,COMMA,EXIT
      0394 R

                            ;   UNTIL       ( a -- )
                            ;               Terminate a BEGIN-UNTIL indefinite loop structure.

                                            $COLON  IMEDD+5,'UNTIL',UNTIL
0F9B                        2       UNTIL:                                          ;
3913                        2       ORG     _NAME                                   ;
3913  0F9B R 3921 R         2               DW       _CODE,_LINK                   ;
3917  85 55 4E 54 49 4C     2               DB      IMEDD+5,'UNTIL'                       ;
0F9B                        2       ORG     _CODE                                   ;
0F9B  80                    1               DB 80H                                 ;
0F9C  0F50 R 03B7 R 0F2B R                  DW      COMPI,QBRAN,COMMA,EXIT
      0394 R

                            ;   AGAIN       ( a -- )
                            ;               Terminate a BEGIN-AGAIN infinite loop structure.

                                            $COLON  IMEDD+5,'AGAIN',AGAIN
0FA4                        2       AGAIN:                                          ;
3909                        2       ORG     _NAME                                   ;
3909  0FA4 R 3917 R         2               DW       _CODE,_LINK                   ;
390D  85 41 47 41 49 4E     2               DB      IMEDD+5,'AGAIN'                       ;
0FA4                        2       ORG     _CODE                                   ;
0FA4  80                    1               DB 80H                                 ;
0FA5  0F50 R 03CC R 0F2B R                  DW      COMPI,BRAN,COMMA,EXIT
      0394 R

                            ;   IF          ( -- A )
                            ;               Begin a conditional branch structure.

                                            $COLON  IMEDD+2,'IF',IFF
0FAD                        2       IFF:                                          ;
3901                        2       ORG     _NAME                                   ;
3901  0FAD R 390D R         2               DW       _CODE,_LINK                   ;
3905  82 49 46              2               DB      IMEDD+2,'IF'                       ;
0FAD                        2       ORG     _CODE                                   ;
0FAD  80                    1               DB 80H                                 ;
0FAE  0F50 R 03B7 R 07DF R                  DW      COMPI,QBRAN,HERE
0FB4  038D R 0000 0F2B R                    DW      DOLIT,0,COMMA,EXIT
      0394 R

                            ;   AHEAD       ( -- A )
                            ;               Compile a forward branch instruction.

                                            $COLON  IMEDD+5,'AHEAD',AHEAD
0FBC                        2       AHEAD:                                          ;
38F7                        2       ORG     _NAME                                   ;
38F7  0FBC R 3905 R         2               DW       _CODE,_LINK                   ;
38FB  85 41 48 45 41 44     2               DB      IMEDD+5,'AHEAD'                       ;
0FBC                        2       ORG     _CODE                                   ;
0FBC  80                    1               DB 80H                                 ;
0FBD  0F50 R 03CC R 07DF R                  DW      COMPI,BRAN,HERE,DOLIT,0,COMMA,EXIT
      038D R 0000 0F2B R
      0394 R

                            ;   REPEAT      ( A a -- )
                            ;               Terminate a BEGIN-WHILE-REPEAT indefinite loop.

```

```
                                          $COLON  IMEDD+6,'REPEAT',REPEA
0FCB                      2       REPEA:                                        ;
38EB                      2       ORG     _NAME                                 ;
38EB  0FCB R 38FB R       2               DW       _CODE,_LINK                 ;
38EF  86 52 45 50 45 41 54  2             DB      IMEDD+6,'REPEAT'                      ;
0FCB                      2       ORG     _CODE                                 ;
0FCB  80                  1               DB 80H                               ;
0FCC  0FA4 R 07DF R 0442 R                DW      AGAIN,HERE,SWAP,STORE,EXIT
      03D3 R 0394 R

                         ;   THEN        ( A -- )
                         ;                Terminate a conditional branch structure.

                                          $COLON  IMEDD+4,'THEN',THENN
0FD6                      2       THENN:                                        ;
38E1                      2       ORG     _NAME                                 ;
38E1  0FD6 R 38EF R       2               DW       _CODE,_LINK                 ;
38E5  84 54 48 45 4E      2               DB      IMEDD+4,'THEN'                    ;
0FD6                      2       ORG     _CODE                                 ;
0FD6  80                  1               DB 80H                               ;
0FD7  07DF R 0442 R 03D3 R                DW      HERE,SWAP,STORE,EXIT
      0394 R

                         ;   AFT         ( a -- a A )
                         ;                Jump to THEN in a FOR-AFT-THEN-NEXT loop the first time through.

                                          $COLON  IMEDD+3,'AFT',AFT
0FDF                      2       AFT:                                        ;
38D9                      2       ORG     _NAME                                 ;
38D9  0FDF R 38E5 R       2               DW       _CODE,_LINK                 ;
38DD  83 41 46 54         2               DB      IMEDD+3,'AFT'                    ;
0FDF                      2       ORG     _CODE                                 ;
0FDF  80                  1               DB 80H                               ;
0FE0  0436 R 0FBC R 0F8D R                DW      DROP,AHEAD,BEGIN,SWAP,EXIT
      0442 R 0394 R

                         ;   ELSE        ( A -- A )
                         ;                Start the false clause in an IF-ELSE-THEN structure.

                                          $COLON  IMEDD+4,'ELSE',ELSEE
0FEA                      2       ELSEE:                                        ;
38CF                      2       ORG     _NAME                                 ;
38CF  0FEA R 38DD R       2               DW       _CODE,_LINK                 ;
38D3  84 45 4C 53 45      2               DB      IMEDD+4,'ELSE'                    ;
0FEA                      2       ORG     _CODE                                 ;
0FEA  80                  1               DB 80H                               ;
0FEB  0FBC R 0442 R 0FD6 R                DW      AHEAD,SWAP,THENN,EXIT
      0394 R

                         ;   WHILE       ( a -- A a )
                         ;                Conditional branch out of a BEGIN-WHILE-REPEAT loop.

                                          $COLON  IMEDD+5,'WHILE',WHILE
0FF3                      2       WHILE:                                        ;
38C5                      2       ORG     _NAME                                 ;
38C5  0FF3 R 38D3 R       2               DW       _CODE,_LINK                 ;
```

```
38C9  85 57 48 49 4C 45     2              DB      IMEDD+5,'WHILE'                         ;
0FF3                        2       ORG     _CODE                                   ;
0FF3  80                    1              DB 80H                                  ;
0FF4  0FAD R 0442 R 0394 R                 DW      IFF,SWAP,EXIT

                            ;   ABORT"     ( -- ; <string> )
                            ;               Conditional abort with an error message.

                                           $COLON  IMEDD+6,'ABORT"',ABRTQ
0FFA                        2       ABRTQ:                                          ;
38B9                        2       ORG     _NAME                                   ;
38B9  0FFA R 38C9 R         2              DW       _CODE,_LINK                    ;
38BD  86 41 42 4F 52 54 22  2              DB      IMEDD+6,'ABORT"'                        ;
0FFA                        2       ORG     _CODE                                   ;
0FFA  80                    1              DB 80H                                  ;
0FFB  0F50 R 0DDF R 0F68 R                 DW      COMPI,ABORQ,STRCQ,EXIT
      0394 R

                            ;   $"         ( -- ; <string> )
                            ;               Compile an inline string literal.

                                           $COLON  IMEDD+2,'$"',STRQ
1003                        2       STRQ:                                          ;
38B1                        2       ORG     _NAME                                   ;
38B1  1003 R 38BD R         2              DW       _CODE,_LINK                    ;
38B5  82 24 22              2              DB      IMEDD+2,'$"'                        ;
1003                        2       ORG     _CODE                                   ;
1003  80                    1              DB 80H                                  ;
1004  0F50 R 0A84 R 0F68 R                 DW      COMPI,STRQP,STRCQ,EXIT
      0394 R

                            ;   ."         ( -- ; <string> )
                            ;               Compile an inline string literal to be typed out at run time.

                                           $COLON  IMEDD+2,'."',DOTQ
100C                        2       DOTQ:                                          ;
38A9                        2       ORG     _NAME                                   ;
38A9  100C R 38B5 R         2              DW       _CODE,_LINK                    ;
38AD  82 2E 22              2              DB      IMEDD+2,'."'                        ;
100C                        2       ORG     _CODE                                   ;
100C  80                    1              DB 80H                                  ;
100D  0F50 R 0A89 R 0F68 R                 DW      COMPI,DOTQP,STRCQ,EXIT
      0394 R

                            ;; Name compiler

                            ;   ?UNIQUE     ( a -- a )
                            ;               Display a warning message if the word already exists.

                                           $COLON  7,'?UNIQUE',UNIQU
1015                        2       UNIQU:                                          ;
389D                        2       ORG     _NAME                                   ;
389D  1015 R 38AD R         2              DW       _CODE,_LINK                    ;
38A1  07 3F 55 4E 49 51 55  2              DB      7,'?UNIQUE'                        ;
1015                        2       ORG     _CODE                                   ;
1015  80                    1              DB 80H                                  ;
```

```
1016  043B R 0C96 R                        DW      DUPP,NAMEQ              ;?name exists
101A  03B7 R 102E R                        DW      QBRAN,UNIQ1             ;redefinitions are OK
                                           D$      DOTQP,' reDef '         ;but warn the user
101E  0A89 R                1              DW      DOTQP                           ;
1020  00 20 72 65 44 65 66  1              DB      0,' reDef '                         ;
1020                        1       ORG     _LEN                                ;
1020  07                    1              DB      _CODE-_LEN-1                    ;
1028                        1       ORG     _CODE                               ;
1028  044A R 07D4 R 0A4B R                 DW      OVER,COUNT,TYPEE        ;just in case its not planned
102E  0436 R 0394 R         UNIQ1:          DW      DROP,EXIT

                            ;   $,n         ( na -- )
                            ;               Build a new dictionary name using the string at na.

                                           $COLON  3,'$,n',SNAME
1032                        2       SNAME:                                      ;
3895                        2       ORG     _NAME                               ;
3895  1032 R 38A1 R         2              DW       _CODE,_LINK                    ;
3899  03 24 2C 6E           2              DB      3,'$,n'                        ;
1032                        2       ORG     _CODE                               ;
1032  80                    1              DB 80H                              ;
1033  043B R 03F1 R                        DW      DUPP,CAT                ;?null input
1037  03B7 R 105F R                        DW      QBRAN,PNAM1
103B  1015 R                               DW      UNIQU                   ;?redefinition
103D  043B R 051B R 03D3 R                 DW      DUPP,LAST,STORE         ;save na for vocabulary link
1043  07DF R 0442 R                        DW      HERE,SWAP               ;align code address
1047  0635 R                               DW      TWOM                   ;link address
1049  0502 R 03DE R 03DE R                 DW      CRRNT,AT,AT,OVER,STORE
      044A R 03D3 R
1053  0635 R 043B R 0516 R                 DW      TWOM,DUPP,NP,STORE     ;adjust name pointer
      03D3 R
105B  03D3 R 0394 R                        DW      STORE,EXIT              ;save code pointer
105F                        PNAM1:          D$      STRQP,' name'           ;null input
105F  0A84 R                1              DW      STRQP                           ;
1061  00 20 6E 61 6D 65     1              DB      0,' name'                         ;
1061                        1       ORG     _LEN                                ;
1061  05                    1              DB      _CODE-_LEN-1                    ;
1067                        1       ORG     _CODE                               ;
1067  0DB4 R                               DW      THROW

                            ;; FORTH compiler

                            ;   $COMPILE    ( a -- )
                            ;               Compile next word to code dictionary as a token or literal.

                                           $COLON  8,'$COMPILE',SCOMP
1069                        2       SCOMP:                                      ;
3887                        2       ORG     _NAME                               ;
3887  1069 R 3899 R         2              DW       _CODE,_LINK                    ;
388B  08 24 43 4F 4D 50 49  2              DB      8,'$COMPILE'                        ;
1069                        2       ORG     _CODE                               ;
1069  80                    1              DB 80H                              ;
106A  0C96 R 053D R                        DW      NAMEQ,QDUP              ;?defined
106E  03B7 R 1086 R                        DW      QBRAN,SCOM2
1072  03DE R 038D R 0080                   DW      AT,DOLIT,IMEDD,ANDD     ;?immediate
      0461 R
```

```
107A  03B7 R 1082 R                          DW      QBRAN,SCOM1
107E  039B R 0394 R                          DW      EXECU,EXIT              ;its immediate, execute
1082  0F2B R 0394 R          SCOM1:           DW      COMMA,EXIT              ;its not immediate, compile
1086  04EE R 07F6 R          SCOM2:           DW      TNUMB,ATEXE             ;try to convert to number
108A  03B7 R 1092 R                          DW      QBRAN,SCOM3
108E  0F5F R 0394 R                          DW      LITER,EXIT              ;compile number as integer
1092  0DB4 R                 SCOM3:           DW      THROW                   ;error

                             ;   CCOMPILE     ( a -- )
                             ;               Compile next byte to code dictionary as machine code.

                                             $COLON  8,'CCOMPILE',CCOMP
1094                         2       CCOMP:                                          ;
3879                         2       ORG     _NAME                                   ;
3879  1094 R 388B R          2               DW       _CODE,_LINK                    ;
387D  08 43 43 4F 4D 50 49  2               DB      8,'CCOMPILE'                        ;
1094                         2       ORG     _CODE                                   ;
1094  80                     1               DB 80H                                  ;
1095  0C96 R 053D R                          DW      NAMEQ,QDUP              ;?defined
1099  03B7 R 10B1 R                          DW      QBRAN,CCOM2
109D  03DE R 038D R 0080                     DW      AT,DOLIT,IMEDD,ANDD     ;?immediate
      0461 R
10A5  03B7 R 10AD R                          DW      QBRAN,CCOM1
10A9  039B R 0394 R                          DW      EXECU,EXIT              ;its immediate, execute
10AD  0436 R 0394 R          CCOM1:           DW      DROP,EXIT               ;its not immediate,drop
10B1  04EE R 07F6 R          CCOM2:           DW      TNUMB,ATEXE             ;try to convert to number
10B5  03B7 R 10BD R                          DW      QBRAN,CCOM3
10B9  0F3A R 0394 R                          DW      CCOMMA,EXIT             ;compile as code byte
10BD  0DB4 R                 CCOM3:           DW      THROW                   ;error

                             ;   OVERT        ( -- )
                             ;               Link a new word into the current vocabulary.

                                             $COLON  5,'OVERT',OVERT
10BF                         2       OVERT:                                          ;
386F                         2       ORG     _NAME                                   ;
386F  10BF R 387D R          2               DW       _CODE,_LINK                    ;
3873  05 4F 56 45 52 54      2               DB      5,'OVERT'                        ;
10BF                         2       ORG     _CODE                                   ;
10BF  80                     1               DB 80H                                  ;
10C0  051B R 03DE R 0502 R                   DW      LAST,AT,CRRNT,AT,STORE,EXIT
      03DE R 03D3 R 0394 R

                             ;   ;            ( -- )
                             ;               Terminate a colon definition.

                                             $COLON  IMEDD+COMPO+1,';',SEMIS
10CC                         2       SEMIS:                                          ;
3869                         2       ORG     _NAME                                   ;
3869  10CC R 3873 R          2               DW       _CODE,_LINK                    ;
386D  C1 3B                  2               DB      IMEDD+COMPO+1,';'                        ;
10CC                         2       ORG     _CODE                                   ;
10CC  80                     1               DB 80H                                  ;
10CD  0F50 R 0394 R 0E1F R                   DW      COMPI,EXIT,LBRAC,OVERT,EXIT
      10BF R 0394 R

```

```
                                ;   ]           ( -- )
                                ;               Start compiling the words in the input stream.

                                                $COLON  1,']',RBRAC
10D7                            2       RBRAC:                                          ;
3863                            2       ORG     _NAME                                   ;
3863  10D7 R 386D R             2               DW       _CODE,_LINK                   ;
3867  01 5D                     2               DB      1,']'                     ;
10D7                            2       ORG     _CODE                                   ;
10D7  80                        1               DB 80H                                 ;
10D8  038D R 1069 R 04E9 R                      DW      DOLIT,SCOMP,TEVAL,STORE,EXIT
      03D3 R 0394 R

                                ;   call,       ( ca -- )
                                ;               Assemble a call instruction to doLST.

                                                $COLON  5,'call,',CALLC
10E2                            2       CALLC:                                          ;
3859                            2       ORG     _NAME                                   ;
3859  10E2 R 3867 R             2               DW       _CODE,_LINK                   ;
385D  05 63 61 6C 6C 2C         2               DB      5,'call,'                        ;
10E2                            2       ORG     _CODE                                   ;
10E2  80                        1               DB 80H                                 ;
10E3  038D R 0080 0F3A R                        DW      DOLIT,CALLL,CCOMMA,EXIT  ;Direct Threaded Code
      0394 R

                                ;   :           ( -- ; <string> )
                                ;               Start a new colon definition using next word as its name.

                                                $COLON  1,':',COLON
10EB                            2       COLON:                                          ;
3853                            2       ORG     _NAME                                   ;
3853  10EB R 385D R             2               DW       _CODE,_LINK                   ;
3857  01 3A                     2               DB      1,':'                     ;
10EB                            2       ORG     _CODE                                   ;
10EB  80                        1               DB 80H                                 ;
10EC  0BC3 R 1032 R                             DW      TOKEN,SNAME
10F0  10E2 R 10D7 R 0394 R                      DW      CALLC,RBRAC,EXIT

                                ;   IMMEDIATE   ( -- )
                                ;               Make the last compiled word an immediate word.

                                                $COLON  9,'IMMEDIATE',IMMED
10F6                            2       IMMED:                                          ;
3845                            2       ORG     _NAME                                   ;
3845  10F6 R 3857 R             2               DW       _CODE,_LINK                   ;
3849  09 49 4D 4D 45 44 49      2               DB      9,'IMMEDIATE'                           ;
10F6                            2       ORG     _CODE                                   ;
10F6  80                        1               DB 80H                                 ;
10F7  038D R 0080 051B R                        DW      DOLIT,IMEDD,LAST,AT,AT,ORR
      03DE R 03DE R 046A R
1103  051B R 03DE R 03D3 R                      DW      LAST,AT,STORE,EXIT
      0394 R

                                ;; Defining words

```

```
                            ;   USER        ( u -- ; <string> )
                            ;               Compile a new user variable.

                                           $COLON  4,'USER',USER
110B                        2       USER:                                        ;
383B                        2       ORG     _NAME                                 ;
383B  110B R 3849 R         2              DW       _CODE,_LINK                   ;
383F  04 55 53 45 52        2              DB      4,'USER'                       ;
110B                        2       ORG     _CODE                                 ;
110B  80                    1              DB 80H                                ;
110C  0BC3 R 1032 R 10BF R                 DW      TOKEN,SNAME,OVERT,CALLC
      10E2 R
1114  0F50 R 0496 R 0F2B R                 DW      COMPI,DOUSE,COMMA,EXIT
      0394 R

                            ;   CREATE      ( -- ; <string> )
                            ;               Compile a new array entry without allocating code space.

                                           $COLON  6,'CREATE',CREAT
111C                        2       CREAT:                                        ;
382F                        2       ORG     _NAME                                 ;
382F  111C R 383F R         2              DW       _CODE,_LINK                   ;
3833  06 43 52 45 41 54 45  2              DB      6,'CREATE'                       ;
111C                        2       ORG     _CODE                                 ;
111C  80                    1              DB 80H                                ;
111D  0BC3 R 1032 R 10BF R                 DW      TOKEN,SNAME,OVERT,CALLC
      10E2 R
1125  0F50 R 048C R 0394 R                 DW      COMPI,DOVAR,EXIT

                            ;   VARIABLE    ( -- ; <string> )
                            ;               Compile a new variable initialized to 0.

                                           $COLON  8,'VARIABLE',VARIA
112B                        2       VARIA:                                        ;
3821                        2       ORG     _NAME                                 ;
3821  112B R 3833 R         2              DW       _CODE,_LINK                   ;
3825  08 56 41 52 49 41 42  2              DB      8,'VARIABLE'                       ;
112B                        2       ORG     _CODE                                 ;
112B  80                    1              DB 80H                                ;
112C  111C R 038D R 0000                   DW      CREAT,DOLIT,0,COMMA,EXIT
      0F2B R 0394 R

                            ;   CODE        ( -- )
                            ;               Start a new code definition using next word as its name.

                                           $COLON  4,'CODE',CODE
1136                        2       CODE:                                        ;
3817                        2       ORG     _NAME                                 ;
3817  1136 R 3825 R         2              DW       _CODE,_LINK                   ;
381B  04 43 4F 44 45        2              DB      4,'CODE'                       ;
1136                        2       ORG     _CODE                                 ;
1136  80                    1              DB 80H                                ;
1137  0BC3 R 1032 R                        DW      TOKEN,SNAME
113B  038D R 1094 R 04E9 R                 DW      DOLIT,CCOMP,TEVAL,STORE,EXIT
      03D3 R 0394 R
```

```
                             ;   ENDCODE     ( -- )
                             ;               Terminate a code definition

                                            $COLON  IMEDD+COMPO+7,'ENDCODE',ENDCD
1145                        2       ENDCD:                                          ;
380B                        2       ORG     _NAME                                   ;
380B  1145 R 381B R         2               DW       _CODE,_LINK                   ;
380F  C7 45 4E 44 43 4F 44  2               DB      IMEDD+COMPO+7,'ENDCODE'                       ;
1145                        2       ORG     _CODE                                   ;
1145  80                    1               DB 80H                                 ;
1146  038D R 0048 0F3A R                    DW      DOLIT,48H,CCOMMA,DOLIT,84H,CCOMMA       ;$NEXT
      038D R 0084 0F3A R
1152  038D R 0048 0F3A R                    DW      DOLIT,48H,CCOMMA,DOLIT,28H,CCOMMA
      038D R 0028 0F3A R
115E  0E1F R 10BF R 0394 R                  DW      LBRAC,OVERT,EXIT

                             ;; Tools

                             ;   _TYPE       ( b u -- )
                             ;               Display a string. Filter non-printing characters.

                                            $COLON  5,'_TYPE',UTYPE
1164                        2       UTYPE:                                          ;
3801                        2       ORG     _NAME                                   ;
3801  1164 R 380F R         2               DW       _CODE,_LINK                   ;
3805  05 5F 54 59 50 45     2               DB      5,'_TYPE'                        ;
1164                        2       ORG     _CODE                                   ;
1164  80                    1               DB 80H                                 ;
1165  0413 R                                DW      TOR                     ;start count down loop
1167  03CC R 1175 R                         DW      BRAN,UTYP2              ;skip first pass
116B  043B R 03F1 R 076E R  UTYP1:           DW      DUPP,CAT,TCHAR,EMIT     ;display only printable
      0A0A R
1173  061F R                                DW      ONEP                     ;increment address
1175  039D R 116B R         UTYP2:           DW      DONXT,UTYP1             ;loop till done
1179  0436 R 0394 R                         DW      DROP,EXIT

                             ;   dm+         ( a u -- a )
                             ;               Dump u bytes from , leaving a+u on the stack.

                                            $COLON  3,'dm+',DMP
117D                        2       DMP:                                          ;
37F9                        2       ORG     _NAME                                   ;
37F9  117D R 3805 R         2               DW       _CODE,_LINK                   ;
37FD  03 64 6D 2B           2               DB      3,'dm+'                      ;
117D                        2       ORG     _CODE                                   ;
117D  80                    1               DB 80H                                 ;
117E  044A R 038D R 0004                    DW      OVER,DOLIT,4,UDOTR      ;display address
      0AA3 R
1186  0A2F R 0413 R                         DW      SPACE,TOR               ;start count down loop
118A  03CC R 119A R                         DW      BRAN,PDUM2              ;skip first pass
118E  043B R 03F1 R 038D R  PDUM1:           DW      DUPP,CAT,DOLIT,3,UDOTR  ;display numeric data
      0003 0AA3 R
1198  061F R                                DW      ONEP                     ;increment address
119A  039D R 118E R         PDUM2:           DW      DONXT,PDUM1             ;loop till done
119E  0394 R                                DW      EXIT
```

```
                                ;   DUMP        ( a u -- )
                                ;               Dump u bytes from a, in a formatted manner.

                                                $COLON  4,'DUMP',DUMP
11A0                            2       DUMP:                                           ;
37EF                            2       ORG     _NAME                                   ;
37EF  11A0 R 37FD R             2               DW      _CODE,_LINK                     ;
37F3  04 44 55 4D 50            2               DB      4,'DUMP'                        ;
11A0                            2       ORG     _CODE                                   ;
11A0  80                        1               DB 80H                                  ;
11A1  04CB R 03DE R 0413 R                      DW      BASE,AT,TOR,HEX         ;save radix, set hex
      0920 R
11A9  038D R 0010 06F1 R                        DW      DOLIT,16,SLASH          ;change count to lines
11AF  0413 R                                    DW      TOR                     ;start count down loop
11B1  0A62 R 038D R 0010    DUMP1:              DW      CR,DOLIT,16,DDUP,DMP    ;display numeric
      055B R 117D R
11BB  054A R 054A R                             DW      ROT,ROT
11BF  0A2F R 0A2F R 1164 R                      DW      SPACE,SPACE,UTYPE       ;display printable characters
11C5  0A11 R 056F R                             DW      NUFQ,INVER              ;user control
11C9  03B7 R 11D5 R                             DW      QBRAN,DUMP2
11CD  039D R 11B1 R                             DW      DONXT,DUMP1             ;loop till done
11D1  03CC R 11D9 R                             DW      BRAN,DUMP3
11D5  0405 R 0436 R         DUMP2:              DW      RFROM,DROP              ;cleanup loop stack, early exit
11D9  0436 R 0405 R 04CB R  DUMP3:              DW      DROP,RFROM,BASE,STORE   ;restore radix
      03D3 R
11E1  0394 R                                    DW      EXIT

                                ;   .S          ( ... -- ... )
                                ;               Display the contents of the data stack.

                                                $COLON  2,'.S',DOTS
11E3                            2       DOTS:                                           ;
37E7                            2       ORG     _NAME                                   ;
37E7  11E3 R 37F3 R             2               DW      _CODE,_LINK                     ;
37EB  02 2E 53                  2               DB      2,'.S'                        ;
11E3                            2       ORG     _CODE                                   ;
11E3  80                        1               DB 80H                                  ;
11E4  0A62 R 078B R                             DW      CR,DEPTH                ;stack depth
11E8  0413 R                                    DW      TOR                     ;start count down loop
11EA  03CC R 11F4 R                             DW      BRAN,DOTS2              ;skip first pass
11EE  040C R 079E R 0AC5 R  DOTS1:              DW      RAT,PICK,DOT            ;index stack, display contents
11F4  039D R 11EE R         DOTS2:              DW      DONXT,DOTS1             ;loop till done
                                                D$      DOTQP,' <sp'
11F8  0A89 R                    1               DW      DOTQP                           ;
11FA  00 20 3C 73 70            1               DB      0,' <sp'                        ;
11FA                            1       ORG     _LEN                                    ;
11FA  04                        1               DB      _CODE-_LEN-1                    ;
11FF                            1       ORG     _CODE                                   ;
11FF  0394 R                                    DW      EXIT

                                ;   !CSP        ( -- )
                                ;               Save stack pointer in CSP for error checking.

                                                $COLON  4,'!CSP',STCSP
1201                            2       STCSP:                                          ;
37DD                            2       ORG     _NAME                                   ;
```

```
37DD  1201 R 37EB R         2               DW       _CODE,_LINK                     ;
37E1  04 21 43 53 50        2               DB      4,'!CSP'                         ;
1201                        2       ORG     _CODE                                     ;
1201  80                    1               DB 80H                                   ;
1202  041C R 04E4 R 03D3 R                  DW      SPAT,CSP,STORE,EXIT     ;save pointer
      0394 R


                            ;   ?CSP        ( -- )
                            ;               Abort if stack pointer differs from that saved in CSP.

                                            $COLON  4,'?CSP',QCSP
120A                        2       QCSP:                                            ;
37D3                        2       ORG     _NAME                                     ;
37D3  120A R 37E1 R         2               DW       _CODE,_LINK                     ;
37D7  04 3F 43 53 50        2               DB      4,'?CSP'                         ;
120A                        2       ORG     _CODE                                     ;
120A  80                    1               DB 80H                                   ;
120B  041C R 04E4 R 03DE R                  DW      SPAT,CSP,AT,XORR        ;compare pointers
      0473 R
                                            D$      ABORQ,'stacks'          ;abort if different
1213  0DDF R                1               DW      ABORQ                            ;
1215  00 73 74 61 63 6B 73  1               DB      0,'stacks'                          ;
1215                        1       ORG     _LEN                                      ;
1215  06                    1               DB      _CODE-_LEN-1                     ;
121C                        1       ORG     _CODE                                     ;
121C  0394 R                                DW      EXIT


                            ;   >NAME       ( ca -- na | F )
                            ;               Convert code address to a name address.

                                            $COLON  5,'>NAME',TNAME
121E                        2       TNAME:                                            ;
37C9                        2       ORG     _NAME                                     ;
37C9  121E R 37D7 R         2               DW       _CODE,_LINK                     ;
37CD  05 3E 4E 41 4D 45     2               DB      5,'>NAME'                         ;
121E                        2       ORG     _CODE                                     ;
121E  80                    1               DB 80H                                   ;
121F  0502 R                                DW      CRRNT                   ;vocabulary link
1221  062D R 03DE R 053D R  TNAM1:           DW      TWOP,AT,QDUP            ;check all vocabularies
1227  03B7 R 1259 R                         DW      QBRAN,TNAM4
122B  055B R                                DW      DDUP
122D  03DE R 043B R         TNAM2:           DW      AT,DUPP                  ;?last word in a vocabulary
1231  03B7 R 1245 R                         DW      QBRAN,TNAM3
1235  055B R 0BE5 R 0473 R                  DW      DDUP,NAMET,XORR         ;compare
123B  03B7 R 1245 R                         DW      QBRAN,TNAM3
123F  0635 R                                DW      TWOM                   ;continue with next word
1241  03CC R 122D R                         DW      BRAN,TNAM2
1245  0442 R 0436 R 053D R  TNAM3:           DW      SWAP,DROP,QDUP
124B  03B7 R 1221 R                         DW      QBRAN,TNAM1
124F  0442 R 0436 R 0442 R                  DW      SWAP,DROP,SWAP,DROP,EXIT
      0436 R 0394 R
1259  0436 R 038D R 0000    TNAM4:           DW      DROP,DOLIT,0,EXIT       ;false flag
      0394 R

                            ;   .ID         ( na -- )
                            ;               Display the name at address.
```

```

                                           $COLON  3,'.ID',DOTID
1261                        2       DOTID:                                      ;
37C1                        2       ORG     _NAME                               ;
37C1  1261 R 37CD R         2               DW       _CODE,_LINK                ;
37C5  03 2E 49 44           2               DB      3,'.ID'                    ;
1261                        2       ORG     _CODE                               ;
1261  80                    1               DB 80H                              ;
1262  053D R                                DW      QDUP                  ;if zero no name
1264  03B7 R 1274 R                         DW      QBRAN,DOTI1
1268  07D4 R 038D R 001F                    DW      COUNT,DOLIT,01FH,ANDD   ;mask lexicon bits
      0461 R
1270  1164 R 0394 R                         DW      UTYPE,EXIT            ;display name string
1274                        DOTI1:           D$      DOTQP,' {noName}'
1274  0A89 R                1               DW      DOTQP                       ;
1276  00 20 7B 6E 6F 4E 61  1               DB      0,' {noName}'                   ;
1276                        1       ORG     _LEN                                ;
1276  09                    1               DB      _CODE-_LEN-1                ;
1280                        1       ORG     _CODE                               ;
1280  0394 R                                DW      EXIT

                            ;   WORDS       ( -- )
                            ;               Display the names in the context vocabulary.

                                           $COLON  5,'WORDS',WORDS
1282                        2       WORDS:                                      ;
37B7                        2       ORG     _NAME                               ;
37B7  1282 R 37C5 R         2               DW       _CODE,_LINK                ;
37BB  05 57 4F 52 44 53     2               DB      5,'WORDS'                    ;
1282                        2       ORG     _CODE                               ;
1282  80                    1               DB 80H                              ;
1283  0A62 R 04FD R 03DE R                  DW      CR,CNTXT,AT           ;only in context
1289  03DE R 053D R         WORS1:           DW      AT,QDUP               ;?at end of list
128D  03B7 R 12A1 R                         DW      QBRAN,WORS2
1291  043B R 0A2F R 1261 R                  DW      DUPP,SPACE,DOTID      ;display a name
1297  0635 R 0A11 R                         DW      TWOM,NUFQ            ;user control
129B  03B7 R 1289 R                         DW      QBRAN,WORS1
129F  0436 R                                DW      DROP
12A1  0394 R                WORS2:           DW      EXIT

                            ;; Hardware reset

                            ;   VER         ( -- n )
                            ;               Return the version number of this implementation.

                                           $COLON  3,'VER',VERSN
12A3                        2       VERSN:                                      ;
37AF                        2       ORG     _NAME                               ;
37AF  12A3 R 37BB R         2               DW       _CODE,_LINK                ;
37B3  03 56 45 52           2               DB      3,'VER'                    ;
12A3                        2       ORG     _CODE                               ;
12A3  80                    1               DB 80H                              ;
12A4  038D R 0101 0394 R                    DW      DOLIT,VER*256+EXT,EXIT

                            ;   hi          ( -- )
                            ;               Display the sign-on message of eForth.
```

```


                                            $COLON  2,'hi',HI
12AA                        2       HI:                                         ;
37A7                        2       ORG     _NAME                                  ;
37A7  12AA R 37B3 R         2               DW       _CODE,_LINK                   ;
37AB  02 68 69              2               DB      2,'hi'                       ;
12AA                        2       ORG     _CODE                                  ;
12AA  80                    1               DB 80H                                 ;
12AB  0380 R 0A62 R                         DW      STOIO,CR                 ;initialize I/O
                                            D$      DOTQP,'eForth v'
12AF  0A89 R                1               DW      DOTQP                          ;
12B1  00 65 46 6F 72 74 68  1               DB      0,'eForth v'                        ;
12B1                        1       ORG     _LEN                                   ;
12B1  08                    1               DB      _CODE-_LEN-1                   ;
12BA                        1       ORG     _CODE                                  ;
12BA  04CB R 03DE R 0920 R                  DW      BASE,AT,HEX
12C0  12A3 R 08BB R 08D5 R                  DW      VERSN,BDIGS,DIG,DIG
      08D5 R
12C8  038D R 002E 08C4 R                    DW      DOLIT,'.',HOLD
12CE  08E0 R 08FE R 0A4B R                  DW      DIGS,EDIGS,TYPEE
12D4  04CB R 03D3 R 0A62 R                  DW      BASE,STORE,CR,EXIT
      0394 R

                            ;   'BOOT        ( -- a )
                            ;                The application startup vector.

                                            $COLON  5,"'BOOT",TBOOT
12DC                        2       TBOOT:                                         ;
379D                        2       ORG     _NAME                                  ;
379D  12DC R 37AB R         2               DW       _CODE,_LINK                   ;
37A1  05 27 42 4F 4F 54     2               DB      5,"'BOOT"                       ;
12DC                        2       ORG     _CODE                                  ;
12DC  80                    1               DB 80H                                 ;
12DD  048C R                                DW      DOVAR
12DF  12AA R                                DW      HI                       ;application to boot

                            ;   SEE          ( --word-- )
                            ;                Decompiles word.
                                            $COLON  3,'SEE',SEE
12E1                        2       SEE:                                         ;
3795                        2       ORG     _NAME                                  ;
3795  12E1 R 37A1 R         2               DW       _CODE,_LINK                   ;
3799  03 53 45 45           2               DB      3,'SEE'                       ;
12E1                        2       ORG     _CODE                                  ;
12E1  80                    1               DB 80H                                 ;
12E2  0F17 R                                DW      TICK
12E4  0A62 R 061F R                         DW      CR,ONEP
12E8  043B R 043B R 0A2F R      SEE1:        DW      DUPP,DUPP,SPACE,DOT,AT,DUPP
      0AC5 R 03DE R 043B R
12F4  03B7 R 12FA R                         DW      QBRAN,SEE2
12F8  121E R                                DW      TNAME
12FA  053D R                    SEE2:        DW      QDUP
12FC  03B7 R 1306 R                         DW      QBRAN,SEE3
1300  1261 R                                DW      DOTID
1302  03CC R 130C R                         DW      BRAN,SEE4
1306  043B R 03DE R 0AB8 R      SEE3:        DW      DUPP,AT,UDOT
```

```
130C  062D R 0A11 R             SEE4:        DW      TWOP,NUFQ
1310  03B7 R 12E8 R                         DW      QBRAN,SEE1
1314  0436 R 0394 R                         DW      DROP,EXIT


                            ;   ADCINIT      ( -- )
                            ;                Init routine for starting ADC Interrupts
                                            $CODE 7,'ADCINIT',ADCINIT
1318                        1      ADCINIT:                                           ;
3789                        1      ORG     _NAME                                    ;
3789  1318 R 3799 R         1              DW       _CODE,_LINK                     ;
378D  07 41 44 43 49 4E 49  1              DB      7,'ADCINIT'                         ;
1318                        1      ORG     _CODE                                    ;
1318  68 FF                                 DB 68H,0FFH             ;V<FF
131A  69 C6                                 DB 69H,0C6H             ;A<C6
131C  63 F2                                 DB 63H,0F2H             ;(V/F2)<A
131E  69 00                                 DB 69H,0                ;A<0
1320  63 F3                                 DB 63H,0F3H             ;(V/F3)<A
1322  4D C8                                 DB 4DH,0C8H             ;ANM <A
1324  48 48                                 DB 48H,48H              ;SKIT FAD, reset INTFAD
1326  00                                    DB 00                   ;NOP
1327  64 0E FE                                     DB 64H,0EH,0FEH         ;ENABLE INTAD
                                            $NEXT
132A  48 84                 1               DB 48H,84H                              ;
132C  48 28                 1               DB 48H,28H                              ;

                            ;   TMIDI        ( n -- )
                            ;                Wait for last transmit, then send midi byte n.
                                            $CODE 5,'TMIDI',TMIDI
132E                        1      TMIDI:                                           ;
377F                        1      ORG     _NAME                                    ;
377F  132E R 378D R         1              DW       _CODE,_LINK                     ;
3783  05 54 4D 49 44 49     1              DB      5,'TMIDI'                         ;
132E                        1      ORG     _CODE                                    ;
132E  A1                                    DB 0A1H                 ;POP BC
132F  0B                                    DB 0BH                  ;A<C
1330  48 4A                                 DB 48H,4AH              ;SKIT FST, skip if interrupt
1332  FD                                    DB 0FDH                 ;JMP TO SKIT
1333  4D D8                                 DB 4DH,0D8H             ;MOV TXB,A
                                            $NEXT
1335  48 84                 1               DB 48H,84H                              ;
1337  48 28                 1               DB 48H,28H                              ;

                            ;   DELAY         ( n -- )
                            ;                Wait for n loops.
                                            $CODE 5,'DELAY',DELAY
1339                        1      DELAY:                                           ;
3775                        1      ORG     _NAME                                    ;
3775  1339 R 3783 R         1              DW       _CODE,_LINK                     ;
3779  05 44 45 4C 41 59     1              DB      5,'DELAY'                         ;
1339                        1      ORG     _CODE                                    ;
1339  A1                                    DB 0A1H                 ;POP BC
133A  53                                    DB 53H                  ;C<C-1, Skip if borrow
133B  FE                                    DB 0FEH                 ;JMP
133C  52                                    DB 52H                  ;B<B-1, Skip if borrow
133D  FC                                    DB 0FCH                 ;JMP
```

```
                                             $NEXT
133E  48 84                 1                DB 48H,84H                             ;
1340  48 28                 1                DB 48H,28H                             ;

                            ;   LCD           ( n -- )
                            ;                 Load control n to LCD display.
                                             $CODE 4,'LCD',LCD
1342                        1       LCD:                                           ;
376B                        1       ORG      _NAME                                  ;
376B  1342 R 3779 R         1                DW       _CODE,_LINK                   ;
376F  04 4C 43 44           1                DB      4,'LCD'                       ;
1342                        1       ORG      _CODE                                  ;
1342  A1                                     DB 0A1H                  ;POP BC
1343  0B                                     DB 0BH                   ;A<C
1344  14 00 A0                                       DB 14H,0,0A0H            ;BC<A000
1347  39                                     DB 39H                   ;(BC)<A
                                             $NEXT
1348  48 84                 1                DB 48H,84H                             ;
134A  48 28                 1                DB 48H,28H                             ;

                            ;  LLI            ( --- )
                            ;                 Sets RS=0 for LCD setup commands.
                                             $CODE 3,'LLI',LLI
134C                        1       LLI:                                           ;
3763                        1       ORG      _NAME                                  ;
3763  134C R 376F R         1                DW       _CODE,_LINK                   ;
3767  03 4C 4C 49           1                DB      3,'LLI'                       ;
134C                        1       ORG      _CODE                                  ;
134C  64 0A EF                                       DB 64H,0AH,0EFH          ;Pc<Pc AND EF
                                             $NEXT
134F  48 84                 1                DB 48H,84H                             ;
1351  48 28                 1                DB 48H,28H                             ;

                            ;  LLC            ( --- )
                            ;                 Sets RS=1 for LCD character loading
                                             $CODE 3,'LLC',LLC
1353                        1       LLC:                                           ;
375B                        1       ORG      _NAME                                  ;
375B  1353 R 3767 R         1                DW       _CODE,_LINK                   ;
375F  03 4C 4C 43           1                DB      3,'LLC'                       ;
1353                        1       ORG      _CODE                                  ;
1353  64 1A 10                                       DB 64H,1AH,10H           ;Pc<Pc OR 10
                                             $NEXT
1356  48 84                 1                DB 48H,84H                             ;
1358  48 28                 1                DB 48H,28H                             ;

                            ;  LI             ( n --- )
                            ;                 load LCD setup instruction n, exit ready for char loads
                                             $COLON 2,'LI',LI
135A                        2       LI:                                           ;
3753                        2       ORG      _NAME                                  ;
3753  135A R 375F R         2                DW       _CODE,_LINK                   ;
3757  02 4C 49              2                DB      2,'LI'                       ;
135A                        2       ORG      _CODE                                  ;
135A  80                    1                DB 80H                                 ;
135B  134C R 1342 R 1353 R                   DW      LLI,LCD,LLC,EXIT
```

```
     0394 R

                            ;   LCDINIT     ( -- )
                            ;               Initialize LCD display.
                                           $COLON 7,'LCDINIT',LCDINIT
1363                        2       LCDINIT:                                          ;
3747                        2       ORG     _NAME                                   ;
3747  1363 R 3757 R         2              DW       _CODE,_LINK                     ;
374B  07 4C 43 44 49 4E 49  2              DB      7,'LCDINIT'                        ;
1363                        2       ORG     _CODE                                   ;
1363  80                    1              DB 80H                                   ;
1364  038D R 0D7A 1339 R                   DW      DOLIT,0D7AH,DELAY
136A  038D R 0038 135A R                   DW      DOLIT,038H,LI
1370  038D R 047E 1339 R                   DW      DOLIT,047EH,DELAY
1376  038D R 0038 135A R                   DW      DOLIT,038H,LI
137C  038D R 0017 1339 R                   DW      DOLIT,017H,DELAY
1382  038D R 0038 135A R                   DW      DOLIT,038H,LI
1388  038D R 0017 1339 R                   DW      DOLIT,017H,DELAY
138E  038D R 0038 135A R                   DW      DOLIT,038H,LI
1394  038D R 0017 1339 R                   DW      DOLIT,017H,DELAY
139A  038D R 0008 135A R                   DW      DOLIT,08H,LI
13A0  038D R 0017 1339 R                   DW      DOLIT,017H,DELAY
13A6  038D R 0001 135A R                   DW      DOLIT,01H,LI
13AC  038D R 01CC 1339 R                   DW      DOLIT,01CCH,DELAY
13B2  038D R 0002 135A R                   DW      DOLIT,02H,LI
13B8  038D R 01CC 1339 R                   DW      DOLIT,01CCH,DELAY
13BE  038D R 0006 135A R                   DW      DOLIT,06H,LI
13C4  038D R 0017 1339 R                   DW      DOLIT,17H,DELAY
13CA  038D R 000E 135A R                   DW      DOLIT,0EH,LI
13D0  038D R 0017 1339 R                   DW      DOLIT,17H,DELAY
13D6  0394 R                               DW      EXIT

                            ;  #DISP        ( n,p --- )
                            ;               Display n as a 3-digit number at LCD position p.
                                           $COLON  5,'#DISP',NDISP
13D8                        2       NDISP:                                          ;
373D                        2       ORG     _NAME                                   ;
373D  13D8 R 374B R         2              DW       _CODE,_LINK                     ;
3741  05 23 44 49 53 50     2              DB      5,'#DISP'                          ;
13D8                        2       ORG     _CODE                                   ;
13D8  80                    1              DB 80H                                   ;
13D9  043B R 135A R 0442 R                 DW      DUPP,LI,SWAP
13DF  08BB R 08D5 R 08D5 R                 DW      BDIGS,DIG,DIG,DIG,EDIGS
      08D5 R 08FE R
13E9  0436 R 043B R 03F1 R                 DW      DROP,DUPP,CAT,LCD,ONEP
      1342 R 061F R
13F3  043B R 03F1 R 1342 R                 DW      DUPP,CAT,LCD,ONEP,CAT,LCD,LI,EXIT
      061F R 03F1 R 1342 R
      135A R 0394 R

                            ;  DISP         ( a,p --- )
                            ;               Display packed string at a to LCD position p.
                                           $COLON  4,'DISP',DISP
1403                        2       DISP:                                          ;
3733                        2       ORG     _NAME                                   ;
3733  1403 R 3741 R         2              DW       _CODE,_LINK                     ;
```

```
3737  04 44 49 53 50        2              DB      4,'DISP'                        ;
1403                        2       ORG     _CODE                                   ;
1403  80                    1              DB 80H                                  ;
1404  135A R 043B R 03F1 R                 DW      LI,DUPP,CAT,TWOM,TOR
      0635 R 0413 R
140E  061F R                   DISP1:       DW      ONEP
1410  043B R 03F1 R 1342 R                 DW      DUPP,CAT,LCD
1416  039D R 140E R                        DW      DONXT,DISP1
141A  0436 R 0394 R                        DW      DROP,EXIT

                             ;  CASE         ( n --- )
                             ;               Execute one of a list of words pointed to by n.
                                           $COLON  4,'CASE',CASE
141E                        2       CASE:                                          ;
3729                        2       ORG     _NAME                                   ;
3729  141E R 3737 R         2              DW       _CODE,_LINK                    ;
372D  04 43 41 53 45        2              DB      4,'CASE'                        ;
141E                        2       ORG     _CODE                                   ;
141E  80                    1              DB 80H                                  ;
141F  0405 R 0442 R 063D R                 DW      RFROM,SWAP,TWOSL,PLUS
      0565 R
1427  07F6 R 0394 R                        DW      ATEXE,EXIT

                             ;  INCR         ( n,nmax --- n+1 )
                             ;               Increment n mod nmax.
                                           $COLON  4,'INCR',INCR
142B                        2       INCR:                                          ;
371F                        2       ORG     _NAME                                   ;
371F  142B R 372D R         2              DW       _CODE,_LINK                    ;
3723  04 49 4E 43 52        2              DB      4,'INCR'                        ;
142B                        2       ORG     _CODE                                   ;
142B  80                    1              DB 80H                                  ;
142C  044A R 061F R 05E1 R                 DW      OVER,ONEP,LESS
1432  03B7 R 1440 R                        DW      QBRAN,INCR1
1436  0436 R 038D R 0000                   DW      DROP,DOLIT,0
143C  03CC R 1442 R                        DW      BRAN,INCR2
1440  061F R                   INCR1:       DW      ONEP
1442  0394 R                   INCR2:       DW      EXIT

                             ;  DECR         ( n,nmax --- n-1 )
                             ;               Decrement n mod nmax.
                                           $COLON  4,'DECR',DECR
1444                        2       DECR:                                          ;
3715                        2       ORG     _NAME                                   ;
3715  1444 R 3723 R         2              DW       _CODE,_LINK                    ;
3719  04 44 45 43 52        2              DB      4,'DECR'                        ;
1444                        2       ORG     _CODE                                   ;
1444  80                    1              DB 80H                                  ;
1445  044A R 0626 R 0453 R                 DW      OVER,ONEM,ZLESS
144B  03B7 R 1457 R                        DW      QBRAN,DECR1
144F  0442 R 0436 R                        DW      SWAP,DROP
1453  03CC R 145B R                        DW      BRAN,DECR2
1457  0436 R                   DECR1:       DW      DROP
1459  0626 R                               DW      ONEM
145B  0394 R                   DECR2:       DW      EXIT
```

```
                            ;  SW@          ( --- n )
                            ;               Read Roland switches as a byte.
                                           $CODE   3,'SW@',SWAT
145D                        1       SWAT:                                          ;
370D                        1       ORG     _NAME                                   ;
370D  145D R 3719 R         1              DW       _CODE,_LINK                    ;
3711  03 53 57 40           1              DB      3,'SW@'                       ;
145D                        1       ORG     _CODE                                   ;
145D  4C C0                                DB      4CH,0C0H        ;;A<PA
145F  6A 00                                DB      6AH,0           ;;B<0
1461  1B                                   DB      1BH             ;;C<A
1462  B1                                   DB      0B1H            ;;PUSH BC
                                           $NEXT
1463  48 84                 1              DB 48H,84H                              ;
1465  48 28                 1              DB 48H,28H                              ;

                            ;  S@           ( --- n )
                            ;               Return number of the lowest Roland switch on.
                                           $CODE   2,'S@',SAT
1467                        1       SAT:                                          ;
3705                        1       ORG     _NAME                                   ;
3705  1467 R 3711 R         1              DW       _CODE,_LINK                    ;
3709  02 53 40              1              DB      2,'S@'                        ;
1467                        1       ORG     _CODE                                   ;
1467  4C C0                                DB      4CH,0C0H        ;;A<PA
1469  6B 00                                DB      6BH,0           ;;C<0
146B  74 11 FF                                     DB      74H,11H,0FFH    ;;A<A EXOR FF
146E  74 49 FF                                     DB      74H,49H,0FFH    ;;A AND FF, SKIP IF NO ZERO
1471  C4                                   DB      0C4H            ;; JMP OUT
1472  43                                   DB      43H             ;; C<C+1, LOOP1
1473  48 01                                DB      48H,1           ;; A SHIFT RIGHT, SKIP IF CARRY
1475  FC                                   DB      0FCH            ;; JMP LOOP1
1476  6A 00                                DB      6AH,0           ;;B<0, OUT
1478  B1                                   DB      0B1H            ;;PUSH BC
                                           $NEXT
1479  48 84                 1              DB 48H,84H                              ;
147B  48 28                 1              DB 48H,28H                              ;

                            ;  LED!         ( n --- )
                            ;               Turn on/off Roland LED's.
                                           $CODE   4,'LED!',LEDB
147D                        1       LEDB:                                          ;
36FB                        1       ORG     _NAME                                   ;
36FB  147D R 3709 R         1              DW       _CODE,_LINK                    ;
36FF  04 4C 45 44 21        1              DB      4,'LED!'                        ;
147D                        1       ORG     _CODE                                   ;
147D  A1                                   DB      0A1H            ;;POP BC
147E  0B                                   DB      0BH             ;;A<C
147F  74 09 FC                                     DB      74H,9H,0FCH     ;;A<A AND FC
1482  74 19 01                                     DB      74H,19H,1       ;;A<A OR 1
1485  4D C1                                DB      4DH,0C1H        ;;PB<A
                                           $NEXT
1487  48 84                 1              DB 48H,84H                              ;
1489  48 28                 1              DB 48H,28H                              ;

                            ;  eUPDAT       ( --- )
```

```
                             ;               Move data from Slider Ram to Edit Buffer.
                                            $CODE   6,'eUPDAT',EUPDAT
148B                         1       EUPDAT:                                            ;
36EF                         1       ORG     _NAME                                      ;
36EF  148B R 36FF R          1              DW       _CODE,_LINK                       ;
36F3  06 65 55 50 44 41 54  1              DB       6,'eUPDAT'                          ;
148B                         1       ORG     _CODE                                      ;
148B  68 FF                                DB       68H,0FFH        ;;V<FF
148D  6A C6                                DB       6AH,0C6H        ;;B<C6
148F  01 F0                                DB       1,0F0H          ;;A<(V/F0)
1491  1B                                   DB       1BH             ;;C<A
1492  29                                   DB       29H             ;;A<(BC)
1493  63 04                                DB       63H,4           ;;(V/04)<A
1495  69 38                                DB       69H,38H         ;;A<70
1497  60 43                                DB       60H,43H         ;;C<C+A
1499  29                                   DB       29H             ;;A<(BC)
149A  63 06                                DB       63H,6H          ;;(V/06)<A
149C  69 38                                DB       69H,38H         ;;A<38
149E  60 43                                DB       60H,43H         ;;C<C+A
14A0  29                                   DB       29H             ;;A<(BC)
14A1  63 07                                DB       63H,7H          ;;(V/07)<A
                                           $NEXT
14A3  48 84                  1             DB 48H,84H                                  ;
14A5  48 28                  1             DB 48H,28H                                  ;

                             ;  eLOAD        ( --- )
                             ;               Load Edit Buffer data into Slider Memory.
                                           $CODE   5,'eLOAD',ELOAD
14A7                         1       ELOAD:                                             ;
36E5                         1       ORG     _NAME                                      ;
36E5  14A7 R 36F3 R          1              DW       _CODE,_LINK                       ;
36E9  05 65 4C 4F 41 44      1              DB       5,'eLOAD'                          ;
14A7                         1       ORG     _CODE                                      ;
14A7  68 FF                                DB       68H,0FFH        ;;V<FF
14A9  6A C6                                DB       6AH,0C6H        ;;B<C6
14AB  01 00                                DB       1,0             ;;A<(V/00)
14AD  1B                                   DB       1BH             ;;C<A
14AE  01 04                                DB       1,4             ;;A<(V/04)
14B0  39                                   DB       39H             ;;(BC)<A
14B1  69 38                                DB       69H,38H         ;;A<38
14B3  60 43                                DB       60H,43H         ;;C<C+A
14B5  49 00                                DB       49H,0           ;;(BC)<0
14B7  69 70                                DB       69H,70H         ;;A<38
14B9  60 43                                DB       60H,43H         ;;C<C+A
14BB  01 06                                DB       1,6             ;;A<(V/06)
14BD  39                                   DB       39H             ;;(BC)<A
14BE  69 38                                DB       69H,38H         ;;A<38
14C0  60 43                                DB       60H,43H         ;;C<C+A
14C2  01 07                                DB       1,7             ;;A<(V/07)
14C4  39                                   DB       39H             ;;(BC)<A
                                           $NEXT
14C5  48 84                  1             DB 48H,84H                                  ;
14C7  48 28                  1             DB 48H,28H                                  ;

                             ;  esUPDAT      ( --- )
                             ;               Update only the Slider data of the Edit Buffer.
```

```
                                             $CODE   7,'esUPDAT',ESUPDAT
14C9                         1       ESUPDAT:                                          ;
36D9                         1       ORG     _NAME                                    ;
36D9  14C9 R 36E9 R          1               DW       _CODE,_LINK                     ;
36DD  07 65 73 55 50 44 41   1               DB      7,'esUPDAT'                        ;
14C9                         1       ORG     _CODE                                    ;
14C9  68 FF                                  DB      68H,0FFH        ;;V<FF
14CB  6A C6                                  DB      6AH,0C6H        ;;B<C6
14CD  01 F0                                  DB      1,0F0H          ;;A<(V/F0)
14CF  1B                                     DB      1BH             ;;C<A
14D0  29                                     DB      29H             ;;A<(BC)
14D1  63 04                                  DB      63H,4           ;;(V/04)<A
                                             $NEXT
14D3  48 84                  1               DB 48H,84H                               ;
14D5  48 28                  1               DB 48H,28H                               ;

                             ;  eSLD#        ( --- FF00 )
                             ;               Edit Buffer Slider number.
                                             $COLON  5,'eSLD#',ESLDN
14D7                         2       ESLDN:                                           ;
36CF                         2       ORG     _NAME                                    ;
36CF  14D7 R 36DD R          2               DW       _CODE,_LINK                     ;
36D3  05 65 53 4C 44 23      2               DB      5,'eSLD#'                         ;
14D7                         2       ORG     _CODE                                    ;
14D7  80                     1               DB 80H                                   ;
14D8  038D R FF00 0394 R                     DW      DOLIT,0FF00H,EXIT

                             ;  eFLD         ( --- FF01 )
                             ;               Edit Buffer LCD Field.
                                             $COLON  4,'eFLD',EFLD
14DE                         2       EFLD:                                           ;
36C5                         2       ORG     _NAME                                    ;
36C5  14DE R 36D3 R          2               DW       _CODE,_LINK                     ;
36C9  04 65 46 4C 44         2               DB      4,'eFLD'                         ;
14DE                         2       ORG     _CODE                                    ;
14DE  80                     1               DB 80H                                   ;
14DF  038D R FF01 0394 R                     DW      DOLIT,0FF01H,EXIT

                             ;  eBYTE1       ( --- FF07 )
                             ;               Edit Buffer Midi Status/Chnl byte.
                                             $COLON  6,'eBYTE1',EBYTE1
14E5                         2       EBYTE1:                                           ;
36B9                         2       ORG     _NAME                                    ;
36B9  14E5 R 36C9 R          2               DW       _CODE,_LINK                     ;
36BD  06 65 42 59 54 45 31   2               DB      6,'eBYTE1'                         ;
14E5                         2       ORG     _CODE                                    ;
14E5  80                     1               DB 80H                                   ;
14E6  038D R FF07 0394 R                     DW      DOLIT,0FF07H,EXIT

                             ;  eBYTE2       ( --- FF06 )
                             ;               Edit Buffer Midi Key#, Controller#, or Program# byte.
                                             $COLON  6,'eBYTE2',EBYTE2
14EC                         2       EBYTE2:                                           ;
36AD                         2       ORG     _NAME                                    ;
36AD  14EC R 36BD R          2               DW       _CODE,_LINK                     ;
36B1  06 65 42 59 54 45 32   2               DB      6,'eBYTE2'                         ;
```

```
14EC                        2       ORG    _CODE                                    ;
14EC  80                    1              DB 80H                                   ;
14ED  038D R FF06 0394 R                   DW      DOLIT,0FF06H,EXIT

                            ;  eBYTE3       ( --- FF04 )
                            ;               Edit Buffer Slider value.
                                           $COLON  6,'eBYTE3',EBYTE3
14F3                        2       EBYTE3:                                             ;
36A1                        2       ORG    _NAME                                    ;
36A1  14F3 R 36B1 R         2              DW       _CODE,_LINK                     ;
36A5  06 65 42 59 54 45 33  2              DB      6,'eBYTE3'                        ;
14F3                        2       ORG    _CODE                                    ;
14F3  80                    1              DB 80H                                   ;
14F4  038D R FF04 0394 R                   DW      DOLIT,0FF04H,EXIT

                            ;  eFLAG        ( --- FF05 )
                            ;               Edit Buffer Midi Flag byte.
                                           $COLON  5,'eFLAG',EFLAG
14FA                        2       EFLAG:                                              ;
3697                        2       ORG    _NAME                                    ;
3697  14FA R 36A5 R         2              DW       _CODE,_LINK                     ;
369B  05 65 46 4C 41 47     2              DB      5,'eFLAG'                         ;
14FA                        2       ORG    _CODE                                    ;
14FA  80                    1              DB 80H                                   ;
14FB  038D R FF05 0394 R                   DW      DOLIT,0FF05H,EXIT

                            ;  FLD0         ( --- 80 )
                            ;               LCD Field start.
                                           $COLON  4,'FLD0',FLD0
1501                        2       FLD0:                                              ;
368D                        2       ORG    _NAME                                    ;
368D  1501 R 369B R         2              DW       _CODE,_LINK                     ;
3691  04 46 4C 44 30        2              DB      4,'FLD0'                         ;
1501                        2       ORG    _CODE                                    ;
1501  80                    1              DB 80H                                   ;
1502  038D R 0080 0394 R                   DW      DOLIT,080H,EXIT

                            ;  FLD1         ( --- 86 )
                            ;               LCD Field start.
                                           $COLON  4,'FLD1',FLD01
1508                        2       FLD01:                                              ;
3683                        2       ORG    _NAME                                    ;
3683  1508 R 3691 R         2              DW       _CODE,_LINK                     ;
3687  04 46 4C 44 31        2              DB      4,'FLD1'                         ;
1508                        2       ORG    _CODE                                    ;
1508  80                    1              DB 80H                                   ;
1509  038D R 0086 0394 R                   DW      DOLIT,086H,EXIT

                            ;  FLD2         ( --- 8A )
                            ;               LCD Field start.
                                           $COLON  4,'FLD2',FLD2
150F                        2       FLD2:                                              ;
3679                        2       ORG    _NAME                                    ;
3679  150F R 3687 R         2              DW       _CODE,_LINK                     ;
367D  04 46 4C 44 32        2              DB      4,'FLD2'                         ;
150F                        2       ORG    _CODE                                    ;
```

```
150F  80                    1              DB 80H                                  ;
1510  038D R 008A 0394 R                   DW      DOLIT,08AH,EXIT

                            ;  FLD3         ( --- 8D )
                            ;               LCD Field start.
                                           $COLON  4,'FLD3',FLD3
1516                        2       FLD3:                                          ;
366F                        2       ORG     _NAME                                   ;
366F  1516 R 367D R         2              DW       _CODE,_LINK                    ;
3673  04 46 4C 44 33        2              DB      4,'FLD3'                        ;
1516                        2       ORG     _CODE                                   ;
1516  80                    1              DB 80H                                  ;
1517  038D R 008D 0394 R                   DW      DOLIT,08DH,EXIT

                            ;  FLD4         ( --- C0 )
                            ;               LCD Field start.
                                           $COLON  4,'FLD4',FLD4
151D                        2       FLD4:                                          ;
3665                        2       ORG     _NAME                                   ;
3665  151D R 3673 R         2              DW       _CODE,_LINK                    ;
3669  04 46 4C 44 34        2              DB      4,'FLD4'                        ;
151D                        2       ORG     _CODE                                   ;
151D  80                    1              DB 80H                                  ;
151E  038D R 00C0 0394 R                   DW      DOLIT,0C0H,EXIT

                            ;  FLD5         ( --- C9 )
                            ;               LCD Field start.
                                           $COLON  4,'FLD5',FLD5
1524                        2       FLD5:                                          ;
365B                        2       ORG     _NAME                                   ;
365B  1524 R 3669 R         2              DW       _CODE,_LINK                    ;
365F  04 46 4C 44 35        2              DB      4,'FLD5'                        ;
1524                        2       ORG     _CODE                                   ;
1524  80                    1              DB 80H                                  ;
1525  038D R 00C9 0394 R                   DW      DOLIT,0C9H,EXIT

                            ;  FLD6         ( --- CD )
                            ;               LCD Field start.
                                           $COLON  4,'FLD6',FLD6
152B                        2       FLD6:                                          ;
3651                        2       ORG     _NAME                                   ;
3651  152B R 365F R         2              DW       _CODE,_LINK                    ;
3655  04 46 4C 44 36        2              DB      4,'FLD6'                        ;
152B                        2       ORG     _CODE                                   ;
152B  80                    1              DB 80H                                  ;
152C  038D R 00CD 0394 R                   DW      DOLIT,0CDH,EXIT

                            ;  L0           ( --- a )
                            ;               Packed string. 'a' is addr of count byte.
                                           $COLON  2,'L0',L0
1532                        2       L0:                                          ;
3649                        2       ORG     _NAME                                   ;
3649  1532 R 3655 R         2              DW       _CODE,_LINK                    ;
364D  02 4C 30              2              DB      2,'L0'                        ;
1532                        2       ORG     _CODE                                   ;
1532  80                    1              DB 80H                                  ;
```

```
                                           SD$ 'Slider'
1533  038D R                1              DW DOLIT
1535  1539 R 0394 R         1              DW      _LEN,EXIT                       ;
1539  00 53 6C 69 64 65 72  1              DB      0,'Slider'                         ;
1539                        1       ORG     _LEN                                   ;
1539  06                    1              DB      _CODE-_LEN-1                    ;
1540                        1       ORG     _CODE                                  ;

                            ;  L1           ( --- a )
                            ;               Packed string. 'a' is addr of count byte.
                                           $COLON  2,'L1',L1
1540                        2       L1:                                         ;
3641                        2       ORG     _NAME                                  ;
3641  1540 R 364D R         2              DW       _CODE,_LINK                    ;
3645  02 4C 31              2              DB      2,'L1'                       ;
1540                        2       ORG     _CODE                                  ;
1540  80                    1              DB 80H                                  ;
                                           SD$ 'Setup#'
1541  038D R                1              DW DOLIT
1543  1547 R 0394 R         1              DW      _LEN,EXIT                       ;
1547  00 53 65 74 75 70 23  1              DB      0,'Setup#'                         ;
1547                        1       ORG     _LEN                                   ;
1547  06                    1              DB      _CODE-_LEN-1                    ;
154E                        1       ORG     _CODE                                  ;

                            ;  L2           ( --- a )
                            ;               Packed string. 'a' is addr of count byte.
                                           $COLON  2,'L2',L2
154E                        2       L2:                                         ;
3639                        2       ORG     _NAME                                  ;
3639  154E R 3645 R         2              DW       _CODE,_LINK                    ;
363D  02 4C 32              2              DB      2,'L2'                       ;
154E                        2       ORG     _CODE                                  ;
154E  80                    1              DB 80H                                  ;
                                           SD$ '* MIDI Running *'
154F  038D R                1              DW DOLIT
1551  1555 R 0394 R         1              DW      _LEN,EXIT                       ;
1555  00 2A 20 4D 49 44 49  1              DB      0,'* MIDI Running *'                       ;
1555                        1       ORG     _LEN                                   ;
1555  10                    1              DB      _CODE-_LEN-1                    ;
1566                        1       ORG     _CODE                                  ;

                            ;  L20          ( --- a )
                            ;               Packed string. 'a' is addr of count byte.
                                           $COLON  3,'L20',L20
1566                        2       L20:                                         ;
3631                        2       ORG     _NAME                                  ;
3631  1566 R 363D R         2              DW       _CODE,_LINK                    ;
3635  03 4C 32 30           2              DB      3,'L20'                       ;
1566                        2       ORG     _CODE                                  ;
1566  80                    1              DB 80H                                  ;
                                           SD$ 'Ch '
1567  038D R                1              DW DOLIT
1569  156D R 0394 R         1              DW      _LEN,EXIT                       ;
156D  00 43 68 20           1              DB      0,'Ch '                         ;
156D                        1       ORG     _LEN                                   ;
```

```
156D  03                      1               DB      _CODE-_LEN-1                    ;
1571                          1       ORG     _CODE                                    ;

                              ;  L21           ( --- a )
                              ;                Packed string. 'a' is addr of count byte.
                                              $COLON  3,'L21',L21
1571                          2       L21:                                            ;
3629                          2       ORG     _NAME                                    ;
3629  1571 R 3635 R           2               DW       _CODE,_LINK                    ;
362D  03 4C 32 31             2               DB      3,'L21'                        ;
1571                          2       ORG     _CODE                                    ;
1571  80                      1               DB 80H                                  ;
                                              SD$ 'Off'
1572  038D R                  1               DW DOLIT
1574  1578 R 0394 R           1               DW      _LEN,EXIT                       ;
1578  00 4F 66 66             1               DB      0,'Off'                         ;
1578                          1       ORG     _LEN                                     ;
1578  03                      1               DB      _CODE-_LEN-1                    ;
157C                          1       ORG     _CODE                                    ;

                              ;  L40           ( --- a )
                              ;                Packed string. 'a' is addr of count byte.
                                              $COLON  3,'L40',L40
157C                          2       L40:                                            ;
3621                          2       ORG     _NAME                                    ;
3621  157C R 362D R           2               DW       _CODE,_LINK                    ;
3625  03 4C 34 30             2               DB      3,'L40'                        ;
157C                          2       ORG     _CODE                                    ;
157C  80                      1               DB 80H                                  ;
                                              SD$ 'Key#    '
157D  038D R                  1               DW DOLIT
157F  1583 R 0394 R           1               DW      _LEN,EXIT                       ;
1583  00 4B 65 79 23 20 20    1               DB      0,'Key#    '                          ;
1583                          1       ORG     _LEN                                     ;
1583  08                      1               DB      _CODE-_LEN-1                    ;
158C                          1       ORG     _CODE                                    ;

                              ;  L41           ( --- a )
                              ;                Packed string. 'a' is addr of count byte.
                                              $COLON  3,'L41',L41
158C                          2       L41:                                            ;
3619                          2       ORG     _NAME                                    ;
3619  158C R 3625 R           2               DW       _CODE,_LINK                    ;
361D  03 4C 34 31             2               DB      3,'L41'                        ;
158C                          2       ORG     _CODE                                    ;
158C  80                      1               DB 80H                                  ;
                                              SD$ 'Key# A-T'
158D  038D R                  1               DW DOLIT
158F  1593 R 0394 R           1               DW      _LEN,EXIT                       ;
1593  00 4B 65 79 23 20 41    1               DB      0,'Key# A-T'                          ;
1593                          1       ORG     _LEN                                     ;
1593  08                      1               DB      _CODE-_LEN-1                    ;
159C                          1       ORG     _CODE                                    ;

                              ;  L42           ( --- a )
                              ;                Packed string. 'a' is addr of count byte.
```

```
                                          $COLON  3,'L42',L42
159C                         2       L42:                                          ;
3611                         2       ORG      _NAME                                  ;
3611  159C R 361D R          2                DW       _CODE,_LINK                   ;
3615  03 4C 34 32            2                DB      3,'L42'                       ;
159C                         2       ORG      _CODE                                  ;
159C  80                     1                DB 80H                                 ;
                                              SD$ 'Control#'
159D  038D R                 1                DW DOLIT
159F  15A3 R 0394 R          1                DW       _LEN,EXIT                     ;
15A3  00 43 6F 6E 74 72 6F  1                DB      0,'Control#'                         ;
15A3                         1       ORG      _LEN                                   ;
15A3  08                     1                DB      _CODE-_LEN-1                   ;
15AC                         1       ORG      _CODE                                  ;

                             ;  L43          ( --- a )
                             ;               Packed string. 'a' is addr of count byte.
                                              $COLON  3,'L43',L43
15AC                         2       L43:                                          ;
3609                         2       ORG      _NAME                                  ;
3609  15AC R 3615 R          2                DW       _CODE,_LINK                   ;
360D  03 4C 34 33            2                DB      3,'L43'                       ;
15AC                         2       ORG      _CODE                                  ;
15AC  80                     1                DB 80H                                 ;
                                              SD$ 'Program#'
15AD  038D R                 1                DW DOLIT
15AF  15B3 R 0394 R          1                DW       _LEN,EXIT                     ;
15B3  00 50 72 6F 67 72 61  1                DB      0,'Program#'                         ;
15B3                         1       ORG      _LEN                                   ;
15B3  08                     1                DB      _CODE-_LEN-1                   ;
15BC                         1       ORG      _CODE                                  ;

                             ;  L44          ( --- a )
                             ;               Packed string. 'a' is addr of count byte.
                                              $COLON  3,'L44',L44
15BC                         2       L44:                                          ;
3601                         2       ORG      _NAME                                  ;
3601  15BC R 360D R          2                DW       _CODE,_LINK                   ;
3605  03 4C 34 34            2                DB      3,'L44'                       ;
15BC                         2       ORG      _CODE                                  ;
15BC  80                     1                DB 80H                                 ;
                                              SD$ 'Ch Press'
15BD  038D R                 1                DW DOLIT
15BF  15C3 R 0394 R          1                DW       _LEN,EXIT                     ;
15C3  00 43 68 20 50 72 65  1                DB      0,'Ch Press'                         ;
15C3                         1       ORG      _LEN                                   ;
15C3  08                     1                DB      _CODE-_LEN-1                   ;
15CC                         1       ORG      _CODE                                  ;

                             ;  L45          ( --- a )
                             ;               Packed string. 'a' is addr of count byte.
                                              $COLON  3,'L45',L45
15CC                         2       L45:                                          ;
35F9                         2       ORG      _NAME                                  ;
35F9  15CC R 3605 R          2                DW       _CODE,_LINK                   ;
35FD  03 4C 34 35            2                DB      3,'L45'                       ;
```

```
15CC                         2       ORG     _CODE                                  ;
15CC  80                     1               DB 80H                                 ;
                                             SD$ 'Ptch Whl'
15CD  038D R                 1               DW DOLIT
15CF  15D3 R 0394 R          1               DW      _LEN,EXIT                      ;
15D3  00 50 74 63 68 20 57   1               DB      0,'Ptch Whl'                        ;
15D3                         1       ORG     _LEN                                   ;
15D3  08                     1               DB      _CODE-_LEN-1                   ;
15DC                         1       ORG     _CODE                                  ;

                             ;  L4X           ( --- a )
                             ;                Packed string. 'a' is addr of count byte.
                                             $COLON  3,'L4X',L4X
15DC                         2       L4X:                                         ;
35F1                         2       ORG     _NAME                                  ;
35F1  15DC R 35FD R          2               DW       _CODE,_LINK                   ;
35F5  03 4C 34 58            2               DB      3,'L4X'                      ;
15DC                         2       ORG     _CODE                                  ;
15DC  80                     1               DB 80H                                 ;
                                             SD$ '********'
15DD  038D R                 1               DW DOLIT
15DF  15E3 R 0394 R          1               DW      _LEN,EXIT                      ;
15E3  00 2A 2A 2A 2A 2A 2A   1               DB      0,'********'                        ;
15E3                         1       ORG     _LEN                                   ;
15E3  08                     1               DB      _CODE-_LEN-1                   ;
15EC                         1       ORG     _CODE                                  ;

                             ;  L50           ( --- a )
                             ;                Packed string. 'a' is addr of count byte.
                                             $COLON  3,'L50',L50
15EC                         2       L50:                                         ;
35E9                         2       ORG     _NAME                                  ;
35E9  15EC R 35F5 R          2               DW       _CODE,_LINK                   ;
35ED  03 4C 35 30            2               DB      3,'L50'                      ;
15EC                         2       ORG     _CODE                                  ;
15EC  80                     1               DB 80H                                 ;
                                             SD$ '***'
15ED  038D R                 1               DW DOLIT
15EF  15F3 R 0394 R          1               DW      _LEN,EXIT                      ;
15F3  00 2A 2A 2A            1               DB      0,'***'                        ;
15F3                         1       ORG     _LEN                                   ;
15F3  03                     1               DB      _CODE-_LEN-1                   ;
15F7                         1       ORG     _CODE                                  ;

                             ;  LSTAT         ( n --- )
                             ;                Choose a midi status label.
                                             $COLON  5,'LSTAT',LSTAT
15F7                         2       LSTAT:                                         ;
35DF                         2       ORG     _NAME                                  ;
35DF  15F7 R 35ED R          2               DW       _CODE,_LINK                   ;
35E3  05 4C 53 54 41 54      2               DB      5,'LSTAT'                       ;
15F7                         2       ORG     _CODE                                  ;
15F7  80                     1               DB 80H                                 ;
15F8  141E R 15DC R 157C R                   DW      CASE,L4X,L40,L41,L42,L43,L44,L45,L4X,EXIT
      158C R 159C R 15AC R
      15BC R 15CC R 15DC R
```

```
     0394 R

                              ;  eDISP        ( --- )
                              ;               Display the Edit buffer on the LCD
                                             $COLON  5,'eDISP',EDISP
160C                          2      EDISP:                                         ;
35D5                          2      ORG     _NAME                                  ;
35D5  160C R 35E3 R           2              DW       _CODE,_LINK                  ;
35D9  05 65 44 49 53 50       2              DB      5,'eDISP'                      ;
160C                          2      ORG     _CODE                                  ;
160C  80                      1              DB 80H                                 ;
160D  1532 R 1501 R 1403 R                   DW      L0,FLD0,DISP,ESLDN,CAT,FLD01,NDISP
      14D7 R 03F1 R 1508 R
      13D8 R
161B  14F3 R 03F1 R 038D R                   DW      EBYTE3,CAT,DOLIT,80H,ANDD
      0080 0461 R
1625  03B7 R 1633 R                          DW      QBRAN,EDISP1
1629  1571 R 150F R 1403 R                   DW      L21,FLD2,DISP
162F  03CC R 1639 R                          DW      BRAN,EDISP2
1633  1566 R 150F R 1403 R     EDISP1:        DW       L20,FLD2,DISP
1639  14E5 R 03F1 R 043B R     EDISP2:        DW       EBYTE1,CAT,DUPP,DOLIT,0FH,ANDD,FLD3,NDISP
      038D R 000F 0461 R
      1516 R 13D8 R
1649  063D R 063D R 063D R                   DW      TWOSL,TWOSL,TWOSL,TWOSL,DOLIT,7H,ANDD
      063D R 038D R 0007
      0461 R
1657  15F7 R 151D R 1403 R                   DW      LSTAT,FLD4,DISP,EBYTE2,CAT,FLD5,NDISP
      14EC R 03F1 R 1524 R
      13D8 R
1665  14F3 R 03F1 R 038D R                   DW      EBYTE3,CAT,DOLIT,7FH,ANDD,FLD6,NDISP
      007F 0461 R 152B R
      13D8 R
1673  038D R 0006 14DE R                     DW      DOLIT,6H,EFLD,CSTOR,EXIT
      03E9 R 0394 R

                              ;  BL/R         ( fld --- pos )
                              ;               Translates LCD field number to a position number.
                                             $COLON  4,'BL/R',BLR
167D                          2      BLR:                                         ;
35CB                          2      ORG     _NAME                                  ;
35CB  167D R 35D9 R           2              DW       _CODE,_LINK                   ;
35CF  04 42 4C 2F 52          2              DB      4,'BL/R'                       ;
167D                          2      ORG     _CODE                                  ;
167D  80                      1              DB 80H                                 ;
167E  043B R 14DE R 03F1 R                   DW      DUPP,EFLD,CAT,CASE
      141E R
1686  1501 R 1508 R 150F R                   DW      FLD0,FLD01,FLD2,FLD3,FLD4,FLD5,FLD6
      1516 R 151D R 1524 R
      152B R
1694  0394 R                                 DW      EXIT

                              ;  BLEFT        ( --- )
                              ;               Moves the LCD cursor to next field. Loads eFLD.
                                             $COLON  5,'BLEFT',BLEFT
1696                          2      BLEFT:                                         ;
35C1                          2      ORG     _NAME                                  ;
```

```
35C1  1696 R 35CF R         2               DW       _CODE,_LINK                    ;
35C5  05 42 4C 45 46 54     2               DB      5,'BLEFT'                        ;
1696                        2       ORG     _CODE                                   ;
1696  80                    1               DB 80H                                  ;
1697  038D R 0040 147D R                    DW      DOLIT,40H,LEDB
169D  14DE R 03F1 R 038D R                  DW      EFLD,CAT,DOLIT,6,DECR,BLR,LI,EXIT
      0006 1444 R 167D R
      135A R 0394 R

                            ;  BRIGHT        ( --- )
                            ;                Moves the LCD cursor to next field. Loads eFLD.
                                            $COLON  6,'BRIGHT',BRIGHT
16AD                        2       BRIGHT:                                          ;
35B5                        2       ORG     _NAME                                   ;
35B5  16AD R 35C5 R         2               DW       _CODE,_LINK                    ;
35B9  06 42 52 49 47 48 54  2               DB      6,'BRIGHT'                        ;
16AD                        2       ORG     _CODE                                   ;
16AD  80                    1               DB 80H                                  ;
16AE  038D R 0080 147D R                    DW      DOLIT,80H,LEDB
16B4  14DE R 03F1 R 038D R                  DW      EFLD,CAT,DOLIT,6,INCR,BLR,LI,EXIT
      0006 142B R 167D R
      135A R 0394 R

                            ;  BLOAD         ( --- )
                            ;                Load Buffer data shown on LCD into Slider Memory.
                                            $COLON  5,'BLOAD',BLOAD
16C4                        2       BLOAD:                                          ;
35AB                        2       ORG     _NAME                                   ;
35AB  16C4 R 35B9 R         2               DW       _CODE,_LINK                    ;
35AF  05 42 4C 4F 41 44     2               DB      5,'BLOAD'                        ;
16C4                        2       ORG     _CODE                                   ;
16C4  80                    1               DB 80H                                  ;
16C5  038D R 0004 147D R                    DW      DOLIT,4,LEDB
16CB  14A7 R 0394 R                         DW      ELOAD,EXIT

                            ;  BMIDI         ( --- )
                            ;                Start the Midi program.
                                            $COLON  5,'BMIDI',BMIDI
16CF                        2       BMIDI:                                          ;
35A1                        2       ORG     _NAME                                   ;
35A1  16CF R 35AF R         2               DW       _CODE,_LINK                    ;
35A5  05 42 4D 49 44 49     2               DB      5,'BMIDI'                        ;
16CF                        2       ORG     _CODE                                   ;
16CF  80                    1               DB 80H                                  ;
16D0  038D R 0001 135A R                    DW      DOLIT,1,LI
16D6  154E R 1501 R 1403 R                  DW      L2,FLD0,DISP
16DC  0394 R                                DW      EXIT

                            ;  BUP           ( --- )
                            ;                Increment value in LCD cursor field.
                                            $COLON  3,'BUP',BUP
16DE                        2       BUP:                                          ;
3599                        2       ORG     _NAME                                   ;
3599  16DE R 35A5 R         2               DW       _CODE,_LINK                    ;
359D  03 42 55 50           2               DB      3,'BUP'                        ;
16DE                        2       ORG     _CODE                                   ;
```

```
16DE  80                    1              DB 80H                                   ;
16DF  038D R 0010 147D R                   DW      DOLIT,10H,LEDB
16E5  038D R 0001 14DE R                   DW      DOLIT,1,EFLD,CAT
      03F1 R
16ED  038D R 0007 0461 R                   DW      DOLIT,7,ANDD,CASE
      141E R
16F5  1730 R 1747 R 1779 R                 DW      UD0,UD1,UD2,UD3,UD4,UD5,UD6,UD7,EXIT
      17B6 R 17F0 R 1846 R
      173E R 1739 R 0394 R

                            ;  BDOWN        ( --- )
                            ;               Decrement value in LCD cursor field.
                                           $COLON  3,'BDOWN',BDOWN
1707                        2       BDOWN:                                          ;
3591                        2       ORG     _NAME                                   ;
3591  1707 R 359D R         2              DW       _CODE,_LINK                     ;
3595  03 42 44 4F 57 4E     2              DB      3,'BDOWN'                         ;
1707                        2       ORG     _CODE                                   ;
1707  80                    1              DB 80H                                   ;
1708  038D R 0020 147D R                   DW      DOLIT,20H,LEDB
170E  038D R 0000 14DE R                   DW      DOLIT,0,EFLD,CAT
      03F1 R
1716  038D R 0007 0461 R                   DW      DOLIT,7,ANDD,CASE
      141E R
171E  1730 R 1747 R 1779 R                 DW      UD0,UD1,UD2,UD3,UD4,UD5,UD6,UD7,EXIT
      17B6 R 17F0 R 1846 R
      173E R 1739 R 0394 R

                            ;  U/D0         ( i/d --- )
                            ;               Field increment/decrement routine.
                                           $COLON  4,'U/D0',UD0
1730                        2       UD0:                                          ;
3587                        2       ORG     _NAME                                   ;
3587  1730 R 3595 R         2              DW       _CODE,_LINK                     ;
358B  04 55 2F 44 30        2              DB      4,'U/D0'                         ;
1730                        2       ORG     _CODE                                   ;
1730  80                    1              DB 80H                                   ;
1731  0436 R 1501 R 135A R                 DW      DROP,FLD0,LI,EXIT
      0394 R

                            ; U/D7          ( i/d --- )
                            ;               Field increment/decrement routine. (bogus field)
                                           $COLON  4,'U/D7',UD7
1739                        2       UD7:                                          ;
357D                        2       ORG     _NAME                                   ;
357D  1739 R 358B R         2              DW       _CODE,_LINK                     ;
3581  04 55 2F 44 37        2              DB      4,'U/D7'                         ;
1739                        2       ORG     _CODE                                   ;
1739  80                    1              DB 80H                                   ;
173A  0436 R 0394 R                        DW      DROP,EXIT

                            ;  U/D6         ( i/d --- )
                            ;               Field increment/decrement routine.
                                           $COLON  4,'U/D6',UD6
173E                        2       UD6:                                          ;
3573                        2       ORG     _NAME                                   ;
```

```
3573  173E R 3581 R         2               DW       _CODE,_LINK                    ;
3577  04 55 2F 44 36        2               DB      4,'U/D6'                        ;
173E                        2       ORG     _CODE                                    ;
173E  80                    1               DB 80H                                  ;
173F  0436 R 152B R 135A R                  DW      DROP,FLD6,LI,EXIT
      0394 R

                              ;  U/D1         ( i/d --- )
                              ;               Field increment/decrement routine.
                                            $COLON  4,'U/D1',UD1
1747                        2       UD1:                                          ;
3569                        2       ORG     _NAME                                    ;
3569  1747 R 3577 R         2               DW       _CODE,_LINK                    ;
356D  04 55 2F 44 31        2               DB      4,'U/D1'                        ;
1747                        2       ORG     _CODE                                    ;
1747  80                    1               DB 80H                                  ;
1748  14D7 R 03F1 R 038D R                  DW      ESLDN,CAT,DOLIT,37H,ROT
      0037 054A R
1752  03B7 R 175C R                         DW      QBRAN,UD1A
1756  142B R                                DW      INCR
1758  03CC R 175E R                         DW      BRAN,UD1B
175C  1444 R                  UD1A:          DW      DECR
175E  1762 R 0394 R           UD1B:          DW      CFLD1,EXIT

                              ;  CFLD1        ( sld# --- )
                              ;               Change Slider# in field 1.  Update Edit buffer & LCD.
                                            $COLON  5,'CFLD1',CFLD1
1762                        2       CFLD1:                                         ;
355F                        2       ORG     _NAME                                    ;
355F  1762 R 356D R         2               DW       _CODE,_LINK                    ;
3563  05 43 46 4C 44 31     2               DB      5,'CFLD1'                        ;
1762                        2       ORG     _CODE                                    ;
1762  80                    1               DB 80H                                  ;
1763  14D7 R 03F1 R 148B R                  DW      ESLDN,CAT,EUPDAT,EDISP
      160C R
176B  038D R 0001 14DE R                    DW      DOLIT,1,EFLD,CSTOR,FLD01,LI,EXIT
      03E9 R 1508 R 135A R
      0394 R

                              ;  U/D2         ( i/d --- )
                              ;               Field increment/decrement routine.
                                            $COLON  4,'U/D2',UD2
1779                        2       UD2:                                          ;
3555                        2       ORG     _NAME                                    ;
3555  1779 R 3563 R         2               DW       _CODE,_LINK                    ;
3559  04 55 2F 44 32        2               DB      4,'U/D2'                        ;
1779                        2       ORG     _CODE                                    ;
1779  80                    1               DB 80H                                  ;
177A  0436 R 14F3 R 03F1 R                  DW      DROP,EBYTE3,CAT,DUPP,DOLIT,80H,ANDD
      043B R 038D R 0080
      0461 R
1788  03B7 R 17A0 R                         DW      QBRAN,UD2A
178C  038D R 007F 0461 R                    DW      DOLIT,7FH,ANDD,EBYTE3,CAT,L20,FLD2,DISP
      14F3 R 03F1 R 1566 R
      150F R 1403 R
179C  03CC R 17B0 R                         DW      BRAN,UD2B
```

```
17A0  038D R 0080 046A R       UD2A:        DW      DOLIT,80H,ORR,EBYTE3,CSTOR,L21,FLD2,DISP
      14F3 R 03E9 R 1571 R
      150F R 1403 R
17B0  150F R 135A R 0394 R     UD2B:        DW      FLD2,LI,EXIT

                             ;  U/D3         ( i/d --- )
                             ;               Field increment/decrement routine.
                                            $COLON  4,'U/D3',UD3
17B6                         2       UD3:                                              ;
354B                         2       ORG     _NAME                                      ;
354B  17B6 R 3559 R          2              DW       _CODE,_LINK                       ;
354F  04 55 2F 44 33         2              DB      4,'U/D3'                           ;
17B6                         2       ORG     _CODE                                      ;
17B6  80                     1              DB 80H                                     ;
17B7  14E5 R 03F1 R 038D R                  DW      EBYTE1,CAT,DOLIT,0FH,ANDD,DOLIT,0FH,ROT
      000F 0461 R 038D R
      000F 054A R
17C7  03B7 R 17D1 R                         DW      QBRAN,UD3A
17CB  142B R                                DW      INCR
17CD  03CC R 17D3 R                         DW      BRAN,UD3B
17D1  1444 R                   UD3A:        DW      DECR
17D3  17D7 R 0394 R            UD3B:        DW      CFLD3,EXIT

                             ;  CFLD3        ( chnl --- )
                             ;               Change midi channel in field 3.
                                            $COLON  5,'CFLD3',CFLD3
17D7                         2       CFLD3:                                             ;
3541                         2       ORG     _NAME                                      ;
3541  17D7 R 354F R          2              DW       _CODE,_LINK                       ;
3545  05 43 46 4C 44 33      2              DB      5,'CFLD3'                           ;
17D7                         2       ORG     _CODE                                      ;
17D7  80                     1              DB 80H                                     ;
17D8  043B R 14E5 R 03F1 R                  DW      DUPP,EBYTE1,CAT,DOLIT,0F0H
      038D R 00F0
17E2  0461 R 046A R 14E5 R                  DW      ANDD,ORR,EBYTE1,CSTOR,FLD3,NDISP,EXIT
      03E9 R 1516 R 13D8 R
      0394 R

                             ;  U/D4         ( i/d --- )
                             ;               Field increment/decrement routine.
                                            $COLON  4,'U/D4',UD4
17F0                         2       UD4:                                              ;
3537                         2       ORG     _NAME                                      ;
3537  17F0 R 3545 R          2              DW       _CODE,_LINK                       ;
353B  04 55 2F 44 34         2              DB      4,'U/D4'                           ;
17F0                         2       ORG     _CODE                                      ;
17F0  80                     1              DB 80H                                     ;
17F1  14E5 R 03F1 R 038D R                  DW      EBYTE1,CAT,DOLIT,70H,ANDD
      0070 0461 R
17FB  0645 R 0645 R 0645 R                  DW      TWOSR,TWOSR,TWOSR,TWOSR,DOLIT,7,ROT
      0645 R 038D R 0007
      054A R
1809  03B7 R 1813 R                         DW      QBRAN,UD4A
180D  142B R                                DW      INCR
180F  03CC R 1815 R                         DW      BRAN,UD4B
1813  1444 R                   UD4A:        DW      DECR
```

```
1815  1819 R 0394 R            UD4B:        DW      CFLD4,EXIT

                            ;  CFLD4        ( status --- )
                            ;               Change Midi operation label in field 4.
                                           $COLON  5,'CFLD4',CFLD4
1819                        2       CFLD4:                                          ;
352D                        2       ORG     _NAME                                   ;
352D  1819 R 353B R         2              DW       _CODE,_LINK                    ;
3531  05 43 46 4C 44 34     2              DB      5,'CFLD4'                         ;
1819                        2       ORG     _CODE                                   ;
1819  80                    1              DB 80H                                  ;
181A  043B R 063D R 063D R                 DW      DUPP,TWOSL,TWOSL,TWOSL,TWOSL
      063D R 063D R
1824  038D R 0080 046A R                   DW      DOLIT,80H,ORR,EBYTE1,CAT
      14E5 R 03F1 R
182E  038D R 000F 0461 R                   DW      DOLIT,0FH,ANDD,ORR,EBYTE1,CSTOR
      046A R 14E5 R 03E9 R
183A  15F7 R 151D R 1403 R                 DW      LSTAT,FLD4,DISP,FLD4,LI,EXIT
      151D R 135A R 0394 R

                            ;  U/D5         ( i/d --- )
                            ;               Field increment/decrement routine.
                                           $COLON  4,'U/D5',UD5
1846                        2       UD5:                                          ;
3523                        2       ORG     _NAME                                   ;
3523  1846 R 3531 R         2              DW       _CODE,_LINK                    ;
3527  04 55 2F 44 35        2              DB      4,'U/D5'                        ;
1846                        2       ORG     _CODE                                   ;
1846  80                    1              DB 80H                                  ;
1847  038D R 00CF 14E5 R                   DW      DOLIT,0CFH,EBYTE1,CAT,DOLIT,0F0H,ANDD,LESS
      03F1 R 038D R 00F0
      0461 R 05E1 R
1857  03B7 R 186B R                        DW      QBRAN,UD5A
185B  15EC R 1524 R 1403 R                 DW      L50,FLD5,DISP,FLD5,LI,DROP
      1524 R 135A R 0436 R
1867  03CC R 1875 R                        DW      BRAN,UD5B
186B  043B R 14EC R 03E9 R     UD5A:        DW      DUPP,EBYTE2,CSTOR,FLD5,NDISP
      1524 R 13D8 R
1875  0394 R                   UD5B:        DW      EXIT




                            ;
                            ;===============================================================

= 3527                              LASTN           EQU     _NAME+4                 ;last name address

= 3523                              NTOPP           EQU     _NAME-0      ;next available memory in ROM name
dictionary
= 1877                              CTOPP           EQU     $+0          ;next available memory in ROM code
dictionary
=                           ROMSPC          EQU     NTOPP-CTOPP  ;UNUSED DICTIONARY ROM SPACE

1877                        MAIN    ENDS
                            END     ORIG
```

```
Macros:

               N a m e              Lines

$CODE  . . . . . . . . . . . . .       9
$COLON . . . . . . . . . . . . .       2
$NEXT  . . . . . . . . . . . . .       2
$USER  . . . . . . . . . . . . .       4
D$ . . . . . . . . . . . . . . .       7
SD$  . . . . . . . . . . . . . .       8

Segments and Groups:

                N a m e             Length   Align  Combine Class

MAIN . . . . . . . . . . . . . .    3FFC    PARA    NONE

Symbols:

                N a m e             Type     Value   Attr

ABOR1  . . . . . . . . . . . . .    L NEAR  0DE8    MAIN
ABORQ  . . . . . . . . . . . . .    L NEAR  0DDF    MAIN
ABORT  . . . . . . . . . . . . .    L NEAR  0DDA    MAIN
ABRTQ  . . . . . . . . . . . . .    L NEAR  0FFA    MAIN
ABS1 . . . . . . . . . . . . . .    L NEAR  05B7    MAIN
ABSS . . . . . . . . . . . . . .    L NEAR  05AC    MAIN
ACCEP  . . . . . . . . . . . . .    L NEAR  0D34    MAIN
ACCP1  . . . . . . . . . . . . .    L NEAR  0D3B    MAIN
ACCP2  . . . . . . . . . . . . .    L NEAR  0D59    MAIN
ACCP3  . . . . . . . . . . . . .    L NEAR  0D5D    MAIN
ACCP4  . . . . . . . . . . . . .    L NEAR  0D61    MAIN
ADCINIT  . . . . . . . . . . . .    L NEAR  1318    MAIN
AFT  . . . . . . . . . . . . . .    L NEAR  0FDF    MAIN
AGAIN  . . . . . . . . . . . . .    L NEAR  0FA4    MAIN
AHEAD  . . . . . . . . . . . . .    L NEAR  0FBC    MAIN
ALLOT  . . . . . . . . . . . . .    L NEAR  0F24    MAIN
ANDD . . . . . . . . . . . . . .    L NEAR  0461    MAIN
AT . . . . . . . . . . . . . . .    L NEAR  03DE    MAIN
ATEXE  . . . . . . . . . . . . .    L NEAR  07F6    MAIN

BACK1  . . . . . . . . . . . . .    L NEAR  0CF8    MAIN
BASE . . . . . . . . . . . . . .    L NEAR  04CB    MAIN
BASEE  . . . . . . . . . . . . .    NUMBER  000A
BCOMP  . . . . . . . . . . . . .    L NEAR  0F49    MAIN
BDIGS  . . . . . . . . . . . . .    L NEAR  08BB    MAIN
BDOWN  . . . . . . . . . . . . .    L NEAR  1707    MAIN
BEGIN  . . . . . . . . . . . . .    L NEAR  0F8D    MAIN
BITIME . . . . . . . . . . . . .    L NEAR  052A    MAIN
BKSLA  . . . . . . . . . . . . .    L NEAR  0BAD    MAIN
BKSP . . . . . . . . . . . . . .    L NEAR  0CCF    MAIN
BKSPP  . . . . . . . . . . . . .    NUMBER  0008
BLANK  . . . . . . . . . . . . .    L NEAR  0767    MAIN
BLEFT  . . . . . . . . . . . . .    L NEAR  1696    MAIN
BLOAD  . . . . . . . . . . . . .    L NEAR  16C4    MAIN
BLR  . . . . . . . . . . . . . .    L NEAR  167D    MAIN
```

```
BMIDI  . . . . . . . . . . . . .       L NEAR  16CF    MAIN
BRAN . . . . . . . . . . . . . .       L NEAR  03CC    MAIN
BRAN1  . . . . . . . . . . . . .       L NEAR  03C5    MAIN
BRIGHT . . . . . . . . . . . . .       L NEAR  16AD    MAIN
BUP  . . . . . . . . . . . . . .       L NEAR  16DE    MAIN
BYE  . . . . . . . . . . . . . .       L NEAR  032B    MAIN

CALLC  . . . . . . . . . . . . .       L NEAR  10E2    MAIN
CALLL  . . . . . . . . . . . . .       NUMBER  0080
CASE . . . . . . . . . . . . . .       L NEAR  141E    MAIN
CAT  . . . . . . . . . . . . . .       L NEAR  03F1    MAIN
CATCH  . . . . . . . . . . . . .       L NEAR  0D91    MAIN
CCOLD  . . . . . . . . . . . . .       NEAR    0300    MAIN
CCOM1  . . . . . . . . . . . . .       L NEAR  10AD    MAIN
CCOM2  . . . . . . . . . . . . .       L NEAR  10B1    MAIN
CCOM3  . . . . . . . . . . . . .       L NEAR  10BD    MAIN
CCOMMA . . . . . . . . . . . . .       L NEAR  0F3A    MAIN
CCOMP  . . . . . . . . . . . . .       L NEAR  1094    MAIN
CELLL  . . . . . . . . . . . . .       NUMBER  0002
CFLD1  . . . . . . . . . . . . .       L NEAR  1762    MAIN
CFLD3  . . . . . . . . . . . . .       L NEAR  17D7    MAIN
CFLD4  . . . . . . . . . . . . .       L NEAR  1819    MAIN
CHAR . . . . . . . . . . . . . .       L NEAR  0BB8    MAIN
CHAR1  . . . . . . . . . . . . .       L NEAR  0A43    MAIN
CHAR2  . . . . . . . . . . . . .       L NEAR  0A45    MAIN
CMOV1  . . . . . . . . . . . . .       L NEAR  080A    MAIN
CMOV2  . . . . . . . . . . . . .       L NEAR  081A    MAIN
CMOVE  . . . . . . . . . . . . .       L NEAR  0803    MAIN
CNTXT  . . . . . . . . . . . . .       L NEAR  04FD    MAIN
CODE . . . . . . . . . . . . . .       L NEAR  1136    MAIN
CODEE  . . . . . . . . . . . . .       NUMBER  0300
COLD . . . . . . . . . . . . . .       L NEAR  0300    MAIN
COLD1  . . . . . . . . . . . . .       L NEAR  0301    MAIN
COLDD  . . . . . . . . . . . . .       NUMBER  0100
COLON  . . . . . . . . . . . . .       L NEAR  10EB    MAIN
COMMA  . . . . . . . . . . . . .       L NEAR  0F2B    MAIN
COMPI  . . . . . . . . . . . . .       L NEAR  0F50    MAIN
COMPO  . . . . . . . . . . . . .       NUMBER  0040
CONSO  . . . . . . . . . . . . .       L NEAR  0EBF    MAIN
COUNT  . . . . . . . . . . . . .       L NEAR  07D4    MAIN
CP . . . . . . . . . . . . . . .       L NEAR  0511    MAIN
CR . . . . . . . . . . . . . . .       L NEAR  0A62    MAIN
CREAT  . . . . . . . . . . . . .       L NEAR  111C    MAIN
CRR  . . . . . . . . . . . . . .       NUMBER  000D
CRRNT  . . . . . . . . . . . . .       L NEAR  0502    MAIN
CSP  . . . . . . . . . . . . . .       L NEAR  04E4    MAIN
CSTOR  . . . . . . . . . . . . .       L NEAR  03E9    MAIN
CTOP . . . . . . . . . . . . . .       NUMBER  C390
CTOPP  . . . . . . . . . . . . .       NEAR    1877    MAIN

DAT  . . . . . . . . . . . . . .       L NEAR  07C7    MAIN
DDROP  . . . . . . . . . . . . .       L NEAR  0555    MAIN
DDUP . . . . . . . . . . . . . .       L NEAR  055B    MAIN
DECIM  . . . . . . . . . . . . .       L NEAR  092B    MAIN
DECR . . . . . . . . . . . . . .       L NEAR  1444    MAIN
DECR1  . . . . . . . . . . . . .       L NEAR  1457    MAIN
```

```
DECR2  . . . . . . . . . . . . .      L NEAR  145B    MAIN
DELAY  . . . . . . . . . . . . .      L NEAR  1339    MAIN
DEPTH  . . . . . . . . . . . . .      L NEAR  078B    MAIN
DGTQ1  . . . . . . . . . . . . .      L NEAR  095B    MAIN
DIG  . . . . . . . . . . . . . .      L NEAR  08D5    MAIN
DIGIT  . . . . . . . . . . . . .      L NEAR  0893    MAIN
DIGS . . . . . . . . . . . . . .      L NEAR  08E0    MAIN
DIGS1  . . . . . . . . . . . . .      L NEAR  08E1    MAIN
DIGS2  . . . . . . . . . . . . .      L NEAR  08ED    MAIN
DIGTQ  . . . . . . . . . . . . .      L NEAR  0936    MAIN
DISP . . . . . . . . . . . . . .      L NEAR  1403    MAIN
DISP1  . . . . . . . . . . . . .      L NEAR  140E    MAIN
DMP  . . . . . . . . . . . . . .      L NEAR  117D    MAIN
DNEGA  . . . . . . . . . . . . .      L NEAR  0588    MAIN
DOLIT  . . . . . . . . . . . . .      L NEAR  038D    MAIN
DONXT  . . . . . . . . . . . . .      L NEAR  039D    MAIN
DOSTR  . . . . . . . . . . . . .      L NEAR  0A71    MAIN
DOT  . . . . . . . . . . . . . .      L NEAR  0AC5    MAIN
DOT1 . . . . . . . . . . . . . .      L NEAR  0AD8    MAIN
DOTI1  . . . . . . . . . . . . .      L NEAR  1274    MAIN
DOTID  . . . . . . . . . . . . .      L NEAR  1261    MAIN
DOTO1  . . . . . . . . . . . . .      L NEAR  0E3F    MAIN
DOTOK  . . . . . . . . . . . . .      L NEAR  0E2A    MAIN
DOTPR  . . . . . . . . . . . . .      L NEAR  0B97    MAIN
DOTQ . . . . . . . . . . . . . .      L NEAR  100C    MAIN
DOTQP  . . . . . . . . . . . . .      L NEAR  0A89    MAIN
DOTR . . . . . . . . . . . . . .      L NEAR  0A92    MAIN
DOTS . . . . . . . . . . . . . .      L NEAR  11E3    MAIN
DOTS1  . . . . . . . . . . . . .      L NEAR  11EE    MAIN
DOTS2  . . . . . . . . . . . . .      L NEAR  11F4    MAIN
DOUSE  . . . . . . . . . . . . .      L NEAR  0496    MAIN
DOVAR  . . . . . . . . . . . . .      L NEAR  048C    MAIN
DOVOC  . . . . . . . . . . . . .      L NEAR  052F    MAIN
DROP . . . . . . . . . . . . . .      L NEAR  0436    MAIN
DSTOR  . . . . . . . . . . . . .      L NEAR  07BA    MAIN
DTRA1  . . . . . . . . . . . . .      L NEAR  0842    MAIN
DTRA2  . . . . . . . . . . . . .      L NEAR  0858    MAIN
DTRAI  . . . . . . . . . . . . .      L NEAR  083B    MAIN
DUMP . . . . . . . . . . . . . .      L NEAR  11A0    MAIN
DUMP1  . . . . . . . . . . . . .      L NEAR  11B1    MAIN
DUMP2  . . . . . . . . . . . . .      L NEAR  11D5    MAIN
DUMP3  . . . . . . . . . . . . .      L NEAR  11D9    MAIN
DUPP . . . . . . . . . . . . . .      L NEAR  043B    MAIN

EBYTE1 . . . . . . . . . . . . .      L NEAR  14E5    MAIN
EBYTE2 . . . . . . . . . . . . .      L NEAR  14EC    MAIN
EBYTE3 . . . . . . . . . . . . .      L NEAR  14F3    MAIN
EDIGS  . . . . . . . . . . . . .      L NEAR  08FE    MAIN
EDISP  . . . . . . . . . . . . .      L NEAR  160C    MAIN
EDISP1 . . . . . . . . . . . . .      L NEAR  1633    MAIN
EDISP2 . . . . . . . . . . . . .      L NEAR  1639    MAIN
EFLAG  . . . . . . . . . . . . .      L NEAR  14FA    MAIN
EFLD . . . . . . . . . . . . . .      L NEAR  14DE    MAIN
ELOAD  . . . . . . . . . . . . .      L NEAR  14A7    MAIN
ELSEE  . . . . . . . . . . . . .      L NEAR  0FEA    MAIN
EMIT . . . . . . . . . . . . . .      L NEAR  0A0A    MAIN
```

```
ENDCD  . . . . . . . . . . . . .      L NEAR  1145    MAIN
EQUAL  . . . . . . . . . . . . .      L NEAR  05B9    MAIN
ERR  . . . . . . . . . . . . . .      NUMBER  001B
ESLDN  . . . . . . . . . . . . .      L NEAR  14D7    MAIN
ESUPDAT  . . . . . . . . . . . .      L NEAR  14C9    MAIN
EUPDAT . . . . . . . . . . . . .      L NEAR  148B    MAIN
EVAL . . . . . . . . . . . . . .      L NEAR  0E57    MAIN
EVAL1  . . . . . . . . . . . . .      L NEAR  0E58    MAIN
EVAL2  . . . . . . . . . . . . .      L NEAR  0E6C    MAIN
EXE1 . . . . . . . . . . . . . .      L NEAR  0801    MAIN
EXECU  . . . . . . . . . . . . .      L NEAR  039B    MAIN
EXIT . . . . . . . . . . . . . .      L NEAR  0394    MAIN
EXPEC  . . . . . . . . . . . . .      L NEAR  0D69    MAIN
EXT  . . . . . . . . . . . . . .      NUMBER  0001
EXTRC  . . . . . . . . . . . . .      L NEAR  08AC    MAIN

FHEAD  . . . . . . . . . . . . .      L NEAR  0507    MAIN
FILE . . . . . . . . . . . . . .      L NEAR  0E96    MAIN
FILL . . . . . . . . . . . . . .      L NEAR  0822    MAIN
FILL1  . . . . . . . . . . . . .      L NEAR  082D    MAIN
FILL2  . . . . . . . . . . . . .      L NEAR  0833    MAIN
FIND . . . . . . . . . . . . . .      L NEAR  0C21    MAIN
FIND1  . . . . . . . . . . . . .      L NEAR  0C3C    MAIN
FIND2  . . . . . . . . . . . . .      L NEAR  0C60    MAIN
FIND3  . . . . . . . . . . . . .      L NEAR  0C68    MAIN
FIND4  . . . . . . . . . . . . .      L NEAR  0C78    MAIN
FIND5  . . . . . . . . . . . . .      L NEAR  0C84    MAIN
FIND6  . . . . . . . . . . . . .      L NEAR  0C6C    MAIN
FLD0 . . . . . . . . . . . . . .      L NEAR  1501    MAIN
FLD01  . . . . . . . . . . . . .      L NEAR  1508    MAIN
FLD2 . . . . . . . . . . . . . .      L NEAR  150F    MAIN
FLD3 . . . . . . . . . . . . . .      L NEAR  1516    MAIN
FLD4 . . . . . . . . . . . . . .      L NEAR  151D    MAIN
FLD5 . . . . . . . . . . . . . .      L NEAR  1524    MAIN
FLD6 . . . . . . . . . . . . . .      L NEAR  152B    MAIN
FLINK  . . . . . . . . . . . . .      L NEAR  050C    MAIN
FOR  . . . . . . . . . . . . . .      L NEAR  0F84    MAIN
FORTH  . . . . . . . . . . . . .      L NEAR  0538    MAIN

HAFBIT . . . . . . . . . . . . .      L NEAR  0525    MAIN
HAND . . . . . . . . . . . . . .      L NEAR  0EA7    MAIN
HANDL  . . . . . . . . . . . . .      L NEAR  04F8    MAIN
HERE . . . . . . . . . . . . . .      L NEAR  07DF    MAIN
HEX  . . . . . . . . . . . . . .      L NEAR  0920    MAIN
HI . . . . . . . . . . . . . . .      L NEAR  12AA    MAIN
HLD  . . . . . . . . . . . . . .      L NEAR  04F3    MAIN
HOLD . . . . . . . . . . . . . .      L NEAR  08C4    MAIN

IFF  . . . . . . . . . . . . . .      L NEAR  0FAD    MAIN
IMEDD  . . . . . . . . . . . . .      NUMBER  0080
IMMED  . . . . . . . . . . . . .      L NEAR  10F6    MAIN
INCR . . . . . . . . . . . . . .      L NEAR  142B    MAIN
INCR1  . . . . . . . . . . . . .      L NEAR  1440    MAIN
INCR2  . . . . . . . . . . . . .      L NEAR  1442    MAIN
INN  . . . . . . . . . . . . . .      L NEAR  04DA    MAIN
INTE1  . . . . . . . . . . . . .      L NEAR  0E13    MAIN
```

```
INTE2  . . . . . . . . . . . . .       L NEAR  0E1D   MAIN
INTER  . . . . . . . . . . . . .       L NEAR  0DEE   MAIN
INVER  . . . . . . . . . . . . .       L NEAR  056F   MAIN
ISLO . . . . . . . . . . . . . .       L NEAR  0EB8   MAIN

KEY  . . . . . . . . . . . . . .       L NEAR  0A01   MAIN
KEY1 . . . . . . . . . . . . . .       L NEAR  0A02   MAIN
KTAP . . . . . . . . . . . . . .       L NEAR  0D09   MAIN
KTAP1  . . . . . . . . . . . . .       L NEAR  0D26   MAIN
KTAP2  . . . . . . . . . . . . .       L NEAR  0D2A   MAIN

L0 . . . . . . . . . . . . . . .       L NEAR  1532   MAIN
L1 . . . . . . . . . . . . . . .       L NEAR  1540   MAIN
L2 . . . . . . . . . . . . . . .       L NEAR  154E   MAIN
L20  . . . . . . . . . . . . . .       L NEAR  1566   MAIN
L21  . . . . . . . . . . . . . .       L NEAR  1571   MAIN
L40  . . . . . . . . . . . . . .       L NEAR  157C   MAIN
L41  . . . . . . . . . . . . . .       L NEAR  158C   MAIN
L42  . . . . . . . . . . . . . .       L NEAR  159C   MAIN
L43  . . . . . . . . . . . . . .       L NEAR  15AC   MAIN
L44  . . . . . . . . . . . . . .       L NEAR  15BC   MAIN
L45  . . . . . . . . . . . . . .       L NEAR  15CC   MAIN
L4X  . . . . . . . . . . . . . .       L NEAR  15DC   MAIN
L50  . . . . . . . . . . . . . .       L NEAR  15EC   MAIN
LAST . . . . . . . . . . . . . .       L NEAR  051B   MAIN
LASTN  . . . . . . . . . . . . .       NUMBER  3527
LBRAC  . . . . . . . . . . . . .       L NEAR  0E1F   MAIN
LCD  . . . . . . . . . . . . . .       L NEAR  1342   MAIN
LCDINIT  . . . . . . . . . . . .       L NEAR  1363   MAIN
LEDB . . . . . . . . . . . . . .       L NEAR  147D   MAIN
LESS . . . . . . . . . . . . . .       L NEAR  05E1   MAIN
LESS1  . . . . . . . . . . . . .       L NEAR  05F2   MAIN
LF . . . . . . . . . . . . . . .       NUMBER  000A
LI . . . . . . . . . . . . . . .       L NEAR  135A   MAIN
LITER  . . . . . . . . . . . . .       L NEAR  0F5F   MAIN
LLC  . . . . . . . . . . . . . .       L NEAR  1353   MAIN
LLI  . . . . . . . . . . . . . .       L NEAR  134C   MAIN
LSTAT  . . . . . . . . . . . . .       L NEAR  15F7   MAIN

MASKK  . . . . . . . . . . . . .       NUMBER  7F1F
MAX  . . . . . . . . . . . . . .       L NEAR  05F8   MAIN
MIN  . . . . . . . . . . . . . .       L NEAR  0604   MAIN
MMOD1  . . . . . . . . . . . . .       L NEAR  06BF   MAIN
MMOD2  . . . . . . . . . . . . .       L NEAR  06CD   MAIN
MMOD3  . . . . . . . . . . . . .       L NEAR  06DD   MAIN
MODD . . . . . . . . . . . . . .       L NEAR  06EA   MAIN
MSMOD  . . . . . . . . . . . . .       L NEAR  06AA   MAIN
MSTA1  . . . . . . . . . . . . .       L NEAR  0751   MAIN
MSTAR  . . . . . . . . . . . . .       L NEAR  0738   MAIN

NAMEE  . . . . . . . . . . . . .       NUMBER  3FFD
NAMEQ  . . . . . . . . . . . . .       L NEAR  0C96   MAIN
NAMET  . . . . . . . . . . . . .       L NEAR  0BE5   MAIN
NAMQ1  . . . . . . . . . . . . .       L NEAR  0CA5   MAIN
NAMQ2  . . . . . . . . . . . . .       L NEAR  0CA7   MAIN
NAMQ3  . . . . . . . . . . . . .       L NEAR  0CC5   MAIN
```

```
NDISP  . . . . . . . . . . . . .    L NEAR  13D8   MAIN
NEGAT  . . . . . . . . . . . . .    L NEAR  057B   MAIN
NEXT . . . . . . . . . . . . . .    L NEAR  0F92   MAIN
NEXT1  . . . . . . . . . . . . .    L NEAR  03AF   MAIN
NP . . . . . . . . . . . . . . .    L NEAR  0516   MAIN
NTIB . . . . . . . . . . . . . .    L NEAR  04DF   MAIN
NTOP . . . . . . . . . . . . . .    NUMBER  C7FF
NTOPP  . . . . . . . . . . . . .    NUMBER  3523
NUFQ . . . . . . . . . . . . . .    L NEAR  0A11   MAIN
NUFQ1  . . . . . . . . . . . . .    L NEAR  0A24   MAIN
NULLS  . . . . . . . . . . . . .    L NEAR  0DCF   MAIN
NUMBQ  . . . . . . . . . . . . .    L NEAR  0963   MAIN
NUMQ1  . . . . . . . . . . . . .    L NEAR  098A   MAIN
NUMQ2  . . . . . . . . . . . . .    L NEAR  09AC   MAIN
NUMQ3  . . . . . . . . . . . . .    L NEAR  09DA   MAIN
NUMQ4  . . . . . . . . . . . . .    L NEAR  09E0   MAIN
NUMQ5  . . . . . . . . . . . . .    L NEAR  09EC   MAIN
NUMQ6  . . . . . . . . . . . . .    L NEAR  09EE   MAIN

ONEM . . . . . . . . . . . . . .    L NEAR  0626   MAIN
ONEP . . . . . . . . . . . . . .    L NEAR  061F   MAIN
ORIG . . . . . . . . . . . . . .    L NEAR  0000   MAIN
ORR  . . . . . . . . . . . . . .    L NEAR  046A   MAIN
OVER . . . . . . . . . . . . . .    L NEAR  044A   MAIN
OVERT  . . . . . . . . . . . . .    L NEAR  10BF   MAIN

PACE . . . . . . . . . . . . . .    L NEAR  0A26   MAIN
PACKS  . . . . . . . . . . . . .    L NEAR  0862   MAIN
PAD  . . . . . . . . . . . . . .    L NEAR  07E6   MAIN
PADD . . . . . . . . . . . . . .    NUMBER  C300
PAREN  . . . . . . . . . . . . .    L NEAR  0BA2   MAIN
PARS . . . . . . . . . . . . . .    L NEAR  0AE7   MAIN
PARS1  . . . . . . . . . . . . .    L NEAR  0B06   MAIN
PARS2  . . . . . . . . . . . . .    L NEAR  0B28   MAIN
PARS3  . . . . . . . . . . . . .    L NEAR  0B2A   MAIN
PARS4  . . . . . . . . . . . . .    L NEAR  0B30   MAIN
PARS5  . . . . . . . . . . . . .    L NEAR  0B48   MAIN
PARS6  . . . . . . . . . . . . .    L NEAR  0B5A   MAIN
PARS7  . . . . . . . . . . . . .    L NEAR  0B64   MAIN
PARS8  . . . . . . . . . . . . .    L NEAR  0B70   MAIN
PARSE  . . . . . . . . . . . . .    L NEAR  0B78   MAIN
PDUM1  . . . . . . . . . . . . .    L NEAR  118E   MAIN
PDUM2  . . . . . . . . . . . . .    L NEAR  119A   MAIN
PICK . . . . . . . . . . . . . .    L NEAR  079E   MAIN
PLUS . . . . . . . . . . . . . .    L NEAR  0565   MAIN
PNAM1  . . . . . . . . . . . . .    L NEAR  105F   MAIN
PRESE  . . . . . . . . . . . . .    L NEAR  0E74   MAIN
PSTOR  . . . . . . . . . . . . .    L NEAR  07AB   MAIN

QBRAN  . . . . . . . . . . . . .    L NEAR  03B7   MAIN
QCSP . . . . . . . . . . . . . .    L NEAR  120A   MAIN
QDUP . . . . . . . . . . . . . .    L NEAR  053D   MAIN
QKEY . . . . . . . . . . . . . .    L NEAR  09FA   MAIN
QRX  . . . . . . . . . . . . . .    L NEAR  032E   MAIN
QSTAC  . . . . . . . . . . . . .    L NEAR  0E43   MAIN
QUERY  . . . . . . . . . . . . .    L NEAR  0D76   MAIN
```

```
QUEST  . . . . . . . . . . . . .      L NEAR  0AE0    MAIN
QUIT . . . . . . . . . . . . . .      L NEAR  0ECC    MAIN
QUIT1  . . . . . . . . . . . . .      L NEAR  0ED3    MAIN
QUIT2  . . . . . . . . . . . . .      L NEAR  0ED5    MAIN
QUIT3  . . . . . . . . . . . . .      L NEAR  0F01    MAIN
QUIT4  . . . . . . . . . . . . .      L NEAR  0F11    MAIN

RAT  . . . . . . . . . . . . . .      L NEAR  040C    MAIN
RBRAC  . . . . . . . . . . . . .      L NEAR  10D7    MAIN
RECUR  . . . . . . . . . . . . .      L NEAR  0F79    MAIN
REPEA  . . . . . . . . . . . . .      L NEAR  0FCB    MAIN
RFROM  . . . . . . . . . . . . .      L NEAR  0405    MAIN
ROMSPC . . . . . . . . . . . . .      TEXT  NTOPP-CTOPP
ROT  . . . . . . . . . . . . . .      L NEAR  054A    MAIN
RPAT . . . . . . . . . . . . . .      L NEAR  03FB    MAIN
RPP  . . . . . . . . . . . . . .      NUMBER  C2F0
RPSTO  . . . . . . . . . . . . .      L NEAR  0400    MAIN
RZERO  . . . . . . . . . . . . .      L NEAR  04A8    MAIN

SAME1  . . . . . . . . . . . . .      L NEAR  0BF5    MAIN
SAME2  . . . . . . . . . . . . .      L NEAR  0C17    MAIN
SAMEQ  . . . . . . . . . . . . .      L NEAR  0BEE    MAIN
SAT  . . . . . . . . . . . . . .      L NEAR  1467    MAIN
SCOM1  . . . . . . . . . . . . .      L NEAR  1082    MAIN
SCOM2  . . . . . . . . . . . . .      L NEAR  1086    MAIN
SCOM3  . . . . . . . . . . . . .      L NEAR  1092    MAIN
SCOMP  . . . . . . . . . . . . .      L NEAR  1069    MAIN
SEE  . . . . . . . . . . . . . .      L NEAR  12E1    MAIN
SEE1 . . . . . . . . . . . . . .      L NEAR  12E8    MAIN
SEE2 . . . . . . . . . . . . . .      L NEAR  12FA    MAIN
SEE3 . . . . . . . . . . . . . .      L NEAR  1306    MAIN
SEE4 . . . . . . . . . . . . . .      L NEAR  130C    MAIN
SEMIS  . . . . . . . . . . . . .      L NEAR  10CC    MAIN
SERIN  . . . . . . . . . . . . .      L NEAR  0520    MAIN
SIGN . . . . . . . . . . . . . .      L NEAR  08EF    MAIN
SIGN1  . . . . . . . . . . . . .      L NEAR  08FC    MAIN
SLASH  . . . . . . . . . . . . .      L NEAR  06F1    MAIN
SLMOD  . . . . . . . . . . . . .      L NEAR  06DF    MAIN
SNAME  . . . . . . . . . . . . .      L NEAR  1032    MAIN
SPACE  . . . . . . . . . . . . .      L NEAR  0A2F    MAIN
SPACS  . . . . . . . . . . . . .      L NEAR  0A36    MAIN
SPAN . . . . . . . . . . . . . .      L NEAR  04D5    MAIN
SPAT . . . . . . . . . . . . . .      L NEAR  041C    MAIN
SPP  . . . . . . . . . . . . . .      NUMBER  C1F0
SPSTO  . . . . . . . . . . . . .      L NEAR  0429    MAIN
SSMOD  . . . . . . . . . . . . .      L NEAR  0753    MAIN
STAR . . . . . . . . . . . . . .      L NEAR  0731    MAIN
STASL  . . . . . . . . . . . . .      L NEAR  075E    MAIN
STCSP  . . . . . . . . . . . . .      L NEAR  1201    MAIN
STOIO  . . . . . . . . . . . . .      L NEAR  0380    MAIN
STORE  . . . . . . . . . . . . .      L NEAR  03D3    MAIN
STR  . . . . . . . . . . . . . .      L NEAR  090D    MAIN
STRCQ  . . . . . . . . . . . . .      L NEAR  0F68    MAIN
STRQ . . . . . . . . . . . . . .      L NEAR  1003    MAIN
STRQP  . . . . . . . . . . . . .      L NEAR  0A84    MAIN
SUBB . . . . . . . . . . . . . .      L NEAR  059B    MAIN
```

```
SWAP . . . . . . . . . . . . . .        L NEAR  0442    MAIN
SWAT . . . . . . . . . . . . . .        L NEAR  145D    MAIN
SZERO  . . . . . . . . . . . . .        L NEAR  04A3    MAIN

TAP  . . . . . . . . . . . . . .        L NEAR  0CFA    MAIN
TBOOT  . . . . . . . . . . . . .        L NEAR  12DC    MAIN
TCHA1  . . . . . . . . . . . . .        L NEAR  0789    MAIN
TCHAR  . . . . . . . . . . . . .        L NEAR  076E    MAIN
TECHO  . . . . . . . . . . . . .        L NEAR  04C1    MAIN
TEMIT  . . . . . . . . . . . . .        L NEAR  04B2    MAIN
TEMP . . . . . . . . . . . . . .        L NEAR  04D0    MAIN
TEVAL  . . . . . . . . . . . . .        L NEAR  04E9    MAIN
TEXPE  . . . . . . . . . . . . .        L NEAR  04B7    MAIN
THENN  . . . . . . . . . . . . .        L NEAR  0FD6    MAIN
THROW  . . . . . . . . . . . . .        L NEAR  0DB4    MAIN
TIB  . . . . . . . . . . . . . .        L NEAR  07ED    MAIN
TIBB . . . . . . . . . . . . . .        NUMBER  C200
TIC  . . . . . . . . . . . . . .        NUMBER  0027
TICK . . . . . . . . . . . . . .        L NEAR  0F17    MAIN
TICK1  . . . . . . . . . . . . .        L NEAR  0F22    MAIN
TMIDI  . . . . . . . . . . . . .        L NEAR  132E    MAIN
TNAM1  . . . . . . . . . . . . .        L NEAR  1221    MAIN
TNAM2  . . . . . . . . . . . . .        L NEAR  122D    MAIN
TNAM3  . . . . . . . . . . . . .        L NEAR  1245    MAIN
TNAM4  . . . . . . . . . . . . .        L NEAR  1259    MAIN
TNAME  . . . . . . . . . . . . .        L NEAR  121E    MAIN
TNUMB  . . . . . . . . . . . . .        L NEAR  04EE    MAIN
TOKEN  . . . . . . . . . . . . .        L NEAR  0BC3    MAIN
TOR  . . . . . . . . . . . . . .        L NEAR  0413    MAIN
TPROM  . . . . . . . . . . . . .        L NEAR  04C6    MAIN
TQKEY  . . . . . . . . . . . . .        L NEAR  04AD    MAIN
TTAP . . . . . . . . . . . . . .        L NEAR  04BC    MAIN
TWOM . . . . . . . . . . . . . .        L NEAR  0635    MAIN
TWOP . . . . . . . . . . . . . .        L NEAR  062D    MAIN
TWOSL  . . . . . . . . . . . . .        L NEAR  063D    MAIN
TWOSR  . . . . . . . . . . . . .        L NEAR  0645    MAIN
TXSTO  . . . . . . . . . . . . .        L NEAR  0347    MAIN
TYPE1  . . . . . . . . . . . . .        L NEAR  0A52    MAIN
TYPE2  . . . . . . . . . . . . .        L NEAR  0A5A    MAIN
TYPEE  . . . . . . . . . . . . .        L NEAR  0A4B    MAIN

UD0  . . . . . . . . . . . . . .        L NEAR  1730    MAIN
UD1  . . . . . . . . . . . . . .        L NEAR  1747    MAIN
UD1A . . . . . . . . . . . . . .        L NEAR  175C    MAIN
UD1B . . . . . . . . . . . . . .        L NEAR  175E    MAIN
UD2  . . . . . . . . . . . . . .        L NEAR  1779    MAIN
UD2A . . . . . . . . . . . . . .        L NEAR  17A0    MAIN
UD2B . . . . . . . . . . . . . .        L NEAR  17B0    MAIN
UD3  . . . . . . . . . . . . . .        L NEAR  17B6    MAIN
UD3A . . . . . . . . . . . . . .        L NEAR  17D1    MAIN
UD3B . . . . . . . . . . . . . .        L NEAR  17D3    MAIN
UD4  . . . . . . . . . . . . . .        L NEAR  17F0    MAIN
UD4A . . . . . . . . . . . . . .        L NEAR  1813    MAIN
UD4B . . . . . . . . . . . . . .        L NEAR  1815    MAIN
UD5  . . . . . . . . . . . . . .        L NEAR  1846    MAIN
UD5A . . . . . . . . . . . . . .        L NEAR  186B    MAIN
```

```
UD5B . . . . . . . . . . . . . .      L NEAR  1875    MAIN
UD6  . . . . . . . . . . . . . .      L NEAR  173E    MAIN
UD7  . . . . . . . . . . . . . .      L NEAR  1739    MAIN
UDOT . . . . . . . . . . . . . .      L NEAR  0AB8    MAIN
UDOTR  . . . . . . . . . . . . .      L NEAR  0AA3    MAIN
ULAST  . . . . . . . . . . . . .      L NEAR  019E    MAIN
ULES1  . . . . . . . . . . . . .      L NEAR  05DB    MAIN
ULESS  . . . . . . . . . . . . .      L NEAR  05C8    MAIN
UMM1 . . . . . . . . . . . . . .      L NEAR  065E    MAIN
UMM2 . . . . . . . . . . . . . .      L NEAR  0690    MAIN
UMM3 . . . . . . . . . . . . . .      L NEAR  0692    MAIN
UMM4 . . . . . . . . . . . . . .      L NEAR  069E    MAIN
UMMOD  . . . . . . . . . . . . .      L NEAR  064D    MAIN
UMST1  . . . . . . . . . . . . .      L NEAR  0707    MAIN
UMST2  . . . . . . . . . . . . .      L NEAR  0727    MAIN
UMSTA  . . . . . . . . . . . . .      L NEAR  06FA    MAIN
UNIQ1  . . . . . . . . . . . . .      L NEAR  102E    MAIN
UNIQU  . . . . . . . . . . . . .      L NEAR  1015    MAIN
UNTIL  . . . . . . . . . . . . .      L NEAR  0F9B    MAIN
UP . . . . . . . . . . . . . . .      L NEAR  0491    MAIN
UPLUS  . . . . . . . . . . . . .      L NEAR  047C    MAIN
UPP  . . . . . . . . . . . . . .      NUMBER  FF00
USER . . . . . . . . . . . . . .      L NEAR  110B    MAIN
UTYP1  . . . . . . . . . . . . .      L NEAR  116B    MAIN
UTYP2  . . . . . . . . . . . . .      L NEAR  1175    MAIN
UTYPE  . . . . . . . . . . . . .      L NEAR  1164    MAIN
UZERO  . . . . . . . . . . . . .      L NEAR  014E    MAIN

VARIA  . . . . . . . . . . . . .      L NEAR  112B    MAIN
VER  . . . . . . . . . . . . . .      NUMBER  0001
VERSN  . . . . . . . . . . . . .      L NEAR  12A3    MAIN
VOCSS  . . . . . . . . . . . . .      NUMBER  0006

WHILE  . . . . . . . . . . . . .      L NEAR  0FF3    MAIN
WITHI  . . . . . . . . . . . . .      L NEAR  0610    MAIN
WORDD  . . . . . . . . . . . . .      L NEAR  0BDC    MAIN
WORDS  . . . . . . . . . . . . .      L NEAR  1282    MAIN
WORS1  . . . . . . . . . . . . .      L NEAR  1289    MAIN
WORS2  . . . . . . . . . . . . .      L NEAR  12A1    MAIN

XIO  . . . . . . . . . . . . . .      L NEAR  0E87    MAIN
XORR . . . . . . . . . . . . . .      L NEAR  0473    MAIN

ZLESS  . . . . . . . . . . . . .      L NEAR  0453    MAIN

@FILENAME  . . . . . . . . . . .      TEXT  TST
_CODE  . . . . . . . . . . . . .      NEAR    1846    MAIN
_LEN . . . . . . . . . . . . . .      NUMBER  0002
_LINK  . . . . . . . . . . . . .      NEAR    3527    MAIN
_NAME  . . . . . . . . . . . . .      NUMBER  3523
_USER  . . . . . . . . . . . . .      NUMBER  0050
```

```
   2953 Source  Lines
   6417 Total   Lines
    463 Symbols

  50298 + 348662 Bytes symbol space free

      0 Warning Errors
      0 Severe  Errors
```