

Fig-FORTH 1.0 for BBC Micro (6502 Assembler)#

LST OFF

* This public domain publication
* is provided through the
* courtesy of Forth Interest
* Group P.O. Box 1105, San
* Carlos, CA 94070
* Further distribution must
* include this notice
* Last amended 2/2/87

TTL 'FIG Forth V.1.0'

SSIZE EQU 256 ; size of disk sector
NBUF EQU 2 ; no of buffers in RAM
SECTOR EQU 400 ; no of sects/drive
SECTL EQU 800 ; sector limit 2 drives
BMAG EQU \$404 ; total buffer magnitude

BOS EQU \$02 ; bottom of FORTH stack
TOS EQU \$70 ; top of FORTH stack
N EQU \$78 ; scratch workspace
IP EQU \$80 ; interpretive pointer
W EQU \$83 ; codefield pointer
UP EQU \$85 ; user area pointer
XSAVE EQU \$87 ; temp store for X reg

ORIG EQU \$1900 ; origin of FORTHs dictionary
MEM EQU \$5800 ; top of assigned memory + 1
UAREA EQU \$480 ; 128 bytes of user area
DAREA EQU \$5800 ; disk buffer area

RUBOUT EQU \$7F ; DEL

TIBX EQU \$100 ; terminal input buffer

; MOS entry points

OSBYTE EQU -12
OSWORD EQU -15
OSWRCH EQU -18
OSRDCH EQU -32
OSNEWL EQU -25
OSCLI EQU -9
OSASCI EQU -29

ORG \$1900

NOP ; these 2 locations are stamped
NOP ; on by BASIC initialisation
NOP ; adjust so that CFA does not
NOP ; cross page boundary so that
NOP ; JMP(W-1) works properly !!!

ENTER JMP COLD+2 ; cold start

```

REENTR  JMP    WARM    ; warm start

        DW    $6502    ; for 6502
        DW    $0000
        DW    NTOP    ; top word in FORTH
        DW    RUBOUT
        DW    UAREA    ; pointer to user area
        DW    TOS
        DW    $1FF    ; top of return stack
        DW    TIBX    ; terminal input buffer
        DW    $1F     ; initial WIDTH
        DW    $00     ; warning : 0=no disk
        DW    TOP     ; initial FENCE
        DW    TOP     ; initial top of dictionary
        DW    VLO     ; initial VOC-LINK pointer
        DW    0      ; fiddle for JMP W-1

* LIT *

L22     DFB     $83
        ASC    'LI'
        DFB    $D4

        DW    0      ; bottom word LFA contains 0
LIT     DW     *+2    ; CFA points to itself

        LDA    (IP),Y
        PHA
        INC    IP
        BNE    L30
        INC    IP+1
L30     LDA    (IP),Y
L31     INC    IP
        BNE    PUSH
        INC    IP+1
PUSH    DEX          ; adjust FORTH stack ptr
        DEX
PUT     STA    1,X    ; store (high) byte on FTH stack
        PLA
        STA    0,X    ; " (low) "
NEXT    LDY    #1
        LDA    (IP),Y ; fetch CFA pointed to by IP
        STA    W+1
        DEY
        LDA    (IP),Y
        STA    W
        CLC
        LDA    IP
        ADC    #2     ; bump IP
        STA    IP
        BCC    L54
        INC    IP+1
L54     JMP    W-1    ; W-1 contains JMP (aaaa)

* CLIT *

L35     DFB     $84
        ASC    'CLI'
        DFB    $D4

```

```

    DW    L22    ; LFA
CLIT   DW    *+2    ; CFA (points to itself)

    LDA    (IP),Y
    PHA
    TYA
    BEQ    L31    ; forced branch into LIT
SETUP  ASL      ; A = no of (16-bit) words to be
    STA    N-1    ; tfr ed to scratchpad
L63    LDA    0,X    ; from FTH stack
    STA    N,Y
    INX
    INY
    CPY    N-1    ; # of bytes
    BNE    L63
    LDY    #0
    RTS

```

* EXECUTE *

```

L75    DFB    $87
    ASC    'EXECUT'
    DFB    $C5

    DW    L35    ; LFA
EXEC   DW    *+2    ; CFA

    LDA    0,X    ; pokes address from top of
    STA    W    ; FTH stack into W
    LDA    1,X
    STA    W+1
    INX
    INX
    JMP    W-1    ; vector through W

```

* BRANCH *

```

L89    DFB    $86
    ASC    'BRANC'
    DFB    $C8

    DW    L75    ; LFA
BRANCH DW    *+2    ; CFA

    CLC
    LDA    (IP),Y    ; adds following (signed) 16-bit
    ADC    IP    ; value to IP, thus forcing a
    PHA    ; relative branch
    INY
    LDA    (IP),Y
    ADC    IP+1
    STA    IP+1
    PLA
    STA    IP
    JMP    NEXT+2    ; Y already = 1

```

* 0BRANCH *

```

L107  DFB  $87
      ASC  '0BRANC'
      DFB  $C8

      DW  L89   ; LFA
ZBRAN  DW   *+2  ; CFA

      INX      ; test top stack item
      INX
      LDA  $FE,X  ; if false then BRANCH
      ORA  $FF,X
      BEQ  BRANCH+2
BUMP  CLC      ; else bump IP
      LDA  IP
      ADC  #2   ; by 2
      STA  IP
      BCC  L122
      INC  IP+1
L122  JMP  NEXT

* (LOOP) *

L127  DFB  $86
      ASC  '(LOOP'
      DFB  $A9

      DW  L107  ; LFA
PLOOP  DW   L130 ; CFA

L130  STX  XSAVE
      TSX
      INC  $101,X  ; bump loop count by 1
      BNE  PL1    ; (on ret'n stack)
      INC  $102,X  ;      "
PL1   CLC
      LDA  $103,X  ; tests loop count vs loop limit
      SBC  $101,X
      LDA  $104,X
      SBC  $102,X
PL2   LDX  XSAVE
      ASL
      BCC  BRANCH+2
      PLA      ; drop loop parameters
      PLA
      PLA
      PLA
      JMP  BUMP   ; leave loop

* (+LOOP) *

L154  DFB  $87
      ASC  '(+LOOP'
      DFB  $A9   ; (there is an extra parm. on stack)
           ; (c.f. (LOOP))
      DW  L127  ; LFA
PPLOO  DW   *+2  ; CFA

      INX
      INX

```

```

STX  XSAVE
LDA  $FF,X
PHA
PHA
LDA  $FE,X
TSX
INX
INX
CLC
ADC  $101,X ; add increment to loop count
STA  $101,X
PLA  ; inc. h.
ADC  $102,X
STA  $102,X
PLA
BPL  PL1 ; full parm comp'son test if inc. +ve
CLC
LDA  $101,X ; reverse comparison
SBC  $103,X
LDA  $102,X
SBC  $104,X
JMP  PL2

```

* (DO) *

```

L185  DFB  $84 ;
      ASC  '(DO'
      DFB  $A9 ; (transfers loop parameters from)
           ; (FORTH stack to ret'n stack)
      DW  L154 ; LFA
PDO   DW  *+2 ; CFA

      LDA  3,X ; loop limit hi
      PHA
      LDA  2,X ; loop limit lo
      PHA
      LDA  1,X ; loop start hi
      PHA
      LDA  0,X ; loop start lo
      PHA
POPTWO INX ; drop FORTH stack item
      INX
POP    INX ; drop another FORTH stack item
      INX
      JMP  NEXT

```

* I *

```

L207  DFB  $81,$C9 ; 'I'

      DW  L185 ; LFA - copy loop counter to FTH stack
I     DW  R+2 ; CFA - same as 'R'

```

* DIGIT *

```

L214  DFB  $85
      ASC  'DIGI' ; converts ASCII chr to binary equiv
      DFB  $D4 ; in relevant BASE leaving num on
           ; FTH stack + tf if valid ff only

```

```

    DW    L207    ; if not valid char
DIGIT   DW     *+2

    SEC
    LDA    2,X    ; get char
    SBC    #$30   ; unprintable ?
    BMI    L234
    CMP    #$A    ; 0-9 ?
    BMI    L227
    SEC
    SBC    #7     ; A-F ?
    CMP    #$A
    BMI    L234
L227    CMP    0,X    ; compare with number base
    BPL    L234
    STA    2,X    ; number valid - stack it
    LDA    #1     ; with tf
    PHA
    TYA
    JMP    PUT    ; exit (true) char valid

L234    TYA
    PHA
    INX
    INX
    JMP    PUT    ; exit (false) char invalid

* (FIND) *

L243    DFB    $86    ; dictionary search for word
    ASC    '(FIND'    ; from NFA on top of F. stack
    DFB    $A9    ; which matches text at addr.

    DW    L214    ; beneath it on stack
PFIND   DW     *+2    ; CFA (self)

    LDA    #2
    JSR    SETUP
    STX    XSAVE
L249    LDY    #0
    LDA    (N),Y
    EOR    (N+2),Y
    AND    #$3F
    BNE    L281
L254    INY
    LDA    (N),Y
    EOR    (N+2),Y
    ASL
    BNE    L280
    BCC    L254
    LDX    XSAVE
    DEX
    DEX
    DEX
    DEX
    CLC
    TYA
    ADC    #5
    ADC    N

```

```
STA 2,X
LDY #0
TYA
ADC N+1
STA 3,X
STY 1,X
LDA (N),Y
STA 0,X
LDA #1
PHA
JMP PUSH ; exit (true)
```

```
L280 BCS L284
L281 INY
LDA (N),Y
BPL L281
L284 INY
LDA (N),Y
TAX
INY
LDA (N),Y
STA N+1
STX N
ORA N
BNE L249
LDX XSAVE
LDA #0
PHA
JMP PUSH ; exit (false)
```

* ENCLOSE *

```
L301 DFB $87
ASC 'ENCLOS'
DFB $C5

DW L243 ; LFA
ENCL DW *+2 ; CFA
```

```
LDA #2
JSR SETUP ; copy 2 words to scratchpad
TXA
SEC
SBC #8
TAX ; bump stack ptr by 8 bytes
STY 3,X ; Y=0
STY 1,X
DEY
L313 INY
LDA (N+2),Y
CMP N
BEQ L313
STY 4,X
L318 LDA (N+2),Y
BNE L327
STY 2,X
STY 0,X
TYA
CMP 4,X
```

```
BNE L326
INC 2,X
L326 JMP NEXT
```

```
L327 STY 2,X
INY
CMP N
BNE L318
STY 0,X
JMP NEXT
```

* EMIT *

```
L337 DFB $84
ASC 'EMI'
DFB $D4
DW L301 ; LFA
```

```
EMIT DW XEMIT ; vectored
```

* KEY *

```
L344 DFB $83
ASC 'KE'
DFB $D9

DW L337 ; LFA
```

```
KEY DW XKEY ; vectored
```

* ?TERMINAL *

```
L351 DFB $89
ASC '?TERMINA'
DFB $CC
```

```
DW L344 ; LFA
QTERM DW XQTER ; vectored
```

* CR *

```
L358 DFB $82
ASC 'C'
DFB $D2
```

```
DW L351 ; LFA
CR DW XCR ; vectored
```

* CMOVE *

```
L365 DFB $85
ASC 'CMOV'
DFB $C5
```

```
DW L358 ; LFA
CMOVE DW *+2 ; CFA
```

```
LDA #3
JSR SETUP
```



```
L370   CPY   N
      BNE   L375
      DEC   N+1
      BPL   L375
      JMP   NEXT   ; finished
```

```
L375   LDA   (N+4),Y
      STA   (N+2),Y
      INY
      BNE   L370
      INC   N+5
      INC   N+3
      JMP   L370
```

```
* U* *
```

```
L386   DFB   $82
      ASC   'U'
      DFB   $AA

      DW   L365   ; LFA
USTAR   DW   *+2   ; CFA
```

```
      LDA   2,X
      STA   N
      STA   2,X
      LDA   3,X
      STA   N+1
      STY   3,X
      LDY   #16
L396   ASL   2,X
      ROL   3,X
      ROL   0,X
      ROL   1,X
      BCC   L411
      CLC
      LDA   N
      ADC   2,X
      STA   2,X
      LDA   N+1
      ADC   3,X
      STA   3,X
      LDA   #0
      ADC   0,X
      STA   0,X
L411   DEY
      BNE   L396
      JMP   NEXT
```

```
* U/ *
```

```
L418   DFB   $82
      ASC   'U'
      DFB   $AF

      DW   L386   ; LFA
USLASH   DW   *+2   ; CFA

      LDA   4,X
```

```

LDY 2,X
STY 4,X
ASL
STA 2,X
LDA 5,X
LDY 3,X
STY 5,X
ROL
STA 3,X
LDA #16
STA N
L433 ROL 4,X
ROL 5,X
SEC
LDA 4,X
SBC 0,X
TAY
LDA 5,X
SBC 1,X
BCC L444
STY 4,X
STA 5,X
L444 ROL 2,X
ROL 3,X
DEC N
BNE L433
JMP POP

```

* AND *

```

L453 DFB $83
ASC 'AN'
DFB $C4

DW L418 ; LFA
ANDD DW *+2 ; CFA

```

```

LDA 0,X
AND 2,X
PHA
LDA 1,X
AND 3,X

```

```

BINARY INX
INX
JMP PUT

```

* OR *

```

L469 DFB $82
ASC 'O'
DFB $D2

DW L453 ; LFA
OR DW *+2 ; CFA

```

```

LDA 0,X
ORA 2,X
PHA

```

```
LDA 1,X
ORA 3,X
INX
INX
JMP PUT
```

* XOR *

```
L484 DFB $83
      ASC 'XO'
      DFB $D2

      DW L469 ; LFA
XOR DW *+2 ; CFA
```

```
LDA 0,X
EOR 2,X
PHA
LDA 1,X
EOR 3,X
INX
INX
JMP PUT
```

* SP@ *

```
L499 DFB $83
      ASC 'SP'
      DFB $C0

      DW L484 ; LFA
SPAT DW *+2 ; CFA
```

```
TXA
PUSH0A PHA
LDA #0
JMP PUSH
```

* SP! *

```
L511 DFB $83
      ASC 'SP'
      DFB $A1

      DW L499 ; LFA
SPSTO DW *+2 ; CFA
```

```
LDY #6
LDA (UP),Y
CLC ; MJR
ADC #2 ; MJR
TAX
JMP NEXT
```

* RP! *

```
L522 DFB $83
      ASC 'RP'
      DFB $A1
```

DW L511 ; LFA
RPSTO DW *+2 ; CFA

STX XSAVE
LDY #8
LDA (UP),Y
TAX
TXS
LDX XSAVE
JMP NEXT

* ;S *

L536 DFB \$82
ASC ';' ;'
DFB \$D3

DW L522
SEMIS DW *+2

PLA
STA IP
PLA
STA IP+1
JMP NEXT

* LEAVE *

L548 DFB \$85
ASC 'LEAV'
DFB \$C5

DW L536
LEAVE DW *+2

STX XSAVE
TSX
LDA \$101,X
STA \$103,X
LDA \$102,X
STA \$104,X
LDX XSAVE
JMP NEXT

* >R *

L563 DFB \$82
ASC '>'
DFB \$D2

DW L548 ; LFA
TOR DW *+2 ; CFA

LDA 1,X
PHA
LDA 0,X
PHA
INX

```
INX
JMP  NEXT
```

```
* R> *
```

```
L577  DFB  $82
      ASC  'R'
      DFB  $BE

      DW  L563  ; LFA
RFROM DW  *+2  ; CFA

      DEX
      DEX
      PLA
      STA  0,X
      PLA
      STA  1,X
      JMP  NEXT
```

```
* R *
```

```
L591  DFB  $81,$D2

      DW  L577  ; LFA
R     DW  *+2  ; CFA

      STX  XSAVE  ; copy
      TSX                ; top of
      LDA  $101,X  ; m/c stack
      PHA                ; to
      LDA  $102,X  ; 4th stack
      LDX  XSAVE  ; = 'I'
      JMP  PUSH
```

```
* 0= *
```

```
L605  DFB  $82
      ASC  '0'
      DFB  $BD

      DW  L591  ; LFA
ZEQU  DW  *+2  ; CFA

      LDA  0,X
      ORA  1,X
      STY  1,X
      BNE  L613
      INY
L613  STY  0,X
      JMP  NEXT
```

```
* 0< *
```

```
L619  DFB  $82
      ASC  '0'
      DFB  $BC

      DW  L605  ; LFA
```

ZLESS DW *+2 ; CFA

ASL 1,X ; leave true
TYA ; if BOS
ROL A ; -ve else
STY 1,X ; leave false
STA 0,X
JMP NEXT

* + *

L632 DFB \$81,\$AB

DW L619 ; LFA
PLUS DW *+2 ; CFA

CLC
LDA 0,X
ADC 2,X
STA 2,X
LDA 1,X
ADC 3,X
STA 3,X
INX
INX
JMP NEXT

* D+ *

L649 DFB \$82

ASC 'D'
DFB \$AB

DW L632 ; LFA
DPLUS DW *+2 ; CFA

CLC
LDA 2,X
ADC 6,X
STA 6,X
LDA 3,X
ADC 7,X
STA 7,X
LDA 0,X
ADC 4,X
STA 4,X
LDA 1,X
ADC 5,X
STA 5,X
JMP POPTWO

* MINUS *

L670 DFB \$85

ASC 'MINU'
DFB \$D3

DW L649 ; LFA
MINUS DW *+2 ; CFA

```
SEC
TYA
SBC 0,X ; leave
STA 0,X ; 2's compliment
TYA ; of BOS
SBC 1,X
STA 1,X
JMP NEXT
```

* DMINUS *

```
L685 DFB $86
ASC 'DMINU'
DFB $D3

DW L670 ; LFA
DMINUS DW *+2 ; CFA
```

```
SEC
TYA
SBC 2,X
STA 2,X
TYA
SBC 3,X
STA 3,X
JMP MINUS+3
```

* OVER *

```
L700 DFB $84
ASC 'OVE'
DFB $D2

DW L685 ; LFA
OVER DW *+2 ; CFA
```

```
LDA 2,X
PHA
LDA 3,X
JMP PUSH
```

* DROP *

```
L711 DFB $84
ASC 'DRO'
DFB $D0

DW L700 ; LFA
DROP DW POP ; CFA
```

* SWAP *

```
L718 DFB $84
ASC 'SWA'
DFB $D0

DW L711 ; LFA
SWAP DW *+2
```

```
LDA 2,X
PHA
LDA 0,X
STA 2,X
LDA 3,X
LDY 1,X
STY 3,X
JMP PUT
```

* DUP *

```
L733 DFB $83
      ASC 'DU'
      DFB $D0

      DW L718 ; LFA
DUP   DW  *+2 ; CFA
```

```
LDA 0,X
PHA
LDA 1,X
JMP PUSH
```

* +! *

```
L744 DFB $82
      ASC '+'
      DFB $A1

      DW L733 ; LFA
PSTORE DW  *+2
```

```
CLC
LDA (0,X)
ADC 2,X
STA (0,X)
INC 0,X
BNE L754
INC 1,X
L754 LDA (0,X)
      ADC 3,X
      STA (0,X)
      JMP POPTWO
```

* TOGGLE *

```
L762 DFB $86
      ASC 'TOGGL'
      DFB $C5

      DW L744 ; LFA
TOGGLE DW  *+2 ; CFA
```

```
LDA (2,X)
EOR 0,X
STA (2,X)
JMP POPTWO
```


* @ *

L773 DFB \$81,\$C0

DW L762 ; LFA
AT DW *+2 ; CFA

LDA (0,X)
PHA
INC 0,X
BNE L781
INC 1,X
L781 LDA (0,X)
JMP PUT

* C@ *

L787 DFB \$82

ASC 'C'
DFB \$C0

DW L773 ; LFA
CAT DW *+2 ; CFA

LDA (0,X)
STA 0,X
STY 1,X
JMP NEXT

* ! *

L798 DFB \$81,\$A1

DW L787 ; LFA
STORE DW *+2 ; CFA

LDA 2,X
STA (0,X)
INC 0,X
BNE L806
INC 1,X
L806 LDA 3,X
STA (0,X)
JMP POPTWO

* C! *

L813 DFB \$82

ASC 'C'
DFB \$A1

DW L798 ; LFA
CSTORE DW *+2 ; CFA

LDA 2,X
STA (0,X)
JMP POPTWO

* : *

L823 DFB \$C1,\$BA

DW L813 ; LFA
COLON DW DOCOL ; CFA

DW QEXEC
DW SCSP
DW CURR
DW AT
DW CON
DW STORE
DW CREATE
DW RBRACK
DW PSCOD

DOCOL LDA IP+1
PHA
LDA IP
PHA
CLC
LDA W
ADC #2
STA IP
TYA
ADC W+1
STA IP+1
JMP NEXT

* ; *

L853 DFB \$C1,\$BB

DW L823 ; LFA
DW DOCOL ; CFA

DW QCSP
DW COMP
DW SEMIS
DW SMUDGE
DW LBRACK
DW SEMIS

* CONSTANT *

L867 DFB \$88
ASC 'CONSTAN'
DFB \$D4

DW L853 ; LFA
CONST DW DOCOL ; CFA

DW CREATE
DW SMUDGE
DW COMMA
DW PSCOD

DOCON LDY #2
LDA (W),Y

```
PHA
INY
LDA (W),Y
JMP PUSH
```

```
* VARIABLE *
```

```
L885 DFB $88
ASC 'VARIABL'
DFB $C5

DW L867 ; LFA
VAR DW DOCOL ; CFA
```

```
DW CONST
DW PSCOD
```

```
DOVAR CLC
LDA W
ADC #2
PHA
TYA
ADC W+1
JMP PUSH
```

```
* USER *
```

```
L902 DFB $84
ASC 'USE'
DFB $D2

DW L885 ; LFA
USER DW DOCOL ; CFA
```

```
DW CONST
DW PSCOD
```

```
DOUSE LDY #2
CLC
LDA (W),Y
ADC UP
PHA
LDA #0
ADC UP+1
JMP PUSH
```

```
* 0 *
```

```
L920 DFB $81,$B0

DW L902 ; LFA
ZERO DW DOCON ; CFA

DW 0
```

```
* 1 *
```

```
L928 DFB $81,$B1
```

DW L920 ; LFA
ONE DW DOCON ; CFA

DW 1

* 2 *

L936 DFB \$81,\$B2

DW L928 ; LFA
TWO DW DOCON ; CFA

DW 2

* 3 *

L944 DFB \$81,\$B3

DW L936 ; LFA
THREE DW DOCON ; CFA

DW 3

* BL *

L952 DFB \$82
ASC 'B'
DFB \$CC

DW L944 ; LFA
BL DW DOCON ; CFA

DW 32 ; ASCII blank

* C/L *

L960 DFB \$83
ASC 'C/'
DFB \$CC

DW L952 ; LFA
CSLL DW DOCON ; CFA

DW 64 ; 64 chars/line

DW SEMIS ; MJR - padding

* FIRST *

L968 DFB \$85
ASC 'FIRS'
DFB \$D4

DW L960 ; LFA
FIRST DW DOCON ; CFA

DW DAREA ; bottom of disk
; buffer

* LIMIT *

L976 DFB \$85
ASC 'LIMI'
DFB \$D4

DW L968 ; LFA
LIMIT DW DOCON ; CFA

DW \$5800 ; end of buffers-see Harrison

* B/BUF *

L984 DFB \$85
ASC 'B/BU'
DFB \$C6

DW L976 ; LFA
BBUF DW DOCON ; CFA

DW 256 ; sector size

* B/SCR *

L992 DFB \$85
ASC 'B/SC'
DFB \$D2

DW L984 ; LFA
BSCR DW DOCON ; CFA

DW 4 ; blocks per screen

L1000 DFB \$87
ASC '+ORIGI'
DFB \$CE

DW L992 ; LFA
PORIG DW DOCOL ; CFA

DW LIT
DW ORIG
DW PLUS
DW SEMIS

* TIB *

L1010 DFB \$83
ASC 'TI'
DFB \$C2

DW L1000 ; LFA
TIB DW DOUSE ; CFA

DFB \$A

* WIDTH *

L1018 DFB \$85

ASC 'WIDT'
DFB \$C8

DW L1010 ; LFA
WIDTH DW DOUSE ; CFA

DFB \$C

* WARNING *

L1026 DFB \$87
ASC 'WARNIN'
DFB \$C7

DW L1018 ; LFA
WARN DW DOUSE ; CFA

DFB \$E

* FENCE *

L1034 DFB \$85
ASC 'FENC'
DFB \$C5

DW L1026 ; LFA
FENCE DW DOUSE ; CFA

DFB \$10

* DP *

L1042 DFB \$82
ASC 'D'
DFB \$D0

DW L1034 ; LFA
DP DW DOUSE ; CFA

DFB \$12

* VOC-LINK *

L1050 DFB \$88
ASC 'VOC-LIN'
DFB \$CB

DW L1042 ; LFA
VOCLNK DW DOUSE ; CFA

DFB \$14

* BLK *

L1058 DFB \$83
ASC 'BL'
DFB \$CB

DW L1050 ; LFA

BLK DW DOUSE ; CFA

DFB \$16

* IN *

L1066 DFB \$82

ASC 'I'

DFB \$CE

DW L1058 ; LFA

IN DW DOUSE ; CFA

DFB \$18

* OUT *

L1074 DFB \$83

ASC 'OU'

DFB \$D4

DW L1066 ; LFA

OUT DW DOUSE ; CFA

DFB \$1A

* SCR *

L1082 DFB \$83

ASC 'SC'

DFB \$D2

DW L1074 ; LFA

SCR DW DOUSE ; CFA

DFB \$1C

* OFFSET *

L1090 DFB \$86

ASC 'OFFSE'

DFB \$D4

DW L1082 ; LFA

OFFSET DW DOUSE ; CFA

DFB \$1E

* CONTEXT *

L1098 DFB \$87

ASC 'CONTEX'

DFB \$D4

DW L1090 ; LFA

CON DW DOUSE ; CFA

DFB \$20

* CURRENT *

L1106 DFB \$87
ASC 'CURREN'
DFB \$D4

DW L1098 ; LFA
CURR DW DOUSE ; CFA

DFB \$22

* STATE *

L1114 DFB \$85
ASC 'STAT'
DFB \$C5

DW L1106 ; LFA
STATE DW DOUSE ; CFA

DFB \$24

* BASE *

L1122 DFB \$84
ASC 'BAS'
DFB \$C5

DW L1114 ; LFA
BASE DW DOUSE ; CFA

DFB \$26

* DPL *

L1130 DFB \$83
ASC 'DP'
DFB \$CC

DW L1122 ; LFA
DPL DW DOUSE ; CFA

DFB \$28

* FLD *

L1138 DFB \$83
ASC 'FL'
DFB \$C4

DW L1130 ; LFA
FLD DW DOUSE ; CFA

DFB \$2A

* CSP *

L1146 DFB \$83
ASC 'CS'

DFB \$D0

DW L1138 ; LFA
CSP DW DOUSE ; CFA

DFB \$2C

* R# *

L1154 DFB \$82
ASC 'R'
DFB \$A3

DW L1146 ; LFA
RNUM DW DOUSE ; CFA

DFB \$2E

* HLD *

L1162 DFB \$83
ASC 'HL'
DFB \$C4

DW L1154 ; LFA
HLD DW DOUSE ; CFA

DFB \$30

* 1+ *

L1170 DFB \$82
ASC '1'
DFB \$AB

DW L1162 ; LFA
ONEP DW DOCOL ; CFA

DW ONE
DW PLUS
DW SEMIS

* 2+ *

L1180 DFB \$82
ASC '2'
DFB \$AB

DW L1170 ; LFA
TWOP DW DOCOL ; CFA

DW TWO
DW PLUS
DW SEMIS

* HERE *

L1190 DFB \$84
ASC 'HER'

DFB \$C5

DW L1180 ; LFA
HERE DW DOCOL ; CFA

DW DP
DW AT
DW SEMIS

* ALLOT *

L1200 DFB \$85
ASC 'ALLO'
DFB \$D4

DW L1190 ; LFA
ALLOT DW DOCOL ; CFA

DW DP
DW PSTORE
DW SEMIS

* , *

L1210 DFB \$81,\$AC

DW L1200 ; LFA
COMMA DW DOCOL ; CFA

DW HERE
DW STORE
DW TWO
DW ALLOT
DW SEMIS

* C, *

L1222 DFB \$82
ASC 'C'
DFB \$AC

DW L1210 ; LFA
CCOMMA DW DOCOL ; CFA

DW HERE
DW CSTORE
DW ONE
DW ALLOT
DW SEMIS

* - *

L1234 DFB \$81,\$AD

DW L1222 ; LFA
SUB DW DOCOL ; CFA

DW MINUS
DW PLUS

```

    DW    SEMIS

* = *

L1244   DFB    $81,$BD

    DW    L1234    ; LFA
EQUALS  DW    DOCOL    ; CFA

    DW    SUB
    DW    ZEQU
    DW    SEMIS

* U< *

L1246   DFB    $82
    ASC   'U'
    DFB   $BC

    DW    L1244    ; LFA
ULESS   DW    DOCOL    ; CFA

    DW    SUB
    DW    ZLESS
    DW    SEMIS

* < *

L1254   DFB    $81,$BC

    DW    L1246    ; LFA
LESS    DW    *+2    ; CFA

    SEC
    LDA   2,X
    SBC   0,X
    LDA   3,X
    SBC   1,X
    STY   3,X    ; zero hi byte
    BVC   L1258
    EOR   #$80    ; correct o/flow
L1258   BPL   L1260
    INY   ; invrt flag
L1260   STY   2,X
    JMP   POP

* > *

L1264   DFB    $81,$BE

    DW    L1254    ; LFA
GREAT   DW    DOCOL    : CFA

    DW    SWAP
    DW    LESS
    DW    SEMIS

* ROT *

```

L1274 DFB \$83
ASC 'RO'
DFB \$D4

DW L1264 ; LFA
ROT DW DOCOL ; CFA

DW TOR
DW SWAP
DW RFROM
DW SWAP
DW SEMIS

* SPACE *

L1286 DFB \$85
ASC 'SPAC'
DFB \$C5

DW L1274 ; LFA
SPACE DW DOCOL

DW BL
DW EMIT
DW SEMIS

* -DUP *

L1296 DFB \$84
ASC '-DU'
DFB \$D0

DW L1286 ; LFA
DDUP DW DOCOL ; CFA

DW DUP
DW ZBRAN
DW 4
DW DUP
DW SEMIS

* TRAVERSE *

L1308 DFB \$88
ASC 'TRAVERS'
DFB \$C5

DW L1296 ; LFA
TRAV DW DOCOL ; CFA

DW SWAP
DW OVER
DW PLUS
DW CLIT
DFB \$7F
DW OVER
DW CAT
DW LESS
DW ZBRAN

DW -15
DW SWAP
DW DROP
DW SEMIS

* LATEST *

L1328 DFB \$86
ASC 'LATES'
DFB \$D4

DW L1308 ; LFA
LATEST DW DOCOL ; CFA

DW CURR
DW AT
DW AT
DW SEMIS

* LFA *

L1339 DFB \$83
ASC 'LF'
DFB \$C1

DW L1328 ; LFA
LFA DW DOCOL ; CFA

DW CLIT
DFB 4
DW SUB
DW SEMIS

* CFA *

L1350 DFB \$83
ASC 'CF'
DFB \$C1

DW L1339 ; LFA
CFA DW DOCOL ; CFA

DW TWO
DW SUB
DW SEMIS

* NFA *

L1360 DFB \$83
ASC 'NF'
DFB \$C1

DW L1350 ; LFA
NFA DW DOCOL ; CFA

DW CLIT
DFB 5
DW SUB
DW LIT

DW -1
DW TRAV
DW SEMIS

* PFA *

L1373 DFB \$83
ASC 'PF'
DFB \$C1

DW L1360 ; LFA
PFA DW DOCOL ; CFA

DW ONE
DW TRAV
DW CLIT
DFB 5
DW PLUS
DW SEMIS

* !CSP *

L1386 DFB \$84
ASC '!CS'
DFB \$D0

DW L1373 ; LFA
SCSP DW DOCOL ; CFA

DW SPAT
DW CSP
DW STORE
DW SEMIS

* ?ERROR *

L1397 DFB \$86
ASC '?ERRO'
DFB \$D2

DW L1386 ; LFA
QERROR DW DOCOL ; CFA

DW SWAP
DW ZBRAN
DW 8
DW ERROR
DW BRANCH
DW 4
DW DROP
DW SEMIS

* ?COMP *

L1412 DFB \$85
ASC '?COM'
DFB \$D0

DW L1397 ; LFA

QCOMP DW DOCOL ; CFA

DW STATE
DW AT
DW ZEQU
DW CLIT
DFB 17
DW QERROR
DW SEMIS

* ?EXEC *

L1426 DFB \$85
ASC '?EXE'
DFB \$C3

DW L1412 ; LFA
QEXEC DW DOCOL ; CFA

DW STATE
DW AT
DW CLIT
DFB 18
DW QERROR
DW SEMIS

* ?PAIRS *

L1439 DFB \$85
ASC '?PAIR'
DFB \$D3

DW L1426 ; LFA
QPAIR DW DOCOL ; CFA

DW SUB
DW CLIT
DFB 19
DW QERROR
DW SEMIS

* ?CSP *

L1451 DFB \$84
ASC '?CS'
DFB \$D0

DW L1439 ; LFA
QCSP DW DOCOL ; CFA

DW SPAT
DW CSP
DW AT
DW SUB
DW CLIT
DFB 20
DW QERROR
DW SEMIS

* ?LOADING *

L1466 DFB \$88
ASC '?LOADIN'
DFB \$C7

DW L1451 ; LFA
QLOAD DW DOCOL ; CFA

DW BLK
DW AT
DW ZEQU
DW CLIT
DFB 22
DW QERROR
DW SEMIS

* COMPILE *

L1480 DFB \$87
ASC 'COMPIL'
DFB \$C5

DW L1466 ; LFA
COMP DW DOCOL ; CFA

DW QCOMP
DW RFROM
DW DUP
DW TWOP
DW TOR
DW AT
DW COMMA
DW SEMIS

* ~[*

L1495 DFB \$81,\$DB

DW L1480 ; LFA
LBRACK DW DOCOL ; CFA

DW ZERO
DW STATE
DW STORE
DW SEMIS

*] *

L1507 DFB \$81,\$DD

DW L1495 ; LFA
RBRACK DW DOCOL ; CFA

DW CLIT
DFB \$C0
DW STATE
DW STORE
DW SEMIS

* SMUDGE *

L1519 DFB \$86
ASC 'SMUDG'
DFB \$C5

DW L1507 ; LFA
SMUDGE DW DOCOL ; CFA

DW LATEST
DW CLIT
DFB 32
DW TOGGLE
DW SEMIS

* HEX *

L1531 DFB \$83
ASC 'HE'
DFB \$D8

DW L1519 ; LFA
HEX DW DOCOL ; CFA

DW CLIT
DFB 16
DW BASE
DW STORE
DW SEMIS

* DECIMAL *

L1543 DFB \$87
ASC 'DECIMA'
DFB \$CC

DW L1531 ; LFA
DECIM DW DOCOL ; CFA

DW CLIT
DFB 10
DW BASE
DW STORE
DW SEMIS

* (;CODE) *

L1555 DFB \$87
ASC '(;COD'
DFB \$A9

DW L1543 ; LFA
PSCOD DW DOCOL ; CFA

DW RFROM
DW LATEST
DW PFA
DW CFA

DW STORE
DW SEMIS

* ;CODE *

L1568 DFB \$85
ASC ' ;COD '
DFB \$C5

DW L1555 ; LFA
DW DOCOL
DW QCSP
DW COMP
DW PSCOD
DW LBRACK
DW SMUDGE
DW SEMIS

* <BUILDS *

L1582 DFB \$87
ASC '<BUILD '
DFB \$D3

DW L1568 ; LFA
BUILD DW DOCOL ; CFA

DW ZERO
DW CONST
DW SEMIS

* DOES> *

L1592 DFB \$85
ASC 'DOES '
DFB \$BE

DW L1582 ; LFA
DOES DW DOCOL ; CFA

DW RFROM
DW LATEST
DW PFA
DW STORE
DW PSCOD

DODOE LDA IP+1
PHA
LDA IP
PHA
LDY #2
LDA (W),Y
STA IP
INY
LDA (W),Y
STA IP+1
CLC
LDA W
ADC #4

PHA
LDA W+1
ADC #0
JMP PUSH

* COUNT *

L1622 DFB \$85
ASC 'COUN'
DFB \$D4

DW L1592 ; LFA
COUNT DW DOCOL ; CFA

DW DUP
DW ONEP
DW SWAP
DW CAT
DW SEMIS

* TYPE *

L1634 DFB \$84
ASC 'TYP'
DFB \$C5

DW L1622 ; LFA
TYPE DW DOCOL ; CFA

DW DDUP
DW ZBRAN
DW 24
DW OVER
DW PLUS
DW SWAP
DW PDO
DW I
DW CAT
DW EMIT
DW PLOOP
DW -8
DW BRANCH
DW 4
DW DROP
DW SEMIS

* -TRAILING *

L1657 DFB \$89
ASC '-TRAILIN'
DFB \$C7

DW L1634 ; LFA
DTRAI DW DOCOL ; CFA

DW DUP
DW ZERO
DW PDO
DW OVER

DW OVER
DW PLUS
DW ONE
DW SUB
DW CAT
DW BL
DW SUB
DW ZBRAN
DW 8
DW LEAVE
DW BRANCH
DW 6
DW ONE
DW SUB
DW PLOOP
DW \$FFE0
DW SEMIS

* (.") *

L1685 DFB \$84
ASC '(. "'
DFB \$A9

DW L1657 ; LFA
PDOTQ DW DOCOL ; CFA

DW R
DW COUNT
DW DUP
DW ONEP
DW RFROM
DW PLUS
DW TOR
DW TYPE
DW SEMIS

* ." *

L1701 DFB \$C2
ASC '. '
DFB \$A2

DW L1685 ; LFA
DW DOCOL ; CFA

DW CLIT
DFB 34
DW STATE
DW AT
DW ZBRAN
DW 20
DW COMP
DW PDOTQ
DW WORD
DW HERE
DW CAT
DW ONEP
DW ALLOT

DW BRANCH
DW 10
DW WORD
DW HERE
DW COUNT
DW TYPE
DW SEMIS

* EXPECT *

L1729 DFB \$86
ASC 'EXPEC'
DFB \$D4

DW L1701 ; LFA
EXPECT DW DOCOL ; CFA

DW OVER
DW PLUS
DW OVER
DW PDO
DW KEY
DW DUP
DW CLIT
DFB 17 ; adjust as appropriate
DW PORIG ; rel. NOPS at ORG
DW AT
DW EQUALS
DW ZBRAN
DW 31
DW DROP
DW CLIT
DFB \$7F
DW OVER
DW I
DW EQUALS
DW DUP
DW RFROM
DW TWO
DW SUB
DW PLUS
DW TOR

* DW SUB
DW DROP ; MJR
DW BRANCH
DW 39
DW DUP
DW CLIT
DFB 13
DW EQUALS
DW ZBRAN
DW 14
DW LEAVE
DW DROP
DW BL
DW ZERO
DW BRANCH
DW 4
DW DUP

DW I
DW CSTORE
DW ZERO
DW I
DW ONEP
DW STORE
DW EMIT
DW PLOOP
DW \$FFA9
DW DROP
DW SEMIS

* QUERY *

L1788 DFB \$85
ASC 'QUER'
DFB \$D9

DW L1729 ; LFA
QUERY DW DOCOL ; CFA

DW TIB
DW AT
DW CLIT
DFB 80
DW EXPECT
DW ZERO
DW IN
DW STORE
DW SEMIS

* <ASCII NULL> *

L1804 DFB \$C1,\$80

DW L1788 ; LFA
DW DOCOL ; CFA

DW BLK
DW AT
DW ZBRAN
DW 42
DW ONE
DW BLK
DW PSTORE
DW ZERO
DW IN
DW STORE
DW BLK
DW AT
DW ZERO
DW BSCR
DW USLASH
DW DROP
DW ZEQU
DW ZBRAN
DW 8
DW QEXEC
DW RFROM

DW DROP
DW BRANCH
DW 6
DW RFROM
DW DROP
DW SEMIS

* FILL *

L1838 DFB \$84
ASC 'FIL'
DFB \$CC

DW L1804 ; LFA
FILL DW DOCOL ; CFA

DW SWAP
DW TOR
DW OVER
DW CSTORE
DW DUP
DW ONEP
DW RFROM
DW ONE
DW SUB
DW CMOVE
DW SEMIS

* ERASE *

L1856 DFB \$85
ASC 'ERAS'
DFB \$C5

DW L1838 ; LFA
ERASE DW DOCOL ; CFA

DW ZERO
DW FILL
DW SEMIS

* BLANKS *

L1866 DFB \$86
ASC 'BLANK'
DFB \$D3

DW L1856 ; LFA
BLANKS DW DOCOL ; CFA

DW BL
DW FILL
DW SEMIS

* HOLD *

L1876 DFB \$84
ASC 'HOL'
DFB \$C4

DW L1866 ; LFA
HOLD DW DOCOL ; CFA

DW LIT
DW -1
DW HLD
DW PSTORE
DW HLD
DW AT
DW CSTORE
DW SEMIS

* PAD *

L1890 DFB \$83
ASC 'PA'
DFB \$C4

DW L1876 ; LFA
PAD DW DOCOL ; CFA

DW HERE
DW CLIT
DFB 68
DW PLUS
DW SEMIS

* WORD *

L1902 DFB \$84
ASC 'WOR'
DFB \$C4

DW L1890 ; LFA
WORD DW DOCOL ; CFA

DW BLK
DW AT
DW ZBRAN
DW 12
DW BLK
DW AT
DW BLOCK
DW BRANCH
DW 6
DW TIB
DW AT
DW IN
DW AT
DW PLUS
DW SWAP
DW ENCL
DW HERE
DW CLIT
DFB 34
DW BLANKS
DW IN
DW PSTORE

DW OVER
DW SUB
DW TOR
DW R
DW HERE
DW CSTORE
DW PLUS
DW HERE
DW ONEP
DW RFROM
DW CMOVE
DW SEMIS

* UPPER *

L1943 DFB \$85
ASC 'UPPE'
DFB \$D2

DW L1902 ; LFA
UPPER DW DOCOL ; CFA

DW OVER
DW PLUS
DW SWAP
DW PDO
DW I
DW CAT
DW CLIT
DFB 95
DW GREAT
DW ZBRAN
DW 9
DW I
DW CLIT
DFB 32
DW TOGGLE
DW PLOOP
DW \$FFEA
DW SEMIS

* (NUMBER) *

L1968 DFB \$88
ASC '(NUMBER'
DFB \$A9

DW L1943 ; LFA
PNUMB DW DOCOL ; CFA

DW ONEP
DW DUP
DW TOR
DW CAT
DW BASE
DW AT
DW DIGIT
DW ZBRAN
DW 44

DW SWAP
DW BASE
DW AT
DW USTAR
DW DROP
DW ROT
DW BASE
DW AT
DW USTAR
DW DPLUS
DW DPL
DW AT
DW ONEP
DW ZBRAN
DW 8
DW ONE
DW DPL
DW PSTORE
DW RFROM
DW BRANCH
DW \$FFC6
DW RFROM
DW SEMIS

* NUMBER *

L2007 DFB \$86
ASC 'NUMBE'
DFB \$D2

DW L1968 ; LFA
NUMBER DW DOCOL ; CFA

DW ZERO
DW ZERO
DW ROT
DW DUP
DW ONEP
DW CAT
DW CLIT
DFB 45
DW EQUALS
DW DUP
DW TOR
DW PLUS
DW LIT
DW -1
DW DPL
DW STORE
DW PNUMB
DW DUP
DW CAT
DW BL
DW SUB
DW ZBRAN
DW 21
DW DUP
DW CAT
DW CLIT

DFB 46
DW SUB
DW ZERO
DW QERROR
DW ZERO
DW BRANCH
DW \$FFDD
DW DROP
DW RFROM
DW ZBRAN
DW 4
DW DMINUS
DW SEMIS

* -FIND *

L2052 DFB \$85
ASC '-FIN'
DFB \$C4

DW L2007 ; LFA
DFIND DW DOCOL ; CFA

DW BL
DW WORD
DW HERE
DW COUNT
DW UPPER
DW HERE
DW CON
DW AT
DW AT
DW PFIND
DW DUP
DW ZEQU
DW ZBRAN
DW \$A
DW DROP
DW HERE
DW LATEST
DW PFIND
DW SEMIS

* (ABORT) *

L2078 DFB \$87
ASC '(ABORT'
DFB \$A9

DW L2052 ; LFA
PABORT DW DOCOL ; CFA

DW ABORT
DW SEMIS

* ERROR *

L2087 DFB \$85
ASC 'ERRO'

DFB \$D2

DW L2078 ; LFA
ERROR DW DOCOL ; CFA

DW WARN
DW AT
DW ZLESS
DW ZBRAN
DW 4
DW PABORT
DW HERE
DW COUNT
DW TYPE
DW PDOTQ
DFB 4
ASC ' ? '
DW MESS
DW SPSTO
DW DROP
DW DROP ; make room
DW IN ; for 2 error
DW AT ; values
DW BLK
DW AT
DW QUIT
DW SEMIS

* ID. *

L2113 DFB \$83
ASC 'ID'
DFB \$AE

DW L2087 ; LFA
IDDOT DW DOCOL ; CFA

DW PAD
DW CLIT
DFB 32
DW CLIT
DFB 95
DW FILL
DW DUP
DW PFA
DW LFA
DW OVER
DW SUB
DW PAD
DW SWAP
DW CMOVE
DW PAD
DW COUNT
DW CLIT
DFB 31
DW ANDD
DW TYPE
DW SPACE
DW SEMIS

* CREATE *

L2142 DFB \$86

ASC 'CREAT'

DFB \$C5

DW L2113 ; LFA

CREATE DW DOCOL ; CFA

DW FIRST ; ensure

DW HERE ; room

DW CLIT ; exists

DFB \$A0 ; in

DW PLUS ; diction'y

DW ULESS

DW TWO

DW QERROR

DW DFIND

DW ZBRAN

DW \$F

DW DROP

DW NFA

DW IDDOT

DW CLIT

DFB 4

DW MESS

DW SPACE

DW HERE

DW DUP

DW CAT

DW WIDTH

DW AT

DW MIN

DW ONEP

DW ALLOT

DW DP ; code

DW CAT ; field

DW CLIT ; mustn't

DFB \$FD ; cross

DW EQUALS ; page

DW ALLOT ; boundary

DW DUP

DW CLIT

DFB \$A0

DW TOGGLE

DW HERE

DW ONE

DW SUB

DW CLIT

DFB \$80

DW TOGGLE

DW LATEST

DW COMMA

DW CURR

DW AT

DW STORE

DW HERE

DW TWOP

DW COMMA
DW SEMIS

* ~[COMPILE] *

L2200 DFB \$C9
ASC '~[COMPILE'
DFB \$DD

DW L2142 ; LFA
DW DOCOL ; CFA

DW DFIND
DW ZEQU
DW ZERO
DW QERROR
DW DROP
DW CFA
DW COMMA
DW SEMIS

* LITERAL *

L2217 DFB \$C7
ASC 'LITERA'
DFB \$CC

DW L2200 ; LFA
LITER DW DOCOL ; CFA

DW STATE
DW AT
DW ZBRAN
DW 8
DW COMP
DW LIT
DW COMMA
DW SEMIS

* DLITERAL *

L2232 DFB \$C8
ASC 'DLITERA'
DFB \$CC

DW L2217 ; LFA
DLIT DW DOCOL ; CFA

DW STATE
DW AT
DW ZBRAN
DW 8
DW SWAP
DW LITER
DW LITER
DW SEMIS

* ?STACK *

L2248 DFB \$86
ASC '?STAC'
DFB \$CB

DW L2232 ; LFA
QSTACK DW DOCOL ; CFA

DW CLIT
DFB TOS
DW SPAT
DW ULESS
DW ONE
DW QERROR
DW SPAT
DW CLIT
DFB BOS
DW ULESS
DW CLIT
DFB 7
DW QERROR
DW SEMIS

* INTERPRET *

L2269 DFB \$89
ASC 'INTERPRE'
DFB \$D4

DW L2248 ; LFA
INTER DW DOCOL ; CFA

DW DFIND
DW ZBRAN
DW 30
DW STATE
DW AT
DW LESS
DW ZBRAN
DW \$A
DW CFA
DW COMMA
DW BRANCH
DW 6
DW CFA
DW EXEC
DW QSTACK
DW BRANCH
DW 28
DW HERE
DW NUMBER
DW DPL
DW AT
DW ONEP
DW ZBRAN
DW 8
DW DLIT
DW BRANCH
DW 6
DW DROP

DW LITER
DW QSTACK
DW BRANCH
DW \$FFC2

* IMMEDIATE *

L2309 DFB \$89
ASC 'IMMEDIAT'
DFB \$C5

DW L2269 ; LFA
DW DOCOL ; CFA

DW LATEST
DW CLIT
DFB 64
DW TOGGLE
DW SEMIS

* VOCABULARY *

L2321 DFB \$8A
ASC 'VOCABULAR'
DFB \$D9

DW L2309 ; LFA
DW DOCOL ; CFA

DW BUILD
DW LIT
DW \$A081
DW COMMA
DW CURR
DW AT
DW CFA
DW COMMA
DW HERE
DW VOCLNK
DW AT
DW COMMA
DW VOCLNK
DW STORE
DW DOES
DOVOC DW TWOP
DW CON
DW STORE
DW SEMIS

* FORTH *

L2346 DFB \$85
ASC 'FORT'
DFB \$C8

DW L2321 ; LFA
FORTH DW DODOE ; CFA

DW DOVOC

DW \$A081
XFOR DW NTOP
VLO DW 0

* DEFINITIONS *

L2357 DFB \$8B
ASC 'DEFINITION'
DFB \$D3

DW L2346 ; LFA
DEFIN DW DOCOL ; CFA

DW CON
DW AT
DW CURR
DW STORE
DW SEMIS

* (*

L2369 DFB \$C1,\$A8

DW L2357 ; LFA
DW DOCOL ; CFA

DW CLIT
DFB 41
DW WORD
DW SEMIS

* QUIT *

L2381 DFB \$84
ASC 'QUI'
DFB \$D4

DW L2369 ; LFA
QUIT DW DOCOL ; CFA

DW ZERO
DW BLK
DW STORE
DW LBRACK
DW RPSTO
DW CR
DW QUERY
DW INTER
DW STATE
DW AT
DW ZEQU
DW ZBRAN
DW 9
DW PDOTQ
DFB 4
ASC ' ok '
DW BRANCH
DW -25
DW SEMIS

* ABORT *

L2406 DFB \$85
ASC 'ABOR'
DFB \$D4

DW L2381 ; LFA
ABORT DW DOCOL ; CFA

DW SPSTO
DW DECIM
DW CR
DW PDOTQ
DFB 14
ASC 'FIG-Forth V1.0'
DW CR
DW FORTH
DW DEFIN
DW QUIT

* COLD *

L2423 DFB \$84
ASC 'COL'
DFB \$C4

DW L2406 ; LFA
COLD DW *+2 ; CFA

LDA ORIG+15 ; from cold start area
STA FORTH+6
LDA ORIG+16
STA FORTH+7
LDY #21
BNE L2433
WARM LDY #15
L2433 LDA ORIG+19
STA UP
LDA ORIG+20
STA UP+1
L2437 LDA ORIG+15,Y
STA (UP),Y
DEY
BPL L2437
LDA #<ABORT
STA IP+1
LDA #>ABORT+2
STA IP
CLD
LDA #\$6C
STA W-1
JMP RPSTO+2

* S->D *

L2453 DFB \$84
ASC 'S->'
DFB \$C4

DW L2423 ; LFA
STOD DW DOCOL ; CFA

DW DUP
DW ZLESS
DW MINUS
DW SEMIS

* +- *

L2464 DFB \$82
ASC '+'
DFB \$AD

DW L2453 ; LFA
PM DW DOCOL

DW ZLESS
DW ZBRAN
DW 4
DW MINUS
DW SEMIS

* D+- *

L2476 DFB \$83
ASC 'D+'
DFB \$AD

DW L2464 ; LFA
DPM DW DOCOL ; CFA

DW ZLESS
DW ZBRAN
DW 4
DW DMINUS
DW SEMIS

* ABS *

L2488 DFB \$83
ASC 'AB'
DFB \$D3

DW L2476 ; LFA
ABS DW DOCOL ; CFA

DW DUP
DW PM
DW SEMIS

* DABS *

L2498 DFB \$84
ASC 'DAB'
DFB \$D3

DW L2488 ; LFA

DABS DW DOCOL ; CFA

DW DUP
DW DPM
DW SEMIS

* MIN *

L2508 DFB \$83

ASC 'MI'
DFB \$CE

DW L2498 ; LFA

MIN DW DOCOL ; CFA

DW OVER
DW OVER
DW GREAT
DW ZBRAN
DW 4
DW SWAP
DW DROP
DW SEMIS

* MAX *

L2523 DFB \$83

ASC 'MA'
DFB \$D8

DW L2508 ; LFA

MAX DW DOCOL ; CFA

DW OVER
DW OVER
DW LESS
DW ZBRAN
DW 4
DW SWAP
DW DROP
DW SEMIS

* M* *

L2538 DFB \$82

ASC 'M'
DFB \$AA

DW L2523 ; LFA

MSTAR DW DOCOL ; CFA

DW OVER
DW OVER
DW XOR
DW TOR
DW ABS
DW SWAP
DW ABS
DW USTAR

DW RFROM
DW DPM
DW SEMIS

* M/ *

L2556 DFB \$82
ASC 'M'
DFB \$AF

DW L2538 ; LFA
MSLASH DW DOCOL ; CFA

DW OVER
DW TOR
DW TOR
DW DABS
DW R
DW ABS
DW USLASH
DW RFROM
DW R
DW XOR
DW PM
DW SWAP
DW RFROM
DW PM
DW SWAP
DW SEMIS

* * *

L2579 DFB \$81,\$AA

DW L2556 ; LFA
STAR DW DOCOL ; CFA

DW USTAR
DW DROP
DW SEMIS

* /MOD *

L2589 DFB \$84
ASC '/MO'
DFB \$C4

DW L2579 ; LFA
SLMOD DW DOCOL ; CFA

DW TOR
DW STOD
DW RFROM
DW MSLASH
DW SEMIS

* / *

L2601 DFB \$81,\$AF

DW L2589 ; LFA
SLASH DW DOCOL ; CFA

DW SLMOD
DW SWAP
DW DROP
DW SEMIS

* MOD *

L2612 DFB \$83
ASC 'MO'
DFB \$C4

DW L2601 ; LFA
MOD DW DOCOL ; CFA

DW SLMOD
DW DROP
DW SEMIS

* */MOD *

L2622 DFB \$85
ASC '*/MO'
DFB \$C4

DW L2612 ; LFA
SSMOD DW DOCOL ; CFA

DW TOR
DW MSTAR
DW RFROM
DW MSLASH
DW SEMIS

* */ *

L2634 DFB \$82
ASC '*'
DFB \$AF

DW L2622 ; LFA
SSLASH DW DOCOL ; CFA

DW SSMOD
DW SWAP
DW DROP
DW SEMIS

* M/MOD *

L2645 DFB \$85
ASC 'M/MO'
DFB \$C4

DW L2634 ; LFA
MSMOD DW DOCOL ; CFA

DW TOR
DW ZERO
DW R
DW USLASH
DW RFROM
DW SWAP
DW TOR
DW USLASH
DW RFROM
DW SEMIS

* USE *

L2662 DFB \$83
ASC 'US'
DFB \$C5

DW L2645 ; LFA
USE DW DOVAR ; CFA

DW DAREA

* PREV *

L2670 DFB \$84
ASC 'PRE'
DFB \$D6

DW L2662 ; LFA
PREV DW DOVAR

DW DAREA

* +BUF *

L2678 DFB \$84
ASC '+BU'
DFB \$C6

DW L2670 ; LFA
PBUF DW DOCOL ; CFA

DW LIT
DW SSIZE+4
DW PLUS
DW DUP
DW LIMIT
DW EQUALS
DW ZBRAN
DW 6
DW DROP
DW FIRST
DW DUP
DW PREV
DW AT
DW SUB
DW SEMIS

* UPDATE *

L2700 DFB \$86

ASC 'UPDAT'

DFB \$C5

DW L2678 ; LFA

UPDATE DW DOCOL ; CFA

DW PREV

DW AT

DW AT

DW LIT

DW \$8000

DW OR

DW PREV

DW AT

DW STORE

DW SEMIS

* FLUSH *

L2705 DFB \$85

ASC 'FLUS'

DFB \$C8

DW L2700 ; LFA

DW DOCOL ; CFA

DW LIMIT

DW FIRST

DW SUB

DW BBUF

DW CLIT

DFB 4

DW PLUS

DW SLASH

DW ONEP

DW ZERO

DW PDO

DW LIT

DW \$7FFF

DW BUFFER

DW DROP

DW PLOOP

DW -10

DW SEMIS

* EMPTY-BUFFERS *

L2716 DFB \$8D

ASC 'EMPTY-BUFFER'

DFB \$D3

DW L2705 ; LFA

DW DOCOL ; CFA

DW FIRST

DW LIMIT

DW OVER
DW SUB
DW ERASE
DW SEMIS

* BUFFER *

L2751 DFB \$86
ASC 'BUFFE'
DFB \$D2

DW L2716 ; LFA
BUFFER DW DOCOL ; CFA

DW USE
DW AT
DW DUP
DW TOR
DW PBUF
DW ZBRAN
DW -4
DW USE
DW STORE
DW R
DW AT
DW ZLESS
DW ZBRAN
DW 20
DW R
DW TWOP
DW R
DW AT
DW LIT
DW \$7FFF
DW ANDD
DW ZERO
DW R
DW STORE
DW R
DW PREV
DW STORE
DW RFROM
DW TWOP
DW SEMIS

* BLOCK *

L2788 DFB \$85
ASC 'BLOC'
DFB \$CB

DW L2751 ; LFA
BLOCK DW DOCOL ; CFA

DW OFFSET
DW AT
DW PLUS
DW TOR
DW PREV

DW AT
DW DUP
DW AT
DW R
DW SUB
DW DUP
DW PLUS
DW ZBRAN
DW 52
DW PBUF
DW ZEQU
DW ZBRAN
DW 20
DW DROP
DW R
DW BUFFER
DW DUP
DW R
DW ONE
DW TWO
DW SUB
DW DUP
DW AT
DW R
DW SUB
DW DUP
DW PLUS
DW ZEQU
DW ZBRAN
DW \$FFD6
DW DUP
DW PREV
DW STORE
DW RFROM
DW DROP
DW TWOP
DW SEMIS

* (LINE) *

L2838 DFB \$86

ASC '(LINE'

DFB \$A9

DW L2788 ; LFA

PLINE DW DOCOL

DW TOR
DW CSLL
DW BBUF
DW SSMOD
DW RFROM
DW BSCR
DW STAR
DW PLUS
DW BLOCK
DW PLUS
DW CSLL
DW SEMIS

* .LINE *

L2857 DFB \$85

ASC '.LIN'

DFB \$C5

DW L2838 ; LFA

DLINE DW DOCOL ; CFA

DW PLINE

DW DTRAI

DW TYPE

DW SEMIS

* MESSAGE *

L2868 DFB \$87

ASC 'MESSAG'

DFB \$C5

DW L2857 ; LFA

MESS DW DOCOL ; CFA

DW WARN

DW AT

DW ZBRAN

DW 27

DW DDUP

DW ZBRAN

DW 17

DW CLIT

DFB 4

DW OFFSET

DW AT

DW BSCR

DW SLASH

DW SUB

DW DLINE

DW BRANCH

DW 13

DW PDOTQ

DFB 6

ASC 'MSG # '

DW DOT

DW SEMIS

* LOAD *

L2896 DFB \$84

ASC 'LOA'

DFB \$C4

DW L2868 ; LFA

LOAD DW DOCOL ; CFA

DW BLK

DW AT

DW TOR

DW IN
DW AT
DW TOR
DW ZERO
DW IN
DW STORE
DW BSCR
DW STAR
DW BLK
DW STORE
DW INTER
DW RFROM
DW IN
DW STORE
DW RFROM
DW BLK
DW STORE
DW SEMIS

* --> *

L2924 DFB \$C3

ASC ' -- '

DFB \$BE

DW L2896 ; LFA

DW DOCOL ; CFA

DW QLOAD

DW ZERO

DW IN

DW STORE

DW BSCR

DW BLK

DW AT

DW OVER

DW MOD

DW SUB

DW BLK

DW PSTORE

DW SEMIS

XEMIT TYA ; writes 1

SEC ; ASCII

LDY #\$1A ; char to

ADC (UP),Y ; terminal

STA (UP),Y

INY ; bump OUT

LDA #0

ADC (UP),Y

STA (UP),Y

LDA 0,X ; fetch char

AND #&7F

STX XSAVE

JSR OSWRCH ; display it

LDX XSAVE

JMP POP

* >VDU *

```
L3000   DFB   $84
        ASC   '>VD'
        DFB   $D5

        DW   L2924
        DW   *+2

        LDA   0,X
        JSR   OSWRCH
        JMP   POP
```

```
XKEY    STX   XSAVE    ; reads one keystroke
        JSR   OSRDCH
        BIT   $FF      ; MJR
        BPL   NOESC    ; MJR
        LDA   $7E      ; MJR
        JSR   OSBYTE   ; MJR
        LDA   $FF      ; MJR
        AND   #127     ; MJR
        STA   $FF      ; MJR
        JMP   REENTR   ; MJR
```

```
NOESC   LDX   XSAVE
        JMP   PUSH0A
```

```
XQTER   LDA   #0
        JMP   PUSH0A    ; dummied
```

```
*
* leave boolean representing terminal break *
*
* system dependent test *
*
```

```
XCR     STX   XSAVE    ; CRLF to terminal
        JSR   OSNEWL   ; monitor call
        LDX   XSAVE
        JMP   NEXT
```

```
* -BCD *
```

```
L3050   DFB   $84
        ASC   '-BC'
        DFB   $C4
```

```
        DW   L3000    ; LFA
DBCD    DW   DOCOL    ; CFA
```

```
DW   ZERO
DW   CLIT
DFB  10
DW   USLASH
DW   CLIT
DFB  16
DW   STAR
```

DW OR
DW SEMIS

* ' (TICK) *

L3202 DFB \$C1,\$A7

DW L3050 ; LFA
TICK DW DOCOL ; CFA

DW DFIND
DW ZEQU
DW ZERO
DW QERROR
DW DROP
DW LITER
DW SEMIS

* FORGET *

L3217 DFB \$86
ASC 'FORGE'
DFB \$D4

DW L3202 ; LFA
FORGET DW DOCOL ; CFA

DW TICK
DW NFA
DW DUP
DW FENCE
DW AT
DW ULESS
DW CLIT
DFB \$15
DW QERROR
DW TOR
DW VOCLNK
DW AT
DW R
DW OVER
DW ULESS
DW ZBRAN
DW L3225-*
DW FORTH
DW DEFIN
DW AT
DW DUP
DW VOCLNK
DW STORE
DW BRANCH
DW -24
L3225 DW DUP
DW CLIT
DFB 4
DW SUB
DW PFA
DW LFA

DW AT
DW DUP
DW R
DW ULESS
DW ZBRAN
DW -14
DW OVER
DW TWO
DW SUB
DW STORE
DW AT
DW DDUP
DW ZEQU
DW ZBRAN
DW -39
DW RFROM
DW DP
DW STORE
DW SEMIS

* BACK *

L3250 DFB \$84
ASC 'BAC'
DFB \$CB

DW L3217 ; LFA
BACK DW DOCOL ; CFA

DW HERE
DW SUB
DW COMMA
DW SEMIS

* BEGIN *

L3261 DFB \$C5
ASC 'BEGI'
DFB \$CE

DW L3250 ; LFA
DW DOCOL ; CFA

DW QCOMP
DW HERE
DW ONE
DW SEMIS

* ENDIF *

L3273 DFB \$C5
ASC 'ENDI'
DFB \$C6

DW L3261 ; LFA
ENDIF DW DOCOL ; CFA

DW QCOMP
DW TWO

DW QPAIR
DW HERE
DW OVER
DW SUB
DW SWAP
DW STORE
DW SEMIS

* THEN * ; (= ENDIF)

L3290 DFB \$C4
ASC 'THE'
DFB \$CE

DW L3273 ; LFA
DW DOCOL ; CFA

DW ENDIF
DW SEMIS

* DO *

L3300 DFB \$C2
ASC 'D'
DFB \$CF

DW L3290 ; LFA
DW DOCOL ; CFA

DW COMP
DW PDO
DW HERE
DW THREE
DW SEMIS

* LOOP *

L3313 DFB \$C4
ASC 'LOO'
DFB \$D0

DW L3300 ; LFA
DW DOCOL ; CFA

DW THREE
DW QPAIR
DW COMP
DW PLOOP
DW BACK
DW SEMIS

* +LOOP *

L3327 DFB \$C5
ASC '+LOO'
DFB \$D0

DW L3313 ; LFA
DW DOCOL ; CFA

DW THREE
DW QPAIR
DW COMP
DW PPLOO
DW BACK
DW SEMIS

* UNTIL *

L3341 DFB \$C5
ASC 'UNTI'
DFB \$CC

DW L3327 ; LFA
UNTIL DW DOCOL ; CFA

DW ONE
DW QPAIR
DW COMP
DW ZBRAN
DW BACK
DW SEMIS

* END * ; (=UNTIL)

L3355 DFB \$C3
ASC 'EN'
DFB \$C4

DW L3341 ; LFA
DW DOCOL ; CFA

DW UNTIL
DW SEMIS

* AGAIN *

L3365 DFB \$C5
ASC 'AGAI'
DFB \$CE

DW L3355 ; LFA
AGAIN DW DOCOL ; CFA

DW ONE
DW QPAIR
DW COMP
DW BRANCH
DW BACK
DW SEMIS

* REPEAT *

L3379 DFB \$C6
ASC 'REPEA'
DFB \$D4

DW L3365 ; LFA

DW DOCOL ; CFA

DW TOR

DW TOR

DW AGAIN

DW RFROM

DW RFROM

DW TWO

DW SUB

DW ENDIF

DW SEMIS

* IF *

L3396 DFB \$C2

ASC 'I'

DFB \$C6

DW L3379 ; LFA

IF DW DOCOL ; CFA

DW COMP

DW ZBRAN

DW HERE

DW ZERO

DW COMMA

DW TWO

DW SEMIS

* ELSE *

L3411 DFB \$C4

ASC 'ELS'

DFB \$C5

DW L3396 ; LFA

DW DOCOL ; CFA

DW TWO

DW QPAIR

DW COMP

DW BRANCH

DW HERE

DW ZERO

DW COMMA

DW SWAP

DW TWO

DW ENDIF

DW TWO

DW SEMIS

* WHILE *

L3431 DFB \$C5

ASC 'WHIL'

DFB \$C5

DW L3411 ; LFA

DW DOCOL ; CFA

DW IF
DW TWOP
DW SEMIS

* SPACES *

L3442 DFB \$86
ASC 'SPACE'
DFB \$D3

DW L3431 ; LFA
SPACES DW DOCOL ; CFA

DW ZERO
DW MAX
DW DDUP
DW ZBRAN
DW 12
DW ZERO
DW PDO
DW SPACE
DW PLOOP
DW -4
DW SEMIS

* <# *

L3460 DFB \$82
ASC '<'
DFB \$A3

DW L3442 ; LFA
BDIGS DW DOCOL ; CFA

DW PAD
DW HLD
DW STORE
DW SEMIS

* #> *

L3471 DFB \$82
ASC '#'
DFB \$BE

DW L3460 ; LFA
EDIGS DW DOCOL ; CFA

DW DROP
DW DROP
DW HLD
DW AT
DW PAD
DW OVER
DW SUB
DW SEMIS

* SIGN *

L3486 DFB \$84
ASC 'SIG'
DFB \$CE

DW L3471 ; LFA
SIGN DW DOCOL ; CFA

DW ROT
DW ZLESS
DW ZBRAN
DW 7
DW CLIT
DFB 45
DW HOLD
DW SEMIS

* # *

L3501 DFB \$81,\$A3

DW L3486 ; LFA
DIG DW DOCOL ; CFA

DW BASE
DW AT
DW MSMOD
DW ROT
DW CLIT
DFB 9
DW OVER
DW LESS
DW ZBRAN
DW 7
DW CLIT
DFB 7
DW PLUS
DW CLIT
DFB 48
DW PLUS
DW HOLD
DW SEMIS

* #S *

L3526 DFB \$82
ASC '#'
DFB \$D3

DW L3501 ; LFA
DIGS DW DOCOL ; CFA

DW DIG
DW OVER
DW OVER
DW OR
DW ZEQU
DW ZBRAN
DW -12
DW SEMIS

DW SEMIS

* D.R *

L3541 DFB \$83
ASC 'D.'
DFB \$D2

DW L3526 ; LFA
DDOTR DW DOCOL ; CFA

DW TOR
DW SWAP
DW OVER
DW DABS
DW BDIGS
DW DIGS
DW SIGN
DW EDIGS
DW RFROM
DW OVER
DW SUB
DW SPACES
DW TYPE
DW SEMIS

* D. *

L3562 DFB \$82
ASC 'D'
DFB \$AE

DW L3541 ; LFA
DDOT DW DOCOL ; CFA

DW ZERO
DW DDOTR
DW SPACE
DW SEMIS

* .R *

L3567 DFB \$82
ASC '.'
DFB \$D2

DW L3562 ; LFA
DOTR DW DOCOL ; CFA

DW TOR
DW STOD
DW RFROM
DW DDOTR
DW SEMIS

* . *

L3585 DFB \$81,\$AE

DW L3567 ; LFA
DOT DW DOCOL ; CFA

DW STOD
DW DDOT
DW SEMIS

* ? *

L3595 DFB \$81,\$BF

DW L3585 ; LFA
QUES DW DOCOL ; CFA

DW AT
DW DOT
DW SEMIS

* LIST *

L3605 DFB \$84
ASC 'LIS'
DFB \$D4

DW L3595 ; LFA
LIST DW DOCOL ; CFA

DW DECIM
DW CR
DW DUP
DW SCR
DW STORE
DW PDOTQ
DFB 6
ASC 'SCR # '
DW DOT
DW CLIT
DFB 16
DW ZERO
DW PDO
DW CR
DW I
DW THREE
DW DOTR
DW SPACE
DW I
DW SCR
DW AT
DW DLINE
DW PLOOP
DW -20
DW CR
DW SEMIS

* INDEX *

L3637 DFB \$85
ASC 'INDE'
DFB \$D8

DW L3605 ; LFA
DW DOCOL ; CFA

DW CR
DW ONEP
DW SWAP
DW PDO
DW CR
DW I
DW THREE
DW DOTR
DW SPACE
DW ZERO
DW I
DW DLINE
DW QTERM
DW ZBRAN
DW 4
DW LEAVE
DW PLOOP
DW -26
DW CLIT
DFB 12 ; FF for printer
DW EMIT
DW SEMIS

* TRIAD *

L3666 DFB \$85
ASC 'TRIA'
DFB \$C4

DW L3637 ; LFA
DW DOCOL ; CFA

DW THREE
DW SLASH
DW THREE
DW STAR
DW THREE
DW OVER
DW PLUS
DW SWAP
DW PDO
DW CR
DW I
DW LIST
DW PLOOP
DW -8
DW CR
DW CLIT
DFB 15
DW MESS
DW CR
DW CLIT
DFB 12 ; FF for printer
DW EMIT
DW SEMIS

* VLIST *

L3696 DFB \$85
ASC 'VLIS'
DFB \$D4

DW L3666 ; LFA
VLIST DW DOCOL ; CFA

DW CLIT
DFB \$80
DW OUT
DW STORE
DW CON
DW AT
DW AT
DW OUT
DW AT
DW CSLL
DW GREAT
DW ZBRAN
DW 10
DW CR
DW ZERO
DW OUT
DW STORE
DW DUP
DW IDDOT
DW SPACE
DW SPACE
DW PFA
DW LFA
DW AT
DW DUP
DW ZEQU
DW QTERM
DW OR
DW ZBRAN
DW \$FFD4
DW DROP
DW SEMIS

* MON *

L4000 DFB \$83
ASC 'MO'
DFB \$CE

DW L3696 ; LFA
MON DW *+2 ; CFA

STX XSAVE
BRK ; break out
LDX XSAVE ; to monitor
JMP NEXT ; and reenter

NTOP DFB \$84

ASC 'NOO'
DFB \$D0

DW L4000 ; LFA
NOOP DW DOCOL ; CFA

DW SEMIS ; NULL DEF'N

TOP ; of dictionary LST ON

LST OFF