

```
0
1
2
3 : IMMEDIATE 80 CURRENT @ @ 4 - +! ; 12 USER OFFSET
4 VOCABULARY ASSEMBLER
5 ( ARITHMETIC OPERATORS)
6 ( ASSEMBLER)
7 ( MACROS)
8 ( TRANSFERS)
9 ( COMPILER)
10 ( ERRORS)
11 ( VOCABULARIES)
12 ( NUMBER FORMATTING)
13 ( OUTPUT VOCABULARY)
14 ( EDITOR) FORTH DEFINITIONS FORGET TASK : TASK ;
15 ( LOWER BAUD SERIAL I/O)
16 ( TERMINAL I/O)
17 ( HIGHER BAUD SERIAL I/O)
18 ( RESTART)
19 ( OPTIONS) DECIMAL
20 ( ASSEMBLER)
21 ( STRING EDITOR)
22 ( CHARACTER EDITOR PRIMITIVES)
23 ?
24 ( DISK UTILITIES) FORTH DEFINITIONS FORGET TASK : TASK ;
25 ( DISK COMPARE)
26 ( DISK ERROR CHECKING) HEX
27 ( DOCUMENTATION) FORTH DEFINITIONS FORGET TASK : TASK ;
28 ( RECONFIGURE) FORTH DEFINITIONS FORGET TASK : TASK ;
29
30
31
32
33
34
35
36
37
38
39 ( COMPILER) FORTH DEFINITIONS FORGET TASK : TASK ;
40
41 ( TARGET VOCABULARY)
42 ( TARGET DICTIONARY)
43 ( TARGET COMPILER)
44 ( TARGET ASSEMBLER)
45 ( RAM DEFINING) HERE
46 ( PROM DEFINING) HERE
47 ( COMPILING WORDS) HEX
48 ( INITIALIZE NUCLEUS)
49 ( INITIALIZE SYMBOLS)
50 ( COMPILER OUTPUT)
51 0003 EQU ZERO 0006 EQU 'NEXT
52 0151 EMPLACE 2+ 015D EMPLACE MOVE
53 ( MORE DEFINITIONS)
54 ( CROSS ILLUSTRATION)
55
56
57
58
59
```

3

```
0 : IMMEDIATE 80 CURRENT @ @ 4 - +! ; 12 USER OFFSET
1 : ( 29 WORD ; IMMEDIATE : 8* 2* 2* 2* ;
2 : LOAD BLK C@ <R IN @ <R 0 IN ! 8* OFFSET @ + F9F DROP
3 : BLK ! INTERPRET R> IN ! R> BLK C! ;
4 : <BUILDS 0 CONSTANT ;
5
6 ( COMPILER) 9 LOAD ( VOCAB'S) OB LOAD ( ASSEMBLER) 4 LOAD
7 ( TERMINAL) OF LOAD 10 LOAD ( ARITHMETIC) 5 LOAD
8 : BLOCK OFFSET @ + F9F MIN BLOCK ;
9 / MIN 2 - / LOAD 1E + !
10 ( FORMATTING) OC LOAD ( OUTPUT) OD LOAD
11 ( ERRORS) OA LOAD 12 LOAD ( OPTIONS) 13 LOAD
12 : ERASE-CORE 1FEA 106 ERASE ; DECIMAL
13 : FLUSH 2 0 DO 0 BUFFER ! LOOP ; : TASK ; OK QUIT
14
15 COPYRIGHT (C) FORTH INC. MAY 1978
```

4

```
0 VOCABULARY ASSEMBLER
1 / ASSEMBLER 2+ / FORTH 1+ : VANISH IN-LINE IN-LINE ! ;
2 : CODE CREATE ASSEMBLER ;
3 : ;CODE COMPILE \ ;CODE ASSEMBLER ; IMMEDIATE
4 : PAGE HERE MINUS CZ DROP CZ SWAP OVER SWAP < U* H +! ;
5
6 : OCTAL 8 BASE C! ; : DECIMAL OA BASE C! ;
7 : HEX 10 BASE C! ; ( ASSEMBLE) 14 LOAD
8
9 : CVARIABLE USER ;CODE W GLO S DST W GHI PUSH
10 : VARIABLE CZ CVARIABLE CZ C, ;
11
12 13 PAGE
13 CODE ERASE 2POP BEGIN BEGIN 0#LD A STR A INC
14 T DEC T GLO 0= END T GHI 0= END NEXT
15
```

5

```
0 ( ARITHMETIC OPERATORS)
1 : > SWAP < ; / 0= CODE NOT HERE 2 - !
2
3 : M* <R DUP I U* SWAP FFOO AND CZ R> U* + ;
4 : M/MOD <R I U/ CZ DROP R> SWAP <R U/ R> + ;
5 : */MOD <R M* R> M/MOD ; : */ */MOD SWAP DROP ;
6 : /MOD <R CZ R> M/MOD ; : MOD /MOD DROP ;
7 : * M* CZ DROP ; : / /MOD SWAP DROP ;
8
9 CODE LEAVE 0#LD R INC R INC R INC R STR R DEC
10 R STR R DEC R DEC NEXT
11 CODE J R INC R INC R INC R INC R INC R LDN S DST
12 R DEC R LDN R DEC R DEC R DEC R DEC PUSH
13
14 : MAX OVER OVER < IF SWAP THEN DROP ;
15 : MIN OVER OVER > IF SWAP THEN DROP ;
```

6

```
0 ( ASSEMBLER)
1 OE CONSTANT R   OD CONSTANT S   OC CONSTANT I   OB CONSTANT W
2 OA CONSTANT A   9  CONSTANT T   4  CONSTANT U   3  CONSTANT P
3
4 : #   8 MODE C! ;           : +C   80 MODE 1+ C! ;
5
6 : 1RG   <BUILDS C, DOES> C@ + C, ;
7 : ALU   <BUILDS C, DOES> C@ MODE @ + MODE C@ IF
8         MODE C@ + C, THEN C, 0 MODE ! ;
9
10 00 1RG LDN   10 1RG INC   20 1RG DEC   40 1RG LDA   50 1RG STR
11 60 1RG OUT   68 1RG INP   80 1RG GLO   90 1RG GHI   A0 1RG PLO
12 B0 1RG PHI   D0 1RG SEP   E0 1RG SEX   F0 ALU LD   FO ALU LDX
13 F4 ALU ADD   F5 ALU SD    F6 ALU SHR   F7 ALU SM    FE ALU SHL
14 F1 ALU OR    F2 ALU AND   F3 ALU XOR   72 ALU LDXA  73 ALU STXD
15 7A ALU REQ   7B ALU SEQ   70 ALU RET   78 ALU SAV
```

7

```
0 ( MACROS)
1 : ENTRY   <BUILDS , DOES> @ HERE 2 - ! ;
2
3 : NEXT    F SEP ;           : 0#LD   F GHI ;
4
5 : DST     DUP DEC   STR ;
6 : PUSH    S DST NEXT ;
7 ;S
8
9
10
11
12
13
14
15
```

8

```
0 ( TRANSFERS)
1 ( CODE NOT S INC S LDN 8 # XOR S STR S DEC NEXT)
2 39 CONSTANT Q 3A CONSTANT O= 3B CONSTANT DFL
3 : EFL DFL + ;           : < DFL NOT ;
4 : O< 80 # AND O= NOT ;
5
6 : IF C, HERE 0 C, ;
7 : THEN HERE SWAP C! ;
8 : ELSE 30 IF SWAP THEN ;
9
10 : END C, C, ;
11 : BR 30 END ;
12 : LS 94 + C, ; ( CONDITIONAL LONG SKIP)
13 ;S
14
15
```

9

```
0 ( COMPILER)
1 : (MARK)  HERE - C, ;
2 : (THEN)  HERE OVER - SWAP C! ;
3
4 : DO      \ DO HERE ; IMMEDIATE
5 : LOOP    \ LOOP (MARK) ; IMMEDIATE
6 : +LOOP   \ +LOOP (MARK) ; IMMEDIATE
7
8 : IF      \ IF HERE 0 C, ; IMMEDIATE
9 : ELSE    \ ELSE HERE 0 C, SWAP (THEN) ; IMMEDIATE
10 : WHILE   \ WHILE SWAP (MARK) (THEN) ; IMMEDIATE
11 : THEN    (THEN) ; IMMEDIATE
12
13 : BEGIN   HERE ; IMMEDIATE
14 : END     \ END (MARK) ; IMMEDIATE
15
```

10

```
- 0 ( ERRORS)
1 CODE S!  IPOP  S PLO  A GHI  S PHI  NEXT
2 CODE 'S  S GHI  T PHI  S GLO  S DST  T GHI  PUSH
3
4 : MESSAGE 17 LINE -TRAILING TYPE SPACE ;
5 : QUESTION HERE COUNT TYPE MESSAGE SO @ S! BLK @ QUIT ;
6 ' QUESTION 2 - ' ' 7 + ! ( CHANGE STACK LEVEL) 0
7 : ?STACK 'S 2+ SO @ OVER < IF 1 QUESTION THEN
8 HERE AO + < IF 2 QUESTION THEN ;
9 ' ?STACK 2 - ' INTERPRET 43 + !
- 10 : (NUMBER) DUP 1+ C@ 2D = SWAP OVER + NUMBER C@ 20 -
11 IF 0 QUESTION THEN SWAP IF MINUS THEN 0 ;
12 ' (NUMBER) 2 - ' INTERPRET 7 + !
13 ;S
14
15
```

11

```
- 0 ( VOCABULARIES)
1 : VOCABULARY <BUILDS CURRENT @ 1 - , DOES> CONTEXT ! ;
- 2 : DEFINITIONS CONTEXT @ CURRENT ! ;
3 VOCABULARY FORTH IMMEDIATE FORTH DEFINITIONS
4 CURRENT @ DUP 7 - SWAP !
5
6 : ' -' IF 0 MINUS THEN ;
7 : FORGET CURRENT @ CONTEXT ! ' DUP 8 - H !
- 8 4 - @ CONTEXT @ ! ;
9
10 : IN-LINE \ ['] , ; IMMEDIATE
11 : ['] \ ['] ' , ; IMMEDIATE
- 12 : [SWAP] SWAP ; IMMEDIATE
13 ;S
14
15
```

12

```
0 ( NUMBER FORMATTING)
1 MSG SPACE 1 C, 20 C, MSG CR 6 C, OD C, OA C, 0, 0,
2 : SPACES -DUP IF 0 DO SPACE LOOP THEN ;
3
4 0 VARIABLE HLD : HOLD HLD @ 1 - DUP HLD ! C! ;
5
6 : PAD HERE 41 + ; : <# PAD HLD ! ;
7 : #> DROP HLD @ PAD OVER - ;
8 : SIGN SWAP 0< IF 2D HOLD THEN ;
9 : # BASE C@ /MOD SWAP 9 OVER < IF 7 + THEN 30 + HOLD ;
10 : #S BEGIN # DUP 0= END ;
11 : (.) DUP ABS <# #S SIGN #> ;
12 : . (.) TYPE SPACE ; : ? @ . ;
13 ;S
14
15
```

13

```
0 ( OUTPUT VOCABULARY)
1 : .R <R (.) R> OVER - SPACES TYPE ;
2 : DUMP 0 DO CR DUP 5 .R 10 0 DO I 7 AND 0= 2* SPACES
3 DUP C@ 3 .R 1+ LOOP 10 +LOOP DROP SPACE ;
4
5 : -TRAILING BEGIN 1 - OVER OVER + C@ 20 = 0=
6 OVER 0< + END 1+ ;
7 : LINE <R 40 80 */MOD R> 8 * + BLOCK + 40 ;
8 : TEXT 20 HERE C! HERE DUP 1+ 40 MOVE
9 WORD HERE 1+ PAD 40 MOVE ;
10 14 USER SCR 16 USER R#
11 : LIST SCR ! 10 0 DO CR I 3 .R SPACE I SCR @ LINE
12 -TRAILING TYPE LOOP CR ;
13 : L SCR @ LIST ;
14 : [ I COUNT DUP 1+ R> + <R TYPE ;
15 : [ \ [ 5D WORD HERE C@ 1+ H +! ; IMMEDIATE
```

14

```
0 ( EDITOR) FORTH DEFINITIONS FORGET TASK : TASK ;
1 BASE C@ HEX VOCABULARY EDITOR IMMEDIATE EDITOR DEFINITIONS
2 : LINE OF AND SCR @ LINE ; : HOLD LINE PAD SWAP MOVE ;
3 : GAP 1+ OF OVER - 0 DO OE I - LINE DROP OF I - LINE MOVE
4 UPDATE LOOP ;
5
6 : T CR SPACE SPACE DUP HOLD PAD 40 TYPE DUP 40 * R# ! ;
7 : D DUP HOLD OF SWAP DO I 1+ LINE DROP I LINE MOVE
8 UPDATE LOOP ;
9 : R PAD SWAP LINE MOVE UPDATE ; : I GAP R ;
10 : " 22 TEXT ; : P 5E TEXT R ; : A GAP P ;
11
12 : COPY 8* OFFSET @ + SWAP 8* DUP 8 + SWAP DO
13 DUP FORTH I BLOCK 2 - ! UPDATE 1+ LOOP DROP ;
14 : DELETE 8* BLOCK 0 SWAP ! UPDATE ; : TOP 0 R# ! ;
15 16 LOAD 15 LOAD FORTH DEFINITIONS EDITOR BASE C!
```

15

```
0 ( LOWER BAUD SERIAL I/O)
1 27 PAGE
2 CODE ECHO ( EVEN, ODD) OB # LD A PLO
3 W SEX W LDA SHR S INC
4 BEGIN DFL NOT IF W DEC 7 OUT
5 ELSE P SEP 7 OUT W DEC THEN ( *) 59 # LD
6 BEGIN C4 C, P SEP 1 # SM O= END S LDN
7 +C SHR S STR A DEC A GLO O= END S INC NEXT
8
9 23 PAGE
10 CODE KEY BEGIN 4 EFL END BEGIN 4 EFL NOT END
11 80 # LD P SEP BEGIN T PLO ( *) 60 # LD
12 BEGIN C4 C, P SEP 1 # SM O= END
13 T GLO SHR 4 EFL IF 80 # OR OVER DFL END
14 THEN P SEP DFL END S DST O#LD PUSH
15
```

16

```
0 ( TERMINAL I/O)
1 : TYPE -DUP IF 0 DO DUP C@ ECHO 1+ LOOP THEN DROP ;
2 : COUNT DUP 1+ SWAP C@ ;
3 : MSG <BUILDS DOES> COUNT TYPE ;
4 MSG OK 4 C, 4F C, 4B C, OD C, OA C,
5
6 : EXPECT 0 DO KEY 7F AND
7 DUP OD = IF DROP 20 ECHO 50
8 ELSE DUP 7F = IF DROP 1 0= 0= DUP 7 + ECHO
9 MINUS SWAP OVER + SWAP
10 ELSE DUP ECHO OVER C! 1+ 1
11 THEN THEN +LOOP 0 SWAP ! ;
12
13 CODE R! U GHI R PHI U GLO R PLO NEXT
14 : QUIT 0 STATE C! BEGIN R! 0 BLK C! 0 IN !
15 SO @ 50 EXPECT INTERPRET OK 0 END ;
```

17

```
0 ( HIGHER BAUD SERIAL I/O)
1 25 PAGE
2 CODE ECHO ( EVEN, ODD) OB # LD A PLO
3 W SEX W LDA SHR S INC
4 BEGIN DFL NOT IF W DEC 7 OUT
5 ELSE P SEP 7 OUT W DEC THEN ( *) 1 # LD
6 BEGIN 1 # SM O= END S LDN +C SHR S STR
7 A DEC A GLO O= END S INC NEXT
8
9 23 PAGE
10 CODE KEY BEGIN 4 EFL END BEGIN 4 EFL NOT END 80 # LD
11 C4C4, P SEP BEGIN P SEP BEGIN T PLO ( *) 3 # LD
12 BEGIN 1 # SM O= END T GLO SHR 4 EFL IF
13 80 # OR SWAP DFL END THEN DFL END
14 S DST O#LD PUSH
15
```

18

```
0 ( RESTART)
1 ASSEMBLER C4CO 0 ! HERE 2 ! 1 OUT 1 C,
2 OPERATOR CZ # LD U PHI R PHI CZ # LD U PLO R PLO
3 SO @ 2 - CZ # LD S PHI CZ # LD S PLO / QUESTION
4 CZ # LD I PHI CZ # LD I PLO 3 # LD S DST 0 # LD
5 S DST F PHI 5 ( NEXT) # LD F PLO NEXT ;S
6
7
8
9
10
- 11
12
13
- 14
15
```

19

```
0 ( OPTIONS) DECIMAL
1 : DRO 0 OFFSET ! ; : DR1 2000 OFFSET ! ;
2 14 CONSTANT EDIT
3 20 CONSTANT ASSEMBLE
4 24 CONSTANT DISKING
5 27 CONSTANT PRINTING
6 39 CONSTANT CROSS
7 HEX ;S
8
9
10
11
12
- 13
14
15
```

20

```
0 ( ASSEMBLER)
1 / FORTH 1+ / ASSEMBLER 2+ @ = 0= 5 U* BLK +!
2 BASE C@ HEX HERE SO @ 400 - H !
3
4 ASSEMBLER DEFINITIONS 18 USER MODE
5 6 LOAD 7 LOAD
- 6 CODE NOT S INC S LDN 8 # XOR S STR S DEC NEXT
7 8 LOAD
8 / + 2 - @ ENTRY BINARY / @ 2 - @ ENTRY PUT
9 / +! 2 - @ ENTRY 2POP / ! 2 - @ ENTRY 1POP
10
11 H ! BASE C! FORTH DEFINITIONS ;S
- 12
13
14
- 15
```

21

```
0 ( STRING EDITOR)
1 : C#  R# @ 40 MOD ;          : #LEFT  40 C# - ;
2 : L#  R# @ 40 / ;           : AT   L# LINE  DROP C# + ;
3 : STRING 5E WORD HERE PAD 40 MOVE ;
4 : ERR  IF TOP PAD HERE 40 MOVE 0 QUESTION THEN ;
5 : ILINE AT 1 - #LEFT 1+ PAD COUNT (MATCH) SWAP AT - R# +! ;
6 : FIND  BEGIN 3FF R# @ < ERR ILINE END ;
7 : M  UPDATE R# +! CR SPACE SPACE L# LINE DROP C# TYPE
8   5F ECHO AT #LEFT TYPE L# . ;
9 : REMOVE  DUP 40 /MOD LINE OVER + <R + C# IF AT OVER #LEFT
10  MOVE THEN DROP R# @ SWAP - R> OVER - OVER BLANK MINUS M ;
11 : B  PAD C@ MINUS M ;      : N  FIND 0 M ;      : F  STRING N ;
12 : X  STRING FIND R# @ PAD C@ - REMOVE ;
13 : TILL R# @ STRING ILINE 0= ERR REMOVE ;
14 : C  STRING PAD COUNT <R #LEFT FORTH I - DUP 0< ERR
15   -DUP IF AT DUP I + ROT (MOVE) THEN AT I MOVE R> M ;
```

22

```
0 ( CHARACTER EDITOR PRIMITIVES)
1 ASSEMBLE LOAD EDITOR DEFINITIONS
2 45 PAGE CODE (MATCH) S INC BEGIN S SEX S LDA T PHI
3 S LDA W PHI S LDA W PLO S INC S LDA T PLO
4 T DEC S LDA A PHI S LDN A PLO A INC A GLO
5 STXD A GHI STXD T GLO STXD 0= IF NEXT
6 THEN S DEC S DEC S DEC W SEX
7 BEGIN A LDA SM SWAP 0= END T GHI I # SM T PHI
8 0= NOT IF W INC T DEC T GLO SWAP 0= END THEN
9 S SEX S INC S INC S INC S INC S INC S INC
10 A GLO STXD A GHI STXD T GLO STXD NEXT
11
12 11 PAGE CODE BLANK 2POP BEGIN 20 # LD A STR A INC
13 T DEC T GLO 0= END NEXT
14
15 : (MOVE) <R SWAP HERE FORTH I MOVE HERE SWAP R> MOVE ;
```

23

```
0 ?
1 STACK EMPTY!
2 DICTIONARY FULL!
3 RESTART!
4 DISK ERROR...
```

```
5
6
7
8
9
10
11
12
13
14
15
```

Copyright 1978 by FORTH, Inc.

RCA 1802 microFORTH

24

```
0 ( DISK UTILITIES)  FORTH DEFINITIONS  FORGET TASK  : TASK ;
1 DECIMAL          2000 CONSTANT NEW  2 CONSTANT NB  7 CONSTANT INC
2 ASSEMBLE LOAD 26 LOAD
3 : FILL  NB 0 DO 3999 BUFFER ! LOOP ; 25 LOAD
4
5 : RIGHT  FILL 8* SWAP 8* INC 0 DO OVER OVER I + DO
6 I NEW + I [BLOCK] 2 - ! UPDATE I J - 1+ NB MOD 0= IF
7 FLUSH THEN INC +LOOP FLUSH LOOP DROP DROP ;
8 : BACKUP  0 250 RIGHT ;
9
10 : SWEEP  FILL 8* SWAP 8* INC 0 DO OVER OVER I + DO
11 I [BLOCK] DROP INC +LOOP LOOP DROP DROP ;
12 ;S
13
14
15
```

25

```
0 ( DISK COMPARE)
1 : MATCH  FILL SWAP DO I 8* DUP [BLOCK] @ SWAP NEW + [BLOCK]
2 @ + IF I 8* 8 0 DO DUP [BLOCK] OVER NEW + [BLOCK]
3 64 0 DO OVER @ OVER @ - IF
4 DROP DROP CR 8 / . J 2* . LEAVE -1 0 0 THEN
5 2+ SWAP 2+ LOOP DROP DROP
6 DUP 0< IF LEAVE THEN 1+ LOOP DROP THEN LOOP ;
7 ;S
8
9
10
11
12
13
14
15
```

26

```
0 ( DISK ERROR CHECKING)  HEX
1 CODE STATUS  S SEX  5 OUT  6 INP  S DEC  NEXT
2 : STATUS  0 STATUS ; : ERROR  STATUS 28 AND ;
3 : LOG  CR . ERROR 4 MESSAGE . ;
4 : NOTIFY  2 - DUP @ 0 ROT ! DUP LOG OFFSET @ - ;
5 : [BLOCK]  BLOCK 0A 0 DO ERROR IF NOTIFY BLOCK
6 ELSE LEAVE THEN LOOP ;
7 DECIMAL ;S
8
9
10
11
12
13
14
15
```

27

```
0 ( DOCUMENTATION)  FORTH DEFINITIONS  FORGET TASK  : TASK ;
1 BASE C@  DECIMAL  MSG LF 1 C, 10 C,
2
3 : INDEX  SWAP OVER SWAP DO  CR LF LF  DUP  I 60 +  MIN I DO
4   CR  I 3 .R 2 I < IF  I 8* BLOCK @ IF
5   SPACE 0 I LINE  -TRAILING TYPE THEN THEN LOOP
6   CR LF 14 MESSAGE CR 60 +LOOP  DROP ;
7
8 : TRIAD  3 / 3 *  DUP 3 + SWAP  OVER OVER DO
9   I 8* BLOCK @ LOOP  + + IF  DO  CR LF LF  I 10 .R
10  CR  I 8* BLOCK @ 0=  I 3 < +  IF 17 0 DO  LF LOOP
11  ELSE I LIST THEN LOOP
12  LF LF 14 MESSAGE CR  ELSE DROP DROP THEN ;
13
14 : SHOW  1+ SWAP 3 / 3 * DO  I TRIAD 3 +LOOP ;
15 BASE C!
```

28

```
0 ( RECONFIGURE)  FORTH DEFINITIONS  FORGET TASK  : TASK ;
1 HEX 2 CONSTANT NB 2000 CONSTANT LIMIT
2 LIMIT NB 83 * - CONSTANT FIRST  / OPERATOR 15 + CONSTANT INIT
3 : B!  80 /MOD 8 + BLOCK + C! UPDATE ;
4 : P!  <R CZ I B! CZ R> 1+ B! ;
5 : DISK  [ / ] OPERATOR 2+  / C@ + ;
6
7 FIRST 20 -  DUP INIT ( U,R) P!
8  DUP / OPERATOR  OVER OVER 3 + ( LINK) P!  P!
9  3 +  DUP DISK CONTEXT P!  DUP DISK CURRENT P!
10 FIRST A0 -  DUP DISK S0 P!  INIT 2+ ( INITIAL S) P!
11 LIMIT 83 -  DUP 4 ( PREV) P!  6 ( USE) P!
12  LIMIT / IN OB + ( LIMIT) P!
13  FIRST / IN OF + ( FIRST) P!
14
15 FIRST . NB 83 * . NB . DECIMAL  FLUSH  FORGET NB
```

29

39

```
0 ( COMPILER)  FORTH DEFINITIONS  FORGET TASK  : TASK ;
1 HEX      VANISH
2
3 VOCABULARY STUB  VOCABULARY HOST  HOST DEFINITIONS
4 : .  BASE C@ SWAP  HEX <# #S #> TYPE SPACE  BASE C! ;
5 : LOG  DUP .  IN @ 20 WORD  IN !  HERE COUNT TYPE  SPACE ;
6
7 F50 CONSTANT NEW      O VARIABLE WO      O VARIABLE H
8 : ADRS  WO @ -  O MAX 80 /MOD 4F MIN  NEW + BLOCK  + ;
9 : TEXT  HERE ;      : HERE  H @ ;      : ORG  H ! ;
10
- 11 CODE NOT  1D C,  OD C,  FB08 ,  5D C,  2D C,  DF C,
12
- 13 : C,  HERE ADRS  C! UPDATE  1 H +! ;
- 14 : ,  CZ C,  CZ C, ;
15 29 LOAD  2A LOAD  2B LOAD  2C LOAD
```

40

41

```
0 ( TARGET VOCABULARY)
1 : CLEAR  50 O DO  NEW I + BLOCK  80 ERASE UPDATE  LOOP FLUSH ;
2
3 STUB DEFINITIONS  : DOES>  \ DOES>  / 2 - ,  HOST ; IMMEDIATE
4 HOST DEFINITIONS  : <BUILDS  \ <BUILDS  STUB ;  IMMEDIATE
5
6 VOCABULARY TARGET
7 TARGET DEFINITIONS  0100 TEXT  0000 TEXT 4 +
8 : X  80 IN +! ;  HOST DEFINITIONS  ! !
9
10 : IMMEDIATE  CURRENT @  DUP @  SWAP OVER 1+ @  SWAP !
11      [ / ] TARGET 2+  OVER OVER @  SWAP 1+ ! ! ;
12 : SMUDGE  8000 [ / ] TARGET 2+ @  +! ;
13
14 : EMPLACE  <BUILDS  ,  DOES>  @ 2 - , ;
15 : EMPLACE  EMPLACE IMMEDIATE ;
```

42

```
0 ( TARGET DICTIONARY)
1 0 VARIABLE LAST
2 : CODE HERE DUP LAST ! 2+ LOG DUP , EMPLACE ;
3 : PAGE HERE MINUS CZ DROP CZ SWAP OVER > * DUP . H +! ;
4
5 : H@ @ ; : H! ! ;
6
7 : INSIDE DUP WO @ < OVER HERE > + 0= ;
8 : C@ INSIDE IF ADRS THEN C@ ;
9 : C! INSIDE IF ADRS UPDATE THEN C! ;
10 : @ <R I 1+ C@ CZ DROP R> C@ CZ DROP ;
11 : ! <R CZ I C! CZ R> 1+ C! ; ;S
12
13
14
15
```

43

```
0 ( TARGET COMPILER)
1 : ,HOST COMPILE HERE FORTH , ;
2 : ;CODE R> H@ LAST H@ ! ;
3 : ;CODE \ ;CODE ,HOST ; FORTH IMMEDIATE
4
5 : -/ TARGET DEFINITIONS -/ HOST DEFINITIONS ;
6 ( MAKE HOST IMMEDIATE) 80 / HOST 9 - +!
7 : / -/ IF 0 QUESTION THEN 2+ H@ ;
8 : \ -/ IF 0 QUESTION THEN 2 - FORTH , ; IMMEDIATE
9
10 0 VARIABLE <LIT> 0 VARIABLE <[']>
11 : COMPILE BEGIN -/ IF TEXT (NUMBER) DROP
12 DUP FFOO AND IF <[']> H@ , , ELSE <LIT> H@ , C, THEN
13 ELSE EXECUTE ?STACK THEN 0 END ; ;S
14
15
```

44

```
0 ( TARGET ASSEMBLER)
1 : EQU LOG CONSTANT ; : LABEL HERE EQU ;
2 : BEGIN HERE ; 0 USER MODE 6 LOAD 7 LOAD 8 LOAD
3 DECIMAL
4 : LOAD CR DUP FORTH . LOAD ; : CR CR SPACE ;
5 45 CONSTANT RAM 46 CONSTANT PROM 47 CONSTANT COLON
6 48 CONSTANT NUCLEUS 49 CONSTANT SYMBOLS 50 CONSTANT OUTPUT
7 53 CONSTANT MORE : COMPUTER ; ;S
8
9
10
11
12
13
14
15
```

45

```
0 ( RAM DEFINING)  HERE
1 : ACCESSIBLE  IN H@  HERE 2+  CONSTANT  IN H! ;
2
3 'CONSTANT' ORG   : CONSTANT  CODE  , ;CODE
4 'USER' ORG      : USER  CODE  C, ;CODE
5 'VARIABLE' ORG  : TABLE  CODE  ;CODE
6                 : CARIABLE  ACCESSIBLE USER ;CODE
7                 : VARIABLE  ACCESSIBLE CONSTANT ;CODE
8 'DOES>' ORG
9 : DOES>  R> H@  LAST H@ 2+ ! ;CODE          ORG
10 : DOES>  FORTH \ HOST DOES> ,HOST COMPILE ;  FORTH IMMEDIATE
11 : <BUILDS  0 CONSTANT ; ;S
12
13
14
15
```

46

```
0 ( PROM DEFINING)  HERE
1 'CONSTANT' ORG   : CONSTANT  CODE  , ;CODE
2 'USER' ORG      : USER  CODE  C, ;CODE
3 'VARIABLE' ORG  : TABLE  CODE  ;CODE
4
5 0 VARIABLE N     : THERE  N H@ ; ; RES  N +! ;
6 : CARIABLE  IN H@  THERE DUP CONSTANT . SPACE
7   FORTH IN !  THERE CONSTANT DROP 1 RES ;
8 : VARIABLE  CVariable 1 RES ;
9
10 'DOES>' ORG
11 : DOES>  R> H@  LAST H@ 2+ ! ;CODE          ORG
12 : DOES>  FORTH \ HOST DOES> ,HOST COMPILE ;  FORTH IMMEDIATE
13 : <BUILDS  0 CONSTANT ; ;S
14
15
```

47

```
0 ( COMPILING WORDS)  HEX
1 : IN-LINE  \ ['] , ; IMMEDIATE
2 : [']  ' \ IN-LINE ; IMMEDIATE
3 : [SWAP]  SWAP ; IMMEDIATE
4 : ( 29 WORD ; IMMEDIATE
5 : ; \ ;S R> DROP ; IMMEDIATE
6
7 : H:  ; ; HERE
8 'S' ORG : : CODE SMUDGE COMPILE SMUDGE ;CODE
9 DECIMAL  ORG ;S
10
11
12
13
14
15
```

48

```
0 ( INITIALIZE NUCLEUS)
1   FORTH / TARGET 2+ DUP 5 + SWAP !
2 FORGET COMPUTER   : COMPUTER ;
3   O ORG   O WO H!   CLEAR
4
5 4 CONSTANT NUC
6 : COPY   FORTH 4 O DO NEW I + NUC I + BLOCK 2 - ! UPDATE
7   LOOP FLUSH ;      COPY      FORGET NUC
8
9 ( SYMBOLS) HEX   33 LOAD 34 LOAD   DECIMAL
- 10 / LIT 2 - <LIT> H!      / ['] 2 - <[']> H!
11 ;S
12
13
- 14
15
```

49

```
0 ( INITIALIZE SYMBOLS)
1   FORTH / TARGET 2+ DUP 5 + SWAP !
2 FORGET COMPUTER   : COMPUTER ;
3   O ORG   O WO H!   CLEAR
4
5 ( SYMBOLS) HEX   33 LOAD 34 LOAD   DECIMAL
- 6 / LIT 2 - <LIT> H!      / ['] 2 - <[']> H!
7 ;S
8
9
- 10
11
12
13
14
15
```

50

```
0 ( COMPILER OUTPUT)
1   FORTH / TARGET 2+ DUP 5 + SWAP !      ASSEMBLE LOAD
2 HOST DEFINITIONS   FORGET COMPUTER   : COMPUTER ;   HEX
3 : DUMP   FORTH   O DO CR   DUP 5 .R   10 O DO   I 7 AND O=
4   2* SPACES   DUP ADRS C@ 3 .R   1+ LOOP   10 +LOOP   DROP CR ;
5
6 : LEADER   64 O DO   O ECHO   LOOP ;      400 VARIABLE SIZE
7 : PROMS   FORTH LEADER O DO LEADER OFF ECHO   SIZE @ O DO
8   DUP ADRS C@ ECHO 1+ LOOP LOOP DROP LEADER LEADER ;
- 9 ( NOTE: ENTIRE ASSUMES A 4K SPACE AT PAD)
10 : ENTIRE   FORTH PAD NEW 20 + NEW DO   I BLOCK   OVER 80 MOVE
11   80 + LOOP DROP ;      16 PAGE
- 12 CODE SIMULATE   2POP   S LDA   P PHI   S LDA   P PLO
13   BEGIN   P LDA   A STR   A INC   T DEC   T GLO   DUP O= END
14   T GHI   O= END   O C,   ( WAITS FOR RESET, RUN P)
- 15 : SIMULATE   PAD O 1000 SIMULATE ;
```

51

0	0003	EQU	ZERO	0006	EQU	'NEXT
1	0014	EMPLACE	LIT	001F	EMPLACE	[']
2	0025	EMPLACE	EXECUTE	0030	EQU	'VARIABLE'
3	0037	EQU	'USER'	0042	EQU	'CONSTANT'
4	004B	EQU	'DOES>'	005E	EMPLACE	;S
5	0063	EQU	':'	0070	EMPLACE	DO
6	007C	EMPLACE	IF	007F	EMPLACE	END
7	0082	EMPLACE	ELSE	0085	EMPLACE	WHILE
8	0088	EMPLACE	LOOP	008D	EMPLACE	+LOOP
9	0009	LOAD	CR	00C0	EMPLACE	U*
10	00D9	EMPLACE	U/	00FF	EMPLACE	CZ
11	0101	EMPLACE	C@	0109	EMPLACE	C!
12	010F	EMPLACE	O=	011E	EMPLACE	O<
13	0128	EMPLACE	+	0130	EMPLACE	2*
14	0138	EMPLACE	-	0140	EMPLACE	AND
15	0146	ENTRY	BINARY	014E	EMPLACE	1+

52

0	0151	EMPLACE	2+	015D	EMPLACE	MOVE
1	016E	EMPLACE	DUP	0175	EMPLACE	OVER
2	017B	EMPLACE	-DUP	0190	EMPLACE	DROP
3	0195	EMPLACE	SWAP	01A2	EMPLACE	@
4	01AC	EMPLACE	!	01B4	EMPLACE	+!
5	01BE	EMPLACE	<R	01C9	EMPLACE	R>
6	01D4	EMPLACE	I	01E0	EMPLACE	ROT
7	01F2	ENTRY	PUT	01F7	ENTRY	2POP
8	01FB	ENTRY	1POP	0200	ORG	;S
9						
10						
11						
12						
13						
14						
15						

53

0	(	MORE DEFINITIONS)				
1	0	CONSTANT 0	: <	- 0<	; :	> SWAP <
2	1	CONSTANT 1	: =	- 0=	; ' 0=	EMPLACE NOT
3						
4		17 PAGE				
5	CODE	MINUS	BEGIN	S INC	S SEX	O#LD SM STXD
6	O#LD	+C	SM	S STR	NEXT	
7	CODE	ABS	S LDN	SHL	DFL NOT	END NEXT
8						
9						
10						
11						
12						
13						
14						
15						

54

```
0 ( CROSS ILLUSTRATION)
1 NUCLEUS LOAD  PROM LOAD  COLON LOAD  MORE LOAD  CR
2 HEX 1000 RES  DECIMAL
3
4   H: NOTHING  ;           : LITTLE  DROP ;
5 H: LAZY  <BUILDS NOTHING DOES>  LITTLE ;           LAZY ME
6
7 H: ARRAY  0 C VARIABLE  1 - RES ;           100 ARRAY BUFFER
8   BUFFER 100 +
9 : FILL  IN-LINE BUFFER DO 255 I C! LOOP ;
10 : TIMES  0 DO DUP <R EXECUTE R> LOOP DROP ;
11 : TEST  BEGIN ['] FILL 5 TIMES  0 END ;
12
13 HERE ZERO !  HEX 2000  CZ # LD  R PHI  CZ # LD  R PLO
14  F PHI  1FEO  CZ # LD  S PHI  CZ # LD  S PLO  'NEXT # LD
15  F PLO  'TEST  CZ # LD  I PHI  CZ # LD  I PLO  NEXT
```

55

56