

Table of Contents

- [Main Kernal](#)
- [I/O Definitions](#)

Main Kernal#

PRT2C

```
ok
128 0 pall      ATARIVF.FB Scr 0 Dr 0
0
1
2
3
4
5
6
7
8
9
10
11
12
13 ende 123
14
15
    ATARIVF.FB Scr 1 Dr 0
0 \ Atari 8bit VolksForth Kernel           cas02jan07
1 forth definitions
2 : (C [compile] ( ; IMMEDIATE \ : ) ; IMMEDIATE
3
4 $2200 DISPLACE ! \ Memory Start Address = $2200
5 TARGET DEFINITIONS $2200  HERE!
6
7 HEX
8 &01 &126  +THRU
9 decimal
10 \ ASSEMBLER NONRELOCATE
11
12 CR .( Unresolved: )
13 .UNRESOLVED CR CR
14
15 CR .( SAVE-TARGET 6502-FORTH83)
    ATARIVF.FB Scr 2 Dr 0
0 \ FORTH PREAMBLE AND ID      10JAN85BP)           cas11aug06
1
2 ASSEMBLER
3  NOP  0 JMP  HERE 2- >LABEL >COLD
4  NOP  0 JMP  HERE 2- >LABEL >RESTART
5
6  HERE DUP ORIGIN!
7
8
9
10
11
12
13
14
```

```

15
  ATARIVF.FB Scr 3 Dr 0
0 \ USERVARIABLES AND COLDSTART VALUES                cas10jan07
1
2 0 JMP  0 JSR  HERE 2- >LABEL >WAKE  END-CODE
3
4 $0D6 ALLOT
5
6 | CREATE LOGO ," volksFORTH-83 Rev. 3.81.4 10jan07"
7
8
9
10
11
12
13
14
15

```

```

  ATARIVF.FB Scr 4 Dr 0
0 \ ZERO PAGE VARIABLES & NEXT 03APR85BP)              cas11aug06
1 \ Zero Page $A0 - $C8 used
2 A0 DUP      >LABEL RP      2+ \ RP      = $A0
3   DUP      >LABEL UP      2+ \ UP      = $A2
4   DUP      >LABEL PUTA    1+ \ PUTA    = $A4
5   DUP      >LABEL SP      2+ \ SP      = $A5
6   DUP      >LABEL NEXT    \ NEXT    = $A7
7   DUP 5 + >LABEL IP      \ IP      = $AB
8     13 + >LABEL W        \ W        = $BE
9     W 8 + >LABEL N        \ N        = $C6
10
11
12
13
14
15

```

```

  ATARIVF.FB Scr 5 Dr 0
0 \ NEXT, MOVED INTO ZERO PAGE 08APR85BP)
1
2 LABEL BOOTNEXT
3   -1 STA          \ -1 IS DUMMY SP
4   IP )Y LDA  W 1+ STA
5   -1 LDA  W STA  \ -1 IS DUMMY IP
6   CLC IP LDA  2 # ADC  IP STA
7   CS NOT ?[ LABEL WJMP -1 ) JMP ]?
8   IP 1+ INC  WJMP BCS END-CODE
9
10
11
12
13
14
15

```

```

  ATARIVF.FB Scr 6 Dr 0
0 \
1 HERE BOOTNEXT - >LABEL BOOTNEXTLEN
2
3 CODE END-TRACE ( PATCH NEXT FOR TRACE )
4 $A5 # LDA  NEXT $A + STA
5   IP # LDA  NEXT $B + STA

```

```

6 $69 # LDA NEXT $C + STA
7 $02 # LDA NEXT $D + STA
8 NEXT JMP END-CODE
9
10
11
12
13
14
15

```

ATARIVF.FB Scr 7 Dr 0

```

0 \ ;C: NOOP 08jab07cas
1
2 CREATE RECOVER ( -- addr ) ASSEMBLER
3 PLA W STA PLA W 1+ STA
4 W WDEC 0 JMP END-CODE
5
6 HERE 2- >LABEL >RECOVER
7 \ HANDCRAFTED FORWARD REFERENCE FOR JMP COMMAND
8
9 COMPILER ASSEMBLER ALSO DEFINITIONS
10 H : ;C: 0 T RECOVER JSR
11 END-CODE ] H ;
12 TARGET
13 CODE NOOP NEXT HERE 2- ! END-CODE
14
15

```

ATARIVF.FB Scr 8 Dr 0

```

0 \ USER VARIABLES 17MAR84KS) 08jab07cas
1
2 CONSTANT ORIGIN 8 UALLOT DROP
3 \ FOR MULTITASKER
4 \ DATASTACK = $7000
5 \ RETURNSTACK = $7500
6 USER S0 $7000 S0 ! USER R0 $7500 R0 !
7 USER DP USER OFFSET 0 OFFSET !
8 USER BASE &10 BASE ! USER OUTPUT
9 USER INPUT
10 USER ERRORHANDLER \ POINTER FOR ABORT" -CODE
11 USER VOC-LINK
12 USER UDP \ POINTS TO NEXT FREE ADDR IN USER
13
14
15

```

ATARIVF.FB Scr 9 Dr 0

```

0 \ MANIPULATE SYSTEM POINTERS 29JAN85BP)
1
2 CODE SP@ ( -- ADDR)
3 SP LDA N STA SP 1+ LDA N 1+ STA
4 N # LDX
5 LABEL XPUSH
6 SP 2DEC 1 ,X LDA SP )Y STA
7 0 ,X LDA 0 # LDX PUTA JMP END-CODE
8
9 CODE SP! ( ADDR --)
10 SP X) LDA TAX SP )Y LDA
11 SP 1+ STA SP STX 0 # LDX
12 NEXT JMP END-CODE
13

```

14
15

ATARIVF.FB Scr 10 Dr 0

```
0 \  
1 CODE UP@ ( -- ADDR)  
2 UP # LDX XPUSH JMP END-CODE  
3  
4 CODE UP! ( ADDR --) UP # LDX  
5 LABEL XPULL SP )Y LDA 1 ,X STA  
6 DEY SP )Y LDA 0 ,X STA  
7 LABEL (XYDROP 0 # LDX 1 # LDY  
8 LABEL (DROP SP 2INC NEXT JMP  
9 END-CODE RESTRICT
```

10
11
12
13
14
15

ATARIVF.FB Scr 11 Dr 0

```
0 \  
1 CODE RP@ ( -- ADDR )  
2 RP # LDX XPUSH JMP END-CODE  
3  
4 CODE RP! ( ADDR -- )  
5 RP # LDX XPULL JMP END-CODE RESTRICT  
6  
7 CODE >R ( 16B -- )  
8 RP 2DEC SP X) LDA RP X) STA  
9 SP )Y LDA RP )Y STA (DROP JMP  
10 END-CODE RESTRICT
```

11
12
13
14
15

ATARIVF.FB Scr 12 Dr 0

```
0 \  
1 CODE R> ( -- 16B)  
2 SP 2DEC RP X) LDA SP X) STA  
3 RP )Y LDA SP )Y STA  
4 LABEL (RDROP 2 # LDA  
5  
6 LABEL (NRDROP CLC RP ADC RP STA  
7 CS ?[ RP 1+ INC ]?  
8 NEXT JMP END-CODE RESTRICT  
9
```

10
11
12
13
14
15

ATARIVF.FB Scr 13 Dr 0

```
0 \  
1 CODE R@ ( -- 16B)  
2 SP 2DEC RP )Y LDA SP )Y STA RP X) LDA PUTA JMP  
3 END-CODE
```

4

08APR85BP)

cas10jan07

```

5 CODE RDROP      (RDROP HERE 2- !   END-CODE  RESTRICT
6
7 CODE EXIT
8  RP X) LDA  IP STA
9  RP )Y LDA  IP 1+ STA
10 (RDROP JMP   END-CODE
11
12 \ CODE UNNEST
13 \ RP X) LDA  IP STA  RP )Y LDA  IP 1+ STA  (rdrop JMP  end-code
14
15
    ATARIVF.FB Scr 14 Dr 0
0  \ ?EXIT EXECUTE  PERFORM                08APR85BP)                cas08jan07
1
2 CODE ?EXIT      ( FLAG -- )
3  SP X) LDA  SP )Y ORA
4  PHP  SP 2INC  PLP
5  ' EXIT @ BNE  NEXT JMP
6 END-CODE
7
8 CODE EXECUTE    ( ADDR --)
9  SP X) LDA  W STA
10 SP )Y LDA  W 1+ STA
11 SP 2INC      W 1- JMP   END-CODE
12
13 : PERFORM ( ADDR -- )  @ EXECUTE ;
14
15
    ATARIVF.FB Scr 15 Dr 0
0  \ C@  C!                10JAN85BP)                cas08jan07
1
2 CODE C@ ( ADDR -- 8B)
3  SP X) LDA  N STA  SP )Y LDA  N 1+ STA
4 LABEL (C@  0 # LDA  SP )Y STA
5  N X)  LDA  PUTA JMP   END-CODE
6
7 CODE C! ( 16B ADDR --)
8  SP X) LDA  N STA  SP )Y LDA  N 1+ STA
9  INY SP )Y LDA  N X) STA DEY
10 LABEL (2DROP
11 SP LDA  CLC  4 # ADC  SP STA
12  CS ?[ SP 1+ INC  ]?
13 NEXT JMP   END-CODE
14
15
    ATARIVF.FB Scr 16 Dr 0
0  \ @ ! +! ctoggle                08APR85BP)                cas08jan07
1
2 : CTOGGLE      ( 8B ADDR --) UNDER C@ XOR SWAP C! ;
3
4 CODE @ ( ADDR -- 16B)
5  SP X) LDA  N STA  SP )Y LDA  N 1+ STA
6  N )Y LDA  SP )Y STA
7  N X) LDA  PUTA JMP   END-CODE
8
9 CODE ! ( 16B ADDR --)
10 SP X) LDA  N STA  SP )Y LDA  N 1+ STA
11 INY SP )Y LDA  N X) STA
12 INY SP )Y LDA  1 # LDY

```

```

13 LABEL (!
14 N )Y STA (2DROP JMP END-CODE
15
    ATARIVF.FB Scr 17 Dr 0
0 \ +! DROP                24MAY84KS)                cas08jan07
1
2 CODE +! ( N ADDR --)
3 SP X) LDA N STA SP )Y LDA N 1+ STA
4 INY SP )Y LDA CLC N X) ADC N X) STA
5 INY SP )Y LDA 1 # LDY N )Y ADC
6 (! JMP END-CODE
7
8 CODE DROP ( 16B --)
9 (DROP HERE 2- ! END-CODE
10
11
12
13
14
15
    ATARIVF.FB Scr 18 Dr 0
0 \ swap                    cas08jan07
1 CODE SWAP ( 16B1 16B2 -- 16B2 16B1 )
2 SP )Y LDA TAX
3 3 # LDY SP )Y LDA N STA
4 TXA SP )Y STA
5 N LDA 1 # LDY SP )Y STA
6 INY 0 # LDX
7 SP )Y LDA N STA SP X) LDA SP )Y STA
8 DEY
9 N LDA PUTA JMP END-CODE
10
11
12
13
14
15
    ATARIVF.FB Scr 19 Dr 0
0 \ DUP ?DUP                08MAY85BP)                cas08jan07
1
2 CODE DUP ( 16B -- 16B 16B)
3 SP 2DEC
4 3 # LDY SP )Y LDA 1 # LDY SP )Y STA
5 INY SP )Y LDA DEY
6 PUTA JMP END-CODE
7
8 CODE ?DUP ( 16B -- 16B 16B / FALSE)
9 SP X) LDA SP )Y ORA
10 0= ?[ NEXT JMP ]?
11 ' DUP @ JMP END-CODE
12
13 \ : ?DUP ( 16B -- 16B 16B / FALSE) DUP IF DUP THEN ;
14 \ : DUP ( n - n n ) SP@ @ ;
15
    ATARIVF.FB Scr 20 Dr 0
0 \ OVER                    13JUN84KS)                cas08jan07
1
2 CODE OVER ( 16B1 16B2 - 16B1 16B3 16B1)
3 SP 2DEC 4 # LDY SP )Y LDA SP X) STA

```

```
4 INY SP )Y LDA 1 # LDY SP )Y STA
5 NEXT JMP END-CODE
6
7
8 \ \ : ROT >R SWAP R> SWAP ;
9 : OVER >R DUP R> SWAP ;
```

```
10
11
12
13
14
15
```

ATARIVF.FB Scr 21 Dr 0

```
0 \ ROT
1 CODE ROT ( 16B1 16B2 16B3 -- 16B2 16B3 16B1)
2 3 # LDY SP )Y LDA N 1+ STA
3 1 # LDY SP )Y LDA 3 # LDY SP )Y STA
4 5 # LDY SP )Y LDA N STA
5 N 1+ LDA SP )Y STA
6 1 # LDY N LDA SP )Y STA
7 INY SP )Y LDA N 1+ STA
8 SP X) LDA SP )Y STA
9 4 # LDY SP )Y LDA SP X) STA
10 N 1+ LDA SP )Y STA
11 1 # LDY NEXT JMP END-CODE
```

cas08jan07

```
12
13
14
15
```

ATARIVF.FB Scr 22 Dr 0

```
0 \ -ROT NIP UNDER PICK ROLL -ROLL
1 : -ROT ( 16B1 16B2 16B3 -- 16B3 16B1 16B2)
2 ROT ROT ;
3
4 : NIP ( 16B1 16B2 -- 16B2) SWAP DROP ;
5
6 : UNDER ( 16B1 16B2 -- 16B2 16B1 16B2) SWAP OVER ;
7
8 : PICK ( N -- 16B.N ) 1+ 2* SP@ + @ ;
9
10 : ROLL ( N --) DUP >R PICK SP@ DUP 2+ R> 1+ 2* CMOVE> DROP ;
11
12 : -ROLL ( N --)
13 >R DUP SP@ DUP 2+ DUP 2+ SWAP
14 R@ 2* CMOVE R> 1+ 2* + ! ;
```

cas08jan07

```
15
```

ATARIVF.FB Scr 23 Dr 0

```
0 \ DOUBLE WORD STACK MANIP. 21APR83KS)
1
2 : 2SWAP ( 32B1 32B2 -- 32B2 32B1) ROT >R ROT R> ;
3
4 CODE 2DROP ( 32B -- )
5 (2DROP HERE 2- ! END-CODE
6
7 : 2DUP ( 32B -- 32B 32B) OVER OVER ;
8
9 \ : 2DROP ( 32B -- ) DROP DROP ;
```

```
10
11
```

```

12
13
14
15
  ATARIVF.FB Scr 24 Dr 0
0 \ + AND OR XOR          08APR85BP)          cas08jan07
1 COMPILER ASSEMBLER ALSO DEFINITIONS
2
3 H : DYADOP ( OPCODE --) T
4   INY SP X) LDA  DUP C, SP C, SP )Y STA
5   DEY SP )Y LDA  3 # LDY  C, SP C, SP )Y STA
6   (XYDROP JMP  H ;
7 TARGET
8
9 CODE +      ( N1 N2 -- N3) CLC      $71 DYADOP  END-CODE
10 CODE OR    ( 16B1 16B2 -- 16B3)  $11 DYADOP  END-CODE
11 CODE AND   ( 16B1 16B2 -- 16B3)  $31 DYADOP  END-CODE
12 CODE XOR   ( 16B1 16B2 -- 16B3)  $51 DYADOP  END-CODE
13
14
15
  ATARIVF.FB Scr 25 Dr 0
0 \ - NOT NEGATE          24DEC83KS)          cas08jan07
1
2 CODE -      ( N1 N2 -- N3)
3   INY SP )Y LDA  SEC  SP X) SBC  SP )Y STA  INY  SP )Y LDA
4   1 # LDY  SP )Y SBC  3 # LDY  SP )Y STA  (XYDROP JMP  END-CODE
5
6 CODE NOT    ( 16B1 -- 16B2)  CLC
7 LABEL (NOT TXA  SP X) SBC  SP X) STA  TXA SP )Y SBC  SP )Y STA
8   NEXT JMP  END-CODE
9
10 CODE NEGATE ( N1 -- N2 ) SEC  (NOT BCS  END-CODE
11
12 \ : -      NEGATE + ;
13
14
15
  ATARIVF.FB Scr 26 Dr 0
0 \ DNEGATE SETUP        14JUN84KS)          cas08jan07
1
2 CODE DNEGATE ( D1 -- -D1)
3   INY  SEC
4   TXA  SP )Y SBC  SP )Y STA  INY
5   TXA  SP )Y SBC  SP )Y STA
6   TXA  SP X) SBC  SP X) STA  1 # LDY
7   TXA  SP )Y SBC  SP )Y STA
8   NEXT JMP  END-CODE
9 LABEL SETUP ( QUAN  IN A)
10 .A ASL  TAX  TAY  DEY
11   [[ SP )Y LDA  N ,Y STA  DEY  0< ?]
12 TXA  CLC  SP ADC  SP STA
13   CS ?[ SP 1+ INC  ]?
14 0 # LDX  1 # LDY  RTS  END-CODE
15
  ATARIVF.FB Scr 27 Dr 0
0 \ D+
1
2 CODE D+      ( D1 D2 -- D3)

```

```

3 2 # LDA SETUP JSR INY
4 SP )Y LDA CLC N 2+ ADC SP )Y STA INY
5 SP )Y LDA N 3 + ADC SP )Y STA
6 SP X) LDA N ADC SP X) STA 1 # LDY
7 SP )Y LDA N 1+ ADC SP )Y STA
8 NEXT JMP END-CODE
9
10
11
12
13
14
15

```

ATARIVF.FB Scr 28 Dr 0

```

0 \ 1+ 2+ 3+ 4+ 1- 2- 08APR85BP) cas08jan07
1
2 CODE 1+ ( N1 -- N2) 1 # LDA
3 LABEL N+ CLC SP X) ADC
4 CS NOT ?[ PUTA JMP ]?
5 SP X) STA SP )Y LDA 0 # ADC SP )Y STA
6 NEXT JMP END-CODE
7
8 CODE 2+ ( N1 -- N2) 2 # LDA N+ BNE END-CODE
9 CODE 3+ ( N1 -- N2) 3 # LDA N+ BNE END-CODE
10 CODE 4+ ( N1 -- N2) 4 # LDA N+ BNE END-CODE
11 | CODE 6+ ( N1 -- N2) 6 # LDA N+ BNE END-CODE
12
13
14
15

```

ATARIVF.FB Scr 29 Dr 0

```

0 \ 1- 2- NUMBER CONSTANTS 24DEC83KS) cas08jan07
1 CODE 1- ( N1 -- N2) SEC
2 LABEL (1- SP X) LDA 1 # SBC
3 CS ?[ PUTA JMP ]?
4 SP X) STA SP )Y LDA 0 # SBC SP )Y STA
5 NEXT JMP END-CODE
6 CODE 2- ( N1 -- N2) CLC (1- BCC END-CODE
7
8 -1 CONSTANT TRUE 0 CONSTANT FALSE
9 ' TRUE ALIAS -1 ' FALSE ALIAS 0
10
11 1 CONSTANT 1 2 CONSTANT 2
12 3 CONSTANT 3 4 CONSTANT 4
13
14 : ON ( ADDR -- ) TRUE SWAP ! ;
15 : OFF ( ADDR -- ) FALSE SWAP ! ;

```

ATARIVF.FB Scr 30 Dr 0

```

0 \ WORDS FOR NUMBER LITERALS 24MAY84KS) cas08jan07
1
2 CODE CLIT ( -- 8B)
3 SP 2DEC IP X) LDA SP X) STA TXA SP )Y STA IP WINC
4 NEXT JMP END-CODE RESTRICT
5
6 CODE LIT ( -- 16B)
7 SP 2DEC IP )Y LDA SP )Y STA IP X) LDA SP X) STA
8 LABEL (BUMP IP 2INC NEXT JMP END-CODE RESTRICT
9
10 : LITERAL ( 16B --) DUP $FF00 AND

```

```

11 IF COMPILE LIT , EXIT THEN COMPILE CLIT C, ;
12 IMMEDIATE RESTRICT
13
14 \\ : LIT R> DUP 2+ >R @ ;
15 \\ : CLIT R> DUP 1+ >R C@ ;
    ATARIVF.FB Scr 31 Dr 0
0 \ COMPARISION CODE WORDS 13JUN84KS) cas08jan07
1 CODE 0< ( N -- FLAG) SP )Y LDA 0< ?[
2 LABEL PUTTRUE $FF # LDA $24 C, ]?
3 LABEL PUTFALSE TXA SP )Y STA
4 PUTA JMP END-CODE
5
6 CODE 0= ( 16B -- FLAG)
7 SP X) LDA SP )Y ORA PUTTRUE BEQ PUTFALSE BNE END-CODE
8
9 CODE UWITHIN ( U1 [LOW UP[ -- FLAG)
10 2 # LDA SETUP JSR 1 # LDY SP X) LDA N CMP
11 SP )Y LDA N 1+ SBC
12 CS NOT ?[ ( N>SP) SP X) LDA N 2+ CMP
13 SP )Y LDA N 3 + SBC
14 PUTTRUE BCS ]?
15 PUTFALSE JMP END-CODE
    ATARIVF.FB Scr 32 Dr 0
0 \ COMPARISION CODE WORDS 13JUN84KS)
1
2 CODE < ( N1 N2 -- FLAG)
3 SP X) LDA N STA SP )Y LDA N 1+ STA
4 SP 2INC
5 N 1+ LDA SP )Y EOR ' 0< @ BMI
6 SP X) LDA N CMP SP )Y LDA N 1+ SBC
7 ' 0< @ 2+ JMP END-CODE
8
9 CODE U< ( U1 U2 -- FLAG)
10 SP X) LDA N STA SP )Y LDA N 1+ STA
11 SP 2INC
12 SP X) LDA N CMP SP )Y LDA N 1+ SBC
13 CS NOT ?[ PUTTRUE JMP ]?
14 PUTFALSE JMP END-CODE
15
    ATARIVF.FB Scr 33 Dr 0
0 \ COMPARISION WORDS 24DEC83KS) cas08jan07
1
2 \ : 0< $8000 AND 0<> ;
3
4 : > ( N1 N2 -- FLAG) SWAP < ;
5 : 0> ( N -- FLAG) NEGATE 0< ;
6 : 0<> ( N -- FLAG) 0= NOT ;
7 : U> ( U1 U2 -- FLAG) SWAP U< ;
8 : = ( N1 N2 -- FLAG) - 0= ;
9 : D0= ( D -- FLAG) OR 0= ;
10 : D= ( D1 D2 -- FLAG) DNEGATE D+ D0= ;
11 : D< ( D1 D2 -- FLAG) ROT 2DUP -
12 IF > NIP NIP ELSE 2DROP U< THEN ;
13
14
15
    ATARIVF.FB Scr 34 Dr 0
0 \ MIN MAX UMAX UMIN EXTEND DABS ABS KS)
1

```

```

2 | : MINIMAX ( N1 N2 FLAG -- N3)
3   RDROP IF SWAP THEN DROP ;
4
5 : MIN ( N1 N2 -- N3) 2DUP > MINIMAX ; -2 ALLOT
6 : MAX ( N1 N2 -- N3) 2DUP < MINIMAX ; -2 ALLOT
7 : UMAX ( U1 U2 -- U3) 2DUP U< MINIMAX ; -2 ALLOT
8 : UMIN ( U1 U2 -- U3) 2DUP U> MINIMAX ; -2 ALLOT
9
10 : EXTEND ( N -- D) DUP 0< ;
11
12 : DABS ( D -- UD) EXTEND IF DNEGATE THEN ;
13 : ABS ( N -- U) EXTEND IF NEGATE THEN ;
14
15
    ATARIVF.FB Scr 35 Dr 0
0 \ LOOP PRIMITIVES 08FEB85BP/KS) cas08jan07
1
2 | : DODO RDROP R> 2+ DUP >R ROT >R SWAP >R >R ;
3
4 : (DO ( LIMIT STAR -- ) OVER - DODO ; -2 ALLOT RESTRICT
5
6 : (?DO ( LIMIT START -- )
7 OVER - ?DUP IF DODO THEN R> DUP @ + >R DROP ; RESTRICT
8
9 : BOUNDS ( START COUNT -- LIMIT START ) OVER + SWAP ;
10
11 CODE ENDLLOOP 6 # LDA (NRDROP JMP END-CODE RESTRICT
12
13 \\ DODO PUTS "INDEX \ LIMIT \
14 ADR.OF.DO" ON RETURN-STACK
15
    ATARIVF.FB Scr 36 Dr 0
0 \ (LOOP (+LOOP 08APR85BP)
1 CODE (LOOP
2 CLC 1 # LDA RP X) ADC RP X) STA
3 CS ?[ RP )Y LDA 0 # ADC RP )Y STA
4 CS ?[ NEXT JMP ]? ]?
5 LABEL DOLOOP 5 # LDY
6 RP )Y LDA IP 1+ STA DEY
7 RP )Y LDA IP STA 1 # LDY
8 NEXT JMP END-CODE RESTRICT
9
10 CODE (+LOOP
11 CLC SP X) LDA RP X) ADC RP X) STA
12 SP )Y LDA RP )Y ADC RP )Y STA
13 .A ROR SP )Y EOR
14 PHP SP 2INC PLP DOLOOP BPL
15 NEXT JMP END-CODE RESTRICT
    ATARIVF.FB Scr 37 Dr 0
0 \ LOOP INDICES 08APR85BP)
1
2 CODE I ( -- N) 0 # LDY
3 LABEL LOOPINDEX SP 2DEC CLC
4 RP )Y LDA INY INY
5 RP )Y ADC SP X) STA DEY
6 RP )Y LDA INY INY
7 RP )Y ADC 1 # LDY SP )Y STA
8 NEXT JMP END-CODE RESTRICT
9

```

```

10 CODE J ( -- N)
11 6 # LDY LOOPINDEX BNE
12          END-CODE RESTRICT
13
14
15
    ATARIVF.FB Scr 38 Dr 0
0 \ BRANCHING                24DEC83KS)
1
2 CODE BRANCH
3 CLC IP LDA IP X) ADC N STA
4 IP 1+ LDA IP )Y ADC IP 1+ STA N LDA IP STA
5 NEXT JMP          END-CODE RESTRICT
6
7 CODE ?BRANCH
8 SP X) LDA SP )Y ORA PHP SP 2INC PLP
9 ' BRANCH @ BEQ (BUMP JMP END-CODE RESTRICT
10
11 \\ : BRANCH R> DUP @ + >R ; RESTRICT
12
13 : ?BRANCH
14 0= R> OVER NOT OVER 2+ AND -ROT
15 DUP @ + AND OR >R ; RESTRICT
    ATARIVF.FB Scr 39 Dr 0
0 \ RESOLVE LOOPS AND BRANCHES 03FEB85BP)                cas11aug06
1
2 : >MARK ( -- ADDR) HERE 0 , ;
3 : >RESOLVE ( ADDR --) HERE OVER - SWAP ! ;
4 : <MARK ( -- ADDR) HERE ;
5 : <RESOLVE ( ADDR --) HERE - , ;
6 : ?PAIRS ( N1 N2 -- ) - ABORT" UNSTRUCTURED" ;
7
8
9
10
11
12
13
14
15
    ATARIVF.FB Scr 40 Dr 0
0 \ CASE?                    04MAY85BP)                cas08jan07
1
2 LABEL PUSHA
3 0 # CMP 0< ?[ PHA OFF # LDA ][
4 LABEL PUSH0A PHA 0 # LDA ]?
5 LABEL PUSH TAX SP 2DEC
6 TXA 1 # LDY SP )Y STA
7 PLA 0 # LDX PUTA JMP
8
9 CODE CASE?
10 ( 16B1 16B2 -- 16B1 FALSE / TRUE )
11 1 # LDA SETUP JSR N LDA SP X) CMP
12 0= ?[ N 1+ LDA SP )Y CMP 0= ?[ PUTTRUE JMP ]? ]?
13 TXA PUSH0A JMP END-CODE
14 \\ : CASE? ( 16B1 16B2 -- 16B1 f )
15 OVER = DUP IF NIP THEN ;
    ATARIVF.FB Scr 41 Dr 0
0 \ BRANCHING                03FEB85BP)                cas08jan07

```

```

1
2 : IF      COMPILER ?BRANCH >MARK 1 ;           IMMEDIATE RESTRICT
3 : THEN    ABS 1 ?PAIRS >RESOLVE ;             IMMEDIATE RESTRICT
4 : ELSE    1 ?PAIRS COMPILER BRANCH >MARK
5           SWAP >RESOLVE -1 ;                 IMMEDIATE RESTRICT
6 : BEGIN   <MARK 2 ;                           IMMEDIATE RESTRICT
7 : WHILE   2 ?PAIRS 2 COMPILER ?BRANCH
8           >MARK -2 2SWAP ;                   IMMEDIATE RESTRICT
9
10 | : (REPTIL <RESOLVE BEGIN DUP -2
11     = WHILE DROP >RESOLVE REPEAT ;
12
13 : REPEAT 2 ?PAIRS COMPILER BRANCH (REPTIL ; IMMEDIATE RESTRICT
14 : UNTIL  2 ?PAIRS COMPILER ?BRANCH (REPTIL ; IMMEDIATE RESTRICT
15
    ATARIVF.FB Scr 42 Dr 0
0 \ LOOPS                                29JAN85KS/BP)
1
2 : DO      COMPILER (DO >MARK 3 ;           IMMEDIATE RESTRICT
3
4 : ?DO     COMPILER (?DO >MARK 3 ;           IMMEDIATE RESTRICT
5
6 : LOOP    3 ?PAIRS COMPILER (LOOP
7           COMPILER ENDLLOOP >RESOLVE ; IMMEDIATE RESTRICT
8
9 : +LOOP   3 ?PAIRS COMPILER (+LOOP
10          COMPILER ENDLLOOP >RESOLVE ; IMMEDIATE RESTRICT
11
12 : LEAVE   ENDLLOOP R> 2- DUP @ + >R ;           RESTRICT
13
14 \\ RETURNSTACK: CALLADR \ INDEX
15          LIMIT \ ADR OF DO
    ATARIVF.FB Scr 43 Dr 0
0 \ UM*                                BP/KS13.2.85)                cas02jan07
1 CODE UM* ( U1 U2 -- UD)
2 SP )Y LDA N STA SP X) LDA N 1+ STA
3 INY N 2+ STX N 3+ STX $10 # LDX
4 [[ N 3+ ASL N 2+ ROL N 1+ ROL N ROL
5 CS ?[ CLC SP )Y LDA N 3+ ADC N 3+ STA
6          INY SP )Y LDA DEY N 2+ ADC N 2+ STA
7          CS ?[ N 1+ INC 0= ?[ N INC ]? ]? ]?
8          DEX 0= ?]
9 N 3+ LDA SP )Y STA INY N 2+ LDA SP )Y STA 1 # LDY
10 N LDA SP )Y STA N 1+ LDA SP X) STA
11 NEXT JMP END-CODE
12
13 \\ : UM* ( U1 U2 -- UD3) >R 0 0 0 R> $10 0
14 DO DUP 2/ >R 1 AND IF 2OVER D+ THEN
15 >R >R 2DUP D+ R> R> R> LOOP DROP 2SWAP 2DROP ;
    ATARIVF.FB Scr 44 Dr 0
0 \ M* 2*                                04JUL84KS)                cas02jan07
1
2 : M*      ( N1 N2 -- D)
3 DUP 0< DUP >R IF NEGATE THEN
4 SWAP DUP 0< IF NEGATE R> NOT >R THEN
5 UM* R> IF DNEGATE THEN ;
6
7 : *      ( N N -- PROD) UM* DROP ;
8

```

```

9 CODE 2* ( N1 -- N2)
10 SP X) LDA .A ASL SP X) STA
11 SP )Y LDA .A ROL SP )Y STA
12 NEXT JMP END-CODE
13

```

```

14 \\ | : 2* DUP + ;
15

```

```

ATARIVF.FB Scr 45 Dr 0

```

```

0 \ UM/MOD 04JUL84KS)

```

```

cas02jan07

```

```

1
2 | : DIVOVL
3 TRUE ABORT" DIVISION OVERFLOW" ;
4

```

```

5 CODE UM/MOD ( UD U -- UREM UQUOTE)
6 SP X) LDA N 5 + STA
7 SP )Y LDA N 4+ STA SP 2INC
8 SP X) LDA N 1+ STA
9 SP )Y LDA N STA INY
10 SP )Y LDA N 3+ STA INY
11 SP )Y LDA N 2+ STA $11 # LDX CLC
12 [[ N 6 + ROR SEC N 1+ LDA N 5 + SBC
13 TAY N LDA N 4+ SBC
14 CS NOT ?[ N 6 + ROL ]?
15 CS ?[ N STA N 1+ STY ]?

```

```

ATARIVF.FB Scr 46 Dr 0

```

```

0 \ um/mod cont.

```

```

cas02jan07

```

```

1
2 N 3 + ROL N 2+ ROL N 1+ ROL N ROL
3 DEX 0= ?]
4 1 # LDY N ROR N 1+ ROR
5 CS ?[ ;C: DIVOVL ; ASSEMBLER ]?
6 N 2+ LDA SP )Y STA INY
7 N 1+ LDA SP )Y STA INY
8 N LDA SP )Y STA 1 # LDY
9 N 3 + LDA

```

```

10 PUTA JMP END-CODE
11
12
13
14
15

```

```

ATARIVF.FB Scr 47 Dr 0

```

```

0 \ 2/ M/MOD 24DEC83KS)

```

```

1
2 : M/MOD ( D N -- MOD QUOT)
3 DUP >R ABS OVER
4 0< IF UNDER + SWAP THEN
5 UM/MOD R@
6 0< IF NEGATE OVER IF SWAP R@ + SWAP 1-
7 THEN THEN RDROP ;
8

```

```

9 CODE 2/ ( N1 -- N2)
10 SP )Y LDA .A ASL
11 SP )Y LDA .A ROR SP )Y STA
12 SP X) LDA .A ROR
13 PUTA JMP END-CODE
14
15

```

```

ATARIVF.FB Scr 48 Dr 0

```

```

0 \ /MOD / MOD */MOD */ U/MOD UD/MOD KS)                cas08jan07
1
2 : /MOD      ( N1 N2 -- REM QUOT)          >R EXTEND R> M/MOD ;
3 : /         ( N1 N2 -- QUOT)             /MOD NIP ;
4 : MOD       ( N1 N2 -- REM)              /MOD DROP ;
5 : */MOD     ( N1 N2 N3 -- REM QUOT)     >R M* R> M/MOD ;
6 : */        ( N1 N2 N3 -- QUOT)        */MOD NIP ;
7 : U/MOD     ( U1 U2  -- UREM UQUOT)     0 SWAP UM/MOD ;
8 : UD/MOD    ( UD1 U2 -- UREM UDQUOT)
9         >R 0 R@ UM/MOD R> SWAP >R UM/MOD R> ;
10
11
12
13
14
15

```

ATARIVF.FB Scr 49 Dr 0

```

0 \ CMOVE      BP 08APR85)                cas08jan07
1
2 CODE CMOVE  ( FROM TO QUAN --)
3 3 # LDA SETUP JSR DEY
4 [[ [[ N CPY 0= ?[ N 1+ DEC 0< ?[
5         1 # LDY NEXT JMP ]? ]?
6 N 4 + )Y LDA N 2+ )Y STA INY 0= ?]
7 N 5 + INC N 3 + INC ]] END-CODE
8
9
10
11
12
13
14
15

```

ATARIVF.FB Scr 50 Dr 0

```

0 \ CMOVE> (CMOVE>                cas08jan07
1 CODE CMOVE> ( FROM TO QUAN --)
2 3 # LDA SETUP JSR
3 CLC N 1+ LDA N 3 + ADC N 3 + STA
4 CLC N 1+ LDA N 5 + ADC N 5 + STA
5 N 1+ INC N LDY CLC CS ?[
6 LABEL (CMOVE>
7 DEY N 4 + )Y LDA N 2+ )Y STA ]?
8 TYA (CMOVE> BNE
9 N 3 + DEC N 5 + DEC N 1+ DEC
10 (CMOVE> BNE 1 # LDY
11 NEXT JMP END-CODE
12
13 : MOVE      ( FROM TO QUAN --) >R 2DUP U< IF R> CMOVE> EXIT THEN
14           R> CMOVE ;
15

```

ATARIVF.FB Scr 51 Dr 0

```

0 \ PLACE COUNT ERASE      16FEB85BP/KS)        cas08jan07
1
2 : PLACE ( ADDR LEN TO --) OVER >R ROT OVER 1+ R> MOVE C! ;
3
4 CODE COUNT ( ADDR -- ADDR+1 LEN)
5 SP X) LDA N STA CLC 1 # ADC SP X) STA
6 SP )Y LDA N 1+ STA 0 # ADC SP )Y STA
7 SP 2DEC (C@ JMP END-CODE

```

```

8
9 \ : COUNT ( ADR -- ADR+1 LEN ) DUP 1+ SWAP C@ ;
10
11 : ERASE ( ADDR QUAN --)      0 FILL ;
12
13
14
15
    ATARIVF.FB Scr 52 Dr 0
0 \ FILL                          11JUN85BP)
1
2 CODE FILL ( ADDR QUAN 8B -- )
3 3 # LDA SETUP JSR DEY
4 N LDA N 3 + LDX
5 0<> ?[ [[ [[ N 4 + )Y STA INY 0= ?]
6         N 5 + INC DEX 0= ?]
7     ]? N 2+ LDX
8 0<> ?[ [[ N 4 + )Y STA INY DEX 0= ?]
9     ]? 1 # LDY
10 NEXT JMP END-CODE
11
12 \\ : FILL ( ADDR QUAN 8B --) SWAP ?DUP
13     IF >R OVER C! DUP 1+ R> 1- CMOVE EXIT THEN 2DROP ;
14
15
    ATARIVF.FB Scr 53 Dr 0
0 \ HERE PAD ALLOT , C, COMPILE 24DEC83KS)          cas08jan07
1
2 : HERE ( -- ADDR) DP @ ;
3
4 : PAD ( -- ADDR) HERE $42 + ;
5
6 : ALLOT ( N --) DP +! ;
7
8 : , ( 16B --) HERE ! 2 ALLOT ;
9
10 : C, ( 8B --) HERE C! 1 ALLOT ;
11
12 : COMPILE R> DUP 2+ >R @ , ; RESTRICT
13
14
15
    ATARIVF.FB Scr 54 Dr 0
0 \ INPUT STRINGS 24DEC83KS)          cas09jan07
1
2 VARIABLE #TIB 0 #TIB !
3 VARIABLE >TIB $100 >TIB ! \ $80 ALLOT
4 VARIABLE >IN 0 >IN !
5 VARIABLE SPAN 0 SPAN !
6
7 : TIB ( -- ADDR ) >TIB @ ;
8
9 : QUERY TIB $80 EXPECT SPAN @ #TIB ! >IN OFF ;
10
11
12
13
14
15

```

```

    ATARIVF.FB Scr 55 Dr 0
0 \ SCAN SKIP /STRING          12OCT84BP)          cas08jan07
1 \ todo: combine scan and skip!
2
3 : SCAN ( ADDR0 LEN0 CHAR -- ADDR1 LEN1) >R
4     BEGIN DUP WHILE OVER C@ R@ -
5     WHILE 1- SWAP 1+ SWAP REPEAT RDROP ;
6
7 : SKIP ( ADDR LEN DEL -- ADDR1 LEN1) >R
8     BEGIN DUP WHILE OVER C@ R@ =
9     WHILE 1- SWAP 1+ SWAP REPEAT RDROP ;
10
11
12 : /STRING ( ADDR0 LEN0 +N - ADDR1 LEN1)
13     OVER UMIN ROT OVER + -ROT - ;
14
15

```

```

    ATARIVF.FB Scr 56 Dr 0
0 \ CAPITAL          03APR85BP)          cas08jan07
1
2 LABEL (CAPITAL \ FOR ASCII ONLY
3 ASCII a # CMP
4 CS ?[ ASCII z 1+ # CMP
5     CC ?[     SEC ASCII a ASCII A - # SBC
6     ]? ]? RTS END-CODE
7
8 CODE CAPITAL ( CHAR -- CHAR' )
9 SP X) LDA (CAPITAL JSR SP X) STA NEXT JMP END-CODE
10
11
12
13
14
15

```

```

    ATARIVF.FB Scr 57 Dr 0
0 \ CAPITALIZE          03APR85BP)          cas08jan07
1
2 CODE CAPITALIZE ( STRING -- STRING )
3 SP X) LDA N STA SP )Y LDA N 1+ STA
4 N X) LDA N 2+ STA DEY
5 [[ N 2+ CPY 0= ?[ 1 # LDY NEXT JMP ]?
6     INY N )Y LDA (CAPITAL JSR N )Y STA
7 ]] END-CODE
8
9 \\ : CAPITALIZE ( STRING -- STRING )
10     DUP COUNT BOUNDS ?DO I C@ CAPITAL I C! THEN LOOP ;
11
12 \\ CAPITAL ( CHAR -- CHAR )
13     ASCII A ASCII Z 1+ UWITHIN
14     IF I C@ [ ASCII A ASCII A - ] LITERAL - ;
15

```

```

    ATARIVF.FB Scr 58 Dr 0
0 \ (WORD          08APR85BP)
1
2 | CODE (WORD ( CHAR ADR0 LEN0 -- ADR)
3     \ N : LENGTH OF SOURCE
4     \ N+2 : PTR IN SOURCE / NEXT CHAR
5     \ N+4 : STRING START ADDRESS
6     \ N+6 : STRING LENGTH

```

```

7 N 6 + STX          \ 0 =: STRING_LENGTH
8 3 # LDY [[ SP )Y LDA N ,Y STA DEY 0< ?]
9 1 # LDY CLC >IN    LDA N 2+ ADC N 2+ STA
10                \ >IN+ADR0 =: N+2
11 >IN 1+ LDA N 3 + ADC N 3 + STA SEC N LDA >IN SBC N STA
12                \ LEN0->IN =: N
13 N 1+ LDA >IN 1+ SBC N 1+ STA
14 CC ?[ SP X) LDA >IN STA \ STREAM EXHAUSTED
15 SP )Y LDA >IN 1+ STA
  ATARIVF.FB Scr 59 Dr 0
0 \ (WORD              08APR85BP)
1
2 ][ 4 # LDY [[ N LDA N 1+ ORA          \ SKIP CHAR'S
3      0= NOT ?[[ N 2+ X) LDA SP )Y CMP \ WHILE COUNT <>0
4      0=      ?[[ N 2+ WINC N WDEC ]]?
5      N 2+ LDA N 4 + STA          \ SAVE STRING_START_ADRESS
6      N 3 + LDA N 5 + STA
7      [[ N 2+ X) LDA SP )Y CMP PHP    \ SCAN FOR CHAR
8      N 2+ WINC N WDEC PLP
9      0= NOT ?[[ N 6 + INC          \ COUNT STRING_LENGTH
10     N LDA N 1+ ORA
11     0= ?] ]? ]?                \ FROM COUNT = 0 IN SKIP)
12 SEC 2 # LDY
13     \ ADR_AFTER_STRING - ADR0 =: >IN)
14 N 2+ LDA SP )Y SBC >IN STA INY
15 N 3 + LDA SP )Y SBC >IN 1+ STA
  ATARIVF.FB Scr 60 Dr 0
0 \ (WORD              08APR85BP)                cas08jan07
1
2 ]? \ FROM 1ST ][, STREAM WAS EXHAUSTED
3     \ WHEN WORD CALLED)
4 CLC 4 # LDA SP ADC SP STA
5 CS ?[ SP 1+ INC ]? \ 2DROP
6 USER' DP # LDY UP )Y LDA
7 SP X) STA N STA INY
8 UP )Y LDA 1 # LDY
9 SP )Y STA N 1+ STA          \ DP @
10 DEY N 6 + LDA \ STORE COUNT BYTE FIRST
11 [[ N )Y STA N 4 + )Y LDA INY
12     N 6 + DEC 0< ?]
13 $20 # LDA N )Y STA          \ ADD A BLANK
14 1 # LDY NEXT JMP END-CODE
15
  ATARIVF.FB Scr 61 Dr 0
0 \ SOURCE WORD PARSE NAME          08APR85BP)                cas21dec05
1
2 : SOURCE ( -- ADDR LEN)
3     TIB #TIB @ ;
4
5 : WORD ( CHAR -- ADDR) SOURCE (WORD ;
6
7 : PARSE ( CHAR -- ADDR LEN) >R SOURCE >IN @ /STRING OVER SWAP
8     R> SCAN >R OVER - DUP R> 0<> - >IN +! ;
9
10 : NAME ( -- ADDR) BL WORD CAPITALIZE EXIT ;
11
12 \\ : WORD ( CHAR -- ADDR) >R
13 SOURCE OVER SWAP >IN @ /STRING R@ SKIP OVER SWAP R>
14 SCAN >R ROT OVER SWAP - R> 0<> - >IN !

```

```

15 OVER - HERE PLACE BL HERE COUNT + C! HERE ;
   ATARIVF.FB Scr 62 Dr 0
0 \ STATE ASCII , " ( " " 24DEC83KS)
1
2 VARIABLE STATE 0 STATE !
3
4 : ASCII BL WORD 1+ C@ STATE @
5     IF [COMPILE] LITERAL THEN ; IMMEDIATE
6
7 : , " ASCII " PARSE HERE OVER 1+ ALLOT PLACE ;
8
9 : "LIT R> R> UNDER COUNT + >R >R ; RESTRICT
10
11 : ( " "LIT ; RESTRICT
12
13 : " COMPILE ( " , " ; IMMEDIATE RESTRICT
14
15
   ATARIVF.FB Scr 63 Dr 0
0 \ ." ( .( \ \ HEX DECIMAL 08SEP84KS) cas08jan07
1
2 : (. " "LIT COUNT TYPE ; RESTRICT
3 : ." COMPILE (. " , " ; IMMEDIATE RESTRICT
4 : ( ASCII ) PARSE 2DROP ; IMMEDIATE
5 : .( ASCII ) PARSE TYPE ; IMMEDIATE
6 : \ >IN @ C/L / 1+ C/L * >IN ! ; IMMEDIATE
7 ' \ ALIAS \
8
9 : \NEEDS NAME FIND NIP IF [COMPILE] \ THEN ;
10
11 : HEX $10 BASE ! ; : DECIMAL $0A BASE ! ;
12
13
14
15
   ATARIVF.FB Scr 64 Dr 0
0 \ NUMBER CONV.: DIGIT? ACCUMULATE KS) cas08jan07
1 : DIGIT? ( CHAR -- DIGIT TRUE/ FALSE )
2 ASCII 0 - DUP 9 U>
3 IF [ ASCII A ASCII 9 - 1- ] LITERAL - DUP 9 U>
4     IF [ 2SWAP ( UNSTRUKTURIERT) ] THEN
5     BASE @ OVER U> ?DUP ?EXIT THEN DROP FALSE ;
6
7 : ACCUMULATE ( +D0 ADR DIGIT - +D1 ADR)
8 SWAP >R SWAP BASE @ UM* DROP ROT BASE @ UM* D+ R> ;
9
10 : CONVERT ( +D1 ADDR0 -- +D2 ADDR2)
11     1+ BEGIN COUNT DIGIT? WHILE ACCUMULATE REPEAT 1- ;
12
13 | : END? ( -- FLAG ) PTR @ 0= ;
14 | : CHAR ( ADDR0 -- ADDR1 CHAR ) COUNT -1 PTR +! ;
15 | : PREVIOUS ( ADDR0 -- ADDR0 CHAR) 1- COUNT ;
   ATARIVF.FB Scr 65 Dr 0
0 \ ?NONUM ?NUM FIXBASE? 13FEB85KS) cas08jan07
1
2 VARIABLE DPL -1 DPL !
3
4 | : ?NONUM ( FLAG -- EXIT IF TRUE )
5     IF RDROP 2DROP DROP RDROP FALSE THEN ;

```

```

6
7 | : ?NUM      ( FLAG -- EXIT IF TRUE )
8   IF RDROP DROP R> IF DNEGATE THEN
9   ROT DROP DPL @ 1+ ?DUP ?EXIT DROP TRUE THEN ;
10 | : FIXBASE? ( CHAR - CHAR FALSE / NEWBASE TRUE )
11   ASCII & CASE? IF $0A TRUE EXIT THEN
12   ASCII $ CASE? IF $10 TRUE EXIT THEN
13   ASCII H CASE? IF $10 TRUE EXIT THEN
14   ASCII % CASE? IF $02 TRUE EXIT THEN FALSE ;
15
  ATARIVF.FB Scr 66 Dr 0
0 \ PUNCTUATION ?DPL PTR          13FEB85KS)          cas08jan07
1
2 | : PUNCTUATION? ( CHAR -- FLAG)
3   ASCII , OVER = SWAP ASCII . = OR ;
4
5 | : ?DPL      DPL @ -1 = ?EXIT 1 DPL +! ;
6
7 | VARIABLE PTR      \ POINTS INTO STRING
8
9
10
11
12
13
14
15
  ATARIVF.FB Scr 67 Dr 0
0 \ (NUMBER NUMBER          13FEB85KS)          cas09jan07
1 : NUMBER? ( STRING - STRING FALSE / N 0< / D 0> )
2   BASE PUSH DUP COUNT PTR ! DPL ON
3   0 >R ( +SIGN)
4   0.0 ROT          END? ?NONUM CHAR
5   ASCII - CASE?
6   IF RDROP TRUE >R END? ?NONUM CHAR THEN FIXBASE?
7   IF BASE !      END? ?NONUM CHAR THEN
8   BEGIN DIGIT? 0= ?NONUM
9   BEGIN ACCUMULATE ?DPL END? ?NUM
10  CHAR DIGIT? 0= UNTIL
11  PREVIOUS PUNCTUATION? 0= ?NONUM
12  DPL OFF END? ?NUM CHAR REPEAT ;
13 DEFER 'NUMBER? ' NUMBER? IS 'NUMBER?
14 : NUMBER ( STRING -- D )
15 'NUMBER? ?DUP 0= ABORT" ?" 0< IF EXTEND THEN ;
  ATARIVF.FB Scr 68 Dr 0
0 \ HIDE REVEAL IMMEDIATE RESTRICT      KS)          cas08jan07
1 VARIABLE LAST      0 LAST !
2
3 | : LAST? ( -- FALSE / ACF TRUE) LAST @ ?DUP ;
4
5 : HIDE      LAST? IF 2- @ CURRENT @ ! THEN ;
6
7 : REVEAL   LAST? IF 2- CURRENT @ ! THEN ;
8
9 : RECURSIVE REVEAL ; IMMEDIATE RESTRICT
10
11 | : FLAG! ( 8B --) LAST? IF UNDER C@ OR OVER C! THEN DROP ;
12
13 : IMMEDIATE $40 FLAG! ;

```

```

14 : RESTRICT $80 FLAG! ;
15
    ATARIVF.FB Scr 69 Dr 0
0 \ CLEARSTACK HALLOT HEAP HEAP?11FEB85BP)          cas08jan07
1
2 CODE CLEARSTACK USER' S0 # LDY
3     UP )Y LDA SP STA INY UP )Y LDA SP 1+ STA
4     1 # LDY NEXT JMP END-CODE
5
6 : HALLOT ( QUAN -- ) S0 @ OVER - SWAP
7     SP@ 2+ DUP ROT - DUP S0 !
8     2 PICK OVER - MOVE CLEARSTACK S0 ! ;
9
10 : HEAP ( -- ADDR) S0 @ 6+ ;
11
12 : HEAP? ( ADDR -- FLAG) HEAP UP@ UWITHIN ;
13
14 | : HEAPMOVE ( FROM -- FROM) DUP HERE OVER -
15     DUP HALLOT HEAP SWAP CMOVE HEAP OVER - LAST +! REVEAL ;
    ATARIVF.FB Scr 70 Dr 0
0 \ DOES> ; 30DEC84KS/BP)          cas08jan07
1
2 LABEL (DODOES> RP 2DEC
3     IP 1+ LDA RP )Y STA IP LDA RP X) STA \ PUT IP ON RP
4     CLC W X) LDA 3 # ADC IP STA
5     TXA W )Y ADC IP 1+ STA \ W@ + 3 -> IP
6 LABEL DOCREATE
7     2 # LDA CLC W ADC PHA TXA W 1+ ADC PUSH JMP END-CODE
8
9 | : (;CODE R> LAST @ NAME> ! ;
10
11 : DOES> COMPILE (;CODE $4C C,
12     COMPILE (DODOES> ; IMMEDIATE RESTRICT
13
14
15
    ATARIVF.FB Scr 71 Dr 0
0 \ 6502-ALIGN ?HEAD \ 08SEP84BP)          cas08jan07
1
2 | : 6502-ALIGN/1 ( ADR -- ADR' ) DUP $FF AND $FF = - ;
3
4 | : 6502-ALIGN/2 ( LFA -- LFA )
5     HERE $FF AND $FF =
6     IF DUP DUP 1+ HERE OVER - 1+ CMOVE> \ LFA NOW INVALID
7     1 LAST +! 1 ALLOT THEN ;
8
9 VARIABLE ?HEAD 0 ?HEAD !
10
11 : | ?HEAD @ ?EXIT -1 ?HEAD ! ;
12
13
14
15
    ATARIVF.FB Scr 72 Dr 0
0 \ WARNING CREATE 30DEC84BP)          cas10jan07
1
2 VARIABLE WARNING 0 WARNING !
3
4 | : EXISTS?

```

```

5     WARNING @ 0= ?EXIT
6     LAST @ CURRENT @ (FIND NIP
7     IF SPACE LAST @ .NAME ." EXISTS " ?CR THEN ;
8
9 : CREATE HERE 0 , CURRENT @@ ,
10    NAME C@ DUP 1 $20 UWITHIN NOT ABORT" INVALID NAME"
11    HERE LAST ! 1+ ALLOT EXISTS? ?HEAD @
12    IF 1 ?HEAD +! DUP 6502-ALIGN/1 , \ POINTER TO CODE
13        HEAPMOVE $20 FLAG! 6502-ALIGN/1 DP !
14    ELSE 6502-ALIGN/2 DROP THEN REVEAL 0 ,
15    ;CODE DCREATE JMP END-CODE
    ATARIVF.FB Scr 73 Dr 0
0 \ NFA?                                30DEC84BP)
1 | CODE NFA? ( VOCABTHREAD CFA -- NFA / FALSE)
2   SP X) LDA N 4 + STA SP )Y LDA N 5 + STA SP 2INC
3   [[ [[ SP X) LDA N 2+ STA SP )Y LDA N 3 + STA
4     N 2+ ORA 0= ?[ PUTFALSE JMP ]?
5     N 2+ )Y LDA SP )Y STA N 1+ STA
6     N 2+ X) LDA SP X) STA N STA
7     N 1+ ORA 0= ?[ NEXT JMP ]? \ N=LINK
8     N 2INC N X) LDA PHA SEC 01F # AND
9     N ADC N STA CS ?[ N 1+ INC ]?
10    PLA 020 # AND 0= NOT
11    ?[ N )Y LDA PHA
12        N X) LDA N STA PLA N 1+ STA ]?
13    N LDA N 4 + CMP 0= ?] \ VOCABTHREAD=0
14    N 1+ LDA N 5 + CMP 0= ?] \ D.H. LEERES VOCABULARY
15    ' 2+ @ JMP END-CODE \ IN NFA? IST ERLAUBT
    ATARIVF.FB Scr 74 Dr 0
0 \ >NAME NAME> >BODY .NAME          03FEB85BP)          cas08jan07
1
2 : >NAME ( CFA -- NFA / FALSE) VOC-LINK
3   BEGIN @ DUP WHILE 2DUP 4 - SWAP
4     NFA? ?DUP IF -ROT 2DROP EXIT THEN REPEAT NIP ;
5
6 | : (NAME> ( NFA -- CFA) COUNT $1F AND + ;
7
8 : NAME> ( NFA -- CFA) DUP (NAME> SWAP C@ $20 AND IF @ THEN ;
9
10 : >BODY ( CFA -- PFA) 2+ ;
11
12 : .NAME ( NFA --)
13   ?DUP IF DUP HEAP? IF ." |" THEN COUNT $1F AND TYPE
14   ELSE ." ???" THEN SPACE ;
15
    ATARIVF.FB Scr 75 Dr 0
0 \ CREATE: : ; CONSTANT VARIABLE      09JAN85KS/BP)          cas10jan07
1
2 : CREATE: CREATE HIDE 0 ] ;
3 : : CREATE: ;CODE HERE >RECOVER ! \ RESOLVE FWD. REFERENCE
4   RP 2DEC IP LDA RP X) STA IP 1+ LDA RP )Y STA
5   W LDA CLC 2 # ADC IP STA TXA W 1+ ADC IP 1+ STA
6   NEXT JMP END-CODE
7
8 : ; 0 ?PAIRS COMPILE EXIT \ exit was unnest
9 [COMPILE] [ REVEAL ; IMMEDIATE RESTRICT
10
11 : CONSTANT ( 16B --) CREATE ,
12   ;CODE SP 2DEC 2 # LDY W )Y LDA SP X) STA INY

```

```

13      W )Y LDA  1 # LDY  SP )Y STA  NEXT JMP  END-CODE
14
15 : VARIABLE  CREATE  2 ALLOT  ;
    ATARIVF.FB Scr 76 Dr 0
0 \ UALLOT USER ALIAS          10JAN85KS/BP)                cas08jan07
1
2 : UALLOT ( QUAN -- OFFSET)
3     DUP UDP @ +  $FF U> ABORT" USERAREA FULL"
4     UDP  @ SWAP UDP +!  ;
5
6 : USER  CREATE  2 UALLOT C,
7     ;CODE  SP 2DEC  2 # LDY W )Y LDA  CLC UP      ADC  SP X) STA
8     TXA  INY UP 1+ ADC  1 # LDY  SP )Y STA  NEXT JMP  END-CODE
9
10 : ALIAS  ( CFA --)
11     CREATE LAST @ DUP C@ $20 AND
12     IF  -2 ALLOT  ELSE  $20 FLAG! THEN  (NAME> !  ;
13
14
15
    ATARIVF.FB Scr 77 Dr 0
0 \ VOC-LINK VP CURRENT CONTEXT ALSO  BP)                cas08jan07
1 CREATE  VP          $10 ALLOT
2
3 VARIABLE CURRENT
4
5 : CONTEXT ( -- ADR  )  VP DUP @ + 2+  ;
6
7 | : THRU.VOCSTACK  ( -- FROM TO ) VP 2+ CONTEXT  ;
8 \ "ONLY FORTH ALSO ASSEMBLER" GIVES VP :
9 \  COUNTWORD = 6 \ONLY\FORTH\ASSEMBLER
10
11 : ALSO      VP @
12 $A > ERROR" VOCABULARY STACK FULL"
13 CONTEXT @  2 VP +!  CONTEXT !  ;
14
15 : TOSS  -2 VP +!  ;
    ATARIVF.FB Scr 78 Dr 0
0 \ VOCABULARY FORTH ONLY FORTH-83 KS/BP)
1
2 : VOCABULARY CREATE  0 , 0 ,
3     HERE VOC-LINK @ ,  VOC-LINK ! DOES>  CONTEXT !  ;
4
5 \ NAME \ CODE \ THREAD \ COLDTHREAD \ VOC-LINK
6
7 VOCABULARY FORTH
8
9 VOCABULARY ONLY
10 ] DOES>  [ ONLYPATCH ]  0 VP ! CONTEXT !  ALSO  ;  ' ONLY !
11
12 : ONLYFORTH  ONLY FORTH ALSO DEFINITIONS  ;
13
14
15
    ATARIVF.FB Scr 79 Dr 0
0 \ DEFINITIONS ORDER WORDS  13JAN84BP/KS)
1
2 : DEFINITIONS  CONTEXT @ CURRENT !  ;
3

```

```

4 | : .VOC ( ADR -- ) @ 2- >NAME .NAME ;
5
6 : ORDER
7 THRU.VOCSTACK DO I .VOC -2 +LOOP 2 SPACES CURRENT .VOC ;
8
9 : WORDS CONTEXT @
10 BEGIN @ DUP STOP? 0= AND
11 WHILE ?CR DUP 2+ .NAME SPACE REPEAT DROP ;
12
13
14
15
    ATARIVF.FB Scr 80 Dr 0
0 \ (FIND                                08APR85BP)
1
2 CODE (FIND ( STRING THREAD
3     -- STRING FALSE / NAMEFIELD TRUE)
4 3 # LDY [[ SP )Y LDA N ,Y STA DEY 0< ?]
5 N 2+ X) LDA 01F # AND N 4 + STA
6 LABEL FINDLOOP 0 # LDY
7 N )Y LDA TAX INY
8 N )Y LDA N 1+ STA N STX N ORA
9 0= ?[ 1 # LDY 0 # LDX PUTFALSE JMP ]?
10 INY N )Y LDA 01F # AND N 4 + CMP
11 FINDLOOP BNE \ COUNTBYTE MATCH
12 CLC 2 # LDA N ADC N 5 + STA
13 0 # LDA N 1+ ADC N 6 + STA
14 N 4 + LDY
15 [[ N 2+ )Y LDA N 5 + )Y CMP
    ATARIVF.FB Scr 81 Dr 0
0 \ FIND (cont.)
1 FINDLOOP BNE DEY 0= ?]
2 3 # LDY N 6 + LDA SP )Y STA DEY
3 N 5 + LDA SP )Y STA
4 DEY 0 # LDX PUTTRUE JMP END-CODE
5
6
7
8
9
10
11
12
13
14
15
    ATARIVF.FB Scr 82 Dr 0
0 \ FOUND                                29JAN85BP)
1
2 | CODE FOUND ( NFA -- CFA N )
3 SP X) LDA N STA SP )Y LDA N 1+ STA
4 N X) LDA N 2+ STA $1F # AND SEC N ADC N STA
5 CS ?[ N 1+ INC ]?
6 N 2+ LDA $20 # AND
7 0= ?[ N LDA SP X) STA N 1+ LDA
8 ][ N X) LDA SP X) STA N )Y LDA ]? SP )Y STA
9 SP 2DEC N 2+ LDA 0< ?[ INY ]?
10 .A ASL
11 0< NOT ?[ TYA $FF # EOR TAY INY ]?

```

cas08jan07

cas08jan07

```

12 TYA SP X) STA
13 0< ?[ $FF # LDA 24 C, ]?
14 TXA 1 # LDY SP )YSTA
15 NEXT JMP END-CODE
   ATARIVF.FB Scr 83 Dr 0
0  \ \
1
2 | : FOUND ( NFA -- CFA N )
3     DUP C@ >R (NAME>
4         R@ $20 AND IF @ THEN
5         -1 R@ $80 AND IF 1- THEN
6         R> $40 AND IF NEGATE THEN ;
7
8
9
10
11
12
13
14
15
   ATARIVF.FB Scr 84 Dr 0
0  \ FIND ' [' 13JAN85BP)
1
2 : FIND ( STRING -- CFA N / STRING FALSE)
3     CONTEXT DUP @ OVER 2- @ = IF 2- THEN
4     BEGIN UNDER @ (FIND IF NIP FOUND EXIT THEN
5         OVER VP 2+ U>
6         WHILE SWAP 2- REPEAT NIP FALSE ;
7
8 : ' ( -- CFA ) NAME FIND 0= ABORT" WHAT?" ;
9
10 : [COMPILE] ' , ; IMMEDIATE RESTRICT
11
12 : [' ' [COMPILE] LITERAL ; IMMEDIATE RESTRICT
13
14 : NULLSTRING? ( STRING -- STRING FALSE / TRUE)
15     DUP C@ 0= DUP IF NIP THEN ;
   ATARIVF.FB Scr 85 Dr 0
0  \ >INTERPRET 28FEB85BP)
1
2 LABEL JUMP
3 INY CLC W )Y LDA 2 # ADC IP STA
4 INY W )Y LDA 0 # ADC IP 1+ STA
5 1 # LDY NEXT JMP END-CODE
6 VARIABLE >INTERPRET
7
8 JUMP ' >INTERPRET !
9
10 \ \ MAKE VARIABLE >INTERPRET TO SPECIAL
11     DEFER
12
13
14
15
   ATARIVF.FB Scr 86 Dr 0
0  \ INTERPRET INTERACTIVE 31DEC84KS/BP)
1
2 DEFER NOTFOUND

```

```

3
4 : NO.EXTENSIONS ( STRING -- ) ERROR" WHAT?" ; \ STRING NOT 0
5
6 ' NO.EXTENSIONS IS NOTFOUND
7
8 : INTERPRET >INTERPRET ; -2 ALLOT
9
10 | : INTERACTIVE ?STACK NAME FIND ?DUP
11     IF 1 AND IF EXECUTE >INTERPRET THEN
12         ABORT" COMPILE ONLY" THEN NULLSTRING? ?EXIT NUMBER?
13     0= IF NOTFOUND THEN >INTERPRET ; -2 ALLOT
14
15 ' INTERACTIVE >INTERPRET !
    ATARIVF.FB Scr 87 Dr 0
0 \ COMPILING [ ]                20DEC84BP)                cas08jan07
1
2 | : COMPILING
3 ?STACK NAME FIND ?DUP
4 IF 0> IF EXECUTE >INTERPRET THEN
5     , >INTERPRET THEN
6 NULLSTRING? ?EXIT 'NUMBER? ?DUP
7     IF 0> IF SWAP [COMPILE] LITERAL THEN
8         [COMPILE] LITERAL
9     ELSE NOTFOUND THEN >INTERPRET ; -2 ALLOT
10
11 : [ [' ] INTERACTIVE IS >INTERPRET STATE OFF ; IMMEDIATE
12
13 : ] [' ] COMPILING IS >INTERPRET STATE ON ;
14
15
    ATARIVF.FB Scr 88 Dr 0
0 \ PERFORM DEFER IS                03FEB85BP)                cas08jan07
1
2 | : CRASH TRUE ABORT" CRASH" ;
3
4 : DEFER CREATE [' ] CRASH ,
5     ;CODE 2 # LDY W )Y LDA PHA INY W )Y LDA
6     W 1+ STA PLA W STA 1 # LDY W 1- JMP END-CODE
7
8 : (IS R> DUP 2+ >R @ ! ;
9
10 | : DEF? ( CFA -- ) @ [' ] NOTFOUND @ OVER =
11     SWAP [' ] >INTERPRET @ = OR NOT ABORT" NOT DEFERRED" ;
12
13 : IS ( ADR -- ) ' DUP DEF? >BODY
14     STATE @ IF COMPILE (IS , EXIT THEN ! ; IMMEDIATE
15
    ATARIVF.FB Scr 89 Dr 0
0 \ ?STACK                08SEP84KS)                cas08jan07
1 | CREATE ALARM 1 ALLOT 0 ALARM C!
2 | : STACKFULL ( -- )
3     DEPTH $20 > ABORT" TIGHT STACK"
4     ALARM C@ 0= IF -1 ALARM C! TRUE ABORT" DICTIONARY FULL" THEN
5         ." STILL FULL" ;
6
7 CODE ?STACK USER' DP # LDY
8     SEC SP LDA UP )Y SBC N STA INY SP 1+ LDA UP )Y SBC
9     0= ?[ 1 # LDY ;C: STACKFULL ; ASSEMBLER ]? alarm stx
10     USER' S0 # LDY UP )Y LDA SP CMP INY

```

```

11      UP )Y LDA  SP 1+ SBC  1 # LDY  CS ?[  NEXT JMP ]?
12      ;C: TRUE ABORT" STACK EMPTY" ; -2 ALLOT
13
14  \ \ : ?STACK  SP@  HERE - 100 U< IF STACKFULL THEN
15      SP@  S0 @ U> ABORT" STACK EMPTY" ;
    ATARIVF.FB Scr 90 Dr 0
0  \ .STATUS PUSH LOAD          08SEP84KS)          cas08jan07
1
2  DEFER .STATUS      ' NOOP IS .STATUS
3
4  | CREATE PULL  0  ] R> R> ! ;
5
6  : PUSH ( ADDR -- )
7      R> SWAP DUP >R @ >R  PULL >R >R ; RESTRICT
8
9
10
11
12
13
14
15
    ATARIVF.FB Scr 91 Dr 0
0  \ RDEPTH DEPTH          cas08jan07
1
2  : RDEPTH  ( -- +N)  R0 @  RP@ 2+ - 2/ ;
3
4  : DEPTH   ( -- +N)  SP@ S0 @ SWAP - 2/ ;
5
6
7
8
9
10
11
12
13
14
15
    ATARIVF.FB Scr 92 Dr 0
0  \ QUIT (QUIT ABORT      07JUN85BP)          cas08jab07
1
2  | : PROMPT  STATE @  IF ." ] "  EXIT  THEN  ."  OK" ;
3
4  : (QUIT
5      BEGIN .STATUS CR QUERY INTERPRET PROMPT  REPEAT ; -2 ALLOT
6
7  DEFER 'QUIT      ' (QUIT IS 'QUIT
8
9  : QUIT      R0 @ RP! [COMPILE] [ 'QUIT ; -2 ALLOT
10
11 : STANDARDI/O  [ OUTPUT ] LITERAL OUTPUT 4 CMOVE ;
12
13 DEFER 'ABORT      ' NOOP IS 'ABORT
14
15 : ABORT CLEARSTACK END-TRACE 'ABORT STANDARDI/O QUIT ; -2 ALLOT
    ATARIVF.FB Scr 93 Dr 0
0  \ (ERROR ABORT" ERROR"      20MAR85BP)          cas08jan07
1

```

```

2 VARIABLE R#      0 R#  !
3
4 : (ERROR ( STRING -- )
5     STANDARDI/O SPACE HERE .NAME COUNT TYPE  SPACE ?CR
6     QUIT ; -2 ALLOT
7
8 ' (ERROR  ERRORHANDLER  !
9
10 : (ABORT"      "LIT SWAP IF
11     >R CLEARSTACK R> ERRORHANDLER PERFORM
12     EXIT THEN  DROP ;  RESTRICT
13
14
15
    ATARIVF.FB Scr 94 Dr 0
0 \ ABORT" ERROR"                                cas08jan07
1
2 | : (ERR"  "LIT SWAP
3     IF ERRORHANDLER  PERFORM EXIT THEN DROP ;  RESTRICT
4
5 : ABORT"  COMPILER (ABORT"  ," ;  IMMEDIATE  RESTRICT
6
7 : ERROR"  COMPILER (ERR"  ," ;  IMMEDIATE  RESTRICT
8
9
10
11
12
13
14
15
    ATARIVF.FB Scr 95 Dr 0
0 \ -TRAILING                                08APR85BP)                                cas08jan07
1
2 $20 CONSTANT BL
3
4 CODE -TRAILING ( ADDR N1 -- ADR  N2 )
5 TYA  SETUP JSR
6 SP X) LDA  N 2+ STA  CLC
7 SP )Y LDA  N 1+ ADC  N 3 + STA
8 N LDY  CLC  CS ?[
9 LABEL (-TRAIL
10 DEY  N 2+ )Y LDA  BL # CMP
11 0<> ?[ INY  0= ?[ N 1+ INC ]?
12     TYA PHA  N 1+ LDA PUSH JMP ]?
13 ]?  TYA  (-TRAIL BNE
14 N 3 + DEC N 1 + DEC  (-TRAIL BPL
15 TYA PUSH0A JMP  END-CODE
    ATARIVF.FB Scr 96 Dr 0
0 \ SPACE SPACES                                29JAN85KS/BP)
1
2 : SPACE                                BL EMIT ;
3
4 : SPACES ( U --)  0  ?DO SPACE LOOP ;
5
6 \\
7 : -TRAILING ( ADDR N1 -- ADDR N2)
8 2DUP  BOUNDS
9     ?DO 2DUP + 1- C@ BL -

```

```

10     IF LEAVE THEN 1- LOOP ;
11
12
13
14
15
    ATARIVF.FB Scr 97 Dr 0
0 \ HOLD <# #> SIGN # #S          24DEC83KS)          cas08jan07
1 | : HLD ( -- ADDR)      PAD 2- ;
2
3 : HOLD ( CHAR -- )      -1 HLD +! HLD @ C! ;
4
5 : <#                      HLD HLD ! ;
6
7 : #> ( 32B -- ADDR +N ) 2DROP HLD @ HLD OVER - ;
8
9 : SIGN ( N -- ) 0< IF ASCII - HOLD THEN ;
10
11 : #      ( +D1 -- +D2) BASE @ UD/MOD ROT $9 OVER <
12   IF [ ASCII A ASCII 9 - 1- ] LITERAL +
13   THEN ASCII 0 + HOLD ;
14
15 : #S      ( +D -- 0 0 ) BEGIN # 2DUP D0= UNTIL ;
    ATARIVF.FB Scr 98 Dr 0
0 \ PRINT NUMBERS          24DEC83KS)
1
2 : D.R -ROT UNDER DABS <# #S ROT SIGN #>
3     ROT OVER MAX OVER - SPACES TYPE ;
4
5 : .R      SWAP EXTEND ROT D.R ;
6
7 : U.R    0 SWAP D.R ;
8
9 : D.     0 D.R SPACE ;
10
11 : .      EXTEND D. ;
12
13 : U.     0 D. ;
14
15
    ATARIVF.FB Scr 99 Dr 0
0 \ .S C/L L/S            24DEC83KS)          cas21cas08jan07
1
2 : .S SP@ S0 @ OVER - $20 UMIN BOUNDS ?DO I @ U. 2 +LOOP ;
3
4 &40 CONSTANT C/L      \ SCREEN LINE LENGTH
5 &24 CONSTANT L/S      \ LINES PER SCREEN
6
7
8
9
10
11
12
13
14
15
    ATARIVF.FB Scr 100 Dr 0
0 \ MULTITASKER PRIMITIVES  BP03NOV85)          cas08jan07

```


9
10
11
12
13
14
15

ATARIVF.FB Scr 104 Dr 0

0 \\
1
2
3
4
5
6
7
8
9

cas11aug06

10
11
12
13
14
15

ATARIVF.FB Scr 105 Dr 0

0 \\
1
2
3
4
5
6
7
8
9

cas11aug06

10
11
12
13
14
15

ATARIVF.FB Scr 106 Dr 0

0 \\
1
2
3
4
5
6
7
8
9

cas11aug06

10
11
12
13
14
15

ATARIVF.FB Scr 107 Dr 0

0 \\ cas11aug06
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

ATARIVF.FB Scr 108 Dr 0

0 \\ cas11aug06
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

ATARIVF.FB Scr 109 Dr 0

0 \\ cas11aug06
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

ATARIVF.FB Scr 110 Dr 0

0 \\ cas11aug06
1
2
3
4
5
6
7

```

8
9
10
11
12
13
14
15
    ATARIVF.FB Scr 111 Dr 0
0  \ \                                     cas11aug06
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
    ATARIVF.FB Scr 112 Dr 0
0  \ LIMIT FIRST                           cas08jan07
1
2  $BC00 CONSTANT LIMIT
3  VARIABLE FIRST
4
5
6
7
8
9
10
11
12
13
14
15
    ATARIVF.FB Scr 113 Dr 0
0  \ ENDPOINTS OF FORGET      04JAN85BP/KS)   cas08jan07
1  | : \? ( NFA -- FLAG )   C@ $20 AND ;
2
3  | : FORGET? ( ADR NFA -- FLAG ) \ CODE IN HEAP OR ABOVE ADR ?
4      NAME> UNDER 1+ U< SWAP HEAP? OR ;
5
6  | : ENDPOINTS ( ADDR -- ADDR SYMB)
7  HEAP VOC-LINK @ >R
8  BEGIN R> @ ?DUP \ THROUGH ALL VOCABS
9  WHILE DUP >R 4 - >R \ LINK ON RETURNST.
10 BEGIN R> @ >R OVER 1- DUP R@ U< \ UNTIL LINK OR
11     SWAP R@ 2+ NAME> U< AND \ CODE UNDER ADR
12 WHILE R@ HEAP? [ 2DUP ] UNTIL \ SEARCH FOR A NAME IN HEAP
13 R@ 2+ \? IF OVER R@ 2+ FORGET?
14     IF R@ 2+ (NAME> 2+ UMAX THEN \ THEN UPDATE SYMB
15 THEN REPEAT RDROP REPEAT ;

```

```

ATARIVF.FB Scr 114 Dr 0
0 \ REMOVE                      23JUL85WE
1
2 | CODE REMOVE ( DIC SYMB THR - DIC SYMB)
3   5 # LDY [[ SP )Y LDA N ,Y STA DEY 0< ?] USER' S0 # LDY
4   CLC UP )Y LDA 6 # ADC N 6 + STA
5   INY UP )Y LDA 0 # ADC N 7 + STA 1 # LDY
6   [[ N X) LDA N 8 + STA N )Y LDA N 9 + STA N 8 + ORA 0<>
7   ?[[ N 8 + LDA N 6 + CMP N 9 + LDA N 7 + SBC CS
8     ?[ N 8 + LDA N 2 + CMP N 9 + LDA N 3 + SBC
9     ][ N 4 + LDA N 8 + CMP N 5 + LDA N 9 + SBC
10    ]? CC
11    ?[ N 8 + X) LDA N X) STA N 8 + )Y LDA N )Y STA
12    ][ N 8 + LDA N STA N 9 + LDA N 1+ STA ]?
13    ]]? (DROP JMP END-CODE
14
15

```

```

ATARIVF.FB Scr 115 Dr 0
0 \ REMOVE- FORGET-WORDS 29APR85BP)
1
2 | : REMOVE-WORDS ( DIC SYMB -- DIC SYMB)
3   VOC-LINK BEGIN @ ?DUP WHILE DUP >R 4 - REMOVE R> REPEAT ;
4
5 | : REMOVE-TASKS ( DIC --)
6   UP@ BEGIN 1+ DUP @ UP@ - WHILE 2DUP @ SWAP HERE UWITHIN
7   IF DUP @ 1+ @ OVER ! 1- ELSE @ THEN REPEAT 2DROP ;
8
9 | : REMOVE-VOCS ( DIC SYMB -- DIC SYMB)
10  VOC-LINK REMOVE THRU.VOCSTACK
11  DO 2DUP I @ -ROT UWITHIN
12  IF [ ' FORTH 2+ ] LITERAL I ! THEN -2 +LOOP
13  2DUP CURRENT @ -ROT UWITHIN
14  IF [ ' FORTH 2+ ] LITERAL CURRENT ! THEN ;
15

```

```

ATARIVF.FB Scr 116 Dr 0
0 \ FORGET-WORDS                      cas08jan07
1
2 DEFER CUSTOM-REMOVE
3 ' NOOP IS CUSTOM-REMOVE
4
5
6 | : FORGET-WORDS ( DIC SYMB --)
7   OVER REMOVE-TASKS REMOVE-VOCS
8   REMOVE-WORDS CUSTOM-REMOVE
9   HEAP SWAP - HALLOT DP ! 0 LAST ! ;
10
11
12
13
14
15

```

```

ATARIVF.FB Scr 117 Dr 0
0 \ DELETING WORDS FROM DICT. 13JAN83KS)
1
2 : CLEAR HERE DUP UP@ FORGET-WORDS DP ! ;
3
4 : (FORGET ( ADR --) DUP HEAP? ABORT" IS SYMBOL"
5   ENDPOINTS FORGET-WORDS ;
6

```

```

7 : FORGET ' DUP [ DP ] LITERAL @ U< ABORT" PROTECTED"
8   >NAME DUP HEAP? IF NAME> ELSE 2- 2- THEN (FORGET ;
9
10 : EMPTY [ DP ] LITERAL @
11   UP@ FORGET-WORDS [ UDP ] LITERAL @ UDP ! ;
12
13
14
15
    ATARIVF.FB Scr 118 Dr 0
0 \ SAVE BYE STOP? ?CR          20OCT84KS/BP)          cas08jan07
1
2 : SAVE
3   HERE UP@ FORGET-WORDS VOC-LINK @
4   BEGIN DUP 2- 2- @ OVER 2- ! @ ?DUP 0= UNTIL
5   UP@ ORIGIN $100 CMOVE ;
6
7 : BYE (BYE ;
8
9 | : END?    KEY #CR = IF TRUE RDROP THEN ;
10
11 : STOP?    ( -- FLAG) KEY? IF END? END? THEN FALSE ;
12
13 : ?CR     COL C/L $A - U> IF CR THEN ;
14
15
    ATARIVF.FB Scr 119 Dr 0
0 \ IN/OUTPUT STRUCTURE          02MAR85BP)          cas08jan07
1 | : OUT:   CREATE DUP C, 2+ DOES> C@ OUTPUT @ + PERFORM ;
2
3   : OUTPUT: CREATE: DOES> OUTPUT ! ;
4 0 OUT: EMIT   OUT: CR      OUT: TYPE
5   OUT: DEL    OUT: PAGE    OUT: AT    OUT: AT?  DROP
6
7 : ROW      ( -- ROW)  AT? DROP ;
8 : COL      ( -- COL)  AT? NIP ;
9
10 | : IN:    CREATE DUP C, 2+ DOES> C@ INPUT @ + PERFORM ;
11
12   : INPUT:  CREATE: DOES> INPUT ! ;
13
14 0 IN: KEY    IN: KEY?    IN: DECODE  IN: EXPECT  DROP
15
    ATARIVF.FB Scr 120 Dr 0
0 \ ALIAS ONLY DEFINITIONEN     29JAN85BP)
1
2 ONLY DEFINITIONS FORTH
3
4 : SEAL 0 [' ] ONLY >BODY ! ; \ KILL ALL WORDS IN ONLY)
5
6   ' ONLY ALIAS ONLY
7   ' FORTH ALIAS FORTH
8   ' WORDS ALIAS WORDS
9   ' ALSO ALIAS ALSO
10 ' DEFINITIONS ALIAS DEFINITIONS
11 HOST TARGET
12
13
14

```

```

15
  ATARIVF.FB Scr 121 Dr 0
0 \ 'COLD                                     cas08jan07
1 | : INIT-VOCABULARYS   VOC-LINK @
2   BEGIN DUP 2- @ OVER 4 - ! @ ?DUP 0= UNTIL ;
3
4 DEFER 'COLD      ' NOOP IS 'COLD
5
6 | : (COLD INIT-VOCABULARYS ONLYFORTH 'COLD PAGE LOGO COUNT TYPE
7   CR (RESTART ; -2 ALLOT
8
9 DEFER 'RESTART  ' NOOP IS 'RESTART
10
11 | : (RESTART ['] (QUIT IS 'QUIT
12   'RESTART [ ERRORHANDLER ] LITERAL @ ERRORHANDLER !
13   ['] NOOP IS 'ABORT ABORT ; -2 ALLOT
14
15

```

```

  ATARIVF.FB Scr 122 Dr 0
0 \ COLD BOOTSYSTEM RESTART      09JUL85WE)          cas08jan07
1 CODE COLD      HERE >COLD !
2 ' (COLD >BODY $100 U/MOD # LDA PHA # LDA PHA
3 LABEL BOOTSYSTEM CLI 0 # LDY
4 CLC S0 LDA 6 # ADC N STA S0 1+ LDA 0 # ADC N 1+ STA
5 [[ ORIGIN ,Y LDA N )Y STA INY 0= ?]
6 $C lda HERE 9 + sta $D lda HERE 5 + sta
7 LABEL WARMBOOT $e474 jsr BOOTNEXTLEN 1- # LDY
8 [[ BOOTNEXT ,Y LDA PUTA ,Y STA DEY 0< ?]
9 CLC S0 LDA 6 # ADC UP STA S0 1+ LDA 0 # ADC UP 1+ STA
10 USER' S0 # LDY UP )Y LDA SP STA INY UP )Y LDA SP 1+ STA
11 USER' R0 # LDY UP )Y LDA RP STA INY UP )Y LDA RP 1+ STA
12 0 # LDX 1 # LDY TXA RP X) STA RP )Y STA
13 PLA IP STA PLA IP 1+ STA
14 LABEL DOSINI 0 # lda $D sta 0 # lda $C sta
15 LABEL XYNEXT 0 # LDX 1 # LDY NEXT JMP END-CODE

```

```

  ATARIVF.FB Scr 123 Dr 0
0 \ ( RESTART PARAM.-PASSING TO FORTH  BP)          cas08jan07
1
2 CODE RESTART      HERE >RESTART !
3 ' (RESTART >BODY $100 U/MOD
4 # LDA PHA # LDA PHA WARMBOOT JMP END-CODE
5
6 >RESTART @ $100 U/MOD DOSINI 1+ C! DOSINI 5 + C!
7
8
9
10
11
12
13
14
15

```

```

  ATARIVF.FB Scr 124 Dr 0
0 \ CODE FOR PARAMETER-PASSING TO FORTH          cas11aug06
1 CR .( Include Atari 8bit IO definitions )
2 include atariio.fb CR
3
4 HOST ' TRANSIENT 8 + @
5 TRANSIENT FORTH CONTEXT @ 6 + !

```

6 TARGET
7
8 FORTH ALSO DEFINITIONS
9
10 : FORTH-83 ; \ LAST WORD IN DICTIONARY
11
12
13
14
15

ATARIVF.FB Scr 125 Dr 0

0 \ SYSTEM DEPENDENT CONSTANTS BP/KS)
1
2 VOCABULARY ASSEMBLER
3 ASSEMBLER DEFINITIONS
4 TRANSIENT ASSEMBLER
5 PUSHA CONSTANT PUSHA \ PUT A SIGN-EXTENDED ON STACK
6 PUSH0A CONSTANT PUSH0A \ PUT A ON STACK
7 PUSH CONSTANT PUSH \ MSB IN A AND LSB ON JSR-STACK
8 RP CONSTANT RP
9 UP CONSTANT UP
10 SP CONSTANT SP
11 IP CONSTANT IP
12 N CONSTANT N
13 PUTA CONSTANT PUTA
14 W CONSTANT W
15 SETUP CONSTANT SETUP

ATARIVF.FB Scr 126 Dr 0

0 \ NEXT XYNEXT LABELS cas11aug06
1 NEXT CONSTANT NEXT
2 XYNEXT CONSTANT XYNEXT
3 (2DROP CONSTANT POPTWO
4 (DROP CONSTANT POP
5
6
7
8
9
10
11
12
13
14
15

ATARIVF.FB Scr 127 Dr 0

0 \ SYSTEM PATCHUP cas11aug06
1
2 FORTH DEFINITIONS
3
4 \$BC00 ' LIMIT >BODY ! \$BC00 FIRST !
5 \$BA00 S0 ! \$BB80 R0 !
6
7 S0 @ DUP S0 2- ! 6 + S0 7 - !
8 HERE DP !
9
10 HOST TUDP @ TARGET UDP !
11 HOST TVOC-LINK @ TARGET VOC-LINK !
12 HOST MOVE-THREADS
13

14
15
ok

I/O Definitions#

PRT2C

ok

14 0 pall ATARIIO.FB Scr 0 Dr 0

0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

ATARIIO.FB Scr 1 Dr 0

0 \ loadscreen fuer ATARI 8bit cas11aug06

1 \ 800 / 600 XL / 800 XL / 1200 XL / 130 XE / 65 XE / 800 XE

2
3 1 &14 +thru

4
5
6
7
8
9
10
11
12
13
14
15

ATARIIO.FB Scr 2 Dr 0

0 \ 65KEY? GETKEY cas09jan07

1
2 | \$02FC Constant CH
3 | CODE 65KEY? (-- FLAG) CH lda clc 1 # adc push0a jmp end-code
4
5 LABEL GETCHK \$E425 lda pha \$E424 lda pha rts
6 | CODE GETKEY (-- 8B) \$FF sty \$FE stx GETCHK jsr
7 | \$FE ldx \$FF ldy push0a jmp end-code
8
9 | \$02F0 Constant CRSINH
10 | CODE CURON (--) 01 # lda
11 LABEL CRS01 CRSINH sta NEXT JMP END-CODE
12 | CODE CUROFF (--) 00 # lda CRS01 JMP END-CODE
13
14 | : 65KEY (-- 8B)
15 CURON BEGIN PAUSE 65KEY? UNTIL CUROFF GETKEY ;

```

ATARIIO.FB Scr 3 Dr 0
0 \ DECODE EXPECT KEYBOARD      BP28MAY85)                cas09jan07
1 $7E CONSTANT #BS   $9B CONSTANT #CR   &27 CONSTANT #ESC
2
3 | : 65DECODE ( ADDR CNT1 KEY -- ADDR CNT2)
4   #BS CASE? IF DUP IF DEL 1- THEN EXIT THEN
5   #CR CASE? IF DUP SPAN ! EXIT THEN
6   >R 2DUP + R@ SWAP C! R> EMIT 1+ ;
7
8 | : 65EXPECT ( ADDR LEN1 -- ) SPAN ! 0
9   BEGIN DUP SPAN @ U<
10  WHILE KEY DECODE
11  REPEAT 2DROP SPACE ;
12
13 INPUT: KEYBOARD [ HERE INPUT ! ]
14 65KEY 65KEY? 65DECODE 65EXPECT [
15

```

```

ATARIIO.FB Scr 4 Dr 0
0 \ (emit 65emit                )                cas09jan07
1
2 LABEL OUTCHK
3   $E407 lda pha $E406 lda pha txa rts
4
5 | Code (emit ( 8b -- ) $FF sty $FE stx
6           SP X) lda tax OUTCHK jsr
7           $FE ldx $FF ldy (drop jmp end-code
8
9
10
11
12
13
14
15

```

```

ATARIIO.FB Scr 5 Dr 0
0 \ EMIT CR DEL PAGE AT AT?      25JAN85RE)                cas09jan07
1
2 | &40 Constant c/row
3
4 | : 65emit ( 8b -- ) (emit ;
5
6 | : 65CR   #CR 65emit ;
7
8 | : 65DEL  #bs 65emit SPACE #bs 65emit ;
9
10 | : 65PAGE &125 EMIT ;
11
12 | : 65at ( row col -- ) $55 ! $54 C! ;
13
14 | : 65AT? ( -- ROW COL ) $54 C@ $55 @ ;
15

```

```

ATARIIO.FB Scr 6 Dr 0
0 \                                cas09jan07
1
2 | : 65type ( adr len -- ) bounds ?DO I c@ emit LOOP ;
3
4
5
6

```

```

7
8
9
10
11
12
13
14
15
    ATARIIO.FB Scr 7 Dr 0
0 \ TYPE DISPLAY (BYE          BP 28MAY85RE)                cas09dec05
1
2 OUTPUT: DISPLAY [ HERE OUTPUT ! ]
3 65EMIT 65CR 65TYPE 65DEL 65PAGE 65AT 65AT? [
4
5 \ fix dosini vector and jump through dosvec
6 | code (bye warmboot 1+ lda $0C sta warmboot 2+ lda
7     $0D sta $000A ) jmp end-code
8
9
10
11
12
13
14
15
    ATARIIO.FB Scr 8 Dr 0
0 \ FileInterface                cas09jan07
1
2
3 \ definitions for fileinterface
4
5 &4 CONSTANT R/O    &8 CONSTANT W/O    &12 CONSTANT R/W
6 3 CONSTANT IO-OPEN    5 CONSTANT IO-GETREC 7 CONSTANT IO-GETCHR
7 9 CONSTANT IO-PUTREC $B CONSTANT IO-PUTCHR $C CONSTANT IO-CLOSE
8
9 $340 CONSTANT ICFLG  $342 CONSTANT ICCOM  $343 CONSTANT ICSTA
10 $344 CONSTANT ICBAL  $345 CONSTANT ICBAH  $348 CONSTANT ICBLL
11 $349 CONSTANT ICBLH  $34A CONSTANT ICAX1  $34B CONSTANT ICAX2
12 $E456 CONSTANT CIOV
13
14
15
    ATARIIO.FB Scr 9 Dr 0
0 \ definitions for fileinterface                cas09jan07
1
2 label freeiobc0 70 # lda label freeiobc2 tay ICFLG ,y lda
3     $FF # cmp 0<> ?[ tya sec $10 # sbc freeiobc2 bne ]?
4     tya rts
5
6 | code freeiobc freeiobc0 jsr .a lsr .a lsr .a lsr .a lsr pha
7     push0a jmp end-code
8
9 label getfileid sp x) lda .a ASL .a ASL .a ASL .a ASL tay rts
10
11 label getparam 2 # ldy sp )y lda ICBLL ,x sta
12     iny sp )y lda ICBLH ,x sta
13     iny sp )y lda ICBAL ,x sta
14     iny sp )y lda ICBAH ,x sta

```

```

15             rts
    ATARIIO.FB Scr 10 Dr 0
0 \ definitions for fileinterface                cas13dec05
1
2 code close-file  getfileid jsr tax IO-CLOSE # lda ICCOM ,x sta
3             CIOV jsr  sp 2inc ICSTA ,x lda  0>= ?[ 0 # lda ]? pha
4             PUSH0A jmp end-code
5
6 code open-file   freeiocb0 jsr tax  IO-OPEN # lda ICCOM ,y sta
7             4 # ldy  sp )y lda  ICBAL ,x sta
8             iny  sp )y lda  ICBAH ,x sta
9             0 # ldy  sp )y lda  ICAX1 ,x sta
10            tya             ICAX2 ,x sta
11            CIOV jsr sp 2inc 0 # ldy ICSTA ,x lda  sp )y sta
12            0>= ?[ 0 # lda sp )y sta ]? 0 # lda tay iny sp )y sta
13            iny iny sp )y sta txa clc .a lsr .a lsr .a lsr
14            .a lsr dey sp )y sta xynext jmp end-code
15

```

```

    ATARIIO.FB Scr 11 Dr 0
0 \ definitions for fileinterface                cas11aug06
1
2 code read-file ( caddr u fileid -- u2 ior )
3   getfileid jsr  tax  getparam jsr
4     IO-GETCHR # lda  ICCOM ,x sta
5     CIOV jsr  sp 2inc  0 # ldy
6     ICSTA ,x lda  sp )y sta
7     0>= ?[ 0 # lda  sp )y sta ]?  tya iny sp )y sta
8     clc  iny  ICBLL ,x lda  sp )y sta
9     iny  ICBLH ,x lda  sp )y sta  xynext jmp end-code
10
11
12
13
14
15

```

```

    ATARIIO.FB Scr 12 Dr 0
0 \ definitions for fileinterface                cas11aug06
1
2 code read-line ( caddr u fileid -- u2 flag ior )
3   getfileid jsr  tax  getparam jsr
4     IO-GETREC # lda  ICCOM ,x sta
5     CIOV jsr  0 # ldy
6     ICSTA ,x lda
7     0>= ?[ tya ]? sp )y sta
8     4 # ldy ICBLL ,x lda  sp )y sta
9     ICBAL ,x adc  tay dey n sty  5 # ldy ICBLH ,x lda sp )y sta
10    ICBAH ,x adc  n 1+ sta  0 # lda  tay  n )y sta iny
11    sp )y sta iny sp )y sta iny sp )y sta xynext jmp end-code
12
13
14
15

```

```

    ATARIIO.FB Scr 13 Dr 0
0 \ definitions for fileinterface                cas11aug06
1
2 code write-file ( caddr u fileid -- ior )
3   getfileid jsr  tax  getparam jsr
4     IO-PUTCHR # lda  ICCOM ,x sta
5     CIOV jsr  sp 2inc sp 2inc  0 # ldy

```

```
6      ICSTA ,x lda sp )y sta
7      0>= ?[ 0 # lda sp )y sta ]?
8      xynext jmp end-code
```

```
9
10
11
12
13
14
15
```

ok

display PRT2C

```
      ATARIIO.FB Scr 14 Dr 0
0 \ definitions for fileinterface                                cas09jan07
1 VARIABLE SOURCE-ID 0 SOURCE-ID !
2 | $580 CONSTANT FNBUF
3 : REFILL tib $50 erase tib $50 SOURCE-ID @ READ-LINE
4     ROT 1 - #tib ! >in off nip ;
5 : INCLUDE-FILE ( fileid -- )
6 SOURCE-ID ! BEGIN REFILL $80 < WHILE INTERPRET .STATUS REPEAT
7 SOURCE-ID @ CLOSE-FILE ABORT" File Error" ;
8 : INCLUDED ( caddr u -- )
9 SOURCE-ID @ >R R/O OPEN-FILE DUP $80 < IF DROP
10 INCLUDE-FILE HERE $50 ERASE #TIB @ >IN ! ELSE
11 ." FileError:" . ABORT THEN R> SOURCE-ID ! ;
12 : FILE" FNBUF $50 BL FILL HERE $50 BL FILL ASCII " WORD
13     COUNT FNBUF SWAP CMOVE FNBUF 0 ;
14 : INCLUDE" ( FNAME ) FILE" INCLUDED ; IMMEDIATE
15
```