PART 4. 8086 SPECIFIC UTILITIES

CHAPTER 22. DEBUGGER

The souce code of the debugger is in CPU8086.BLK, screens 18 to 20, and in UTILITY.BLK, screens 49 to 51.

The debugger in F83 is designed to let user single step through the execution of a high level definition. To invoke the debugger, type

DEBUG xxx

where xxx is the name of the colon definition you want to trace. DEBUG patches the NEXT routine with a special routine DEBNEXT to display the contents of the data stack at every step when DEBNEXT is encountered. The real single step action occurs only when the word xxx is executed. When xxx is executed, you will get a single step trace showing the word within xxx that is about to be executed, and the contents of the data stack. At each step, you can use the commands C (continue), F (Forth) , and Q (quit) to control the stepping process. F allows you the execute any Forth word to poke around, until you type RESUME to continue with the debugging. Q stops the debugging process and restores xxx to its original condition. See Section 3.5 for a discussion on how to use the debugger.

22.1. LOW LEVEL SUPPORTING WORDS

VOCABULARY BUG — The vocabulary holding all the debugger supporting words.

BUG ALSO DEFINITIONS — Declare BUG as the current vocabulary to add new words to it.

VARIABLE 'DEBUG — A variable holding the code field address of the word to be traced.

VARIABLE <IP — Lower limit of tracing range for IP, the interpretive pointer.

VARIABLE IP> — Upper limit of IP for tracing.

VARIABLE CNT — Count of tracing through debug NEXT.

ASSEMBLER HEX — Invoke the assembler to use the LABEL word in ASSEMBLER vocabulary.

LABEL FNEXT — A machine code subroutine restoring NEXT back to its original form.

- 0AD # AL MOV — AD is the machine code of indirect jump.
- AL >NEXT #) MOV — Put this jump code in >NEXT.
- D88B # AX MOV — The address of the real NEXT code.
- AX >NEXT 1+ #) MOV — Put this address after the jump code. This is the original NEXT.

```
        RET

LABEL DNEXT             A copy of NEXT that gets executed during debugging in
                        place of the regular NEXT.
```

```
        AX LODS                 Load IP into AX and increment IP by 2.
        AX W MOV                Copy IP into W register.
        0 [W] JMP               Indirect jump through W register.

LABEL DEBNEXT                   The debugger's version of NEXT. If IP is between <IP and
                                IP>, the contents of the execution variable 'DEBUG are
                                executed. The word pointed to by 'DEBUG must drop the
                        IP pushed on data stack by DEBNEXT and must be
        terminated by PNEXT for more tracing.
        <IP #) IP CMP U> IF     IP greater than <IP?
                IP> #) IP CMP
                 U<= IF         IP less than IP>? If both true, do the following:
                        CNT #) AL MOV
                        AL INC
                        AL CNT #) MOV           Increment CNT.
                        2 # AL CMP              AL=2?
                        0= IF                   Yes.  Do the following every other time.
                                AL AL SUB
                                AL CNT #) MOV   Clear CNT counter.
                                FNEXT #) CALL   Restore the original NEXT.
                                IP PUSH         Push current IP on data stack.
                                'DEBUG #) W MOV Copy the execution address in 'DEBUG to W
                                                register.
                                0 [W] JMP       Indirect jump through W. The trace routine is
                                                executed.
                        THEN
                THEN
        THEN
        DNEXT #) JMP            Non of the about cases are true. Execute the regular NEXT.

CODE PNEXT      ( --- )         Patch the regular NEXT in Forth to jump to DEBNEXT.
                                This puts us in the debug mode and allows for tracing single
                                steps.
        0E9 # AL MOV            E9 is the machine code of JMP.
        AL >NEXT #) MOV         Copy this code into >NEXT.
        DEBNEXT >NEXT 3 + - # AX MOV
        AX >NEXT 1+ #) MOV      Copy the address of DEBNEXT to the cell next to >NEXT.
                                >NEXT is now patched to execute DEBUG by jumping to
                                DEBNEXT.
        NEXT
        END-CODE

FORTH DEFINITIONS               Next instruction must be defined in FORTH vocabulary.

CODE UNBUG      ( --- )         Restore FORTH's NEXT to its original condition and
                                disable tracing.
        FNEXT #) CALL           Call FNEXT to fix NEXT.
        NEXT
```

END-CODE

22.2. HIGH LEVEL TRACE COMMANDS

```
BUG ALSO DEFINITIONS        Put the following words in the BUG vocabulary.  Only the
                            very last word TRACE needs to be in the FORTH
                            vocabulary for ease of accessing.

: L.ID          ( nfa len --- )     Print the name of a word left justified in a field of at least ten
                                    characters.
        SWAP DUP .ID                Print the name.
        DUP NAME>               Get the code field address.
        1- - + SPACES               Fill in spaces.
        ;

VARIABLE SLOW                       When true, step continuously. When false, single step.
VARIABLE RES                        When true, resume debugging.

: (DEBUG)       ( low-addr hi-addr --- )        Prepare to trace the words between the specified
                                    range for IP pointer.
        1 CNT !                     Store 1 into CNT to run DEBNEXT every other time
                                    through NEXT.
        IP> !                       Set the high limit of IP.
        <IP !                       Set the low limit.
        PNEXT                       Patch NEXT to DEBNEXT.
        ;

: 'UNNEST       ( pfa --- pfa' )  From the starting address of a parameter field, find the end
                                    of this field indicated by UNNEST.
        BEGIN
                1+ DUP @            Get the execution address of the next word.
                ['] UNNEST          Is it UNNEST?
        UNTIL                       Exit if UNNEST is found and leave its address on stack.     ;

: TRACE         ( ip --- )          Display the contents of the data stack and the name of the
                                    next word about to execute in the routine being debugged.  It
                                    then waits for a key unless SLOW is true. If the key is C, F,
                                    or Q, special action is taken; otherwise, a single step is
                                    performed.
        >R                          Save ip.
        .S                          Display the data stack.
        R>                          Restore ip.
        CR @ >NAME                  Get the name field of the word pointed to by ip.
        10 L.ID                     Print its name.
        SLOW @ NOT                  If SLOW is false,
        KEY? OR                     or a key is received, do the following:
        IF
                SLOW OFF            Reset SLOW flag.
                RES OFF             Reset RESume flag also.
                ."   --->"          Print a prompting message.
```

```
KEY                 Wait for a key here.
UPC                 Change the key code to upper case always.
ASCII C OVER = IF   If the key is C,
SLOW @ NOT          complement SLOW.
SLOW ! THEN
```

```
            ACSII F OVER = IF   If this key is F,
            DROP                throw away the key code,
            BEGIN               and entry a Forth interpreter loop.
                  QUERY RUN     Interpret any Forth command.
                  RES @         Continue if RES is false.
            UNTIL
            THEN
            ASCII Q OVER =      If the key is Q,
            ABORT" Unbug"       abort debugger.
            DROP THEN
      PNEXT                     Patch NEXT again to continue tracing the next word.     ;

' TRACE 'DEBUG !                Vector 'DEBUG to execute TRACE. This is the function of
                                DEBNEXT.

FORTH DEFINITIONS               Put the final debugger commands in the regular FORTH
                          vocabulary so that the user can invoke it conveniently. Other
                          debugging words are hidden in the BUG vocabulary.

: DEBUG     ( --- )             Patch NEXT to DEBNEXT and set the range of IP to be
                          debugged.
      '                         Get the execution address of the next word following
                          DEBUG.
      2-                        Convert cfa to pfa.
      DUP [ BUG ] 'UNNEST       Find the IP range.
      (DEBUG)                   Set IP range and patch NEXT.
      ;

: RESUME    ( --- 0 )           Turn on RES to enable tracing to continue.
      [ BUG ] RES ON            Set RES flag.
      0                         Leave a dummy stack item to replace the key code dropped.
      PNEXT                     Patch NEXT to continue debugging.
      ;

ONLY FORTH ALSO DEFINITIONS     Re-initialize the vocabulary searching order.
```