```
BOOT     START      0          BOOTSTRAP LOADER FOR SIC/XE
.
. THIS BOOTSTRAP READS OBJECT CODE FROM DEVICE F1 AND ENTERS IT
. INTO MEMORY STARTING AT ADDRESS 80 (HEXADECIMAL). AFTER ALL OF
. THE CODE FROM DEVF1 HAS BEEN SEEN ENTERED INTO MEMORY, THE
. BOOTSTRAP EXECUTES A JUMP TO ADDRESS 80 TO BEGIN EXECUTION OF
. THE PROGRAM JUST LOADED.  REGISTER X CONTAINS THE NEXT ADDRESS
. TO BE LOADED.
.
         CLEAR      A          CLEAR REGISTER A TO ZERO
         LDX        #128       INITIALIZE REGISTER X TO HEX 80
LOOP     JSUB       GETC       READ HEX DIGIT FROM PROGRAM BEING LOADED
         RMO        A,S        SAVE IN REGISTER S
         SHIFTL     S,4        MOVE TO HIGH-ORDER 4 BITS OF BYTE
         JSUB       GETC       GET NEXT HEX DIGIT
         ADDR       S,A        COMBINE DIGITS TO FORM ONE BYTE
         STCH       0,X        STORE AT ADDRESS IN REGISTER X
         TIXR       X,X        ADD 1 TO MEMORY ADDRESS BEING LOADED
         J          LOOP       LOOP UNTIL END OF INPUT IS REACHED
.
. SUBROUTINE TO READ ONE CHARACTER FROM INPUT DEVICE AND
. CONVERT IT FROM ASCII CODE TO HEXADECIMAL DIGIT VALUE. THE
. CONVERTED DIGIT VALUE IS RETURNED IN REGISTER A. WHEN AN
. END-OF-FILE IS READ, CONTROL IS TRANSFERRED TO THE STARTING
. ADDRESS (HEX 80).
.
GETC     TD         INPUT      TEST INPUT DEVICE
         JEQ        GETC       LOOP UNTIL READY
         RD         INPUT      READ CHARACTER
         COMP       #4         IF CHARACTER IS HEX 04 (END OF FILE),
         JEQ        80             JUMP TO START OF PROGRAM JUST LOADED
         COMP       #48        COMPARE TO HEX 30 (CHARACTER '0')
         JLT        GETC       SKIP CHARACTERS LESS THAN '0'
         SUB        #48        SUBTRACT HEX 30 FROM ASCII CODE
         COMP       #10        IF RESULT IS LESS THAN 10, CONVERSION IS
         JLT        RETURN        COMPLETE. OTHERWISE, SUBTRACT 7 MORE
         SUB        #7            (FOR HEX DIGITS 'A' THROUGH 'F')
RETURN   RSUB                  RETURN TO CALLER
INPUT    BYTE       X'F1'      CODE FOR INPUT DEVICE
         END        LOOP
```

Figure 3.3 Bootstrap loader for SIC/XE.

You should work through the execution of this bootstrap routine by hand with several bytes of sample input, keeping track of the exact contents of all registers and memory locations as you go. This will help you become familiar with the machine-level details of how loading is performed.

For simplicity, the bootstrap routine in Fig. 3.3 does not do any error checking it assumes that its input is correct. You are encouraged to think about the