The NanoAssembler

by Roger Wood
HCM Staff

This companion to NanoProcessor shows you how an assembler can provide easy access to machine language—by translating simple instructions into the computer's native tongue.

In the last issue (*HCM* Vol. 5, No. 5), we presented *NanoProcessor*, a program that introduced the concepts of machine-language programming. This program demonstrated how a microprocessor works at its most fundamental level. Although entering and running simple programs on the *NanoProcessor* can be fun, longer and more complicated machine-language routines are another story. Even with short programs, you probably discovered what a time-consuming and error-prone process it can be to enter machine language one bit at a time.

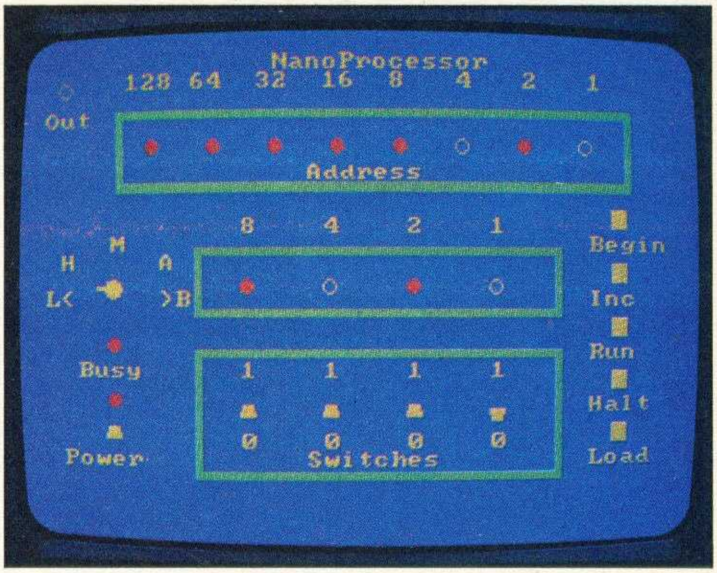
To alleviate the difficulties involved in working with machine language, early computer users created programs called "assemblers." An assembler is a human-to-machine translator. It operates from a "dictionary" of mnemonics (a combination of letters that humans can understand), translating these mnemonics into the numbers of machine code. Using assemblers, you can write a program with the more easily remembered mnemonics, and let the computer create the actual machine language (the ones and zeros).

Thus, we present the *NanoAssembler*; a program that will teach you how to use assemblers. With the *NanoAssembler*, you will be able to write long, complicated programs for the *NanoProcessor* much more easily than you would using machine language.

Source Code To Object Code

You may find that many people refer to "assembly-language programs" and "machine-language programs" interchangeably, as though they were the same thing. Actually, an assembly-language program is a text file—known as a "source file"—that the computer cannot execute directly. It is simply a series of text lines comprising mnemonics, numbers, and labels. Before the computer can run such a program, the source file must be "assembled" or translated into a machine-language file—also known as an "object file."

Take a look at Sample Program 1, which you can load and run on the *NanoProcessor*. You may recognize this program, as it is identical to Sample Program 1 in the last issue. The two left-most columns, entitled Addr and Code, contain the machine language, (object code), which makes up the program. You can enter this object code bit by bit, or you can enter the more easily read and (with some training) understood assembly language (source code), contained in the Line, Label, Mnemonic, and Remark columns. The Remark column is like a REM statement in BASIC. It makes the program much easier to read and understand.



Our *NanoAssembler* package consists of two BASIC programs: the *NanoEditor* and the *NanoAssembler*. The *NanoEditor* is a simple text editor that lets you enter your program as source code and save it to disk (or alternatively tape on the Atari, Commodore, and TI computers). *NanoAssembler* can then read and translate that file into a corresponding file of object code, which you can save to disk or tape. You can then load the object code into the *NanoProcessor* and run it.

Creating A Program

We will use Sample Program 1 to demonstrate how the *NanoEditor* and the *NanoAssembler* work. To start, Load and RUN The *NanoEditor*. You begin with this menu:

1) EDIT
2) FILES
3) PRINT
4) EXIT

Choose the Edit option, which allows you to create and modify files. The *Editor* now displays the command prompt: CMD. You may enter one of 5 single-letter commands:

Command	Function
A	Add a line of text
E	Edit a line of text
D	Delete a line of text
I	Insert a line of text
L	List

To begin creating a new file—in this case Sample Program 1—press A. In response, the *Editor* displays line 001, with a flashing cursor waiting for your input. For each line of source code, the *Editor* provides a line number ranging from 001 to 200. When you enter the Add-a-line mode, the program always displays the cursor on a new line of source code—one line past the last line in memory. You can automatically advance to the next line by pressing [ENTER] or [RETURN]. To exit the Add-a-line mode, press the [ESCAPE] key (see your computer's Control Capsule if your machine does not have an Escape key).

the Print option, just select it from the main menu (3). After you save (and print) the source file, select the Exit option from the main menu. The program gives you a chance to change your mind before ending, so you don't need to worry about losing the program in memory due to an erroneous keypress.

Now it is time to load and RUN the *NanoAssembler*. The program prompts you to load your source file for assembly. As the program translates your source code into machine code, it lists the source file, the addresses, and object code to either the screen or a printer (if you have one).

Passing Through

The actual assembly of the program occurs in two steps, or "passes." Thus, the *NanoAssembler* is a "two-pass" assembler. The first pass does most of the work, determining the correct machine-language instructions and the instruction addresses. However, sorting out labels requires a second pass because, until it identifies all address labels, the program may not know the exact address of each instruction.

Try assembling Sample Program 1. If you have entered it correctly, the *NanoAssembler* should output the assembled version, as shown in Figure 1, to the screen or printer. If you have made an error in entering the program into the *NanoEditor*, the *NanoAssembler* informs you of the line number in the source code that contains the error, and states the type of error. For example, if in line 1 you enter LDA #3 instead of LDA# 3, when you try to assemble the program the computer displays the error: ILLEGAL USE OF LABEL IN LINE 1. Here, the computer interprets the code as a LoaD A *addr* instruction (object code = 2), instead of a LoaD A immediate instruction (object code = 1). Then, when the computer evaluates the "label" #3, it finds that the label is illegal because it does not begin with a letter.

Figure 1: Instruction Set

Dec.	Binary	Nibbles per instr.	Mnemonic	Flags* affected C Z	Function
0	%0000	1	ADD	Y Y	Add the contents of B register to the contents of A register—result in A.
1	%0001	2	LDA#	N Y	Load A with number following instruction.
2	%0010	3	LDA *addr*	N Y	Load A with number at location specified by *addr*.
3	%0011	3	STA *addr*	N N	Store the contents of A at location specified by *addr*.
4	%0100	1	TAB	N N	Transfer contents of A to B.
5	%0101	1	TBA	N Y	Transfer contents of B to A.
6	%0110	1	RRC	Y Y	Rotate A right one bit through carry.
7	%0111	1	RLC	Y Y	Rotate A left one bit through carry.
8	%1000	1	AND	Y Y	Logically AND A and B—Result in A.
9	%1001	1	OR	Y Y	Logically OR A and B—Result in A.
10	%1010	1	XOR	Y Y	Logically XOR A and B—Result in A.
11	%1011	3	BZ *addr*	N N	Branch to *addr* if Zero flag is set.
12	%1100	3	BNZ *addr*	N N	Branch to *addr* if Zero flag is not set.
13	%1101	3	BCS *addr*	N N	Branch to *addr* if Carry flag is set.
14	%1110	3	BCC *addr*	N N	Branch to *addr* if Carry flag is not set.
15	%1111	3	JMP *addr*	N N	Branch to *addr* unconditionally.
Assembler Directives:					
n/a	n/a	0	ORG	n/a	Use to specify a particular address (e.g., specify starting address of program).
n/a	n/a	0	EQU	n/a	Equate label with value—assigns the value to the right of the EQU statement to the label to the left.
n/a	n/a	1	DN	n/a	Define Nibble—assigns the value to the right of the DN statement to the label at the left.

**Flags affected refers to whether or not the instruction has any effect on the flags in the status register. The C column stands for the Carry flag (did the operation result in a carry being generated?), and the Z stands for the Zero flag (did the operation result in a zero?). A Y appears in the column if the flag is affected by the instruction. An N indicates the flag is not changed by the instruction.*

After displaying the program, *NanoAssembler* prompts you to save the object file. The saved file is identical in format to the ones you loaded and saved with the *NanoProcessor* last issue; that is, the file contains the contents of all addresses from 0 through 255. To see that your program works properly, load and RUN the *NanoProcessor*. You can then load and run the program you've just created according to the instructions detailed in Vol. 5, No. 5.

For a short program such as Sample Program 1, this process may seem a bit time consuming. For longer and more complex programs, however, the ease of writing and debugging provided by an assembler more than makes up for the added steps.

Assembler Directives

Figure 1 displays the 16 instructions that we detailed in the *NanoProcessor*. You may specify any of these instructions when writing an assembly-language program with the *NanoEditor*. The *NanoAssembler*, in turn, converts these instructions into their machine codes. There are three additional commands, known as assembler directives, that the *Assembler* understands:

Directive	Purpose
ORG	Start object code here
DN	Define a nibble
EQU	Define a label

The ORG command directs the *NanoAssembler* to assemble the program at a specified address between 0 and 255. For an example of this instruction, see line 1 of Sample Program 2. This program is a slightly modified version of Sample Program 2 that we presented in last issue's *NanoProcessor*. It performs a two nibble addition of numbers located at addresses 240 and 241, placing the answer in addresses 248 and 249. The ORG statement makes the starting address %1010.

The DN instruction allows you to include a particular value at any address. Just specify the address using the ORG directive, and then define the value to be placed at that address with the DN directive. Lines 22 through 24 of Sample Program 2 define the two nibbles that the program adds.

Figure 2

Decimal	Binary	Hexadecimal
0	%0000	$0
1	%0001	$1
2	%0010	$2
3	%0011	$3
4	%0100	$4
5	%0101	$5
6	%0110	$6
7	%0111	$7
8	%1000	$8
9	%1001	$9
10	%1010	$A
11	%1011	$B
12	%1100	$C
13	%1101	$D
14	%1110	$E
15	%1111	$F

Sample Program 1

```
Addr   Code    Line  Label   Mnemonic    Remark
               001           LDA# 3      ;Get first number
 0     %0001
 1     %0011
               002           TAB         ;Move to B
 2     %0100
               003           LDA# 7      ;Get second number
 3     %0001
 4     %0111
               004           ADD         ;Figure sum
 5     %0000
               005   HERE    JMP HERE    ;Jump self to stop
 6     %1111
 7     %0110
 8     %0000
```

The EQU command lets you identify any address with a particular label. Lines 2 through 6 of Sample Program 2 use this directive. These statements make Sample Program 2 more readable by assigning descriptive labels to the 5 data addresses: NIB1 and NIB2 for the two numbers to be added; LONIB and HINIB for the low and high nibbles of the answer; and OUT for the OUT light. (See last issue's *NanoProcessor* for a complete explanation of how Program 2 uses these 4 locations.)

The other change to Program 2 in this issue is in the use of the OUT light located at the upper-left of the *NanoProcessor* screen. When you assemble Sample Program 2 and run it, you will find that the OUT light is off when the program begins, but it turns on when the program is complete. Thus, you do not need to know what address the program will end on. Instead, the OUT light signals that the program is finished.

Sample Program 3 accesses the *NanoProcessor*'s "sound chip." Any time you store a number at either location 254 or 255, the *NanoProcessor* responds with a tone. With 16 different values possible at each of these locations, you can make a total of 32 different tones. Sample Program 3 plays a C scale.

We hope that you have found these *Nano* programs instructive and enjoyable. With what you have learned, you should be able to create your own "machine-language" routines. Feel free to let us know in "Letters to the Editor" of any programs you create, so we may share them with our readers.

HCM Glossary Terms: assembler, label, object code, op-code, operand, pass, source code.

For your type-in listings, see HCM PROGRAM LISTING CONTENTS.

HCM

Three Number Systems Supported

Machine language on the *NanoProcessor* can be entered only in binary. The *NanoAssembler*, however, understands decimal and hexadecimal in addition to binary. Last issue we explained how to convert between decimal and binary—this issue we introduce you to hexadecimal.

As we explained in the previous issue, decimal is a base 10 system. It uses ten digits (0 through 9) to represent numbers. Similarly, binary is a base 2 system and uses two digits (0 and 1). Hexadecimal is a base 16 number system and uses 16 different digits—0 through 9 plus the letters A through F. (See Figure 2 for a conversion chart.) As the conversion chart shows, we can express the number 11 decimal as either the binary number %1010 or the hexadecimal number $B. (Note that the % symbol denotes a binary number, and the $ symbol a hexadecimal number.)

To convert a two-digit hexadecimal number (say $C8) to decimal you simply find the decimal equivalent of the left-most digit (i.e., $C = 12), and multiply it by 16. Then simply add the decimal equivalent of the right-most digit (12*16+8=200). Hexadecimal is a particularly useful system in assembly language because it can express any nibble as a single character or any byte as two characters.

Sample Program 2

```
Addr   Code    Line  Label   Mnemonic    Remark
               001           ORG 10
               002   NIB1    EQU $F0
               003   NIB2    EQU $F1
               004   LONIB   EQU $F8
               005   HINIB   EQU $F9
               006   OUT     EQU $FD
               007           LDA# 0      ;Turn OUT light off
 10    %0001
 11    %0000
               008           STA OUT
 12    %0011
 13    %1101
 14    %1111
               009           LDA NIB1    ;Get first number
 15    %0010
 16    %0000
 17    %1111
               010           TAB         ;Move to B
 18    %0100
               011           LDA NIB2    ;Get second number
 19    %0010
 20    %0001
 21    %1111
               012           ADD         ;Figure sum
 22    %0000
               013           STA LONIB   ;Low to memory
 23    %0011
 24    %1000
 25    %1111
               014           BCC NIB     ;One nibble answer
 26    %1110
 27    %0010
 28    %0010
               015           LDA# 1
 29    %0001
 30    %0001
               016           JMP STHI    ;All done
 31    %1111
 32    %0100
 33    %0010
               017   NIB     LDA# 0      ;Zero A
 34    %0001
 35    %0000
               018   STHI    STA HINIB   ;High to memory
 36    %0011
 37    %1001
 38    %1111
               019           LDA# ON     ;Set OUT light
 39    %0001
 40    %0001
               020           STA OUT
 41    %0011
 42    %1101
 43    %1111
               021   HERE    JMP HERE    ;Jump self to end
 44    %1111
 45    %1100
 46    %0010
               022           ORG $F0
               023           DN $A
               024           DN $C
```

Sample Program 3

```
Addr   Code    Line  Label   Mnemonic    Remark
               001   SOUND   EQU 254
               002           LDA# 2
  0    %0001
  1    %0010
               003           TAB
  2    %0100
               004           AND
  3    %1000
               005           RRC
  4    %0110
               006           STA SOUND
  5    %0011
  6    %1110
  7    %1111
               007           ADD
  8    %0000
               008           STA SOUND
  9    %0011
 10    %1110
 11    %1111
               009           ADD
 12    %0000
               010           STA SOUND
 13    %0011
 14    %1110
 15    %1111
               011           LDA# 6
 16    %0001
 17    %0110
               012           STA SOUND
 18    %0011
 19    %1110
 20    %1111
               013           ADD
 21    %0000
               014           STA SOUND
 22    %0011
 23    %1110
 24    %1111
               015           ADD
 25    %0000
               016           STA SOUND
 26    %0011
 27    %1110
 28    %1111
               017           ADD
 29    %0000
               018           STA SOUND
 30    %0011
 31    %1110
 32    %1111
               019           LDA# $D
 33    %0001
 34    %1101
               020           STA SOUND
 35    %1101
 36    %1110
 37    %1111
               021   HERE    JMP HERE
 38    %1111
 39    %0110
 40    %0010
```

NANOEDITOR *Continued*

IBM PC/PCjr, TANDY 1000

```
A  900 GOSUB 580:IF K$=CHR$(27) THEN RETUR
       N ELSE IF LI$(LN)="" THEN 900
T  910 X1=X1+2:LN=LN+1:GOTO 890
V  920 'DELETE LINE
N  930 IF LN>NL THEN COLOR 2:PRINT:PRINT"N
       O SUCH LINE":COLOR 15:GOSUB 740:RET
       URN
M  940 COLOR 11:PRINT:PRINT"   DELETING LI
       NE  ";LN$:COLOR 15
E  950 FOR IT=LN TO NL:LI$(IT)=LI$(IT+1):N
       EXT:LI$(NL)=SP$+SP$:NL=NL-1:GOSUB 7
       40
H  960 RETURN
E  970 'INSERT
Z  980 IF LN>NL OR NL=199 THEN COLOR 2:LOC
       ATE 10,5:PRINT"LAST LINE IS"NL;"CAN
       'T INSERT":COLOR 15:GOSUB 740:RETUR
       N:ELSE X1=5
S  990 FOR IT=NL TO LN STEP -1:LI$(IT+1)=L
       I$(IT):NEXT IT:NL=NL+1:LI$(LN)=SP$+
       SP$
K 1000 GOSUB 580:IF K$=CHR$(27) THEN 950 E
       LSE IF LI$(LN)="" THEN 1000
U 1010 RETURN
W 1020 COLOR 5,0:CLS:LOCATE 5,5:PRINT"SOUR
       CE CODE IS A MAXIMUM":LOCATE 6,5:PR
       INT"200 LINES!":GOSUB 740:RETURN
G 1030 'FILES
O 1040 COLOR 12,1:CLS:X=3:Y=10:W=20:H=13:G
       OSUB 1180
T 1050 LOCATE 5,12:COLOR 5:PRINT" LOAD/SAV
       E MENU";
R 1060 COLOR 15:LOCATE 8,15:PRINT"1)  SAVE
       ";:LOCATE 10,15:PRINT"2)  LOAD";:
L 1070 LOCATE 13,13,1:COLOR 14:PRINT"YOUR
       CHOICE:   ";:LOCATE ,26:K$=INPUT$(1
       )
Q 1080 PRINT K$;:IF K$=CHR$(27) THEN RETUR
       N ELSE IF K$ > "2" OR K$ < "1" THEN
         1070
U 1090 ON ASC(K$)-48 GOTO 1770,1680
I 1100 'PRINT
A 1110 FOR LN = 1 TO NL:GOSUB 520:GOSUB 73
       0:LPRINT LN$;TAB(5) A$;TAB(15) B$;T
       AB(42) C$:NEXT:RETURN
Y 1120 'END PROGRAM
Q 1130 LOCATE 20,1:COLOR 8,7:PRINT"ARE YOU
         SURE YOU WANT TO END THIS":PRINT"S
       ESSION ? (Y/N) ";
L 1140 K$=INKEY$:IF K$="" THEN 1140 ELSE C
       OLOR 14,1:PRINT K$;
F 1150 IF NOT(K$="y" OR K$="Y") THEN RETUR
       N 290
B 1160 CLS:ON ERROR GOTO 0:END
F 1170 UL=222:UR=221:LL=222:LR=221:H1=223:
       H2=220:V2=221:V1=222:GOTO 1220
R 1180 'DISPLAY A SINGLE-SIDED BOX
E 1190 UL=218:UR=191:LL=192:LR=217:H1=196:
       H2=H1:V1=179:V2=V1:GOTO 1220
X 1200 'DISPLAY A DOUBLE-SIDED BOX
Q 1210 UL=201:UR=187:LL=200:LR=188:H1=205:
       H2=H1:V1=186:V2=V1:GOTO 1220
T 1220 '  DISPLAYS A BOX
C 1230 LOCATE X, Y:PRINT CHR$(UL);STRING$(
       W-2,H1);CHR$(UR);
J 1240 LOCATE X + H - 1,Y:PRINT CHR$(LL);S
       TRING$(W-2,H2);CHR$(LR);
T 1250 FOR I=X+1 TO X+H-2:LOCATE I,Y:PRINT
         CHR$(V1);
S 1260 LOCATE I,Y+W-1:PRINT CHR$(V2);:NEXT
       :RETURN
J 1270 'INPUT ROUTINE
O 1280 SP=1:INS=0:RT=0:IF SE=1 AND LEN(S$)
       >0 THEN SP = LEN(S$):SE=0:ELSE SE=0
Y 1290 LOCATE X,Y + SP - 1,1
P 1300 K$ = INKEY$:IF K$ = "" THEN GOTO 13
       00
C 1310 IF MC$<=K$ AND XC$>=K$ THEN GOSUB 1
       410:IF RT=1 THEN K$=CHR$(9):RETURN
       ELSE GOTO 1290
H 1320 IF K$ = CHR$(0)+CHR$(77) THEN GOSUB
          1480:IF RT=1 THEN RETURN ELSE 1290
T 1330 IF K$ = CHR$(0)+CHR$(75) THEN GOSUB
          1490:IF RT=1 THEN RETURN ELSE 1290
Q 1340 IF K$ = CHR$(0) + CHR$(82) THEN GOS
       UB 1500:GOTO 1290
H 1350 IF K$ = CHR$(18) THEN GOSUB 1500:GO
       TO 1300
N 1360 IF K$ = CHR$(8) THEN GOSUB 1510:IF
       RT=1 THEN RETURN ELSE GOTO 1300
S 1370 IF K$ = CHR$(0) + CHR$(83) THEN GOS
       UB 1540:GOTO 1290
J 1380 IF K$ = CHR$(127) THEN GOSUB 1540:G
       OTO 1290
D 1390 IF K$ = CHR$(13) OR K$=CHR$(9) OR K
       $=CHR$(27) THEN RETURN
K 1400 GOTO 1290
Y 1410 CS = LEN(S$):IF INS = 0 THEN GOTO 1
       450
G 1420 IF CS >= L THEN RETURN
T 1430 S$ = LEFT$(S$, SP - 1) + K$ + RIGHT
       $(S$,CS - SP + 1):LOCATE X,Y:PRINT
       S$;
N 1440 SP=SP+1:IF SP=L+1 THEN SP=SP-1:LOCA
       TE X,Y+SP-1:RETURN:ELSE RETURN
E 1450 IF CS=L AND SP=L+1 THEN RETURN ELSE
          PRINT K$;
V 1460 IF SP < CS+1 THEN S$=LEFT$(S$,SP-1)
       +K$+RIGHT$(S$,CS-SP):ELSE S$=S$+K$
Y 1470 SP=SP+1:IF SP=L+1 THEN SP=SP-1:LOCA
       TE X,Y+SP-1:RT=1:RETURN:ELSE RETURN
V 1480 IF SP<LEN(S$)+1 AND SP<L THEN SP=SP
       +1:LOCATE X,Y+SP-1,1:RETURN:ELSE RT
       =1:RETURN
Y 1490 IF SP>1 THEN SP=SP-1:LOCATE X,Y+SP-
       1, 1:RETURN:ELSE RT=1:RETURN
M 1500 IF INS=1 THEN INS=0:RETURN:ELSE INS
       =1:RETURN
U 1510 CS=LEN(S$):IF CS>0 AND SP<=CS+1 AND
         SP>1 THEN S$=LEFT$(S$,SP-2)+RIGHT$
       (S$,CS-SP+1):LOCATE X,Y:PRINT S$;"
       ";:SP=SP-1:LOCATE X,Y+SP-1
R 1520 IF CS < 1 THEN RT=1
J 1530 RETURN
N 1540 CS=LEN(S$):IF CS>0 AND SP<=CS THEN
       S$=LEFT$(S$,SP-1)+RIGHT$(S$,CS-SP):
       LOCATE X,Y:PRINT S$;" ";:LOCATE X,Y
       +SP-1:IF SP>CS+1 THEN SP=CS+1
N 1550 RETURN
R 1560 'TRAP FILE NAME ERROR
T 1570 GOSUB 1590:RESUME 1580
C 1580 RETURN 280
L 1590 'DISPLAY FILE ERROR MESSAGE
N 1600 COLOR 8,1:CLS:X=7:Y=5:W=30:H=5:GOSU
       B 1210
A 1610 COLOR 0,5:LOCATE 9,9,0:PRINT "   INV
       ALID FILE NAME   ";
S 1620 FOR J = 1 TO 4000:NEXT:LOCATE ,,1:C
       OLOR 15,0
M 1630 RETURN
H 1640 'GET FILE NAME
D 1650 CLS:WIDTH 40:X=5:Y=5:W=30:H=5:COLOR
          4,1:GOSUB 1210:COLOR 15:LOCATE X+2
       ,(40-LEN(P$))/2:PRINT P$;
G 1660 X=11:Y=1:W=39:H=5:COLOR 15:GOSUB 11
       80:LOCATE 13,3:PRINT"FILE NAME: ";
F 1670 X=13:Y=14:MC$="!":XC$="z":COLOR 14:
       S$="":L=20:GOSUB 1270:F$=S$:RETURN
K 1680 'LOAD FILE
Y 1690 P$ = "LOAD SOURCE FILE":GOSUB 1650:
       ON ERROR GOTO 1560
C 1700 OPEN "I", 1, F$ + ".S"
Q 1710 INPUT #1,NL:FOR I = 1 TO NL
E 1720 INPUT #1, S$:QT = 1
I 1730 WHILE (QT>0):QT=INSTR(QT,S$,CHR$(2)
       ):IF QT>0 THEN MID$(S$,QT,1)=CHR$(3
       4)
N 1740 WEND          'REPLACE QUOTATION MAR
       KS...
J 1750 LI$(I) = S$:NEXT:CLOSE:LN=1:FOR I=N
       L+1 TO 200:LI$(I)=SP$+SP$:NEXT
W 1760 RETURN
H 1770 'SAVE FILE
F 1780 CLS:WIDTH 40:P$ = "SAVE SOURCE FILE
       "
E 1790 GOSUB 1640:ON ERROR GOTO 1560
N 1800 OPEN "O", 1, F$ + ".S"
N 1810 WRITE #1, NL
I 1820 FOR I = 1 TO NL:S$ = LI$(I):QT = 1
Z 1830 WHILE (QT>0):QT= INSTR(QT,S$,CHR$(3
       4)):IF QT>0 THEN MID$(S$,QT,1)=CHR$
       (2)
O 1840 WEND          'REPLACE QUOTATION MARK
       S WITH NON-VOLATILE CHARACTERS
R 1850 WRITE #1, S$:NEXT:CLOSE
V 1860 RETURN
```

HCM

NANOASSEMBLER

TI-99/4A

```
P  100 REM ******************
Q  110 REM * NANOASSEMBLER  *
P  120 REM ******************
M  130 REM COPYRIGHT 1985
H  140 REM EMERALD VALLEY PUBLISHING CO
A  150 REM BY ROGER WOOD
R  160 REM HOME COMPUTER MAGAZINE
G  170 REM VERSION 5.6.1
T  180 REM TI BASIC AND EXTENDED BASIC
M  190 REM
F  200 CALL CLEAR
W  210 PRINT TAB(7);"The NanoAssembler": :
         : :TAB(6);"PLACE ALPHA LOCK DOWN":
         :"    PLEASE WAIT WHILE I SET UP"
J  220 DIM OP$(18)
P  230 DIM NN(18)
Q  240 DIM LI$(200)
Z  250 DIM LB$(25)
W  260 DIM AD(255)
Z  270 GOSUB 2990
A  280 RESTORE 2820
M  290 GOSUB 2740
B  300 GOSUB 2910
Q  310 IF (K<49)+(K>50)THEN 300
X  320 GOSUB 1360
L  330 IF FL$="" THEN 280
X  340 GOSUB 1630
L  350 GOSUB 1860
I  360 GOSUB 470
O  370 RESTORE 2840
Q  380 GOSUB 2740
E  390 GOSUB 2910
```

Continued

```
L  400 IF (K<49)+(K>50)THEN 370
D  410 GOSUB 1430
W  420 IF FL$="" THEN 380
P  430 GOSUB 1700
S  440 GOTO 1970
N  450 STOP
C  460 REM ASSEMBLE
R  470 PRINT "EXECUTING FIRST PASS"
N  480 CA=0
K  490 LB=0
B  500 FOR LN=1 TO NL
J  510 LFLAG=0
V  520 GOSUB 1560
Y  530 IF SEG$(TL$,1,6)="      " THEN 750
U  540 IF LB=24 THEN 1230
L  550 LB$(LB)=SEG$(TL$,1,6)
P  560 X=POS(LB$(LB)," ",1)
D  570 IF X=0 THEN 640
B  580 FOR IT=6 TO X STEP -1
Z  590 IF SEG$(LB$(LB),IT,1)=" " THEN 610
F  600 GOTO 1210
I  610 LB$(LB)=SEG$(LB$(LB),1,IT-1)
U  620 NEXT IT
N  630 IF LB=0 THEN 670
C  640 FOR IT=0 TO LB-1
F  650 IF LB$(LB)=SEG$(LB$(IT),2,LEN(LB$(I
       T)))THEN 1250
K  660 NEXT IT
G  670 IF (ASC(LB$(LB))>64)*(ASC(LB$(LB))<
       91)THEN 690
Z  680 GOTO 1210
B  690 FOR IT=1 TO LEN(LB$(LB))
F  700 X=ASC(SEG$(LB$(LB),IT,1))
Q  710 IF ((X>64)*(X<92))+((X>47)*(X<58))T
       HEN 730
E  720 GOTO 1210
Q  730 NEXT IT
H  740 LFLAG=1
R  750 IF SEG$(TL$,7,1)<>" " THEN 1190
P  760 GOSUB 1780
N  770 IF CA+NN(OP)>256 THEN 1310
A  780 ON OP+1 GOTO 1990,2210,2120,2120,19
       90,1990,1990,1990,1990,1990,1990,21
       20,2120,2120,2120,2120,2020,2060,22
       60
X  790 IF LFLAG=0 THEN 820
N  800 LB$(LB)=CHR$(CA)&LB$(LB)
C  810 LB=LB+1
O  820 CA=CA+NN(OP)
W  830 NEXT LN
G  840 PRINT "EXECUTING SECOND PASS"
I  850 CA=0
G  860 FOR LN=1 TO NL
H  870 GOSUB 1560
E  880 GOSUB 1780
Z  890 EA=CA+NN(OP)
D  900 IF OP=17 THEN 1050
M  910 IF OP<>18 THEN 950
P  920 GOSUB 2420
B  930 EA=V
U  940 GOTO 1050
F  950 IF NN(OP)<3 THEN 1070
D  960 IF AD(CA+1)<>-1 THEN 1070
Y  970 FOR IT=0 TO LB
Y  980 IF SEG$(TL$,X+1,LL)=SEG$(LB$(IT),2,
       LEN(LB$(IT)))THEN 1010
A  990 NEXT IT
G 1000 GOTO 1290
R 1010 V=ASC(LB$(IT))
C 1020 AD(CA+1)=V-INT(V/16)*16
I 1030 AD(CA+2)=INT(V/16)
I 1040 GOTO 1070
A 1050 PRINT #PR:"       ";LN;LI$(LN)
I 1060 GOTO 1120
K 1070 PRINT #PR:"       ";LN;LI$(LN)
F 1080 FOR IT=0 TO NN(OP)-1
O 1090 GOSUB 1500
B 1100 PRINT #PR:OT$
F 1110 NEXT IT
R 1120 CA=EA
U 1130 NEXT LN
L 1140 PRINT "PRESS ANY KEY TO CONTINUE"
X 1150 GOSUB 2910
H 1160 IF PR=0 THEN 1180
L 1170 CLOSE #1
C 1180 RETURN
K 1190 PRINT #PR:"ILLEGAL CODE IN LINE";LN
Z 1200 GOTO 1330
W 1210 PRINT #PR:"ILLEGAL LABEL IN LINE";L
       N
R 1220 GOTO 1330
Z 1230 PRINT #PR:"TOO MANY LABELS--
              25 MAXIMUM"
R 1240 GOTO 1330
P 1250 PRINT #PR:"DUPLICATE LABEL IN LINE"
       ;LN
Z 1260 GOTO 1330
P 1270 PRINT #PR:"NUMBER OUT OF RANGE IN L
       INE";LN
P 1280 GOTO 1330
M 1290 PRINT #PR:"ILLEGAL USE OF LABEL IN
       LINE";LN
K 1300 GOTO 1330
N 1310 PRINT #PR:"ADDRESS OUT OF RANGE IN
       LINE";LN
C 1320 GOTO 1330
T 1330 GOSUB 2950
Y 1340 GOTO 1970
D 1350 REM FILES
H 1360 ON K-48 GOTO 1370,1410
Z 1370 PRINT "INPUT FILE NAME FOR DSK1"
O 1380 INPUT FL$
S 1390 FL$="DSK1."&SEG$(FL$,1,8)&"_S"
T 1400 GOTO 1420
A 1410 FL$="CS1"
F 1420 RETURN
I 1430 ON K-48 GOTO 1440,1480
N 1440 PRINT "INPUT FILE NAME FOR DSK1"
K 1450 INPUT FL$
T 1460 FL$="DSK1."&SEG$(FL$,1,10)
N 1470 GOTO 1490
D 1480 FL$="CS1"
O 1490 RETURN
I 1500 OT$=STR$(CA+IT)
A 1510 OT$=SEG$("000",1,3-LEN(OT$))&OT$
R 1520 OU$=STR$(AD(CA+IT))
T 1530 OU$=SEG$("000",1,3-LEN(OU$))&OU$
T 1540 OT$=OT$&" "&OU$
N 1550 RETURN
A 1560 TL$=SEG$(LI$(LN),1,24)
O 1570 FOR IT=LEN(TL$)TO 1 STEP -1
I 1580 IF SEG$(TL$,IT,1)<>" " THEN 1610
L 1590 TL$=SEG$(TL$,1,LEN(TL$)-1)
H 1600 NEXT IT
U 1610 LL=IT
N 1620 RETURN
D 1630 OPEN #1:FL$,INTERNAL,INPUT ,FIXED 6
       4
B 1640 INPUT #1:NL
V 1650 FOR IT=1 TO NL
C 1660 INPUT #1:LI$(IT)
G 1670 NEXT IT
Z 1680 CLOSE #1
W 1690 RETURN
G 1700 OPEN #1:FL$,INTERNAL,OUTPUT,FIXED 6
       4
A 1710 PRINT "SAVING FILE. . ."
Q 1720 FOR IT=0 TO 245 STEP 7
K 1730 PRINT #1:AD(IT);AD(IT+1);AD(IT+2);A
       D(IT+3);AD(IT+4);AD(IT+5);AD(IT+6)
A 1740 NEXT IT
N 1750 PRINT #1:AD(252);AD(253);AD(254);AD
       (255)
A 1760 CLOSE #1
D 1770 RETURN
B 1780 FOR IT=0 TO 18
Z 1790 IF SEG$(TL$,8,LEN(OP$(IT)))=OP$(IT)
       THEN 1820
N 1800 NEXT IT
M 1810 GOTO 1190
S 1820 OP=IT
Z 1830 X=8+LEN(OP$(OP))
K 1840 IF (OP=9)*(SEG$(TL$,X,1)="G")THEN 1
       800
Q 1850 RETURN
S 1860 PRINT "SEND ASSEMBLY TO:": :"1. SCR
       EEN":"2. PRINTER"
W 1870 GOSUB 2860
H 1880 IF K=50 THEN 1910
V 1890 PR=0
S 1900 RETURN
A 1910 PRINT "ENTER PRINTER DEVICE :"
K 1920 INPUT PR$
V 1930 PR=1
Z 1940 OPEN #PR:PR$
N 1950 RETURN
M 1960 REM END PROGRAM
U 1970 PRINT
P 1980 END
K 1990 IF X<>11 THEN 1190
E 2000 AD(CA)=OP
A 2010 GOTO 790
O 2020 GOSUB 2420
O 2030 IF V=-1 THEN 1290
T 2040 AD(CA)=V
G 2050 GOTO 790
W 2060 GOSUB 2420
O 2070 IF V=-1 THEN 1290
U 2080 IF LFLAG=0 THEN 1190
A 2090 LB$(LB)=CHR$(V)&LB$(LB)
I 2100 LB=LB+1
A 2110 GOTO 830
N 2120 GOSUB 2420
I 2130 AD(CA)=OP
O 2140 IF V<>-1 THEN 2180
T 2150 AD(CA+1)=-1
V 2160 AD(CA+2)=-1
D 2170 GOTO 790
U 2180 AD(CA+1)=V-INT(V/16)*16
Q 2190 AD(CA+2)=INT(V/16)
L 2200 GOTO 790
N 2210 GOSUB 2420
F 2220 IF V=-1 THEN 1290
T 2230 AD(CA)=OP
V 2240 AD(CA+1)=V
U 2250 GOTO 790
S 2260 GOSUB 2420
T 2270 IF (V=-1)+(LFLAG=1)THEN 1290
X 2280 CA=V
N 2290 GOTO 830
Y 2300 V=0
W 2310 FOR IT=0 TO LEN(T$)-1
Y 2320 V=V+VAL(SEG$(T$,IT+1,1))*2^(LEN(T$)
       -IT-1)
E 2330 NEXT IT
F 2340 IF (((OP=1)+(OP=16))*(V>15))+(V>255
       )THEN 1270
G 2350 RETURN
```

Continued

NANOASSEMBLER *Continued*

TI-99/4A

```
K 2360 V=0
Q 2370 FOR IT=0 TO LEN(T$)-1
O 2380 V=V+16^(LEN(T$)-IT-1)*(POS(HX$,SEG$
       (T$,IT+1,1),1)-1)
C 2390 NEXT IT
G 2400 IF (((OP=1)+(OP=16))*(V>15))+(V>255
       )THEN 1270
F 2410 RETURN
B 2420 IF SEG$(TL$,X,1)<>" " THEN 1270
A 2430 IF SEG$(TL$,X+1,1)="" THEN 1190
A 2440 Y=POS(NM$,SEG$(TL$,X+1,1),1)
K 2450 IF Y<>0 THEN 2480
F 2460 V=-1
D 2470 RETURN
F 2480 IF Y<=2 THEN 2500
Y 2490 Y=3
A 2500 ON Y GOTO 2510,2580,2650
N 2510 T$=""
A 2520 FOR IT=X+2 TO LL
Q 2530 N$=(SEG$(TL$,IT,1))
Z 2540 IF (N$<"0")+(N$>"1")THEN 1270
N 2550 T$=T$&N$
T 2560 NEXT IT
A 2570 GOTO 2300
K 2580 T$=""
R 2590 FOR IT=X+2 TO LL
Y 2600 N$=(SEG$(TL$,IT,1))
G 2610 IF POS(HX$,N$,1)=0 THEN 1270
V 2620 T$=T$&N$
L 2630 NEXT IT
G 2640 GOTO 2360
Q 2650 T$=""
M 2660 FOR IT=X+1 TO LL
X 2670 N$=(SEG$(TL$,IT,1))
A 2680 IF (N$<"0")+(N$>"9")THEN 1270
K 2690 T$=T$&N$
N 2700 NEXT IT
N 2710 V=VAL(T$)
N 2720 RETURN
E 2730 REM PRINT MENU
S 2740 CALL CLEAR
P 2750 READ NI
B 2760 FOR IT=1 TO NI
K 2770 READ A,A$
F 2780 PRINT :TAB(A);A$
Y 2790 NEXT IT
L 2800 PRINT : : : : :
N 2810 RETURN
E 2820 DATA 7,5,The NanoAssembler,1," ",1,
       LOAD SOURCE FILE FROM:
L 2830 DATA 8,(1) DISK (DSK1),8,(2) CASSET
       TE (CS1),1," ",5,YOUR CHOICE:
K 2840 DATA 7,5,The NanoAssembler,1," ",7,
       SAVE OBJECT FILE TO:
N 2850 DATA 8,(1) DISK (DSK1),8,(2) CASSET
       TE (CS1),1," ",5,YOUR CHOICE:
A 2860 CALL KEY(0,K,S)
N 2870 IF S=0 THEN 2860
E 2880 IF (K<49)+(K>50)THEN 2870
R 2890 PRINT CHR$(K)
N 2900 RETURN
I 2910 CALL KEY(0,K,S)
R 2920 IF S=0 THEN 2910
R 2930 CALL HCHAR(18,20,K)
N 2940 RETURN
E 2950 FOR IT=1 TO 1000
F 2960 NEXT IT
I 2970 RETURN
R 2980 REM PROGRAM INIT
X 2990 RESTORE 3060
K 3000 FOR IT=0 TO 18
J 3010 READ OP$(IT),NN(IT)
P 3020 NEXT IT
F 3030 HX$="0123456789ABCDEF"
J 3040 NM$="%$0123456789"
G 3050 RETURN
E 3060 DATA ADD,1,LDA#,2,LDA,3,STA,3,TAB,1
       ,TBA,1,RRC,1,RLC,1,AND,1,OR,1,XOR,1
A 3070 DATA BZ,3,BNZ,3,BCS,3,BCC,3,JMP,3,D
       N,1,EQU,0,ORG,0
```

HCM

NANOEDITOR

TI-99/4A

```
P  100 REM **************
Q  110 REM * NANOEDITOR *
P  120 REM **************
M  130 REM COPYRIGHT 1985
H  140 REM EMERALD VALLEY PUBLISHING CO
A  150 REM BY ROGER WOOD
R  160 REM HOME COMPUTER MAGAZINE
G  170 REM VERSION 5.6.1
T  180 REM TI BASIC AND EXTENDED BASIC
M  190 REM
F  200 CALL CLEAR
D  210 PRINT TAB(9);"The NanoEditor": : :
       :TAB(6);"PLACE ALPHA LOCK DOWN": :"
          PLEASE WAIT WHILE I SET UP"
Q  220 DIM LI$(200)
T  230 GOSUB 3450
A  240 RESTORE 3320
V  250 GOSUB 3120
J  260 GOSUB 3370
Y  270 IF (K<49)+(K>52)THEN 260
E  280 ON K-48 GOSUB 310,1590,2030,2160
W  290 GOTO 240
D  300 REM EDIT
D  310 CALL CLEAR
A  320 LN$="CMD"
I  330 L=1
Y  340 RW=1
D  350 B=ASC(SEG$(CMD$,1,1))
V  360 T=ASC(SEG$(CMD$,LEN(CMD$),1))
R  370 CN$=SEG$(C$,1,3)
Z  380 S$=""
C  390 E=0
B  400 GOSUB 2250
W  410 IF ESC=0 THEN 430
S  420 RETURN
K  430 OP=POS(CMD$,S$,1)
Q  440 IF (S$="")+(OP=0)THEN 380
H  450 IF OP=1 THEN 610
M  460 CN$=SEG$(C$,1,6)
S  470 LN$="L#?"
A  480 L=3
R  490 RW=2
S  500 B=ASC("0")
Y  510 T=ASC("9")
H  520 S$=""
Z  530 GOSUB 2250
E  540 IF ESC=0 THEN 560
S  550 RETURN
U  560 IF LEN(S$)=0 THEN 520
M  570 LN=VAL(S$)
I  580 IF (LN>0)*(LN<201)THEN 600
W  590 GOTO 520
E  600 LN$=SEG$("000",1,3-LEN(S$))&S$
A  610 ON OP GOSUB 640,850,970,1160,1360
B  620 GOTO 310
X  630 REM ADD
D  640 IF NL=200 THEN 790
W  650 LN=NL+1
Q  660 GOSUB 3230
B  670 FOR RW=3 TO 21 STEP 2
N  680 GOSUB 3200
B  690 S$=""
A  700 GOSUB 2250
M  710 IF ESC=0 THEN 730
J  720 RETURN
H  730 IF LEN(S$)=0 THEN 690
P  740 LI$(LN)=S$
D  750 NL=LN
A  760 LN=LN+1
Y  770 IF NL=200 THEN 810
N  780 NEXT RW
B  790 CALL CLEAR
I  800 GOTO 670
G  810 PRINT "SOURCE CODE IS A MAXIMUM OF
       200 LINES!"
Y  820 GOSUB 3410
F  830 RETURN
A  840 REM DELETE
X  850 IF LN<=NL THEN 890
F  860 PRINT "NO SUCH LINE"
P  870 GOSUB 3410
C  880 RETURN
D  890 PRINT "DELETING LINE";LN
H  900 FOR IT=LN TO NL
R  910 LI$(IT)=LI$(IT+1)
T  920 NEXT IT
J  930 NL=NL-1
S  940 LI$(NL+1)=""
S  950 RETURN
R  960 REM EDIT
P  970 IF LN<=NL THEN 1010
K  980 PRINT "THERE IS NO LINE ";LN$
E  990 GOSUB 3410
Z 1000 RETURN
I 1010 GOSUB 3230
L 1020 FOR RW=3 TO 21 STEP 2
V 1030 GOSUB 3200
F 1040 S$=LI$(LN)
F 1050 GOSUB 2250
Y 1060 IF ESC=0 THEN 1080
G 1070 RETURN
W 1080 IF LEN(S$)=0 THEN 890
K 1090 LI$(LN)=S$
N 1100 IF NL=LN THEN 950
S 1110 LN=LN+1
I 1120 NEXT RW
A 1130 CALL CLEAR
E 1140 GOTO 1020
P 1150 REM INSERT
C 1160 IF (LN<=NL)*(NL<200)THEN 1200
O 1170 PRINT "LAST LINE IS";NL;"CAN'T INSE
       RT"
C 1180 GOSUB 3410
N 1190 RETURN
A 1200 RW=3
W 1210 GOSUB 3230
C 1220 GOSUB 3200
T 1230 GOSUB 3280
X 1240 S$=""
P 1250 GOSUB 2250
U 1260 IF ESC=0 THEN 1280
E 1270 RETURN
T 1280 IF LEN(S$)=0 THEN 1250
M 1290 FOR IT=NL TO LN STEP -1
N 1300 LI$(IT+1)=LI$(IT)
R 1310 NEXT IT
```

Continued

NANOEDITOR *Continued*

```
F 1320 NL=NL+1
R 1330 LI$(LN)=S$
Y 1340 RETURN
D 1350 REM LIST
P 1360 IF LN<=NL THEN 1380
N 1370 GOTO 980
D 1380 CALL CLEAR
N 1390 FOR RW=3 TO 21 STEP 2
H 1400 S$=LI$(LN)
D 1410 GOSUB 3200
O 1420 GOSUB 3280
D 1430 GOSUB 3000
A 1440 GOSUB 3060
A 1450 IF LN<NL THEN 1490
A 1460 PRINT " <PRESS ANY KEY TO CONTINUE>
       "
V 1470 GOSUB 3370
D 1480 RETURN
R 1490 LN=LN+1
Q 1500 NEXT RW
M 1510 PRINT "      <PRESS FCTN 9 TO QUIT>
             <OR RETURN TO CONTINUE>"
V 1520 GOSUB 3370
Y 1530 KS=POS(CN$,CHR$(K),1)
K 1540 IF (KS<1)+(KS>2)THEN 1520
F 1550 IF KS=1 THEN 1380
S 1560 RETURN
S 1570 CA=0
K 1580 REM FILES
N 1590 RESTORE 3340
B 1600 GOSUB 1850
C 1610 IF K=51 THEN 1690
X 1620 DC=K-48
L 1630 RESTORE 3350
R 1640 GOSUB 1850
G 1650 IF K=51 THEN 1690
D 1660 SL=K-48
J 1670 ON DC GOSUB 1700,1750
Z 1680 CLOSE #1
W 1690 RETURN
E 1700 PRINT "FILE NAME FOR DSK1"
Z 1710 INPUT FL$
U 1720 IF FL$="" THEN 1770
M 1730 FL$="DSK1."&SEG$(FL$,1,8)&"_S"
H 1740 GOTO 1760
T 1750 FL$="CS1"
U 1760 ON SL GOSUB 1780,1810
D 1770 RETURN
K 1780 OPEN #1:FL$,INTERNAL,OUTPUT,FIXED 6
       4
A 1790 PRINT "SAVING FILE. . ."
Q 1800 GOTO 1890
P 1810 OPEN #1:FL$,INTERNAL,INPUT ,FIXED 6
       4
Y 1820 PRINT "LOADING FILE. . ."
A 1830 GOSUB 1940
N 1840 RETURN
O 1850 GOSUB 3120
A 1860 GOSUB 3370
Z 1870 IF (K<49)+(K>51)THEN 1860
L 1880 RETURN
F 1890 PRINT #1:NL
P 1900 FOR IT=1 TO NL
L 1910 PRINT #1:LI$(IT)
M 1920 NEXT IT
J 1930 RETURN
Q 1940 INPUT #1:NL
I 1950 FOR IT=1 TO NL
P 1960 INPUT #1:LI$(IT)
F 1970 NEXT IT
I 1980 RETURN
E 1990 FOR IT=1 TO 200
Y 2000 LI$(IT)=""
R 2010 NEXT IT
Y 2020 REM PRINT
L 2030 RESTORE 3360
Q 2040 GOSUB 3120
H 2050 INPUT PR$
S 2060 IF PR$<>"" THEN 2080
P 2070 RETURN
I 2080 OPEN #1:PR$
V 2090 FOR LN=1 TO NL
M 2100 GOSUB 3200
N 2110 PRINT #1:LN$;"  ";LI$(LN)
U 2120 NEXT LN
A 2130 CLOSE #1
H 2140 RETURN
M 2150 REM END PROGRAM
K 2160 PRINT "ARE YOU SURE YOU WANT TO END
       THIS SESSION? (Y/N) ";
I 2170 CALL KEY(0,K,S)
T 2180 IF S=0 THEN 2170
I 2190 IF (K<>89)*(K<>78)THEN 2170
G 2200 PRINT CHR$(K)
T 2210 IF K=89 THEN 2230
O 2220 RETURN
A 2230 CALL CLEAR
L 2240 END
T 2250 ESC=0
L 2260 GOSUB 3280
Q 2270 IF S$="" THEN 2290
P 2280 GOSUB 3040
Q 2290 GOSUB 3000
F 2300 CALL GCHAR(R,C,G)
M 2310 CALL HCHAR(R,C,95)
L 2320 CALL KEY(0,K,S)
D 2330 CALL HCHAR(R,C,G)
Z 2340 IF S=0 THEN 2310
E 2350 IF (K<B)+(K>T)THEN 2410
S 2360 CALL HCHAR(R,C,K)
T 2370 CALL SOUND(10,440,0)
N 2380 GOSUB 2880
X 2390 E=(E*-(E>=X))+(X*-(E<X))
A 2400 GOTO 2300
N 2410 S=POS(CN$,CHR$(K),1)+1
I 2420 ON S GOTO 2300,2770,2730,2450,2380,
       2430,2580,2640
C 2430 GOSUB 2940
K 2440 GOTO 2300
D 2450 Y=R
S 2460 Z=C
E 2470 FOR I=X TO E
G 2480 K=R
K 2490 S=C
I 2500 GOSUB 2900
S 2510 CALL GCHAR(R,C,G)
B 2520 CALL HCHAR(K,S,G)
M 2530 NEXT I
U 2540 E=E-1
A 2550 R=Y
N 2560 C=Z
A 2570 GOTO 2300
I 2580 NS=23
G 2590 IF R=RW THEN 2600
C 2600 NS=55
Y 2610 GOSUB 3000
U 2620 CALL HCHAR(R,C,32,NS)
X 2630 GOTO 2300
N 2640 IF C>=14 THEN 2680
S 2650 C=14
D 2660 X=7-(R>RW)*24
V 2670 GOTO 2720
G 2680 IF R>RW THEN 2720
K 2690 C=7
G 2700 X=24
P 2710 R=R+1
X 2720 GOTO 2300
B 2730 ESC=1
Y 2740 IF S$="" THEN 2760
V 2750 GOSUB 3040
D 2760 RETURN
D 2770 GOSUB 3000
Z 2780 S$=""
M 2790 CALL GCHAR(R,C,G)
G 2800 S$=S$&CHR$(G)
A 2810 GOSUB 2890
P 2820 IF E+1<>X THEN 2790
O 2830 IF LEN(S$)=0 THEN 2870
Y 2840 IF SEG$(S$,LEN(S$),1)<>" " THEN 287
       0
I 2850 S$=SEG$(S$,1,LEN(S$)-1)
A 2860 GOTO 2830
L 2870 RETURN
D 2880 IF X=L-1 THEN 2930
A 2890 X=X+1
E 2900 C=C+1
Z 2910 R=R-(C>30)
I 2920 C=C+(C>30)*24
K 2930 RETURN
W 2940 IF X=0 THEN 2990
P 2950 X=X-1
S 2960 C=C-1
J 2970 R=R+(C<7)
Y 2980 C=C-(C<7)*24
G 2990 RETURN
X 3000 R=RW
F 3010 C=7
Q 3020 X=0
U 3030 RETURN
S 3040 E=LEN(S$)
N 3050 GOSUB 3000
J 3060 IF X=LEN(S$)THEN 3100
I 3070 CALL HCHAR(R,C,ASC(SEG$(S$,X+1,1)))
M 3080 GOSUB 2890
W 3090 GOTO 3060
E 3100 RETURN
P 3110 REM PRINT MENU
V 3120 CALL CLEAR
A 3130 READ NI
K 3140 FOR IT=1 TO NI
B 3150 READ A,A$
S 3160 PRINT :TAB(A);A$
R 3170 NEXT IT
X 3180 PRINT : : : : :
P 3190 RETURN
M 3200 LN$=STR$(LN)
S 3210 LN$=SEG$("000",1,3-LEN(LN$))&LN$
P 3220 RETURN
R 3230 L=48
E 3240 B=ASC(" ")
R 3250 T=ASC("Z")
O 3260 CN$=C$
Y 3270 RETURN
H 3280 FOR IT=3 TO 5
A 3290 CALL HCHAR(RW,IT,ASC(SEG$(LN$,IT-2,
       1)))
G 3300 NEXT IT
X 3310 RETURN
X 3320 DATA 7,8,The NanoEditor,10,(1) EDIT
       ,10,(2) FILES,10,(3) PRINT,10,(4) E
       XIT
N 3330 DATA 1," ",5,"YOUR CHOICE: "
J 3340 DATA 6,5,FILE DEVICE MENU,7,(1) DIS
       K (DSK1),7,(2) CASSETTE (CS1),7,(3)
        EXIT,1," ",5,"YOUR CHOICE: "
```

Continued

NANOEDITOR *Continued*

TI-99/4A

```
U3350 DATA 6,7,LOAD / SAVE MENU,10,(1) SA
      VE,10,(2) LOAD,10,(3) EXIT,1," ",5,
      "YOUR CHOICE: "
W3360 DATA 4,10,PRINT FILE,1," ",3,ENTER
      PRINTER DEVICE,3,OR PRESS [ENTER] T
      O EXIT,
P3370 CALL KEY(0,K,S)
D3380 IF S=0 THEN 3370
M3390 CALL HCHAR(18,20,K)
```

```
F3400 RETURN
U3410 FOR IT=1 TO 1000
N3420 NEXT IT
I3430 RETURN
X3440 REM PROGRAM INIT
M3450 C$=CHR$(13)&CHR$(15)&CHR$(3)&CHR$(9
      )&CHR$(8)&CHR$(7)&CHR$(1)
L3460 CMD$="ADEIL"
O3470 RETURN
```

HCM

BASID'S BASIC LOADER

COMMODORE 64

```
H  100 REM ************************
M  110 REM * BASID'S BASIC LOADER *
T  120 REM ************************
M  130 REM COPYRIGHT 1985
R  140 REM EMERALD VALLEY PUBLISHING CO.
F  150 REM BY RANDY THOMPSON
N  160 REM HOME COMPUTER MAGAZINE
A  170 REM VERSION 5.6.1
P  180 REM C-64 BASIC
T  190 PRINT"■SHIFT CLR■INITIALIZING..."
M  200 FOR I=49152 TO 50461:READ D:C=C+D:P
       OKE I,D:NEXT
N  210 IF C<>157106 THEN PRINT"** ERROR IN
        DATA STATEMENTS **":END
J  220 SYS 49152:PRINT"** BA■CTRL RVSON■SI
       D■CTRL RVSOFF■ ACTIVATED **"
M  230 DATA 169,51,141,4,3,169,192,141
J  240 DATA 5,3,169,246,141,6,3,169
S  250 DATA 192,141,7,3,169,43,141,8
Y  260 DATA 3,169,193,141,9,3,169,82
F  270 DATA 141,10,3,169,193,141,11,3
O  280 DATA 169,132,141,0,3,169,193,141
V  290 DATA 1,3,96,166,122,160,4,132
G  300 DATA 15,189,0,2,16,7,201,255
P  310 DATA 240,62,232,208,244,201,32,240
Q  320 DATA 55,133,8,201,34,240,85,36
H  330 DATA 15,112,45,201,63,208,4,169
Z  340 DATA 153,208,37,201,48,144,4,201
S  350 DATA 60,144,29,132,113,160,0,132
P  360 DATA 11,136,134,122,202,200,232,189
E  370 DATA 0,2,56,249,158,160,240,245
L  380 DATA 201,128,208,47,5,11,164,113
M  390 DATA 232,200,153,251,1,201,0,240
O  400 DATA 56,56,233,58,240,4,201,73
E  410 DATA 208,2,133,15,56,233,85,208
I  420 DATA 160,133,8,189,0,2,240,224
A  430 DATA 197,8,240,220,200,153,251,1
E  440 DATA 232,208,240,166,122,230,11,200
Q  450 DATA 185,157,160,16,250,185,158,160
L  460 DATA 208,181,240,15,189,0,2,16
R  470 DATA 189,153,253,1,198,123,169,255
Y  480 DATA 133,122,96,160,0,185,153,193
Z  490 DATA 208,2,200,232,189,0,2,56
L  500 DATA 249,153,193,240,245,201,128,20
       8
J  510 DATA 4,5,11,208,153,166,122,230
Q  520 DATA 11,200,185,152,193,16,250,185
P  530 DATA 153,193,208,224,240,198,16,15
W  540 DATA 36,15,48,11,201,255,240,7
I  550 DATA 201,204,176,6,76,36,167,76
R  560 DATA 243,166,56,233,203,170,132,73
E  570 DATA 160,255,202,240,8,200,185,153
O  580 DATA 193,16,250,48,245,200,185,153
Z  590 DATA 193,48,5,32,71,171,208,245
K  600 DATA 76,239,166,32,115,0,32,52
F  610 DATA 193,76,174,167,201,204,144,4
J  620 DATA 201,223,144,6,32,121,0,76
R  630 DATA 237,167,56,233,204,10,170,189
C  640 DATA 226,193,72,189,225,193,72,76
B  650 DATA 115,0,169,0,133,13,32,115
L  660 DATA 0,201,204,144,4,201,221,144
G  670 DATA 6,32,121,0,76,141,174,56
Z  680 DATA 233,204,10,72,32,115,0,32
A  690 DATA 241,174,104,168,185,7,194,133
C  700 DATA 85,185,8,194,133,86,32,84
R  710 DATA 0,76,141,173,224,11,208,14
Y  720 DATA 201,76,144,10,201,95,176,6
P  730 DATA 56,233,76,76,69,193,76,139
H  740 DATA 227,86,79,204,70,82,209,87
B  750 DATA 65,86,197,65,84,84,65,67
V  760 DATA 203,68,69,67,65,217,83,85
G  770 DATA 211,82,69,204,83,79,85,78
O  780 DATA 196,70,73,76,84,69,210,76
K  790 DATA 207,66,65,206,72,201,82,73
P  800 DATA 78,199,83,89,78,195,82,69
Z  810 DATA 211,67,85,212,80,85,76,83
K  820 DATA 197,76,83,212,73,78,73,212
T  830 DATA 0,170,194,195,194,209,194,13
Z  840 DATA 195,38,195,250,194,19,195,44
H  850 DATA 195,65,195,106,195,110,195,114
N  860 DATA 195,149,195,170,195,191,195,21
       7
A  870 DATA 195,252,195,18,196,46,196,58
P  880 DATA 196,67,196,86,196,115,196,128
J  890 DATA 196,121,196,140,196,152,196,16
       4
I  900 DATA 196,184,196,188,196,192,196,20
       5
W  910 DATA 196,221,196,237,196,243,196,11
G  920 DATA 197,157,0,212,157,145,194,96
W  930 DATA 32,235,183,76,66,194,32,247
H  940 DATA 183,201,0,240,3,76,72,178
K  950 DATA 166,20,138,202,240,14,202,240
```

```
Q  960 DATA 9,202,240,3,76,72,178,162
O  970 DATA 14,96,162,7,96,32,48,194
W  980 DATA 165,21,208,240,165,20,201,16
N  990 DATA 176,234,133,2,96,32,85,194
N 1000 DATA 10,10,10,10,133,2,96,189
U 1010 DATA 150,194,74,74,74,74,168,76
H 1020 DATA 162,179,32,48,194,165,21,240
E 1030 DATA 3,76,72,178,165,20,201,2
Z 1040 DATA 176,247,232,232,232,232,201,0
L 1050 DATA 96,0,0,0,0,0,0,0
F 1060 DATA 0,0,0,0,0,0,0,0
E 1070 DATA 0,0,0,0,0,0,0,0
J 1080 DATA 0,0,0,32,158,183,224,16
B 1090 DATA 144,3,76,72,178,173,169,194
C 1100 DATA 41,240,133,2,138,5,2,162
K 1110 DATA 24,76,41,194,32,48,194,165
X 1120 DATA 20,32,41,194,232,165,21,76
S 1130 DATA 41,194,32,48,194,165,21,240
I 1140 DATA 3,76,72,178,165,20,240,249
I 1150 DATA 201,5,176,245,168,169,8,10
T 1160 DATA 136,208,252,133,2,232,232,232
C 1170 DATA 232,189,145,194,41,15,5,2
J 1180 DATA 76,41,194,32,101,194,232,232
U 1190 DATA 232,232,232,232,189,145,194,41
T 1200 DATA 15,5,2,76,41,194,32,101
A 1210 DATA 194,76,255,194,32,85,194,232
A 1220 DATA 232,232,232,232,232,189,145,19
       4
R 1230 DATA 41,240,5,2,76,41,194,32
A 1240 DATA 85,194,76,24,195,32,122,194
H 1250 DATA 240,8,189,145,194,9,1,76
R 1260 DATA 41,194,189,145,194,41,254,76
U 1270 DATA 41,194,32,48,194,201,3,208
Z 1280 DATA 2,169,4,166,21,240,3,76
J 1290 DATA 72,178,166,20,224,2,176,247
S 1300 DATA 224,0,240,5,13,168,194,208
U 1310 DATA 5,73,255,45,168,194,162,23
F 1320 DATA 76,41,194,169,16,208,6,169
Q 1330 DATA 32,208,2,169,64,133,2,32
O 1340 DATA 158,183,224,2,144,3,76,72
O 1350 DATA 178,165,2,224,0,240,5,13
Q 1360 DATA 169,194,208,5,73,255,45,169
S 1370 DATA 194,162,24,76,41,194,32,122
T 1380 DATA 194,240,8,189,145,194,9,4
Q 1390 DATA 76,41,194,189,145,194,41,251
N 1400 DATA 76,41,194,32,122,194,240,8
F 1410 DATA 189,145,194,9,2,76,41,194
A 1420 DATA 189,145,194,41,253,76,41,194
L 1430 DATA 32,158,183,224,16,144,3,76
S 1440 DATA 72,178,138,32,104,194,173,168
X 1450 DATA 194,41,15,5,2,162,23,76
S 1460 DATA 41,194,32,138,173,32,247,183
L 1470 DATA 201,8,144,3,76,72,178,165
P 1480 DATA 20,162,21,32,41,194,160,3
N 1490 DATA 70,21,102,20,136,208,249,232
P 1500 DATA 165,20,76,41,194,32,48,194
C 1510 DATA 165,21,201,16,176,222,232,232
K 1520 DATA 165,20,32,41,194,232,165,21
H 1530 DATA 76,41,194,169,26,160,196,76
D 1540 DATA 30,171,42,42,32,82,65,78
U 1550 DATA 68,89,32,84,72,79,77,80
J 1560 DATA 83,79,78,32,42,42,0,169
F 1570 DATA 0,162,24,32,41,194,202,16
Z 1580 DATA 250,96,173,169,194,41,15,168
K 1590 DATA 76,162,179,32,54,194,189,145
N 1600 DATA 194,133,99,189,146,194,133,98
R 1610 DATA 162,144,56,76,73,188,32,54
H 1620 DATA 194,160,0,189,149,194,41,240
M 1630 DATA 240,14,133,2,169,8,160,1
Q 1640 DATA 10,36,2,208,3,200,208,248
F 1650 DATA 76,162,179,32,54,194,76,111
Y 1660 DATA 194,32,54,194,232,76,111,194
P 1670 DATA 32,54,194,189,150,194,41,15
H 1680 DATA 168,76,162,179,32,54,194,189
I 1690 DATA 151,194,41,15,168,76,162,179
A 1700 DATA 32,54,194,189,149,194,41,1
K 1710 DATA 168,76,162,179,32,54,194,201
B 1720 DATA 3,208,2,169,4,45,168,194
I 1730 DATA 240,2,169,1,168,76,162,179
A 1740 DATA 169,16,208,6,169,32,208,2
O 1750 DATA 169,64,45,169,194,240,2,169
X 1760 DATA 1,168,76,162,179,32,54,194
A 1770 DATA 189,149,194,41,4,240,2,169
J 1780 DATA 1,168,76,162,179,32,54,194
C 1790 DATA 189,149,194,41,2,240,2,169
W 1800 DATA 1,168,76,162,179,173,168,194
M 1810 DATA 76,114,194,173,166,194,133,99
Q 1820 DATA 169,0,160,3,14,167,194,42
A 1830 DATA 136,208,249,133,98,162,144,56
P 1840 DATA 76,73,188,32,54,194,189,147
K 1850 DATA 194,133,99,189,148,194,133,98
O 1860 DATA 162,144,56,76,73,188
```

HCM