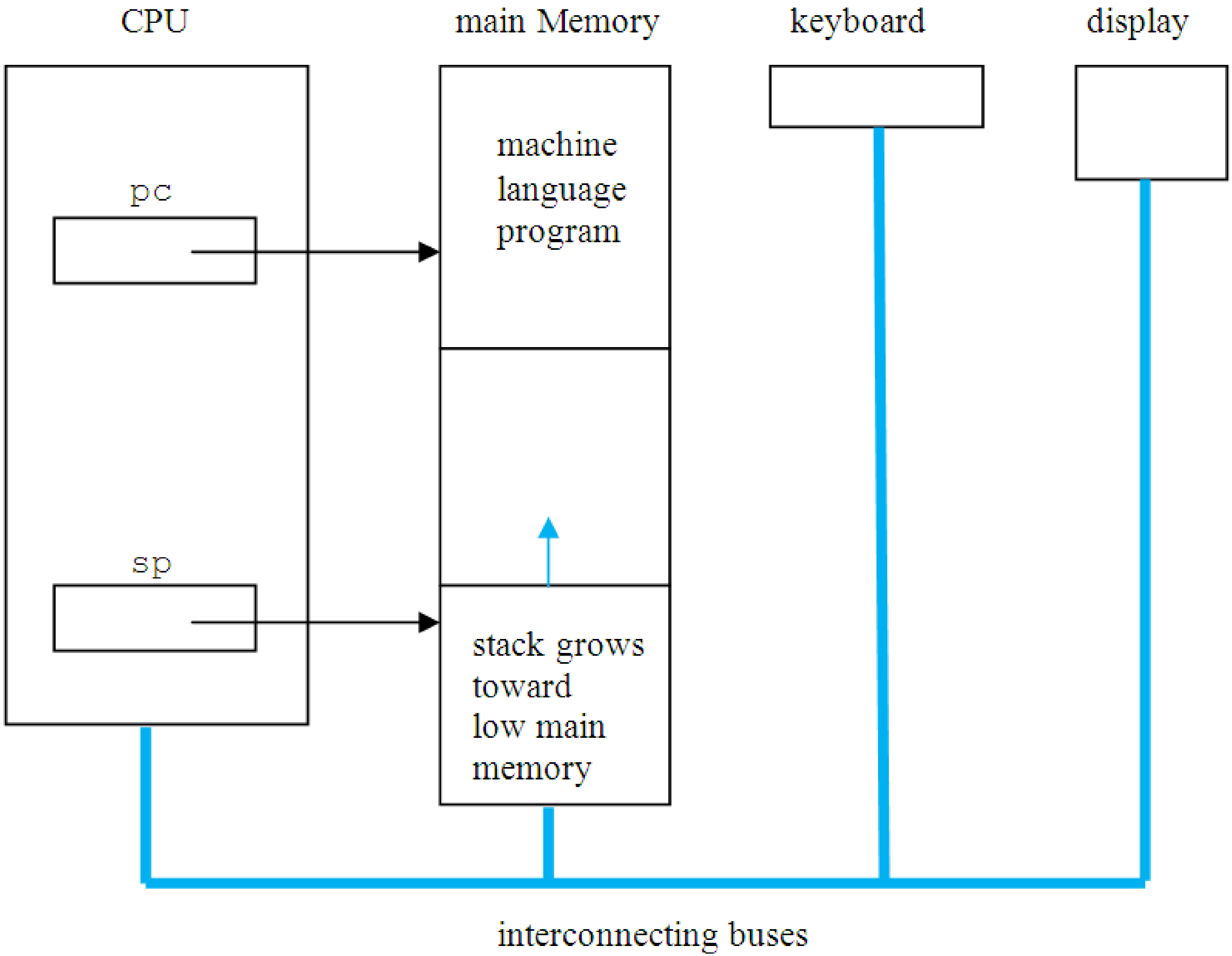
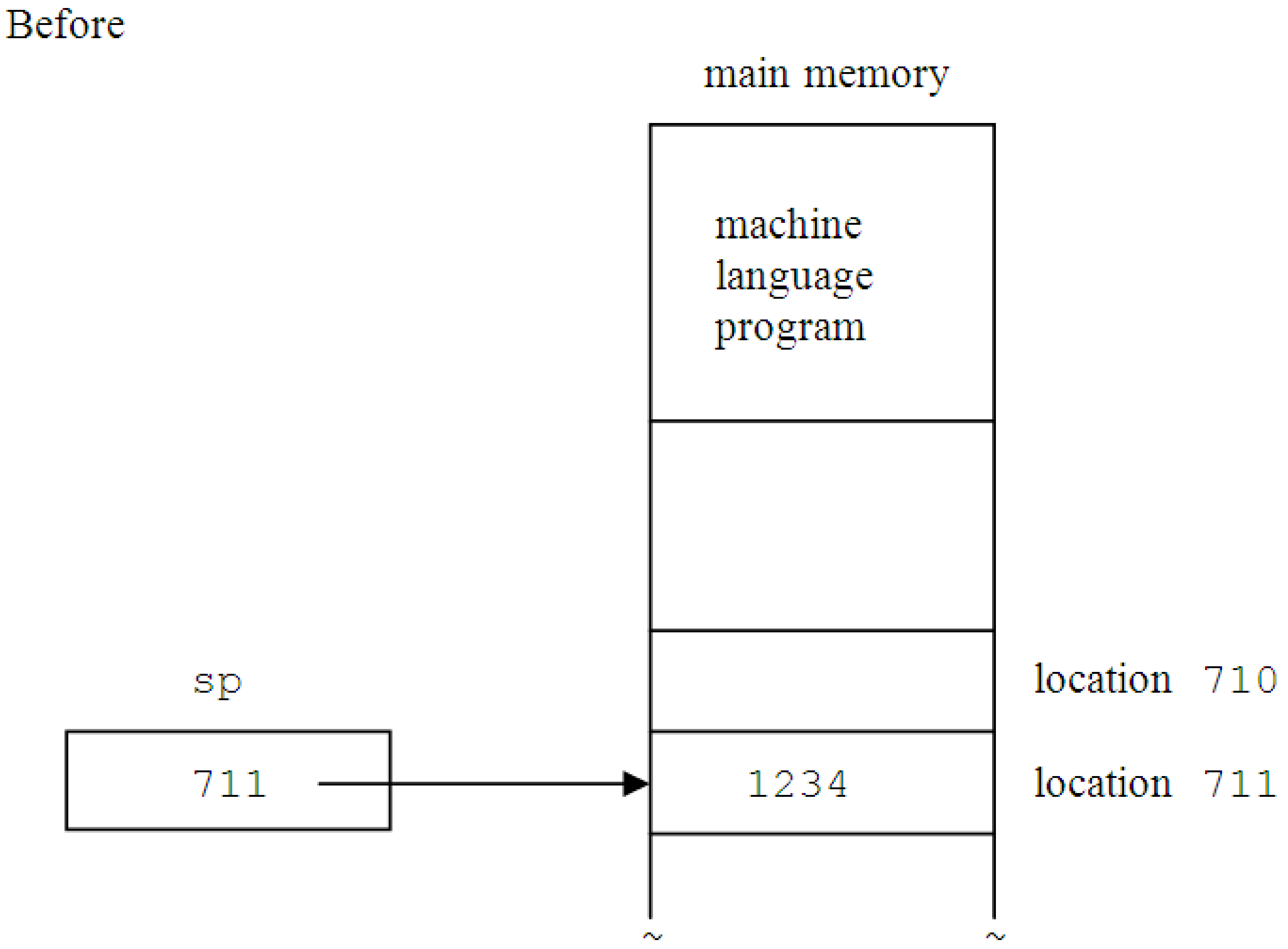
Chapter 11

Assembly Language

J1 Computer Architecture

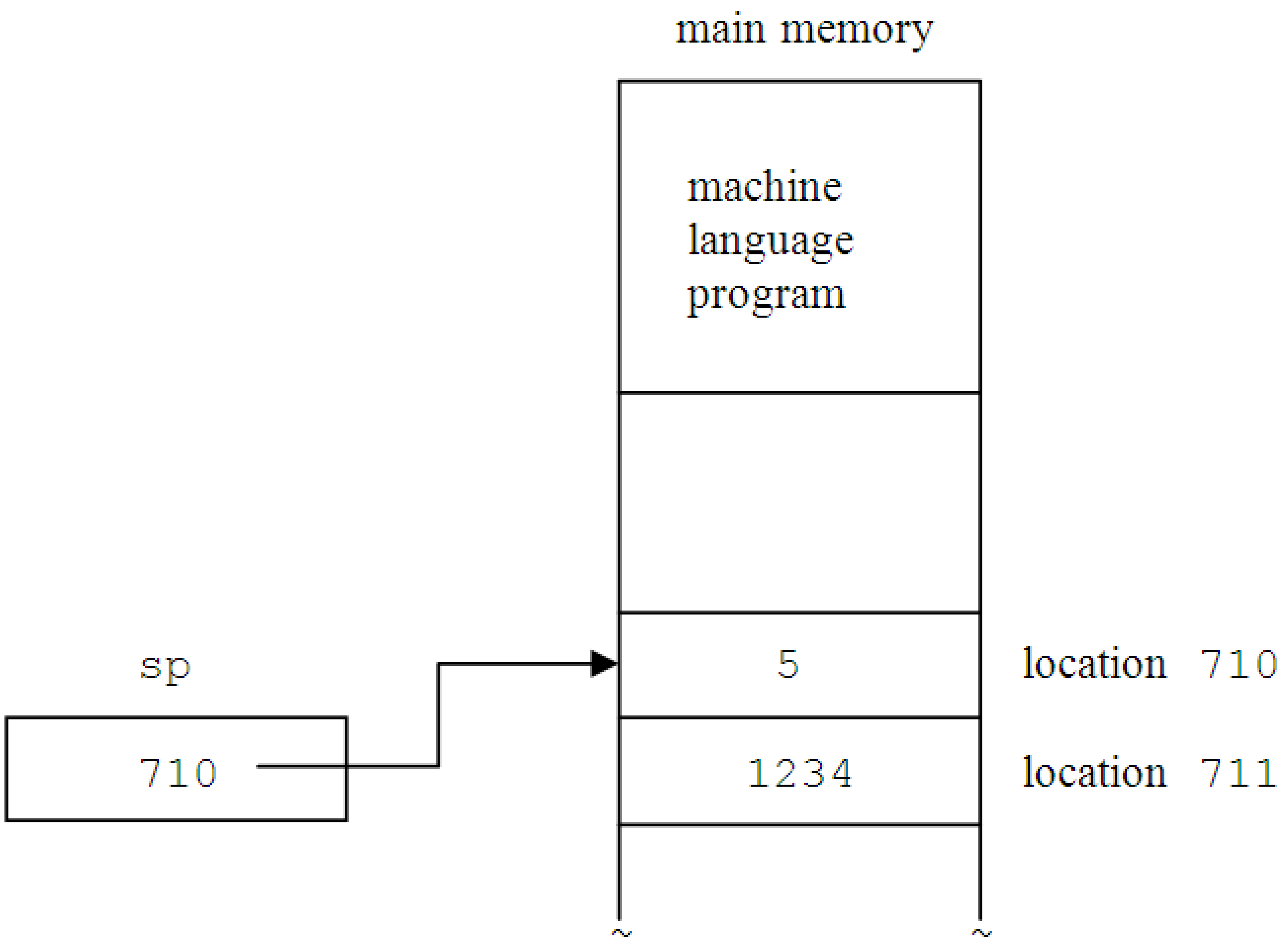


Push operation: before



Push operation: after

After



Machine language instructions

push value from location 4:

0000 000000000100

push constant 7:

0001 000000000111

push word constant 3

1111101110000000 0000000000000011

halt

1111111111111111

Assembly language instructions

```
p         4         ; push the word at address 4
pc        7         ; push 7
pwc       3         ; push 3
dout                ; pop and display
halt                ; terminate program
```

Define word

```
dw  15
```

Pushing characters

```
pc  'A'
pc  '\n'      or      pc          10
```

aout instruction

```
pc  'A'
aout

pc  '\n'
aout
```

Using labels

```
          p         x      ; this instruction is at address 0
          halt             ; this instruction is at address 1
x:        dw        12     ; this data item is at address 2
```

Complete program

```
          p         @1   ; push 1
          pc        2    ; push 2
          add            ; pop twice,add,push
          pwc       4    ; push 4
          add            ; pop twice,add,push
          dout           ; pop and display
          pc        '\n' ; push newline
          aout           ; pop and output
          halt           ; terminate
@1:       dw        1    ; data
```

Using the assembler

```
a fig1104.a
```

or

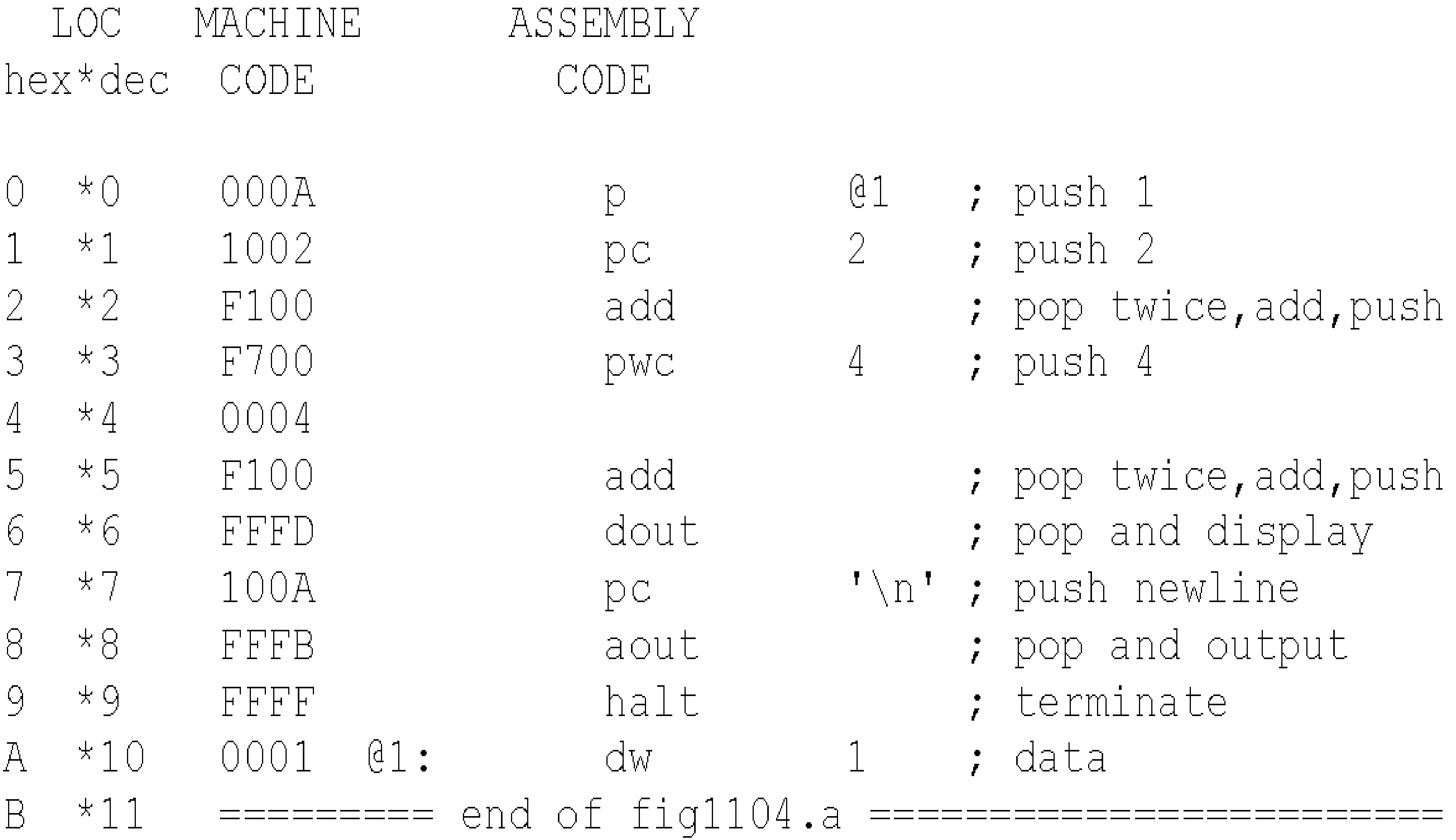
```
./a fig1104.a
```

```
e fig1104
```

or

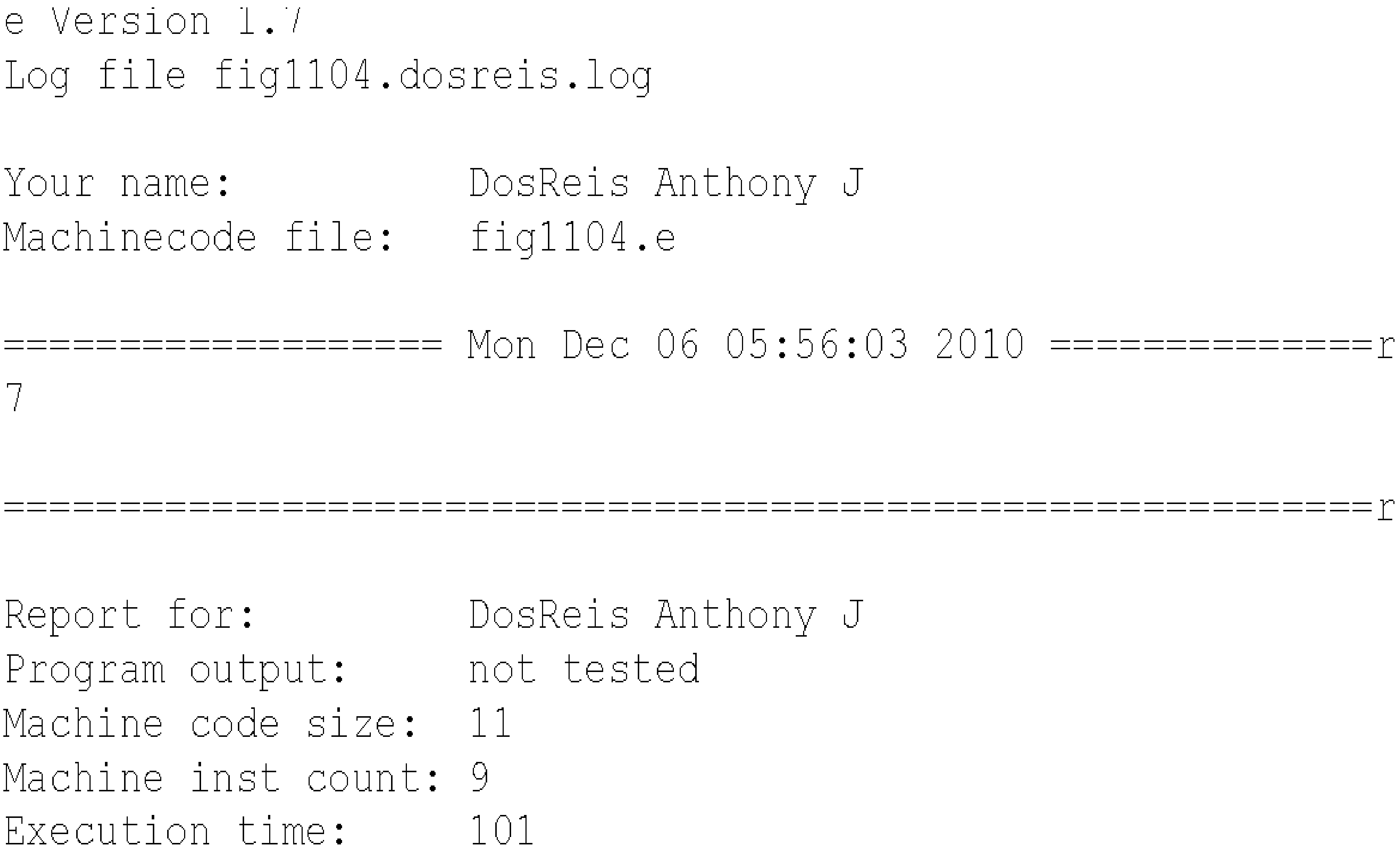
```
e fig1104 /c
```

Assembler listing



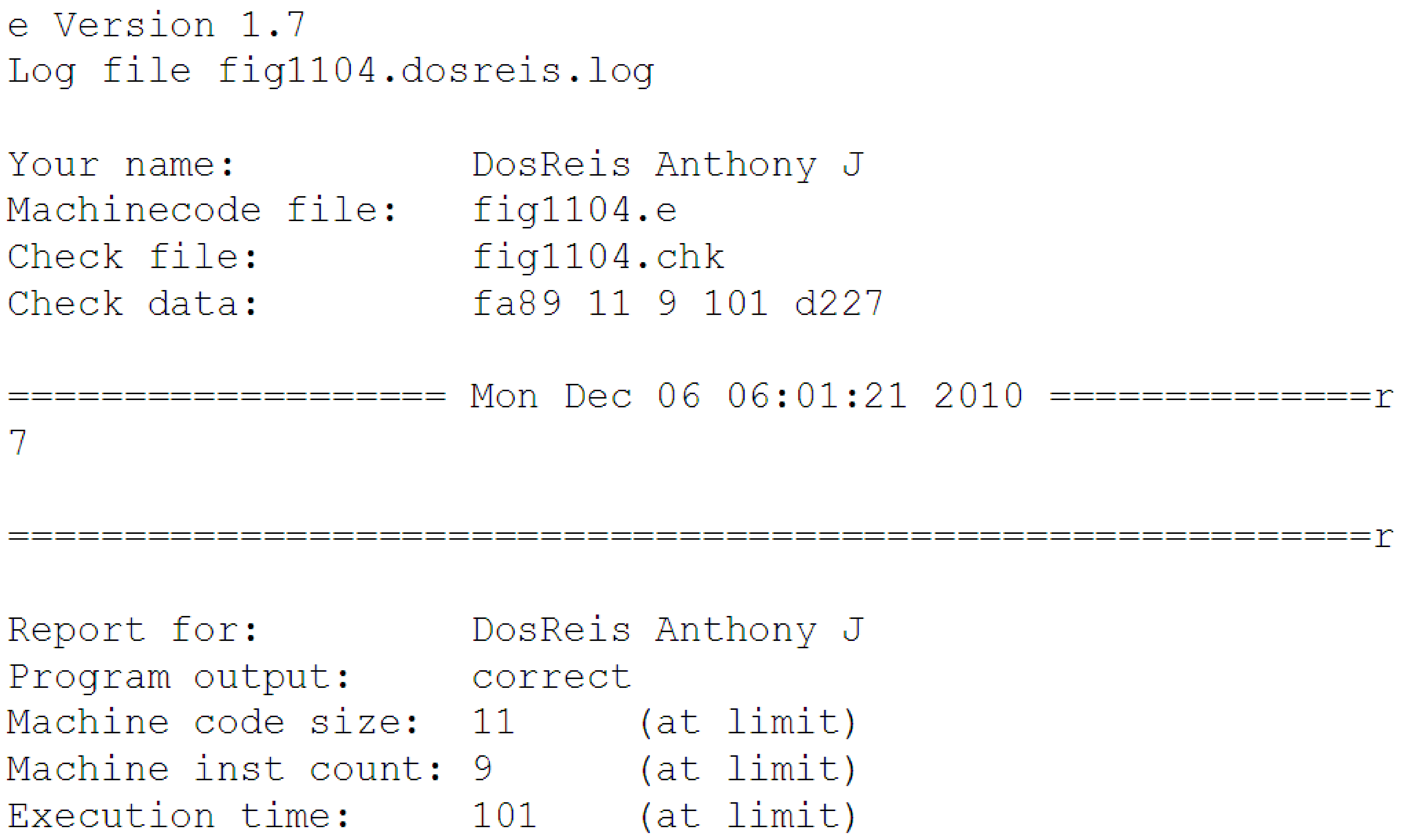
```
  LOC    MACHINE       ASSEMBLY
hex*dec   CODE           CODE

0   *0    000A              p            @1   ; push 1
1   *1    1002              pc           2    ; push 2
2   *2    F100              add               ; pop twice,add,push
3   *3    F700              pwc          4    ; push 4
4   *4    0004
5   *5    F100              add               ; pop twice,add,push
6   *6    FFFD              dout              ; pop and display
7   *7    100A              pc           '\n' ; push newline
8   *8    FFFB              aout              ; pop and output
9   *9    FFFF              halt              ; terminate
A   *10   0001  @1:         dw           1    ; data
B   *11   ========= end of fig1104.a ====================
```

log file



```
e Version 1.7
Log file fig1104.dosreis.log

Your name:          DosReis Anthony J
Machinecode file:   fig1104.e

=================== Mon Dec 06 05:56:03 2010 ==============r
7

===========================================================r

Report for:         DosReis Anthony J
Program output:     not tested
Machine code size:  11
Machine inst count: 9
Execution time:     101
```

log file with checking active

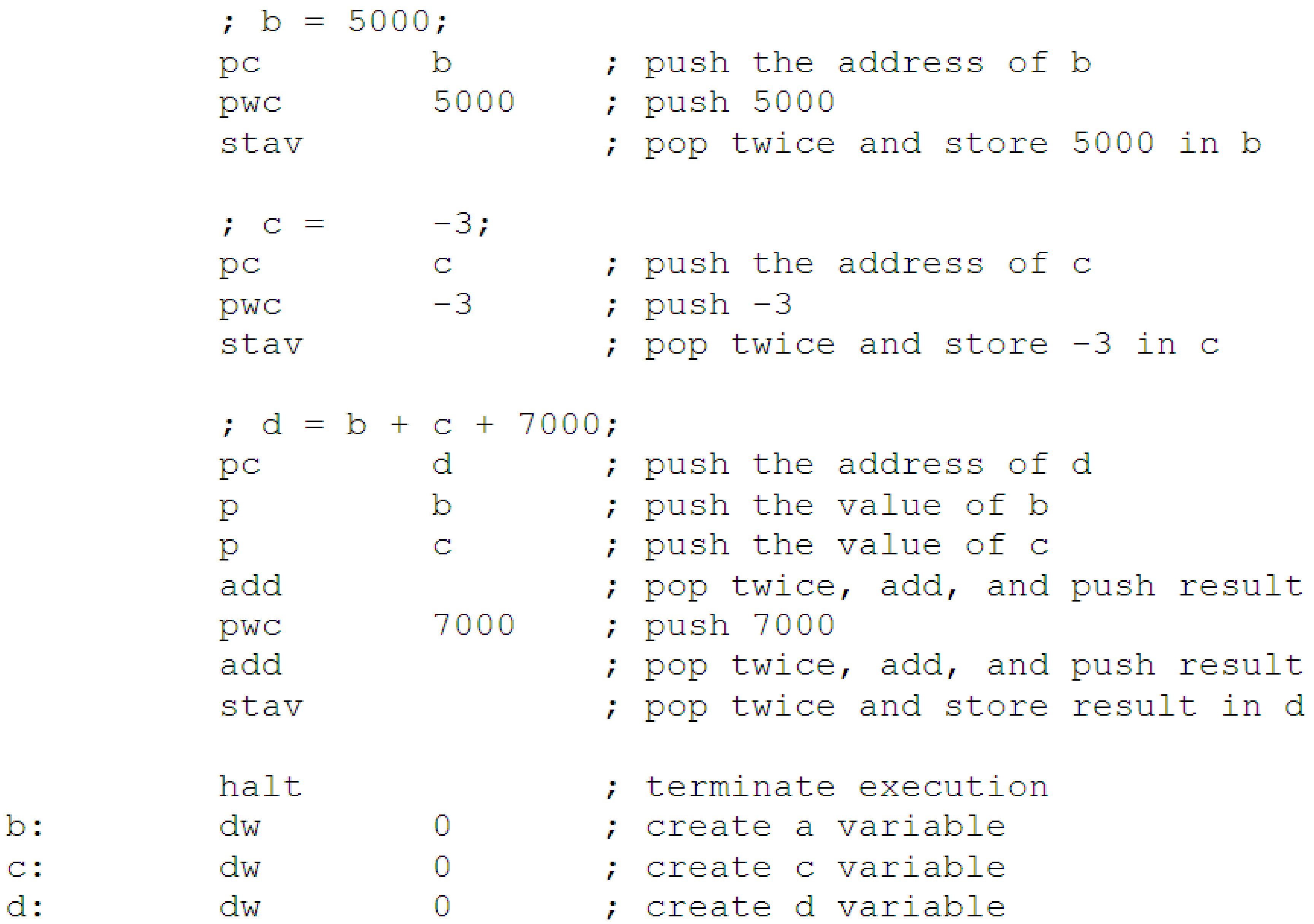


```
e Version 1.7
Log file fig1104.dosreis.log

Your name:          DosReis Anthony J
Machinecode file:   fig1104.e
Check file:         fig1104.chk
Check data:         fa89 11 9 101 d227

=================== Mon Dec 06 06:01:21 2010 ==============r
7

============================================================r

Report for:         DosReis Anthony J
Program output:     correct
Machine code size:  11      (at limit)
Machine inst count: 9       (at limit)
Execution time:     101     (at limit)
```

stav instruction

```
           pc        x        ; push address of x
           p         y        ; push value of y
           stav               ; pop and store
           halt
x:         dw        0
y:         dw        77
```

Compiling assignment statements



```
          ; b = 5000;
          pc        b       ; push the address of b
          pwc       5000    ; push 5000
          stav              ; pop twice and store 5000 in b

          ; c =     -3;
          pc        c       ; push the address of c
          pwc       -3      ; push -3
          stav              ; pop twice and store -3 in c

          ; d = b + c + 7000;
          pc        d       ; push the address of d
          p         b       ; push the value of b
          p         c       ; push the value of c
          add               ; pop twice, add, and push result
          pwc       7000    ; push 7000
          add               ; pop twice, add, and push result
          stav              ; pop twice and store result in d

          halt              ; terminate execution
b:        dw        0       ; create a variable
c:        dw        0       ; create c variable
d:        dw        0       ; create d variable
```

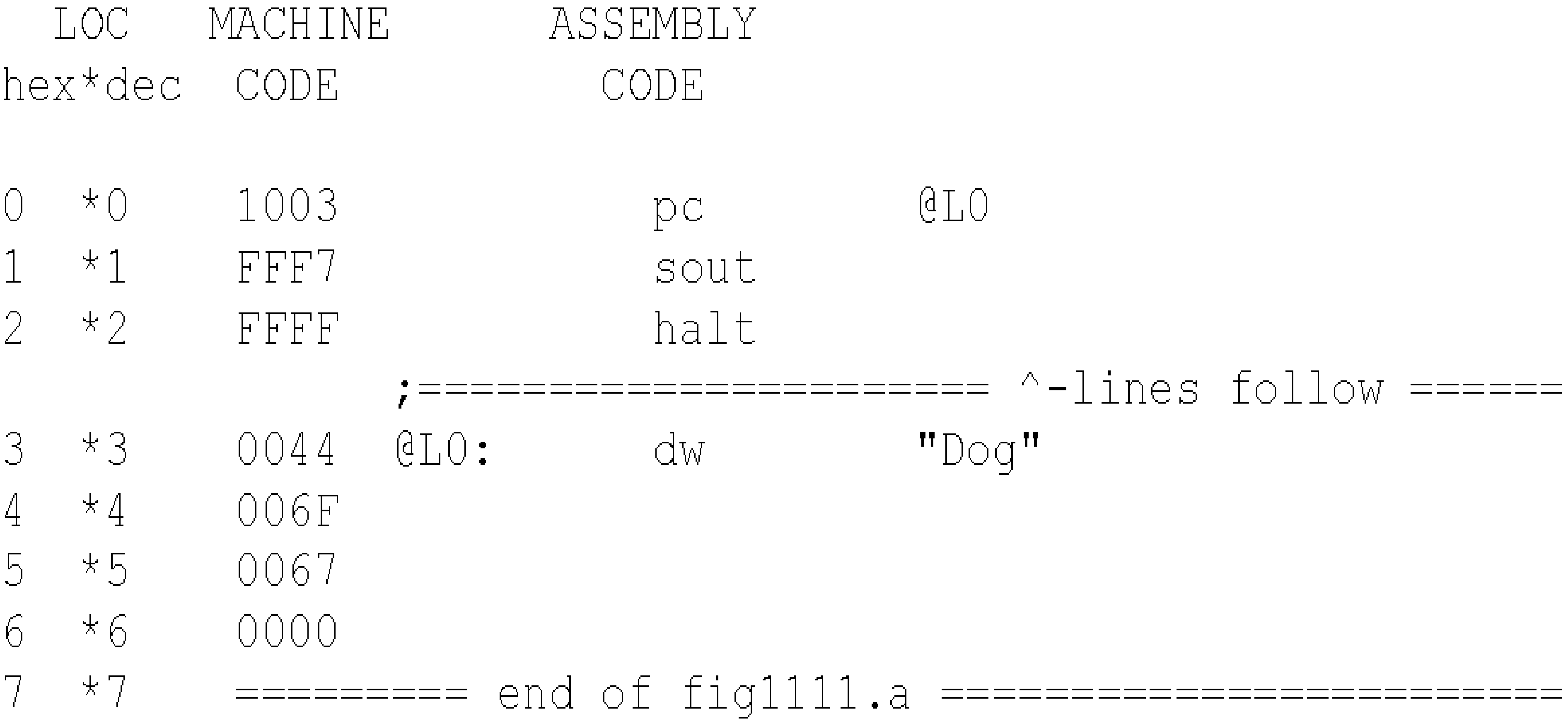
Compiling println()

```
println(q);

  p       q
  dout
  pc      '\n'
  aout
```

Outputting strings

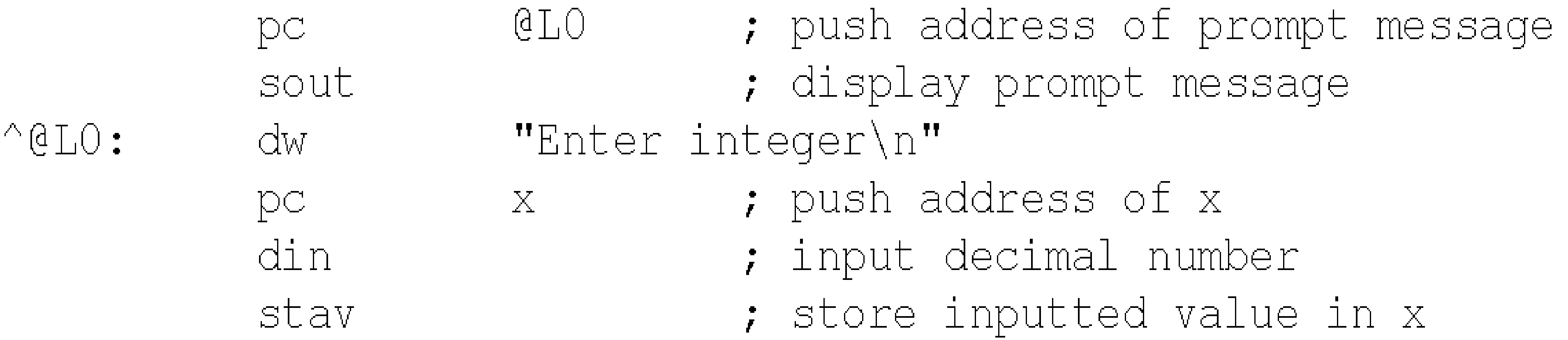
```
            pc         @L0
            sout
^@L0:       dw         "Dog"
            halt
```

Caret line moved to end



```
  LOC   MACHINE      ASSEMBLY
hex*dec  CODE          CODE

0  *0    1003             pc          @L0
1  *1    FFF7             sout
2  *2    FFFF             halt
                ;===================== ^-lines follow ======
3  *3    0044   @L0:      dw          "Dog"
4  *4    006F
5  *5    0067
6  *6    0000
7  *7    ========= end of fig1111.a =======================
```

Inputting a decimal number



```
          pc        @L0       ; push address of prompt message
          sout                ; display prompt message
^@L0:     dw        "Enter integer\n"
          pc        x         ; push address of x
          din                 ; input decimal number
          stav                ; store inputted value in x
```

entry directive

```
         entry     cat       ; specifies entry point
x:       dw        3
y:       dw        4
cat:     p         x
         p         y
         add
         dout
         halt
```