Lab #1

Assembly Language 1

Florida Polytechnic University

EEL4746C: Microcomputers

Fall 2018

Student Name: Peter A. Dranishnikov

Student ID: U0000005258

Lab Partner(s): N/A

Section: 01

Experiment Date: September 24th, 2018

Table of Contents

Introduction

The purpose of this lab was to trace through a program using the GNU Debugger for AVR.

Discussion

The AVR assembly code below was assembled, linked, and loaded into the GNU Debugger for AVR. The debugger was used to step through the machine code line-by-line and observe the change in the registers and memory. Briefly, the source code intended to set several constants, loaded those constants into several registers, and determine a conditional branch on equality (which intended to not branch since constants a and b were not equal). Note that during the lab, two other different programs were assembled, linked, loaded, and executed in the debugger for demonstration purposes during the lab session, so the source code will not be shown in this document.

```
####################################################################
# Filename: lab1-3.asm
# Version: 1.1 (lab manual version with documentation changes)
# Description: performs a comparison on 2 constants, then branches to load a value if equal
# Author: Original: Youssif Al-Nashif Derived: Peter A. Dranishnikov
# Target: Atmel AtMega328p AVR
# Assembler: avr-as
# Last modified: September 24th, 2018 (09/24/18)
####################################################################

.global start
.text

.set a, 10
.set b, 25
.set c, 15
.set d, -10
.set e, 246
start:
      ldi r16, a
      ldi r17, b
      ldi r18, 0
      cp r16, r17 ;compare content of r16 & r17
      breq if ;branch if equal
else:
      ldi r18, c
      rjmp endif
if:
      ldi r18, d
endif:
      mov r18, r18
loop:
      rjmp loop
.end
Anything can be written beyond this point
```

Experimental procedure

The source code above was assembled using avr-as targeting the atmega328p microprocessor. Then the resulting object file was linked to an executable with avr-ld. The microprocessor simulator was started, with the executable loaded through avr-gdb.

Results/Measurements/Observations

Appendix A contains the full terminal log for all steps. No memory values had changed since there were no instructions in the source code that wrote directly to memory. The following table shows the change of registers for each stepping:

Instruction	Changes in registers	Branch
(reset vector)	PC=0x0000, rest uninitialized	
`ldi r16, a`	r16=0x0A, PC+=0x2	
`ldi r17, b`	r17=0x19 PC+=0x2	
`ldi r18, 0`	r18=0x00 PC+=0x2	
`cp r16, r17`	SREG=0x15 (Flags: Carry, Negative, Sign) PC+=0x2	
`breq if`	PC+=0x2	Not taken (Zero flag clear)
`ldi r18, c`	r18=0x0F PC+=0x02	
`rjmp endif`	PC+=0x04	
`mov r18, r18`	PC+=0x02	
`rjmp loop`	PC=0x12	

Result Discussion

The expected behavior matched the actual behavior during stepping runtime.

Conclusion

It is possible to step through an assembled program and observe the changes in registers using the GNU debugger and the simulavr simulator.

Answers to lab's questions

Step 24 with code 2: Why there is an error?

The first .byte directive was not aligned to any ending byte value.

Step 26 with code 2 correction: What is the difference from the previous code?

The .byte directive was changed to end on the next byte. Additionally, the directives in the loop of the original source code were converted to instructions by the assembler.

Appendix A: Full terminal log

```
user@ubuntu:~$ cd atmega328p/asm/lab1/
user@ubuntu:~/atmega328p/asm/lab1$ ls
lab1-1.asm  lab1-1.o  lab1-1.x  lab1-2.asm  lab1-2.o  lab1-2.x  lab1-3.asm
user@ubuntu:~/atmega328p/asm/lab1$ avr-as -mmcu=atmega328p -ggdb -o lab1-3.o lab1-3.asm
user@ubuntu:~/atmega328p/asm/lab1$ avr-as -mmcu=atmega328p -ggdb -o lab1-3.o lab1-3.asm -a
GAS LISTING lab1-3.asm             page 1


   1                    ####################################################################
   2                    # Filename: lab1-3.asm
   3                    # Version: 1.1 (lab manual version with documentation changes)
   4                    # Description: performs a comparison on 2 constants, then branches
to load a value if equal
   5                    # Author: Original: Youssif Al-Nashif Derived: Peter A. Dranishnikov
   6                    # Target: Atmel AtMega328p AVR
   7                    # Assembler: avr-as
   8                    # Last modified: September 24th, 2018 (09/24/18)
   9                    ####################################################################
  10
  11                    .global start
  12                    .text
  13
  14                    .set a, 10
  15                    .set b, 25
  16                    .set c, 15
  17                    .set d, -10
  18                    .set e, 246
  19                    start:
  20 0000 0AE0                ldi r16, a
  21 0002 19E1                ldi r17, b
  22 0004 20E0                ldi r18, 0
  23 0006 0117                cp r16, r17 ;compare content of r16 & r17
  24 0008 01F0                breq if ;branch if equal
  25                    else:
  26 000a 2FE0                ldi r18, c
  27 000c 00C0                rjmp endif
  28                    if:
  29 000e 26EF                ldi r18, d
  30                    endif:
  31 0010 222F                mov r18, r18
  32                    loop:
  33 0012 00C0                rjmp loop
  34                    .end

GAS LISTING lab1-3.asm             page 2


DEFINED SYMBOLS
          lab1-3.asm:19     .text:0000000000000000 start
          lab1-3.asm:14     *ABS*:000000000000000a a
          lab1-3.asm:15     *ABS*:0000000000000019 b
          lab1-3.asm:16     *ABS*:000000000000000f c
          lab1-3.asm:17     *ABS*:fffffffffffffff6 d
          lab1-3.asm:18     *ABS*:00000000000000f6 e
          lab1-3.asm:28     .text:000000000000000e if
          lab1-3.asm:25     .text:000000000000000a else
          lab1-3.asm:30     .text:0000000000000010 endif
          lab1-3.asm:32     .text:0000000000000012 loop

NO UNDEFINED SYMBOLS
user@ubuntu:~/atmega328p/asm/lab1$ avr-ld -o lab1-3.x lab1-3.o
user@ubuntu:~/atmega328p/asm/lab1$ avr-gdb
GNU gdb (GDB) 7.10.1
Copyright (C) 2015 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "--host=x86_64-linux-gnu --target=avr".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word".
(gdb) target remote localhost:1212
Remote debugging using localhost:1212
0x00000000 in ?? ()
(gdb) file lab1-3.x
A program is being debugged already.
Are you sure you want to change the file? (y or n) y
Reading symbols from lab1-3.x...done.
(gdb) load
Loading section .text, size 0x14 lma 0x0
Start address 0x0, load size 20
Transfer rate: 160 bits in <1 sec, 20 bytes/write.
(gdb) disassemble /r 0x000000, +1
Dump of assembler code from 0x0 to 0x800001:
=> 0x00000000 <start+0>:     0a e0 ldi   r16, 0x0A   ; 10
   0x00000002 <start+2>:     19 e1 ldi   r17, 0x19   ; 25
   0x00000004 <start+4>:     20 e0 ldi   r18, 0x00   ; 0
   0x00000006 <start+6>:     01 17 cp    r16, r17
   0x00000008 <start+8>:     11 f0 breq  .+4         ;  0xe <if>
   0x0000000a <else+0>: 2f e0 ldi   r18, 0x0F   ; 15
   0x0000000c <else+2>: 01 c0 rjmp  .+2         ;  0x10 <endif>
   0x0000000e <if+0>:   26 ef ldi   r18, 0xF6   ; 246
   0x00000010 <endif+0>:     22 2f mov   r18, r18
   0x00000012 <loop+0>: ff cf rjmp  .-2         ;  0x12 <loop>
   0x00000014:    ff ff .word 0xffff      ; ????
   0x00000016:    ff ff .word 0xffff      ; ????
   0x00000018:    ff ff .word 0xffff      ; ????
   0x0000001a:    ff ff .word 0xffff      ; ????
   0x0000001c:    ff ff .word 0xffff      ; ????
   0x0000001e:    ff ff .word 0xffff      ; ????
   0x00000020:    ff ff .word 0xffff      ; ????
   0x00000022:    ff ff .word 0xffff      ; ????
   0x00000024:    ff ff .word 0xffff      ; ????
   0x00000026:    ff ff .word 0xffff      ; ????
   0x00000028:    ff ff .word 0xffff      ; ????
   0x0000002a:    ff ff .word 0xffff      ; ????
   0x0000002c:    ff ff .word 0xffff      ; ????
   0x0000002e:    ff ff .word 0xffff      ; ????
   0x00000030:    ff ff .word 0xffff      ; ????
   0x00000032:    ff ff .word 0xffff      ; ????
   0x00000034:    ff ff .word 0xffff      ; ????
   0x00000036:    ff ff .word 0xffff      ; ????
   0x00000038:    ff ff .word 0xffff      ; ????
   0x0000003a:    ff ff .word 0xffff      ; ????
   0x0000003c:    ff ff .word 0xffff      ; ????
   0x0000003e:    ff ff .word 0xffff      ; ????
---Type <return> to continue, or q <return> to quit---q
Quit
(gdb) disassemble /m 0x000000, +1
Dump of assembler code from 0x0 to 0x800001:
20          ldi r16, a
=> 0x00000000 <start+0>:     ldi   r16, 0x0A   ; 10

21          ldi r17, b
   0x00000002 <start+2>:     ldi   r17, 0x19   ; 25

22          ldi r18, 0
   0x00000004 <start+4>:     ldi   r18, 0x00   ; 0

23          cp r16, r17 ;compare content of r16 & r17
   0x00000006 <start+6>:     cp    r16, r17

24          breq if ;branch if equal
   0x00000008 <start+8>:     breq  .+4         ;  0xe <if>

25    else:
26          ldi r18, c
   0x0000000a <else+0>: ldi   r18, 0x0F   ; 15

27          rjmp endif
   0x0000000c <else+2>: rjmp  .+2         ;  0x10 <endif>

28    if:
29          ldi r18, d
   0x0000000e <if+0>:   ldi   r18, 0xF6   ; 246

30    endif:
31          mov r18, r18
   0x00000010 <endif+0>:     mov   r18, r18

32    loop:
33          rjmp loop
   0x00000012 <loop+0>: rjmp  .-2         ;  0x12 <loop>
---Type <return> to continue, or q <return> to quit---q
Quit
(gdb) x /24b 0x0100
0x800100:   -86   -86   -86   -86   -86   -86   -86   -86
0x800108:   -86   -86   -86   -86   -86   -86   -86   -86
0x800110:   -86   -86   -86   -86   -86   -86   -86   -86
(gdb) x /17xb 0x0100
```

```
0x800100:   0xaa  0xaa  0xaa  0xaa  0xaa  0xaa  0xaa  0xaa
0x800108:   0xaa  0xaa  0xaa  0xaa  0xaa  0xaa  0xaa  0xaa
0x800110:   0xaa
(gdb) x /32x 0x0100
0x800100:   0xaa  0xaa  0xaa  0xaa  0xaa  0xaa  0xaa  0xaa
0x800108:   0xaa  0xaa  0xaa  0xaa  0xaa  0xaa  0xaa  0xaa
0x800110:   0xaa  0xaa  0xaa  0xaa  0xaa  0xaa  0xaa  0xaa
0x800118:   0xaa  0xaa  0xaa  0xaa  0xaa  0xaa  0xaa  0xaa
(gdb) x /32xw 0x0100
0x800100:   0xaaaaaaaa  0xaaaaaaaa  0xaaaaaaaa  0xaaaaaaaa
0x800110:   0xaaaaaaaa  0xaaaaaaaa  0xaaaaaaaa  0xaaaaaaaa
0x800120:   0xaaaaaaaa  0xaaaaaaaa  0xaaaaaaaa  0xaaaaaaaa
0x800130:   0xaaaaaaaa  0xaaaaaaaa  0xaaaaaaaa  0xaaaaaaaa
0x800140:   0xaaaaaaaa  0xaaaaaaaa  0xaaaaaaaa  0xaaaaaaaa
0x800150:   0xaaaaaaaa  0xaaaaaaaa  0xaaaaaaaa  0xaaaaaaaa
0x800160:   0xaaaaaaaa  0xaaaaaaaa  0xaaaaaaaa  0xaaaaaaaa
0x800170:   0xaaaaaaaa  0xaaaaaaaa  0xaaaaaaaa  0xaaaaaaaa
(gdb) x /32xh 0x0100
0x800100:   0xaaaa      0xaaaa      0xaaaa      0xaaaa      0xaaaa      0xaaaa      0xaaaa
      0xaaaa
0x800110:   0xaaaa      0xaaaa      0xaaaa      0xaaaa      0xaaaa      0xaaaa      0xaaaa
      0xaaaa
0x800120:   0xaaaa      0xaaaa      0xaaaa      0xaaaa      0xaaaa      0xaaaa      0xaaaa
      0xaaaa
0x800130:   0xaaaa      0xaaaa      0xaaaa      0xaaaa      0xaaaa      0xaaaa      0xaaaa
      0xaaaa
(gdb) i r
r0             0xaa     170
r1             0xaa     170
r2             0xaa     170
r3             0xaa     170
r4             0xaa     170
r5             0xaa     170
r6             0xaa     170
r7             0xaa     170
r8             0xaa     170
r9             0xaa     170
r10            0xaa     170
r11            0xaa     170
r12            0xaa     170
r13            0xaa     170
r14            0xaa     170
r15            0xaa     170
r16            0xaa     170
r17            0xaa     170
r18            0xaa     170
r19            0xaa     170
r20            0xaa     170
r21            0xaa     170
r22            0xaa     170
r23            0xaa     170
r24            0xaa     170
r25            0xaa     170
r26            0xaa     170
r27            0xaa     170
r28            0xaa     170
```

```
r29            0xaa     170
r30            0xaa     170
r31            0xaa     170
SREG           0x0      0
SP             0x0      0x0 <start>
---Type <return> to continue, or q <return> to quit---
PC2            0x0      0
pc             0x0      0x0 <start>
(gdb) si
21          ldi r17, b
(gdb) i r
r0             0xaa     170
r1             0xaa     170
r2             0xaa     170
r3             0xaa     170
r4             0xaa     170
r5             0xaa     170
r6             0xaa     170
r7             0xaa     170
r8             0xaa     170
r9             0xaa     170
r10            0xaa     170
r11            0xaa     170
r12            0xaa     170
r13            0xaa     170
r14            0xaa     170
r15            0xaa     170
r16            0xa      10
r17            0xaa     170
r18            0xaa     170
r19            0xaa     170
r20            0xaa     170
r21            0xaa     170
r22            0xaa     170
r23            0xaa     170
r24            0xaa     170
r25            0xaa     170
r26            0xaa     170
r27            0xaa     170
r28            0xaa     170
r29            0xaa     170
r30            0xaa     170
r31            0xaa     170
SREG           0x0      0
SP             0x0      0x0 <start>
---Type <return> to continue, or q <return> to quit---
PC2            0x2      2
pc             0x2      0x2 <start+2>
(gdb) si
22          ldi r18, 0
(gdb) i r
r0             0xaa     170
r1             0xaa     170
r2             0xaa     170
r3             0xaa     170
r4             0xaa     170
```

```
r5             0xaa     170
r6             0xaa     170
r7             0xaa     170
r8             0xaa     170
r9             0xaa     170
r10            0xaa     170
r11            0xaa     170
r12            0xaa     170
r13            0xaa     170
r14            0xaa     170
r15            0xaa     170
r16            0xa      10
r17            0x19     25
r18            0xaa     170
r19            0xaa     170
r20            0xaa     170
r21            0xaa     170
r22            0xaa     170
r23            0xaa     170
r24            0xaa     170
r25            0xaa     170
r26            0xaa     170
r27            0xaa     170
r28            0xaa     170
r29            0xaa     170
r30            0xaa     170
r31            0xaa     170
SREG           0x0      0
SP             0x0      0x0 <start>
---Type <return> to continue, or q <return> to quit---
PC2            0x4      4
pc             0x4      0x4 <start+4>
(gdb) si
23          cp r16, r17 ;compare content of r16 & r17
(gdb) i r
r0             0xaa     170
r1             0xaa     170
r2             0xaa     170
r3             0xaa     170
r4             0xaa     170
r5             0xaa     170
r6             0xaa     170
r7             0xaa     170
r8             0xaa     170
r9             0xaa     170
r10            0xaa     170
r11            0xaa     170
r12            0xaa     170
r13            0xaa     170
r14            0xaa     170
r15            0xaa     170
r16            0xa      10
r17            0x19     25
r18            0x0      0
r19            0xaa     170
r20            0xaa     170
```

```
r21            0xaa     170
r22            0xaa     170
r23            0xaa     170
r24            0xaa     170
r25            0xaa     170
r26            0xaa     170
r27            0xaa     170
r28            0xaa     170
r29            0xaa     170
r30            0xaa     170
r31            0xaa     170
SREG           0x0      0
SP             0x0      0x0 <start>
---Type <return> to continue, or q <return> to quit---
PC2            0x6      6
pc             0x6      0x6 <start+6>
(gdb) si
24          breq if ;branch if equal
(gdb) i r
r0             0xaa     170
r1             0xaa     170
r2             0xaa     170
r3             0xaa     170
r4             0xaa     170
r5             0xaa     170
r6             0xaa     170
r7             0xaa     170
r8             0xaa     170
r9             0xaa     170
r10            0xaa     170
r11            0xaa     170
r12            0xaa     170
r13            0xaa     170
r14            0xaa     170
r15            0xaa     170
r16            0xa      10
r17            0x19     25
r18            0x0      0
r19            0xaa     170
r20            0xaa     170
r21            0xaa     170
r22            0xaa     170
r23            0xaa     170
r24            0xaa     170
r25            0xaa     170
r26            0xaa     170
r27            0xaa     170
r28            0xaa     170
r29            0xaa     170
r30            0xaa     170
r31            0xaa     170
SREG           0x15     21
SP             0x0      0x0 <start>
---Type <return> to continue, or q <return> to quit---
PC2            0x8      8
pc             0x8      0x8 <start+8>
```

```
(gdb) si
else () at lab1-3.asm:26
26          ldi r18, c
(gdb) i r
r0             0xaa     170
r1             0xaa     170
r2             0xaa     170
r3             0xaa     170
r4             0xaa     170
r5             0xaa     170
r6             0xaa     170
r7             0xaa     170
r8             0xaa     170
r9             0xaa     170
r10            0xaa     170
r11            0xaa     170
r12            0xaa     170
r13            0xaa     170
r14            0xaa     170
r15            0xaa     170
r16            0xa      10
r17            0x19     25
r18            0x0      0
r19            0xaa     170
r20            0xaa     170
r21            0xaa     170
r22            0xaa     170
r23            0xaa     170
r24            0xaa     170
r25            0xaa     170
r26            0xaa     170
r27            0xaa     170
r28            0xaa     170
r29            0xaa     170
r30            0xaa     170
r31            0xaa     170
SREG           0x15     21
SP             0x0      0x0 <start>
---Type <return> to continue, or q <return> to quit---
PC2            0xa      10
pc             0xa      0xa <else>
(gdb) si
27          rjmp endif
(gdb) i r
r0             0xaa     170
r1             0xaa     170
r2             0xaa     170
r3             0xaa     170
r4             0xaa     170
r5             0xaa     170
r6             0xaa     170
r7             0xaa     170
r8             0xaa     170
r9             0xaa     170
r10            0xaa     170
r11            0xaa     170
```

```
r12            0xaa     170
r13            0xaa     170
r14            0xaa     170
r15            0xaa     170
r16            0xa      10
r17            0x19     25
r18            0xf      15
r19            0xaa     170
r20            0xaa     170
r21            0xaa     170
r22            0xaa     170
r23            0xaa     170
r24            0xaa     170
r25            0xaa     170
r26            0xaa     170
r27            0xaa     170
r28            0xaa     170
r29            0xaa     170
r30            0xaa     170
r31            0xaa     170
SREG           0x15     21
SP             0x0      0x0 <start>
---Type <return> to continue, or q <return> to quit---
PC2            0xc      12
pc             0xc      0xc <else+2>
(gdb) si
endif () at lab1-3.asm:31
31          mov r18, r18
(gdb) i r
r0             0xaa     170
r1             0xaa     170
r2             0xaa     170
r3             0xaa     170
r4             0xaa     170
r5             0xaa     170
r6             0xaa     170
r7             0xaa     170
r8             0xaa     170
r9             0xaa     170
r10            0xaa     170
r11            0xaa     170
r12            0xaa     170
r13            0xaa     170
r14            0xaa     170
r15            0xaa     170
r16            0xa      10
r17            0x19     25
r18            0xf      15
r19            0xaa     170
r20            0xaa     170
r21            0xaa     170
r22            0xaa     170
r23            0xaa     170
r24            0xaa     170
r25            0xaa     170
r26            0xaa     170
```

```
r27            0xaa     170
r28            0xaa     170
r29            0xaa     170
r30            0xaa     170
r31            0xaa     170
SREG           0x15     21
SP             0x0      0x0 <start>
---Type <return> to continue, or q <return> to quit---
PC2            0x10     16
pc             0x10     0x10 <endif>
(gdb) x /32xh 0x0100
0x800100:   0xaaaa      0xaaaa      0xaaaa      0xaaaa      0xaaaa      0xaaaa      0xaaaa
      0xaaaa
0x800110:   0xaaaa      0xaaaa      0xaaaa      0xaaaa      0xaaaa      0xaaaa      0xaaaa
      0xaaaa
0x800120:   0xaaaa      0xaaaa      0xaaaa      0xaaaa      0xaaaa      0xaaaa      0xaaaa
      0xaaaa
0x800130:   0xaaaa      0xaaaa      0xaaaa      0xaaaa      0xaaaa      0xaaaa      0xaaaa
      0xaaaa
(gdb) si
loop () at lab1-3.asm:33
33          rjmp loop
(gdb) x /32xh 0x0100
0x800100:   0xaaaa      0xaaaa      0xaaaa      0xaaaa      0xaaaa      0xaaaa      0xaaaa
      0xaaaa
0x800110:   0xaaaa      0xaaaa      0xaaaa      0xaaaa      0xaaaa      0xaaaa      0xaaaa
      0xaaaa
0x800120:   0xaaaa      0xaaaa      0xaaaa      0xaaaa      0xaaaa      0xaaaa      0xaaaa
      0xaaaa
0x800130:   0xaaaa      0xaaaa      0xaaaa      0xaaaa      0xaaaa      0xaaaa      0xaaaa
      0xaaaa
(gdb) i r
r0             0xaa     170
r1             0xaa     170
r2             0xaa     170
r3             0xaa     170
r4             0xaa     170
r5             0xaa     170
r6             0xaa     170
r7             0xaa     170
r8             0xaa     170
r9             0xaa     170
r10            0xaa     170
r11            0xaa     170
r12            0xaa     170
r13            0xaa     170
r14            0xaa     170
r15            0xaa     170
r16            0xa      10
r17            0x19     25
r18            0xf      15
r19            0xaa     170
r20            0xaa     170
r21            0xaa     170
r22            0xaa     170
r23            0xaa     170
```

```
r24            0xaa     170
r25            0xaa     170
r26            0xaa     170
r27            0xaa     170
r28            0xaa     170
r29            0xaa     170
r30            0xaa     170
r31            0xaa     170
SREG           0x15     21
SP             0x0      0x0 <start>
---Type <return> to continue, or q <return> to quit---
PC2            0x12     18
pc             0x12     0x12 <loop>
(gdb) si
33          rjmp loop
(gdb) i r
r0             0xaa     170
r1             0xaa     170
r2             0xaa     170
r3             0xaa     170
r4             0xaa     170
r5             0xaa     170
r6             0xaa     170
r7             0xaa     170
r8             0xaa     170
r9             0xaa     170
r10            0xaa     170
r11            0xaa     170
r12            0xaa     170
r13            0xaa     170
r14            0xaa     170
r15            0xaa     170
r16            0xa      10
r17            0x19     25
r18            0xf      15
r19            0xaa     170
r20            0xaa     170
r21            0xaa     170
r22            0xaa     170
r23            0xaa     170
r24            0xaa     170
r25            0xaa     170
r26            0xaa     170
r27            0xaa     170
r28            0xaa     170
r29            0xaa     170
r30            0xaa     170
r31            0xaa     170
SREG           0x15     21
SP             0x0      0x0 <start>
---Type <return> to continue, or q <return> to quit---
PC2            0x12     18
pc             0x12     0x12 <loop>
(gdb) si
33          rjmp loop
(gdb) i r
```

```
r0             0xaa     170
r1             0xaa     170
r2             0xaa     170
r3             0xaa     170
r4             0xaa     170
r5             0xaa     170
r6             0xaa     170
r7             0xaa     170
r8             0xaa     170
r9             0xaa     170
r10            0xaa     170
r11            0xaa     170
r12            0xaa     170
r13            0xaa     170
r14            0xaa     170
r15            0xaa     170
r16            0xa      10
r17            0x19     25
r18            0xf      15
r19            0xaa     170
r20            0xaa     170
r21            0xaa     170
r22            0xaa     170
r23            0xaa     170
r24            0xaa     170
r25            0xaa     170
r26            0xaa     170
r27            0xaa     170
r28            0xaa     170
r29            0xaa     170
r30            0xaa     170
r31            0xaa     170
SREG           0x15     21
SP             0x0      0x0 <start>
---Type <return> to continue, or q <return> to quit---
PC2            0x12     18
pc             0x12     0x12 <loop>
(gdb) continue
Continuing.
^C
Program received signal SIGINT, Interrupt.
loop () at lab1-3.asm:33
33          rjmp loop
(gdb) i r
r0             0xaa     170
r1             0xaa     170
r2             0xaa     170
r3             0xaa     170
r4             0xaa     170
r5             0xaa     170
r6             0xaa     170
r7             0xaa     170
r8             0xaa     170
r9             0xaa     170
r10            0xaa     170
r11            0xaa     170
```

```
r12            0xaa     170
r13            0xaa     170
r14            0xaa     170
r15            0xaa     170
r16            0xa      10
r17            0x19     25
r18            0xf      15
r19            0xaa     170
r20            0xaa     170
r21            0xaa     170
r22            0xaa     170
r23            0xaa     170
r24            0xaa     170
r25            0xaa     170
r26            0xaa     170
r27            0xaa     170
r28            0xaa     170
r29            0xaa     170
r30            0xaa     170
r31            0xaa     170
SREG           0x15     21
SP             0x0      0x0 <start>
---Type <return> to continue, or q <return> to quit---
PC2            0x12     18
pc             0x12     0x12 <loop>
(gdb) x /32xh 0x0100
0x800100:   0xaaaa      0xaaaa      0xaaaa      0xaaaa      0xaaaa      0xaaaa      0xaaaa
      0xaaaa
0x800110:   0xaaaa      0xaaaa      0xaaaa      0xaaaa      0xaaaa      0xaaaa      0xaaaa
      0xaaaa
0x800120:   0xaaaa      0xaaaa      0xaaaa      0xaaaa      0xaaaa      0xaaaa      0xaaaa
      0xaaaa
0x800130:   0xaaaa      0xaaaa      0xaaaa      0xaaaa      0xaaaa      0xaaaa      0xaaaa
      0xaaaa
(gdb)
```