



ECE375 Lab 3

TA: Youngbin Jin

School of Electrical Engineering and Computer Science
Oregon State University



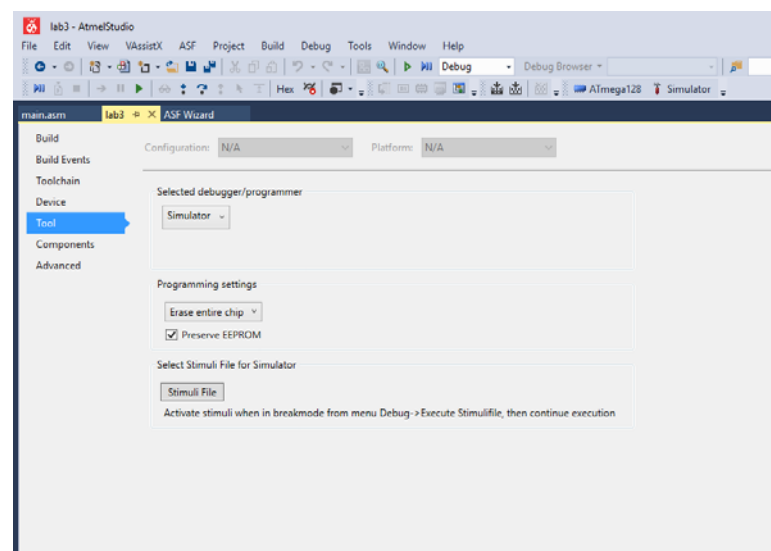
Demo Session

- Check-off to your lab TA only.
- You have to include source code in your lab report. If you are missing one of them (lab report, asm file, and check off), you can not receive credit for Demo.
- Don't forget comments!
 - Your AVR assembly code need to be well-commented.

AVR Simulation

- Learn how to use the Atmel Studio simulator
- Set Break Points
- Open Processor, I/O port, and Memory Window

Simulator



Break Points

The screenshot shows the Atmel Studio IDE with the assembly code for 'main.asm'. Several breakpoints are set, indicated by red dots in the left margin. The code includes an initialization routine and a main program with two loops. Comments indicate where to set breakpoints for specific instructions.

```

main.asm - lab3
File Edit View Assistant ASF Project Build Debug Tools Window Help
lab3 Debug
main.asm lab3 ASF Wizard
;*****
; The initialization routine
; initialize Stack Pointer
INIT:
    ldi mpr, low(RAMEND)
    out SPH, mpr
    ldi mpr, high(RAMEND)
    out SPH, mpr
;*****
; Main Program
;*****
MAIN:
    cli r0 ; *** SET BREAKPOINT HERE *** (#1)
    dec r0 ; initialize r0 value
;*****
    cli r1 ; *** SET BREAKPOINT HERE *** (#2)
    ldi i, $04
    ldi r1
    inc r1
    ldi r1
    dec i
    brne LOOP ; *** SET BREAKPOINT HERE *** (#3)
;*****
    cli r2 ; *** SET BREAKPOINT HERE *** (#4)
    ldi i, $0F
    inc r2 ; initialize r2 value
    loop2:
    cp r2, i
    brne LOOP2 ; *** SET BREAKPOINT HERE *** (#5)
;*****
    ; initialize r3 value
    mov r3, r2 ; *** SET BREAKPOINT HERE *** (#6)
;*****
; Note: At this point, you need to enter several values
; directly into the Data Memory. FUNCTION is written to
; expect memory locations $0101:$0106 and $0103:$0102
; to represent two 16-bit operands.
;
; So at this point, the contents of r0, r1, r2, and r3
; MUST be manually typed into Data Memory locations

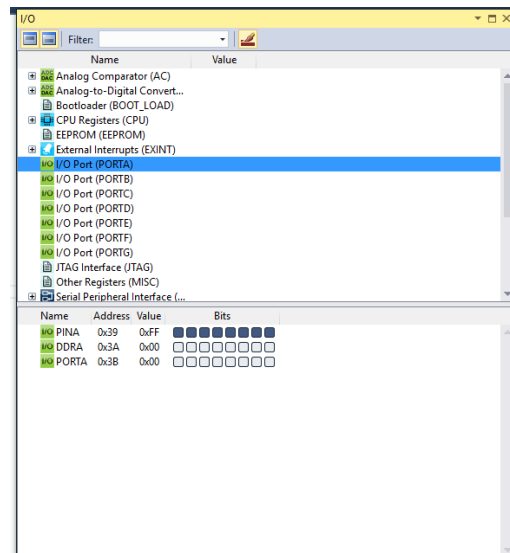
```

Processor Status

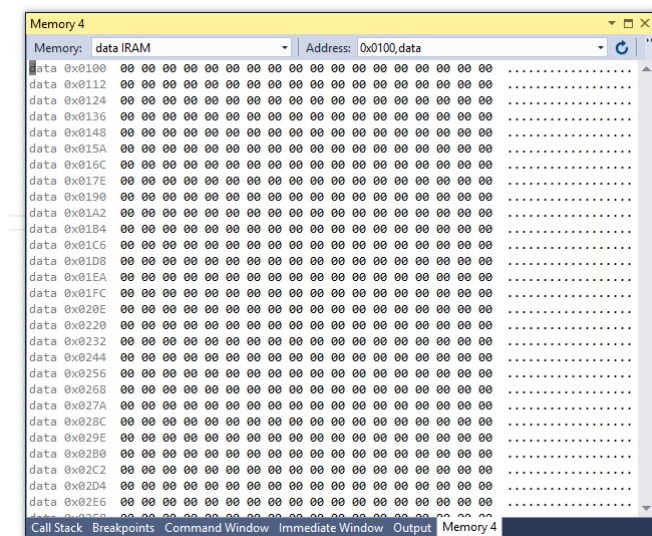
The screenshot shows the 'Processor Status' window, which displays the current state of the processor. It includes a table of registers and their values, as well as other status information like the program counter, stack pointer, and cycle counter.

Name	Value
Program Counter	0x00000000
Stack Pointer	0x0000
X Register	0x0000
Y Register	0x0000
Z Register	0x0000
Status Register	0x00000000
Cycle Counter	0
Frequency	1.000 MHz
Stop Watch	0.00 µs
Registers	
R00	0x00
R01	0x00
R02	0x00
R03	0x00
R04	0x00
R05	0x00
R06	0x00
R07	0x00
R08	0x00
R09	0x00
R10	0x00
R11	0x00
R12	0x00
R13	0x00
R14	0x00
R15	0x00
R16	0x00
R17	0x00

I/O Ports



Memory



Questions?

