Temple University

College of Engineering

Department of Electrical and Computer Engineering (ECE)

Student Lab Report Cover Page

Course Number	**:**	3613
Course Section	**:**	002
Experiment #	**:**	Lab 2

Student Name (print)	**:**	Robert Bara
TUid#	**:**	915614617
Date	**:**	9/9/2020

Grade	**:**	______ **/100**
TA Name	**:**	Sung Choi

LAB 2.

Write assembly code to complete four given tasks and examine the result using the Atmel Studio IDE.

Lab 2 includes four activities. For each activity, write simple assembly code and run it using the Atmel Studio IDE. Then, examine the result to prove the code performs correctly to satisfy the given tasks. Write assembly code to complete four given task and examine the result using the Atmel Studio IDE. Follow the example to complete the activities.

<Example>

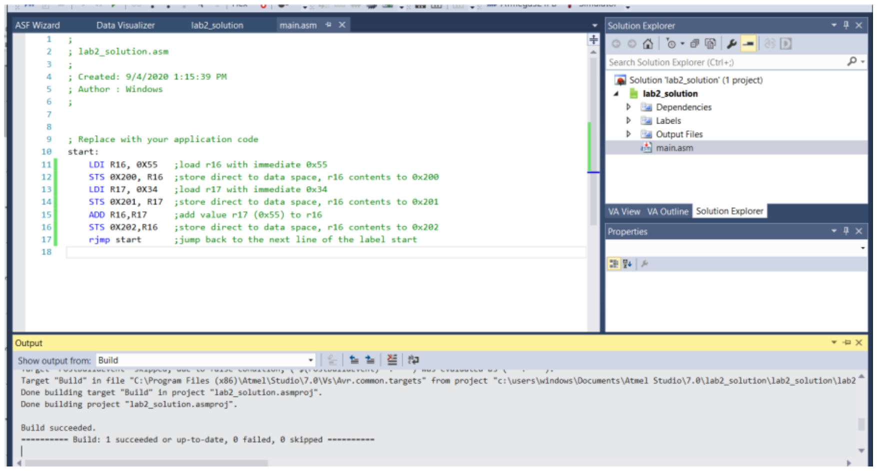
Task: Add numbers, 0x55 and 0x34. Store each value to the memory location, 0x200, and 0x201, respectively. The result of addition must be stored at the location 0x202.

1) Code: AVR Assembly code (add comments for every line of code)

```
start:
        LDI R16, 0X55           ;load r16 with immediate 0x55
        STS 0X200, R16  ;store direct to data space, r16 contents to 0x200
        LDI R17, 0X34           ;load r17 with immediate 0x34
        STS 0X201, R17  ;store direct to data space, r16 contents to 0x201
        ADD R16, R17            ;add value r17 (0x55) to r16
        STS 0X202, R16  ;store direct to data space, r16 contents to 0x202
        RJMP start              ;jump back to the next line of the label start
```

2) Run: Build the code on Atmel Studio IDE and check there is no error.

The code was built successfully with no error.



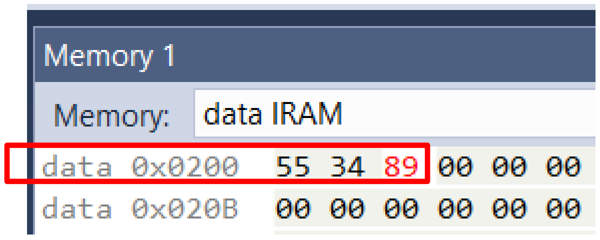
```
;
; lab2_solution.asm
;
; Created: 9/4/2020 1:15:39 PM
; Author : Windows
;


; Replace with your application code
start:
    LDI R16, 0X55   ;load r16 with immediate 0x55
    STS 0X200, R16  ;store direct to data space, r16 contents to 0x200
    LDI R17, 0X34   ;load r17 with immediate 0x34
    STS 0X201, R17  ;store direct to data space, r16 contents to 0x201
    ADD R16,R17     ;add value r17 (0x55) to r16
    STS 0X202,R16   ;store direct to data space, r16 contents to 0x202
    rjmp start      ;jump back to the next line of the label start
```

```
Target "Build" in file "C:\Program Files (x86)\Atmel\Studio\7.0\Vs\Avr.common.targets" from project "c:\users\windows\Documents\Atmel Studio\7.0\lab2_solution\lab2_solution\lab2
Done building target "Build" in project "lab2_solution.asmproj".
Done building project "lab2_solution.asmproj".

Build succeeded.
========== Build: 1 succeeded or up-to-date, 0 failed, 0 skipped ==========
```

<Figure. Atmel Studio view of build solution – Build successfully>

3) Result: Examine the contents of registers and memory when you execute code. Show all results that are required from the task. **Put the appropriate label and description for each result.**

Store 0x55 to the memory location 0x200 and 0x34 to the memory location 0x201, and Store the sum into the memory location 0x202



```
data 0x0200   55 34 89 00 00 00
data 0x020B   00 00 00 00 00 00
```

<Figure. RAM location and contents>

ACTIVITIES:

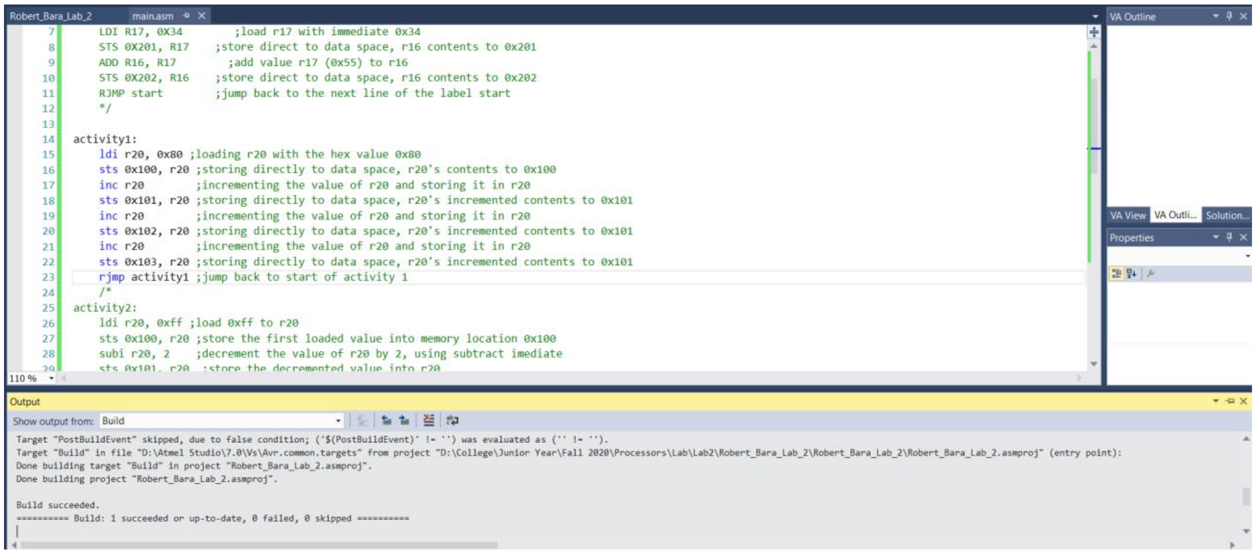
Read the activities carefully and complete the task. Fill out each section, code, run, and result based on the given tasks.

Activity 1. Load a number 0x80 to R20 and increment 3 times. Store the contents of R20 to the memory location 0x100, 0x101, 0x102, and 0x103 for every time the value increases.

1) Code:

```
activity1:
          ldi r20, 0x80 ;loading r20 with the hex value 0x80
          sts 0x100, r20 ;storing directly to data space, r20's contents to 0x100
          inc r20                 ;incrementing the value of r20 and storing it in r20
          sts 0x101, r20 ;storing directly to data space, r20's incremented contents
   to 0x101
          inc r20        ;incrementing the value of r20 and storing it in r20
          sts 0x102, r20 ;storing directly to data space, r20's incremented contents
   to 0x101
          inc r20        ;incrementing the value of r20 and storing it in r20
          sts 0x103, r20 ;storing directly to data space, r20's incremented contents
   to 0x101
          rjmp activity1 ;jump back to start of activity 1
```

2) Run:



Robert_Bara_Lab_2   main.asm

```
    LDI R17, 0X34        ;load r17 with immediate 0x34
    STS 0X201, R17    ;store direct to data space, r16 contents to 0x201
    ADD R16, R17        ;add value r17 (0x55) to r16
    STS 0X202, R16    ;store direct to data space, r16 contents to 0x202
    RJMP start        ;jump back to the next line of the label start
    */

activity1:
    ldi r20, 0x80 ;loading r20 with the hex value 0x80
    sts 0x100, r20 ;storing directly to data space, r20's contents to 0x100
    inc r20        ;incrementing the value of r20 and storing it in r20
    sts 0x101, r20 ;storing directly to data space, r20's incremented contents to 0x101
    inc r20        ;incrementing the value of r20 and storing it in r20
    sts 0x102, r20 ;storing directly to data space, r20's incremented contents to 0x101
    inc r20        ;incrementing the value of r20 and storing it in r20
    sts 0x103, r20 ;storing directly to data space, r20's incremented contents to 0x101
    rjmp activity1 ;jump back to start of activity 1
    /*
activity2:
    ldi r20, 0xff ;load 0xff to r20
    sts 0x100, r20 ;store the first loaded value into memory location 0x100
    subi r20, 2    ;decrement the value of r20 by 2, using subtract imediate
    sts 0x101, r20  ;store the decremented value into r20
```

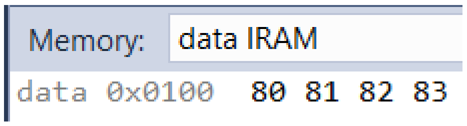
Output

Show output from: Build

```
Target "PostBuildEvent" skipped, due to false condition; ('$(PostBuildEvent)' != '') was evaluated as ('' != '').
Target "Build" in file "D:\Atmel Studio\7.0\Vs\Avr.common.targets" from project "D:\College\Junior Year\Fall 2020\Processors\Lab\Lab2\Robert_Bara_Lab_2\Robert_Bara_Lab_2\Robert_Bara_Lab_2.asmproj" (entry point):
Done building target "Build" in project "Robert_Bara_Lab_2.asmproj".
Done building project "Robert_Bara_Lab_2.asmproj".

Build succeeded.
========== Build: 1 succeeded or up-to-date, 0 failed, 0 skipped ==========
```

[Figure: Build successfully for Activity 1]

3) Result:



Memory: data IRAM

```
data 0x0100   80 81 82 83
```

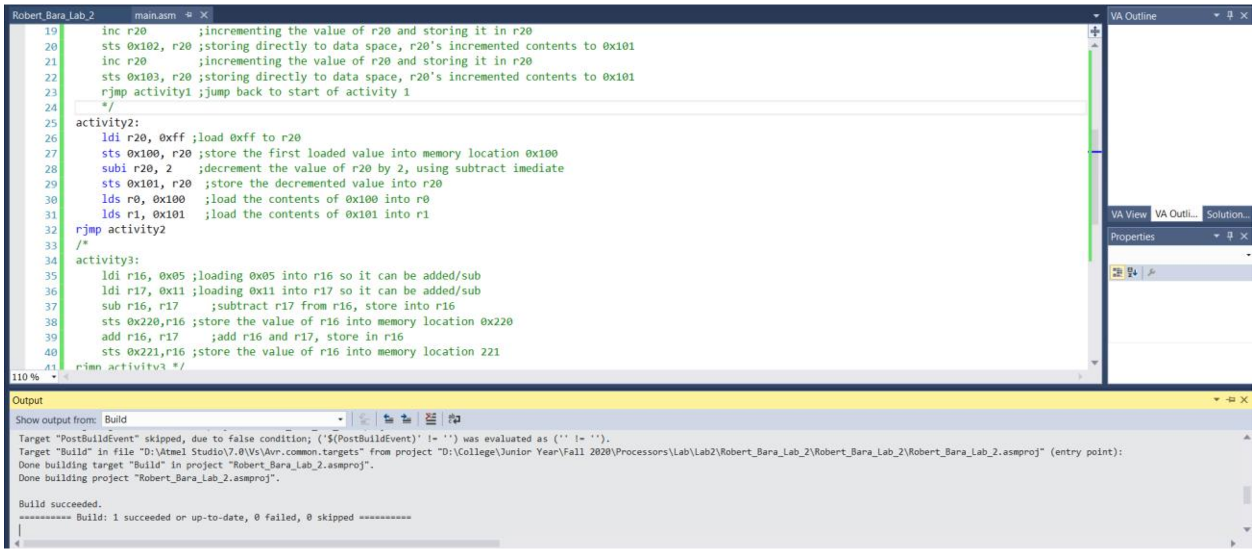
[Figure: Ram and Contents for Activity 1]

Activity 2. Load a number 0xff to R20 and decrement the value of R20 by 2. Store the first loaded value 0xff into the memory location 0x100 and the value after decrement into 0x101. Load the contents of 0x100 to R0 and the contents of 0x101 to R1.

1) Code:

```
activity2:
    ldi r20, 0xff ;load 0xff to r20
    sts 0x100, r20 ;store the first loaded value into memory location 0x100
    subi r20, 2    ;decrement the value of r20 by 2, using subtract imediate
    sts 0x101, r20  ;store the decremented value into r20
    lds r0, 0x100   ;load the contents of 0x100 into r0
    lds r1, 0x101   ;load the contents of 0x101 into r1
    rjmp activity2
```

2) Run:

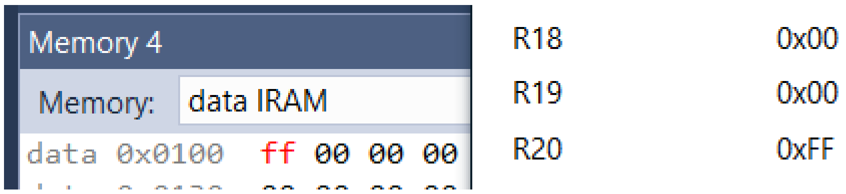


```
    inc r20        ;incrementing the value of r20 and storing it in r20
    sts 0x102, r20 ;storing directly to data space, r20's incremented contents to 0x101
    inc r20        ;incrementing the value of r20 and storing it in r20
    sts 0x103, r20 ;storing directly to data space, r20's incremented contents to 0x101
    rjmp activity1 ;jump back to start of activity 1
    */
activity2:
    ldi r20, 0xff ;load 0xff to r20
    sts 0x100, r20 ;store the first loaded value into memory location 0x100
    subi r20, 2    ;decrement the value of r20 by 2, using subtract imediate
    sts 0x101, r20  ;store the decremented value into r20
    lds r0, 0x100   ;load the contents of 0x100 into r0
    lds r1, 0x101   ;load the contents of 0x101 into r1
rjmp activity2
/*
activity3:
    ldi r16, 0x05 ;loading 0x05 into r16 so it can be added/sub
    ldi r17, 0x11 ;loading 0x11 into r17 so it can be added/sub
    sub r16, r17     ;subtract r17 from r16, store into r16
    sts 0x220,r16 ;store the value of r16 into memory location 0x220
    add r16, r17     ;add r16 and r17, store in r16
    sts 0x221,r16 ;store the value of r16 into memory location 221
rjmp activity3 */
```

```
Target "PostBuildEvent" skipped, due to false condition; ('$(PostBuildEvent)' != '') was evaluated as ('' != '').
Target "Build" in file "D:\Atmel Studio\7.0\Vs\Avr.common.targets" from project "D:\College\Junior Year\Fall 2020\Processors\Lab\Lab2\Robert_Bara_Lab_2\Robert_Bara_Lab_2\Robert_Bara_Lab_2.asmproj" (entry point):
Done building target "Build" in project "Robert_Bara_Lab_2.asmproj".
Done building project "Robert_Bara_Lab_2.asmproj".

Build succeeded.
========== Build: 1 succeeded or up-to-date, 0 failed, 0 skipped ==========
```

[Figure: Build successfully for Activity 2]

3) Result:



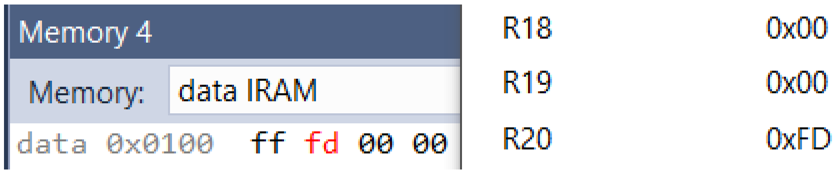
Memory 4

Memory: data IRAM

```
data 0x0100  ff 00 00 00
```

R18	0x00
R19	0x00
R20	0xFF

[Figure: Ram location and contents & Processor Status for first loaded values]



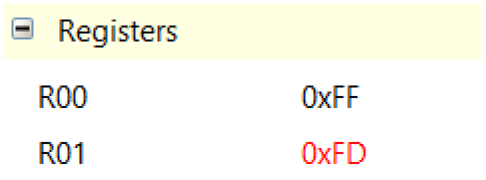
Memory 4

Memory: data IRAM

```
data 0x0100  ff fd 00 00
```

R18	0x00
R19	0x00
R20	0xFD

[Figure: Ram location and contents & Processor Status for decremented by 2 values]



Registers

R00	0xFF
R01	0xFD

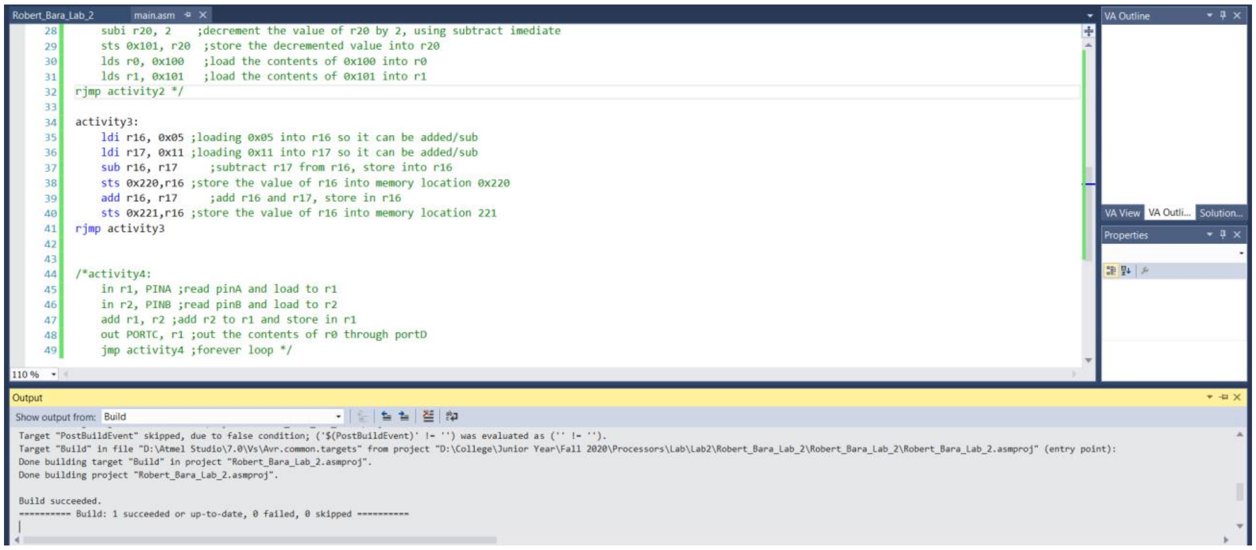
[Figure: Load the contents of 0x100 and 0x101 into r0 and r1]

Activity 3. Subtract a number 0x05 from 0x11. Store this value into the memory location 0x220. Add the number 0x05 and 0x11. Store this value into the memory location 0x221.

1) Code:

```
activity3:
	ldi r16, 0x05 ;loading 0x05 into r16 so it can be added/sub
	ldi r17, 0x11 ;loading 0x11 into r17 so it can be added/sub
	sub r16, r17   ;subtract r17 from r16, store into r16
	sts 0x220,r16 ;store the value of r16 into memory location 0x220
	add r16, r17   ;add r16 and r17, store in r16
	sts 0x221,r16 ;store the value of r16 into memory location 221
	rjmp activity3
```

2) Run:

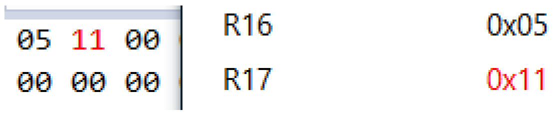


```
	subi r20, 2    ;decrement the value of r20 by 2, using subtract imediate
	sts 0x101, r20  ;store the decremented value into r20
	lds r0, 0x100   ;load the contents of 0x100 into r0
	lds r1, 0x101   ;load the contents of 0x101 into r1
rjmp activity2 */

activity3:
	ldi r16, 0x05 ;loading 0x05 into r16 so it can be added/sub
	ldi r17, 0x11 ;loading 0x11 into r17 so it can be added/sub
	sub r16, r17     ;subtract r17 from r16, store into r16
	sts 0x220,r16 ;store the value of r16 into memory location 0x220
	add r16, r17     ;add r16 and r17, store in r16
	sts 0x221,r16 ;store the value of r16 into memory location 221
rjmp activity3


/*activity4:
	in r1, PINA ;read pinA and load to r1
	in r2, PINB ;read pinB and load to r2
	add r1, r2 ;add r2 to r1 and store in r1
	out PORTC, r1 ;out the contents of r0 through portD
	jmp activity4 ;forever loop */
```

```
Target "PostBuildEvent" skipped, due to false condition; ('$(PostBuildEvent)' != '') was evaluated as ('' != '').
Target "Build" in file "D:\Atmel Studio\7.0\Vs\Avr.common.targets" from project "D:\College\Junior Year\Fall 2020\Processors\Lab\Lab2\Robert_Bara_Lab_2\Robert_Bara_Lab_2\Robert_Bara_Lab_2.asmproj" (entry point):
Done building target "Build" in project "Robert_Bara_Lab_2.asmproj".
Done building project "Robert_Bara_Lab_2.asmproj".

Build succeeded.
========== Build: 1 succeeded or up-to-date, 0 failed, 0 skipped ==========
```

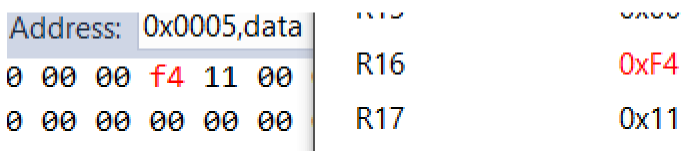
[Figure: Build Successfully for activity 3]

3) Result:



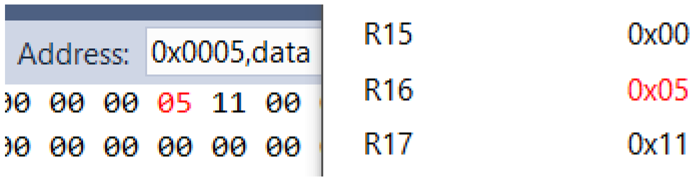
```
05 11 00     R16     0x05
00 00 00     R17     0x11
```

[Figure: Load Registers and Ram locations before subtraction]



```
Address: 0x0005,data
0 00 00 f4 11 00     R16     0xF4
0 00 00 00 00 00     R17     0x11
```

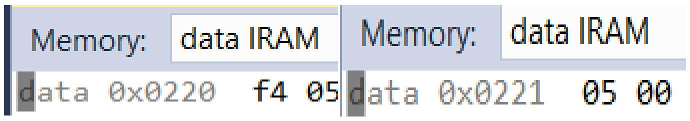
[Figure: Load Registers and Ram locations after subtraction]



```
Address: 0x0005,data
)0 00 00 05 11 00
)0 00 00 00 00 00
```

R15	0x00
R16	0x05
R17	0x11

[Figure: Load Registers and Ram locations after addition]



```
Memory: data IRAM
data 0x0220  f4 05
Memory: data IRAM
data 0x0221  05 00
```

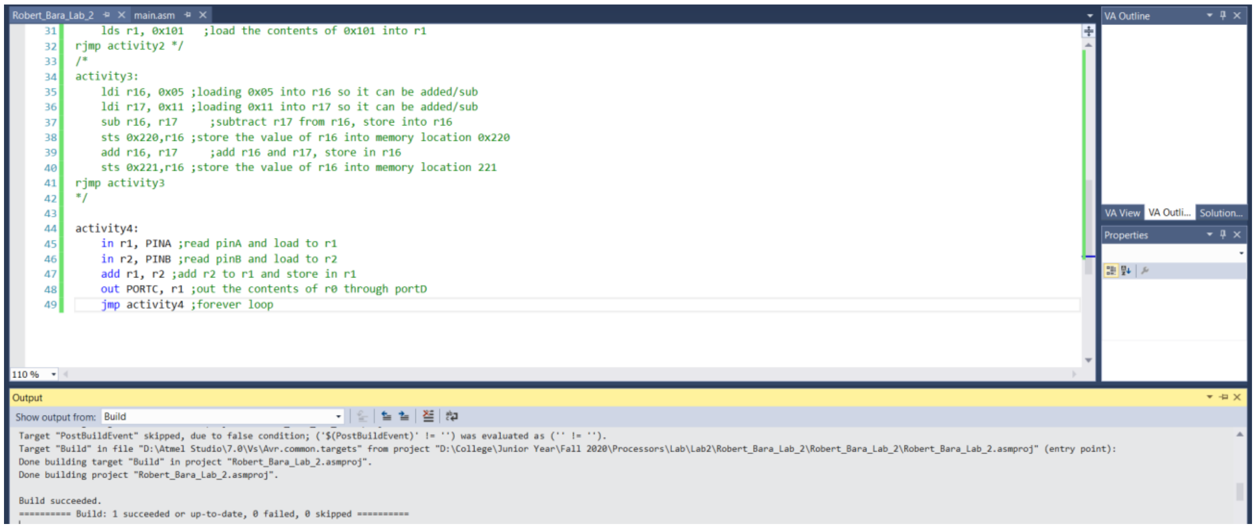
[Figure: Ram locations upon completed code cycle]

Activity 4. Load R1 with PINA and R2 with PINB. Add the contents of R1 and R2 and out the sum of value through PORTC. (Follow the steps of "How to read PINx values and make output in simulation mode" attached at the end of this lab manual.)

1) Code:

```
activity4:
	in r1, PINA ;read pinA and load to r1
	in r2, PINB ;read pinB and load to r2
	add r1, r2 ;add r2 to r1 and store in r1
	out PORTC, r1 ;out the contents of r0 through portD
	jmp activity4 ;forever loop
```

2) Run:

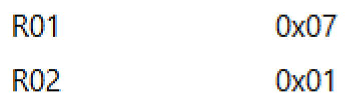


```
	lds r1, 0x101	;load the contents of 0x101 into r1
rjmp activity2 */
/*
activity3:
	ldi r16, 0x05 ;loading 0x05 into r16 so it can be added/sub
	ldi r17, 0x11 ;loading 0x11 into r17 so it can be added/sub
	sub r16, r17	;subtract r17 from r16, store into r16
	sts 0x220,r16 ;store the value of r16 into memory location 0x220
	add r16, r17	;add r16 and r17, store in r16
	sts 0x221,r16 ;store the value of r16 into memory location 221
rjmp activity3
*/

activity4:
	in r1, PINA ;read pinA and load to r1
	in r2, PINB ;read pinB and load to r2
	add r1, r2 ;add r2 to r1 and store in r1
	out PORTC, r1 ;out the contents of r0 through portD
	jmp activity4 ;forever loop
```

```
Target "PostBuildEvent" skipped, due to false condition; ('$(PostBuildEvent)' != '') was evaluated as ('' != '').
Target "Build" in file "D:\Atmel Studio\7.0\Vs\Avr.common.targets" from project "D:\College\Junior Year\Fall 2020\Processors\Lab\Lab2\Robert_Bara_Lab_2\Robert_Bara_Lab_2\Robert_Bara_Lab_2.asmproj" (entry point):
Done building target "Build" in project "Robert_Bara_Lab_2.asmproj".
Done building project "Robert_Bara_Lab_2.asmproj".

Build succeeded.
========== Build: 1 succeeded or up-to-date, 0 failed, 0 skipped ==========
```

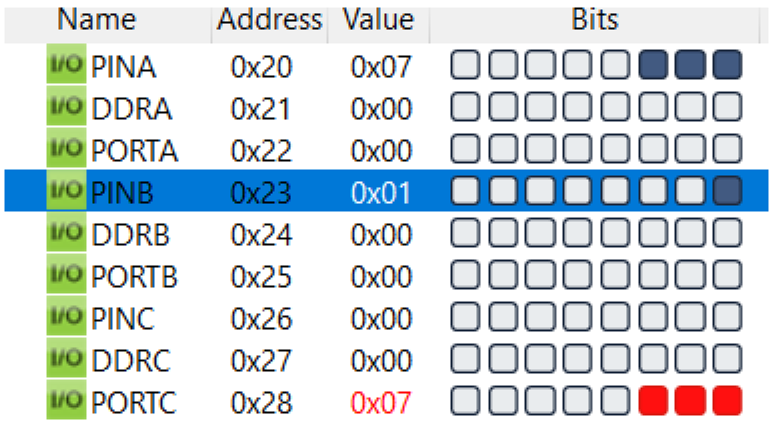
[Figure: Build successful for activity 4]

3) Result:



R01	0x07
R02	0x01

[Figure: Processor Status before the addition]



Name	Address	Value	Bits
PINA	0x20	0x07	☐☐☐☐☐■■■
DDRA	0x21	0x00	☐☐☐☐☐☐☐☐
PORTA	0x22	0x00	☐☐☐☐☐☐☐☐
PINB	0x23	0x01	☐☐☐☐☐☐☐■
DDRB	0x24	0x00	☐☐☐☐☐☐☐☐
PORTB	0x25	0x00	☐☐☐☐☐☐☐☐
PINC	0x26	0x00	☐☐☐☐☐☐☐☐
DDRC	0x27	0x00	☐☐☐☐☐☐☐☐
PORTC	0x28	0x07	☐☐☐☐☐🟥🟥🟥

[Figure: Port IO status after the cycle is completed]

LAB 2 Grading Rubric

Activity	Task	Full Points	Earned Points	Comment
1	Code	10		Complete code (5pts) and comments (5pts)
	Run	5		No syntax error
	Result	10		R20 and the memory contents of 0x100, 0x101, 0x102, 0x103 (2pts ea.)
Subtotal		25		
2	Code	10		Complete code (5pts) and comments (5pts)
	Run	5		No syntax error
	Result	10		R20 before and after decrementing (1pt ea.), memory contents of 0x100 and 0x101 (2 pts ea.), R0 and R1 (2 pts ea.)
Subtotal		25		

3	Code	10		Complete code (5) and comments (5)
	Run	5		No syntax error
	Result	10		Contents of 0x220 and 0x221 (5 pts ea.)
Subtotal		25		
3	Code	10		Complete code (5) and comments (5)
	Run	5		No syntax error
	Result	10		R1,R2, PINA, PINB, and PORTC values (2pts ea.)
Subtotal		25		
Total		**100**		