Joseph Bellow

CPE301 – SPRING 2016

Design Assignment 0

DO NOT REMOVE THIS PAGE DURING SUBMISSION:

The student understands that all required components should be submitted in complete for grading of this assignment.

NO	SUBMISSION ITEM	COMPLETED (Y/N)	MARKS (/MAX)
0.	COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS		
1.	INITIAL CODE OF TASK 1/A		
2.	INCREMENTAL / DIFFERENTIAL CODE OF TASK 2/B		
3.	INCREMENTAL / DIFFERENTIAL CODE OF TASK 3/C		
4.	INCREMENTAL / DIFFERENTIAL CODE OF TASK 4/D		
5.	INCREMENTAL / DIFFERENTIAL CODE OF TASK 5/E		
6.	SCHEMATICS		
7.	SCREENSHOTS OF EACH TASK OUTPUT		
8.	SCREENSHOT OF EACH DEMO		
9.	VIDEO LINKS OF EACH DEMO		
10.	GOOGLECODE LINK OF THE DA		

0.	COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS		

We used no components for this project other than the Atmel Studio 7 software.

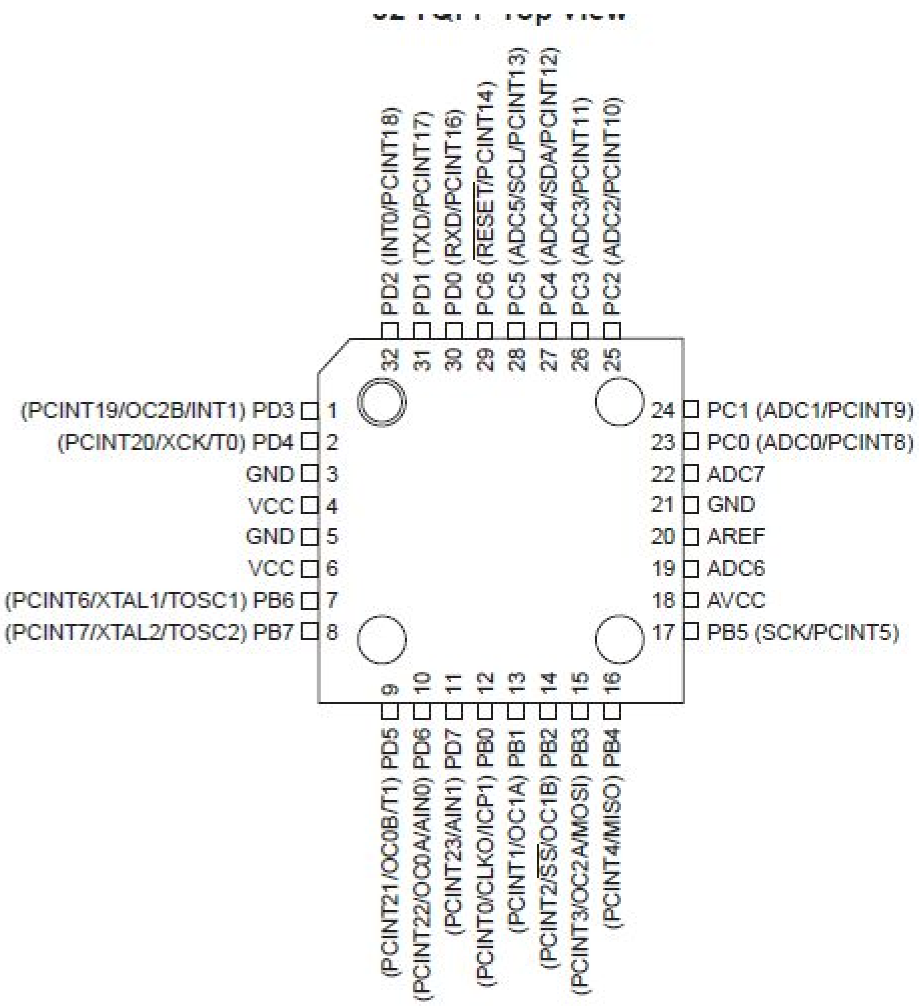
1.	INITIAL CODE OF TASK 1/A		

```
start:

  ldi r16, 30             ;stores the decimal value 30 into register 16

        ldi r17, 34             ;stores the decimal value 34 into register 17

        add r16, r17    ;adds r17 to r16 and stores value in r16 now equal to 62

        ldi r17, 45             ;stores decimal value 45 into r17

        add r16, r17    ;adds r17 to r16 and stores value in r16 now equal to 112

        brvs l1                 ;branch to l1 if overflow bit v in status register is high

        ldi r17, 42             ;stores decimal value 42 into r17

        add r16, r17    ;adds r17 to r16 and stores value in r16 now equal to 155

        brvs l1                 ;branch to l1 if overflow bit v in status register is high

        ldi r17, 50             ;stores decimal value 50 into r17

        add r16, r17    ;adds r17 to r16 and stores value in r16 now equal to 189

        brvs l1                 ;branch to l1 if overflow bit v in status register is high

        cbi portb,2             ;clear bit in port b position 2

        rjmp l2 ;If the sum produces an overflow set PORTB.2 pin = HIGH else PORTB.2 pin = LOW

        l1:

        sbi portb, 2    ;sets port b pin 2 high

        rjmp l1                 ;keeps looping l1 to keep port b pin 2 high

        l2:

        nop
```

2.	Task B		

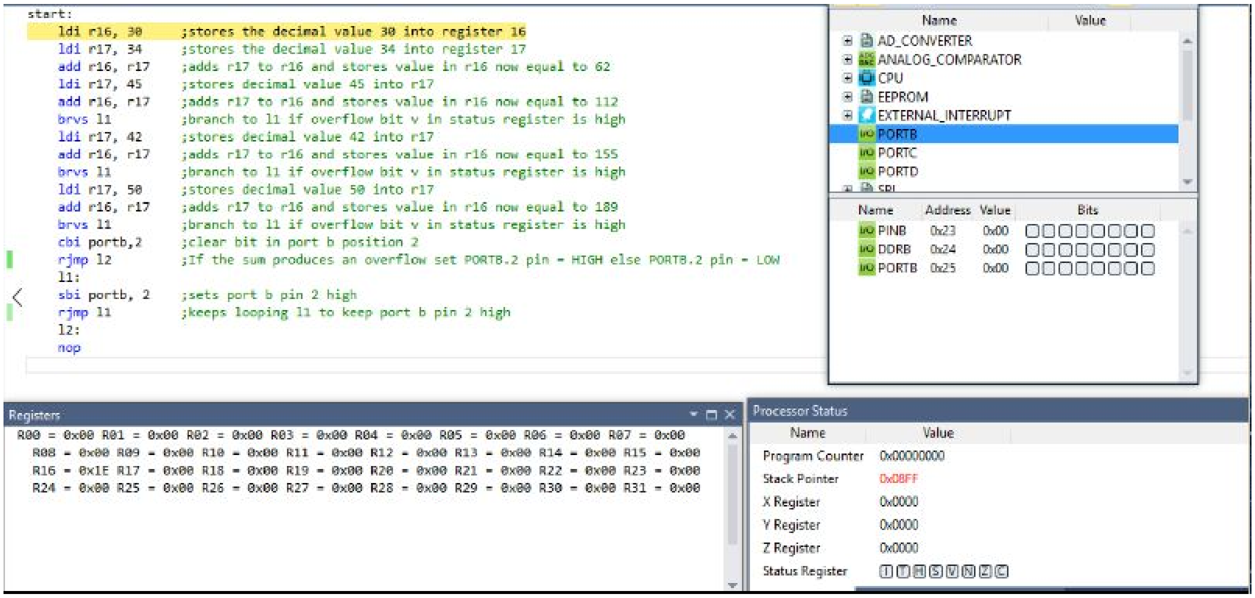
Execution time was found to be 1.38 us. A picture is shown in section 7.

6.	SCHEMATICS		



7.	SCREENSHOTS OF EACH TASK OUTPUT		

Screenshot at start of task A.



```
start:
    ldi r16, 30      ;stores the decimal value 30 into register 16
    ldi r17, 34      ;stores the decimal value 34 into register 17
    add r16, r17     ;adds r17 to r16 and stores value in r16 now equal to 62
    ldi r17, 45      ;stores decimal value 45 into r17
    add r16, r17     ;adds r17 to r16 and stores value in r16 now equal to 112
    brvs l1          ;branch to l1 if overflow bit v in status register is high
    ldi r17, 42      ;stores decimal value 42 into r17
    add r16, r17     ;adds r17 to r16 and stores value in r16 now equal to 155
    brvs l1          ;branch to l1 if overflow bit v in status register is high
    ldi r17, 50      ;stores decimal value 50 into r17
    add r16, r17     ;adds r17 to r16 and stores value in r16 now equal to 189
    brvs l1          ;branch to l1 if overflow bit v in status register is high
    cbi portb,2      ;clear bit in port b position 2
    rjmp l2          ;If the sum produces an overflow set PORTB.2 pin = HIGH else PORTB.2 pin = LOW
    l1:
    sbi portb, 2     ;sets port b pin 2 high
    rjmp l1          ;keeps looping l1 to keep port b pin 2 high
    l2:
    nop
```

Name	Value
AD_CONVERTER	
ANALOG_COMPARATOR	
CPU	
EEPROM	
EXTERNAL_INTERRUPT	
PORTB	
PORTC	
PORTD	
SPI	

Name	Address	Value	Bits
PINB	0x23	0x00	□□□□□□□□
DDRB	0x24	0x00	□□□□□□□□
PORTB	0x25	0x00	□□□□□□□□

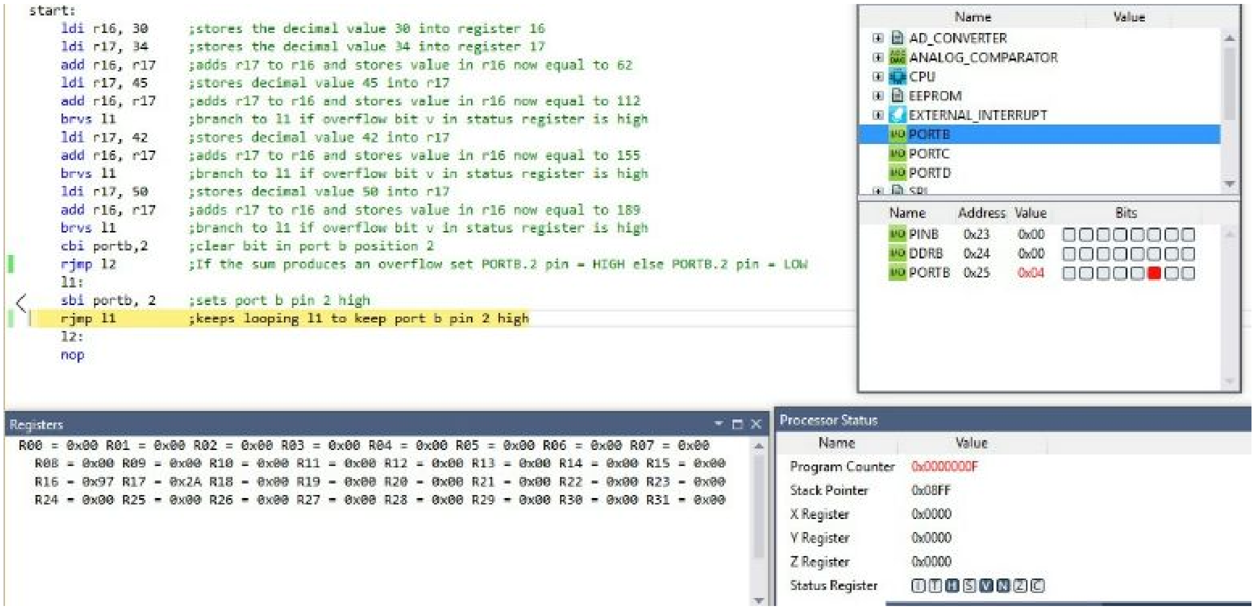
Registers

```
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 = 0x1E R17 = 0x00 R18 = 0x00 R19 = 0x00 R20 = 0x00 R21 = 0x00 R22 = 0x00 R23 = 0x00
  R24 = 0x00 R25 = 0x00 R26 = 0x00 R27 = 0x00 R28 = 0x00 R29 = 0x00 R30 = 0x00 R31 = 0x00
```

Processor Status

Name	Value
Program Counter	0x00000000
Stack Pointer	0x08FF
X Register	0x0000
Y Register	0x0000
Z Register	0x0000
Status Register	I T H S V N Z C

Screenshot at end of task A showing output on port B pin 2.



```
start:
    ldi r16, 30      ;stores the decimal value 30 into register 16
    ldi r17, 34      ;stores the decimal value 34 into register 17
    add r16, r17     ;adds r17 to r16 and stores value in r16 now equal to 62
    ldi r17, 45      ;stores decimal value 45 into r17
    add r16, r17     ;adds r17 to r16 and stores value in r16 now equal to 112
    brvs l1          ;branch to l1 if overflow bit v in status register is high
    ldi r17, 42      ;stores decimal value 42 into r17
    add r16, r17     ;adds r17 to r16 and stores value in r16 now equal to 155
    brvs l1          ;branch to l1 if overflow bit v in status register is high
    ldi r17, 50      ;stores decimal value 50 into r17
    add r16, r17     ;adds r17 to r16 and stores value in r16 now equal to 189
    brvs l1          ;branch to l1 if overflow bit v in status register is high
    cbi portb,2      ;clear bit in port b position 2
    rjmp l2          ;If the sum produces an overflow set PORTB.2 pin = HIGH else PORTB.2 pin = LOW
    l1:
    sbi portb, 2     ;sets port b pin 2 high
    rjmp l1          ;keeps looping l1 to keep port b pin 2 high
    l2:
    nop
```

Name	Value
AD_CONVERTER	
ANALOG_COMPARATOR	
CPU	
EEPROM	
EXTERNAL_INTERRUPT	
PORTB	
PORTC	
PORTD	
SPI	

Name	Address	Value	Bits
PINB	0x23	0x00	□□□□□□□□
DDRB	0x24	0x00	□□□□□□□□
PORTB	0x25	0x04	□□□□□■□□

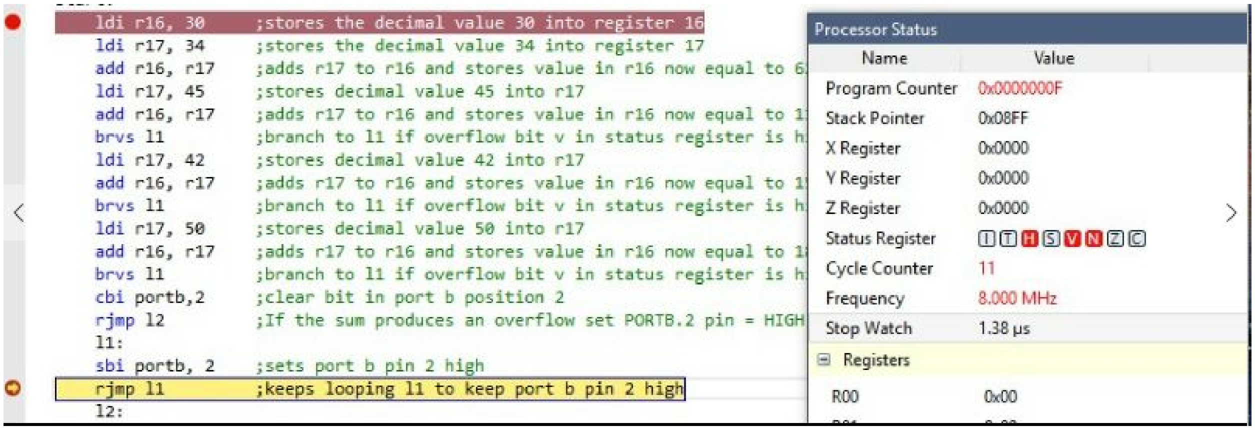
Registers

```
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 = 0x97 R17 = 0x2A R18 = 0x00 R19 = 0x00 R20 = 0x00 R21 = 0x00 R22 = 0x00 R23 = 0x00
  R24 = 0x00 R25 = 0x00 R26 = 0x00 R27 = 0x00 R28 = 0x00 R29 = 0x00 R30 = 0x00 R31 = 0x00
```

Processor Status

Name	Value
Program Counter	0x0000000F
Stack Pointer	0x08FF
X Register	0x0000
Y Register	0x0000
Z Register	0x0000
Status Register	I T H S V N Z C

Screenshot for task B checking execution time.



```
    ldi r16, 30       ;stores the decimal value 30 into register 16
    ldi r17, 34       ;stores the decimal value 34 into register 17
    add r16, r17      ;adds r17 to r16 and stores value in r16 now equal to 6
    ldi r17, 45       ;stores decimal value 45 into r17
    add r16, r17      ;adds r17 to r16 and stores value in r16 now equal to 1
    brvs l1           ;branch to l1 if overflow bit v in status register is h
    ldi r17, 42       ;stores decimal value 42 into r17
    add r16, r17      ;adds r17 to r16 and stores value in r16 now equal to 1
    brvs l1           ;branch to l1 if overflow bit v in status register is h
    ldi r17, 50       ;stores decimal value 50 into r17
    add r16, r17      ;adds r17 to r16 and stores value in r16 now equal to 1
    brvs l1           ;branch to l1 if overflow bit v in status register is h
    cbi portb,2       ;clear bit in port b position 2
    rjmp l2           ;If the sum produces an overflow set PORTB.2 pin = HIGH
    l1:
    sbi portb, 2      ;sets port b pin 2 high
    rjmp l1           ;keeps looping l1 to keep port b pin 2 high
    l2:
```

Processor Status

Name	Value
Program Counter	0x0000000F
Stack Pointer	0x08FF
X Register	0x0000
Y Register	0x0000
Z Register	0x0000
Status Register	I T H S V N Z C
Cycle Counter	11
Frequency	8.000 MHz
Stop Watch	1.38 µs
⊟ Registers	
R00	0x00

8.	SCREENSHOT OF EACH DEMO		

Not applicable to this assignment.

9.	VIDEO LINKS OF EACH DEMO		
Not applicable to this assignment			
10.	GOOGLECODE LINK OF THE DA		
http:// @svn or github repository link			

Student Academic Misconduct Policy
http://studentconduct.unlv.edu/misconduct/policy.html

"This assignment submission is my own, original work".
Joseph Bellow