CPE301 – FALL 2019

Design Assignment 1B

Student Name: Worku Tafara
Student #: 2001245644
Student Email: tafarw1@unlv.nevada.edu
Primary Github address: https://github.com/WorkuT1226/CPE301.git
Directory:

Submit the following for all Labs:

- In the document, for each task submit the modified or included code (only) with highlights and justifications of the modifications. Also, include the comments.
- Use the previously create a Github repository with a random name (no CPE/301, Lastname, Firstname). Place all labs under the root folder ESD301/DA, sub-folder named LABXX, with one document and one video link file for each lab, place modified asm/c files named as LabXX-TYY.asm/c.
- If multiple asm/c files or other libraries are used, create a folder LabXX-TYY and place these files inside the folder.
- The folder should have a) Word document (see template), b) source code file(s) and other include files, c) text file with youtube video links (see template).

- **COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS**

List of Components used
Block diagram with pins used in the Atmega328P

- **INITIAL/MODIFIED/DEVELOPED CODE OF TASK 1/A**

```
.EQU STARTADDS = 0x0200 //set STARTADDS to 0x200
.EQU COUNTER = 250 //set counter to 250

.org 0

CLR R0 //clear R0
LDI XL, LOW(STARTADDS) //sets low bits of register X (R26)
```

```
LDI XH, HIGH(STARTADDS)//set high bits of register X (R27)
LDI YL, LOW(0x300)//set low bits of register Y(R28) to 0x300
LDI YH, HIGH(0x300)//set high bits of register Y(R29) to 0x300
LDI ZL, LOW(0x500)//set low bits of register Y(R30) to 0x500
LDI ZH, HIGH(0x500)//set high bits of register Y(R31) to 0x500

LDI R22, 6 //load immediate value 6 into R22
LDI R21, Counter //load counter value into R21

BEGIN:
        MOV R23, R22 //move value in R22 to R23.
        ST      X+,  R22 //store value X is pointing at to R22 then increment by 1.
        JMP L1 //jump to L1 target

L1:
        SUBI R23, 0x05 //subtract immediate value 0x05 from value in R23
        BREQ L3 //branch if equal jump to L3 target
        BREQ L2 //branch if equal jump to L2 target.
        JMP L1 //jump to L1 target

L2:     ST Y+, R22 //store value Y is pointing at to R22 then increment by 1.
        ADD R16, R22 //add value in R22 to R16
        ADC R17, R0 //add with carry value in R0 to R17
        JMP END // jump to END target

L3:
        ST Z+, R22 //store value Z is pointing at to R28 then increment by 1
        ADD R18, R22 // add value in R28 to R18.
        ADC R19, R0 //add with carry value in R0 to R19.
        JMP END //jump to END target

END:    INC R22 //increment R28 by 1.
        DEC R21 //decrement counter by 1
        BRNE BEGIN //branch if not equal back to BEGIN target.
        BREAK //break loop
```

- **DEVELOPED MODIFIED CODE OF TASK 2/A from TASK 1/A**
- **SCHEMATICS**

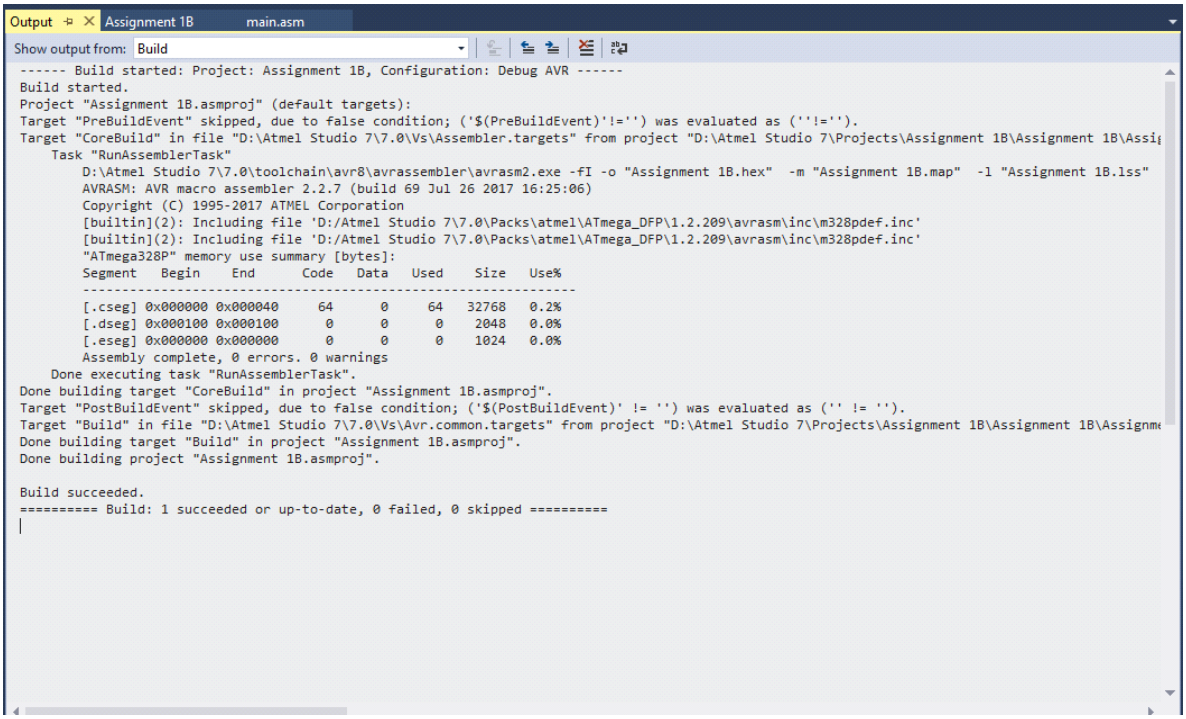
- **SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)**

```
L3:
        ST Z+, R22 //store value Z is pointing at to R28 then increment by 1
        ADD R18, R22 // add value in R28 to R18.
        ADC R19, R0 //add with carry value in R0 to R19.
        JMP END //jump to END target

END:    INC R22 //increment R28 by 1.
        DEC R21 //decrement counter by 1
        BRNE BEGIN //branch if not equal back to BEGIN target.
        BREAK //break loop
```

Output Assignment 1B main.asm*

```
.EQU STARTADDS = 0x0200 //set STARTADDS to 0x200
.EQU COUNTER = 250 //set counter to 250

.org 0

CLR R0 //clear R0
LDI XL, LOW(STARTADDS) //sets low bits of register X (R26)
LDI XH, HIGH(STARTADDS)//set high bits of register X (R27)

LDI YL, LOW(0x300)//set low bits of register Y(R28) to 0x300
LDI YH, HIGH(0x300)//set high bits of register Y(R29) to 0x300


LDI ZL, LOW(0x500)//set low bits of register Y(R30) to 0x500
LDI ZH, HIGH(0x500)//set high bits of register Y(R31) to 0x500

LDI R22, 6 //load immediate value 6 into R22
LDI R21, Counter //load counter value into R21

BEGIN:
        MOV R23, R22 //move value in R22 to R23.
        ST  X+,  R22 //store value X is pointing at to R22 then increment by 1.
        JMP L1 //jump to L1 target

L1:
        SUBI R23, 0x05 //subtract immediate value 0x05 from value in R23
        BREQ L3 //branch if equal jump to L3 target
        BREQ L2 //branch if equal jump to L2 target.
        JMP L1 //jump to L1 target

L2:     ST Y+, R22 //store value Y is pointing at to R22 then increment by 1.
        ADD R16, R22 //add value in R22 to R16
        ADC R17, R0 //add with carry value in R0 to R17
        JMP END // jump to END target
```

```
prog 0xFFCA  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
prog 0xFFDC  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
prog 0xFFEE  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
prog 0x0000  00 24 a0 e0 b2 e0 c0 e0 d3 e0 e0 e0 f5 e0 76 e0 4a ef
prog 0x0012  67 2f 7d 93 0c 94 0d 00 65 50 41 f0 11 f0 0c 94 0d 00
prog 0x0024  79 93 07 0f 10 1d 0c 94 1c 00 71 93 27 0f 30 1d 0c 94
prog 0x0036  1c 00 73 95 4a 95 51 f7 98 95 ff ff ff ff ff ff ff ff
prog 0x0048  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
prog 0x005A  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
```



- **SCREENSHOT OF EACH DEMO (BOARD SETUP)**

- **VIDEO LINKS OF EACH DEMO**

- **GITHUB LINK OF THIS DA**

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

"This assignment submission is my own, original work".
Worku Tafara