



Comment: My lab 0  
was graded live by  
Andrei Pielea



Name/Semester: Aaron Mills / Spring 2021

Grade: /5

Z-23547104

- [5] 0) This lab is designed to help students get acquainted with the MSP430 Launchpad microcontroller training kit. Type the following sample assembly language program which starts at address 0x0200 (&0200h), or simply \$200. The program adds the contents of three consecutive memory locations starting at address \$200. The sum is stored at location \$206. In the following subsections, various commands are listed for you to explore.

In the Code Composer, create a new Assembly Project and insert the following code into section label "Main loop here". You can also copy the entire skeleton program from the text file provided on your Canvas course.

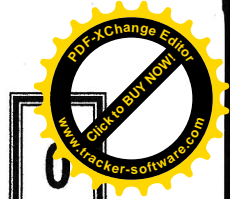
```
LAB1      mov.w  #01, &0200h      ;set a number on location $0200
          mov.w  #02, &0202h      ;set a number on location $0202
          mov.w  #03, &0204h      ;set a number on location $0204

LINEA     clr    R7                ;clear the entire R7 register
          clr    R8                ;clear the entire R8 register
          clr    R9                ;clear the entire R9 register
          clr    R10               ;clear the entire R10 register

LINEB     mov.w  &0200h, R7        ;copy a word from &0200h to R7
          mov.w  &0202h, R8        ;copy a word from &0202h to R8
          mov.w  &0204h, R9        ;copy a word from &0204h to R9

LINEC     mov.b  R7, R10           ;start accumulator in R10 with value form R7
          add.b  R8, R10           ;add to it the content R8
          add.b  R9, R10           ;add to it the content R9
          mov.b  R10, &0206h       ;now store the sum back in memory

Mainloop  jmp     Mainloop        ;Infinite Loop
```



Name/Semester: Aaron Mills / Spring 2021

Grade: /5

[1] 0.a) Exercise 1: Default program execution

Z-23547104

- Insert break point at the Mainloop line
- Build and execute program
- Record value of core registers when program stops at the breakpoint

R7 0x0001, R8 0x0002, R9 0x0003, R10 0x0006, SR 0x0000, NZVC 0000

[2] 0.b) Exercise 2: Memory manipulation

- Soft Reset the micro
- Insert break point at LINEB label
- Insert break point at LINEC label
- Keep the break point at Mainloop line
- Run the program so it stops at LINEB
- Record Values of the following registers:

R7 0x0000, R8 0x0000, R9 0x0000, R10 0x0000, SR 0x0000, NZVC 0000

- Using the memory browser, modify the content of following memory locations by manually typing the new values over the initial values (in decimal notation):

▪ 0x0200 = 02, 0x0202 = 03, 0x0204 = 10 *decimal*

- Run the code, and now it will stop at LINEC
- Record the updated values of the registers:

R7 0x0002, R8 0x0003, R9 0x000A, R10 0x0000, SR 0x0000, NZVC 0000

- Run the code, and now it will stop at Mainloop
- Record the values again:

R7 0x0002, R8 0x0003, R9 0x000A, R10 0x000F, SR 0x0000, NZVC 0000

[2] 0.c) Exercise 3: Register manipulation

- Soft Reset the micro
- Remove the break points at LINEB and keep the ones at LINEC and Mainloop
- Execute program to stop at LINEC
- Record the new values:

R7 0x0001, R8 0x0002, R9 0x0003, R10 0x0000, SR 0x0000, NZVC 0000

- Manually modify the contents of Registers as follows:

▪ R7 = 05, R8 = 01, R9 = 0 (decimal notation)

- Run the code till it stops at the last breakpoint, Mainloop
- Record the new values:

R7 0x0005, R8 0x0001, R9 0x0000, R10 0x0006, SR 0x0000, NZVC 0000