• **CDA3331C • Intro to Microcomputers •** **Lab Assignment** **3**

Name/Semester: Aaron P. Mills / Spring 2021	**Grade:**	**/20**

[2] 3.b) Run your program and verify the results by examining registers

R4 = (a) = 0x0005, in decimal = 5

R5 = (X) = 0x0134, in decimal = 308

R7 = (F) = 0x027C, in decimal = 636

[2] 3.c) Manually change the contents of R4 to 5, a new input value for variable a. Run your program and observe the results

R4 = (a) = 0x0006, in decimal = 6

R5 = (X) = 0x06D4, in decimal = 1748

R7 = (F) = 0x0DBC, in decimal = 3516

[2] 3.d) Manually change the contents of R4 to -7, a new input value for variable a. Run your program and observe the results

R4 = (a) = 0xFFF9, in decimal = -7

R5 = (X) = 0x2E34, in decimal = 11828

R7 = (F) = 0x5C7C, in decimal = 23676

[2] 3.e) what is the maximum value of "a" that the function can execute correctly and why?

a=7 is the maximum value the function can compute correctly. Any integer greater than 7 creates an overflow, meaning x will not be recorded properly. This will then cause the rest of the computation to fail.

• **CDA3331C • Intro to Microcomputers •** **Lab Assignment** **3**

Name/Semester: Aaron P. Mills	**Grade:** /20

Z - 23547104

[20] 3) Implement the following arithmetic function using subroutines as a technique for efficient coding. The only input to the function is variable (a) that is initialized in register R4 at the beginning of the program to 5 and maintained thereafter. The X calculation result is stored in R5. The final answer, F, is stored in R7.

$$F = \left(\frac{4X + 40}{2}\right) \quad \text{where } X = \sum_{i=0}^{i=|a|} (2(i!))$$

[12] 3.a) Write an MSP430 assembly language program that implements the above function using subroutines for Multiplication and Factorial. The overall program structure is given in the skeleton below, which is available on Canvas as well. Please note that the multiply subroutine is included in the skeleton code. Please review how it works:

```
RESET       mov.w   #__STACK_END,SP         ;Initialize stack pointer
StopWDT     mov.w   #WDTPW+WDTHOLD,&WDTCTL  ;Stop WDT

LAB2        mov.w   #5, R4      ;Load "a" into R4

CLEAR       clr     R5          ;clear the entire register
            clr     R6          ;clear the entire register
            clr     R7          ;clear the entire register

XCALC       ...     ...         ;the X calculation part of your program
            ...     ...         ;taking value of R4 as an input
            ...     ...         ;and returning result X in R5

FCALC       ...     ...         ;the final part of your program
            ...     ...         ;taking inputs from R5
            ...     ...         ;and returning result F in R7

MainLoop jmp    Mainloop        ;Infinite Loop


MULT        ...     ...         ;Included in the skeleton code
```

[4] Main Program – [3] XCALC, [1] FCALC

[8] Factorial Subroutine