Course: CDA-4205L Professor: Yan Zhang PhD

Assignment: Lab 6 Due: 11/25/21 at 11:59 pm

Description and Instructions

For this lab you will be looking at the RISC-V instruction set covered in chapters 2-4 of your textbook.

You will need to complete the following questions and show all work when necessary. You will also have a coding portions. For the coding portions you are required to create a lab report outlining any equation and calculations you performed in order to code your solutions. You will also need to answer some questions regarding your solution and provide some screenshots in your report. Feel free to use this document as a foundation for your report or create your own. All solutions need to be typed, no hand written solutions will be accepted.

To submit you will need to upload a single PDF file lab report which will include the answers to the questions below as well as sections outlining your solutions for the coding portion. You will also need to provide a section that includes the code you wrote. Make sure that the code is properly formatted and runs properly in the RISC-V simulation we covered in class.

All submissions should be done through Canvas by the due date or will be subjected to the penalties outlined in the syllabus. We encourage collaboration, however, you must submit your own original work must be submitted and cheating will not be tolerated. Your solutions will need to follow strict adherence to the RISC-V coding style. This means that you your solutions should be case sensitive, if commenting, make sure you use // to represent the commented section. A new line will be associated with a new line of code and use of indentation is needed to separate label, instructions, and registers (both destination and source).

RISC-V Code Problem

(5pts) 1. You will need to write a program that will sort an array of the 21 positive integers. You can initialize this array first by manually storing the following numbers.
[5,6,8,4,2,11,0,4,74,89,11,4,2,5,25,12,8,14,4,9,13]
Note: you can sort ascending or descending but the result should be sorted array.

```
addi a0,x0,1024
addi t1,x0,5
sw t1,0(a0)

addi t1,x0,6
sw t1,4(a0)

addi t1,x0,8
sw t1,8(a0)

addi t1,x0,4
sw t1,12(a0)

addi t1,x0,2
sw t1,16(a0)

addi t1,x0,11
sw t1,20(a0)

addi t1,x0,0
sw t1,24(a0)

addi t1,x0,4
sw t1,28(a0)

addi t1,x0,74
sw t1,32(a0)

addi t1,x0,89
sw t1,36(a0)

addi t1,x0,11
sw t1,40(a0)

addi t1,x0,4
sw t1,44(a0)

addi t1,x0,2
sw t1,48(a0)

addi t1,x0,5
sw t1,52(a0)

addi t1,x0,25
sw t1,56(a0)

addi t1,x0,12
```

```
sw t1,60(a0)

addi t1,x0,8
sw t1,64(a0)

addi t1,x0,14
sw t1,68(a0)

addi t1,x0,4
sw t1,72(a0)

addi t1,x0,9
sw t1,76(a0)

addi t1,x0,13
sw t1,80(a0)

addi x9, x0, 1024 #start of array

addi x10, x0, 1108 #end of array

outerloop:

beq x9, x10, exit

add x7, x0, x9


innerloop:

lw x23, 0(x9)

beq x7, x10, incrementfirstloop

lw x24, 0(x7)

blt x24, x23, swap

beq x0, x0, continue

swap:

add x25, x0, x0

add x25, x0, x23

sw x24, 0(x9)

sw x25, 0(x7)

continue:

addi x7, x7, 4

beq x0, x0, innerloop

incrementfirstloop:

addi x9, x9, 4

beq x0, x0, outerloop
exit:
nop
```

(5pts) 2. You will need to write a program that will calculate the factorial of n integer. Note: the final result will be saved in the memory.

```
addi x22, x0,1024#adress to store value

addi t1,x0,1
addi t5,x0,1
addi a0, x0,6 #n to be changed
addi t0,a0,0

factorial:
mult t2,t0,t5
addi t5,t2,0
sub t3,t0,t1
addi t0, t3,0
bne t0,x0, factorial

sw t2,0(x22)
```