Electrical Engineering
& Computer Science
University of Missouri

ECE 4270: Computer Architecture, Fall 2021

LAB 2: MIPS Assembler

Scope

In this lab assignment, you will develop an assembler for MIPS ISA. It will take a program written in MIPS assembly and will convert into MIPS machine code. In the previous lab assignment, you have developed an instruction-level simulator for MIPS and test it out with some test programs provided with the assignment. For this assignment, you will use the output of your MIPS assembler as an input to MIPS simulator you have developed during the previous lab assignment.

Your MIPS assembler should accept an input file that contains the program written in MIPS assembly and should generate an output file that contains the corresponding MIPS machine code. The machine code will be similar to ones that you used in the first lab assignment (each line represents an instruction and encoded as 32-bit hexadecimal value). As an example, the following C code excerpt:

```
sum = 0;
for(i = 0; i < 10; i++){
sum += 1;
}
```

can be written in MIPS ISA as:

```
address instruction
-----------------------------------------------
[0x04000000] addiu $a0, $zero, 0 //set sum initially to 0
[0x04000004] addiu $a1, $zero, 0xA // upper bound for i is 10
[0x04000008] addiu $a2, $zero, 0 // i is initially 0
[0x0400000c] addiu $a0, $a0, 0x1 //update sum
[0x04000010] addiu $a2, $a2, 0x1 // increment i
[0x04000014] bne $a1, $a2, -8 // if branch taken, then jump to head of loop at 0x400000c
[0x04000018] addiu $v0, $zero, 0xA //set $v0 to 10 to exit
[0x0400001c] syscall
```

Of course, you can write a better code for the same C excerpt; however, for illustration purposes bear with this one.

Your MIPS assembler should convert the above MIPS instructions to the following machine code:

```
24040000
2405000a
24060000
24840001
24c60001
14a6fffe
2402000a
0000000c
```

After generating above instruction stream and saving it into an output file, you should be able to run MIPS simulator that you developed during the first lab assignment with this file.

Once you have a working MIPS assembler, you should write the MIPS assembly program for the following problems and convert them into MIPS machine code using your MIPS assembler and test them out using your instruction-level MIPS simulator.

Problem 1: You have an array that contains 10 integers, A = {5,3,6,8,9,1,4,7,2,10}. Use bubble sort algorithm to sort it in ascending order. In your MIPS program, you should store these values into the data segment of the memory, before trying to sort them out.

Problem 2: Write a MIPS program that finds the Fibonacci number of a given value. Calculate the Fibonacci number of 10 to test your MIPS program.

If you have to use an instruction that you haven't implemented in MIPS simulator already, you should also implement that instruction and specify it in your lab report.

Grading Rubric

Code: MIPS assembler, and solutions to the problems (75)
Report: 25 points

Code and MIPS programs (75 points):

In order to get a full credit for the code, your MIPS assembler should covert the MIPS programs you wrote for the given problems correctly, and MIPS simulator should run them properly.

Lab report (25 points):
Your report should give details about the work distribution within the group (who did what), milestones in your work and your implementation decisions (why did you choose the way you did it, and/or how did you do that).

If you added new instructions to MIPS simulator to support the operations you needed in solving the problems, please specify them in the report, as well.

Submission

You should submit the lab report along with the MIPS assembler code you developed (provide makefile, as well) and MIPS programs for the given problems. If you added and implemented new instructions to the MIPS simulator, then submit MIPS simulator code as well. One submission per group is required. Please, generate a pdf file of your report and name it lab2_report_groupX.pdf, then place it into the folder called lab2_groupX (where X is your group number). The folder lab2_groupX should also contain the src/ and input/ folders that contain your assembler code and MIPS programs you wrote for the problems, respectively. Then, please compress the lab2_groupX folder as lab2_groupX.tar.gz and submit it through the Canvas.

Due Date

Your lab is due on:
10/12 for Section I (before the lab)
10/14 for Section II (before the lab)