

Computer Architecture

EECE7352

Prof. Kaeli

Homework 2:

For part A of this homework, you will need to use the BRISC-V simulator. The simulator is located online at:

<https://ascslab.org/research/briscv/simulator/simulator.html>

For this lab we will only be using the 32-bit base RISC-V ISA (RV32I). This base ISA supports simple integer operations, including load, store, branch and jump instructions. Instructions for floating point and integer multiply may not be included in RV32I. Read the RISC-V specification on Canvas to see which instructions are available in the RV32I base instruction set. Note that the instruction sheet and RISC-V specification both include instructions for extensions, in addition to the base RV32I instructions. Do not use instructions from any extensions, they may not be supported by the simulator we are using.

There are two example programs on Canvas (the assembly code is already provided with the simulator). We provide source code (gcd.c and fibonacci.c) and assembly code (gcd.s and fibonacci.s) in order to show you example programs in RISC-V assembly. These programs also provide you with examples of the calling convention used by RISC-V. These programs are provided here for your reference.

Pay attention to three things in the assembly code:

- How is the stack pointer managed?
- How is the frame pointer (s0) managed?
- How are function arguments and return values passed?

To test your assembly code, load it in the BRISC-V simulator. A manual with directions for the simulator can be found here:

<https://ascslab.org/research/briscv/simulator/manual.html>

Part A: (30 points)

- 1) Write a RISC-V assembly program to return the larger of two integer values. The two input values should be initialized in main(). The main() function should call the larger(int x, int y) function, passing integer two values as arguments. The compare function should return the larger of the two numbers. If the values are equal, then return a value of zero.

- a. Write a RISC-V assembly program to implement and test the `larger(int x, int y)` function. Submit your assembly code on Canvas.
 - b. What is the largest number that `larger` can accept as input?
- 2) Write a recursive RISC-V assembly program to compute the factorial of a value that is initialized in `main()`. We provide a recursive factorial program in the `c` program example on Canvas.
 - a. Submit your assembly code on Canvas.
- 3) What is the largest integer value that you can compute the factorial in your program on the RV32I ISA? Explain why.

Part B: (30 points)

For this problem you will use the `qsort.c` (quicksort) program provided, and you need to produce a dynamic instruction mix table (similar to Figure A.29 in your textbook) to characterize the execution of the quicksort program. You can perform this study on any architecture of your choice. There are a number of approaches you can take to produce this data.

- 1) You could instrument the code to capture the execution frequency of each basic block, and then, using an assembly listing of the program, provide instruction counts (this is slightly imprecise, but very acceptable for this assignment).
- 2) You could find a tracing program that can capture an instruction trace. You would then have to write a program to count individual instructions (challenging, but not impossible).
- 3) You could find a tool out on the Internet that provides this capability already for you. While this sounds easy, it may be a bit of work to learn the particular tool you have chosen to use.

Please make sure to explain how you produced the data in your table and provide details of the tools that you used.

Part C. (15 points)

For this part of the assignment, write two different benchmark programs on your own that contain significant floating-point content. Compile the programs on X86 and generate an assembly listing of the benchmarks. Then identify 4 different floating-point instructions used in each program (a total of 8) and explain both the operands used by each instruction and the operation performed on the operands by the instruction.

Part D. (15 points)

For this problem you will need to read through Appendix K in your text, covering a number of instruction sets, and then answer the following questions:

- 1) Name 2 CISC instruction set architectures and 2 RISC instruction set architectures.
- 2) Describe 3 characteristics of the DEC Alpha instruction set.
- 3) Discuss the differences/similarities between MIPS and PowerPC in terms of how they handle conditional branches.
- 4) Provide an example of how register windows work on the SPARC ISA.
- 5) In your opinion, which generation of the Intel x86 architecture was the most significant advancement from the previous generation of the ISA.

Part E. (10 points)

Complete problem 1.16 at the end of Chapter 1 and Problems A.1, A.7 (only a, b and c) and A.12 from Appendix A in the text.

Part F. (20 points of Extra Credit to your quiz grade)

Read the Amdahl, Blaauw and Brooks 1964 paper on the IBM 360 Architecture. Given the timeframe of the paper, what do you find the most impressive feature of the architecture as described by the authors? Justify why you feel this is such a great feature. Also, discuss the representation of the various data types supported on this important ISA, and contrast it with the RISC-V.