

ECEC-355

Project 2: Single-cycle RISC-V Simulation

Instructor: Anup Das

TA: Shihao Song {shihao.song@drexel.edu}

1 Introduction

This project is intended to be a comprehensive introduction to single-cycle RISC-V simulation. You may work on this project in a team of up to two members. This project is due on **July 25, 2020**.

2 Required Reading

Chapter 4. The Processor, Sections 4.1 – 4.4.

3 Framework Overview

1. You will be working under directory *DREXEL-DISCO-RISC-V-Simulator-C-Impl/project_2_3_4_5*. To navigate to the directory:

```
$ cd DREXEL-DISCO-RISC-V-Simulator-C-Impl/project_2_3_4_5
```

2. To compile and run the simulator:

```
$ make  
$ ./RVSim ../cpu_traces/project_two
```

4 Review of Byte-addressable Data Memory

1. Assume we have a size-of-24-bytes data memory as shown in Figure 1. How does `int arr[3] = {19088743, 2882400001, 169552957}` get stored (assume `arr[0]` points to 0th location)? **Hints: Endianness.** Report the following in your report.

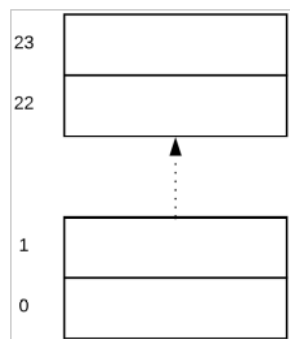


Figure 1: Endianness

- (a) How much space (in byte) does one integer occupy considering a RISC-V architecture?
 - (b) Draw the data mapping.
2. Continuing with 1, what is the value of x10 (assume x23 = 0)?

```
ld x8, 0(x23)
ld x9, 16(x23)
add x10, x8, x9
```

5 Extending the Framework

Extend *Core.h,c* to simulate the behavior of a single-cycle RISC-V CPU as shown in Figure 2. Test your simulator using *../cpu_traces/project_two* with the following configurations:

1. Set x25 to 4;
2. Set x10 to 4;
3. Set data memory from 0th location to `uint64_t arr[] = {16, 128, 8, 4}`
4. Set x22 to 1;

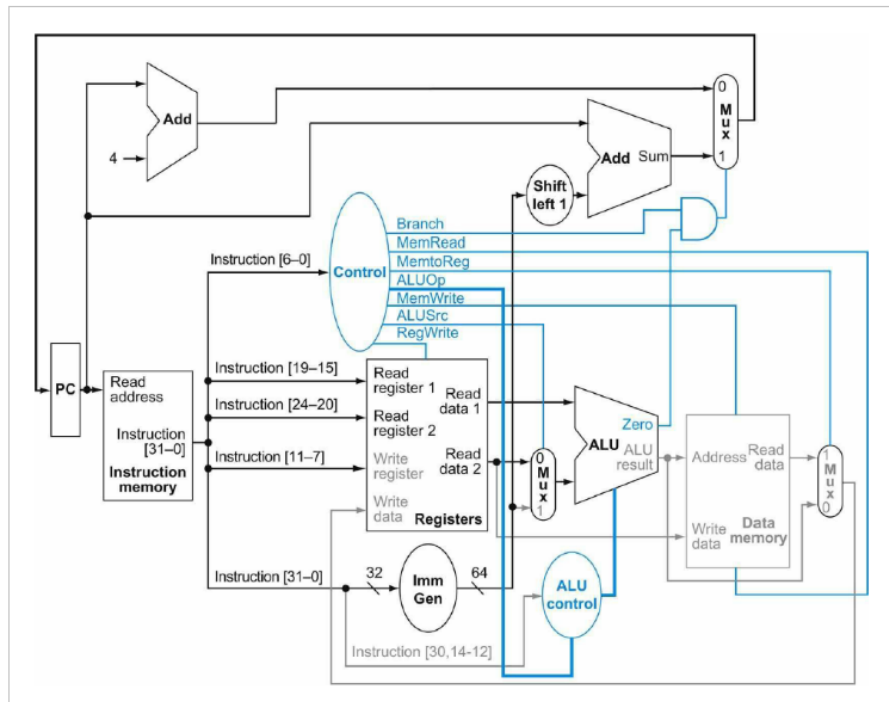


Figure 2: A Single-cycle RISC-V CPU

Report the following in your final report:

1. Value of x9 after completion of simulation.
2. Value of x11 after completion of simulation.

6 Submission

1. Summarize your experiment in Section 4 and 5. Compile your report in PDF format.
2. All the source codes.
3. Zip above and submit through Bblearn.