

CS311: Lab Report

Assignment 3 – Single Cycle Processor Simulation

B Siddharth Prabhu

200010003@iitdh.ac.in

Devdatt N

200010012@iitdh.ac.in

23 September 2022

1 Introduction

In this assignment, we simulated the working of a single cycle processor using Java. Each stage of processing (and the latches in between!) are coded as Java classes. As of now, one instruction occurs per cycle, i.e., one instruction must exit the processor before the next can enter. While this ensures correctness, it is at the cost of time and efficiency. We will work on improving this in the coming assignments, using the notion of *pipelining*.

2 Input and Output of the Program

Inputs include:

1. Full path to configuration file, `src/configuration/config.xml` .
2. Full path to statistics file, `stats.txt` , which stores statistics of the simulation run.
3. Full path to object file, for example `text_cases/descending.out` whose execution is to be simulated.

We run the program for a given object file (e.g. `descending.out`) using the following command-line argument(s), which have been put into a shell script for simplicity:

```
#!/bin/bash
ant;
ant make-jar;
java -jar jars/simulator.jar src/configuration/config.xml stats.txt
↪ test_cases/descending.out;
```

Output includes the statistics file which must be created at the required location. (`ant;`

and `ant make-jar;` do not have to be used in runs after the first simulation.) Pre-existing jar/bin folders may raise minor issues, which can be overcome by simply removing them (since they generate again anyway).

3 Scope for Future Improvement

We have tabulated number of cycles, number of instructions, CPI, and IPC for all given programs in the next section. We can improve on this using methods such as pipelining. We can't lose any correctness, so we'll still have to account for data/control hazards when doing that.

4 Tabulation of Observations

Object File	Number of Instructions	Number of Cycles	CPI	IPC
descending.out	365	365	1.0	1.0
evenorodd.out	6	6	1.0	1.0
fibonacci.out	94	94	1.0	1.0
prime.out	34	34	1.0	1.0
palindrome.out	56	56	1.0	1.0

Table 1: Observation Table