CS311: Lab Report
Assignment 4 - Pipelined Core Model Simulation

B Siddharth Prabhu Devdatt N
`200010003@iitdh.ac.in` `200010012@iitdh.ac.in`

07 October 2022

1   Introduction

In this assignment, we simulated the working of a pipelined processor model using Java. Each stage of processing (and the latches in between!) are coded as Java classes. The instructions queue into the processor, and if a conflict is detected, then the instructions in the previous latches are discarded. While this ensures correctness, it is at the cost of time and efficiency. We will work on improving this in the coming assignments.

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:

```bash
#!/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 are generated 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 software methods like code reordering/removal of dead code, or hardware methods like interlocks.

4   Tabulation of Observations

Object File	Number of Cycles	Number of OF stalls	Number of wrong branch instructions
`descending.out`	658	126	220
`evenorodd.out`	16	8	4
`fibonacci.out`	157	44	36
`prime.out`	75	15	28
`palindrome.out`	163	43	18

Table 1: Observation Table