#Yue Huang 1003839484 yuefrank.huang@mail.utoronto.ca
#Jiayi Zhao 1003936592 jiay.zhao@mail.utoronto.ca

Lab1 Report

1.After we run gcc.eio using sim-safe, we got CPI of 1.6642 for question 1 and CPI of 1.4079 for question 2.

Given CPI, the slowdown performance can be calculated as :

(New execution time- Old execution time)/Old execution time , where execution time = CPI * instruction number *cycle time.

In this case, instruction number and cycle time are same for new execution and old execution. Thus slowdown performance is calculated as;

(New CPI-Old CPI)/Old CPI

For question 1, the slowdown is (1.6642-1)/1=66.42%

For question 2, the slowdown is (1.4079-1)/1=40.79%

For the new CPI, we calculated by using

New CPI=New execute cycles/total number of instructions

We counted the total number of cycles to stall in sim-safe.

Thus, New CPI=((instruction number* old CPI)+ total number of cycles to stall)/instruction number. where old CPI is 1.0.

2.For question1 in section 3, there are two kinds of stalls, one-cycle stall and two-cycle stall. We design the micro-benchmarks respectively to validate the correctness.

To detect one cycle stall, we designed benchmark (i.e.)

```
addu $8,$10,$9
addu $4,$4,1
addu $9,$10,$8
```

To detect two cycle stall, we designed benchmark (i.e.)

```
addu $7,$5,$6
addu $6,$5,$7
```

we can put the instruction into a loop and iterate for 10^6 times to get the statistics.

We used flag -O1 to compile the microbenchmark.