CS311 (Computer Architecture Lab) **Assignment-04**

VIDIT JIGNESH PARIKH

April 6, 2025

1 Introduction

This assignment is about the simulated working of a discrete event model using Java. An event is a tuple in the following form: < `event_time`, `event_type`, `requesting_element`, `processing_element`, `payload` >.
The event queue is a list of events with a particular timestamp against them, that is, events ordered by time. And, an event is said to **fire** when the current clock cycle is equal to the event time. When this happens, the `handleEvent()` function of the processing element is invoked. Handling of an event may in turn lead to more events being generated, for the same clock cycle, or for some future clock cycle. Main Memory Latency is modeled, which is reflected in both - the IF and the MA stages.

2 Tabulation of Observations

Benchmark Object File	**No. Of Cycles**	**No. of Instructions**	**CPI**	**Throughput (IPC)**
`descending.out`	13196	329	40.10942	0.024931798
`evenordodd.out`	246	6	41.0	0.024390243
`fibonacci.out`	3464	86	40.279068	0.02482679
`prime.out`	1218	30	40.6	0.024630541
`palindrome.out`	2274	56	40.607143	0.024626208

Table 1: Observation Table for Assignment-05

3 Observations & Conclusions

The introduction of discrete event model helps us to model our processor to replicate a real-life processor even more, owing to the fact that for any real life scenario, the computations like ALU operations (mult, div, etc.) or memory access would not be infinitely fast, they would have a finite time lag which is significant especially when compared to the time for a cycle and hence introducing the same helps to predict and model the functionality of a real life processor, better!

- Main Memory Latency has been modeled. It is reflected in IF stage and during the load/ stores processes.
- Latencies of different functional units have been modeled. This is reflected in EX stage, and includes components such as ALU, multiplier, divider, etc.
- Main memory latency is observed to be roughly around 40% for our processor configuration. (The general latency could be anywhere between 20-50%, and hence our observations are in line with the general trends.
- If programs have more data hazards and control hazards, then it will take more cycles per instruction.