Computer Architecture Assignment 5

Srihari K G 210030035
Aryan Gulhane 210010006

October 17, 2023

1. **Discrete Event Simulator Model**

   Abstract: In this assignment, we simulate the 5 staged pipelined processor with discrete event simulator model. In this assignment we have modelled the latency involved in Instruction fetch, read and write involved with Memory and the arithmetic opertaions.It includes main memory latency of 40 cycles, Multiplier (4), Divider(10). And the simulation is done in the following order.

   while (not end of simulation)
   {
   performRW
   performMA
   performEX
   eventQueue.processEvents
   performOF
   performIF
   increment clock by 1

   }

   Following tables provide information on the number of cycles required for each benchmark program, number of instructions fetched in IF stage [1] and the throughput in terms of instructions per cycle with both latency included and not included in a 5 stage pipelined processor.

2. **Statistics for 5 staged pipelined processor without latency**

ToyRISC Program File	Number of Instructions fetched	Number of Cycles	Instruction Per Cycle
Prime.asm	36	65	0.55
Palindrome.asm	56	133	0.42
fibonacci.asm	88	158	0.55
evenorodd.asm	8	18	0.44
descending.asm	352	637	0.55

3. **Statistics for 5 staged pipelined processor with latency included**

ToyRISC Program File	Number of Instructions fetched	Number of Cycles	Instruction Per Cycle
Prime.asm	29	1485	0.019
Palindrome.asm	49	2885	0.016
fibonacci.asm	78	4165	0.018
evenorodd.asm	6	325	0.018
descending.asm	277	16085	0.017

[1] The Number of instructions fetched is the actual number of times the instruction was requested to memory including all the cases handled for data and control hazards

4. **Interpretation and comments** We see that the throughput in terms of Instructions per cylce (IPC) of the processor simulation with no latency is roughly 0.5 whereas the IPC of benchmark programs in the processor simulation with latency involved is roughly 0.019. We may note that IPC is drastically decreased because for each instruction read and for every load and store there are 40 cycles required. And we see more number of cycles running for descending code as it has more number of instructions.