COMPUTER ARCHITECTURE LAB 4

CS 311

Author
Saksham Chhimwal (210010046)
Shubh Agarwal (210020047)

Observations

Test Case	Static Inst.	Dynamic Inst.	No. of Cycles	Stalls	Wrong B.T.*
evenorodd.asm	10	6	245	2	0
fibonacci.asm	22	94	3175	12	16
palindrome.asm	17	56	2004	18	7
descending.asm	22	365	11463	69	88
prime.asm	17	34	1180	4	5

Table 1: Number of Instructions executed and stalls for various programs.

Test Case	IPC	Frequency
evenorodd.asm	0.008163265	1.0 GHz
fibonacci.asm	0.017007874	1.0 GHz
palindrome.asm	0.006487026	1.0 GHz
descending.asm	0.012125971	1.0 GHz
prime.asm	0.01779661	1.0 GHz

Table 2: IPC and Frequency

Static Instructions The number of instructions persent inside the program.

Dynamic Instructions The number of instructions executed by the program to produce the desired result.

No. of Cycle The number of clock cycles taken to complete the execution.

Stalls The number of stalls that were made during the program execution.

Wrong Branch Taken The number of times the program fetched wrong program code.

IPC Ratio of Instruction executed per cycle i.e. (Correct Instructions / Cycles).

Conclusion/Comments

When compared to report from the last time in which we did not need to incorporate the latency the number of dynamic instructions is less than the last time. A possible reason for this is that our processor is able to process the instructions before the new instruction enters the pipeline. The number of branch locks remain the same as this has no relation with how the instructions are fed in the pipeline.