Program	Number of cycles taken	Number of times an instruction on a wrong branch path entered the pipeline	Number of times the OF stage needed to stall because of a data hazard
descending.asm (Lab 4 Given Test Case)	459	88	91
evenorodd.asm (Lab 4 Given Test Case)	16	0	8
Lab 2 -> descending.asm	289	27	85
Lab 2 -> even-odd.asm	14	0	4
Lab 2 -> fibonacci.asm	128	9	38
Lab 2 -> palindrome.asm	32	4	9
Lab 2 -> prime.asm	26	1	12

Observations:

- The number of cycles taken with using pipeline is definitely less than the number of cycles taken without using pipeline. (As we can see from previous lab report)
- The more complex the program is, the more is the frequency that instruction enters a wrong path in pipeline and also, the more is the frequency of Data Hazard occurring due to Read After Write Hazard.
- The frequency of instructions entering wrong path also depends upon the input data values stored in register and memory and whether the branch conditions satisfy to true or not, satisfying to true leads to instruction being fetched on wrong data path.
- The more number of times branch condition satisfies to true, the more is number of times instruction entered wrong path in pipeline and we need to set that instruction to Nop and hence it will lead to wastage of that cycle and hence increasing the number of cycles.
- The more is the Read After Write operations from the same position, the more will be the Number of times OF stage needed to be stalled and it will therefore also increase the number of cycles because of stalling.
- But the total number of cycles taken will be still less than the case when we are not using pipelining.

Submitted By:

Mayank Singla