CS311 Computer Architecture Lab

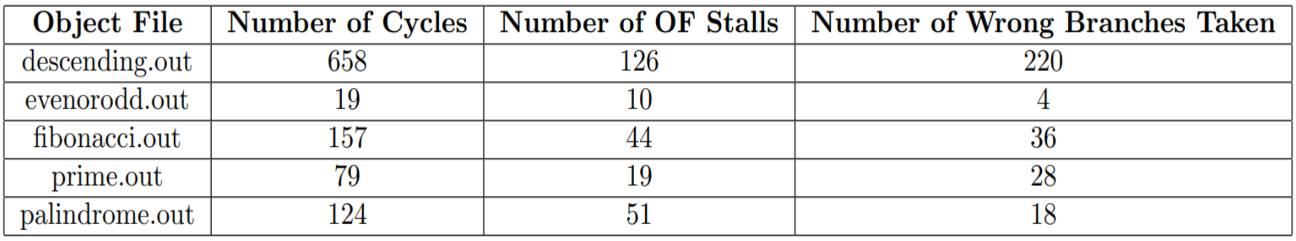
Assignment 4

Mukta Vedpathak(220110014)

Bidipta Saha(220150002)

- The assignment involves implementing a pipelined simulator for ToyRISC using Java, which involves accounting for various types of hazards like data hazards and control hazards using the proper interlocks. The pipelining results in fewer cycles required for execution of the object files than a non-pipelined models. However the fruitfulness of pipelining depends of the nature of the program and number of hazards.

- The results of execution of benchmark programs provided:



Object File	Number of Cycles	Number of OF Stalls	Number of Wrong Branches Taken
descending.out	658	126	220
evenorodd.out	19	10	4
fibonacci.out	157	44	36
prime.out	79	19	28
palindrome.out	124	51	18

- **Observations-** The pipelining leads to reduction in number of cycles required as compared to a non-pipelined model. In a scenario where there are no hazards, there would be no stalls or bubbles, which would lead to the number of cycles in pipelined model being slightly more than one-fifth the number of cycles required in the corresponding non-pipelined model. It would be "slightly more than one fifth" because we have five stages in our model. However branches,write operations,etc need to be addressed in a pipelined model and therefore higher the frequency of such problems, the more stalls and bubbles we have. For example, the descending.asm file has many branches and therefore highest number of wrong branches taken, which therefore leads to its execution needing highest number of cycles(658).