

# **ASSIGNMENT 4**

**PREKSHA PATANGE**

**MC23BT029**

## STATISTICS REPORT

Assembly program	No. of Instructions	Cycles	No. of OF stages stalled	No. of wrong branch Instructions
fibonacci	157	157	44	36
prime	79	79	19	28
palindrome	124	124	51	18
descending	658	658	126	220
evenorodd	19	19	10	4

Table 1: No. of Instructions, No. of Cycles, No. of OF stalls and No. of wrong branch Instructions of Different Programs

### Comment on the Observation

- Ideally, a 5-stage pipeline should achieve up to a  $5\times$  speedup, but this is reduced due to stalls. For example, in the Descending Sort benchmark, 220 wrong branches and 126 stalls significantly increase execution time, reducing the pipeline's effectiveness.
- Branch-intensive programs (like sorting and Fibonacci) suffer more from branch mispredictions than in a single-cycle processor. Single-cycle processors do not suffer from pipeline flushing, so their performance remains consistent.
- Even/Odd checking (19 cycles in both cases) is too simple to benefit significantly from pipelining. The instruction dependency in small programs makes the pipelining speedup negligible.
- The number of wrong branch instructions and the number of OF stage stalls are affected depending on the number of data hazards in the program.