ECE 486/586

COMPUTER ARCHITECTURE

SPRING 2024

PROJECT REPORT

ON

MIPS LITE 5 STAGE PIPELINE SIMULATOR

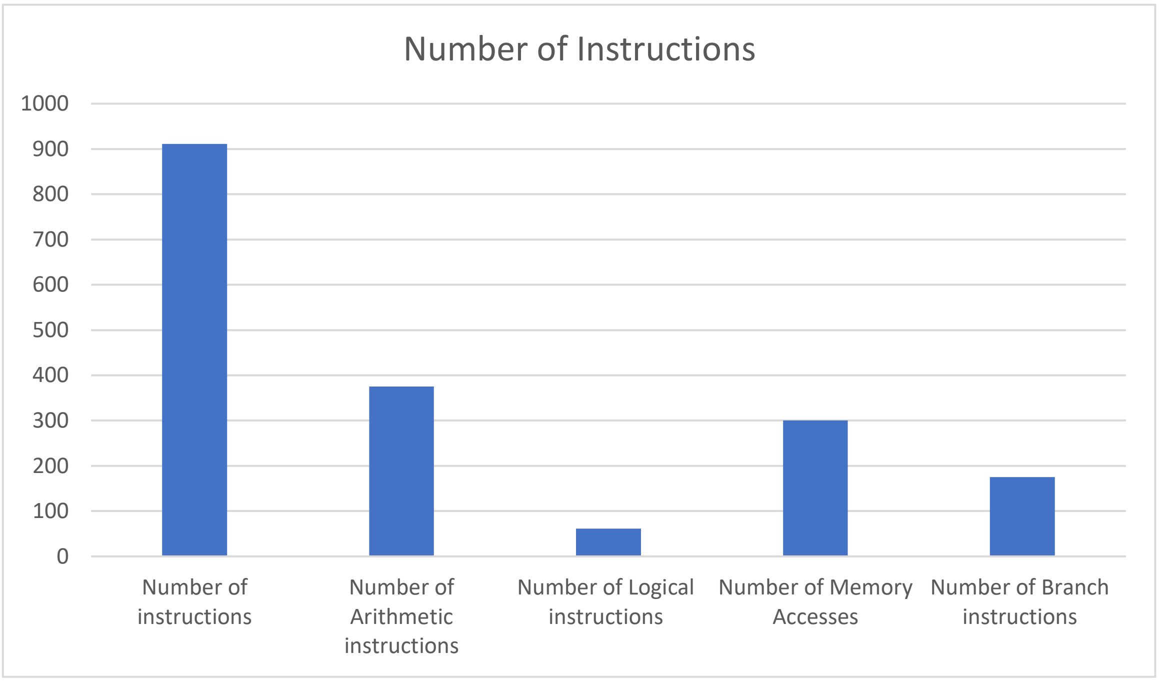
TEAM-8

Akhileswari Sirigineedi	**PSU ID: 941575537**	**akhi@pdx.edu**
Dhruva Teja Koppisetty	**PSU ID: 952138674**	**dhruva@pdx.edu**
Enoch Akanksh Vasimalla	**PSU ID: 926900483**	**enochv@pdx.edu**
Keerthanaa Bhoopathy	**PSU ID: 942229350**	**keerthb@pdx.edu**

1. Total number of instructions and a breakdown of instruction frequencies for the following instruction types: Arithmetic, Logical, Memory Access, Control Transfer.

   ANS.

Number of instructions	911
Number of Arithmetic instructions	375
Number of Logical instructions	61
Number of Memory Accesses	300
Number of Branch instructions	175



2. Final state of program counter, general purpose registers and memory (You only need to include the register and memory locations whose state has changed during the program execution)

ANS.

Final State General Purpose Register	Final State
R0	0
R11	1044
R12	1836
R13	2640
R14	25
R15	-188

R16	213
R17	29
R18	3440
R19	-1
R20	-2
R21	-1
R22	76
R23	3
R24	-1
R25	3

Final State Program Counter = 112

Final State Memory

Memory Accessed [2400] are: 2

Memory Accessed [2404] are: 4

Memory Accessed [2408] are: 6

Memory Accessed [2412] are: 8

Memory Accessed [2416] are: 10

Memory Accessed [2420] are: 12

Memory Accessed [2424] are: 14

Memory Accessed [2428] are: 16

Memory Accessed [2432] are: 18

Memory Accessed [2436] are: 29

Memory Accessed [2440] are: 22

Memory Accessed [2444] are: 24

Memory Accessed [2448] are: 26

Memory Accessed [2452] are: 28

Memory Accessed [2456] are: 30

Memory Accessed [2460] are: 32

Memory Accessed [2464] are: 34

Memory Accessed [2468] are: 36

```
Memory Accessed [2472] are:     38
Memory Accessed [2476] are:     59
Memory Accessed [2480] are:     42
Memory Accessed [2484] are:     44
Memory Accessed [2488] are:     46
Memory Accessed [2492] are:     48
Memory Accessed [2496] are:     50
Memory Accessed [2500] are:     52
Memory Accessed [2504] are:     54
Memory Accessed [2508] are:     56
Memory Accessed [2512] are:     58
Memory Accessed [2516] are:     89
Memory Accessed [2520] are:     62
Memory Accessed [2524] are:     64
Memory Accessed [2528] are:     66
Memory Accessed [2532] are:     68
Memory Accessed [2536] are:     70
Memory Accessed [2540] are:     72
Memory Accessed [2544] are:     74
Memory Accessed [2548] are:     76
Memory Accessed [2552] are:     78
Memory Accessed [2556] are:    119
Memory Accessed [2560] are:     82
Memory Accessed [2564] are:     84
Memory Accessed [2568] are:     86
Memory Accessed [2572] are:     88
Memory Accessed [2576] are:     90
Memory Accessed [2580] are:     92
Memory Accessed [2584] are:     94
Memory Accessed [2588] are:     96
Memory Accessed [2592] are:     98
Memory Accessed [2596] are:    149
Memory Accessed [2600] are:      2
Memory Accessed [2604] are:      4
Memory Accessed [2608] are:      6
Memory Accessed [2612] are:      8
Memory Accessed [2616] are:     10
Memory Accessed [2620] are:     12
Memory Accessed [2624] are:     14
Memory Accessed [2628] are:     16
Memory Accessed [2632] are:     18
Memory Accessed [2636] are:     29
```

3. Describe the stall conditions in both the “no forwarding” and “forwarding” cases and how long you stalled the pipeline for each stall condition (e.g., in the “no forwarding” case, if a consumer instruction comes right after a producer instruction, then the stall penalty is 2 cycles)

ANS.

Forwarding-Case:

- There will be no stalls and thus no stall penalty if a dependent instruction immediately follows its producer instruction.
- If a dependent instruction immediately follows a LOAD instruction, the stall penalty will be 1 cycle.
- When a branch instruction is taken or a jump register instruction is executed, the stall penalty will be 2 cycles.

No Forwarding-Case:

- The stall penalty is 2 cycles if a dependent instruction directly follows its producer instruction.
- The stall penalty is reduced to 1 cycle if there is one intermediate instruction between the producer and dependent instructions.
- There is no stall penalty (0 cycles) if two non-dependent instructions separate the producer and dependent instructions.
- A stall penalty of 2 cycles occurs when a taken branch instruction or a jump register instruction is executed.

4. In the case of “no forwarding”, the total number of data hazards and the average stall penalty per hazard

   Number of stall cycles: 554

   Number of Data Hazards: 307

   Average stall penalty per hazard = Total number of stalls/Total number of data hazards

   = 554/307 = 1.804

5. In the case of “forwarding”, the number of data hazards which could not be fully eliminated by forwarding.

   Number of Data Hazards: 60

6. Execution time in terms of number of clock cycles for the “no forwarding” and the “forwarding” scenarios.

   Number of clock cycles in non-forwarding: 1707
   Number of clock cycles during forwarding: 1213

7. Speedup achieved by “forwarding” as compared to “no forwarding”.

   Speedup achieved = (Execution time) no-forwarding / (Execution time) forwarding Speedup

   =1707/1213 = 1.407