ECE 586

COMPUTER ARCHITECTURE

Final Project Report on
MIPS-LITE 5-STAGE PIPELINE SIMULATOR

Submitted by:

Balaji Rao Vavintaparthi

Gagan Ganapathy Machiyanda Belliappa

Moksh Deepak Jadhav

Naveen Babu Vanamala

1. Instruction types: Arithmetic, Logical, Memory Access, Control Transfer
   The breakdown of instruction frequencies for these instruction types is listed below:

Total Number of instructions	911
Number of Arithmetic instructions	375
Number of Logical instructions	61
Number of Memory Access instructions	300
Number of Control Transfer instructions	175

2. Final states of program counter and general-purpose registers are listed below:

   Final state of program counter = 112

   Final states of register files are listed below:

General Purpose Register	**Final State**
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 Memory Values

Address: Contents	Address: Contents	Address: Contents
2400: 2	2492: 48	2584: 94
2404: 4	2496: 50	2588: 96
2408: 6	2500: 52	2592: 98
2412: 8	2504: 54	2596: 150
2416: 10	2508: 56	2600: 2
2420: 12	2512: 58	2604: 4
2424: 14	2516: 90	2608: 6
2428: 16	2520: 62	2612: 8
2432: 18	2524: 64	2616: 10
2436: 30	2528: 66	2620: 12
2440: 22	2532: 68	2624: 14
2444: 24	2536: 70	2628: 16
2448: 26	2540: 72	2632: 18
2452: 28	2544: 74	2636: 30
2456: 30	2548: 76	
2460: 32	2552: 78	
2464: 34	2556: 120	

2468: 36	2560: 82	
2472: 38	2564: 84	
2476: 60	2568: 86	
2480: 42	2572: 88	
2484: 44	2576: 90	
2488: 46	2580: 92	

Stall conditions in the cases of "no forwarding" and "forwarding" can be summarized as follows:

a. In MIPS architecture, Write After Read (WAR) and Read After Write (WAR) hazards are not present.

b. Pipeline stalling is only necessary to handle hazards related to Read After Write (RAW) dependencies.

c. The utilization of forwarding techniques aids in minimizing stalls and enhancing overall execution time

1. Scenario 1: No Forwarding

- When there are two or more instructions between the producer and the dependent consumer instruction, no stalls are needed as the consumer instruction will have its operands available when required.
- When there is only one instruction gap between the producer and the dependent consumer instruction, a one-cycle stall is introduced to ensure that the necessary information is ready in the memory or register file when needed.
- When the dependent consumer instruction immediately follows the producer instruction, a two-cycle stall is introduced to ensure the required contents are available when needed.

2. Scenario 2: Forwarding

- When there is a gap of two or more instructions between the producer and the dependent consumer instruction, no stalls are required as the consumer instruction can access its operands from the memory or register file without any delays.
- When there is a gap of one instruction between the producer and the dependent consumer instruction, no stalls are introduced because forwarding allows the dependent consumer instruction to directly receive the required data from the producer instruction.
- When the dependent consumer instruction immediately follows the producer instruction, only a one-cycle stall is needed to ensure the availability of the required contents before the consumer instruction needs them.

When the dependent consumer instruction immediately follows the producer instruction, there is no requirement for a two-cycle stall. By employing the forwarding technique, the necessary data can be directly provided from the producer instruction to the dependent consumer instruction. This eliminates the

need to wait for the data to be stored in memory or the register file. As a result, only a one-cycle stall is necessary to ensure that the required contents are available before the consumer instruction requires them.

In Non-Forwarding case

a. Total Number of stalls – 557
b. Total Number of data hazards – 309

So average Stall penalty = 557/ 309 = 1.802589

In the case of forwarding the no data hazards which could not be fully eliminated by forwarding

Total number of data hazards = 174

Total clock cycles in

Non-Forwarding – 1782

Forwarding – 1089

Utilizing the forwarding technique in the simulator execution leads to a reduction in the number of clock cycles compared to the scenario without forwarding. This reduction in clock cycles indicates that the forwarding technique effectively minimizes stalls and delays in the pipeline, thereby reducing the overall execution time.

Now using Amdahl's law

$$\text{Speedup} = \frac{\textit{Execution Time (Non-Forwarding)}}{\textit{Execution Time (Forwarding)}}$$

Speedup = 1782/ 1089 = 1.636