COMPUTER ARCHITECTURE

(ECE586)

Project Report SPRING-2023

Team 40

Somu Jayanth Kumar Reddy (927148993)

Sai Bhargav Reddy Gujjula (959909816)

Bhuvan Yadav Nagulla (965201149)

SaiNikhil Reddy Lokasani (920732545)

OBJECTIVE

To develop an execution-driven MIPS lite pipeline simulator. The simulator will take the provided memory image as its input. It will implement two key features:

1. A functional simulator which simulates the MIPS lite ISA and captures the impact of instruction execution on machine state.

2. A timing simulator which models the timing details for the 5 stage pipeline. The output of the simulator will include:

(a) A breakdown of instruction frequencies in the instruction trace into different instruction types,

(b) Final machine state (register values, memory contents etc.)

(c) Execution time (in cycles) of the instruction trace on the 5 stage pipeline.

IMPLEMENTATION

We implemented MIPS-lite design using Python language.

SIMULATION RESULTS

Instruction counts:

****

Total number of instructions: 1286

Arithmetic instructions: 527

Logical instructions: 86

Memory Access instructions: 425

Control Transfer instructions: 248

****

****

Final register states->

Program counter: 112

R 11 : 1144

R 12 : 1936

R 13 : 2740

R 14 : 170

R 15 : -124

R 16 : 294

R 17 : 70

R 18 : 3540

R 19 : -1

R 20 : -2

R 21 : -2

R 22 : 76

R 23 : 3

R 24 : -1

R 25 : 106

Stalls without forwarding: 783

Stalls with forwarding: 85

Timing simulator output without forwarding:

Total number of clock cycles: 2411

Program Halted

Timing simulator output with forwarding:

Total no. of cycles(with forwarding): 1713

Program Halted

Stalls without forwarding: 783

average cycle stall for hazards: 1.804147465437788

Single stalls: 85

Double stalls: 349

No. of RAW hazards: 434

Penalties because of branches: 338

No. of branches leading to penalties: 169

Average Branch Penality: 2.0 cycles

Stalls with forwarding: 85

Total no. of cycles(without forwarding): 2411

Total no. of cycles(with forwarding): 1713

Speedup acheive by forwarding: 1.407472270869819

Stall conditions for forwarding and no forwarding cases:

Forwarding case

- There will be no delays if the dependent instructioncomes soon after the producer instruction. As a result, the stall penalty is zero (0).
- The stall penalty is one (1) if the dependent instruction comes soon after the producer instruction and the producer instruction is a LOAD instruction.
- The stall penalty is 2 if the instruction is a branch and istaken, or if the instruction jump register is executed.

No Forwarding Case

- The stall penalty is two cycles if the dependent instruction comes shortly after the producer instruction.
- The stall penalty is 1 cycle if an intermediate instruction separates the producer and dependent instructions.
- If there are two instructions between the producer and the dependent instruction and there is no dependence between them, the stall penalty is 0.
- The stall penalty is 2 if the instruction is a branch and is taken, or if the instruction jump register is executed.

No Forwarding:

The total number of data hazards and the average stall penalty per hazard:

The total number of Data Hazards: **434**

Single stalls: **85**

Double stalls: **349**

The total number of stalls: **783**

Average stall penalty per hazard = Total number of stalls/Total number of data hazards = 783/434 = **1.804147465437788**.

Branch penalties because of branch instructions: **338**

Number of branches leading to branch penalties: **169**

Average Branch penalty: **2.0 cycles**

Forwarding:

The number of data hazards which could not be fully eliminated by forwarding

The total number of stalls: **85**

- Execution time in terms of number of clock cycles for the no forwarding and the forwarding scenarios:

The number of clock cycles for no forwarding: **2411**

The number of clock cycles for forwarding: **1713**

- Speedup achieved by forwarding as compared to no forwarding:

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

Speedup = 2411/1713 = **1.407472270869819**.