Desmond Oketch- SCT212-0083/2021

COMPUTER TECHNOLOGY

Lab 2

Computer Architecture – tutorial 2

Loop Overview

```
loop:  LD    R1, 0(R2)
       DADDI R1, R1, 1
       SD    0(R2), R1
       DADDI R2, R2, 4
       DSUB  R4, R3, R2
       BNEZ  R4, loop
```

- **Initial Condition**: R3 = R2 + 396
- **Loop Iterations**: Since each iteration increments R2 by 4, and R3 = R2 + 396, the loop runs for 99 iterations (396 / 4 = 99).

a. No Forwarding, Branch Resolved in ID, Pipeline Flushing

Assumptions:

- Pipeline Stages: IF, ID, EX, MEM, WB
- No Forwarding: Data hazards cause stalls
- Branch Handling: Resolved in ID stage; mispredicted branches cause pipeline flushes
- Memory Access: 1 cycle

Data Hazards and Stalls:

1. LD R1, 0(R2) → DADDI R1, R1, 1: RAW hazard; DADDI needs R1 after it's written back → 2 stalls

2. DADDI R1, R1, 1 → SD 0(R2), R1: RAW hazard; SD needs R1 after it's written back → 2 stalls
3. DADDI R2, R2, 4 → DSUB R4, R3, R2: RAW hazard; DSUB needs updated R2 → 2 stalls
4. DSUB R4, R3, R2 → BNEZ R4, loop: RAW hazard; BNEZ needs R4 after it's written back → 2 stalls

Pipeline Timing per Iteration:

- Instruction Cycles: 6 instructions × 5 stages = 30 cycles
- Stalls: 4 hazards × 2 stalls = 8 cycles
- Branch Penalty: Assuming 2-cycle penalty due to flushing

Total per Iteration: 30 (instructions) + 8 (stalls) + 2 (branch penalty) = 40 cycles

Total for 99 Iterations: 99 × 40 = 3960 cycles

b.With Forwarding, Predict Branch as Not Taken

Assumptions:

- Forwarding: Data hazards are mitigated
- Branch Prediction: Predict not taken; mispredicted branches cause flushes
- Branch Resolution: In ID stage

Data Hazards and Stalls:

- LD followed by DADDI: Load-use hazard; even with forwarding, need 1 stall
- Other instructions: Hazards resolved via forwarding; no stalls

Pipeline Timing per Iteration:

- Instruction Cycles: 6 instructions × 1 cycle = 6 cycles
- Stalls: 1 cycle (load-use hazard)
- Branch Penalty: Assuming 2-cycle penalty for misprediction

Total per Iteration: 6 (instructions) + 1 (stall) + 2 (branch penalty) = 9 cycles

Total for 99 Iterations: 99 × 9 = 891 cycles

c.Delayed Branch with Forwarding

Assumptions:

- Delayed Branch: The instruction following the branch is always executed
- Forwarding: Data hazards are mitigated
- Branch Resolution: In ID stage

Instruction Scheduling:

To utilize the branch delay slot effectively, we can reorder instructions:

```
loop:  LD    R1, 0(R2)
       DADDI R1, R1, 1
       SD    0(R2), R1
       DADDI R2, R2, 4
       DSUB  R4, R3, R2
       DADDI R5, R5, 0   ; NOP or useful instruction
       BNEZ  R4, loop
```

Here, DADDI R5, R5, 0 is a placeholder for the delay slot. Ideally, we place a useful instruction here to avoid wasting a cycle.

Pipeline Timing per Iteration:

- None, as the delay slot is utilized
- Instruction Cycles: 6 instructions × 1 cycle = 6 cycles
- Stalls: 1 cycle (load-use hazard between LD and DADDI)

Branch Penalty: Total per Iteration: 6 (instructions) + 1 (stall) = 7 cycles

Total for 99 Iterations: 99 × 7 = 693 cycles