National University of Computer and Emerging Sciences, Lahore Campus

| | | | |
|---|---|---|---|
| Course: | Advance Computer Architecture | Course Code: | EE502 |
| Program: | MS(Computer Science) | Semester: | Spring 2017 |
| Duration | 75 mins | Total Marks: | 50 |
| Date: | 14-03-2017 | Weight | 20 |
| Section: | A | Page(s): | 2 |
| Exam: | Mid | | |

**Question1: True/False** **Marks 5**

1. Pipelining increases the latency of a single instruction.
2. Machine cycle is determined by the time required for the fastest pipeline stage.
3. A single memory port is accessed for instruction fetch and data read in the same clock cycle would be a data hazard.
4. Ideal Speedup from pipelining is equal to the number of pipeline stages.
5. Pipelined implementation increases the clock frequency of the processor over single-cycle implementation.

**Question2:** **Marks 5**

What is the difference between static and dynamic branch prediction? Explain at least one technique from both types of predictions.

**Question3:** **Marks 15**

Consider the following code sequence

```
lw R1,20(R3)
mul R3,R4,R1
add R1, R1, 4
sw R3, 20(R1)
```

Consider the following clock cycle times for a 5-stage pipelined datapath

| Without Forwarding | With ALU-ALU forwarding only | With Full Forwarding (ALU-ALU and MEM-ALU) |
|---|---|---|
| 400ps | 500ps | 600ps |

**a.** What is the total execution time of the above code sequence when pipeline without forwarding is used? [5]

Answer: _____

**Rough work:**

b. What is the total execution time of the above code sequence when pipeline with ALU-ALU forwarding is used? [5]

Answer: _____

**Rough work:**

c. What is the total execution time of the above code sequence when pipeline with full forwarding is used? [5]
Answer: _____
**Rough work:**

**Question4:** **Marks 10**

The table below describes the performance of two processors, the rAlpha and the c86 with a compiler for a common program.

| | GHz | Compiler | |
|---|---|---|---|
| | | Instructions | Average CPI |
| rAlpha | 3.4 | 7000 | 1.2 |
| c86 | 2.6 | 1500 | 2.2 |

Which is the best compiler-machine combination that provides better performance?

**Question 5:** **Marks 15**

Consider the following MIPS assembly code to be executed on a pipelined CPU with frequency of 1 GHz.

```
Loop: ld R2, 100 (R4)
      addi R2, R2, 4
      ld R3, 200 (R4)
      addi R3, R3, 4
      add R5, R2, R3
      sw R5, 300 (R4)
      addi R4, R4, 4
      beq R4, R6, Loop
```

Your task is to do dynamic scheduling based on Tomasulo algorithms. There are:

- 2 LOAD/STORE Functional Units (LDU1, LDU2) with latency of 2 cycles
- 2 ALU/BR Function Units (ALU1, ALU2) with latency 1 cycle
- 2 RESERVATION STATIONS (RS1, RS2) for ALU/BR operation
- 2 Buffers for LOAD/STORE (LDST1, LDST2)

Moreover you have to identify:

- Structural hazards for RS in ISSUE phase
- RAW hazards and Structural hazards for FUs in START EXECUTE phase

Assume static branch prediction (backward taken) is used.

a) Show a single iteration of the loop using tomasulo
b) If the clock frequency is 1GHz, what would be average CPI.

*Instruction status:*

| Instruction | Issue | Exec Start | Exec Comp | Write Result |
|---|---|---|---|---|
| Loop: ld R2, 100 (R4) | | | | |
| addi R2, R2, 4 | | | | |
| ld R3, 200 (R4) | | | | |
| addi R3, R3, 4 | | | | |
| add R5, R2, R3 | | | | |
| sw R5, 300 (R4) | | | | |
| addi R4, R4, 4 | | | | |
| beq R4, R6, Loop | | | | |

## Reservation Stations:

| Time | Name | Busy | Op | S1 Vj | S2 Vk | RS Qj | RS Qk |
|---|---|---|---|---|---|---|---|
| | RS1 | No | | | | | |
| | RS2 | No | | | | | |

| | Busy | Address |
|---|---|---|
| LDST1 | No | |
| LDST2 | No | |

## Register result status:

| R0 | R1 | R3 | R4 | R5 | R6 | R7 | ... |
|---|---|---|---|---|---|---|---|
| | | | | | | | |