

1. Problems in this exercise refer to the following sequence of instructions:

```
LW    $5, -16($5)
SW    $5, -16($5)
ADD   $5, $5, $5
```

(a) **Question** Indicate dependences and their type.

**Answer**

```
RAW: I1 -> I2 ($5), I1 -> I3 ($5)
WAR: I2 -> I3 ($5), I1 -> I3 ($5)
WAW: I1 -> I3 ($5)
```

(b) **Question** Assume there is not forwarding in this pipeline processor, indicate hazards and add NOP instructions to eliminate them.

**Answer**

With no forwarding, the only hazards are the RAW.

```
∴ LW    $5, -16($5)
    NOP
    NOP
    SW    $5, -16($5)
    ADD   $5, $5, $5
```

(c) **Question** Assume there is full forwarding, indicate hazards and add NOP instructions to eliminate unresolved cases.

**Answer**

The only hazard with full forwarding is the RAW:

```
∴ LW    $5, -16($5)
    NOP
    SW    $5, -16($5)
    ADD   $5, $5, $5
```

The remaining problem in this exercise assumes the following clock cycle times:

Without forwarding: 200 ps

With full forwarding: 250 ps

With ALU-ALU forwarding only: 220 ps

(d) **Question** What is the total execution time of this instruction sequence without forwarding and with full forwarding? What is the speed-up achieved by adding full forwarding to a pipeline that had no forwarding?

**Answer**

With full forwarding, the instructions need 4 cycles. With no forwarding, the instructions need 5 cycles.

$$\therefore \frac{200 \cdot 5}{250 \cdot 4} = 1$$

- (e) **Question** What is the total execution time of this instruction sequence with only ALU-ALU forwarding? What is the speed-up over a no-forwarding pipeline?

**Answer**

Total time:  $220 \cdot 5 = 1100$  ps

Speed up:  $\frac{1000}{1100} = 0.91$

For this problem, assume that all branches are resolved in ID stage and are perfectly predicted (this eliminates all control hazards). For the following fragment of MIPS code:

```
LW $5, -16($5)
SW $4, -16($4)
LW $3, -20($4)
BEQ $2, $0, Label ; assume $2 ≠ $0
ADD $1, $5, $4
Label:
SUB $2, $1, $3
```

If we only have one memory (for both instruction and data), there is a structural hazard every time we need to fetch an instruction in the same cycle in which another instruction accesses data. To guarantee forward progress, this hazard must always be resolved in favor of the instruction that accesses data.

- (a) **Question** What is the total execution time of this instruction sequence in the 5-stage pipeline that only has one memory?

**Answer**

- (b) **Question** How can the structural hazard be eliminated by adding NOP to the code? (Please show a modified version of the program with the added NOP instructions)

**Answer**

2. (a) **Question** We wish to add the instruction "ADDI" (Add immediate) and "LUI" (load upper immediate) to the shown single-cycle simple processor. Add any datapath and control signals and show the value of the control signals while executing the new "ADDI" and "LUI" instruction.