## Computer Architecture and Engineering

# CS152 Quiz #4 Solutions

## Problem Q4.1: Out-of-Order Scheduling

### Problem Q4.1.A

| | Time | | | | OP | Dest | Src1 | Src2 |
|---|---|---|---|---|---|---|---|---|
| | Decode → ROB | Issued | WB | Committed | | | | |
| $I_1$ | -1 | 0 | 1 | 2 | L.D | T0 | R2 | - |
| $I_2$ | 0 | 2 | 12 | 13 | MUL.D | T1 | T0 | F0 |
| $I_3$ | 1 | **13** | **15** | **16** | ADD.D | **T2** | **T1** | **F0** |
| $I_4$ | **2** | **3** | **5** | **17** | ADDI | **T3** | **R2** | **-** |
| $I_5$ | **3** | **4** | **6** | **18** | L.D | **T4** | **T3** | **-** |
| $I_6$ | **4** | **7** | **17** | **19** | MUL.D | **T5** | **T4** | **T4** |
| $I_7$ | **5** | **18** | **20** | **21** | ADD.D | **T6** | **T5** | **T2** |

**Table Q4.1-1**

Common mistakes: Forgetting in-order commit, integer result bypassing, single-ported ROB/register files, issue dependent on writeback of sources and completed decode

### Problem Q4.1.B

| | Time | | | | OP | Dest | Src1 | Src2 |
|---|---|---|---|---|---|---|---|---|
| | Decode → ROB | Issued | WB | Committed | | | | |
| $I_1$ | -1 | 0 | 1 | 2 | L.D | T0 | R2 | - |
| $I_2$ | 0 | 2 | 12 | 13 | MUL.D | T1 | T0 | F0 |
| $I_3$ | 3 | **13** | **15** | **16** | ADD.D | **T0** | **T1** | **F0** |
| $I_4$ | **14** | **15** | **17** | **18** | ADDI | **T1** | **R2** | **-** |
| $I_5$ | **17** | **18** | **19** | **20** | L.D | **T0** | **T1** | **-** |
| $I_6$ | **19** | **20** | **30** | **31** | MUL.D | **T1** | **T0** | **T0** |
| $I_7$ | **21** | **31** | **33** | **34** | ADD.D | **T0** | **T1** | **F3** |

**Table Q4.1-2**

Common mistakes: Forgetting in-order commit, not reusing T0 and T1 appropriately

## Problem Q4.2: Fetch Pipelines

| | |
|---|---|
| PC | PC Generation |
| F1 | ICache Access |
| F2 | |
| D1 | Instruction Decode |
| D2 | |
| RN | Rename/Reorder |
| RF | Register File Read |
| EX | Integer Execute |

### Problem Q4.2.A — Pipelining Subroutine Returns

Immediately after what pipeline stage does the processor know that it is executing a subroutine return instruction?

D2

Immediately after what pipeline stage does the processor know the subroutine return address?

RF

How many pipeline bubbles are required when executing a subroutine return?

6

### Problem Q4.2.B — Adding a BTB

A subroutine can be called from many different locations and thus a single subroutine return can return to different locations. A BTB holds only the address of the last caller.

### Problem Q4.2.C — Adding a Return Stack

Normally, instruction fetch needs to wait until the return instruction finishes the RF stage before the return address is known. With the return stack, as soon as the return instruction is decoded in D2, instruction fetch can begin fetching from the return address. This saves 2 cycles.

A return address is pushed after a JAL/JALR instruction is decoded in D2. A return address is popped after a JR r31 instruction is decoded in D2.