Q1 Pipelining RISC-V

2 Points

You can find the lecture slides for todays lecture here!

You can access the YouTube playlist here!

Q1.1 Which of the following are true?

2 Points

Because we break up the pipeline into 5 stages, we can achieve a 5x speedup.

- ○ True
- ◉ False

The time to complete one instruction is shorter in pipelined designs.

- ○ True
- ◉ False

Q2 Pipeline Hazards

1 Point

Q2.1
1 Point

I swear that I have watched this video and the previous video and that if I have not and say that I have watched it, I will receive an F in the class.

- (●) I have watched the video.
- ( ) I have not watched the video.

Q3 Pipelining Datapath
3 Points

Q3.1
3 Points

Should we absorb the +4 unit of the IF stage into the ALU? Why or why not?

- ○ Yes we should, because this will allow us to use less hardware and therefore a cheaper datapath which is worth the lowering of performance.
- ◉ No we should not, because this will incur a structural hazard between the IF and EX stages, and the stall necessary to resolve this is worse than less hardware.

If a register file cannot be written and read in a single cycle, that would cause a structural hazard affecting (choose one):

- ○ IF and WB stages
- ◉ ID and WB
- ○ EX and WB
- ○ ID and MA
- ○ MA and WB

Q4 Data Hazards

10 Points

Q4.1

5 Points

Which stages are involved in a data hazard?

- [ ] IF
- [x] ID
- [ ] EX
- [ ] MEM
- [x] WB

What do we mean when we say to insert a "bubble" into the pipeline?

- [x] Stall the pipeline for a cycle
- [x] Execute a `nop` psuedoinstruction instead of the next instruction
- [ ] Pour soapy water onto your motherboard, watch it foam up, and blow on it until a bubble forms

For which types of instructions is the result that is to be written back to `rd` ready at the end of the `EX` stage?

- [x] R
- [ ] I
- [ ] S
- [ ] B
- [x] U
- [ ] J

Q4.2

5 Points

Assuming we cannot read/write ('double pump') register file nor forward, how many stalls do we need to account for a data hazard caused by an `add` instruction?

- ○ 0
- ○ 1
- ○ 2
- ◉ 3
- ○ 4

Assuming we can double-pump but cannot forward, how many stalls do we need to account for a data hazard caused by an `add` instruction?

- ○ 0
- ○ 1
- ◉ 2
- ○ 3
- ○ 4

Assuming we can double-pump and forward, how many stalls do we need to account for a data hazard caused by an `add` instruction?

- ◉ 0
- ○ 1
- ○ 2
- ○ 3
- ○ 4

Does the following code need to stall without forwarding:

```
lw  t0,0(t0)
add t1, t0, t0
```

- ◉ yes
- ○ no

Does the following code need to stall without forwarding:

```
addi t1, t0, 1
addi t3, t0, 2
addi t3, t0, 4
addi t5, t1, 5
```

- (●) yes
- ( ) no

Lecture 22 - Pipelining II

GRADED

STUDENT
Zachary Zhu

TOTAL POINTS
16 / 16 pts

QUESTION 1
Pipelining RISC-V — **2** / 2 pts

1.1 Which of the following are true? — **2** / 2 pts

QUESTION 2
Pipeline Hazards — **1** / 1 pt

2.1 (no title) — **1** / 1 pt

QUESTION 3
Pipelining Datapath — **3** / 3 pts

3.1 (no title) — **3** / 3 pts

QUESTION 4
Data Hazards — **10** / 10 pts

4.1 (no title) — **5** / 5 pts

4.2 (no title) — **5** / 5 pts