Computer Organization(H)
Southern University of Science and Technology
Mengxuan Wu
12212006

Theory Assignment 3

Mengxuan Wu

Problem 1

a)

The single cycle time for pipelined processor is the longest time of all stages, which is 350ps.

The single cycle time for non-pipelined processor is the sum of all stages, which is 1250ps.

b)

For the pipelined processor, the total latency will be 350ps×5 = 1750ps.

For the non-pipelined processor, the total latency will be 1250ps.

c)

I will split the longest stage into two stages, and each stage will take 175ps. The new clock cycle time will be the longest time of remaining stages, which is 300ps.

d)

The `lw` and `sw` commands will utilize memory. Hence, the memory utilization will be $20\% + 15\% = 35\%$.

e)

The `lw` and `ALU/Logic` commands will utilize the write-register port. Hence, the write-register port utilization will be $45\% + 20\% = 65\%$.

f)

For multi-cycle organization, the time cost for each instruction will be:

Instruction	Stage	Time
ALU/Logic	IF, ID, EX, WB	1400ps
Branch	IF, ID, EX	1050ps
Load	IF, ID, EX, MEM, WB	1750ps
Store	IF, ID, EX, MEM	1400ps

The execution time for multi-cycle organization will be the weighted sum of each instruction's time cost:

$$\begin{aligned}\text{Execution Time} &= 0.45 \times 1400 + 0.2 \times 1050 + 0.2 \times 1750 + 0.15 \times 1400 \\ &= 1400\text{ps}\end{aligned}$$

The overall comparison is shown in the table below:

Organization	Clock Cycle Time	Execution Time
Single-cycle	1250ps	1250ps
Multi-cycle	350ps	1400ps
Pipelined	350ps	1750ps

Problem 2

a)

$$\begin{aligned}\text{Execution Time} &\approx 250\text{ps} \times 5n &&= 1250n\text{ps} \\ \text{Execution Time}' &\approx 300\text{ps} \times 1.05n &&= 315n\text{ps}\end{aligned}$$

The speedup ratio will be:

$$\begin{aligned}\text{Speedup} &= \frac{1250n}{315n} \\ &\approx 3.968\end{aligned}$$

b)

Let the ratio of `NOP` in the forwarding pipelined processor be k. If the forwarding pipelined processor is faster than the non-forwarding pipelined processor, we have:

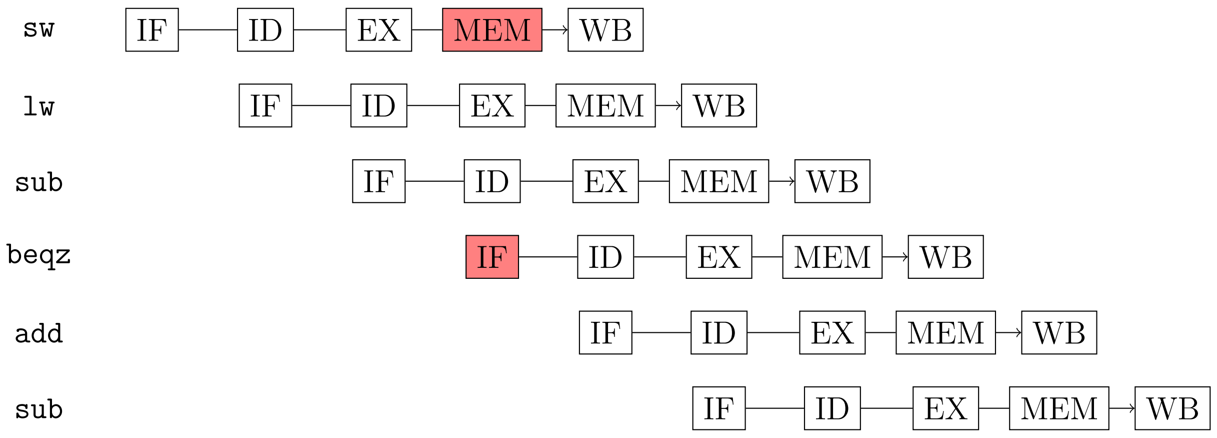
$$\begin{aligned}250\text{ps} \times 5n &> 300\text{ps} \times (1+k) \times n \\ 950n\text{ps} &> 300kn\text{ps} \\ 3.167 &> k\end{aligned}$$

Hence, the ratio of `NOP` in the forwarding pipelined processor at most is 3.167 if it is faster than the non-forwarding pipelined processor.

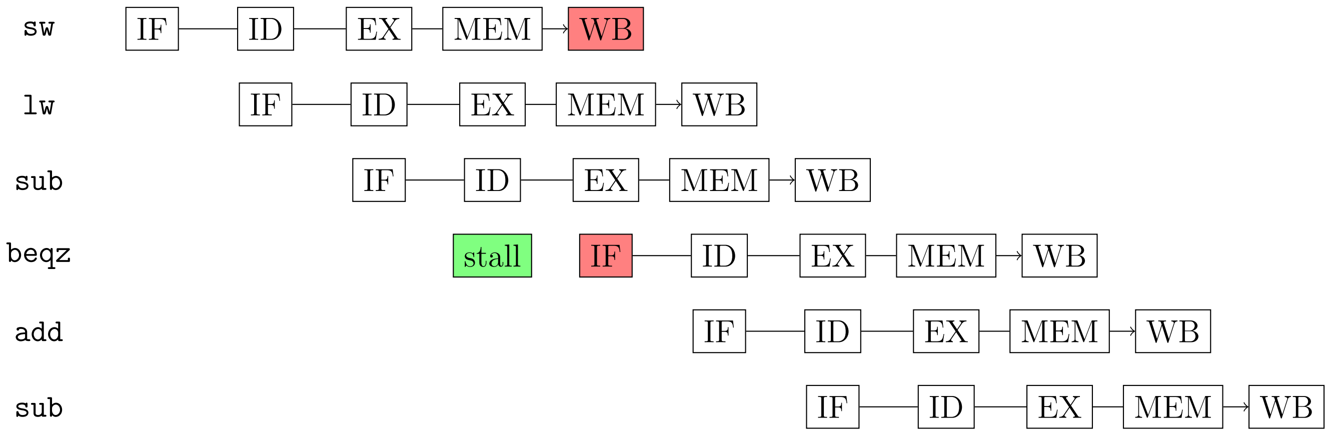
Problem 3

a)

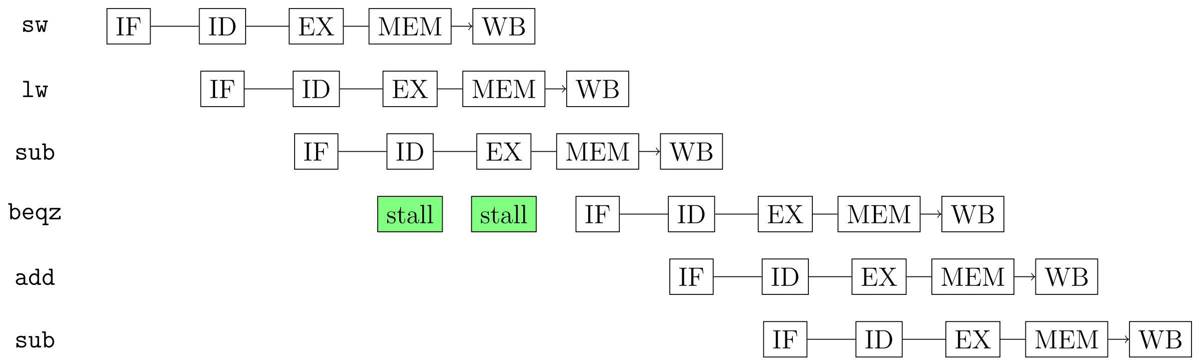
The first hazard is the data hazard between the `lw` and `beqz` instructions.



Hence, we stall the `beqz` instruction for one cycle. But the hazard still exists.



Hence, we stall the `beqz` instruction for one more cycle. The hazard is resolved.



Although at some points we seem to be executing the `WB` and `IF` stages simultaneously, but the instruction does not actually write back to memory and the `WB` stage is just for pipelining (For example, the `WB` stage of the `lw` overlaps with the `IF` stage of the `beqz` instruction, but the `lw` instruction does not write back to memory).

b)

No, we add proper hardware to solve structural hazards, not by reordering instructions.

c)

The `NOP` instruction is itself an instruction, and needs to be fetched in the `IF` stage. Hence, it will not resolve this structural hazard caused by `IF` stage overlaps with other memory stages.

Problem 4

a)

Memory	ALU/Branch
	`li x12, 0`
	`jal ENT`
	`bne x12, x13, TOP`
	`slli x5, x12, 2`
	`add x6, x10, x5`
`lw x7, 0(x6)`	
`lw x29, 4(x6)`	
	`sub x30, x7, x29`
	`add x31, x11, x5`
`sw x30, 0(x31)`	`addi x12, x12, 2`
	`bne x12, x13, TOP`
	`slli x5, x12, 2`
	`add x6, x10, x5`
`lw x7, 0(x6)`	
`lw x29, 4(x6)`	
	`sub x30, x7, x29`
	`add x31, x11, x5`
`sw x30, 0(x31)`	`addi x12, x12, 2`
	`bne x12, x13, TOP`

b)

For each iteration, we need to run every line from label `TOP` to label `ENT`.

Assume the time cost for each instruction is 1 cycle. And we consider the stall caused by load-use hazard. For a one-issue processor, one iteration will take 10 cycles. For a two-issue processor, one iteration will take 9 cycles.

Hence, the speedup ratio will be $\frac{10}{9} = 1.111$.

c)

An optimized version of the code for one-issue processor is shown below:

```
slli x5, x13, 2
add x6, x10, x5
add x7, x11, x5
Loop:
beqz x5, Exit
lw x29, 0(x6)
lw x30, 4(x6)
addi x5, x5, -8
sub x31, x29, x30
sw x31, 0(x7)
addi x6, x6, -8
addi x7, x7, -8
j Loop
Exit:
```

d)

An optimized version of the code for two-issue processor is shown below:

Memory	ALU/Branch
`lw x29, 0(x6)`	`beqz x5, Exit`
`lw x30, 4(x6)`	`addi x5, x5, -8`
	`sub x31, x29, x30`
`sw x31, 0(x7)`	`addi x6, x6, -8`
	`addi x7, x7, -8`
	`j Loop`

e)

The speed up ratio will be $\frac{9}{6} = 1.5$.