

VE370 Homework 4 579370910084 Lan Wang

1. (15 points) Given this instruction:

lw $\times 5$, -4 ($\times 2$)

As the instruction goes through the pipeline, what will be stored in the pipeline registers:

IF: what's in PC

ID: what's in IF/ID

EX: what's in ID/EX?

MEM: what's in EX/MEM

WB: what's in MEM/WB?

IF: PC stores the address of this lw instruction

ID: the address of this lw instruction & the binary code of this instruction

EX: the control signals for lw & the address of this lw instruction
& readData 1 & readData 2 & the extended immediate

& the input for ALU Control of lw & writeRegister No.

MEM: the control signals for lw (without ALUSrc & ALUOp)

& ALUResult (including zero) & writeRegister No. & readData 2

& the address generated by the address of this lw instruction
and immediate

WB: readData (from memory) & ALUResult

& control signals (Mem-to-Reg and RegWrite)

2. (1) What is the clock cycle time? (2 points)

(2) What is the execution time of a sw instruction in the pipelined processor? (3 points)

(3) If we can split one stage of the pipelined datapath into two new stages, each with half the latency of the original stage, which stage would you split and what is the new clock cycle time of the processor? (5 points)

(4) Using the processor to run a program of 1,000 instructions, what is the total execution time?
What is the CPI? (10 points)

$$(1) 350 \text{ ps}$$

$$(2) 350 \times 4 = 1400 \text{ ps}$$

$$(3) \text{ ID stage. } 300 \text{ ps}$$

$$(4) \text{ CPI} = \frac{1000+4}{1000} = 1.004$$

$$\begin{aligned} T_{\text{total}} &= IC \times CPI \times T_C \\ &= 1000 \times \frac{1000+4}{1000} \times 350 = 35400 \text{ ps} \end{aligned}$$

3. (10 points) Assume that x11 is initialized to 11 and x12 is initialized to 22. Suppose you executed the code below on a pipelined processor that does not handle data hazards at all.

L1: addi x11, x12, 5
L2: add x13, x12, x11
L3: addi x14, x11, 15

- (1) Indicate data dependencies, if any, in above instruction sequence. (which register between which instructions) (5 points)
(2) What would the final values of registers x13 and x14 be? (5 points)

→ L1 & L2 , L1 & L3

→ x13 : 33

x14 : 26

4. (30 points) Given the following instructions:

L1: sw x18, -12(x8)
L2: lw x3, 8(x18)
L3: add x6, x3, x3
L4: or x8, x9, x6

- a) Assume there is no forwarding in this pipelined processor. Indicate hazards and add NOP instructions to eliminate them. How many clock cycles will it take to execute the instructions? (10 points)
b) Assume there is ALU-ALU forwarding. Indicate hazards and add NOP instructions to eliminate them. How many clock cycles will it take to execute the instructions? (10 points)
c) Assume there is full forwarding. Indicate hazards and add NOP instructions to eliminate them. How many clock cycles will it take to execute the instructions? (10 points)

a) data hazards : L2 & L3 , L3 & L4

add two NOP between L2 & L3 , another two between L3 & L4

$8+4 = 12$ clock cycles

b) data hazards : L2 & L3 .

add two NOP between L2 & L3

$6+4 = 10$ clock cycles

c) data hazards : L2 & L3

only one NOP is needed between L2 & L3

$5+4 = 9$ clock cycles

5. (25 points) Given this MIPS assembly instruction sequence executed by the pipelined processor:

```
sub x6, x2, x1
lw x3, 8(x6)
lw x2, 0(x6)
or x3, x5, x3
sw x3, 0(x5)
```

- If the processor has forwarding, but we forgot to implement the hazard detection unit, what happens when this code executes? (5 points)
- If there is forwarding, for the first five cycles during the execution of this code, specify which signals are asserted in each cycle by hazard detection and forwarding units. (10 points)
- If there is no forwarding, what new inputs and output signals do we need for the hazard detection unit? Using this instruction sequence as an example, explain why each signal is needed. (10 points)

a) Nothing will happen since there is no load use data hazard.

	PC Write	IF/ID Write	Hazard	Forward A	Forward B
CC1			X	XX	XX
CC2			0	XX	XX
CC3			0	00	00
CC4			0	10	00
CC5			0	01	00

b) We need:
 ID/EX. reg Write
 EX/MEM. reg Write
 EX/MEM. rd

If there is no forwarding, we can only add nop to ensure the correctness. So, the unit needs to add nop(s) if data used in ID depends on what produced by EX or MEM;

Only the register number of writing destination in MEM stage needs to be added, since ID/EX. rd is already existed;
 No new outputs are needed.

Then if ID/EX. reg Write == 1 & (IF/ID.Rs1 == ID/EX. rd | IF/ID.Rs2 == ID/EX. rd),
 add 1 nop

if EX/MEM. reg Write == 1 & (IF/ID.Rs1 == EX/MEM. rd | IF/ID.Rs2 == EX/MEM. rd),
 add 2 nops