Homework 4

VE370 - Intro to Computer Organization Summer 2022

* Name: Huang Yucheng ID: 519021910885

Exercise 1

Given this instruction:

`lw x5,-4(x2)`

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?

Answer:
The machine code is `1111 1111 1100 0001 0010 0010 1000 0011`
IF: in PC, it stores the address of lw
ID: in IF/ID, it stores the following things

1. the things in PC, actually the address of lw
2. the instruction `0xFFC12283`

EX: in ID/EX, it stores the following things

1. the things in PC, actually the address of lw
2. Read data 1: content in `x2`
3. Read data 2: `Imm[4:0]`, actually `11100` = 28
4. ImmGenOut: -4, actually `0xFFFFFFFC`
5. `Inst[30, 14-12]`, actually 1, 010
6. Rd = 5
7. RegWrite = 1
8. Branch = 0
9. ALUSrc = 1
10. ALUOp = 00
11. MemRead = 1
12. MemWrite = 0
13. MemToReg = 1

MEM: in EX/MEM, it stores the following things

1. the address of lw + Imm $\times 2$

2. ALU result: content in `x2-4`
3. Read data 2: `Imm[4:0]`, actually `11100` = 28
4. Rd = 5
5. RegWrite = 1
6. Branch = 0
7. MemRead = 1
8. MemWrite = 0
9. MemToReg = 1
10. Zero = X

WB: in MEM/WB, it stores the following things

1. ALU result: content in `x2-4`
2. Memory content in `x2-4`
3. Rd = 5
4. RegWrite = 1
5. MemToReg = 1

Exercise 2

Assume that individual stages of the RISC-V pipelined datapath have the following latencies:

IF	ID	EX	MEM	WB
250ps	350ps	150ps	300ps	200ps

Also, assume that instructions executed by the processor are broken down as follows:

ALU/Logic	Jump/Branch	Load	Store
45%	20%	20%	15%

(1) What is the clock cycle time?

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

(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?

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

Answer:

(1) 350ps

(2) $350 \times 5 = 1750$ps

(3) ID Stage since it has the longest time, and new clock cycle is 300ps

(4) CPI = $\frac{1000+4}{1000} = 1.004$, and total time = $1004 \times 350 = 351400$ps

Exercise 3

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)

(2) What would the final values of registers `x13` and `x14` be?

Answer:

(1) There is data dependencies in `L1` and `L2`, `L1` and `L3`, because they both call `x11` and `L1` operates `x11`

(2) `x13` will be 33, and `x14` will be 26.

Exercise 4

Given the following instructions:

```
L1: sw x18,-12(x8)
L2: lw x3, 8(x8)
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?

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?

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?

Answer:

a) We have two hazards here, one is load hazard between `L2` and `L3`, the other is EX hazard between `L3` and `L4`, so we need to add two `NOP` in these two gaps.

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

Now, we have 8 sentences so that the total clock cycle should be $8 + 4 = 12$

b) Since it is ALU-ALU forwarding, we don't need to add NOP between `L3` and `L4`, so we need to add two `NOP` between `L2` and `L3`.

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

Now, we have 6 sentences so that the total clock cycle should be $6 + 4 = 10$

c) Since it is full forwarding, we don't need to add NOP between `L3` and `L4`, but we need to add one `NOP` between `L2` and `L3`.

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

Now, we have 5 sentences so that the total clock cycle should be $5 + 4 = 9$

Exercise 5

Given this 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)
```

a) If the processor has forwarding, but we forgot to implement the hazard detection unit, what happens when this code executes?

b) 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.

c) 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.

Answer:

a) Nothing will happen because there is no loaduse hazard, so that hazard detection will not detect anything.

b) Since there is no stall, PCWrite and IF/IDWrite will always be 1 and Hazard will always be 0.

(1) ForwardA = XX; ForwardB = XX

(2) ForwardA = XX; ForwardB = XX

(3) ForwardA = 00; ForwardB = 00

(4) ForwardA = 10; ForwardB = 00

(5) ForwardA = 01; ForwardB = 01

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

We need to add three inputs here EX/MEM.RegRd, ID/EX.RegWrite, EX/MEM.RegWrite.

When it satisfies ID/EX.RegWrite ==1, ID/EX.RegRd == IF/ID.RegisterRs1 or ID/EX.RegRd == IF/ID.RegisterRs2, we add the first NOP.

When it satisfies EX/MEM.RegWrite ==1, EX/MEM.RegRd == IF/ID.RegisterRs1 or EX/MEM.RegRd == IF/ID.RegisterRs2, we add the second NOP.

Since we already have the ID/EX block, we don't need new outputs here.