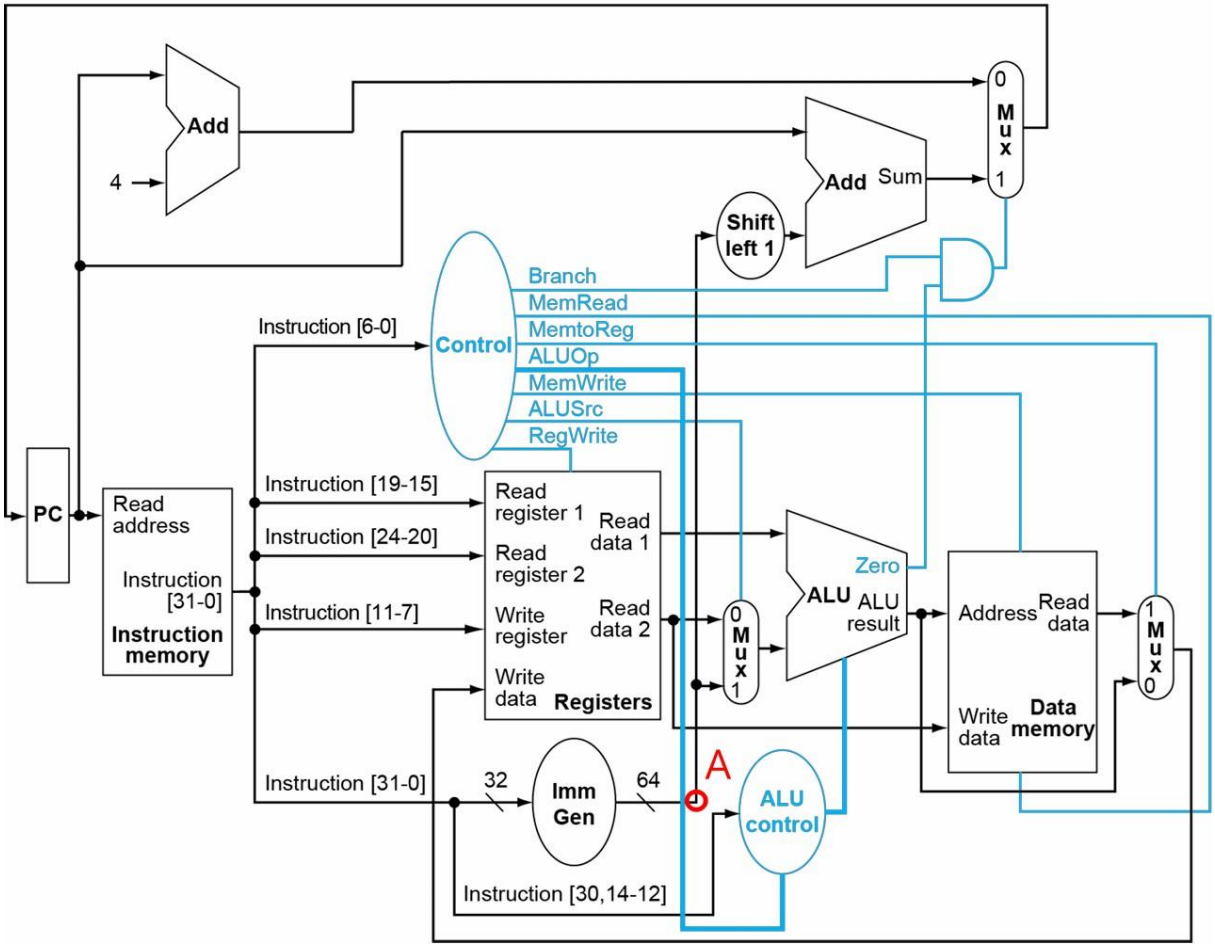
CS4100 Computer Architecture

Spring 2024, Homework 4

Due: 23:59, 5/12/2024

1. (14 points) Consider the single-cycle processor below:



(a) (4 points) What instruction(s) may fail to run correctly if the interconnections labeled **A** above have been cut? Choose the answer(s) from add, sd, ld, and beq.

sd, ld, beq

(b) (4 points) What instruction(s) would still be correct if the input signals of **Read register 1** and **Read register 2** were swapped? Choose the answer(s) from add, sd, ld, and beq.

add, beq

(c) (6 points) What control signal(s) must be set to 0 when the program runs the instruction **sd x13 0(x12)**? What are the **respective** consequences if they are set to 1? Choose the answer(s) from Branch, MemRead, MemtoReg, MemWrite, ALUSrc, and RegWrite.

Branch, MemRead, RegWrite.

若 Branch 被設成 1，則可能會拿到錯誤的下一個指令的位址。

若 MemRead 被設成 1，因為 MemWrite 此時是 1 ，同時 access memory 會有衝突。

若 RegWrite 被設成 1，則可能改到 Instruction[11:7] 所表示的 reg 的值。

2. (11 points) Consider the execution of the machine instruction $01EEA523_{hex}$ on the single-cycle processor.

   $01EEA523_{16}$ = 0000000_11110_11101_010_01010_0100011$_2$ => sw x30, 10(x29)

   (a) (7 points) What are the values of the signals: Branch, MemRead, MemtoReg, ALUOp, MemWrite, ALUSrc and RegWrite?

   Branch: 0, MemRead: 0, MemtoReg: 0, ALUOp: 00, MemWrite: 1, ALUSrc: 1, RegWrite: 0.

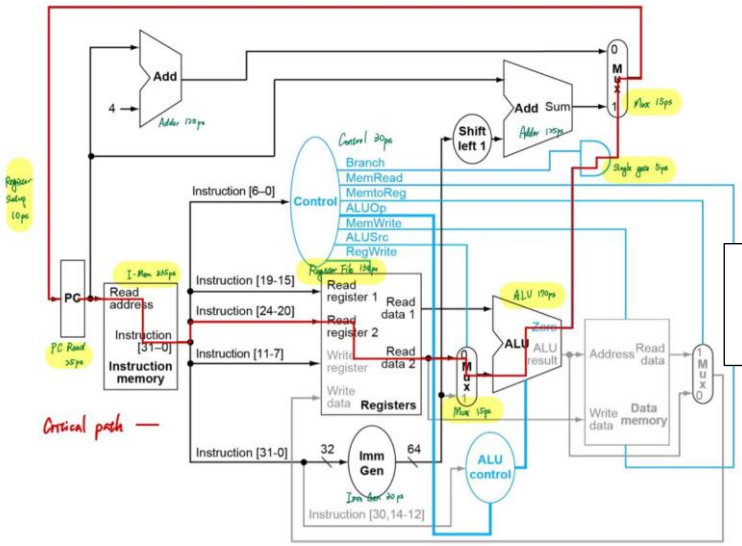
   (b) (4 points) What are the input values of the ALU? You can use Reg[x] to denote the value of register x.

   One of input values is Reg[x29], and the other one is 10.

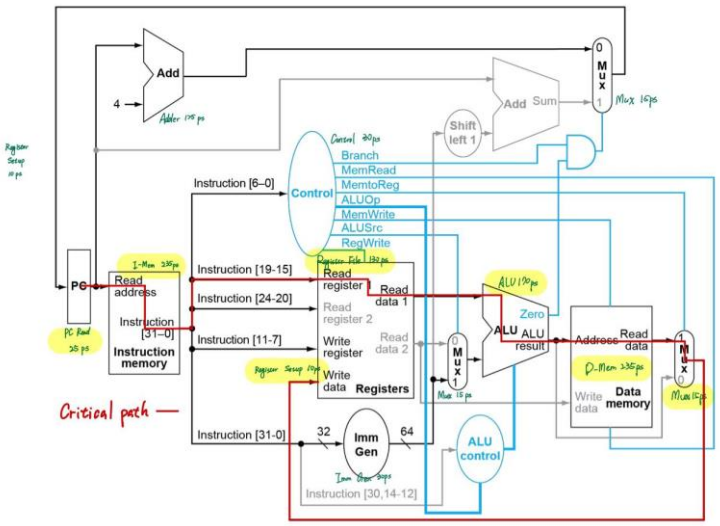
3. (8 points) Assume that the logic blocks used to implement the single-cycle processor have the following delay values:

I-Mem/ D-Mem	Register File	Mux	ALU	Adder
235ps	130ps	15ps	170ps	125ps
Single Gate	PC Read	Register Setup	Imm Gen	Control
5ps	25ps	10ps	30ps	30ps

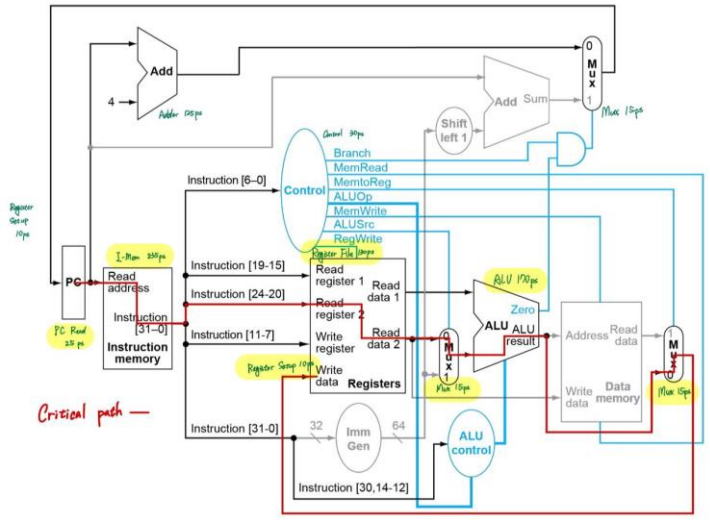
"I-Mem/D-Mem" is the amount of time to access the Instruction or Data Memory.
"Register File" is the amount of time to read rs1 and rs2 after a rising clock edge.
"PC Read" is the amount of time needed after a rising clock edge for the new PC value to appear on the output; this delay value applies to the PC only.
"Register Setup" is the amount of time a register's data input must be stable before a rising clock edge; this delay value applies to both the PC and Register File.
"Control" is the total amount of time for the Control unit and the ALU control unit to produce the 4-bit "ALU operation".



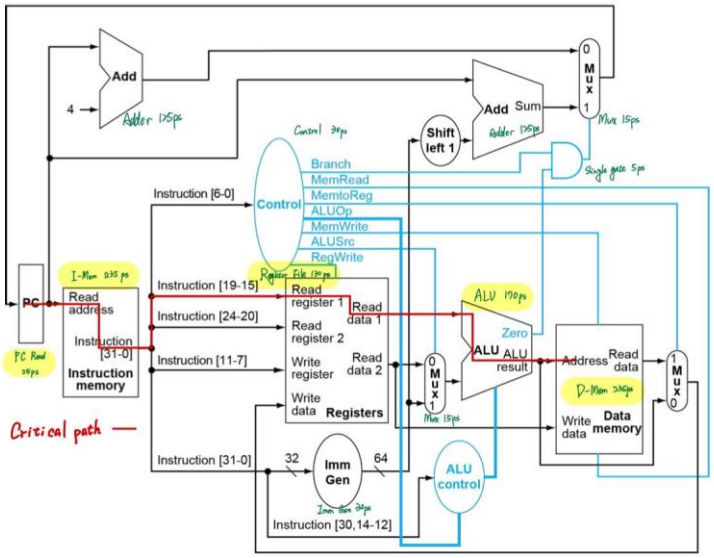
branch:
25 + 235 + 130 + 15 + 170 + 5 + 15 + 10
= 605 ps



load:
25 + 235 + 130 + 170 + 235 + 15 + 10
= 820 ps



add:
25 + 235 + 130 + 15 + 170 + 15 + 10
= 600 ps



store:
25 + 235 + 130 + 170 + 235
= 795 ps

(a) (4 points) Consider the four instructions: add, ld, sd, beq, which is the least time consuming one and how much time does it need?

The pictures above show that add is the least time consuming instruction, and it needs 600 ps.

(b) (4 points) What is the minimum clock period for this processor? You must justify your answer.

The minimum clock period is equal to the most time consuming instruction, which is ld. The pictures show that ld needs 820 ps. Therefore, the minimum clock period is 820 ps.

4. (5 points) Consider a non-pipelined processor with a clock rate of 4GHz, executing a program which has 1200 instructions and an average CPI of 4. The same processor is upgraded to a pipelined processor with 5 stages, but the clock rate is reduced to 2GHz. What is the maximum speedup when running the same program on the pipelined processor?

   $[(4 * 1200) / (4 * 10^9)] / [(1200 + 4) / (2 * 10^9)]$

   $= (4800 / 4) / (1204 / 2) \fallingdotseq 1.993$

   The maximum speedup is about 1.993.

5. (10 points) Consider the instruction sequence below:

   ld x28, 4(x5)

   add x29, x6, x7

   sub x30, x28, x29

   sd x30, 0(x11)

   (a) (3 points) List all the data dependencies, regardless of whether they cause any hazard or not.

   x28 of sub instruction depends on x28 of ld instruction.
   x29 of sub instruction depends on x29 of add instruction.
   x30 of sd instruction depends on x30 of sub instruction.

   (b) (3 points) Assume that the instruction sequence is executed on a five-stage pipelined processor with forwarding. Are there any hazards that will require the pipeline to stall because they cannot be resolved by forwarding? Justify your answer.

   若使用 forwarding 的話，ld 指令的 rd 當作下一行指令的 rs 需要多 stall 一個 cycle 才能正常運作，剩下的 hazard 的問題不需要 stall 就可以解決，而 sub 指令與 ld 指令中間還隔著一行 add 指令，因此不需要 stall 就可以解決 hazard 的問題。

   (c) (4 points) Assuming that the instruction sequence is executed on a five-stage pipelined processor without the forwarding unit and the hazard detection unit, insert a minimum number of NOP (no operation) instructions to ensure correct execution.

ld	IF	ID	EX	MEM	WB							
add		IF	ID	EX	MEM	WB						
sub			-	-	IF	ID	EX	MEM	WB			
sd						-	-	IF	ID	EX	MEM	WB

   ld x28, 4(x5)
   add x29, x6, x7
   nop
   nop
   sub x30, x28, x29
   nop
   nop
   sd x30, 0(x11)

6. (12 points) Consider the following sequence of branch outcomes: NT, T, T, NT, NT, NT, NT, T, where NT denotes not-taken and T denotes taken.

(a) (4 points) What are the respective accuracy rates of the always-not-taken predictor and the always-taken predictor for the sequence of branch outcomes?

The accuracy rate of always-not-taken predictor is 0.625.

The accuracy rate of always-taken predictor is 0.375.

(b) (4 points) Assume that a 1-bit dynamic predictor starts at the T state. Complete the following table and write down the accuracy rate of this predictor for the sequence of branch outcomes.

Ground truth	NT	T	T	NT	NT	NT	NT	T
State	T	NT	T	T	NT	NT	NT	NT
Decision	T	NT	T	T	NT	NT	NT	NT
Correctness	X	X	O	X	O	O	O	X

The accuracy rate is 0.5.

(c) (4 points) Assume that a 2-bit dynamic predictor starts at the "strongly predict taken" state. Complete the following table and write down the accuracy rate of this predictor for the sequence of branch outcomes.

Define:

Strongly predict taken: SPT

Weak predict taken: WPT

Weak predict not taken: WPNT

Strongly predict not taken: SPNT

Ground truth	NT	T	T	NT	NT	NT	NT	T
State	SPT	WPT	SPT	SPT	WPT	WPNT	SPNT	WPNT
Decision	T	T	T	T	T	NT	NT	NT
Correctness	X	O	O	X	X	O	O	X

The accuracy rate is 0.5.

7. (30 points) Consider the following instruction sequence running on the five-stage pipelined processor:

beq x11, x12, Label
sd x15, 0(x23)
ld x15, 0(x24)
add x11, x6, x12
sub x11, x13, x12

Assume x11 ≠ x12.

Note that in the following questions, structural hazards are considered only in (a) and (b).

(a) (10 points) Assume the processor predicts each branch instruction to be not taken. If we only have one memory (for both instructions and data), there is a structural hazard every time we need to fetch an instruction in the same cycle in which another instruction accesses data. To guarantee the processor to work correctly, this structural hazard must always be resolved in favor of the instruction that accesses data. In other words, there is a hazard detection unit in the IF stage, and if a structural hazard occurs, the instruction in the IF stage needs to stall for that cycle. What is the total execution time of this instruction sequence? We have learned that data hazards can be eliminated by adding NOPs to the code, so can you do the same with this structural hazard and why?

beq	IF	ID	EX	MEM	WB						
sd		IF	ID	EX	MEM	WB					
ld			IF	ID	EX	MEM	WB				
add				IF	ID	EX	MEM	WB			
sub					-	-	IF	ID	EX	MEM	WB

Red means not used. Total time: 11 cycles

不行，因為就算是 nop 還是得去 memory 抓指令，所以無法解決 structural hazard 的問題。

(b) (5 points) Assume we use the same processor in (a). What is the minimum number of cycles you can achieve by adjusting the order of the instructions without losing the correctness? Also give the new sequence of instructions after re-ordering.

beq x11, x12, Label

add x11, x6, x12

sub x11, x13, x12

sd x15, 0(x23)

ld x15, 0(x24)

beq	IF	ID	EX	MEM	WB				
add		IF	ID	EX	MEM	WB			
sub			IF	ID	EX	MEM	WB		
sd				IF	ID	EX	MEM	WB	
ld					IF	ID	EX	MEM	WB

Red means not used.

(c) (5 points) Assuming Stall on Branch (i.e., wait until the branch outcome is determined before fetching next instruction), what speedup is achieved on this instruction sequence if branch outcomes are determined in the ID stage, relative to the execution where branch outcomes are determined in the MEM stage?

beq	IF	ID	EX	MEM	WB					
sd			IF	ID	EX	MEM	WB			
ld				IF	ID	EX	MEM	WB		
add					IF	ID	EX	MEM	WB	
sub						IF	ID	EX	MEM	WB

10 cycles if the branch outcome is determined in the ID stage.

beq	IF	ID	EX	MEM	WB							
sd					IF	ID	EX	MEM	WB			
ld						IF	ID	EX	MEM	WB		
add							IF	ID	EX	MEM	WB	
sub								IF	ID	EX	MEM	WB

12 cycles if the branch outcome is determined in the MEM stage.

speedup = 12 / 10 = 1.2

(d) (5 points) Assume the processor predicts each branch instruction to be not taken. Also assume each individual pipeline stage of IF, ID, EX, MEM, and WB has the latency of 210 ps, 160 ps, 220 ps, 180 ps, and 100 ps, respectively. If we change load/store instructions to use a register (without an offset) as the address, these instructions no longer need to use the ALU. As a result, MEM and EX stages can be overlapped and the pipeline has only 4 stages. Assuming this change does not affect the clock period, what speedup is achieved in this instruction sequence compared to the original five-stage one?

beq	IF	ID	EX	MEM	WB				
sd		IF	ID	EX	MEM	WB			
ld			IF	ID	EX	MEM	WB		
add				IF	ID	EX	MEM	WB	
sub					IF	ID	EX	MEM	WB

9 cycles * 220 ps / cycle (最久的 latency) = 1980 ps

beq	IF	ID	EX	MEM	WB				
sd		IF	ID	MEM	WB				
ld			IF	ID	MEM	WB			
add				IF	ID	EX	MEM	WB	
sub					IF	ID	EX	MEM	WB

9 cycles * 220 ps / cycle (最久的 latency) = 1980 ps

speedup = 1

(e) (5 points) Given the pipeline stage latencies in (d), repeat the speedup calculation of (d) by considering the (possible) change in the clock period as follows. When EX and MEM are done in a single stage (called EX/MEM stage), most of their work can be done in parallel. As a result, the EX/MEM stage now has a latency that is the larger of the original two, plus 25 ps needed for the work that could not be done in parallel.

beq	IF	ID	EX/MEM	WB				
sd		IF	ID	EX/MEM	WB			
ld			IF	ID	EX/MEM	WB		
add				IF	ID	EX/MEM	WB	
sub					IF	ID	EX/MEM	WB

8 cycles * (220 + 25) ps / cycle (最久的 latency) = 1960 ps

speedup = 1980 / 1960≒1.0102

8. (10 points) Consider the execution of the following sequence of four instructions on a five-stage pipelined processor which uses the always-not-taken strategy for branch prediction and has both the forwarding unit and hazard detection unit.

add x12, x6, x5
sub x10, x11, x12
beq x11, x12, LABEL
sd x11, 0(x12)

Suppose the third instruction is detected to have an invalid target address and cause an exception in the ID stage. Fill in the following table by showing what instructions are in the IF, ID, EX, MEM and WB stages. Note that each instruction in your answer should be one chosen from the given four instructions, the NOP instruction (or bubble), and the first instruction of the exception handler.

Denote "add x12, x6, x5" as "add".
Denote " sub x10, x11, x12" as "sub".
Denote " beq x11, x12, LABEL " as "beq".
Denote " sd x11, 0(x12)" as "sd".
Denote " the first instruction of the exception handler " as "FIEH".

	IF	ID	EX	MEM	WB
Cycle 1	add	-	-	-	-
Cycle 2	sub	add	-	-	-
Cycle 3	beq	sub	add	-	-
Cycle 4	sd	beq	sub	add	-
Cycle 5	FIEH	bubble	bubble	sub	add