Name: _____.

Computer Science and Engineering 431 (CMPEN 431)

Exam 1

Fall 2022

9/27/2022

Name: _____.

Q1	10	
Q2	15	
Q3	15	
Q4	15	
Q5	15	
Q6	25	
Q7	15	
Total	110	

Note:

You have **1 hour and 15 minutes** to complete the exam.

This exam is **closed notes and book**.

Please **write down solution steps** for partial credits to be considered.

Please also **write you name on top of each page**.

I strongly recommend **solving easy problems first**, as problems are not in the order of complexity.

Name: _____.

Q1. Select True or False (10 pt, 1pt each).

- (True / False) MIPS is a RISC architecture.
- (True / False) In the instruction decode stage of MIPS, registers are read only after the instruction is fully decoded.
- (True / False) In a single-issue pipelined processor (what we learned in class), the ideal CPI is 1.
- (True / False) In general, CPI cannot go below 1.
- (True / False) MIPS uses a separate instruction and data memory to avoid structural hazard.
- (True / False) MIPS uses a separate register file hardware for instruction decode (ID) and writeback (WB) stage to avoid structural hazard.
- (True / False) Data forwarding reduces control hazard.
- (True / False) Even without BTB or branch delay slots, the pipelined MIPS processor we studied do not need to stall if a branch predictor correctly predicts branch not taken.
- (True / False) Even without BTB or branch delay slots, the pipelined MIPS processor we studied do not need to stall if a branch predictor correctly predicts branch taken.
- (True / False) BHT and BTB is accessed only when branch or jump instruction is being executed.

Name: _____.

Q2. Answer the following questions (15pt total).

- Q2-1. Write a MIPS assembly program that is equivalent to the following C program. Assume that `tmp` is stored in register $t0, and the base address of `Array` is stored in register $s0. (5pt)

```
int32_t Array[100]; int32_t tmp;
…
Array[3] = tmp;
```

sw $t0, 12($s0)

- Q2-2. From the below MIPS instructions, what value would the 16-bit immediate field of the beq instruction be holding? You can answer in decimals. **Explain your answer briefly**. (5pt)

```
L: add $1, $2, $3
   sub $2, $3, $5
   andi $2, $2, 1
   beq $2, $4, L
   lw $2, 0($5)
```

-4

It encodes the number of instructions to branch, counting from the next instruction after beq.

- Q2-3: Assume a branch that has the following outcome pattern (T: Taken, N: Not taken). (5pt)
  N N N T N N N T N N T T
  Assuming a 2-bit predictor with four states of strong not taken (SN), weak not taken (wn), weak taken (wt), and strong taken (ST), how many branches will be predicted correctly if we start from state wt?

	N	N	N	T	N	N	N	T	N	N	T	T
cur state	wt	wn	SN	SN	wn	SN	SN	SN	wn	SN	SN	wn	wt
correct?	X	O	O	X	O	O	O	X	O	O	X	X	7

Name: _____.

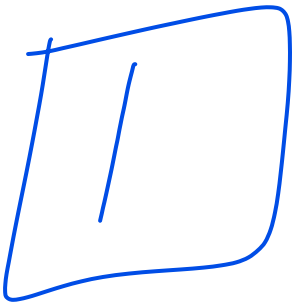
Q3. Consider two machines executing the exact same 10,000 instructions (instruction count = 10,000).

Machine A - Single-cycle MIPS processor. Clock cycle time (CCT) is 1000ps.

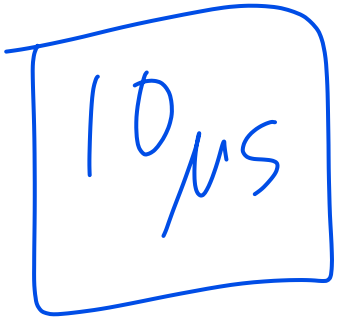
Machine B - Multi-cycle processor. CCT is 250ps. Instructions take the following number of cycles to finish: Branch (3 cycles), R-type (4 cycles), Load (5 cycles), Store (4 cycles).

Assume that among the 10,000 instructions, 20% is Branch, 50% is R-type, 10% is Load, and 20% is Store. Answer the following questions. (15pt total)

- Q3-1: What is the cycle-per-instruction (CPI) of machine A? (3pt)



- Q3-2: What is the execution time of running the 10,000 instructions on machine A? Answer must be in **microseconds (us)**. (2pt)



- Q3-3: What is the CPI of machine B? (3pt)

Name: _____.

- Q3-4: What is the execution time of running the 10,000 instructions on machine B? Answer must be in **microseconds (us)**. (2pt)

9.75 μs

- Q3-5: What is the speedup of machine B, relative to machine A? Answer in **times** (e.g., 2**x** if B is 2 times faster). The answer can be in a form of non-reduced fraction (e.g., something like 4.5/3.7x is fine). (2pt)

$\frac{10}{9.75}$ x

- Q3-6: Will machine B always be faster or slower than A, or will it depend on the program you execute? Choose one side and explain briefly. (3pt)

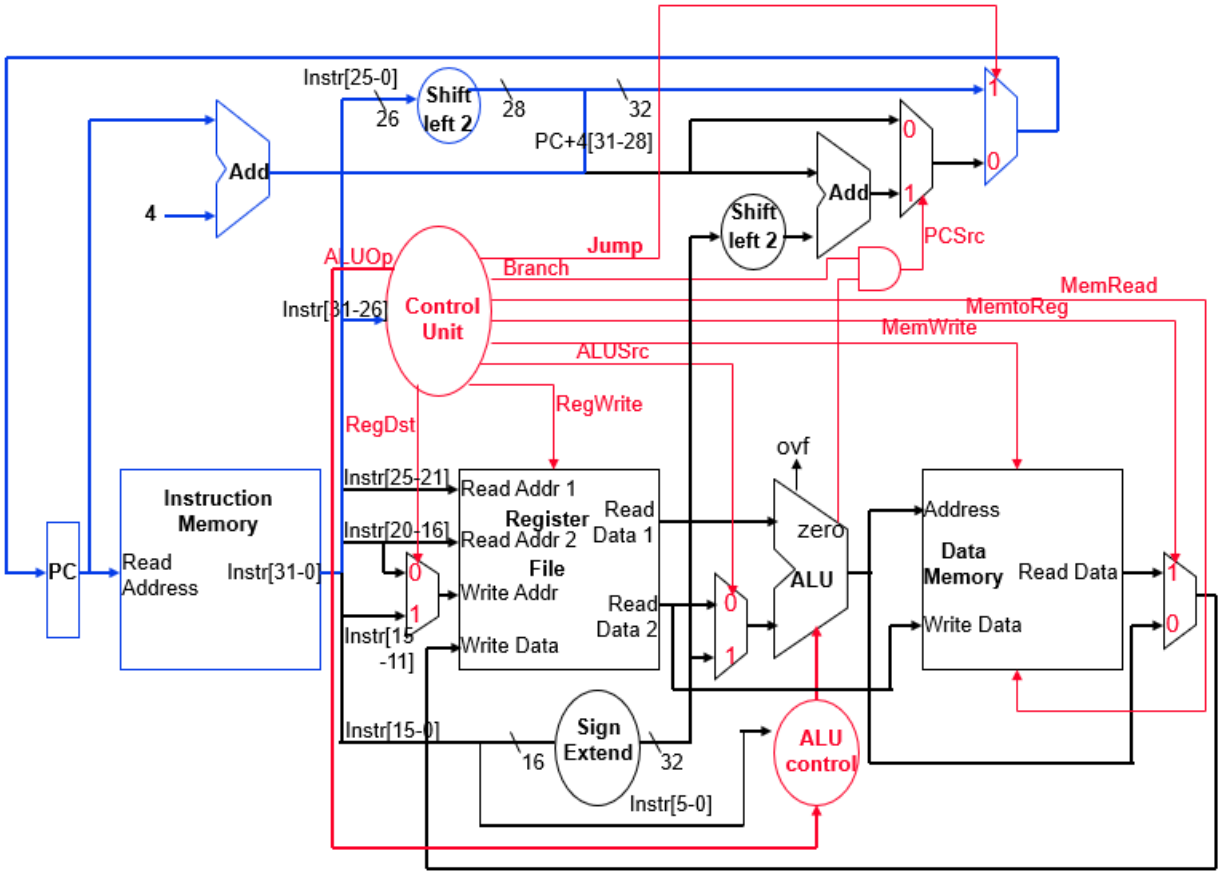
No. It depends on the instruction mix.

Name: _____.

(this page is intentionally left as blank. Use it as needed)

Name: _____.

Q4. Below is the diagram of a MIPS single-cycle CPU hardware. No need to explain your answer for this question. (15pt)



- Q4-1: What are each control signal's value, when you run a store? Fill 0, 1, or X (does not matter) below. (4pt)

RegDst	RegWrite	ALUSrc	MemWrite	MemtoReg	MemRead	Branch	Jump
X	0	1	1	X	0	0	0

- Q4-2: What operation (e.g., add, sub, and, or, xor) will the ALU execute, when you run a store instruction? (3.5pt)

add

- Q4-3: What are each control signal's value, when you run a beq instruction? Fill 0, 1, or X (does not matter) below. (4pt)

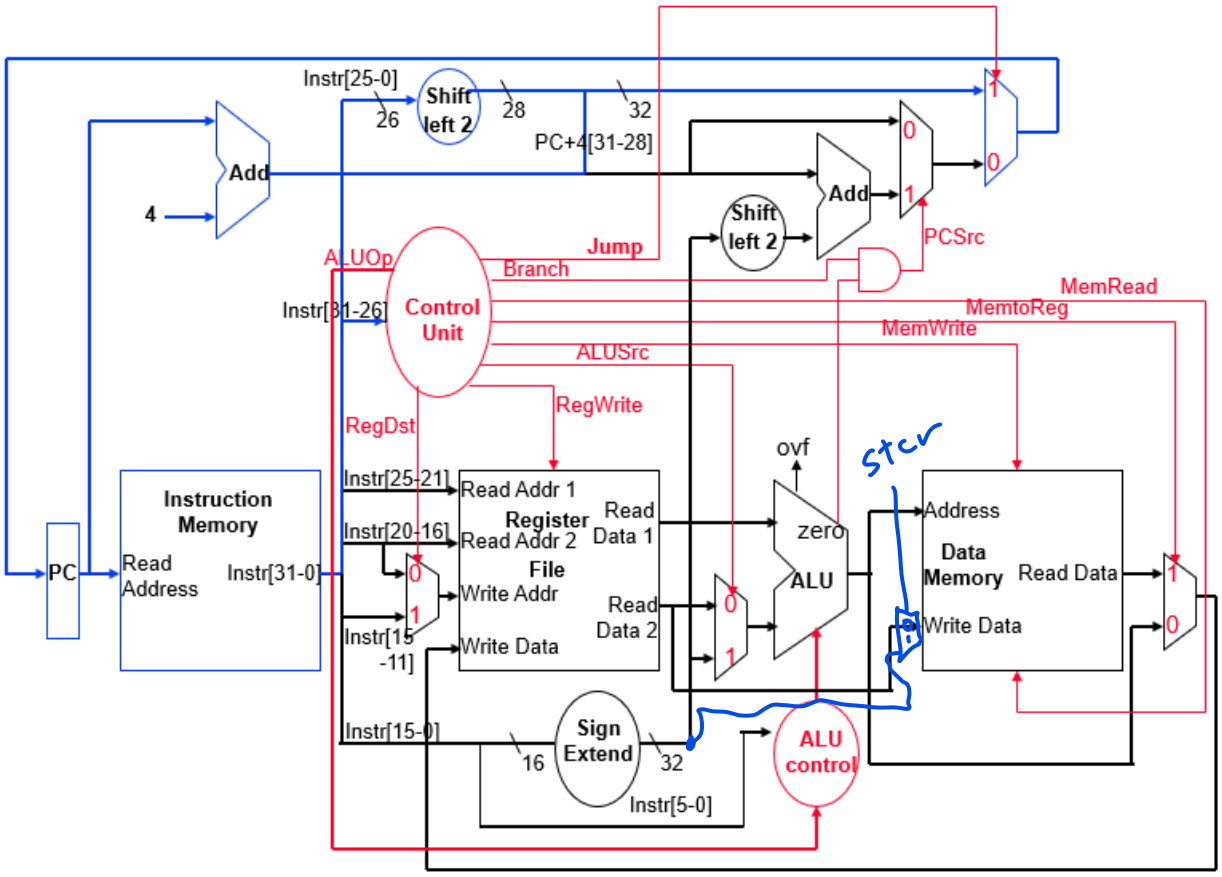
RegDst	RegWrite	ALUSrc	MemWrite	MemtoReg	MemRead	Branch	Jump
X	0	0	0	X	0	1	0

- Q4-4: What operation (e.g., add, sub, and, or, xor) will the ALU execute, when you run a beq instruction? (3.5pt)

sub

Name: _____.

Q5: Below is the diagram of a MIPS single-cycle CPU hardware. (15pt)



Assume you want to design a new store-constant-with-reg-offset (**stcr**) instruction, which stores the 32bit sign-extended immediate value to the memory. The memory address to use for the store is calculated by adding the values in the register $rs and $rt. In other words,

MEM[REG[$rs] + REG[$rt]] <- SignExtend(imm)

On the provided diagram, show how stcr instruction can be supported by adding new control signals, muxes, wires, and gates. You **MUST NOT** add a new adder and use the existing ALU if any addition is needed. Also, show what the value of each of the control signals below must be, as well as any new control signals you added.

RegDst	RegWrite	ALUSrc	MemWrite	MemtoReg	MemRead	Branch	Jump	stcr
X	0	0	1	X	0	0	0	1

Name: _____.

(this page is intentionally left as blank. Use it as needed)

Name: _____.

Q6: Assume a 5-stage MIPS pipeline with Fetch (F), Decode (D), Execute (E), Memory (M), and Writeback (W) stage. Read the below questions and fill the pipeline diagram. Show a stalled stage with a lowercase alphabet. *Showing the potential data hazard (register read-after-write dependence) and forwarding in the diagram is not necessary, but you will get a partial credit if you do so correctly, in case your answer is incorrect (if you get the answer correctly no need to draw the dependence and forwarding). Assume that the registers are already ready if an instruction writing to that register is not shown. No need to explain your answer. (25pt)

- Q6-1: Fill the pipeline diagram, **assuming NO forwarding support at all**. The decode stage will be stalled until the necessary register values are written back. (Hint: as we learned in class, register writeback happens at the first half of the cycle, and register read happens at the second half of the cycle). (5pt)

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
lw $1, 0($2)	F	D	E	M	W														
add $2, $1, $3		f	d	e	D	E	M	W											
lw $4, 0($2)			f	f	F	d	d	D	t	M	W								
add $5, $4, $3						f	f	F	d	d	D	E	M	W					
sw $5, 0($6)									f	f	F	d	d	D	E	M	W		

- Q6-2: Fill the pipeline diagram, **only assuming forwarding support from EX/MEM and MEM/WB stage registers to the input of the ALU (E stage)**, as we've learned in class. (Hint: We do not assume any forwarding support to the input to the M stage). (5pt)

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
lw $1, 0($2)	F	D	E	M	W														
add $2, $1, $3		F	d	D	E	M	W												
lw $4, 0($2)			f	F	D	E	M	W											
add $5, $4, $3					F	d	D	E	M	W									
sw $5, 0($6)						f	F	d	d	D	E	M	W						

OR same answer w/ Q6-3 is also correct.

Name: _____.

- Q6-3: Fill the pipeline diagram, **assuming full forwarding support, including forwarding into the input of the M stage**. (Hint: forwarding can happen to the input of the E and M stage). (5pt)

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
lw $1, 0($2)	F	D	E	M	W														
add $2, $1, $3		F	D	E	M	W													
lw $4, 0($2)			F	D	E	M	W												
add $5, $4, $3				F	D	E	M	W											
sw $5, 0($6)					F	D	E	M	W										

either forwarding is fine

- Q6-4: Fill the pipeline diagram. Assume **branch decision is made at the M stage** (unoptimized branch hardware). Assume **full forwarding**. Assume the branch is **taken**, but the branch predictor **mispredicts** and try to execute the next instruction sequentially. Put FL when an instruction is flushed. (Hint: FL should come at the same cycle (column) with beq's M stage). (5pt)

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
add $1, $1, $1	F	D	E	M	W														
beq $1, $2, L		F	D	E	M	W													
addi $3, $4, 5			F	D	FL														
L: addi $5, $4, 7				F	FL	F	D	E	M	W									

- Q6-5: Fill the pipeline diagram. Assume **branch decision is made at the D stage** (early-branch decision). Assume **full forwarding**. Assume the branch is **taken**, but the branch predictor **mispredicts** and try to execute the next instruction sequentially. Put FL when an instruction is flushed. (Hint: FL should come at the same cycle (column) with beq's D stage). (5pt)

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
add $1, $1, $1	F	D	E	M	W														
beq $1, $2, L		F	D	E	M	W													
addi $3, $4, 5			F	FL															
L: addi $5, $4, 7				FL	F	D	E	M	W										

Name: _____.

Q7: Below is a pseudo-code for data forwarding from MEM/WB state register to the input of the ALU. Answer the below questions. (15pt)

```
if (MEM/WB.RegWrite and (MEM/WB.RegisterRd != 0))
   and (MEM/WB.RegisterRd == ID/EX.RegsterRs)
   and !((EX/MEM.RegWrite and (EX/MEM.RegisterRd != 0)) and
         (EX/MEM.RegisterRd == ID/EX.RegisterRs)))
       ForwardA = 01
```

① ② ③

- Q7-1: Explain briefly what code ① (`MEM/WB.RegWrite`) is checking. (5pt)

Checks if the instruction currently running in the MEM/WB stage will write to a register (R-type or I-type other than the store).

- Q7-2: Explain briefly what code ② (`MEM/WB.RegisterRd == ID/EX.RegsterRs`) is checking. (5pt)

Checks if the register the instruction in the MEM/WB stage is writing to is the same as the (first) source register of the instruction in ID/EX stage (i.e., if there will be a data hazard).

- Q7-3: Explain briefly what code ③ (`!((EX/MEM.RegWrite and (EX/MEM.RegisterRd != 0)) and(EX/MEM.RegisterRd == ID/EX.RegisterRs)))` is checking. (5pt)

Checks if there will be any forwarding from the EX/MEM stage
(only forward from the MEM/WB stage if there is no forwarding from the EX/MEM stage)