

Department of Electrical and Computer Engineering
The University of Texas at Austin

ECE 460N Spring 2025

Instructor: Yale N. Patt

TAs: Luke Mason, Margaret Lee, Jenna May, Roy Mor, Rathna Sivakumar

Final Exam

May 2nd, 2025

Name: _____

PART A

Problem 1 (15 points): _____

Problem 2 (15 points): _____

Problem 3 (15 points): _____

Problem 4 (15 points): _____

Total Part A (45 points): _____

PART B

Problem 5 (25 points): _____

Problem 6 (30 points): _____

Total Part B (55 points): _____

IMPORTANT: Part A is graded out of 45. You will complete three of the four problems and X out the problem you did not complete. If you fail to do so, we will X out problem 1.

Note: Please be sure that your answers to all questions (and all supporting work that is required) are contained in the space provided.

Note: Please be sure your name is recorded on each sheet of the exam.

Please read the following sentence, and if you agree, sign where requested:
I have not given nor received any unauthorized help on this exam.

Signature: _____

GOOD LUCK!

Name: _____

Problem 1 (15 pts): Pipelining

If you choose not to complete this problem, put an X over Problem 1 on the cover page.

Consider an in-order, pipelined LC-3b machine with five stages: FETCH (F), DECODE (D), AGEX (A), MEM (M), and STORE (S). Each stage takes 1 cycle, and the pipeline does not support data forwarding.

Stalls work exactly like lab 6. Instructions stall in **DECODE stage** in the event of a dependency stall. There is no branch predictor. When a branch instruction is executed, the target is fetched when the branch is in **STORE stage**.

The machine executes the following LC-3b assembly program. Assume there are no I-cache stalls while fetching the instructions.

Your job: Complete the timing diagram.

I1	AND R0, R0, #0
I2	LOOP ADD R1, R0, #-1
I3	BRn EXIT
I4	ADD R0, R0, #1
I5	EXIT LSHF R3, R3, #2

Cycle:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
I1	F	D	A	M	S													

Name: _____

Problem 2 (15 pts): Physical Memory

If you choose not to complete this problem, put an X over Problem 2 on the cover page.

Chester wants his code to run faster. His boss, Aniket, informs him that his code doesn't optimize for row buffer hits and interleaving. Unfortunately, he does not remember the physical memory's address mapping. Luckily, you're here to help.

Mapping 1:	R	R	R	R	R	B	C	C	C	C	BoB	BoB
Mapping 2:	B	C	C	C	C	R	R	R	R	R	BoB	BoB
Mapping 3:	R	R	R	R	R	C	C	C	C	B	BoB	BoB
Mapping 4:	C	C	C	C	B	R	R	R	R	R	BoB	BoB
Mapping 5:	C	C	C	C	R	R	R	R	R	B	BoB	BoB

Chester makes the four memory accesses shown below. Access 2 gets a row buffer hit. Access 4 has a bank conflict with Access 3. Which address mapping is used? Explain briefly how you arrived at your answer.

Access 1:	1	0	0	1	0	1	0	1	0	1	0	1
Access 2:	1	0	1	1	0	1	0	1	0	1	1	1
Access 3:	0	1	0	1	0	1	1	1	1	1	0	1
Access 4:	0	0	1	1	1	1	0	0	1	0	1	1

Mapping #:	Explain <i>briefly</i> :

Name: _____

Problem 3 (15 pts): Floating Point

If you choose not to complete this problem, put an X over Problem 3 on the cover page.

Computers A and B support an 8-bit floating point format based on the IEEE floating point standard. They both round-toward-zero if necessary (i.e. cut off the extra low order bits).

Sign	Exponent	Fraction
------	----------	----------

Part a (5 pts): Computer A's floating point format uses 3 exponent bits and 4 fraction bits. It performs $x + x$, where x is the value with floating point representation 00010001. What is the result of the addition, represented in the same 8-bit floating point format?

Result:

--	--	--	--	--	--	--	--

Part b (10 pts): Your job is to determine the number of exponent bits and fraction bits of Computer B. Computer B performs repeated addition of y , an 8-bit floating point number. Below are the intermediate results for $y + y$ and $(y + y) + y$.

After $y + y$

0	0	1	1	0	1	0	1
---	---	---	---	---	---	---	---

After $(y + y) + y$

0	0	1	1	1	0	0	1
---	---	---	---	---	---	---	---

Exponent bits:

--

Fraction bits:

--

Name: _____

Problem 4 (15 pts): Cache

If you choose not to complete this problem, put an X over Problem 4 on the cover page.

You were looking through documentation for two different caches, and you realized some information was missing. Memory has 16-bit addresses and is byte-addressable. WB and WT stand for Writeback and Writethrough. Both caches use random replacement.

Part a (3 pts): What bit(s), other than the tag itself, must be contained in a tag store entry if it is a writeback cache?

--

Part b (12 pts) : Use the given information to fill in the blanks about the caches below.

Cache Format			#Sets	#Ways	Cache Line Size	Data Store Size (Bytes)	WB or WT	Tag Store Size (Bits)
Tag	Index	Offset						
[15:9]	[8:1]	[0:0]		2			WT	
[15:11]	[10:4]	[3:0]				8192 B (2 ¹³)		3584 bits (2 ⁹ * 7)

Name: _____

Problem 5 (25 pts): Virtual Memory

An LC-3b ISA processor, byte-addressable, is augmented with VAX-like two-level virtual memory. Physical addresses are 12 bits. System space ranges from x0000 to x7FFF, and user space ranges from x8000 to xFFFF. PTEs in this system are 2 bytes and use the following format.

15	14	13		?	?		0
Valid	Prot	0 Padding				PFN	

The processor has a 2-entry TLB that uses least recently used (LRU) replacement. The TLB is initially empty and only contains PTEs of user pages.

A short program is executed, which accesses the following physical memory addresses in the order shown below. The **program starts at virtual address x8000**. Also shown are the corresponding values stored in each memory location **after program execution**. Memory locations in the table contain word values at each physical address.

Physical Address (hex)	Physical Address (binary)	Word Value (hex)	Word Value (binary)
x04E	0 0 0 0 0 1 0 0 1 1 1 0	xC00B	1 1 0 0 0 0 0 0 0 0 0 0 1 0 1 1
x580	0 1 0 1 1 0 0 0 0 0 0 0	x8012	1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0
x900	1 0 0 1 0 0 0 0 0 0 0 0	x6580	0 1 1 0 0 1 0 1 1 0 0 0 0 0 0 0
x04E	0 0 0 0 0 1 0 0 1 1 1 0	xC00B	1 1 0 0 0 0 0 0 0 0 0 0 1 0 1 1
x5FE	0 1 0 1 1 1 1 1 1 1 1 0	x8006	1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0
x37C	0 0 1 1 0 1 1 1 1 1 0 0	x5E08	0 1 0 1 1 1 1 0 0 0 0 0 1 0 0 0
x902	1 0 0 1 0 0 0 0 0 0 0 1 0	x1DA4	0 0 0 1 1 1 0 1 1 0 1 0 0 1 0 0
x904	1 0 0 1 0 0 0 0 0 0 1 0 0	x7580	0 1 1 1 0 1 0 1 1 0 0 0 0 0 0 0
x050	0 0 0 0 0 1 0 1 0 0 0 0	xC011	1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1
x880	1 0 0 0 1 0 0 0 0 0 0 0	x800A	1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0
x500	0 1 0 1 0 0 0 0 0 0 0 0	x5E08	0 1 0 1 1 1 1 0 0 0 0 0 1 0 0 0
x906	1 0 0 1 0 0 0 0 0 0 1 1 0	xF025	1 1 1 1 0 0 0 0 0 0 0 1 0 0 1 0 1

PROBLEM CONTINUES ON NEXT PAGE

Name: _____

Part a (6 pts): Specify the assembly language program that was executed given the above memory access sequence.

<p>.ORIG x8000</p>

Part b (5 pts) : What is the page size?

--

Part c (6 pts): What is the PBR and SBR, assuming both page tables start at the beginning of a page? *Hint: the program starts at virtual address x8000*

PBR:	SBR:
------	------

Part d (8 pts): Fill in the contents of the TLB after the program has finished. Remember, the TLB only contains PTEs of user pages.

VPN	PTE

Name: _____

Question 6 (30 pts): Tomasulo's Algorithm

A microarchitecture based on the Tomasulo algorithm, augmented with a ROB, has these specifications:

- The instruction cycle is 4 stages: FETCH, DECODE, EXECUTE, and WRITEBACK.
- EXECUTE is 2 cycles for ADD, 3 for MUL, and 1 for BR. Other stages are 1 cycle each.
- The processor is multiple-issue, i.e. FETCH, DECODE, and WRITEBACK support **up to two** instructions per cycle.
- There is one pipelined adder, one pipelined multiplier, and one branch unit, with three reservation stations each.
- After DECODE, an instruction is stored in a reservation station if one is available. It may enter EXECUTE immediately if all sources are available. Reservation stations are allocated top to bottom.
- Dependent instructions may begin execution the cycle after the last source value is written back.
- Reservation Stations are deallocated at the end of WRITEBACK, and can be allocated for another instruction in the same clock cycle.
- A branch predictor predicts the direction and target of a branch during FETCH. The predicted target is fetched in the cycle after FETCH of the branch.
- If a branch is mispredicted, the correct target is fetched after WRITEBACK of the branch and the pipeline is flushed.
- Keep in mind, the RAT and reservation stations may contain speculatively executed instructions.
- Note: The system only has 4 general purpose registers, R0, R1, R2, and R3.

Your job: using the partial information shown on pages 9 and 10, answer the following:

Part a (16 pts): An Aggie with a 4.0 GPA executes a program segment on this processor. Unfortunately, he never learned how to code, so the program segment contains an **infinite loop**. Complete the program segment on page 9.

Part b (10 pts): Fill in the bolded boxes of the RAT and reservation stations on page 10.

Part c (4 pts): The branch predictor uses a 2-bit saturating counter to predict the branch direction. List the possible value(s) of the 2-bit counter before the program snippet is executed.

--

PROBLEM CONTINUES ON NEXT PAGE

Name: _____

An Aggie with a 4.0 GPA executes a program segment on this processor. Unfortunately, he never learned how to code, so the program segment contains an **infinite loop**.

INST#	Instruction
I1	ADD R2, R0, #0
I2	BRz LOC
I3	LOOP __ __, R1, R2
I4	__ __, R1, __
I5	ADD R2, R0, #-1
I6	BRp LOOP
I7	LOC MUL __, __, __
I8	ADD R2, __, R3

A partial timing diagram is shown below. You do not need to fill out the timing diagram, but the information given **will be used to solve part a and b**. Use the diagram for any scratch work.

*Note: a * indicates the instruction is waiting in a reservation station.*

#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	...
I1	F	D	E	E	W														...
I2	F	D	*	*	*	E	W												...
I3		F	D																...
I4		F	D						W										...
I5			F	D															...
I6			F	D															...
				F	D														...
				F	D														...
																			...
																			...
																			...
																			...
...

PROBLEM CONTINUES ON NEXT PAGE

Name: _____

The state of the register alias table (RAT) is partially shown at the end of cycle 4 and cycle 5:

RAT After Cycle 4

	V	Tag	Value
R0	1	θ	3
R1	0		1
R2	0		56
R3	0		2

RAT After Cycle 5

	V	Tag	Value
R0	0		3
R1	0		1
R2	0		
R3			

The state of the reservation stations are partially shown at the end of **cycle 5**. Remember, speculatively executed instructions may appear in reservation stations and the RAT.

ADD Reservation Stations

	V	Tag	Value	V	Tag	Value
α	0	ρ	3	0		2
β	1	θ	1	1		3
γ	1		3	1	-	-1

MUL Reservation Stations

	V	Tag	Value	V	Tag	Value
π	1	θ	1	1		3
ρ	1			1		
θ	-	-	-	-	-	-

The following blank timing diagram has been provided for extra scratch work:

#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	...
																			...
																			...
																			...
																			...
																			...
																			...
																			...
																			...
																			...
																			...
																			...
																			...
																			...
																			...
...