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

ECE 460N Spring 2023
Instructor: Yale N. Patt
TAs: Michael Chen, Ali Mansoorshahi
Final Exam
April 28, 2023

Name: _____

PART A:

Problem 1 (20 points): _____

Problem 2 (15 points): _____

Problem 3 (15 points): _____

Total Part A (50 points): _____

PART B:

Problem 4 (30 points): _____

Problem 5 (30 points): _____

Total Part B (60 points): _____

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!
(HAVE A GREAT SUMMER)

Name: _____

Problem 1 (20 Points): Answer the following questions.
Note: For each of the four answers below, if you leave the box empty, you will receive one point.

Part a (5 points): Many operate instructions require three addresses, two for the locations of the two source operands and one for the destination where the result is to be stored. ISAs are often characterized as 3-address, 2-address, 1-address, and zero-address depending on the number of operands that are **explicitly** identified in the instruction. LC-3b is a 3-address machine. The two registers containing source operands and the register that the result will be stored in are all identified in the LC-3b instruction. For example ADD R0, R1, R2. Zero address means NONE of addresses are explicitly identified in the instruction. An ADD instruction is simply: ADD. How will the microarchitecture know where to find the source operands and where to store the result of the ADD?

Explain in 15 words or fewer: Operands are taken off the stack and the result is pushed back onto the stack

Part b (5 points): An n-wide issue microarchitecture is one in which n instructions are fetched from the Instruction Cache every cycle. ...or, at least that is the hope. ...and a good deal of the time, n useful instructions are fetched from the Instruction Cache in the same cycle. Not always! Three things can get in the way of n useful instructions being fetched in a single cycle. Name 2 of them.

Explain in 10 words or fewer: Branches. Taken branches cause the following instructions to not be useful.	Explain in 10 words or fewer: iCache misses. Requires going to main memory to fetch new instruction bytes.

Part c (10 points): Recall my 6-bit floating point data type that I used in class to show the characteristics of the IEEE Floating Point Standard. Can the number 11 $\frac{1}{2}$ be represented exactly with my 6-bit floating point data type? If yes, show me the six bit representation. If not, explain why not?

No. 11.5 requires 4 fraction bits to be represented exactly 1.0111 * 2^3

If 11 $\frac{1}{2}$ can not be represented exactly with my 6 bit data type, how would I represent 11 $\frac{1}{2}$ inexactly with each of the 4 rounding modes?

Rounding Mode: To Zero Value: 1.01*2^3	Rounding Mode: Unbiased nearest even Value: 1.10*2^3	Rounding Mode: up Value: 1.10 * 2^3	Rounding Mode: Down Value: 1.01 * 2^3

Name: _____

Problem 2 (15 points): The memory system for an LC-3b machine with vector capabilities has one channel and one rank. Latency to the DRAM arrays is 5 cycles. The memory is **byte addressable**, and has a 2 byte data bus. The machine is connected to 64KB of physical memory.

Part a (5 points) How many bits of bank address does the machine need to maximize performance?

3

Part b (5 points): We wish to execute VLD V1, A, where VLD is a vector load instruction, V1 is a vector register, VSTRIDE is 1, VLENGTH is 12, A is memory address 0x0300, and every element is a 16-bit word. What bits of the address should be bank address bits if we wish to minimize the latency of executing the VLD instruction?

Your job: Identify which bits of the address are bank bits. Put an x in each bank bit of the address.

												x	x	x	

Part c (5 points) We determine that the machine has 256 rows. Given the number of banks from part A, how many columns does the machine have?

16

Name: _____

Question 3 (15 Points): An LC-3b system, with a 16-bit physical address space, has a 1 KB physically indexed, physically tagged data-cache. The cache is 2-way set associative, and uses perfect LRU replacement. A cache line is 16 Bytes.

We execute the following code snippet on the machine:

- `sum` and `i` are kept in registers

```
char A[1024];
int sum = 0;
for(int i = 0; i < 1024; i = i + 1) {
      sum = sum + A[i];
}
```

Note: each char is 1 byte.

Part a (5 points) Show the address layout. Use "I" for index, "T" for tag, and "O" for offset into the cache line.

T	T	T	T	T	T	T	I	I	I	I	I	O	O	O	O

Part b (4 points) What is the cache hit ratio of the data cache?

15/16

Part c (3 points) Does the cache hit ratio change if we change the associativity of the cache to 16-way? Yes/No (Circle one). If yes what is the new cache hit ratio?

NO

Part d (3 points) Does the cache hit ratio change if we change the line size of the cache to 64B but maintain the size of the cache? Yes/No (Circle one). If yes what is the new cache hit ratio?

YES 63/64

Name: _____

Problem 4 (30 points): An out-of-order processor executes its instructions according to the Tomasulo algorithm devised by Robert Tomasulo.

- Fetch and Decode take one cycle each.
- Register Rename occurs during decode.
- The ADD functional unit takes 3 cycles, at the end of which the result is stored in an output buffer.
- The MUL functional unit takes 5 cycles, at the end of which the result is stored in an output buffer.
- The DIV functional unit takes 9 cycles, at the end of which the result is stored in an output buffer.
- There is one of each functional unit. The ADD and MUL are pipelined, but DIV is not pipelined.
- The reservation stations are allocated in a top-down circular manner. Reservation stations are allocated at the end of decode and deallocated at the end of writeback.
- If two instructions are ready to dispatch to the same functional unit in the same cycle, the older instruction is dispatched and the younger instruction waits. If multiple instructions are ready to dispatch to different functional units in the same cycle, all of them may be dispatched.
- One cycle is needed to write the result to a destination register. Only one result can be written in a single clock cycle. If two instructions want to write results in the same clock cycle, the older instruction writes, and the younger instruction waits.
- If the machine supports reservation station bypass, an instruction can start execution immediately after decode if it does not need to wait for any dependency. Otherwise the instruction must wait at least 1 cycle in the reservation station.
- If the machine supports data forwarding, an instruction can start execution during the writeback cycle of the instruction it is dependent on. Otherwise the dependent instruction must wait until the cycle after writeback to start execution.
- SR1 will populate the left side of the reservation stations and SR2 the right.
- For DIV, SR1 is the dividend and SR2 is the divisor. (DR = SR1/SR2)

A friend used this processor to profile some code and got the following results.

```
ADD R0, R1, R1
MUL R0, R1, R1
```

Takes 9 cycles

```
ADD R0, R1, R1
MUL R2, R0, R0
```

Takes 11 cycles

Each correct answer earns you 1 point. Each incorrect answer is minus 1 point. Leaving an answer blank earns 0 points.

Part a (1 points). Does the machine support reservation station bypass? **YES**

Part b (1 points). Does the machine support data forwarding? **YES**

Name: _____

Part c (23 points). Program A shown below finishes processing by the end of CYCLE 16.

Your job: Fill in the missing fields in Program A.

I1	ADD	R7	R6	R5
I2	MUL	R6	R2	R3
I3	ADD	R1	R0	R6
I4	DIV	R3	R4	R0
I5	MUL	R5	R2	R6
I6	ADD	R0	R1	R4
I7	ADD	R3	R2	R4

Snapshots of the Register Alias Table and the reservation stations for the ADD, MUL, and DIV functional units are shown below, (a) before I1 is fetched, (b) after cycle 6, and (c) after cycle 11.

Your job: Fill in the missing entries identified in boldface.

	V	Tag	Value
R0	1	-	1
R1	1	-	10
R2	1	-	22
R3	1	-	3
R4	1	-	40
R5	1	-	50
R6	1	-	60
R7	1	-	70

Before I1 is fetched

	V	Tag	Value
R0	1	-	1
R1	0		10
R2	1	-	22
R3			
R4	1	-	40
R5	0		
R6	0		
R7		α	

After cycle 6

	V	Tag	Value
R0	0	γ	1
R1	0		10
R2	1	-	22
R3			
R4	1	-	40
R5	0		
R6	1		
R7		α	

After cycle 11

		V	Tag	Value	V	Tag	Value
ADD	α	1	-	60	1	-	50
	β	1	-	1	0		
	γ						
MUL	σ	1	-	22	1	-	3
	ρ	1	-	22	0		
	π						
DIV	ϕ	1					
	θ						
	τ						

After cycle 6

		V	Tag	Value	V	Tag	Value
ADD	α	1	-	22	1	-	40
	β	1	-	1	1		
	γ	0			1	-	40
MUL	σ	1	-	22	1	-	3
	ρ	1	-	22	1		
	π						
DIV	ϕ	1					
	θ						
	τ						

After cycle 11

Name: _____

Part d (5 points): For program A, if R0 contained 0x0000 initially, would bad things happen without a reorder buffer? Explain in 20 words or fewer.

Yes. Divide by zero exception will occur after older instructions have already retired meaning no precise exceptions.

We have provided templates for you to use in solving this problem. They are for your use only. **They will not be graded**. Use F for Fetch, D for Decode, RS for Reservation Station Allocation, A for ADD, M for MUL, Di for Divide, and Register number (R#) for WRITEBACK, and * if no progress is made during that clock cycle for that instruction.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
I1																
I2																
I3																
I4																
I5																
I6																
I7																

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
I1																
I2																
I3																
I4																
I5																
I6																
I7																

Name: _____

Name: _____

Problem 5 (30 Points): Consider the following specification for a machine.

- LC-3b with VAX-like 2 level virtual memory
- 64KB of physical memory, 2 Bytes for each PTE
- Virtual memory is partitioned into 2 regions: User space starts at x0000, system space at x8000
- SBR points to the start of a frame. Every PBR points to the start of a page.
  Note: A page table may take more than one page/frame
- The machine includes a 4 entry fully associative TLB using perfect LRU replacement. **The TLB contains PTEs of both user pages and system pages.**
- The TLB contains both Data and Instruction access mappings
- Although unrealistic, assume pages are evicted from memory to disk using perfect LRU.
- The machine can process programs in one of two modes, based on the page size. The page size is set during boot time (when the operating system is configured), and can not be changed while the machine is running. We refer to the two page sizes as huge-pages and base-pages. Huge pages are larger than base pages.
- The machine switches between processes (i.e. performs a context switch) every 4 instructions. This means saving the state of the machine, and then loading the next process. This is done using a set of hardware registers and does not involve memory.

The virtual address has the following format

Region	VPN	Offset

We perform an experiment on the machine with only two processes resident. Each process executes the corresponding code segment shown below:

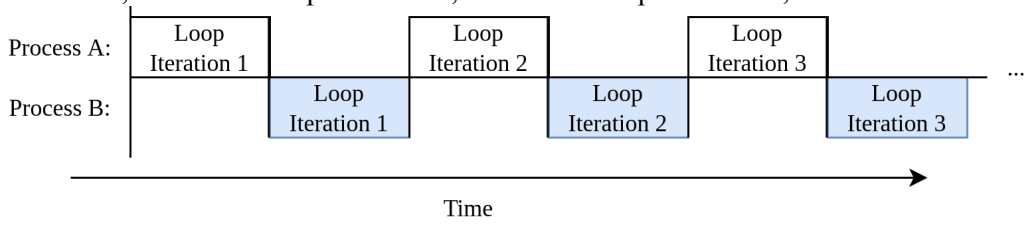
Process A

```
      ; R1 contains x4100
      ; R2 contains x00C8 (#200)
      .ORIG x4000
LOOP  STB R2, R1, #0
      ADD R1, R1, #1
      ADD R2, R2, #-1
      BRp LOOP ; switch to process B
      HALT
```

Process B

```
      ; R1 contains x4100
      ; R2 contains xFF38 (#-200)
      .ORIG x4000
LOOP  STB R2, R1, #0
      ADD R1, R1, #1
      ADD R2, R2, #1
      BRn LOOP ; switch to process A
      HALT
```

The code executes as follows: Process A Loop-iteration 1, Process B Loop-iteration 1, Process A Loop-iteration 2, Process B Loop-iteration 2, Process A Loop-iteration 3, etc.



Both processes start execution at virtual address x4000. Each code segment fits in one page.

We perform the experiment twice, once when the mode is base-page size and once when the mode is huge-page size. With the page size set to base-page, 1200 TLB misses occur. With page-size set to huge-page, 4 TLB misses occur.

Your job: Answer the questions on the next page.

Name: _____

Part a (6 points). What is the largest possible page size in base-page mode?

256

Part b (6 points). What is the smallest possible page size in huge-page mode?

512

Part c (6 points). IF we have 2KB of physical memory, how many times would we have to evict a page to disk during the execution of the two threads using the page size of part a?

0

Part d (6 points). IF we have 2KB of physical memory, how many times would we have to evict a page to disk during the execution of the two threads using the page size of part b?

797

Part e (6 points). The system is executing in huge-page mode with 64KB of physical memory. The first iteration of process A makes the following physical accesses in no particular order:

x0800, x0802, x0804, x0806, x0900, 0x1040, 0x4002

Some time later, an iteration of process B make the following physical accesses in no particular order:

x0400, x0402, x0404, x0406, x05A1

Your job: Fill in the following blanks

SBR **x4000**

Process A PBR **x8200**