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

EE 460N Fall 2022
Instructor: Yale N. Patt
TAs: Kayvan Mansoorshahi, Michael Chen
Final Exam
December 9, 2022

Name: SOLUTIONS

Part A:

Problem 1 (15 points):_____

Problem 2 (15 points):_____

Problem 3 (15 points):_____

Problem 4 (15 points):_____

Part B:

Problem 5 (35 points):_____

Problem 6 (35 points):_____

Total (130 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!

Name:_____

Problem 1 (15 points):
We wish to design a 128KB on-chip N-way set associative physical cache to support a virtual memory system organized into 8KB pages. We want to be able to access the TLB, the tag store, and the data concurrently. We want a 32 byte line size. We want to have the maximum number of sets possible.

What is N, the associativity of the cache. Please show your work.

16 WAY

CACHE SIZE = $2^{\text{INDEX}} \times N \times 2^{\text{LINE SIZE}}$

128KB = 2^{17}

32 BYTES = 2^5

FRAME NO.	8 KB PAGE

CONCURRENT ACCESS → 13 BITS FOR INDEX AND LINE

$13 = \text{INDEX} + 5$

$\text{INDEX} = 8$

$\therefore 2^{17} = 2^8 \times N \times 2^5$

$2^4 = N$

Name:_____

Problem 2 (15 points):
You are working for HotShot Software Company #10 as a consultant. They have a critical piece of code that they need to run in 10 ms or faster. Through previous experiments they know the code is 60% parallelizeable and takes 20 ms to run on 1 processor. Assume the parallelizeable part of the code scales perfectly with more processors.

Part a (7 points): At least how many processors does it take to meet their performance goal of 10 ms? Show your work.

$\frac{20}{10} = 2 = \frac{1}{\frac{.6}{p} + (1-\alpha)}$ = 6 processors

Part b (8 points): How much speed up is possible given infinite processors? Show your work.

$\frac{1}{\frac{.6}{\infty} + (1-.6)} = \frac{1}{.4} = \frac{10}{4} = 2.5$

Name:_____

Problem 3 (15 points):

You are the chief architect on a new Out of Order machine with the following specification:

- The machine has a five stage pipeline (FETCH, DECODE, REGISTER RENAME, EXECUTE, WRITEBACK). All stages except EXECUTE take one clock cycle each.
- Tomasulo's original algorithm (no ROB).
- Instructions with no dependencies can start executing immediately after REGISTER RENAME, and still allocate a reservation station.
- Each functional unit has enough reservation stations to handle any reasonable demand.
- There are two functional units. A pipelined adder that takes 2 cycles, and a pipelined multiplier that takes 4 cycles.
- Entries are put into reservation stations at the end of REGISTER RENAME and removed at the end of WRITE-BACK.
- The result of a functional unit is broadcast during WRITEBACK. Any dependent instructions can begin executing in the next cycle.
- The write-back bus supports only one result being stored at a time. Earlier instructions take priority for WRITE-BACK.

Analysis shows that the following snippet of code is most important for your customers.

1	MUL R1, R2, R3
2	ADD R6, R1, R3
3	ADD R4, R1, R6
4	MUL R2, R2, R5
5	ADD R2, R1, R2

Your job: Fill in the timing diagram below for the code given above. Use F for FETCH, D for DECODE, R for REGISTER RENAME, A for the add, M for multiply, and WB for WRITEBACK. Use - to indicate waiting in a reservation station, and * to indicate a stall that cycle.

Instruction	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
	F	D	R	M	M	M	M	WB												
		F	D	R	-	-	-	-	A	A	WB									
			F	D	R	-	-	-	-	-	-	A	A	WB						
				F	D	R	M	M	M	M	*	WB								
					F	D	R	-	-	-	-	-	A	A	WB					

Name:_____

Problem 4 (15 points):

The following sentences pertain to a physical memory consisting of one megabyte of DRAM. Three successive load instructions requests data from DRAM memory.

```
Address of 1st load: 10110110000000010101
Address of 2nd load: 11000110100010100111
Address of 3rd load: 11000110111000110101
```

1. There is only one channel and within that channel only one rank.
2. Each chip provides one byte of memory.
3. There are 16 banks → 4 BITS FOR BANK
4. Each load results in 32 bytes of memory provided on an 256 bit bus. → 5 BITS OF B ON B
5. Row address bits are contiguous.
6. Column address bits are contiguous.
7. Bank address bits are contiguous.
8. The second load instruction must wait for the first load to complete before it can start. 1ST, 2ND; SAME BANK
9. The memory controller does not have to supply the third load instruction's row address bits. 2ND, 3RD, SAME BANK, SAME ROW

Your job: Identify the address bits used to specify the bank, row, column, etc.

Bank:	Row:	Column:	Byte on bus:
15:12	19:16	11:5	4:0

ONLY PLACES WITH 4 CONTIGUOUS BITS OF ADDR. IDENTICAL FOR ALL 3 ADDRESSES

ROW BITS SAME FOR 2ND 3RD ADDRESSES

ONLY BITS LEFT

STATEMENT 4 ABOVE

Name:_____

Problem 5 (35 points):
On Exam 2, we extended the LC-3b with VAX-like virtual memory. Its characteristics are repeated below:

- 16-bit virtual addresses.
- The memory management system uses the two-level page table scheme like the VAX.
- Virtual memory is partitioned into two halves. User space starts at x0000 and system space starts at x8000.
- There are 16KB of physical memory.
- The page size is 256 bytes.
- The user page table starts at the beginning of the page. The system page table starts at the beginning of a frame.
- LC-3b is little endian.
- A PTE contains 16 bits and has the following form.

PTE:

Valid	Read Permission	Write Permission	0s	PFN
1 bit	1 bit	1 bit		

We added a new instruction **STDW R0,R1,R2** that stores a double word (32 bits) into four consecutive bytes of memory. R2 contains the starting address of the four bytes of memory. R0 contains the value written into the first two bytes. R1 contains the value written into the second two bytes.

For example, if R2 contains x3000, R0 contains x1234, and R1 contains x5678, **STDW R0,R1,R2** will store x1234 into x3000 and x3001, and will store x5678 into x3002 and x3003.

Part a (6 points): What is the maximum number of physical memory accesses required to perform STDW? Ignore instruction fetch. Explain, use 20 words or fewer.

6 accesses. VAX requires 3 accesses per load/store.

Problem 5 (Continued)

Part b (17 points): Suppose we add a 2 entry, fully associative TLB, that only maps user pages, with LRU replacement. Let's consider one instance of the execution of STDW R0, R1, R2. Below is a snapshot of the registers, the TLB, and a portion of physical memory. All locations required by STDW R0, R1, R2 are shown. Addresses are physical addresses:

Register	Contents
R1	x1107
R2	x5FFE

V	VPN	PTE
1	x44	xE015
0	x5F	xC023

Registers **before**

Address	Contents
x31FE	x01
x31FF	x19
x3200	x07
x3201	x11
x3202	x04
x3203	x16

Address	Contents
x352C	x24
x352D	x00
x351E	x12
x351F	x40
x3520	x12
x3521	xC0

TLB **before**

Address	Contents
x12BC	x70
x12BD	xC0
x12BE	x08
x12BF	xE0
x12C0	x32
x12C1	xE0

Address	Contents
x08FE	x09
x08FF	x30
x0900	xA4
x0901	xF5
x0902	x07
x0903	x11

Memory **After**

After the instruction is fetched and decoded, multiple memory access could be required to process this instruction. Assume no exceptions occur during translation.

Your Job: Complete the table and TLB below based on the given machine state shown above. Use as many blank rows as you need. Draw a line through unused rows.

Access #	VA	PA	Data	TLB Hit	R/W
1	—	x3520	xC012	NO	R
2	x90BE	x12BE	xE008	NO	R
3	x5FFE	x08FE	x3009	NO	W
4	—	x3520	xC012	NO	R
5	x90C0	x12C0	xE032	NO	R
6	x6000	x3200	x1107	NO	W
7					
8					

V	VPN	PTE
1	x60	xE032
1	x5F	xE008

Table 1: TLB

Part c (12 points): Fill in the following values:

R0 x3009 PBR x9000 SBR x3500

Name:_____

Problem 6 (35 points):
A student in class is testing his Lab 3 LC-3b simulator with the following program.

```
        .ORIG x3000
        AND R0, R0, #0
        ADD R1, R2, R6
        LSHF R3, R4, #4
        BRnz DEST1
        LEA R5, NUM1
        JSR DEST2
DEST1   LEA R5, NUM2
DEST2   LDW R6, R5, #0
        HALT
NUM1    .FILL x1234
NUM2    .FILL x5678
        .END
```

The contents of the general purpose registers before and after executing the program are shown below.

Register	Value
R0	x3829
R1	x54A7
R2	x6329
R3	x08FC
R4	x5BE6
R5	x2398
R6	x1754
R7	xE5F0

Before executing the program

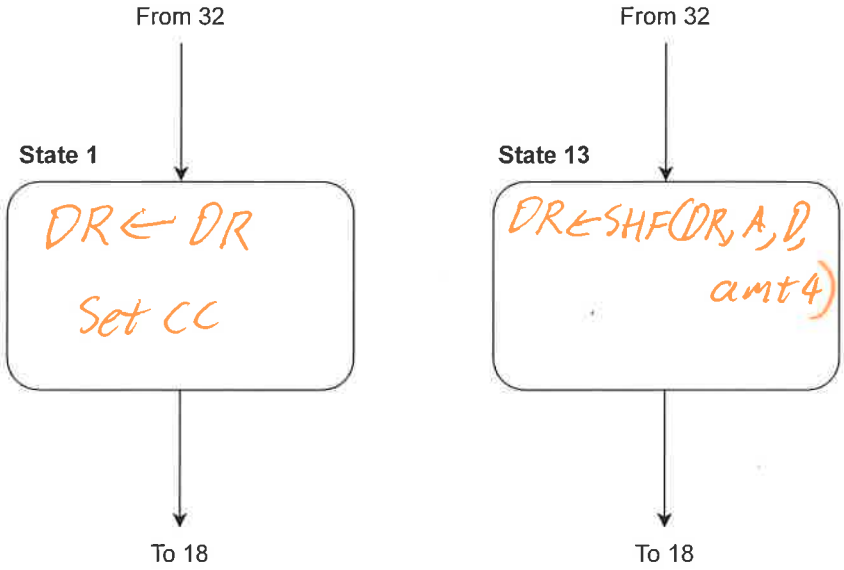
Register	Value
R0	x0000
R1	x54A7
R2	x6329
R3	x8FC0
R4	x5BE6
R5	x3012
R6	x6D40
R7	x3012

After executing the program

If you examine the registers before and after execution, you can see that something is broken. The problem is due to incorrect control signals implementing states 1,6, and 13 in the state machine. Furthermore, we have been told that the only control signals that can be incorrect are among the following: **LD.CC**, **LD.MAR**, **SR1MUX**, and **ALUK**. Given this information, you can find the incorrect control signals.

Name:_____

Part a (17 points): Specify what is actually happening in states 1 and 13 below.



Part b (18 points):
The table below shows the relevant piece of the control store that is causing the incorrect behavior, i.e., the control store values for four control signals in each of three states.

Your job: Fill in the values for these 12 entries in the piece of the control store shown in the figure.

Note: Write "X" if the value is a don't care.

	LD.CC	LD.MAR	SR1MUX	ALUK
State 1	1	?	0	11
State 6	?	0	?	X
State 13	0	?	0	X

? => cannot be determined for given program. either answer accepted