The University of Texas at Austin
EE460N/EE382N.1 Exam 1 (Fall 2015)

Mattan Erez, Alex Hsu, Tommy Huynh, Anoop Naravaram, Kishore Punniyamurthy

Your signature is your promise that you have **not cheated** and will not cheat on this exam, nor will you help others to cheat on this exam. Cheating means exchanging any information whatsoever with any other entity (with the exception of private searches). **Plus -- no typing!**

Signature: | Name: | EID:

Note: Please be sure that your answers to all questions (and all supporting work that is required) **are contained in the box provided for each question;** we reserve the right to ignore anything else.

Note: **"I DON'T KNOW"** is a valid answer that automatically gives you 15% of the maximum possible grade. Be sure to either erase, or cross-out everything else and write I DON'T KNOW in, large, capital letters. A blank answer does not get you the 15%. No partial I DON'T KNOW credit (only on entire questions/parts that have a marked point value on the exam)

Note: Please put your name at the top of each page of the exam!

Note: For all questions, unless otherwise stated, find the most efficient solution. Efficiency counts.

The exam has a total of 4 questions and 5 pages. It's a good idea to read through first some questions might be easier for you than others.

For each page that doesn't include your name and eid you will be deducted 1 point!

Question 1 (36 points total):

A network SoC consists of an LC-3b processor core, its private cache, a network interface (NI), and a memory controller that allows both the LC-3b and the NI to access a common main memory. All are connected via a coherent bus using the snooping MSI protocol. The processor cache is write-back, write-allocate, and its capacity is 128B.

Operation notation

```
RD/LD x4000          : read / load the value at memory location x4000
WR/ST x1, x4000      : write / store the value 1 into memory location x4000
```

Hints:

- Remember that you solve this question by posing and testing hypotheses for what different parameters might be.
- It is easier to test some hypotheses than others (some have fewer parameters to explore).
- You are required to get a consistent answer, not to prove that it's unique.

<<<Continued on next page>>>

Name: EID:

The cache is empty initially. You are given the following trace:

#	Operation	Hex Address	Binary Address	Status
1	LD	x3000	0011 0000 0000 0000	Miss
2	ST xFF	x3064	0011 0000 0110 0100	Miss
3	LD	x39021	0011 1001 0000 0010	Miss
4	LD	x3A08	0011 1010 0000 1000	Miss
5	**I/O WR x42**	x3A08	0011 1010 0000 1000	-
6	ST x66	x3010	0011 0000 0001 0000	Hit
7	**I/O RD**	x3010	0011 0000 0001 0000	-
8	ST xFE	x3020	0011 0000 0010 0000	Miss
9	ST xAB	x3166	0011 0001 0110 0110	Miss
10	**I/O WR x21**	x3910	0011 1001 0001 0000	-
11	LD	x3002	0011 0000 0000 0010	Miss
12	LD	x3066	0011 0000 0110 0110	Hit

Part a (24 pt): What are the cache parameters?What are the cache parameters?

i. Block size = __32_____ Bytes

ii. Type (circle one): ~~direct-mapped~~ ~~set-associative~~ **fully associative**

iii. Associativity = __4____ ways (write "n/a" if direct mapped)

iv. Replacement policy (circle one): ~~LRU LFU LIFO~~ **FIFO** N/A (N/A if direct-mapped)

Part b (4 pt): What is the cache function (i.e., what bits comprise the tag, the index, and the offset)?

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Tag											**Offset**				

Part c (8 pt): Show the content of the cache tag and the cache maintenance information (required to maintain the operation of the cache described) for the accessed cache line after each of the following operations; also show the content of main memory at that time. Before these operations, each memory location contains the least significant byte of its address (i.e. M[xABCD] = xCD)

Access # (from above)	Tag	Maintenance	Main Memory
4	**b 0011 1010 000**	**State:S FIFO:3**	M[x3A08] = **x08**
5	**b 0011 1010 000**	**State:I FIFO:3**	M[x3A08] = **x42**
6	**b 0011 0000 000**	**State:M FIFO:0**	M[x3010] = **x10**
7	**b 0011 0000 000**	**State:S FIFO:0**	M[x3010] = **x66**

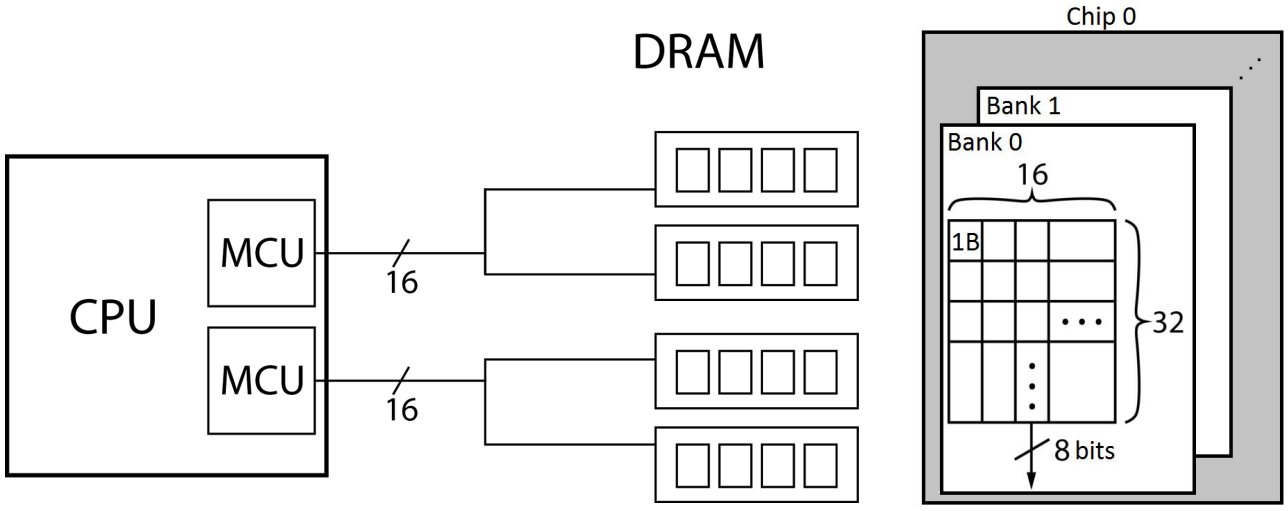
Name: EID:

Question 2 (10 points total):

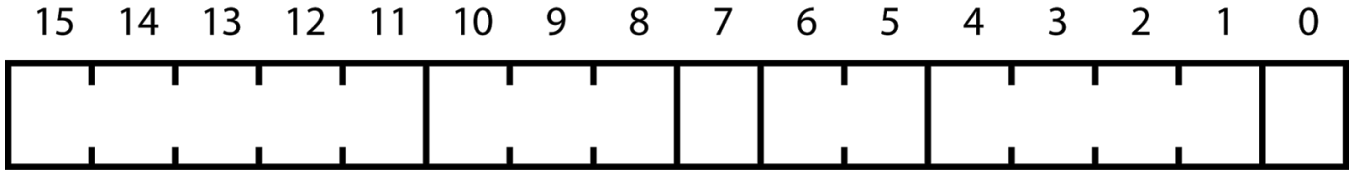
Your colleague at Little Computer Inc. designed a cool new byte-addressable memory system with a 16-bit address space and gave you the following memory diagrams. Unfortunately, the fields in the bit-swizzling went missing. Complete the documentation.

Diagrams:

MCU = memory controller unit and is responsible for scheduling and sending commands to DRAM chips.



Bit-swizzling:



Note: Fields are all contiguous.

ROW	BANK	CH	RANK	COLUMN	N/A

Question 3 (22 points total):

You recently got hired to work on the memory subsystem for a 32-bit, byte-addressable processor. You also found the following information in the documentation:

- 1 GHz memory bus
- DDR
- Burst length = 4
- Time between precharge command and activate command = 4 ns
- Time between activate command and column command = 4 ns
- Time between a column command and data becomes ready for transfer = 2 ns

...	11	10	9	8	7	6	5	4	3	2	1	0
row		column			bank				?		n/a	

Part a (5 pt): What are bits 2 and 3 of the address mapped to in DRAM address? If you have a hunch but are not sure if your answer is legal, ask us.

Column

Part b (12 pt): Calculate the time it takes to read the following patterns (assume all accesses are ready to be made at time 0 and that the memory controller does its best to minimize time):

Access pattern	Time (ns)
16, 32, 64 (b0001 0000, b0010 0000, b0100 0000)	**16ns**
256, 512, 1024 (b0001 0000 0000, b0010 0000 0000, b0100 0000 0000)	**16ns**
2048, 4096, 8192 (b0000 1000 0000 0000, b0001 0000 0000 0000, b0010 0000 0000 0000)	**36ns**

Part c (9 pt): You tested your calculation on the real system and observed that the time decreased by 3ns or more per pattern. What single parameter (other than clock frequency) could be wrong in the documentation and what value could it be instead (no value can be 0ns)? **Short answer! It's possible that there are multiple correct answers.**

Activate or precharge taking only 1ns or burst of only 2 transfers.

Name: EID:

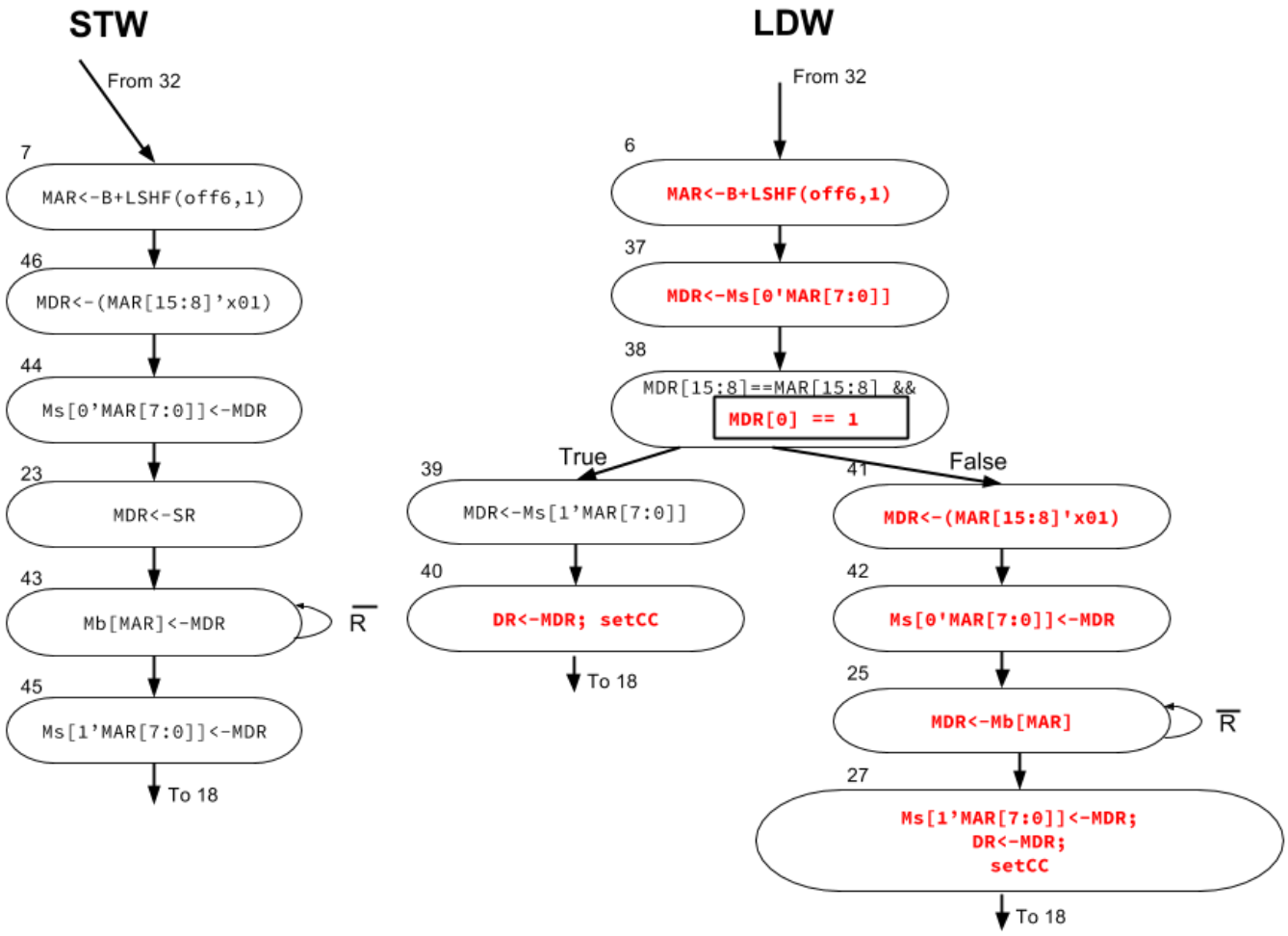
Question 4 (32 points total):

You are given a modified LC3b microarchitecture that includes two memories instead of one (both are attached to the same bus and share the single MAR and MDR in the datapath). One memory, Mb, is 64KiB and has about a 100-cycle access latency for reads and writes. The second, Ms, is 512B and has a 1-cycle latency (and no need for the $\overline{R}$ signal). Both use a 16-bit interface. The control path may choose which memory is accessed by setting an appropriate control bit to that selects between Mb and Ms. You are also told that LDB, LDW, STB, and STW have been modified. **You are also given the modification to STW below.**

Part a (22 pt):

Fill in the microcode for LDW using pseudo-microcode in the same style as the LC3b state diagram. Note that you may not add or remove any states and each state must include some operation.

Note: the notation `x1234'56` represents concatenation `x123456`.



Part b (10 pt):

What is this modification implementing? Be as specific as you possibly can (i.e., if there any characteristics or parameters, specify them).

Ms is a 256B direct-mapped, write-allocate, write-through, data cache with 2B cache lines; it is a data cache because the question says only LDB, LDW, STB, and STW are modified and not the instruction fetch state. BIG problem here. This was meant to be 256B (half of Ms was supposed tobe data and half tag array), but it's actually 256 entries here and 2B each, which doesn't add up :-(