Department of Electrical and Computer Engineering

The University of Texas at Austin

EE 460N, Spring 2022

HW 1

Due: Feb 18th 11:59 PM

Mattan Erez, Instructor
Chet Pena, Divija Gogineni, Ying-Wei Wu, TAs

By having this homework submitted with my name selected, I acknowledge that I contributed meaningfully to (most) of the solutions to problems in this homework set and did not simply do only some subset of the problems.

You are welcome to do assignments in groups. However, make sure everyone in your group both contributes and is on top of the entire assignment (splitting up the work is not allowed).

Make sure you hand in clearly-written text and clearly-drawn figures. We will use online submission and grading through gradescope, so make sure you find a scanner or download a good scanner app for your phone (and find somewhere with good lighting to scan). I like Scanbot, which works well and is free. You can use CamScanner mobile app too. If we can't easily read an assignment we will not grade it!

Because of the poor student-to-TA ratio, we will likely sample-grade. This means we will grade a subset of the assigned problems in every assignment; the same subset will be graded for all who submit

Problem 1

The following program computes the square (k*k) of a positive integer k, stored in location `0x4000` and stores the result in location `0x4002`. The result is to be treated as a 16-bit unsigned number.

Assumptions:

- A memory access takes 5 cycles
- The system call initiated by the `HALT` instruction takes 20 cycles to execute. This **does not** include the number of cycles it takes to execute the `HALT` instruction itself.

```
        .ORIG X3000
        AND R0, R0, #0
        LEA R3, NUM
        LDW R3, R3, #0
        LDW R1, R3, #0
        ADD R2, R1, #0
LOOP    ADD R0, R0, R1
        ADD R2, R2, #-1
        BRP LOOP
        STW R0, R3, #1
        HALT
NUM     .FILL x4000
        .END
```

1. How many cycles does each instruction take to execute on the LC-3b microarchitecture described in Appendix C? Link to the handout: https://drive.google.com/drive/u/1/folders/1EAHMpBerYzKperNWq-nxK7Awyk6Md2ZB
2. How many cycles does the entire program take to execute? (answer in terms of k)

Problem 2

Consider the following possibilities for saving the return address of a subroutine:

1. In a processor register.
2. In a memory location associated with the subroutine; i.e., a different memory location is used for each different subroutine.
3. On a stack.

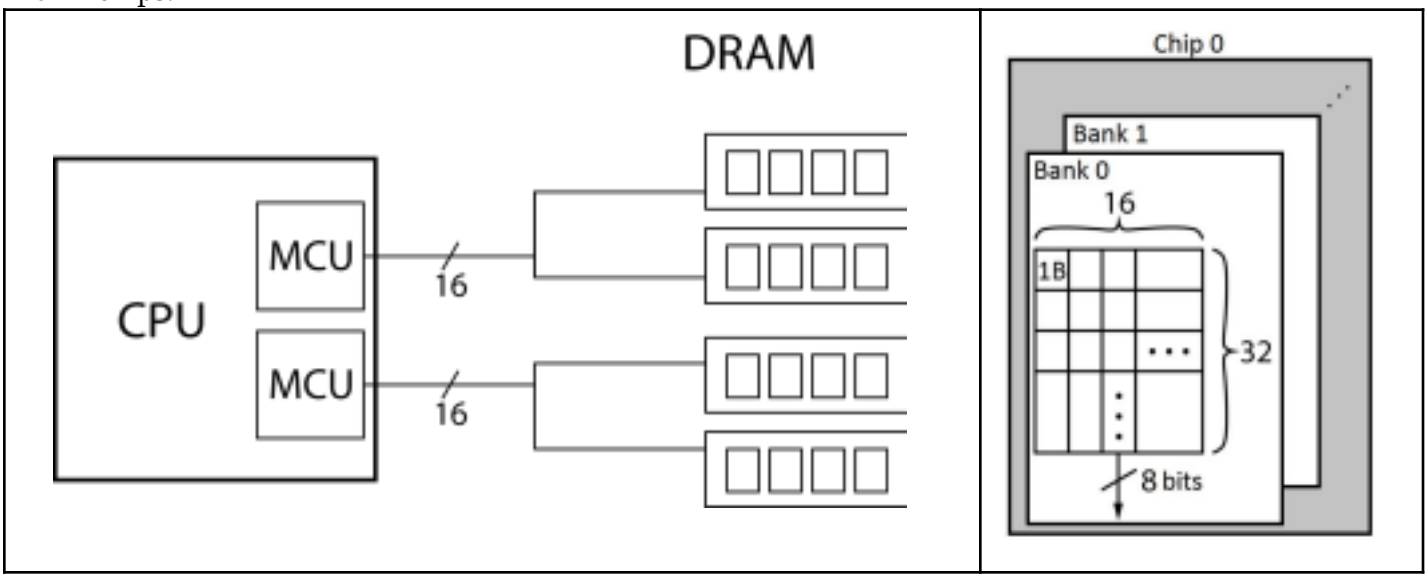
Which of these possibilities supports subroutine nesting, and which supports subroutine recursion (that is, a subroutine that calls itself)?

Problem 3

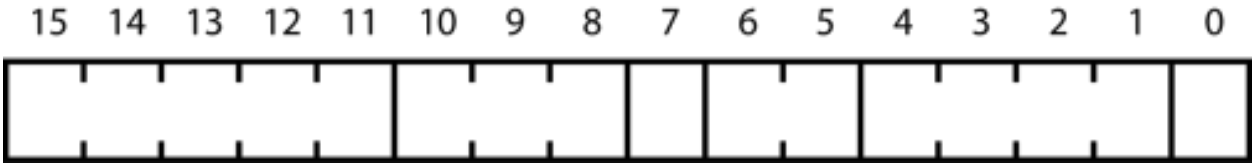
Your colleague at Little Computer Inc. designed a cool new byte-addressable memory system with a 16-bit address space and gave you the following memory diagrams. Unfortunately, the fields in the bit-swizzling went missing. Complete the documentation.

Diagrams:

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



Bit-swizzling:



Note: Fields are all contiguous.

Problem 4

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 the DRAM address?

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)	
256, 512, 1024 (b0001 0000 0000, b0010 0000 0000, b0100 0000 0000)	
2048, 4096, 8192 (b0000 1000 0000 0000, b0001 0000 0000 0000, b0010 0000 0000 0000)	

Part c (5 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.**