Homework 5

October 4, 2025

T1

What is the problem with the following LC-3 program? What may happen when it is executed? Give two possible way to fix the problem.

```
.ORIG x3000
        AND R1, R1, #0
PROGRAM ADD R1, R1, #1
        TRAP x23        ; IN
        LEA R0, MESSAGE
        TRAP x22        ; PUTS
MESSAGE .STRINGZ "Program reporting"
.END                    ; HALT
```

T2

In LC-3 programming practice, when using LC3Tools for running and debugging programs, you may observe the following phenomenon: after the program executes the `HALT` instruction and stops, the stored values in registers are significantly different from your expectations. Traditionally, we solve this issue by setting breakpoints.

> You may take a look at the following screenshot for reference.

Registers

R0	x0000	0	
R1	x7FFF	32767	
R2	x0003	3	
R3	x0004	4	
R4	x0005	5	
R5	x0006	6	
R6	x2FFE	12286	
R7	x0008	8	
PSR	x0002	2	CC: Z
PC	x036C	876	
MCR	x0000	0	

Memory

x3000	x5020	20512	*AND R0, R0, #0*
x3001	x1021	4129	*ADD R0, R0, #1*
x3002	x1221	4641	*ADD R1, R0, #1*
x3003	x1461	5217	*ADD R2, R1, #1*
x3004	x16A1	5793	*ADD R3, R2, #1*
x3005	x18E1	6369	*ADD R4, R3, #1*
x3006	x1B21	6945	*ADD R5, R4, #1*
x3007	x1D61	7521	*ADD R6, R5, #1*
x3008	x1FA1	8097	*ADD R7, R6, #1*
x3009	xF025	61477	*HALT*
x300A	x0000	0	

1) Now, please explain why this situation occurs.

2) The following shows a fragment of the HALT service routine. Under normal circumstances, will the instruction at address x036C be executed? If not, explain the purpose of this instruction.

Address	Instruction
x0366	*LEA R0, TRAP_HALT_MSG*
x0367	*PUTS*
x0368	*LDI R0, OS_MCR*
x0369	*LD R1, MASK_HI*
x036A	*AND R0, R0, R1*
x036B	*STI R0, OS_MCR*
x036C	*BRnzp TRAP_HALT*

T3

List at least three differences between TRAP and interrupt.

T4

1) What is system space?
2) What is user space?
3) Which memory addresses are inaccessible in user mode?
4) What is an ACV?
5) What determines the accessible memory range in the current state?
6) List all methods to transition from user mode to privilege mode.

T5

The following is an LC-3 program fragment. What is the value of register R0 after executing the code at label F.

> Note: you need to show how you arrived at your answer.

```
A   LD R0, E
    LEA R7, E
B   .FILL x1021
    ADD R2, R0, #0
    LD R1, D
    LD R3, B
    ADD R3, R1, R3
    ST R3, C
C   .BLKW 1
    RET
D   .STRINGZ "!"
E   LD R1, A
F   ADD R0, R0, R1
```

T6

During the execution of an LC-3 program, an instruction in the program starts executing at clock cycle T and requires 15 cycles to complete. The table lists ALL five clock cycles during the processing of this instruction, which require use of the bus. The table shows for each of those clock cycles: which clock cycle, the state of the state machine, the value on the bus, and the important control signals that are active during that clock cycle.

1) Fill in the missing entries in the table.

2) What is the instruction being processed?

3) Where in memory is that instruction?

4) How many clock cycles does it take memory to read or write?

5) There is enough information above for you to know the contents of three memory locations. What are they, and what are their contents?

Cycle	State	Bus	ImportantControl Signals for This Cycle
T	18	x3010	LD.MAR=1,LD.PC=1,PCMux=PC+1,GatePC=1
T+4			
T+6		x3013	
T+10		x4567	
T+14		x0000	LD.REG=1,LD.CC=1,GateMDR=1,DR=001

T7

Consider the following program:

```
        .ORIG x3000
        LD R0, A
        LD R1, B
        BRz DONE
        --------- (a)
        --------- (b)
        BRnzp AGAIN
DONE    ST R0, A
        HALT
A       .FILL x0--- (c)
B       .FILL x0001
        .END
```

The program uses only R0 and R1. Note lines (a) and (b) indicate two missing instructions. Complete line (c). Note also that one of the instructions in the program must be labeled AGAIN, and that label is missing.

After execution of the program, the contents of A is x1800.

During execution, we examined the computer during each clock cycle and recorded some information for certain clock cycles, producing the table shown below. The table is ordered by the cycle number in which the information was collected. Note that each memory access takes five clock cycles.

Cycle Number	State Number	Control Signals
1	18	LD.MAR: ☐ LD.REG: ☐ GateMDR: ☐ LD.PC: ☐ PCMUX: ☐ GatePC: ☐
☐	0	LD.MAR: ☐ LD.REG: ☐ BEN ☐ LD.PC: ☐ LD.CC: ☐
☐	☐	LD.REG: 1 DR: 000 GateMDR: ☐ GateALU: ☐ GateMARMUX: ☐
57	1	LD.MAR: ☐ ALUK: ☐ GateALU: ☐ LD.REG: ☐ DR: ☐ GatePC: ☐
77	22	ADDR1MUX: ☐ ADDR2MUX: ☐ LD.PC: ☐ LD.MAR ☐ PCMUX: ☐
101	15	

Fill in the missing instructions in the program, and complete the program by labeling the appropriate instruction AGAIN. Also, fill in the missing information in the table.

Given values for A and B, what does the program do?

T8

There are two parts of an element in a **singly linked list**: the data and a pointer to the next element. The data and the pointer are in adjacent memory locations, with the data preceding the next element pointer. They are both 16-bit integers. The address of a list element is the starting address of such element. Example below:

List with 4 elements

x4000	x0041
x4001	x4100
x4100	x0042
x4101	x5000
x5000	x0043
x5001	x5090
x5090	x0044
x5091	x0000

You need to implement the function below, which takes R0 as its only parameter. R0 is the address of an existing list element, and you need to delete such element from the list (the element is not the last element). You also need to output the element's data to the monitor (for convenience, we consider the data an 8-bit character and ignore other bits). When the function returns, R0 should carry the address of the next element of the deleted element. All other registers are caller saved.

> Note: the addresses of the elements are allowed to change.

```
____ ________#0
____ __________
TRAP x21
____ __, R2, __
____ R1, R0, __
____ R2, ______
____ ________#0
____ ________#1
____ R2, ______
RET
```