HW4

T1

Many ISAs have a conditional load instruction (LDC), which loads a value from memory into a register based on the condition codes. We could add that instruction to the LC-3 ISA using the unused opcode. Further we could use the BEN bit (BEN = (IR[11] AND N) OR (IR[10] AND Z) OR (IR[9] AND P)) the same way we use BEN to determine whether to take the conditional branch. The LDC instruction has three operands: DR, PC offset, and the nzp bits.

If a program contained an LDC instruction in memory location x4000, what is the largest memory address that can provide the value to be loaded into DR?

T2

List five addressing modes in LC3. Given instructions ADD, NOT, LEA,LDR and JMP, categorize them into operate instructions, data movement instructions, or control instructions. For each instruction mentioned above, list addressing modes that can be used

T3

In class we showed the first few states of the finite state machine that is required for processing instructions of a computer program written for LC-3. In the first state, the computer does two things, represented as:

```
MAR <-- PC
PC <-- PC+1
```

1. Why does the microarchictecture put the contents of the PC into the MAR?
2. Why does the microarchitecture increment the PC?

T4

1. Write a **single** LC3 assembly instruction that copies the content of `R5` to `R4`.
2. Write a **single** LC3 assembly instruction that clears the content of `R3` (i.e. `R3 = 0`).
3. Write **3** LC3 assembly instructions that does `R1 = R6 - R7`.
    - You are ONLY allowed to change the value of `R1`.
    - You may assume that the initial value of `R1` is 0.
4. Write **3** LC3 assembly instructions that multiply the value at label `DATA` by 2. (`Mem[DATA] = Mem[DATA] * 2`)
    - You are ONLY allowed to change the value of `R1`.
    - You don't need to restore or clear the value of the register you used.
    - No need to consider overflow.
5. Set condition codes based on the value of `R1` using only `one` LC-3 instruction.

- You are not allowed to change any value in the registers.

T5

How many times does the LC-3 make a read or write request to memory during the processing of the LD instruction?

How many times during the processing of the LDI instruction? How many times during the processing of the LEA instruction?

Also indicate what phases the instructions don't need. Processing includes all phases of the instruction cycle.

T6

Suppose we changed the LC-3 to have only four registers instead of 8. Fewer registers is in general a bad idea since it means loading from memory and storing to memory more often, but we can still ask the question: would there be any benefit to reducing the number of registers? For each of the following, answer yes or no, and explain your answer.

1. If we keep the basic format of all instructions as they currently are (and keep each instruction 16 bits), is there any benefit for operate (0001, 0101, 1001) instructions, if we reduce the number of registers to 4?
2. Is there any benefit for load (0010) and store (0011) instructions, if we reduce the number of registers to 4?
3. Is there any benefit for conditional branch (0000) instructions, if we reduce the number of registers to 4?

T7

Write a short LC-3 program that compares the two numbers in `R1` and `R2` and puts the value 0 in `R0` if `R1 = R2`, 1 if `R1 > R2`, and −1 if `R1 < R2`.

T8

The content in PC is x3010. The content of the following memory unit is as follows

Address	Value
x304E	x70A4
x304F	x70A3
x3050	x70A2
x70A2	x70A4
x70A3	x70A3
x70A4	x70A2
x3010	1110 0110 0011 1110
x3011	0110 1000 1100 0001

Address	Value
x3012	0110 1111 0000 0001
x3013	0110 1101 1111 1111

1. After the execution of the following code, What is the value stored in R6 ?
2. Can you use one LEA instruction to do the same task as the three instructions above do?(Only consider loading value into R6.)

T9

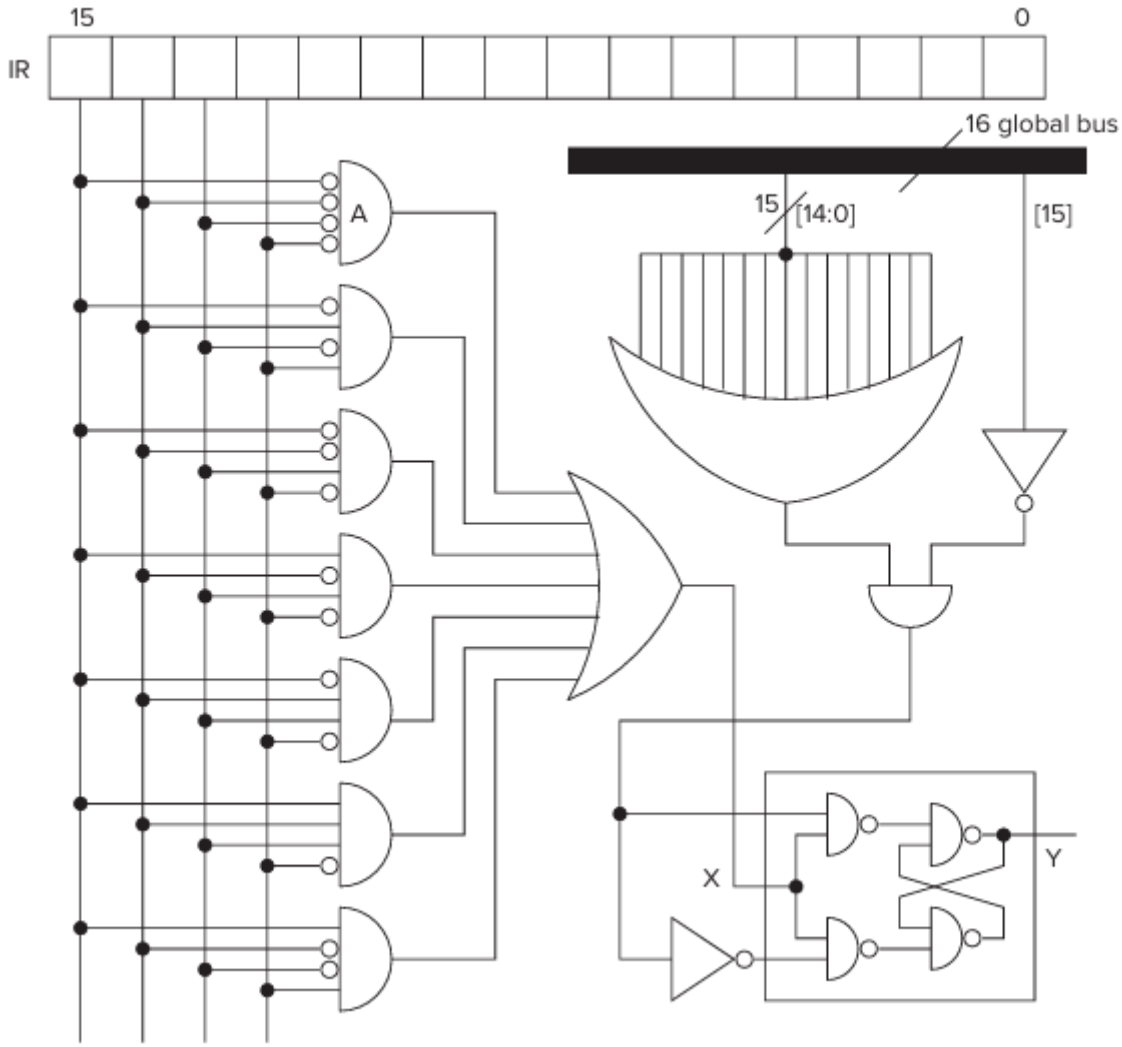
Shown below are the contents of registers before and after the LC-3 instruction at location x3210 is executed. Your job: Identify the instruction stored in x3210. Note: There is enough information below to uniquely specify the instruction at x3210

	Before	After
R0:	xFF1D	xFF1D
R1:	x301C	x301C
R2:	x2F11	x2F11
R3:	x5321	x5321
R4:	x331F	x331F
R5:	x1F22	x1F22
R6:	x01FF	x01FF
R7:	x341F	x3211
PC:	x3210	x3220
N:	0	0
Z:	1	1
P:	0	0

T10

Apart of the implementation of the LC-3 architecture is shown in the following diagram.

1. What information does Y provide?
2. The signal X is the control signal that gates the gated D latch. Is there an error in the logic that produces X?


15
0
IR
16 global bus
15
[14:0]
[15]
A
X
Y