Introduction to Computing: Homework 5

Fall 2025
Dr. Ramesh Yerraballi
The University of Texas at Austin
Due: Tuesday 10/21, 11:59pm

Instructions:

You may discuss the problem set solutions with your fellow classmates but the write up must be your own. Please use the TAs and the Instructor for help before you seek out a friend or classmate. Show your work. You may handwrite or type your answers.

Questions

1. (15 pts) This problem tests your knowledge of the instruction cycle for processing the AND instruction. You are asked to show the values of several control signals in every clock cycle of the sequence that is used to process the AND instruction. The instruction cycle starts with state 18 as shown in the following table.
Your job: Identify each state in the sequence, and show the values of the control signals listed during each state in the sequence. Use the convention specified below. For a particular state, if the value of a control signal does not matter, fill it with an X. You may not have to use all the rows.
Note: Assume a memory access takes one clock cycle.

Cycle	State	LD.PC	LD.MAR	LD.MDR	LD.REG	LD.CC	GateALU	GatePC	ALUK	PCMUX		
1	18											
2												
3												
4												
5												
6												
7												
8												
9												
10												

⇧
Use these extra columns if needed

```
LD.PC     0: load not enabled         GateALU   0: do not pass signal
          1: load enabled                       1: pass signal

LD.MAR    0: load not enabled         GatePC    0: do not pass signal
          1: load enabled                       1: pass signal

LD.MDR    0: load not enabled         ALUK      00: ADD
          1: load enabled                       01: AND
                                                10: NOT
LD.REG    0: load not enabled                   11: Pass input A
          1: load enabled
                                      PCMUX     00: PC+1
LD.CC     0: load not enabled                   01: BUS
          1: load enabled                       10: from adder
```

2. (8 pts)
   - What is an addressing mode?

   - List the five addressing modes of the LC-3.

   - What addressing mode(s) are used by the instruction x3F20?

   - What addressing mode(s) are used by the instruction x14B5?

3. (8 pts) How many times does the LC-3 make a **read or write** request to memory during the processing of the LD, STR, LDI, and BR instruction? Processing includes **all** phases of the instruction cycle.

4. (15 pts) An LDR instruction, located at x4950, uses R4 as its base register. The value currently in R4 is x6880.
   - What is the range of addresses that this instruction can load from?

   - Suppose we redefine the LDR offset to be zero-extended, rather than sign-extended. Then what would be the new range of addresses that this instruction could load from?

   - What would be the range of accessible addresses if this was an LD instruction with a signed offset?

5. (10 pts) (Adapted from 4.8) Suppose a 32-bit instruction has the following format:

OPCODE	DR	SR1	SR2	UNUSED

   If there are 250 opcodes and 100 registers, and every register is available as a source or destination for every opcode,

   a. What is the minimum number of bits required to represent the OPCODE?

   b. What is the minimum number of bits required to represent the Destination Register (DR)?

   c. What is the maximum number of UNUSED bits in the instruction encoding?

6. (14 pts) Suppose that an instruction cycle of the LC-3 has just finished and another one is about to begin. The following table describes the values in select LC-3 registers and memory locations:

Register	Value
IR	x3000
PC	x3002
R0	x3001
R1	x3429

Memory Location	Value
x3000	x5020
x3001	x2001
x3002	x6200
x3003	x3001

For each phase of the new instruction cycle, specify the values that PC, IR, MAR, MDR, R0, and R1 will have *at the end* of the phase in the following table:

	PC	IR	MAR	MDR	R0	R1
Fetch						
Decode						
Evaluate Address						
Fetch Operands						
Execute						
Store Result						

Hint: Take note of where the PC is pointing to. Remember to consider when MAR and MDR actually get modified.

7. (15 pts) You're given the memory *after* the following program has been executed. Fill in the missing entries.

Addr	Label	Instruction	Memory
x3000	—	LD R0, C	
x3001	—	AND R1, R1, #0	0101 001 001 1 00000
x3002	—		0001 001 001 1 00001
x3003	—	AND R2, R2, #0	
x3004	A	AND R3, R0, __	0101 011 000 0 00 001
x3005	—	BR_ B	
x3006	—	ADD __, R2, #1	0001 010 010 1 00001
x3007	B	ADD R1, R1, R1	0001 001 001 0 00 001

x3008	—	BR_ A	
x3009	—	ST R2, __	0011 010 000000010
x300A	—	HALT	1111 0000 00100101
x300B	C	.FILL x39FF	0011 1001 1111 1111
x300C	D	.BLKW #1	0000 0000 0000 0100

- What does this program do?

8. (15 pts) The PC is loaded with x3000 and 4 instructions are executed. The table below contains the contents of the various registers **at the end of execution** for each of the instructions. Assume all other registers are initially x0000. Complete the table.

	PC	MAR	MDR	IR	R0	R1
Before 1st instruction	x3000	—	—	—	x0000	x0000
After 1st instruction		x3005			x8667	
After 2nd instruction				x5021		x0000
After 3rd instruction				x0401		
After 4th instruction						x0001

- What does this program do?