Homework 5 CS/ECE 252: Sec 1 & 2 [Due at lecture on Wed, Mar 26]
Primary contact for this hw: Deepika Muthukumar [deepikam at cs dot wisc dot edu]

Instructions: You must do this homework alone. Please hand in ONE copy of the homework listing your section number, full name (as appear in Learn@UW) and UW ID. You must staple all pages of your homework together to receive full credit.

Problem 1 (6 points)

Write LC-3 instructions in hex for implementing the following tasks.

1. Move the value in R1 into R5 (Give a single instruction)
   AND R5 R1 R1: 0x5A41
2. Store the 2's complement of R4 into R0 without overwriting R4
   R0 = NOT(R4): 0x913F
   R0 = R0+1: 0x1021
3. Load the value at 0x0003 into R1
   R1 = R1 AND 0x0: 0x5260
   R1 = M[value(R1)+0x3]: 0x6243

Problem 2 (3 points)

Suppose the following LC-3 program is loaded into memory starting at location x30FF:

x30FF: 1110 0010 0000 0001: LEA R1, 1: 0x3101 stored in R1
x3100: 0110 0100 0100 0001: LDR R2, R1, 1: 0x1441 stored in R2
x3101: 1111 0000 0010 0101: HALT
x3102: 0001 0100 0100 0001
x3103: 0001 0100 1000 0010
If the program is executed, what is the value in R2 at the end of execution?
0x1441

Problem 3 (2 points)

A memory's addressability is 32 bits. What does that tell about the size of the MAR and MDR?
The MDR is 32 bits. The statement does not tell anything about the size of the MAR.

Problem 4 (2 points)

We wish to execute a single LC-3 instruction that will subtract the decimal number 18 from register 1 and put the result into register 2. Can we do it? If yes, show how it would be done. If not,

explain why not.

No, this cannot be done in a single instruction because the smallest integer possible for the immediate field in the ADD instruction is -16 since there are only 5 bits available. This operation would need a minimum of two instructions.

Problem 5 (4 points)

Let's say we have a memory consisting of 1024 locations and each location contains 16 bits.

a. How many bits are required for the address? 10 bits

b. If we use the PC-relative addressing mode, and want to allow control transfer between instructions 25 locations away, how many bits of a branch instruction are needed to specify the PC-relative offset? 6 bits

Problem 6 (2 points)

Consider a 16 bit processor (each instruction is 16 bits wide). A typical instruction looks like ADD Destination, Source 1, Source 2, where destination, source 1 and source 2 are registers. If the first 4-bits are used to differentiate instructions (Opcode), what is the maximum number of registers that the processor can support?

4 bits per register: 16 registers

Problem 7 (4 points)

A LD instruction in LC-3 is located at 0x4200.

a. What are the largest and smallest addresses from where the data can be loaded using this instruction?

   Largest Address = 0x4200 + 0x1 + 0xFF = 0x4300
   Smallest Address = 0x4200 + 0x1 - 0x100 = 0x4101

b. If LC-3 used unsigned number for offset instead of 2s complement number, what would be the largest and smallest address from where the data can be loaded using this instruction?

   Largest Address = 0x4200 + 0x1 + 0x1FF = 0x4400
   Smallest Address = 0x4200 + 0x1 + 0x00 = 0x4201

Problem 8 (4 points)

The LC-3 does not have an opcode for logical function OR. That is, there is no instruction in the LC-3 ISA that performs the OR operation. However, we can write a sequence of instructions to implement the OR operation. The four instruction sequence below performs the OR of the contents of register 1 and register 2 and puts the result in register 4. Fill in the two missing instructions so that the four instruction sequence will do the OR operation.

1. 1001 0110 0111 1111
2.
3. 0101 1100 1100 0101
4.

A OR B = NOT(NOT A AND NOT B)

1. 1001 0110 0111 1111: R3=NOT R1
2. 1001 1010 1011 1111: R5=NOT R2
3. 0101 1100 1100 0101: R6=R3 AND R5
4. 1001 1001 1011 1111: R4=NOT R6

Problem 9 (3 points)

What does the following LC-3 instruction do? (Be specific; that is, give specific register numbers or memory locations.)

Address	Binary	Hex
0x450A	0011 0110 0000 0111	0x3607

The instruction is: ST R3 PC+7. So, specifically, the value currently in R3 would be stored at address 0x450B + 7 = 0x4512.