Homework 5 [Due at lecture on Wed, Mar 18]

Primary contact for this homework on Piazza is Sujith Surendran

Problem 1 (3 points)

Consider a 16 bit processor (each instruction is 16 bits wide). All the instructions have the following format:

<table>
<thead>
<tr>
<th>Opcode</th>
<th>Destination</th>
<th>Source 1</th>
<th>Source 2</th>
<th>Immval</th>
</tr>
</thead>
</table>

Opcode Destination, Source 1, Source 2, immval

where destination, source 1 and source 2 are registers, and immval is the PC relative value of the offset in 2’s complement form.

If there are 8 different instructions supported and there are 4 registers, specify:
   a. The number of bits required for the opcode field
   b. Number of bits required for destination, Source1, Source2 fields
   c. The maximum and minimum values of the immval field.

Answer: Opcode: 3 bits
Bits for register fields : 2 each
Bits for immediate value = 7 => Max value = $2^6 - 1 = 63$, Min value = -64

Problem 2 (2 points)

Assume there exists another computer which can address $2^{32}$ memory locations, and assume that each of the memory locations contain 16 bits of data.
   a. What is the size of MAR ?
   b. What is the size of MDR ?
Size of MAR = 32 bits
Size of MDR = 16 bits

Problem 3 (10 points)

Write LC-3 instructions (in binary) to do the following:

a. Place a value 0x0005 into R1
b. Place the value at memory location 0x3005 into R2.
c. Store 2’s complement of R3 into R4
d. Move the value of R5 into R6
e. Do OR operation on R5, R6 and store the result into R0, ie, R0 = R5 OR R6

Note: You can assume that each of these instructions start at memory location x3000

a. AND R1, R1, #0 \[\rightarrow 0101 001 001 1 00000\]
   ADD R1, R1, #5 \[\rightarrow 0001 001 001 1 00101\]

b. LD R2, PCoffset = 4 \[\rightarrow 0010 010 000000100\]

c. NOT R3, R3 \[\rightarrow 1001 011 011 111111\]
   ADD R4, R3, #1 \[\rightarrow 0001 100 011 1 00001\]

d. ADD R6, R5, #0 \[\rightarrow 0001 110 101 1 00000\]

e. NOT R5, R5 \[\rightarrow 1001 101 101 111111\]
   NOT R6, R6 \[\rightarrow 1001 110 110 111111\]
   AND R0, R5, R6 \[\rightarrow 0101 000 101 0 00 110\]
   NOT R0, R0 \[\rightarrow 1001 000 000 111111\]
Problem 4 (9 points)

LD R0, x3100  0010 000 011111111
ADD R0, R0, #1  0001 000 000 1 00001
NOT R1, R0  1001 001 000 1 11111
AND R2, R1, xFFF8  0101 010 001 1 11000
ST R0, x3100  0011 000 011111011
ST R1, x3101  0011 001 011111011
ST R2, x3102  0011 010 011111011

a. (4 points) Fill in the comments for the above program describing what each instruction does. For example, if one of the instructions was 1001 0100 1111 1111, then the comment for this instruction can be either "This instruction stores the complement of value at R3 into R2" or "R2 <-- NOT(R3)".

b. (3 points) If the initial condition of the memory locations 0x3100 to 0x3103 before executing the code are as shown below (ie, all are 0), what are the final values at these memory locations after executing the code?

<table>
<thead>
<tr>
<th>Address</th>
<th>Initial Value</th>
<th>Final Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x3100</td>
<td>0x0000</td>
<td></td>
</tr>
<tr>
<td>0x3101</td>
<td>0x0000</td>
<td></td>
</tr>
<tr>
<td>0x3102</td>
<td>0x0000</td>
<td></td>
</tr>
<tr>
<td>0x3103</td>
<td>0x0000</td>
<td></td>
</tr>
</tbody>
</table>

Answer: x3100 => x0001
         x3101 => xFFFE
         x3012 => xFFF8

c. (2 points) If the whole memory is shifted up by 0x1000, ie, if the instructions start at 0x2000 (instead of 0x3000) and the data values start at 0x2100 (instead of 0x3100), will these instructions still work? Specify why or why not.

Yes they will still work because all the addresses for loads and stores are PC relative
Problem 5 (4 points)

A LD instruction in LC3 is located at 0x4200.

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

b. If LC3 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?

Answer:

a. Largest = x4201+0xFF = x4300
   Smallest = x4201 - 0x100 = x4101

b. Largest = x4201 + 0xFF = x4300
   Smallest = x4201 - 0xFF = x4102

Problem 6 (2 points)

The purpose of this problem is to get you setup with the PennSim LC-3 simulator, which will be important for subsequent homeworks. To get started with PennSim, look at the information on the Computing page. Specifically, go through the PennSim Guide.

Download the PennSim simulator from the Computing page and run it. You should be able to run PennSim on any computer (with Java 1.6 or higher installed). They do not require any installation; they are self-contained executable files.

- To run on a Windows/Mac computer: double click the PennSim.jar file.
- To run on a linux machine: open Terminal and navigate to the folder containing PennSim.jar. Execute the following command "java -jar PennSim.jar"

a. (2 points) Once you have opened the simulator, list down the registers which are used by the simulator (visible on the upper left corner).