NCSU ECE 109 Sections 002, 602, 604, and 605 and UNCA CSCI 255.001
Exam 2 Spring 2008
3 April, 2008

This is a closed book and closed notes exam. It is to be turned in by 5:50 pm. Calculators, PDA's, cell phones, and other electronic or communication devices may not be used during this exam. Please read and sign the following statement:

I have neither given not received unauthorized assistance on this test.

Name: _____

If you want partial credit for imperfect answers, explain the reason for your answer!

Problem 1 (10 points) Data Path
Describe how the LC-3 instruction

```
STR R5,R4,#3
```

causes data to be routed through the components, such as registers, buses, and MUXes, of the LC-3 data path. This `STR` instruction may refer to registers or memory elements whose values aren't known. Don't worry about these unknowns. Just use words like "contents of register *x*" in your answer.

Problem 2 (15 points) LC-3 programming

Write two short LC-3 programs to solve the following small problems. Answers that are unnecessary long or complicated will not receive full credit.

Write LC/3 code to compute `1-R3` and store the result in `R5`. That is, do `R5 ← 1-R3`
Write LC/3 code to compare R4 to zero and to (1) set R5 to 3, if R4 is positive; or (2) set R5 to 1, if R4 is zero or negative.

Problem 2 (25 points) Hand assembled
Use the symbol table shown below in this question.

ASHEVILLE	x4626
KINSTON	x4695
RALEIGH	x4685
WILMINGTON	x46A6

Write the appropriate 16-bit LC-3 machine language word, in binary or hex, for each assembly language statement shown in the left column of the table below. Assume that the instruction is located at address x4600 in all cases. If the assembly language statement is illegal, state the reason why.

ADD R0,R2,#12	
ADD R7,R7,#-12	
AND R2,R3,R4	
AND R2,R3,#14	
BR ASHEVILLE	
BRnz KINSTON	
BRz x15	
LD R2,RALEIGH	
LDR R2,R2,#-2	
LEA R2,WILMINGTON	
OR R4,R3,R4	
STR R3,R4,#20	

Problem 3 (25 points) Hand executed

The binary program shown in the left column below is loaded into memory starting at `x3000`. In the right column, write the LC/3 assembly instructions or appropriate psuedo-ops corresponding to this program. Be sure to include appropriate labels and `.ORIG` and `.END` statements.

Binary	Assembly
1010010000000101	
0001001010111101	
0101011010101100	
0000101000000001	
0001010010100001	
1111000000100101	
0011000000000111	
0000000000000101	

Suppose the program shown above is executed on the LC-3, starting with its first instruction. As the program is executed, instructions will be fetched, register or memories locations *may* be modified with values computed by the instruction, and the N, Z, and P bits *may* be changed. Document these changes in the table below. Fill in the cells of the table for each executed instruction until either (1) ten instructions are executed or (2) the `HALT` (`TRAP x25`) instruction is reached.

Do not simply fill in a row for each word shown in the above table. Some of these words **will never be** executed and **should not** appear in the table. Some of these words may be executed more than once. Entries in the "Effect of the instruction" should be very specific, such as "Adds R5 and R3 and obtains xFF which is stored in R4".

Instruction address	Register or memory location modified	Effect of the instruction	**NZP**
x3000			

Problem 4 (16 2/3 points)
Assume that the eight LC/3 registers have the values shown on the left below and that the eight words of memory starting at memory location `x3040` have the values shown on the right.

Register	***Value***
R0	x0000
R1	x0000
R2	x2101
R3	x3101
R4	x0000
R5	x5101
R6	x0000
R7	x0000

Address	***Value***
x3040	x0000
x3041	x0000
x3042	x2121
x3043	x3131
x3044	x0000
x3045	x5151
x3046	x0000
x3047	x0000

For the six addresses shown below, write a single LC/3 instruction to load the value **stored in** the specified memory location into register 4. (For example, when `x3041` is specified, `x1111` should be stored in R4.) Assume that each instruction is located at memory address `x3021`.

If this location cannot be loaded in one instruction, state why this is not possible.

x0000	
x2120	
x2121	
x3041	
x3131	
x5202	

Problem 5 (8 1/3 points)
Assume the same register and memory contents as in Problem 4. Also, continue to assume that you are writing an instruction to be stored at memory location `x3021`.

However, this time assume the leftmost column refers to **values** and **not** to memory locations. (For example, when `x3041` is specified, `x3041` should be stored in R4.) For each row, write a single instruction to store these values into register 4. If this value cannot be generated in one instruction, state why this is not possible.

x3041	
x3131	
x5202	