**NCSU ECE 109 Sections 602 and 603 and UNCA CSCI 255.001**
**Exam 2 Spring 2009 Answers**
26 March, 2009

**Problem 1 (8 points) Overflow**
Add the following pairs of five-bit two's complement numbers **and indicate which additions result in an overflow**.

| | |
|---|---|
| 10000<br>+ 01111<br>**11111**<br>*No overflow* | 10000<br>+ 10000<br>**00000**<br>*overflow* |
| 11000<br>+ 11111<br>**10111**<br>*No overflow* | 00011<br>+ 01110<br>**10001**<br>*overflow* |

**Problem 2 (6 points) Decoder and MUX**
If a decoder has 5 inputs, how many outputs does it have?

**It has 32 inputs, because $2^5$ is 32.**

If a MUX is to select from 16 data inputs, how many selector inputs does it have?

**It has 4 selector inputs, because 16 is $2^4$.**

**Problem 3 (12 points) Truth to Gates**
Draw a circuit, at the gate level, that will implement the following truth table, where A, B, and C are inputs and where Z is the single output.

| A | B | C | Z |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 |

**Problem 4 (6 points) Memory**
A computer memory has 16-bit words stored in 2 M locations. What is the size of this memory in bits?

**It has 32 M bits. That's 16 × 2 M.**

How many address bits are needed to address the 2 M words of this memory?

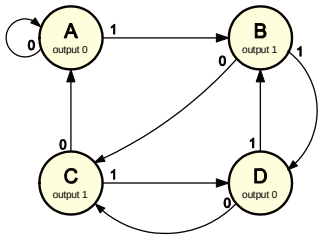
**It needs 21. That's because 2 M is $2 \times 2^{20}$ or $2^{21}$.**

**Problem 5 (6 points) Data path**
On the first execution cycle of an LC-3 instruction, two operations are performed: (1) the PC is loaded in the MAR, and (2) the PC is incremented by one. Briefly describe how this is done with the textbook's LC-3 data path.

**To transfer the PC to MAR, GatePC is asserted (turned on) to place the PC on the bus and LD.MAR is asserted to store the bus value into the MAR. To increment the PC, the selector inputs of PCMUX are set to allow PC+1 to pass and LD.PC is asserted to load the PC with the incremented value.**

**Problem 6 (15 points) Finite State Machines**

A picture of a finite state machine is given below. The FSM has four states, labeled A to D, and receives single input bit. There is also a single output bit which is shown inside the state. The starting state for the FSM is A.




*Subproblem A:*

Suppose the FSM is started (in state A) and then receives three successive inputs `0 1 1`. After the 3rd input bit is received, which state is the FSM in and what is its output?

**The FSM make transitions from A to A on 0 and to B on 1 and to D on the final 1. Its output at D is 0.**

*Subproblem B:*

In the table below give a two-bit state encoding for the four states of the FSM. There are many (actually 24) possible correct answers for this subproblem.

| State as letter | State as two bits | |
|---|---|---|
| A | **0** | **0** |
| B | **0** | **1** |
| C | **1** | **0** |
| D | **1** | **1** |

*Subproblem C:*

Complete the following table to show the state transitions of the FSM at the top of the page using your state encodings from the previous subproblem.

| *present state* | | *input* | *next state* | |
|---|---|---|---|---|
| **0** | **0** | **0** | **0** | **0** |
| **0** | **0** | **1** | **0** | **1** |
| **0** | **1** | **0** | **1** | **0** |
| **0** | **1** | **1** | **1** | **1** |
| **1** | **0** | **0** | **0** | **0** |
| **1** | **0** | **1** | **1** | **1** |
| **1** | **1** | **0** | **1** | **0** |
| **1** | **1** | **1** | **0** | **1** |

**Problem 7 (27 points) Decoding and executing LC-3 instructions**

In this problem, you are going to describe what happens when an LC-3 instruction located at address `x4004` is executed. The values of the `NZP` bits, eight LC-3 registers, and memory locations from `x4000` to `x4007` when the instruction begins execution are shown below. The memory location for `x4004` is vacant because its value varies throughout the problem.

| | |
|---|---|
| N | 0 |
| Z | 1 |
| P | 0 |

| | |
|---|---|
| R0 | x0004 |
| R1 | x0005 |
| R2 | x0006 |
| R3 | x0007 |
| R4 | x4000 |
| R5 | x4001 |
| R6 | x4002 |
| R7 | x4003 |

| | |
|---|---|
| M[x4000] | x4004 |
| M[x4001] | x4005 |
| M[x4002] | x4006 |
| M[x4003] | x4007 |
| M[x4004] | *see below* |
| M[x4005] | x0001 |
| M[x4006] | x0002 |
| M[x4007] | x0003 |

Below, there are eleven boxes. In the upper-left corner of each box, there is a 16-bit value. Assume that this 16-bit quantity becomes the value of memory location `x4004` and is fetched from memory and executed as an instruction of the LC-3 computer. Within each box, write the result of executing its instruction. Mention every `NZP` bit, LC-3 register, or memory location changed by the instruction. If the PC is changed by a branch, be sure to mention the updated value for the PC. If the PC is merely increased by one, you do not need to mention that change. It nothing is changed, explicitly say this is the case.

These instructions are *not* executed in order. They are all run *independently* with the register and memory values shown on the previous page.

| |
|---|
| **0001001010000011 ADD R1,R2,R3**<br>***R1 will be equal to 13 (xD) and the P bit will be set.*** |
| **0001001000100011 ADD R1,R0,#3**<br>***R1 will be equal to 7 and the P bit will be set.*** |
| **0001001000111111 ADD R1,R0,#-1**<br>***R1 will be equal to 3 and the P bit will be set.*** |
| **0101001111100000 AND R1,R7,#0**<br>***R1 will be equal to 0 and the Z bit will be set.*** |
| **0101010010000010 AND R2,R2,R2**<br>***R2 will be equal to 6 and the P bit will be set.*** |
| **0000100111111101 BRn #-3 [x4002]**<br>***Since the N bit isn't set, there are no changes (except for incrementing PC).*** |
| **0000010000000001 BRz #+1 [x4006]**<br>***Since the Z bit is set, the PC is changed to x4006.*** |
| **0010011000000001 LD R3,#+1 [x4006]**<br>***R3 will be equal to 2 (from memory location x4006) and the P bit will be set.*** |
| **0110011101000000 LDR R3,R5,#0**<br>***R3 will be equal to x4005 (from memory location x4001) and the P bit will be set.*** |
| **1110000000000000 LEA R0,#0 [x4005]**<br>***R0 will be set to x4005 (the present PC) and the P bit will be set.*** |
| **0011011000000010 ST R3,#2 [x4007]**<br>***Memory location x4007 will be set to 7. No change of NZP bits.*** |

**Problem 8 (20 points) Decoding and executing an LC-3 program**

In the table below is an LC-3 program that is loaded into five memory locations starting at x3000. First, complete the table by decoding the instructions into the rightmost column of the table.

| | | |
|---|---|---|
| x3000 | 0101000000100000 | **AND R0,R0,#0** |
| x3001 | 0001000000100101 | **ADD R0,R0,#5** |
| x3002 | 0001000000111101 | **ADD R0,R0,#-3** |
| x3003 | 0000001111111110 | **BRp #-2 [x3002]** |
| x3004 | 1111000000100101 | HALT ;TRAP x25 |

Next, describe the order in which the instructions of this program are executed until it reaches the HALT and give the final values of any registers changed by the program at that time.

**(1) Instruction at x3000 is executed. R0 is set to 0.**
**(2) Instruction at x3001 is executed. R0 is set to 5.**
**(3) Instruction at x3002 is executed. R0 is set to 2.**
**(4) Instruction at x3003 is executed. Branch is taken to x3002.**
**(5) Instruction at x3002 is executed. R0 is set to -1.**
**(6) Instruction at x3003 is executed. Branch is not taken.**
**(7) HALT.**

**R0 is -1 at end of program.**