ECE 199 Exam I Spring 2004

Wednesday, February 25th, 2004

Name:

- **Be sure your exam booklet has 12 pages**
- **Write your name at the top of each page**
- **This is a closed book exam**
- **You are allowed one handwritten 8.5 x 11 sheet of notes**
- **Absolutely no interaction between students is allowed**
- **Show all of your work**
- **Be sure to clearly indicate any assumptions that you make**
- **More challenging questions are marked with a *****
- **Don’t panic, and good luck!**

Problem 1	20 points	______________________
Problem 2	20 points	______________________
Problem 3	20 points	______________________
Problem 4	20 points	______________________
Problem 5	20 points	______________________
Total	100 points	

 Name: ____

Problem 1 (20 points): Short Answer

Part A (5 points)**:** Convert the following numbers.

-15 (from decimal to 8-bit 2’s complement)

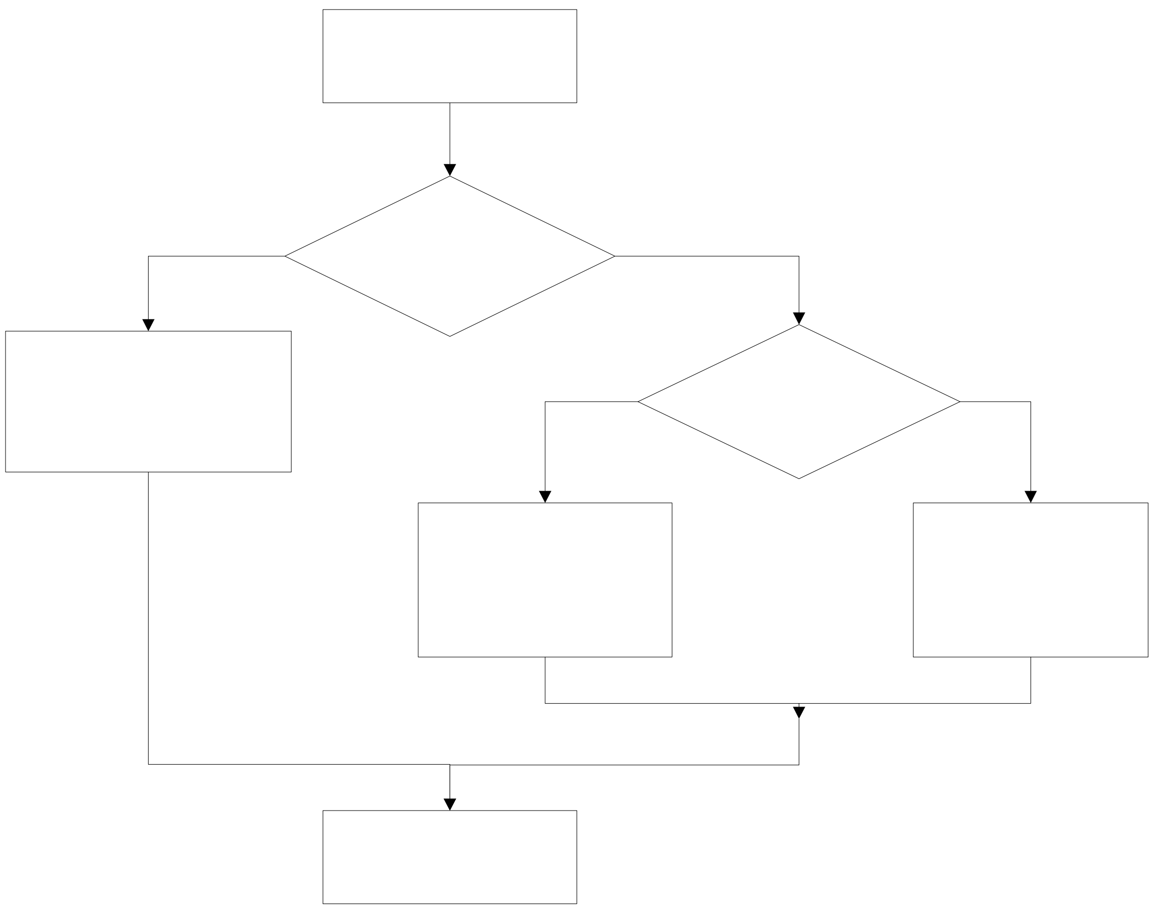
11011001 (from 8-bit 2’s complement to decimal)

Part B (5 points): Is the pair of functions f_1, f_2 together logically complete? Prove that your answer is correct.

A	**B**	f_1	f_2
0	0	1	0
0	1	1	0
1	0	0	1
1	1	0	0

Part C (5 points): There are now slightly fewer than 1.1001×2^{32} living humans on Earth. Assuming that this number may increase as much as 32-fold in the next 100 years, how many bits are necessary to represent any living human at any time in that period?

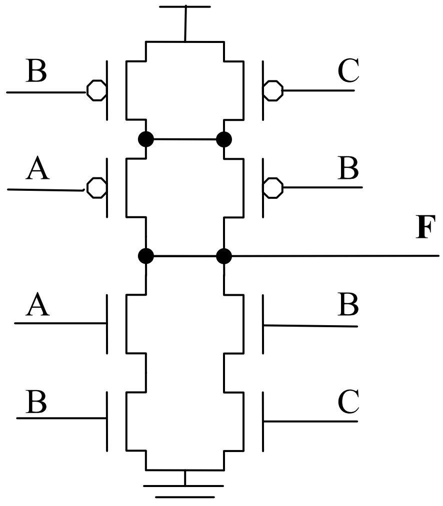
Part D (5 points): Given the structure below, fill in the blanks as necessary to implement a flowchart that changes a lowercase ASCII letter to uppercase and ignores non-letters. Write “empty” in a box if no action should be taken.



 Name: ___________________________________

Problem 2 (20 Points): Transistors and Combinational Logic

Part A (5 points): Fill in the truth table for the circuit below.



A	**B**	**C**	**F**
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

*****Part B** (3 points): The above circuit can be simplified to use fewer transistors without changing the logical function. Draw such a circuit.

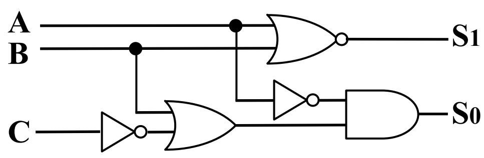
 Name: ____

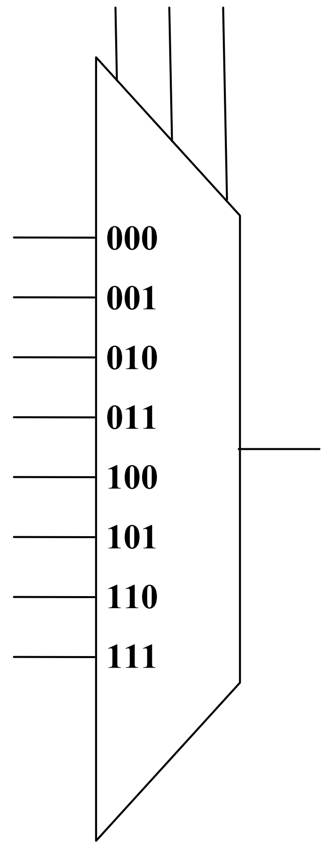
Part C (5 points): Use a 3-input decoder and one multi-input gate to implement a majority function, i.e., a 3-input function that outputs a 1 when two or more (the majority) of its inputs are 1's. Be sure to label all relevant parts of the decoder.

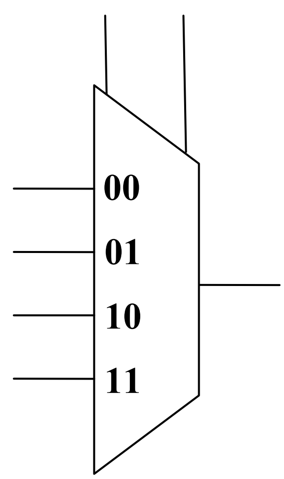
 Name: ____

Part D (7 points)

Use the two MUXes (one two-input and one three-input) provided to the right below to construct an alternative version of the circuit to the left. You may use the MUXes, the inputs A, B, and C, and as many 0's and 1's as you need, but you **may not** use extra gates. Be sure to label all inputs and outputs for each mux. *Hint: Writing out the truth table may help.*



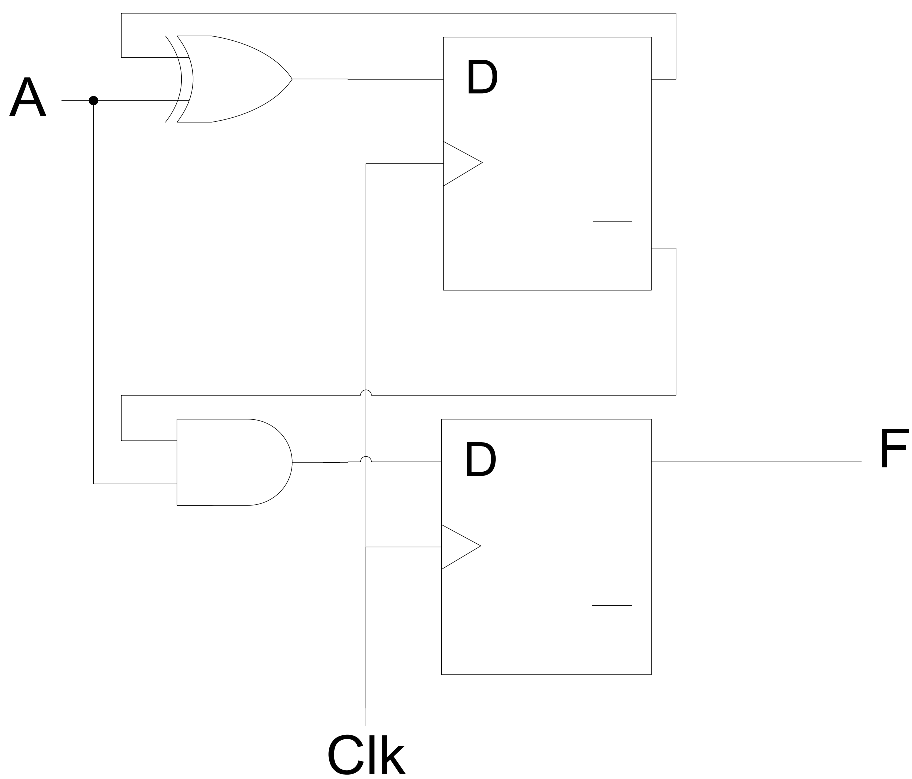




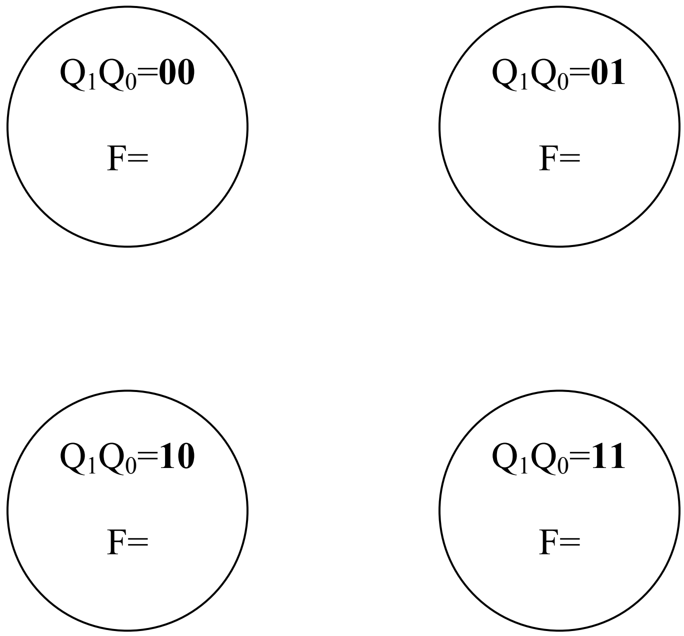
 Name: ____

Problem 3 (20 points): Finite State Machines

For the following question, refer to the circuit below.



Part A (7 points): Draw the finite state machine transition diagram for the circuit. Be sure to fill in all input transition arcs and state outputs. The state diagram ($Q = Q_1Q_0$) has been partially filled in to help you get started. *Hint: For each value of A, write the next state in terms of the current state, then label the diagram below.*



Part B (3 points): Assuming that the circuit is started in state Q_1Q_0 = 00, are all states reachable? In other words, is it possible for the state of the circuit to take on any given combination of 0s and 1s through some sequence of inputs? Explain your answer.

Part C (5 points): Fill in the table for the new state Q_{new} ($=Q_1Q_0$) and the output F after each input transition has completed. Assume that the circuit is initially in state Q_1Q_0=00.

A	0	1	1	0	0	1	0	1	1	0	1	1
$\mathbf{Q_{new}}$												
$\mathbf{F_{new}}$												

Part D (5 points): Summarize what this circuit does in one sentence.

Problem 4 (20 points): The Von Neumann Model

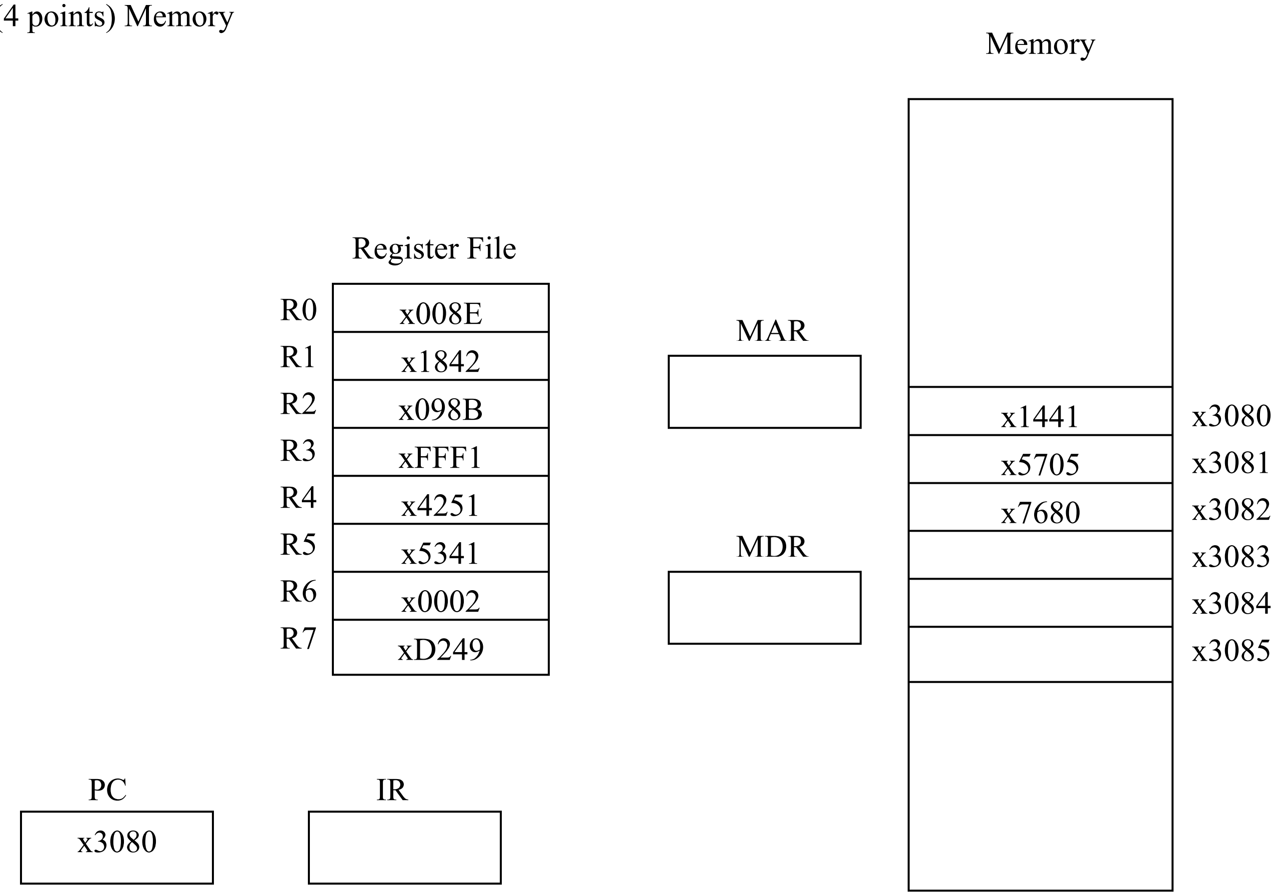
For the following question, refer to the LC-3 Von Neumann model below. The current PC and register file is provided, and the LC-3 computer is about to perform a **FETCH**.

Part A (4 points): The data in memory locations x3080, x3081 and x3082 are indicated in hexadecimal on the diagram. For each data word shown in memory, write a register transfer language (RTL) statement corresponding to the instruction represented by the word. For example, the word x907F is "R0 ← NOT R1" in RTL. Note that we will not grade on RTL syntax, but what you write must convey the action taken by the LC-3 when processing each instruction.

Address	RTL
x3080	
x3081	
x3082	

Part B (16 points): In the diagram below, the LC-3 is about to start the instruction cycle for the instruction at x3080. Update the diagram to reflect the state of the machine after the execution of three full instruction cycles. Clearly denote the final values in the following components within the diagram below.

i) (4 points) Register File
ii) (4 points) IR and PC
iii) (4 points) MAR and MDR
iv) (4 points) Memory



 Name: ____

Problem 5 (20 Points): Programming

Part A (15 points): The code fragment below calculates **(R1 × R1) + 6**, leaving the result in R3. First translate the instructions to register transfer language (RTL). For example, the instruction 1001 0000 0111 1111 is "R0 ← NOT R1" in RTL. After translating the instructions, fill in the missing bits to complete the program and update the RTL to reflect your additions.

Address	Instruction Bits	RTL
x3000	0101 0110 1110 0000	
x3001		
x3002	0001 0110 1100 0001	
x3003	0001 0100 1011 1111	
x3004	0000 1011	
x3005	0001 0110 1110 0110	

Note that we will not grade on RTL syntax, but what you write must convey the action taken by the LC-3 when processing each instruction.

*****Part B** (5 points): If R1 and R3 hold 16-bit 2's complement numbers, for what values of R1 is the result in R3 correct (i.e., equal to the expression **(R1 × R1) + 6**) after executing the code above? Explain your answer.

 Name: ____

Use this page for scratchwork

Instruction	15 14 13 12	11 10 9	8 7 6	5	4 3	2 1 0
ADD+	0001	DR	SR1	0	00	SR2
ADD+	0001	DR	SR1	1	imm5	
AND+	0101	DR	SR1	0	00	SR2
AND+	0101	DR	SR1	1	imm5	
BR	0000	n z p	PCoffset9			
JMP	1100	000	BaseR	000000		
JSR	0100	1 PCoffset11				
JSRR	0100	0 00	BaseR	000000		
LD+	0010	DR	PCoffset9			
LDI+	1010	DR	PCoffset9			
LDR+	0110	DR	BaseR	offset6		
LEA+	1110	DR	PCoffset9			
NOT+	1001	DR	SR	111111		
RET	1100	000	111	000000		
RTI	1000	000000000000				
ST	0011	SR	PCoffset9			
STI	1011	SR	PCoffset9			
STR	0111	SR	BaseR	offset6		
TRAP	1111	0000	trapvect8			
reserved	1101					

Figure A.2 Format of the entire LC-3 instruction set. Note: + indicates instructions that modify condition codes