CS/ECE 252: INTRODUCTION TO COMPUTER ENGINEERING

UNIVERSITY OF WISCONSIN—MADISON

Prof. Mark D. Hill

TAs: Sujith Surendran, Lisa Ossian

Midterm Examination 2

In Class (50 minutes)

Friday, March 13, 2014

Weight: 17.5%

NO: BOOK(S), NOTE(S), OR CALCULATORS OF ANY SORT.

The exam has **eleven** pages. **Circle your final answers**. Plan your time carefully since some problems are longer than others. You **must turn in the pages 1-11**. Use the blank sides of the exam for scratch work.

LAST NAME: _____

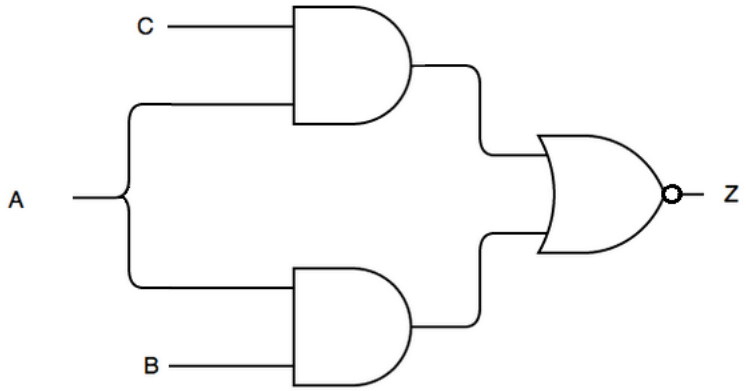
FIRST NAME: _____

ID#: _____

Problem	Maximum Points	Points Earned
1	7	
2	3	
3	3	
4	5	
5	2	
6	3	
7	7	
Total	30	

Problem 1 **(7 points)**

Consider the gate-level circuit below.



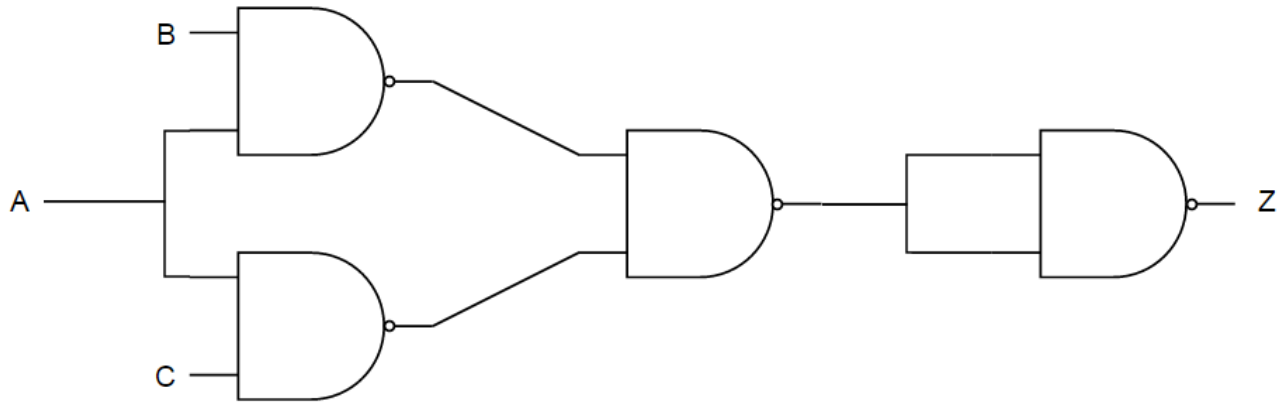
a. (2 points) Fill out the truth table for Z.

A	B	C	Z
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

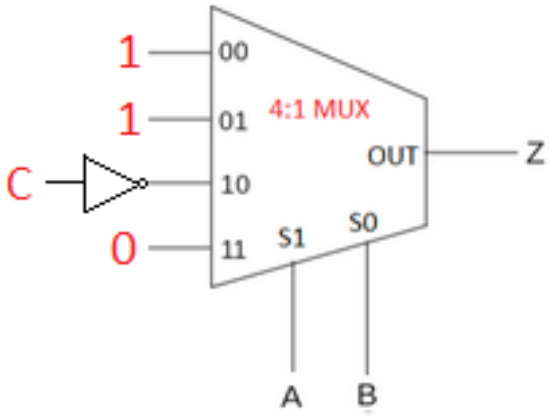
b. (1 point) Write the AND-OR logic expression for Z as a function of inputs A, B, and C. (Do not simplify the expression.) For example, Z = (A AND B AND C) OR (NOT(A)).

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

c. (2 points) Implement a logic circuit for Z using only 2-input NAND gates.

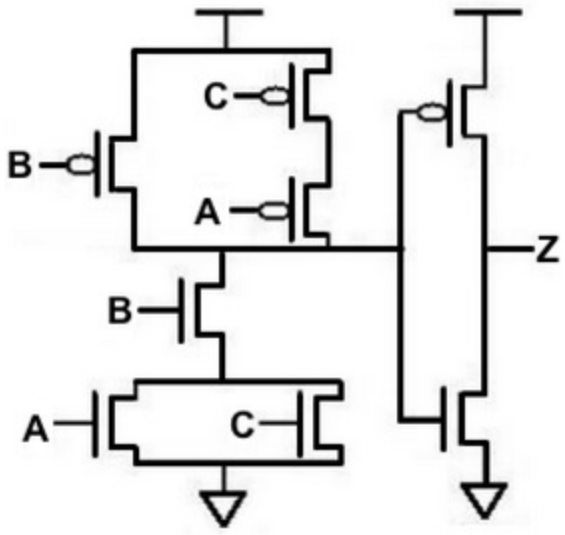


e. (2 points) Implement a logic circuit for Z using only one 4x1 multiplexer and NOT gates where A and B are connected to the select lines. Draw your answer using the 4x1 multiplexer below. Do not use any additional logic gates.



Problem 2 **(3 points)**

Complete the truth table for the following transistor-level circuit.



A	B	C	Z
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

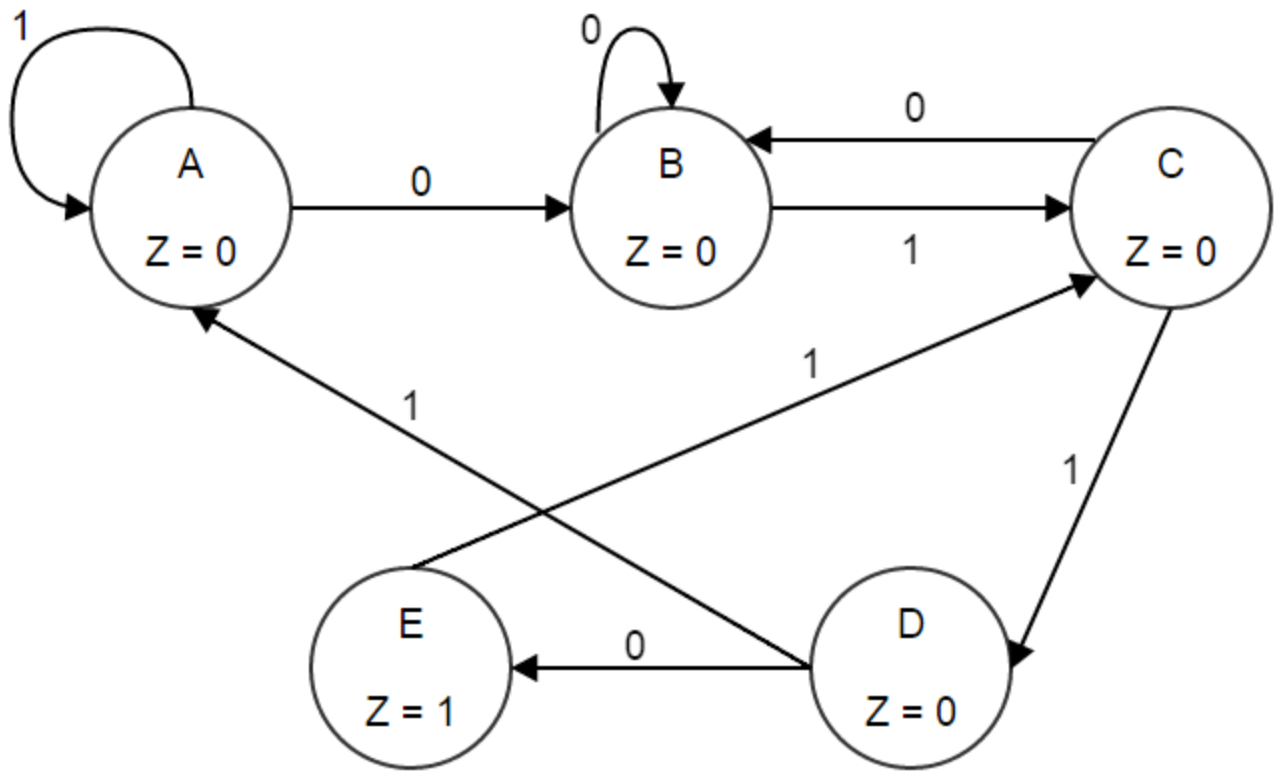
Problem 3 **(3 points)**

Complete the finite state machine (FSM) below for recognizing the bit sequence `0110`. The machine takes one input every clock cycle, which can be `1` or `0`. When the machine finds the sequence `0110`, it should output a `1`. Otherwise, it should output a `0`.

The transitions and output for the initial state (**A**) have been completed for you. Clearly show all possible state transitions and the output (**Z**) for the four remaining states.

Note: The machine should also recognize overlapping input sequences. (See the sample input and output given below.)

Sample Input	`0111 0110 1101 0110`
Sample Output	`0000 0001 0010 0001`



Problem 4 **(5 points)**

The finite state machine (FSM) below (in **Figure 1(a)**) recognizes a certain bit sequence. The machine takes one input every clock cycle, which can be 1 or 0. The machine outputs a '1' when this certain bit sequence is recognized; otherwise it outputs a '0'. Each state is represented as S. For example, the state marked as "0" has S = 0. Z is the output in each state. S' represents the next state and the labels on each transition is the input value that triggers the transition. Assume that the initial state is 0.

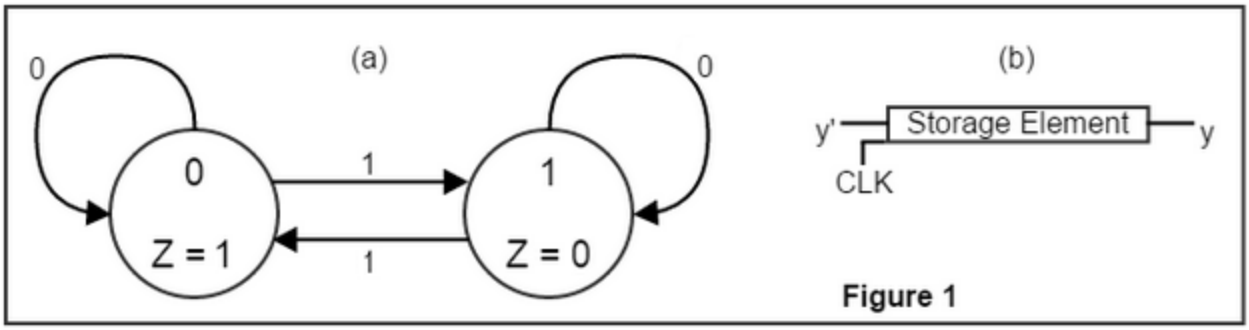


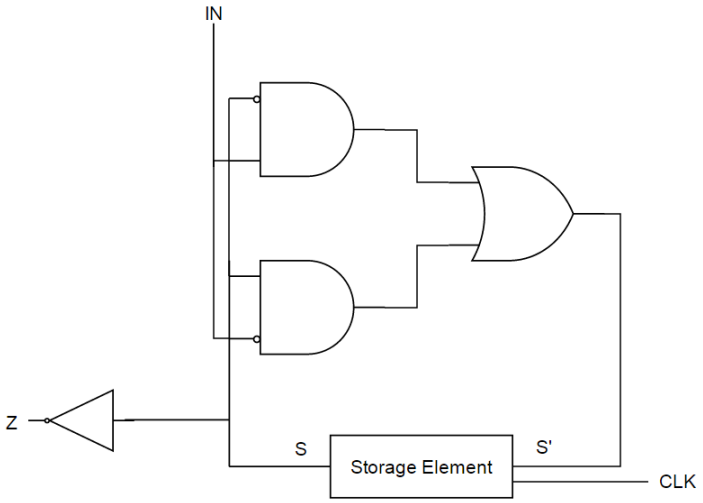
Figure 1

a. (1 point) Complete the Output truth table to the right for the FSM.

b. (1 point) Complete the Next State truth table to the right for the FSM.

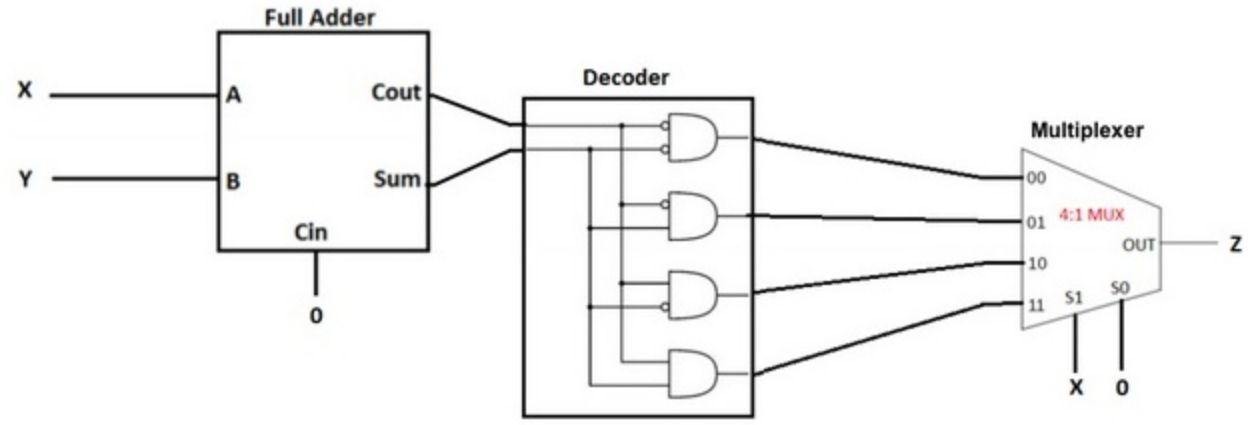
Output		Next State		
S	Z	S	IN	S'
0	1	0	0	0
1	0	0	1	1
		1	0	1
		1	1	0

c. (3 points) Draw the logic circuit which implements the above FSM using combinational logic and flip flops. Use representation shown in **Figure 1(b)** for any 1-bit flip flop required in the circuit (where CLK is the clock). You can use any kind of logic gates for implementing the combinational logic. Note: You should implement the state (S) as well as the output (Z).



Problem 5 **(2 points)**

Consider the following circuit containing a single-bit full adder, a decoder, and a multiplexer. X and Y are inputs to this circuit, and the circuit produces an output Z. Fill in the truth table below for this combinational circuit.



X	Y	Z
0	0	1
0	1	0
1	0	0
1	1	1

Problem 6 **(3 points)**

Suppose a 32-bit instruction takes the following format:

OPCODE	SR	DR	UNUSED	IMM

If there are 128 opcodes and n registers, and the IMM field contains 7 bits:

a. (1 point) What is the minimum number of bits required to represent the OPCODE?

2^7 = 128 opcodes, so 7 bits.

b. (1 point) If the source register (SR) and destination register (DR) each have 5 bits, what is the maximum number of registers there can be?

2^5 = 32 registers.

c. (1 point) If the UNUSED bits were given to the register fields (SR and DR), how many more registers could we have in our computer?

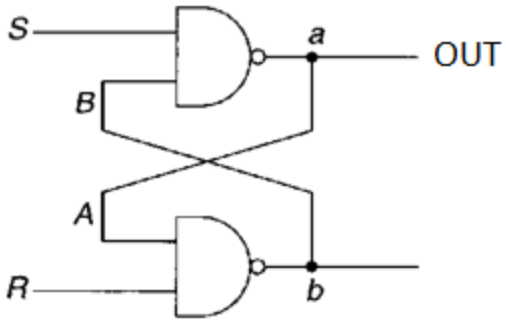
unused bits = 32 - 7 - 5*2 - 7 = 8
We have 8 more bits, so we can give 8/2 = 4 bits to each register field.
Each register field currently contains 5 bits, so we would have 5+4 = 9 bits.
2^9 - 2^5 = 512 - 32 = 480 more registers.

Problem 7 - Short Answer (7 points)

a. (1 point) How many boolean functions can be implemented with n variables?

Given n inputs, 2^{2^n} functions are possible.

b. (2 points) An RS latch is shown in the figure below. Assuming that the inputs to this R-S latch are changed every second (as shown in the table below), specify what would be the value of the "OUT" signal for each of these sets of inputs.



Time (seconds)	0	1	2	3
S	1	1	0	1
R	0	1	1	1
OUT	0	0	1	1

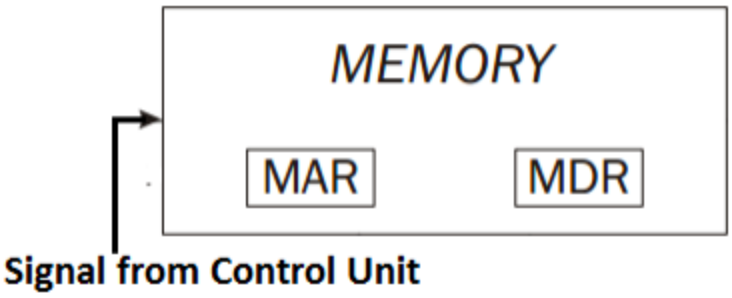
c. (2 points) How many n-type transistors are present in one 32-bit wide register using gated-D latches? Explain your answer.

Each Gated D latch has 4 NAND gates and 1 inverter. Each NAND gate has 2 n-type transistors, and each inverter has 1 n-type transistor. Therefore, we have 32*(4*2+1) = 288 n-type transistors.

d. (0.5 point) What is an opcode?

The portion of a machine language instruction that specifies the operation to be performed.

e. (1.5 points) The figure below shows a block diagram of the Von Neumann model.



List the steps in writing a value x0002 to a location x3011 in the memory. Your steps should mention the MAR and MDR wherever applicable.

1) Write the data x0002 to MDR.

2) Write the address x3011 into the MAR.

3) Send a “write” signal to the memory.