Homework 4 CS/ECE 252: Sec 1 & 2 [Due at lecture on Fri, Feb 21]

Primary contact for this hw: Lisa Ossian [ossianli at cs dot wisc dot edu]

Instructions: You must do this homework alone. Please hand in ONE copy of the homework listing your section number, full name (as appear in Learn@UW) and UW ID. You must staple all pages of your homework together to receive full credit.

Problem 1 (2 points)

Assume you have a 32-input multiplexer. How many output lines does this mux have? How many select lines does it have?

1 output line. There are 2^5 inputs, so there are 5 select lines.

Problem 2 (2 points)

a) Draw a state diagram for a finite state machine that outputs 1 when a binary number with an odd number of 1's is read. For example, for the input "001110111," your machine should output "001011010."
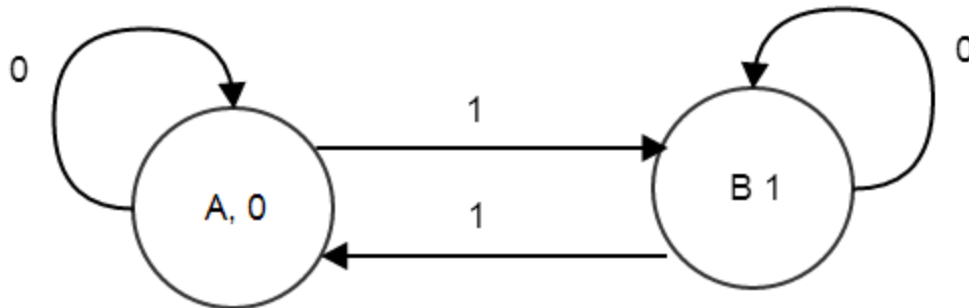
b) How many flip-flops are needed to implement the FSM designed in part a?

1 flip-flop.

Problem 3 (6 points)

A logic circuit has two 2-bit unsigned binary numbers X[1:0] and Y[1:0] as the inputs and it has two 1-bit outputs. One of the outputs is ONEGTR and the other is XLTY.

The ONE_GTR output is true when X[1:0] is one greater than Y[1:0]. For example, if X[1:0] is 01 and Y[1:0] is 10, ONE_GTR is false.

The X1_XOR_Y1 output is true when either X[1] or Y[1] is true, but not when both are true.
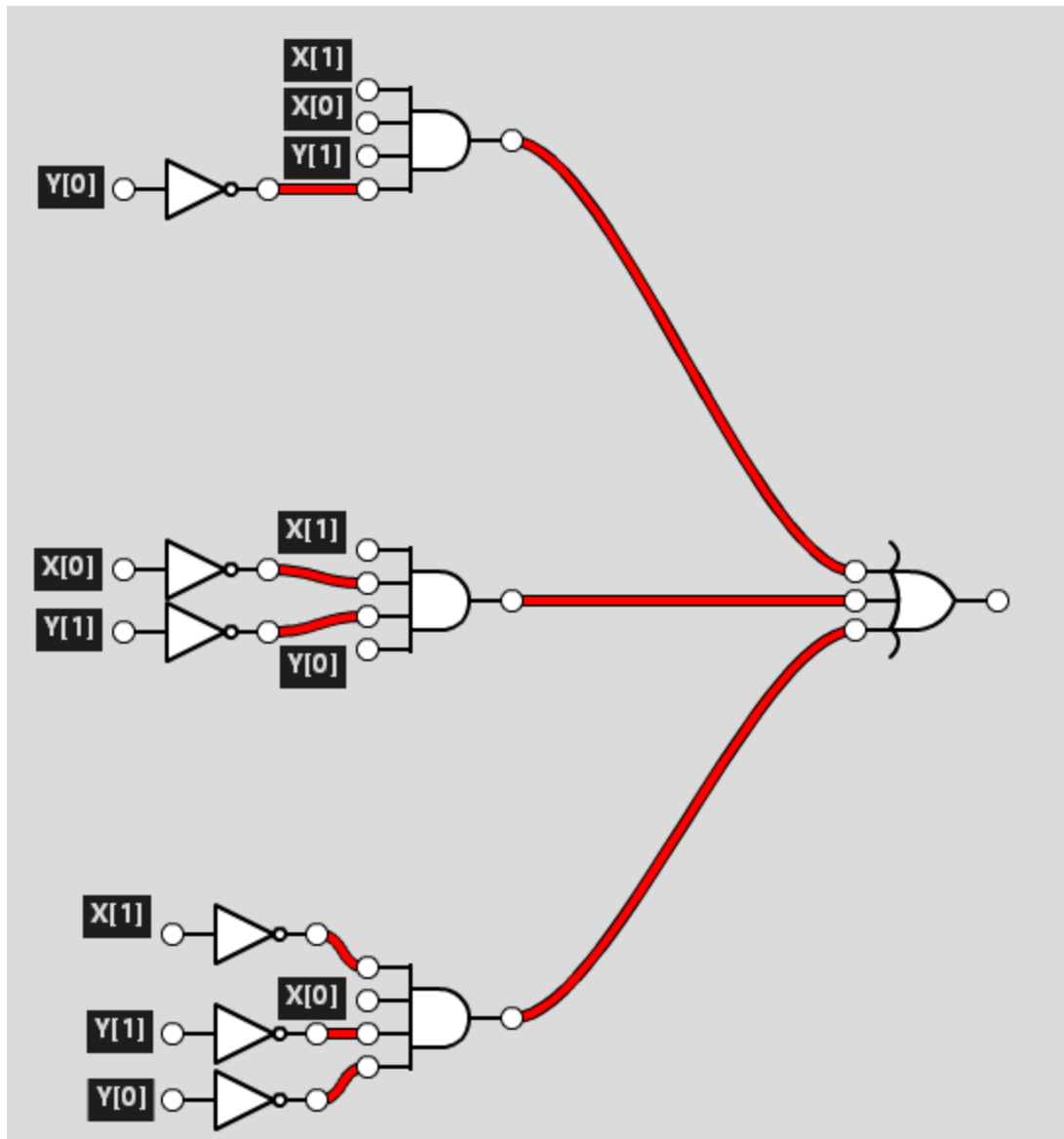
a) Write a truth table for these two functions.

X[1]	X[0]	Y[1]	Y[0]	ONE_GTR	X1_XOR_Y1
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	0	1
0	0	1	1	0	1
0	1	0	0	1	0
0	1	0	1	0	0
0	1	1	0	0	1
0	1	1	1	0	1
1	0	0	0	0	1
1	0	0	1	1	1
1	0	1	0	0	0
1	0	1	1	0	0
1	1	0	0	0	1
1	1	0	1	0	1
1	1	1	0	1	0
1	1	1	1	0	0

b) Write the logic equations for both functions.

ONE_GTR = X[1].X[0].Y[1].Y[0]' + X[1].X[0]'.Y[1]'.Y[0] + X[1]'.X[0].Y[1]'.Y[0]'

X1_XOR_Y1 = X[1].Y[1]' + X[1]'.Y[1]

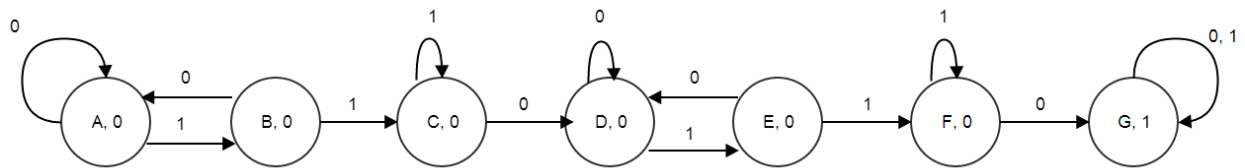c) Draw the Gate level circuit for ONE_GTR using AND, OR and NOT gates

Problem 4 (6 points)

a) Consider a finite state machine that outputs 1 for binary numbers that contain two occurrences of "110." Give two examples of numbers for which the FSM would output 1.

110110
001010110010110
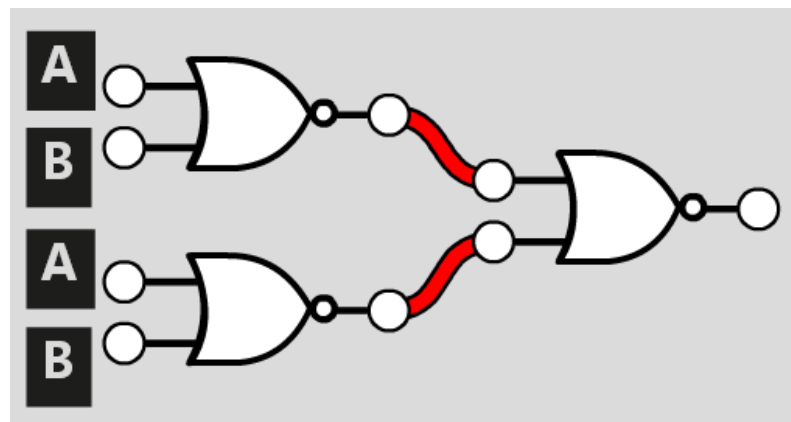..etc.

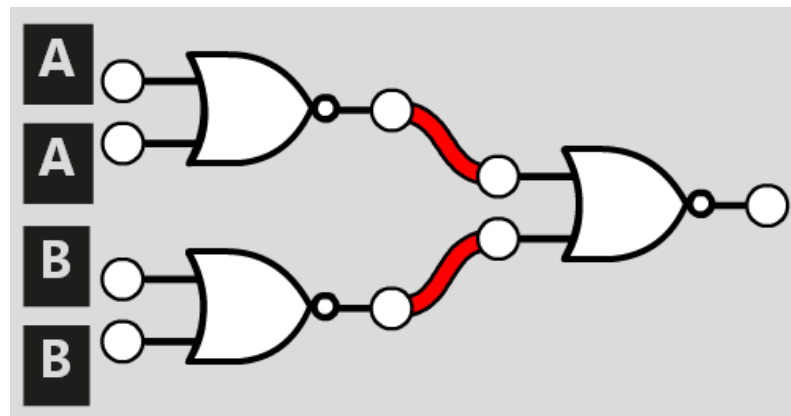b) Draw the FSM described in part a.

Problem 5 (4 points)

In this problem, we will prove that a NOR gate, by itself, is logically complete.

a) Construct a logic circuit that performs the AND function, a logic circuit that perform the OR function, and a logic circuit that perform the NOT function. Use only NOR gates in these three logic circuits.
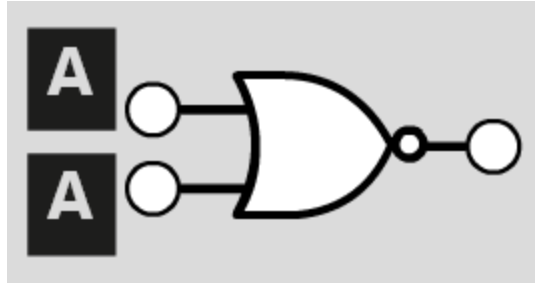
AND

OR
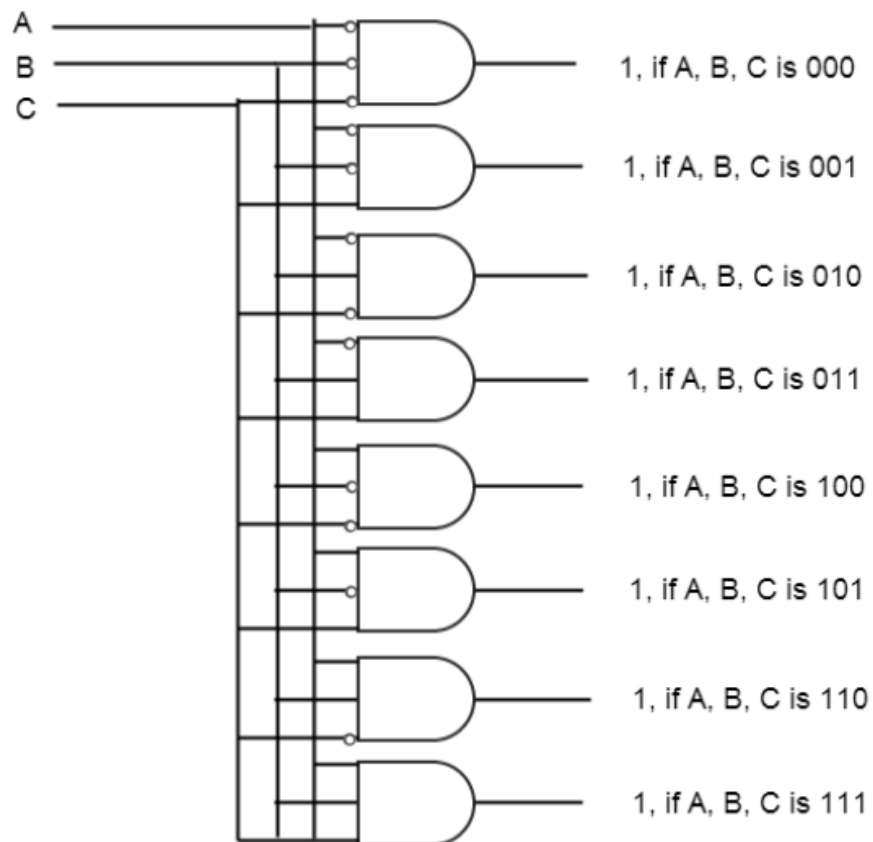
b) Write the logic equation corresponding to each circuit.

AND = (A NOR B) NOR (A NOR B)
OR = (A NOR A) NOR (B NOR B)
NOT = A NOR A

Problem 6 (3 points)

a) Draw a three-input decoder, using the two-input decoder in Figure 3.11a as an example. As in the figure, remember to note the input conditions that will make each output 1.

b) How many outputs does a seven-input decoder have?
128 outputs.

Problem 7 (3 points)

Suppose a 32-bit instruction takes the following format:

OPCODE	DR	SR1	SR2	UNUSED

If there are 432 opcodes and 32 registers,

a) What is the minimum number of bits required to represent the OPCODE?

9, because $2^9 = 512$.

b) What is the minimum number of bits required to represent the destination register (DR) and the source registers (SR1 and SR2)?

15, because $2^5 = 32$ and we are talking about 3 registers, giving us $5*3 = 15$.

c) What is the maximum number of UNUSED bits in the instruction encoding?

$32 - 9 - 15 = 8$.

Problem 8 (2 points)

a) Assume that a machine cycle takes 1.25 nanoseconds. How many machine cycles happen in one second? (Hint: 1 nanosecond = 10^{-9} seconds).

$1/(1.25 * 10^{-9}) = 8 * 10^8$ machine cycles.

(b) Assume further that a computer needs 16 cycles on average to process one instruction and that the computer processes instructions one at a time from beginning to end. In one second, how many instructions can the computer process?

$(8 * 10^8)/16 = 5*10^7$ instructions.

Problem 9 (2 points)

Name the two components of an instruction and describe what information each contains.

Opcode contains the operation to be performed.
Operands contain the data and locations to be used for the operation.