CS/ECE 252: INTRODUCTION TO COMPUTER ENGINEERING

UNIVERSITY OF WISCONSIN – MADISON

Instructor: Andy Phelps
TAs: Newsha Ardalani, Peter Ohmann, and Jai Menon

<u>Midterm Examination 2</u>
In Class (50 minutes)
Wednesday, March 9
Weight: 15%

<u>NO BOOK(S), NOTE(S), CALCULATOR(S) OF ANY SORT</u>

This exam has 6 front-and-back pages, including a blank page at the end. Plan your time carefully, since

some problems are longer than others. You must turn in pages 1 to 5.

LAST NAME: _____

FIRST NAME: _____

SECTION: _____

ID #: _____

Question	Maximum Points	Points
1	4	
2	4	
3	8	
4	8	
5	4	
6	4	
7	8	
Total	40	

Problem 1 (4 points)

Use the truth table to answer the following questions.

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

a) Write the boolean expression which corresponds to the truth table. Do not simplify the expression.

Hint: Format your answer like: $Z = \_\_\ \_\_\ \_\_ + \_\_\ \_\_\ \_\_ + \ldots$

$Z = \overline{A}\,B\,\overline{C} + A\,\overline{B}\,\overline{C} + A\,B\,C$
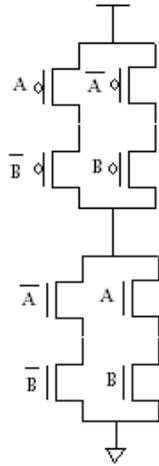
b) Draw the logic gate-level circuit which corresponds to the truth table. Do not simplify the expression.
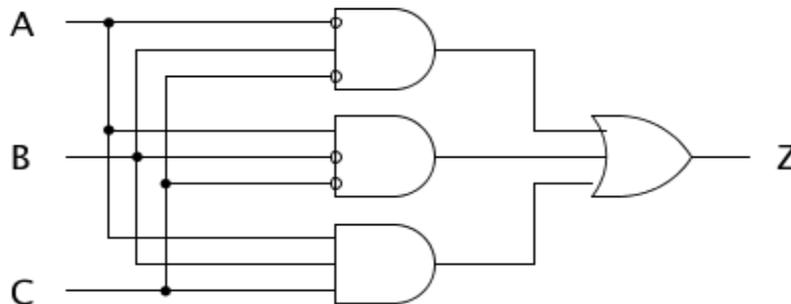
Problem 2 (4 points)

Use the CMOS circuit below to answer the following questions. Inputs with the same name are assumed to be connected to each other. Names with a bar indicate the inversion of that input.

a) Does the circuit above have any flaw? (Yes or No)
   Hint: Evaluate the circuit for all sets of inputs

   NO

b) If it does, identify it. If not, identify the logic function the circuit implements.

   XOR

Problem 3 (8 points)

Suppse a 32-bit instruction takes the following format (not drawn to scale):

OPCODE	--UNUSED--	DEST	SRC1	SRC2

If there are 30 opcodes and 128 registers:

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

   5

b) What is the minimum number of bits required to represent the SRC1, SRC2, and DEST registers?

   (Provide the total number of bits)

   $7 * 3 = 21$

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

   $36 - 21 - 5 = 6$

d) For these questions, you may (and it is probably easier to) provide your answers as 2^n.

   If we decided to redesign the ISA and actually use the UNUSED bits, how many:

   i. opcodes could we have if we turned all the UNUSED bits into more opcode bits?

OPCODE	DEST	SRC1	SRC2

   $2^{5+6} = 2^{11}$

   ii. registers could we have in our computer if we turned all the UNUSED bits into more bits for the DEST, SRC1, and SRC2 fields? Remember, you must distribute the bits across SRC1, SRC2, and DST.
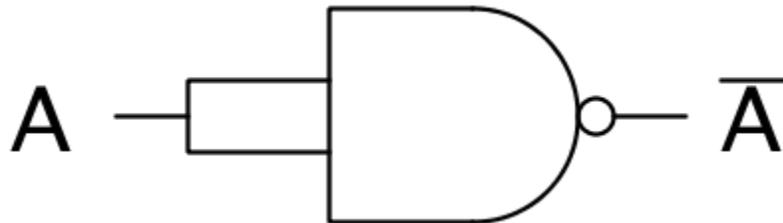
OPCODE	DEST	SRC1	SRC2

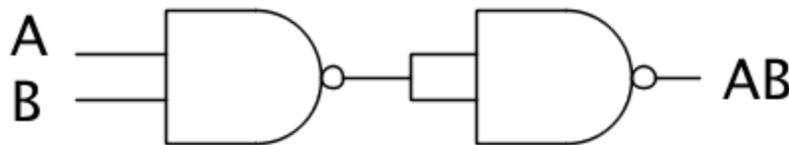 $21 + 6 = 27$ bits. $27 / 3 = 9$. 2^9 registers.

Problem 4 (8 points)

Show that a NAND gate (by itself) is logically complete. To do so,

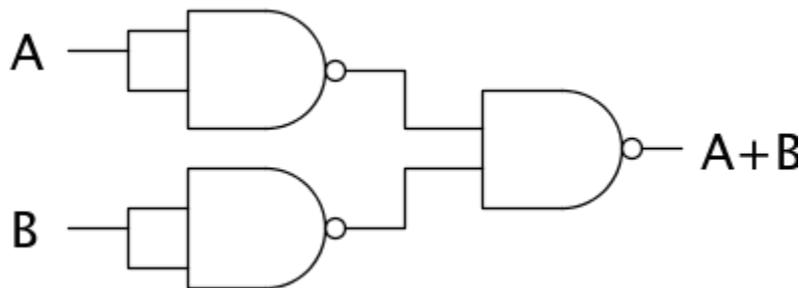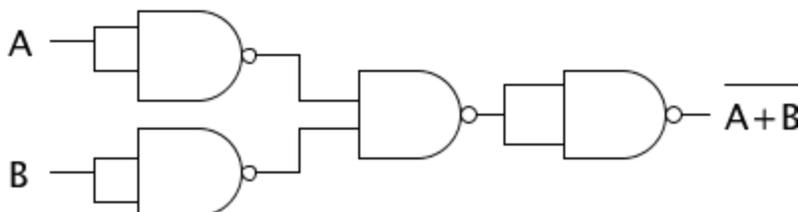   a) Make a NOT gate from only NAND gates (use <u>exactly</u> 1 NAND gate)

$$A \longrightarrow \overline{A}$$

    b) Make an AND gate from only NAND gates (use <u>exactly</u> 2 NAND gates)

$$\begin{matrix} A \\ B \end{matrix} \longrightarrow AB$$

    c) Make an OR gate from only NAND gates (use <u>exactly</u> 3 NAND gates)

$$\begin{matrix} A \\ B \end{matrix} \longrightarrow A+B$$

    d) Make a NOR gate from only NAND gates (use <u>exactly</u> 4 NAND gates)

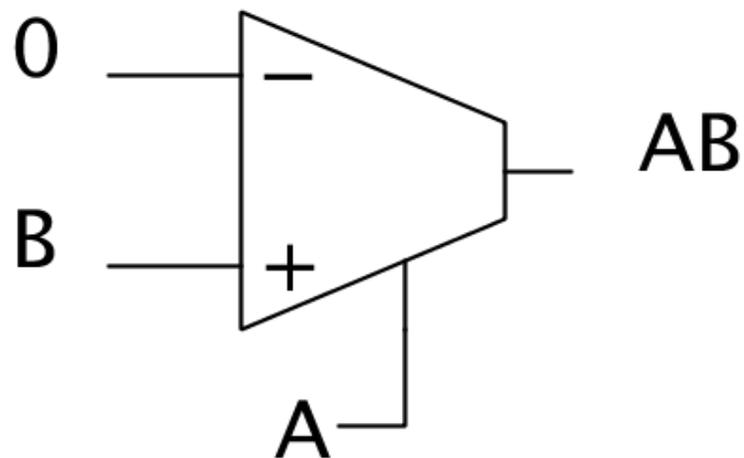$$\begin{matrix} A \\ B \end{matrix} \longrightarrow \overline{A+B}$$

Problem 5 (4 points)

With a 2:1 MUX, we can implement any two-variable function $F(A,B)$ by appropriately connecting 0, 1, A, $\overline{A}$, B, or $\overline{B}$ to the various inputs.
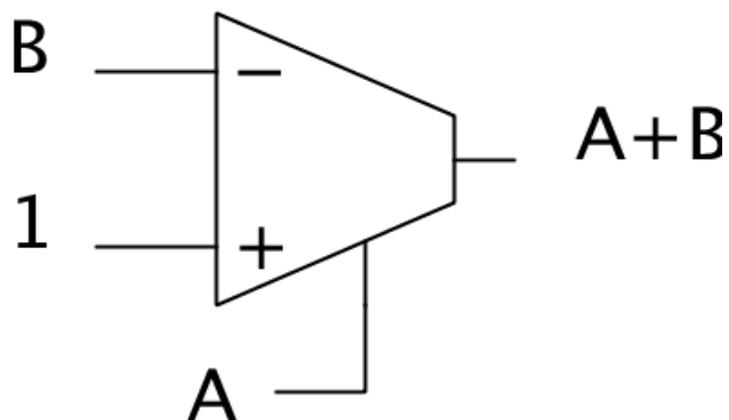
a) Design a 2-input AND gate using only a 2:1 MUX

*(Note: No additional logic elements are to be used)*

b) Design a 2-input OR gate using only a 2:1 MUX

(Note: No additional logic elements are to be used)

Problem 6 (4 points)

The figure below is a finite state machine (FSA) diagram where the start state is S. The machine reads binary input streams from **LEFT TO RIGHT.** Complete the questions below according to the behavior of this machine.
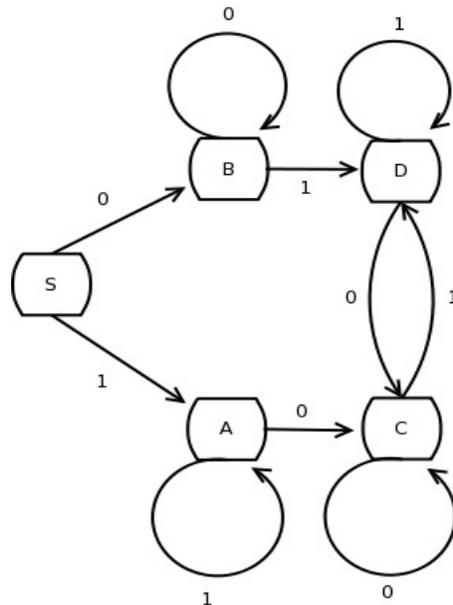
a) For the following input sequences, what is the ending state (i.e. what state is the machine is in after all of the input is read)?

i. 0 1 1 1 1

D

ii. 1 0 0 1 0 0

C

iii. 1 0 1 0 1

D

b) Assume the current state is A. What input sequence of length <u>exactly 2</u> will get the machine to state D?

0 1

Problem 7 (1 point each – 8 points total)

Choose and circle the best answer for each question. Each question has exactly one correct answer.

Multiple Choice

1. The Decode step of the Instruction Cycle always examines which part of the instruction?
    a. Immediate (literal) value
    b. Opcode
    c. Offset
    d. Register
2. What is the difference between a Combinational logic circuit and a Sequential logic circuit?
    a. A change in input for a Sequential logic circuit will propagate immediately to the output, while a Combinational circuit requires a clock pulse before the output changes
    b. A Combinational logic circuit's output depends solely on the values that are present on the inputs now, while a Sequential logic circuit can depend on previous inputs
    c. Combinational logic circuits are built from logic gates (e.g. AND, OR, NOT), while Sequential logic circuits are built from their inversions (e.g. NAND, NOR)
    d. A Combinational logic circuit may have or implement memory, while a Sequential logic circuit operates only based on its current inputs
3. The minimum number of transistors required to implement a CMOS 3-input AND gate is
    a. 4
    b. 6
    c. 8
    d. 10
4. If the number of address bits in a memory is increased by 2 and the addressability is halved, the size of memory (i.e. the number of bits stored in the memory)
    a. Doubles
    b. Remains unchanged
    c. Halves
    d. Increases by 2^(address bits)/addressability

True/False

1. A Decoder would be a standard mechanism for interpreting the opcode part of an instruction.
    a. True
    b. False
2. The MAR is where we put the data we are about to write out to memory.
    a. True
    b. False
3. A Gated-D Latch (with inputs D and Write-Enable) has a restricted combination of inputs; that is, there is a set of inputs which are illegal.
    a. True
    b. False
4. DeMorgan's Law shows that if one inverts both the inputs and output of an AND gate, one has implemented an OR gate.
    a. True
    b. False