

Department of Electrical and Computer Engineering

The University of Texas at Austin

EE 306, Fall 2019

Problem Set 2

Due: 25 September, before class

Yale N. Patt, Instructor

TAs: Sabee Grewal, Arjun Ramesh, Joseph Ryan, Chirag Sakhuja, Meiling Tang, Grace Zhuang

Instructions:

You are encouraged to work on the problem set in groups and turn in one problem set for the entire group. Remember to put all your names on the solution sheet. Also, remember to put the name of the TA and the time for the discussion section you would like the problem set turned back to you. Show your work.

1. What is the largest positive normalized number that can be represented using the IEEE Floating Point standard?
 3.4028235×10^{38}
2. What is the largest positive number that can be represented in a 32 bit 2's complement scheme?
2,147,483,647
3. Professor Patt is trying to decide when he should shave his beard. Your job is to design a logic circuit whose output Y is equal to 1 when Professor Patt should shave his beard, and 0 when he should not. The circuit will receive three input variables (A, B, C) that answer three different yes/no questions (1=yes, 0=no).

A<=Has Professor Patt been uncomfortably warm this summer?

B<=Does Professor Patt want a new, fresher look?

C<=Are beards "cool"?

We think that Professor Patt should shave his beard if he has been uncomfortably warm this summer. He should also shave his beard if he wants a new fresher look and beards are not "cool".

Write the logic equation for Y in terms of A,B,C that solves this problem, and draw the

gate-level diagram.

$$Y = A \text{ OR } (B \text{ AND } (\text{NOT } C))$$

4. (3.11)

- a. Draw a transistor-level diagram for a three-input AND gate and a three-input OR gate. Do this by extending the designs from Figures 3.6a and 3.7a. (Figures can be found in the book on pages 56 & 57 respectively).

For 3-input OR gate: Add one P-type transistor from Vdd to C, in series to the current gate, and one N-type from C to GROUND, in parallel to the current gate.

For 3-input AND gate: Add one P-type transistor from Vdd to C, in parallel to the current gates, and one N-type from C to GROUND, in series to the current gates.

- b. Replace the transistors in your diagrams from part (a) with either a wire or no wire to reflect the circuit's operation when the following inputs are applied:

$$A = 1, B = 0, C = 0$$

Output of 3-input OR will be 1, Output of 3-input AND will be 0.

- c. The transistor circuit shown below produces the accompanying truth table. The inputs to some of the gates of the transistors are not specified. Also, the outputs for some of the input combinations of the truth table are not specified. Complete both specifications. i.e., all transistors will have their gates properly labeled with either A, B, or C, and all rows of the truth table will have a 0 or 1 specified as the output.

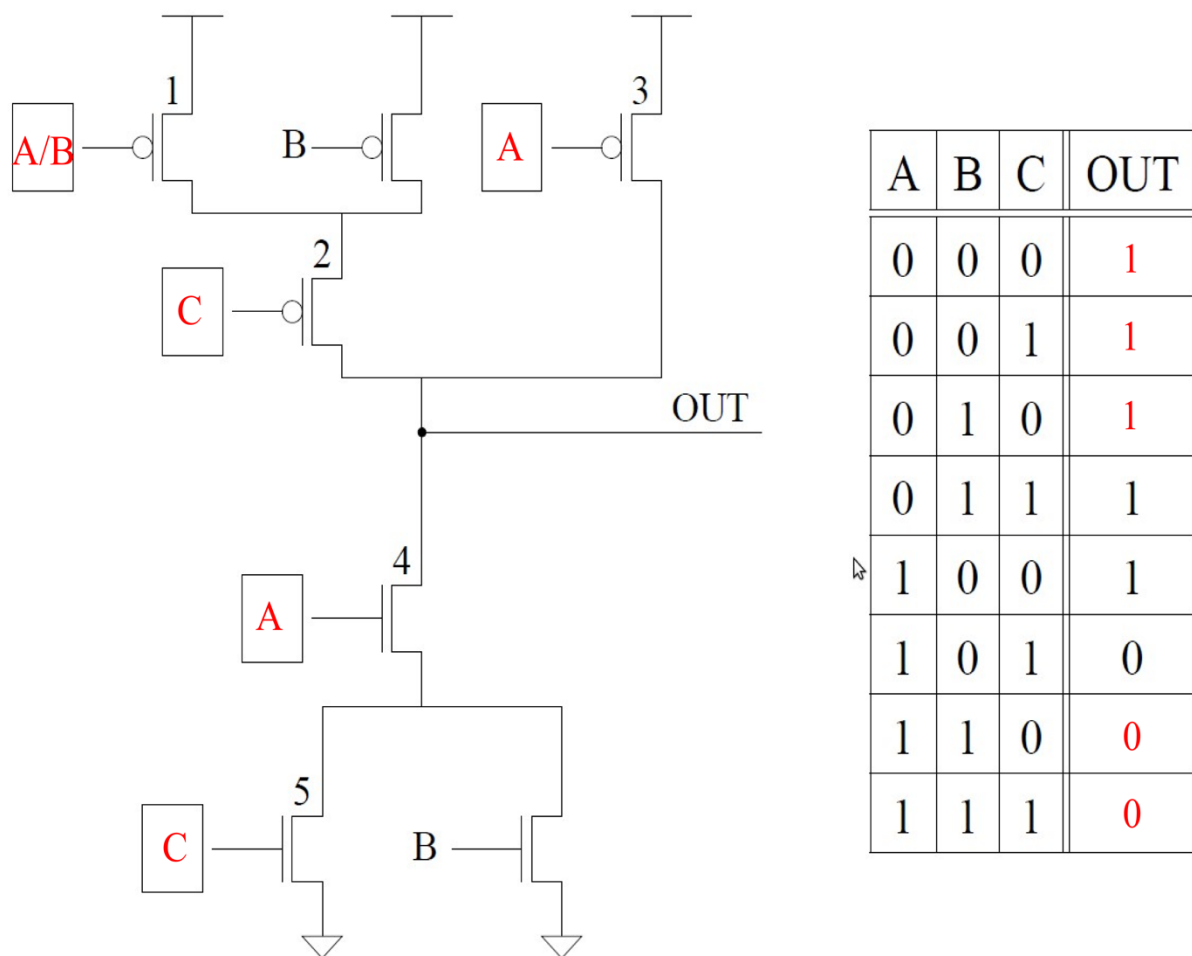


Figure 1

5. Shown below are several logical identities with one item missing in each. X represents the case where it can be replaced by either a 0 or a 1 and the identity will still hold. Your job: Fill in the blanks with either a 0, 1, or X.

For example, in part a, the missing item is X. That is $0 \text{ OR } 0 = 0$ and $0 \text{ OR } 1 = 1$.

- $0 \text{ OR } X = \underline{\hspace{1cm}}$
X
- $1 \text{ OR } X = \underline{\hspace{1cm}}$
1
- $0 \text{ AND } X = \underline{\hspace{1cm}}$
0
- $1 \text{ AND } X = \underline{\hspace{1cm}}$
X
- $\underline{\hspace{1cm}} \text{ XOR } X = X$
0

6. (3.19)

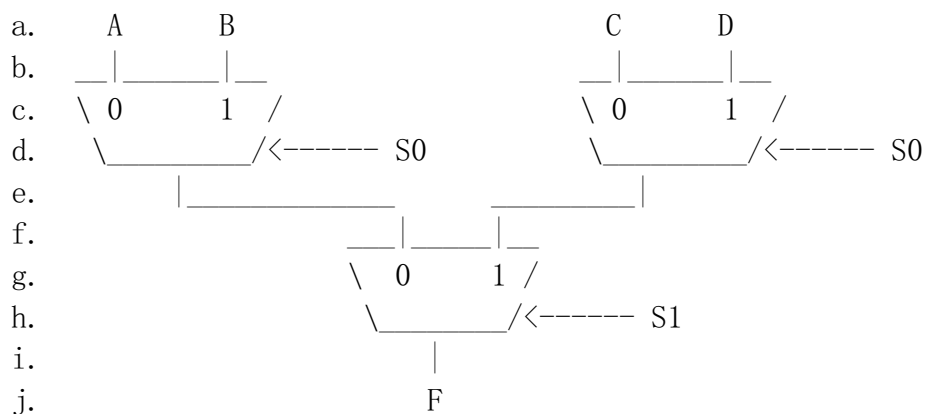
Logic circuit 1 in Figure 3.36 (page 87 of the book) has inputs A, B, C. Logic circuit 2 in Figure 3.37 (page 87 of the book) has inputs A and B. Both logic circuits have an output D. There is a fundamental difference between the behavioral characteristics of these two circuits. What is it? *Hint:* What happens when the voltage at input A goes from 0 to 1 in both circuits?

Figure 3.36 is a 2-input mux, which combinational logic i.e., D is the output of the circuit.

Figure 3.37 is a storage element, which stores the data value previously stored in the latch.

7. (Adapted from 3.22)

A) Implement a 4-to-1 mux using only 2-to-1 muxes making sure to properly connect all of the terminals. Remember that you will have 4 inputs (A, B, C, and D), 2 control signals (S1 and S0), and 1 output (OUT). After implementing the 4-1 mux, fill in the truth table below.



You require 3 muxes. First, the input are A and B and the select line is S0. Second, inputs are C and D and the select line is also S0. Third, is a mux where both its inputs are the outputs of the first two muxes and select line is S1.

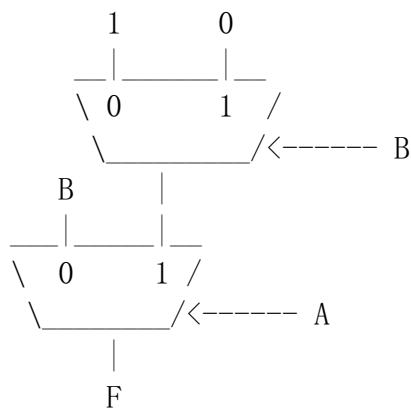
Every time S1 and S0 are both 0, OUT is the same as A.

Every time S1 and S0 are 0 and 1 respectively, OUT is the same as B.

Every time S1 and S0 are 1 and 0 respectively, OUT is the same as C.

Every time S1 and S0 are 1 and 1 respectively, OUT is the same as D.

B) Implement $F = A \text{ xor } B$ using ONLY two 2-to-1 muxes. You are not allowed to use a not gate (A' and B' are not available).



8. (Adapted from 3.25)

Say the speed of a logic structure depends on the largest number of logic gates through which any of the inputs must propagate to reach an output. Assume that a NOT, an AND, and an OR gate all count as one gate delay. For example, the propagation delay for a two-input decoder shown in Figure 3.11 is 2 because some inputs propagate through two gates.

- a. What is the propagation delay for the two-input mux shown in Figure 3.12 (page 61)?

3

- b. What is the propagation delay for the 4-bit adder shown in Figure 3.16 (page 63)?

12

- c. Can you reduce the propagation delay for the circuit shown in Figure 3 by implementing the equation in a different way? If so, how?

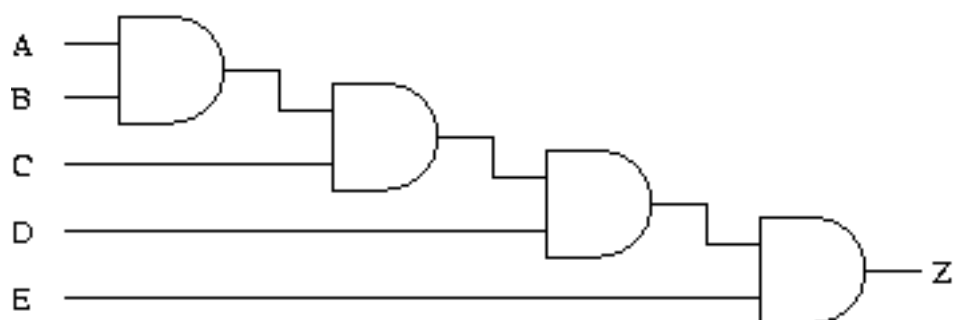


Figure 3

You can construct a tree-like structure.

$$E = ((A \text{ AND } B) \text{ AND } (C \text{ AND } D)) \text{ AND } E)$$

9. (3.26)

Recall that the adder was built with individual "slices" that produced a sum bit and carryout bit based on the two operand bits A and B and the carryin bit. We called such an element a full-adder. Suppose we have a 3-to-8 decoder and two six-input OR gates, as shown in Figure 3 below. Can we connect them so that we have a full-adder? If so, please do. (*Hint: If an input to an OR gate is not needed, we can simply put an input 0 on it and it will have no effect on anything. For example, see the figure below.*)

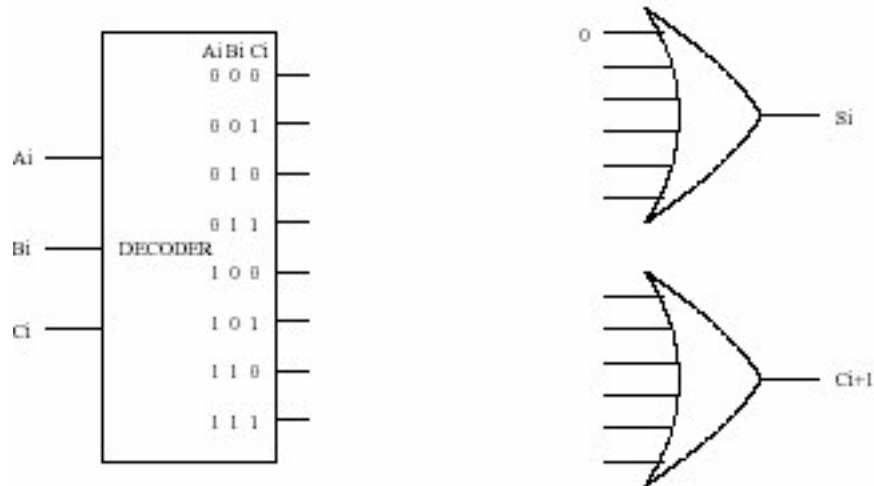


Figure 3

For the OR gate generating the S_i , connect the 4 outputs: 001, 010, 100, and 111 from the decoder to the OR gate. Connect the remaining 2 inputs of the OR gate to 0.

For the OR gate generating the C_{i+1} , connect the 4 outputs: 011, 101, 110, and 111 from the decoder to the OR gate. Connect the remaining 2 inputs of the OR gate to 0.

10. Design the combinational logic circuit for an elevator controller such that the option to go up or down by only one floor is disabled. Assume that the building the elevator is in has 4 floors. Your inputs are the current floor and the next requested floor. The output from the controller should be the next floor to go to. Please show the full truth table for the elevator controller.

Since there are four floors, you will need 2 bits to represent a floor. Let the logic variable $C[1:0]$ represent the current floor, $R[1:0]$ represent the requested floor, and $D[1:0]$ represent the floor the elevator should go to given a current floor and a requested floor. Shown below is the truth table for this combinational logic circuit.

| | | | | | | | |
|-----|-------|----|----|----|--|----|----|
| 11. | C1 | C0 | R1 | R0 | | D1 | D0 |
| 12. | ----- | | | | | | |
| 13. | 0 | 0 | 0 | 0 | | 0 | 0 |
| 14. | 0 | 0 | 0 | 1 | | 0 | 0 |
| 15. | 0 | 0 | 1 | 0 | | 1 | 0 |
| 16. | 0 | 0 | 1 | 1 | | 1 | 1 |
| 17. | 0 | 1 | 0 | 0 | | 0 | 1 |
| 18. | 0 | 1 | 0 | 1 | | 0 | 1 |
| 19. | 0 | 1 | 1 | 0 | | 0 | 1 |
| 20. | 0 | 1 | 1 | 1 | | 1 | 1 |
| 21. | 1 | 0 | 0 | 0 | | 0 | 0 |
| 22. | 1 | 0 | 0 | 1 | | 1 | 0 |
| 23. | 1 | 0 | 1 | 0 | | 1 | 0 |
| 24. | 1 | 0 | 1 | 1 | | 1 | 0 |
| 25. | 1 | 1 | 0 | 0 | | 0 | 0 |
| 26. | 1 | 1 | 0 | 1 | | 0 | 1 |
| 27. | 1 | 1 | 1 | 0 | | 1 | 1 |
| 28. | 1 | 1 | 1 | 1 | | 1 | 1 |

29.

A logic circuit consisting of 6 gated D latches and 1 inverter is shown below:

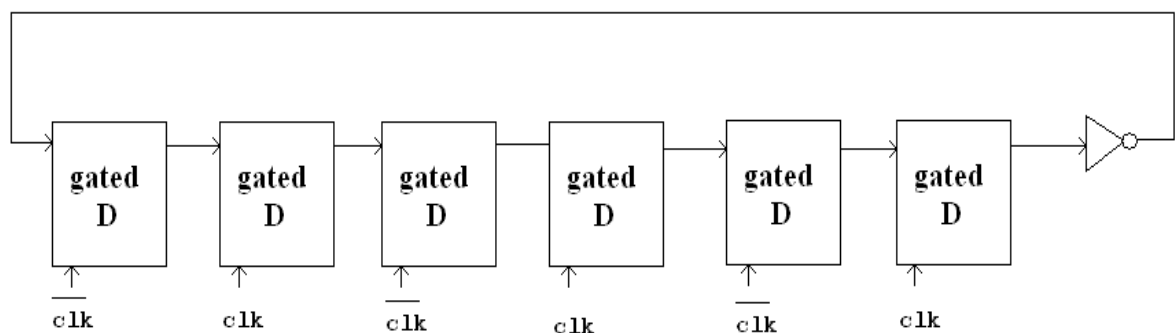


Figure 5

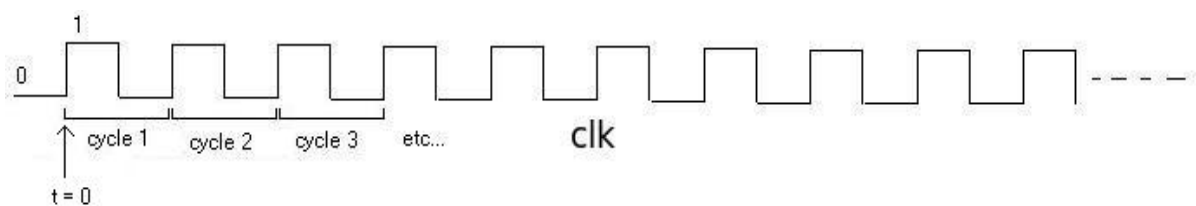


Figure 6

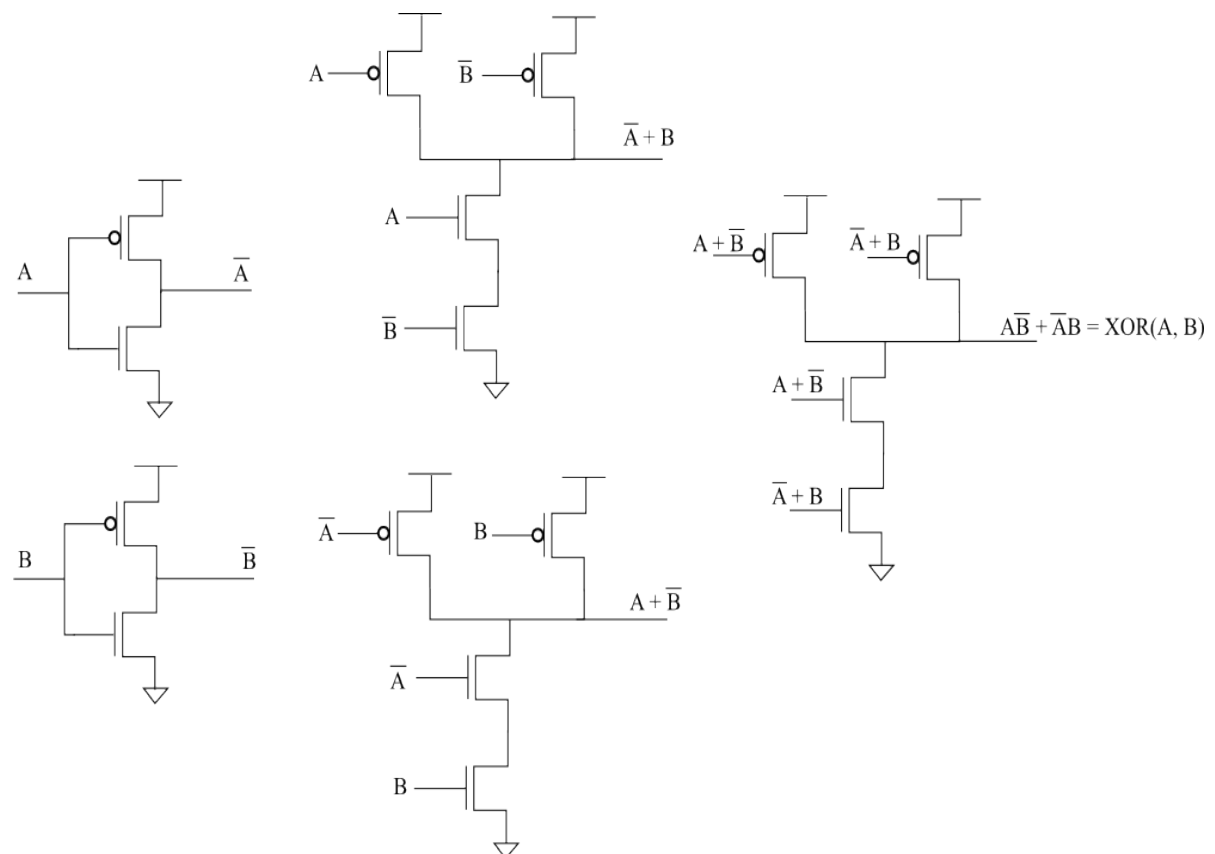
Let the state of the circuit be defined by the state of the 6 D latches. Assume initially the state is 000000 and clk starts at the point labeled t0.

Question: What is the state after 50 cycles. How many cycles does it take for a specific state to show up again?

Every 6 clock cycles a pattern repeats (shown below). Because $50 = 6 \cdot 8 + 2$ after 50 cycles the state will be the same as after 2 cycles. It will be in state 111000 after 50 cycles

30.

Draw the transistor level circuit of a 2 input XOR gate



31. A comparator circuit has two 1-bit inputs, A and B, and three 1-bit outputs, G (greater), E (equal), and L (less than). Refer to figures 3.40 and 3.41 on page 92 in the book for this problem.

a. Draw the truth table for a 1-bit comparator.

| A | B | G | E | L |
|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 |

32. b. Implement G, E and L for a 1-bit comparator using AND, OR, and NOT gates.

33. $G = AB'$, $L = A'B$, $E = A'B' + AB$

34. c. Figure 3.41 performs one-bit comparisons of the corresponding bits of two unsigned integer A[3:0] and B[3:0]. Using the 12 one-bit results of these 4 one-bit comparators, construct a logic circuit to output a 1 if unsigned integer A is larger than unsigned integer B (the logic circuit should output 0 otherwise). The inputs to your logic circuit are the outputs of the 4 one-bit comparators and should be labeled G[3], E[3], L[3], G[2], E[2], L[2], ... L[0]. (Hint: You may not need to use all 12 inputs.)

35. $Y = G[3] + E[3]G[2] + E[3]E[2]G[1] + E[3]E[2]E[1]G[0]$

2. One of Professor Patt's students is always late to meetings, so Professor Patt wants you to design an alarm clock to help his student be on time. Your job is to design a logic circuit whose output Z is equal to 1 when the alarm clock should go off. The circuit will receive four input variables (A, B, C, D) that answer four different yes/no question (1=yes, 0=no):

A <= Is it going to be sunny today?

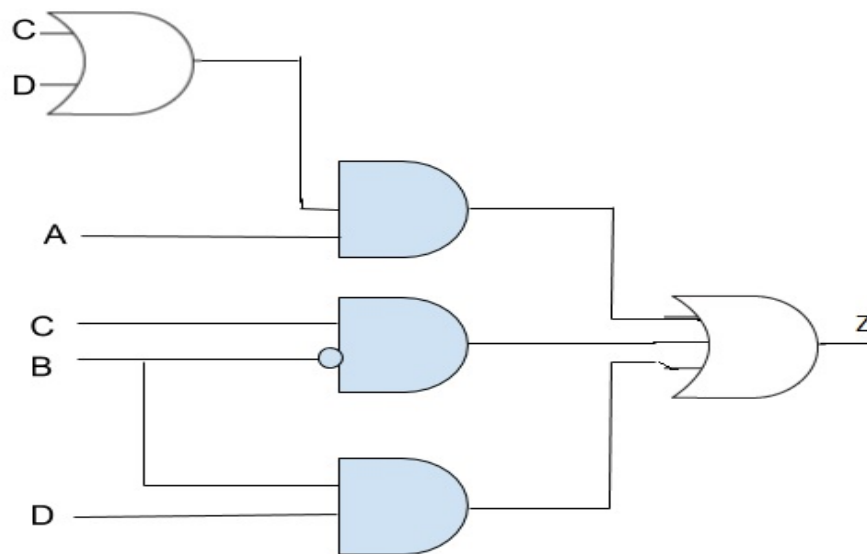
B <= Is it the weekend?

C <= Is it 7:00am?

D <= Is it 9:00am?

Professor Patt wants the alarm clock to go off if it's sunny and it's either 7:00am or 9:00am. The alarm clock should go off if it's the weekend and it's 9:00am. The alarm clock should also go off if it's not the weekend and it's 7:00am. Write the truth table and draw a gate-level diagram that performs this logic.

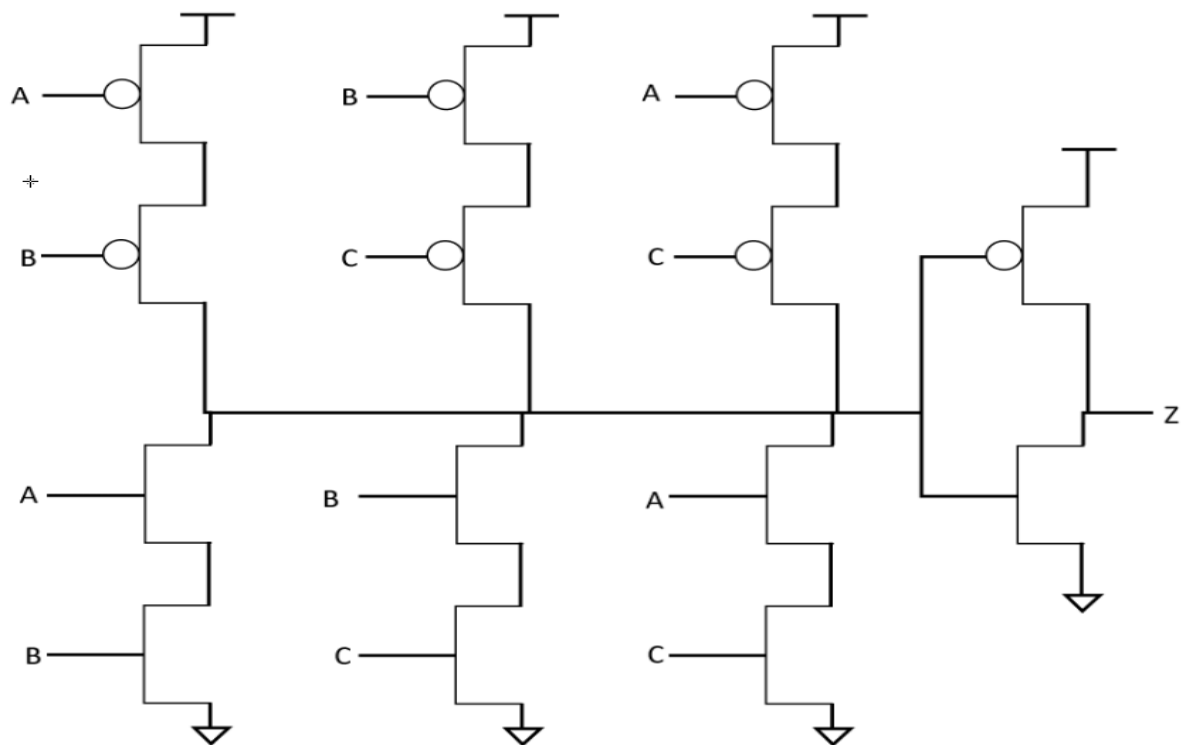
| A | B | C | D | ALARM |
|---|---|---|---|-------|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | x |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | x |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | x |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | x |



3. In class, we showed that $\text{NOT}((\text{NOT } A) \text{ AND } (\text{NOT } B)) == A \text{ OR } B$ using their truth tables. This equivalence is known as DeMorgan's law. Similarly, show that $\text{NOT}((\text{NOT } A) \text{ OR } (\text{NOT } B)) == A \text{ AND } B$.

| A | B | NOT A | NOT B | (NOT A) OR (NOT B) | NOT ((NOT A) OR (NOT B)) | A AND B |
|---|---|-------|-------|--------------------|--------------------------|---------|
| 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 | 1 |

4. Draw the transistor level circuit of a 3 input majority gate.



5. **Updated on 09/21/2019.** Prove that NAND is logically complete.
NOT: $A \text{ NAND } A \Rightarrow \text{NOT } A$
AND: $(A \text{ NAND } B) \text{ NAND } (A \text{ NAND } B) = \text{NOT } (A \text{ NAND } B) \Rightarrow A \text{ AND } B$