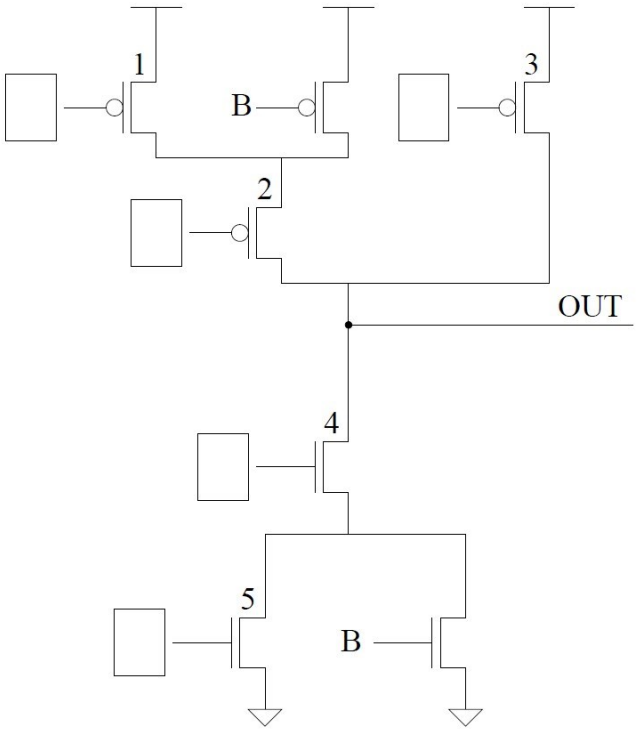
Homework02

1. According to the following equations, draw the circuits and write the matching truth tables.The circuits can be drawn either in transistor-level or symbols.
   a. X = NOT (NOT(A) OR (A AND B AND C))
   b. Y = NOT ((A OR B) AND (A AND B AND C))

2. a. How many ways can you use gates to implement Z = A XOR B? Try to figure them out as many as possible and draw them down.(i.e. NOR gate/2-1 mux …).How many transistors at least will you use?

   b. (3.22) 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, write out the truth table.

3. 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.



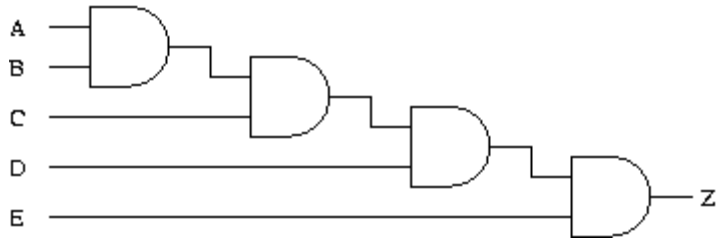
A	B	C	OUT
0	0	0	
0	0	1	
0	1	0	
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	
1	1	1	

4. (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?

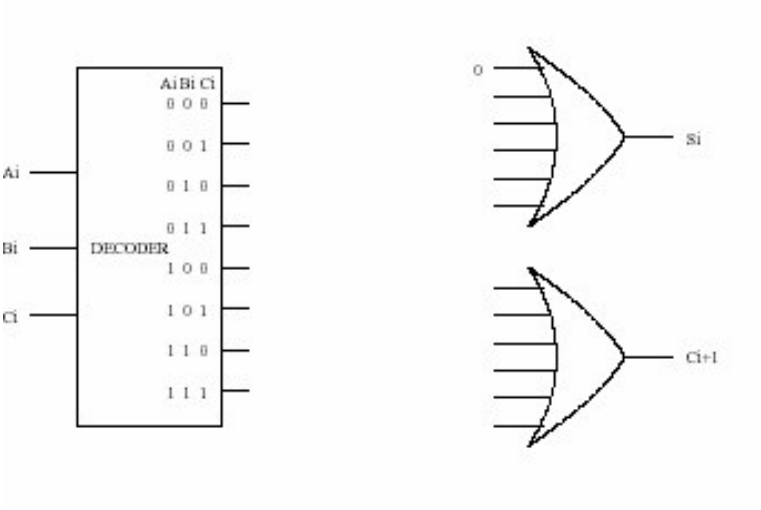
5. (Adapted from 3.25)

   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)?
b. What is the propagation delay for the 4-bit adder shown in Figure 3.16 (page 63)?
c. Can you reduce the propagation delay for the circuit shown in the Figure below by implementing the equation in a different way? If so, how?

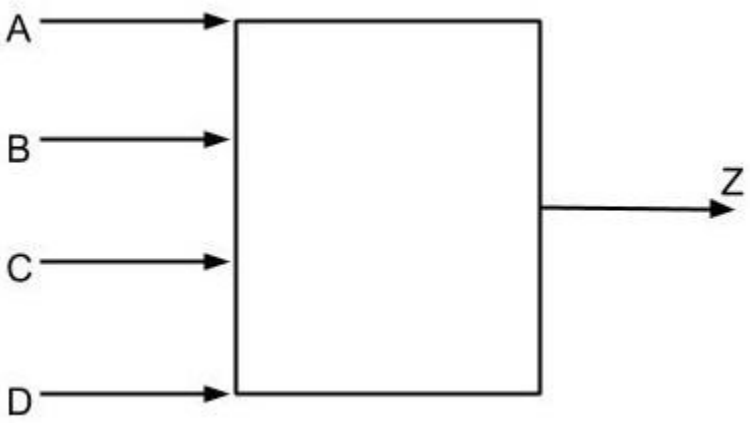


6. (3.26) Recall that the adder was built with individual "slices" that produced a sum bit and carryout bit based on the two operand bits A and B and the carryin bit. We called such an element a full-adder. Suppose we have a 3-to-8 decoder and two six-input OR gates, as shown in Figure 3 below. Can we 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.)



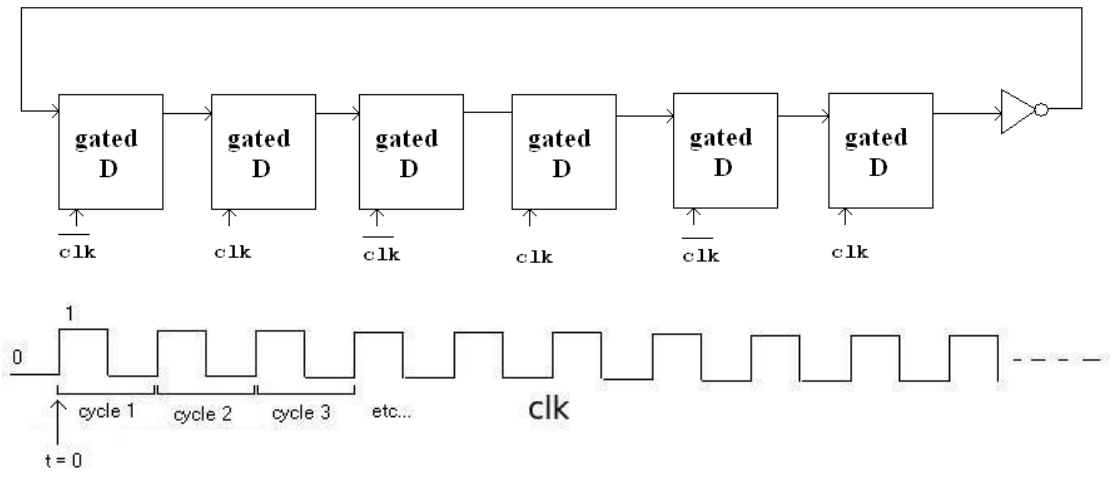
7. One of Professor Patt's students is always late to meetings, so Professor Patt wants you to design an alarm clock to help his student be on time. Your job is to design a logic circuit whose output Z is equal to 1 when the alarm clock should go off. The circuit will receive four input variables (A, 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.



8. (Adapted from 3.30)
   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.
   b. Implement G, E and L for a 1-bit comparator using AND, OR, and NOT gates.
   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.)

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



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 cyles. How many cycles does it take for a specific state to show up again?

10. We want to make a state machine for the scoreboard of the Texas vs. Oklahoma Football game. The following information is required to determine the state of the game:
    1) Score: 0 to 99 points for each team
    2) Down: 1, 2, 3, or 4
    3) Yards to gain: 0 to 99
    4) Quarter: 1, 2, 3, 4
    5) Yardline: any number from Home 0 to Home 49, Visitor 0 to Visitor 49, 50
    6) Possesion: Home, Visitor
    7) Time remaining: any number from 0:00 to 15:00, where m:s (minutes,

seconds)
a. What is the minimum number of bits that we need to use to store the state required?
b. Suppose we make a separate logic circuit for each of the seven elements on the scoreboard, how many bits would it then take to store the state of the scoreboard?
c. Why might the method of part b be a better way to specify the state than the method of part a?

11. We wish to design a controller for an elevator such that if you push a button for a desired floor, the controller will output the floor number that the elevator should go to. However, if there is a request to move only one floor or to move zero floors, the elevator should remain on the current floor. There are four floors in the building.

    a. Construct a complete truth table for the elevator controller. It is not necessary to draw the logic here; the truth table is sufficient.

    b. Design the state machine for the sequential logic circuit for an elevator controller which performs the same operation. The input to the state machine is the next requested floor. There will be a state for each floor the elevator could be on. Draw a finite state machine that describes the behavior of the elevator controller.

       Hint:
       a. What information does the controller need in order to output the floor to go to?
       b. How many input bits will that require?
       c. How many output bits will the controller have to supply?