# Textbook Challenge Questions:

## *Chapter 1&2&3*

1. **The solutions to the quadratic equation $x^2 - 11x + 11 = 0$ are $x = 3$ and $x = 6$. What is the base of the numbers?**

2. Determine the base of the numbers in each case for the following operations to be correct:

   a. $\frac{14}{2} = 5$

   b. $\frac{54}{4} = 13$

   c. $24 + 17 = 40$

3. Find the 16's complement of C3DF.

   a. Convert C3DF to binary.

   b. Find the 2's complement of the binary result in part a).

   c. Convert the answer in part b) to hexadecimal and compare with the answer in a).

4. Perform the subtraction on the unsigned numbers using the 10's complement of the subtrahend. Where the result should be negative, find its 10's complement and affix a minus sign. Verify your answers.

5. **Perform the subtraction on the given unsigned binary numbers using the 2's complement of the subtrahend. Where the result should be negative, find its 2's complement and affix a minus sign.**

6. Convert decimal +49 and +29 to binary, using the signed-2's complemented representation and enough digits to accommodate the numbers. Then perform the binary equivalent of the following. Convert the answers back to decimal and verify that they are correct.

   a. $(+29) + (-49)$

   b. $(-29) + (+49)$

   c. $(-29) + (-49)$

7. **The following decimal numbers are shown in sign-magnitude form: +9286, and +801. Convert them to signed-10's complement form and perform the following operations.**

   a. (+9286)+(+801)

   b. (+9286)+(-801)

   c. (-9286)+(+801)

   d. (-9286)+(-801)

8. **Formulate a weighted binary code for the decimal digits using the following weights:**

   a. 6, 3, 1, 1

   b. 6, 4, 2, 1

9. **Assign a binary code in some orderly manner to the 52 playing cards. Use the minimum number of bits.**

**10. Show that the dual of $(x + y')z$ is equal to its complement.**

11. Simplify the following functions, and implement them with two-level NAND gate circuits:

a. $F(A,B,C,D) = AC'D' + A'C + ABC + AB'C + A'C'D'$

b. $F(A,B,C,D) = A'B'C'D + CD + AC'D$

c. $F(A,B,C) = (A' + C' + D')(A' + C')(C' + D')$

d. $F(A,B,C,D) = A' + B + D' + B'C$

12. Draw a NAND logic diagram that implements the complement of the following function:

$$F(A, B, C, D) = \sum m(0, 1, 2, 3, 6, 10, 11, 14)$$

13. Simplify the following functions, and implement them with two-level NOR gate circuits:

a. $F = wx' + y'z' + w'yz'$

b. $F(w, x, y, z) = \sum m(0, 3, 12, 15)$

c. $F(x, y, z) = [(x + y)(x + z')]'$

14. Implement the following Boolean function $F$, using the two-level forms of logic:

a. NAND-AND

b. AND-NOR

c. OR-NAND

d. NOR-OR

15. Derive the circuits for a three-bit parity generator and four-bit parity checker using an odd parity bit.

16. Implement the following four Boolean expressions with three half-adders:

$$D = A \oplus B \oplus C$$
$$E = A'BC + AB'C$$
$$F = ABC' + (A' + B')C$$
$$G = ABC$$

# Chapter 4

1. –

   a. Construct a BCD to 9's complement converter using four 2X1 MUX.

   b. Construct a BCD adder-subtractor circuit that takes the 9's CF as the Addend. Use block diagrams for the components.

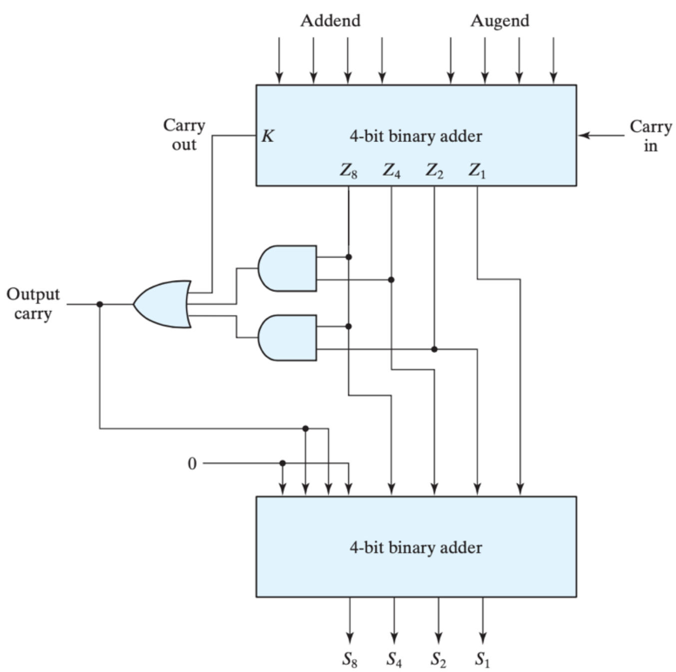



**FIGURE 4.14**
Block diagram of a BCD adder

2. Construct a 5-32 line decoder with four 3-to-8-line decoders with enable and a 2-to-4 line decoder. Use block diagrams for the components.

3. Design a four-input priority encoder with the inputs stated below, but with input $D_0$ having the highest priority and input $D_3$ the lowest priority.

**Table 4.8**
*Truth Table of a Priority Encoder*

| Inputs | | | | Outputs | | |
|---|---|---|---|---|---|---|
| $D_0$ | $D_1$ | $D_2$ | $D_3$ | $x$ | $y$ | $V$ |
| 0 | 0 | 0 | 0 | X | X | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| X | 1 | 0 | 0 | 0 | 1 | 1 |
| X | X | 1 | 0 | 1 | 0 | 1 |
| X | X | X | 1 | 1 | 1 | 1 |

4. Construct a 16X1 multiplexer with two 8X1 and one 2X1 multiplexers. Use block diagrams.

5. Implement the following Boolean functions with a multiplexer:

   a. $F(A, B, C, D) = \sum m(0, 2, 5, 8, 10, 14)$

   b. $F(A, B, C, D) = \prod M(2, 6, 11)$

6. Implement a full adder with two 4X1 multiplexers.

7. An 8X1 multiplexer has inputs $A, B, and\ C$ connected to the selection inputs $S_2, S_1, and\ S_0$, respectively. Determine the Boolean function that the multiplexer implements given the following inputs:

   a. $I_1 = I_2 = I_7 = 0; I_3 = I_5 = 1; I_0 = I_4 = D; and\ I_6 = D'$

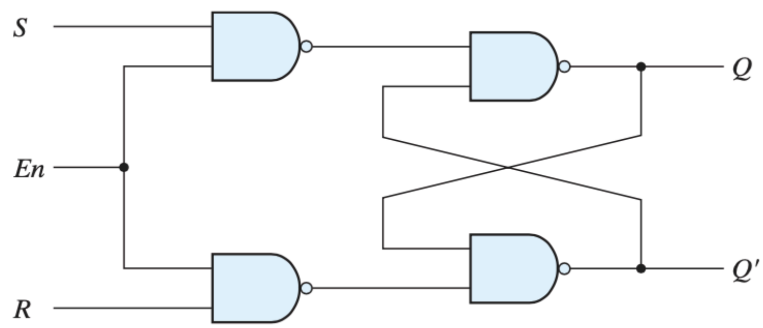
8. Implement the following Boolean function with a $4X1$ multiplexer and external gates.

   a. $F_1(A, B, C, D) = \sum m(1, 3, 4, 11, 12, 13, 14, 15)$

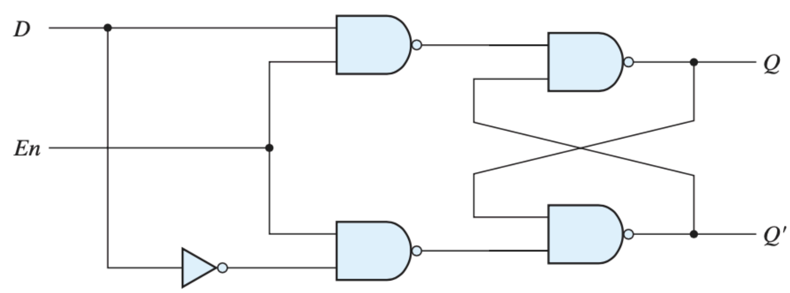
      Connect inputs $A\ and\ B$ to the selection lines. The input requirements for the four data lines will be a function of variables $C\ and\ D$. These values are obtained by expressing $F$ as a function of $C\ and\ D$ for each of these four cases when $AB = 00, 01, 10, and\ 11$. These functions may have to be implemented with external gates.

# *Chapter 5*

1. The *D* latch is constructed with four NAND gates and an inverter. Consider the following three ways for obtaining a *D* latch. In each case, draw the logic diagram and verify the circuit operation




(a) Logic diagram




(a) Logic diagram

   a. Use NOR gates for the SR latch part and AND gates for the other two. An inverter may be needed.

   b. Use NOR gates for all four gates. Inverters may be needed.

   c. Use four NAND gates only (without an inverter).

2. Construct a JK flip-flop using a D flip-flop, a 2-to-1 multiplexer, and an inverter.

3. **Explain the differences among a truth table, a state table, a characteristic table, and an excitation table. Also explain the difference among a Boolean equation, a state equation, a characteristic equation, and a flip-flop input equation.**

4. A sequential circuit with two $D$ flip-flops $A$ and $B$, two inputs $x\ and\ y$, and one output $z$ is specified by the following next-state and output equations.

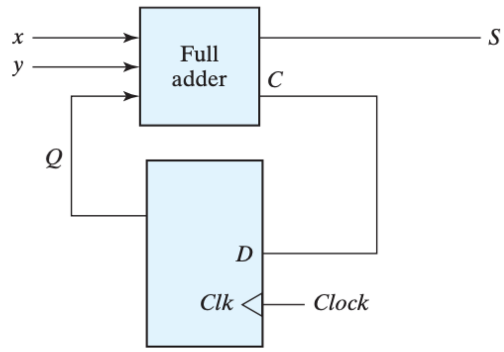
$$A^+ = xy' + xB$$
$$B^+ = xA + xB'$$
$$z = A$$

a. Draw the logic diagram for the circuit.

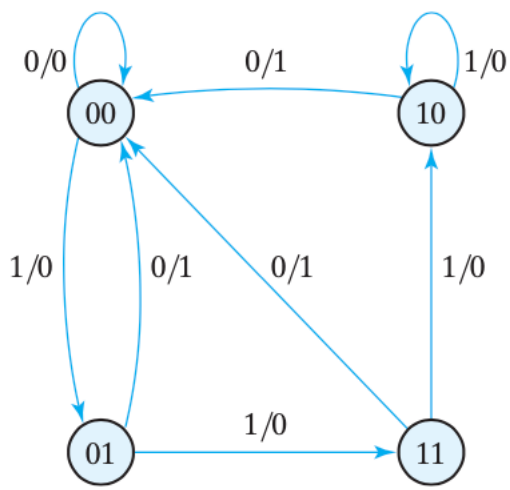
b. List the state table for the sequential circuit.

c. Draw the corresponding state diagram.

5. A sequential circuit has one flip-flop $Q$ and two inputs $x$ and $y$, and one output $S$. It consists of a full-adder circuit connected to a $D$ flip-flop, as shown below. Derive the state table and state diagram of the sequential circuit.



6. For the circuit described by the state diagram below:




a. Determine the state transitions and output sequence that will be generated when the input sequence is 010110111011110 is applied to the circuit and it is initially in the state 00.

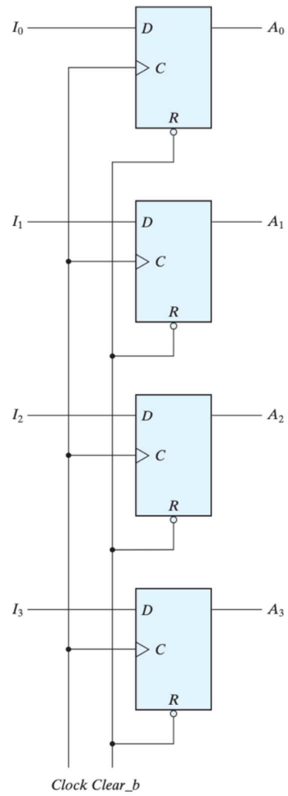
b. Find all of the equivalence states and draw a simpler, but equivalent, state diagram.

c. Using D flip-flops, design the equivalent machine (including its logic diagram) described by the state diagram in part b).

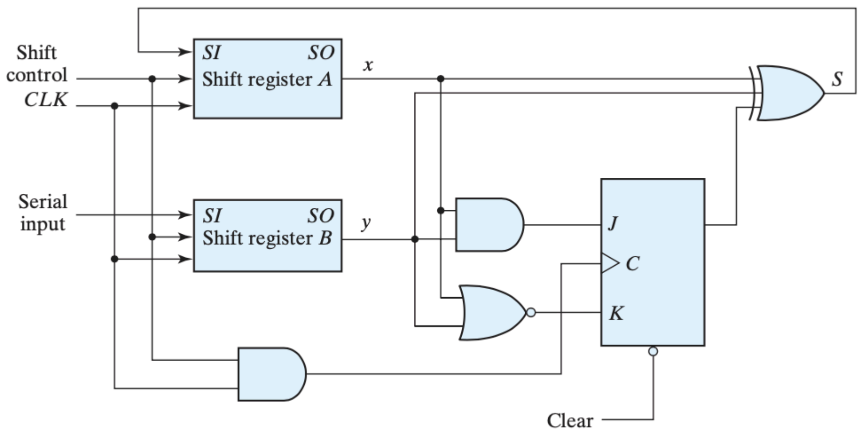
7. **Design a one-input, one-output serial 2's comeplementer. The circuit accepts a string of bits from the input and generates the 2's compelement of the output. The circuit can be reset asynchronously to start and end the operation.**

# Chapter 6

1. Include a 2-input NAND gate in the register below, and connect the gate output to the $C$ inputs of all the flip-flops. One input of the NAND gate receives the clock pulses from the clock generator, and the other input of the NAND gate provides a parallel load control. Explain the operation of the modified register. Explain why this circuit might have operational problems. [Aziz said a question like this will be on the final].

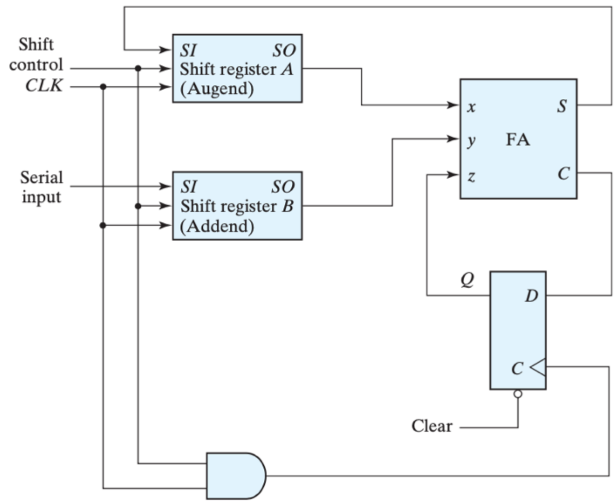


2. The serial adder below uses two four-bit registers. Register *A* holds the binary number 0101 and register *B* holds 0111. The carry flip-flop is initially reset to 0. List the binary values in register *A* and the carry flip-flop after each shift.

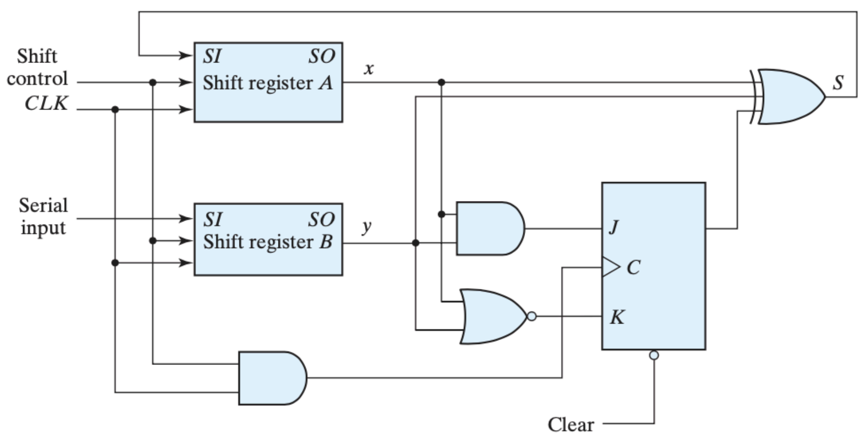


3. Two ways for implementing a serial adder is shown below. It is necessary to modify the circuits to convert them to serial subtractors.

   a. Using the circuit below, show the changes needed to perform the $A + 2's\ CF\ of\ B$.




   b. Using the circuit below, show the changes needed by modifying the table below from an adder to a subtractor circuit.




| Present State | Inputs | | Next State | Output | Flip-Flop Inputs | |
|---|---|---|---|---|---|---|
| $Q$ | $x$ | $y$ | $Q$ | $S$ | $J_Q$ | $K_Q$ |
| 0 | 0 | 0 | 0 | 0 | 0 | X |
| 0 | 0 | 1 | 0 | 1 | 0 | X |
| 0 | 1 | 0 | 0 | 1 | 0 | X |
| 0 | 1 | 1 | 1 | 0 | 1 | X |
| 1 | 0 | 0 | 0 | 1 | X | 1 |
| 1 | 0 | 1 | 1 | 0 | X | 0 |
| 1 | 1 | 0 | 1 | 0 | X | 0 |
| 1 | 1 | 1 | 1 | 1 | X | 0 |

4. **Design a serial 2's complementor with a shift register and a flip-flop. The binary number is shifted out form one side and it's 2's complementor is shifted into the other side of the shift register.**

5. Draw the logic diagram of a four-bit binary ripple countdown counter using:
   [a form of this will be on the exam]

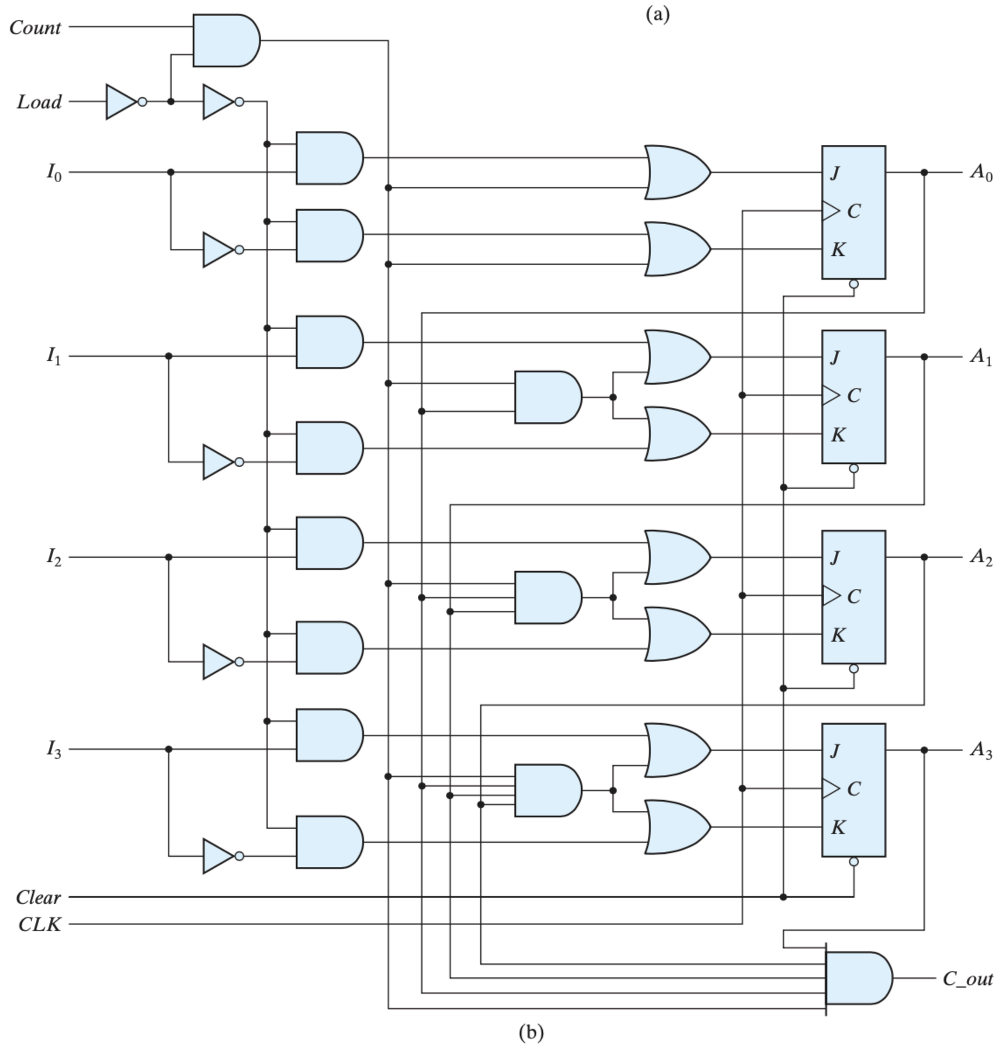
   a. Flip-flops that trigger on the positive-edge of the clock.

   b. Flip-flops that trigger on the negative-edge of the clock.

6. Show that a BCD ripple counter can be constructed using a four-bit binary ripple counter with asynchronous clear and a NAND gate that detects the occurrence of count 1010.

7. A flip-flop has a 3 ns delay from the time the clock edge occurs to the time the output is complemented. What is the maximum delay in a 10-bit binary ripple counter that uses these flip-flops? What is the maximum frequency at which the counter can operate reliably? [Aziz said this could be on the exam]

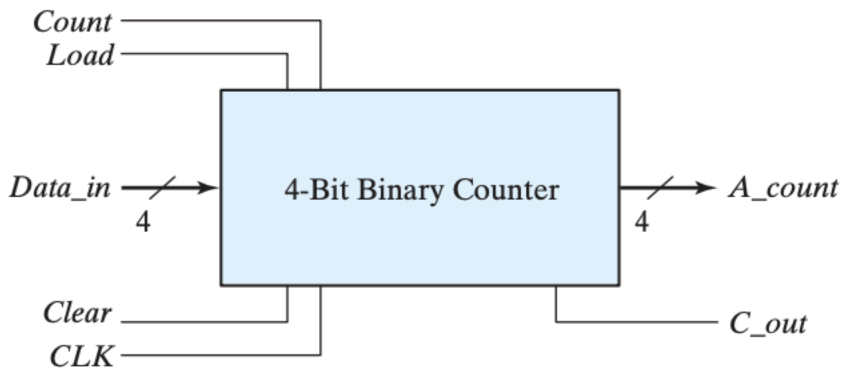
8. Enclose the binary counter with parallel load (shown below) in a block diagram showing all the inputs and outputs.




a. Show the connections of four such blocks to produce a 16-bit counter with parallel load.

b. Construct a binary counter that counts from 0 through binary 127.

9. For the circuit below, give three alternatives for a mod-10 counter. (The count evolves through a sequence of 12 distinct states)




(a)

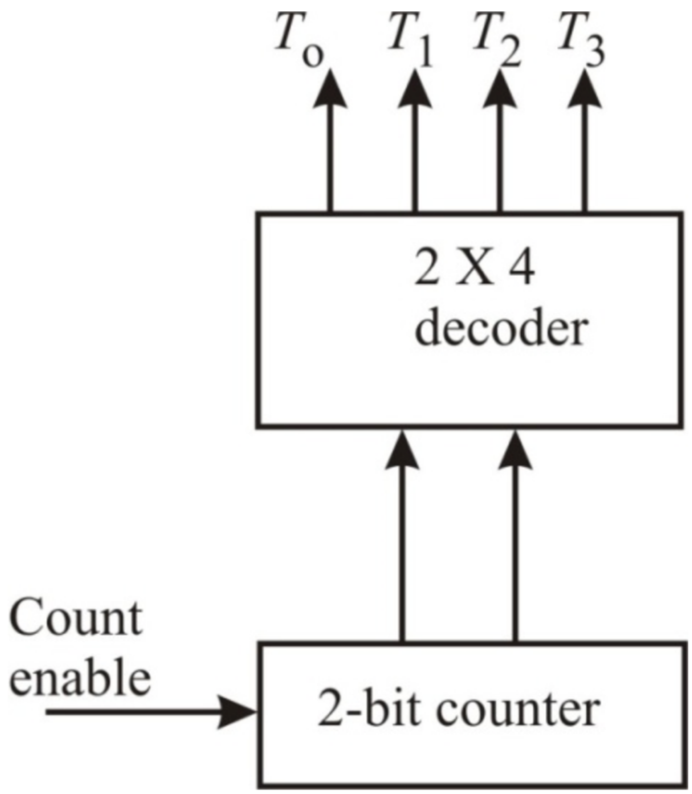
a. Using an AND gate and the load input.

b. Using the output carry.

c. Using a NAND gate and the asynchronous clear input.

10. Design a counter with *T* flip-flops that goes through the following binary repeated sequence: 0, 1, 3, 7, 6, 4. Show that when binary states 010 and 101 are considered as don't care conditions, the counter may not operate properly. Find a way to correct the design. [Major for Aziz's section.]

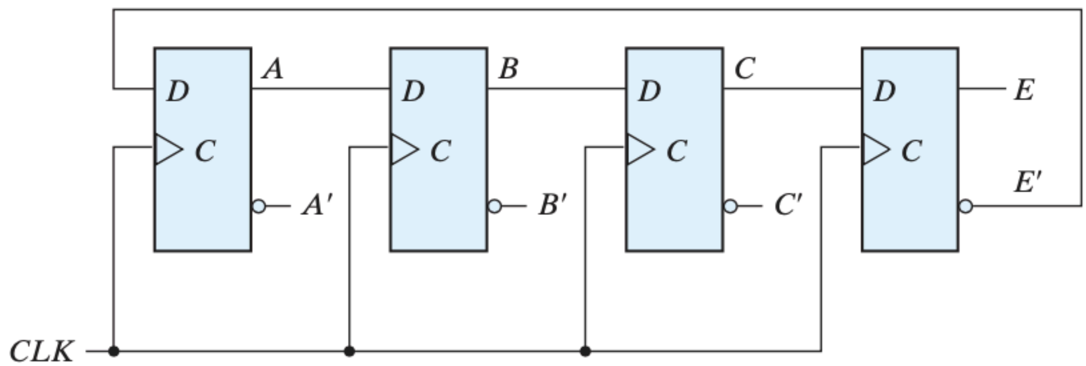
11. It is necessary to generate six repeated timing signals $T_0$ *through* $T_5$ similar to the ones shown below. Design the circuit using:




a. Flip-flops only.

b. A counter and a decoder.

12. List the eight unused states in the switch-tail ring counter below. Determine the next state for each of these states and show that, if the counter finds itself in an invalid state, it does not return to a valid state. Modify the circuit and so that the counter produces the same sequence of states and that the circuit reaches a valid state from any one of the unused states.



# Funky Exam Practice Examples

1. **The subsequent signed binary numbers are represented in 1's complement notation. Perform the following operations. (Your results should be represented using the same notation as the operands). Check your operation by converting both your operands and your results into decimal numbers and indicate the cases where overflow occurs.**

    a. **001011-111011**

    b. **011111+010000**

    c. **100111+101000**

2. A BCD to 7-segment decoder has 4 inputs $(W, X, Y, Z)$ and 7 outputs $(a, b, c, d, e, f, g)$ that select the correspondent segments of the LED display represented in the figure below. The numeric representation of the decimal numbers is given in the figure below as well.


a. Consider the decoder's outputs to being in "don't care" states (marked with "d") or "X") for any of the 6 unused input combinations.

   i. Find the minimized expression for the output $c$ only.

   ii. Draw the logic diagram of the two-level NOR circuit that implements the above expression, $c$.

b. Redo part a) considering that the decoder has to display the letter "E" (for Error) on the 7-segment display if any of the unused combinations is presented to the decoder's inputs.

3. Given the following Boolean functions:

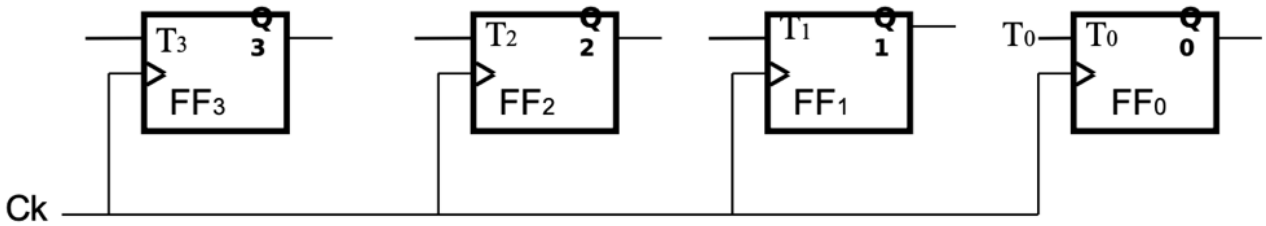
$F_1(a, b, c) = \sum m(1, 4, 7)$ with don't care inputs $\sum d(0, 2)$
$F_2(a, b, c) = \sum m(2, 1, 5, 6)$

a. Use a NAND (active low) implementation decoder with external NAND gates only to implement $F_1$ in its product-of-sum (POS) form. (Assume NAND gates with any number of inputs are available).

b. Use 2-to-4 decoders to build the necessary decoder to implement $F_2$. Use an external OR gate at the output of the decoder. (Assume OR gates with any number of inputs are available) to implement $F_2$ in its sum-of-products form.

4. The figure below shows four T-type flip-flops clocked for synchronous counting. Using AND gates, design a logic circuit such that at the positive edge clock, $FF_1$ changes the states when $Q_0 = 1$, $FF_2$ changes the states when $Q_0Q_1 = 1$, and $FF_3$ changes the states when $Q_0Q_1Q_2 = 1$.

   a. Complete the figure below to include your logic with AND gates.




   b. What is the function of this circuit if $T_0 = 1$?

   c. What is the function of this circuit if $T_0 = 0$?

   d. What is the modulus of this counter?

   e. What is the maximum modulus of this counter?

5. Realize (develop) the circuit for full adder using 3-8 line decoder with:
    a. Two OR gates
        i. Write the truth table.
        ii. Write the simplified expressions for the outputs, (using K-map)
        iii. Draw the circuit.
    b. Two NOR gates
        i. Write the truth table.
        ii. Write the simplified expressions for the outputs, (using K-map)
        iii. Draw the circuit.

6. **Derive the logic equations for a 4-to-2 priority encoder. Write the truth table first and show the highest and lowest priority inputs. Use K-map to simplify the expressions.**

7. Design a combinational circuit that generates the 10's complement of a BCD (binary Coded Decimal) digit.
    a. Build the truth table of your circuit.
    b. Simplify the outputs in their sum-of-products form using K-maps.
        i. List all prime implicants.
        ii. List essential prime implicants.
        iii. Draw your circuit with NAND gates only.

8. Given the following Boolean functions:

$$F(a, b, c) = \sum m(0, 2, 3, 7)$$
$$G(a, b, c) = \sum m(1, 4, 6, 7)$$

a. Use a NAND implementation decoder with external NAND gates only to implement $F$ and $G$ in their sum-of-products (SOP) form. (Assume that any type of gate with any number of inputs is available).

b. Use a multiplexer to implement $G$.

9. Design a synchronous BCD counter. Use negative edge-triggered D Flip-flops provided with a clock.

   a. Draw the state diagram of the counter.

   b. Build the counter's state table showing the synchronous inputs of the D flip-flops as well.

   c. Using K-maps, find the minimal sum-of-products (SOP) of the equations for the inputs to the flip-flops. Assume the next states of the unused combinations to be "don't care" states.

   d. Draw the logic diagram of the counter.

   e. Modify your diagram to show how your BCD counter can be cleared when it reaches the value 7. Use active low clear and NAND gates.

10. Design a serial ADDER with JK flip-flops and draw the corresponding logic circuit. The two binary numbers to be added, $X$ and $Y$, are stored in two shift registers $A$ and $B$ respectively. The result (sum) of the adder must be stored in the register $B$. Use a full adder to add the two numbers. Answer the following questions:

   a. Give a table that shows:

      i. The inputs and outputs for the full-adder

      ii. The states (present and next) for the JK flip-flop

      iii. The JK flip-flop inputs.

   b. Obtain the logic expressions for the JK inputs and the outputs of the full-adder.

   c. Draw the logic circuit of the serial adder. (Use a diagram to represent the full adder and the registers).

11. Design a code converter that converts a decimal digit from the code 8, 4, -2, -1 to 8, 4, 2, 1 code.

    a. Build the truth table for this converter.

    b. Design the circuit with a decoder.

12. Design a 4-bit combinational circuit for a 2's complementor. The circuit generates at the output the 2's complement of the input binary numbers.

a. Complete the following truth table. $A, B, C, D$ indicates the input binary number to be complemented using 2's complement and $W, X, Y, Z$ represents the output 2's complement of the input binary number. The variable $D$ is the least significant bit and $A$ is the most significant bit of the binary number.

| A | B | C | D | W | X | Y | Z |
|---|---|---|---|---|---|---|---|
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |

b. Obtain the Boolean function $W$ in its sum-of-products (SOP). Use K-map to simplify $W$ and draw its corresponding logic circuit.

c. Obtain the Boolean function $Z$ in its product-of-sums (POS). Use K-map to simplify $Z$.

d. Use a decoder to implement $Z$.