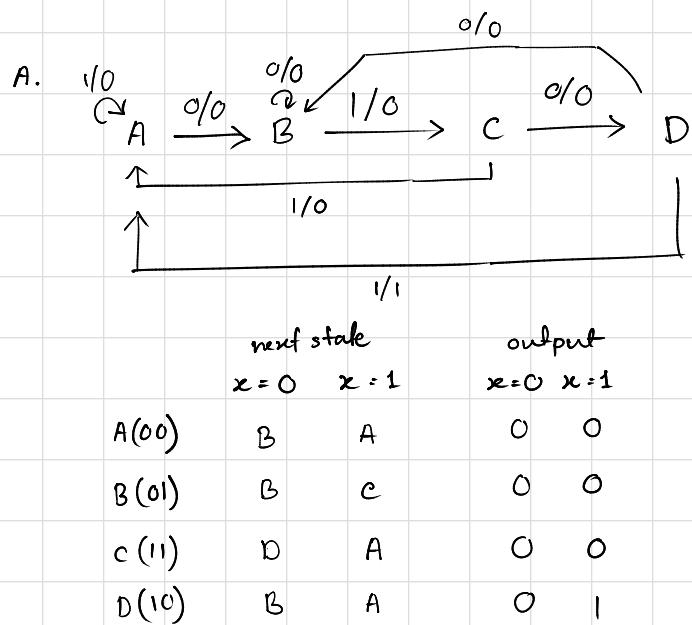


1.	You have designed a new flip-flop named X and the excitation table for X flip-flop is as follows:															
	<table border="1"> <tr> <th>$Q(t)$</th> <th>$Q(t+1)$</th> <th>X</th> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </table>	$Q(t)$	$Q(t+1)$	X	0	0	1	0	1	0	1	0	0	1	1	1
$Q(t)$	$Q(t+1)$	X														
0	0	1														
0	1	0														
1	0	0														
1	1	1														
	You want to design a sequential circuit using the X flip-flop, which could recognize the sequence 0101. While designing the circuit, consider the overlaps of the sequence. The sequential circuit will give output "1" when the given sequence is recognized, otherwise will provide output "0".															
	For such a circuit, complete the following steps:															
A.	Draw the state diagram while assigning the following coding to the states: 1 st state is coded as 00, 2 nd state as 01, 3 rd state as 11, and 4 th state as 10 [2]															
B.	Draw the state table with output and input function for X flip-flop. [2]															
C.	Minimize the functions of output and flip-flop inputs. [2]															
D.	Based on the designed sequential circuit, provide the outputs for the following input sequences: i) 000100101011 ii) 010110100101 [2]															

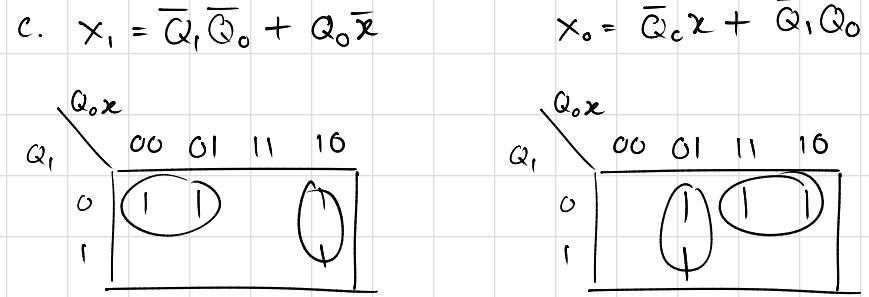
DISCLAIMER: Following the absurd rules of our great Benzin; only considering a single bit when looking for overlaps.



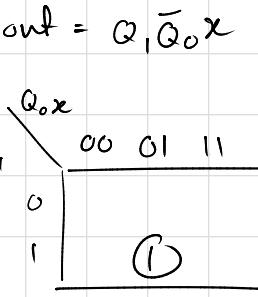
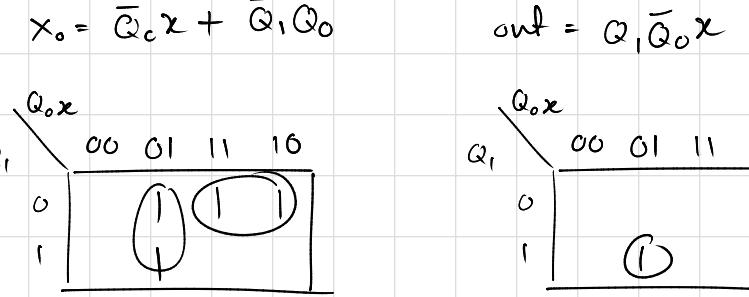
B. State table

Q_1, Q_0 in	Q_1, Q_0 out	X_1, X_0
0 0 0	0 1 0	1 0
0 0 1	0 0 0	1 1
0 1 0	0 1 0	1 1
0 1 1	1 1 0	0 1
1 0 0	0 0 0	0 0
1 0 1	0 0 1	0 1
1 1 0	1 0 0	1 0
1 1 1	0 0 0	0 0

$$x_1 = \overline{Q}_1 \overline{Q}_0 + Q_0 \overline{x}$$



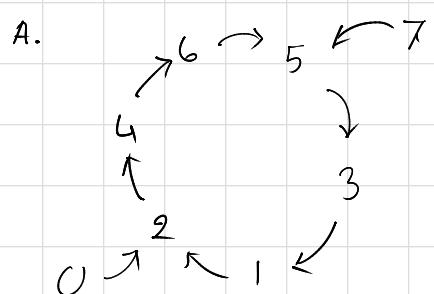
$$x_0 = \overline{Q}_1 x + \overline{Q}_1 Q_0$$



D. i) in: 0 0 0 1 0 0 0 1 0 1 1	out: 0 0 0 0 0 0 0 0 1 0 0 0
ii) in: 0 1 0 1 1 0 1 0 0 1 0 1	out: 0 0 0 1 0 0 0 0 0 0 0 1

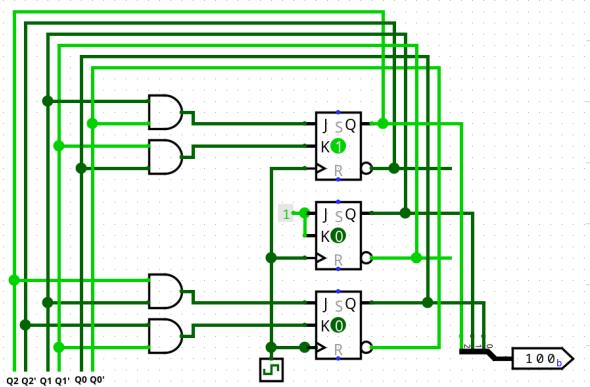
2.	Design a synchronous counter to count the following arbitrary sequence using JK flip flops and basic gates.
	Sequence : 5 → 3 → 1 → 2 → 4 → 6 → 5 → Here, the next sequence of the missing numbers 0 is 2, and for missing number 7 the next sequence is 5.
A.	Draw the state Diagram
B.	Find the state table with JK flip flop inputs
C.	Minimize the functions of flip flop inputs
D.	Draw the circuit diagram using block Diagram of JK flip flops and basic gates

$Q(t)$	$Q(t+1)$	J	K	Operation
0	0	0	x	No change/reset
0	1	1	x	Set/complement
1	0	x	1	Reset/complement
1	1	x	0	No change/set

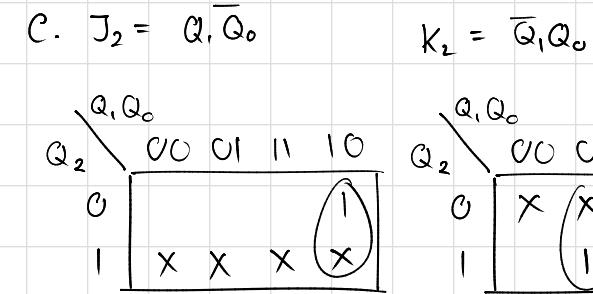


B. State table

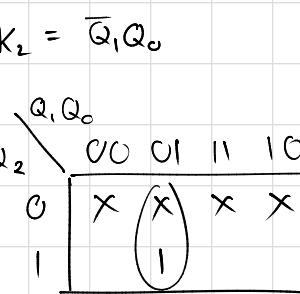
Q_2, Q_1, Q_0	Q_2, Q_1, Q_0	J_2, K_2	J_1, K_1	J_0, K_0
0 0 0	0 1 0	0 x	1 x	0 x
0 0 1	0 1 0	0 x	1 x	x 1
0 1 0	1 0 0	1 x	x 1	0 x
0 1 1	0 0 1	0 x	x 1	x 0
1 0 0	1 1 0	x 0	1 x	0 x
1 0 1	0 1 1	x 1	1 x	x 0
1 1 0	1 0 1	x 0	x 1	1 x
1 1 1	1 0 1	x 0	x 1	x 0



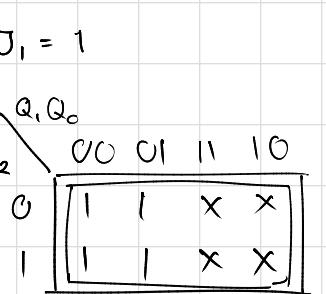
$$J_2 = Q_1 \overline{Q}_0$$



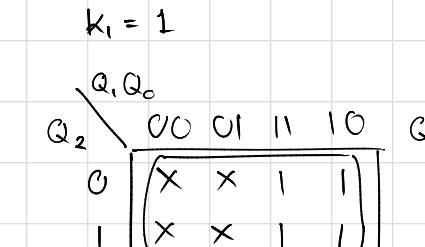
$$K_2 = \overline{Q}_1 Q_0$$



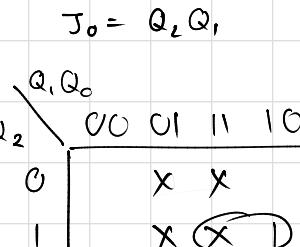
$$J_1 = 1$$



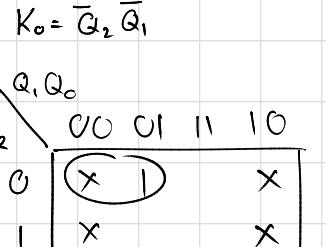
$$K_1 = 1$$



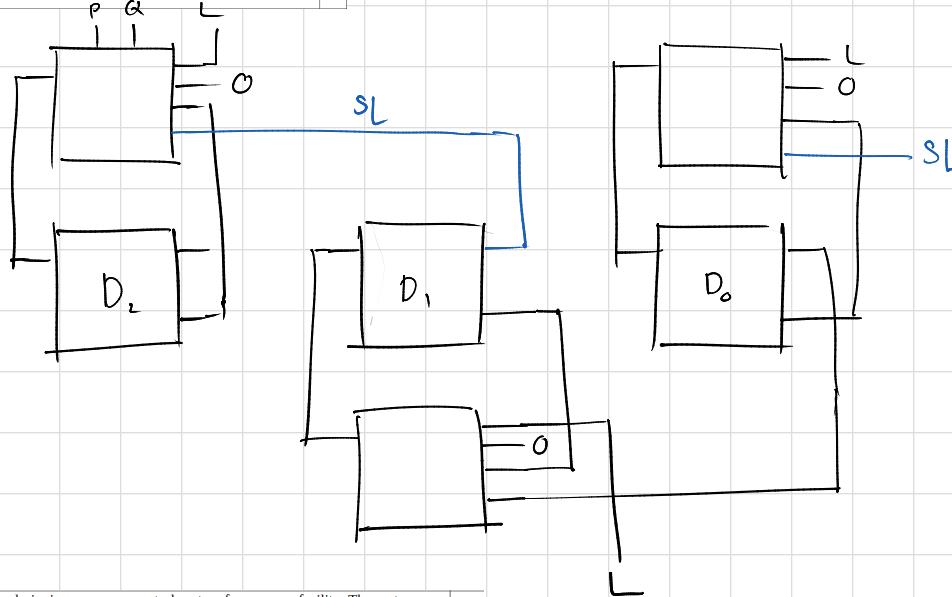
$$J_0 = Q_2 Q_1$$



$$K_0 = \overline{Q}_2 \overline{Q}_1$$



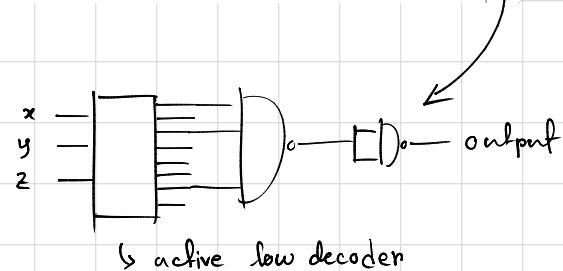
P	Q	Operation
0	0	Parallel Load
0	1	Clear Register to 0
1	0	Toggle
1	1	Shift Left



5 Suppose you are designing an access control system for a secure facility. The system uses keycards with embedded codes. There are 3 security guards in the facility and each have a keycard that is encoded with a 3 bit unique identifier. The keycards are shown below: [4]

010 110 000

The access control system is only enabled from 8:00 am to 3:00 pm. That means even if the guards try to enter the facility at night, the control system won't be able to give access. Design a combinational circuit that has an active low enable or inverted clock that recognizes the unique identifier on the keycard and grants access only to authorized individuals using decoder and NAND gate. Show the truth table and logic circuit decoder block diagram.



$$B. Y_3 = I_{15} + I_{14} + I_{13} + I_{12} + I_{11} + I_{10} + I_9 + I$$

$$Y_2 = I_{15} + I_{14} + I_{13} + I_{12} + I_7 + I_6 +$$

$$Y_1 = I_{15} + I_{14} + I_{11} + I_{10} + I_7 + I_6 + I_3 + I_2$$

$$Y_0 = I_{15} + I_{13} + I_{11} + I_9 + I_7 + I_5 + I_3$$

1

You thought a specialized alarm system could be quite helpful. You'll put in the first letter of the tasks you may have (given below) as the input and a priority encoder will give a 3-bit output. Here is the list of tasks with their input-output labels and priority:

Priority	Task	Input-Output
1 (highest)	Project Submissions	P-010
2	Class Tests	C-011
3	Lab Tests	L-111
4	Quizzes	Q-001
5 (lowest)	Assignment Submissions	A-110

Now, design a priority encoder that would work as your personal task manager and alarm system for this hectic week. To do it, complete the following steps:

- A. Derive the truth table of the priority encoder including the valid bit.
 - B. Derive the Boolean expressions for all the outputs.
 - C. Draw the logic diagram using basic gates.

$$y_L = \bar{P} \bar{C} \bar{Q} + \bar{P} \bar{C} L$$

$$Y_t = \bar{L}\bar{Q} + L + C + P$$

$$Y_0 = \bar{P}C + \bar{P}Q + \bar{P}L$$

The diagram illustrates the state transitions of a system over time. The horizontal axis represents time, and the vertical axis represents the state of the PC and Q output.

- PC:** The PC value changes from 000 to 001, then to 011, 010, 110, 111, 101, and finally 100.
- Q Output:** The Q output shows a sequence of bits: X, 1, 1, 1, 1, 1, 1, 1. A blue box highlights the first four bits (X, 1, 1, 1). A green box highlights the next two bits (1, 1). A red box highlights the last three bits (1, 1, 1).

The logic diagram illustrates a 4-bit adder circuit. It features four input columns labeled P, C, L, and Q, each with four inputs. The outputs of the first three columns (P, C, L) are connected to the inputs of four AND gates. The outputs of these four AND gates serve as inputs to four OR gates. The outputs of the OR gates from the first three columns are summed at a fifth OR gate. The output of this fifth OR gate is combined with the fourth input from column Q through another AND gate. The final output is produced by a fifth OR gate. The labels **C.**, **0 1 1**, and **V** are present near the circuit.

C.	I ₁₅	*	*	*	*
	I ₁₄	*	*	*	*
	I ₁₃	*	*	*	*
	I ₁₂	*	*	*	
	I ₁₁	*	*	*	*
	I ₁₀	*	*	*	
	I ₉	*	*	*	*
	I ₈	*			
	I ₇		*	*	*
	I ₆		*	*	
	I ₅		*		*
	I ₄		*		
	I ₃			*	*
	I ₂			*	
	I ₁				*
	I ₀				

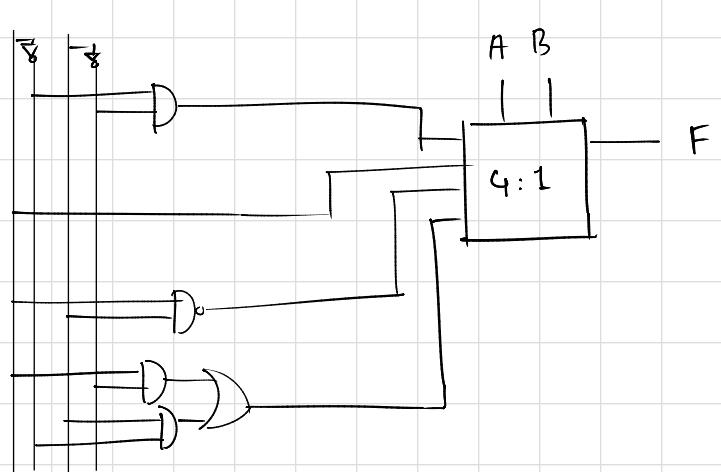
7 Implement the following Boolean function using a 4:1 MUX and necessary basic gates.

[4]

$$F(A, B, C, D) = \Pi_M(1, 2, 3, 4, 5, 11, 12, 15)$$

A	B	C	D	F
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
<hr/>				0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	1
<hr/>				1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
<hr/>				0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

C D



$$F = C\bar{D} + CD$$

$$= C$$

$$F = \bar{C}D$$

$$F = \bar{C}D + C\bar{D}$$

