USN | | | | | | | | | | |

Internal Assessment Test 2 – January 2022

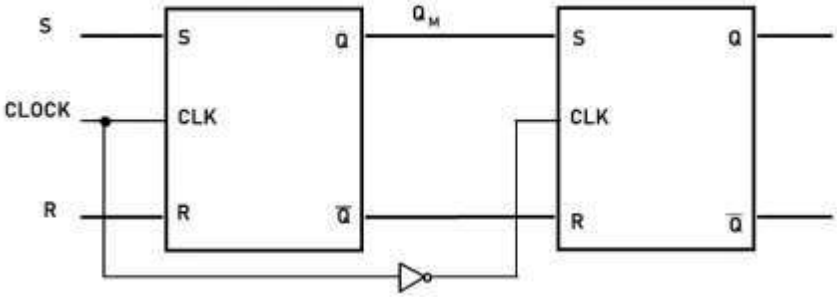
Sub:	Analog and Digital Electronics					Sub Code:	18CS33	Branch:	CSE	
Date:	25/1/2022	Duration:	90 min's	Max Marks:	50	Sem/Sec:	3rd / A,B,C		OBE	

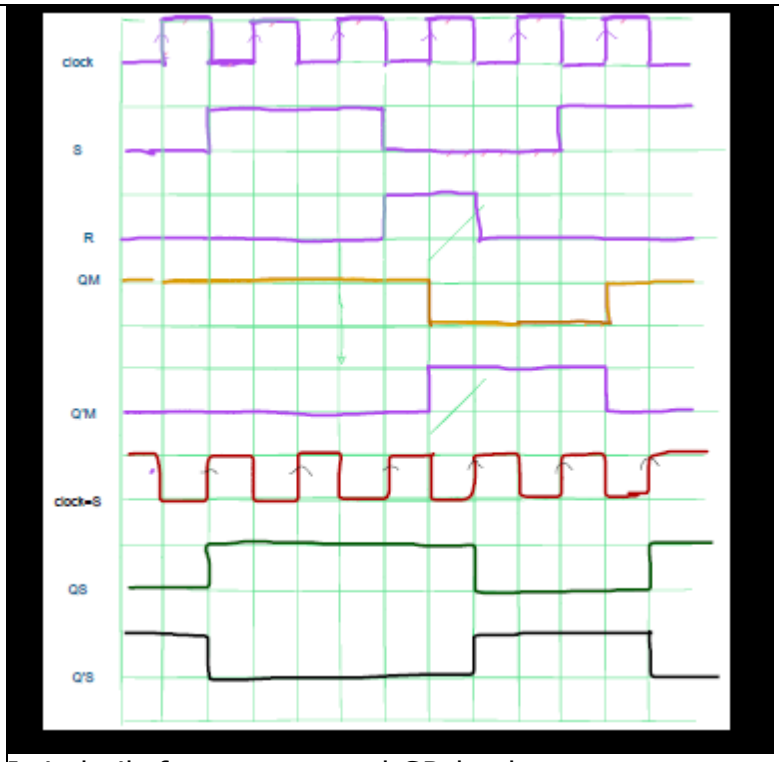
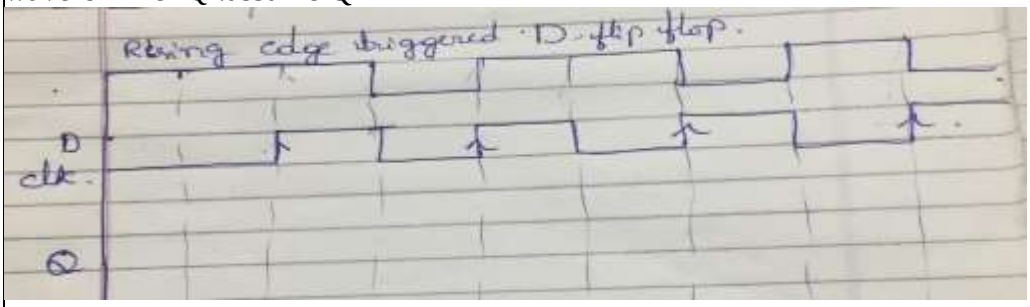
Answer any FIVE FULL Questions

		MARKS	CO	RBT
1 (a)	Explain the operation of master and slave JK flip flop with truth table and timing diagram	[7]	CO4	L4
	ANS:			
(b)	Master Slave SR Flip flop			

S	R	Q	Q_{next}	Q_{next}'
0	0	0	0	1
0	0	1	1	0
0	1	0	0	1
0	1	1	0	1
1	0	0	1	0
1	0	1	1	0
1	1	0	×	×
1	1	1	×	×

Truth Table

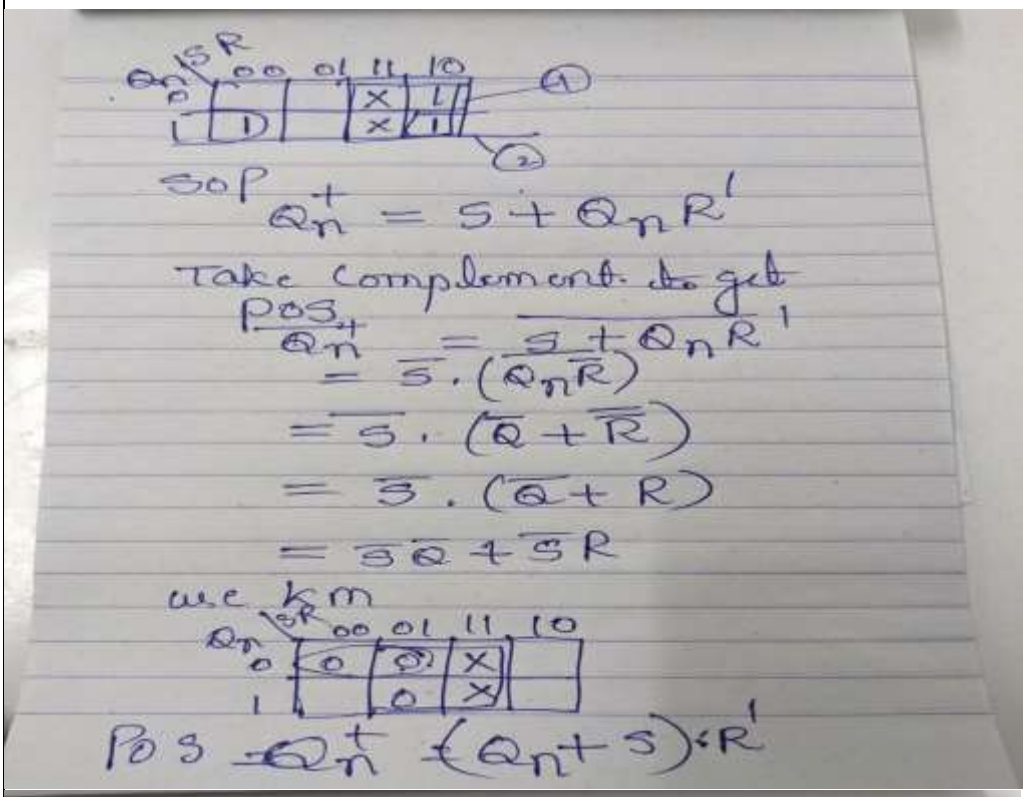


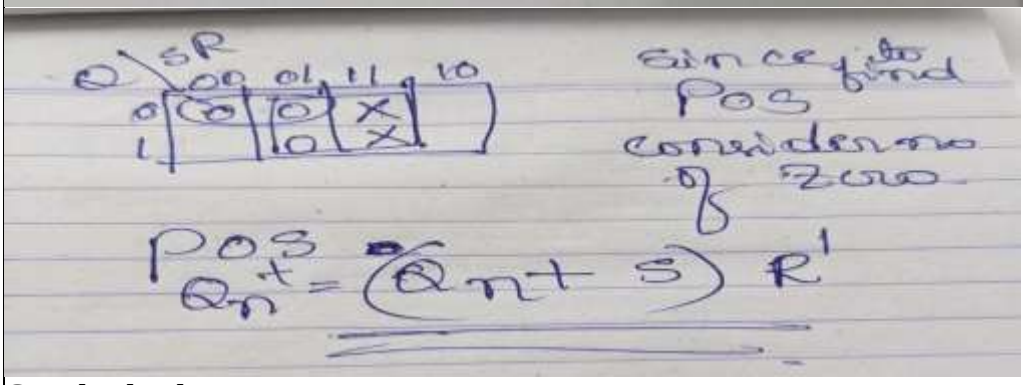
	 It is built from two gated SR latches: one a master, and the other a slave. The master takes the flip-flop's inputs: `S` (set), `R` (reset), and `C` (clock). 2. The clock input is fed to the latch's gate input. 3. The slave takes the master's outputs as inputs (`Q` to `S` and `Qn` to `R`), and the complement of the flip-flop's clock input. 4. The slave's outputs are the flip-flop's outputs. 5. This difference in clock inputs between the two latches *disconnects* them and eliminates the transparency between the flip-flop's inputs and outputs. 6. The schematic below shows a master-slave SR flip-flop. 7. The two inputs `S` and `R` are used to set and reset the data respectively. 8. The clock input `C` is used to control both the master and slave latches making sure only one of the latches can set its data at any given time. 9. When `C` has the value `1`, the master latch can set its data and the slave latch cannot. When `C` has the value `0`, the slave can set its data and the master cannot			
	Given a rising-edge-triggered D flip-flop with the following inputs, sketch the waveform for *Q* .assume Q=1 	[3]	CO4	L3
2	Derive the characteristic equations for the following latches and flip-flops in product of-sums form.	[10]	CO4	L3

(a) S-R latch or flip-flop
(b) Gated D latch
(c) D flip-flop
(d) J-K flip-flop
(e) T flip-flop

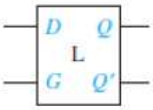
ANS:

INPUTS			OUTPUTS	REMARKS
S	**R**	**Q_n (Present State)**	**Q_{n+1} (Next State)**	**States and Conditions**
0	0	X	Q_n	Hold State condition S = R = 0
0	1	X	0	Reset state condition S = 0 , R = 1
1	0	X	1	Set state condition S = 1 , R = 0
1	1	X	Indeterminate	Indeterminate state condition S = R = 1

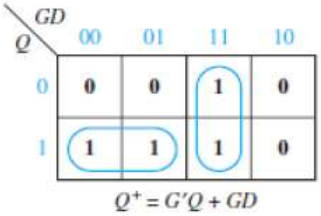


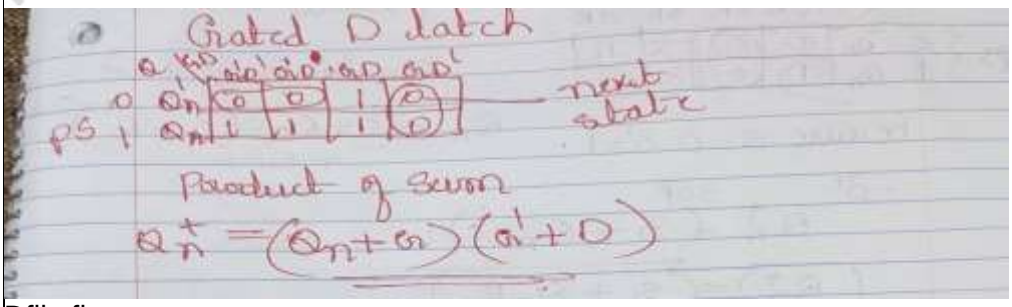


Gated D latch

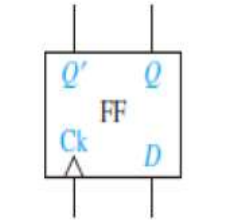


G D Q	Q^+
0 0 0	0
0 0 1	1
0 1 0	0
0 1 1	1
1 0 0	0
1 0 1	0
1 1 0	1
1 1 1	1

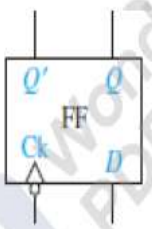




Dflip flop:



(a) Rising-edge trigger

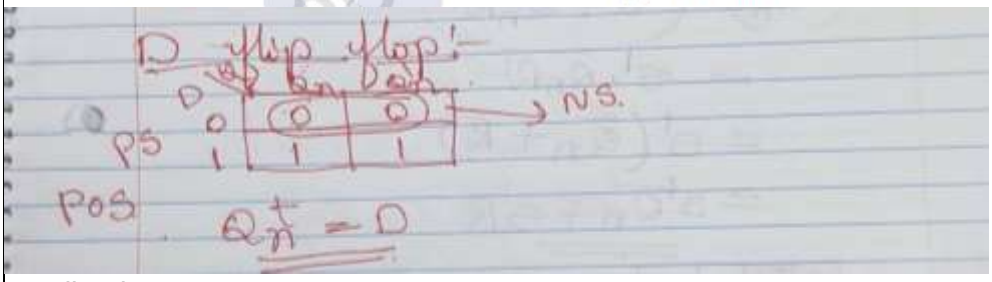


(b) Falling-edge trigger

D Q	Q^+
0 0	0
0 1	0
1 0	1
1 1	1

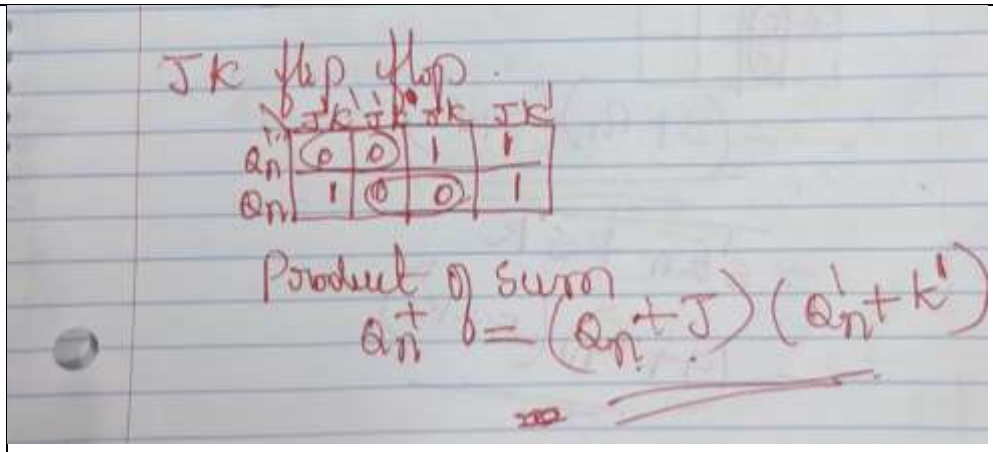
$Q^+ = D$

(c) Truth table

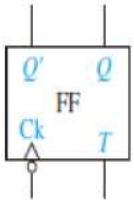


JK Flip Flop:

INPUTS J	INPUTS K	Q_n (Present State)	OUTPUTS Q_{n+1} (Next State)	REMARKS States and Conditions
0	0	X	Q_n	Hold State condition J = K = 0
0	1	X	0	Reset state condition J = 0 , K = 1
1	0	X	1	Set state condition J = 1 , K = 0
1	1	X	Q'_n	Toggle state condition J = K = 1



T Flip Flop :

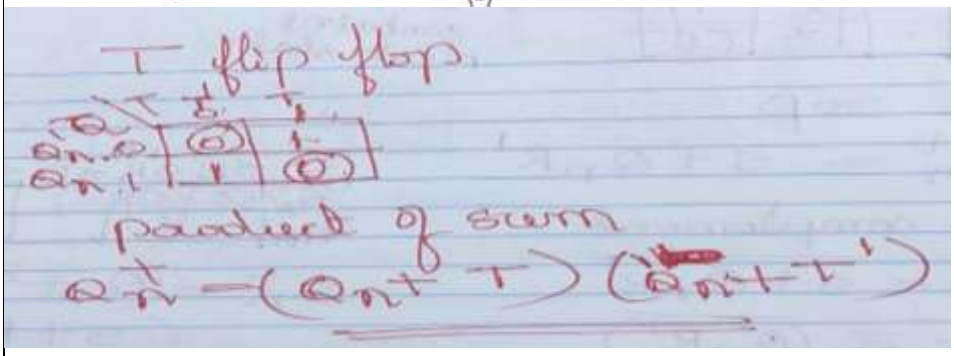


(a)

T Q	Q^+
0 0	0
0 1	1
1 0	1
1 1	0

(b)

$$Q^+ = T'Q + TQ' = T \oplus Q$$



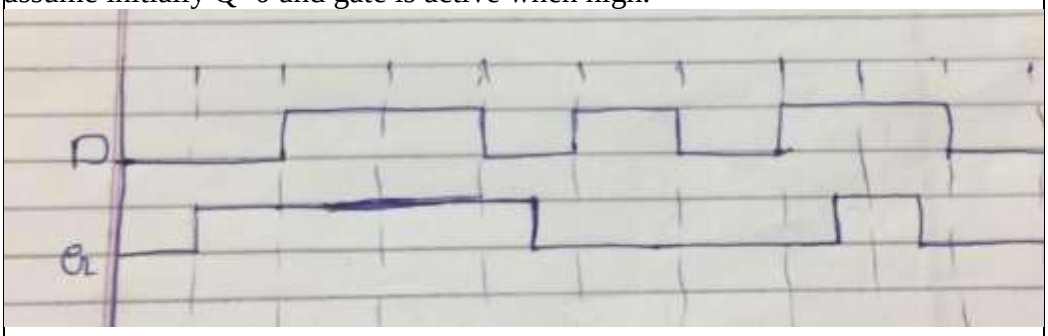
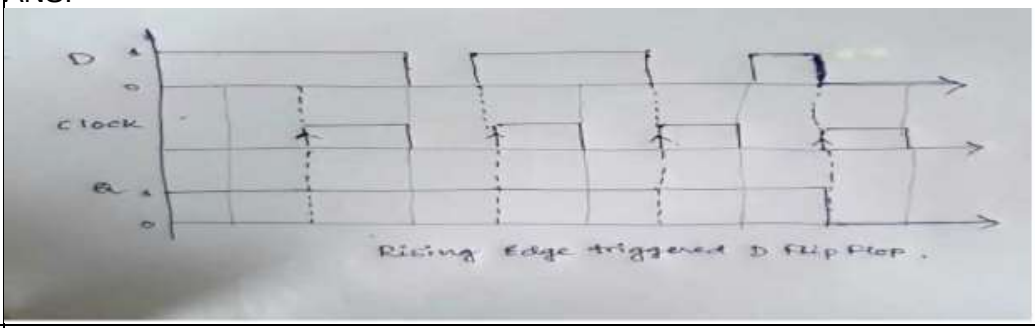
3	Write a structural VHDL module for a 4 bit adder using Full adder component.
	Full Adder
	Library ieee;
	use ieee.std_logic_1164.all;
	
	entity fa is port(a,b,c:in bit; sum,carry:out bit);
	end fa;
	
	architecture data of fa is
	begin
	sum<= a xor b xor c;
	carry <= ((a and b) or (b and c) or (a and c));
	end data;
	
	4bit adder
	
	library IEEE;
	use IEEE.STD_LOGIC_1164.all;
	
	entity 4bitadder is
	port(a : in STD_LOGIC_VECTOR(3 downto 0);
	b : in STD_LOGIC_VECTOR(3 downto 0);
		[10]	CO5	L3

```
ca : out STD_LOGIC;
sum : out STD_LOGIC_VECTOR(3 downto 0)
);
end 4bitadder;

architecture adder4 of 4bitadder is
Component fa is
port (a : in STD_LOGIC;
b : in STD_LOGIC;
c : in STD_LOGIC;
sum : out STD_LOGIC;
ca : out STD_LOGIC
);
end component;

signal s : std_logic_vector (2 downto 0);
signal temp: std_logic;

begin
temp<='0';
u0 : fa port map (a(0),b(0),temp, sum(0),s(0));
u1 : fa port map (a(1),b(1),s(0),sum(1),s(1));
u2 : fa port map (a(2),b(2),s(1),sum(2),s(2));
ue : fa port map (a(3),b(3),s(2),sum(3),ca);
end adder4;
```

4 (a)	Explain edge-triggered D Flip Flop with truth table and timing diagram and why edge the trigger is important	[6]	CO4	L4
(b)	Given a gated D latch with the following inputs, sketch the waveform for Q. assume initially Q=0 and gate is active when high.  ANS:  	[4]	CO4	L3
5	Design 3 bit Synchronous Down Counter using SR flip flop ANS:	[10]	CO4	L3

Design 3bit counter (down) synchronous counter using SR flip flop.

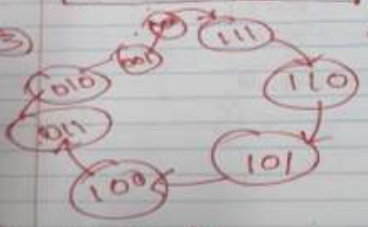
① Truth table.
SR flip flop

S	R	Q_n	Q_{n+1}
0	0	Q_n	Q_n
0	1	0	0
1	0	1	1
1	1	X	X

② Excitation table

PS	NS	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

③ State Diagram

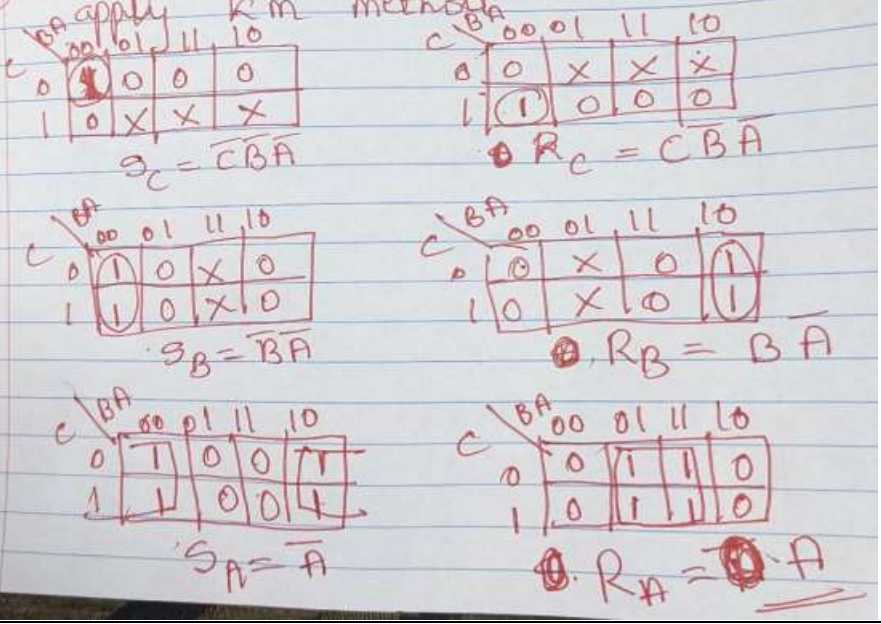


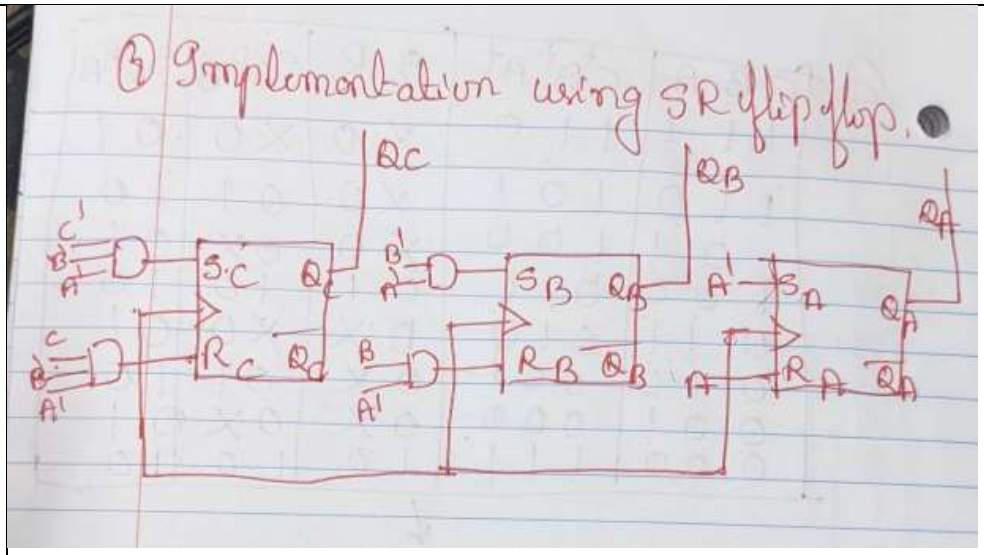
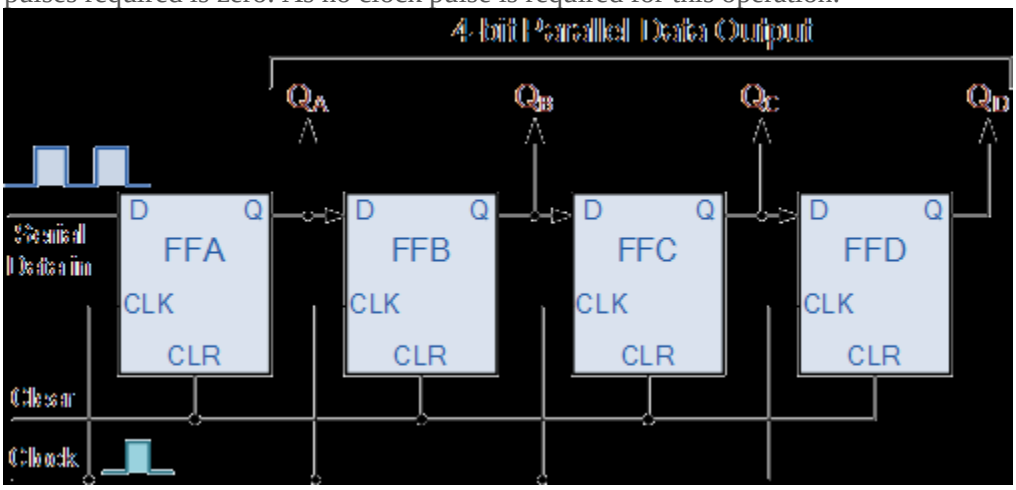
④ Synthesis table

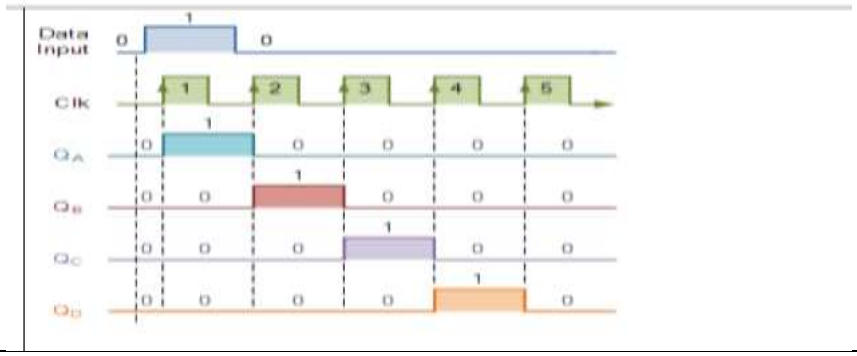
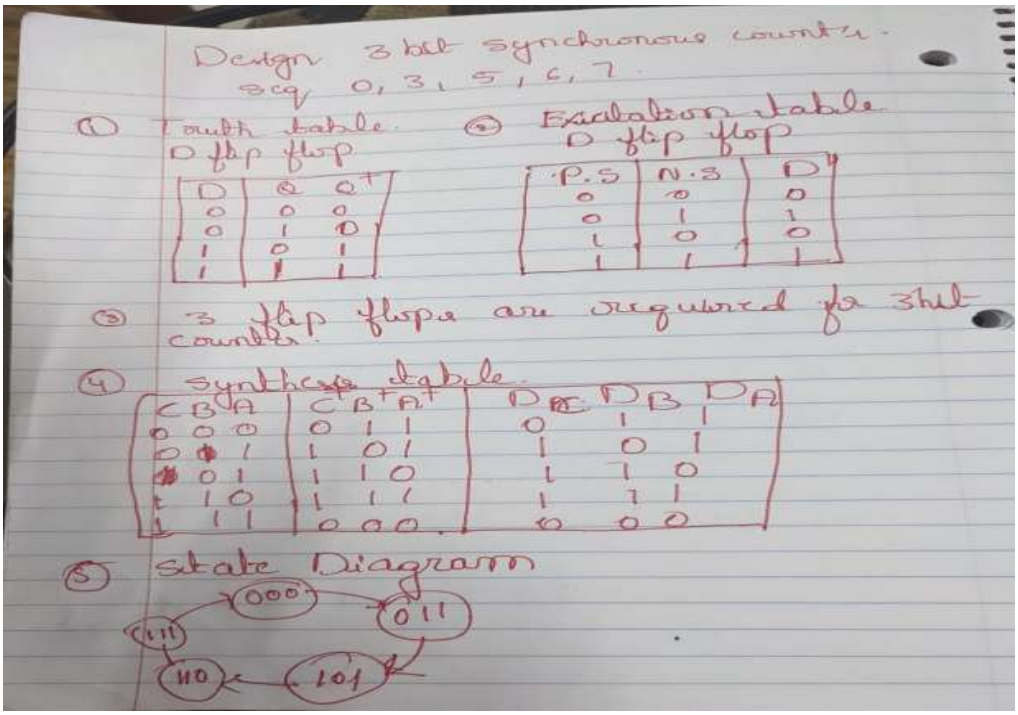
⑤

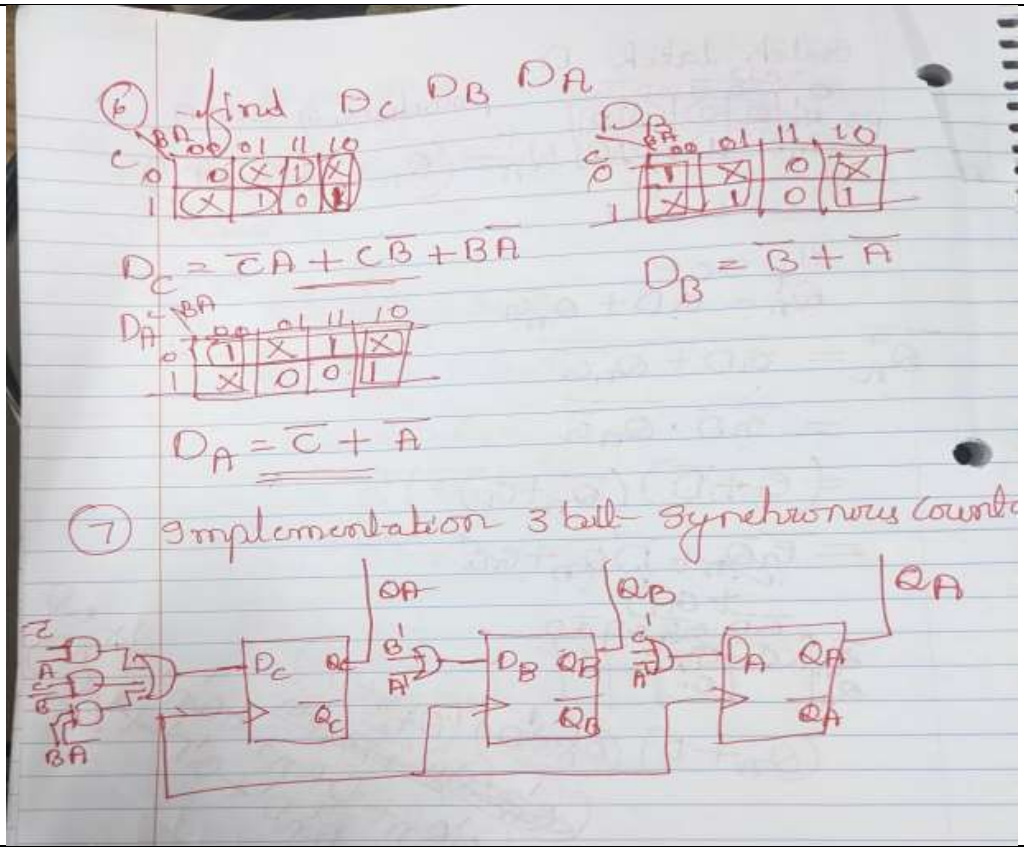
C	B	A	C^+	B^+	A^+	S_C	R_C	S_B	R_B	S_A	R_A
1	1	1	1	1	0	X	0	X	0	0	1
1	1	0	1	0	1	X	0	0	1	1	0
1	0	1	1	0	0	X	0	0	X	0	1
1	0	0	0	1	1	0	1	1	0	1	0
0	1	1	0	1	0	0	X	X	0	0	1
0	1	0	0	0	1	0	X	0	1	1	0
0	0	1	0	0	0	0	X	0	X	0	1
0	0	0	1	1	1	1	0	1	0	1	0

⑥ To find $S_C R_C S_B R_B S_A R_A$ apply km method



				
6	Write neat Sketch and Explain the working principle of Serial in parallel out(SIPO with timing Diagram ANS: **Serial-in to Parallel-out (SIPO)** - the register is loaded with serial data, one bit at a time, with the stored data being available at the output in parallel form. For 'n' bit serial input data which need to be stored, the number of clock pulses required is equal to 'n'. and For 'n' bit parallel output data that need to be stored, the number of clock pulses required is zero. As no clock pulse is required for this operation.  The operation is as follows. Let's assume that all the flip-flops (FFA to FFD) have just been RESET (CLEAR input) and that all the outputs Q_A to Q_D are at logic level "0" ie, no parallel data output. If a logic "1" is connected to the DATA input pin of FFA then on the first clock pulse the output of FFA and therefore the resulting Q_A will be set HIGH to logic "1" with all the other outputs still remaining LOW at logic "0". Assume now that the DATA input pin of FFA has returned LOW again to logic "0" giving us one data pulse or 0-1-0. The second clock pulse will change the output of FFA to logic "0" and the output of FFB and Q_B HIGH to logic "1" as its input D has the logic "1" level on it from Q_A. The logic "1" has now moved or been "shifted" one place along the register to the right as it is now at Q_A. When the third clock pulse arrives this logic "1" value moves to the output of FFC (Q_C) and so on until the arrival of the fifth clock pulse which sets all the outputs Q_A to Q_D back again to logic level "0" because the input to FFA has remained constant at logic level "0". The effect of each clock pulse is to shift the data contents of each stage one place to the right, and this is shown in the following table until the complete data value of 0-0-0-1	[10]	CO4	L4,

	is stored in the register. This data value can now be read directly from the outputs of Q_A to Q_D. Then the data has been converted from a serial data input signal to a parallel data output. 	Clock Pulse No	QA	QB	QC	QD
0	0	0	0	0
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1
5	0	0	0	0 | 			
7	Design 3 bit Synchronous Counter for the following sequence 0,3,5,6,7 using D flip flop ANS: 	[10]	CO4	L3



Course Outcomes		Modules covered	PO1	PO2	PO3	PO4	PO5	PO6		PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	Design and analyze application of analog circuits using photo devices, timer IC, power supplyand regulator IC and op-amp.	1	2	3	3	2	2	0	0	0	0	0	0	0	0	2	0	2	2
CO2	Explain the basic principles of A/D and D/A conversion circuits and develop the same	1	2	2	3	2	2	0	0	0	0	0	0	0	0	2	0	2	2
CO3	Simplify digital circuits using Karnaugh Map , and Quine-McClusky Methods	2	2	3	3	3	2	0	0	0	0	0	0	0	0	2	0	2	2
CO4	Explain Gates and flip flops and make us in designing different data processing circuits, registers and counters and compare the types.	3, 4, 5	2	3	3	3	3	0	0	0	0	0	0	0	0	2	0	2	2
CO5	Develop simple HDL programs	3, 4, 5	1	1	1	1	3	0	0	0	0	0	0	0	0	2	0	2	1

COGNITIVE LEVEL	REVISED BLOOMS TAXONOMY KEYWORDS
L1	List, define, tell, describe, identify, show, label, collect, examine, tabulate, quote, name, who, when, where, etc.
L2	summarize, describe, interpret, contrast, predict, associate, distinguish, estimate, differentiate, discuss, extend
L3	Apply, demonstrate, calculate, complete, illustrate, show, solve, examine, modify, relate, change, classify, experiment, discover.
L4	Analyze, separate, order, explain, connect, classify, arrange, divide, compare, select, explain, infer.
L5	Assess, decide, rank, grade, test, measure, recommend, convince, select, judge, explain, discriminate, support, conclude, compare, summarize.

PROGRAM OUTCOMES (PO), PROGRAM SPECIFIC OUTCOMES (PSO)				CORRELATION LEVELS	
PO1	Engineering knowledge	PO7	Environment and sustainability	0	No Correlation
PO2	Problem analysis	PO8	Ethics	1	Slight/Low
PO3	Design/development of solutions	PO9	Individual and team work	2	Moderate/ Medium
PO4	Conduct investigations of complex problems	PO10	Communication	3	Substantial/ High
PO5	Modern tool usage	PO11	Project management and finance		
PO6	The Engineer and society	PO12	Life-long learning		
PSO1	Develop applications using different stacks of web and programming technologies				
PSO2	Design and develop secure, parallel, distributed, networked, and digital systems				
PSO3	Apply software engineering methods to design, develop, test and manage software systems.				
PSO4	Develop intelligent applications for business and industry				