CBCS SCHEME

USN □□□□□□□□□□

18CS33

Third Semester B.E. Degree Examination, Feb./Mar. 2022
Analog and Digital Electronics

Time: 3 hrs. Max. Marks: 100

Note: Answer any FIVE full questions, choosing ONE full question from each module.

Module-1

1 a. What is biasing? Mention different BJT biasing techniques. Explain voltage divider bias. (08 Marks)

b. Explain relaxation oscillator. (06 Marks)

c. Write a note on opto coupler. (06 Marks)

OR

2 a. Explain active filters. List advantages of active filters over passive filters. (06 Marks)

b. Explain with diagram, R-2R ladder type D to A converter. (08 Marks)

c. Define op-amp. Explain the performance parameters of op-amp. (06 Marks)

Module-2

3 a. Explain Don't Care condition with an example. (04 Marks)

b. Reduce the following functions using K-map technique:
F(P, Q, R, S) = Σm(0, 1, 4, 8, 9, 10) + d(2, 11) (08 Marks)

c. Using Quine McClusky method, simplify the expression:
F(P, Q, R, S) = Σm(0, 3, 5, 6, 7, 11, 14)
Write the gate diagram for the same. (08 Marks)

OR

4 a. Explain entered variable map method. (05 Marks)

b. Apply Quine McClusky method to find the essential prime implicants for the Boolean expression f(a, b, c, d) = Σm(1, 3, 6, 7, 9, 10, 12, 13, 14, 15) (07 Marks)

c. For the below expression, draw the logic diagram using AOI logic for minimal sum. Obtain minimal sum using K-map.
F(a, b, c, d) = Σm(1, 2, 3, 5, 6, 7, 11, 12, 13, 14, 15) (08 Marks)

Module-3

5 a. What is hazard? List the types of hazards. Explain static 0 and static 1 hazard. (06 Marks)

b. Differentiate between combinational and sequential circuit. (06 Marks)

c. Implement the following using PLA:
A(x, y, z) = Σm (1, 2, 4, 6)
B(x, y, z) = Σm (0, 1, 6, 7)
C(x, y, z) = Σm (2, 6) (08 Marks)

OR

6 a. Implement the following function using 8:1 multiplexer:
f(a, b, c, d) = Σm (0, 1, 5, 6, 8, 10, 12, 15) (07 Marks)

b. What is programmable logic array? How does PLA differ from PAL? (06 Marks)

c. Realize the following using 3:8 decoder:
(i) f(a, b, c) = Σm (1, 2, 3, 4) (ii) f(a, b, c) = Σm (3, 5, 7) (07 Marks)

compulsorily draw diagonal cross lines on the remaining blank pages.
2. Any revealing of identification, appeal to evaluator and /or equations written eg, 42+8 = 50, will be treated as malpractice.

Module-4

7 a. What are the three different models for writing a module body in VHDL? Give example for any one model. (06 Marks)

b. Derive characteristic equation for JK, T, D and SR flip flop. (08 Marks)

c. Give VHDL code for 4:1 multiplexer using conditional assign statement. (06 Marks)

OR

8 a. Using structural model, write VHDL code for Half Adder. (06 Marks)

b. Derive the excitation table for JK and SR flip flop. How SR flip flop is converted to T flip flop? (08 Marks)

c. With logic diagram, explain JK flip flop. (06 Marks)

Module-5

9 a. Define counter. Design synchronous counter for the sequence 0, 4, 1, 2, 6, 0, 4 using JK flip-flop. (08 Marks)

b. What is shift register? With a neat diagram, explain 4 bit parallel in serial out shift register. (08 Marks)

c. Write a note on sequential parity checker. (04 Marks)

OR

10 a. With a neat diagram, explain ring counter. (06 Marks)

b. Design and implement MOD 5 synchronous counter using JK flip-flop. Explain with timing diagram. (08 Marks)

c. Write a note on parallel adder with accumulator. (06 Marks)

* * * * *

USN	□□□□□□□□□□	CMRIT CMR INSTITUTE OF TECHNOLOGY, BENGALURU. ACCREDITED WITH A+ GRADE BY NAAC

VTU Solution– March 2022

Sub:	Analog and Digital Electronics				Sub Code:	18CS33	Branch:	ISE
Date:		Duration:	3 Hrs	Max Marks: 100	Sem/Sec:	III / A, B and C		OBE

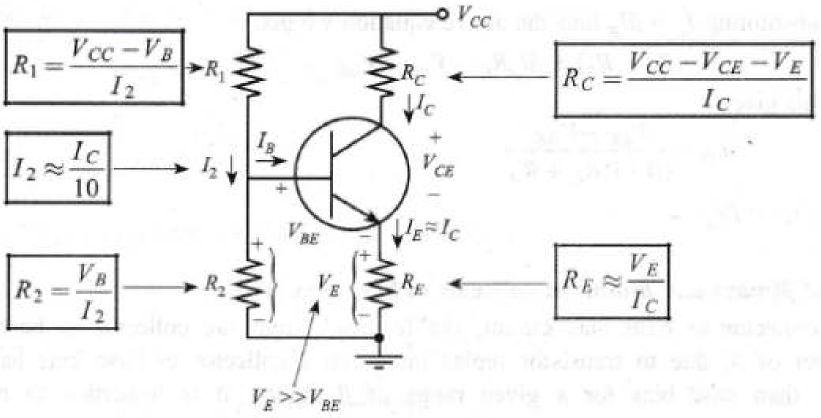
	Answer any FIVE FULL Questions	MARKS	CO	RBT
1	a. What is biasing? Mention different BJT biasing techniques. Explain voltage divider bias.	8	CO1	L2

Solution:

Transistor Biasing is the establishment of suitable dc-values such as IC, VCE, IB, etc., by using a dc-source. When BJT is properly biased, amplification of signal takes place. There are mainly three types of biasing a transistor: Base bias or Fixed bias, Collector-to-Base bias, Voltage-divider bias.

VOLTAGE DIVIDER (EMITTER CURRENT) BIAS CIRCUIT:

Voltage divider bias is the most stable of the three basic transistor biasing circuits. A voltage divider circuit is shown in the following Figure.


V_{CC}
$R_1 = \frac{V_{CC} - V_B}{I_2}$
R_1
R_C
I_C
$R_C = \frac{V_{CC} - V_{CE} - V_E}{I_C}$
$I_2 \approx \frac{I_C}{10}$
I_2
I_B
V_{CE}
V_{BE}
$I_E \approx I_C$
$R_2 = \frac{V_B}{I_2}$
R_2
V_E
R_E
$R_E \approx \frac{V_E}{I_C}$
$V_E >> V_{BE}$

There is an emitter resistor R_E connected in series with Emitter terminal, so that the total dc load in series with the transistor is ($R_C + R_E$). Resistors R_1 and R_2 constitutes a voltage V_B.

Applying KVL to the loop V_{CC}, R_1, and R_2, we get;

$V_{CC} - I_1R_1 - I_2R_2 = 0$ Or, $I_1R_1 + I_2R_2 = V_{CC}$ -------------------- (5)

We have; $I_1 = I_2 + I_B$

Voltage divider bias circuits are normally designed to have a voltage divider current I_2 very much greater than transistor base current I_B. i.e., $I_2 >> I_B$. Hence, $I_1 \approx I_2$ -------------------- (6)

Using 6 in 5; $I_2R_1 + I_2R_2 = V_{CC}$ i.e., $I_2 (R_1 + R_2) = V_{CC}$ Or, $\boldsymbol{I_2 = (V_{CC}) / (R_1 + R_2)}$

V_B is the voltage across R_2. i.e., $V_B = I_2R_2$ Or, $\boldsymbol{V_B = (V_{CC} * R_2) / (R_1 + R_2)}$

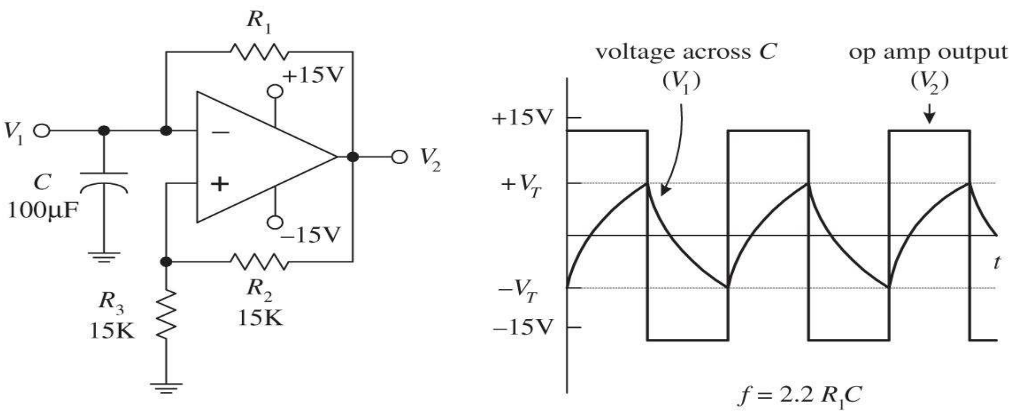
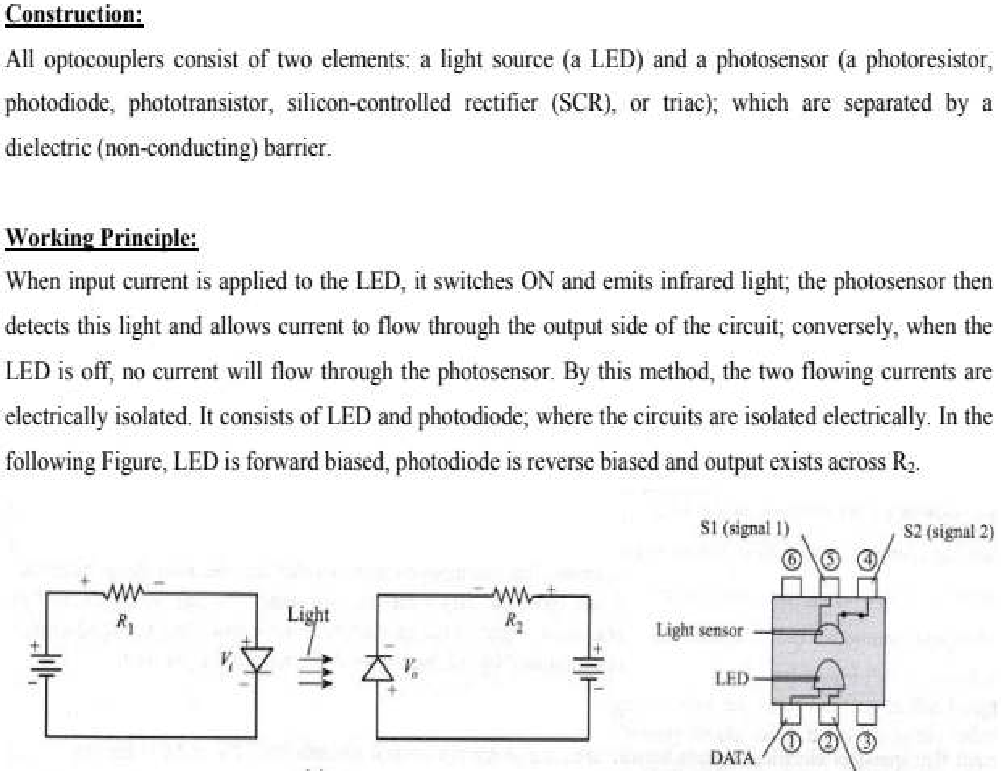
V_E is the voltage across R_E. i.e., $\boldsymbol{V_E = I_ER_E}$

Applying KVL to the base-emitter loop; $V_B - V_{BE} - V_E = 0$ i.e., $\mathbf{V_{BE} = V_B - V_E}$

Or, $V_E = V_B - V_{BE}$ i.e., $I_ER_E = V_B - V_{BE}$ Hence, $\boldsymbol{I_E = (V_B - V_{BE}) / R_E}$

Applying KVL to the collector-emitter loop; $V_{CC} - I_CR_C - V_{CE} - I_CR_E = 0$ $[I_E \approx I_C]$

i.e., $\boldsymbol{V_{CE} = V_{CC} - I_C (R_C + R_E)}$

	b. Explain relaxation oscillator **Solution:** Simple Square-Wave Relaxation Oscillator  R_1 +15V V_1 C 100μF −15V R_3 15K R_2 15K V_2 voltage across C (V_1) op amp output (V_2) +15V $+V_T$ t $-V_T$ −15V $f = 2.2\ R_1C$ Oscillator generally refers to the circuit which produces periodic and repetitive output like a sine wave or square wave. An oscillator can be a mechanical or electronic construction that produces oscillation depending on a few variables.	6	CO1	L3
	c. Write a note on opto coupler  **Construction:** All optocouplers consist of two elements: a light source (a LED) and a photosensor (a photoresistor, photodiode, phototransistor, silicon-controlled rectifier (SCR), or triac); which are separated by a dielectric (non-conducting) barrier. **Working Principle:** When input current is applied to the LED, it switches ON and emits infrared light; the photosensor then detects this light and allows current to flow through the output side of the circuit; conversely, when the LED is off, no current will flow through the photosensor. By this method, the two flowing currents are electrically isolated. It consists of LED and photodiode; where the circuits are isolated electrically. In the following Figure, LED is forward biased, photodiode is reverse biased and output exists across R_2. R_1 Light V_i R_2 V_o S1 (signal 1) S2 (signal 2) ⑥ ⑤ ④ Light sensor LED DATA ① ② ③ The Figure describes the basic operation of an optocoupler. When current is not being applied via Pin 1, the LED is off, and the circuit connected to Pins 4 and 5 is experiencing no current flow. When power is applied to the input circuit, the LED switches on, the sensor detects the light, closes the switch and initiates current flow in the output circuit, as shown in the Figure	6		

2 a. Explain active filter. List advantages of active filters over passive filters. 06 CO1 L2

Solution:

Filter is a frequency selective circuit commonly used in signal processing that passes signal of specified range of frequencies and blocks the signals of frequencies outside the band. Active filters are attractive due to their –

- Flexibility in gain control
- Small component size
- No loading Problem
- Pass band gain
- Use of the inductors can be avoided

Filters are useful in many areas of applications, such as Communication and Signal Processing. They are found in electronic systems like Radio, Television, Telephones, Radars, satellites, and Biomedical instruments.

Active filters offer the following advantages over Passive filters:

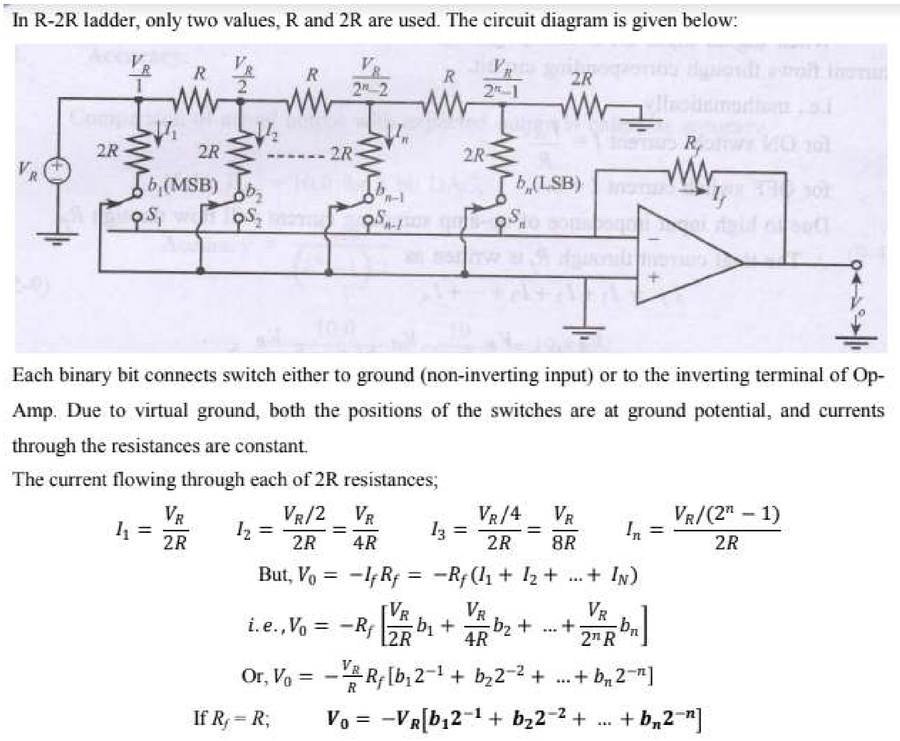
- Gain and frequency adjustment flexibility
- No loading problem & No insertion loss
- Size and weight
- Cost.

Most commonly used active filters are –

- Low-pass filter, High-pass filter, Band-pass filter, Band-stop filter (Band-reject filter), and All-pass filter.

b. Explain with diagram, R-2R ladder type D to A converter 8 CO1 L2

Solution:



In R-2R ladder, only two values, R and 2R are used. The circuit diagram is given below:

Each binary bit connects switch either to ground (non-inverting input) or to the inverting terminal of Op-Amp. Due to virtual ground, both the positions of the switches are at ground potential, and currents through the resistances are constant.

The current flowing through each of 2R resistances;

$$I_1 = \frac{V_R}{2R} \qquad I_2 = \frac{V_R/2}{2R} = \frac{V_R}{4R} \qquad I_3 = \frac{V_R/4}{2R} = \frac{V_R}{8R} \qquad I_n = \frac{V_R/(2^n-1)}{2R}$$

$$\text{But, } V_0 = -I_f R_f = -R_f(I_1 + I_2 + \ldots + I_N)$$

$$i.e., V_0 = -R_f\left[\frac{V_R}{2R}b_1 + \frac{V_R}{4R}b_2 + \ldots + \frac{V_R}{2^n R}b_n\right]$$

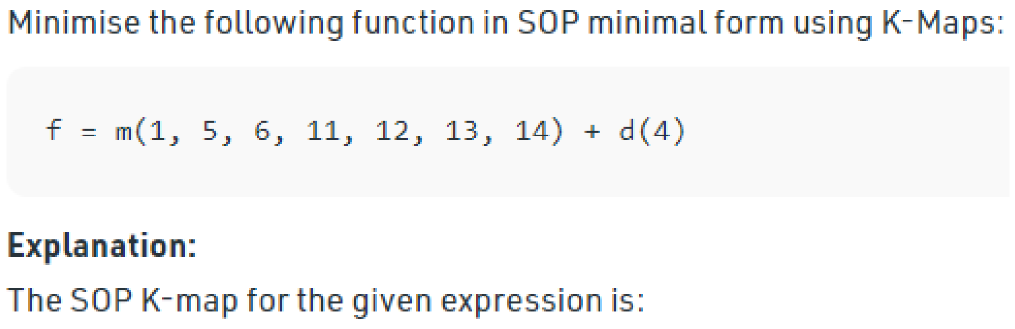
$$\text{Or, } V_0 = -\tfrac{V_R}{R}R_f[b_1 2^{-1} + b_2 2^{-2} + \ldots + b_n 2^{-n}]$$

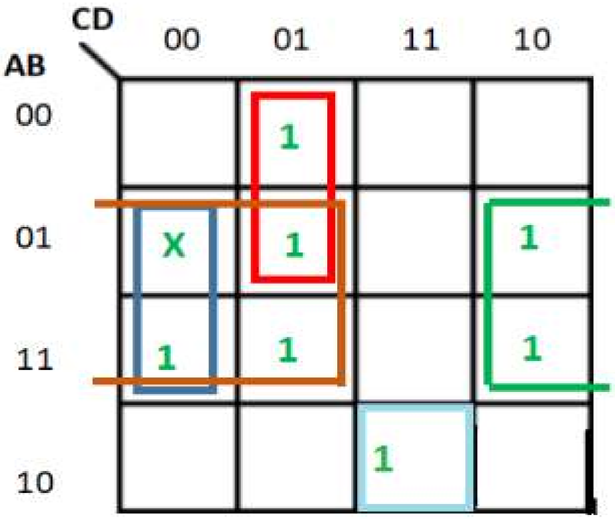
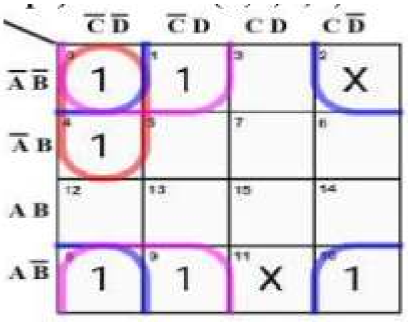
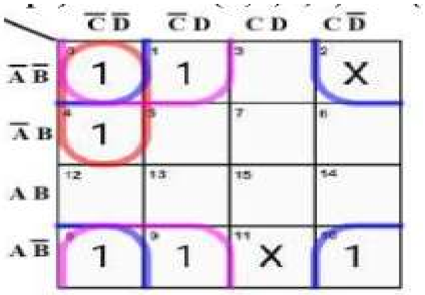
$$\text{If } R_f = R; \qquad \boldsymbol{V_0 = -V_R[b_1 2^{-1} + b_2 2^{-2} + \ldots + b_n 2^{-n}]}$$

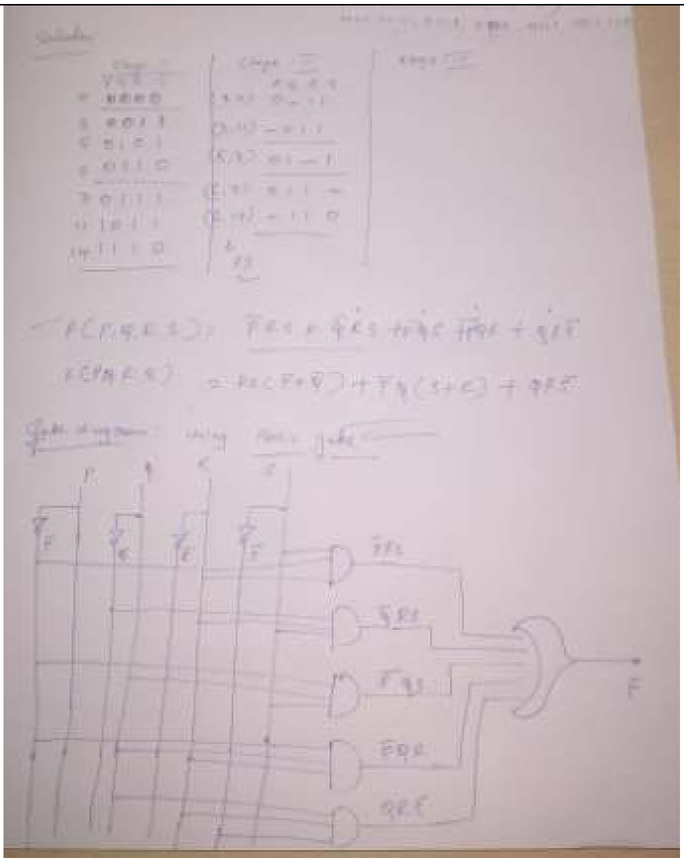
c. Define op-amp. Explain the performance parameters of op-amp. 06

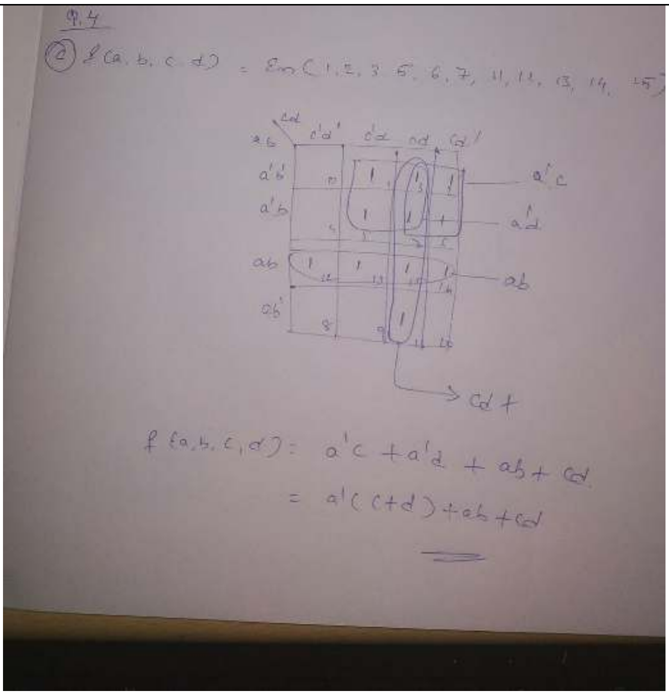
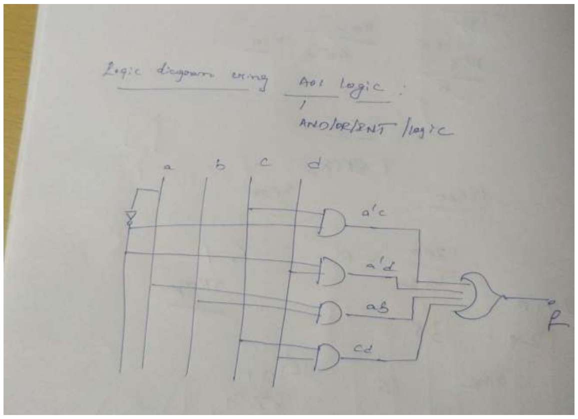
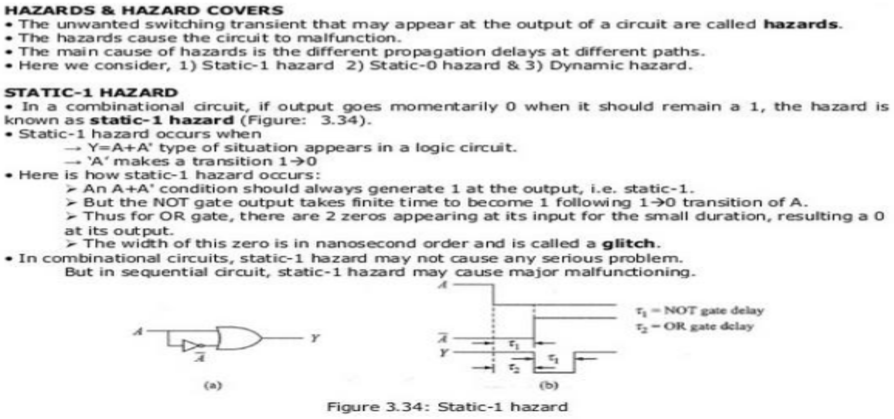
Solution:

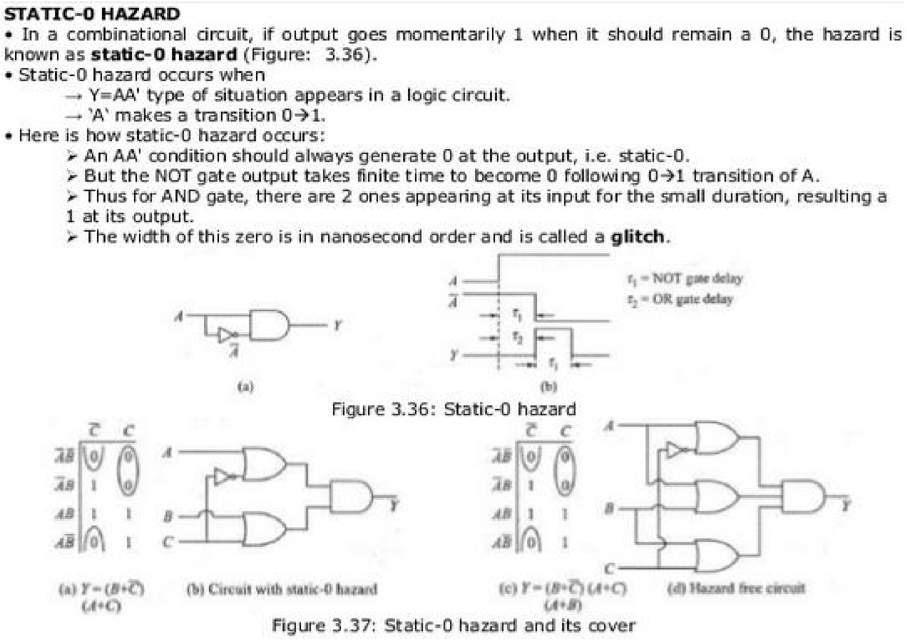
An Op-Amp is a direct-coupled high gain, high bandwidth differential amplifier with very high value of input impedance and very low value of output impedance.

	The ideal Op-Amp model was derived to simplify circuit calculations. The ideal Op-Amp model makes three assumptions: 1. Input resistance (impedance), Ri = ∞ 2. Output resistance (impedance), Ro = 0 3. Open-loop (differential voltage) gain, Ad = ∞ Based on these three assumptions, other assumptions can be derived: 1. Since Ri = ∞, Ii = Ini = 0 2. Since Ro = 0, Vo = Ad * Vd 3. Zero DC input and output offset voltages 4. Bandwidth and slew rate are also infinite, as no frequency dependencies are assumed. 5. Drift is also zero, as there is no changes in performance over time, temperature, power supply variations, and so on 6. Since output voltage depends only on differential input voltage, it rejects any voltage common to both inputs. Hence, common mode gain = 0			
3	a. Explain don’t care conditions with an example. **Solution:** The Don’t Care conditions allow us to replace the empty cell of a K-Map to form a grouping of the variables which is larger than that of forming groups without don’t care. While forming groups of cells, we can consider a Don’t Care cell as 1 or 0 or we can also ignore that cell. Therefore, the “Don’t Care” condition can help us to form a larger group of cells.  Minimise the following function in SOP minimal form using K-Maps: `f = m(1, 5, 6, 11, 12, 13, 14) + d(4)` **Explanation:** The SOP K-map for the given expression is:	4	CO3	L2

	 CD AB 00 01 11 10 00 01 11 10 1 X 1 1 1 1 1 1 1 'herefore, SOP minimal is, f = BC' + BCD' + A'C'D + AB'CD			
	b. Reduce the following function using k Map S= F (A, B, C, D) = Σm (0,1,4,8,9,10) + d (2,11) **Solution:**  $\bar{C}\bar{D}$ $\bar{C}D$ CD $C\bar{D}$ $\bar{A}\bar{B}$ $\bar{A}B$ AB $A\bar{B}$ 1 1 X 1 1 1 X 1 $S = \bar{A}\bar{C}\bar{D} + \bar{B}\bar{D} + \bar{B}\bar{C}$  $\bar{C}\bar{D}$ $\bar{C}D$ CD $C\bar{D}$ $\bar{A}\bar{B}$ $\bar{A}B$ AB $A\bar{B}$ 1 1 X 1 1 1 X 1 $S = \bar{A}\bar{C}\bar{D} + \bar{B}\bar{D} + \bar{B}\bar{C}$	8	CO3	L2
	c. Using QM method, simplify the expression f(p.q.r,s)= Σm(0,3,5,6,7,11,14) Solution:	8		

				
4	**a.** Explain entered variable map method **Solution:** **Minimization procedure for VEM –** Now, let’s see how to find SOP expression if a VEM is given. 1. Write all the variables(original and complimented forms are treated as two different variables) in the map as 0, leave 0’s, minterms and don’t cares as it is and obtain the SOP expression. 2. (a) Select one variable and make all occurrences of that variable as 1, write minterms (1’s) as don’t cares, leave 0’s and don’t cares as it is. Now, obtain the SOP expression. (b) Multiply the obtained SOP expression with the concerned variable. 3. Repeat step 2 for all the variables in the k-map. 4. SOP of VEM is obtained by ORing all the obtained SOP expressions.	5	CO3	L2
	b. Apply QM method to find the essential prime implicants for the Boolean expression f(a,b,c,d)= Σm(1,3,6,7,9,10,12,13,14,15)	7	CO1	L2

	 cd ab c'd' c'd cd cd' a'b' a'b ab ab' 1 1 1 1 1 1 1 1 1 1 1 0 4 12 13 14 8 9 11 10 a'c a'd ab cd + Q.4 (c) f(a, b, c, d) = Σm(1, 2, 3, 5, 6, 7, 11, 12, 13, 14, 15) f(a, b, c, d) = a'c + a'd + ab + cd = a'(c+d) + ab + cd			
	c. For the below expression, draw the logic diagram using AOI logic for minimal sum. Obtain minimal sum using K-map  Logic diagram using AOI logic / AND/OR/INT /logic 	8		
5	a. What is hazard? List the types of hazards. Explain static 0 and static 1 hazard. **Solution:**  **HAZARDS & HAZARD COVERS** • The unwanted switching transient that may appear at the output of a circuit are called **hazards**. • The hazards cause the circuit to malfunction. • The main cause of hazards is the different propagation delays at different paths. • Here we consider, 1) Static-1 hazard 2) Static-0 hazard & 3) Dynamic hazard. **STATIC-1 HAZARD** • In a combinational circuit, if output goes momentarily 0 when it should remain a 1, the hazard is known as **static-1 hazard** (Figure: 3.34). • Static-1 hazard occurs when → Y=A+A' type of situation appears in a logic circuit. → 'A' makes a transition 1→0 • Here is how static-1 hazard occurs: ➢ An A+A' condition should always generate 1 at the output, i.e. static-1. ➢ But the NOT gate output takes finite time to become 1 following 1→0 transition of A. ➢ Thus for OR gate, there are 2 zeros appearing at its input for the small duration, resulting a 0 at its output. ➢ The width of this zero is in nanosecond order and is called a **glitch**. • In combinational circuits, static-1 hazard may not cause any serious problem. But in sequential circuit, static-1 hazard may cause major malfunctioning. A $\bar{A}$ Y (a) A $\bar{A}$ Y τ_1 τ_2 τ_1 τ_1 – NOT gate delay τ_2 – OR gate delay (b) Figure 3.34: Static-1 hazard	6	CO4	L2



STATIC-0 HAZARD

- In a combinational circuit, if output goes momentarily 1 when it should remain a 0, the hazard is known as **static-0 hazard** (Figure: 3.36).
- Static-0 hazard occurs when
  - → Y=AA' type of situation appears in a logic circuit.
  - → 'A' makes a transition 0→1.
- Here is how static-0 hazard occurs:
  - ➢ An AA' condition should always generate 0 at the output, i.e. static-0.
  - ➢ But the NOT gate output takes finite time to become 0 following 0→1 transition of A.
  - ➢ Thus for AND gate, there are 2 ones appearing at its input for the small duration, resulting a 1 at its output.
  - ➢ The width of this zero is in nanosecond order and is called a **glitch**.

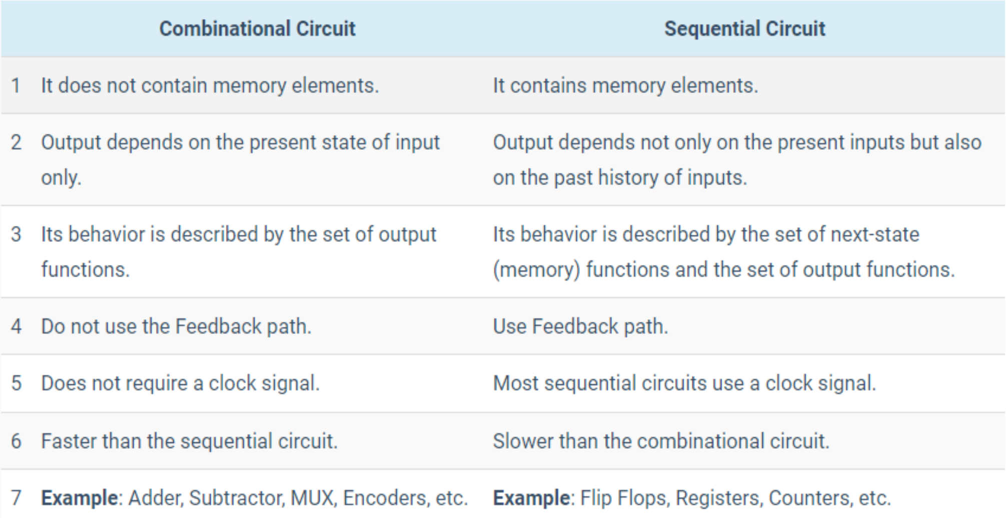
Figure 3.36: Static-0 hazard

(a) $Y=(B+\bar{C})(A+C)$ (b) Circuit with static-0 hazard (c) $Y=(B+\bar{C})(A+C)(A+B)$ (d) Hazard free circuit

Figure 3.37: Static-0 hazard and its cover

b. Differentiate between combinational and sequential circuit. 6 CO4 L1

Solution:

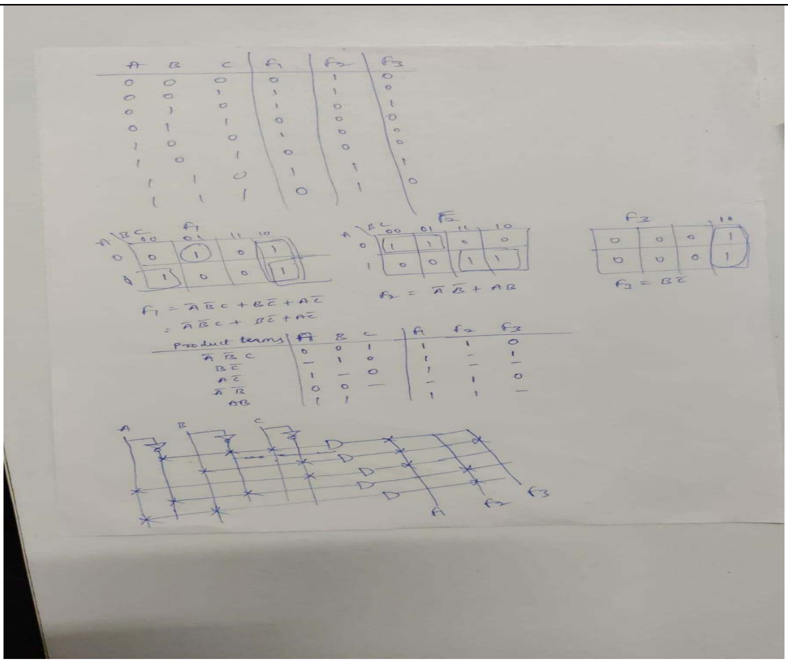


	Combinational Circuit	Sequential Circuit
1	It does not contain memory elements.	It contains memory elements.
2	Output depends on the present state of input only.	Output depends not only on the present inputs but also on the past history of inputs.
3	Its behavior is described by the set of output functions.	Its behavior is described by the set of next-state (memory) functions and the set of output functions.
4	Do not use the Feedback path.	Use Feedback path.
5	Does not require a clock signal.	Most sequential circuits use a clock signal.
6	Faster than the sequential circuit.	Slower than the combinational circuit.
7	**Example**: Adder, Subtractor, MUX, Encoders, etc.	**Example**: Flip Flops, Registers, Counters, etc.

.

c. Implement the following using PLA A=(1,2,4,6) B=(0,1,6,7) C=(2,6) 8

Solution:



A	B	C	f1	f2	f3
0	0	0	0	1	0
0	0	1	1	1	0
0	1	0	1	0	1
0	1	1	0	0	0
1	0	0	1	0	0
1	0	1	0	0	0
1	1	0	1	1	1
1	1	1	0	1	0

f1

A \ BC	00	01	11	10
0	0	1	0	1
1	1	0	0	1

f2

A \ BC	00	01	11	10
0	1	1	0	0
1	0	0	1	1

f3

			10
0	0	0	1
0	0	0	1

$f_1 = \bar{A}\bar{B}C + B\bar{C} + A\bar{C}$

$= \bar{A}\bar{B}C + B\bar{C} + A\bar{C}$

$f_2 = \bar{A}\bar{B} + AB$

$f_3 = B\bar{C}$

Product terms	A	B	C	f1	f2	f3
$\bar{A}\bar{B}C$	0	0	1	1	1	0
$B\bar{C}$	—	1	0	1	—	1
$A\bar{C}$	1	—	0	1	—	—
$\bar{A}\bar{B}$	0	0	—	—	1	0
AB	1	1		1	1	—

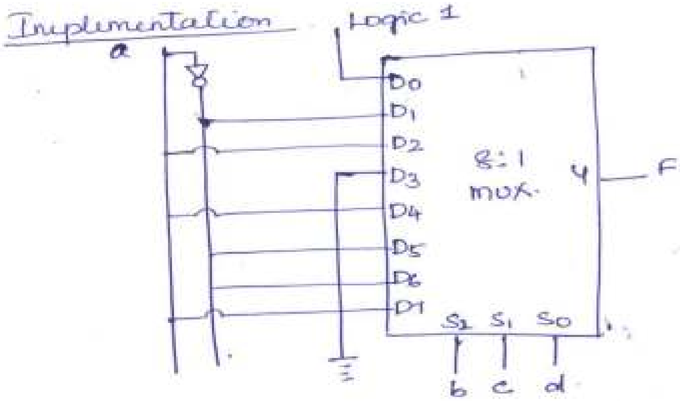
6 a. Implement the following function using 8:1 multiplexer F(a,b,c,d)=Σm(0,1,5,6,8,10,12,15) 7 CO5 L3

Solution:

Implementation Table :

	D0	D1	D2	D3	D4	D5	D6	D7
$\bar{a}$	0	1	2	3	4	5	6	7
a	8	9	10	11	12	13	14	15
	1	$\bar{a}$	a	0	a	$\bar{a}$	$\bar{a}$	a

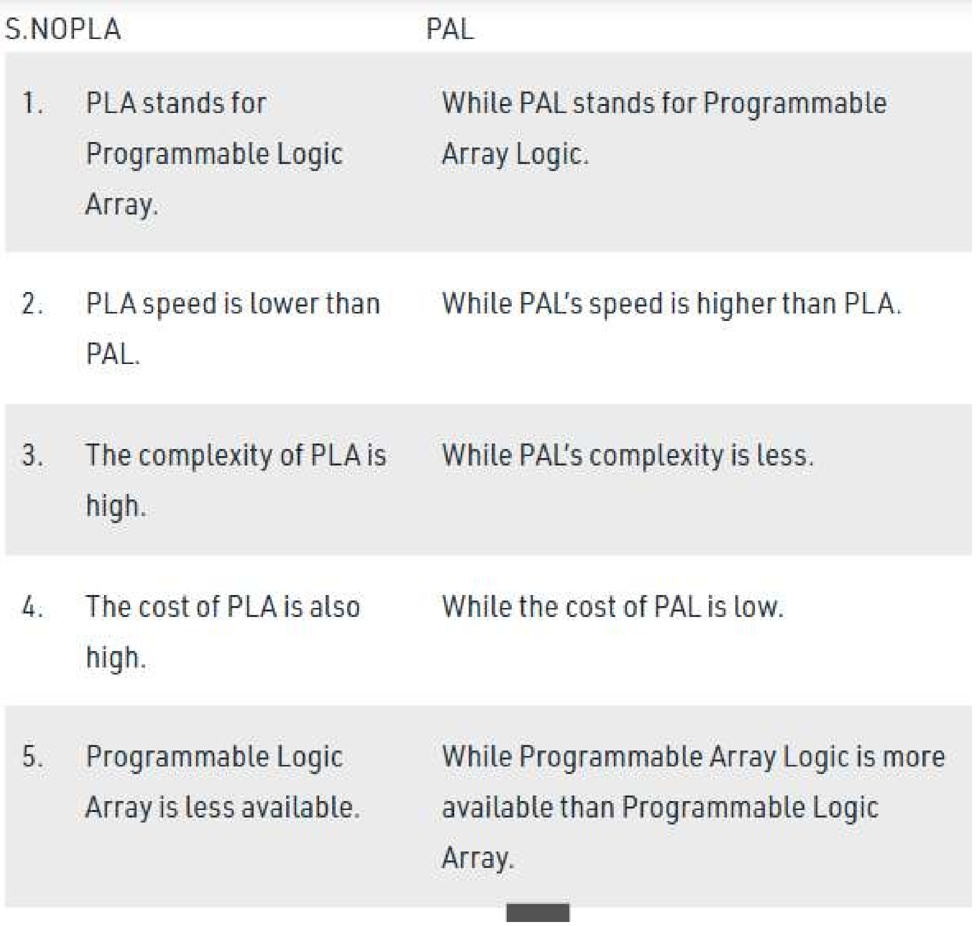
Implementation



b. What is programmable logic array? How does PLA differ from PAL? 6 CO5 L2

Solution:

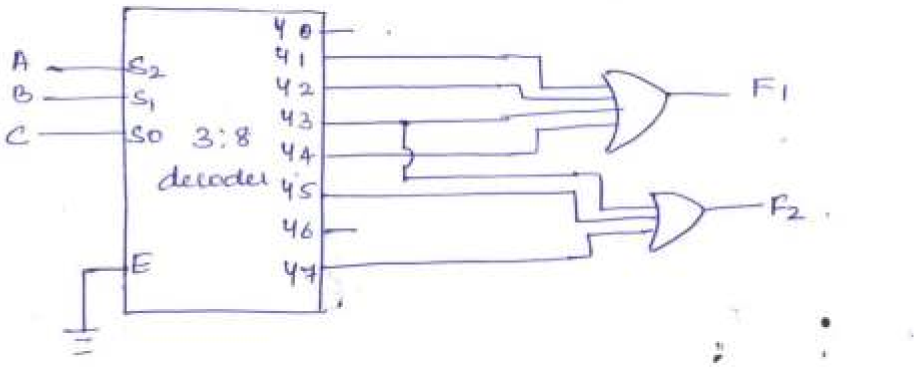
. A programmable logic array (PLA) is a kind of programmable logic device used to implement combinational logic circuits.



S.NO	PLA	PAL
1.	PLA stands for Programmable Logic Array.	While PAL stands for Programmable Array Logic.
2.	PLA speed is lower than PAL.	While PAL's speed is higher than PLA.
3.	The complexity of PLA is high.	While PAL's complexity is less.
4.	The cost of PLA is also high.	While the cost of PAL is low.
5.	Programmable Logic Array is less available.	While Programmable Array Logic is more available than Programmable Logic Array.

c. Realize the following using 3:8 decoder
i) f(a,b,c) = Σm(1,2,3,4) ii) f(a,b,c) = Σm(3,5,7)

7

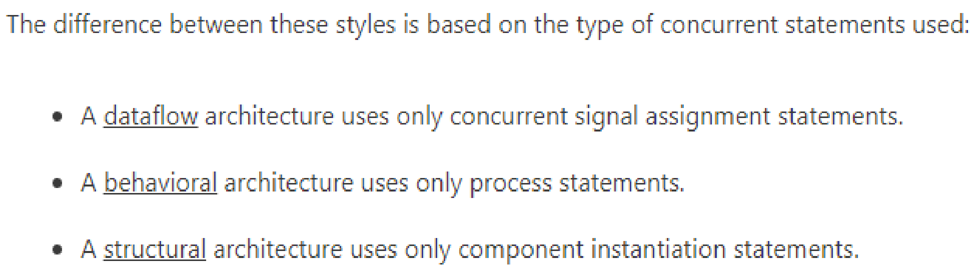


7

a. What are the three different models for writing a module body in VHDL? Give example for any one model

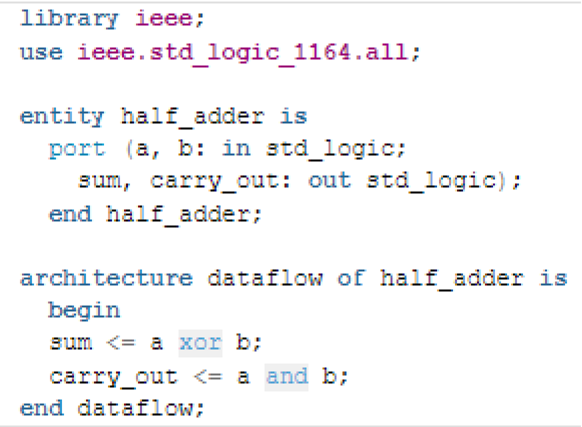
6 CO2 L2

Solution:



The difference between these styles is based on the type of concurrent statements used:

- A dataflow architecture uses only concurrent signal assignment statements.
- A behavioral architecture uses only process statements.
- A structural architecture uses only component instantiation statements.

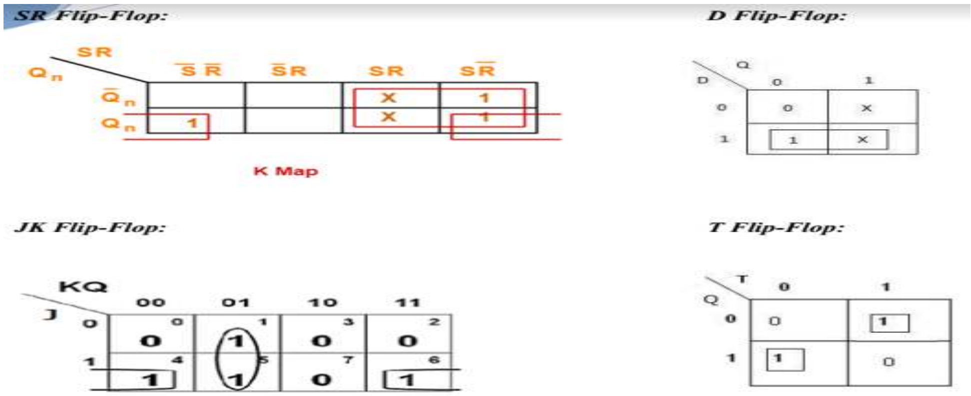


```
library ieee;
use ieee.std_logic_1164.all;

entity half_adder is
  port (a, b: in std_logic;
    sum, carry_out: out std_logic);
  end half_adder;

architecture dataflow of half_adder is
  begin
  sum <= a xor b;
  carry_out <= a and b;
end dataflow;
```

b. Derive characteristics equations for JK,T,D and SR flip flop

8 CO4 L2

Solution:



The characteristic equations for the latches and flip-flops discussed so far are:

$Q^+ = S + R'Q \ (SR = 0)$	(S-R latch or flip-flop)
$Q^+ = GD + G'Q$	(gated D latch)
$Q^+ = D$	(D flip-flop)
$Q^+ = D{\cdot}CE + Q{\cdot}CE'$	(D-CE flip-flop)
$Q^+ = JQ' + K'Q$	(J-K flip-flop)
$Q^+ = T \oplus Q = TQ' + T'Q$	(T flip-flop)

c. Give VHDL code for 4:1 multiplexer using conditional assign statement.

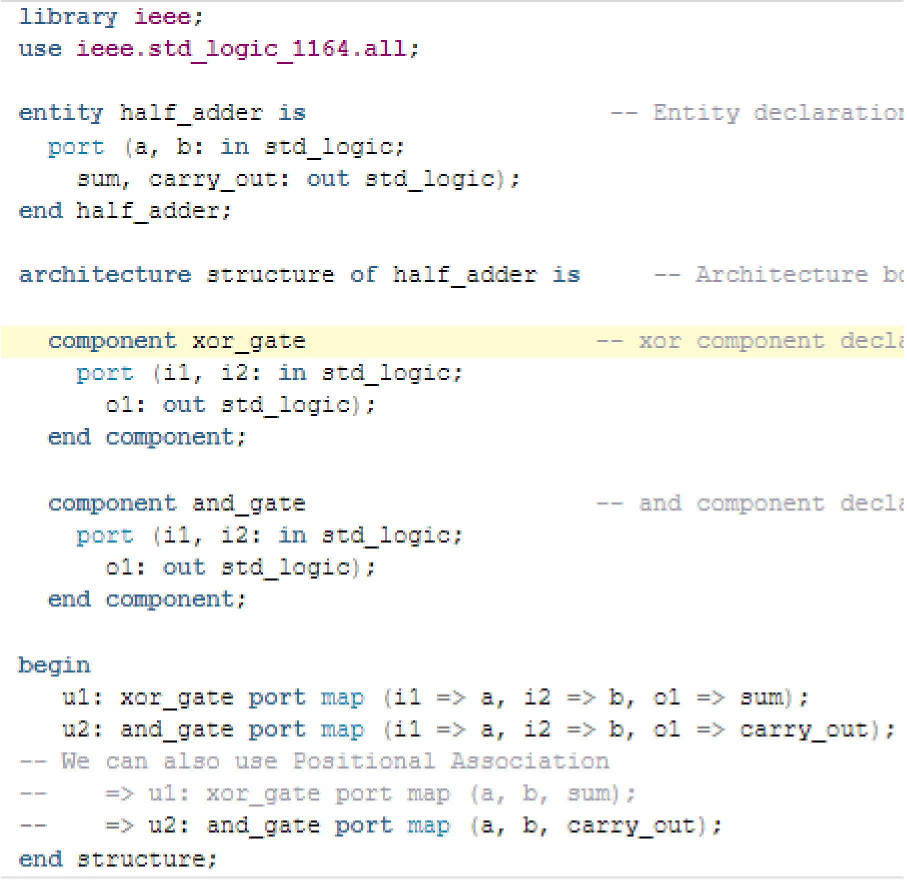
6

Solution:

```
library IEEE; use IEEE.STD_LOGIC_1164.all;
entity mux4 is
 port(d0, d1,
 d2, d3: in STD_LOGIC_VECTOR(3 downto 0);
 s: in STD_LOGIC_VECTOR(1 downto 0);
 y: out STD_LOGIC_VECTOR(3 downto 0));
end;
architecture synth1 of mux4 is
begin
 y &lt;= d0 when s = &quot;00&quot; else
 d1 when s = &quot;01&quot; else
 d2 when s = &quot;10&quot; else
 d3;
end;
```

8 a. Using structural model, write VHDL code for Half Adder. 6 CO5 L2

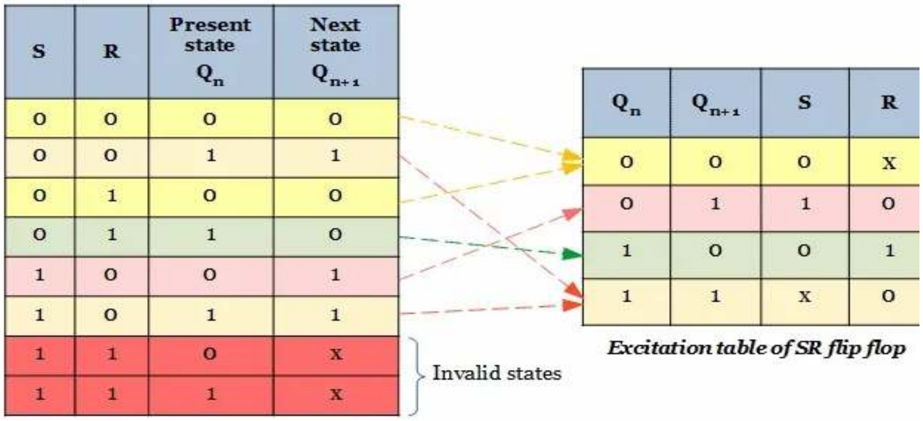
Solution:



```
library ieee;
use ieee.std_logic_1164.all;

entity half_adder is                          -- Entity declaratio
  port (a, b: in std_logic;
    sum, carry_out: out std_logic);
end half_adder;

architecture structure of half_adder is       -- Architecture bo

  component xor_gate                         -- xor component decla
    port (i1, i2: in std_logic;
      o1: out std_logic);
  end component;

  component and_gate                         -- and component decla
    port (i1, i2: in std_logic;
      o1: out std_logic);
  end component;

begin
   u1: xor_gate port map (i1 => a, i2 => b, o1 => sum);
   u2: and_gate port map (i1 => a, i2 => b, o1 => carry_out);
-- We can also use Positional Association
--    => u1: xor_gate port map (a, b, sum);
--    => u2: and_gate port map (a, b, carry_out);
end structure;
```

b. Derive the excitation table for JK flip flop and SR flip flop. How SR flip flop is converted to T flip flop. 8 CO5 L2

Solutions:

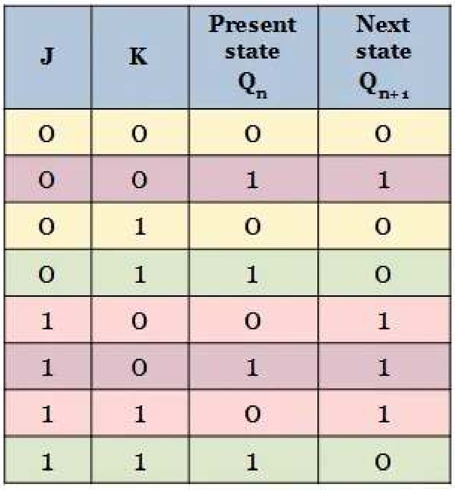


S	R	Present state Q_n	Next state Q_{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	x
1	1	1	x

Invalid states

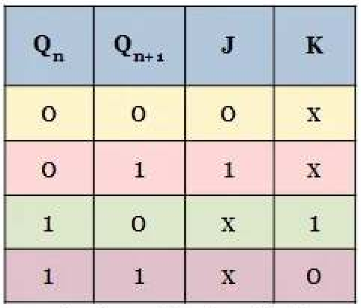
Q_n	Q_{n+1}	S	R
0	0	0	x
0	1	1	0
1	0	0	1
1	1	x	0

Excitation table of SR flip flop



J	K	Present state Q_n	Next state Q_{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Truth table of JK flip flop

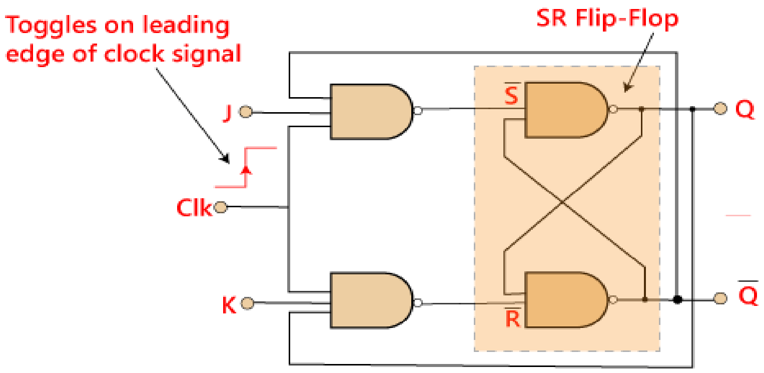


Q_n	Q_{n+1}	J	K
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0

Excitation table of JK flip flop

c. With logic diagram explain JK flip flop. 6

Solution:



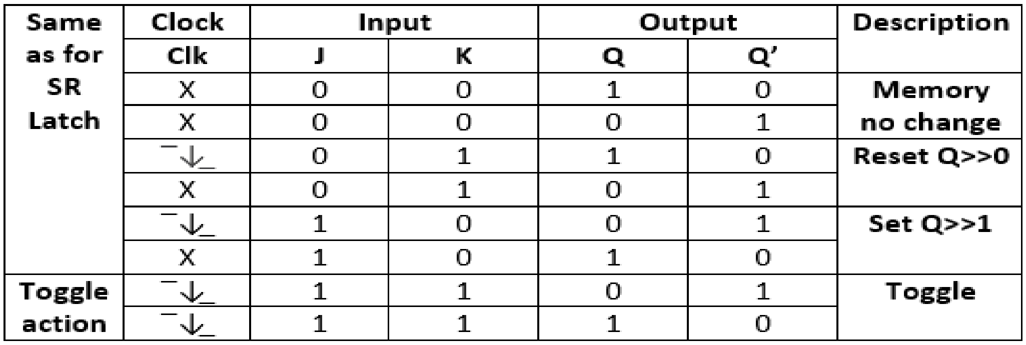
Circuit

In SR flip flop, both the inputs 'S' and 'R' are replaced by two inputs J and K. It means the J and K input equates to S and R, respectively.

The two 2-input AND gates are replaced by two 3-input NAND gates. The third input of each gate is connected to the outputs at Q and Q'. The cross-coupling of the SR flip-flop permits the previous invalid condition of (S = "1", R = "1") to be used to produce the "toggle action" as the two inputs are now interlocked.

If the circuit is "set", the J input is interrupted from the "0" position of Q' through the lower NAND gate. If the circuit is "RESET", K input is interrupted from 0 positions of Q through the upper NAND gate. Since Q and Q' are always different, we can use them to control the input. When both inputs 'J' and 'K' are set to 1, the JK toggles the flip flop as per the given truth table.

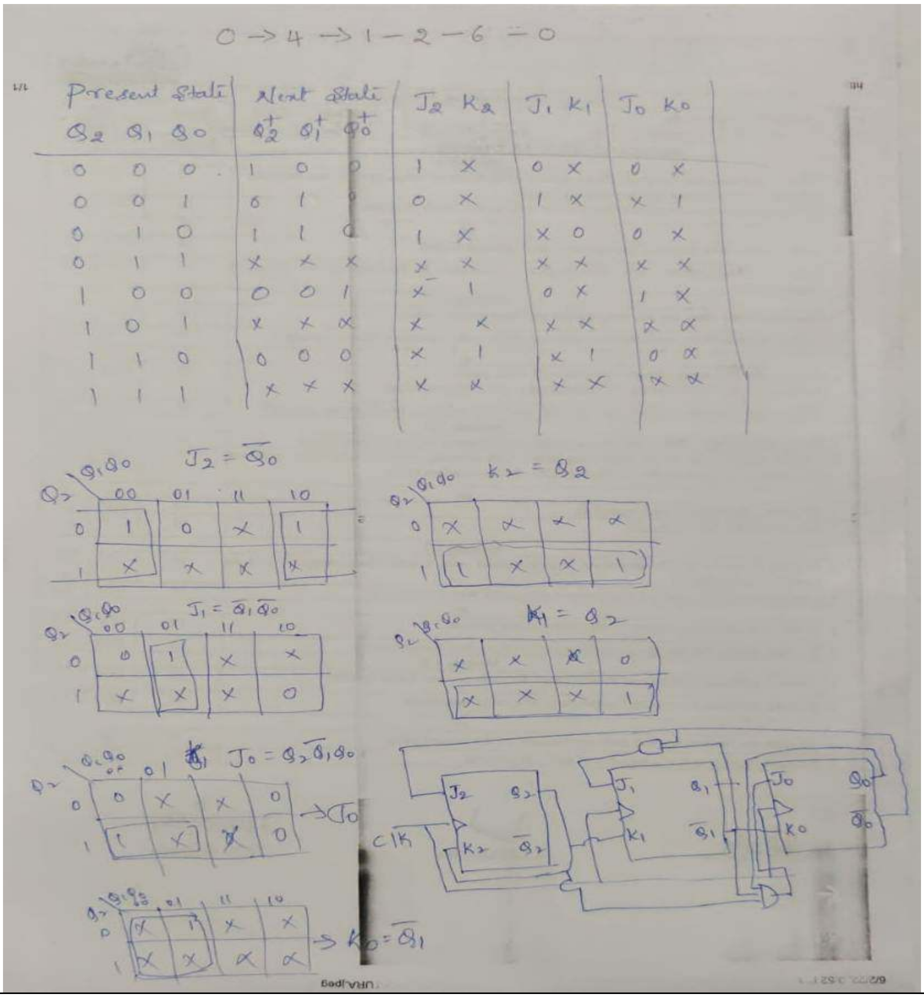
Truth Table:



Same	Clock	Input		Output		Description
as for	Clk	J	K	Q	Q'	
SR Latch	X	0	0	1	0	Memory no change
	X	0	0	0	1	
	↓	0	1	1	0	Reset Q>>0
	X	0	1	0	1	
	↓	1	0	0	1	Set Q>>1
	X	1	0	1	0	
Toggle action	↓	1	1	0	1	Toggle
	↓	1	1	1	0	

9 a. Define counter. Design synchronous counter for the sequence 0,4,1,2,6,0,4 using JK flip flop. 8

Solutions:



0 → 4 → 1 – 2 – 6 – 0

Present State			Next State								
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	1	0	0	1	X	0	X	0	X
0	0	1	0	1	0	0	X	1	X	X	1
0	1	0	1	1	0	1	X	X	0	0	X
0	1	1	X	X	X	X	X	X	X	X	X
1	0	0	0	0	1	X	1	0	X	1	X
1	0	1	X	X	X	X	X	X	X	X	X
1	1	0	0	0	0	X	1	X	1	0	X
1	1	1	X	X	X	X	X	X	X	X	X

$J_2 = \overline{Q_0}$

Q_2 \ Q_1Q_0	00	01	11	10
0	1	0	X	1
1	X	X	X	X

$K_2 = Q_2$

Q_2 \ Q_1Q_0				
0	X	X	X	X
1	1	X	X	1

$J_1 = \overline{Q_1}\,\overline{Q_0}$

Q_2 \ Q_1Q_0	00	01	11	10
0	0	1	X	X
1	X	X	X	0

$K_1 = Q_2$

Q_2 \ Q_1Q_0				
0	X	X	X	0
1	X	X	X	1

$J_0 = Q_2\overline{Q_1}Q_0$

Q_2 \ Q_1Q_0	00	01		
0	0	X	X	0
1	1	X	X	0

→ J_0

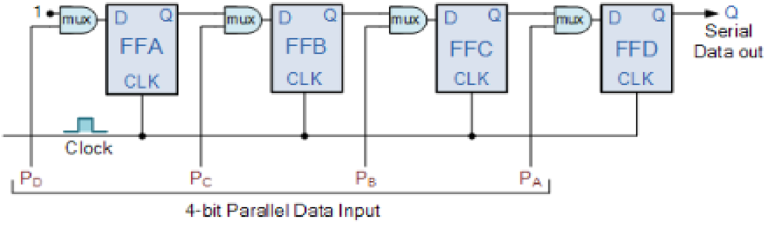
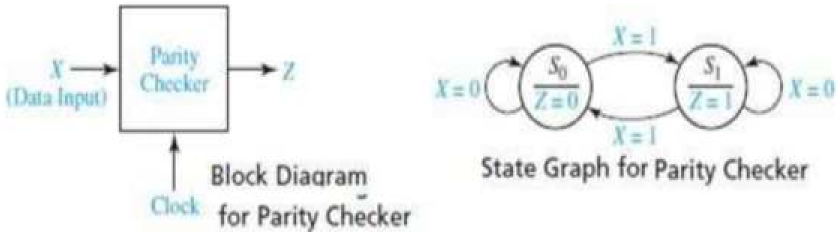
Q_2 \ Q_1Q_0	00	01	11	10
0	X	1	X	X
1	X	X	X	X

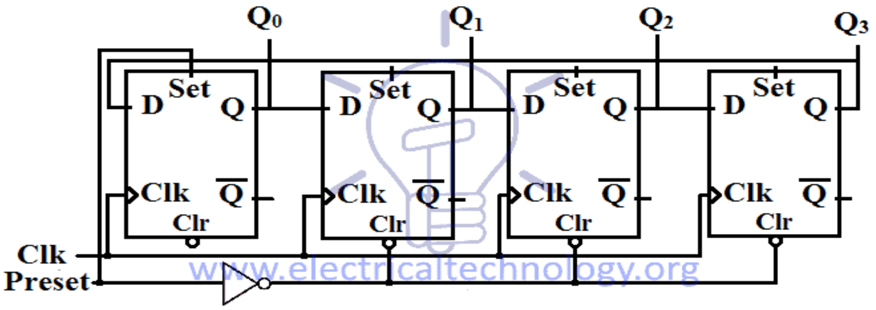
→ $K_0 = \overline{Q_1}$

b. What is shift register? With a neat diagram, explain 4 bit parallel in serial out shift register. 8 CO4 L2

Solution:

A shift register is a digital memory circuit found in calculators, computers, and data-processing systems. Bits (binary digits) enter the shift register at one end and emerge from the other end.

	Parallel-in to Serial-out (PISO) Shift Register The Parallel-in to Serial-out shift register acts in the opposite way to the serial-in to parallel-out one above. The data is loaded into the register in a parallel format in which all the data bits enter their inputs simultaneously, to the parallel input pins P_A to P_D of the register. The data is then read out sequentially in the normal shift-right mode from the register at Q representing the data present at P_A to P_D. This data is outputted one bit at a time on each clock cycle in a serial format. It is important to note that with this type of data register a clock pulse is not required to parallel load the register as it is already present, but four clock pulses are required to unload the data. **4-bit Parallel-in to Serial-out Shift Register**  1 mux D Q FFA CLK FFB FFC FFD Q Serial Data out Clock P_D P_C P_B P_A 4-bit Parallel Data Input			
	c. Write a note on sequential parity checker. **Solutions:**  X (Data Input) Parity Checker Z Clock X = 1 X = 0 S0 Z = 0 S1 Z = 1 X = 0 X = 1 Block Diagram for Parity Checker State Graph for Parity Checker The sequential circuit must -remember‖ whether the total number of 1 inputs received is even or odd; therefore, only two states are required. We will designate these states as S0 and S1, corresponding respectively to an even number of 1's received and an odd number of 1's received. We will start the circuit in state S0 because initially zero 1's have been received, and zero is an evennumber. As indicated in state graph (above Figure), if the circuit is in state S0 (even number of 1's received) and X = 0 is received, the circuit must stay in S0 because the number of 1's received is still even. However, if X = 1 is received, the circuit goes to state S1 because the number of 1's received is then odd. Similarly, if the circuit is in state S1 (odd number of 1's received) a 0 input causes no state change, but a 1 causes a change to S0 because the number of 1's received is then even. The output Z should be 1 whenever the circuit is in state S1 (odd number of 1's received). The output is listed below the state on the state graph.	4	CO4	L2
10	a. With a neat diagram, explain ring counter **Solutions:** Working of Ring Counter Ring counter's state needs to be set before the operation. Since ring counter circulates 1 through all stages, and there are no external inputs except the clock signal. So we need to set its state to initial state 1000 manually. We need to set the first stage flip-flop and clear the rest of the stages to obtain the state 1000.The preset input pin is designed to do this function. The schematic of ring counter is given below:	6	CO5	L2



First, we need to set the initial state 1000 through preset input.

Whenever the first clock edge hits the counter the outputs of each stage shifts to the next succeeding stage. And the output of the last will shift to the first stage making the state 0100.

Upon next clock cycle, each stage will update its state according to its input. So the '1' will be shifted to the third stage making the state 0010. Upon another clock cycle, the '1' will reach the last stage making the 0001.

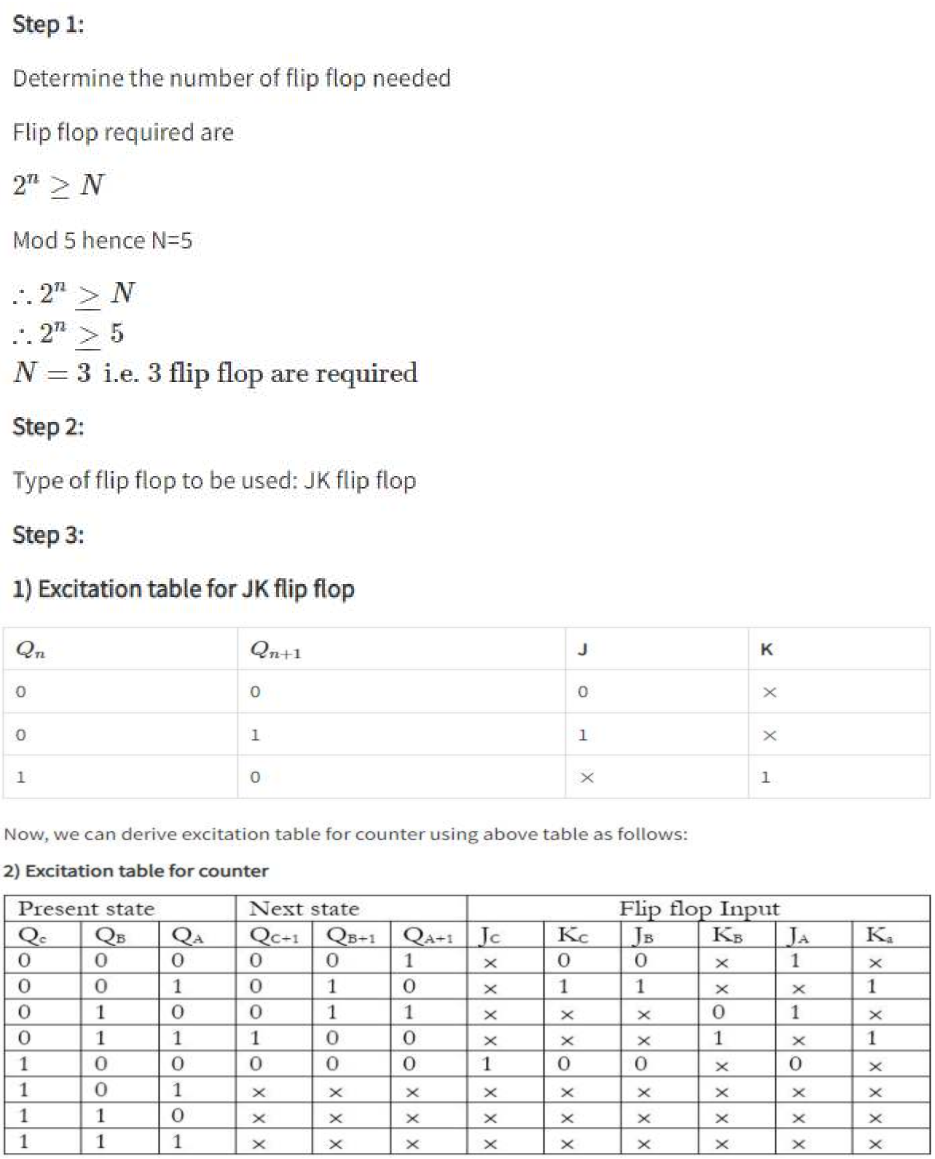
Now upon next clock cycle, '1' from the last stage (flip-flop) will shift back to the first stage making the initial state 1000. And it starts again from the first state repeating itself considering the clock signal is provided. This is how the data inside the ring counter circulates in the ring.

Ring counter divides the frequency of the clock signal by 'n'. n is the bit size of the ring counter. So ring counter can be used as a frequency divider.

b. Design and implement MOD 5 synchronous counter using JK flip flop. Explain with timing diagram.

8 CO1 L2

Solutions:



Step 1:

Determine the number of flip flop needed

Flip flop required are

$2^n \geq N$

Mod 5 hence N=5

$\therefore 2^n \geq N$
$\therefore 2^n \geq 5$
$N = 3$ i.e. 3 flip flop are required

Step 2:

Type of flip flop to be used: JK flip flop

Step 3:

1) Excitation table for JK flip flop

Q_n	Q_{n+1}	J	K
0	0	0	×
0	1	1	×
1	0	×	1

Now, we can derive excitation table for counter using above table as follows:

2) Excitation table for counter

Present state			Next state			Flip flop Input					
Q_c	Q_B	Q_A	Q_{C+1}	Q_{B+1}	Q_{A+1}	J_C	K_C	J_B	K_B	J_A	K_a
0	0	0	0	0	1	×	0	0	×	1	×
0	0	1	0	1	0	×	1	1	×	×	1
0	1	0	0	1	1	×	×	×	0	1	×
0	1	1	1	0	0	×	×	×	1	×	1
1	0	0	0	0	0	1	0	0	×	0	×
1	0	1	×	×	×	×	×	×	×	×	×
1	1	0	×	×	×	×	×	×	×	×	×
1	1	1	×	×	×	×	×	×	×	×	×

Step 4

K-map simplification

For J_C

Q_BQ_A \ Q_C	00	01	11	10
0	0	0	1	0
1	X	X	X	X

$J_c = Q_BQ_A$

For K_C

Q_BQ_A \ Q_C	00	01	11	10
0	X	X	X	X
1	1	X	X	X

$K_C = 1$

For K_B

Q_BQ_A \ Q_C	00	01	11	10
0	×	×	1	0
1	×	×	×	×

$K_B = QA$

For J_A

Q_BQ_A \ Q_C	00	01	11	10
0	1	×	×	1
1	0	×	×	×

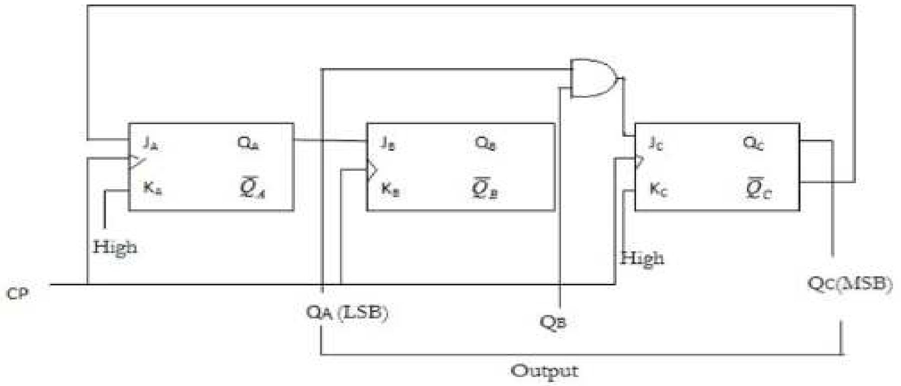
$J_A = \overline{Q}_C$

For K_A

Q_BQ_A \ Q_C	00	01	11	10
0	×	1	1	×
1	×	×	×	×

$K_A = 1$

Step 5 Logic Diagram



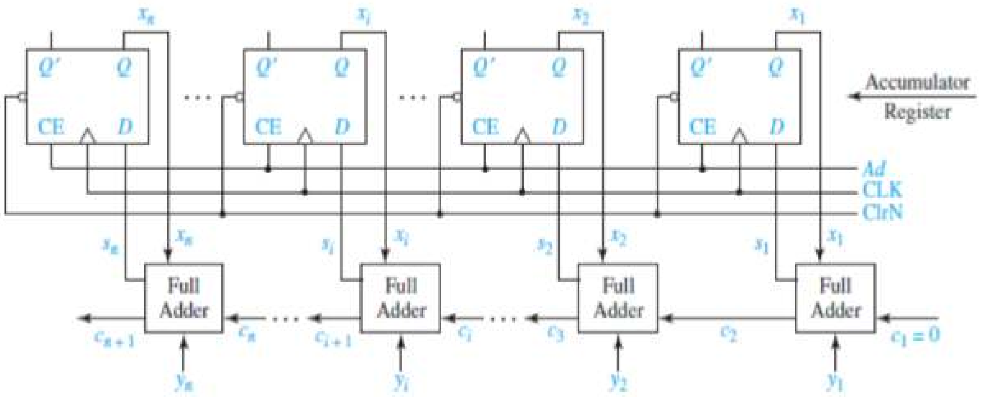
c. Write a note on parallel adder with accumulator. 6

Solution:

Parallel Adder with Accumulator:

In computer circuits, it is frequently desirable to store one number in a register of flip-flops (called an accumulator) and add a second number to it, leaving the result stored in the accumulator.

One way to build a parallel adder with an accumulator is to add a register to the adder as shown in the following Figure.



Suppose that the number $X = x_n \dots x_2x_1$ is stored in the accumulator. Then, the number $Y = y_n \dots y_2y_1$ is applied to the full adder inputs, and after the carry has propagated through the adders, the sum of X and Y

	appears at the adder outputs. An add signal (Ad) is used to load the adder outputs into the accumulator flip-flops on the rising clock edge. If $s_i = 1$, the next state of flip-flop x_i will be 1. If $s_i = 0$, the next state of flip-flop x_i will be 0.Thus, $x_i^+ = s_i$, and if $Ad = 1$, the number X in the accumulator is replaced with the sum of X and Y, following the rising edge of the clock. Observe that the adder with accumulator is an iterative structure that consists of a number of identical cells. Each cell contains a full adder and an associated accumulator flip-flop. Cell i, which has inputs c_i and y_i and outputs $c_i = 1$ and x_i, is referred to as a typical cell. Before addition can take place, the accumulator must be loaded with X. This can be accomplished in several ways. The easiest way is to first clear the accumulator using the asynchronous clear inputs on the flip-flops, and then put the X data on the Y inputs to the adder and add to the accumulator in the normal way. Alternatively, we could add multiplexers at the accumulator inputs so that we could select either the Y input data or the adder output to load into the accumulator. This would eliminate the extra step of clearing the accumulator but would add to the hardware complexity.			