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

Internal Assessment Test 2 – Dec 2022

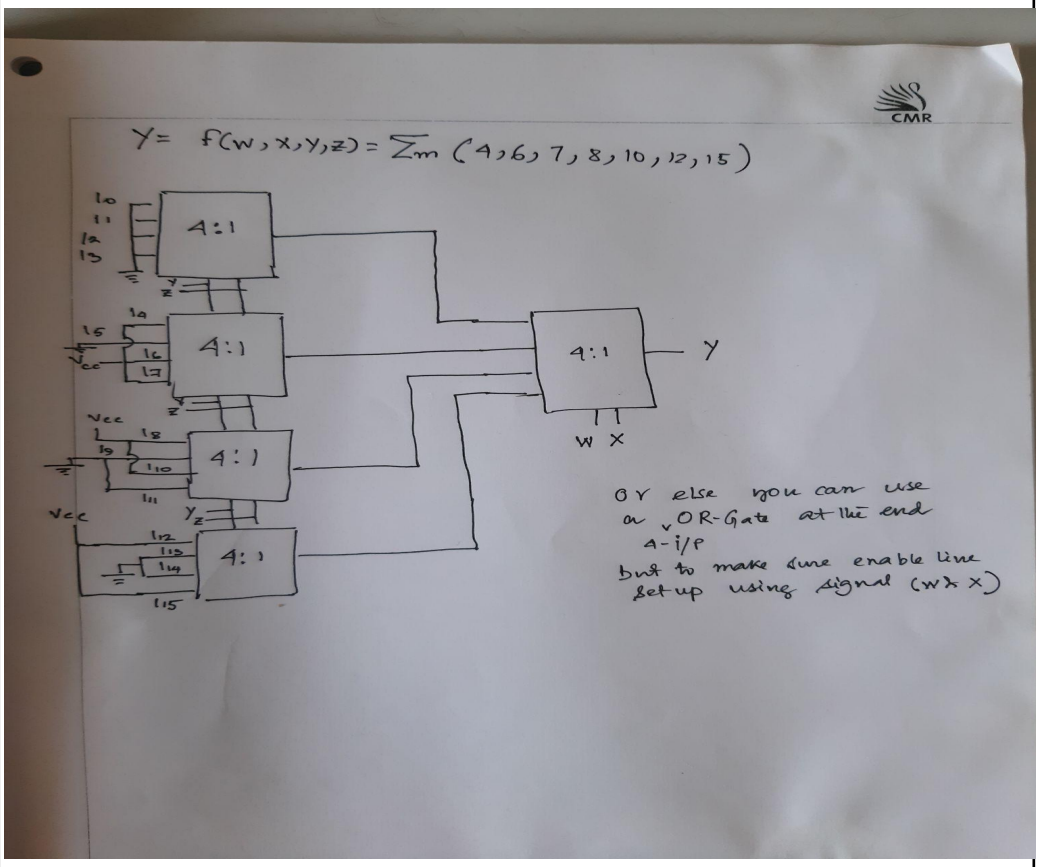
| Sub: | Analog and Digital Electronics | | | | | Sub Code: | 21CS33 | Branch: | CSE |
|---|---|---|---|---|---|---|---|---|---|
| Date: | 27/12/22 | Duration: | 90 mins | Max Marks: | 50 | Sem / Sec: | 3rd semester/A,B,C | | OBE |

Answer any FIVE FULL Questions

| | | MARKS | CO | RBT |
|---|---|---|---|---|
| 1 | A) Realize the Boolean expression **f(w,x,y,z)= Σm (4,6,7,8,10,12,15)** Using 4 to 1 line multiplexer and external gates.<br>B) Define decoder. Draw Truth Table and logic diagram of **3:8 decoder**.<br>C) Differentiate between combinational circuit and sequential circuit. | [4+4+2] | CO2 | L2,L3 |

**ANS:**

**A)**




**B)** A 3 to 8 decoder has three inputs (A, B, C) and eight outputs (D0 to D7).

- Based on the 3 inputs one of the eight outputs is selected.
- The truth table for 3 to 8 decoder is shown in the below table.
- From the truth table, it is seen that only one of eight outputs (D0 to D7) is selected based on three select inputs.

| A | B | C | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

$$D_0 = \bar{A}\bar{B}\bar{C}, \quad D_1 = \bar{A}\bar{B}C, \quad D_2 = \bar{A}B\bar{C},$$
$$D_3 = \bar{A}BC, \quad D_4 = A\bar{B}\bar{C}, \quad D_5 = A\bar{B}C,$$
$$D_6 = AB\bar{C}, \quad D_7 = ABC$$

C)

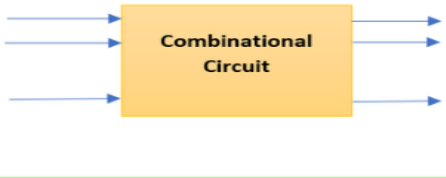
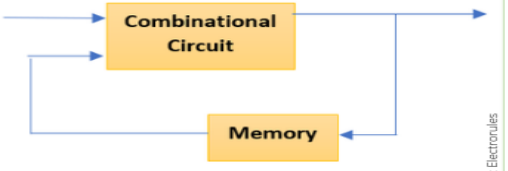
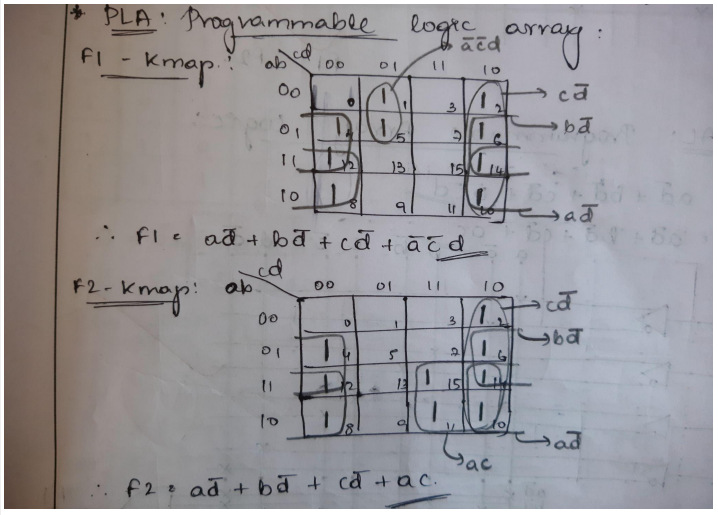
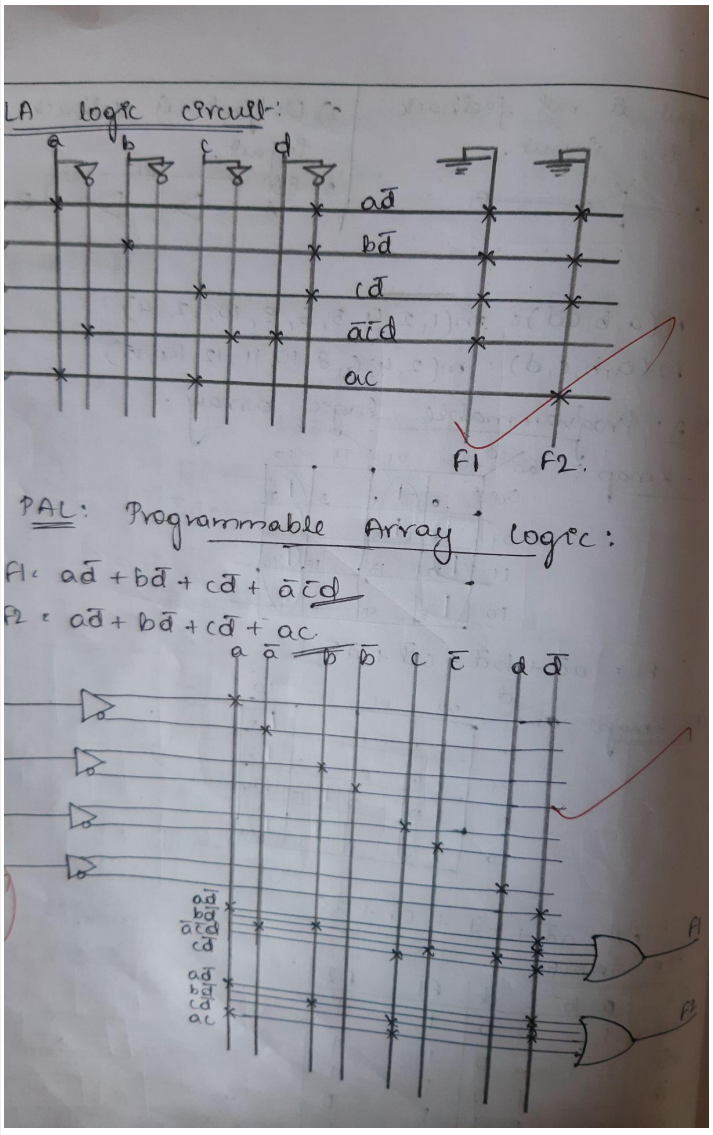
| Combinational Circuit | Sequential Circuit |
|---|---|
| Output only depends on the present input | Output depends on present input and past output |
| Memory element is absent | Memory element is present |
| No clock signal is applied | Clock signal is required |
|   |   |
| Example - Half Adder, Full Adder, Multiplexer | Examples - Flipflop, Counters, Registers |

Image Credits: Electronicles

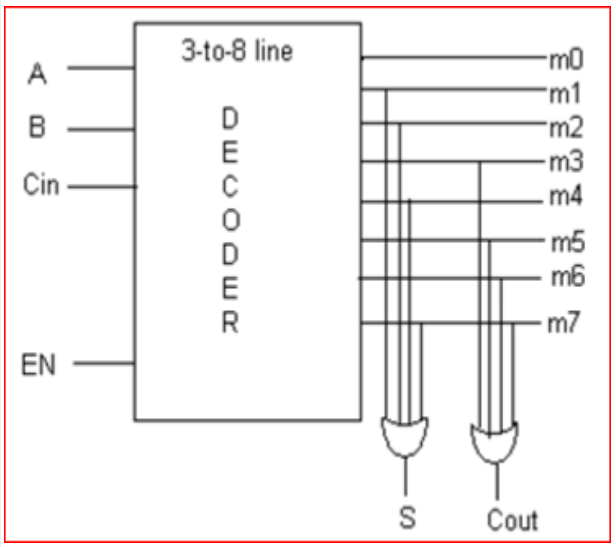
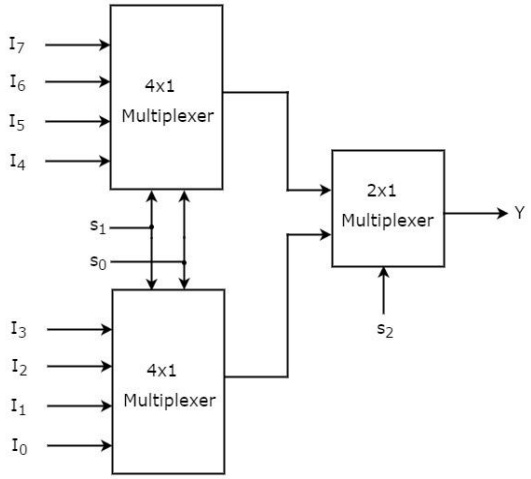
| | | | | |
|---|---|---|---|---|
| 2 | A) Consider two 4 variable functions *F*1 and *F*2.<br>***F*1(*a*, *b*, *c*, *d*) = *m*(1, 2, 4, 5, 6, 8, 10, 12, 14)**<br>***F*2(*a*, *b*, *c*, *d*) = *m*(2, 4, 6, 8, 10, 11, 12, 14, 15)**<br>Realize *F*1 and *F*2 using a **PLA, PAL**.<br>B) Implement **Full Adder** Circuit using Decoder<br><br>**ANS:**<br>  | [4+3+3] | CO2,<br>CO2 | L2,L3 |

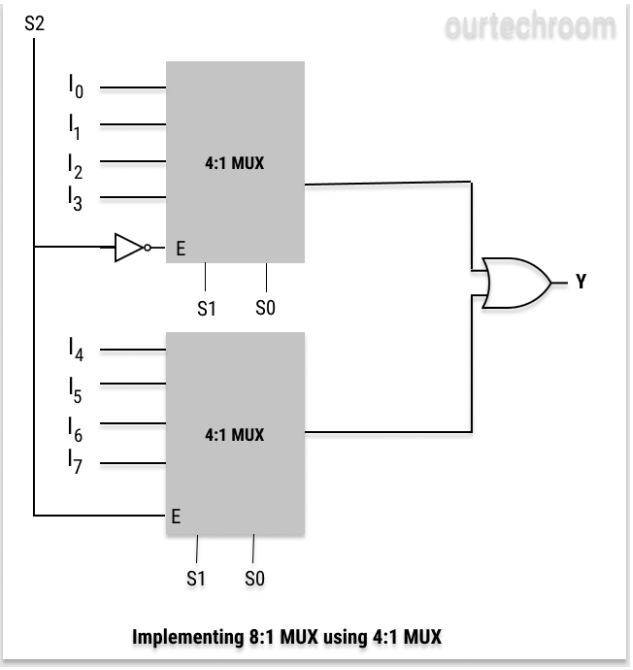
PLA TABLE:

| PI | a | b | c | d | F1 | F2 |
|---|---|---|---|---|---|---|
| $a\bar{d}$ | 1 | – | – | 0 | 1 | 1 |
| $b\bar{d}$ | – | 1 | – | 0 | 1 | 1 |
| $c\bar{d}$ | – | – | 1 | 0 | 1 | 1 |
| $\bar{a}\bar{c}d$ | 0 | – | 0 | 1 | 1 | 0 |
| $ac$ | 1 | – | 1 | – | 0 | 1 |




B)

| | | | | | |
|---|---|---|---|---|---|
| | <table><tr><th>Input bit for number A</th><th>Input bit for number B</th><th>Carry bit input $C_{IN}$</th><th>Sum bit output S</th><th>Carry bit output $C_{OUT}$</th></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr></table><br>  | | | |
| 3 | i) Realize a 8:1 Multiplexer using<br>**A) 4:1 Multiplexer & one 2:1 Multiplexer**<br>**B) Only by using 4:1 Multiplexer**<br>ii) Implement **4:2 Priority Encoder** using basic gates.<br>iii) Realize 2:1 Multiplexer using **tri-state buffer**<br><br>**ANS:**<br>**i) A)**<br>  <br>**i) B)** | [4+4+2] | CO2,<br>CO2,<br>CO2 | L3,L2 |

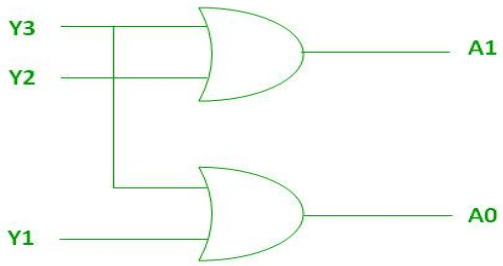



Implementing 8:1 MUX using 4:1 MUX

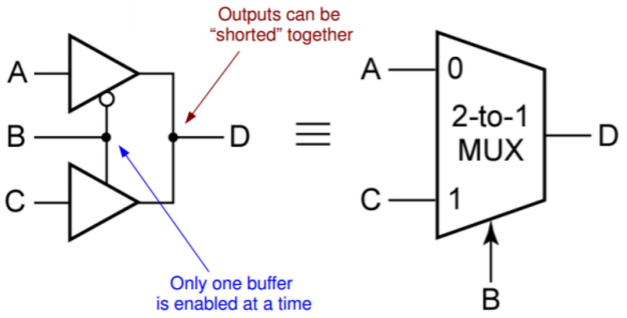
ii)

| INPUTS | | | | OUTPUTS | |
|---|---|---|---|---|---|
| Y3 | Y2 | Y1 | Y0 | A1 | A0 |
| 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 |

```
A1 = Y3 + Y2
A0 = Y3 + Y1
```




iii)




| | | | | |
|---|---|---|---|---|
| 4 | A) Write a VHDL module for a combinational circuit using data flow model<br>**i) 4:1 Multiplexer**<br>**ii) 8 : 3 Encoder**<br>B) Explain the application of the S-R latch is for debouncing switches. | [6+4] | CO2 | L3,L2 |

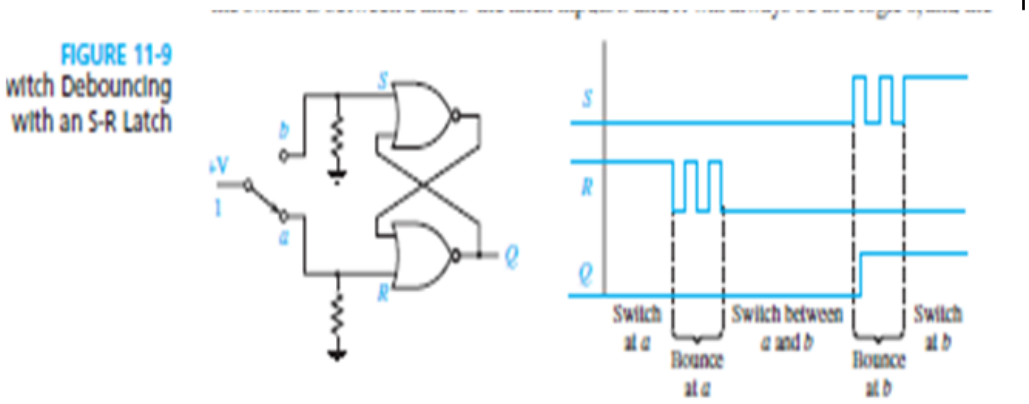
**ANS:**

**a) 4:1 Multiplexer**

```
Library ieee;
useieee.std_logic_1164.all;

entity mux is
port(S1,S0,D0,D1,D2,D3:in bit; Y:out bit);
end mux;

architecture data of mux is
begin
  Y<=(not S0 andnot S1 and D0)or
(S0 andnot S1 and D1)or
(not S0 and S1 and D2)or
(S0 and S1 and D3);
end data;
```

**b) 8 : 3 Encoder**

```
library ieee;
useieee.std_logic_1164.all;

entity encis
  port(i0,i1,i2,i3,i4,i5,i6,i7:in bit; o0,o1,o2:out bit);
end encis;

architecture adder4 of encis
begin
  o0<=i4 or i5 or i6 or i7;
  o1<=i2 or i3 or i6 or i7;
```

| | | | | |
|---|---|---|---|---|
| | o2<=i1 or i3 or i5 or i7;<br>end adder4;<br><br>B)<br>  <br><br>**When a mechanical switch is opened or closed, the switch contacts tend to vibrate or bounce open and closed several times before settling down to their final position.**<br><br>**This produces a noisy transition, and this noise can interfere with the proper operation of a logic circuit. This can avoided by contacting to SR latch.**<br><br>**1. The input to the switch in Figure above is connected to a logic 1 (+V).The pull-down resistors connected to contacts *a* and *b* assure that when the switch is between *a* and *b* the latch inputs *S* and *R* will always be at a logic 0, and the latch output will not change state.**<br><br>**2. The timing diagram shows what happens when the switch is flipped from *a* to*b*.**<br><br>**a. As the switch leaves *a*, bounces occur at the *R* input; when the switch reaches *b*, bounces occur at the *S* input.**<br><br>**b. After the switch reaches *b*, the first time *S* becomes 1, after a short delay the latch switches to the *Q* =1 state and remains there.**<br><br>**c. Thus *Q* is free of all bounces even though the switch contacts bounce.**<br><br>**This debouncing scheme requires a *double throw* switch that switches between two contacts;it will not work with a *single throw* switch that switches between one contact and open.** | | | |
| 5 | A) Derive the characteristic equations for the following latches and flip-flops in SOP form.<br>**a) Gated D latch b) D flip-flop c) J-K flip-flop d)T flip-flop**<br>B) Given a rising-edge-triggered D flip-flop with the following inputs, sketch the waveform for *Q*.assume Q=1<br><br>**ANS:** | [8+2] | CO2 | L2,L3 |

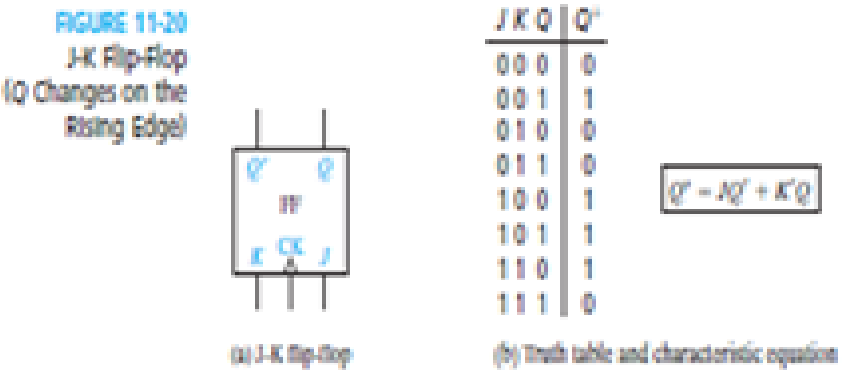
**Gated D latch**

Next state Q+ output 1 when G=0 and Q=1 or G=1 and D=1

**D flip-flop**

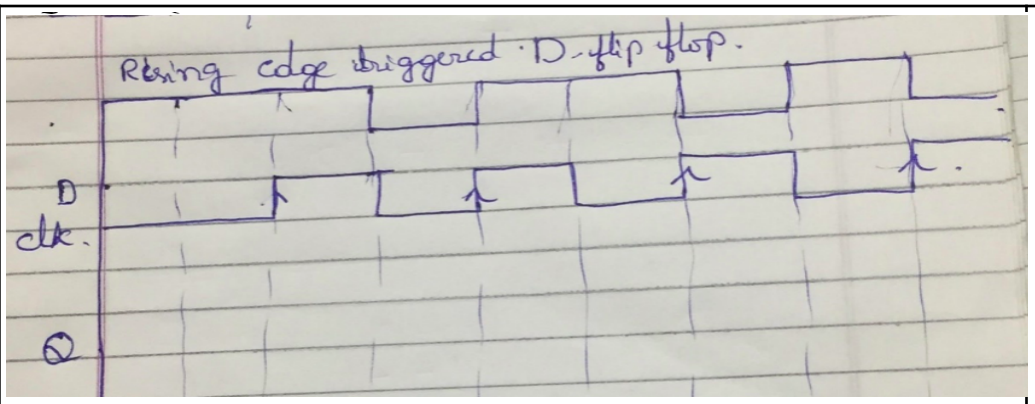
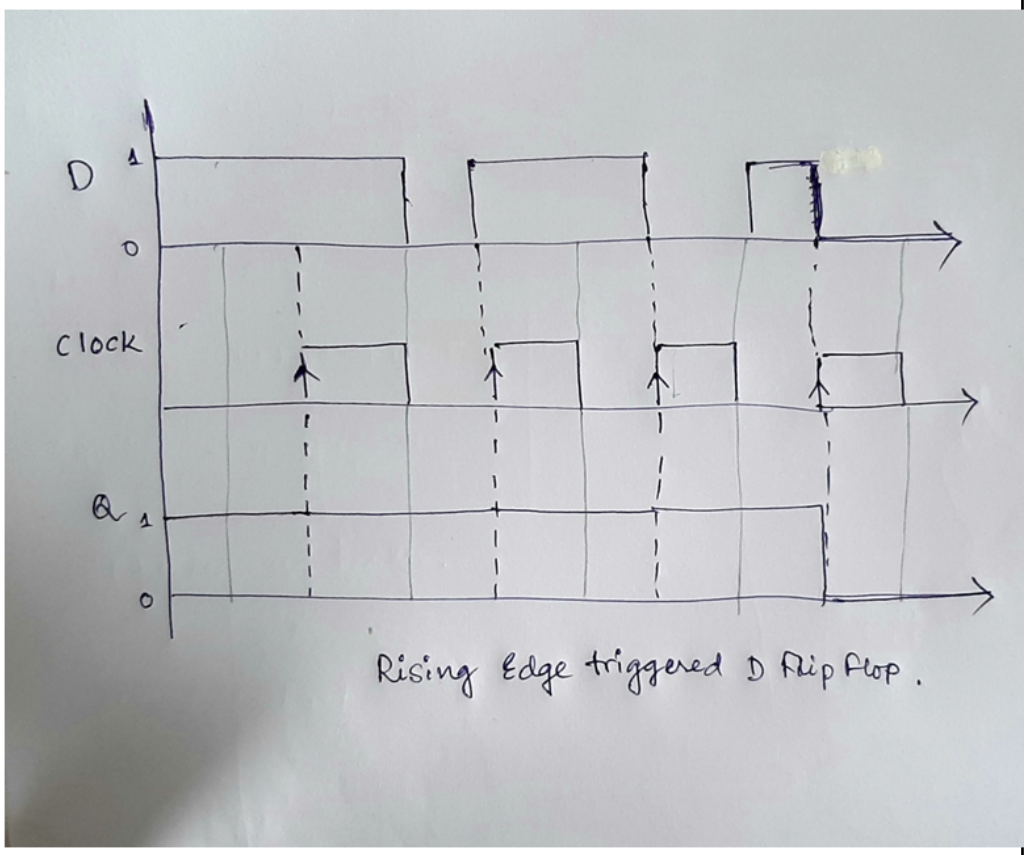
Next state Q+ output 1 when D=1

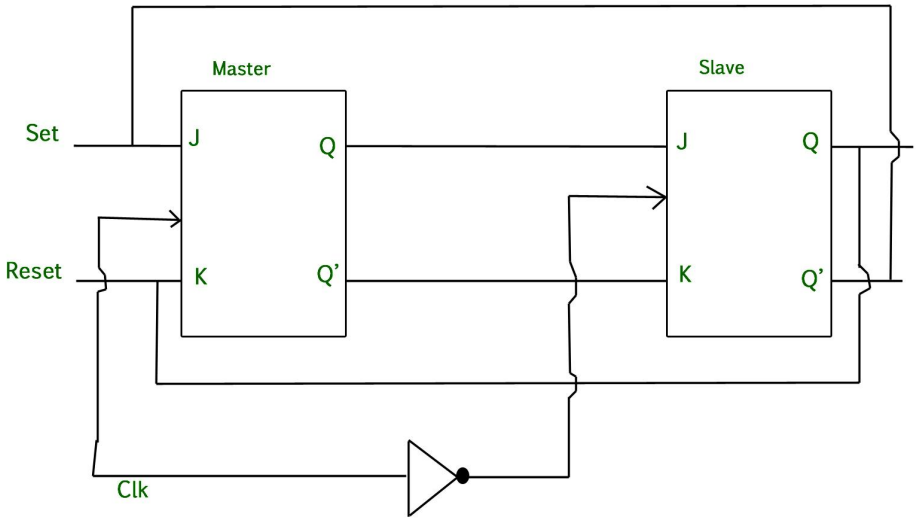
**J-K flip-flop**




Next state Q+ output 1 when J=1 and Q=0 or K=0 and Q=1

**T flip-flop**

| | | | | |
|---|---|---|---|---|
| |  <br>  | | | |
| 6 | Consider a basic **NAND based J-K FlipFlop** circuit with propagation delay of each gate is 10 ns. Assume the input signals state is **J=1 and K=1**, respectively and present state (Q) is 1. Considering the Level triggered clock duration is 1 ms. What will be the impact of clock signal on the output behavior of the circuit? If there are any major behavioral changes in the circuit output then how can this issue be rectified. | [10] | CO2 | L2,L2 |



Master-slave JK Flip-Flop

## Working of a master slave flip flop –

1. When the clock pulse goes to 1, the slave is isolated; J and K inputs may affect the state of the system. The slave flip-flop is isolated until the CP goes to 0. When the CP goes back to 0, information is passed from the master flip-flop to the slave and output is obtained.
2. Firstly the master flip flop is positive level triggered and the slave flip flop is negative level triggered, so the master responds before the slave.
3. If J=0 and K=1, the high Q' output of the master goes to the K input of the slave and the clock forces the slave to reset, thus the slave copies the master.
4. If J=1 and K=0, the high Q output of the master goes to the J input of the slave and the Negative transition of the clock sets the slave, copying the master.
5. If J=1 and K=1, it toggles on the positive transition of the clock and thus the slave toggles on the negative transition of the clock.
6. If J=0 and K=0, the flip flop is disabled and Q remains unchanged.

## CO PO Mapping

| 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 supply and regulator IC, op-amp and basic principles of A/D and D/A conversion circuits and to develop the same. | 1 | 3 | 3 | 3 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 2 |
| CO2 | Formulate logical expressions for minimized SOP, POS forms by use of Karnaugh maps, Quine Mc Cluskey method. | 2 | 3 | 3 | 3 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 2 |
| CO3 | Design combinational logic circuits using gates, encoders, decoders, multiplexers, demultiplexers, Comparators, arithmetic-logic units and to build simple applications. | 3 | 3 | 3 | 3 | 3 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 2 |
| CO4 | Understand the use of latches, flip-flops, Switch Contact Bounce Circuits and Various Representation of FLIP-FLOPs and use them in designing, registers and counters. | 5 | 3 | 3 | 3 | 3 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 2 |
| CO5 | Develop simple HDL programs | 4 | 3 | 3 | 3 | 3 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 2 |

| 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 | | | | |