MODULE-3 Sequential Circuits – 1: 7 Hours

Basic Bistable Element, Latches, SR Latch,Application of SR Latch, A Switch Debouncer, The S R Latch, The gated SR Latch, The gated D Latch, The Master-Slave Flip-Flops (Pulse-Triggered Flip-Flops): The Master-Slave SR Flip-Flops, The Master-Slave JK Flip- Flop, Edge Triggered Flip-Flop: The Positive Edge-Triggered D Flip-Flop, Negative-Edge Triggered D Flip-Flop

Recommended readings:

1. Donald D Givone, “Digital Principles and Design “, Tata McGraw

Hill Edition, 2002.

Unit - 6.1, 6.2, 6.4, 6.5

Introduction :

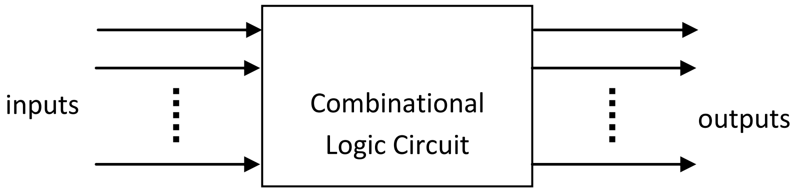
Logic circuit is divided into two types.

1. Combinational Logic Circuit
2. Sequential Logic Circuit

Definition :

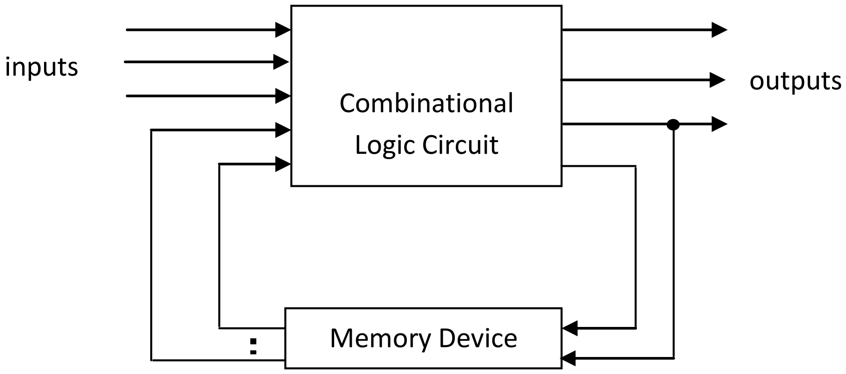
1. Combinational Logic Circuit :

   The circuit in which outputs depends on only present value of inputs. So it is possible to describe each output as function of inputs by using Boolean expression. No memory element involved. No clock input. Circuit is implemented by using logic gates. The propagation delay depends on, delay of logic gates. Examples of combinational logic circuits are : full adder, subtractor, decoder, codeconverter, multiplexers etc.



2. Sequential Circuits :

   Sequential Circuit is the logic circuit in which output depends on present value of inputs at that instant and past history of circuit i.e. previous output. The past output is stored by using memory device. The internal data stored in circuit is called as state. The clock is required for synchronization. The delay depends on propagation delay of circuit and clock frequency. The examples are flip-flops, registers, counters etc.



- Basic Bistable element.
  - Flip-Flop is Bistable element.

- It consist of two cross coupled NOT Gates.
- It has two stable states.
- Q and $\overline{Q}$ are two outputs complement of each other.
- The data stored 1 or 0 in basic bistable element is state of flip-flop.
- 1 – State is set condition for flip-flop.
- 0 – State is reset / clear for flip-flop.
- It stores 1 or 0 state as long power is ON.

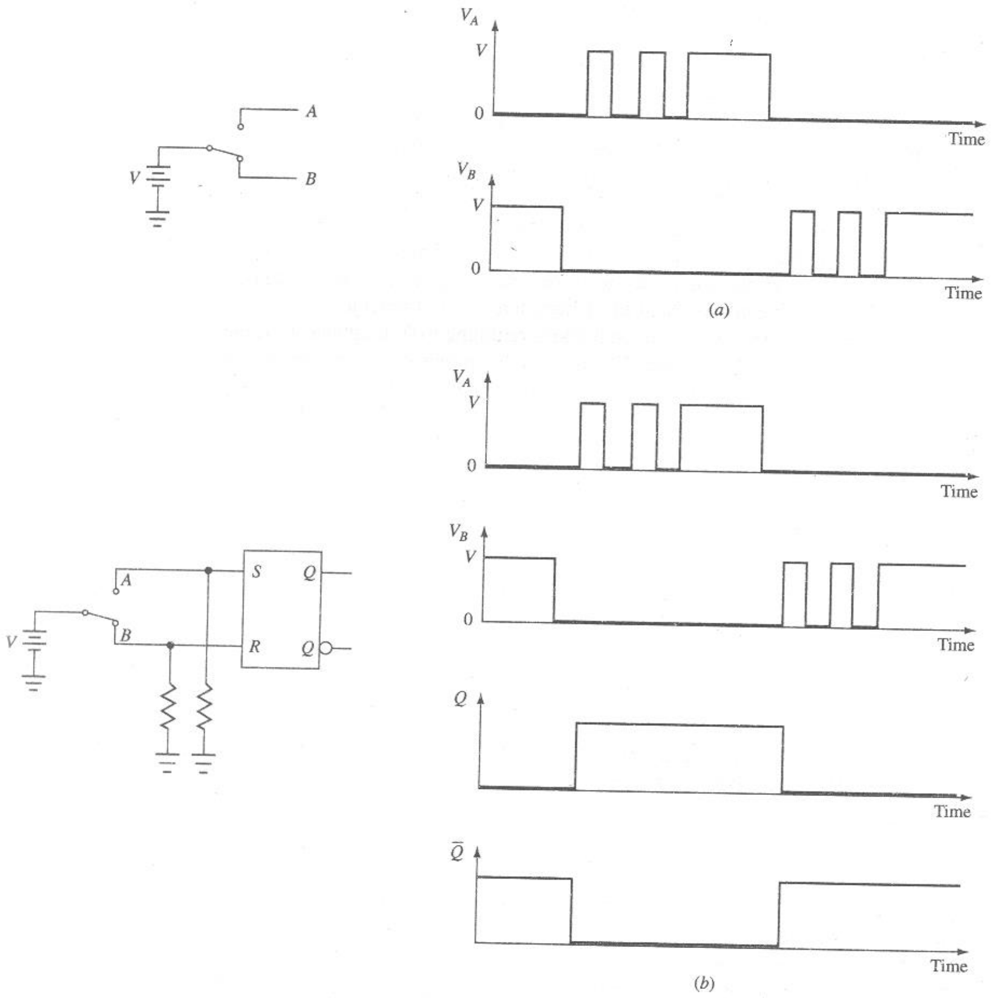
Latches :

S-R Latch : Set-reset Flip-Flop

- Latch is a storage device by using Flip-Flop.
- Latch can be controlled by direct inputs.
- Latch outputs can be controlled by clock or enable input.
- Q and $\overline{Q}$ are present state for output.
- Q^+ and $\overline{Q}^+$ are next states for output.
- The function table / Truth table gives relation between inputs and outputs.
- The S=R=1 condition is not allowed in SR FF as output is unpredictable.

Application of SR Latch :

- A switch debouncer

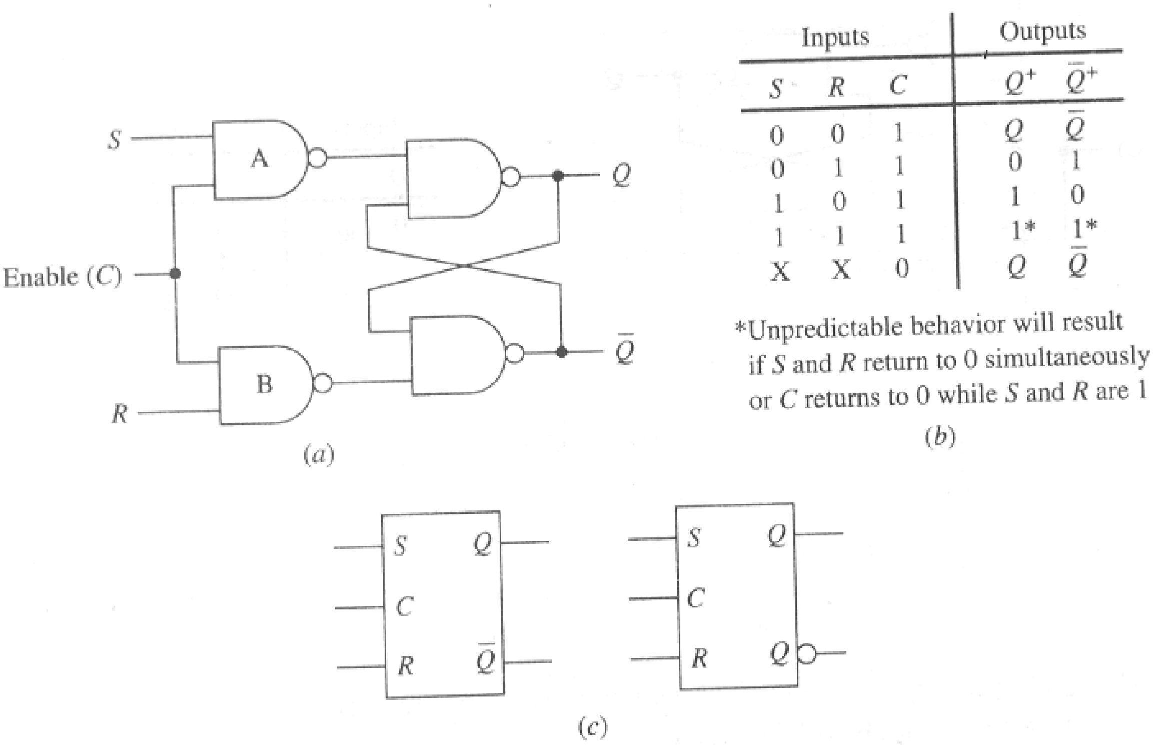


(a)

(b)

- Bouncing problem with Push button switch.
- Debouncing action.
- SR Flip-Flop as switch debouncer.

Gated SR Latch :



(a)

Inputs			Outputs	
S	R	C	Q^+	$\overline{Q}^+$
0	0	1	Q	$\overline{Q}$
0	1	1	0	1
1	0	1	1	0
1	1	1	1*	1*
X	X	0	Q	$\overline{Q}$

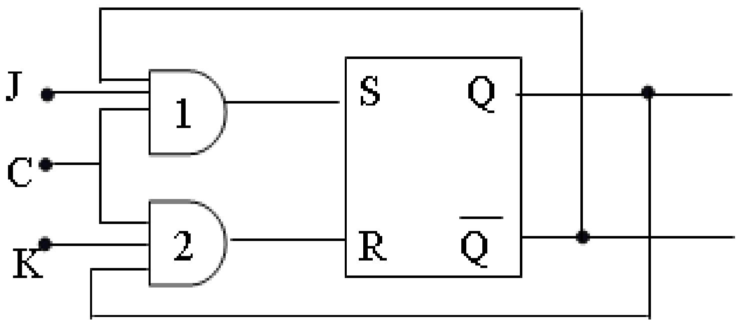
*Unpredictable behavior will result if S and R return to 0 simultaneously or C returns to 0 while S and R are 1

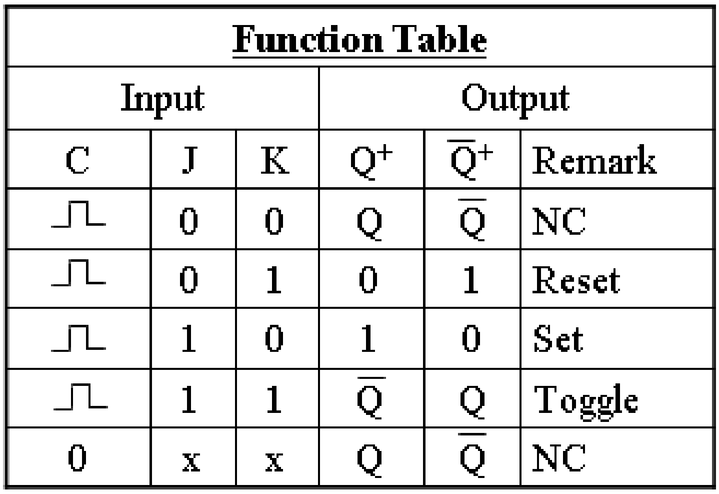
(b)

(c)

- Enable input C is clock input.
- C=1, Output changes as per input condition.
- C=0, No change of state.
- S=1, R=0 is set condition for Flip-flop.
- S=0, R=1 is reset condition for Flip-flop.
- S=R=1 is ambiguous state, not allowed.

JK Flip-Flop by using SR Flip-Flop



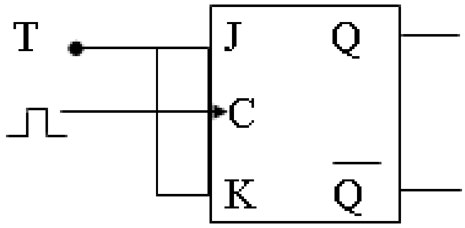


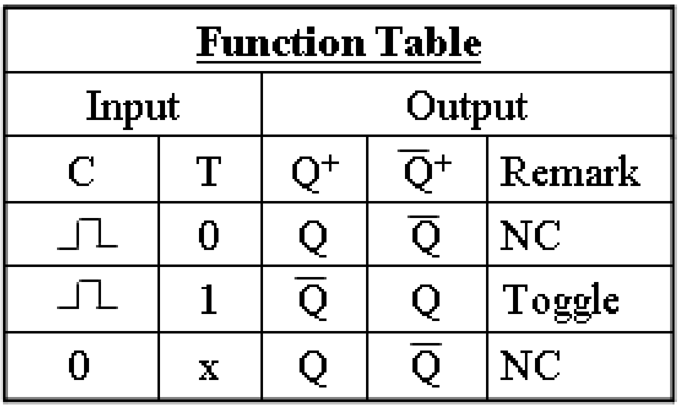
Function Table

Input			Output		
C	J	K	Q^+	$\overline{Q}^+$	Remark
⎍	0	0	Q	$\overline{Q}$	NC
⎍	0	1	0	1	Reset
⎍	1	0	1	0	Set
⎍	1	1	$\overline{Q}$	Q	Toggle
0	x	x	Q	$\overline{Q}$	NC

In SR FF, S=R=1 condition is not allowed.

- JK FF is modified version of SR FF.
- Due to feedback from output to input AND Gate J=K=1 is toggle condition for JK FF.
- The output is complement of the previous output.
- This condition is used in counters.
- T-FF is modified version of JK FF in which T=J=K=1.

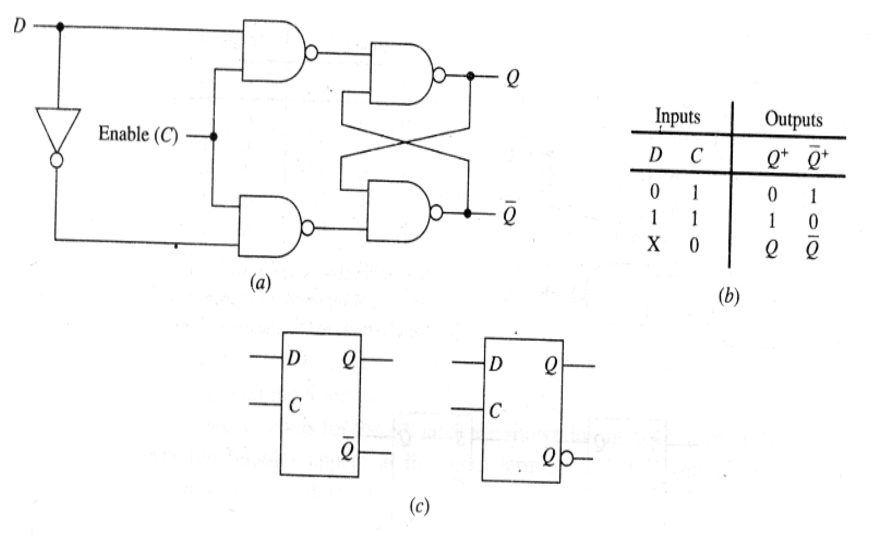




Function Table

Input		Output		
C	T	Q^+	$\overline{Q}^+$	Remark
⎍	0	Q	$\overline{Q}$	NC
⎍	1	$\overline{Q}$	Q	Toggle
0	x	Q	$\overline{Q}$	NC

Gated D Latch :



(a)

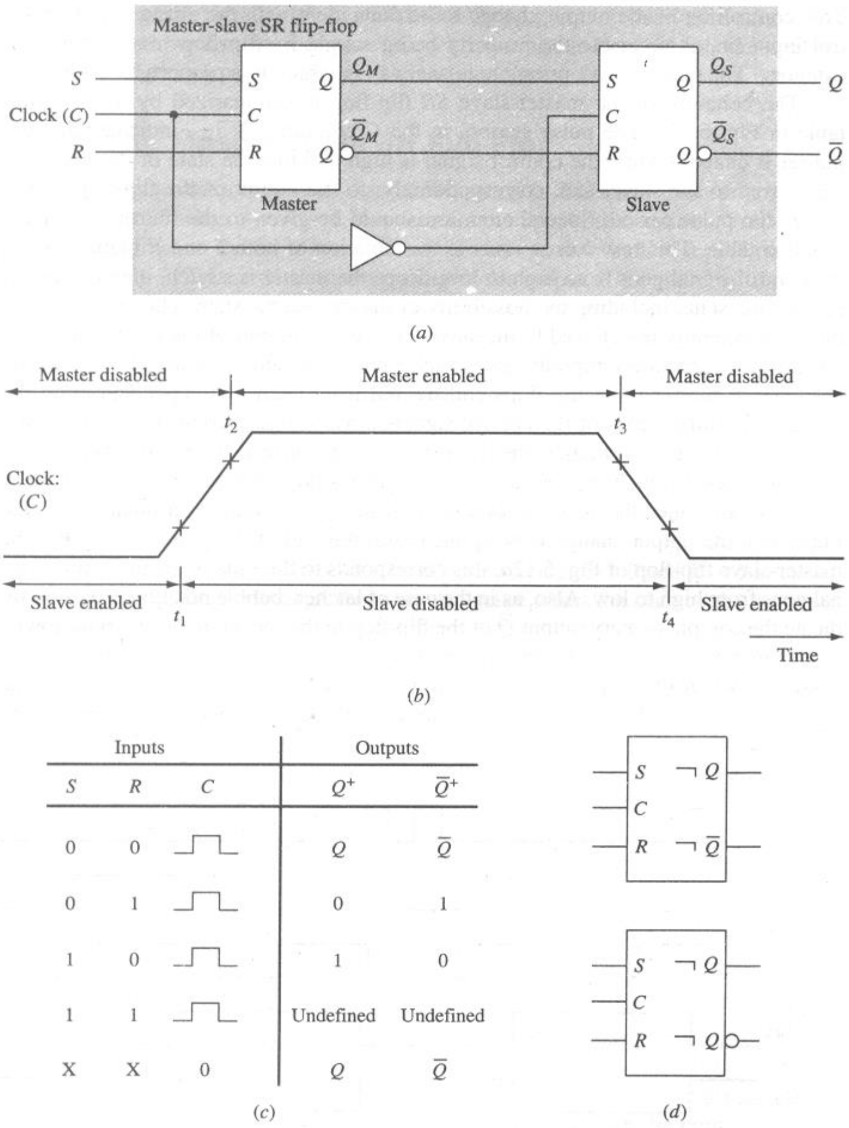
Inputs		Outputs	
D	C	Q^+	$\bar{Q}^+$
0	1	0	1
1	1	1	0
X	0	Q	$\bar{Q}$

(b)

(c)

- D Flip-Flop is Data Flip-Flop.
- D Flip-Flop stores 1 or 0.
- R input is complement of S.
- Only one D input is present.
- D Flip-Flop is a storage device used in register.

Master slave SR Flip-Flop



(a)

(b)

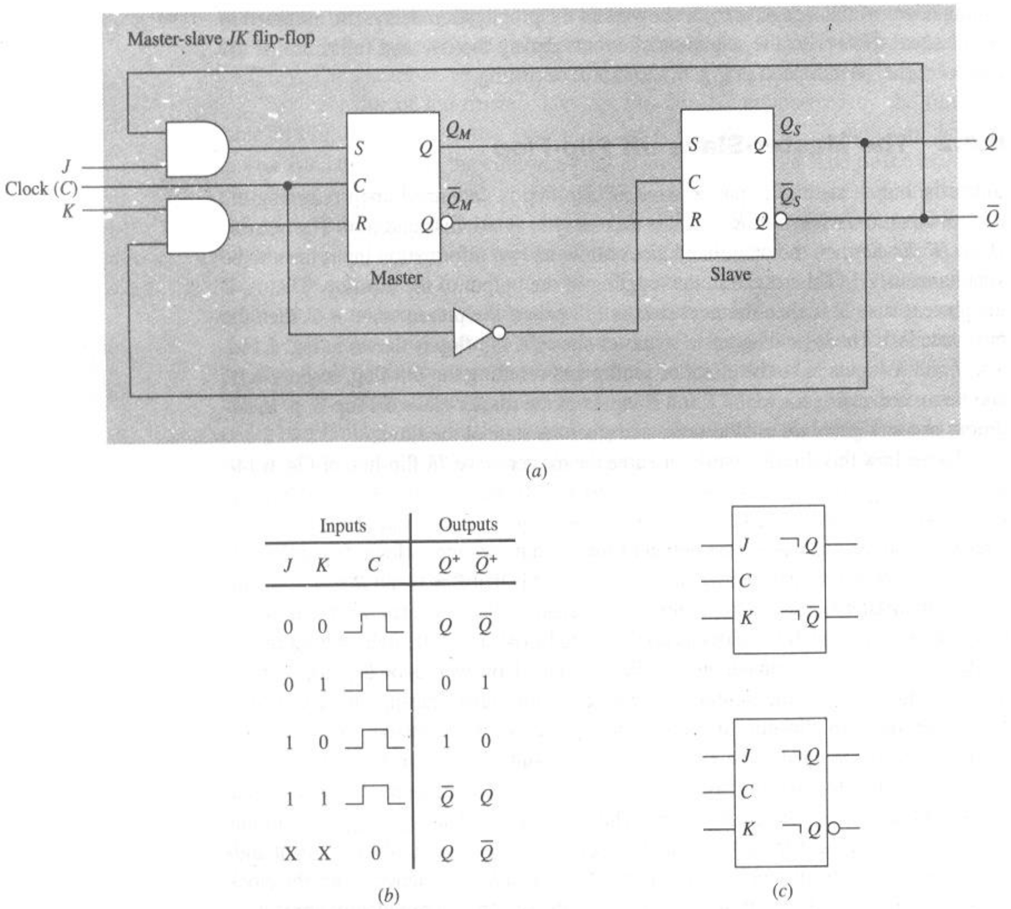
Inputs			Outputs	
S	R	C	Q^+	$\overline{Q}^+$
0	0	⎍	Q	$\overline{Q}$
0	1	⎍	0	1
1	0	⎍	1	0
1	1	⎍	Undefined	Undefined
X	X	0	Q	$\overline{Q}$

(c)

(d)

- Two SR Flip-Flop, 1st is Master and 2nd is slave.
- Master Flip-Flop is positive edge triggered.
- Slave Flip-Flop is negative edge triggered.
- Slave follows master output.
- The output is delayed.

Master slave JK Flip-Flop

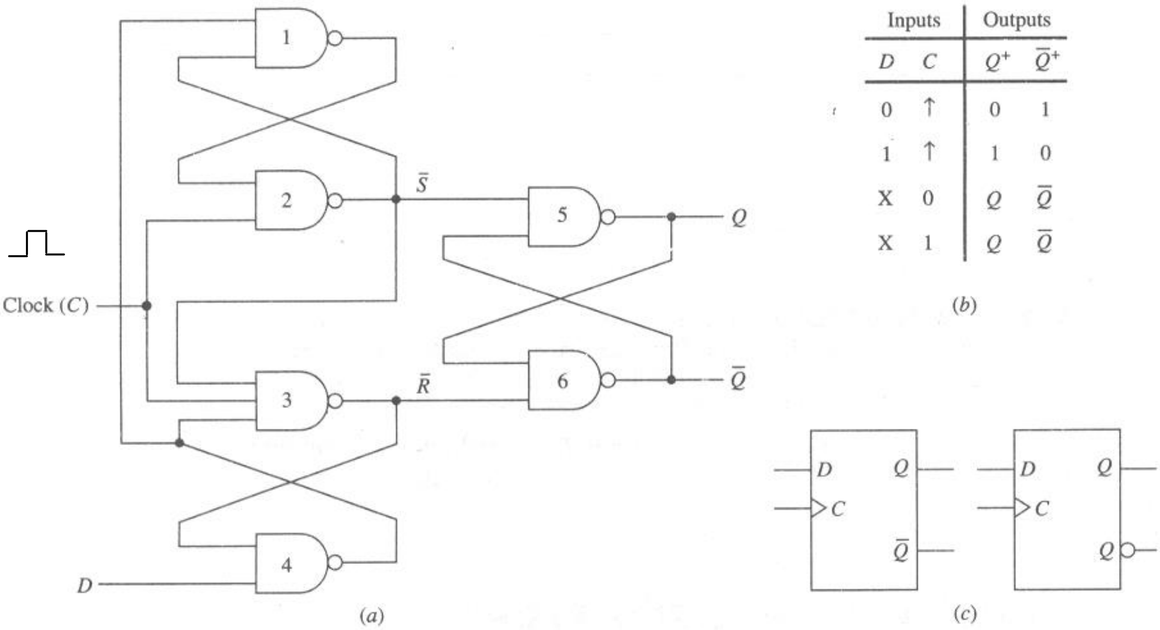


Inputs			Outputs	
J	K	C	Q^+	$\bar{Q}^+$
0	0	⎍	Q	$\bar{Q}$
0	1	⎍	0	1
1	0	⎍	1	0
1	1	⎍	$\bar{Q}$	Q
X	X	0	Q	$\bar{Q}$

(b)

- In SR Flip-Flop the input combination S=R=1 is not allowed.
- JK FF is modified version of SR FF.
- Due to feedback from slave FF output to master, J=K=1 is allowed.
- J=K=1, toggle, action in FF.
- This finds application in counter.

Positive Edge Triggered D Flip-Flop



Inputs		Outputs	
D	C	Q^+	$\bar{Q}^+$
0	↑	0	1
1	↑	1	0
X	0	Q	$\bar{Q}$
X	1	Q	$\bar{Q}$

(b)

- When C=0, the output of AND Gate 2 & 3 is equal to 1.

  $\bar{S} = \bar{R} = 1,$ No Change of State

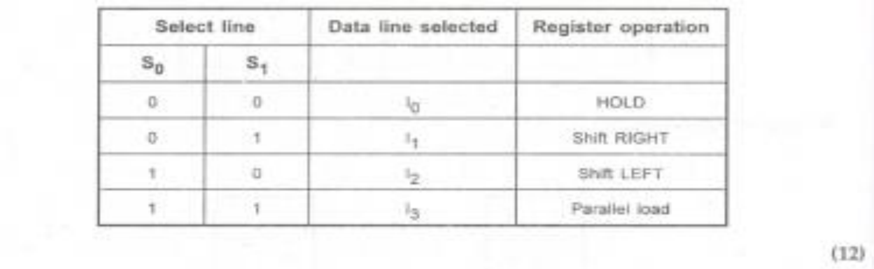
- If C=1, D=1, the output of AND Gate 2 is 0 and 3 is 1.

  $\bar{S} = 0, \bar{R} = 1,\ Q = 1$ and $\bar{Q} = 0$

Recommended question and answer –unit-5

Jan 2009

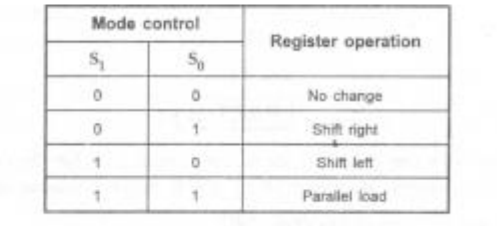
Q.6 a) *Design a 4-bit universal shift register using positive edge triggered* D *flip-flops* to *operate as shown in the table* .



Select line		Data line selected	Register operation
S_0	S_1		
0	0	I_0	HOLD
0	1	I_1	Shift RIGHT
1	0	I_2	Shift LEFT
1	1	I_3	Parallel load

(12)

Ans. : Universal shift register: A register capable of shifting in one direction only is a unidirectional shift register. A register capable of shifting in both directions is a bidirectional shift register. If the register has both shifts (right shift and left shift) and parallel load capabilities, it is referred to as Universal shift register. The Fig. 5 (See next page) shows the 4-bit universal shift register. It has all the capabilities listed above. It consists of four flip-flops and four multiplexers. The four multiplexers have two common selection inputs Sl and So' and they select appropriate input for D flip-flop. The Table 1 shows the register operation depending on the selection inputs of multiplexers. When S1 S0 = 00, input 0 is selected and the present value of the register is applied to the D inputs of the flip-flops. This results r, v change in the register value. When S1S0 = 01, input 1 is selected and circuit connections are such that it operates as a right shift register. When S1S0 = 10, input 2 is selected and circuit connections are such that it operates as a left shift register. Finally, when S1S0= 11, the binary information on ~e parallel input lines is transferred into the register simultaneously and it is a parallel load operation.



Mode control		Register operation
S_1	S_0	
0	0	No change
0	1	Shift right
1	0	Shift left
1	1	Parallel load

Table 1 Mode control and register operation

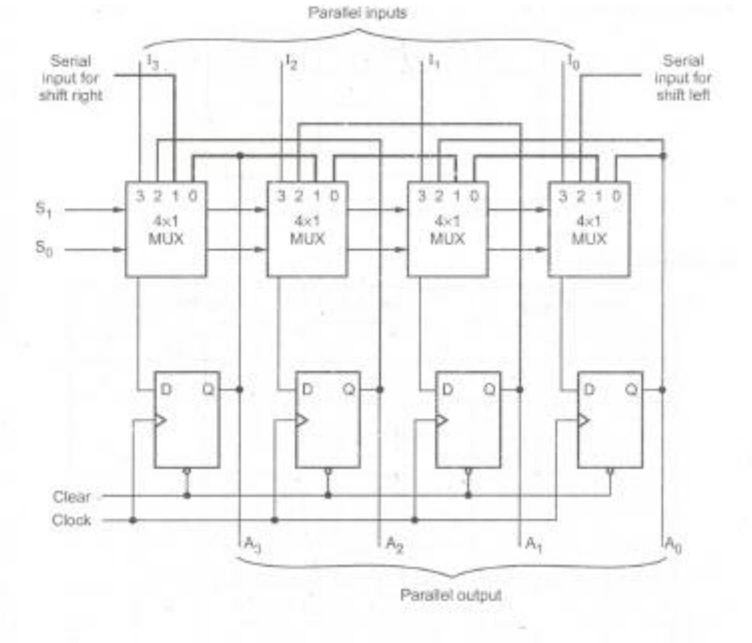


Fig. 5 4-Bit universal shift register

Jan -2008

5. What is the significance of edge triggering ? Explain the working of edge triggered D-flip-flop and T-flip-flop with their functional table. (6)

Ans. : For the edge triggered FF, it is necessary to apply the clock signal in the form of sharp positive and negative spikes instead of in the form of pulse train. These spikes can be derived from the rectangular clock pulses with the help of a passive differentiator as shown in Fig. 14. Edge triggered D Flip-Flop Fig. 15 shows the edge triggered DFF. It consists of gated 0 latch and a differentiator circuit. The clock pulses are applied to the circuit through a differentiator formed by R1C and a rectifier circuit consisting of diode 0 and R2.The NAND gates 1 through 5 form a D latch. The differentiator converts the clock pulse! into positive and negative spikes as shown in the Fig. 16 and the combination of D and R2 will allow only the positive spikes to pass through blocking the negative spikes.

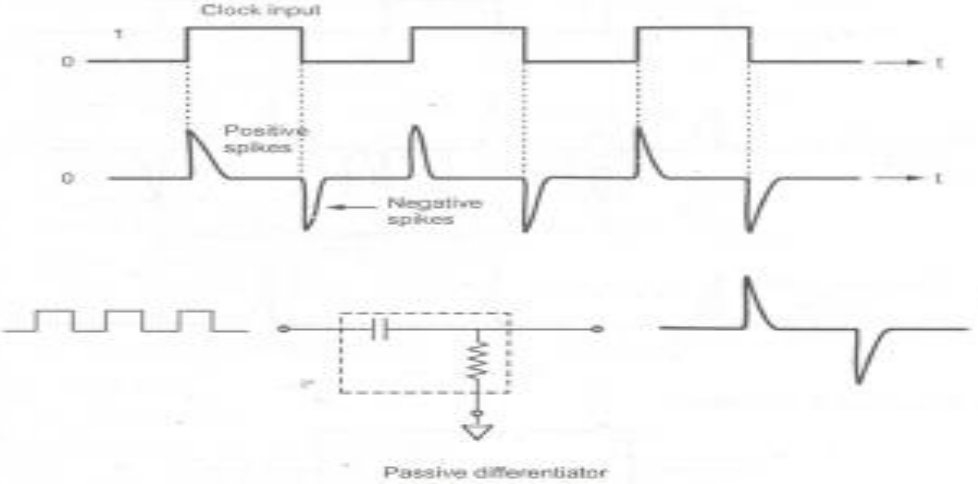


Fig. 14 Use of differentiator to obtain sharp edges

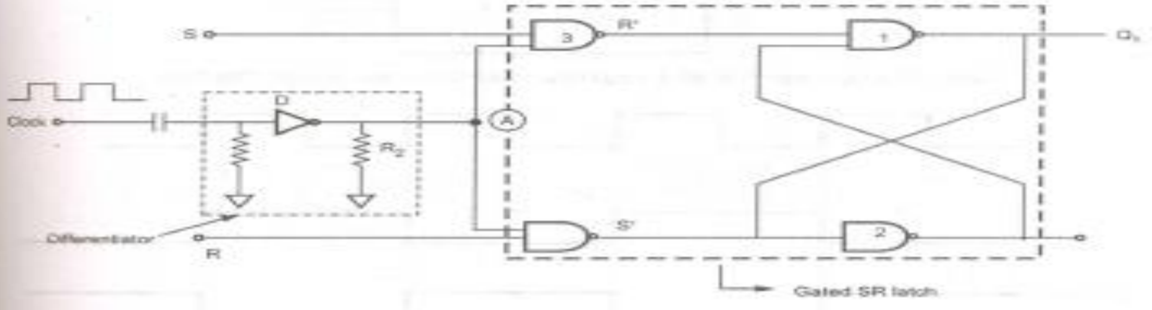


Fig. 15

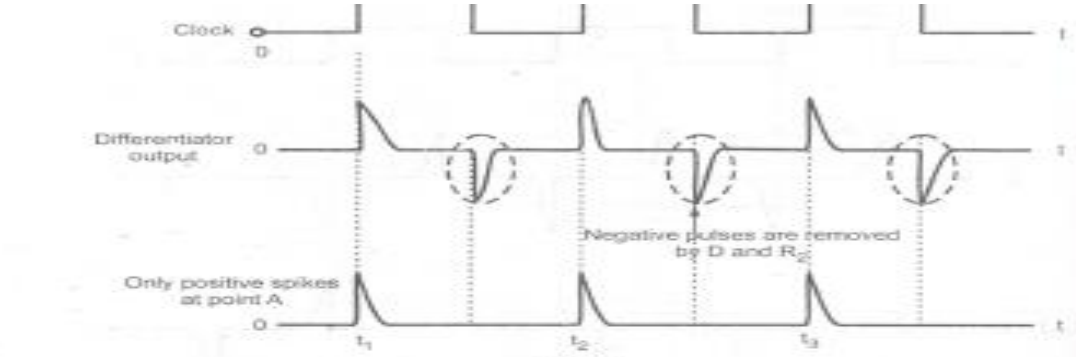

Clock
0
t
Differentiator output
0
t
Negative pulses are removed by D and R₂
Only positive spikes at point A
0
t
t₁
t₂
t₃

Fig. 16

Edge triggered T flip-flop

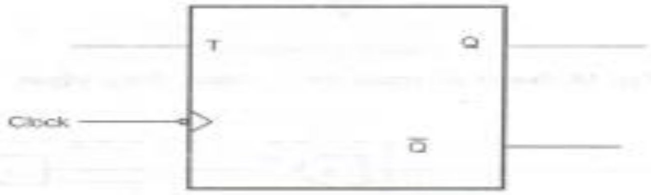
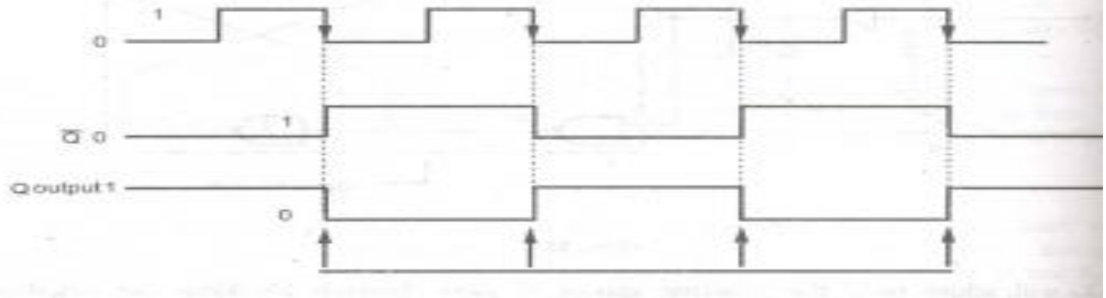


Fig. 17 Logic symbol of a negative edge triggered toggle flip-flop



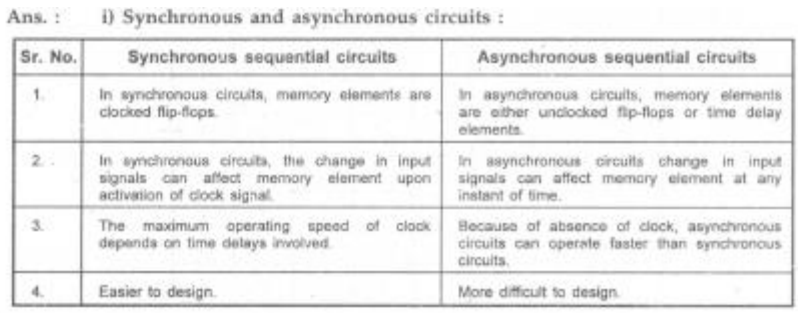
Toggling takes place at every falling edge of clock

Aug 2009

Q.5 a) Clearly distinguish between

i) Synchronous and asynchronous circuits.

ii) Combinational and sequential circuits.



Ans. : i) Synchronous and asynchronous circuits :

Sr. No.	Synchronous sequential circuits	Asynchronous sequential circuits
1.	In synchronous circuits, memory elements are clocked flip-flops.	In asynchronous circuits, memory elements are either unclocked flip-flops or time delay elements.
2.	In synchronous circuits, the change in input signals can affect memory element upon activation of clock signal.	In asynchronous circuits change in input signals can affect memory element at any instant of time.
3.	The maximum operating speed of clock depends on time delays involved.	Because of absence of clock, asynchronous circuits can operate faster than synchronous circuits.
4.	Easier to design.	More difficult to design.

b) *Explain the working of 4-bit asynchronous counter.*

Ans. : **4-bit asynchronous counter :**

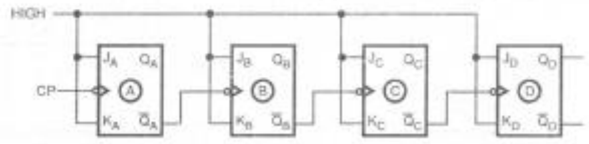


Fig. 10 4-bit asynchronous down counter

1) 4 flip-flops are employed to create a 4-bit asynchronous counter as shown.

2) The clock signal is connected to the clock input of only first stage flip-flop.

3) Because of the inherent propagation delay time through a flip-flop, two flip-flops never trigger simultaneously. Thus, it works in an asynchronous operation.

4) Output of the first flip-flop triggers the second flip-flop and so on.

S) At the output of flip-flops, we get the counted value of the counter.

c) *Explain Johnson counter with its circuit diagram and timing diagram.* (8)

Ans. : **4-bit Johnson counter :**

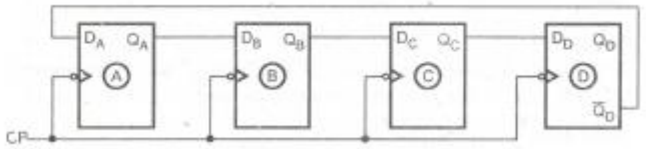


Fig. 11 Four-bit Johnson counter

1) Initially, the register is cleared.

:. all the outputs QA' *QSI QCI* Qo are zero.

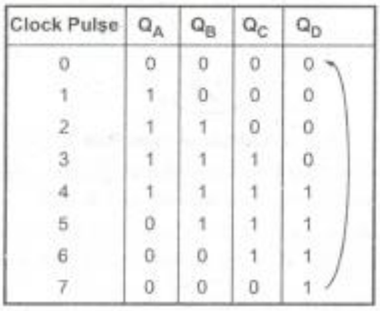
2) The complement of Q 0 is 1 which is connected back to the D input of first stage.

:. DA is, 1.

:. The outPut becomes QA = 1, Qs = 0, Qc = a and Qo = O.

3) The next clock pulse produces Q A = 1, Q B = 1, Q C = a and Q 0 = O.

The sequence is' given as :



Clock Pulse	Q_A	Q_B	Q_C	Q_D
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0
4	1	1	1	1
5	0	1	1	1
6	0	0	1	1
7	0	0	0	1

Table 3 Four-bit Johnson sequence

Aug-2008

b) *Fig.* 2 *shows a BCD counter that produces a 4-bit cutput representing*

code for the number of pulses that have been applied to the counter i;

example, after four pulses have occurred, the counter ouq

(ABeD) = (OlOOh = (04)10' The counter resets to 0000 on the tenth]

starts counting over again. Design the logic circuit that produces a HIGl

Whenever the count is 2, 3 *or* 9. *Use K - mapping and take advai*

"don't care" conditions. Implement the logic circuit using NAND gates.

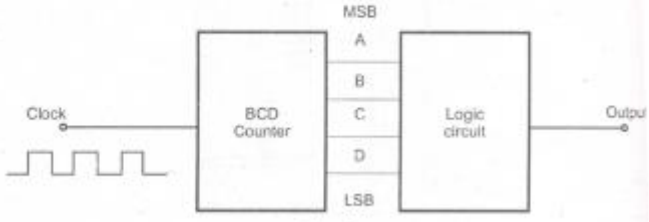
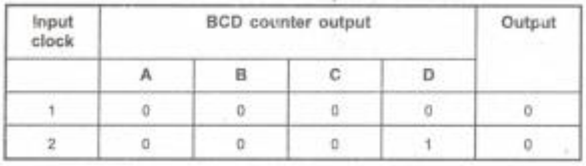
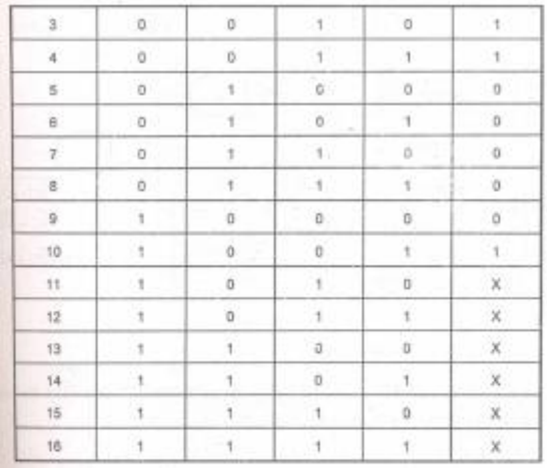


Fig. 2

Ans. :



Input clock	BCD counter output				Output
	A	B	C	D	
1	0	0	0	0	0
2	0	0	0	1	0



3	0	0	1	0	1
4	0	0	1	1	1
5	0	1	0	0	0
6	0	1	0	1	0
7	0	1	1	0	0
8	0	1	1	1	0
9	1	0	0	0	0
10	1	0	0	1	1
11	1	0	1	0	X
12	1	0	1	1	X
13	1	1	0	0	X
14	1	1	0	1	X
15	1	1	1	0	X
16	1	1	1	1	X

BCD numbers are present in between 0 to 9 so from 9 to 15 we take don't care condition and when 2, 3, or 9 detected we take output as 1.

Output = $AD + \bar{B}C$

K-map simplification

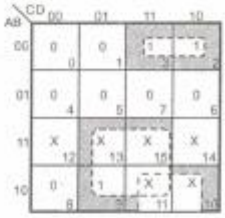
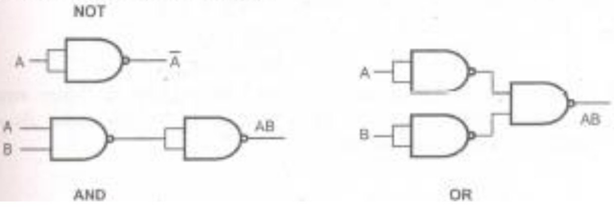


Fig. 2 (a)

Implementation using NAND gate :



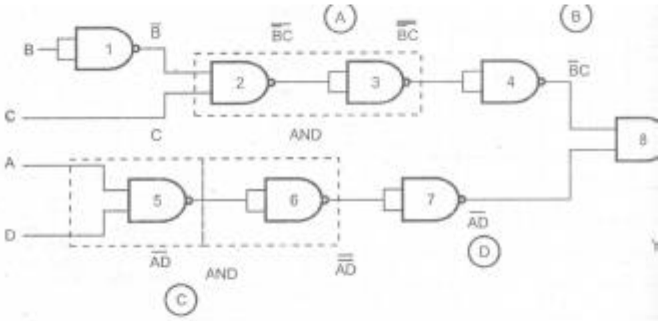


Fig. 2 (c)

We get same data or signal at point Ⓐ and Ⓑ as well as point Ⓒ and (I
remove gate 2, 3 for at point Ⓐ, Ⓑ and 6 , 7 , for point Ⓒ, Ⓓ.

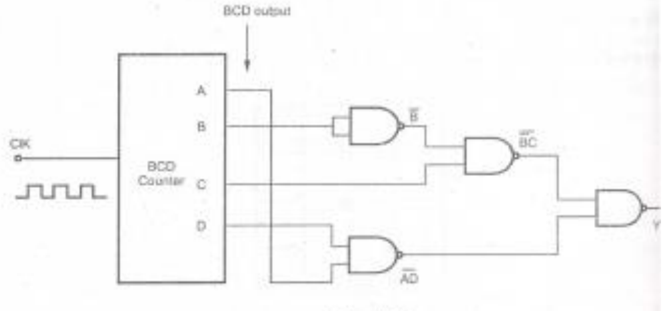


Fig. 2 (d)

Output will be the 1 when output of BCD $(0010)_2 = (2)_{10}$

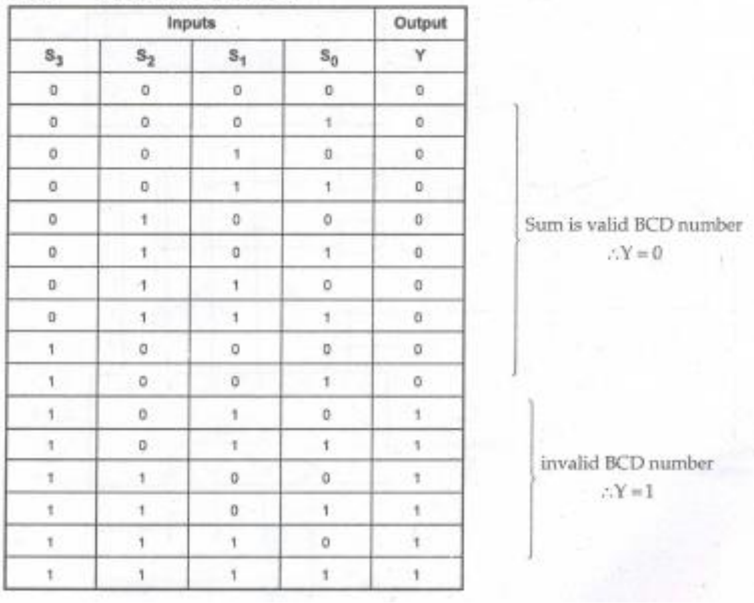
$(0011)_2 = (3)_{10}$

$(0001) = (9)_{10}$

Aug -2008

Q.4 a) *Design a* 4 - *bit BCD adder circuit using* 7483 *IC chip, with self correcting cirucit i.e., .a provision has. to be made in the circuit, in case if the sum of the BCD number exceeds* 9. **(12)**

Ans. : Truth table is given as ,

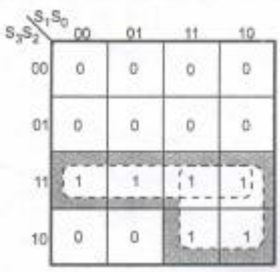


Inputs				Output
S_3	S_2	S_1	S_0	Y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Sum is valid BCD number $\therefore Y = 0$

invalid BCD number $\therefore Y = 1$

K-map simplification



$\therefore Y = S_3S_2 + S_3S_1$

Circuit diagram for 4-bit BCD adder :

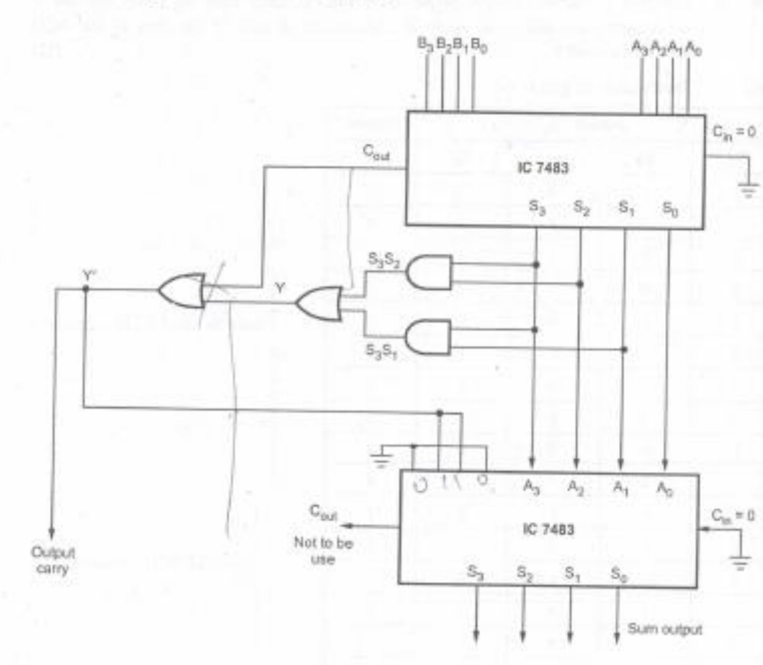
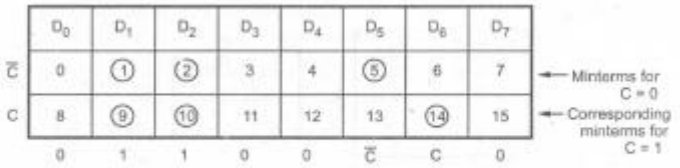


Fig. 7 (b)

Aug-2007

b) *Implement the following Boolean function using 8:1 MUX :*
$F(A,B,C,D) = \Sigma m(1,2,5,9,10,14)$

Sol. : Implementation Table :



	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	
$\overline{C}$	0	①	②	3	4	⑤	6	7	← Minterms for C = 0
C	8	⑨	⑩	11	12	13	⑭	15	← Corresponding minterms for C = 1
	0	1	1	0	0	$\overline{C}$	C	0	

Fig. 10

Multiplexer Implementation :

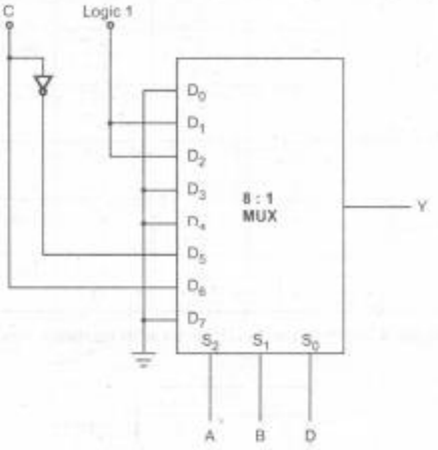


Fig. 11