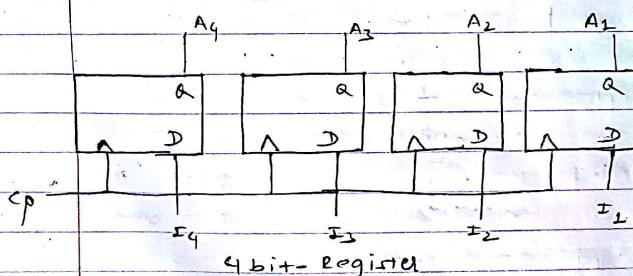


Chap 8 → Register, Counters and Memory Unit.

\* Register      pulse duration  $\rightarrow$  latch  
                  pulse transition  $\rightarrow$  flip flop

Register is a group of flip flops suitable for storing binary information. It is also known as Group of binary storage cell. Each cell is capable of storing one bit of information.

For n-bit registers, there are group of 'n' flip flops and stores binary information of n-bits. Registers also contain combinational gates which controls and predefines how new information is transferred into the register.



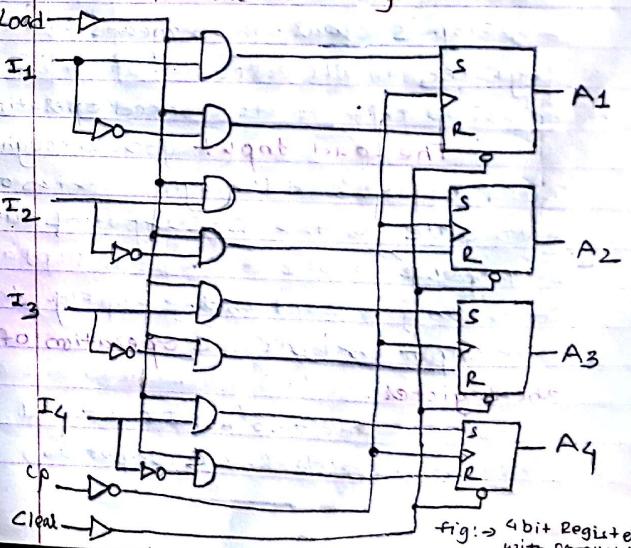
Above figure shows a 4-bit register constructed using 4 D-Flip Flops. The common clock input triggers all the flip flop on the rising edge of the

pulse and the binary data available at the four inputs are transferred into the four bit register.

#### \* Register with parallel Load

The process of transferring information into a register is called Loading.

If all the bits of the register are loaded simultaneously with a single clock pulse, it is called Parallel Loading.



A 4-bit Register with a Load Control input using R-S Flip Flop is shown above.

The control pulse input of the Register receives continuous synchronized pulse which is applied to all flip flops since control pulse is inverted

it will be triggered by the negative edge of the pulse.

Similarly clear input goes to each flip flop through a non-inverting buffer and is also known as special terminal. When this terminal is '0' the flip flop is clear which means that the register are all cleared to '0'. It is required prior to its clocked operation.

The load input goes through a buffer gate and through a series of AND Gate to the R&S input of each flip flop. As the clock pulse ( $C_P$ ) are continuously present in the flip flop, the load input controls the operation of the register.

If the load input is '0' both R & S input is 0

and no change of state occurs with the clock pulse.

When the load control or input is equal to 1, Inputs  $I_1$  to  $I_4$  specifies what information is loaded into the register.

For each input  $I=1$ , the corresponding flip flop input  $S=1, R=0$  the flip flop is set.

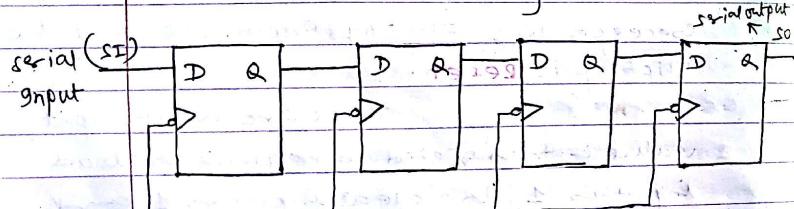
Again for each input  $I=0$ , the corresponding flip flop output  $S=0, R=1$ , the flip flop is reset.

Therefore while input value is transferred to register the load input is 1, the clear input is 1 and a clock pulse goes from 1 to 0. This type of transfer where all bits of register are loaded simultaneously is called parallel load transfer.

## \* Shift Registers

Registers capable of shifting binary information either to right or to the left is called shift Registers.

- Consist of chain of Flipflops connected in cascade. Output of one flipflop is connected to the input of the next flipflop and receives common clock pulse which causes the shift from one stage to the next.



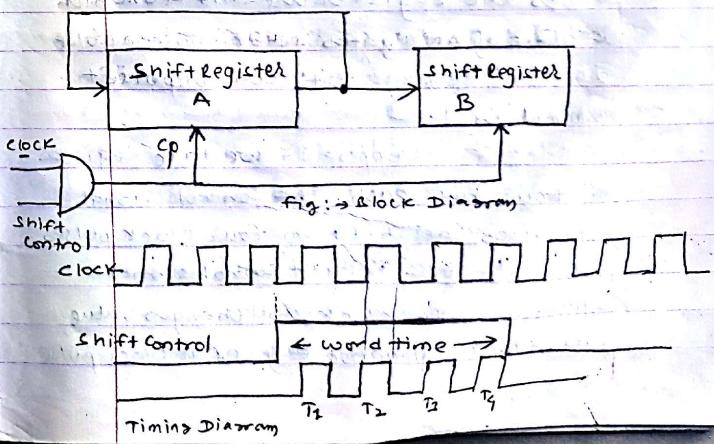
clock pulse (CP) of 1 ms is used here  
fig: Shift Register

Above figure

shows the simple shift Register using D-Flipflop. Here the Q output of the Flipflop is connected to the D Flipflop at its right. Each clock pulse will shift the content of the register to the right. The content is shifted

to the right in every clock pulse during the negative edge of the pulse transition (P.T.). The serial output (SO) is taken from the output of the rightmost FlipFlop.

\* Serial Transfer of shift Register: → When the information is transferred and manipulated one bit at a time is called serial transfer. Here, the content of one register is transferred to another by shifting the bit from one register to the another that is source register to Destination Register.



Above figure shows the block diagram of serial transfer of information from Register A to Register B. The serial output (S.O) of Register A goes to the serial input of Register B. To prevent the loss of information stored the serial output (S.O) of Register A is fed back to serial input (S.I) and the content of B is shifted out through its serial output and is lost if there is no third shift register.

Here, the shift control determines when and how many times the register are shifted. The clock pulses are passed to control pulse terminal of Register when the shift control is 1.

Suppose if we take registers of four bits each. The control signal supervises the shift to four clock pulses. The shift control signal synchronizes the clock and changes value just after the negative edge of a clock pulse.

The output of AND gate to the control pulse terminal produces four pulses  $T_1, T_2, T_3$  and  $T_4$  as shown in above figure.

Example: If binary content of A = 1011 and B = 0010, the serial transfer is shown below.

Clock pulse	Shift register A	Shift register B	Serial output of B
Initial value	1 0 1 1	0 0 1 0	
After $T_1$	1 1 0 1	1 0 0 1	0
After $T_2$	1 1 1 0	1 1 0 0	1
After $T_3$	0 1 1 1	0 1 1 0	0
After $T_4$	1 0 1 1	1 0 1 1	0

Serial Mode	Parallel Mode
1) In serial mode, registers have a single serial input and a single serial output	All bits are transferred simultaneously during one clock pulse

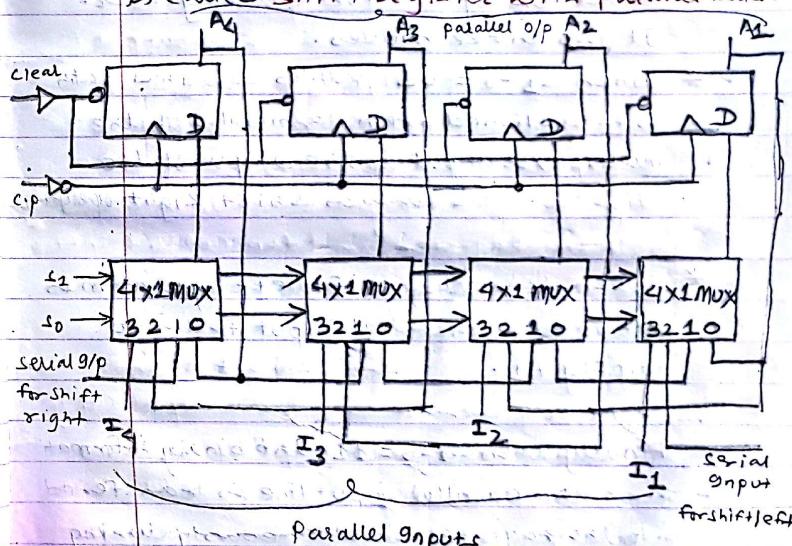
Most of the computers operate in parallel mode because of fast mode of operation.

## \* Bidirectional shift Register with parallel Load

Shift Registers can be used for converting serial data to parallel data and vice versa. The most general shift registers have capabilities as listed below.

1. Clear Control to clear the register to zero.
2. Control pulse Input for clock pulse to synchronize all operation.
3. A shift right control to enable the shift right operation and the serial input and output lines associated with the shift right.
4. A shift left control to enable the shift left operation and serial input and output lines associated with the shift left.
5. A parallel load control to enable parallel transfer and the m-input lines associated with the parallel transfer.
6. m-parallel output lines.
7. A control state that leaves the information in the register unchanged even though clock pulses are continuously applied.

Registers capable of shifting both right and left is called a Bidirectional shift Register. Therefore register that has both shift and parallel load capabilities is called shift register with parallel load.



A bidirectional shift Register with parallel Load is capable of performing three operation: Shift Left, Shift Right and parallel Load.

Above Block Diagram shows that it consist of four D-FlipFlops

four multiplexer which are part of Register. The four multiplexer has two common selection variable  $s_1 s_0$ . Selection variable ( $s_1 s_0$ ) controls the mode of operation of Register. It has three modes.

\* When  $s_1 s_0 = 01$ , path is established from terminal 1 of the multiplexer input to the D input of the FlipFlop. This causes shift Right operation.

\* When  $s_1 s_0 = 10$ , shift left operation is activated with serial input fed into FlipFlop A1.

\* Finally when  $s_1 s_0 = 11$ , the binary information on the parallel input line is transferred into the register simultaneously during the next clock pulse.

#### \* Serial Addition:

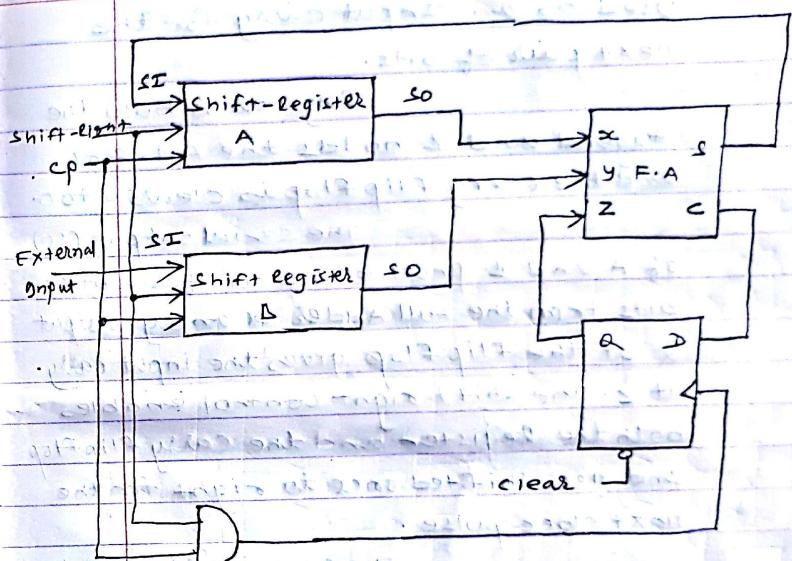


fig: → serial Adder

Above figure shows the serial Adder which adds two binary number serially stored in two shift registers. Bits are added one pair at a time sequentially through a single Full Adder (F.A.).

The carry out of a Full Adder is transferred to a D Flip Flop and the output of this Flip Flop is then

used as an Input carry for the next pair of bits.

Registers A hold the Augend and B hold the Addend and the carry flip flop is cleared too. The serial output (so)

of A and B provide a pair of significant bits from the Full Adder x and y. Output Q of the flip flop gives the input carry at z. The shift right control enables both the Register and the carry flip flop and gets shifted once to right for the next clock pulse.

The sumbit ( $s$ ) enters the shift register A and the output carry is transferred into flip flop Q. Now for each succeeding clock pulse a new sumbit is transferred to A and a new carry to Q and both registers are shifted to right. This process continues until the shift right control is disabled.

For new number to be added to A the content of Register A, new number must be first transferred serially into Register B.

#### \* Difference between parallel Adder and serial Adder

##### parallel Adder

1. parallel Adder uses Registers with parallel load capabilities

##### serial Adder

1. serial Adder uses Shift registers

2. In parallel Adder the number of Full Adder is equal to the no of bits in binary number

2. serial Adder requires only one Full Adder circuit and a carry Flip Flop.

3. parallel Adder is purely combinational circuit

3. serial Adder is a sequential circuit

4. parallel Adder requires more equipment is expensive

4. serial Adder requires less equipment and is cheap

## \* Types of shift Register

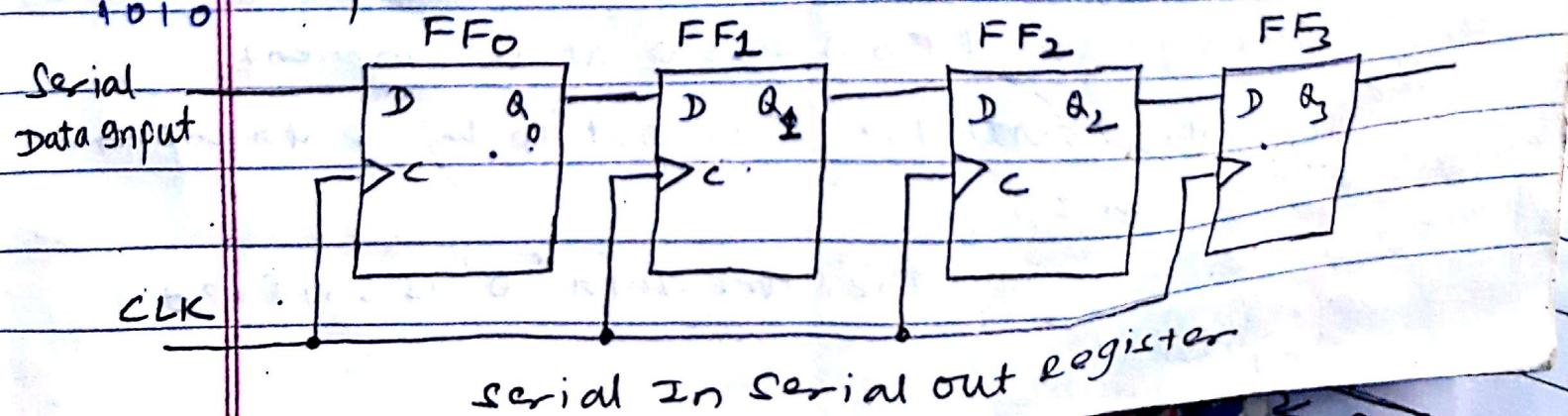
Page \_\_\_\_\_

In shift register the bits of binary number (data) can be shifted from one place to another by two ways.

- shifting data 1 bit at a time in serial fashion beginning with the MSB or LSB. This technique is called serial shifting.
- shifting all the data bits simultaneously is referred as parallel shifting.

On this basis shift register are of different types. They are

- ✓ 1. Serial In - Serial Out Shift Register (SISO):  
Serial In Serial Out shift register (SISO) accepts data serially with one bit at a time on a single line. It produces the stored information on its output in serial form.



Above figure shows a 4-bit Register implemented with D-FlipFlop with four stages. This register can store upto four bits of data.

Let us take, data input 1010 into the Register begining with the LSB '0'. The Register is initially clear.

\* When input '0' is put on serial  $D=0$  Data input  $FF_0$ ,  $FF_0$  is reset and stores the 0 bit for the first clock pulse.

\* When second bit '1' is put to the Data input,  $D=1$  in  $FF_0$ .

When the second pulse occurs, the data input '0' is shifted to the  $FF_1$  and '1' is stored in  $FF_0$ .

\* When third bit '0' is put to the Data input,  $D=0$  in  $FF_0$ . The stored data '1' is

shifted to  $FF_1$  and '0' to  $FF_2$  for the third clock pulse.  $FF_0$  stores '0' at this moment.

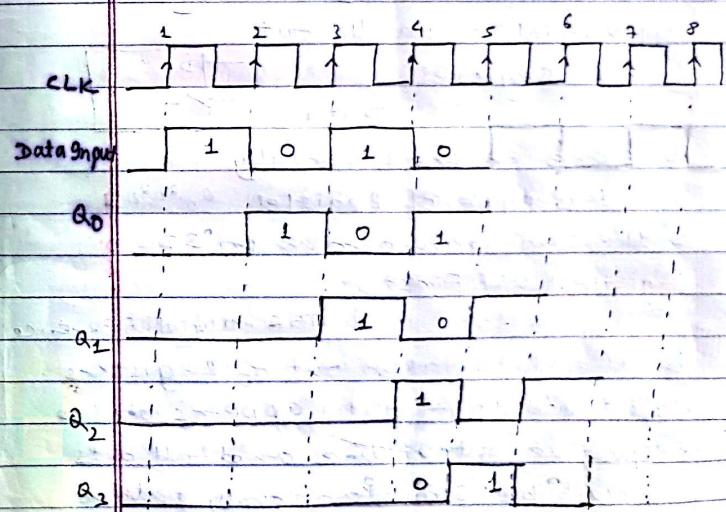
\* When fourth bit '1' is put to the Data input  $D=1$  in  $FF_0$ .

The stored data '0' is shifted.

to  $FF_1$ , stored data '1' to  $FF_2$  and '0' to  $FF_3$  for the fourth pulse.

Therefore completes the serial entry of the four bits into the shift register.

The timing diagram is shown below-



## 2. serial in parallel out

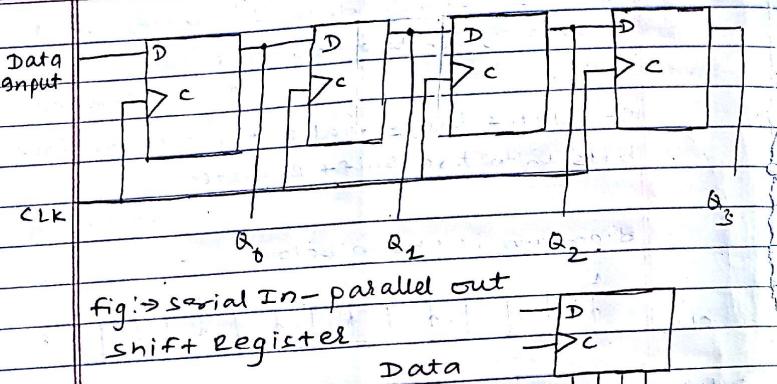
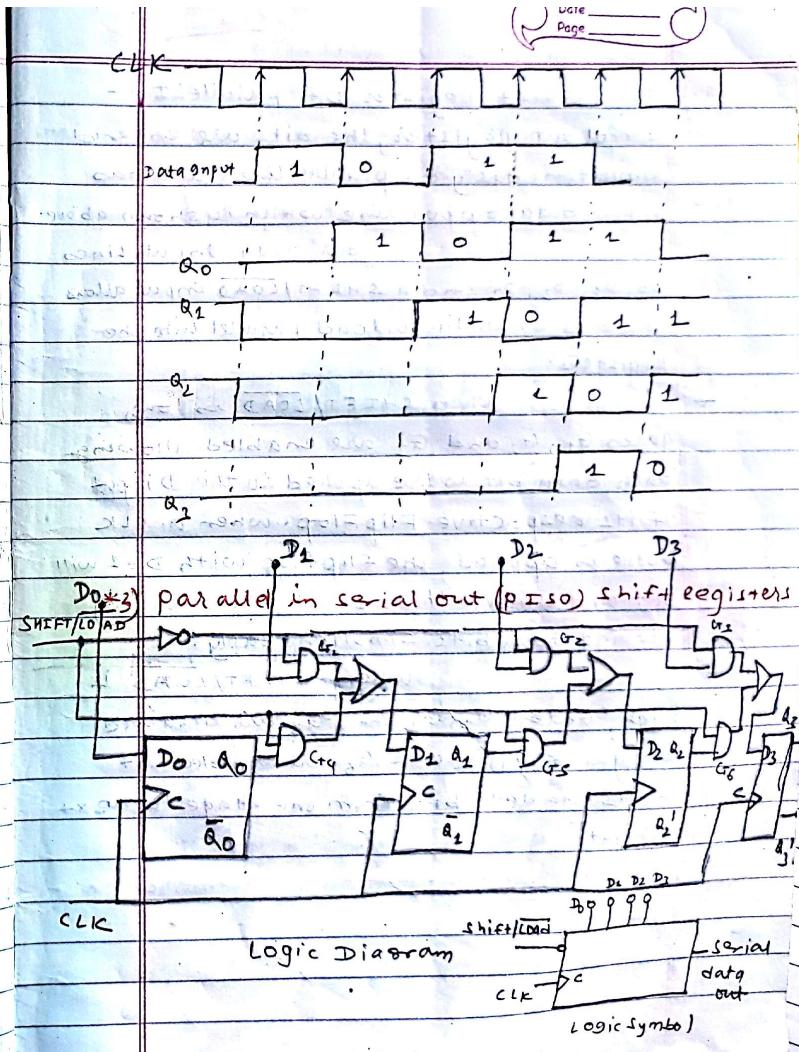


fig: serial In - parallel out  
shift register

bits are entered serially  
into this type of register  
in the same manner as in SISO  
Register. (LSB First).

The only difference  
is the data taken out of registers.  
Here, the output data appears on its  
respective output line and all bits  
are available simultaneously rather  
than bit by bit as serial output.



For a shift register of parallel in - serial out register, the bits are entered simultaneously on parallel lines and has single data output line which is shown above.

Four Data Input Lines  $D_0, D_1, D_2, D_3$  and a SHIFT/LOAD input allows four bits of data to load parallel into the registers.

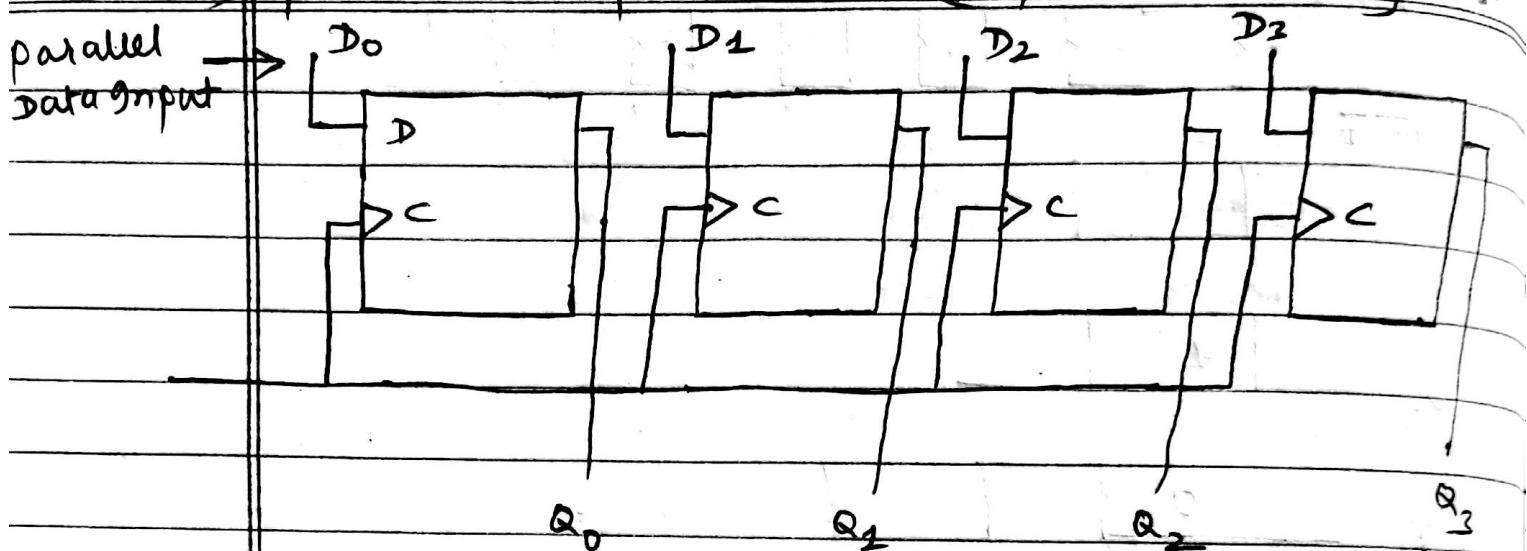
When SHIFT/LOAD is low, gates  $G_1, G_2$  and  $G_3$  are enabled. Each data bit is applied to the D input of its respective FlipFlop. When a CLK pulse is applied, the flipflop with  $D=1$  will set and those with  $D=0$  will reset, thereby storing four bits simultaneously.

When SHIFT/LOAD is high, Gates  $G_1, G_2$  and  $G_3$  are disabled and gates  $G_4, G_5$  and  $G_6$  are enabled. It shifts the data bit from one stage to next to right.

	1 ↑	2 ↑	3 ↑	4 ↑	5 ↑	6 ↑	
SHIFT/LOAD							
$Q_0$	1						
$Q_1$	0						
$Q_2$	1						
$Q_3$	0	1	0	1	0		

on clock pulse-1  
(CLK-1), the parallel data (1010) are loaded into the register making  $Q_3$  '0'. on CLK-2 the '1' from  $Q_2$  is shifted into  $Q_3$ . on the clock pulse CLK-3, 0 is shifted into  $Q_3$ . on CLK-4, the next 1 is shifted into  $Q_3$ . For CLK-5, CLK-6 all data bits will be shifted out and only 0's will remain in the register because no new data have been entered.

#### 4) parallel in parallel out (PIPO) shift register

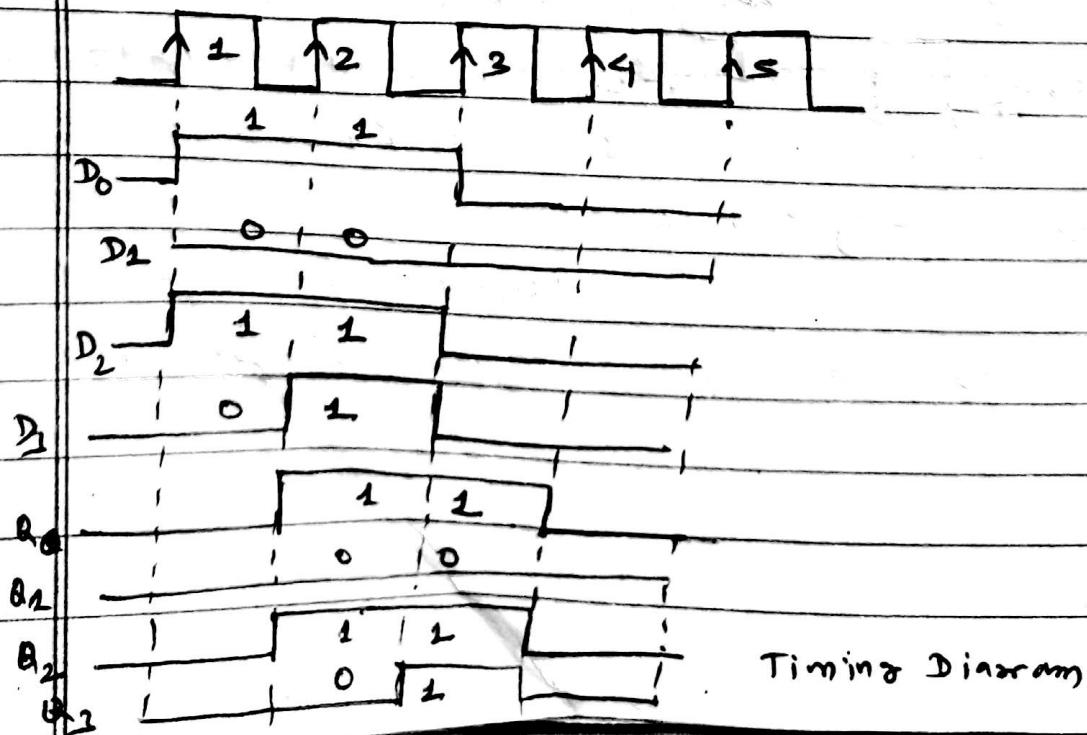


parallel data output

fig:  $\Rightarrow$  PIPO shift register

This kind of

shift register takes data from the parallel input & shifts it to the corresponding output. This type of register is simply used to store data and is sometimes called data register.



## \* Counters

A Counter is a sequential circuit consist of a set of flip flops connected in a suitable manner to count the number of occurrence of an event. It is useful for generating timing sequence to control operation in digital system. It goes through a prescribed sequence of state upon the application of clock pulse.

- pulse counting
- frequency division
- time measurement & control
- timing operation.

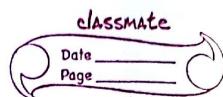
Counters are classified

into two types:

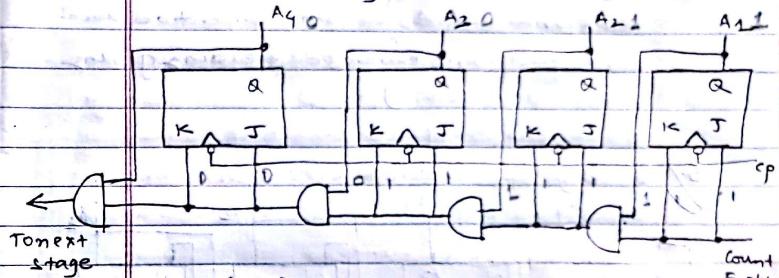
1. Synchronous Counters:

In Synchronous counter the clock input is connected to all of the flip flops. All flip flops are clocked (triggered) at same time.

a. 4-bit Binary Synchronous Counter:



Counter that follows the binary sequence is called Binary counter.



4-bit synchronous binary counter

Above

Figure shows 4-bit binary synchronous counter. It consists of 4 flip flop and counts in binary form.

The clock terminal of all flip flops are connected to common clock source. The first stage A<sub>1</sub> has its J and K equal to 1 if the counter is enabled. The other input J and K becomes equal to 1 if all the lower order bits are equal to 1 and the count is enabled. The AND gate chain generates the required logic for input J and K in each stage.

For Example: → if the present

4 bit counter is  $A_4 A_3 A_2 A_1 = 0011$ , the next count will be 0100. Here,  $A_3$  is complemented,  $A_2$  is complemented and  $A_1$  is not complemented giving 0100. (next count)

b) Binary Up-Down Counter

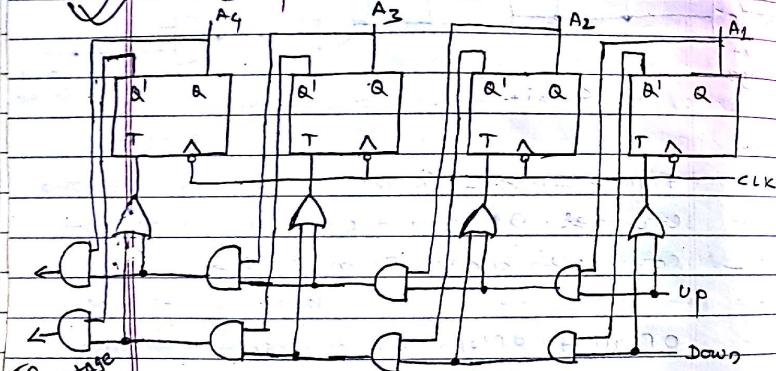


fig: → 4 bit up/down synchronous Counter

known as Bidirectional Counter. Above figure shows 4 bit up/down synchronous Counter. Here T-Flip Flops are used in binary Counter which counts up or down. It has two operation

Date \_\_\_\_\_  
Page \_\_\_\_\_

When the up input control is 1, the circuit counts up where 'T' inputs are determined from the previous value of the normal output in Q.

When the down input control is 1, the circuit counts down, since the complement output  $\bar{Q}$  determine the state of the T inputs.

c) Ring Counter

Ring Counter is the simplest Shift Register Counter.

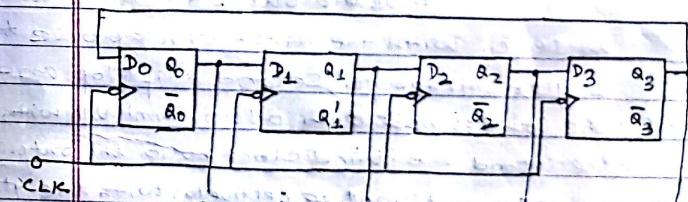


fig: → 4 bit ring counter using D-Flip Flop

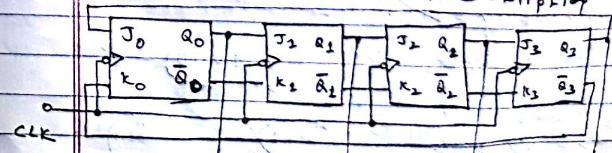
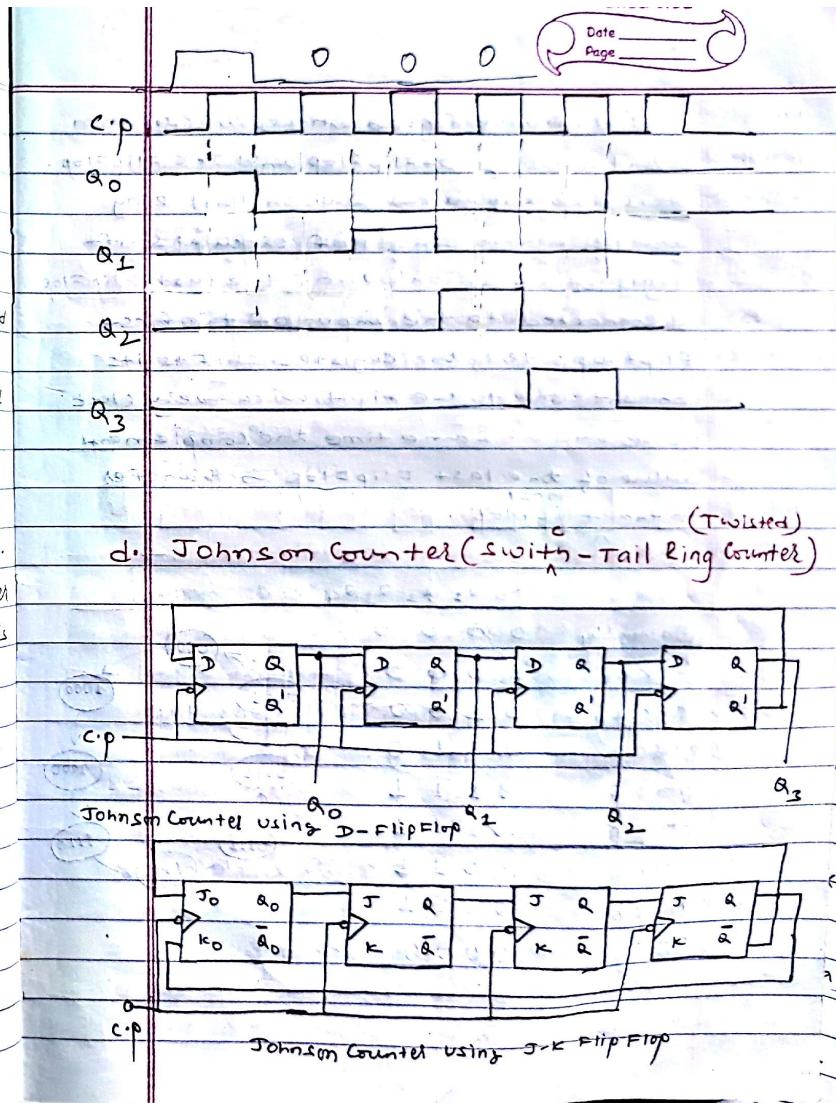
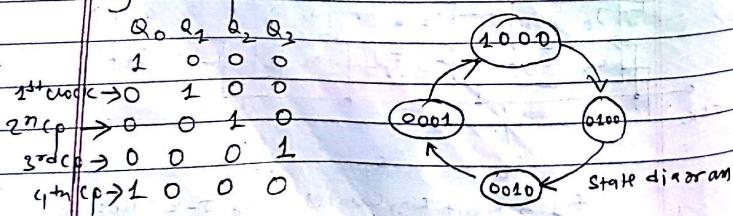


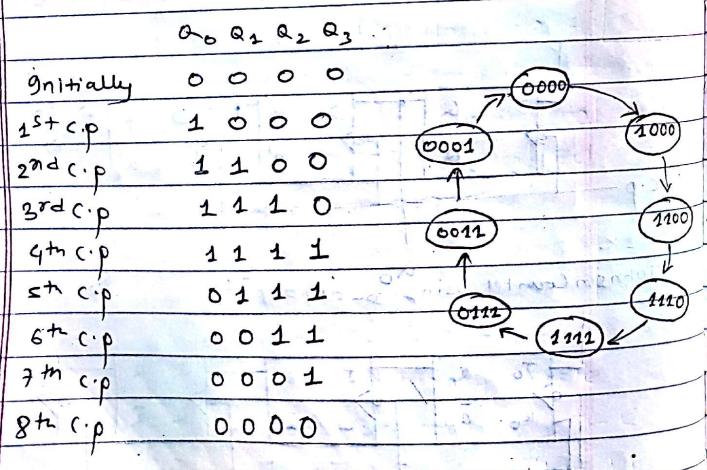
fig: → 4 bit ring counter using J-K Flip Flop

Above figure shows the Ring Counter using different Flip Flop. Here the flipflops are arranged as normal shift register. The output  $Q$  of the flip flop is connected to the input  $D$  of next stage. The output  $Q$  of the last Flip Flop is connected back to input  $D$  of first flip flop which shows the flip flop is arranged in a Ring and is named as Ring Counter.

Let us assume the flip flop  $Q_0$  is said to be logic 1 and other flip flops are set to logic low 0. Thus the output of the counter is 1000. With the first clock pulse this 1 gets shifted to second Flip Flop and the counter output is 0100, similarly with the second and third clock pulse the output is 0010 and 0001 respectively. With the 4th clock pulse the output is again 1000 and cycle repeats.



Above figure shows the Johnson counter using D-FlipFlop and J-K FlipFlop. It is also known as switch tail ring counter which consists of circular shift register whose output of the last FlipFlop is connected to the input of the first flipflop. Here, the register shifts its content one to the right with every clock pulse and at same time the complement value of the last FlipFlop is transferred to the flipflop.



Let us assume that the counter is initially reset to all zero (0's). With the 1<sup>st</sup> clock pulse the output will be 1000 and the cycle changes the output to 0000 again at 8<sup>th</sup> clock pulse. Therefore starting from a cleared state the switch tail ring counter goes through a sequence of 8 states. i.e.  $n$ -bit will go through a sequence of  $2^n$  states.

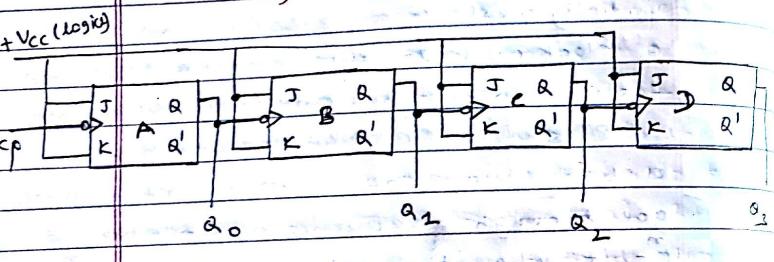
## 2. Asynchronous Counter

Asynchronous Counter are those counters where all the flip flops used in counter are triggered simultaneously. In Asynchronous counter the first flip flop is clocked by the external clock pulse and the output  $Q$  and  $\bar{Q}$  provide clock pulse to the following flip flops.

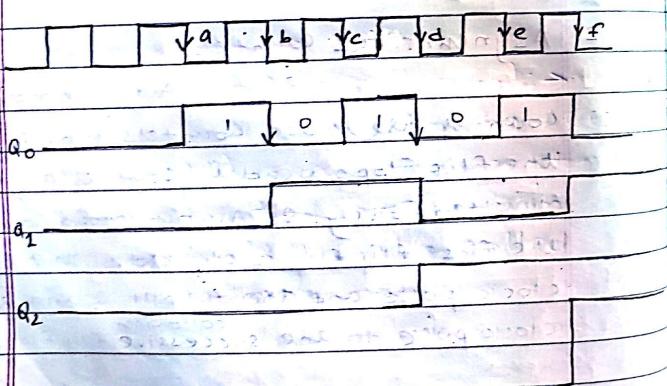
## 3. Ripple Counter $\Rightarrow$

A ripple counter is an arrangement of flip flop where the output of one flip flop drives the clock input of the following flip flop. It is also

called Asynchronous or serial Counter.



This is a 4-bit binary Ripple Counter.



Timing Diagram of 4-bit Ripple Counter

A binary Ripple

Counter can be constructed using J-K Flip Flop. Above figure shows 4-bit Binary Ripple Counter which consists of 4 J-K Flip Flop

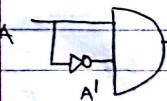
A square clock drives flip flop A. Output of A drives B and output of B drives C and output of C drives D. All J and K inputs are tied to positive Vcc (+Vcc) and the flip flops are negative edge triggered.

Above timing diagram shows the action of counter as the clock runs. For every clocked negative trigger, flip flop will change the state which is clearly seen.

\* Output hazard (Races) →

A race condition or race hazard is the behaviour of an electronic system where the output is dependent on the sequence or timing of other uncontrollable events. The term races is originated due to two signals racing each other to influence the output first.

Consider a two input AND gate fed with a logic signal A on one input and A' on the next. In theory

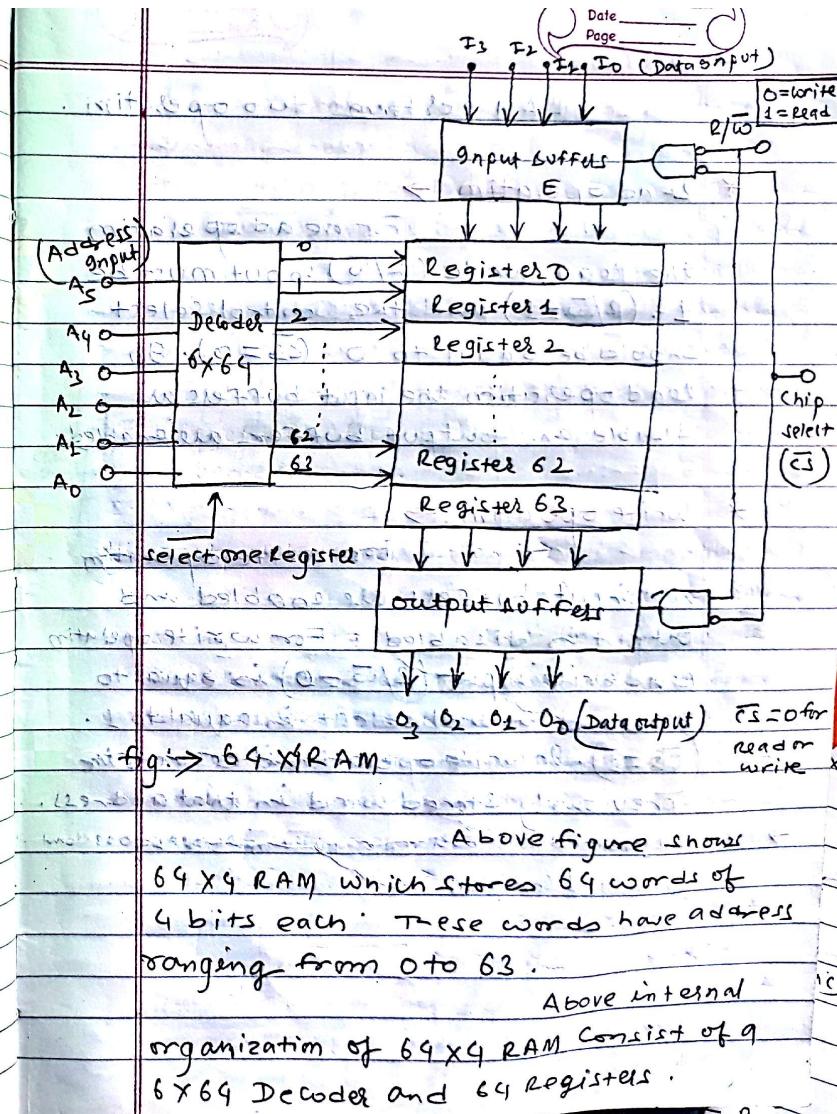


AND gate fed with a logic signal A on one input and A' on the next. In theory

product of A and  $A'$  is always 0. However if changes in the value of A takes longer to propagate to the second input than the first, for a short period it will ensure both inputs are high and thus the output is also high. This condition is called race condition which has to be eliminated on design of the circuits.

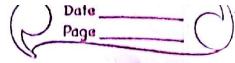
### \* RAM (Random Access Memory) :

Random Access Memory (RAM) is an array of storage cell that stores the information in binary form. RAM is also called volatile memory.



Above figure shows 64x4 RAM which stores 64 words of 4 bits each. These words have address ranging from 0 to 63.

Above internal organization of 64x4 RAM consist of 9x64 Decoder and 64 Registers.



RAM performs two operations.

### \* Read operation :→

For read operation the Read/write ( $R/\bar{W}$ ) input must be 1. ( $R/\bar{W} = 1$ ) and the Control select should be equal to 0 ( $\bar{CS} = 0$ ). In read operation the input buffers are disable and output buffers are enabled.

### \* Write operation :→

For write operation the input buffers are enabled and output is disabled. For write operation read/write input ( $R/\bar{W} = 0$ ) is equal to zero and control select is equal to 1. ( $\bar{CS} = 1$ ). In write operation it destroys the previously stored word in that address.

\* Complemented the lower order position  $A_3 A_2 A_1 = 011$  don't give all 1 condition

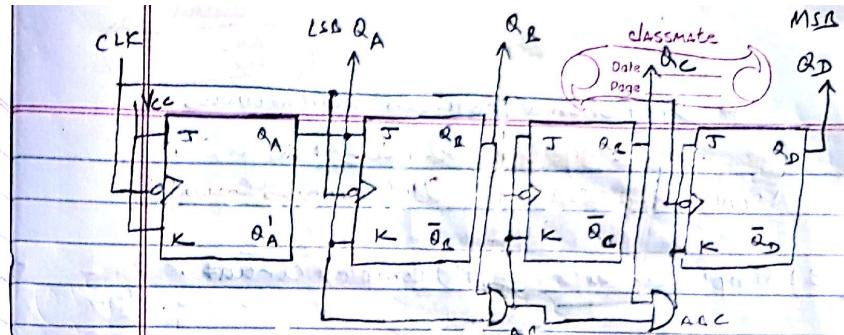


fig :→ 4 bit synchronous up counter

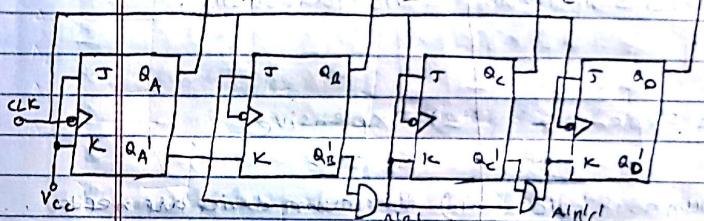


fig :→ 4 bit synchronous Down counter

A synchronous binary counter counts from 0 to  $2^n - 1$ ,  $n \Rightarrow$  number of bits/flip flop. Each flip flop is used to represent 1 bit. The flip flop in the lowest order position is complemented/toggled with every clock pulse and a flip flop in any other position is complemented on the next clock pulse provided all the bits in the lower order position are equal to 1. For example  $A_4 A_3 A_2 A_1 = 0011$ ,  $\text{next count} = 0100$ .  $A_1$  is the lowest order bit, is always complemented.  $A_2$  is complement because all the low order position ( $A_1$  in this case are all 1).  $A_3$  is also complemented because all the low order position  $A_2$  &  $A_3$  are 1's. But  $A_4$  is not

## \* Difference between Synchronous and Asynchronous Counter

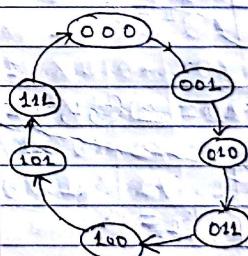
Asynchronous Counter	Synchronous Counter
----------------------	---------------------

- |   |   |
|---|---|
| 1) Simple circuit                               | 2) Complex circuit.                         |
| 2) All FlipFlop are not clocked simultaneously. | 2) All FlipFlop are clocked simultaneously. |
| 3) Less Expensive                               | 3) Expensive                                |
| 4) Minimum no of logic device are needed        | 4) Maximum device are needed.               |

$A_3 A_2 A_1 A_0 = 0100$  next count  $0011$   
 For binary down counter, the low order bit is always toggled,  $A_2$  will be complemented if all the low order position ( $A_2$ ) on this case are all 0's.  $A_2$  is also complemented because all the low order position  $A_2$  and  $A_1$  are 0's.  $A_0$  is not complemented the low order position,  $A_3 A_2 A_1 = 011$  don't give all 0 condition.

- 1) Design a 3 bit synchronous binary up counter using T-K FlipFlop

Present State	Next State	Excitation Input
000	001	$J_C K_A$ 0 X    0 X
001	010	$J_C K_A$ 0 X    0 X
010	011	$J_C K_A$ 0 X    X 0
011	100	$J_C K_A$ 1 X    X 1
100	101	$J_C K_A$ X 0    0 X
101	110	$J_C K_A$ X 0    1 X
110	111	$J_C K_A$ X 0    X 0
111	000	$J_C K_A$ X 1    X 1



State diagram.

$$J_C = \sum 3 \quad d(J_C) = \sum (4, 5, 6, 7)$$

C	$\bar{A}^0$	$\bar{A}^1$	$\bar{A}^2$	
$c'$	0	1	1	2
$c$	X	X	X	X

$J_C = \text{LA} \#$

$$K_C = \sum 7 \quad d(K_C) = \sum (0, 1, 2, 3)$$

C	$\bar{A}^0$	$\bar{A}^1$	$\bar{A}^2$	$\bar{A}^3$
$c'$	0	1	X	X
$c$	4	5	7	6

$K_C = \text{LA} \#$

$$J_B = \sum (1, 5) \quad d(J_B) = \sum (2, 3, 6, 7)$$

C	$\bar{A}^0$	$\bar{A}^1$	$\bar{A}^2$	$\bar{A}^3$
$c'$	0	1	X	X
$c$	4	5	1	X

$J_B = A$

$$K_B = \sum (3, 7) \quad d(K_B) = \sum (0, 1, 4, 5)$$

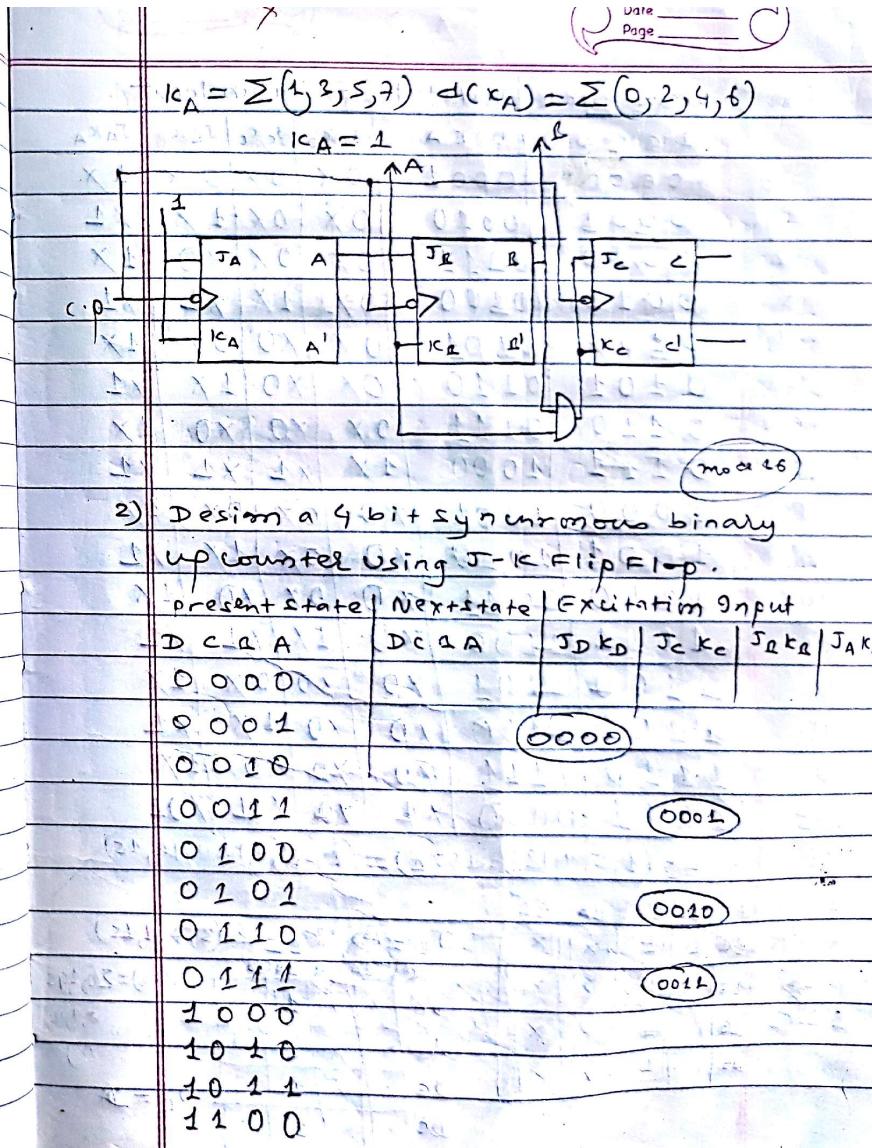
C	$\bar{A}^0$	$\bar{A}^1$	$\bar{A}^2$	$\bar{A}^3$
$c'$	1	X	X	1
$c$	X	X	1	X

$K_B = A$

$$J_A = \sum (0, 2, 4, 6) \quad d(J_A) = \sum (1, 3, 5, 7)$$

C	$\bar{A}^0$	$\bar{A}^1$	$\bar{A}^2$	$\bar{A}^3$
$c'$	1	X	X	1
$c$	1	X	X	1

$J_A = 1$



Present State		Next State		Excitation Input					
D	CBA	D'	CBA	J <sub>D</sub>	K <sub>D</sub>	J <sub>C</sub>	K <sub>C</sub>	J <sub>A</sub>	K <sub>A</sub>
0	0000	0001	0X	0X	0X	1X			
1	0001	0010	0X	0X	1X	X1			
2	0010	0011	0X	0X	X0	1X			
3	0011	0100	0X	1X	X1	X1			
4	0100	0101	0X	X0	0X	1X			
5	0101	0110	0X	X0	1X	X1			
6	0110	0111	0X	X0	X0	1X			
7	0111	1000	1X	X1	X1	X1			
8	1000	1001	X0	0X	0X	1X			
9	1001	1010	X0	0X	1X	X1			
10	1010	1011	X0	0X	X0	1X			
11	1011	1100	X0	1X	X1	X1			
12	1100	1101	X0	X0	0X	1X			
13	1101	1110	X0	X0	1X	X1			
14	1110	1111	X0	X0	X0	1X			
15	1111	0000	X1	X1	X1	X1			

$$J_A = \sum(1, 5, 9, 12) \quad d(J_A) = (2, 3, 6, 7, 10, 11, 14, 15)$$

DC \ D'A' C'A' B'A' BA' BA'

DC	1	X	X	J <sub>D</sub> = A	K <sub>D</sub> = $\sum(8, 7, 11, 15)$
D'C	1	X	X	d(K <sub>D</sub> ) = $\sum(6, 10, 15)$	
DC	1	X	X	d(K <sub>D</sub> ) = $\sum(8, 9, 13)$	
DC'	1	X	X	J <sub>D</sub> = A	
DC'	X	X	X		

$$J_D = \sum(2, 11, 12, 13, 14, 15) \quad d(J_D) = \sum(8, 9, 10, 11, 12, 13, 14, 15)$$

DC	X'A	A'A'	A'A	SA	SA
D'C	0	1	3	2	
DC	4	x	2	1	6
DC	12	11	10	9	8
DC'	8	x	x	10	x

$$J_D = CBA$$

$$K_D = \sum(1, 5, 7) \quad d(K_D) = \sum(0, 1, 2, 3, 4, 5, 6, 7)$$

DC	X	X	X	X
D'C	X	X	X	X
DC				
DC				

$$K_D = CBA$$

$$J_C = \sum(3, 11, 12) \quad d(J_C) = \sum(4, 5, 6, 7, 12, 13, 14, 15)$$

DC	X'A	A'A'	A'A	SA	BA'
D'C			2		
DC	X	X	X	X	
DC	X	X	X	X	
DC'			1		

$$J_C = BA$$

$$K_C = \sum(7, 15) \quad d(K_C) = \sum(0, 1, 2, 3, 8, 9, 10, 11)$$

$$K_C = BA$$

$$J_A = \sum (0, 2, 4, 6, 8, 10, 12, 14)$$

$$d(J_A) = \sum (2, 3, 5, 7, 9, 11, 13, 15)$$

$\Delta$   $\begin{matrix} J_A \\ J'_A \\ K_A \\ K'_A \end{matrix}$

$J'_c'$	1	1	X	X
$J'_c$	X	X	X	1
$J_c$	1	X	X	1
$J_c'$	1	X	X	1

$$J_A = 1$$

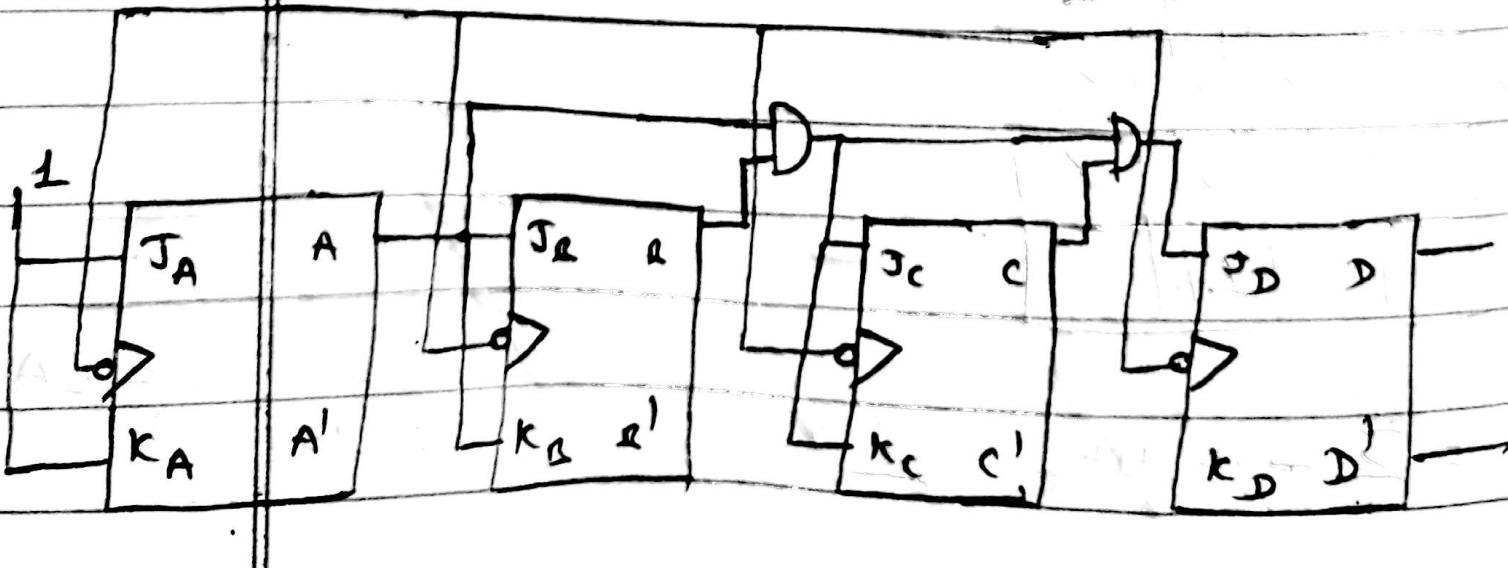
$$K_A = \sum (1, 3, 5, 7, 9, 11, 13, 15)$$

$$d(K_A) = \sum (0, 2, 4, 6, 8, 10, 12, 14)$$

$\Delta$   $\begin{matrix} J_A \\ J'_A \\ K_A \\ K'_A \end{matrix}$

$J'_c'$	X	1	1	X
$J'_c$	X	1	1	X
$J_c$	X	1	1	X
$J_c'$	X	1	1	X

$$K_A = 1$$



3) Design a 3 bit synchronous binary down counter using J-K FlipFlop

Excitation input

Present state	Next state	$J_C$	$K_C$	$J_K$	$K_K$
CBA 7 111	CBA 110	X 0	X 0	X 1	
6 110	101	X 0	X 1	1 X	
5 101	100	X 0	0 X	X 1	
4 100	011	X 1	1 X	1 X	
3 011	010	0 X	X 0	X 1	
2 010	001	0 X	X 1	1 X	
1 001	000	0 X	0 X	X 1	
0 000	111	1 X	1 X	1 X	

$$J_C = \sum 0 \quad d(J_C) = \sum 7, 6, 5, 4 \quad K_C = \sum 4 \quad d(K_C) = \sum (9, 1, 3)$$

$C \backslash B^A$			
$\Sigma A' \quad \Sigma A \quad \Sigma A \quad \Sigma A'$			
$c'$	1		X
$c$	X	X	X

$C \backslash B^A$			
$\Sigma A' \quad \Sigma A \quad \Sigma A \quad \Sigma A'$			
$c'$	X	X	X X
$c$	1		

$C \backslash B^A$			
$\Sigma A' \quad \Sigma A \quad \Sigma A \quad \Sigma A'$			
$c'$	1	X	X
$c$	1	X	X

$c'$	X	X	1
$c$	X	X	1

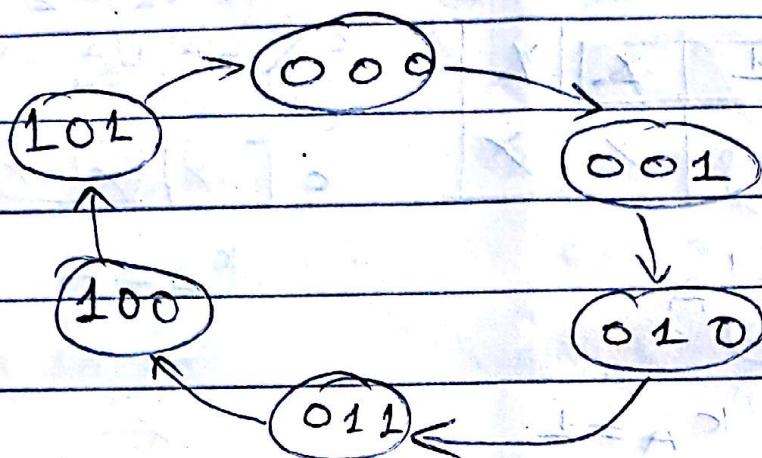
$C \backslash A^A$			
$\Sigma A' \quad \Sigma A \quad \Sigma A \quad \Sigma A'$			
$c'$	1	X	X 1
$c$	1	X	X 1

$c'$	X	1	1	X
$c$	X	1	1	X

# Design mode ⑥ Counter

Page \_\_\_\_\_

Using J-K Flip Flop and S-R Flip Flop



State Diagram of Mod-6 Counter .

present state	next state	Excitation Input		
CBA	CBA	J <sub>C</sub> K <sub>C</sub>	J <sub>R</sub> K <sub>R</sub>	J <sub>A</sub> K <sub>A</sub>
0 000	001	0 X	0 X	1 X
1 001	010	0 X	1 X	X 1
2 010	011	0 X	X 0	1 X
3 011	100	1 X	X 1	X 1
4 100	101	X 0	0 X	1 X
5 101	000	X 1	0 X	X 1
6 110	XXX	X X	XX	XX
7 111	XXX	X X	X X	XX

$$J_C = \sum \beta d(J_C) = \sum (4, 5, 6, 7) \quad K_C = \sum \beta d(K_C) = \sum (0, 1, 2, 3, 6, 7)$$

C	R <sub>A</sub>	R <sub>A'</sub>	S <sub>A</sub>	S <sub>A'</sub>	C'
C	X	X	X	X	1
C'					

$$J_C = RA$$

C	R <sub>A</sub>	R <sub>A'</sub>	S <sub>A</sub>	S <sub>A'</sub>	C'
C	X	X	X	X	X
C'					

$$K_C = A$$

$$J_A = \sum 1 \quad d(J_A) = \sum (2, 3, 6, 7) \quad K_A = \sum 3$$

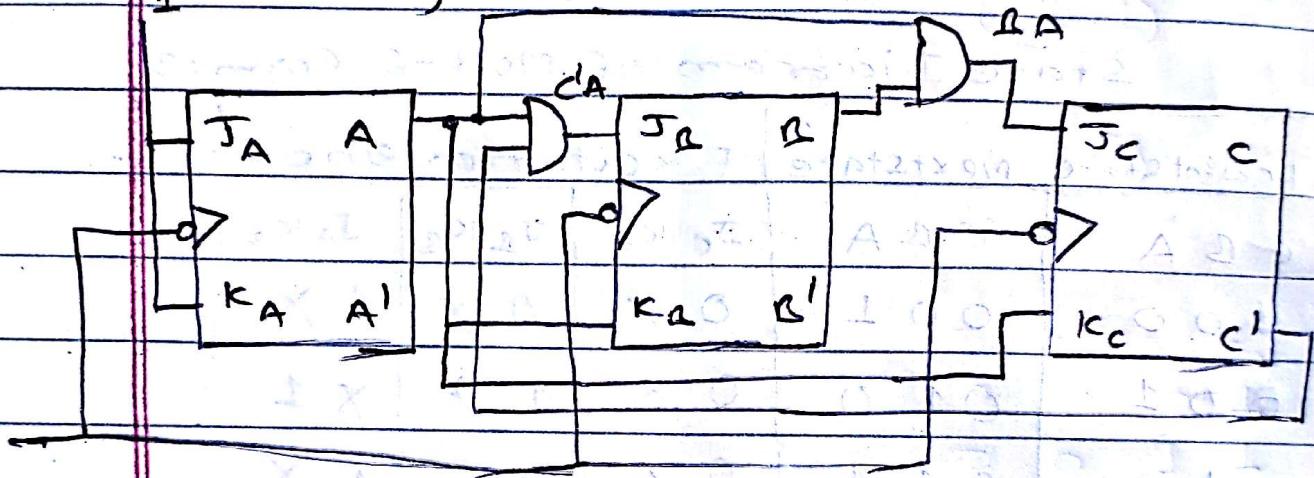
	$J_A$	$K_A$	$J_B$	$K_B$
C'	1	*	*	
C		X	X	

	$J_A$	$K_A$	$J_B$	$K_B$
C'	X	X	1	
C	X	X	X	X

$$J_A = C' A$$

$$K_B = A$$

$$J_A = 1, \quad K_A = 1$$



Using SR Flip Flop

Excitation Input

present state	Next state	$S_C R_C$	$S_B R_B$	$S_A R_A$
0 0 0 0	0 0 1 X	0 X	0 X	1 0
0 0 1 1	0 1 0 X	0 X	1 0	0 1
0 1 0 0	0 1 1 X	0 X	X 0	1 0
0 1 1 1	1 0 0 X	1 0	0 1	0 1
1 0 0 0	1 0 1 X	X 0	0 X	1 0
1 0 1 1	0 0 0 X	0 1	0 X	0 1
1 1 0 0	X X X X	X X	X X	X X
1 1 1 1	X X X X	X X	X X	X X

$$S_C = \sum 3 \quad d(S_C) = \sum(4, 6, 7) \quad R_C = \sum 5 \quad d(R_C) = \sum(6, 1, 2, 6, 7)$$

Value  
Page

C	$\Delta A$	$\Delta A'$	$\Delta A$	$\Delta A'$
C'		1		
C	X	X	X	

C	$\Delta A$	$\Delta A'$	$\Delta A$	$\Delta A'$
C'		X	X	.
C		1	X	X

$$S_C = \Delta A$$

$$R_C = C A$$

$$S_B = \sum 1 \quad d(S_B) = \sum(2, 6, 7) \quad R_B = \sum 3 \quad d(R_B) = \sum(0, 4, 5, 6, 7)$$

C	$\Delta A$	$\Delta A'$	$\Delta A$	$\Delta A'$
C'		1	X	
C		X	X	

C	$\Delta A$	$\Delta A'$	$\Delta A$	$\Delta A'$
C'		X		1
C	X	X	X	X

$$S_B = C' \Delta' A$$

$$R_B = \Delta A$$

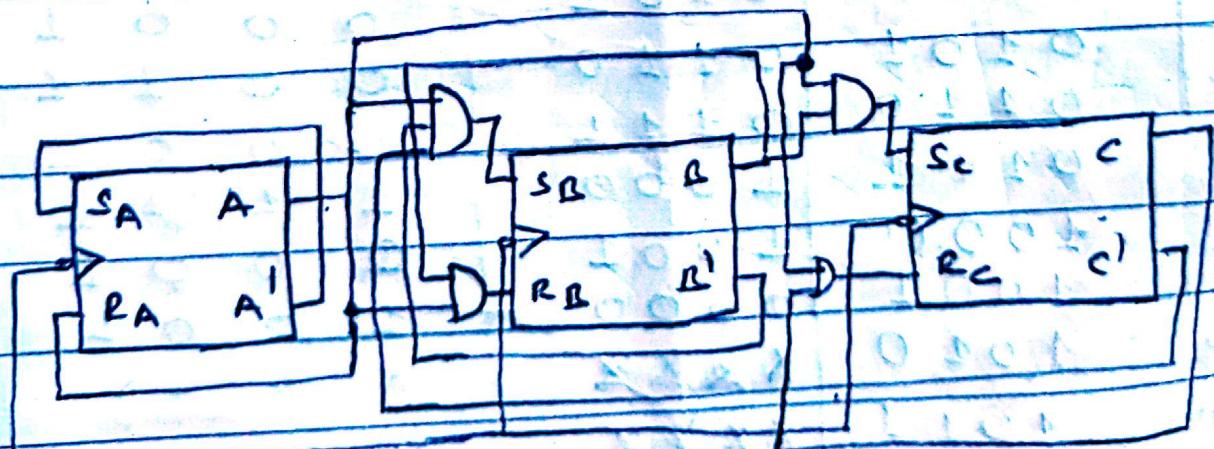
$$S_A = \sum(0, 2, 4) \quad d(S_A) = \sum(6, 7) \quad R_A = \sum(1, 3, 5) \quad d(R_A) = \sum(6, 7)$$

C	$\Delta A$	$\Delta A'$	$\Delta A$	$\Delta A'$
C'	1		1	
C	1	X	X	

C	$\Delta A$	$\Delta A'$	$\Delta A$	$\Delta A'$
C'		1	1	1
C		1	X	X

$$S_A = A'$$

$$R_A = A$$



C.P

