

UNIT-IV

Registers , Counters & PLD'S

SHIFT REGISTER

Introduction to registers

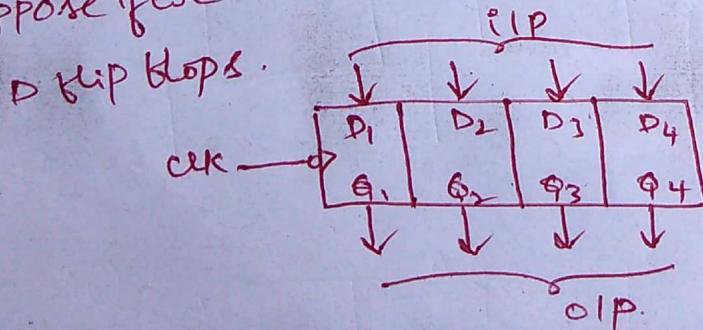
- Flip Flop is a 1 bit memory cell.
- we can't store more than one bit in flip flop.
- so to increase the storage capacity, we have to use group of flip flops.
- This group of flip flop is known as Register.
- The n-bit register consists of 'n' number of flip flops and is capable of storing n-bit word.
- For registers we need 'D' flip flop. →

D	Q _n
0	0
1	1

D → Data.

register means storing the data. in D flip flop whatever the input the same is the output. i.e it storing the data. So D-flip flop is used in Registers.

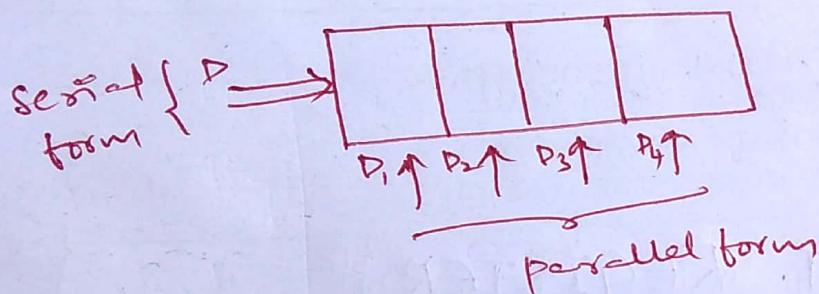
Suppose if we want design 4 bit register, we need 4



Here clock pulse is given internally to all 4 flip flops. Clock pulse acts simultaneously to all 4 flip flops.

Data formats and classification of Registers

- In registers data can be entered in serial or parallel form.
- Serial form means, we can enter one bit at a time.
- Parallel form means, we can enter all bits at a time.



Classification of Registers:

1) depending on IIP & OIP

- Serial in serial out (SISO)
- Serial in parallel out (SIPO)
- Parallel in serial out (PISO)
- Parallel in parallel out (PIPO)

2) depending on application

- Shift register [ex: SISO, SIPO]
- Storage register [ex: SISO, PIPO]

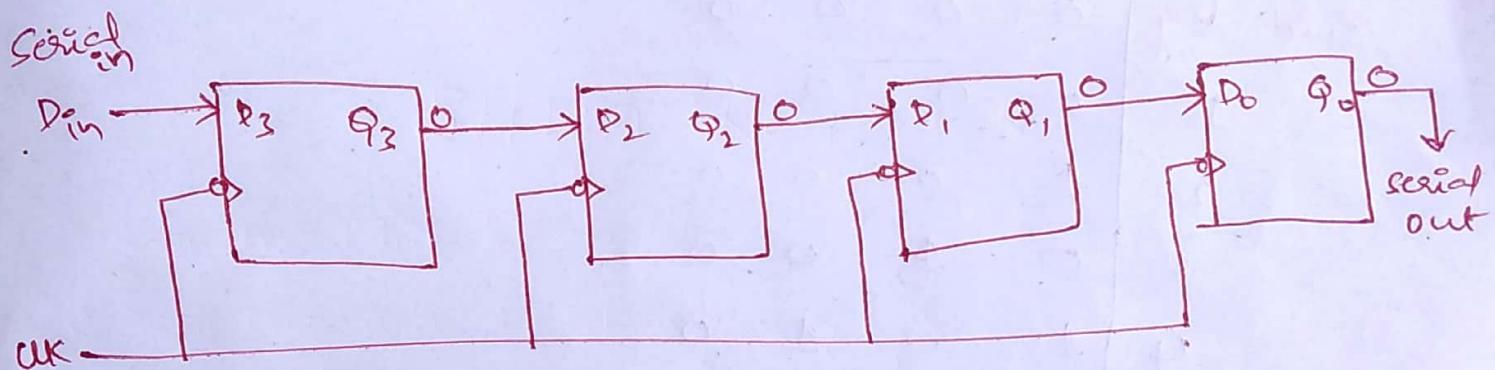
Shift register (serial in serial out)

→ Application of shift register is shifting the data.

In shifting two modes are there.

1. shift right
2. shift left.

Shift right register



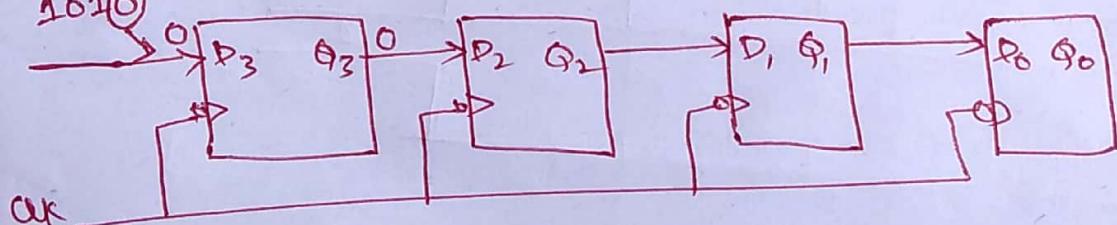
If I want to store the bit 1010,

initially $Q_3 \ Q_2 \ Q_1 \ Q_0 = 0 \ 0 \ 0 \ 0$

initially give input data from LSB to MSB..

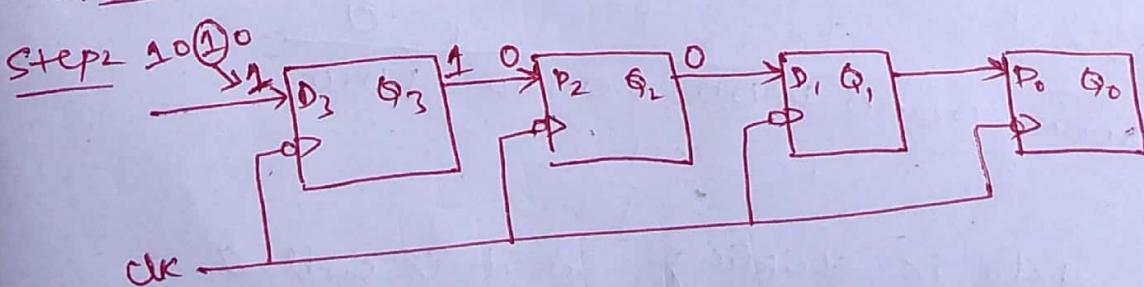
Step 1

1010



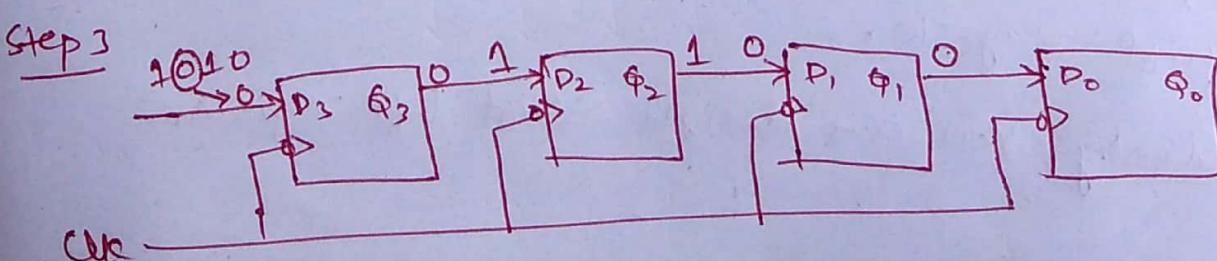
Step 2

1010

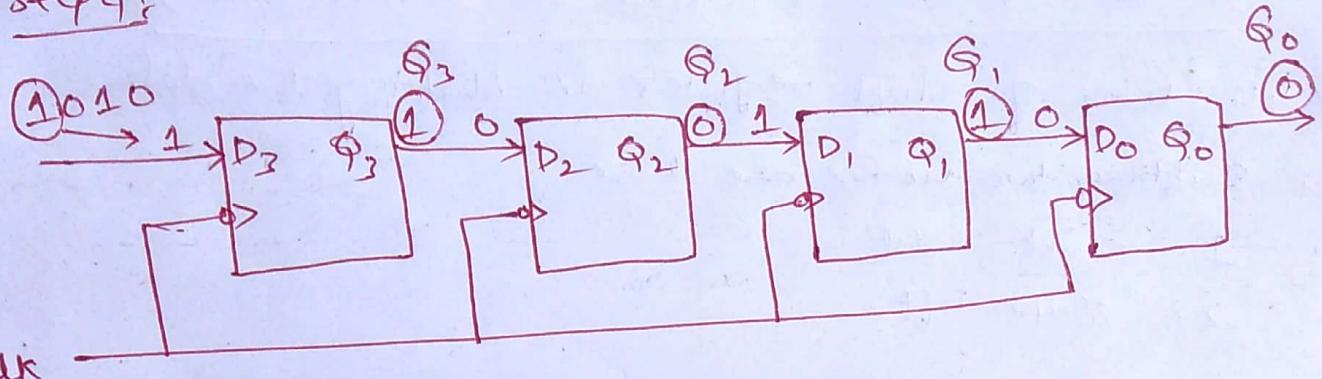


Step 3

1010

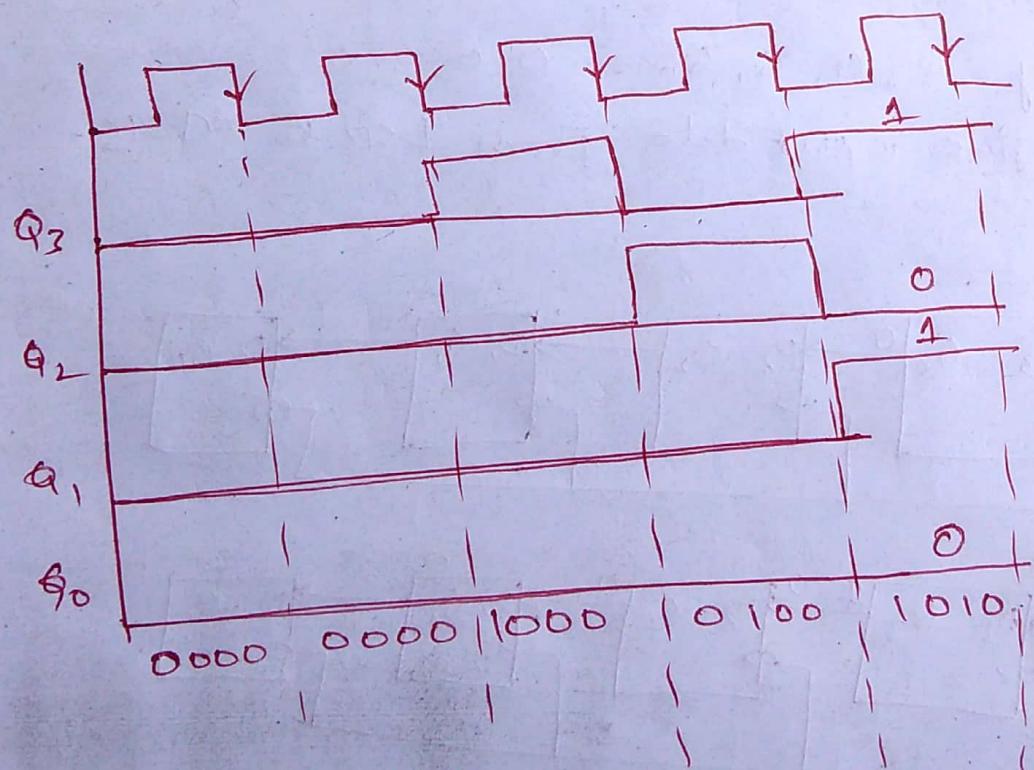


Step 4:



clk	Q ₃	Q ₂	Q ₁	Q ₀
Initially	0	0	0	0
1	0	0	0	0
2	1	0	0	0
3	0	1	0	0
4	1	0	1	0

so now

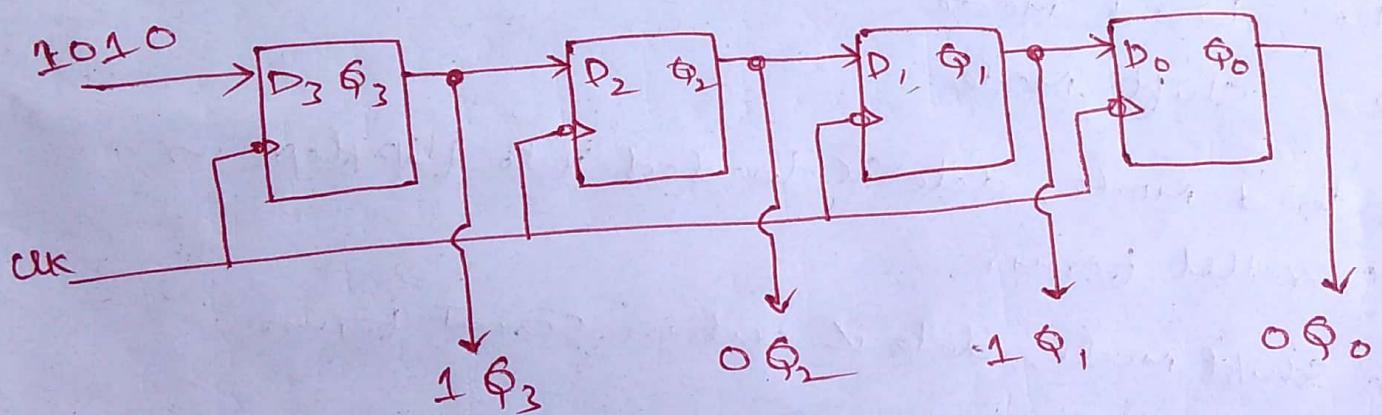


Here data is shifting so it's called shift register.

Similarly you can design shift left register.

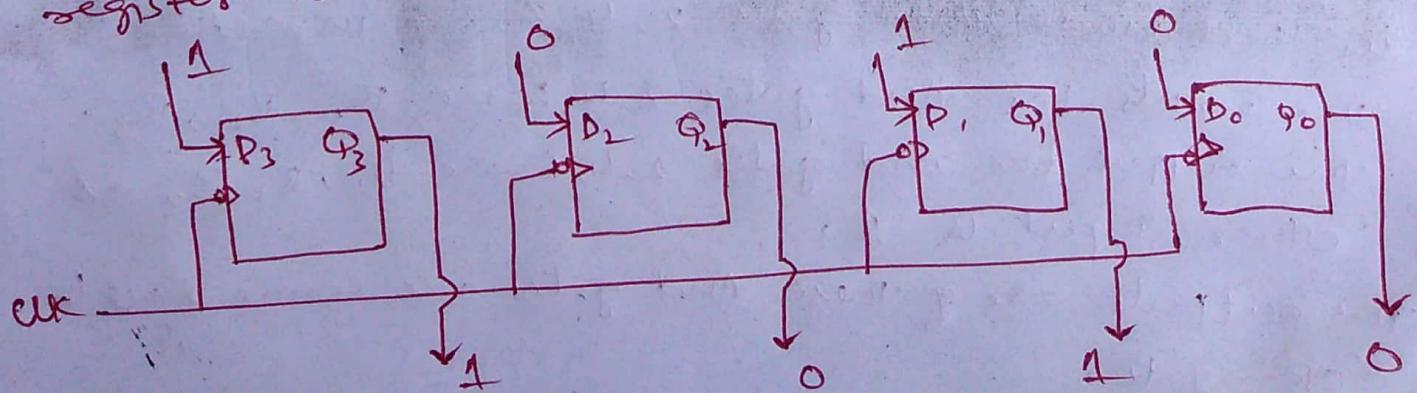
Serial in parallel out (SIPO)

In this SIPO register, data is given in serial form and output is taken in parallel form.

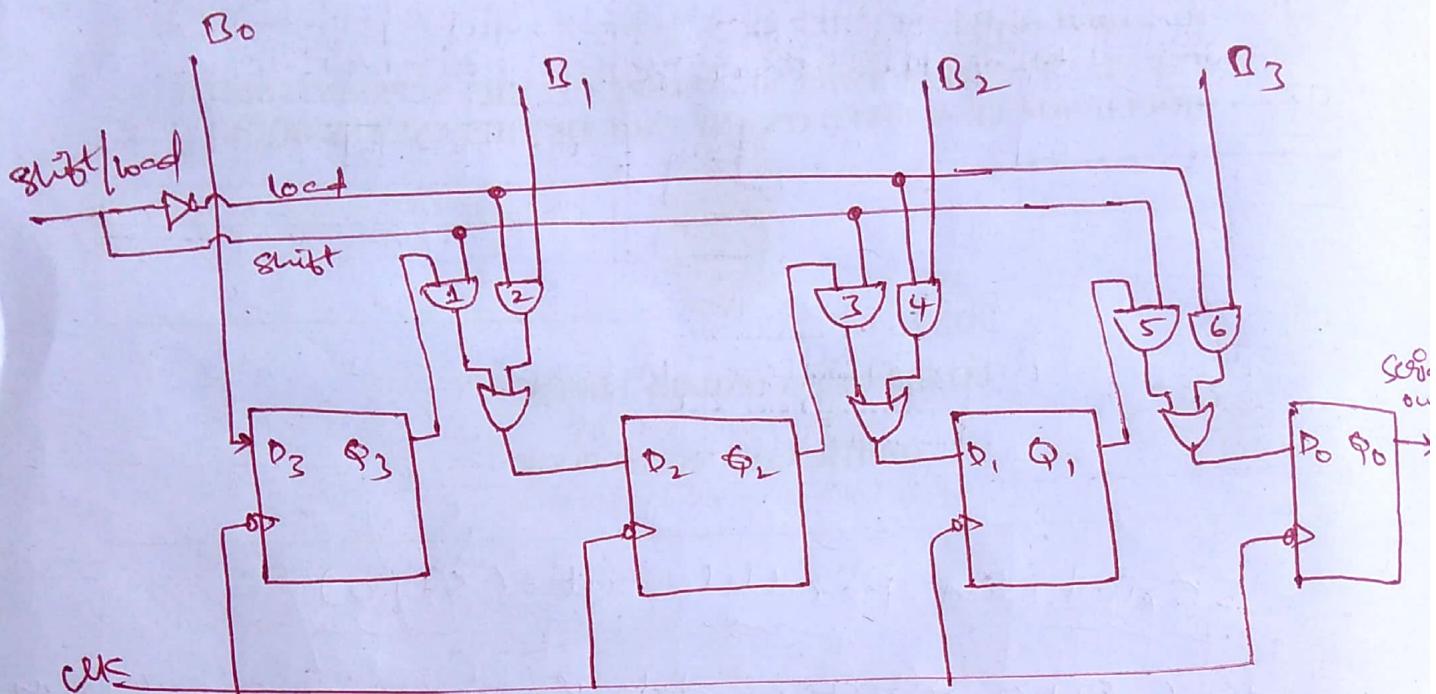


Parallel in parallel out (PIPO)

parallel in, parallel out register is also called as parallel storage register. Here input data is given in parallel form and output also taken in parallel form. This register is used to store the data.



parallel in serial out (PISO)



In this PISO, two modes are there.

1. Load mode

2. Shift mode

In load mode, data is loaded into flip flops in parallel form.

In shift mode, data is out in serial form.

→ When load mode is activated, shift mode is zero.

Similarly when shift mode is activated, load mode is zero.

→ From the diagram, when load mode is activated, shift mode becomes zero. i.e. load = 1, shift = 0.

In load mode AND gates 2, 4, 6 are in active.

The inputs for AND gates 2, 4, 6 are B_1, B_2, B_3 .

The output too AND gates 2, 4, 6 are B_1, B_2, B_3 as other input is 1.

The outputs of these AND gates are connected to OR gate.

The output of OR gate is output of And gate.
Because other input of OR gate is zero.

∴ The inputs of the four flip flops D_3, D_2, D_1, D_0 are B_3, B_2, B_1, B_0 are connected in parallel form.
Now these inputs are stored in flip flops to get out these in serial out form.

∴ To get outputs in serial form we have to activate shift mode.

⇒ In shift mode, shift = 1, load = 0.

In shift mode, the And gates 1, 3, 5 are in active mode and 2, 4, 6 are inactive mode.

For And gates 1, 3, 5 one input is q and other inputs are Q_3, Q_2 and Q_1 .

The output of And gate 1, 3, 5 is Q_3, Q_2, Q_1 ,

The output of And gate 2, 4, 6 is 0.

The outputs of these And gates are connected to the input of OR gate.

So the output of OR gate is Q_2, Q_1, q ,

so the output of OR gate is connected to input of flip flop.

So the inputs of flip flop is Q_2, Q_1 and q ,

so the inputs for four flip flop is B_0, Q_2, Q_1, q ,

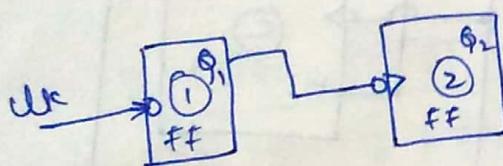
so the output of one flip flop is connected to input of next flip flop. This means date is passing in serial form. In this serial form date is coming out.

COUNTERS

- counter is a sequential circuit, it simply counts.
- you can design your own counter depends upon your need.
- To design counters we use either JK or T flip flop with -ve edge clock pulse.

Types of counters

Ripple / Asynchronous counter



1) FF are connected such a way that o/p of first FF drives the clock of next FF.

2) FF's are not clocked simultaneously
 3) Ckt is simple for more no. of states.
 4) speed is slow as clock is propagated through no. of stages

1) There is no connection b/w o/p of FF₁ and next flip flop.

2) FF's are connected simultaneously.

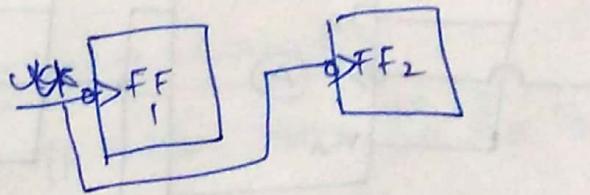
3) Ckt becomes complex as no. of states increases.

4) Speed is high as clock is given at same time.

→ up counter → counts small to big
 ex: 0, 1, 2, 3, ...

→ down counter → counts big to small
 ex: 9, 8, 7, 6, 5, ...

→ updown counter → counts both up and down depends upon mode of counter.



Design of Asynchronous counter

if no. of flip flops are n , then it counts,

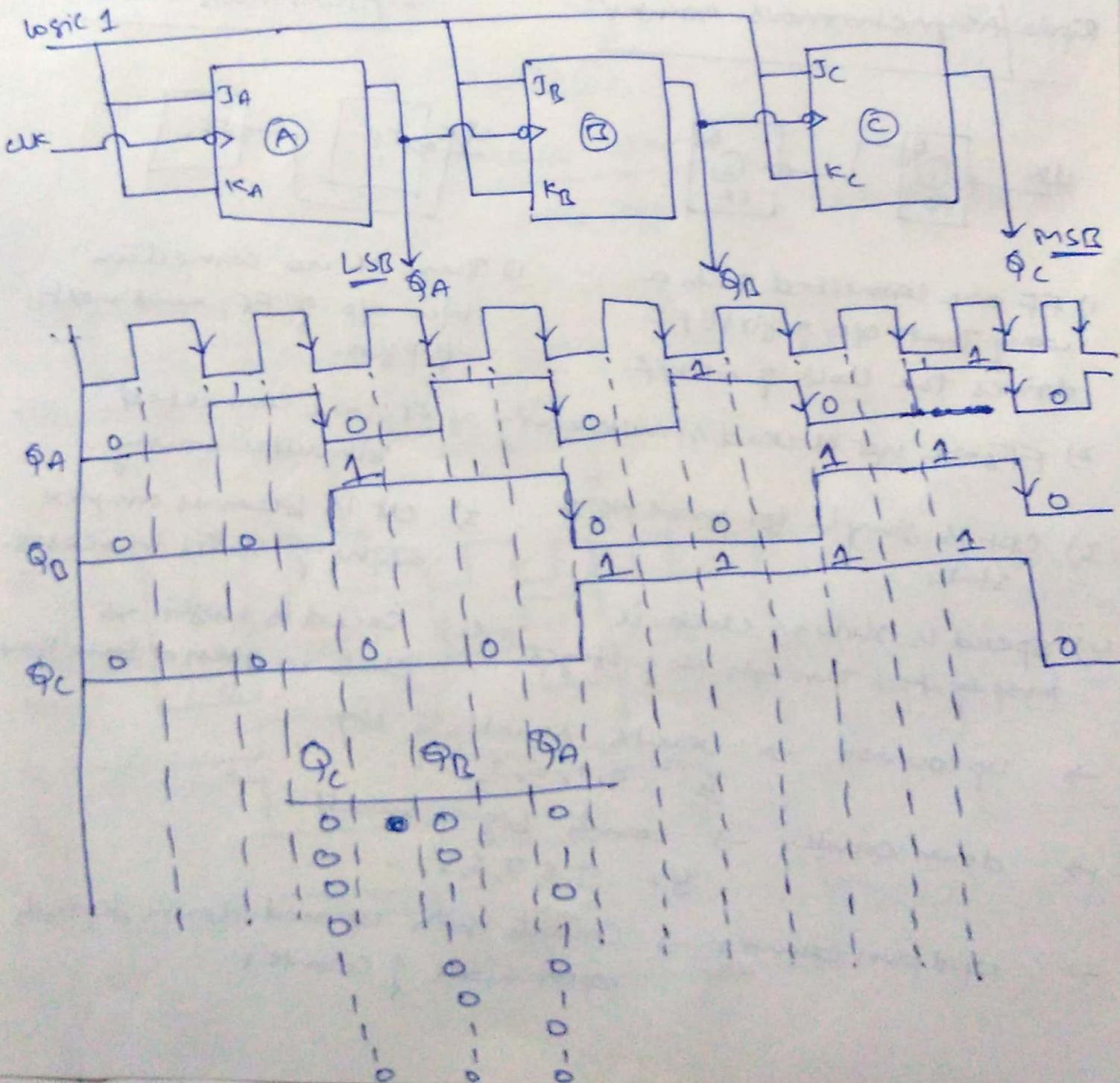
$$\text{no. of states} = 2^n.$$

$$\text{max count} = 2^n - 1$$

if $n = 3$, no. of states $= 2^3 = 8$ i.e. 0, 1, 2, 3, 4, 5, 6, 7, 8
max count $= 2^3 - 1 = 7$.

3 Bit Asynchronous up counter

3 bit means we require 3 FF's.

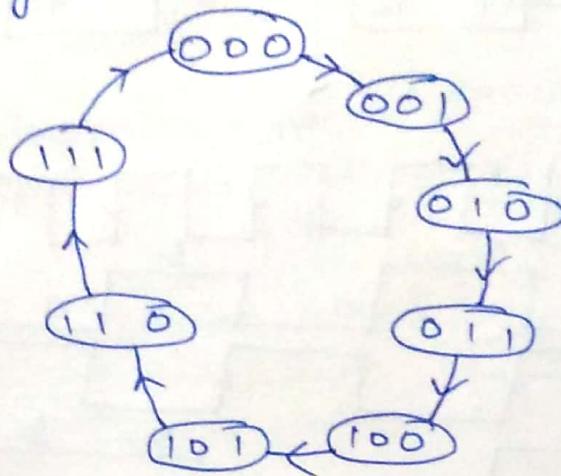


State diagram of a counter

2

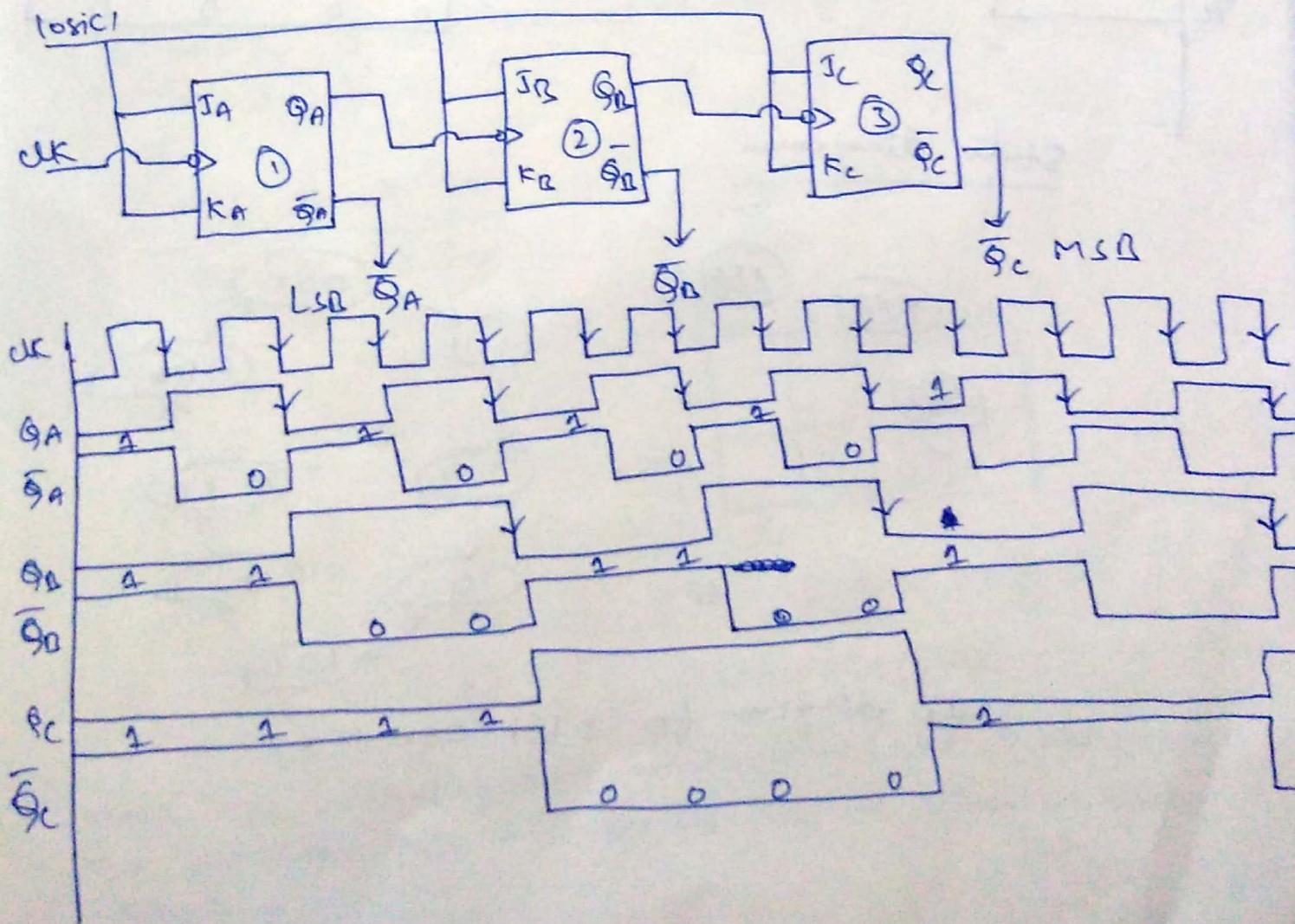
For a 3bit up counter, no. of states = $2^3 = 8$.
 i.e 000 to 111.

up counting means small to big.

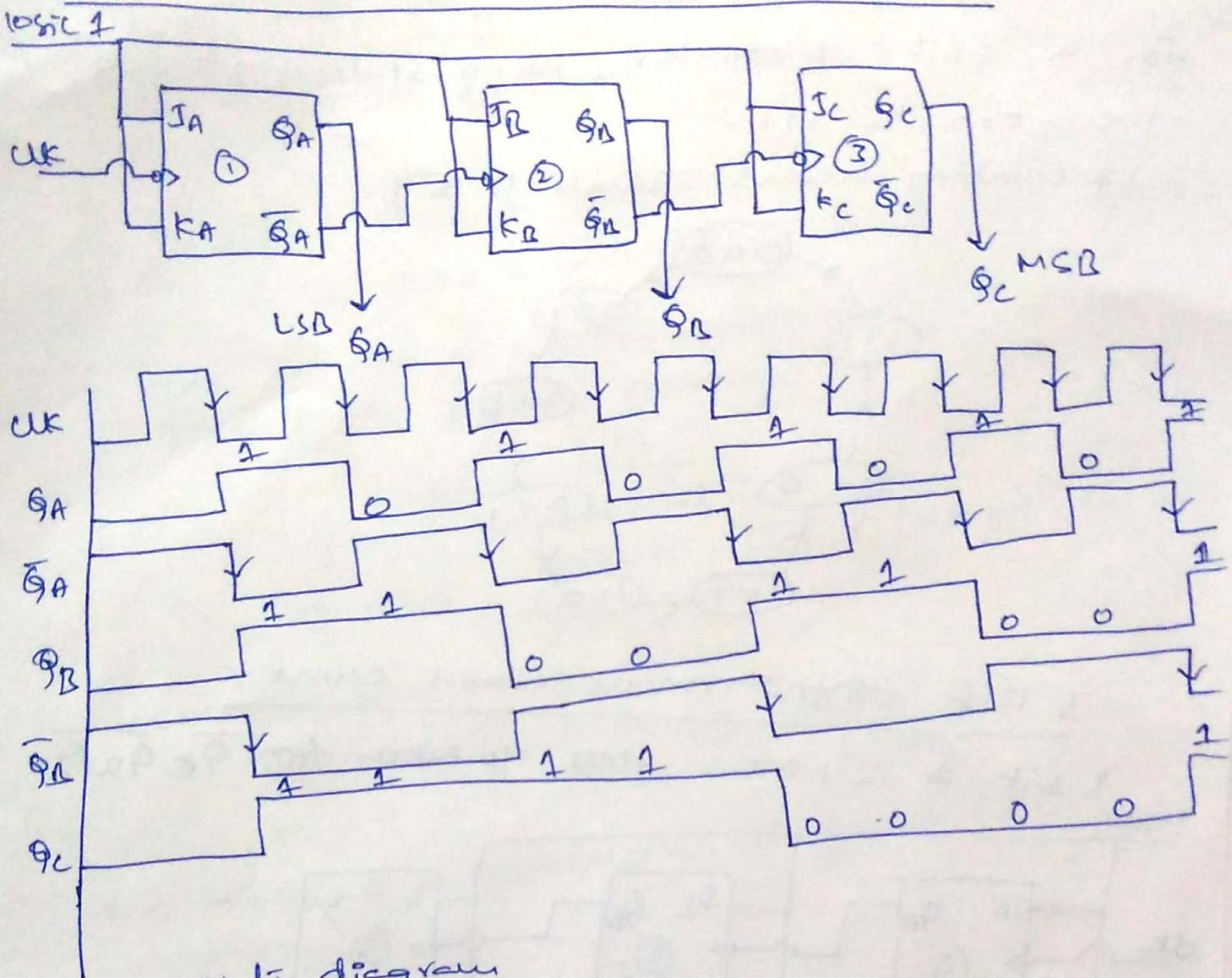


3 bit Asynchronous down counter

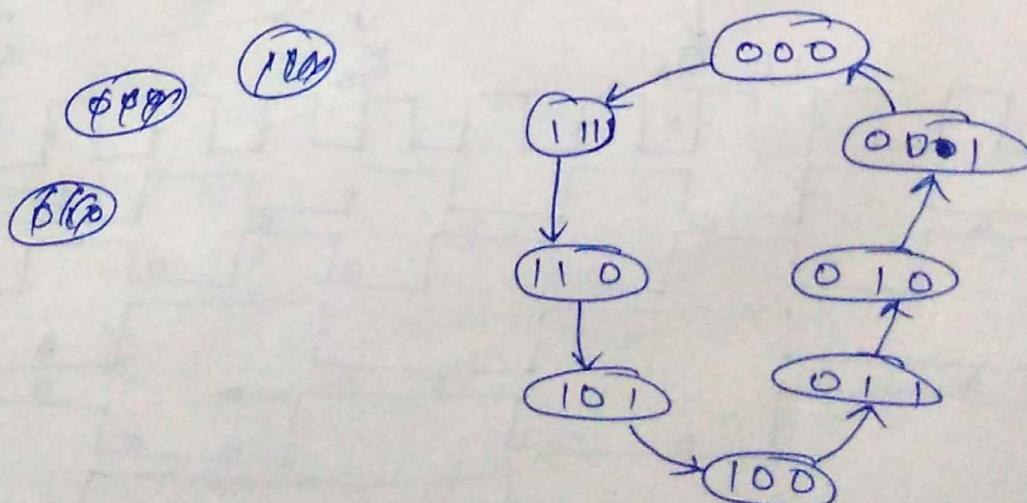
3 Bit Asynchronous



3 Bit Asynchronous down counter



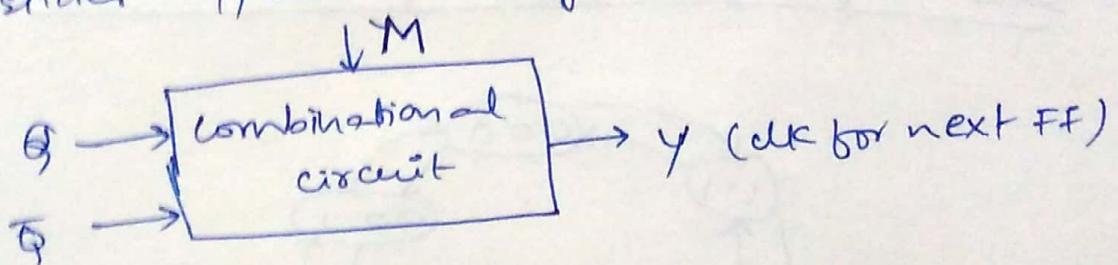
state diagram



state diagram for 3 bit counter.

3 bit up-down Ripple Counter

- A mode control input is used to select either up or down mode.
- if Q_A is clock to next FF, it acts as up counter
if \bar{Q}_A is clock to next FF, it acts as down counter.
- Here I'm designing a combinational circuit, that gives either up/down counting.



Here I'm considering when,

$M = 0 \rightarrow$ it acts as up counter

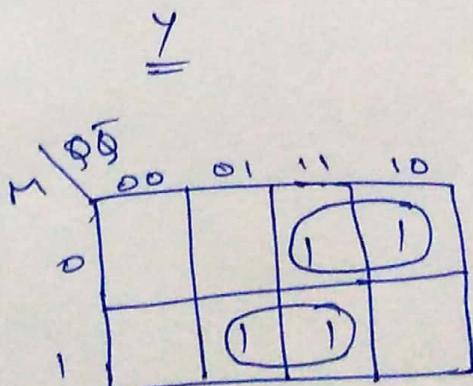
$M = 1 \rightarrow$ it acts as down counter

$M = 1 \rightarrow$ it acts as clk of next FF.

i.e. $M=0$, Q is connected as clk of next FF.

$M=1$, \bar{Q} is " " " " " "

M	Q	\bar{Q}	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
-	-	-	-
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

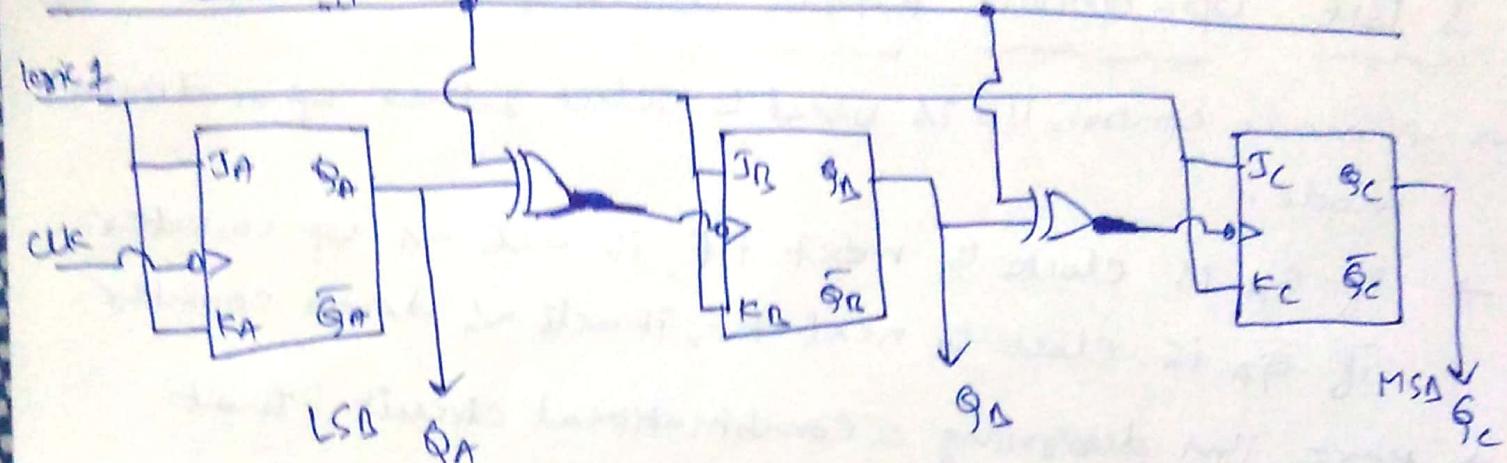


$$Y = \overline{M}Q + M\overline{Q}$$

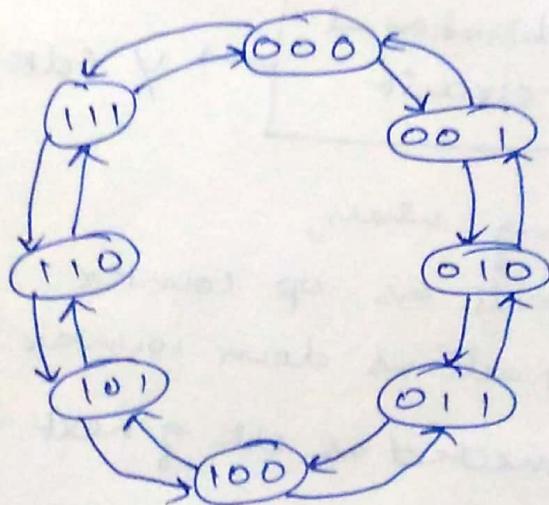
$$Y = M \oplus Q$$

Here I'm inserting the combinational ckt in between the flip flops. This logic decides we use either up/down counting.

mode control input M



state diagram for 3 bit up-down counter



Design of synchronous counter

To design synchronous counter following steps to be followed.

Step 1: Decide the no. of flip flops

Step 2: Excitation table of FF

Step 3: State diagram and circuit excitation table

Step 4: Obtain simplified equations using K map.

Step 5: Draw the logic diagram.

→ Design 2 bit synchronous up counter?

Sol: Step 1: 2 bit means 2 FF's required.

Here i'm considering the JK FF.

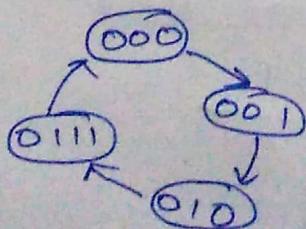
Step 2: Excitation table for JK FF.

PS Q_n	Q_{n+1}	NS	
		I	K
0	0	0	x
0	1	1	+
1	0	x	1
1	1	x	0

Step 3: State diagram.

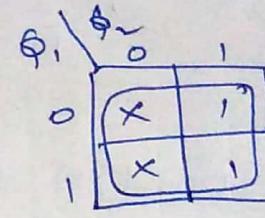
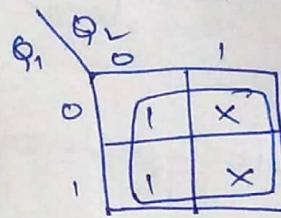
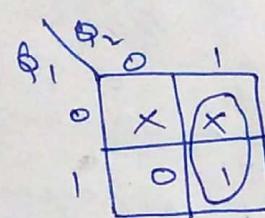
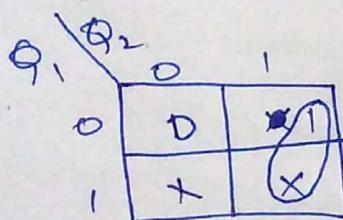
2 bit $\Rightarrow 2^2 = 4 \rightarrow$ no. of states.

$$\text{max. count} = 2^2 - 1 = 4 - 1 = 3.$$



Circuit excitation table :

PS	NS	J ₁	K ₁	J ₂	K ₂
Q ₁ Q ₂	Q ₁ * Q ₂ *				
0 0	0 1	0	x	1	x
0 1	1 0	1	x	x	1
1 0	1 1	x	0	1	x
Step 4: 1 1	0 0	x	1	x	1



$$J_1 = Q_2$$

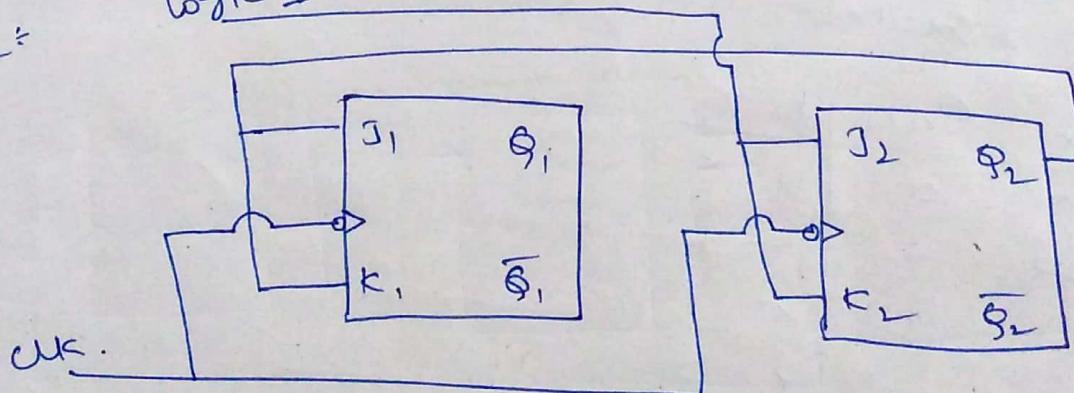
$$K_1 = Q_2$$

$$J_2 = 1$$

$$K_2 = 1$$

Step 5:

logic 1.



⇒ Design 3 bit synchronous down counter?

Step 1: 3 bit means 3 FF's required.

Here i'm Considering T FF.

Step 2: Excitation table for T FF.

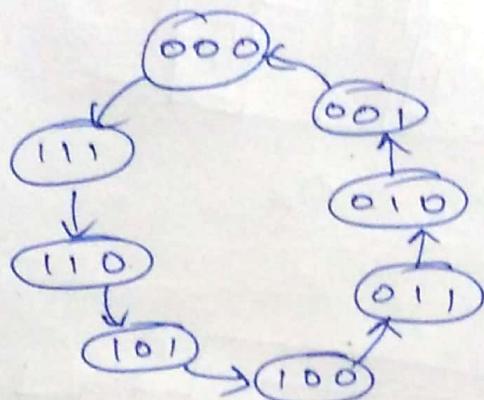
Q _n	S _{n+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

and count

Step 3 : State diagram

3 bit $\Rightarrow 2^3 = 8 \rightarrow$ no. of states

$2^3 - 1 = 8 - 1 = 7 \rightarrow$ max. count.



Circuit Excitation table

PS $Q_C Q_B Q_A$	NS $Q_C^* Q_B^* Q_A^*$	T _A	T _B	T _A
0 0 0	1 1 1	1	1	1
0 0 1	0 0 0	0	0	1
0 1 0	0 0 1	0	1	1
0 1 1	0 1 0	0	0	1
1 0 0	0 1 1	1	1	1
1 0 1	1 0 0	0	0	1
1 1 0	1 0 1	0	1	1
1 1 1	1 1 0	0	0	1

Step 4:

$Q_B Q_A$	00	01	11	10
Q_C	0	1	1	1
0	1	1	1	1
1	1	1	1	1

$$T_C = \overline{Q_B} \overline{Q_A}$$

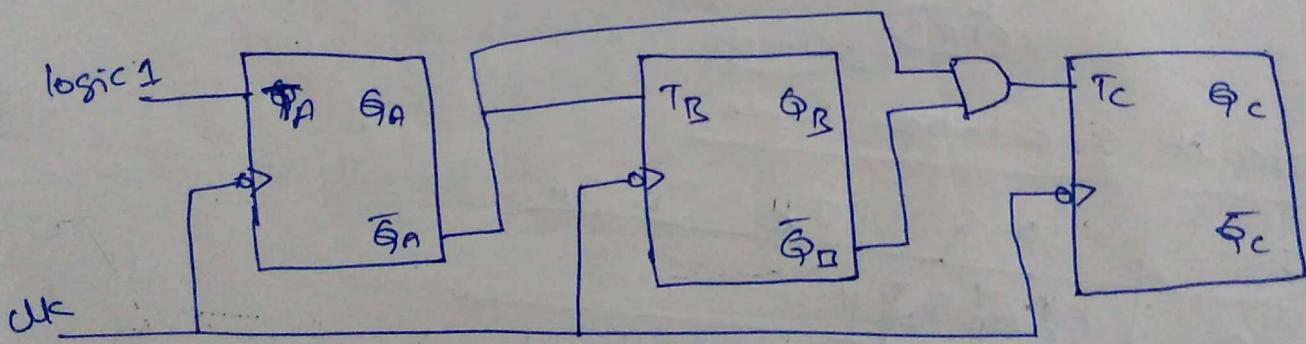
$Q_B Q_A$	00	01	11	10
Q_C	0	1	1	1
0	1	1	1	1
1	1	1	1	1

$$T_B = \overline{Q_A}$$

$Q_B Q_A$	00	01	11	10
Q_C	0	1	1	1
0	1	1	1	1
1	1	1	1	1

$$T_A = 1$$

Step 5



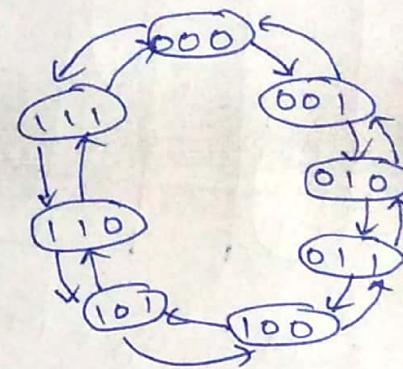
⇒ Design 3 bit synchronous up down counter.

Step 1: 3 bit ⇒ 3 FF's

Here I'm considering T FF.

Step 2: Excitation table for T FF.

Q_n	Q_{n+1}	T
0	0	0
0	1	1
1	0	1
1	1	0



Step 3: State diagram

3 bit ⇒ $2^3 = 8 \rightarrow$ no. of states

max. count ⇒ $2^3 - 1 = 8 - 1 = 7$.

Here I'm considering Mode control P/P M. depending upon mode control P/P, it will count either up/down counting.

I consider, when $M = 0 \rightarrow$ up counting
 $M = 1 \rightarrow$ down counting.

Circuit Excitation Table ↴

M	Q_C	Q_B	Q_A	Q_C^*	Q_B^*	Q_A^*	T_C	T_B	T_A
0	0	0	0	0	0	1	0	0	1
0	0	0	1	0	1	0	0	1	1
0	0	1	0	0	1	1	0	0	1
0	0	1	1	1	0	0	1	1	1
0	1	0	0	1	0	1	0	0	1
0	1	0	1	1	1	0	0	1	1
0	1	1	0	1	1	1	0	0	1
0	-1	-1	-1	0	0	0	1	-1	-1
-1	0	0	0	1	1	1	1	0	1
1	0	0	1	0	0	0	0	0	1
1	0	1	0	0	0	1	0	0	1
1	0	1	1	0	1	0	0	1	1
1	1	0	0	0	1	1	0	0	1
1	1	0	1	1	0	0	0	1	1
1	1	1	0	1	0	1	0	0	1
1	1	1	1	1	1	0	0	0	1

Step 4

MQ_C	$Q_B Q_A$	00	01	11	10
00					
01					
11	1				
10	1				

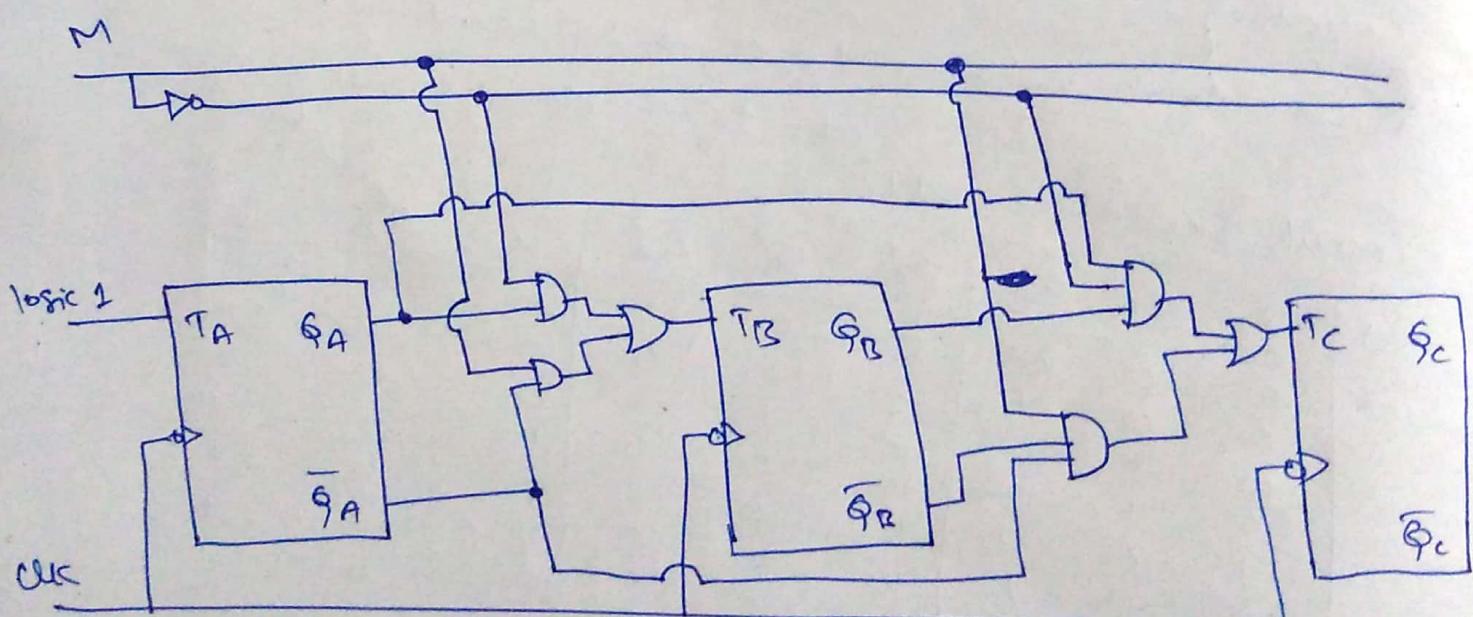
$$T_C = M \bar{Q}_B \bar{Q}_A + \bar{M} Q_B Q_A$$

MQ_C	$Q_B Q_A$	00	01	11	10
00					
01					
11	1	1	1	1	1
10	1	1	1	1	1

$$T_B = \bar{M} Q_A + M \bar{Q}_A$$

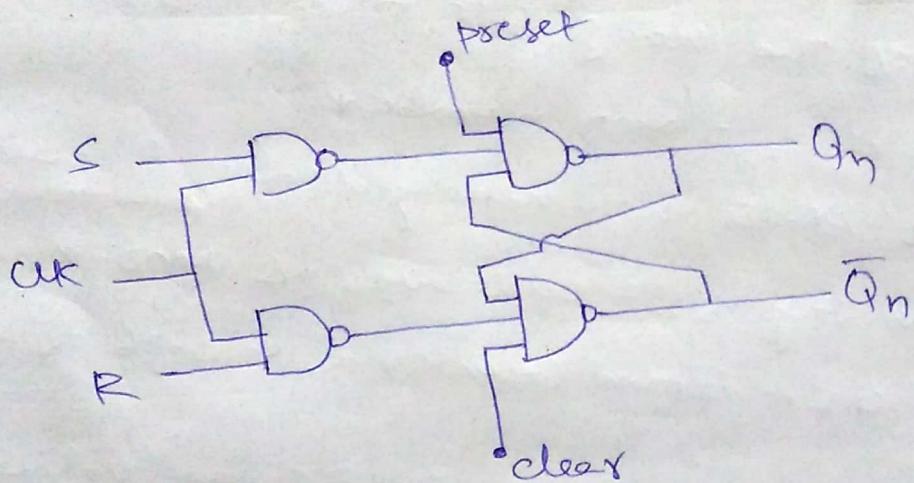
MQ_C	$Q_B Q_A$	00	01	11	10
00		1	1	1	1
01		1	1	1	1
11		1	1	1	1
10		1	1	1	1

$$T_A = 1$$



Preset and clear inputs

- These are called direct inputs or overriding inputs or asynchronous inputs.
- These are called overriding inputs because independent of flip flop inputs when these inputs are present, output of the flip flop changes.



Here, when preset = 0 $\Rightarrow Q_n = 1$

when clear = 0 $\Rightarrow \bar{Q}_n = 1 \Rightarrow Q_n = 0$

whatever be the clock and inputs of the flip flop, ~~then~~
when preset = 0 my Q_n value is 1. Similarly,
when clear = 0 my Q_n value is 0.

→ If preset and clear are equal to 1, then flip flop performs normally and it is independent of preset and clear inputs.

(Note: Roll Numbers should not be written)

Modulus of the Counter & Counting to particular value

- ⇒ 2 bit ripple counter is called MOD-4 or modulus 4 counter.
- ⇒ 3 bit ripple counter is called MOD-8 or modulus 8 counter.
- ⇒ If n is the no. of bits then MOD number = 2^n .
- ⇒ no. of states = MOD number.
- ⇒ Modulus states how many states it having.
- ⇒ Modulus states how many upcounter as MOD-8 upcounter.
- ⇒ So we can say 3 bit up counter as MOD-8 down counter.
Similarly, 4 bit down counter as MOD-16 down counter.
- ⇒ How to stop counting upto particular value

If design MOD-6 counter using MOD-8 counter.

Now we have to design MOD-8 counter.

MOD-8 means 3 bit counter. we can design either up/down counter. from MOD-8 counter we have to design MOD-6 counter. i.e. it must have 6 states and max. count is 5.

In MOD-8 counter,

$$\text{no. of states} = 8$$

$$\text{no. of bits} = 3$$

$$\text{no. of FF's} = 3$$

$$\text{max. count} = 8 - 1 = 7$$

whereas in MOD-6 counter,

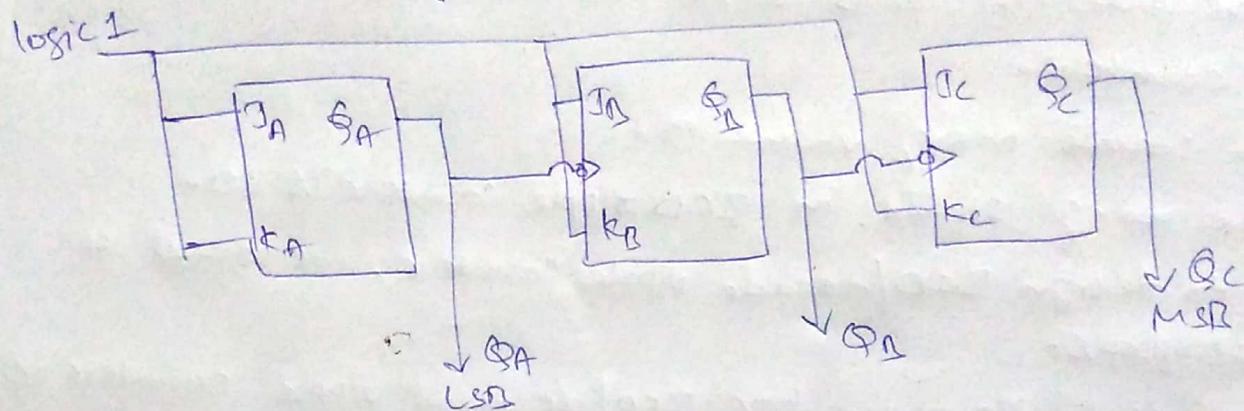
$$\text{no. of states} = 6$$

$$\text{no. of bits} = 3$$

$$\text{no. of FF's} = 3$$

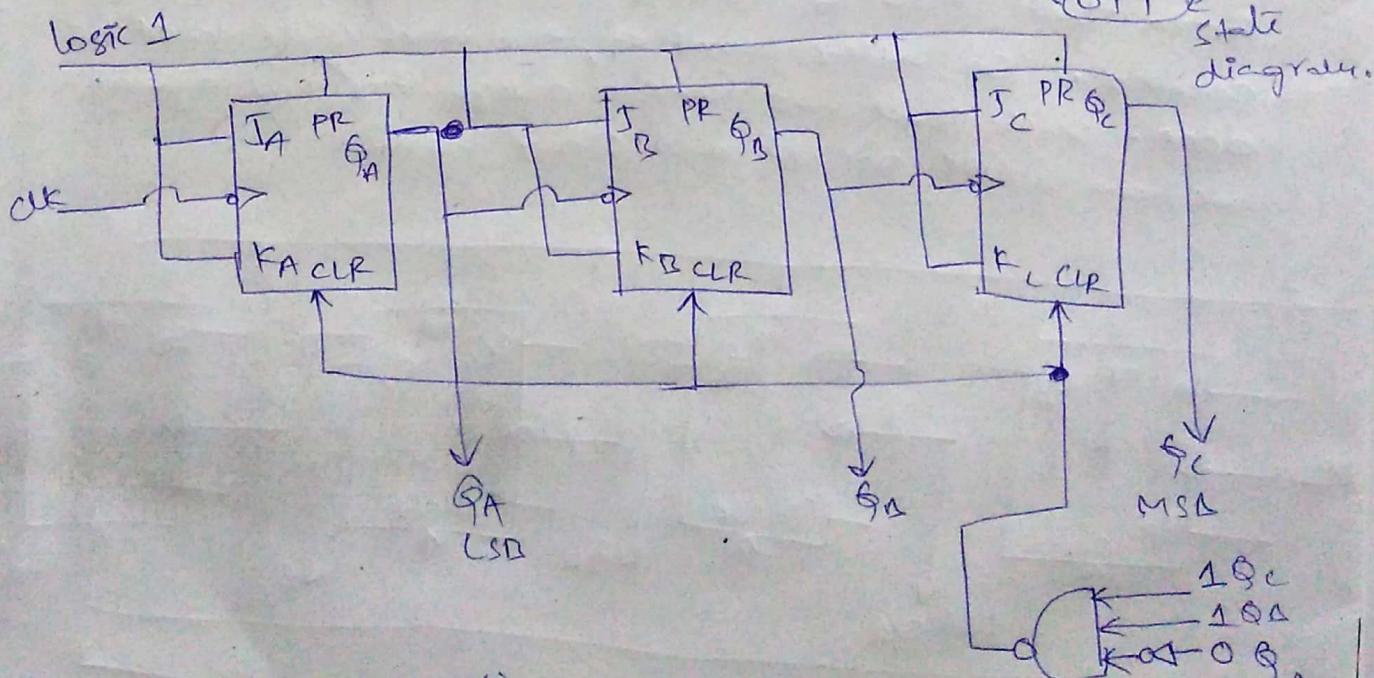
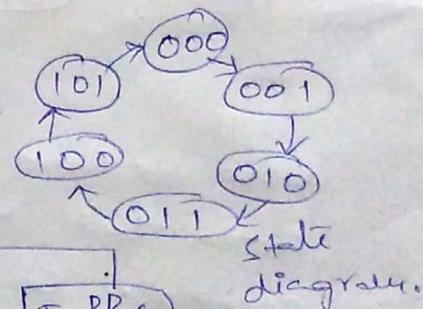
$$\text{max. count} = 6 - 1 = 5$$

First we have to design MOD-8 counter. Here I'm considering up counter.



From this Mod-8 counter, we have to design Mod-6 counter. For this, I use preset and clear inputs. I give preset input to logic 1. So the circuit is independent of preset input. But Mod-8 Counter counts upto 7. But here in Mod-6 Counter, max count is 5 only. So when it reaches Mod-8 Counter counts 5, after that next count is 0, then it acts as 5, after that next count is 1. For that, we give clear i/p = 0. When Mod-8 Counter counts 6.

$$6 = 110$$



Mod-6 Counter using Mod-8 Counter.

Decade (BCD) ripple counter

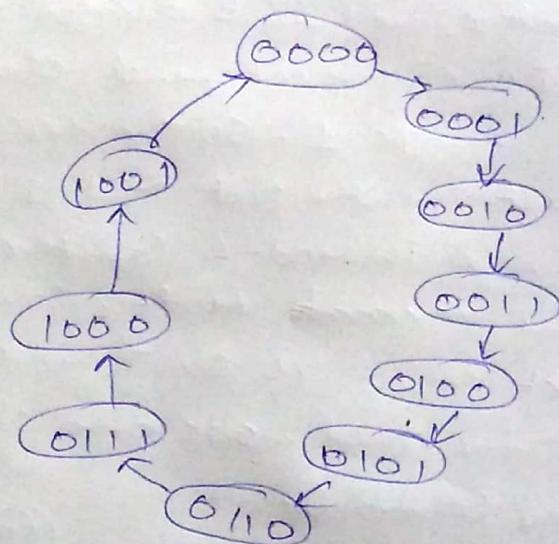
→ Decade (or) BCD ripple counter means Mod-10 counter.

it counts max. value 9.

the no. of states in BCD ripple counter is 10.

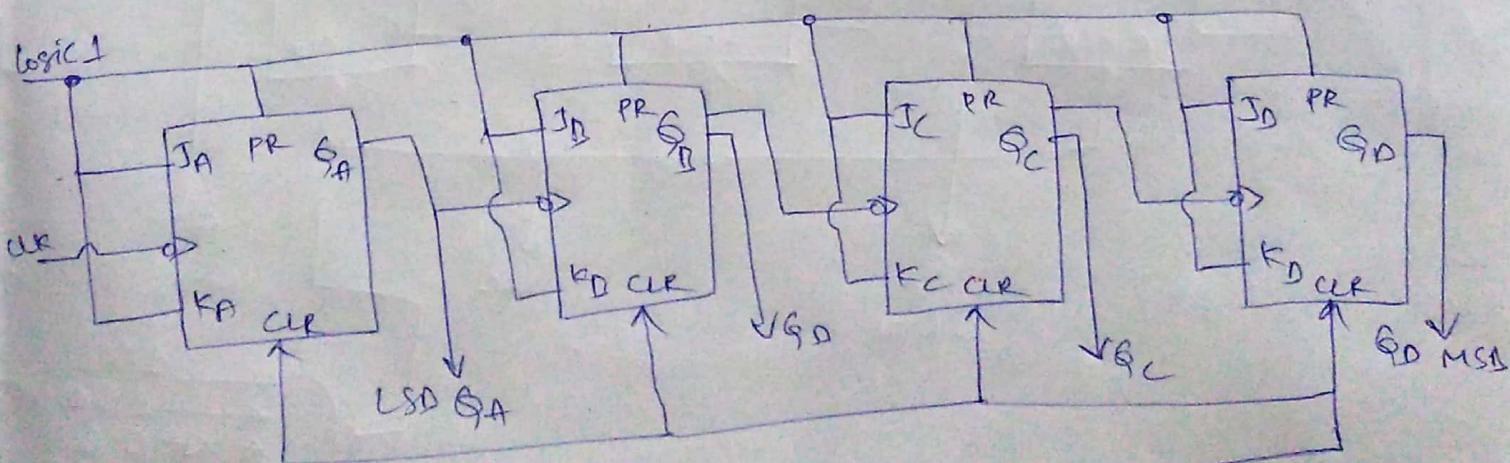
So to design BCD/decade ripple counter, we need 4 flipflops.

The state diagram for BCD/decimal ripple counter is,

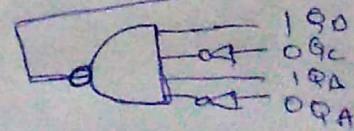


Now we have to design $2^4 = 16$ i.e. MOD 16 counter.

From that MOD-16 counter, we develop Mod 10/BCD counter. So, after count 9, the next state must be 0000. Such that we have to design for this again we have to use preset and clear inputs.



MOD-10 counter using MOD-16 counter.



UNIT-IV

Registers & Counters

The Shift Register:

The Shift Register is another type of sequential logic circuit that can be used for the storage or the transfer of binary data. This sequential device loads the data present on its inputs and then moves or “shifts” it to its output once every clock cycle, hence the name **Shift Register**.

A *shift register* basically consists of several single bit “D-Type Data Latches”, one for each data bit, either a logic “0” or a “1”, connected together in a serial type daisy-chain arrangement so that the output from one data latch becomes the input of the next latch and so on.

Data bits may be fed in or out of a shift register serially, that is one after the other from either the left or the right direction, or all together at the same time in a parallel configuration.

The number of individual data latches required to make up a single **Shift Register** device is usually determined by the number of bits to be stored with the most common being 8-bits (one byte) wide constructed from eight individual data latches.

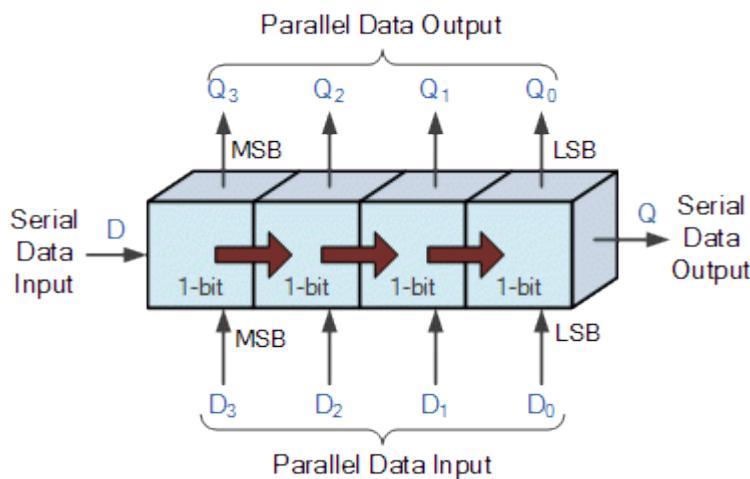
Shift Registers are used for data storage or for the movement of data and are therefore commonly used inside calculators or computers to store data such as two binary numbers before they are added together, or to convert the data from either a serial to parallel or parallel to serial format. The individual data latches that make up a single shift register are all driven by a common clock (Clk) signal making them synchronous devices.

Shift register IC’s are generally provided with a *clear* or *reset* connection so that they can be “SET” or “RESET” as required. Generally, shift registers operate in one of four different modes with the basic movement of data through a shift register being:

- Serial-in to Parallel-out (SIPO) - the register is loaded with serial data, one bit at a time, with the stored data being available at the output in parallel form.

- Serial-in to Serial-out (SISO) - the data is shifted serially “IN” and “OUT” of the register, one bit at a time in either a left or right direction under clock control.
- Parallel-in to Serial-out (PISO) - the parallel data is loaded into the register simultaneously and is shifted out of the register serially one bit at a time under clock control.
- Parallel-in to Parallel-out (PIPO) - the parallel data is loaded simultaneously into the register, and transferred together to their respective outputs by the same clock pulse.

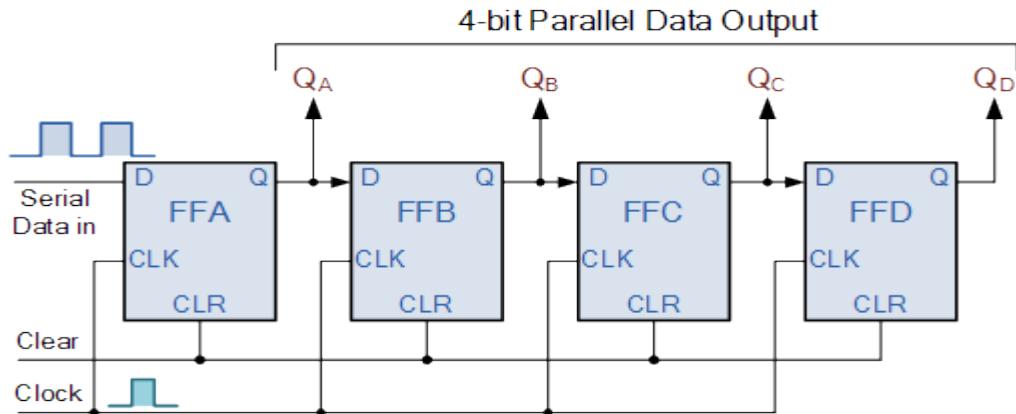
The effect of data movement from left to right through a shift register can be presented graphically as:



Also, the directional movement of the data through a shift register can be either to the left, (left shifting) to the right, (right shifting) left-in but right-out, (rotation) or both left and right shifting within the same register thereby making it *bidirectional*. In this tutorial it is assumed that all the data shifts to the right, (right shifting).

Serial-in to Parallel-out (SIPO) Shift Register

4-bit Serial-in to Parallel-out Shift Register



The operation is as follows. Lets assume that all the flip-flops (FFA to FFD) have just been RESET (CLEAR input) and that all the outputs Q_A to Q_D are at logic level “0” ie, no parallel data output.

If a logic “1” is connected to the DATA input pin of FFA then on the first clock pulse the output of FFA and therefore the resulting Q_A will be set HIGH to logic “1” with all the other outputs still remaining LOW at logic “0”. Assume now that the DATA input pin of FFA has returned LOW again to logic “0” giving us one data pulse or 0-1-0.

The second clock pulse will change the output of FFA to logic “0” and the output of FFB and Q_B HIGH to logic “1” as its input D has the logic “1” level on it from Q_A . The logic “1” has now moved or been “shifted” one place along the register to the right as it is now at Q_B .

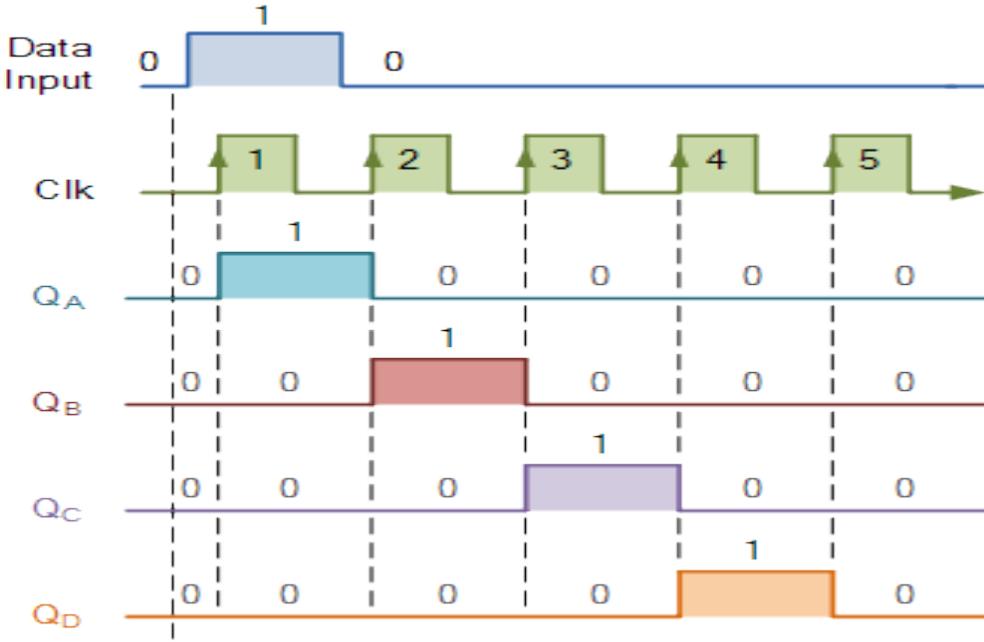
When the third clock pulse arrives this logic “1” value moves to the output of FFC (Q_C) and so on until the arrival of the fifth clock pulse which sets all the outputs Q_A to Q_D back again to logic level “0” because the input to FFA has remained constant at logic level “0”.

The effect of each clock pulse is to shift the data contents of each stage one place to the right, and this is shown in the following table until the complete data value of 0-0-0-1 is stored in the register. This data value can now be read directly from the outputs of Q_A to Q_D . Then the data has been converted from a serial data input signal to a parallel data output. The truth table and

following waveforms show the propagation of the logic “1” through the register from left to right as follows.

Basic Data Movement Through A Shift Register

Clock Pulse No	QA	QB	QC	QD
0	0	0	0	0
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1
5	0	0	0	0



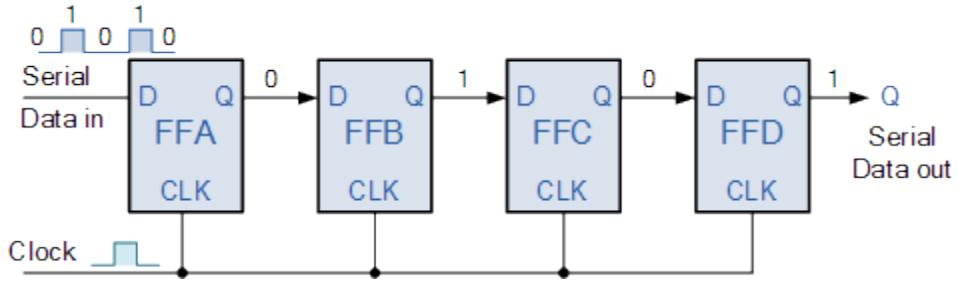
Note that after the fourth clock pulse has ended the 4-bits of data (0-0-0-1) are stored in the register and will remain there provided clocking of the register has stopped. In practice the input data to the register may consist of various combinations of logic “1” and “0”. Commonly available SIPO IC's include the standard 8-bit 74LS164 or the 74LS594.

Serial-in to Serial-out (SISO) Shift Register

This **shift register** is very similar to the SIPO above, except were before the data was read directly in a parallel form from the outputs Q_A to Q_D , this time the data is allowed to flow straight through the register and out of the other end. Since there is only one output, the DATA leaves the shift register one bit at a time in a serial pattern, hence the name **Serial-in to Serial-Out Shift Register or SISO**.

The SISO shift register is one of the simplest of the four configurations as it has only three connections, the serial input (SI) which determines what enters the left hand flip-flop, the serial output (SO) which is taken from the output of the right hand flip-flop and the sequencing clock signal (Clk). The logic circuit diagram below shows a generalized serial-in serial-out shift register.

4-bit Serial-in to Serial-out Shift Register

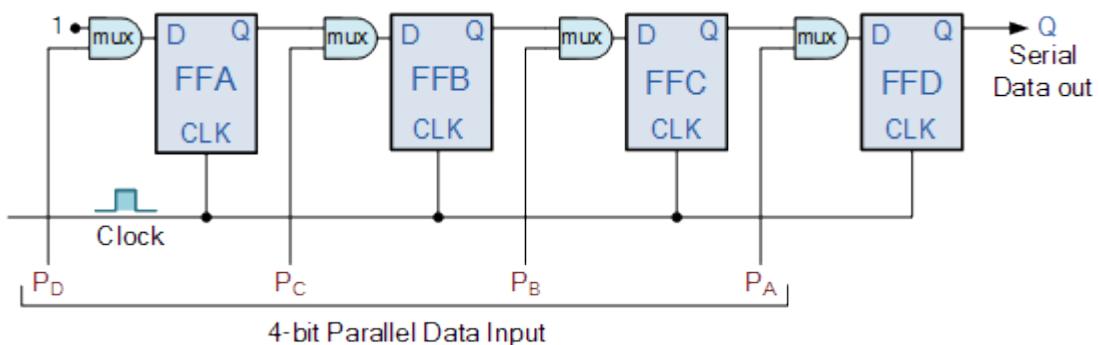


You may think what's the point of a SISO shift register if the output data is exactly the same as the input data. Well this type of **Shift Register** also acts as a temporary storage device or it can act as a time delay device for the data, with the amount of time delay being controlled by the number of stages in the register, 4, 8, 16 etc or by varying the application of the clock pulses. Commonly available IC's include the 74HC595 8-bit Serial-in to Serial-out Shift Register all with 3-state outputs.

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

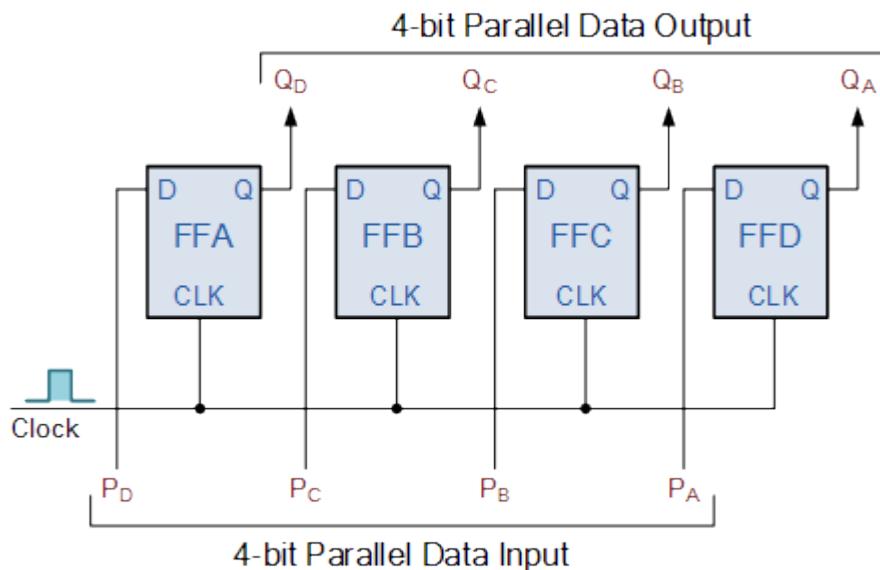


As this type of shift register converts parallel data, such as an 8-bit data word into serial format, it can be used to multiplex many different input lines into a single serial DATA stream which can be sent directly to a computer or transmitted over a communications line. Commonly available IC's include the 74HC166 8-bit Parallel-in/Serial-out Shift Registers.

Parallel-in to Parallel-out (PIPO) Shift Register

The final mode of operation is the Parallel-in to Parallel-out Shift Register. This type of shift register also acts as a temporary storage device or as a time delay device similar to the SISO configuration above. The data is presented in a parallel format to the parallel input pins P_A to P_D and then transferred together directly to their respective output pins Q_A to Q_D by the same clock pulse. Then one clock pulse loads and unloads the register. This arrangement for parallel loading and unloading is shown below.

4-bit Parallel-in to Parallel-out Shift Register



The PIPO shift register is the simplest of the four configurations as it has only three connections, the parallel input (PI) which determines what enters the flip-flop, the parallel output (PO) and the sequencing clock signal (Clk).

Similar to the Serial-in to Serial-out shift register, this type of register also acts as a temporary storage device or as a time delay device, with the amount of time delay being varied by the

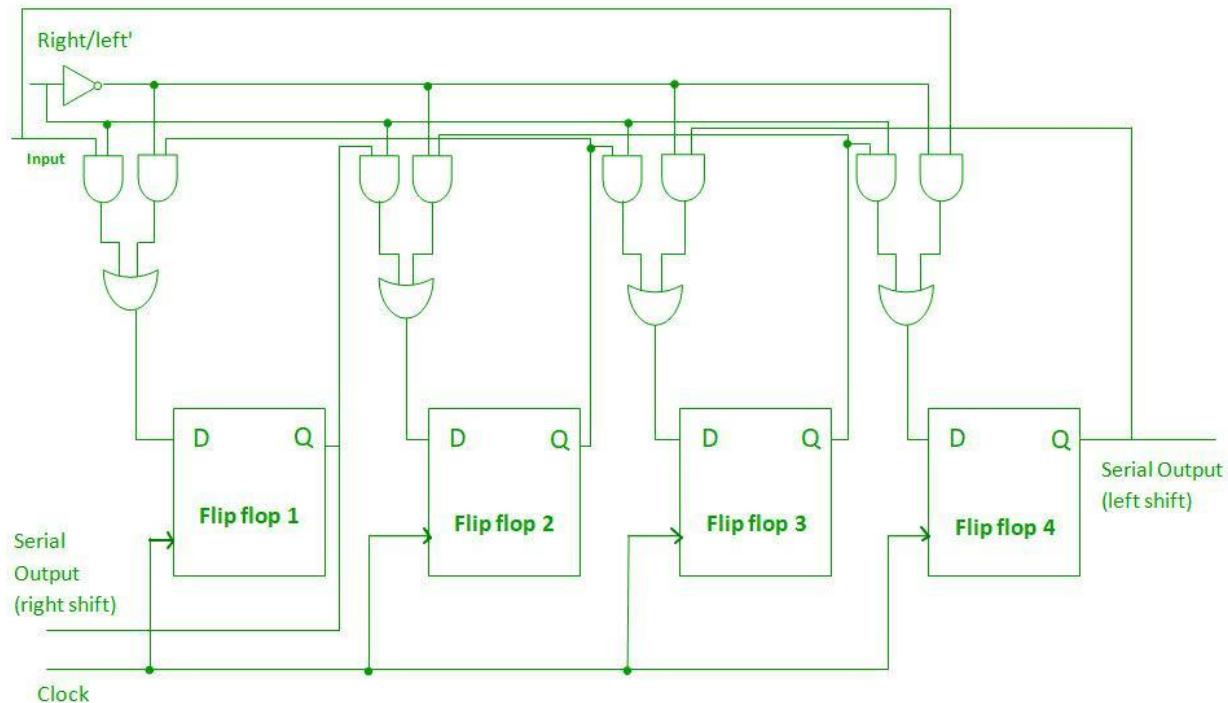
frequency of the clock pulses. Also, in this type of register there are no interconnections between the individual flip-flops since no serial shifting of the data is required.

Bidirectional Shift Register

If we shift a binary number to the left by one position, it is equivalent to multiplying the number by 2 and if we shift a binary number to the right by one position, it is equivalent to dividing the number by 2. To perform these operations we need a register which can shift the data in either direction.

Bidirectional shift registers are the registers which are capable of shifting the data either right or left depending on the mode selected. If the mode selected is 1(high), the data will be shifted towards the right direction and if the mode selected is 0(low), the data will be shifted towards the left direction.

The logic circuit given below shows a Bidirectional shift register. The circuit consists of four D flip-flops which are connected. The input data is connected at two ends of the circuit and depending on the mode selected only one gate is in the active state.



Applications of shift Registers :

- The shift registers are used for temporary data storage.
- The shift registers are also used for data transfer and data manipulation.
- The serial-in serial-out and parallel-in parallel-out shift registers are used to produce time delay to digital circuits.
- The serial-in parallel-out shift register is used to convert serial data into parallel data thus they are used in communication lines where demultiplexing of a data line into several parallel line is required.
- A Parallel in Serial out shift register us used to convert parallel data to serial data

COUNTERS:

A **Counter** is a device which stores (and sometimes displays) the number of times a particular event or process has occurred, often in relationship to a clock signal. Counters are used in digital electronics for counting purpose, they can count specific event happening in the circuit. For example, in UP counter a counter increases count for every rising edge of clock. Not only counting, a counter can follow the certain sequence based on our design like any random sequence 0,1,3,2... .They can also be designed with the help of flip flops.

Counter Classification

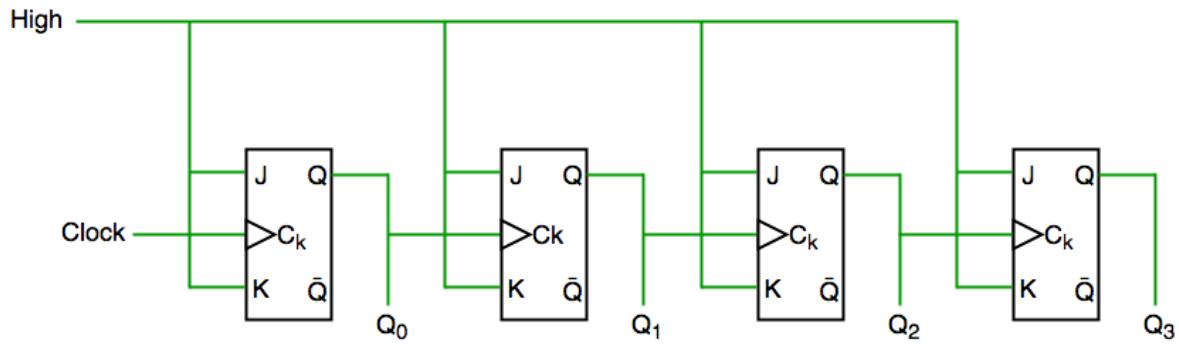
Counters are broadly divided into two categories

1. Asynchronous counter
2. Synchronous counter

1. Asynchronous Counter

In asynchronous counter we don't use universal clock, only first flip flop is driven by main clock and the clock input of rest of the following counters is driven by output of previous flip flops.

We can understand it by following diagram-

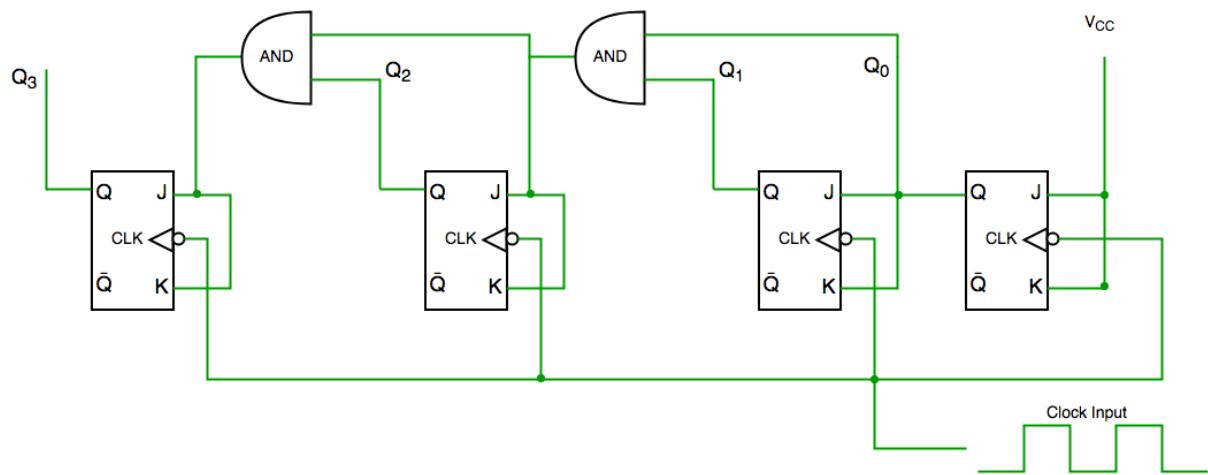


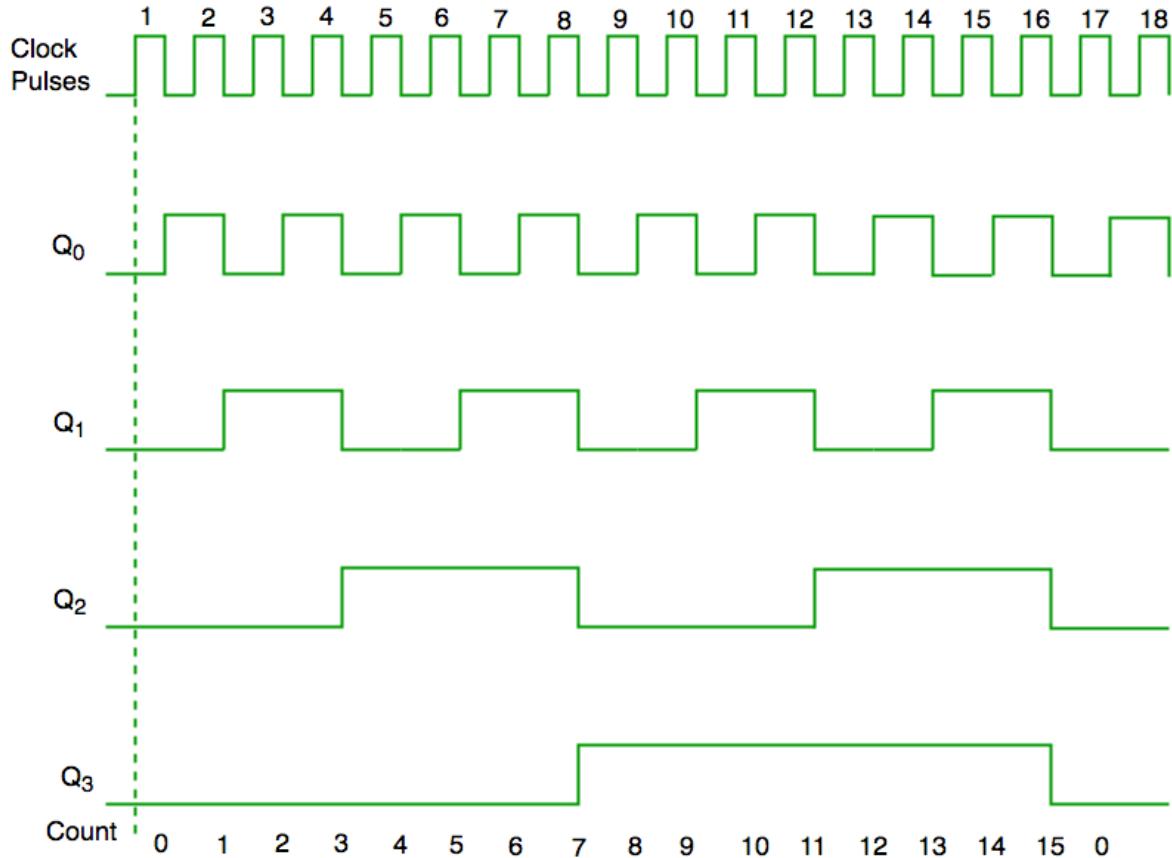
(b) Timing Diagram

It is evident from timing diagram that Q_0 is changing as soon as the rising edge of clock pulse is encountered, Q_1 is changing when rising edge of Q_0 is encountered(because Q_0 is like clock pulse for second flip flop) and so on. In this way ripples are generated through Q_0, Q_1, Q_2, Q_3 hence it is also called **RIPPLE counter**.

2. Synchronous Counter

Unlike the asynchronous counter, synchronous counter has one global clock which drives each flip flop so output changes in parallel. The one advantage of synchronous counter over asynchronous counter is, it can operate on higher frequency than asynchronous counter as it does not have cumulative delay because of same clock is given to each flip flop.





Timing diagram synchronous counter

From circuit diagram we see that Q_0 bit gives response to each falling edge of clock while Q_1 is dependent on Q_0 , Q_2 is dependent on Q_1 and Q_0 , Q_3 is dependent on Q_2,Q_1 and Q_0 .

What is Ring Counter & Johnson Counter?

A **ring counter**, a type of counter in which the output of the last [flip-flop](#) is connected as an input to the first flip-flop is known as a *Ring counter*. The input is shifted between the flip-flops in a ring shape which is why it is known as a Ring counter.

A Ring counter is a synchronous counter. the synchronous counter has a common clock signal that triggers all the Flip-flops at the same time. Ring counter consists of [D-flip flops](#) connected in cascade setup with the output of last Flip-flop connected to the input of first Flip-flop. Each flip-flop constitutes a stage.

Types of Ring Counters

There are two types of the Ring counter.

- A Typical Ring Counter
- Johnson counter

Typical Ring Counter

A typical 4-bit ring counter is made of D-flip flops or JK-flip flop connected in cascade with the non-complemented output of the last stage connected as an input to the first stage. Ring counter has Mod = n ‘n’ is the number of bits. It means 4-bit ring counter has 4 states.

Ring Counter Truth Table

Consider Q_A , Q_B , Q_C , Q_D as the 4 bits of the ring counter. The truth table for 4-bit ring counter is given below.

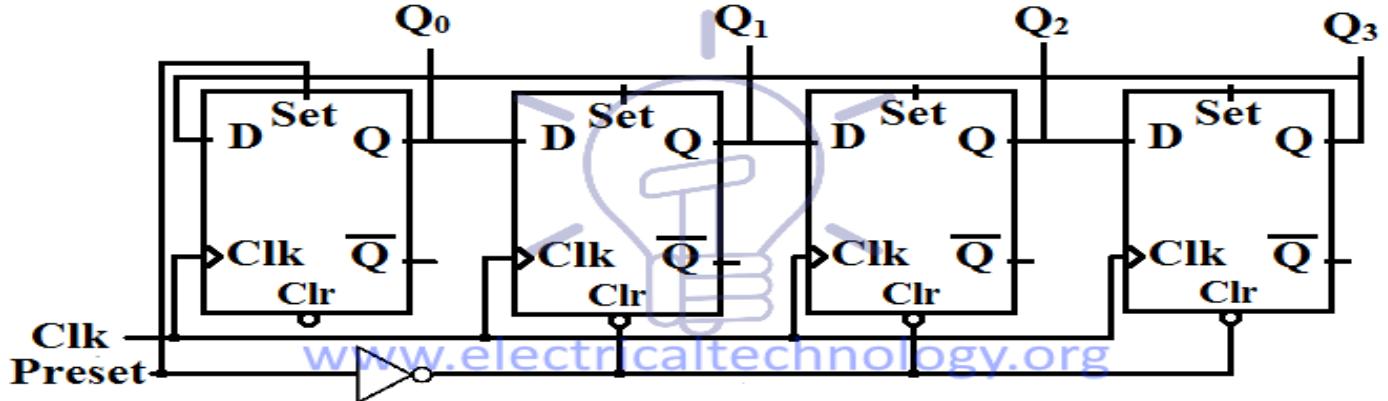
States	Q_A	Q_B	Q_C	Q_D
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1

In ring counter, logic ‘1’ flows through all stages of the counter. In each state, it flows one bit to the right. When it reaches stage 4, it circulates back to stage1 of the counter.

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:



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.

Advantages / Disadvantages of Ring Counter

Advantages

- It doesn't need a decoder (i.e. It is a self decoding circuit)
- It can be implemented using JK and D flip-flops.

Disadvantages

- In ring counter, only 4 of the 15 states are being utilized.

What is Johnson Counter

“Johnson counter” or “twisted ring counter” is a type of synchronous ring counter in which the complemented output of the flip-flop is connected with the input of the first flip-flop. Johnson counter can be made with D-flip flops or [JK-flip flops](#) in cascade setup.

The Mod of Johnson counter is ‘ $2n$ ’, n is the bit size of the counter. Mod is the maximum number of states a counter can obtain.

Johnson Counter Truth Table

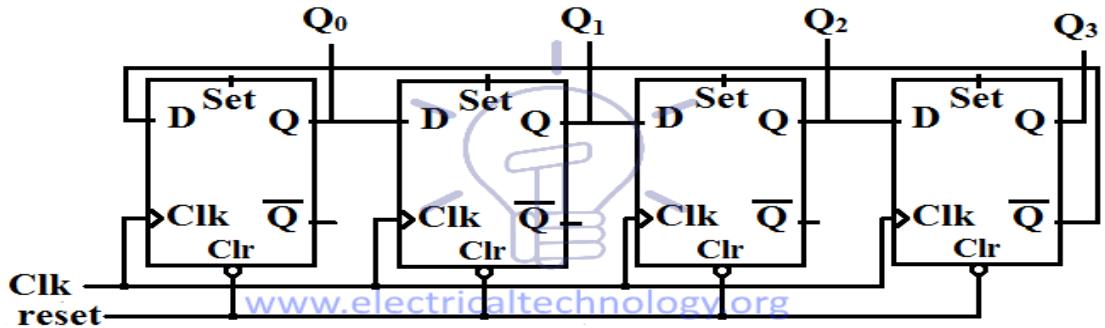
Consider a 4-bit Johnson counter with Q_A , Q_B , Q_C , Q_D as the output of 4 stages of the counter.

The truth table of the 4-bit Johnson counter is given below;

States	Q_A	Q_B	Q_C	Q_D
1	0	0	0	0
2	1	0	0	0
3	1	1	0	0
4	1	1	1	0
5	1	1	1	1
6	0	1	1	1
7	0	0	1	1
8	0	0	0	1

Johnson Counter Schematic Design

The schematic of 4-bit Johnson counter consists of 4 D-flip flops or 4 JK-flip flops. These flip-flops are connected with each other in cascade setup. The output of each flip-flop is connected with the input of the succeeding flip-flop. The complemented output of the last flip-flop is connected with the input of the first flip-flop. The Same clock input is connected with all flip-flops. There is clear input for resetting the state to default 0000. Johnson counter's schematic design is given below.



Working of Johnson Counter

The default state of Johnson counter is 0000 thus before starting the clock input we need to clear the counter using clear input.

Whenever a clock edge hits the counter the output of each flip-flop will transfer to the next stage (flip-flop) but the inverted output of the last flip-flop will shift to the first stage making the state 1000.

Upon next clock cycle, another ‘1’ will stack in from the left side as the inverted output of the last stage will be shifted to the first stage.

On next clock cycle, another ‘1’ will add in from left until the state becomes 1111.

Now that the last flip-flop’s output is ‘1’, the next clock cycle will shift the invert of the last flip-flop which is ‘0’ into the first flip-flop. It will result in stacking ‘0’ from the left side. This stacking of the first 0 will make the state 1111 into 0111.

The next coming clock cycles will stack in 0’s from the left making the states 0011, 0001 & 0000 with each clock cycle. Eventually, it reaches its default state and it starts from the beginning again.

Advantages / Disadvantages of Johnson Counter

Advantages

- More outputs as compared to ring counter.
- It has same number of flip flop but it can count twice the number of states the ring counter can count.
- It count the data in a continuous loop
- It only needs half the number of flip-flops compared to the standard ring counter for the same MOD

Disadvantages

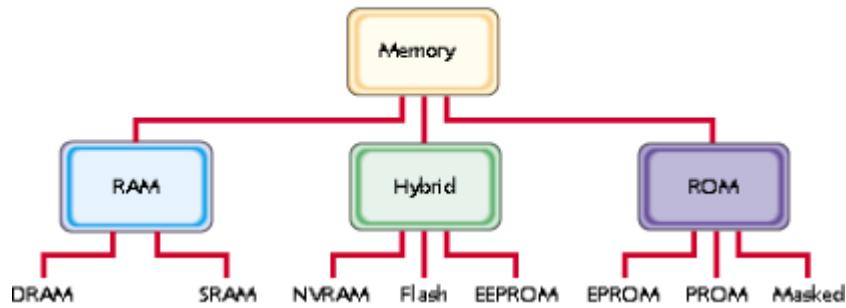
- Only 8 of the 15 states are being used.
- It doesn't count in a binary sequence.

Applications of Ring & Johnson Counters

- a. Johnson counter divides a clock signal's frequency by ' 2^n '. n is the bit size of the counter. Johnson counter uses less number of flip-flops compare to a typical ring counter.
- b. 2-stage Johnson counter, also known as quadrature oscillator produce 4 outputs with 90-degree phase shift. It can easily drive a 2-phase stepper motor.
- c. 3-stage Johnson counter is used as 3-phase square wave generator having 120-degree phase shift between them.
- d. 5-stage Johnson counter is used as decade counter or decade divider.

Other applications of Johnson and Ring counters as follow:

- Frequency counters
- As object counters
- Parallel to serial data conversion logic circuits
- Analog to digital convertors.
- Digital clocks
- Digital triangular wave generator.
- Generating staircase voltage
- Frequency divider circuits of the clock signals. (Where the Input frequency divided by 2)
- Synchronous decade counter or divider circuit
- Timers and Rate measurement. (Time circuits, Washing machines, Alarm clock etc)
- As a 3 phase square wave generator which produces 1200 phase shift(3 stage Johnson counter)



Introduction to Memory Types

Many types of memory devices are available for use in modern computer systems. As an embedded software engineer, you must be aware of the differences between them and understand how to use each type effectively. In our discussion, we will approach these devices from the software developer's perspective. Keep in mind that the development of these devices took several decades and that their underlying hardware differs significantly. The names of the memory types frequently reflect the historical nature of the development process and are often more confusing than insightful. [Figure 1](#) classifies the memory devices we'll discuss as RAM, ROM, or a hybrid of the two.

Types of RAM

The RAM family includes two important memory devices: static RAM (SRAM) and dynamic RAM (DRAM). The primary difference between them is the lifetime of the data they store. SRAM retains its contents as long as electrical power is applied to the chip. If the power is turned off or lost temporarily, its contents will be lost forever. DRAM, on the other hand, has an extremely short data lifetime—typically about four milliseconds. This is true even when power is applied constantly.

In short, SRAM has all the properties of the memory you think of when you hear the word RAM. Compared to that, DRAM seems kind of useless. By itself, it is. However, a simple piece of hardware called a DRAM controller can be used to make DRAM behave more like SRAM. The job of the DRAM controller is to periodically refresh the data stored in the DRAM. By refreshing the data before it expires, the contents of memory can be kept alive for as long as they are needed. So DRAM is as useful as SRAM after all.

When deciding which type of RAM to use, a system designer must consider access time and cost. SRAM devices offer extremely fast access times (approximately four times faster than DRAM) but are much more expensive to produce. Generally, SRAM is used only where access speed is extremely important. A lower cost-per-byte makes DRAM attractive whenever large amounts of RAM are required. Many embedded systems include both types: a small block of SRAM (a few kilobytes) along a critical data path and a much larger block of DRAM for everything else.

Types of ROM

Memories in the ROM family are distinguished by the methods used to write new data to them (usually called programming), and the number of times they can be rewritten. This classification reflects the evolution of ROM devices from hardwired to programmable to erasable-and-programmable. A common feature of all these devices is their ability to retain data and programs forever, even during a power failure.

The very first ROMs were hardwired devices that contained a preprogrammed set of data or instructions. The contents of the ROM had to be specified before chip production, so

the actual data could be used to arrange the transistors inside the chip. Hardwired memories are still used, though they are now called "masked ROMs" to distinguish them from other types of ROM. The primary advantage of a masked ROM is its low production cost. Unfortunately, the cost is low only when large quantities of the same ROM are required.

One step up from the masked ROM is the PROM (programmable ROM), which is purchased in an unprogrammed state. If you were to look at the contents of an unprogrammed PROM, you would see that the data is made up entirely of 1's. The process of writing your data to the PROM involves a special piece of equipment called a device programmer. The device programmer writes data to the device one word at a time by applying an electrical charge to the input pins of the chip. Once a PROM has been programmed in this way, its contents can never be changed. If the code or data stored in the PROM must be changed, the current device must be discarded. As a result, PROMs are also known as one-time programmable (OTP) devices.

An EPROM (erasable-and-programmable ROM) is programmed in exactly the same manner as a PROM. However, EPROMs can be erased and reprogrammed repeatedly. To erase an EPROM, you simply expose the device to a strong source of ultraviolet light. (A window in the top of the device allows the light to reach the silicon.) By doing this, you essentially reset the entire chip to its initial-unprogrammed-state. Though more expensive than PROMs, their ability to be reprogrammed makes EPROMs an essential part of the software development and testing process.

Hybrid types

As memory technology has matured in recent years, the line between RAM and ROM has blurred. Now, several types of memory combine features of both. These devices do not belong to either group and can be collectively referred to as hybrid memory devices. Hybrid memories can be read and written as desired, like RAM, but maintain their contents without electrical power, just like ROM. Two of the hybrid devices, EEPROM and flash, are descendants of ROM devices. These are typically used to store code. The third hybrid, NVRAM, is a modified version of SRAM. NVRAM usually holds persistent data.

EEPROMs are electrically-erasable-and-programmable. Internally, they are similar to EPROMs, but the erase operation is accomplished electrically, rather than by exposure to ultraviolet light. Any byte within an EEPROM may be erased and rewritten. Once written, the new data will remain in the device forever-or at least until it is electrically erased. The primary tradeoff for this improved functionality is higher cost, though write cycles are also significantly longer than writes to a RAM. So you wouldn't want to use an EEPROM for your main system memory.

Flash memory combines the best features of the memory devices described thus far. Flash memory devices are high density, low cost, nonvolatile, fast (to read, but not to write), and electrically reprogrammable. These advantages are overwhelming and, as a direct result, the use of flash memory has increased dramatically in embedded systems. From a software viewpoint, flash and EEPROM technologies are very similar. The major difference is that flash devices can only be erased one sector at a time, not byte-by-byte. Typical sector sizes are in the range 256 bytes to 16KB. Despite this disadvantage, flash is much more popular than EEPROM and is rapidly displacing many of the ROM devices as well.

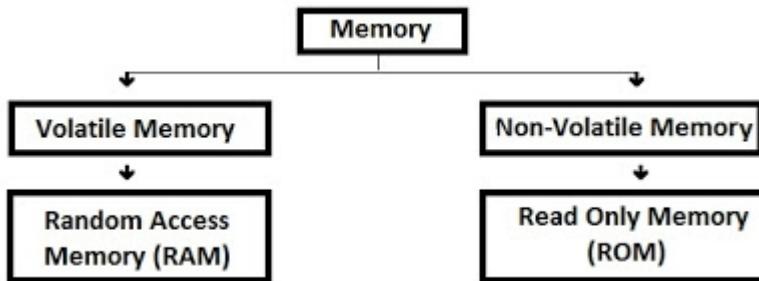
The third member of the hybrid memory class is NVRAM (non-volatile RAM). Nonvolatility is also a characteristic of the ROM and hybrid memories discussed previously. However, an NVRAM is physically very different from those devices. An NVRAM is usually just an SRAM with a battery backup. When the power is turned on, the NVRAM operates just like any other SRAM. When the power is turned off, the NVRAM draws just enough power from the battery to retain its data. NVRAM is fairly common in embedded systems. However, it is expensive-even more expensive than SRAM, because of the battery-so its applications are typically limited to the storage of a few hundred bytes of system-critical information that can't be stored in any better way.

[Table 1](#) summarizes the features of each type of memory discussed here, but keep in mind that different memory types serve different purposes. Each memory type has its strengths and weaknesses. Side-by-side comparisons are not always effective.

COMPUTER MEMORY

Memory is an essential element of a computer. Without its memory, a computer is of hardly any use. Memory plays an important role in saving and retrieving data. The performance of the computer system depends upon the size of the memory. Memory is of following types:

1. Primary Memory / Volatile Memory.
2. Secondary Memory / Non Volatile Memory.



1. Primary Memory / Volatile Memory: Primary Memory is internal memory of the computer. RAM AND ROM both form part of primary memory. The primary memory provides main working space to the computer. The following terms comes under primary memory of a computer are discussed below:

- **Random Access Memory (RAM):** The primary storage is referred to as random access memory (RAM) because it is possible to randomly select and use any location of the memory directly store and retrieve data. It takes same time to any address of the memory as the first address. It is also called read/write memory. The storage of data and instructions inside the primary storage is temporary. It disappears from RAM as soon as the power to the computer is switched off. The memories, which lose their content on failure of power supply, are known as volatile memories .So now we can say that RAM is volatile memory.
- **Read Only Memory (ROM):** There is another memory in computer, which is called Read Only Memory (ROM). Again it is the ICs inside the PC that form the ROM. The storage of program and data in the ROM is permanent. The ROM stores some standard processing programs supplied by the manufacturers to operate the personal computer. The ROM can only be read by the CPU but it cannot be changed. The basic input/output program is stored in the ROM that examines and initializes various equipment attached to the PC when the power switch is ON. The memories, which do

not lose their content on failure of power supply, are known as non-volatile memories. ROM is non-volatile memory.

- **PROM:** There is another type of primary memory in computer, which is called Programmable Read Only Memory (PROM). You know that it is not possible to modify or erase programs stored in ROM, but it is possible for you to store your program in PROM chip. Once the programmers' are written it cannot be changed and remain intact even if power is switched off. Therefore programs or instructions written in PROM or ROM cannot be erased or changed.
- **EPROM:** This stands for Erasable Programmable Read Only Memory, which overcome the problem of PROM & ROM. EPROM chip can be programmed time and again by erasing the information stored earlier in it. Information stored in EPROM exposing the chip for some time ultraviolet light and it erases chip is reprogrammed using a special programming facility. When the EPROM is in use information can only be read.
- **Cache Memory:** The speed of CPU is extremely high compared to the access time of main memory. Therefore the performance of CPU decreases due to the slow speed of main memory. To decrease the mismatch in operating speed, a small memory chip is attached between CPU and Main memory whose access time is very close to the processing speed of CPU. It is called CACHE memory. CACHE memories are accessed much faster than conventional RAM. It is used to store programs or data currently being executed or temporary data frequently used by the CPU. So each memory makes main memory to be faster and larger than it really is. It is also very expensive to have bigger size of cache memory and its size is normally kept small.
- **Registers:** The CPU processes data and instructions with high speed; there is also movement of data between various units of computer. It is necessary to transfer the processed data with high speed. So the computer uses a number of special memory units called registers. They are not part of the main memory but they store data or information temporarily and pass it on as directed by the control unit.

2. Secondary Memory / Non-Volatile Memory: Secondary memory is external and permanent in nature. The secondary memory is concerned with magnetic memory. Secondary memory can be stored on storage media like floppy disks, magnetic disks, magnetic tapes, This memory can also be stored optically on Optical disks - CD-ROM. The following terms comes under secondary memory of a computer are discussed below:

- **Magnetic Tape:** Magnetic tapes are used for large computers like mainframe computers where large volume of data is stored for a longer time. In PC also you can use tapes in the form of cassettes. The cost of storing data in tapes is inexpensive. Tapes consist of magnetic materials that store data permanently. It can be 12.5 mm to 25 mm wide plastic film-type and 500 meter to 1200 meter long which is coated with magnetic material. The deck is connected to the central processor and information is

fed into or read from the tape through the processor. It's similar to cassette tape recorder.

- **Magnetic Disk:** You might have seen the gramophone record, which is circular like a disk and coated with magnetic material. Magnetic disks used in computer are made on the same principle. It rotates with very high speed inside the computer drive. Data is stored on both the surface of the disk. Magnetic disks are most popular for direct access storage device. Each disk consists of a number of invisible concentric circles called tracks. Information is recorded on tracks of a disk surface in the form of tiny magnetic spots. The presence of a magnetic spot represents one bit and its absence represents zero bit. The information stored in a disk can be read many times without affecting the stored data. So the reading operation is non-destructive. But if you want to write a new data, then the existing data is erased from the disk and new data is recorded. For Example-Floppy Disk.

- **Optical Disk:** With every new application and software there is greater demand for memory capacity. It is the necessity to store large volume of data that has led to the development of optical disk storage medium. Optical disks can be divided into the following categories:
 1. **Compact Disk/ Read Only Memory (CD-ROM)**
 2. **Write Once, Read Many (WORM)**
 3. **Erasable Optical Disk**

A memory is just like a human brain. It is used to store data and instruction. Computer memory is the storage space in computer where data is to be processed and instructions required for processing are stored.

The memory is divided into large number of small parts. Each part is called cell. Each location or cell has a unique address which varies from zero to memory size minus one.

For example if computer has 64k words, then this memory unit has $64 * 1024 = 65536$ memory location. The address of these locations varies from 0 to 65535.

Memory is primarily of two types

- **Internal Memory** - cache memory and primary/main memory
- **External Memory** - magnetic disk / optical disk etc.

Characteristics of Memory Hierarchy are following when we go from top to bottom.

- Capacity in terms of storage increases.
- Cost per bit of storage decreases.
- Frequency of access of the memory by the CPU decreases.
- Access time by the CPU increases

RAM

A RAM constitutes the internal memory of the CPU for storing data, program and program result. It is read/write memory. It is called random access memory (RAM).

Since access time in RAM is independent of the address to the word that is, each storage location inside the memory is as easy to reach as other location & takes the same amount of time. We can reach into the memory at random & extremely fast but can also be quite expensive.

RAM is volatile, i.e. data stored in it is lost when we switch off the computer or if there is a power failure. Hence a backup uninterruptible power system(UPS) is often used with computers. RAM is small , both in terms of its physical size and in the amount of data it can hold.

RAM is of two types

- Static RAM (SRAM)
- Dynamic RAM (DRAM)

Static RAM (SRAM)

The word **static** indicates that the memory retains its contents as long as power remains applied. However, data is lost when the power gets down due to volatile nature. SRAM chips use a matrix of 6-transistors and no capacitors. Transistors do not require power to prevent leakage, so SRAM need not have to be refreshed on a regular basis.

Because of the extra space in the matrix, SRAM uses more chips than DRAM for the same amount of storage space, thus making the manufacturing costs higher.

Static RAM is used as cache memory needs to be very fast and small.

Dynamic RAM (DRAM)

DRAM, unlike SRAM, must be continually **refreshed** in order for it to maintain the data. This is done by placing the memory on a refresh circuit that rewrites the data several hundred times per second. DRAM is used for most system memory because it is cheap and small. All DRAMs are made up of memory cells. These cells are composed of one capacitor and one transistor.

ROM

ROM stands for Read Only Memory. The memory from which we can only read but cannot write on it. This type of memory is non-volatile. The information is stored permanently in such memories during manufacture.

A ROM, stores such instruction as are required to start computer when electricity is first turned on, this operation is referred to as bootstrap. ROM chip are not only used in the computer but also in other electronic items like washing machine and microwave oven.

Following are the varioys types of ROM

MROM (Masked ROM)

The very first ROMs were hard-wired devices that contained a pre-programmed set of data or instructions. These kind of ROMs are known as masked ROMs. It is inexpensive ROM.

PROM (Programmable Read only Memory)

PROM is read-only memory that can be modified only once by a user. The user buys a blank PROM and enters the desired contents using a PROM programmer. Inside the PROM chip there are small fuses which are burnt open during programming. It can be programmed only once and is not erasable.

EPROM(Erasable and Programmable Read Only Memory)

The EPROM can be erased by exposing it to ultra-violet light for a duration of upto 40 minutes. Usually, a EPROM eraser achieves this function. During programming an electrical charge is trapped in an insulated gate region. The charge is retained for more than ten years because the charge has no leakage path. For erasing this charge, ultra-violet light is passed through a quartz crystal window(lid). This exposure to ultra-violet light dissipates the charge. During normal use the quartz lid is sealed with a sticker.

EEPROM(Electrically Erasable and Programmable Read Only Memory)

The EEPROM is programmed and erased electrically. It can be erased and reprogrammed about ten thousand times. Both erasing and programming take about 4 to 10 ms (millisecond). In EEPROM, any location can be selectively erased and programmed. EEPROMs can be erased one byte at a time, rather than erasing the entire chip. Hence, the process of re-programming is flexible but slow.

Serial Access Memory

Sequential access means the system must search the storage device from the beginning of the memory address until it finds the required piece of data. Memory device which supports such access is called a Sequential Access Memory or Serial Access Memory. Magnetic tape is an example of serial access memory.

Direct Access Memory

Direct access memory or Random Access Memory, refers to condition in which a system can go directly to the information that the user wants. Memory device which supports such access is called a Direct Access Memory. Magnetic disk, optical disks are an examples of direct access memory.

Cache Memory

Cache memory is a very high speed semiconductor memory which can speed up CPU. It acts as a buffer between the CPU and main memory. It is used to hold those parts of data and program which are most frequently used by CPU. The parts of data and programs are transferred from disk to cache memory by operating system, from where CPU can access them.

Advantages

- Cache memory is faster than main memory.
- It consumes less access time as compared to main memory.
- It stores the program that can be executed within a short period of time.
- It stores data for temporary use.

Disadvantages

- Cache memory has limited capacity.
- It is very expensive.

Virtual memory is a technique that allows the execution of processes which are not completely available in memory. The main visible advantage of this scheme is that programs can be larger than physical memory. Virtual memory is the separation of user logical memory from physical memory.

This separation allows an extremely large virtual memory to be provided for programmers when only a smaller physical memory is available. Following are the situations, when entire program is not required to be loaded fully in main memory.

- User written error handling routines are used only when an error occurred in the data or computation.
- Certain options and features of a program may be used rarely.
- Many tables are assigned a fixed amount of address space even though only a small amount of the table is actually used.
- The ability to execute a program that is only partially in memory would counter many benefits.
- Less number of I/O would be needed to load or swap each user program into memory.
- A program would no longer be constrained by the amount of physical memory that is available.
- Each user program could take less physical memory, more programs could be run the same time, with a corresponding increase in CPU utilization and throughput.

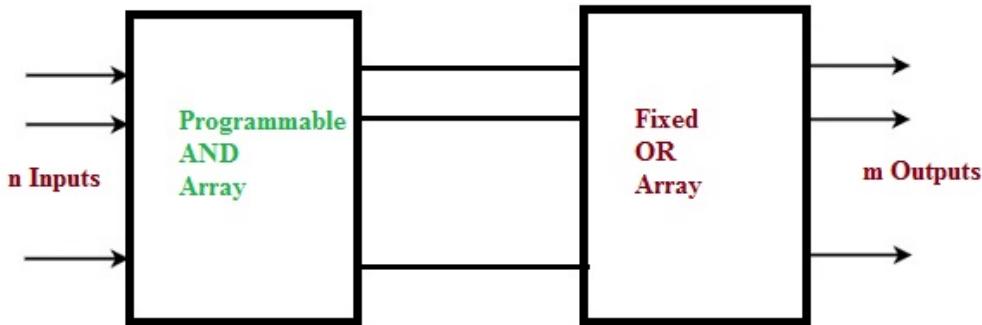
Auxiliary Memory

Auxiliary memory is much larger in size than main memory but is slower. It normally stores system programs, instruction and data files. It is also known as secondary memory. It can also be used as an overflow/virtual memory in case the main memory capacity has been exceeded. Secondary memories can not be accessed directly by a processor. First the data / information of auxiliary memory is transferred to the main memory and then that information can be accessed by the CPU. Characteristics of Auxiliary Memory are following

- **Non-volatile memory** - Data is not lost when power is cut off.
- **Reusable** - The data stays in the secondary storage on permanent basis until it is not overwritten or deleted by the user.
- **Reliable** - Data in secondary storage is safe because of high physical stability of secondary storage device.
- **Convenience** - With the help of a computer software, authorised people can locate and access the data quickly.
- **Capacity** - Secondary storage can store large volumes of data in sets of multiple disks.
- **Cost** - It is much less expensive to store data on a tape or disk than primary memory.

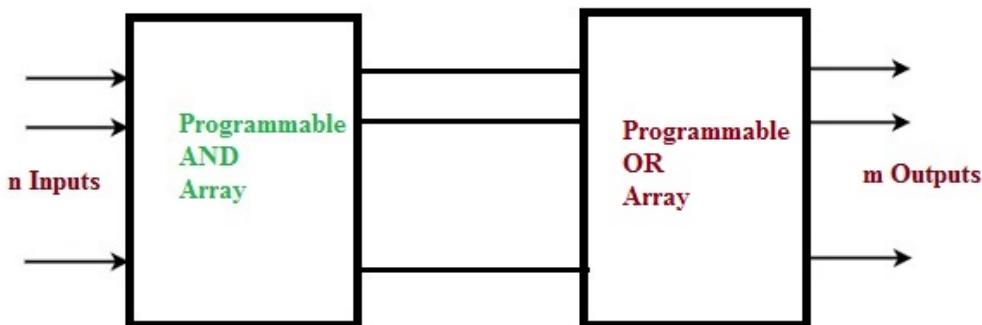
What are PAL and PLA?

Both **Programmable Array Logic** and **Programmable Logic Array** are types of PLDs (programmable logic devices), and these are mainly used for designing combination logic mutually by sequential logic. The main difference among these two is that PAL can be designed with a collection of AND gates and fixed collection of OR gates whereas PLA can be designed with a programmable array of AND although a fixed collection of OR gate. A programmable logic device offers a simple as well as flexible logic circuit designing.



Programmable Array Logic

Previous to programmable logic devices, the **combinational logic circuits** can be designed with multiplexers, and these circuits were rigid as well as compound, then PLDs are developed. The initial programmable logic device was ROM, but it was not successful due to the hardware wastage issues as well as exponential growth enhancement in the every hardware application. To overcome this issue, PAL and PLA were used. These two are programmable, and efficiently uses the hardware.



Programmable Logic Array

Design of Programmable Array Logic (PAL)

The **definition of term PAL or Programmable Array Logic** is one type of PLD which is known as Programmable Logic Device circuit, and working of this PAL is the same as the PLA. The designing of the

programmable array logic can be done with fixed OR gates as well as programmable AND gates. By using this we can implement two easy functions wherever the associates AND gates with each OR gate denote the highest number of product conditions that can be produced in the form of **SOP (sum of product)** of an exact function.

As the logic gates like AND is connected continually toward the OR gates, and that indicates that the produced product term is not distributed with the output functions. The major notion behind PLD development is to fabricate a compound Boolean logic onto a single chip by removing the defective wiring, avoiding the logic design, as well as decreasing the consumption of power.

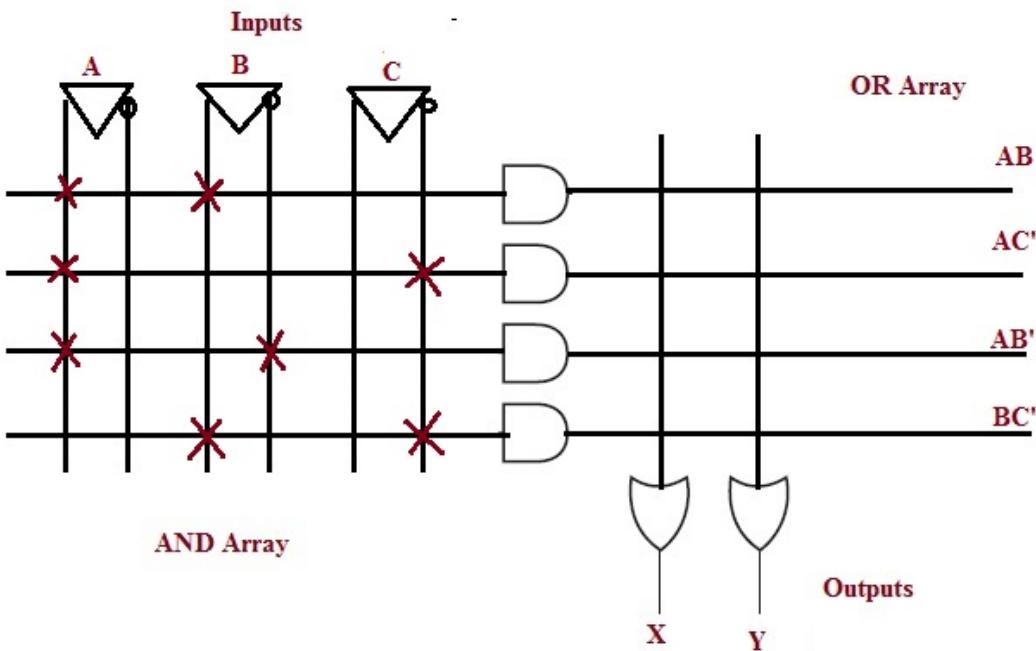
Example of PAL

Implement the following **Boolean expression** with the help of **programmable array logic (PAL)**

$$X = AB + AC'$$

$$Y = AB' + BC'$$

The above given two **Boolean functions** are in the form of **SOP (sum of products)**. The product terms present in the Boolean expressions are X & Y, and one product term that is AC' is common in every equation. So, the total required logic gates for generating the above two equations is AND gates-4 OR programmable gates-2. The equivalent PAL logic diagram is shown below.



PAL Logic Circuit

The AND gates which are programmable have the right of entry for normal as well as complemented variable inputs. In the above logic diagram, the available inputs for each AND gate are A, A', B, B', C, C'. So, in order to generate a single product term with every AND gate, the program is required.

All the product terms are obtainable at the inputs of an each OR gate. Here, the programmable connections on the logic gate can be denoted with the symbol 'X'.

Here, the OR gate inputs are fixed. Thus, the required product terms are associated with each OR gate inputs. As a result, these gates will generate particular Boolean equations. The '·' The symbol represents permanent connections.

Design of Programmable Logic Array (PLA)

The definition of term PLA presents the Boolean function in the form of a sum of product (SOP). The designing of this programmable logic array can be done using the logic gates like AND, OR, and NOT by fabricating on the chip, that makes every input as well as its compliment obtainable toward every AND gate.

An every AND gate's output is connected to the every OR gate. Finally, the output of the OR gate generates the output of the chip. Thus, this is how an appropriate association is finished to use the expressions of the sum of the product. In the programmable logic array, the connections of logic gates like AND & OR are programmable. PLA is expensive and difficult to compare with PAL. The PAL uses two dissimilar developed methods can be used for a programmable logic array for enhancing the effortlessness of programming. In this kind of method, every connection can be done using a fuse on each intersection point wherever the unnecessary connections can be detached by the fuse blowing. The

final technique engages the making of connection while the process of the fabrication using the suitable cover offered for the precise interconnection model.

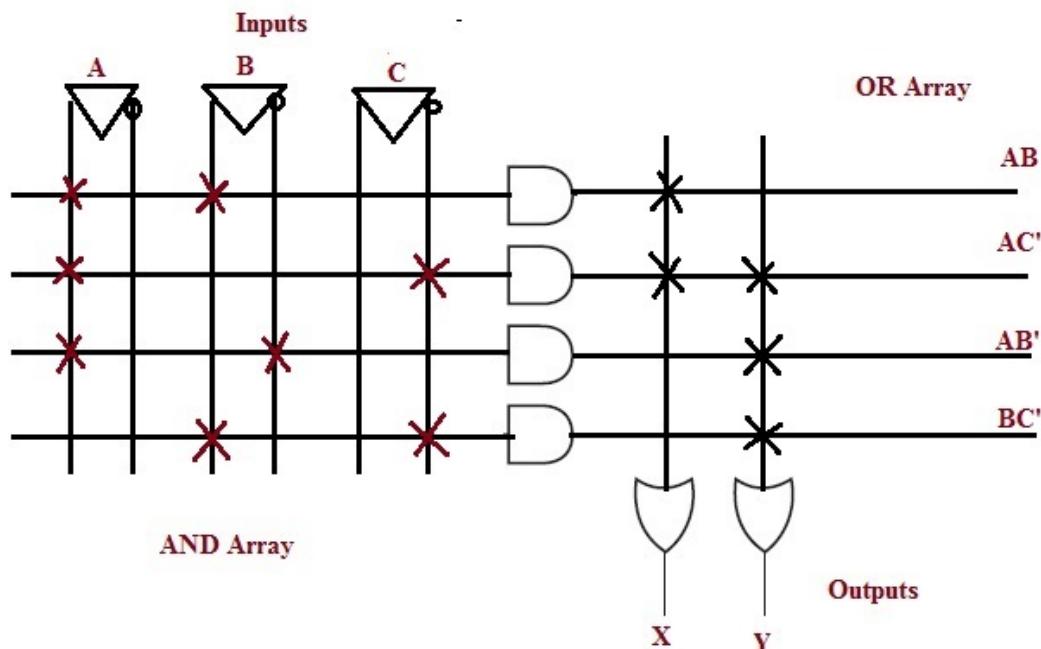
Example of PLA

Implement the following Boolean expression with the help of programmable logic array (PLA)

$$X = AB + AC'$$

$$Y = AB' + BC + AC'$$

The above given two Boolean functions are in the form of SOP (sum of products). The product terms present in the Boolean expressions are X & Y, and one product term that is AC' is common in every equation. So, the total required logic gates for generating the above two equations is AND gates-4, OR programmable OR gates-2. The equivalent PLA logic diagram is shown below.



PLA Logic Circuit

The AND gates which are programmable have the right of entry for normal as well as complemented variable inputs. In the above logic diagram, the available inputs for each AND gate are A, A', B, B', C, C'. So, in order to generate a single product term with every AND gate, the program is required. All the product terms are obtainable at the inputs of each OR gate. Here, the programmable connections on the logic gate can be denoted with the symbol 'X'.

Difference between PAL and PLA

The **Difference between PAL and PLA in Tabular Form** mainly includes **PAL and PLA full form**, construction, availability, flexibility, cost, number of functions, and speed which are discussed below.

Programmable Array Logic (PAL)	Programmable Logic Array (PLA)
The full form of PAL is programmable array logic	The full form of the PLA is a programmable logic array
The construction of PAL can be done using the programmable collection of AND & OR gates	The construction of PLA can be done using the programmable collection of AND & fixed collection of OR gates.
The availability of PAL is less prolific	The availability of PLA is more
The flexibility of PAL programming is more	The flexibility of PLA is less
The cost of a PAL is expensive	The cost of PLA is middle range
The number of functions implemented in PAL is large	The number of functions implemented in PLA is limited
The speed of PAL is slow	The speed of PLA is high

Memory and Programmable Logic

Random-Access Memory

- A memory unit stores binary information in groups of bits called words.
1 byte = 8 bits
- 1 word = 2 bytes
- 2 The communication between a memory and its environment is achieved through data input and output lines, address selection lines, and control lines that specify the direction of transfer.

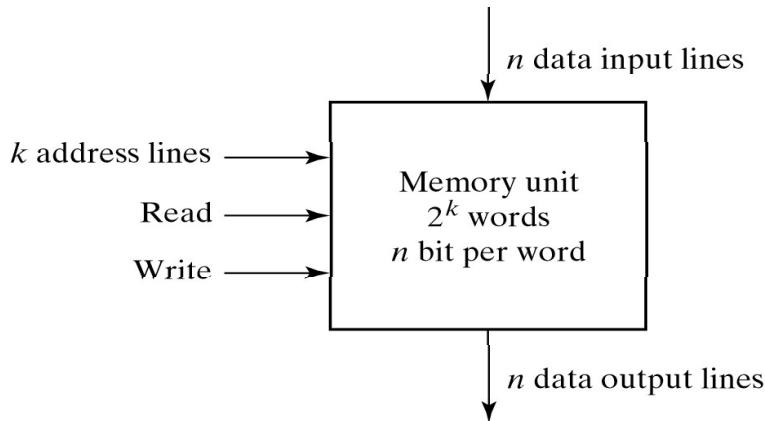


Fig. 7-2 Block Diagram of a Memory Unit

Content of a memory:

- Each word in memory is assigned an identification number, called an address, starting from 0 up to $2^k - 1$, where k is the number of address lines.
- The number of words in a memory with one of the letters K= 2^{10} , M= 2^{20} , or G= 2^{30} .
 $64K = 2^{16}$ $2M = 2^{21}$
 $4G = 2^{32}$

Write and Read operations :

- Transferring a new word to be stored into memory:
 1. Apply the binary address of the desired word to the address lines.
 2. Apply the data bits that must be stored in memory to the data input lines.
 3. Activate the write input.
- Transferring a stored word out of memory:
 1. Apply the binary address of the desired word to the address lines.
 2. Activate the read input.
- Commercial memory sometimes provide the two control inputs for reading and writing in a somewhat different configuration in table 7-1.

Table 7-1
Control Inputs to Memory Chip

Memory Enable	Read/Write	Memory Operation
0	X	None
1	0	Write to selected word
1	1	Read from selected word

Memory address		
Binary	decimal	Memory content
0000000000	0	1011010101011101
0000000001	1	1010101110001001
0000000010	2	0000110101000110
•	•	•
1111111101	1021	1001110100010100
1111111110	1022	0000110100011110
1111111111	1023	1101111000100101

Fig. 7-3 Content of a 1024×16 Memory

Types of memories:

- In random-access memory, the word locations may be thought of as being separated in space, with each word occupying one particular location.
 - In sequential-access memory, the information stored in some medium is not immediately accessible, but is available only certain intervals of time. A magnetic disk or tape unit is of this type.
- There are two basic types of RAM :**
- (i) Dynamic Ram
 - (ii) Static RAM

- Dynamic RAM : loses its stored information in a very short time (for milli sec.) even when power supply is on. D-RAM's are cheaper & lowe

Types of ROMs:

ROM : Read only memory: Its non volatile memory, ie, the information stored in it, is not lost even if the power supply goes off. It's used for the permanent storage of information. It also posses random access property. Information can not be written into a ROM by the users/programmers. In other words the contents of ROMs are decided by the manufactures.

The following types of ROMs an listed below :

(i) PROM : It's programmable ROM. Its contents are decided by the user. The user can store permanent programs, data etc in a PROM. The data is fed into it using a PROM programs.

(ii) EPROM : An EPROM is an erasable PROM. The stored data in EPROM's can be erased by exposing it to UV light for about 20 min. It's not easy to erase it because the EPROM IC has to be removed from the computer and exposed to UV light. The entire data is erased and not selected portions by the user. EPROM's are cheap and reliable.

(iii) EEPROM (Electrically Erasable PROM) : The chip can be erased & reprogrammed on the board easily byte by byte. It can be erased with in a few milliseconds. There is a limit on the number of times the EEPROM's can be reprogrammed, i.e.; usually around 10,000 times.

- A combinational PLD is an integrated circuit with programmable gates divided into an AND array and an OR array to provide an AND-OR sum of product implementation.
- PROM: fixed AND array constructed as a decoder and programmable OR array.
- PAL: programmable AND array and fixed OR array.
- PLA: both the AND and OR arrays can be programmed.
- The required paths in a ROM may be programmed in four different ways.
 1. Mask programming: fabrication process
 2. Read-only memory or PROM: blown fuse /fuse intact
 3. Erasable PROM or EPROM: placed under a special ultraviolet light for a given period of time will erase the pattern in ROM.
 4. Electrically-erasable PROM(EEPROM): erased with an electrical signal instead of ultraviolet light.

READ ONLY MEMORY

A ROM is essentially a memory device in which permanent binary information is stored. The binary information must be specified by the designer and is then embedded in the unit to form the required interconnection pattern. Once the pattern is established it stays within the unit even when power is turned off and on again.

A block diagram of ROM is shown in the Figure 1. It consists of k inputs and n outputs. The inputs provide the address for the memory and the outputs give the data bits of the stored word which is selected by the address. The number of words in a ROM is determined from the fact that k address input lines are needed to specify 2^k words.

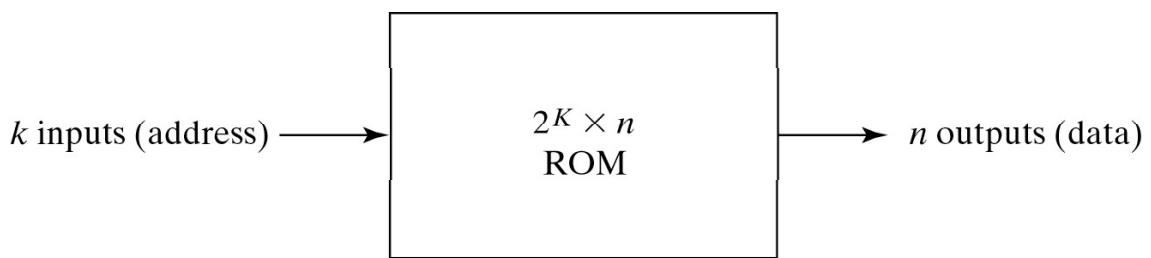


Fig. 7-9 ROM Block Diagram

ROM does not have data inputs because it does not have a write operation.

Consider for example a 32×8 ROM. The unit consists of 32 words of 8 bits each. There are five input lines that form the binary numbers from 0 through 31 for the address. The Figure 2 shows the internal logic construction of the ROM. The five inputs are decoded into 32 distinct outputs by means of a 5×32 decoder. Each output of the decoder represents a memory address. The 32 outputs of the decoder are connected to each of the eight OR gates.

The diagram shows the array logic convention used in complex circuits . Each OR gate must be considered as having 32 inputs. Each output of the decoder is connected to one of the inputs of each OR gate. Since each OR gate has 32 input connections and there are 8 OR gates, the ROM contains $32 \times 8 = 256$ internal connections.

In general, a $2^k \times n$ ROM will have an internal $k \times 2^k$ decoder and n OR gates. Each OR gate has 2^k inputs, which are connected to each of the outputs of the decoder.

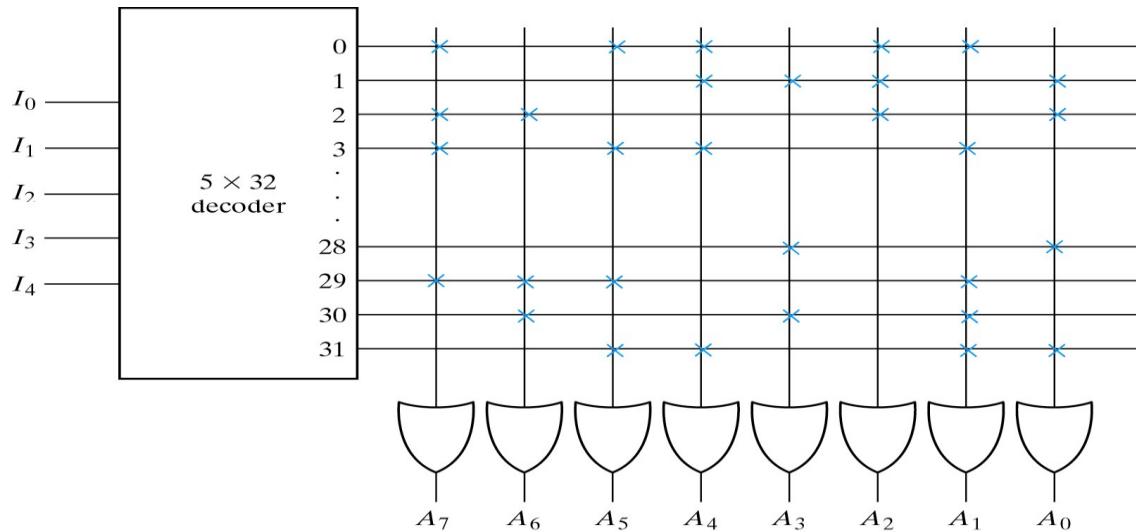


Fig. 7-11 Programming the ROM According to Table 7-3

Inputs					Outputs				
I₄	I₃	I₂	I₁	I₀	A₇	A₆	A₅	A₄	A₃
0	0	0	0	0	1	0	1	1	0
0	0	0	0	1	0	0	0	1	1
0	0	0	1	0	1	1	0	0	0
0	0	0	1	1	1	0	1	1	0
⋮					⋮				
1	1	1	0	0	0	0	0	0	1
1	1	1	0	1	1	1	1	0	0
1	1	1	1	0	0	1	0	0	1
1	1	1	1	1	0	0	1	1	0

Every 0 listed in the truth table specifies a no connection and every 1 listed specifies a path that is obtained by a connection. The four 0's in the word are programmed by blowing the fuse links between output 3 of the decoder and the inputs of the OR gates associated with outputs A_6 , A_3 , A_2 and A_0 . The four 1's in the word are marked in the diagram with a X to denote a connection in place of a dot used for permanent connection in logic diagrams.

When the input of the ROM is 00011, all the outputs of the decoder are 0 except for output 3, which is at logic 1. The signal equivalent to logic 1 at decoder output 3 propagates through the connections to the OR gate outputs of A_7 , A_5 , A_4 and A_1 . The other four outputs remain at 0. The result is that the stored word 10110010 is applied to the eight data outputs.

COMBINATIONAL PLDs

The PROM is a combinational programmable logic device (PLD). A combinational PLD is an integrated circuit with programmable gates divided into an AND array and an OR array to provide an AND-OR sum of product implementation. There are three major types of combinational PLDs and they differ in the placement of the programmable connections in the AND-OR array. The Figure 3 shows the configuration of three PLDs.

The PROM has a fixed AND array constructed as a decoder and programmable OR array. The programmable OR gates implement the Boolean functions in sum of minterms.

The programmable array logic (PAL) has a programmable AND array and a fixed OR array. The AND gates are programmed to provide the product terms for the Boolean functions which are logically summed in each OR gate. The most flexible PLD is the programmable logic array (PLA) where both AND and OR arrays can be programmed. The product terms in the AND array may be shared by any OR gate to provide the required sum of products implementation.

Fig 3 Basic configuration of three PLDs

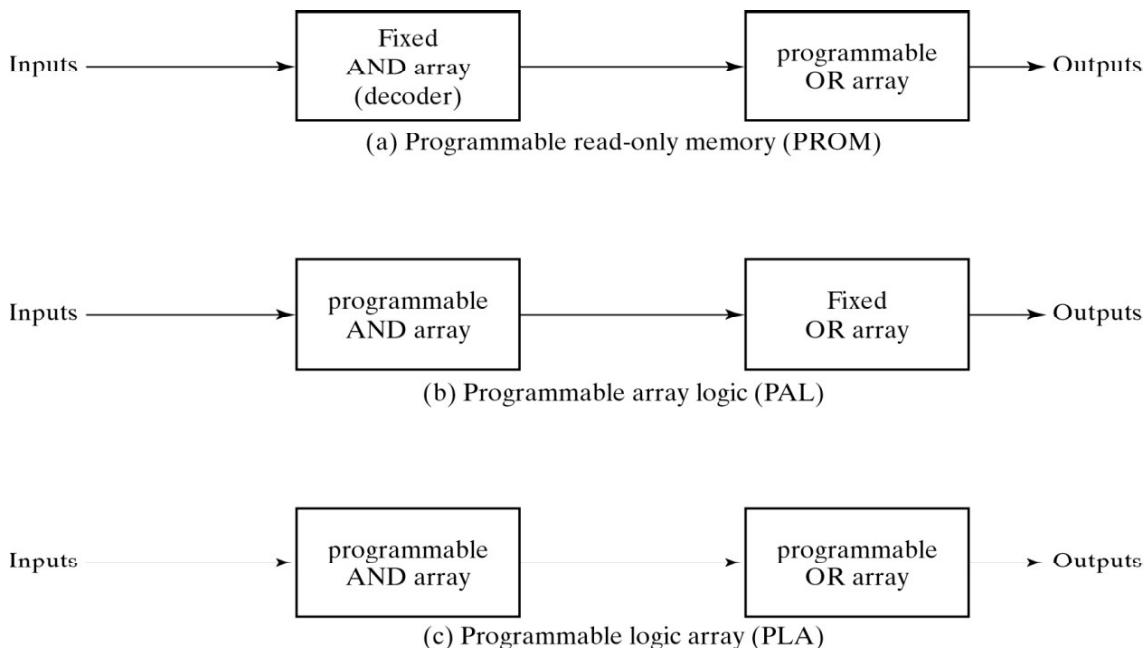


Fig. 7-13 Basic Configuration of Three PLDs

Types of ROMs:

- A combinational PLD is an integrated circuit with programmable gates divided into an AND array and an OR array to provide an AND-OR sum of product implementation.
 - PROM: fixed AND array constructed as a decoder and programmable OR array.
 - PAL: programmable AND array and fixed OR array.
 - PLA: both the AND and OR arrays can be programmed.
 - The required paths in a ROM may be programmed in four different ways.
5. Mask programming: fabrication process
 6. Read-only memory or PROM: blown fuse /fuse intact
 7. Erasable PROM or EPROM: placed under a special ultraviolet light for a given period of time will erase the pattern in ROM.
 8. Electrically-erasable PROM(EEPROM): erased with an electrical signal instead of ultraviolet light.

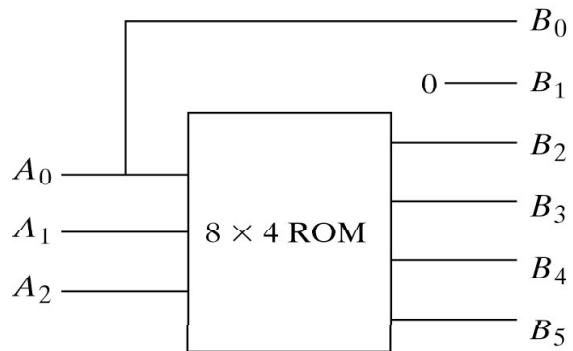
Example of ROM :

- Design a combinational circuit using a ROM. The circuit accepts a 3-bit number and generates an output binary number equal to the square of the input number.

Derive truth table first

Table 7-4
Truth Table for Circuit of Example 7-1

Inputs			Outputs						Decimal
A_1	A_1	A_0	B_5	B_4	B_3	B_2	B_1	B_0	
0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	1	1
0	1	0	0	0	0	1	0	0	4
0	1	1	0	0	1	0	0	1	9
1	0	0	0	1	0	0	0	0	16
1	0	1	0	1	1	0	0	1	25
1	1	0	1	0	0	1	0	0	36
1	1	1	1	1	0	0	0	1	49



(a) Block diagram

A_2	A_1	A_0	B_5	B_4	B_3	B_2
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	1
0	1	1	0	0	1	0
1	0	0	0	1	0	0
1	0	1	0	1	1	0
1	1	0	1	0	0	1
1	1	1	1	1	0	0

(b) ROM truth table

Fig. 7-12 ROM Implementation of Example 7-1



This document was created with the Win2PDF “print to PDF” printer available at
<http://www.win2pdf.com>

This version of Win2PDF 10 is for evaluation and non-commercial use only.

This page will not be added after purchasing Win2PDF.

<http://www.win2pdf.com/purchase/>