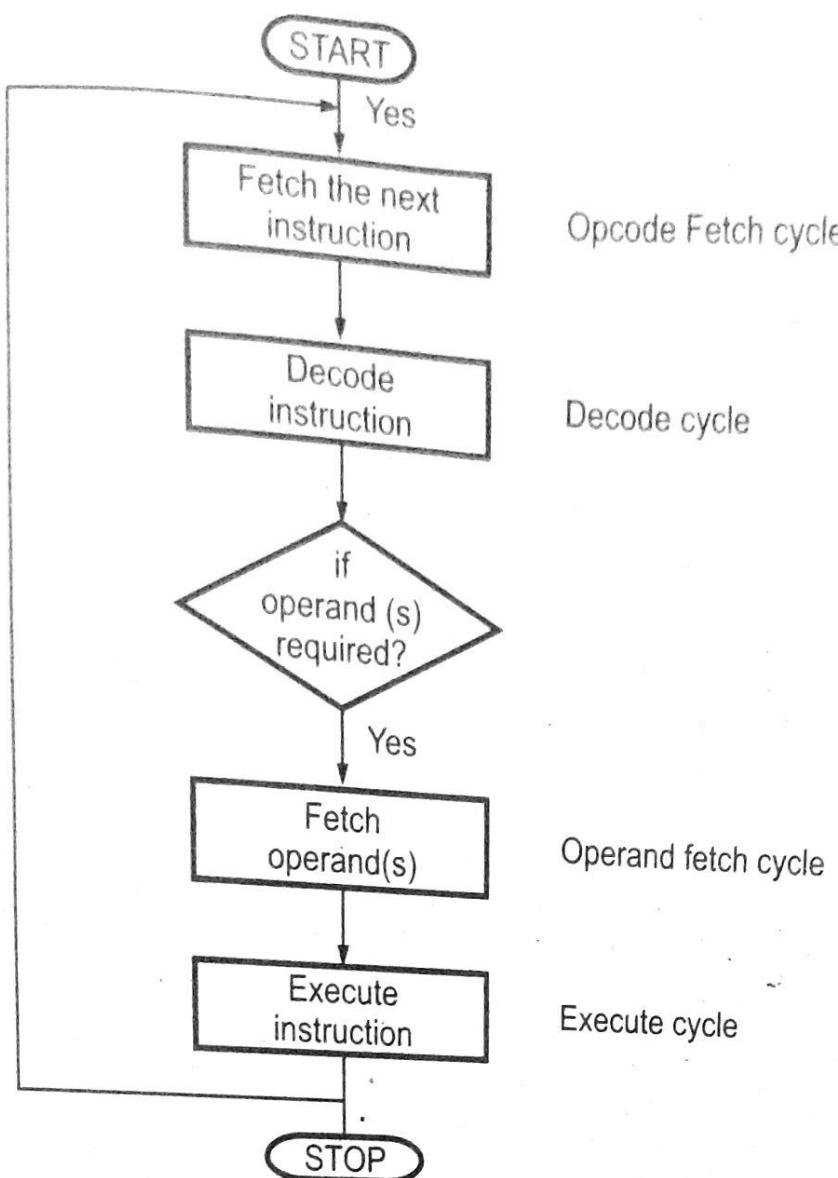


## **11.4 Microprocessor Operation**

- The primary function of a microprocessor is to execute sequence of instructions stored in a memory, which is external to the microprocessor. The sequence of operations involved in processing an instruction constitutes an instruction cycle.
- From the above discussion it is cleared that, the complete instruction cycle involves three operations : **fetch, decode and execution.**  
(See Fig. 11.4.1 on next page)
- **Fetch :**
  - Microprocessor sends the address of the instruction to the memory.
  - Microprocessor also sends memory read control signal to enable memory.
  - Microprocessor reads instruction byte (opcode) sent from memory on data bus and places it in the Instruction Register (IR) in the microprocessor.



**Fig. 11.4.1 Basic instruction cycle**

- When the fetch cycle is used to read instruction it is known as **instruction fetch cycle**. On the other hand, when the fetch cycle is used to read operand it is known as **operand fetch cycle**.

**Decode :**

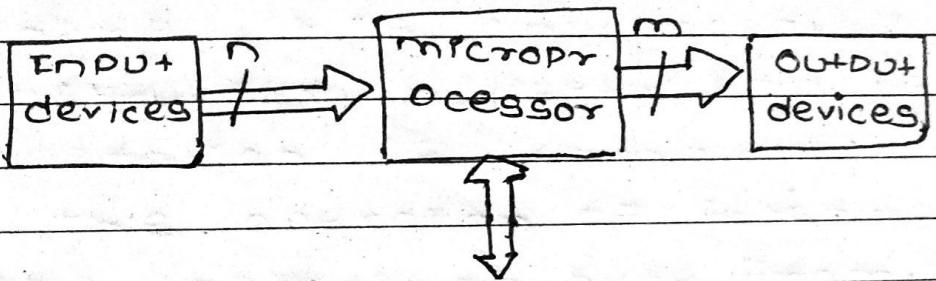
- Microprocessor decodes the opcode of the instruction stored in the instruction register to determine which operation is to be performed.

**Execution :**

- Microprocessor performs the specified operation. This often involves performing an arithmetic or logical operation and storing the result in the destination location.

① What is microprocessor? List different operation and its application?

- Microprocessor is important part of computer architecture without which you will not able to perform anything on computer.
- The microprocessor is integrated circuit that performs all function of CPU of computer.
- It is a programmable device which take in input, performed some operation on it and produce desired output.



- microprocessor consist of ALU, control unit, and register.
- ALU perform Arithmetic and logic operation
- Control unit for controlling instruction and flow of data.
- Register array consist of register identified by letter like B, C, D, H, L and accumulator.

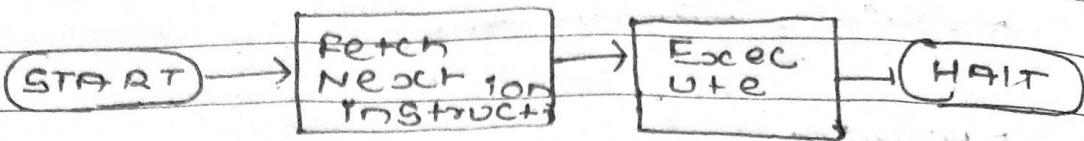
### \* Various operation:-

### \* Application:-

- ① microprocessor based system are widely used in Frequency meters, Spectrum analysers etc.
- ② In industry, they are widely used for controlling various parameter like speed, temperature and pressure.
- ③ In telephone industry they are widely used in digital telephone set, telephone exchange and modems.
- ④ 32 bit microprocessor are widely used in CAD machines.
- ⑤ The car, moruti WagonR also uses 32 bit processor for controlling the MPFI unit.

### \* Operation:-

- The basic function of microprocessor is to execute a program.
- For execution of program microprocessor consider each instruction separately.
- For each instruction the microprocessor require 2 step:-
  - ① Fetch instruction from memory
  - ② Execute instruction



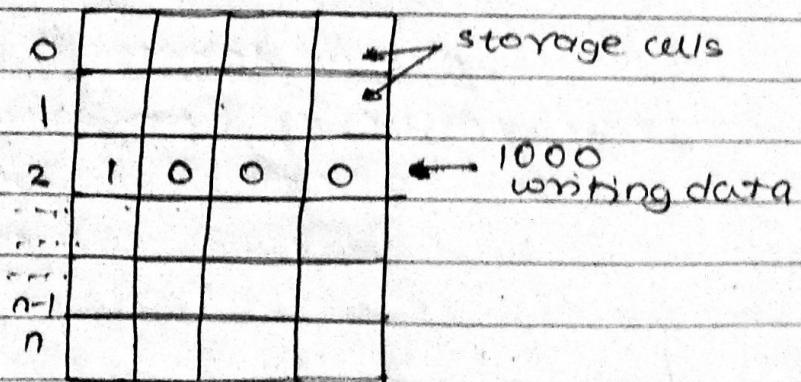
- For fetching an instruction, microprocessor places the content of program counter on address line, make data line available and activate a memory read control signal. Because of this, the instruction code from memory is available on data line.
- The microprocess accept the content of data line and transfer to internal data bus. From there it is only accepted by instruction register and fetch operation is complete.
- The operation of fetching the instruction byte from memory is called instruction Fetch.
- The timing and control get all information about operation specified.
- The timing and control will perform the operation is called execution of instruction.

- 2) Explain memory organization of microprocessor or

Ans:

- Memories are made up of registers. Each register in memory is one storage location. Each location by is identified by an address called memory address.
- The no. of storage locations can vary from a few in some memories to hundreds of thousands in others.
- Each location can accommodate one or more bits. Generally, the total no. of bits that a memory can store is its capacity. Most of the types of the capacity is specified in terms of byte.

Address



- Each register consists of storage elements (FLIP-FLOPS or capacitors), each of which stores one-bit of data. A storage element is called a cell.

- The data stored in a memory by a process called writing and are removed from the memory by the process of reading.

- 3) With the help of block diagram explain fundamental unit of microprocessor.

Ans :-

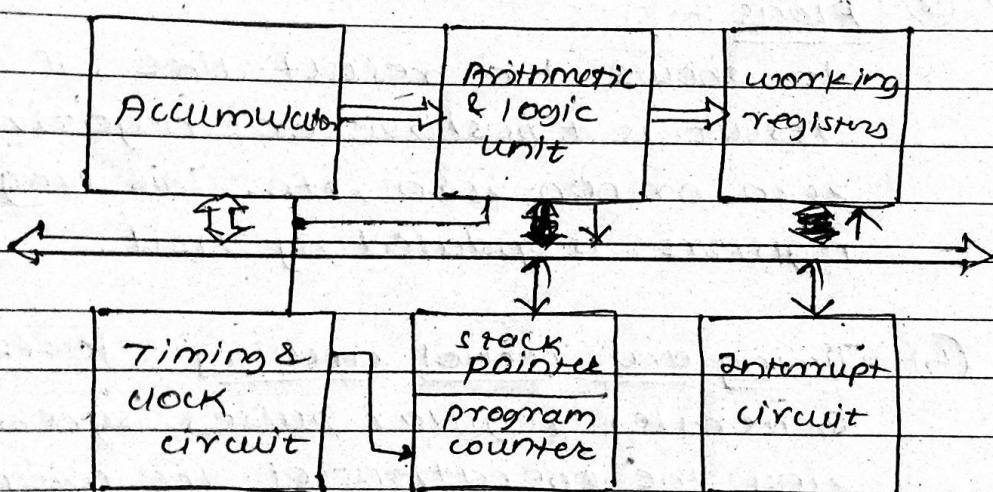


fig. Block diagram of microprocessor.

This architecture consists of different groups as follows:

1. Registers
2. ALU
3. ~~working registers~~ flags
4. Timing & control circuitry
5. system bus
6. program counter.

① Registers :-

small, fast storage locations within the CPU for temporary data storage including the instruction register (IR) and program counter (PC).

② Arithmetic and logic unit :-

performs arithmetic (addition, subtraction, etc) and logic (AND, OR, NOT) operations on data, executing instructions.

③ Flags :-

It shows the result like, if the result is + positive or - negative, zero or non-zero, etc. This flag register is updated by ALU.

④ Timing and control circuitry - [clock]

Generates regular pulses synchronizing the operations of the microprocessor.

⑤ System bus :-

connects all components, facilitating the transfer of data and instructions between them.

⑥ Program counter :-

The microprocessor uses the PC to sequence the execution of instructions. It is used to hold the address of program memory.

Q4. What are functional units of microprocessor



- 1) Arithmetic Logic Unit (ALU)
- 2) Control Unit
- 3) Registers
- 4) Memory Management Unit (MMU):
- 5) Input/Output Interfaces.

These functional units work together in a coordinated manner during the execution of a program.

The control unit fetches instructions from memory, sends them to the ALU for processing. To manage the flow of data between the various registers, the MMU ensures proper addressing of memory. I/O interfaces facilitate communication with external devices.

#### I) Arithmetic Logic Unit (ALU):

The ALU is responsible for performing arithmetic & logic operations. It can perform tasks like addition, subtraction, AND, OR, & other logical operations.

### 2) Control Unit:

The control unit manages the flow of data & instructions within the microprocessor. It fetches instructions from memory, decodes them, & coordinates the execution of operation.

### 3) Registers:

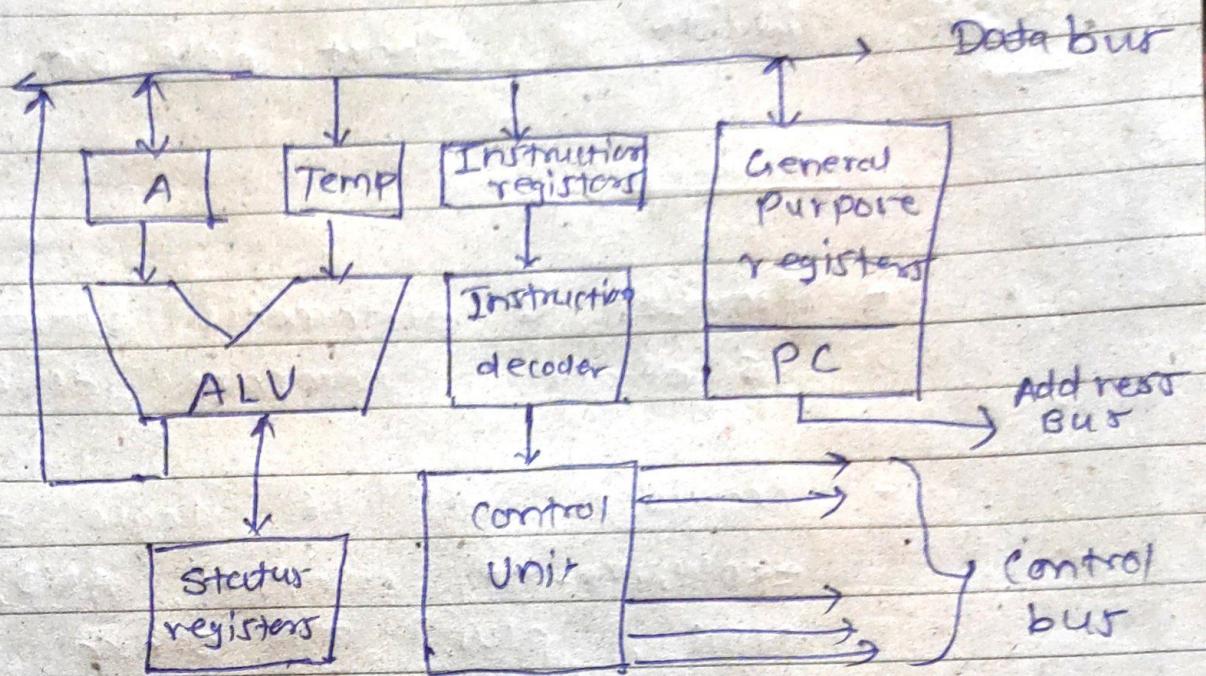
Registers are small, fast storage locations within the microprocessor. They hold data that is being processed or temporarily stored during execution.

### 4) Memory management Unit (MMU):

The MMU is responsible for managing memory resources. It translates virtual addresses to physical addresses facilitating access to data stored in memory. It plays a crucial role in the interaction between the microprocessor & memory.

### 5) Input / Output Interfaces:

The microprocessor sends & receives data to & from peripherals.



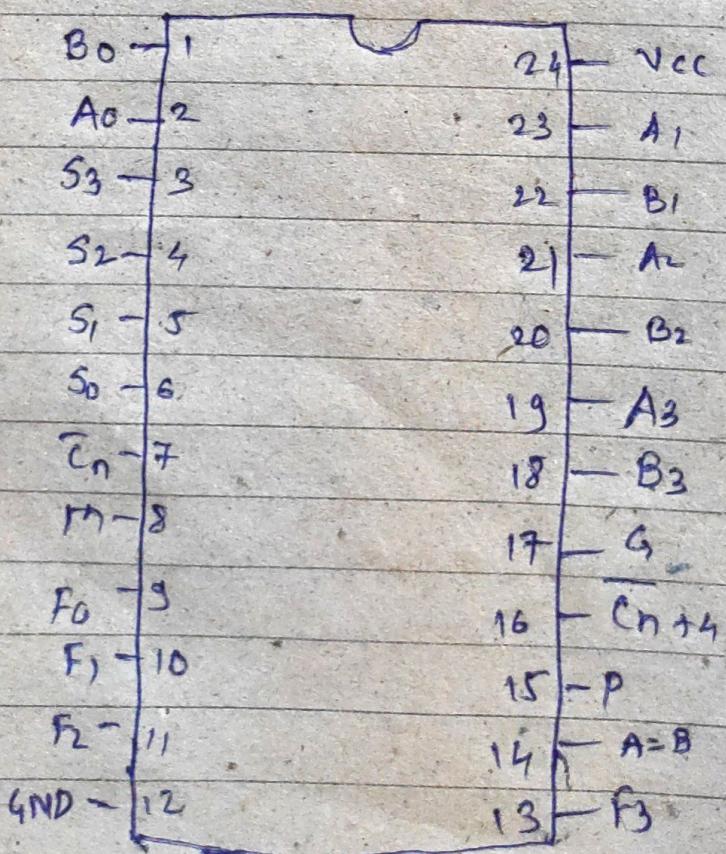
Q5

→ Write a short note on ALU IC 74181.

### Features:

- Provides all 16 arithmetic operations: add, subtract, compare, double etc.
- Provides all 16 logic operations: exclusive-OR, compare, AND, NAND, OR, NOR, plus ten other logic operations.
- Full lookahead for high speed arithmetic operation on long words.

### Connection Diagram:

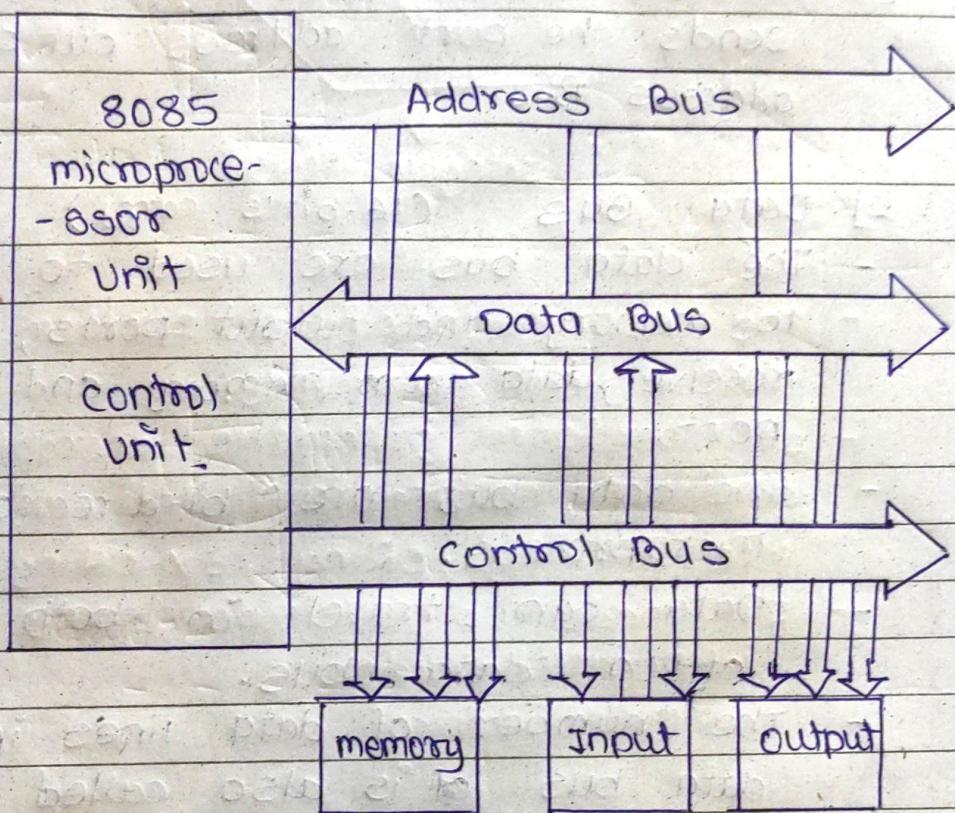


Pin Names	Description
A0 - A3	Operand Input
B0 - B3	Operand Input
S0 - S3	function select input
m	mode control input
$\bar{C}_n$	carry Input (Active low)
F0 - F3	Function output
A = B	comparator output
G	carry Generator output
P	carry propagate output
$\bar{C}_{n+4}$	carry output (Active low)

(8) \*

what is system bus, draw microprocessor bus structure, explain in brief.

- A bus is a collection of wires that are grouped together to serve single purpose.
- In microprocessor there are three sets of communication lines that are called buses



- They are the address bus, the data bus and control bus the three buses together form the "system bus".

### \* System Bus

#### 1] Address Bus

- The bus over which the microprocessor sends out the address of a memory location or I/O location is called as the address bus.
- the length of address Bus is 16 bits

- The address bus is unidirectional in Address bus bits flow only in one direction from the microprocessor unit to memory and I/O devices.
- The address bus is also used to send the port address on the address bus when the microprocessor reads data from or writes data to a port, it sends the port address out on the address Bus.

## 2] Data Bus (16-bit's carry)

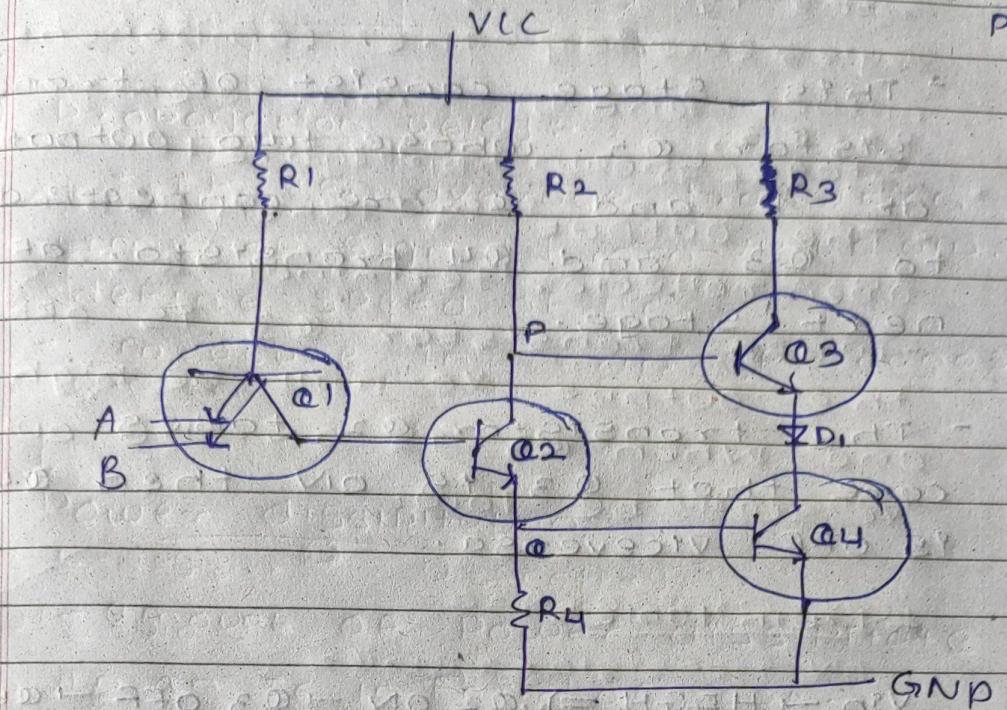
- The data bus are used to send data to memory and output ports, and to receive data from memory and input port.
- The data bus are bi-directional in microprocess.
- Data can travel in both direction of a data bus.
- The number of data lines in the data bus of is also called data bus width.
- The data bus also connected the I/O ports and microprocessor so the microprocessor can write data to or read data from the memory or the I/O port.

## 3] Control Bus .

- the microprocessor uses the control bus to provide the timing signal
- control bus is unidirectional in microprocessors.

- The microprocessor have dedicated hardware unit "control unit" which is used to send control signals
- Some of the control signals are:
  1. memory read
  2. memory write
  3. I/O read
  4. I/O write.

\* TTL :- CS speed more NAND gate with totem pole



TWO INPUT TTL

### \*WORKING OF TTL NAND GATE:-

#### ① Stage 1 (multi-emitter transistor) :-

The multi-emitter transistor work like this, when any of emitter or both emitter are connected to logic 0, then BE junction get forward bias and BC junction get reverse bias and  $I_1$  flows and  $I_2$  become 0.

When both emitter are connected to logic 1 then BE junction are reverse bias and BC junction get forward bias and  $I_1$  become 0 and  $I_2$  flows.

- When all emitter are connected to logic 1 then BE junction are reverse bias and BC junction get forward bias and  $I_1$  become 0 and  $I_2$  flows.

### (2) Stage 2 (Phase Splitter Stage):

- This stage consists of transistor Q2 whose two output at P and Q are connected to Q3 and Q4 transistors of next stage.
- The transistor Q2 takes care that Q3 is ON then Q4 is OFF vice versa.

$V_{in} \rightarrow \text{HIGH} \Rightarrow Q_2 \text{ ON} \rightarrow Q_3 \text{ OFF} \rightarrow Q_4 \text{ ON}$

$V_{in} \rightarrow \text{LOW} \Rightarrow Q_2 \text{ OFF} \rightarrow Q_3 \text{ ON and } Q_4 \text{ OFF}$

### (3) Stage 3 Totem Pole Output:

- The output transistor pair Q3 and Q4 along with diode D1 referred to as totem pole.

#### \* Truth Table ::

Input	Transistor	Output
A B	Q1 Q2 Q3 Q4	Y

0 0	ON OFF ON OFF	1
-----	---------------	---

0 1	ON OFF ON OFF	1
-----	---------------	---

1 0	ON OFF ON OFF	1
-----	---------------	---

1 1	OFF ON OFF ON	0
-----	---------------	---

\* TTL with open collector :-

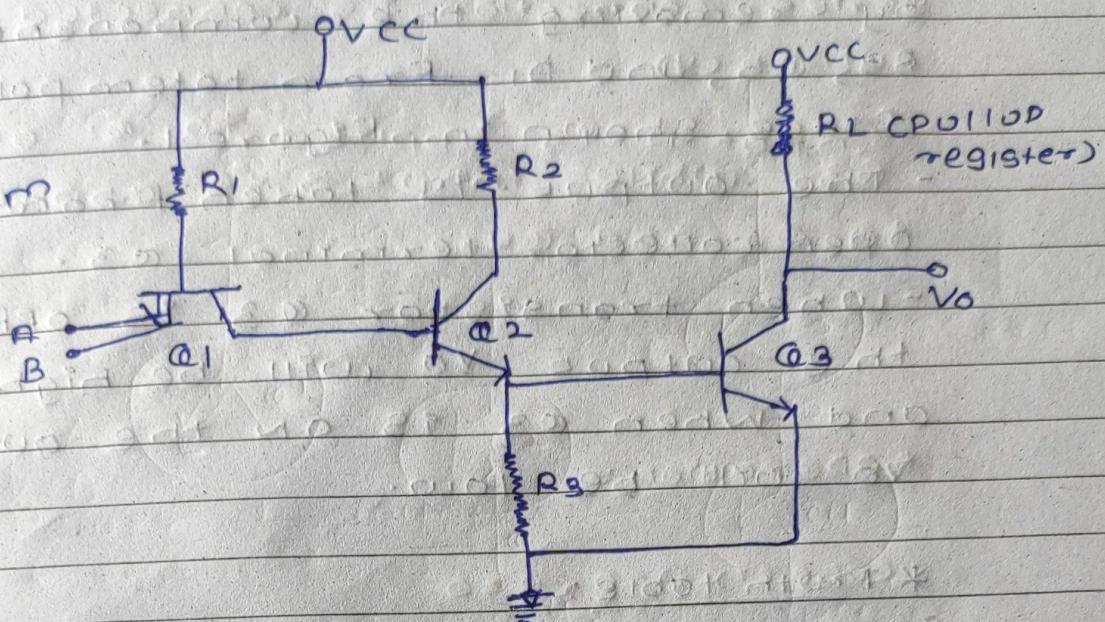


Figure 2

1st  
diagram

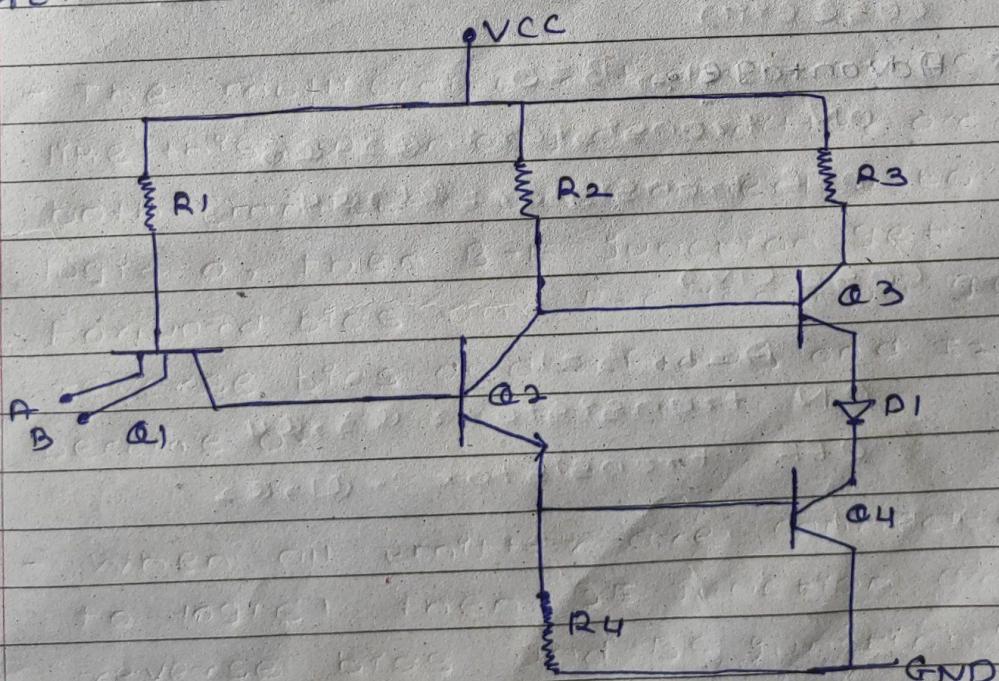


Figure 1

- Page No. \_\_\_\_\_  
Date \_\_\_\_\_
- This type of gate is obtained by removing the component  $R_3$ ,  $Q_3$ , and  $D_1$  from totem pole stage shown in Figure 1.
  - The output is taken from open collector terminal  $Q_3$ .
  - When transistor  $Q_3$  is OFF, the output  $V_O$  will be high and when  $Q_3$  is ON the output  $V_O$  will be low.

\* Truth table:-

?	A	B	$V_O$
	0	0	1
	0	1	1
	1	0	1
	1	1	0

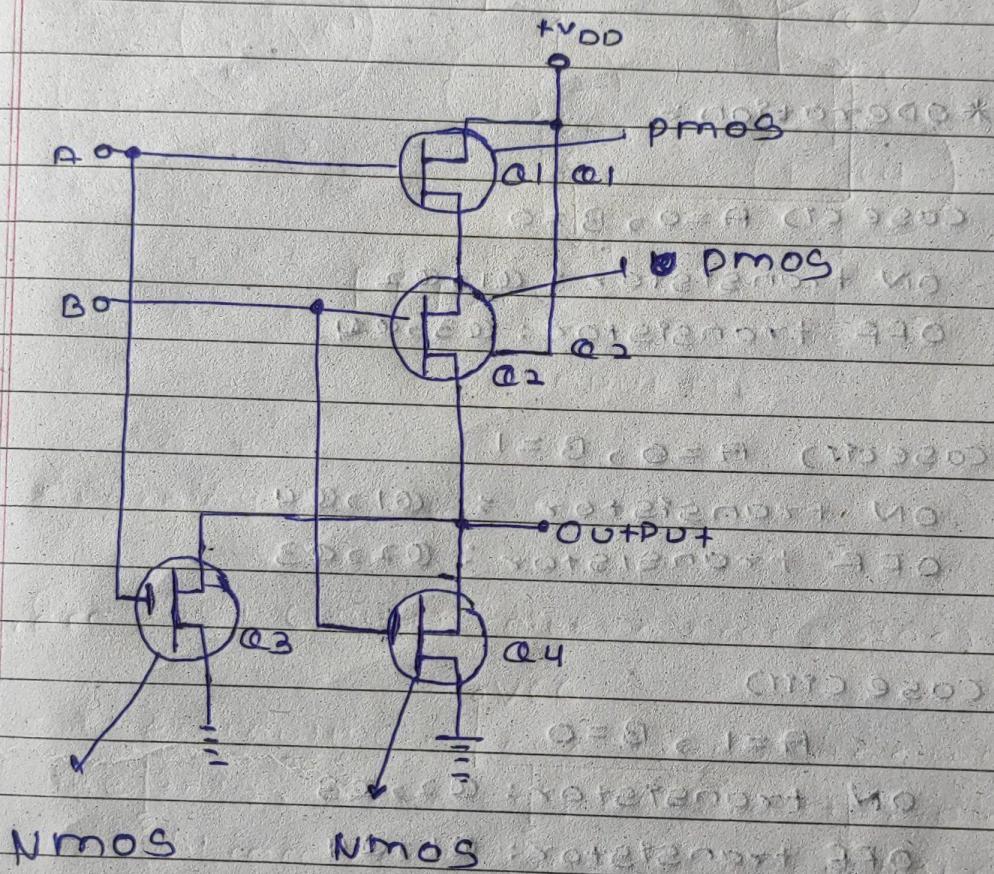
\* Advantage:-

\* CMOS NAND GATE

A) Input  
B) Output  
C) Logic  
D) Power

\* CMOS NOR GATE

Nmos - Parallel  
Pmos - Serial



Nmos

Nmos

- CMOS stands for Complementary MOS, it uses both PMOS and NMOS transistors.

- It consists of two PMOS and two NMOS transistors.

- Two PMOS transistors Q1, Q2 are connected in series, while two NMOS transistors Q3 and Q4 are connected in parallel.

- The common end of parallel series connection is connected at output QD.
- Input A is applied to Q1 and Q3 while input B is applied to Q2 and Q4.

WIRING - 20/09

### \* Operation :-

Case C(i) A=0, B=0

ON transistor : Q1, Q2

OFF transistor : Q3, Q4

Case C(ii) A=0, B=1

ON transistor : Q1, Q4

OFF transistor : Q2, Q3

Case C(iii)

A=1, B=0

ON transistor : Q2, Q3

OFF transistor : Q1, Q4

Case C(iv)

Case C(iv) A=1, B=1 output 2 20mA

ON transistor : Q3, Q4

OFF transistor : Q1, Q2

20mA going out (A0 + 20mA) = 40mA

20mA going out (B0 + 20mA) = 40mA

20mA going out (Q1 + 20mA) = 40mA

20mA going out (Q2 + 20mA) = 40mA

20mA going out (Q3 + 20mA) = 40mA

20mA going out (Q4 + 20mA) = 40mA

A	B	Q1	Q2	Q3	Q4	Y
0	0	ON	ON	OFF	OFF	1
0	1	ON	OFF	OFF	ON	0
1	0	OFF	ON	ON	OFF	0
1	1	OFF	OFF	ON	ON	0

(11)



## Compare TTL and CMOS logic families

CMOS

TTL

Voltage level

Wide range of voltage levels

Fixed voltage levels (typically 5V)

Power consumption

Low

High

Noise immunity

High

Low

Fan-out capability

High fan-out capability

lower fan-out capability

Speed

Slow propagation delay

Fast propagation delay

Power supply

Typically operates at 5V or 3.3V

Typically operates at 5V

Application

Battery-operated devices, high-density ICs

High speed applications, memory systems

Technology

MOSFET

BJT (Bipolar Junction Transistor)

\* Explain tristate logic

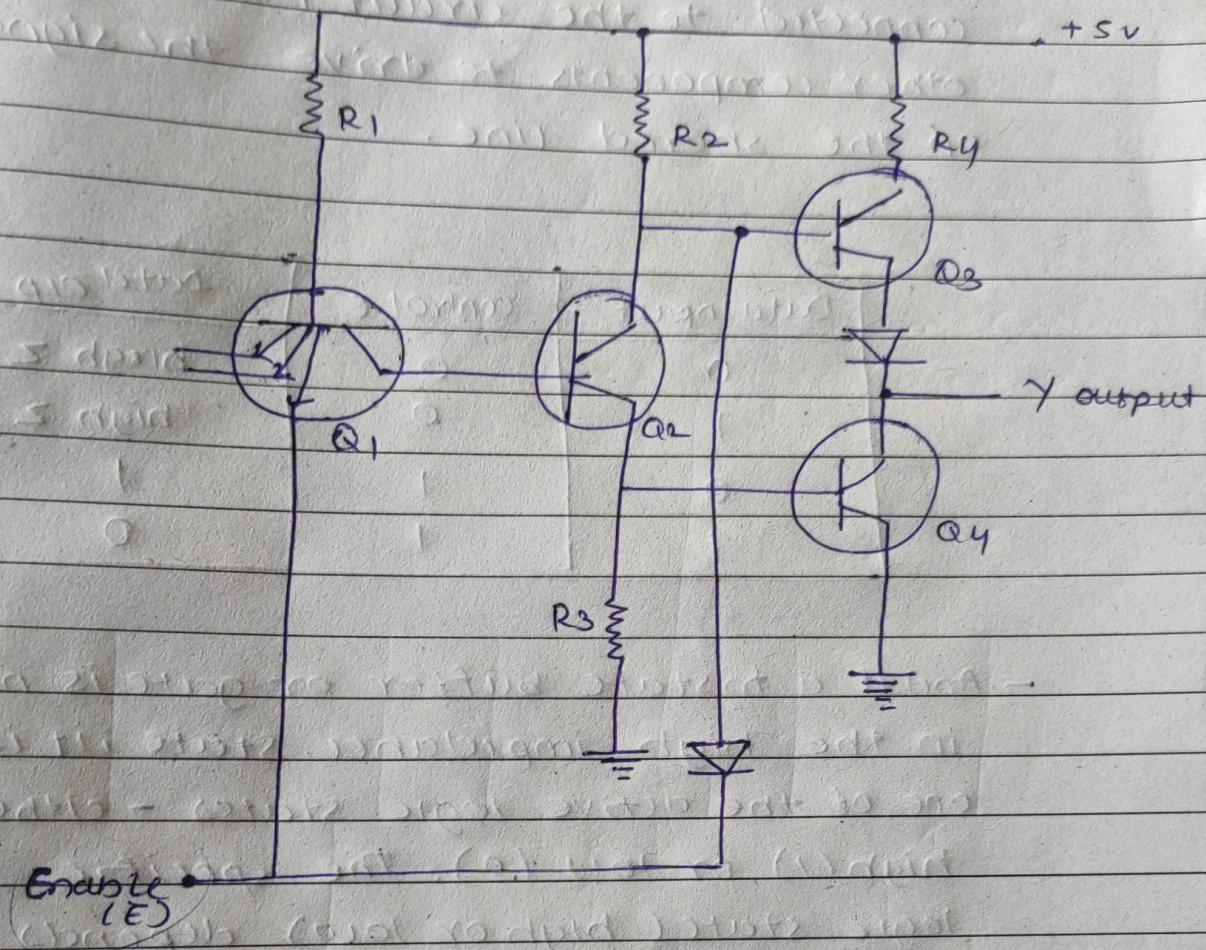


Fig. Tristate 2-input TTL NAND gate

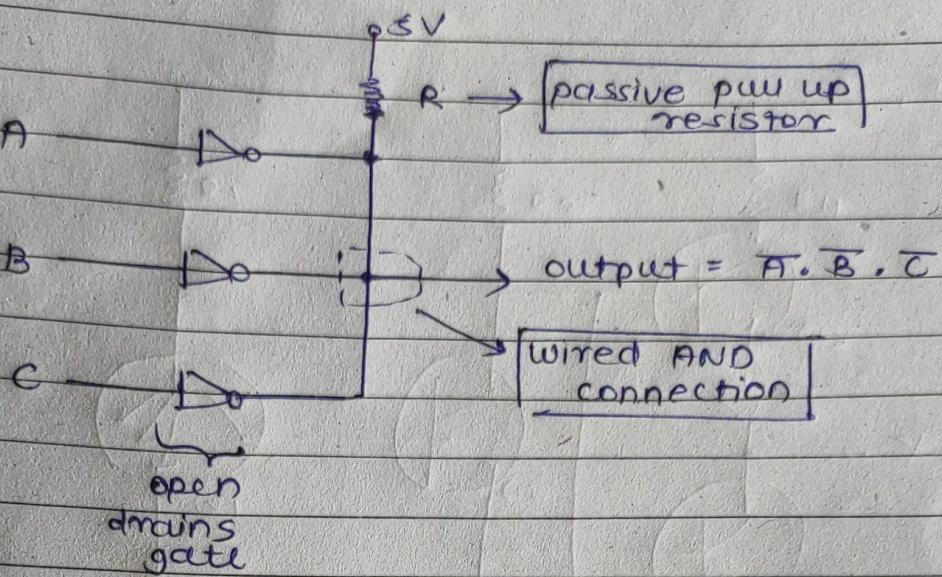
- Tristate logic in digital electronics refers to a circuit or electronic component that has three states: high(1), low(0) and high impedance(>). The high and low states are similar to standard binary logic (1 and 0), but the high impedance state effectively disconnects the outputs from the circuit.

- When a tristate buffer is in the high impedance state, it acts as if it is not connected to the circuit, allowing other components to drive the signal on the shared line.

Data Input	Control E	Data Out
0	0	high z
1	0	high z
0	1	1
1	1	0

- And if a tristate buffer or gate is not in the high impedance state, it is one of the active logic states - either high (1) or low (0). The specific logic state (high or low) depends on the control signal applied to the tristate component.

\* Explain wired logic



- When the outputs of several open drain gates are tied together with a single pull up resistor, then wired logic is formed.
- The open drain gates share a common connection, the common wire is HIGH by default due to pull up resistor
- When any one (or more) of gate outputs pulls it low, the 5V are dropped across  $R_\text{p}$  and the common connection is in low state
- Since the common O/p is HIGH only when all the outputs are in the HIGH state, connecting the O/p in the way performs the logic AND function. This is why it is called as wired-AND connection. This is shown symbolically by the dotted AND gate symbol. There is no actual AND gate there.

## CMOS Characteristics:

### 1) Low Power Consumption:

One of the CMOS's key advantages is its low power usage, making it suitable for battery-operated devices & energy-efficient operations.

### 2) High Noise Immunity:

CMOS circuits are less susceptible to noise, ensuring reliable operation in various environments.

### 3) Wide Voltage Range:

CMOS devices typically operate over a wide range of supply voltages, providing flexibility in design.

### 4) Temperature Stability:

CMOS devices exhibit stable performance across a range of temperatures.

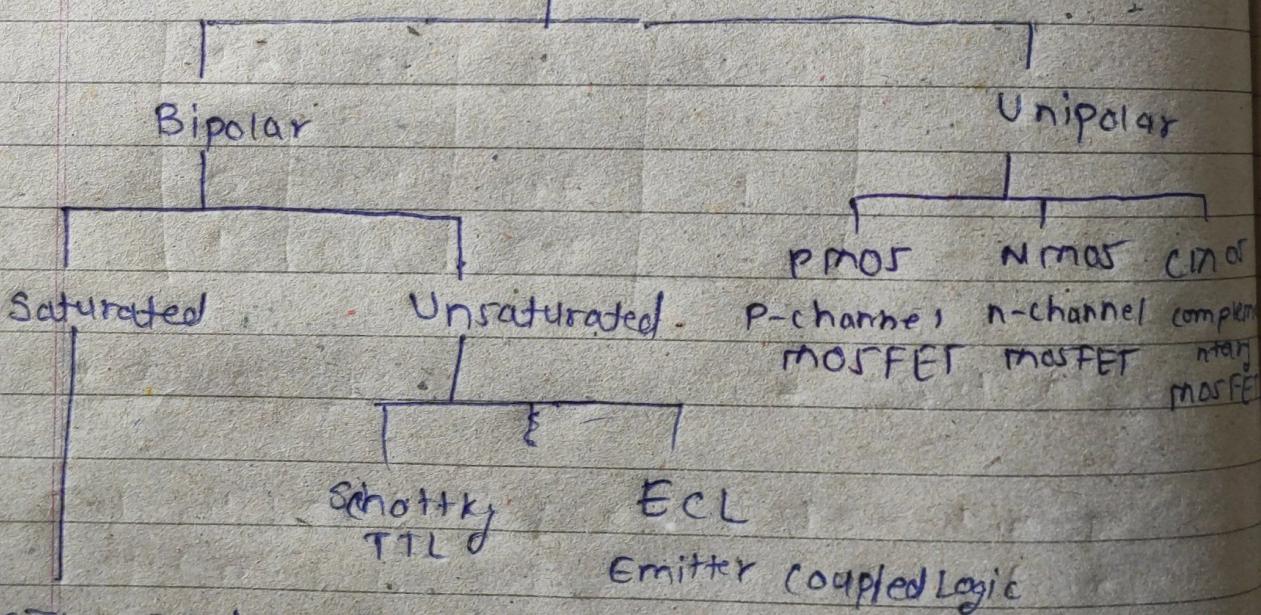
Q. 1 What is logic families. Give the classification of logic family. Also write the characteristics of CMOS.



### Logic families:

A digital logic family is a group of compatible devices with the same logic levels & supply voltages.

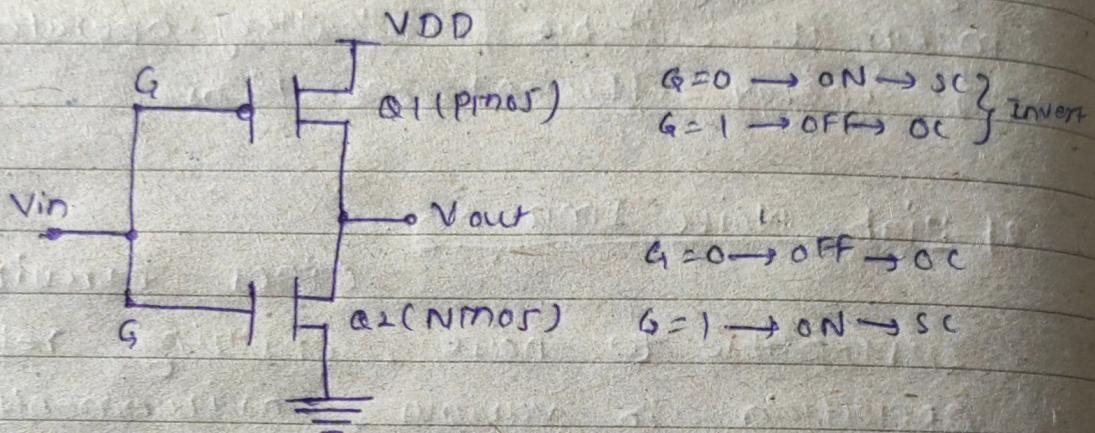
### Classification of logic families



- RTL - Register TL
- PTL - Diode TL
- DCTL - Direct coupled TL
- I<sup>2</sup>L - Integrated Injection Logic
- HTL - High Threshold Logic
- TTL - Transistor TL

**Q7.** Draw & explain circuit diagram of CMOS inverter.

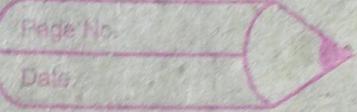
- Complement metal oxide semiconductor
- It behave like logic gate (Inverter)
- CMOS = PMOS + NMOS



Vin	Q1	Q2	Vout
0	ON	OFF	1
1	OFF	ON	0

\* Advantages:

- 1) suitable for LSI
  - 2) circuit is simple
  - 3) less power usage.
  - 4) Fan out is good.
  - 5) Noise margin is more.
- 1) stored in special conducting foam.
  - 2) propagation time is more.
  - 3) cost is more than TTL.



- 1) CMOS uses both PMOS & NMOS transistors.
- 2) CMOS inverter circuit uses both PMOS(Q1) & NMOS(Q2) transistors connected in series.
- 3) The input terminal of the inverter(V<sub>in</sub>) is connected to both the gates(g<sub>1</sub>) of Q1 & Q2.
- 4) The output of the inverter(Y) is connected to both the drain(D) terminals of Q1 & Q2.
- 5) The supply voltage V<sub>DD</sub> is connected to the source terminal of PMOS(Q1) transistor while the source terminal of NMOS(Q2) is connected to the ground terminal.
- 6) The input A can be a logic low(0) or a logic high(1). For some input when one transistor is ON then the other will be OFF.

② Define and mention standard values for TTL logic families.

### (i) Noise Margin

- The maximum value of noise that can be tolerated in a system without creating an error is called as noise margin.

### (ii) Propagation Delay.

- Propagation delay is the time taken by the output to change its state after input changes. It is denoted by  $t_p$ .

### (iii) Power dissipation.

- It represents the power delivered to the gate from the power supply.
- 

### (iv) Fan - out

The number of loads that the output of Gate can drive

### (v) Fan - in

The maximum number of inputs that can be applied to a logic gate.

(vii) Figure of merit

- Product of propagation delay & power dissipation.

(viii) Noise immunity

- maximum noise that a circuit can withstand without changing the output.

## \* characteristics of Digital IC's :-

→ 1) Voltage and Current Parameters :-

→ - Ideally the i/p voltage level of 0 V to +5 V are called logic 0 & 1 level respectively. (high voltage & low voltage)

## 2) fan-in and fan-out :-

→ fan-in is defined as the nos of i/p gate has.

e.g. two gate will have a fan-in equal to 2.

→ fan-out is defined as the max nos of o/p of the same IC family that a gate can drive without falling outside the specified o/p voltage limit.

## 3) Noise margin :-

- Noise is an unwanted electrical disturbance which may induce some voltage in the connecting wire used bet<sup>n</sup> 2 gates.

- Noise immunity is defined as the ability of logic clk to tolerate the noise without causing the o/p to change undesirably.

- The quantitative measure of noise immunity of logic family is known as Noise Margin.

## 4) propagation Delay :-

- The o/p of logic gate does not change its state instantaneously when the state of its i/p is changed.

- There is a time delay bet<sup>n</sup> these two time instant is called as the propagation delay.

## 5) power Dissipation :-

- Due to applied voltage & current flowing thru the logic ICs, some power will be dissipated in it is called power dissipation.

## 6) Speed power product

### \* Noise Margin:

- Noise margin is defined as,

High level noise margin,  $V_{NH} = V_{OH(\min)} - V_{IH(\min)}$

Low level noise margin  $V_{NL} = V_{IL(\max)} - V_{OL(\max)}$

- Substituting the value of above table, the noise margin for TTL family can be calculated

$$\therefore V_{NH} = 2.4 - 2 = 0.4 \text{ V} \quad V_{NL} = 0.8 - 0.4 = 0.4 \text{ V}$$

- Thus, noise margin for TTL family is 0.4 V.

### \* Power dissipation:

- The avg power dissipation for the std. TTL 74 series is approximately 10 mW.
- It is dependent on the parameters such as tolerance, signal level etc.

### \* Propagation Delay:

- The propagation delay of std. TTL gate is approximately 10 ns (max)

### \* Fan-out:

- A std. TTL gate is capable of driving almost 10 other TTL simultaneously.  
Hence fan-out of TTL is 10.

## Unit 4: Algorithmic State Machine

Page No.	
Date	

- It is used in sequential ckt having large number of external i/p's.

### Q.1 What is ASM chart

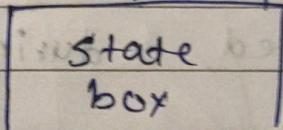
- An algorithm is a group of statements or steps which tell us how to obtain the solution of a problem.
- After writing the algorithm we draw a flowchart to specify the tech-wise procedure of decision path for an algorithm.
- An ASM is similar to conventional flowchart but we interpret in different manner.
- A conventional flowchart does not take into account the time factor but ASM does take into account the time factor.
- An ASM chart is basically a flowchart which represents a hardware algorithm.
- It is used in sequential ckt having large number of external i/p's.

Q.2 State & Explain the basic components of ASM charts.

- 1) The state box
- 2) Conditional box
- 3) The decision box

### 1) The state box

state name binary code

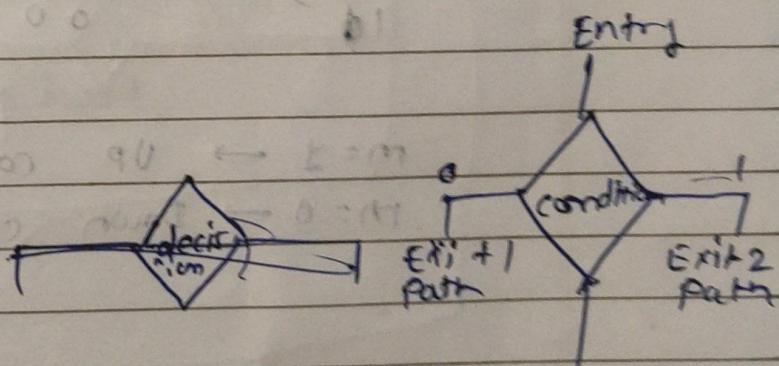


Exit

- State box is used for indicating a state.

- The state box is rectangular shape.
- The input to the state box is indicated by entry & output is indicated by exit.
- Inside this box we write information such as register operations or output signal names.

### 3) Decision Box



- The decision box is diamond shaped box.
- It has 2 or more exit paths
- Decision box makes the decision based on certain conditions if those conditions are written inside the decision box.

## Q Conditional Box:

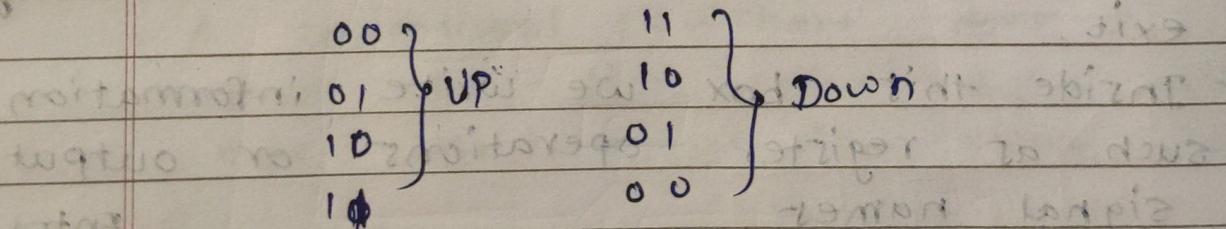
Register operation  
or  
output produced

- It is oval in shape.
- The register operations or output produced listed inside the conditional box are generated during a given state.

Q.3. Draw an ASM chart of state table of 2-bit

NP down counter having a mode control input

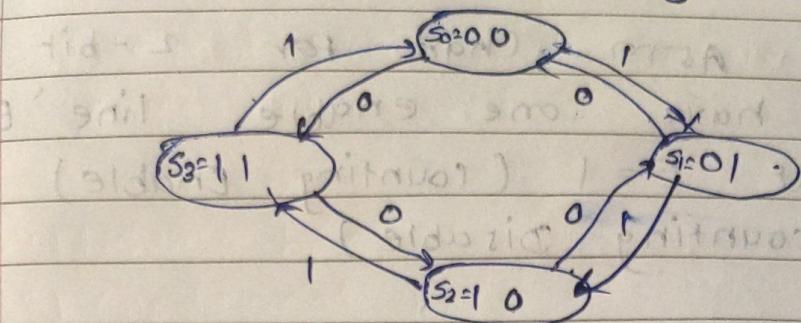
of two-bit UP down counter



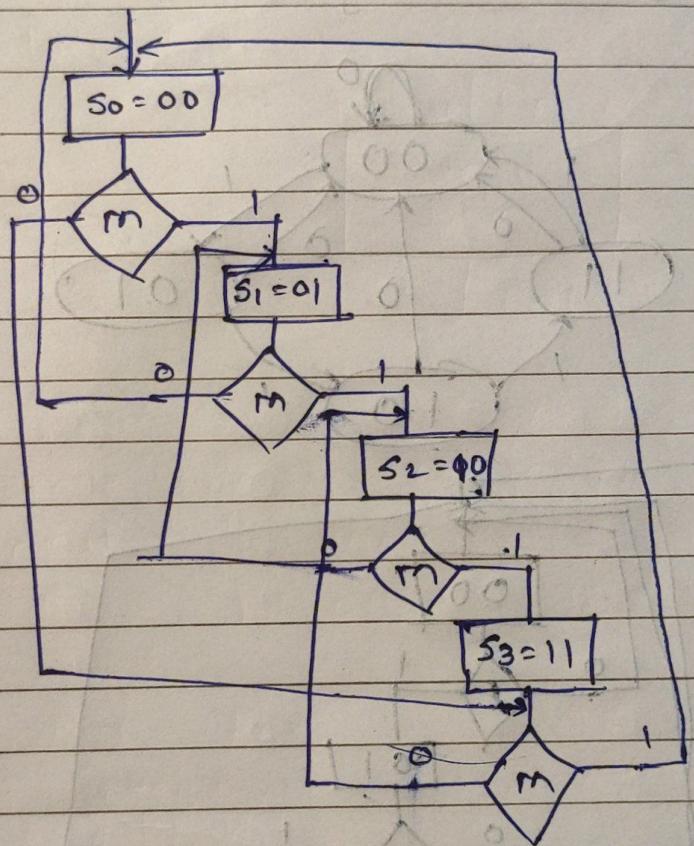
$m=1 \rightarrow$  UP counter

$m=0 \rightarrow$  Down counter

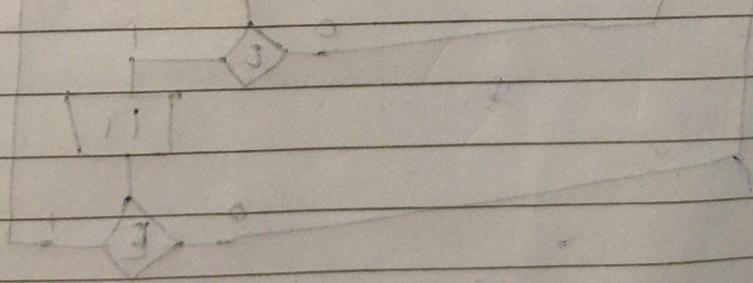
## State Diagram



## transition state



## ASM Chart.



trials MCA

## \* Application of ASM chart:-

- ① ASM is used to represent diagrams of digital integrated circuit
- ② used to describe sequential operation of digital circuit.
- ③ The ASM chart describe the sequence of event and also describe the timing relationship between sequence of state while going from one state to another.
- ④ They are used to design of control unit of computer.
- ⑤ They are also used in general control networks in any digital system.

## Unit-4 DEID

Page No.	
Date	

### ASM and PLD

fixed function  
IC.

#### \* Programmable Logic Device (PLD) :-

- A PLD is an individual programmable electronic chip which is used as an element to build digital circuit.
- It is a IC with AND array and OR array.
- To design a circuit, a designer must select appropriate IC's for circuit.

#### \* Definition :-

- A PLD is an IC and it is capable of implementing logic function.
- It is VLSI chip that contain rectangular structure which allow designer to customize it for any specific application.

#### \* Advantage of PLD :-

- ① Low development cost
- ② Short design cycle
- ③ Design security
- ④ Higher densities
- ⑤ Reduction in Power requirement

#### \* Combinational PLD :-

- Combinational PLD is integrated circuit with programmable gates.

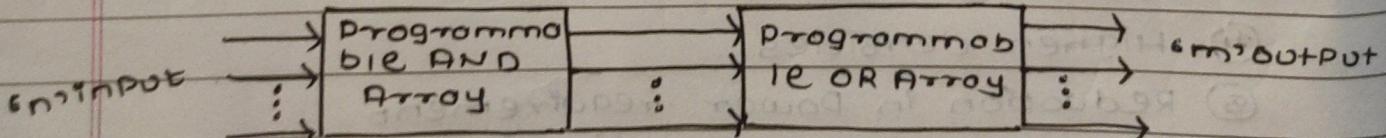
## \* Three fundamental types of PLDs :-

- ① PLA (Programmable Logic Array)
- ② PAL (Programmable Array logic)
- ③ PROM (Programmable Read only memory)

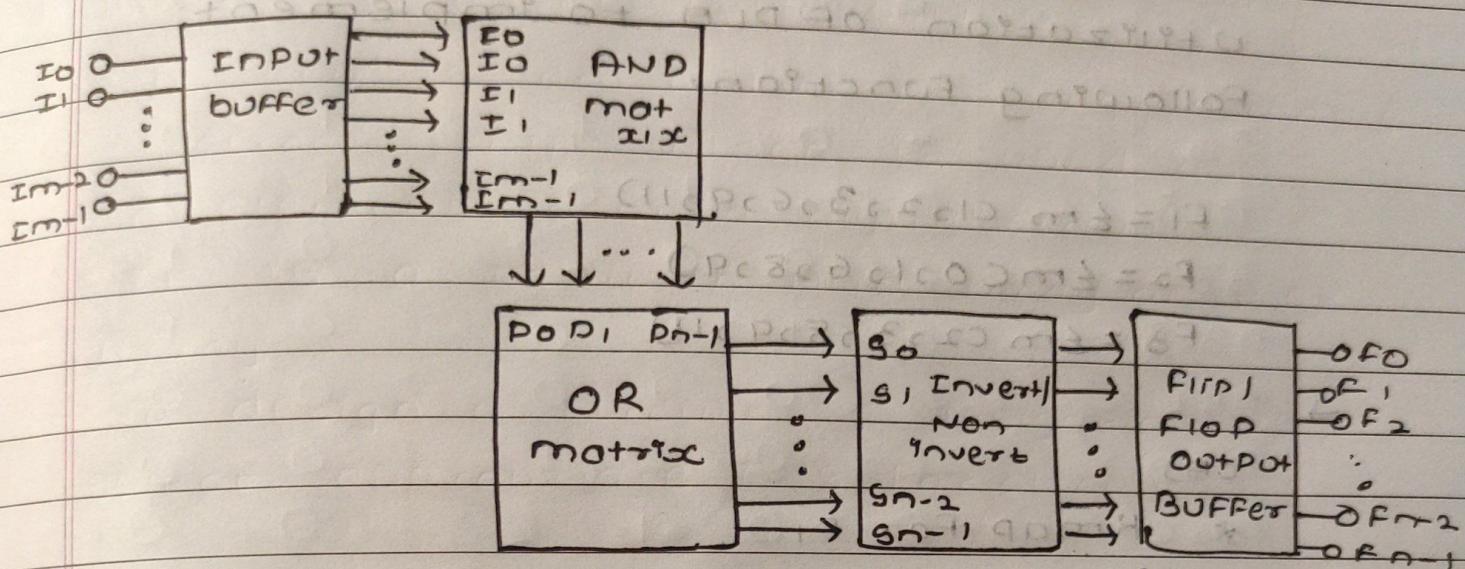
### \* ~~Com~~

#### ① PLA :-

- PLA is programmable logic device that has both programmable AND array and programmable OR array.
- Here input of AND gate are programmable that means each AND gate has both normal and complement input of variable. So based on requirement , we can program any of those inputs.
- So, we can generate only required ~~two~~ product terms by using these AND gates.



## \* Internal Architecture of PLA :-



### \* Advantage :-

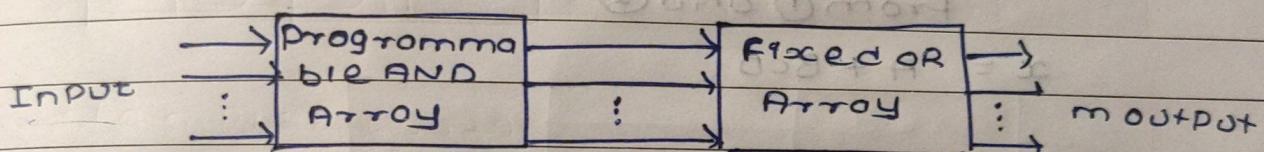
- ① Low development cost
- ② Low space requirement
- ③ Less power requirement
- ④ Less design time
- ⑤ Easy to troubleshoot
- ⑥ Easy design modification

### \* Disadvantage :-

- ① Lack of security
- ② Large power requirement
- ③ It requires large area and poor performance
- ④ Additional cost, power, space etc.

② PAL:-

- The PAL is programmable logic device that has programmable AND and Fixed OR Array.
- The advantage of PAL is that we can generate only required product term of boolean function instead of generating all min terms by using programmable AND.



\* Solved Example:-

$$W(CA, BA, C, D) = \Sigma m(2, 12, 13)$$

$$X(CA, BA, C, D) = \Sigma m(7, 8, 9, 10, 11, 12, 13, 14, 15)$$

AB	CD	$\bar{C}\bar{D}$	$\bar{C}D$	$C\bar{D}$	$CD$
$\bar{A}\bar{B}$		0	1	3	$(1)_2$
$\bar{A}B$		4	5	7	G
AB		1 <sub>12</sub>	1 <sub>13</sub>	15	14
A $\bar{B}$		8	9	11	10

$$\begin{aligned} W = \\ \text{For Pair } (12, 13) \\ AB\bar{C} \dots ① \end{aligned}$$

For Pair (2)

$$\begin{aligned} A\bar{B}C\bar{D} \dots ② \\ = AB\bar{C} + A\bar{B}C\bar{D} \end{aligned}$$

## \* K map for x

<del>AB</del>	<del>CD</del>	<del>CD</del>	<del>CD</del>	<del>CD</del>
<del>AB</del>	0	1	3	2
<del>AB</del>	4	5	7	6
<del>AB</del>	12	13	15	14
<del>AB</del>	18	19	11	10

$$x = A \dots ①$$

$$x = \cancel{BCD} \dots ②$$

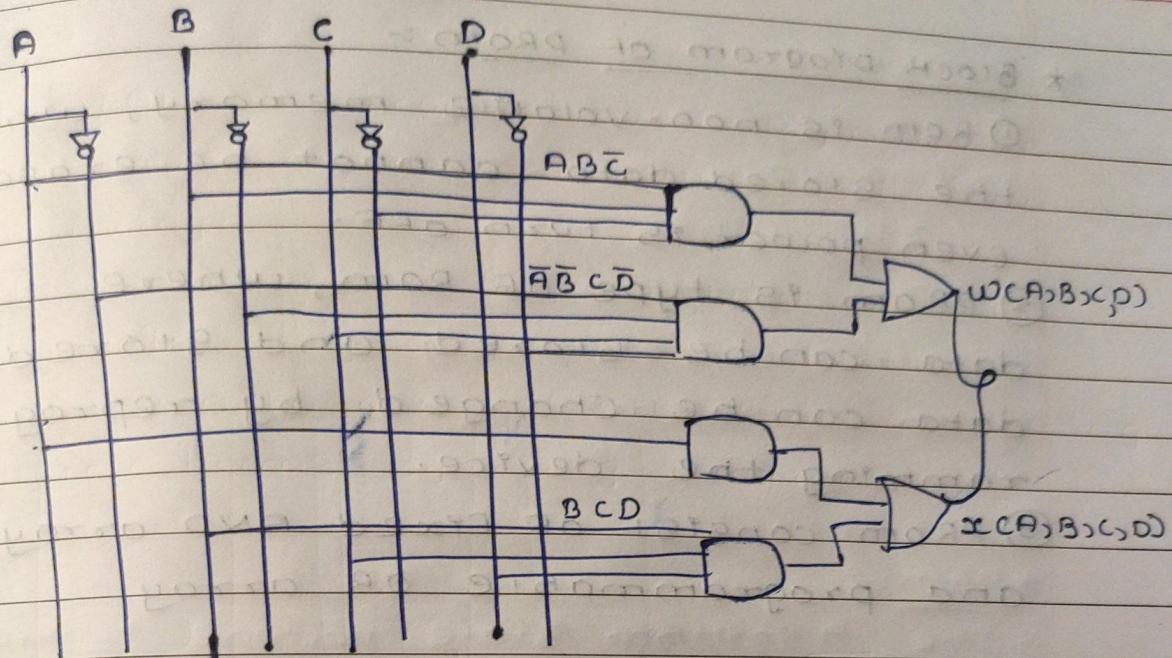
From ① and ②

$$x = A + BCD$$

## \* Programming Table PAL:-

Product Term  $\rightarrow$  Input =  $(A \bar{B} C \bar{D}) (B \bar{C} D) (A \bar{B} \bar{C} \bar{D})$  Output

$A \bar{B} \bar{C}$	1	1	0	-	$y \rightarrow \text{WCA, B, C, D}$
$\bar{A} \bar{B} C \bar{D}$	0	0	1	-	$y \rightarrow \text{SC (A, B, C, D)}$
$A$	1	-	-	-	$y \rightarrow \text{SC (A, B, C, D)}$
$B \bar{C} D$	1	1	1	1	$y \rightarrow \text{SC (A, B, C, D)}$



#### \* Advantages :-

- ① Highly efficient
- ② Highly Secure
- ③ Highly reliability
- ④ they need low Power

#### \* Disadvantages :-

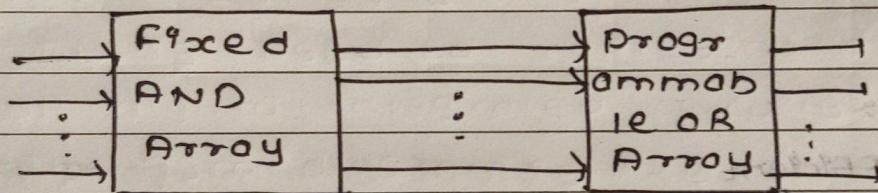
- ① they are expensive
- ② Speed is low ( $F_{max} = 20MHz$ )

#### \* PROM Programmable Read only Memory :-

- The input lines And array are hard drive and output lines to OR array are programmable

\* Block Diagram of PROM :-

- ① ROM is non-volatile memory, where the stored data cannot be erased even power is turn off.
- ② PROM is type of ROM, where data can be stored and stored data can be changed by reprogramming the device.
- ③ PROM consist of fixed AND array and programmable OR array.

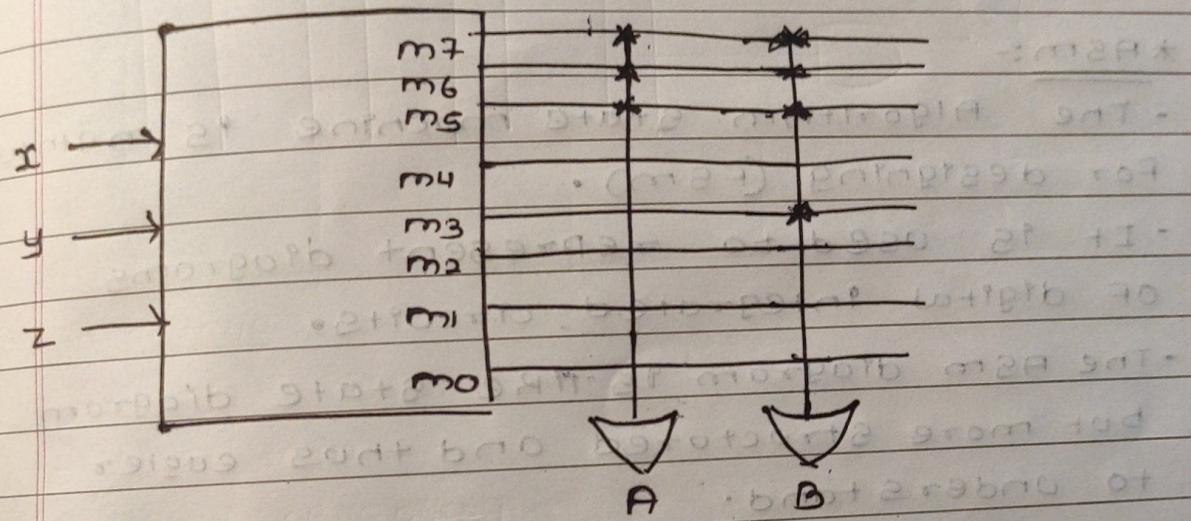
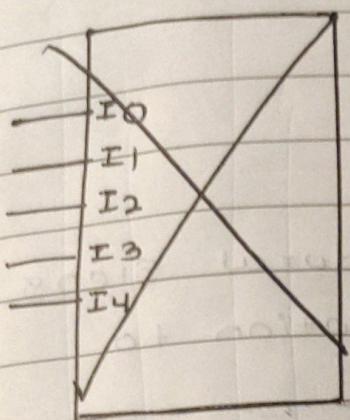


\* Solved Examples:-

- Q. Implement Following boolean Function using PROM.

$$A(x,y,z) = \sum m(5,6,7)$$

$$B(x,y,z) = \sum m(3,5,6,7)$$



## 1.7 COMPARISON OF PROM, PLA AND PAL

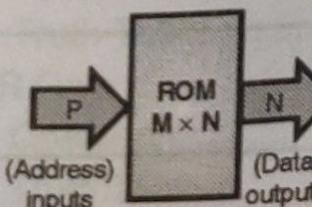
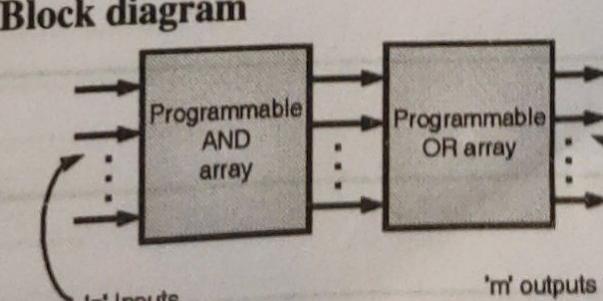
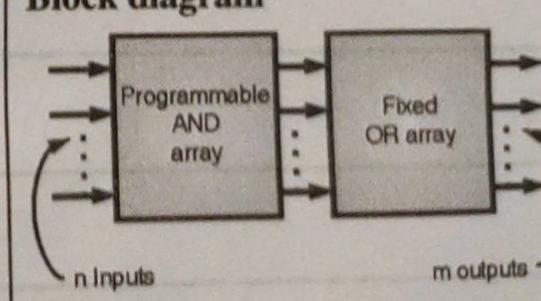
**UQ.** Compare PAL with PLA with suitable examples of logic expressions.

**Q. 6(b), Dec. 15, 10 Marks**

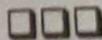
**UQ.** Compare PAL with PLA.

**Q. 1(d), May 16, 5 Marks Q. 1(a), Dec 17, 5 Marks**

We compare ROM/PROM, PLA and PAL as :

Sr. No.	ROM/PROM	PLA	PAL
1.	ROM has fixed AND gate but programmable OR gate array.	PLA has programmable AND gate array and programmable OR gate array.	PAL has programmable AND gate array but fixed OR gate array
2.	All minterms are decoded.	It does not provide full decoding of variables and does not generate all minterms as in the ROM.	AND Array can be programmed to get desired minterms.
3.	It does not require programming like in C and C++. In concept it is similar to PLA	Though its name consists of word 'programmable', it does not require any type of programming like in C and C++.	It does not require any type of programming like in C and C++.
4.	Only Boolean functions in standard SOP form can be implemented using PROM.	Any Boolean function in SOP form can be implemented using PLA.	Any Boolean function in SOP form can be implanted using PAL.
5.	ROM as a combinational circuit  	<b>Block diagram</b>  	<b>Block diagram</b>  

**Fig. 1.7.1**



③ What is ROM? Write a short note on PROM as PLD.

- - ROM stands for Read only memory. It is a type of non-volatile memory that is used primarily in the storage of firmware and software in electronic devices. The data stored in ROM is typically pre-programmed during manufacturing and cannot be modified or written over by normal computer processes.
- PROM on the other hand, stands for programmable read only memory. It is a type of ROM that allows users to program or write data to memory once.
- PLD stands for Programmable logic devices, a broader category that includes various types of programmable device such as PROMs, PALS, PLA. PROM can be considered as specific type of PLD due to their programmable nature, allowing users to define specific logic functions and configurations.

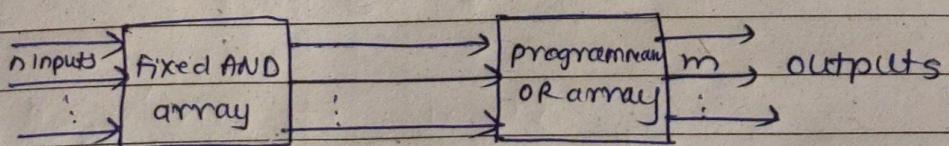


fig. PROM

PLA

Q Define the following multiple output function using PLA [7m]

$$F_1(A, B, C, D) = \Sigma m(3, 7, 8, 9, 11, 15)$$

$$F_2(A, B, C, D) = \Sigma m(3, 4, 5, 7, 10, 14, 15)$$

$\Rightarrow$  (i) To obtain minimal expression

K-map for  $F_1$  :-

		CD	00	01	11	10
		AB	00	01	11	10
AC	BC	00	0	1	1	2
		01	5	4	7	6
AC	BC	11	12	13	15	14
		10	18	19	11	10

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

K-map for  $F_2$  :-

		CD	00	01	11	10
		AB	00	01	11	10
AC	BC	00	0	1	1	2
		01	16	15	1	6
AC	BC	11	12	13	1, 5	14
		10	8	9	11	10

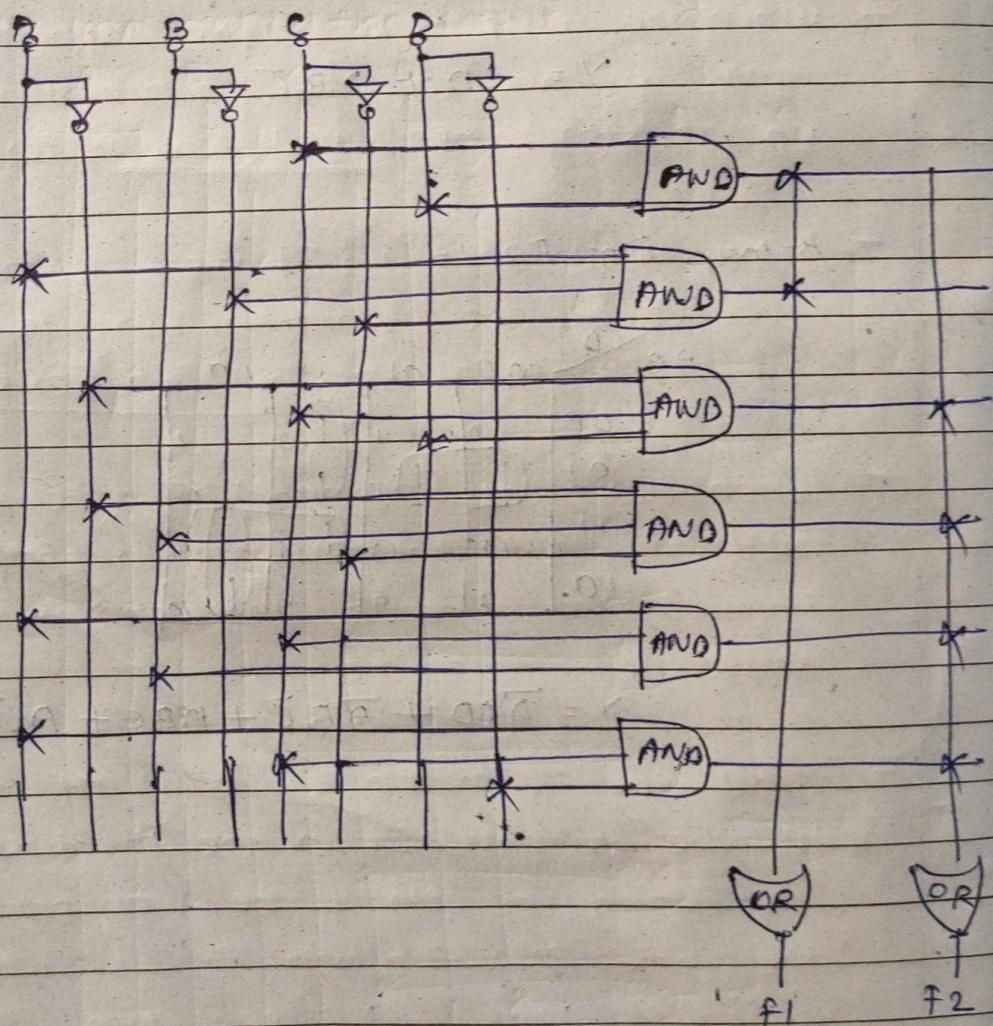
$$Y = \bar{A}CD + \bar{A}\bar{B}\bar{C} + A\bar{B}C + A\bar{C}\bar{D}$$

## (ii) PLA programmable table

product terms	input				outputs	
	A	B	C	D	F1	F2
CD	-	-	1	1	1	0
$A\bar{B}\bar{C}$	1	0	0	-	1	0
$\bar{A}CD$	0	-	1	1	0	1
$\bar{A}\bar{B}\bar{C}$	0	1	0	-	0	1
$A\bar{B}C$	1	1	1	-	0	1
$AC\bar{D}$	1	-	1	0	0	1

fig. PLA programmable table.

## (iii) Implementation of PLA

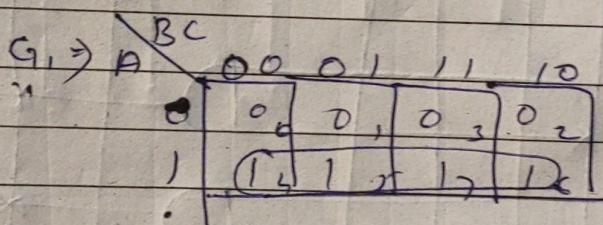


→ Implement 3-bit binary to gray code converter using PLA.

⇒ (i) Truth table

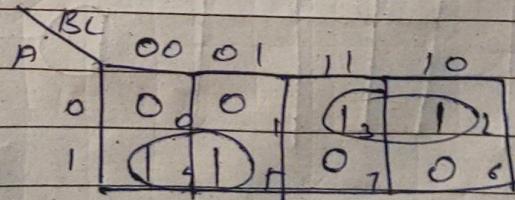
	Inputs			Outputs		
	A	B	C	G <sub>1</sub>	G <sub>2</sub>	G <sub>3</sub>
0	0	0	0	0	0	0
1	0	0	1	1	0	1
2	0	1	0	0	1	1
3	0	1	1	0	1	0
4	1	0	0	1	1	0
5	1	0	1	1	1	1
6	1	1	0	0	0	1
7	1	1	1	1	0	0

(ii) K-map



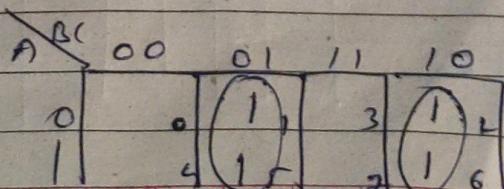
$$G_1 = A$$

$G_2 \Rightarrow$



$$G_2 = \overline{AB} + A\overline{B}$$

$G_3 \Rightarrow$

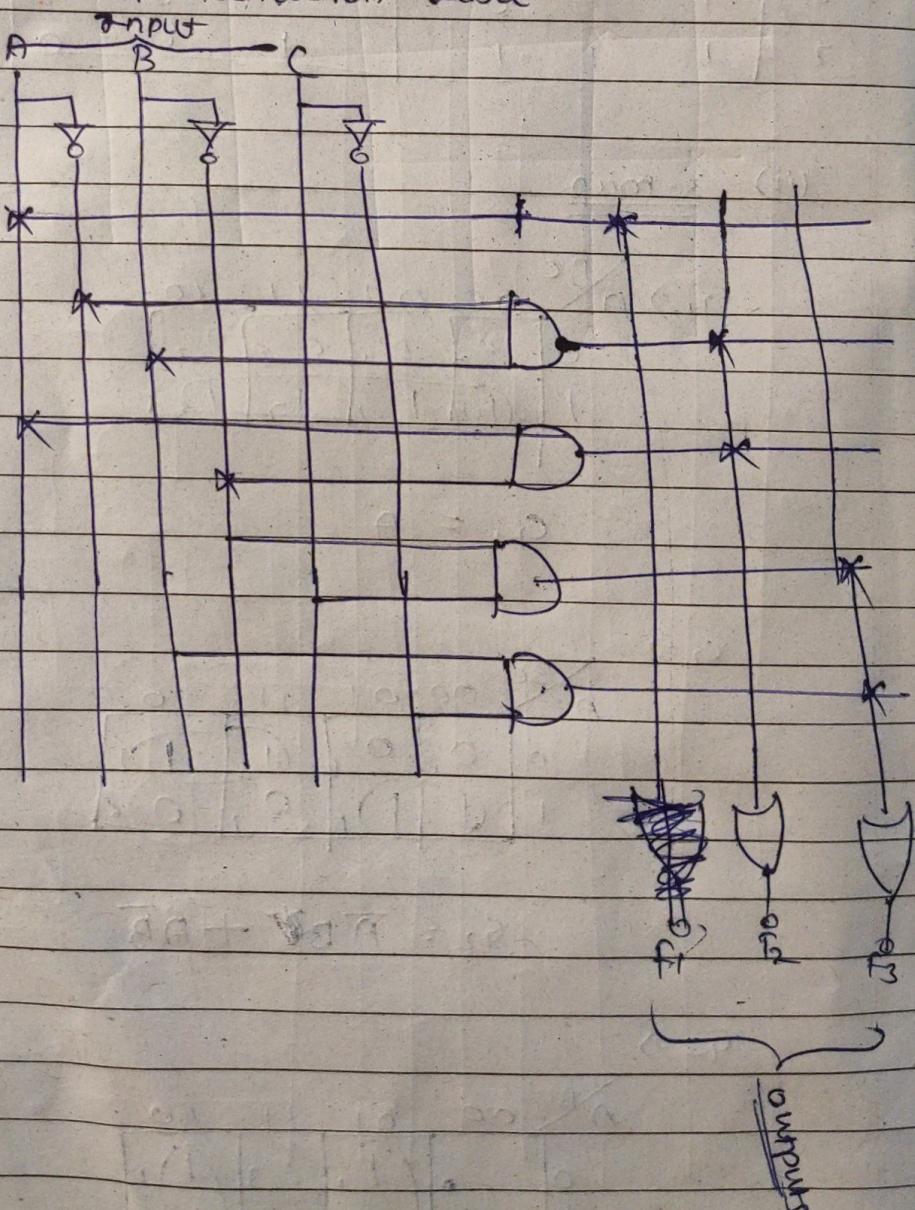


$$G_3 = \overline{BC} + \overline{B}\overline{C}$$

Programmable  
(iii) Implementation table

Product term	Input			Output		
	A	B	C	$f_1$	$f_2$	$f_3$
A	1	-	-	1	0	0
$AB$	0	1	-	0	1	0
$A\bar{B}$	0	0	1	0	1	0
$\bar{B}C$	-	0	1	0	0	1
$B\bar{C}$	0	-	0	0	0	1

(iv) Implementation table



(1)

Implement following boolean function using PAL

$$F_1 = \Sigma m(0, 2, 3, 4, 5, 6, 7, 8, 10, 11, 15)$$

$$F_2 = \Sigma m(1, 2, 8, 12, 13)$$



(i) Minimize expression

(F1)

		CD	00	01	11	10
		AB	00	01	11	10
00	00	D <sub>0</sub>	1	1	1	1
		D <sub>4</sub>	1	1	1	1
01	01	D <sub>5</sub>	1	1	1	1
		D <sub>6</sub>	1	1	1	1
11	11	D <sub>7</sub>	1	1	1	1
		D <sub>12</sub>	1	1	1	1
10	10	D <sub>8</sub>	1	1	1	1
		D <sub>13</sub>	1	1	1	1

$$Y = \overline{B} \overline{D} + \overline{C} D + \overline{A} B$$

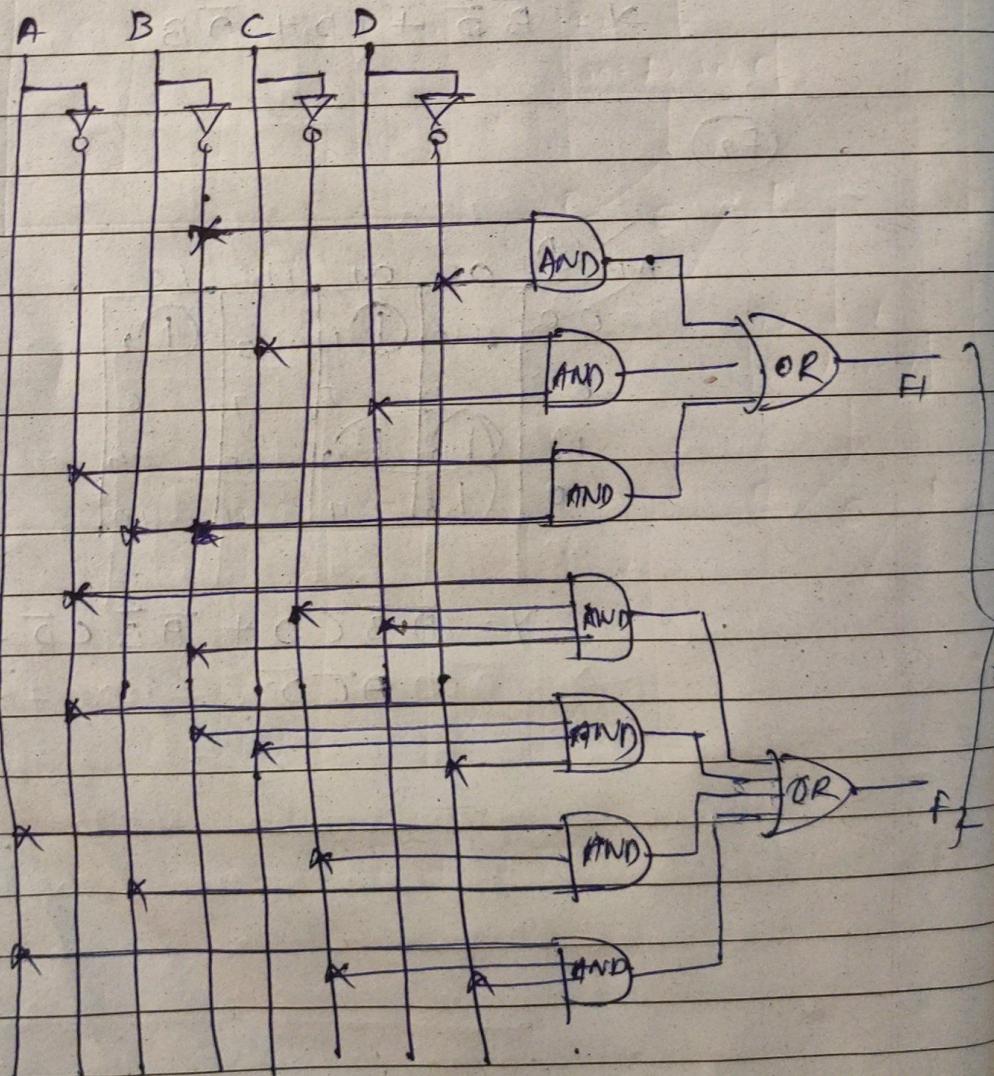
(F2)

		CD	00	01	11	10
		AB	00	01	11	10
00	00	0	1	1	1	1
		4	1	1	1	1
01	01	5	1	1	1	1
		6	1	1	1	1
11	11	12	1	1	1	1
		13	1	1	1	1
10	10	14	1	1	1	1
		15	1	1	1	1

$$Y = \overline{A} \overline{B} \overline{C} D + \overline{A} \overline{B} C \overline{D} + A B \overline{C} + A C \overline{D}$$

## (iii) Programmable table for PAL

product terms	input				output
	A	B	C	D	
$BD$	+	0	-	0	{
$CD$	-	-	1	1	$F_1$
$\bar{AB}$	0	1	0	-	}
$\bar{ABC}D$	0	0	0	1	{
$\bar{ABC}\bar{D}$	0	0	1	0	$F_2$
$ABC$	1	1	0	-	}
$AC\bar{D}$	1	-	0	0	

(iii) Implementation ~~table~~

→ Design BCD to Excess 3 code converter  
and implement it using AAL.

→ (i) Truthtable of BCD to Excess 3 code:-

	BCD				Excess 3			
	A	B	C	D	E <sub>1</sub>	E <sub>2</sub>	E <sub>3</sub>	E <sub>4</sub>
0	0	0	0	0	0	0	1	1
1	0	0	0	1	0	1	0	0
2	0	0	1	0	0	1	0	1
3	0	0	1	1	0	1	1	0
4	0	1	0	0	0	1	1	1
5	0	1	0	1	1	0	0	0
6	0	1	1	0	1	0	0	1
7	0	1	1	1	1	0	1	0
8	1	0	0	0	1	0	1	1
9	1	0	0	1	1	1	0	0

(ii) K-map minimal expression <sup>find</sup>

G $\Rightarrow$		CD				
		AB	00	01	11	10
AB	CD	00	0	1	3	1
00	00	0	1	1	3	1
01	01	1	1	1	1	0
11	11	X	X	X	X	X
10	10	1	0	1	0	1

$$Y = A + BD + BC$$

$E_2 \Rightarrow$ 

$AB \backslash CD$	00	01	11	10
00	0	1	1	1
01	1	5	7	6
11	X <sub>12</sub>	X <sub>13</sub>	X <sub>15</sub>	X <sub>14</sub>
10	8	9	X <sub>11</sub>	X <sub>10</sub>

$$Y = \overline{B}D + \overline{B}C + BC\overline{P}$$

 $E_3 \Rightarrow$ 

$AB \backslash CD$	00	01	11	10
00	1	1	1	2
01	1	5	1	6
11	X <sub>12</sub>	X <sub>13</sub>	X <sub>15</sub>	X <sub>14</sub>
10	1	9	X <sub>11</sub>	X <sub>10</sub>

$$Y = \overline{C}\overline{D} + CD$$

 $E_4 \Rightarrow$ 

$AB \backslash CD$	00	01	11	10
00	1	0	1	2
01	1	4	5	6
11	X <sub>12</sub>	X <sub>13</sub>	X <sub>15</sub>	X <sub>14</sub>
10	1	9	X <sub>11</sub>	X <sub>10</sub>

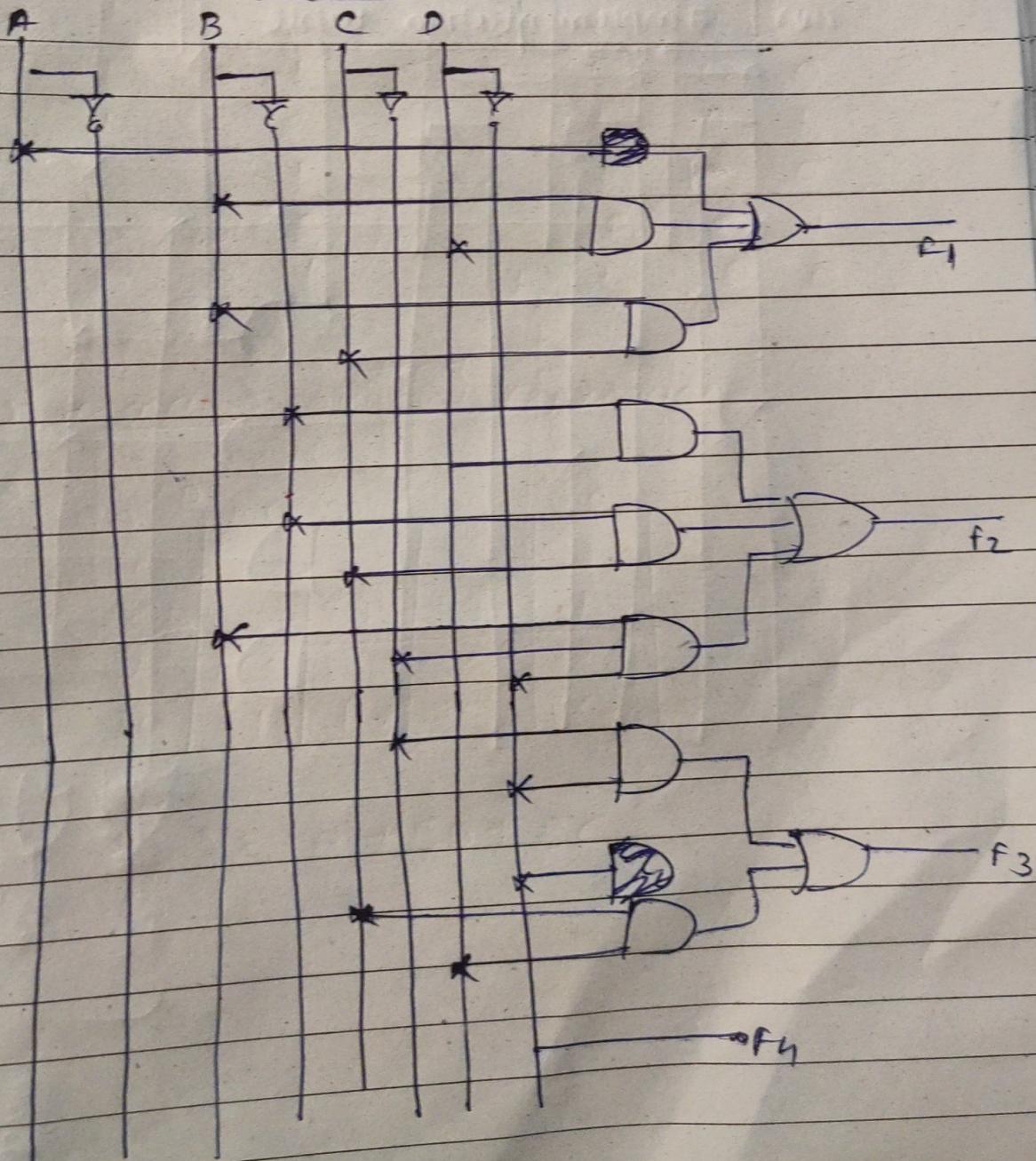
$$Y = \cancel{\overline{CD}} + \cancel{CD}$$

$G_4 \Rightarrow$ 

<del>AB</del> CD	00	01	11	10
00	10	1		12
01	14	r	7	16
11	x	x <sub>3</sub>	x <sub>15</sub>	x <sub>13</sub>
10	1	9	x <sub>11</sub>	x <sub>10</sub>

$$Y = \overline{D}$$

### (iii) Implementation



## (iii) programmable table PPL

Product term Entered	Input	Output
	A B C D	
1	1 - - -	
A	- 1 - 1	{ F1
BD	- 1 1 -	
BC	- 0 - 1	{ F2
$\bar{B}D$	- 0 1 -	
$\bar{B}C$	- 1 0 0	
$B\bar{C}\bar{D}$	- - 0 0	{ F3
$\bar{C}\bar{D}$	0 0 1 1	
CD	- - - 0	{ F4
$\bar{P}$		

Algorithm  
State  
Machine

# ASM Chart

Rainbow

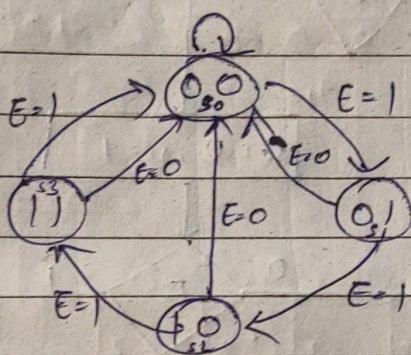
PAGE: / /  
DATE: / /

- \* Design the ASM chart for a 2-bit binary counter having one enable line E such that when:

$E = 1$  (count enable)

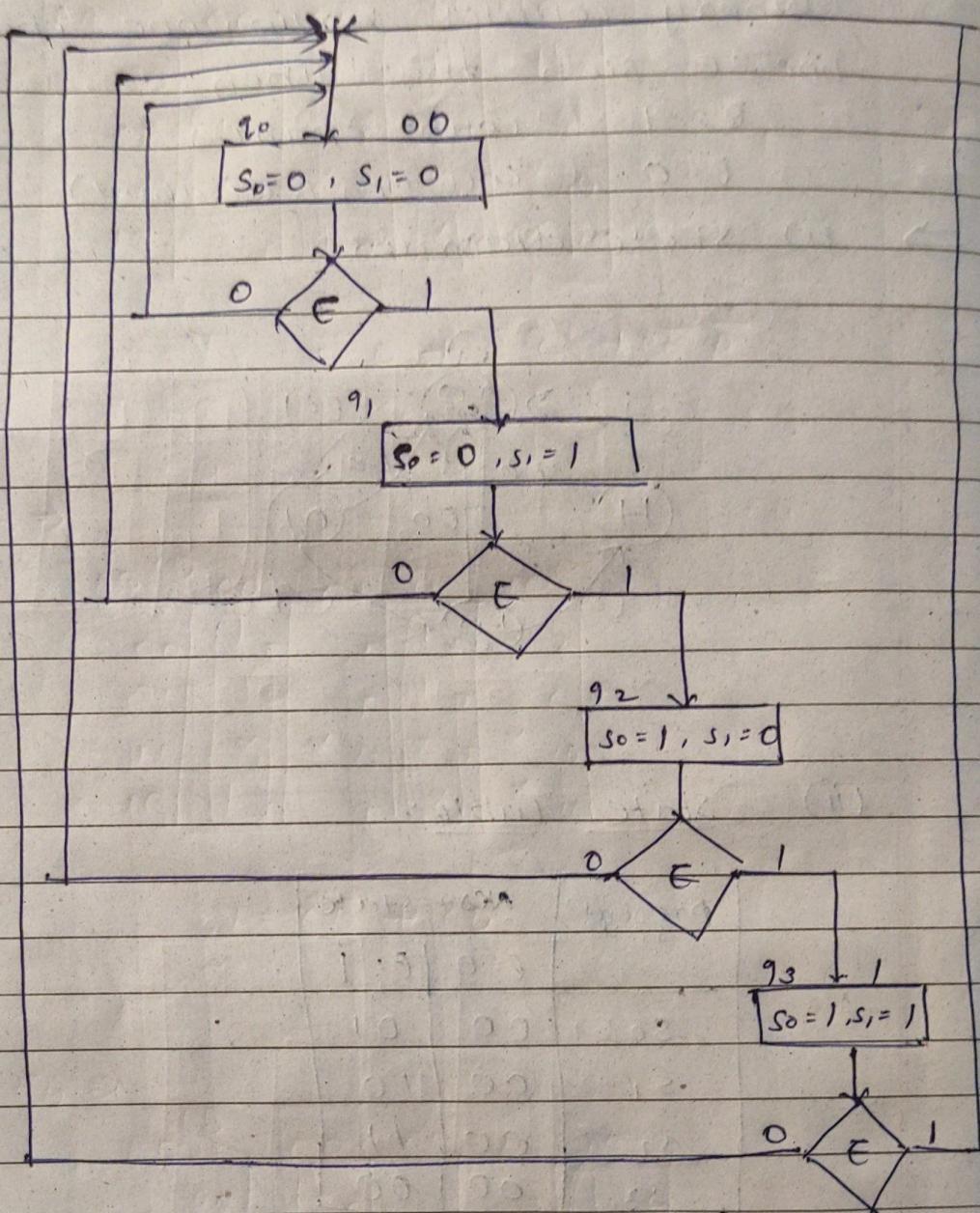
$E = 0$  (counting is disabled)

⇒ (i) state diagram



(ii) State Table :-

Present State	Next state	
	$E=0$	$E=1$
$S_0$	00	01
$S_1$	00	10
$S_2$	00	11
$S_3$	00	00

(iii) ASM chart

UNIT - III

1) what do you mean by excitation table of flip-flop?

- ⇒ - The excitation table has the minimum inputs, which will excite or trigger the flip flop to go from its present state to the next state. It is derived from the truth table.
- The excitation table consists of two columns for the present state ( $Q_n$ ) and the next state ( $Q_{n+1}$ ) and one or two columns for each input. The input column depends on the type of flip flop.

2) distinguish between combinational and sequential switching circuits.

- Combinational Sequential

definition

A combinational circuit is a digital circuit type where the O/P is only a pure function of present I/P. It is a digital circuit type whose O/P relies not just on current values of the I/P signal it has but it depends on the past sequence of inputs as well.

memory unit

Memory device is absent

An integrated storage unit for sequential circuits can store instant results.

Output A circuit's o/p from a combined circuit relies on the i/p at present the output of sequential circuit will depend on the recent o/p and current i/p

Clock There is no combinational clock present in a circuit. However the possibility of a clock is there in the sequential circuit.

e.g. Demultiplexer, decoder, full adder encoder, and half adder are examples of sequential circuits.

3) Difference between synchronous and asynchronous counter.



Synchronous counter

(i) In synchronous counter there are continuous clock input signals with flip-flops used to produce the output.

(ii) In synchronous counter, the operation is faster.

(iii) Synchronous counter is also known as parallel counter.

(iv) It produces less error than asynchronous counter.

(v) Design is complex.

(vi) It can work with flexible no. of count sequences.

Asynchronous counter

(i) In asynchronous counters there are different clock signals used to produce the output.

(ii) In asynchronous counter the operation is slower.

(iii) Asynchronous counter is also known as serial counter.

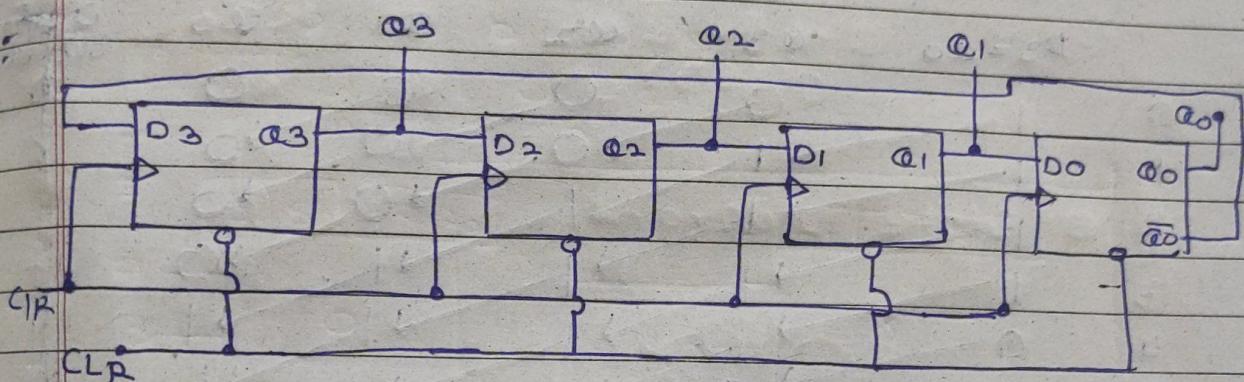
(iv) It produces more error.

(v) Design is simple.

(vi) It can work with a fixed no. of count sequences.

\* Johnson Counter:

- The construction of Johnson Counter is similar to ring counter except the complement of last flip flop output is connected to input of 1st flip flop.



\* Circuit Operation:

• Before the clock is applied:-

- All FFS are RESET using low CLR input. Then CLR is made high.

$$Q_3 \ Q_2 \ Q_1 \ Q_0 = 0000$$

As  $\bar{Q}_0$  is connected to  $D_3$ , input before 1st clock is

$$D_3 \ D_2 \ D_1 \ D_0 = 1000$$

• After 1st clock pulse:-

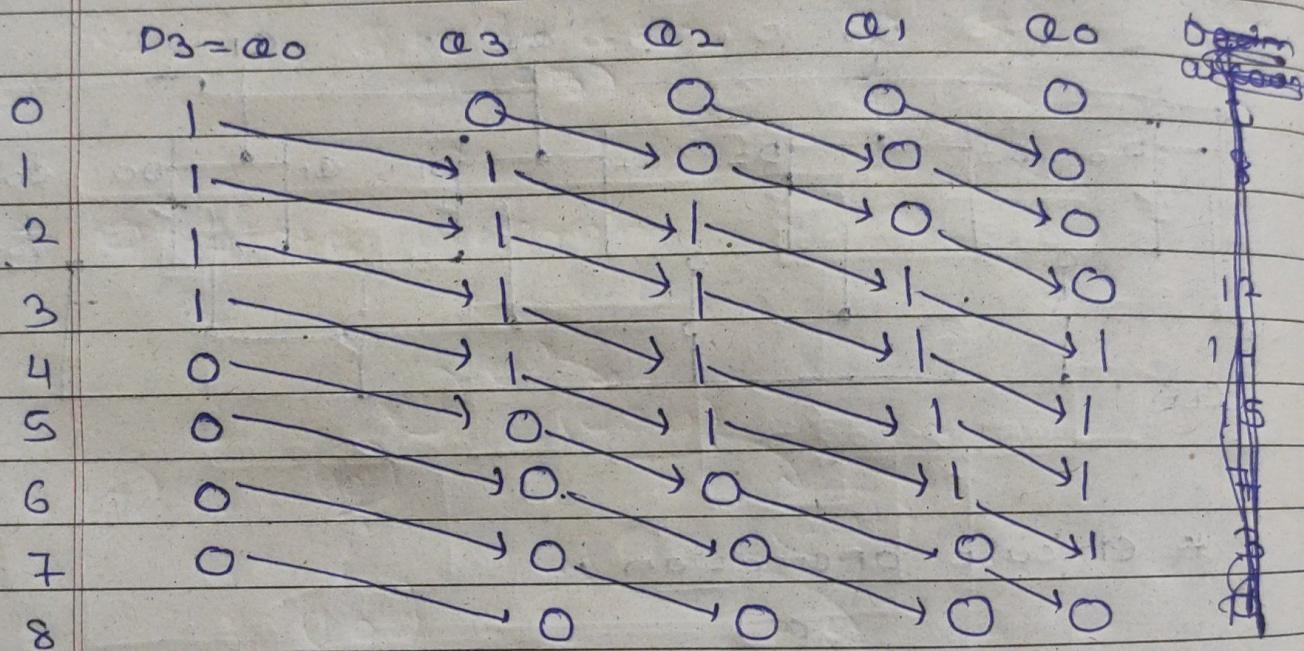
$$Q_3 \ Q_2 \ Q_1 \ Q_0 = 1000$$

As  $\bar{Q}_0$  is connected to  $D_3$ , input before 2nd CLK is

$$D_3 \ D_2 \ D_1 \ D_0 = 1100$$

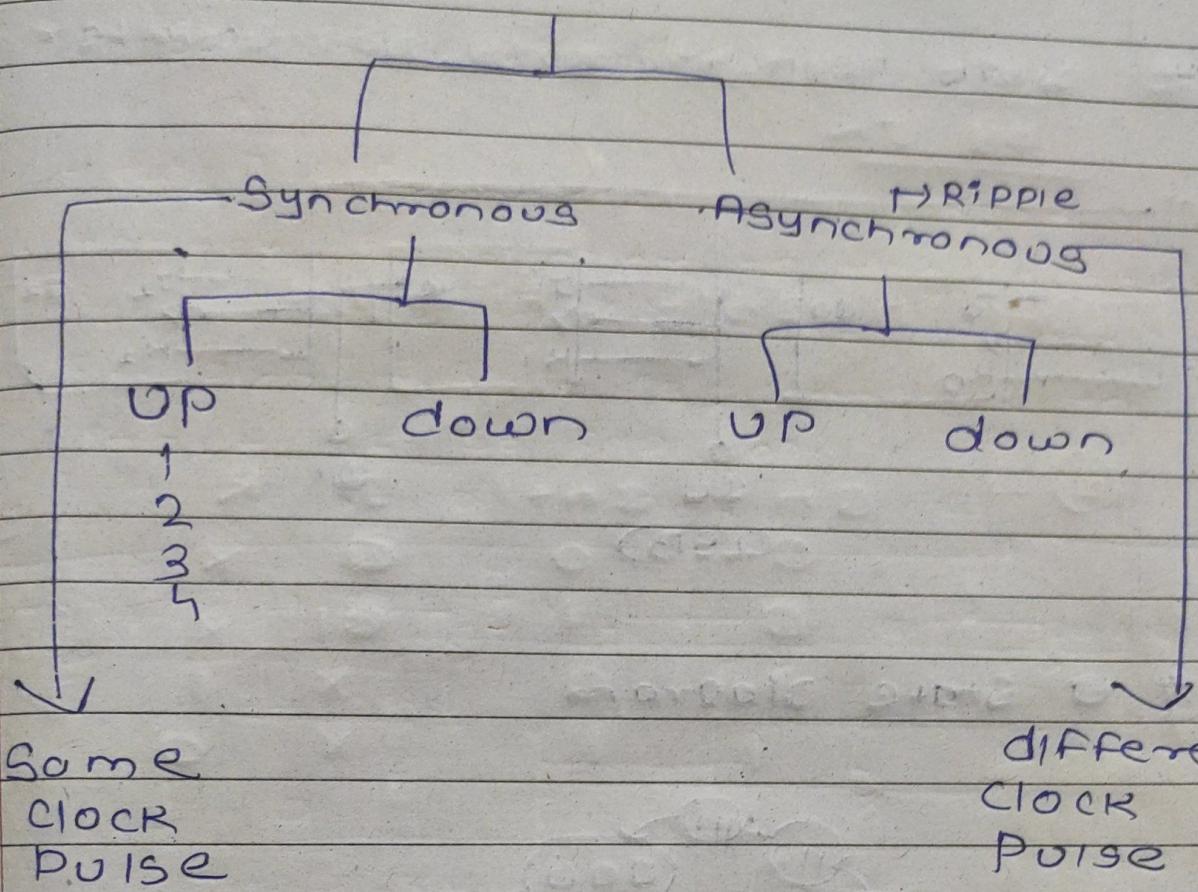
\* AFTER 2<sup>nd</sup> CLOCK PULSE -  
 $\textcircled{Q}_3 \textcircled{Q}_2 \textcircled{Q}_1 \textcircled{Q}_0 = 1100$ .

\* TRUTH TABLE OF Johnson Counter:-

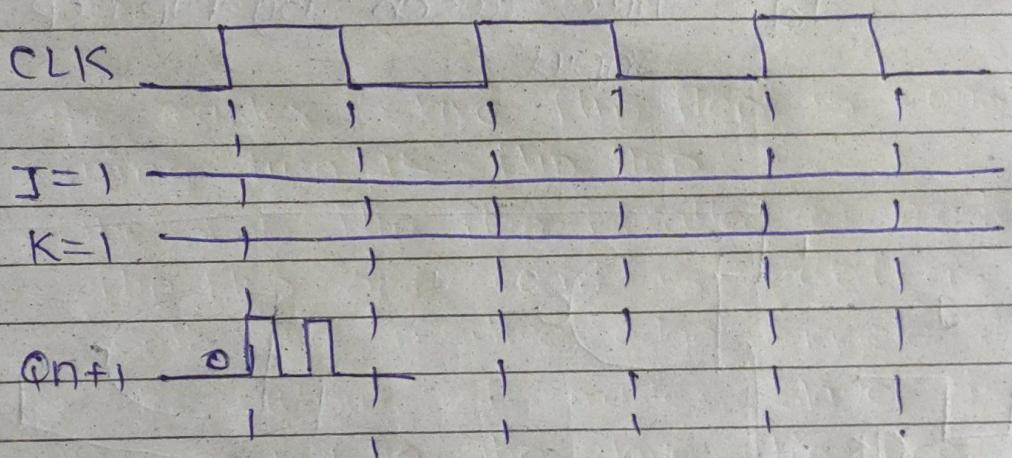
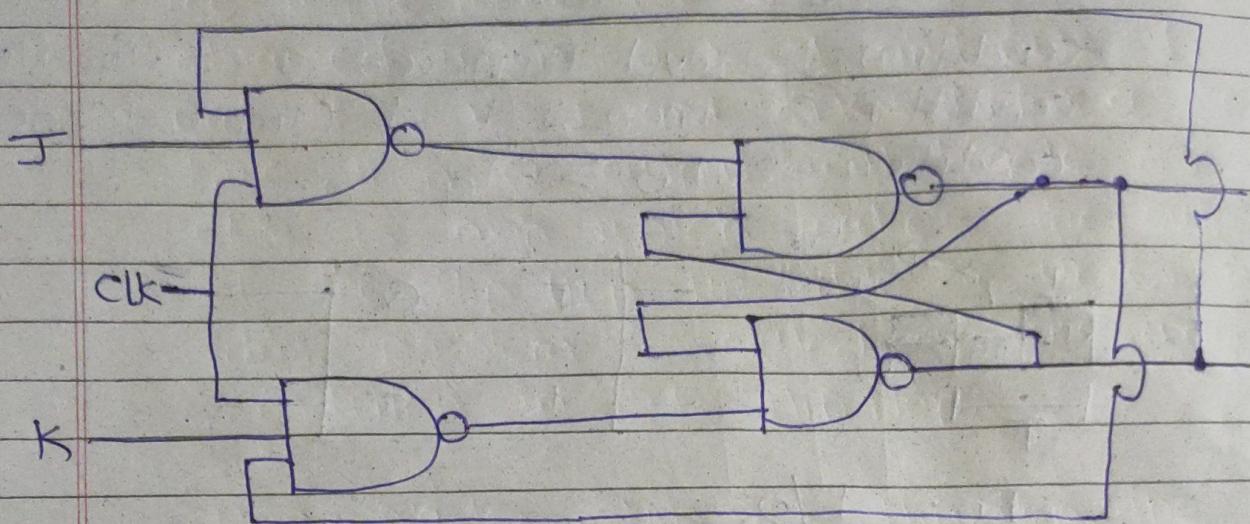


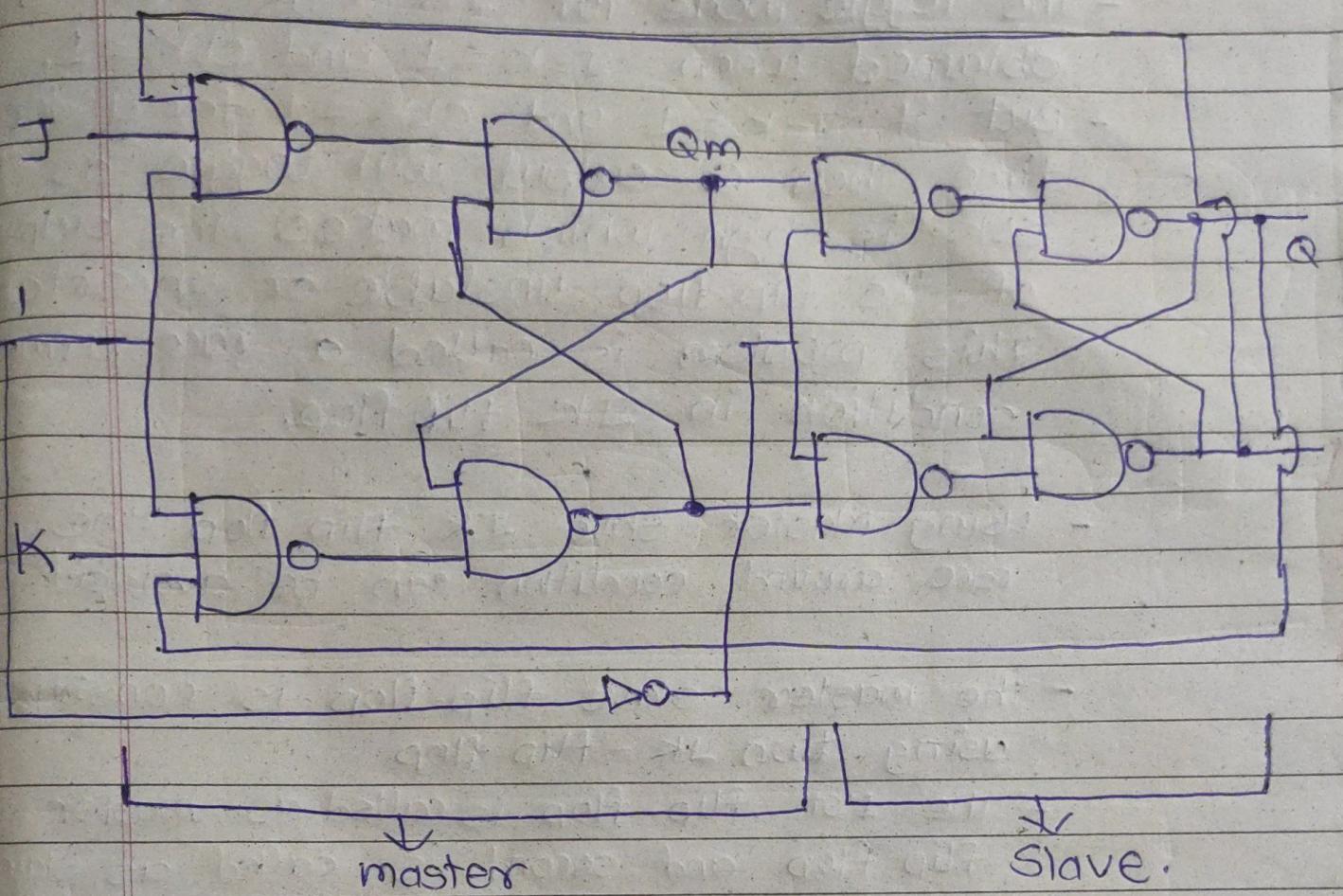
## Counter

K

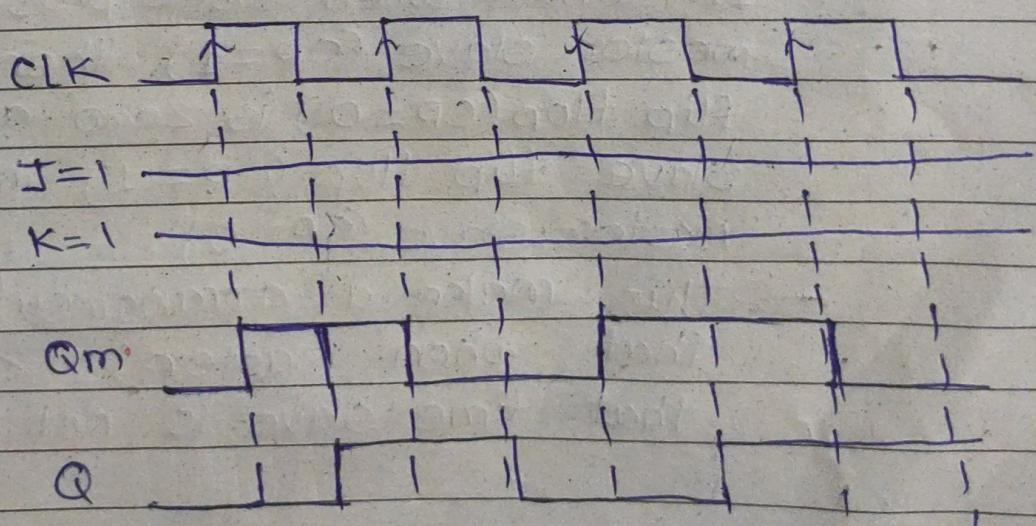


\* Master Slave JK flip-flop.





- when master slave clock is Active and then slave clock is is passive and when slave part is Active master will be passive.



J=1    K=1  
 Qm Active  
 Q Inactive.

- The toggle mode for JK flip-flop is obtained when  $J=K=1$  and  $CLK=1$ .
- But if  $J=K=1$  and  $CLK=1$  for a long time, then Q output will toggle as CLK is high which makes the output of the flip flop unstable or uncertain. This problem is called a race around condition in JK flip flop.
- Using Master Slave JK flip flop the race around condition can be avoided.
- The master slave flip-flop is constructed using two JK-flip flop.
- The first flip-flop is called as master flip flop and second is called as slave.
- Both are positive edge triggered.
- The output of master slave is connected to the slave flip-flop.
- In addition to those flip-flop one inverter is connected to clock pulse of master slave ( $CP=1$ ) then slave flip flop ( $CP=0$ ) is zero and when slave flip flop ( $CP=1$ ) then master slave ( $CP=0$ )
- This make a arrangement such that when master is active that time slave is not active.

## \* Race Around Condition

- when  $J=K=1$  then race around condition occurs.
- It occurs when the time period of the clock pulse (pulse width) is greater than the propagation delay of the flip flop due to that output change or toggle in a single clock period.
- propagation delay is the speed of operation of circuit.
- if toggle even number of times, then the output is same but if it toggles odd number of times then output is complement.

## \* How to avoid it ?

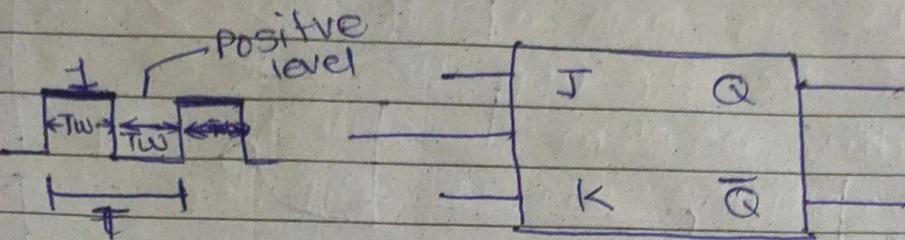
- We have two options
- 1. Master Slave JK FF
- 2. Edge triggered JK FF

## \* Race Around Condition

Condition 1: Level triggered J-K flip flop

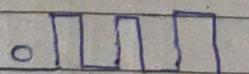
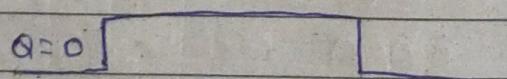
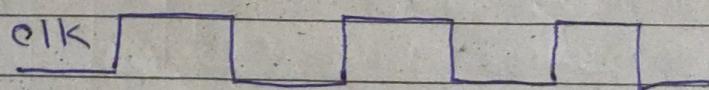
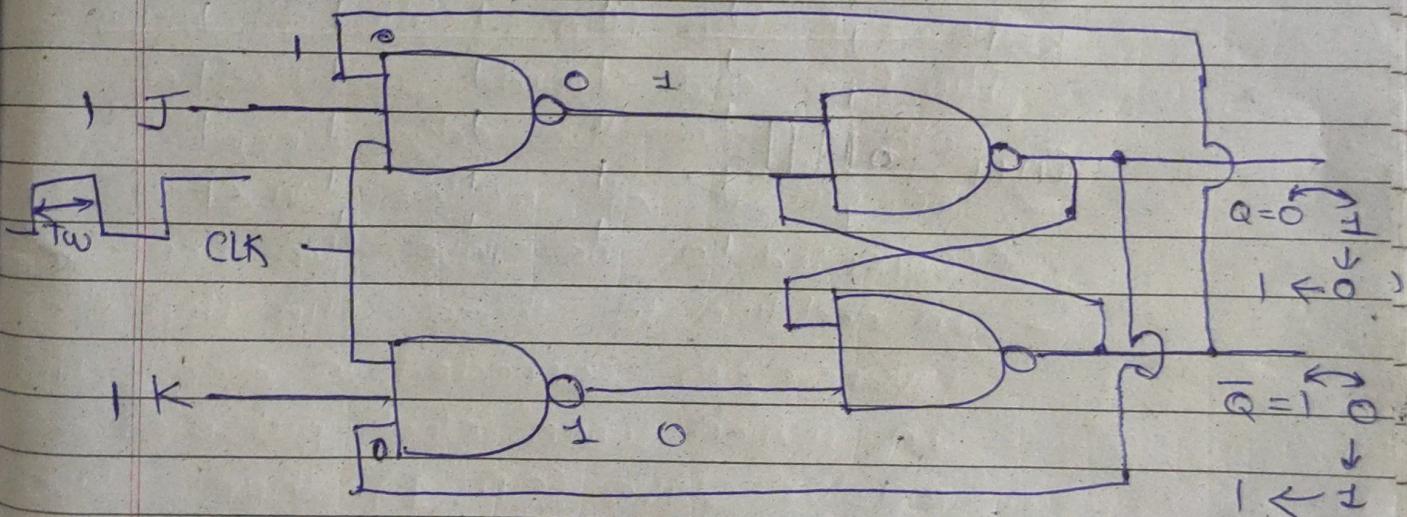
Condition 2: when  $J=K=1$  (Toggle Mode)

Condition 3:  $T_W \gg T_d$



$$T_W = T/2$$

$T_d$  = time taken by flip-flop to process input

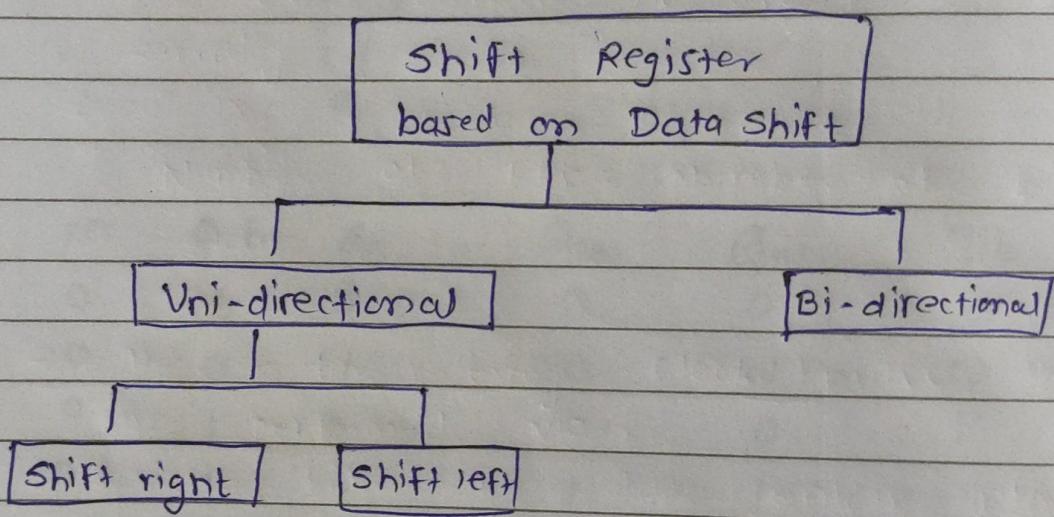


Q. How are shift registers classified?

→ Shift registers are classified in 2 categories.

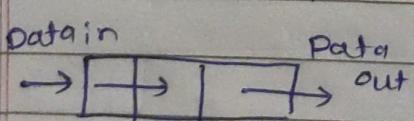
- 1) Depending on direction of data shift.
- 2) Depending on mode of data in or out.

category 1:

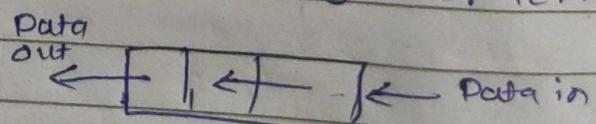


a) Uni-directional Shift register:

- In this type, data bits only in one direction i.e. either towards right or left.



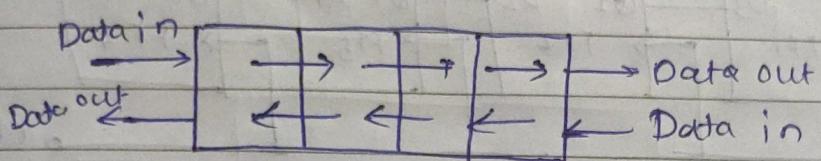
Right shift  
register



Left shift  
register.

### b) Bi-directional shift register:

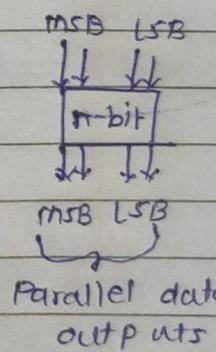
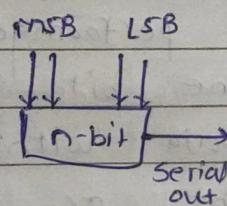
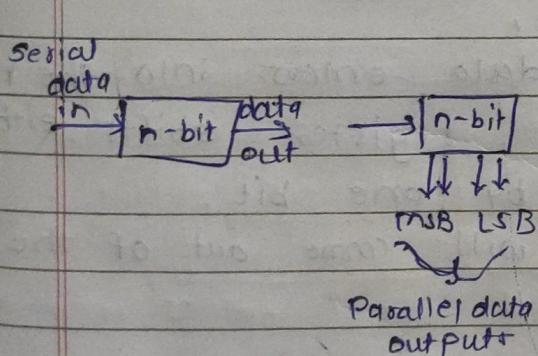
- In this type, data bit moves in both the directions either left or right.



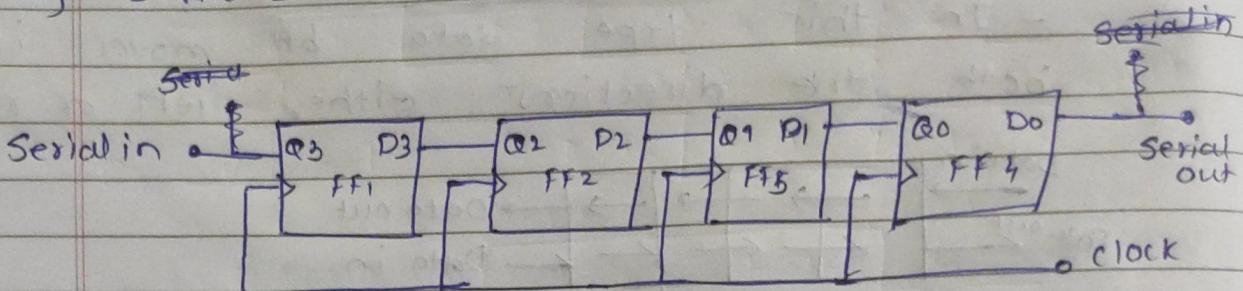
Category 2: Depending on the mode of Data In/out.

#### Classification of shift registers

Serial-In Serial-Out (SISO)	Serial-in Parallel-out (SIPO)	Parallel-In Serial-Out (PISO)	Parallel-In Parallel-out (PIPO)
-----------------------------------	-------------------------------------	-------------------------------------	---------------------------------------



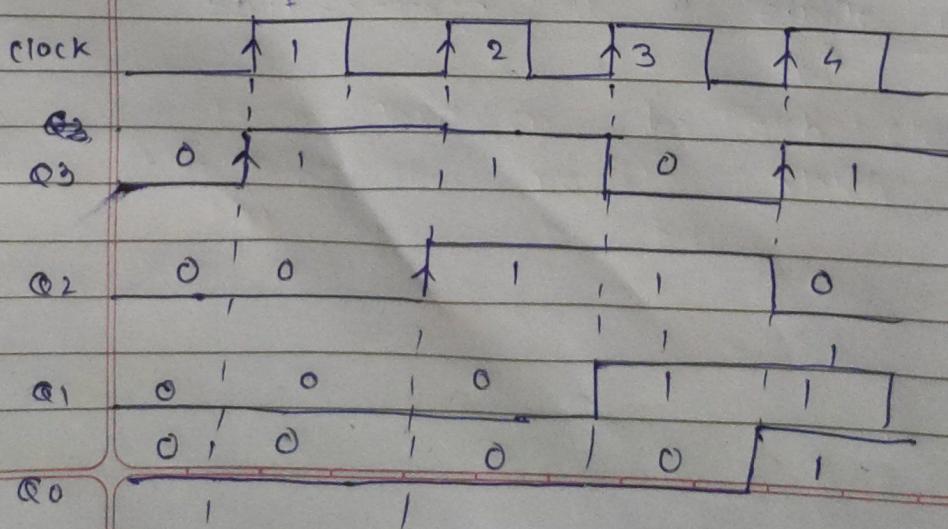
### 1) Serial In Serial Out (SISO)



- As the name suggests, the data enters into the shift register serially & the output is also taken serially.

Number of FFs = Number of bits to store

- Here (for) every clock pulse three functions are performed. viz.,
  1. one bit of data enters into the register.
  2. Data within the register shifts either right or left by one bit.
  3. One data bit will come out of the register.



Initial = 0000

After = 1000

After = 1100

After = 0110

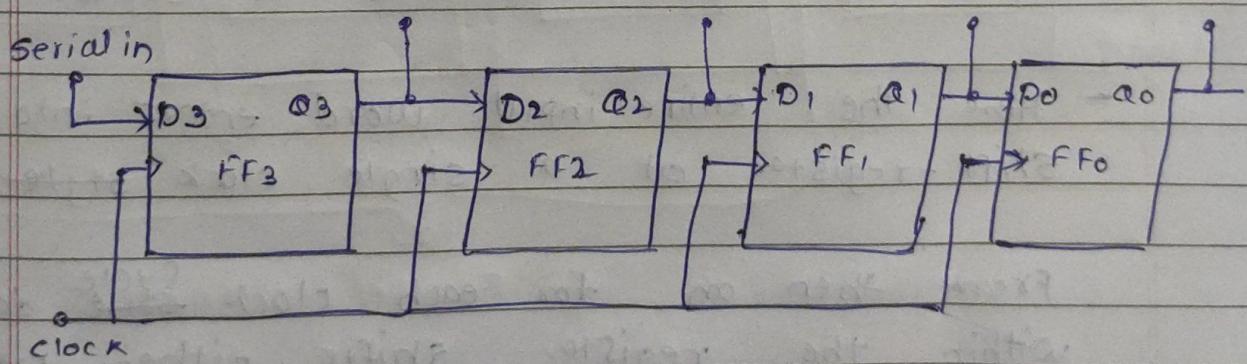
After = 1011

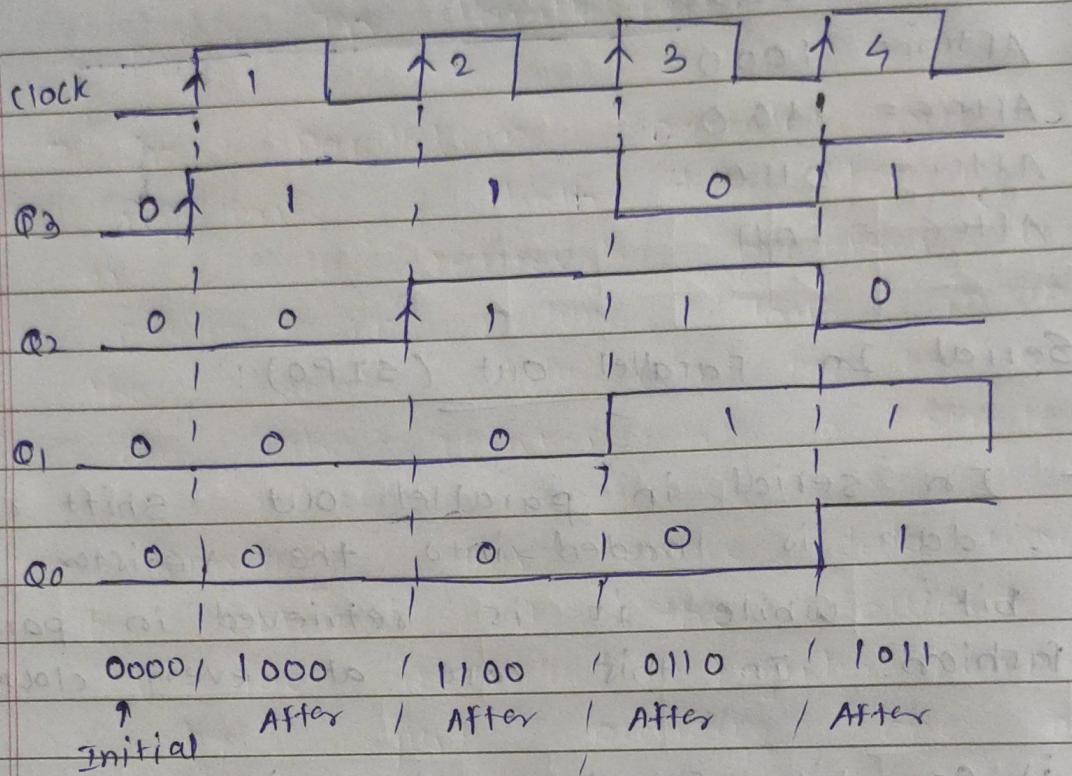
## 2) Serial In Parallel Out (SIPO):

- In serial in parallel out shift registers, data is loaded into the register bit by bit while it is retrieved in parallel fashion. In this case, at every clock pulse

- i) One bit of data enters into the register.
- ii) Data within the register shifts either right or left by one bit.

- After loading the data, the output bits can be read-out in parallel.

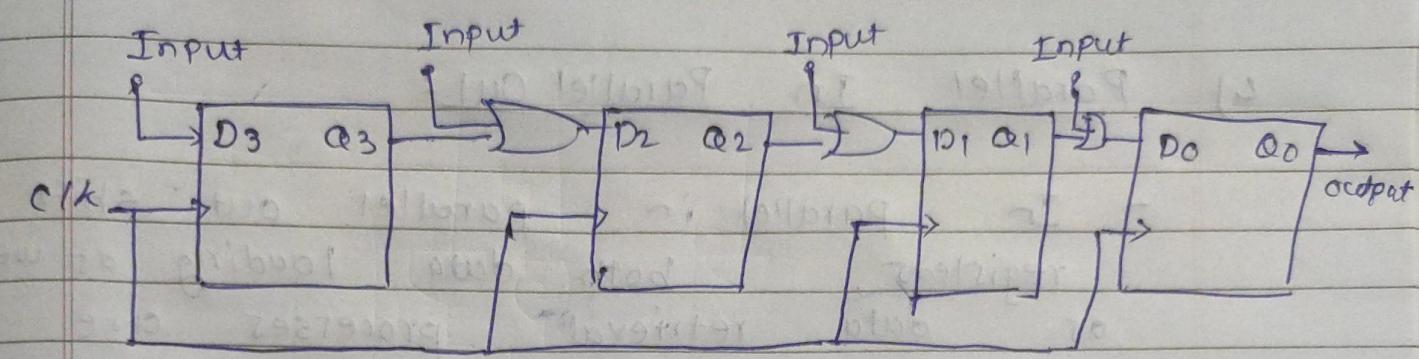




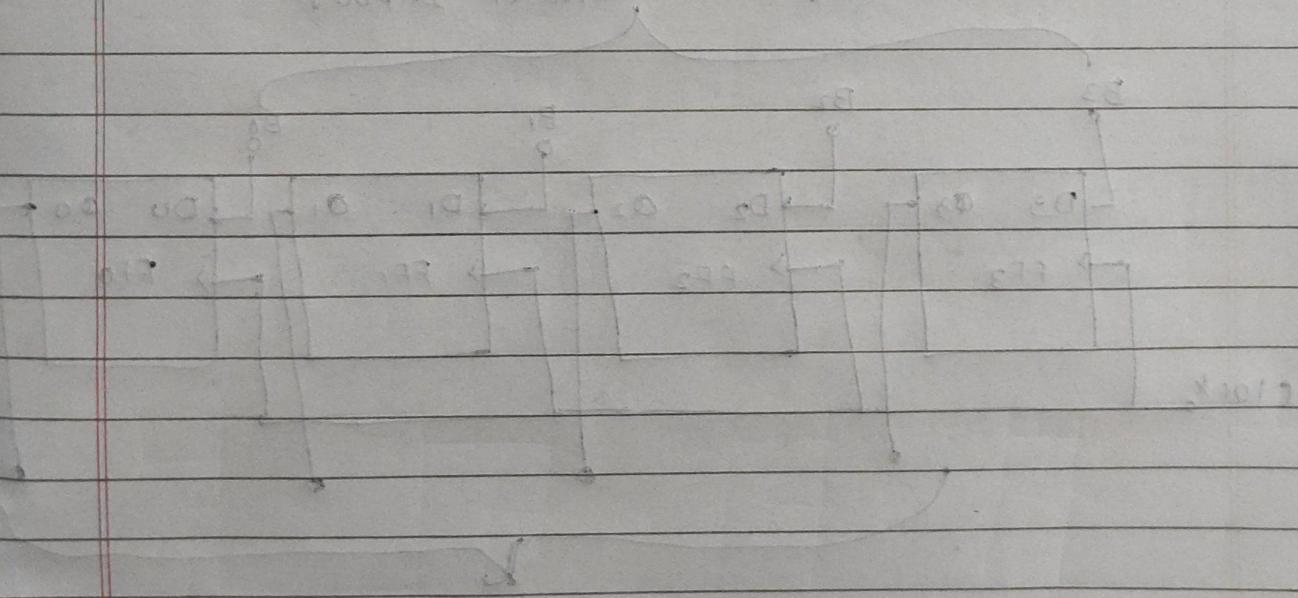
### 3) Parallel In Serial Out (PISO):

- In case of parallel in serial out shift register, the data is loaded in parallel fashion while the data retrieval is serial.
- Here the entire input word enters into the shift-register at a single clock cycle.

From then on, for each clock cycle data within the register shifts either right or left by one bit.



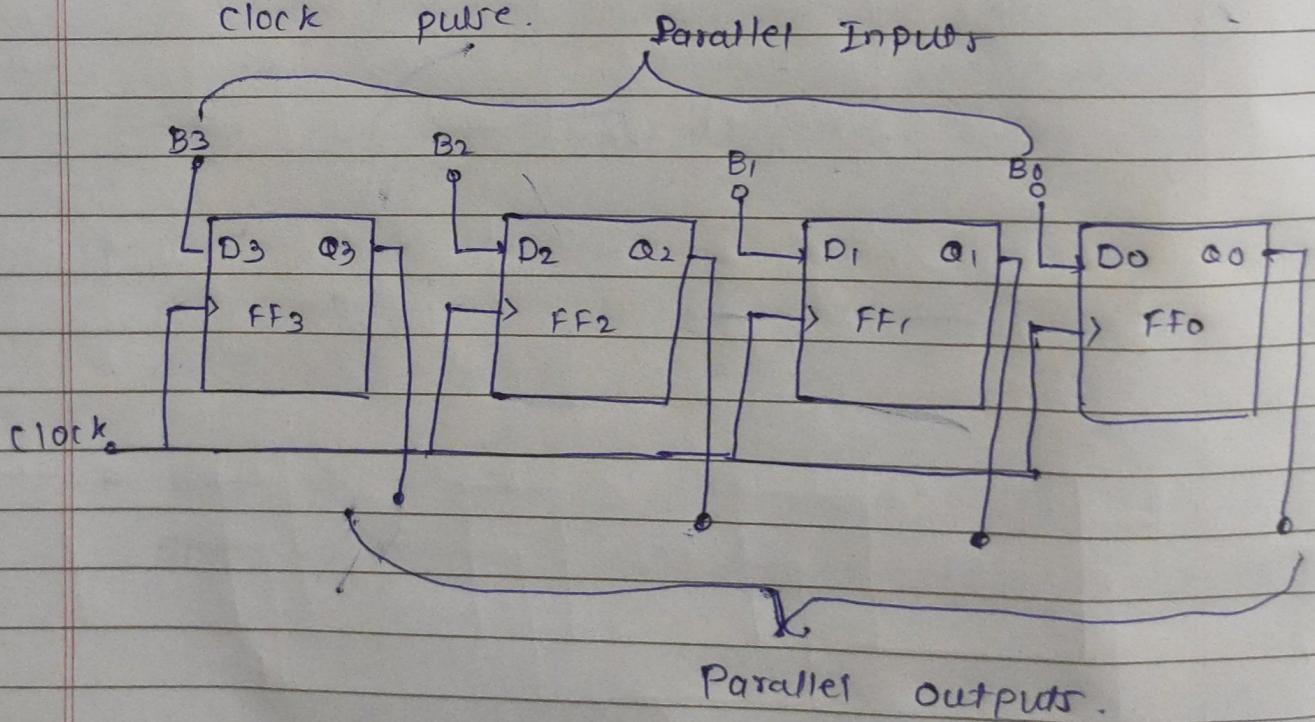
also shows other units such as register, multiplexer etc.



state transition diagram

#### 4) Parallel In Parallel Out:

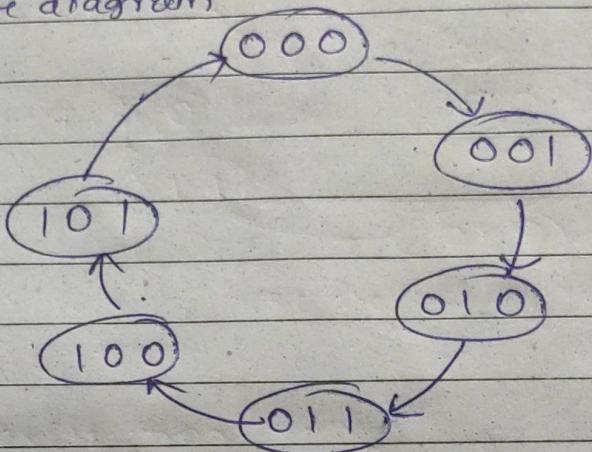
- In parallel in parallel out shift registers, both data loading as well as data retrieval processes are parallel in nature. This means that the entire data word can be entered into the registers at a single clock.
- Similarly this entire data word can be obtained at the output pins of the individual register components by just providing one more clock pulse.



ASYNCHRONOUS COUNTER~~6M~~

MOD 6 asynchronous counter using JK flip flops.

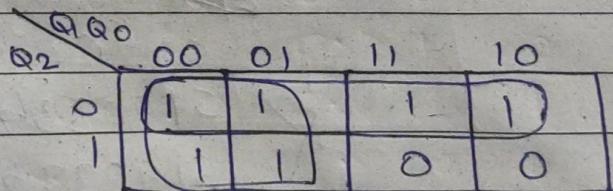
(i) state diagram



(ii) truth table

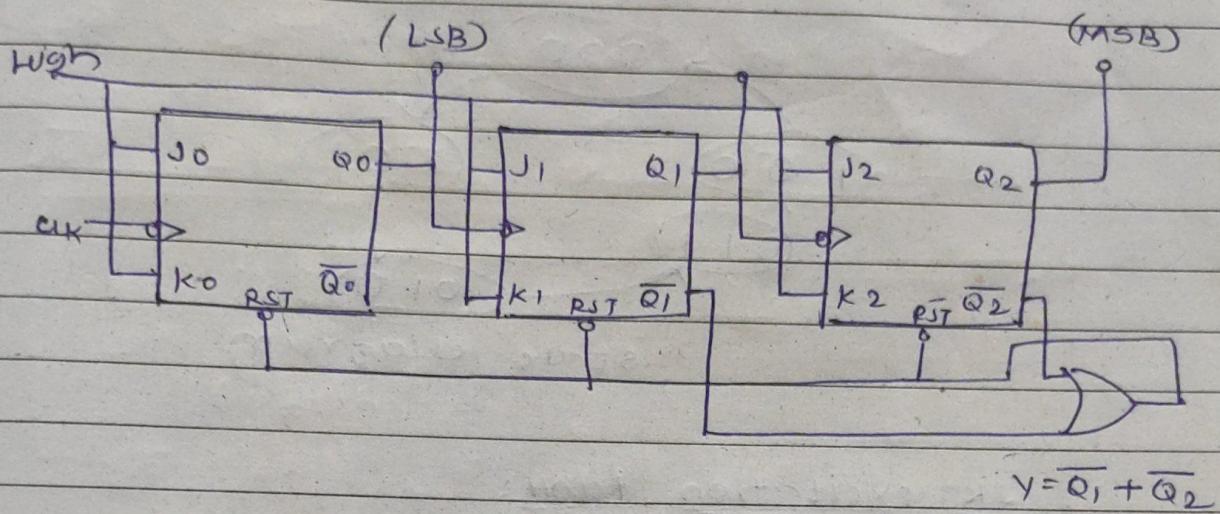
C4K	Q2	Q1	Q0	O1P
0	0	0	0	1
1	0	0	1	1
2	0	1	0	1
3	0	1	1	1
4	1	0	0	1
5	1	0	1	1
6	1	1	0	0

(iii) K-map

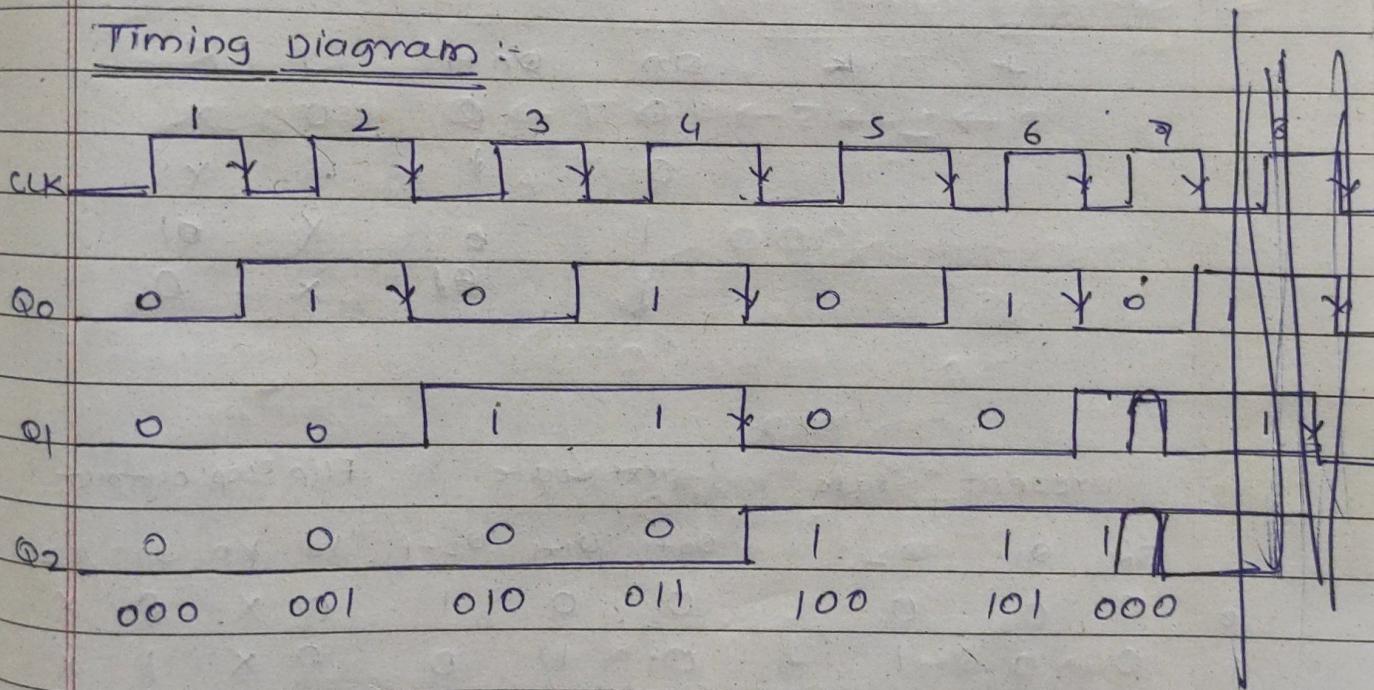


$$Y = \overline{Q_1} + \overline{Q_2}$$

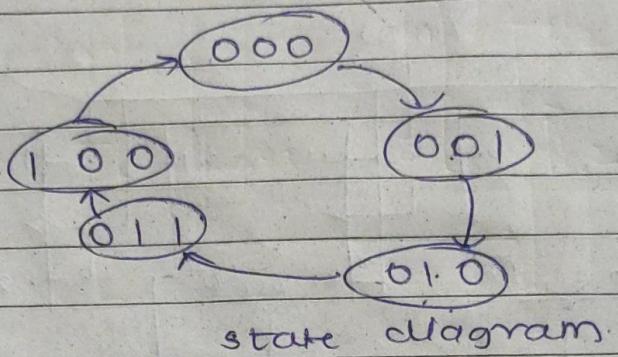
logic diagram:



Timing Diagram :-



\* mod-5 Synchronous using JK flip flop.



JK excitation table:

J	K	Qn	Qn+1	J	K
0	0	0	0	0	X
0	1	1	1	1	X
1	0	0	X	0	1
1	1	1	X	X	0

present state				Next state			Flip flop output				
Q <sub>0</sub>	Q <sub>1</sub>	Q <sub>2</sub>		Q <sub>0+1</sub>	Q <sub>1+1</sub>	Q <sub>2+1</sub>	J <sub>0</sub>	K <sub>0</sub>	J <sub>1</sub>	K <sub>1</sub>	
0	0	0	0	0	0	1	0	X	0	X	
1	0	0	1	0	1	0	0	X	1	X	
2	0	1	0	0	1	1	0	X	X	0	
3	0	1	1	1	0	0	1	X	X	1	
4	1	0	0	0	0	0	X	1	0	X	
5	1	0	1	X	X	X	X	X	X	X	
6	1	1	0	X	X	X	X	X	X	X	
7	1	1	1	X	X	X	X	X	X	X	

$J_0$   
 $\overline{Q_0 Q_2}$

	00	01	11	10
0	0	0	1	0
1	x	x	x	x

$$J_0 = \overline{Q_0 Q_1 Q_2} Q_1 Q_2$$

$K_0$

$\overline{Q_1 Q_2}$

	00	01	11	10
0	x	x	x	x
1	(1)	x	x	x

$$K_0 = 1$$

$J_1$

$\overline{Q_1 Q_2}$

	00	01	11	10
0	0	1	x	x
1	0	x	x	x

$$J_1 = Q_2$$

$\overline{Q_0 Q_2}$

	00	01	11	10
0	x	x	1	0
1	x	x	x	x

$$K_1 = Q_2$$

$J_2$

$\overline{Q_1 Q_2}$

	00	01	11	10
0	(1)	x	x	1
1	0	x	x	x

$$J_2 = \overline{Q_0}$$

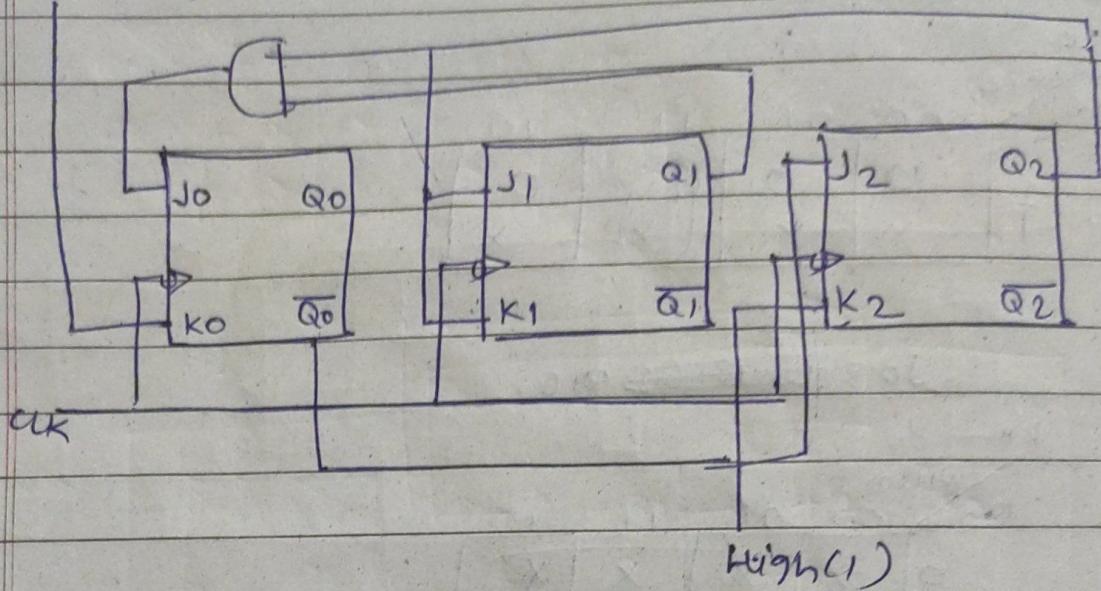
$K_2$

$\overline{Q_1 Q_2}$

	00	01	11	10
0	x	1	1	3x
1	x	x	x	x

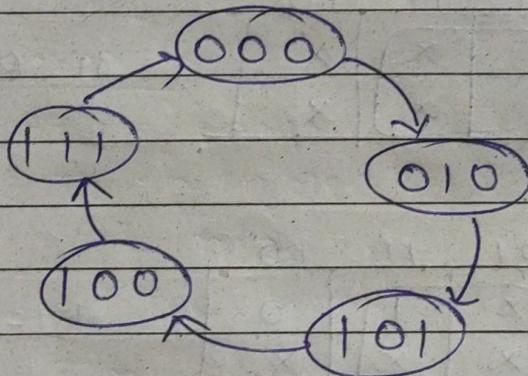
$$K_2 = 1$$

High(1)



\* sequence generator:

$0 \rightarrow 2 \rightarrow 5 \rightarrow 4 \rightarrow 7 \rightarrow 0$  using JK



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

Present state      Next state      Output

	$Q_0$	$Q_1$	$Q_2$	$Q_0+1$	$Q_1+1$	$Q_2+1$	$J_0$	$K_0$	$J_1$	$K_1$	$J_2$	$K_2$
0	0	0	0	0	1	0	0	X	1	X	0	X
1	0	0	1	X	X	X	X	X	X	X	X	X
2	0	1	0	1	0	1	1	X	X	1	1	X
3	0	1	1	X	X	X	X	X	X	X	X	X
4	1	0	0	1	1	1	X	0	1	X	1	X
5	1	0	1	1	0	0	X	0	0	X	X	1
6	1	1	0	X	X	X	X	X	X	X	X	X
7	1	1	1	0	0	0	X	1	X	1	X	1

$J_0$

$Q_0 \backslash Q_1 Q_2$

	00	01	11	10
0	0	X	X	1
1	X	X	X	X

$J_0 = \overline{Q_1} Q_1$

$K_0$

$Q_0 \backslash Q_1 Q_2$

	00	01	11	10
0	X	X	X	X
1	0	0	1	X

$K_0 = Q_1$

$J_1$

$Q_0 \backslash Q_1 Q_2$

	00	01	11	10
0	1	X	X	X
1	1	0	X	X

$J_1 = \overline{Q_2}$

$K_1$

$Q_0 \backslash Q_1 Q_2$

	00	01	11	10
0	X	X	X	1
1	X	X	1	X

$K_1 = 1$

$J_2$

$Q_0 \backslash Q_1 Q_2$

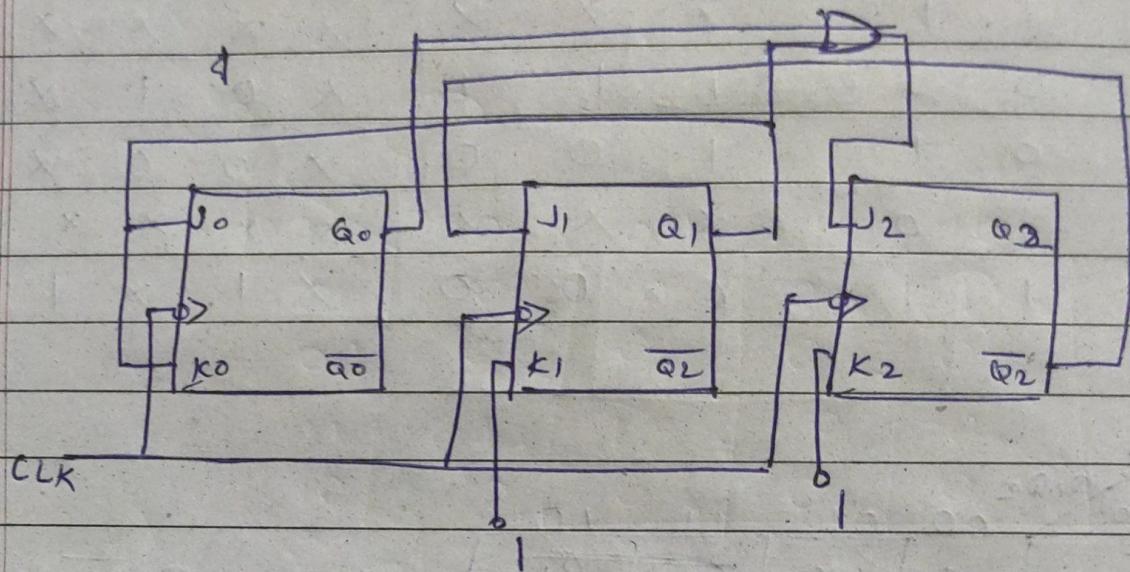
	00	01	11	10
0	0	X	X	1
1	1	X	X	X

$J_2 = \overline{Q_0} + Q_1$

$K_2$

	$Q_1 Q_2$	00	01	11	10
$Q_0$	0	(x) $\times$	$\times$	$\times$	$\times$
1	1	(x) 1	$\times$	$\times$	1

$$K_2 = 1$$



Asynchronous

state table

Truth table

K-map

logic diagram

Time diagram

Synchronous

state diagram

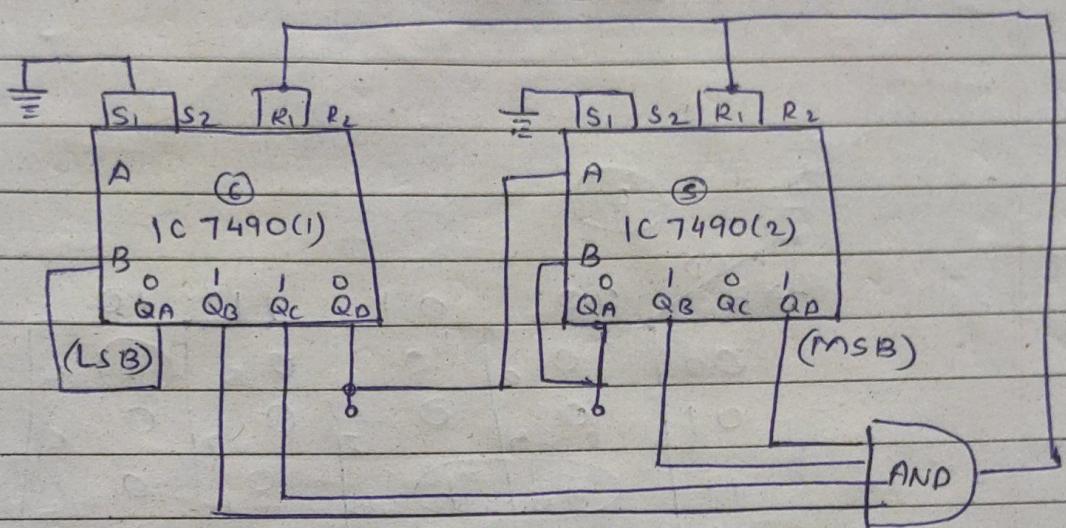
excitation

Present, next state

K-map

diagram.

## \* MOD 56 counter :-



## \* MOD 92

