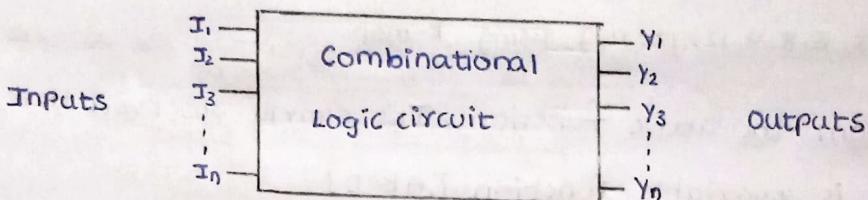


2. COMBINATIONAL AND SEQUENTIAL LOGIC CIRCUIT

For digital system consist of combinational circuit and Sequential circuits.

COMBINATIONAL CIRCUIT

THE Combinational circuit output depends on only present inputs only.



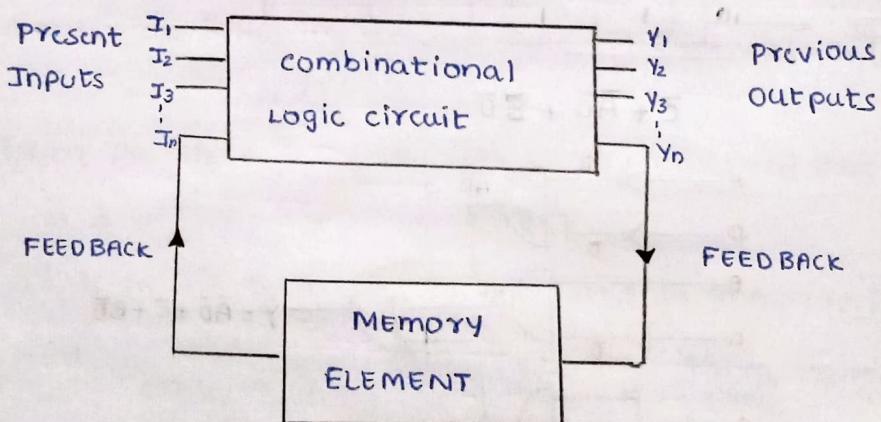
Here $I_1, I_2, I_3 \dots I_n$ are inputs. $Y_1, Y_2, Y_3 \dots Y_n$ are outputs.

Combinational logic circuits may be adder or subtractor, Encoded, decoded, multiplexer, de-multiplexer,

In a combinational logic circuit n input. 2^n outputs.

SEQUENTIAL CIRCUIT

It output depends on present input and previous output.



Here $I_1, I_2, I_3 \dots I_n$ are inputs. $Y_1, Y_2, Y_3 \dots Y_n$ are outputs. Here memory element might be flipflop

Sequential logic circuits may be Register, counter, flipflop.

Differences between Combinational and Sequential logic circuit

COMBINATIONAL LOGIC CIRCUIT	SEQUENTIAL LOGIC CIRCUIT
In combinational circuit the output variable at any instant of time or dependent only on the present input variables.	In sequential circuit the output variable at any instant of time or dependent on the previous output, not only on the present input but also present stage or previous output.
Memory element is not required.	Memory element is required to store the previous history of the input variable.
Combinational circuit are easy to design.	It is are comparatively hard to design.
Memory element or feedback path is not present	Memory element or feedback path is present
It is a simple circuit	It is a complex circuit
It is are faster because the delay between there input and output is low.	It is slower because the output is due to propagation delay along with feedback path delay.

Classification of Sequential Circuits.

1. Synchronous
2. Asynchronous.

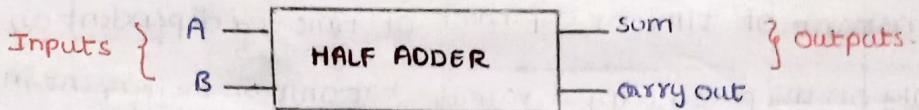
ADDERS

Digital computer performs various arithmetic operations. In most basic operation is adding of two binary digits.

HALF ADDER

A combinational circuit that performs the addition of two bits is called a half adder.

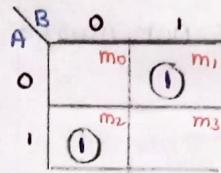
It adds two inputs [A,B] which as single bit. It doesn't take carry from previous sum.



TRUTH TABLE

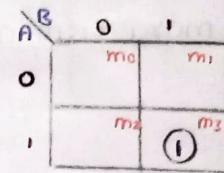
Inputs		Outputs	
A	B	Sum	Cout
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

K-map



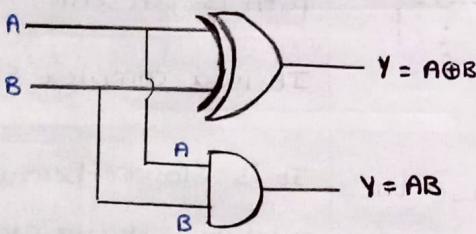
$$\text{Sum} = A\bar{B} + \bar{A}B$$

$$\text{Sum} = A \oplus B$$



$$\text{Carry out} = AB$$

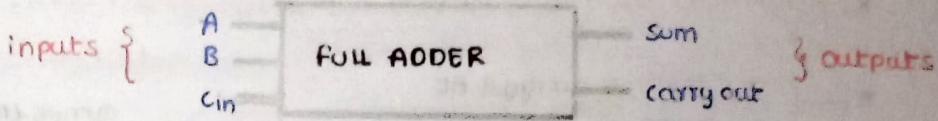
Logic diagram.



FULL ADDER

It is a combinational circuit that performs the addition of three bit is called a full adder.

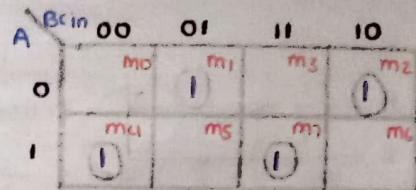
It is designed to add more than two bit. It can add two single bit number along with a carry in [A, B, cin]



TRUTH TABLE

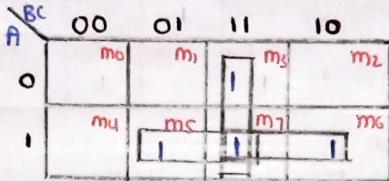
Inputs			Outputs	
A	B	Cin	Sum	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

K-map for sum



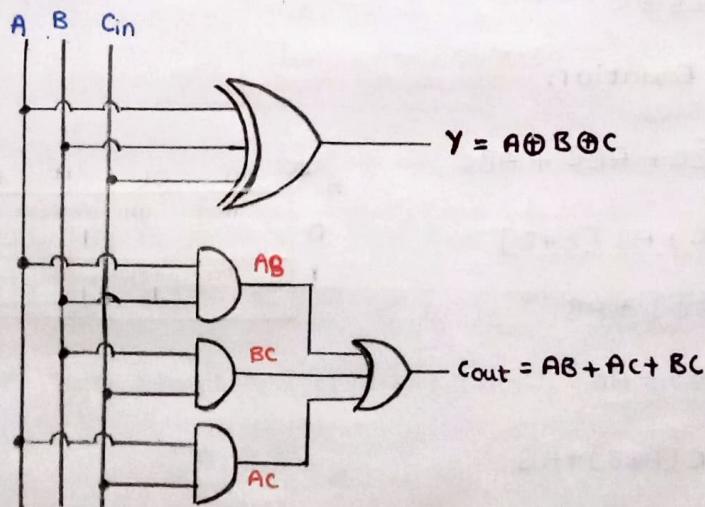
$$\begin{aligned}
 \text{Sum} &= \bar{A}\bar{B}C + A\bar{B}\bar{C} + ABC + \bar{A}BC \\
 &= \bar{B}[\bar{A}C + A\bar{C}] + B[AC + \bar{A}\bar{C}] \\
 &= \bar{B}[A \oplus C] + B[\bar{A} \oplus \bar{C}] \\
 &= \bar{B}[x] + B[\bar{x}] \\
 &= B \oplus x \Rightarrow A \oplus B \oplus C
 \end{aligned}$$

K-map for carry out

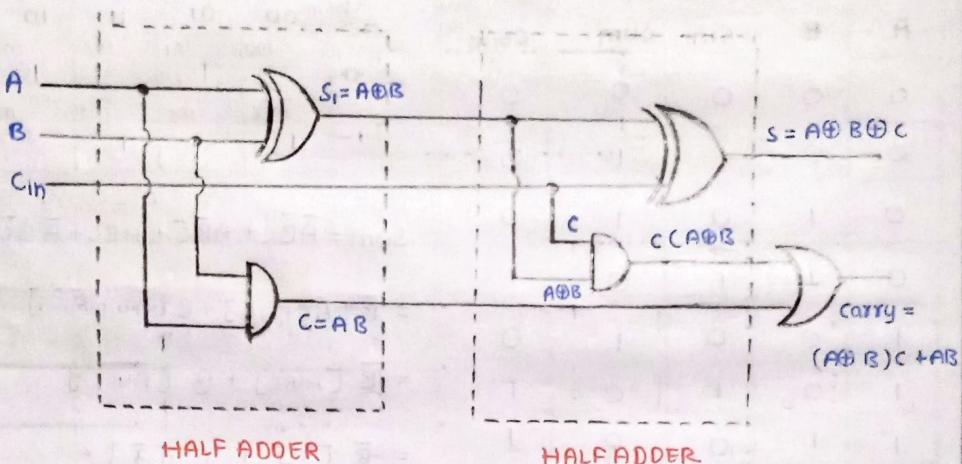
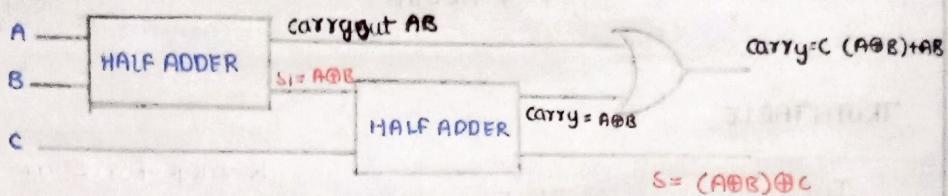


$$\text{Carry} = AC + BC + AB$$

Logical diagram



REALIZATION OF FULL-ADDER USING TWO HALF ADDERS



Sum Equation

$$\bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$$

$$C[\bar{A}\bar{B} + AB] + \bar{C}[\bar{A}B + A\bar{B}]$$

$$C\bar{X} + \bar{C}X$$

$$\therefore X = A \oplus B = \bar{A}B + \bar{B}A$$

$$\text{Sum} = X \oplus C$$

$$\bar{X}_1 = \overline{A \oplus B} = \bar{A}\bar{B} + A\bar{B}$$

$$\text{Sum} = A \oplus B \oplus C$$

		BC	00	01	11	10
		A	m ₀	m ₁	m ₃	m ₂
0	0		1		1	1
	1		m ₄	m ₅	m ₇	m ₆

Carry Equation

$$\bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$$

$$\bar{A}BC + A\bar{B}C + AB[C + \bar{C}]$$

$$C[\bar{A}B + A\bar{B}] + AB$$

$$C[A \oplus B] + AB$$

$$\text{Carry} = C[A \oplus B] + AB$$

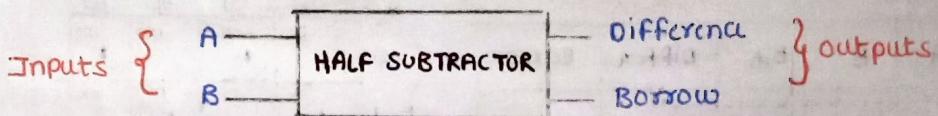
		BC	00	01	11	10
		A	m ₀	m ₁	m ₃	m ₂
0	0		1		1	1
	1		m ₄	m ₅	m ₇	m ₆

HALF SUBTRACTOR

A combinational circuit that performs the subtraction of two bits is called a half subtractor.

It subtracts two inputs [A, B] which as single bit.

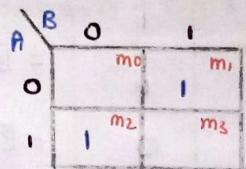
It doesn't take carry from previous subtractor.



TRUTH TABLE

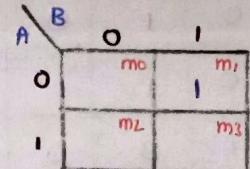
Inputs		Outputs	
A	B	Difference	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

K-map



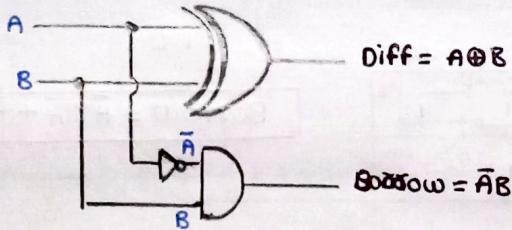
$$\text{Sum} = A\bar{B} + \bar{A}B$$

$$\text{Diff} = A \oplus B$$



$$\text{Borrow} = \bar{A}B$$

LOGIC DIAGRAM

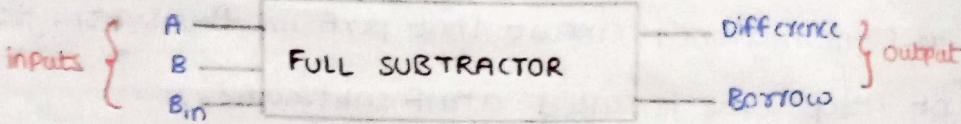


FULL SUBTRACTOR

It is a combinational circuit that performs the subtraction of three bits is called a full subtractor.

It is designed to subtract more than two bits. It can add and subtract two single bit numbers along with a Borrow in

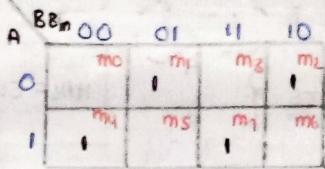
[A, B, Bin]



TRUTH TABLE

Inputs			Outputs	
A	B	B _{in}	Difference	Borrow
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

K-map for Difference



SIMPLIFIED EXPRESSION

$$\text{Difference} = \bar{A}\bar{B}B_{in} + \bar{A}B\bar{B}_{in} + AB\bar{B}_{in} + A\bar{B}\bar{B}_{in}$$

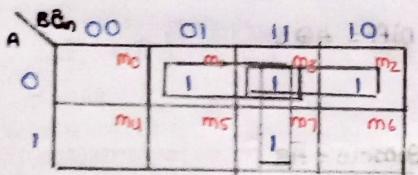
$$= \bar{B}[\bar{A}B_{in} + AB_{in}] + B[A\bar{B}_{in} + \bar{A}\bar{B}_{in}]$$

$$= \bar{B}[A \oplus B_{in}] + B[\bar{A} \oplus B_{in}]$$

$$= \bar{B}[x] + B[\bar{x}]$$

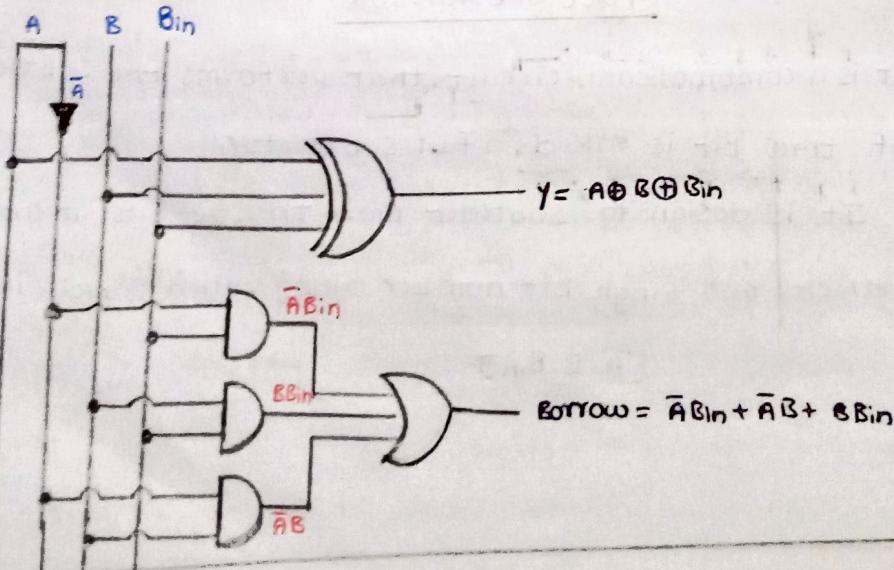
$$= \bar{B} \oplus x \Rightarrow \boxed{\text{Difference} = A \oplus B \oplus B_{in}}$$

K-map for Borrow



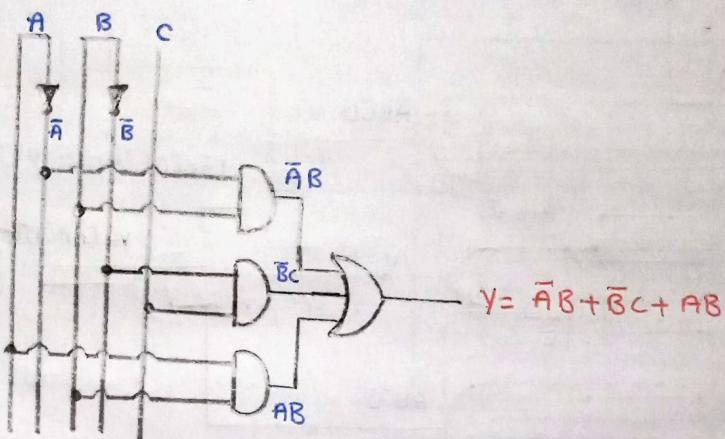
$$\text{Borrow} = \bar{A}B_{in} + \bar{A}B + BB_{in}$$

Logic DIAGRAM

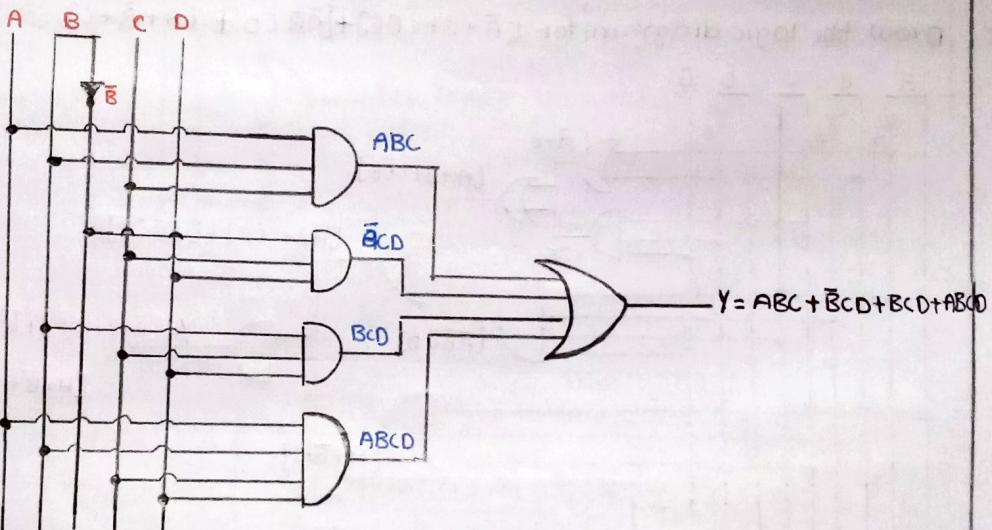


HOME WORK

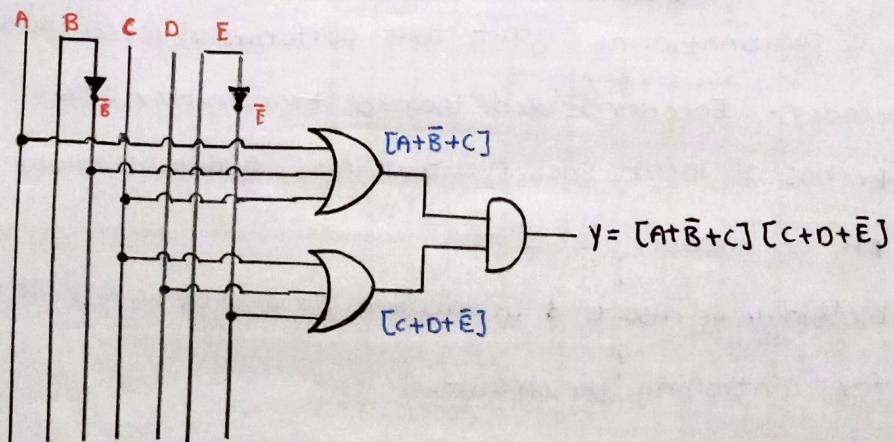
- 1 Draw Logic diagram of $\bar{A}B + \bar{B}C + AB$



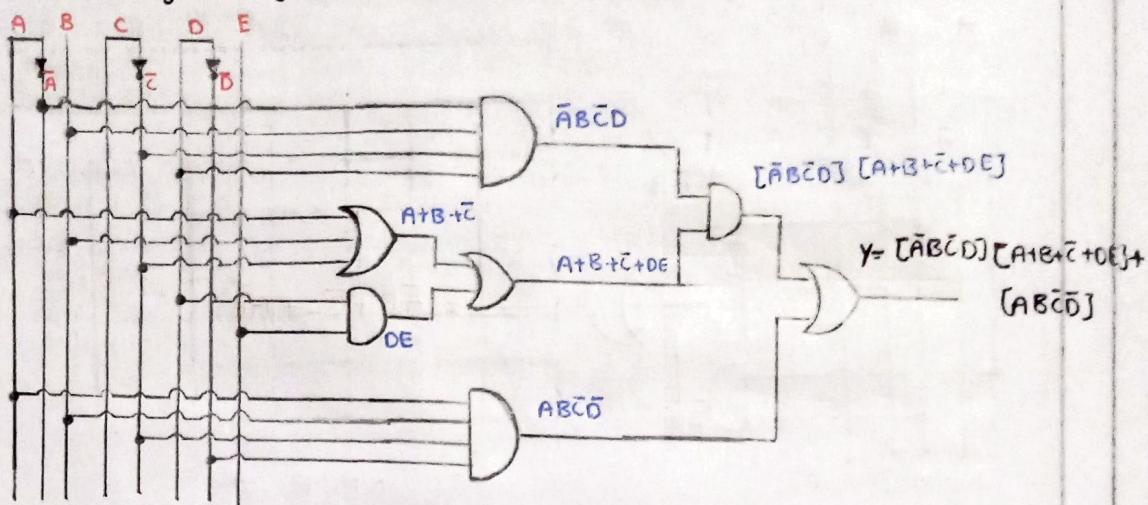
- 2 Draw Logic diagram of $ABC + \bar{B}CD + BCD + ABCD$



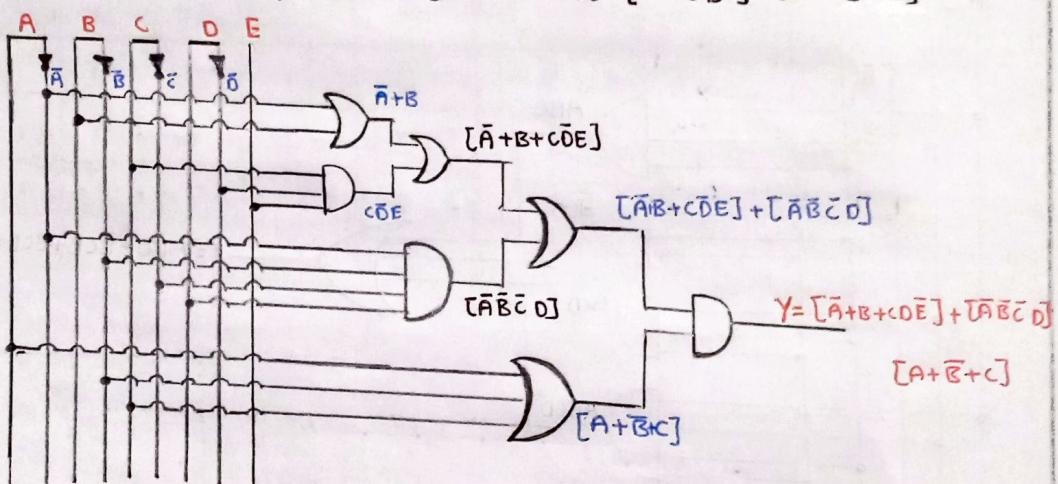
- 3 Draw logic diagram for $[A + \bar{B} + C]$ $[C + D + \bar{E}]$



4 Draw a logic diagram for $[\bar{A}\bar{B}\bar{C}D] \cdot [A+B+\bar{C}+DE] + [\bar{A}B\bar{C}\bar{D}]$



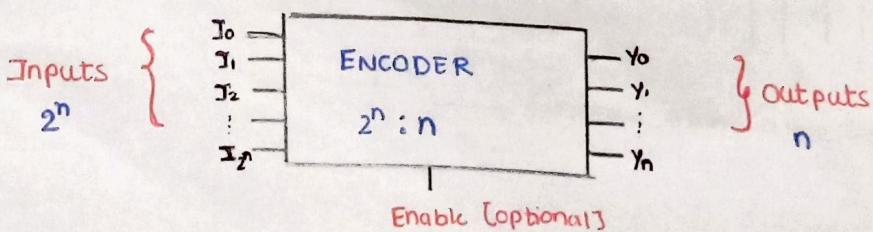
5 Draw the logic diagram for $[\bar{A}+B+C\bar{D}\bar{E}] + [\bar{A}\bar{B}\bar{C}\bar{D}] \cdot [A+\bar{B}+C]$



ENCODER

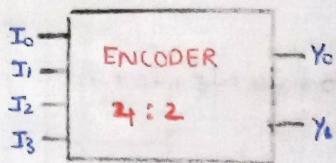
It is a combinational circuit that performs reverse operation of Decoder. Encoder is used toward transmitter side. Encoder has 2^n input lines n output lines. General formula of an encoder is $2^n \times n$.

Enable Single is also used in encoder. It is an optional. It is used for controlling the circuit.



DESIGNING OF 4:2 ENCODER

TRUTH TABLE



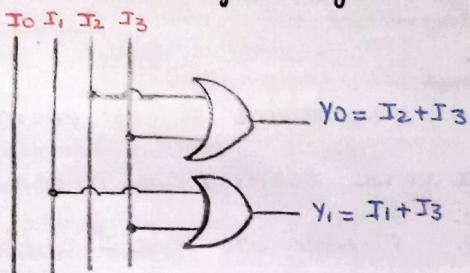
Output Equations:

$$Y_0 = I_2 + I_3$$

$$Y_1 = I_1 + I_3$$

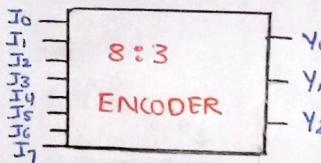
INPUTS						OUTPUTS	
E	J ₃	J ₂	J ₁	J ₀		Y ₀	Y ₁
0	x	x	x	x		0	0
1	0	0	0	1		0	0
1	0	0	1	0		0	1
1	0	1	0	0		1	0
1	1	0	0	0		1	1

Logic diagram.



DESIGNING OF 8:3 ENCODER

TRUTH TABLE



Output Equations:

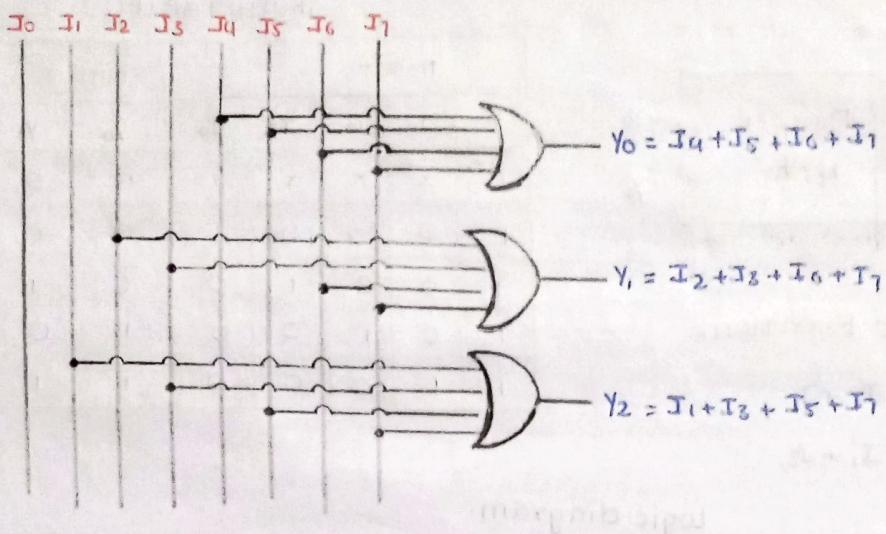
$$Y_0 = I_4 + I_5 + I_6 + I_7$$

$$Y_1 = I_2 + I_3 + I_6 + I_7$$

$$Y_2 = I_1 + I_3 + I_5 + I_7$$

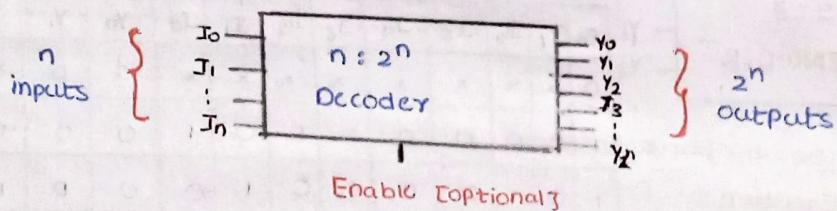
Inputs									Outputs		
E ₀	I ₇	I ₆	I ₅	I ₄	I ₃	I ₂	I ₁	I ₀	Y ₀	Y ₁	Y ₂
0	x	x	x	x	x	x	x	x	0	0	0
1	0	0	0	0	0	0	0	1	0	0	0
1	0	0	0	0	0	0	1	0	0	0	1
1	0	0	0	0	0	1	0	0	0	1	0
1	0	0	0	0	1	0	0	0	0	1	1
1	0	0	0	1	0	0	0	0	1	0	0
1	0	0	1	0	0	0	0	0	1	0	1
1	1	0	0	0	0	0	0	0	1	1	0
1	1	0	0	0	0	0	0	0	1	1	1

Logic diagram

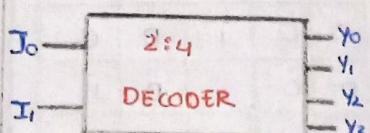


DECODER

It is combinational circuit that performs reverse operation of Encoder. Decoder are used towards receiver side in communication to decode the encoded data. Decoder are have 2^n output lines and n inputs. General formula of Decoders is $n \times 2^n$. Enable signal is used in Decoder. It is an optional. It is used to control the circuit.



DESIGNING OF 2:4 DECODER



output equation

$$Y_0 = \bar{I}_0 \bar{I}_1$$

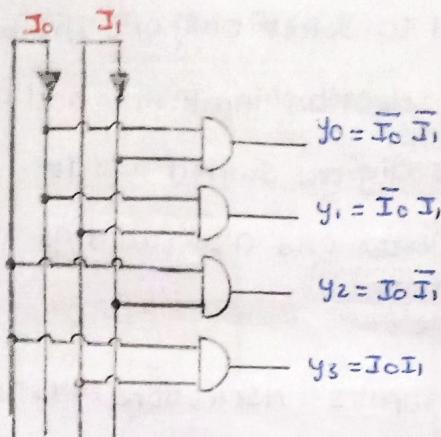
$$Y_1 = \bar{I}_0 I_1$$

$$Y_2 = I_0 \bar{I}_1$$

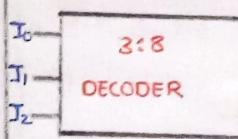
$$Y_3 = I_0 I_1$$

Inputs			Outputs			
En	I_1	I_0	Y_0	Y_1	Y_2	Y_3
0	x	x	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1

Logic diagram.



DESIGNING OF 3:8 DECODER



output equation:

$$y_0 = \bar{I}_2 \bar{I}_1 \bar{I}_0$$

$$y_1 = \bar{I}_2 \bar{I}_1 I_0$$

$$y_2 = \bar{I}_2 I_1 \bar{I}_0$$

$$y_3 = \bar{I}_2 I_1 I_0$$

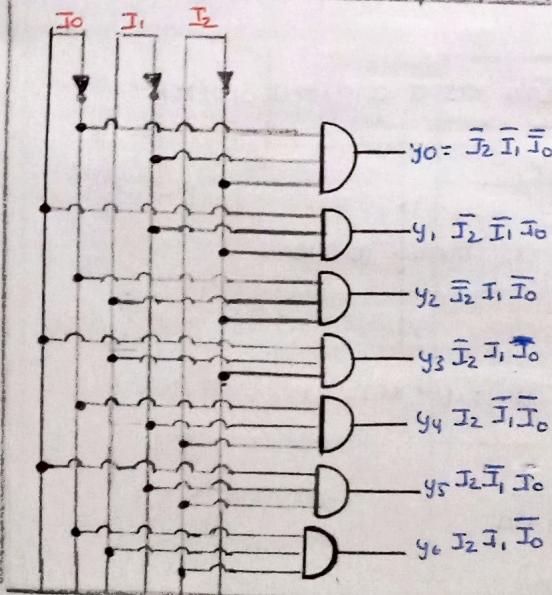
$$y_4 = I_2 \bar{I}_1 \bar{I}_0$$

$$y_5 = I_2 \bar{I}_1 I_0$$

$$y_6 = I_2 I_1 \bar{I}_0$$

$$y_7 = I_2 I_1 I_0$$

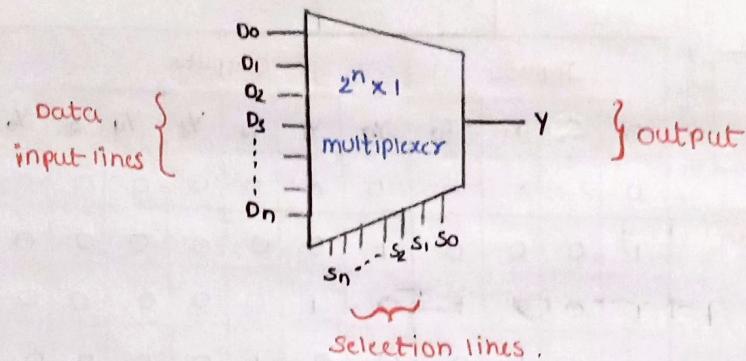
Inputs			Outputs								
E _n	I ₂	I ₁	I ₀	y ₀	y ₁	y ₂	y ₃	y ₄	y ₅	y ₆	y ₇
0	x	x	x	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	0	0	0	0
1	0	0	1	0	1	0	0	0	0	0	0
1	0	1	0	0	0	1	0	0	0	0	0
1	0	1	1	0	0	0	1	0	0	0	0
1	1	0	0	0	0	0	0	1	0	0	0
1	1	0	1	0	0	0	0	0	1	0	0
1	1	1	0	0	0	0	0	0	0	1	0
1	1	1	1	0	0	0	0	0	0	0	1



MULTIPLEXER

It is a combinational circuit used to select only one input among several inputs based on selection input lines and it is MSI $2^n \times 1$. This can act as digital switch and it is denoted by $2^n \times 1$, data selector and also called as Selector variable or Selector line.

for a mux there can be 2^n inputs, n Selection lines and only one output is possible.



Advantages:

1. It reduces number of wires
2. It reduce circuit complexity.
3. It reduce cost and reduces number of lines.

Application

1. It is used in communication as a digital switch
2. It is used as data selector.

NOTE: multiplexer is known as many to one

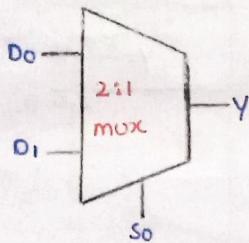
Types

There are many types of multiplexer

1. 2x1
2. 4x1
3. 8x1
4. 16x1 ... so on.

DESIGNING OF 2:1 MULTIPLEXER

2x1 mux will accept 2 inputs and gives 1 output and it is one selection line input and it has also one enable and it is optional.



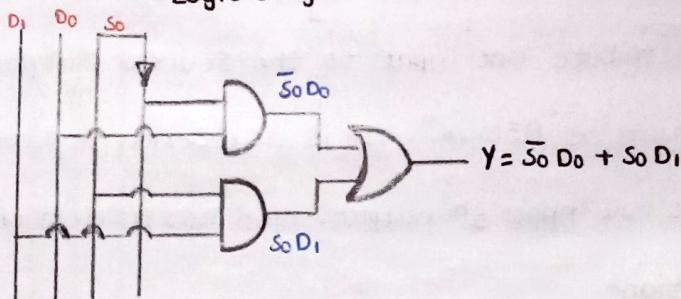
Inputs		Output
E	S0	Y
0	X	0
1	0	D0
1	1	D1

output equation:

$$Y = \bar{S}_0 D_0$$

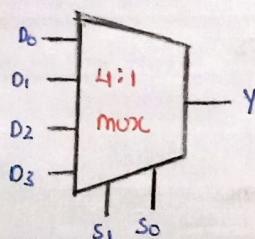
$$Y = S_0 D_1 \quad Y = \bar{S}_0 D_0 + S_0 D_1$$

Logic diagram.



DESIGNING OF 4:1 MULTIPLEXER

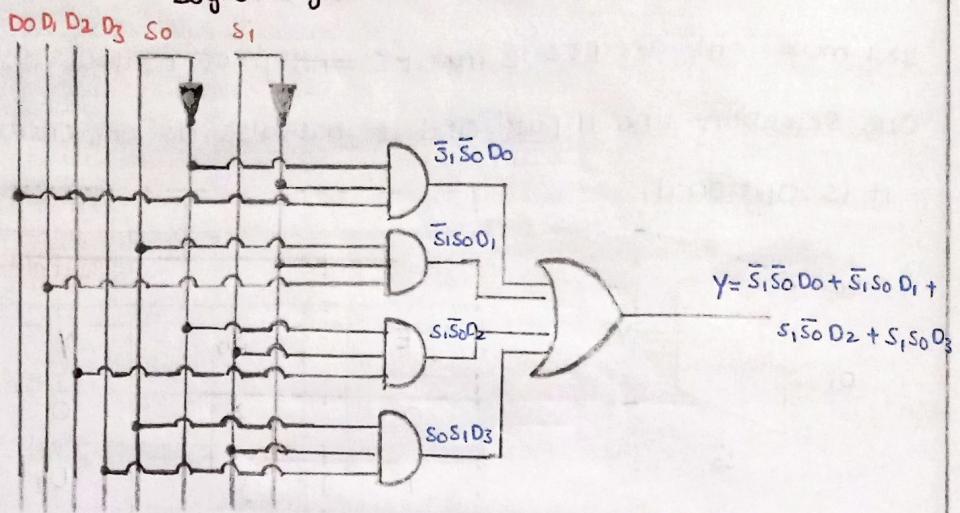
4x1 mux will accept 4 inputs and gives 1 output and it has two selection line input and it also has one enable and it is optional.



$$Y = \bar{S}_0 \bar{S}_1 D_0 + S_0 \bar{S}_1 D_1 + \bar{S}_0 S_1 D_2 + S_0 S_1 D_3$$

Inputs			Outputs
E	S1	S0	Y
0	X	X	0
1	0	0	D0
1	0	1	D1
1	1	0	D2
1	1	1	D3

Logic diagram.



DE MUX

It is a combinational circuit which is used in communication.

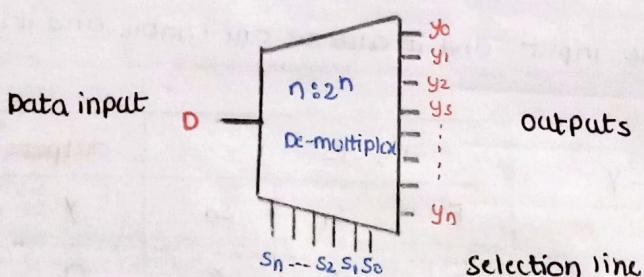
It can distribute one input to the several output. Demultiplex is also called as "DEMUX". It is as a serial to parallel converter.

It accepts one input 2^n outputs and has n selection lines.

Applications

It is used as distribution in communication.

It is as a serial to parallel converter.



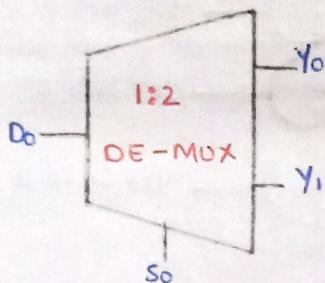
TYPES OF DE-MUX

There are many types of de multiplexers

- 1. 1x2 3. 1x8
- 2. 1x4 4. 1x16

DESIGNING OF 1:2 DE-MULTIPLEXER

It is a combinational circuit it will accept one input data line and distribute it to the output lines [y_0, y_1] based on selection input lines]

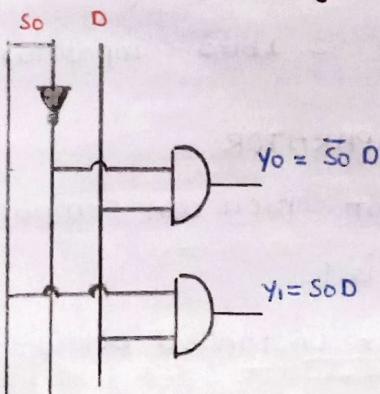


Input		outputs	
E	S_0	y_0	y_1
0	x	0	0
1	0	0	1
1	1	1	0

Output equation:

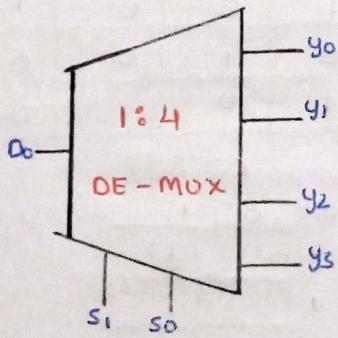
$$y_0 = \overline{S_0} D \quad y_1 = S_0 D$$

Logical diagram



DESIGNING OF 1:4 DE-MULTIPLEXER

It is a combinational circuit it will accept one input data line and distribute it to the output lines [y_0, y_1, y_2, y_3] based on selection input lines].



Input			outputs			
E	S_1	S_0	y_0	y_1	y_2	y_3
0	x	x	0	0	0	0
1	0	0	0	1	0	0
1	0	1	1	0	1	0
1	1	0	0	0	0	1
1	1	1	0	0	0	0

output equation:

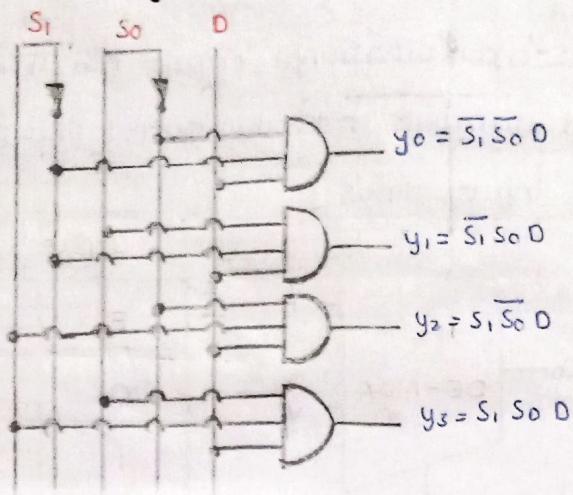
$$y_0 = \bar{s}_1 \bar{s}_0 D$$

$$y_1 = \bar{s}_1 s_0 D$$

$$y_2 = s_1 \bar{s}_0 D$$

$$y_3 = s_1 s_0 D$$

LOGIC DIAGRAM



COMPARATOR

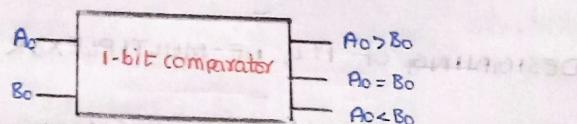
Comparator is a combinational circuit which is used for comparing numbers $[A, B]$. There are many types

- 1. 1bit comparator
- 2. 2bits comparator ... Soon

DESIGNING OF 1 BIT COMPARATOR

It is a combinational circuit which can compare two numbers $[A, B]$ of single bit $[A_0, B_0]$.

Comparison can be better understand below logic



1. $A_0 = 1, B_0 = 0$ $A_0 > B_0$ [Greater]

2. $A_0 = 1, B_0 = 1$ $A_0 = B_0$ [Equal]

3. $A_0 = 0, B_0 = 1$ $A_0 < B_0$ [Lesser]

4. $A_0 = 0, B_0 = 0$ $A_0 = B_0$ [Equal]

Inputs		Outputs		
A_0	B_0	$A > B_0$	$A = B$	$A < B_0$
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0

K-map for $A > B$

A_0	B_0	0	1
0	m ₀	m ₁	
1	m ₂	m ₃	

$A_0 \bar{B}_0$

K-map for $A = B$

A_0	B_0	0	1
0	m ₀	m ₁	1
1	1	m ₂	m ₃

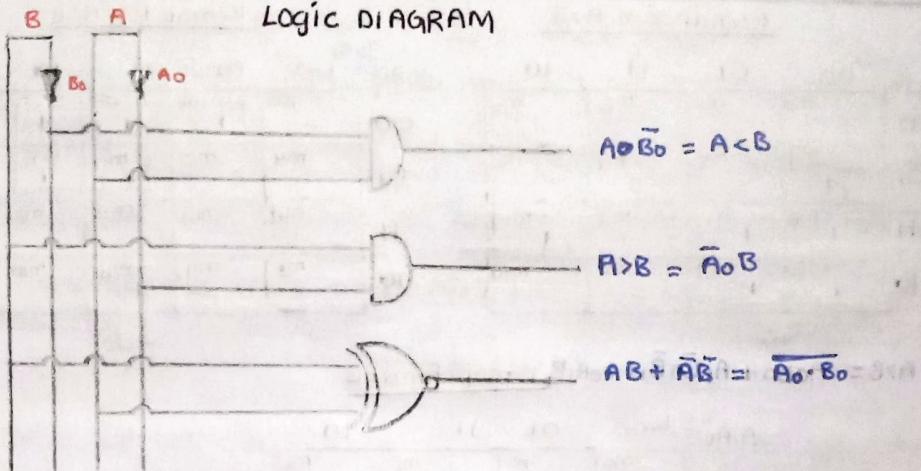
$A_0 + \bar{A}_0 \bar{B}_0$

K-map for $A < B$

A_0	B_0	0	1
0	m ₀	1	m ₃
1	m ₂		m ₃

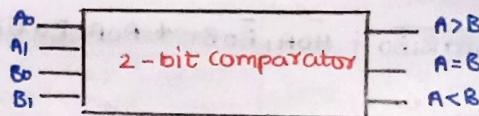
$\bar{A}_0 B_0$

LOGIC DIAGRAM



DESIGNING OF 2 BIT COMPARATOR

- It is a combinational circuit which can compare 2 bits (A, B) of 2 bits. $[A = A_0, A_1]$ $[B = B_0, B_1]$
- 2 bit comparator logic can be better understood by below truth table.



inputs				outputs		
A_1	A_0	B_1	B_0	$A > B$	$A = B$	$A < B$
0	0	0	0	0	1	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	1	0	0
0	1	0	1	0	1	0
0	1	1	0	0	0	1
0	1	1	1	0	0	1
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	0	1	0
1	0	1	1	0	0	1
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	1	0	0

K-map for $A > B$

		00	01	11	10
		m_0	m_1	m_3	m_2
		m_4	m_5	m_7	m_6
		m_8	m_9	m_{15}	m_{14}
		m_{10}	m_{11}	m_{13}	m_{12}

K-map for $A < B$

		00	01	11	10
		m_0	m_1	m_3	m_2
		m_4	m_5	m_7	m_6
		m_{12}	m_{13}	m_{15}	m_{14}
		m_8	m_9	m_{11}	m_{10}

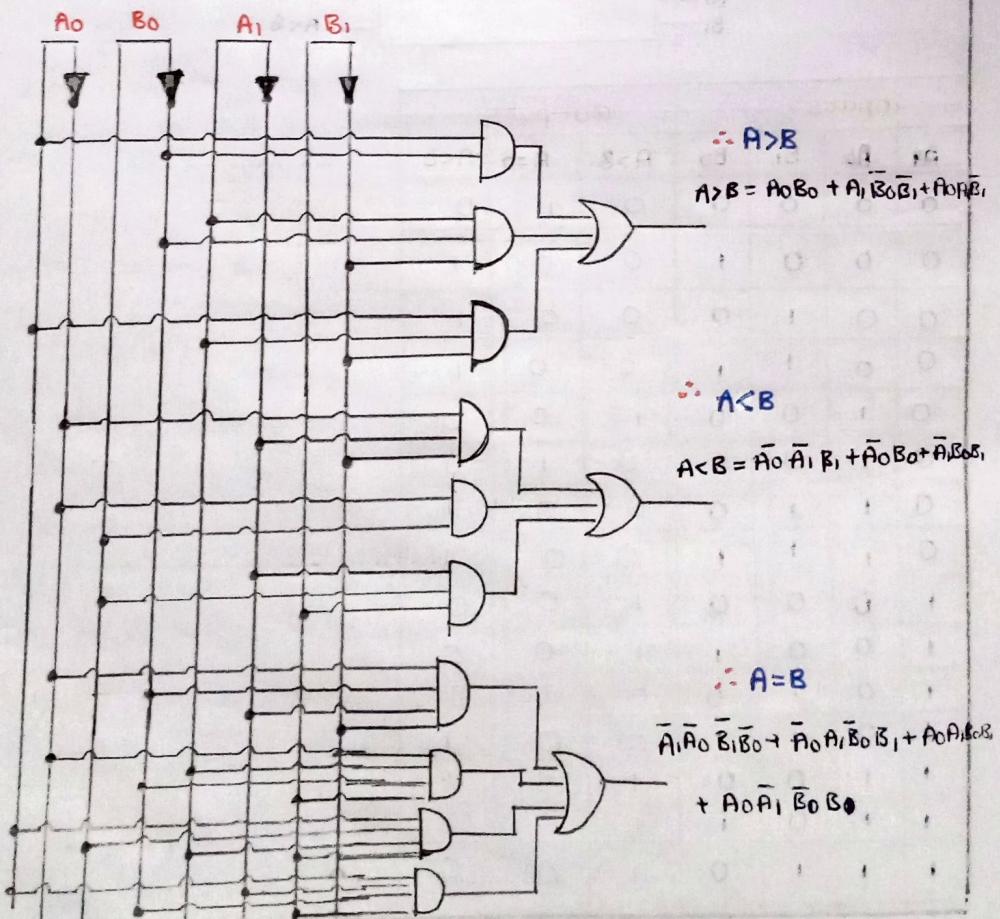
K-map for $A = B$

		00	01	11	10
		m_0	m_1	m_3	m_2
		m_4	m_5	m_7	m_6
		m_2	m_{13}	m_{15}	m_{14}
		m_8	m_9	m_{11}	m_{10}

$$\therefore A > B = \bar{A}_0 \bar{B}_0 + A_1 \bar{B}_0 \bar{B}_1 + \bar{A}_0 A_1 \bar{B}_1$$

$$\therefore A < B = \bar{A}_0 \bar{A}_1 \bar{B}_1 + \bar{A}_0 B_0 + \bar{A}_1 B_0 \bar{B}_1$$

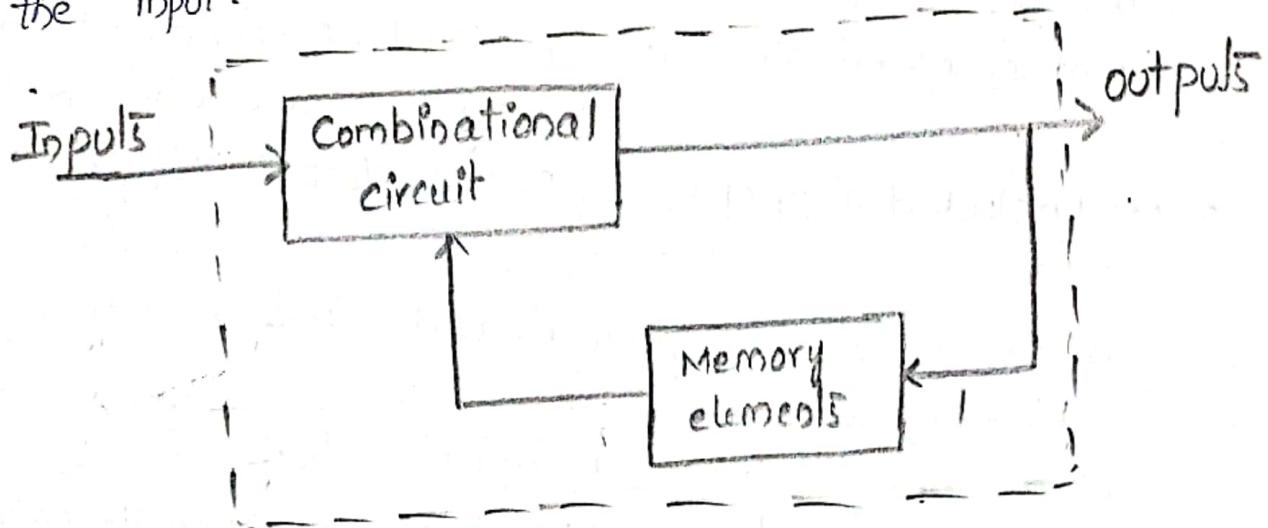
$$A = B = \bar{A}_1 \bar{B}_1 \bar{B}_0 + \bar{A}_0 A_1 \bar{B}_0 \bar{B}_1 + \bar{A}_0 A_1 B_0$$



Sequential logic circuits:

The design of logic circuit using memory and elements and combinational circuit is known as sequential logic circuit.

- Sequential logic circuit generates output in accordance with the sequence in which the input signals are received.
- The outputs always depends on the present inputs and on past outputs, because output is connected back to the input.



Block diagram of sequential circuit

Thus we can specify the sequential circuit by a time sequence of external inputs, internal states (present states and next states) and outputs.

Sequential circuits can be classified into

1. Synchronous Sequential circuits and
2. Asynchronous Sequential circuits depending on the timing of their signals.

- In asynchronous sequential circuits, change in input signals can affect memory elements at any instant of time.
- In synchronous sequential circuits, signals can affect the memory elements only at discrete instants of time.

Asynchronous Sequential Circuits

1. Input signals can affect memory elements at any instant of time.
2. More difficult to design
3. Clock is not required as one of the input
4. Speed of operation is high
5. Ex: Unclocked flipflops

Synchronous Sequential Circuits

1. Input signals can effect memory elements at discrete instant of time.
2. Easier to design.
3. Clock is required as one of the input.
4. Speed of operation is limited by the time delays involved.
5. Ex: Clocked flipflops

Comparison between Sequential and Combinational Circuits

Combinational logic circuits

1. output depends only on the present input
2. Easier to design
3. Speed of operation is high
4. Memory unit is not required
5. Ex: parallel adder

Sequential logic circuits

1. output depends on the present input and past output also
2. comparatively harder to design
3. speed of operation is comparatively low.
4. Memory unit is required to store the past outputs.
5. Ex: Serial adder.

Latch

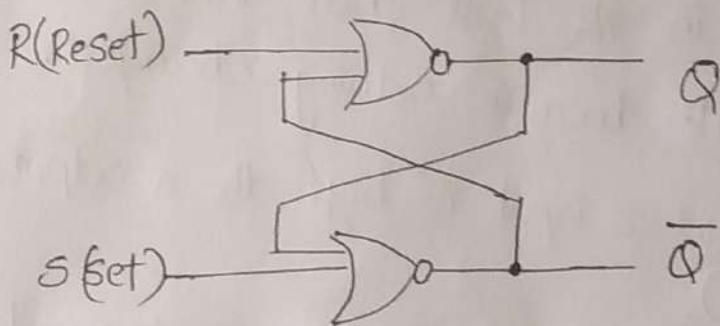
Latches and flipflops both are bistable elements. These are the basic building blocks of most sequential circuits.

- The main difference between latches and flipflop is in the method used for changing their state.
- Many times enable signal is provided with the latch. When enable signal is active output changes occur as input changes.
But when enable signal is not activated input changes don't affect output.

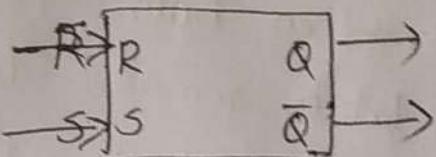
S-R latch:

The simplest type of latch is set-reset (SR) latch. It can be constructed from either two NAND gates or two NOR gates.

SR latch using NOR gates



logic symbol



logic diagram

The two NOR gates are cross coupled so that the output of NOR gate 1 is connected to one of the

Inputs of NOR gate 2 and vice versa.

The latch has two inputs Q and \bar{Q} , and two inputs Set and Reset.

Case 1 : $S=0$ and $R=0$

case 2 : $S=0$ and $R=1$

Case 3 : $S=1$ and $R=0$

Case 4 : $S=1$ and $R=1$

Truth Table

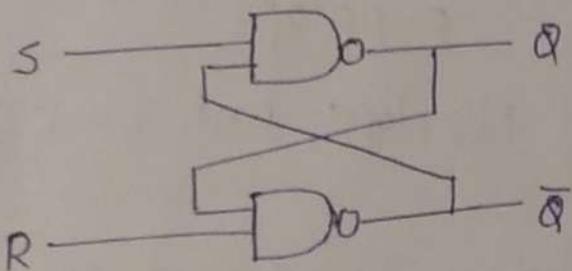
S	R	Q_n	Q_{n+1}	state
0	0	0	0	No change
0	0	1	1	
0	1	0	0	Reset
0	1	1	0	
1	0	0	1	Set
1	0	1	1	
1	1	0	x	Indeterminate
1	1	1	x	

operation :

- When both inputs low, the output does not change and latch remains latched in its last state. This condition is called inactive state because nothing changes.
- When R input is low and S input is high, the Q output of latch is set (at logic 1).
- When R input is high & S input is low, the o/p is Reset.
- When R and S inputs both are high, output is unpredictable. This is called indeterminate condition.

SR latch using NAND gates.

logic diagram



The operation of this latch is the reverse of the operation of NOR gate latch.

Truth table

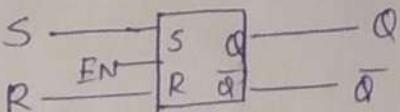
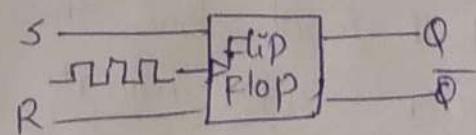
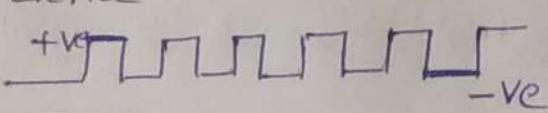
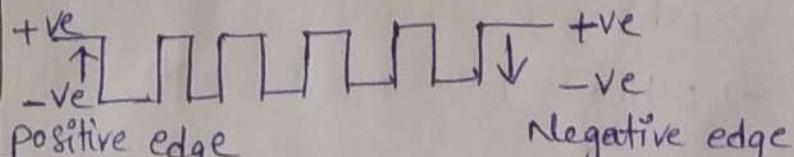
S	R	Q_n	Q_{n+1}	state
0	0	0	X	Indeterminate
0	0	1	X	Indeterminate
0	1	0	∅	Set
0	1	1	∅	Set
1	0	0	0	Reset
1	0	1	0	Reset
1	1	0	0	No change
1	1	1	1	No change

operation

- When both inputs low, the output is unpredictable . this is called Indeterminate condition.
- When R is high, S is low the Q output of latch is set
- When R is low, S is high the Q output of latch is Reset
- When both inputs high, the output does not change.

Differences between Latch and FlipFlop

Latches and FlipFlops both are building blocks of sequential circuits

Latches	FlipFlops
1. Latches do not require clock signal	FlipFlops have clock signals.
2. A power requirement of a latch is less	Power requirement of a flip-flop is more.
3. A latch works based on the enable signal	A flip flop works based on the clock signal.
4. The operation of a latch is faster as they do not have to wait for any clock signal	FlipFlops are comparatively slower than latches due to clock signal
5. A latch is an asynchronous device	A flip flop is a synchronous device.
6. Latch can be built by logic gates	Flip flop can be built by latch with an additional control input.
7. <u>Logic Symbol</u>	<u>Logic Symbol</u>
	
8. Latch is a level triggered device	Flip flop is a edge triggered device
	

Flip Flops:

The storage elements employed in clocked sequential circuits are called flip flops.

- A flip-flop is a binary cell capable of storing one bit of information.
- It has two outputs, one for normal value and one for the complement value of the bit stored in it.
- A flip flop maintains a binary state until directed by a clock pulse to switch states.
- A flip flop is a bistable electronic circuit that has two stable states i.e., the output is either 0 or 1.
- Flip flop is the smallest storage unit of a computer since what it stores is a single bit (0 or 1).
- There are many types of flip flops.
The major difference between them are in the number of inputs applied and a manner how the inputs change the outputs.

1. SR flip flop

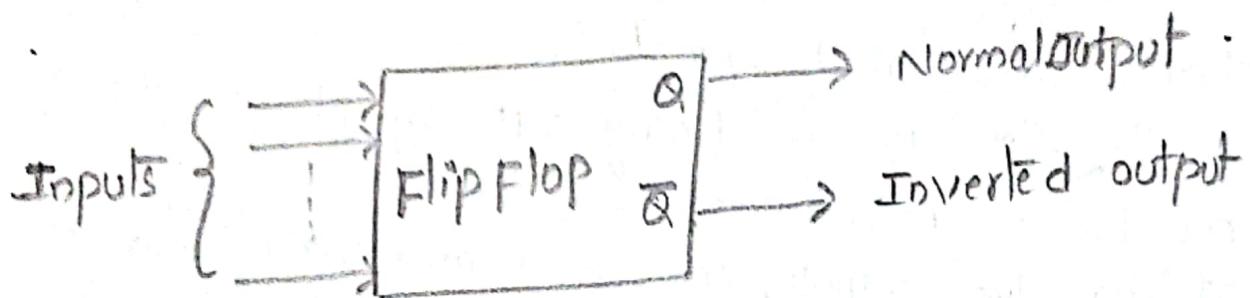
2. JK flip flop

3. D flip flop and

4. T flip flop

- SR flip flop is the commonest of them.

- A flipflop can have one or more inputs. The input signals which command the flipflop to change state are called excitations.



General flipflop symbol.

Applications of flip flops:

There are number of applications of flip flops.

1. The flipflop serves as a storage device
2. It stores a '1' when its Q output is '1', and stores a '0' when its Q output is a '0'.
3. flipflops are the fundamental components of shift registers and counters.

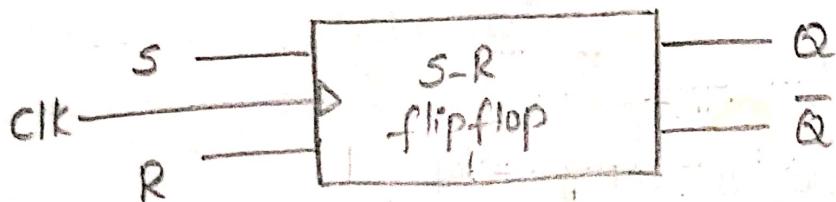
S-R FlipFlop :

In digital circuits, many operations have to be carried out in a proper sequence at the appropriate time. These operations are controlled by clock pulses.

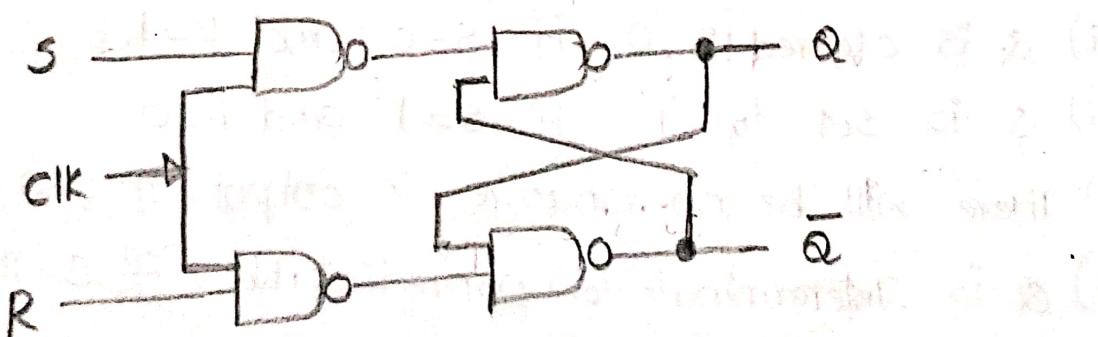
- The flip-flop output is controlled i.e., neither be set nor reset without the preference of the clock pulse.

Such a flip flop is the clocked SR flip flop.

- The SR flip flop can be represented using a graphic symbol.



Graphic symbol of S-R flipflop



Logic diagram of S-R flipflop

- It consists of inputs S, R and clk to set, Reset and for clock respectively.
- The clk input, that is represented with an arrowhead shape is dynamic, as it responds only when there is a

Positive transition (i.e from 0 to 1)

- The output can be Q or \bar{Q} (Q 's complement)

Truth table for S-R flip flop is,

S	R	On	Qn+1 state
0	0	0	0
0	0	1	1 No change
0	1	0	0
0	1	1	0 Reset
1	0	0	1
1	0	1	1 Set
1	1	0	x Indeter-
1	1	1	x minate

Truth table of S-R flip flop

1. Whenever the clock input changes from 0 to 1

i) Q is cleared to '0' if $S=0$ and $R=1$

ii) Q is set to '1' if $S=1$ and $R=0$

iii) There will be no change in output, if $S=0$ and $R=0$

iv) Q is Indeterminate (or) unpredictable, if $S=R=1$.

2. When the clock 'clk' has no signal, then there will be no change in output.

Q_n	S	R	Q_{n+1}
0	0	0	0 Q_n
0	0	1	0 Q_n
0	1	0	1
0	1	1	X
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	X

Characteristic table of SR flipflop.

K-map for characteristic table is

$Q_n \backslash S$	00	01	11	10
0			X 1	
1	1		X 1	

Characteristic eq n is $Q_{n+1} = S + Q_n \bar{R}$

Excitation table for SR flipflop is

present state Q_n	Next state Q_{n+1}	Required inputs	
		S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

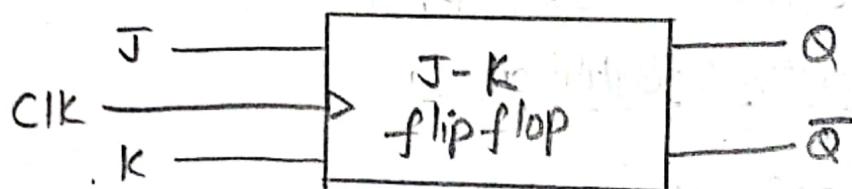
J-K flipflop :

In an S-R flipflop, the state of the output is not predictable when $R=1$ and $S=1$.

The J-K flipflop allows inputs $J=K=1$.

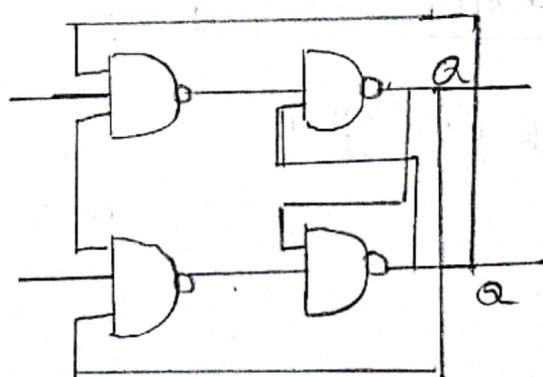
In this situation, the state of the output is changed.

- In J-K, the letter J is for set and the letter K is for Reset (clear).
- When inputs are applied to both J and K simultaneously as 1, the flipflop switches to its complement state i.e $Q=1$, it switches to $Q=0$ and viceversa.



Graphical representation of J-K flip flop

logical diagram for J-K flipflop



Truthtable

J	K	Q_n	Q_{n+1}	state
0	0	0	0	
0	0	1	1	Nochange
0	1	0	0	
0	1	1	0	Reset
1	0	0	1	
1	0	1	1	set
1	1	0	1	Toggle
1	1	1	0	(Complementary)

Characteristic table

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

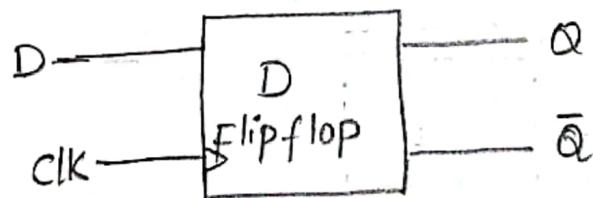
Excitation table

present state	Q_{n+1}	Required inputs	
		J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

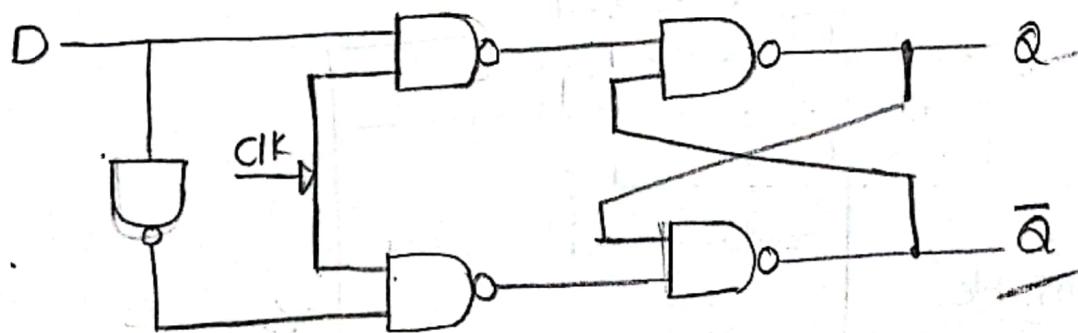
D Flip-Flop:

The D flip-flop is also known as Delay flip-flop (or) Data flip-flop.

- The D flip-flop is a slight modification of the SR flip-flop.
- An SR flip-flop is converted to a D flip-flop by inserting an inverter between S and R and assigning the symbol D to the single input.
- The D input is sampled during the occurrence of a clock transition from 0 to 1.
- D flip-flop is used to either store the data or introduce a delay.



Graphic symbol of D flipflop



Circuit diagram of D flipflop

operation . Truth table for D flipflop

If $D=1$, the output of flipflop goes to the 1 state
but if $D=0$, the output of flipflop goes to the 0 state.

TruthTable

D	Q_n	Q_{n+1}	state
0	0	0	Reset
	1	0	
1	0	0	Set
	1	1	

TruthTable of D-flipflop

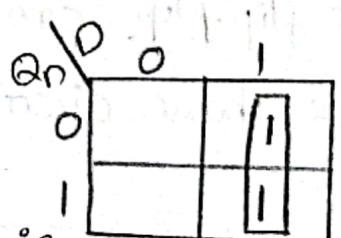
The output terminals of D-flipflop are same as that of S-R flipflop.

- D-flipflop is said to be an extension or improved version of S-R flipflop.
- The output values are applied to the input terminal. The input values and its corresponding outputs are represented in the characteristic table.

Characteristic table :

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

K-map for characteristic equation is



Characteristic equation for D flipflop is,

$$Q_{n+1} = D$$

Excitation table for D-flipflop

<u>PS</u> Q_n	<u>NS</u> Q_{n+1}	<u>Required J Input</u> D
0	0	0
0	1	1
1	0	0
1	1	1

For a D flip flop, the next state is always equal to the D input and it is independent of the present state.
 \therefore D must be 0 if Q_{n+1} has to be 0, and
D must be 1 if Q_{n+1} has to be 1
regardless of the value of Q_n .

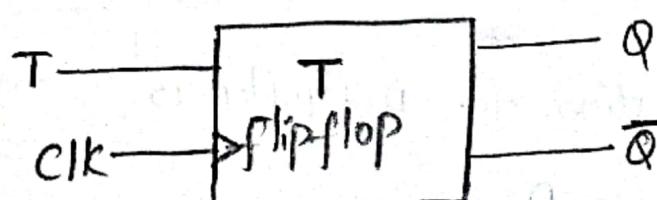
T-flipflop:

A T flipflop has a single control input, labelled T for toggle.

Whenever a pulse is given to the T input, the output changes state.

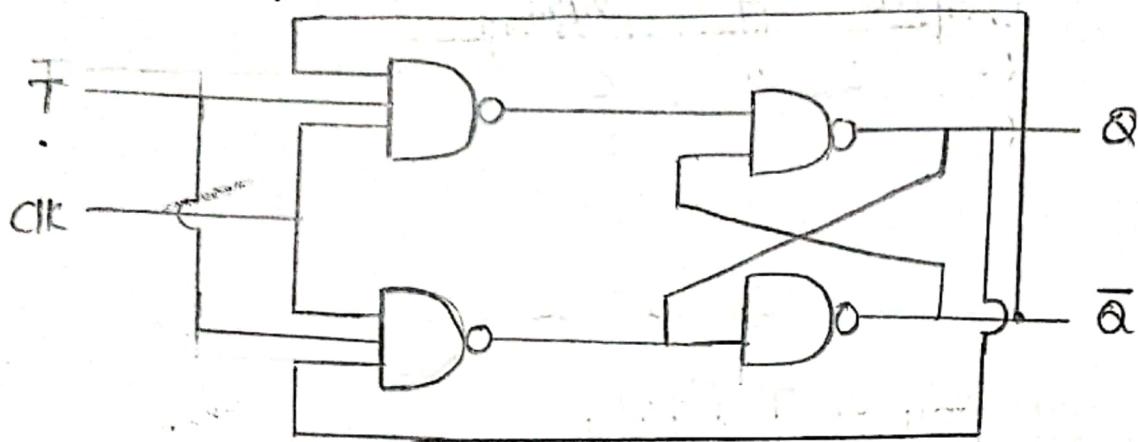
T flipflop can be used as a frequency divider.

The basic circuit diagram for T flipflop is



Circuit diagram

Circuit diagram:



It is easy to convert a J-K flipflop to the functional equivalent of a T flipflop by just connecting J and K together and labelling the common connection as T.

Truth table

T	Qn	Qn+1	state
0	0	0	No change
0	1	1	No change
1	0	1	Toggle
1	1	0	Toggle

Truth table of T flipflop

operation

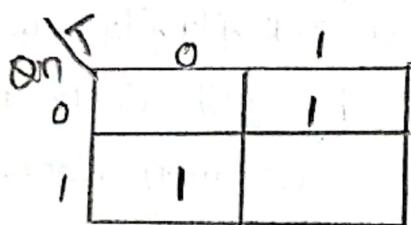
When $T=0$, there is no change of state

When $T=1$, the flipflop toggles.

Characteristic table for T-flipflop :

<u>Present state</u>	<u>Input</u>	<u>Next state</u>	
Q_n	T	Q_{n+1}	
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	1

K-map for Q_{n+1} of T flipflop



The characteristic equation of T-flip flop is

$$Q_{n+1} = \overline{Q_n}T + Q_n\overline{T}$$

Excitation table of T flipflop

<u>present state</u>	<u>next state</u>	<u>Required Input</u>
Q_n	Q_{n+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

Counters:

Counters are one form of registers, which can be used to count the number of clock pulses applied at input over a period of time.

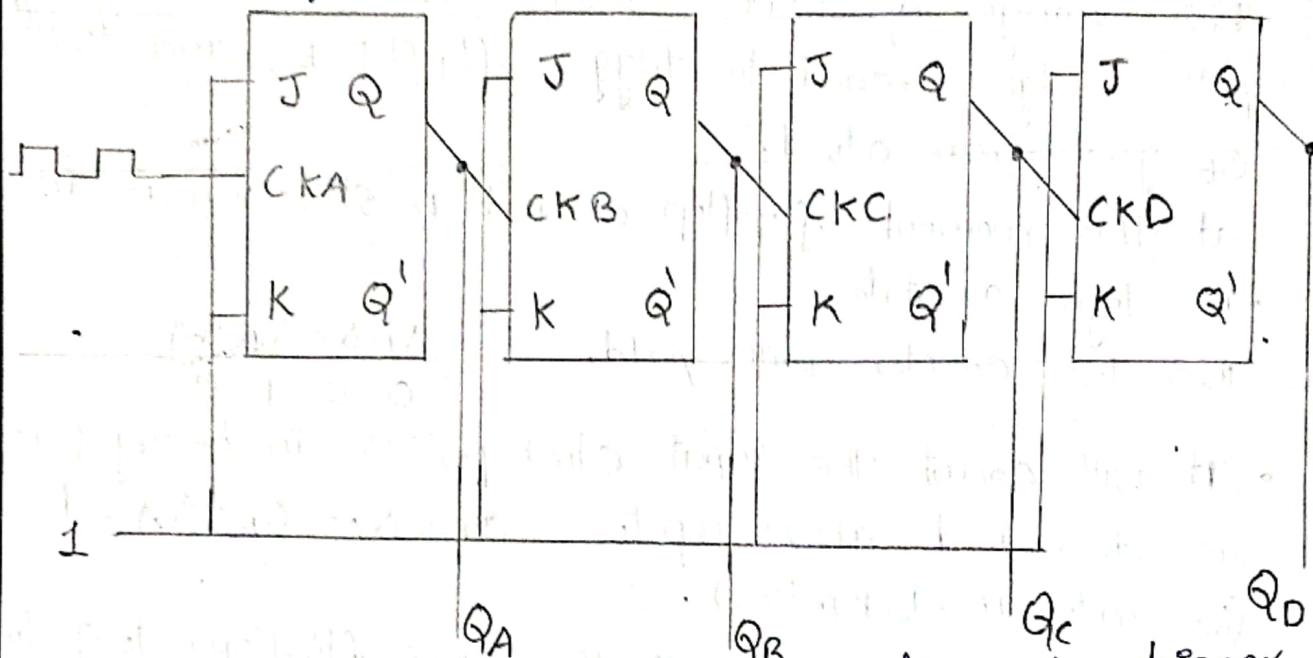
- Counters are found in almost all equipment containing digital logic.
- Counters are mostly constructed using JK flipflops in toggle mode or T flipflops.
- Flipflop programmed as counters are used in a wide variety of counting applications in scientific instruments, industrial controls, computers and communication equipments, as well as in many other areas.
- A counter that follows the binary number sequence is called a binary counter.
- There are two types of counters,
 1. Synchronous counter
 2. Asynchronous counter.
- In synchronous counter, the common clock input is connected to all of the flipflop and they are clocked simultaneously.
- In Asynchronous counter, commonly called ripple counters, the first flipflop is clocked by the external clock pulse and then each successive flipflop is clocked by the output of the previous flipflop.

Asynchronous or Ripple Counter:

Ripple counters are simplest possible counters.

- The circuit diagram of a ripple counter is constructed using JK flipflop.
- The J and K of all the flipflops are connected to high
- (1). so that all the flipflops are in toggle mode.
- Flipflops considered here are all negative edge triggered.
- Ripple counter is a cascaded arrangement of flipflops where the output of one flipflop drives the clock input of the following flipflop.
- A n-bit ripple counter can count up to 2^n states.
- It is also known as MOD n counter.
- Some of the features of ripple counter are
 1. It is an asynchronous counter
 2. Different flipflops are used with a different clock pulse.
 3. All the flipflops are used in toggle mode.
 4. only one flipflop is applied with an external clockpulse and another flipflop clock is obtained from the output of the previous flipflop.
 5. The flipflop applied with an external clock pulse act as LSB (Least Significant Bit) in the counting sequence.

A simple 4-bit binary ripple counter that uses four J-K flipflops.



- Suppose that the flipflops are negative edge-trigger type.
Initially all flipflops are in logic 0 state ($Q_A = Q_B = Q_C = Q_D = 0$).
A clock pulse is applied to the clock input of flipflop A, causing Q_A to change from logic 0 to logic 1.
At this moment, flipflop B, C and D do not change state (i.e. remain in the logic 0 state)
since there is no negative going edge of the clock pulse present at their clock input.
If Q_A and Q_D denote a LSB and MSB, respectively,
after the first clock pulse is applied to the clock input to flipflop Q_A ,

the counter will read

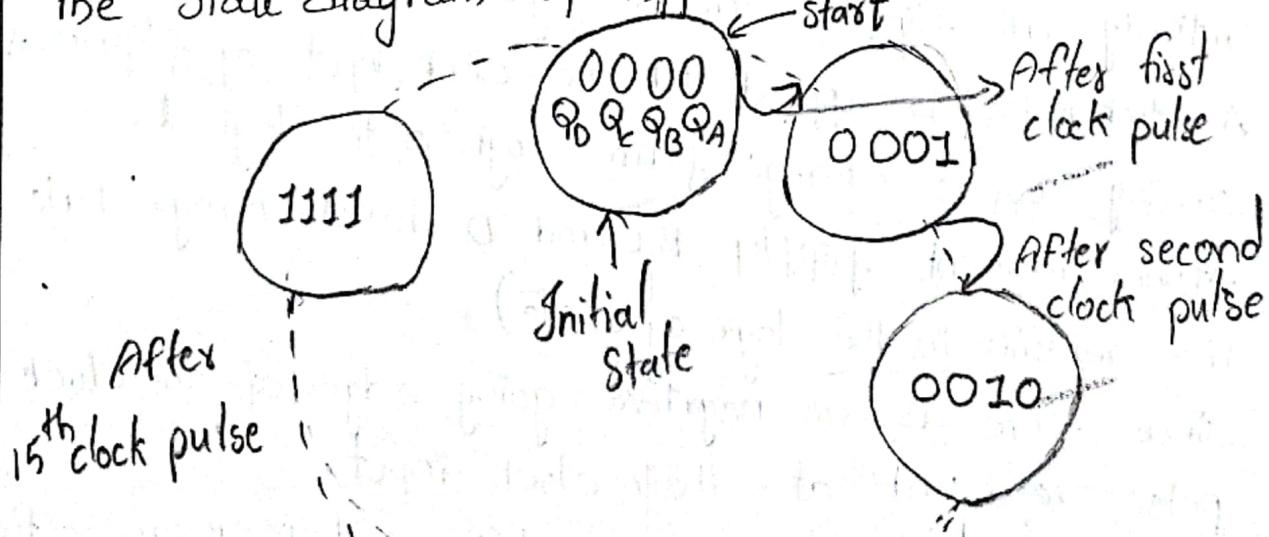
$$\begin{array}{l} Q_D \quad Q_C \quad Q_B \quad Q_A \\ \hline 0 \quad 0 \quad 0 \quad 1 \end{array}$$

- with the arrival of the second clock pulse to flipflop A, Q_A goes from 1 to 0. This change of state creates the negative-going pulse edge needed to trigger flipflop B, and thus Q_B goes from 0 to 1. At this moment flipflop C and D still remain in the logic 0 state. Thus the counter will yield.

$$\begin{array}{l} Q_D \quad Q_C \quad Q_B \quad Q_A \\ \hline 0 \quad 0 \quad 1 \quad 0 \end{array}$$

- It will count the input clock pulses in binary form as described above up to $Q_D = Q_C = Q_B = Q_A = 1$ (i.e upto 15 clock pulses). Clock pulse 16 causes all the four flipflops to go to state 0—that is the counter recycles.

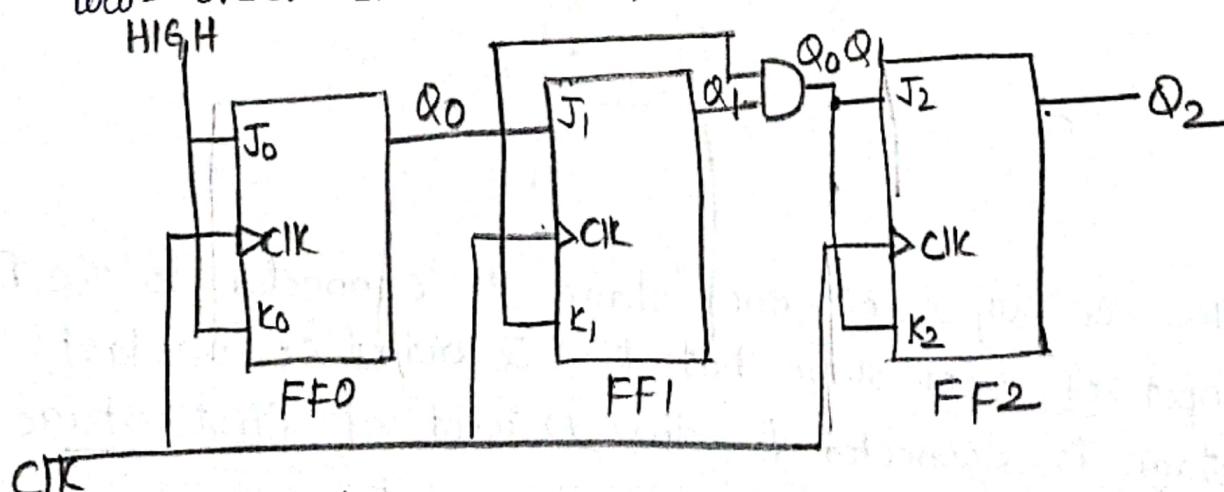
The state diagram of ripple counter is,



Synchronous counter:

To increase the speed of the counter the clock pulse were to be applied to all the flipflop at the same time affecting the change in all the flipflops simultaneously. Such counters are known as synchronous counters.

- In synchronous counters all the four flipflops are controlled by the same clock.
- The J and K inputs of the first flipflop are connected to High (1). Therefore the first flipflop would toggle for each clockpulse.
- The counter is based on the principle that in the sequence 0000; 0001, 0010, 0011, 0100, 0101, 0110 etc, the low order bit is complemented after every count and every other bit is complemented if the low-order bits are equal to 1.



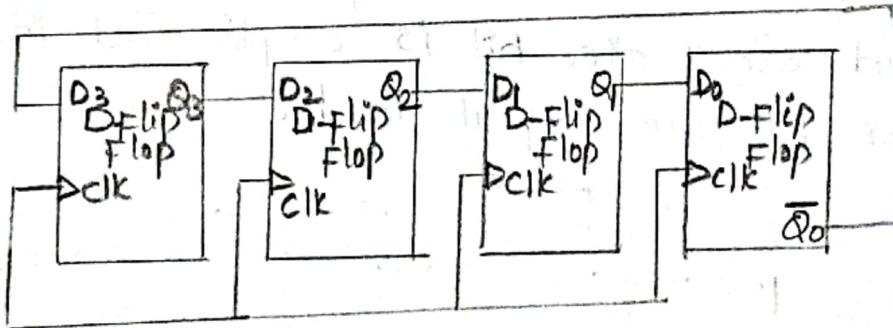
A 3-bit synchronous binary Counter

Twisted Ring counter (Johnson Counter)

Johnson counter also known as creeping counter, is an example of synchronous counter.

- In Johnson counter, the complemented output of last flipflop is connected to input of first flipflop and to implement n -bit Johnson counter we require ' n ' flipflops.
- It is one of the most important type of shift register counter.
- It is informed by the feedback of the output to its own input.
- Other names of Johnson counter are: bit counter, creeping counter, twisted ring counter, walking counter, mobile counter and switch tail counter.

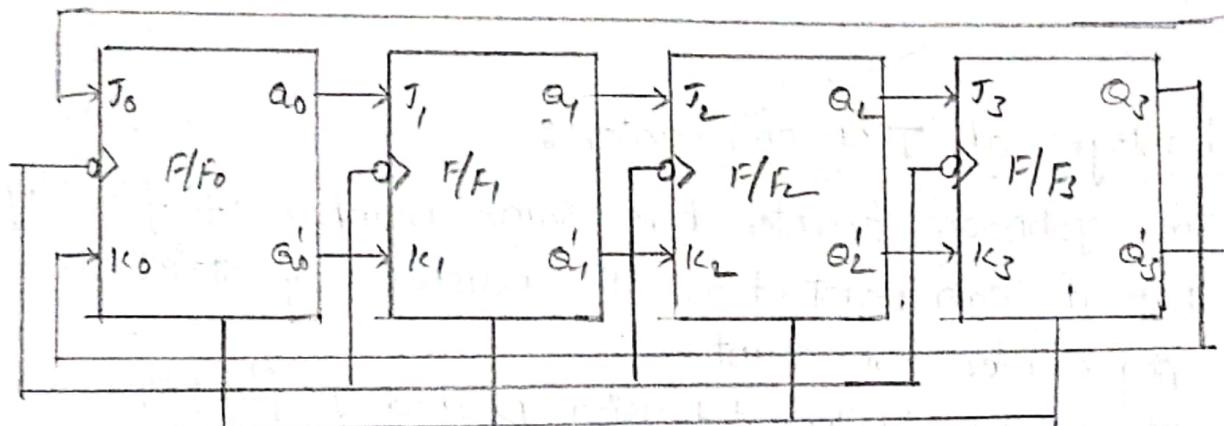
The logic diagram of 4-bit Johnson counter using D flip flops is,



- The Q output of each stage is connected to the D input of next stage but the Q output of the last stage is connected to the D input of first stage, therefore the name twisted ring counter.

This feed back arrangement produces a unique sequence of states.

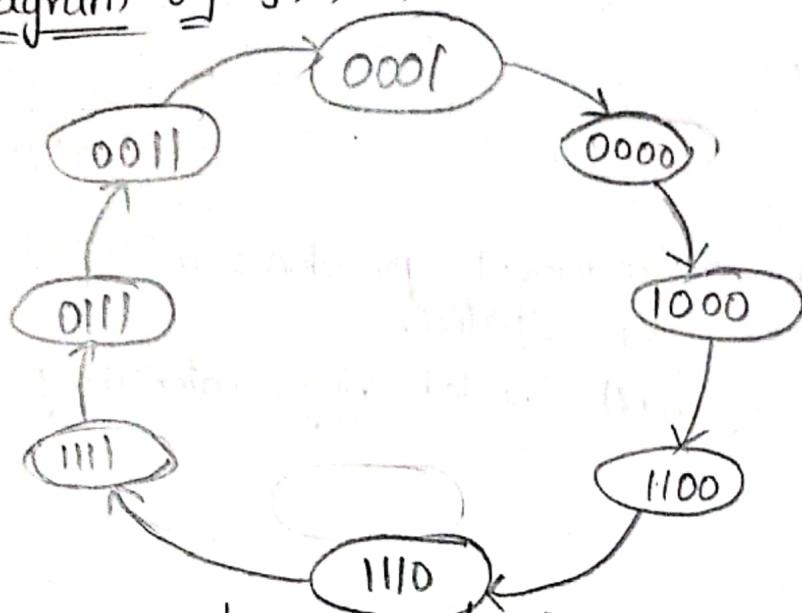
logic diagram of 4-bit twisted ring counter using JK flipflop



Truth table for Johnson counter

State	Q ₀	Q ₁	Q ₂	Q ₃
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0
4	1	1	1	1
5	0	1	1	1
6	0	0	1	1
7	0	0	0	1
8	0	0	0	0

State diagram of Johnson Counter



Advantages of Johnson Counter:

1. The Johnson counter has same number of flipflop but it can count twice the number of states the ring counter can count.
2. It can be implemented using D and JK flipflops
3. Johnson ring counter is used to count the data in a continuous loop.
4. Johnson counter is a self-decoding circuit.

Disadvantages of Johnson counter:

1. Johnson counter doesn't count in a binary sequence.
2. In Johnson counter more number of states being utilized.
3. The number of flipflops needed is one half the number of timing signals.
4. It can be constructed for any number of timing sequence.

Comparison between Asynchronous and Synchronous counter

Asynchronous Counter	Synchronous Counter
<ol style="list-style-type: none">These are serial countersi.e In these counters clock is connected to the first flip flop and its output acts as a clock input to the next flip flop and so onlogic circuit is very simple even for more number of statesspeed of operation is very low	<ol style="list-style-type: none">These are parallel countersIn these counters clock is connected simultaneously to all the flip flops.logic circuit becomes complex as number of states increases.speed of operation is relatively high.

Applications of counters:

Basically counters are useful for generating timing variables to sequence and control the operations in a digital system.

Hence the counters are very useful and versatile devices that are found in many applications.

1. Digital clock
2. Frequency dividers
3. parallel to serial data conversion (Multiplexing)
4. Auto parking control
5. Industrial digital control system etc.

Design of Synchronous counter:

The following steps are to be followed to design a simple synchronous counters.

- ① Determine the number of flip-flops needed using the formula $2^n \geq N$.
where n is the number of flipflops and
 N is the number of clock pulses to be counted.
2. select the type of flipflop to be used.
3. prepare the statetable from the given circuit information and derive the circuit excitation table.
4. Simplify the expressions for the flipflop inputs using any one of the simplification method (K-map or tabulation or algebraic)
5. Draw the logic diagram.

Design of Asynchronous counters:

To design an asynchronous counter the following steps are to be followed.

1. write the counting sequence
2. Tabulate the values of reset signal R for various states of the counter.
3. obtain the minimal expression for R or \bar{R} using K-map or any other method.
4. provide a feedback such that R or \bar{R} resets all the flipflops after the desired count.

Register :

Register is a series of flipflops. (or)
A register is a group of flipflops with each flipflop capable of storing one bit of information.
can store any discrete quantity

- A register with n flipflop of information contains n bits.

- The flipflop hold the binary information and the gates control when and how new information is transferred into the register.

- The transfer of new information into a register is referred to as loading the register.

- The registers can be classified into four possible modes of operation.

They are

1. serial in - serial out

2. serial in - parallel out

3. parallel in - serial out

4. parallel in - parallel out

Applications of registers :

The registers are found in many applications, a few of them are:

1. Time delay 2. Keyboard encoder

3. Ring counter

4. Serial to parallel and parallel to serial data converter.

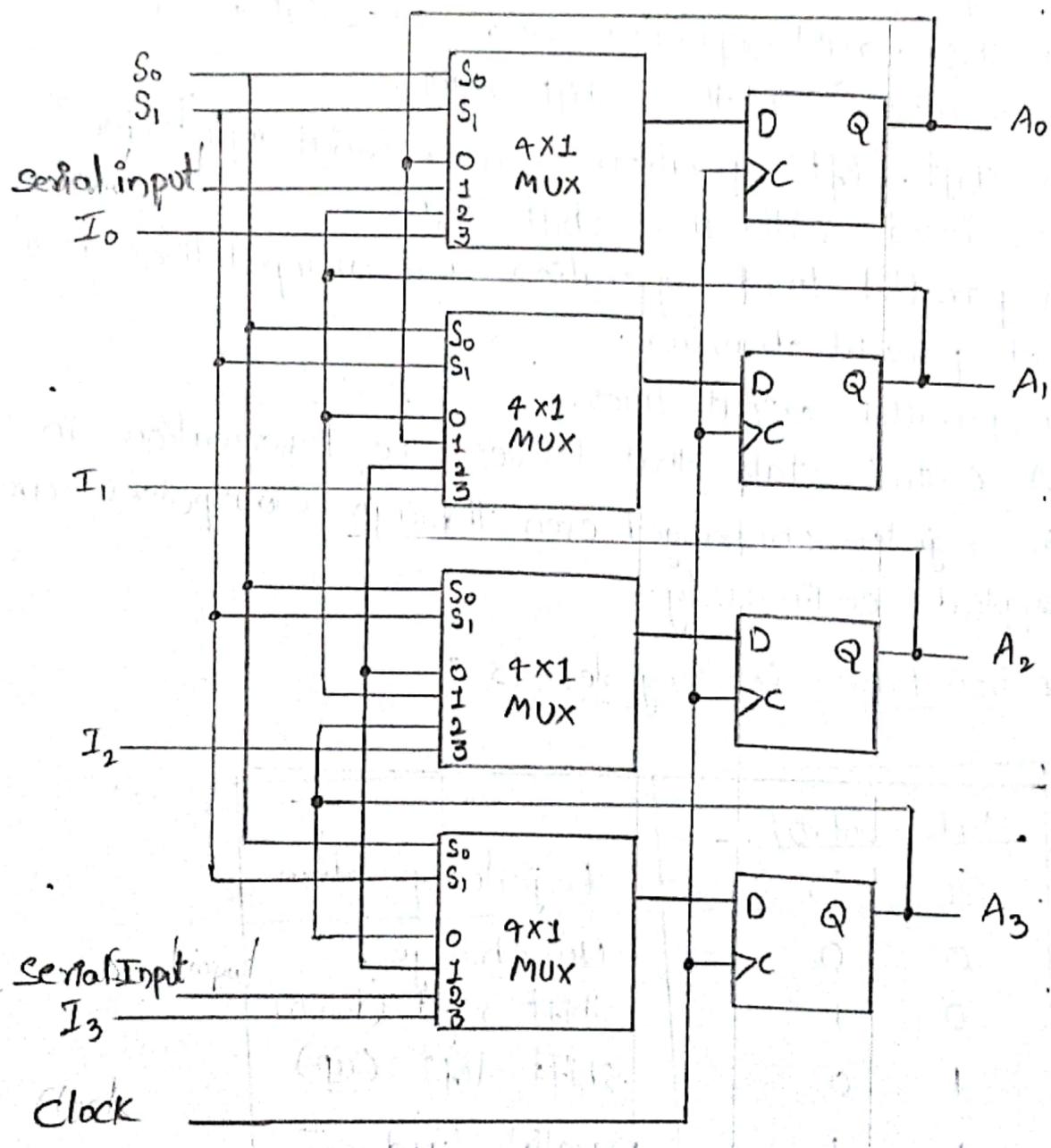
Shift Registers :

A register capable of shifting its binary information in one or both directions is called a shift register.

- The logical configuration of a shift register consists of a chain of flipflops in cascade, with the output of one flipflop connected to the input of the next flipflop.
- All flipflops receive common clock pulses that initiate the shift from one stage to the next.
- The simplest possible shift register is one that uses only flip-flops.
- The output of a given flipflop is connected to the D input of the flip-flop at its right.
- The clock is common to all flip-flops.
- The serial input determines what goes into the left most position during the shift.
- The serial output is taken from the output of the rightmost flipflop.
- sometimes it is necessary to control the shift so that it occurs with certain clock pulses but not with others.
- This can be done by inhibiting the clock from the input of the register if we do not want it to shift.
- When the shift register is used, the shift can be controlled by connecting the clock to the input of an AND gate and the 2nd output of the AND gate

can then control the shift by inhibiting the clock.
it is also possible to provide extra circuits to control the shift operation through the D inputs of the flip-flops rather than the clock input.

Bidirectional shift Register with parallel load



A register capable of shifting in one direction only is called a unidirectional shift register.

- A register that can shift in both directions is called a bidirectional shift register.
- The most general shift register has all the capabilities. Others may have some of these capabilities, with at least one shift operation.
- 1. An input for clock pulses to synchronize all operations.
- 2. A shift-right operation and a serial input line associated with the shift-right.
- 3. A shift-left operation and a serial input line associated with the shift-left.
- 4. A parallel load operation and n input lines associated with parallel transfer.
- 5. n parallel output lines.
- 6. A control state that leaves the information in the register unchanged even though clock pulses are applied continuously.

Function table for Register is :

Mode control		Register operation
S_1	S_0	
0	0	No change
0	1	Shift right (down)
1	0	Shift left (up)
1	1	parallel load