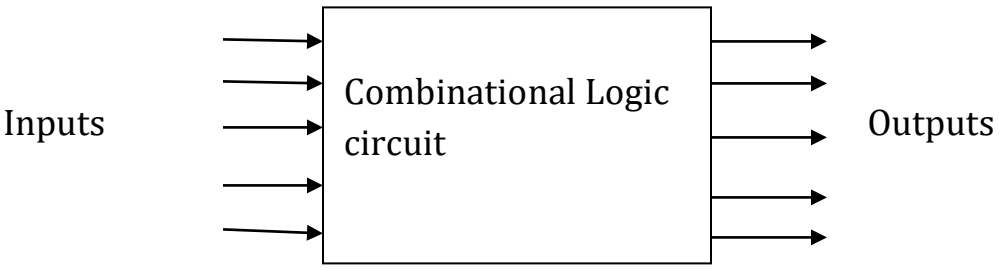
MODULE 5
SEQUENTIAL AND COMBINATIONAL LOGIC

LOGIC CIRCUITS

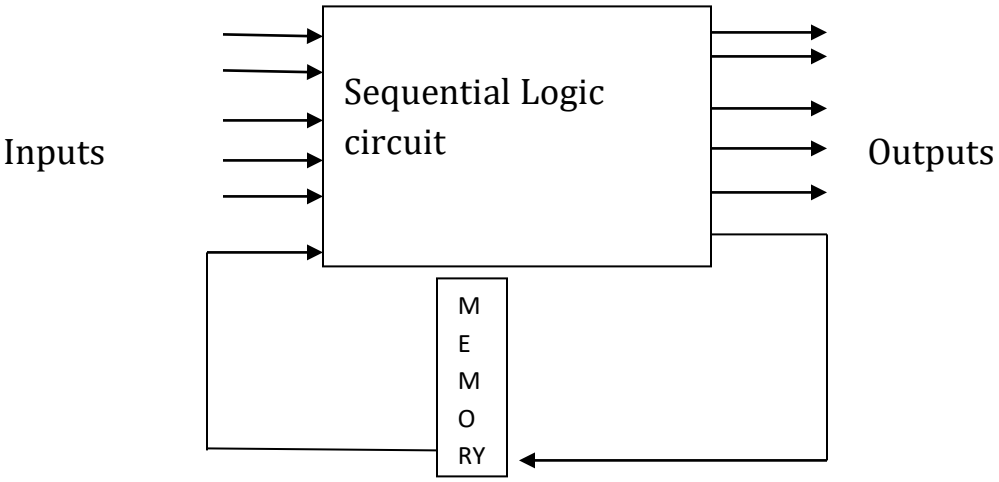
Logic circuits are divided in to two

- Combinational Logic circuit
- Sequential Logic Circuit

In combinational circuit, each output depends entirely on the immediate (present) input to the circuit.
Block diagram is



In Combinational Logic circuit, the output depends on both present and past inputs. It means that this type of circuit involves the memory elements for storing previous input conditions



Sequential circuit can be classified as

- Synchronous sequential circuit
- Asynchronous sequential circuit

Synchronous sequential circuit- A system whose behaviour can be defined from the knowledge of its signal at a discrete instant of time

Asynchronous sequential circuit-Here, the behaviour of sequential circuit depends on the order in which its input signal change and can be affected at any moment of time

MEMORY ELEMENTS

A memory element is a circuit which can remember values for a long time. The activation for the memory element can be done in two forms, namely, pulse-triggered and edge trigged. Pulse triggered circuit is known as latch and edge triggered circuit is known as flip-flop

FLIP-FLOPS

A flip-flop is a memory element that can store 1 bit of information. It is a sequential logic circuit and it can maintain the binary state indefinitely until directed by an input signal

A flip-flop can be constructed either using two NAND gates or using two NOR gates.

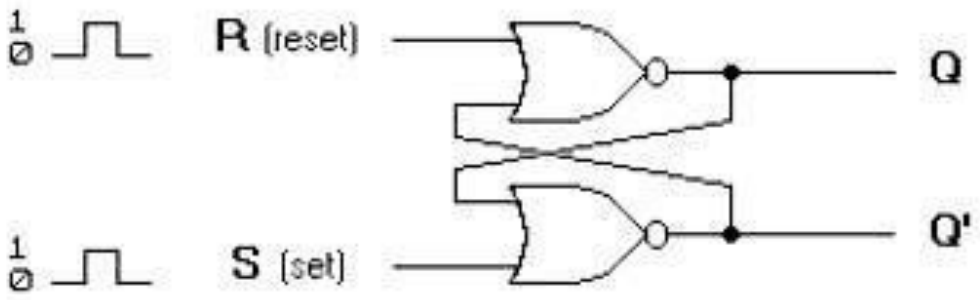
Each flip-flop has two outputs Q and Q'

When the output of Q is 0 then Q' is 1 and vice versa

The basic type of flip-flop is the RS-flip-flop or SR-latch

It has two inputs, set and reset ie S,R respectively

Basic flip-flop circuit with NOR gate

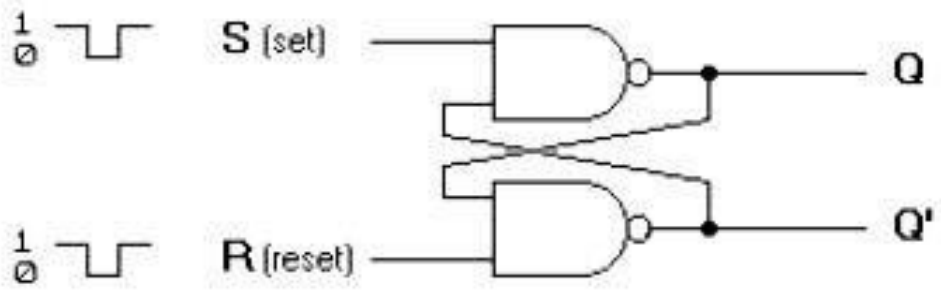


(a) Logic diagram

S R	Q Q'	
1 0	1 0	
0 0	1 0	(after S=1, R=0)
0 1	0 1	
0 0	0 1	(after S=0, R=1)
1 1	0 0	

(b) Truth table

Basic flip-flop circuit with NAND gate



(a) Logic diagram

S R	Q Q'	
1 0	0 1	
1 1	0 1	(after S=1, R=0)
0 1	1 0	
1 1	1 0	(after S=0, R=1)
0 0	1 1	

(b) Truth table

The cross-coupled connection from the output of one gate to the input of other gate constitute a feedback path

A flip-flop has two states. Set and Reset

When Q=1 and Q'=0, it is in set state

When Q=0 and Q'=1, it is in Reset state or clear state

Clocked RS-Flip-flop

By adding clock input to the basic flip-flop circuit, we can make the clocked RS Flip-flop. This circuit responds to the input level during the occurrence of a clock pulse.

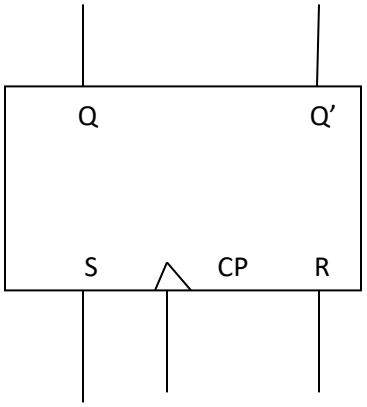
The clocked RS Flip-flop consists of a basic NOR flip-flop and two AND gates. It has 3 inputs S, R and clock pulse (CP)

The output of two AND gate remains at 0 as long as the clock pulse is 0.

When the clock pulse goes to 1, the information from the S and R input is allowed to reach the basic flip-flop

Set state is reached when S=1, R=0 and CP=1 & Reset state is reached when S=0, R=1 and CP=1

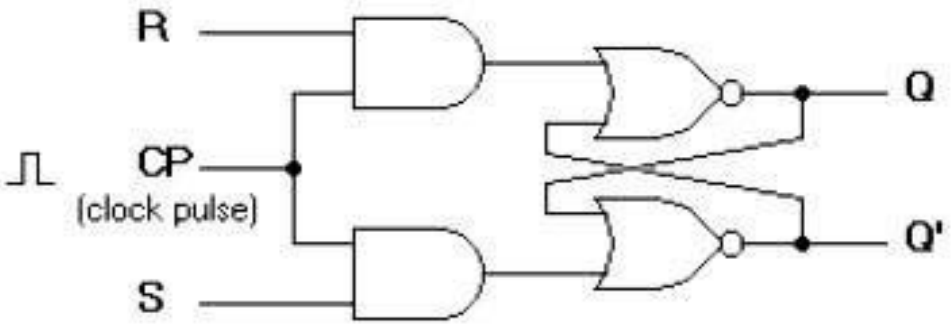
Graphic symbol(block diagram)



The triangle symbol in a CP input is called

Dynamic indicator

Logic diagram and truth table clocked RS Flip-flop



(a) Logic diagram

Q S R	Q(t+1)
0 0 0	0
0 0 1	0
0 1 0	1
0 1 1	indeterminate
1 0 0	1
1 0 1	0
1 1 0	1
1 1 1	indeterminate

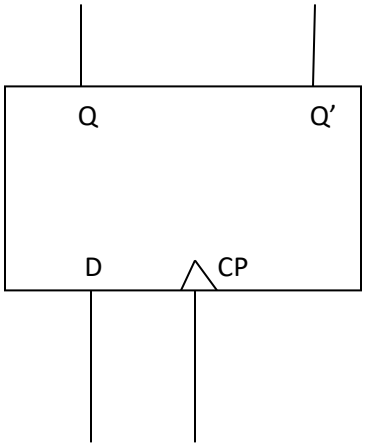
(b) Truth table

From the truth table Q denotes the current state of the flip-flop and Q (t+1) denotes the state after the cock pulse occurred

D Flip-flop

D flip-flop is the modification of clocked RS-flip-flop. It is constructed using NAND gates. The D input goes directly in to the S input and its complement is applied to the R input. The flip-flop enables only if the clock input is 1.It is basically an RS flip-flop with an inverter in the R input. The inverter reduces the number of inputs from two to one. This type of flip-flop is called gated-Latch

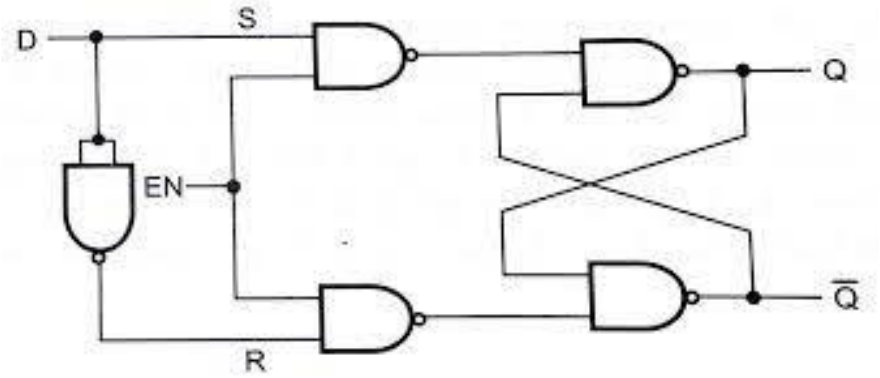
Graphic symbol(block diagram)



Truth table

Q	D	Q(t+1)
0	0	0
0	1	1
1	0	0
1	1	1

Logic diagram

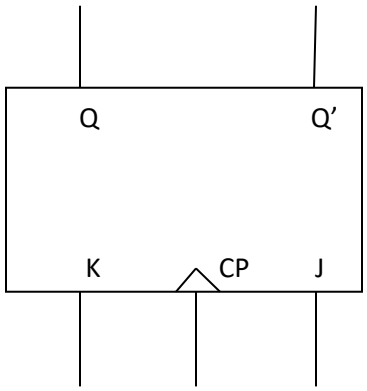


JK Flip-flop

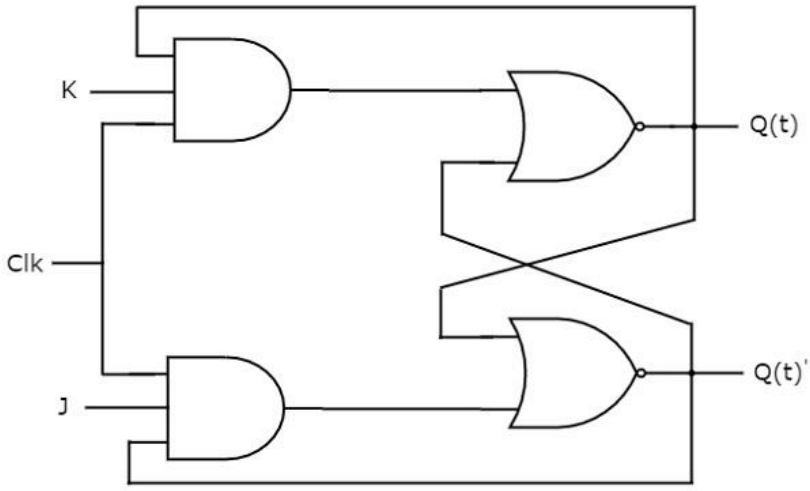
JK flip-flop is the refinement of the RS flip-flop .Here the indeterminate state of RS flip-flop is defined in the JK flip-flop. The inputs J and K are like the inputs S and R to set and clear the flip-flop .i.e. J is for Set(S) and K is for Reset(R)

When both J and K inputs are applied simultaneously(J=1,K=1), then the flip-flop switches to its complemented state. That is, if Q=1 then Q(t+1)=0 and vice versa. This is how JK flip-flop defines indeterminate state of RS flip-flop

Graphic symbol(block diagram)



Logic diagram



To define the indeterminate state, output of Q is ANDed with K and Clock inputs & output of Q' is ANDed with J and Clock inputs

Truth table

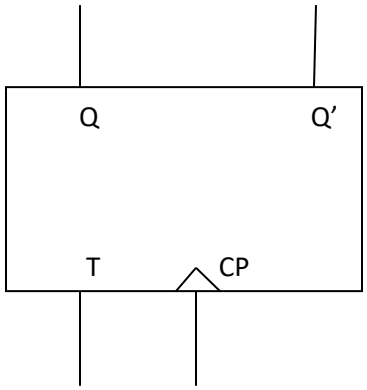
Q	J	K	Q(t+1)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

T Flip-flop

The T Flip-flop is a single input version of JK Flip-flop. T flip-flop is obtained from JK flip-flop if both inputs are tied together. 'T' means that 'toggle'. It denotes the ability of the flip-flop to toggle or changes the state

When T input is 1 , it goes to the complemented state .

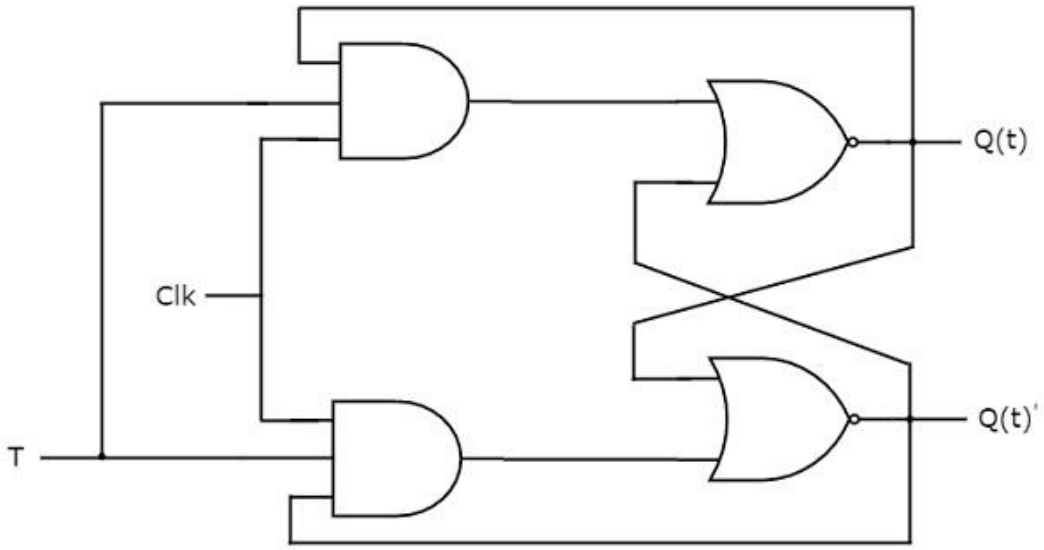
Graphic symbol(block diagram)



Truth table

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

Logic diagram

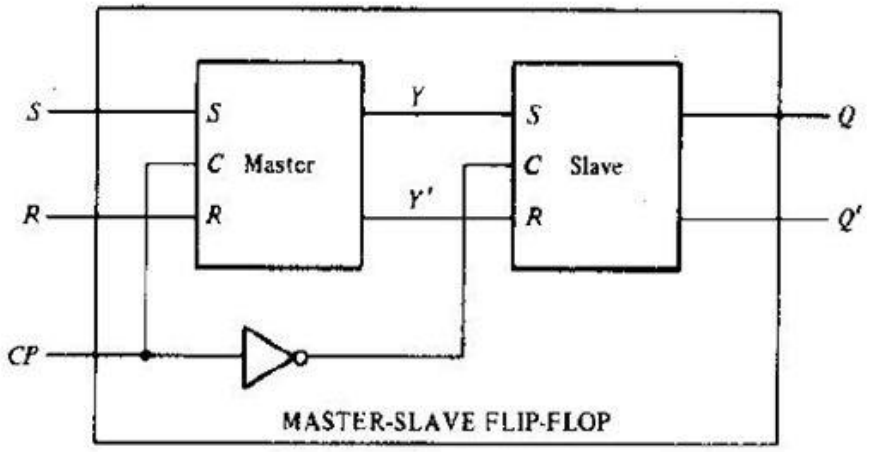


Master – Slave flip-flop

A master-slave flip-flop is constructed from two separate flip-flops.one circuit serves as a master and the other serve as a slave. The overall circuit is referred to as a master-slave flip-flop

When the clock pulse is 0, the clock input of slave is 1, and the slave is enabled. It generate Q and Q' outputs similar to Y and Y' in master flip flop

When the clock pulse is 1,the master flip-flop is enabled. Then the information at the external Rand S input is transmitted to the master flip-flop.



MASTER-SLAVE FLIP-FLOP

ADDERS

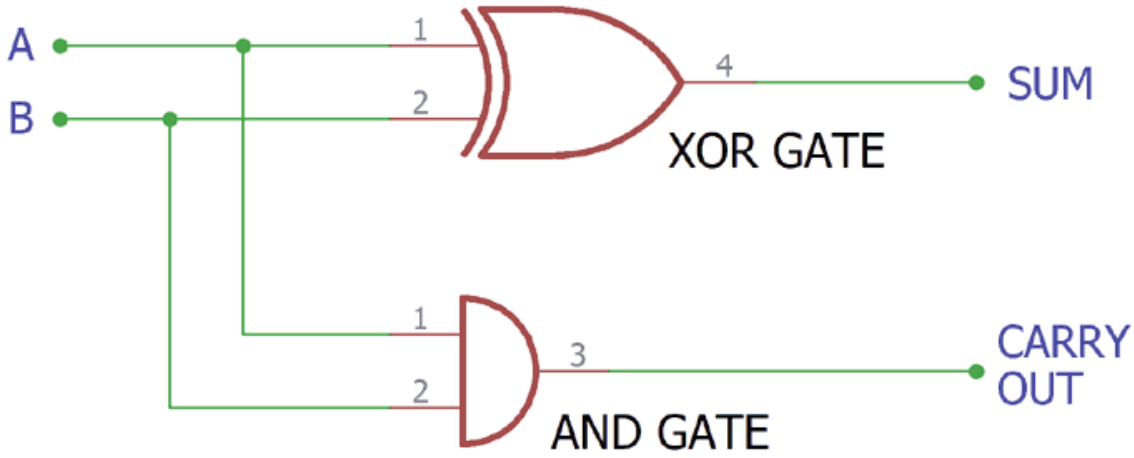
There are two types of adders

- Half adder
- Full adder

Half Adder

A half adder is a combinational circuit which perform addition of two binary digits. A half adder adds two1 bit binary number A,B . Outputs of half adder are the sum of two bits sum(S) and carry (C).A half adder can be constructed from one XOR gate and one AND gate

Block diagram & Logic diagram



Truth table of half adder

Truth Table			
Input		Output	
A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

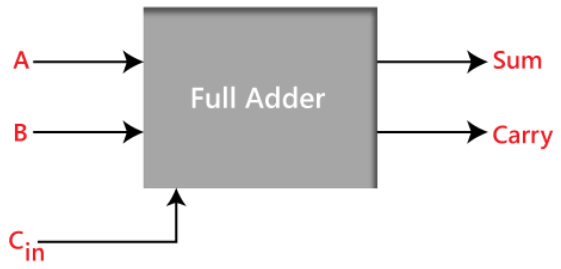
Full adder

Full adder is also a combinational digital circuit. It is used to add two binary numbers which contains more than1bits.

Full adder accepts 3 inputs and generates a Sum(S) output and Carry(C) output.

It is used to add 3 bits including carry generated from the previous bit addition

Block diagram



C_{in} is the input carry and C_{out} is the output carry after the addition

Truth table

Full Adder Truth table

INPUTS			OUTPUTS	
A	B	C_{in}	SUM	CARRY OUT
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

From the truth table

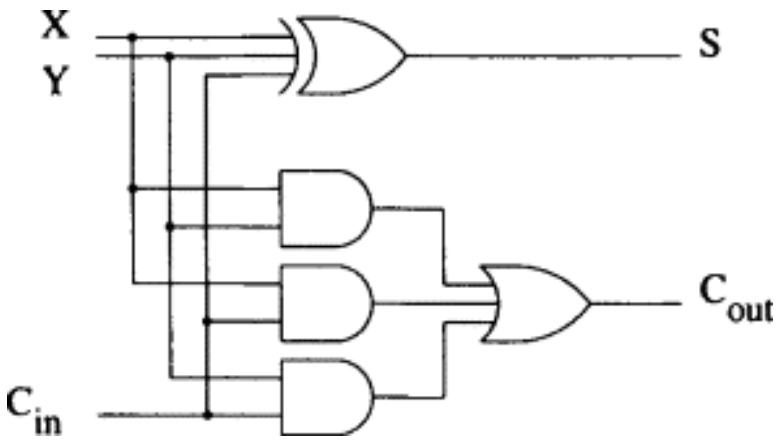
Output equation for sum is

$S=A \oplus B \oplus C_{in}$

Output equation for carry is

$C_{out}= AB + AC_{in} + BC_{in}$

Logic diagram

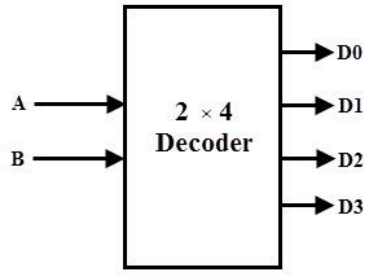


Decoder

A decoder is a combinational circuit that converts binary information from the n-coded inputs to a maximum of 2^n outputs

The basic function of decoder is to detect the presence of a specified combination of bits and to indicate that presence by a specified output level

Block diagram



Truth table of 2*4 decoder

Inputs		Outputs				X
A	B	d_0	d_1	d_2	d_3	
0	0	1	0	0	0	0
0	1	0	1	0	0	1
1	0	0	0	1	0	2
1	1	0	0	0	1	3

From the above truth table

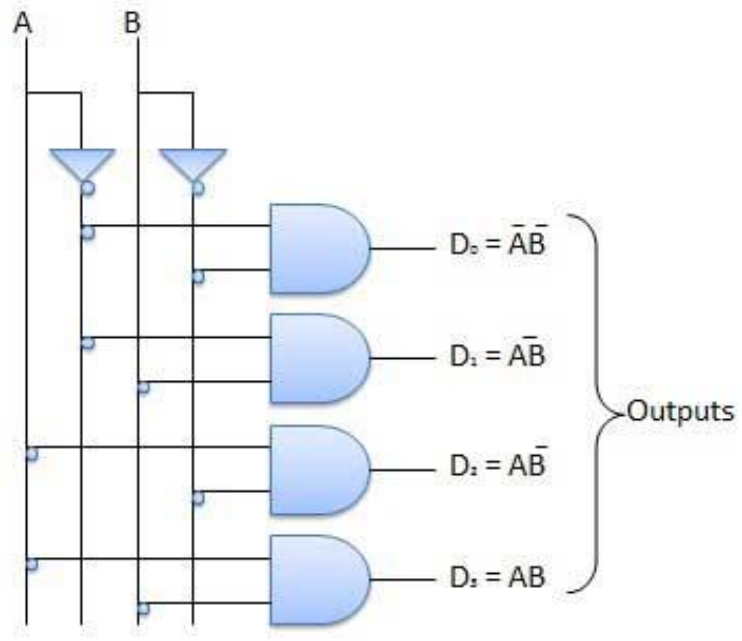
$D0=\bar{A} . \bar{B}$

$D1=\bar{A} . B$

$D2=A . \bar{B}$

$D3=A . B$

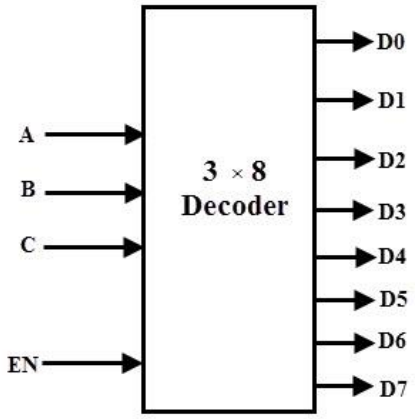
Logic diagram



3 * 8 Decoder

Here, there are 3 inputs and 2^3=8 outputs. Information coming from 3 input lines is passed onto 8 output lines

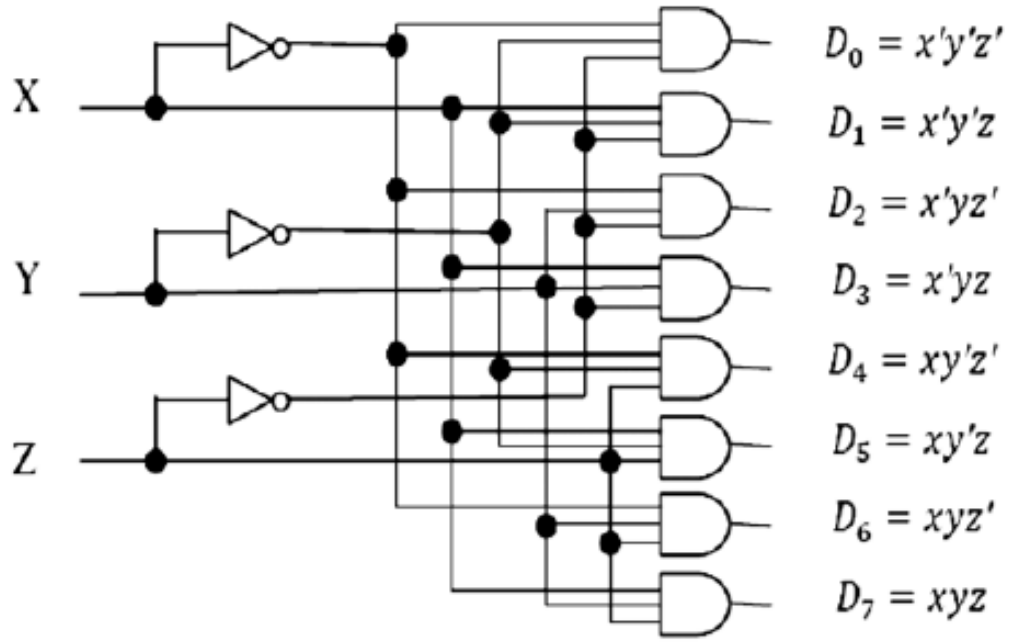
Block diagram



Truth table

Inputs			Outputs							
X	Y	Z	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
0	0	0	**1**	0	0	0	0	0	0	0
0	0	1	0	**1**	0	0	0	0	0	0
0	1	0	0	0	**1**	0	0	0	0	0
0	1	1	0	0	0	**1**	0	0	0	0
1	0	0	0	0	0	0	**1**	0	0	0
1	0	1	0	0	0	0	0	**1**	0	0
1	1	0	0	0	0	0	0	0	**1**	0
1	1	1	0	0	0	0	0	0	0	**1**

Logic Diagram

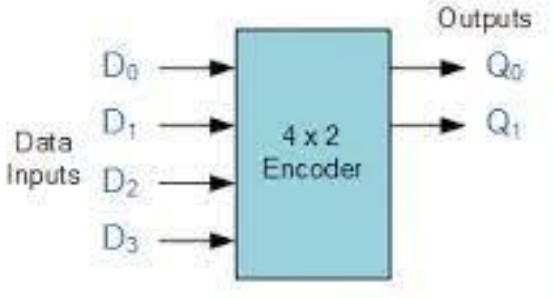


Encoder

Encoding is the reverse of decoding. An encoder has 2^n input lines and n output lines.

It is implemented with OR gates

Block Diagram



Inputs				Outputs	
D_3	D_2	D_1	D_0	Q_1	Q_0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1
0	0	0	0	x	x

Truth table

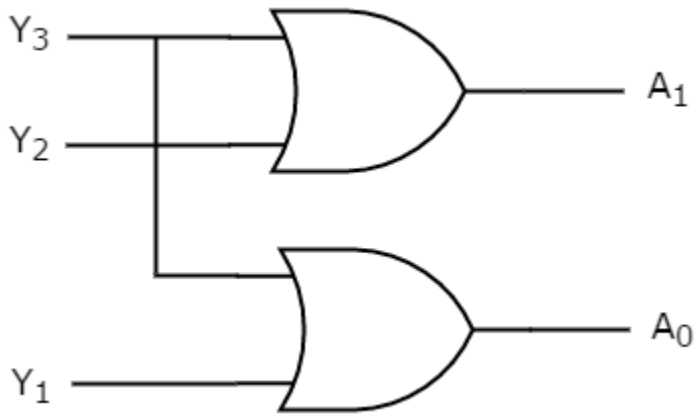
INPUTS				OUTPUTS	
Y3	Y2	Y1	Y0	A1	A0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

From the truth table

A0=Y1+Y3

A1=Y2+Y3

Logic diagram



8 to 3 Encoder

This encoder is also called octal-to-binary encoder. It has 8 inputs and 3 outputs

Truth table

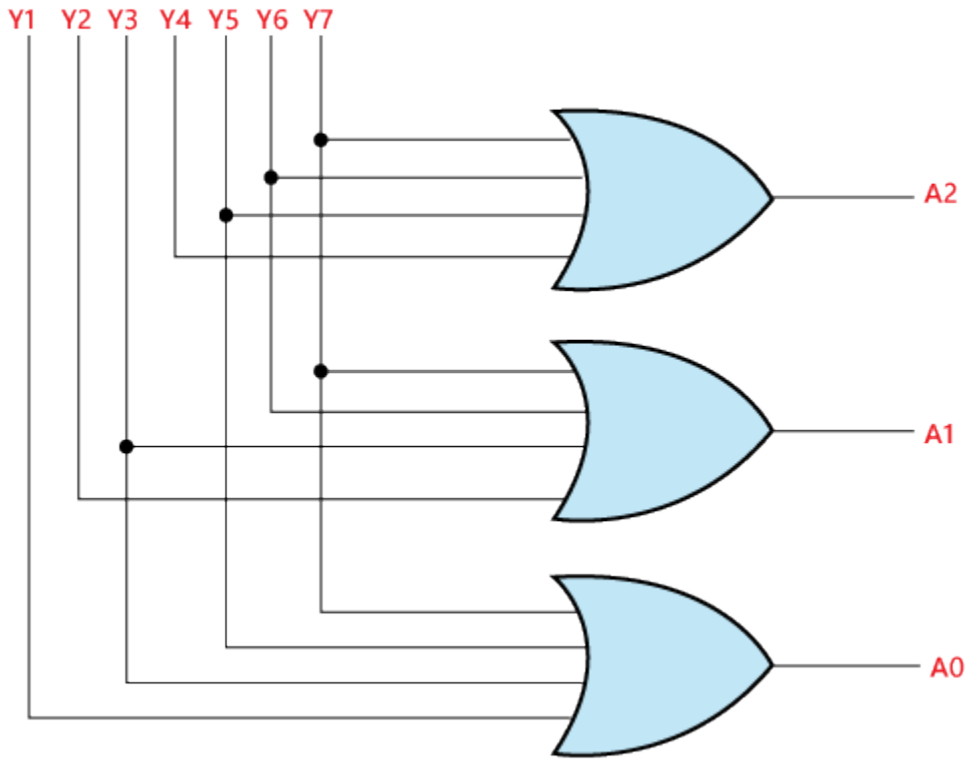
INPUTS								OUTPUTS		
Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0	A2	A1	A0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1

From the truth table

A0=Y1+Y3+Y5+Y7

A1=Y2+Y3+Y6+Y7

A2=Y4+Y5+Y6+Y7



MULTIPLEXERS (DATA SELECTORS)

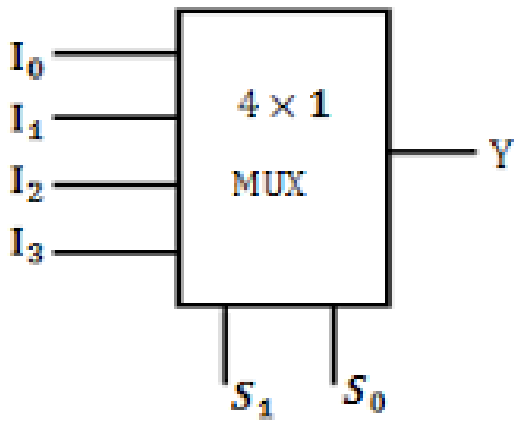
A multiplexer or (MUX) is a device that allows digital information from several sources to be routed on to a single line

It is a combinational circuit which select binary information from one of many input lines and directs it to a single output line

Normally there are 2^n input lines and n selection line whose bit combinations determine which input is selected for transmission

4 to 1 multiplexer is the common type, which has 4 input lines, 1 output line and 2 control lines or selection lines

Block diagram & Truth table



Truth table

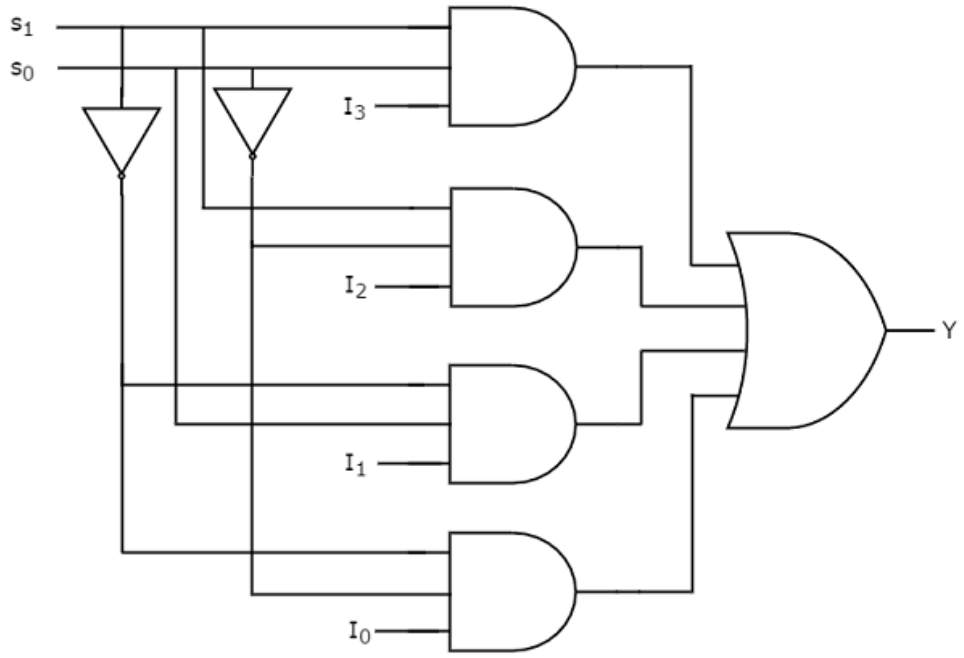
S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

Explanation

- If both selection lines are 0 then the input line I0 is going to the output line
  Ie if S0=0 and S1=0 1then I0 is selected for output
  So Y= I0$\overline{S0}$$\overline{S1}$
- Data output is equal to I1, If S0=1 and S1=0
  So Y=I1S0$\overline{S1}$
- Data output is equal to I2, If S0=0 and S1=1
  So Y=I2$\overline{S0}$S1
- Data output is equal to I3, If S0=0 and S1=1
  So Y=I3S0S1
- Total exzpression for the data output is

  Y= I0$\overline{S0}$$\overline{S1}$ + I1S0$\overline{S1}$ + I2$\overline{S0}$S1 + I3S0S1

Logic diagram



DE-MULTIPLEXERS (DATA DISTRIBUTORS)

De-multiplexer reverses the function of multiplexer.

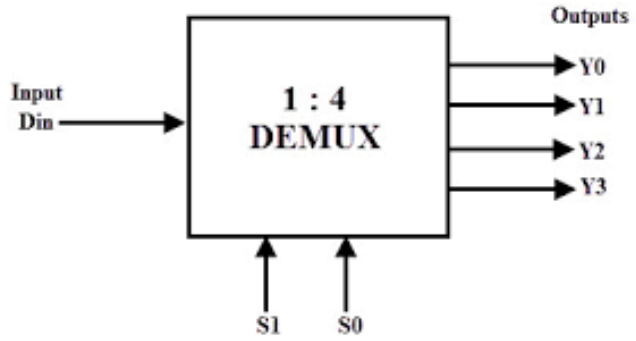
It takes digital information from one source and distributes it to a number of output lines. So they are also called data distributors

De-multiplexer has one input line and 2^n output lines.

The selection of specific output line is controlled by the bit values of n selection lines

1 to 4 de-multiplexer has 1 data input line, 4 output line and 2 selection lines

Block Diagram and Truth table

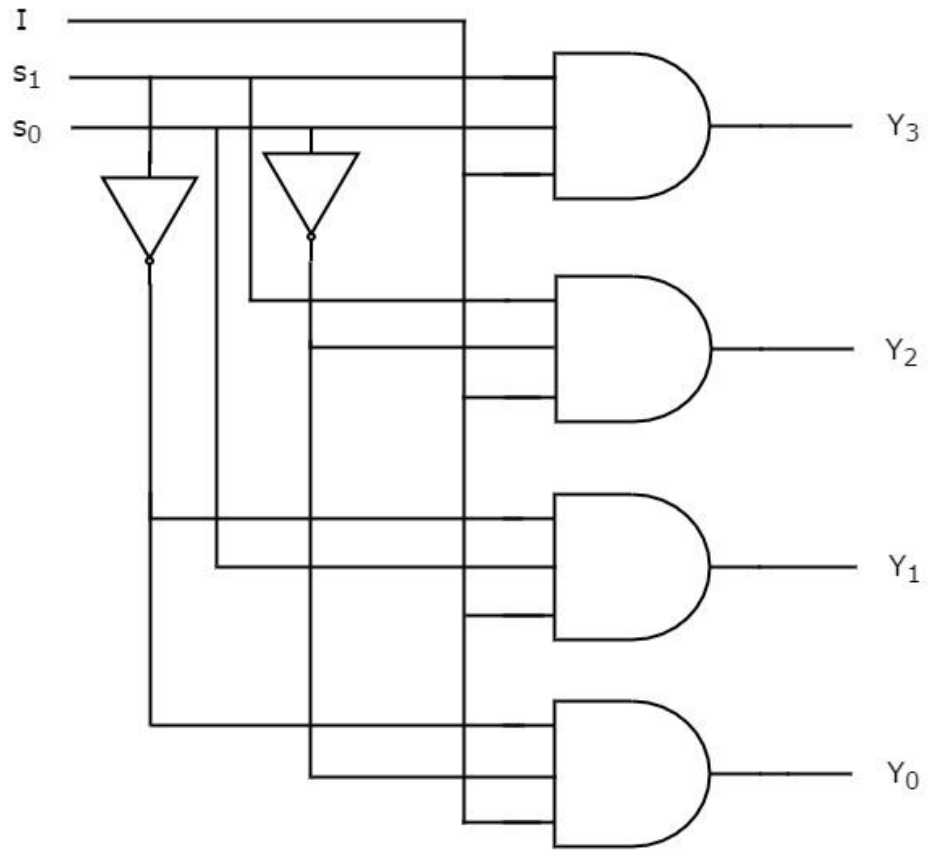


Input	Select Lines	Output Lines
I	S_1 S_0	D_0 D_1 D_2 D_3
I	0 0	1 0 0 0
I	0 1	0 1 0 0
I	1 0	0 0 1 0
I	1 1	0 0 0 1

Explanation

- The single input data I has a path to all four outputs, but only one output line is active based on the selection lines
- D0 is active when S1=0 and S0=0
  Ie $Y=I\bar{S}1\bar{S}0$
- D1 is active when S1=0 and S0=1
  Ie $Y=I\bar{S}1S0$
- D2 is active when S1=1 and S0=0
  Ie $Y=IS1\bar{S}0$
- D3 is active when S1=1 and S0=1
  Ie Y=IS1S0

Logic Diagram



Registers

A register is a group of binary storage cells suitable for holding binary information. A group of flip-flops constitutes a register.

Each flip-flop is a binary cell capable of storing one bit of information. An n-bit register has a group of n flip-flops and is capable of storing any binary information containing n bits

A register consists of a group of flip-flops and gates

Shift Registers

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

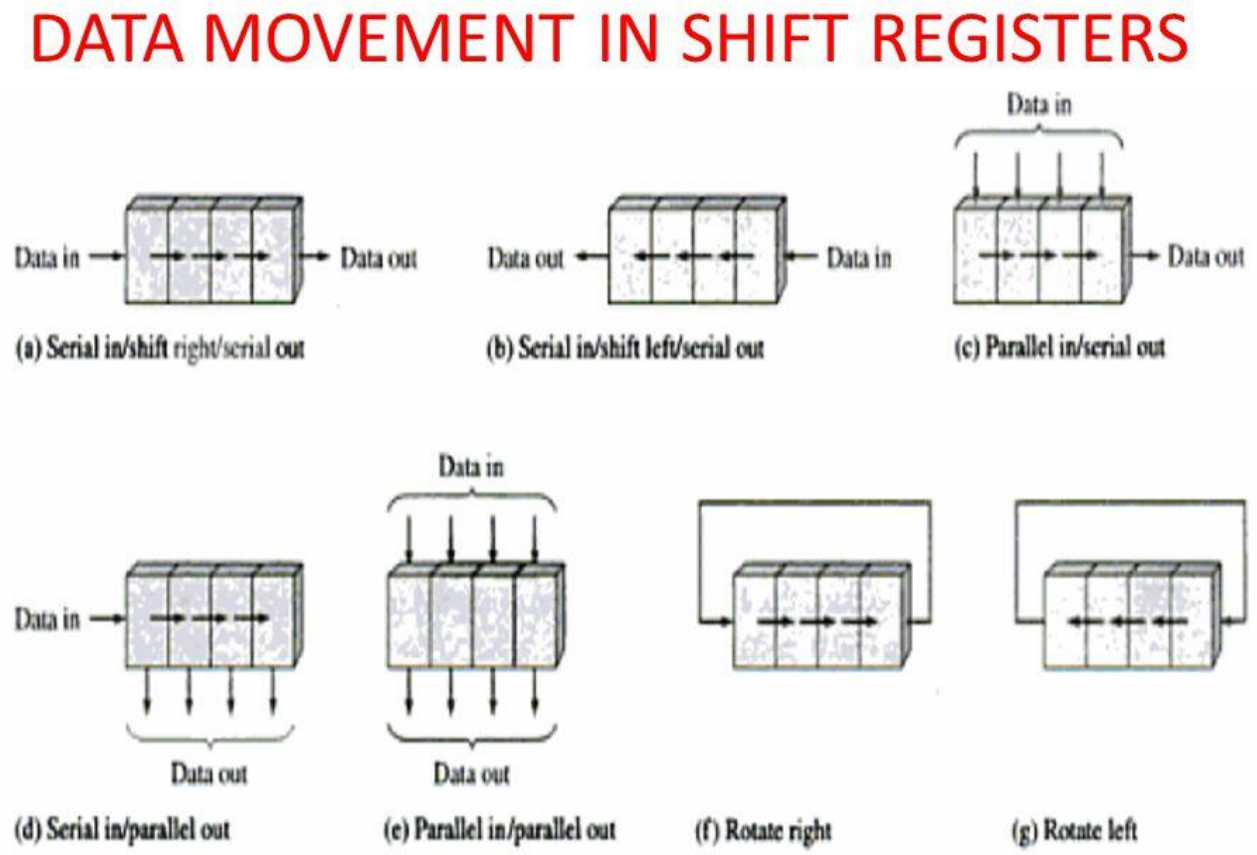
The logical configuration of a shift register consists of a chain flip-flops connected in cascade.ie with the output of one flip-flop connected to the input of next flip-flop.

All flip-flop receive common clock pulse, which causes the shift from one stage to the next

4 types of shift registers

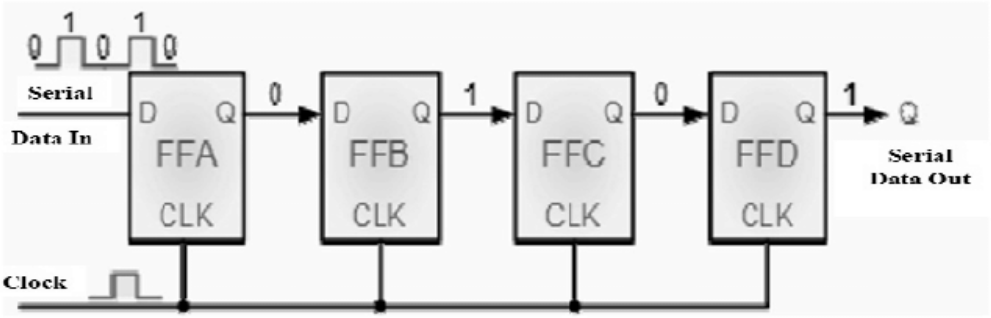
1. Serial-in-Serial-out shift register
2. Serial-in Parallel-out shift register
3. Parallel-in Serial-out shift register
4. Parallel-in Parallel-out shift register

Following figure shows the basic data movement in shift registers



Serial-in Serial-out shift register

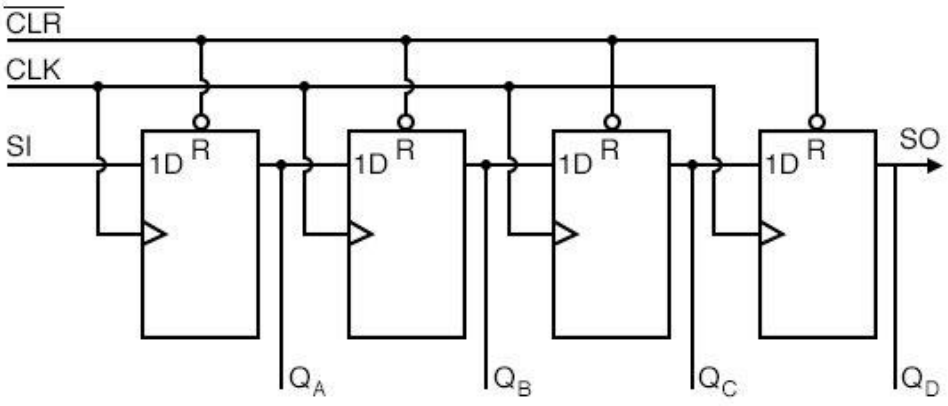
- This register accepts data serially i.e. One bit at a time on a single line
- Also it produces stored information on its output line in serial form



Serial-in Parallel-out shift register

In this type register, the data bits are entered serially. I.e. least significant bit is applied first in the flip-flop like the serial-in serial-out register. The difference is that the way in which data bits are taken out from the register. i. e outputs are available in parallel fashion.

Once the data are stored, each bit appears on its respective output line. All the bits are available simultaneously rather than on a bit by bit basis

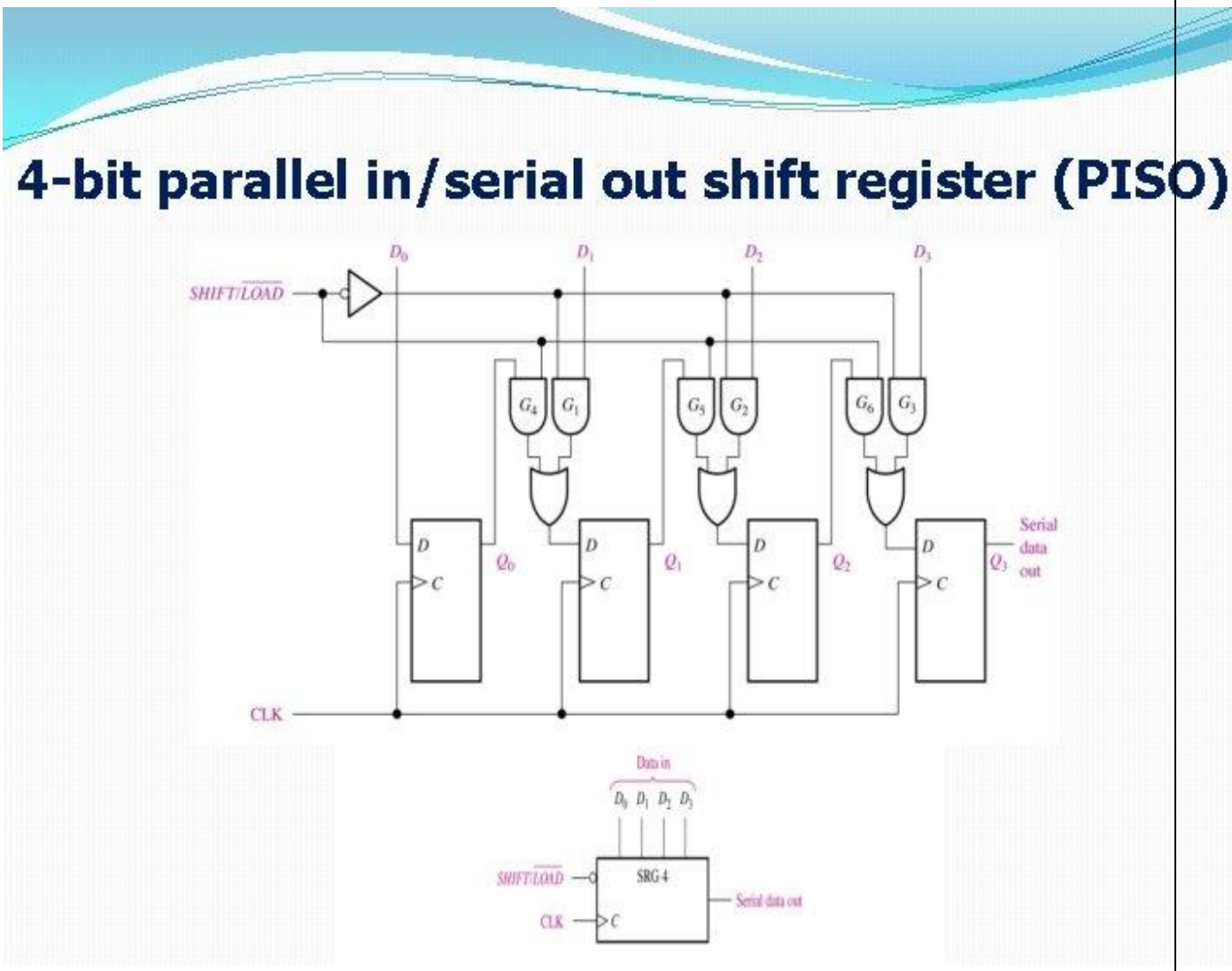


Serial-in/ Parallel-out shift register details

Parallel-in Serial-out shift register

Here the data bits are entered simultaneously through the parallel lines rather on a bit by bit basis

It generates serial output once the data is completely stored in the register



When SHIFT/LOAD=0, then the input data D0, D1, D2, D3 are loaded to the respective flip-flop through the gates G1, G2, G3

When SHIFT/LOAD=1, then the circuit acts like a shift register, which transfers previously loaded data to the next stage serially

Parallel-in Parallel-out shift register

In this type of register, the inputs are received in parallel and the outputs generated also in parallel manner