

# Outline no 5 DLD

## Ch # function of Combinational Logic

### Half adder :-

The half-adder accepts two binary digits on its input and produce two binary digit on its outputs, a sum bit and a carry bit.

Truth table

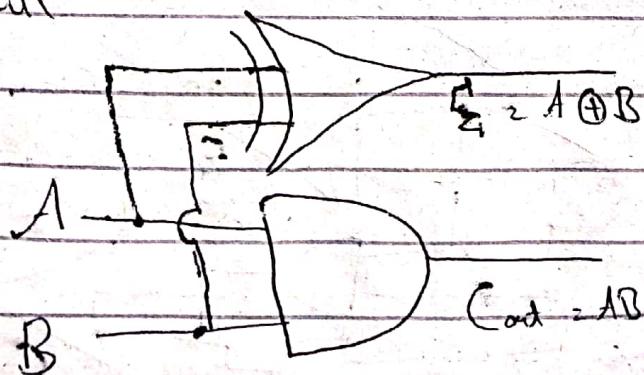
Input		Output		
A	B	Count	$\Sigma$	
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	0	0

Logic of Half-Adder.

$$\text{Count} = AB$$

$$\Sigma = A \oplus B$$

Logical Circuit



## FULL Adder :-

The full adder accepts ~~input~~ input bits and an input carry to generate a sum output and an output carry.

Truth table of full-adder

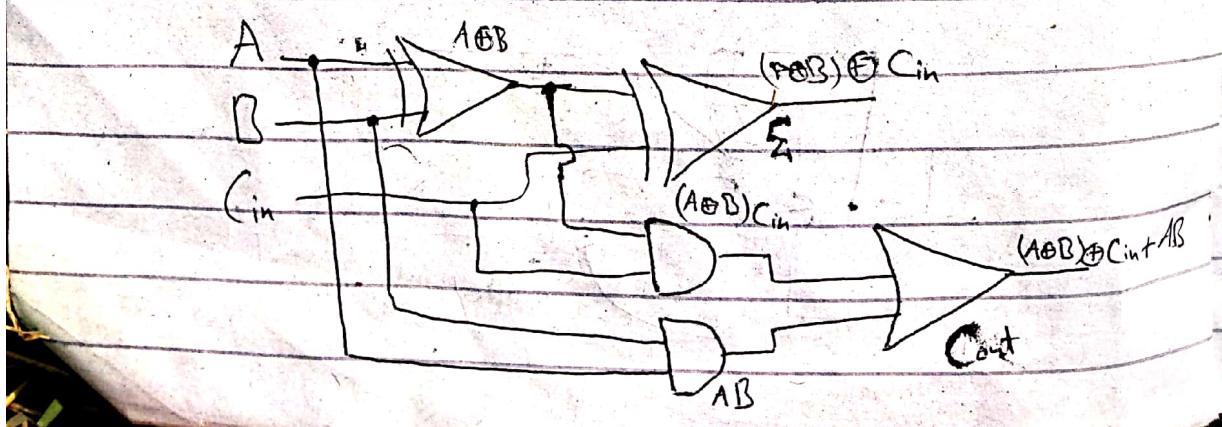
Input			Output	
A	B	Cin	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Full-Adder Logic

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

$$Cout = AB + (A \oplus B) C_{in}$$

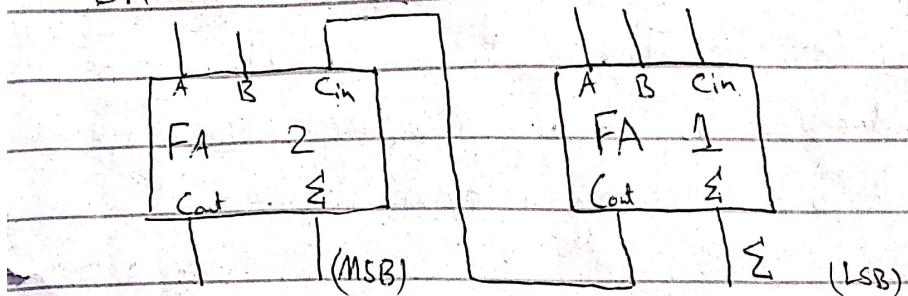
Logical Circuit



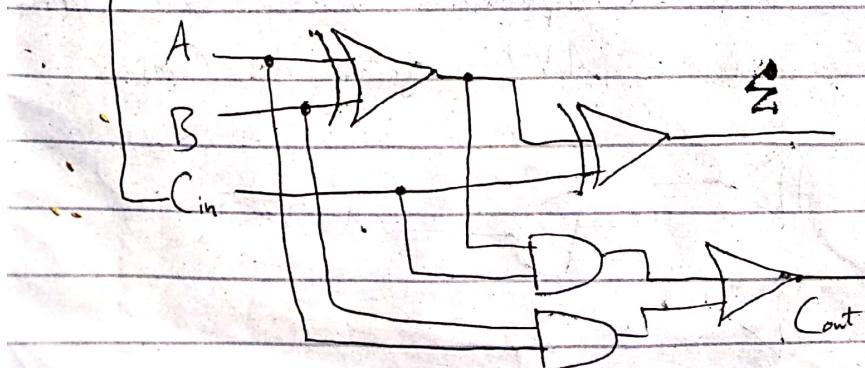
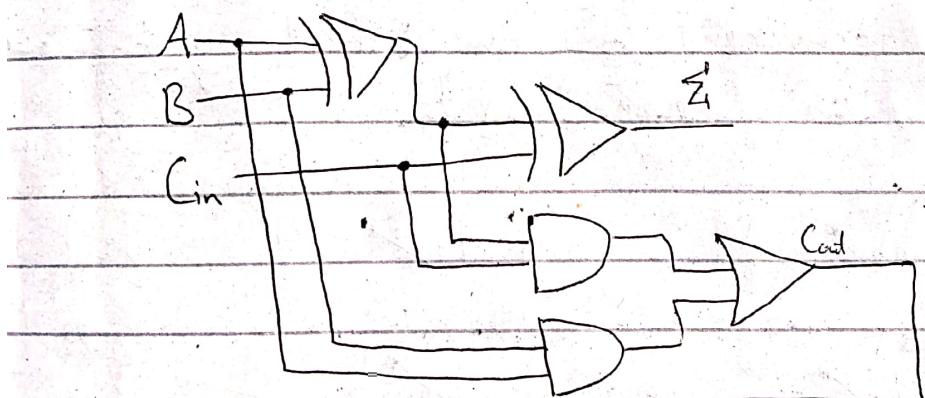
## Parallel binary adder

Two or more full adders are connected to form a parallel binary adder.

In parallel binary adder the carry out of first <sup>full-adder</sup> is become the carry in of the second Full-Adder and so on.



## Circuit Diagram



## Ripple Carry Adder :-

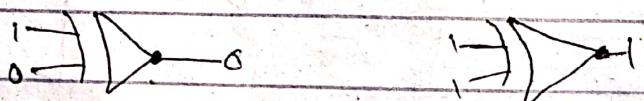
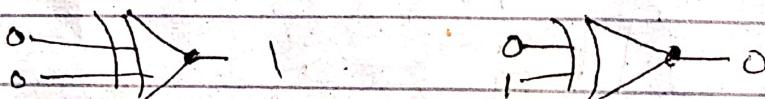
A ripple carry adder is one in which the carry output of each full-adder is connected to the carry input of the next higher-order stage (full-adder);

## Comparators :-

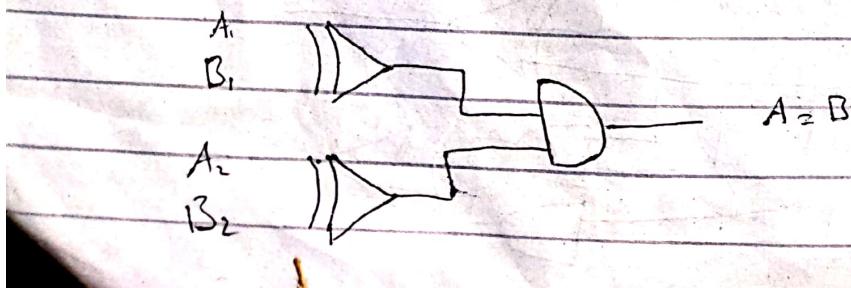
The basic function of Comparators is to compare the magnitude of two binary quantities.

## Equality :-

A exclusive-NOR gate can be used as a basic Comparator because it generates 0 on different inputs and 1 on same inputs.



Two Bit Comparator uses two XNOR gates and AND gate to Compute



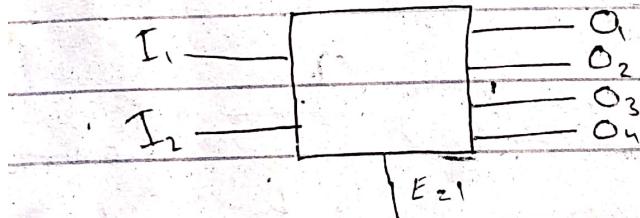
## Decoder:-

A decoder is a logic circuit that accepts a set of inputs that represents a binary number and activates that output which corresponds to the input binary number.

A decoder has ' $n$ ' inputs and an enable line and  $2^n$  output lines.

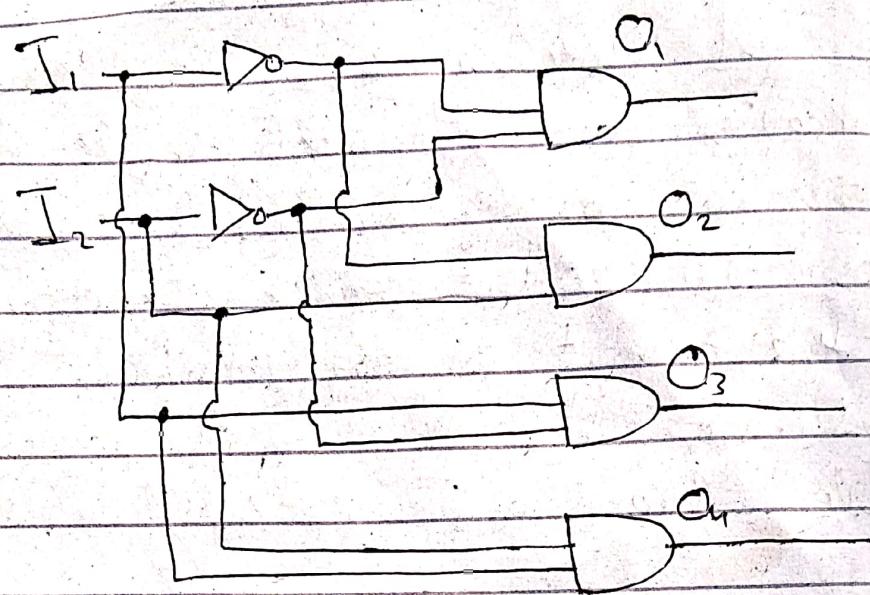
## $2 \times 4$ Decoder :-

it has 2 input lines and 4 output lines

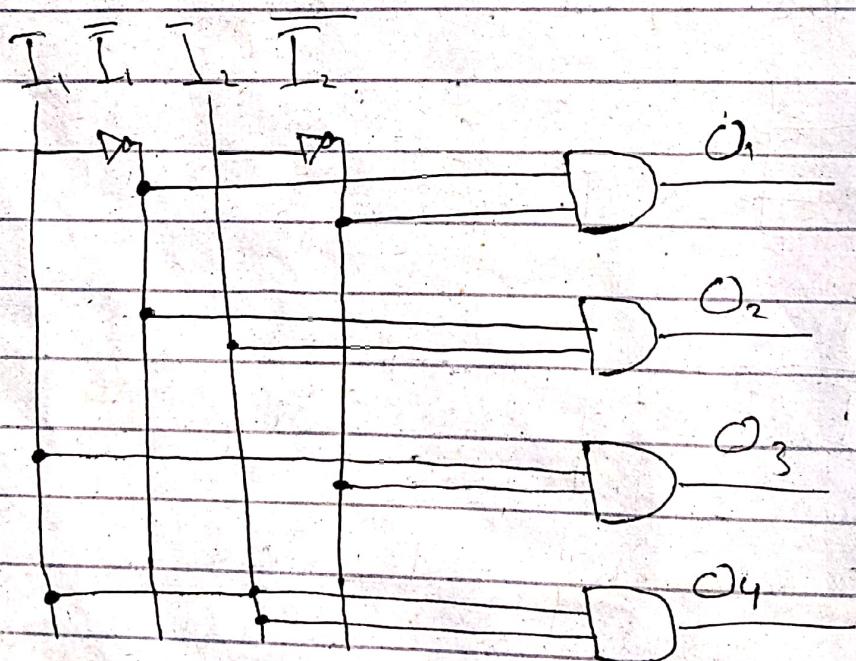


Input	Output
$I_1\ I_2$	$O_1\ O_2\ O_3\ O_4$
0 0	1 0 0 0
0 1	0 1 0 0
1 0	0 0 1 0
1 1	0 0 0 1

Logic diagram



OR



## 4 bit decoder :-

4 bit decoder has 4 input lines and 16 output lines. It requires sixteen decoding gates to generate decoded output. It is also called  $4 \times 16$  decoder.

Binary inputs	Decoding function
- - - - - 0 0 0 0 0 0 0 0	A <sub>0</sub>
- - - 0 0 0 0 - - 0 0 0 0	A <sub>1</sub>
- - 0 0 - 1 0 0 - - 0 0 - - 0 0	A <sub>2</sub>
- 0 - 0 - 0 - 0 - 0 - 0 - 0	A <sub>3</sub>

Table on Page # 224

## BCD to Decimal Decoded :-

BCD to deci.

Decodes Converts each BCD code to

To possible decimal digit indication. It is frequently referred as  $4 \times 10$  decoder.

The method of implementation is same for the 1 of 16 decoder but only first ten inputs and outputs are used and others are at don't care.

Truth Table

Decimal digit      Inputs

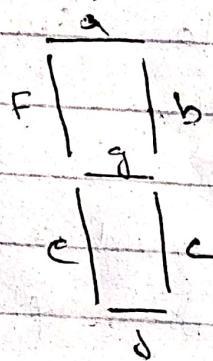
Decoding func.

#	$A_1$	$A_2$	$A_3$	$A_4$	
0	0	0	0	0	$\bar{A}_1, \bar{A}_2, \bar{A}_3, \bar{A}_4$
1	0	0	0	1	$\bar{A}_1, \bar{A}_2, \bar{A}_3, A_4$
2	0	0	1	0	$\bar{A}_1, \bar{A}_2, A_3, \bar{A}_4$
3	0	0	1	1	$\bar{A}_1, \bar{A}_2, A_3, A_4$
4	0	1	0	0	$\bar{A}_1, A_2, \bar{A}_3, \bar{A}_4$
5	0	1	0	1	$\bar{A}_1, A_2, \bar{A}_3, A_4$
6	0	1	1	0	$\bar{A}_1, A_2, A_3, \bar{A}_4$
7	0	1	1	1	$\bar{A}_1, A_2, A_3, A_4$
8	1	0	0	0	$A_1, \bar{A}_2, \bar{A}_3, \bar{A}_4$
9	1	0	0	1	$A_1, \bar{A}_2, \bar{A}_3, A_4$

## BCD to 7-Segment Decoder:-

7-Segment decoder accepts the BCD code on its input and generate seven outputs that is used to display decimal digits digitally.

BCD to



Truth table

Inputs $I_0\ I_1\ I_2\ I_3$	Decoding function							Output
	a	b	c	d	e	f	g	
0 0 0 0	1	1	1	1	1	1	1	0
0 0 0 1	0	1	1	0	0	0	0	1
0 0 1 0	1	1	0	1	1	0	1	2
0 0 1 1	1	1	1	1	0	0	1	3
0 1 0 0	0	1	1	1	0	0	1	4
0 1 0 1	1	0	1	1	0	1	1	5
0 1 1 0	1	0	1	1	1	1	1	6
0 1 1 1	1	1	1	1	0	0	0	7
1 0 0 0	1	1	1	1	1	1	1	8
1 0 0 1	1	1	1	0	0	1	1	9

## Encoders :-

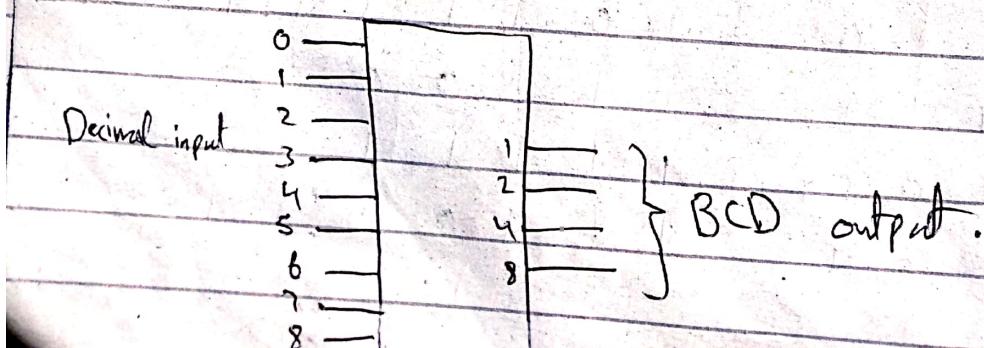
An encoder is a logic circuit that performs a 'reverse' function. An encoder accepts an input level on one of its inputs representing a digit, such as decimal or octal and convert it into coded output such as BCD or Binary.

Encoders can also be devised to encode various symbols or numbers to a Coded format is called and alphabetic character.

"The process of converting from familiar symbol or numbers to a coded format is called encoding."

## The Decimal to BCD Encoder.

This type of encoder has ten inputs (one of each decimal digit) and four outputs corresponding to BCD code.

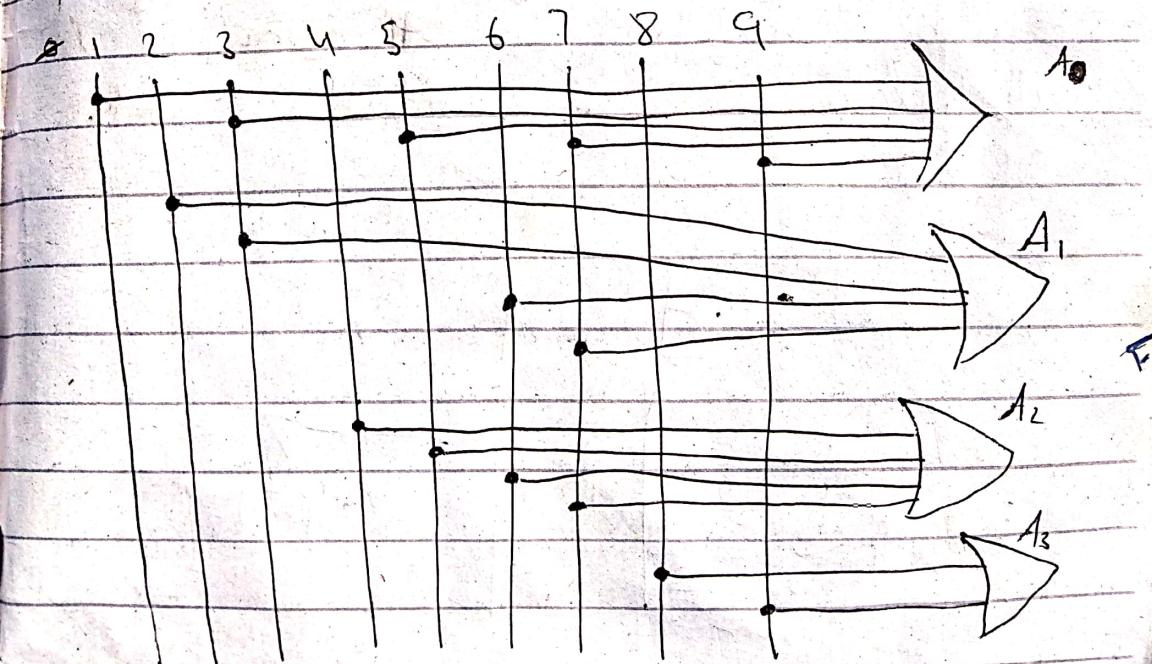


Decimal Digit

BCD Code

$A_3 \ A_2 \ A_1 \ A_0$

0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1



when 0 input mean  $A_0, A_1, A_2, A_3$  are 0  
so there is no need in circuit.

## Code Converters :-

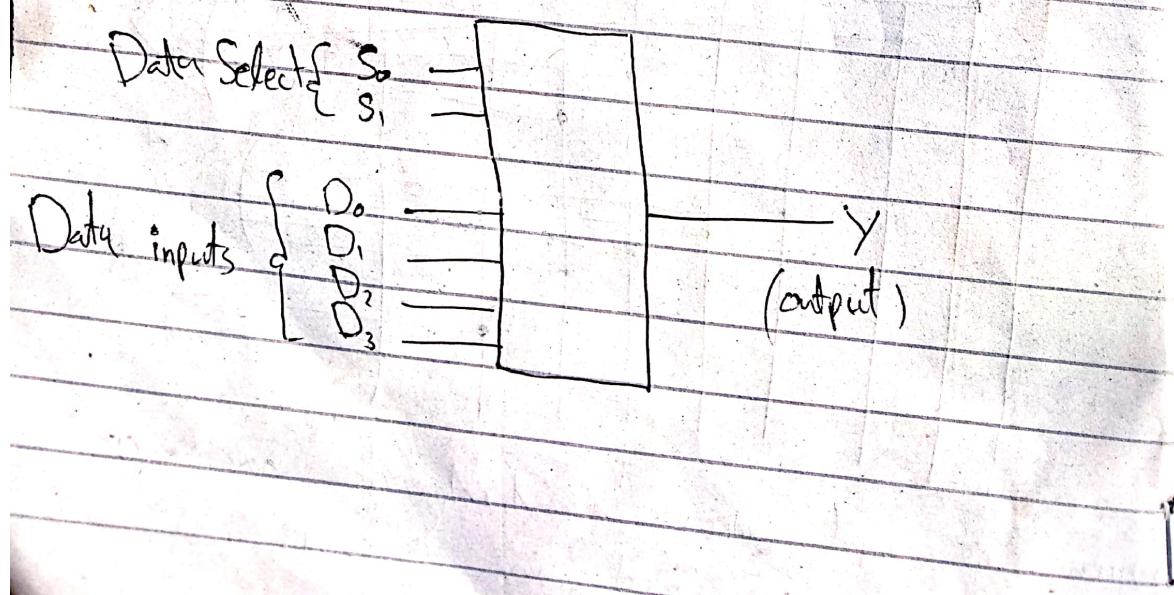
Code converters is  
Combinational logic circuit that ~~convert~~  
One code to other for example binary  
to gray code.

## Multiplexers :-

A multiplexer (MUX) is a device that allows digital information from several sources to be routed onto a single line for transmission to over that line to a common destination.

The basic multiplexer has several data-input lines and a single output line. It also has data-select inputs, which permit digital data on any one of the inputs to be switched to the output line.

Multiplexers are also called as data selector.

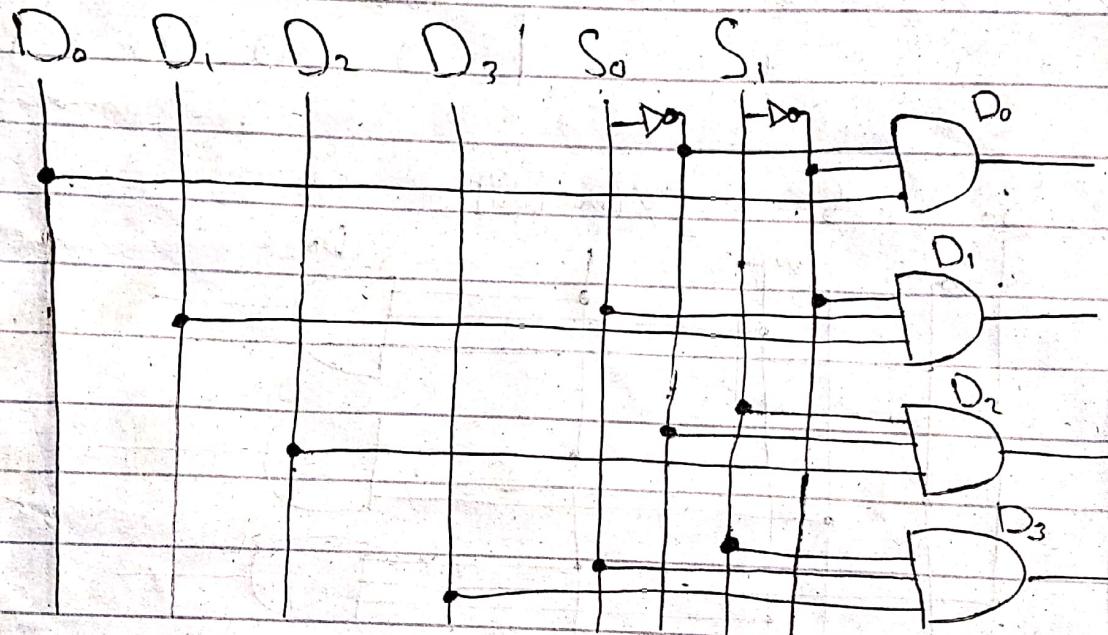


Data-Select Input

$S_1$	$S_0$
0	0
0	1
1	0
1	1

Input Selected

$D_0$   
 $D_1$   
 $D_2$   
 $D_3$



If  $2^n$  are input lines then  $n$  are the data-select lines.

### Demultiplexers :-

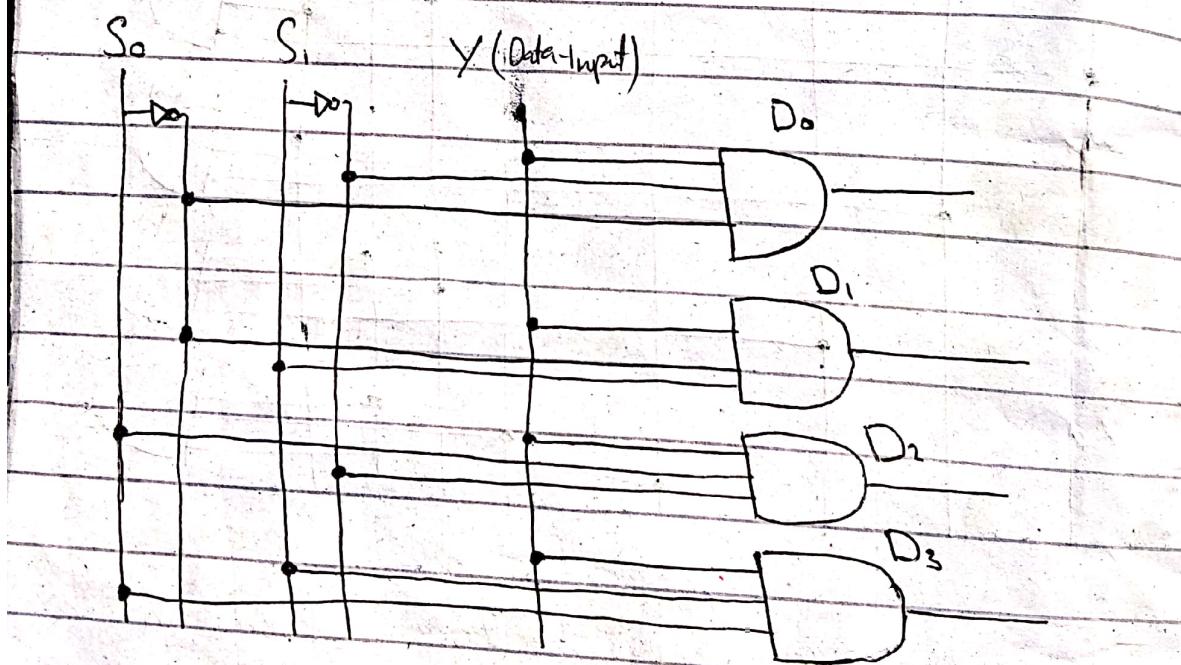
A Demultiplexer (DEMUX)

basically reverse the multiplexing function.

- It takes ~~an~~ digital information from 1 line and distributes it to a given number of output lines. For this reason the demultiplexers is also known as a

Data distributor. Decoders can also be used as demultiplexers.

Data-Select Input		Input Selected			
$S_0$	$S_1$	$D_0$	$D_1$	$D_2$	$D_3$
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

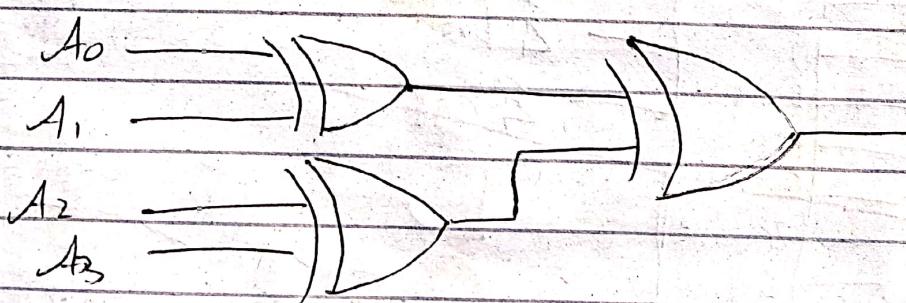
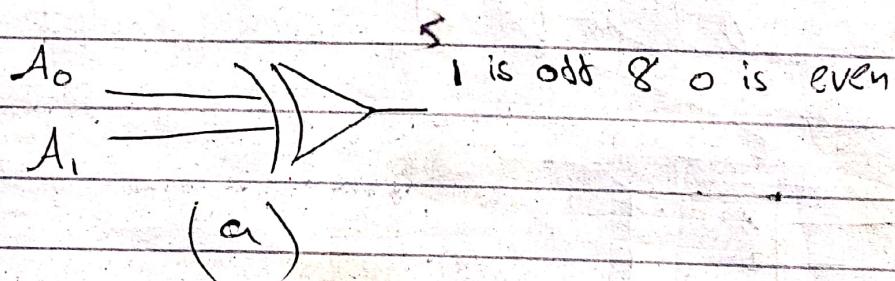


If  $2^n$  output lines in Data-Select inputs the  $2^n$  output lines.

## Parity bit :-

To check the ~~input~~ transmitted data is safe from ~~error~~ we sent a parity bit with data. A parity bit shows the number of 1 in transmitted data is even or odd.

If number of ~~1~~ is odd the parity bit is 1 and if number of 1 is even the parity bit is 0. The exclusive OR gate is used to check the number of 1 is even or odd in given input.

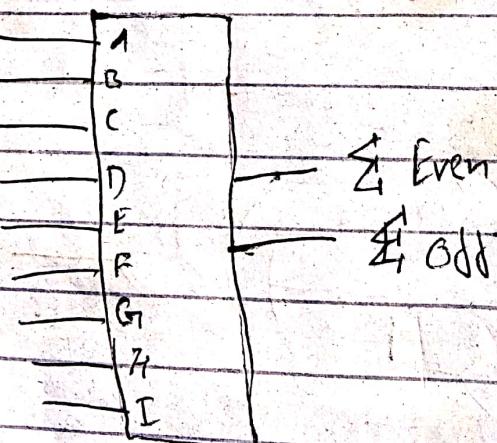


(b)  
if n number of lines as input then  $n-1$  numbers of exclusive OR gates are used to check the number of 1 is even or odd as in (pic B).

## Parity Checks :-

when this device is used as even parity checker, the number of input bits should always be even and when parity error occurs the  $\Sigma$  Even output goes low and the  $\Sigma$  odd output should always be off and the  $\Sigma$  odd output goes high.

when it is used as an odd parity checker the number of inputs bit should always be odd and when a parity error occurs the  $\Sigma$  odd outputs goes low and  $\Sigma$  even output goes high.



input  
0, 2, 4, 6, 8  
1, 3, 5, 7, 9

$\Sigma$  Even       $\Sigma$  Odd  
H                L  
L                H

## Parity Generator -

It is a device that generate parity bit at transmitter.

If used as even parity generator the parity bit taken at the  $\sum$  odd output because this output is a zero if there is an even number of inputs bits and it is a 1 if there is a 1.

When used as an odd parity generator the parity bit is taken at the  $\sum$  even output because it is a 0 when the number of input bits is odd.

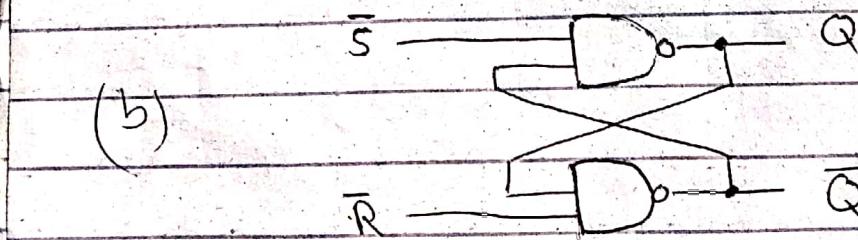
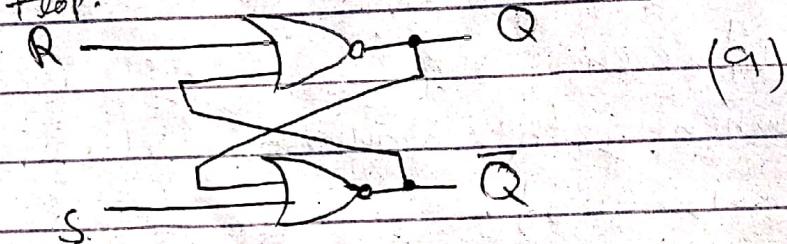
# Outline no # 6

## Latch :-

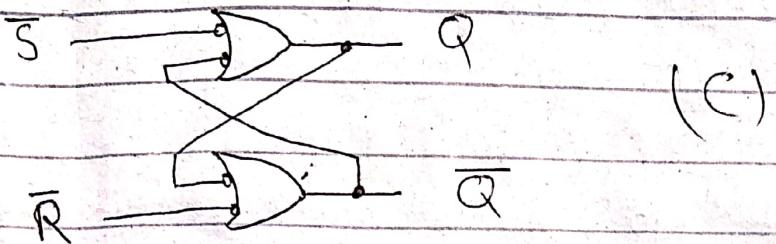
A latch is a type of bistable logic device or multivibrator. A latch is also defined as temporary storage device that stores state information.

## S-R Latch:-

S-R Latch is stand for set-Reset latch which is formed with two cross-coupled NOR gates or NAND gates. It is also known as basic flip flop.



Negative OR gate is equivalent to NAND gate So we can also figure SR latch as



when  $Q$  is 1 and  $\bar{Q}$  is 0  
then latch is **Set**.

when  $Q$  is 0 and  $\bar{Q}$  is 1  
then latch is **RESET**.

Table for pic (a)				NOR table
S	R	Q	$\bar{Q}$	a b A+B $\bar{A}+\bar{B}$
1	0	1	0	(Set) 0 0 0 1
0	0	1	0	0 1 1 0
0	1	0	1	(Reset) 1 0 1 0
0	0	0	1	1 1 1 0
1	1	0	0	(Forbidden) or invalid condition

when input is 0,0 then there is no change in output state,  
output state is same as previous.  
means it stores previous state.

# Important Topics

1.1, 1.2, 1.3, 1M

2.1 - 2.11

3.1 - 3.7

4.1 - 4.10

5.1 - 5.5

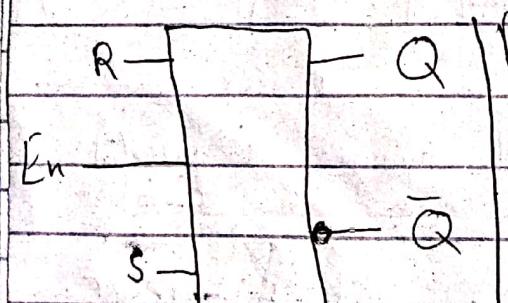
6.1, 6.2, 6.3, 6.4, 6.5, 6.6, 6.7, 6.8, 6.9, 6.10

7.1, 7.2, 7.3

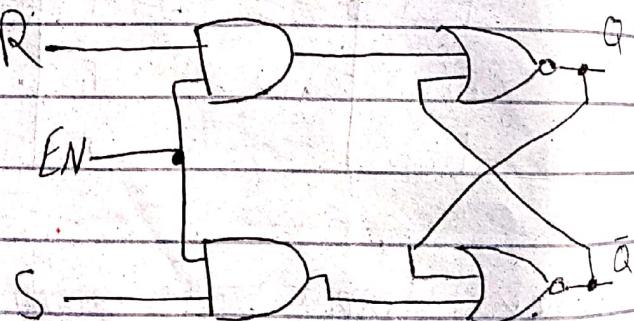
## Gated S-R Latch:-

A gated S-R latch requires an enable input, EN. It is also called as S-R flip flop. It can be constructed from NAND gate and NOR gate.

### Using NOR Gate



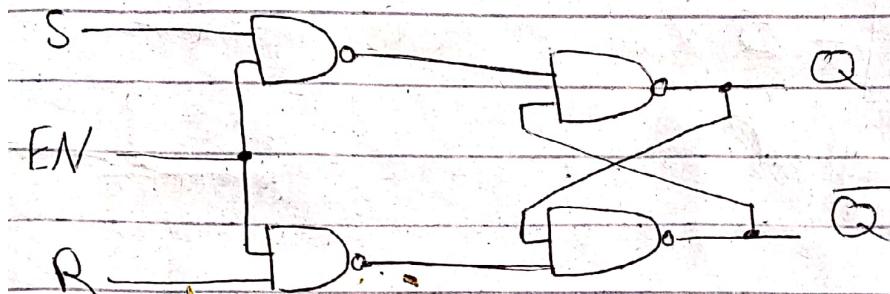
Logic Symbol



Logic Diagram

EN	S	R	$Q_{n+1}$
0	X	X	No change
1	0	0	No change
1	0	1	$Q = 0$ ; Reset state
1	1	0	$Q = 1$ ; SET state
1	1	1	Invalid

Using NAND gate



Logic Diagram



Logic Symbol

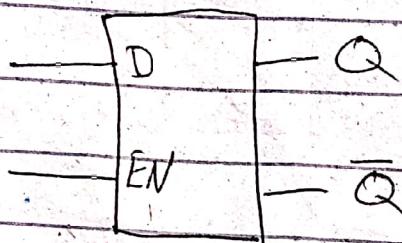
~~Both has Same Output Table~~

Both has Same function Table

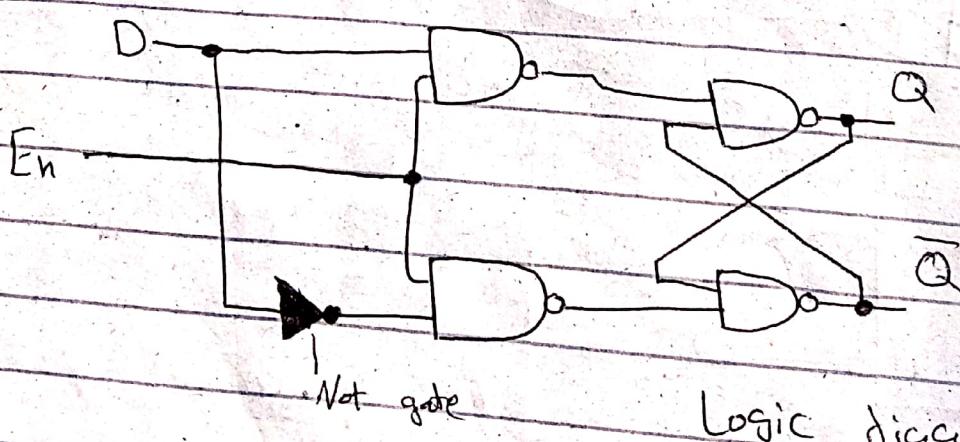


## The Gated D Latch:

An other type of gated latch is D latch.  
~~It is also called D flip flop.~~  
It is different from S-R latch because it has only one input in addition to EN.



Logic Symbol



Logic diagram

EN	D	Q <sub>n+1</sub>
0	X	No change
1	0	$Q = 0$ ; RESET state
1	1	$Q = 1$ ; SET state

function table

## Flip flop:-

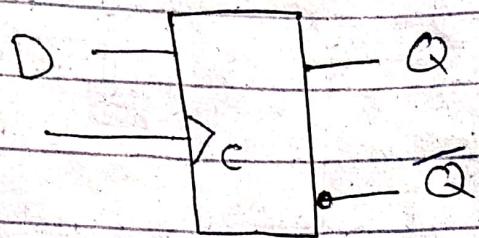
flip flop are synchronous bistable device also known as bistable multivibrator. In case, the term synchronous means that the output changes state only at specified point on the triggering input called the clock (clk).

An edge-triggered flip flop changes state either at the positive edge (rising edge) or at the negative edge (falling edge) of the clock pulse.

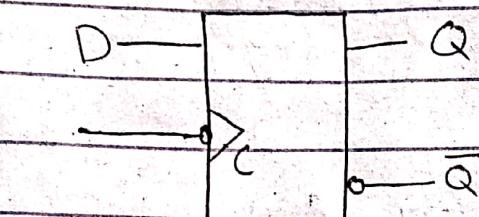
D flip flop: D input of D flip-flop is synchronous  
~~D flip flop is same as~~  
~~D latch only the difference is that input~~  
are transferred to the flip-flop's output only on the triggering edge of the clock pulse.

When D is high the Q output goes high on triggering edge of the clock pulse, and the flip-flop is SET.

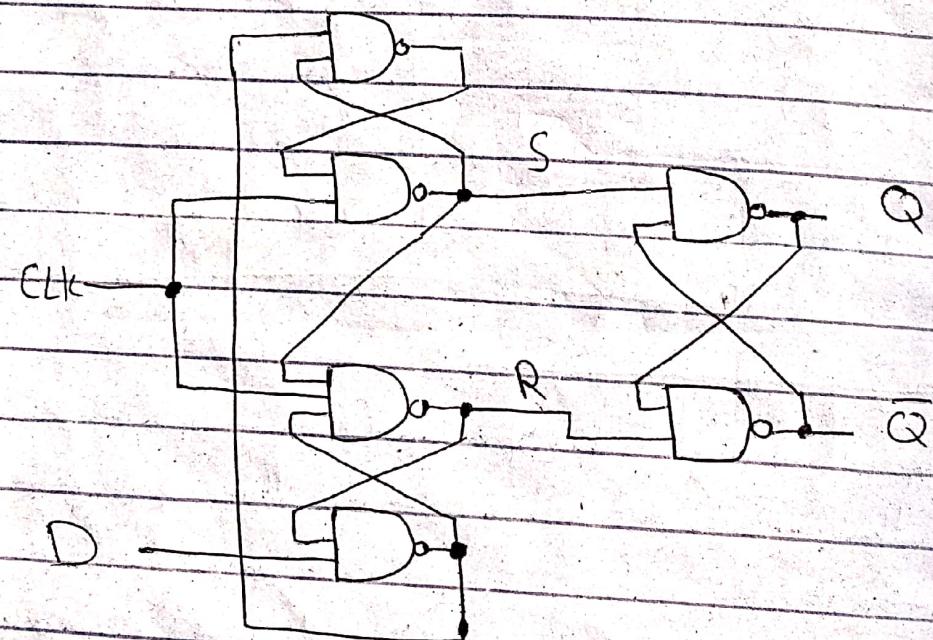
When D is low the Q output goes low on triggering edge of clock pulse, and flip-flop is RESET.



positive edge-triggered symbol



Negative edge-triggered Symbol



Circuit diagram

Inputs		Output
D	CLK	Q    Q̄
0	↑	0    1    RESET
1	↑	1    0    SET

Truth table for positive edge trig

D flip-flop. The operation and truth table for negative edge-triggered flip-flop are same as positive but on table the clock sign  $\downarrow$  falling edge use.

## The J-K Flip-flop:

The J and K inputs of the J-K flip-flop are synchronous inputs because data on these inputs are transferred to the flip-flop's output only on the triggering ~~edge~~ <sup>edge</sup> of the clock pulse.

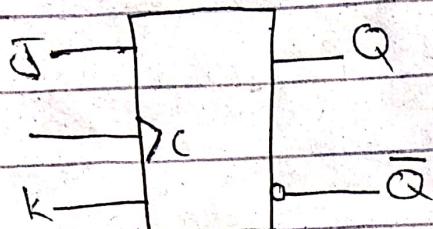
When J is high and K is low, the Q goes high on triggering edge of the clock pulse, and flip flop is Set.

When J is low and K is High the Q output goes low on triggering the edge of the clock pulse and flip-flop is Reset.

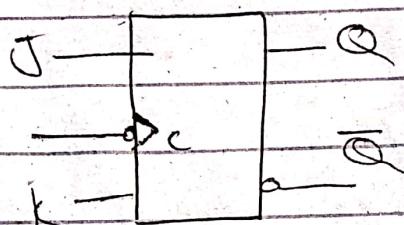
When both J and K are low, the output does not change from its prior state.

When both J and K are High, the ~~output~~ flip-flop changes state. This is called toggle mode.

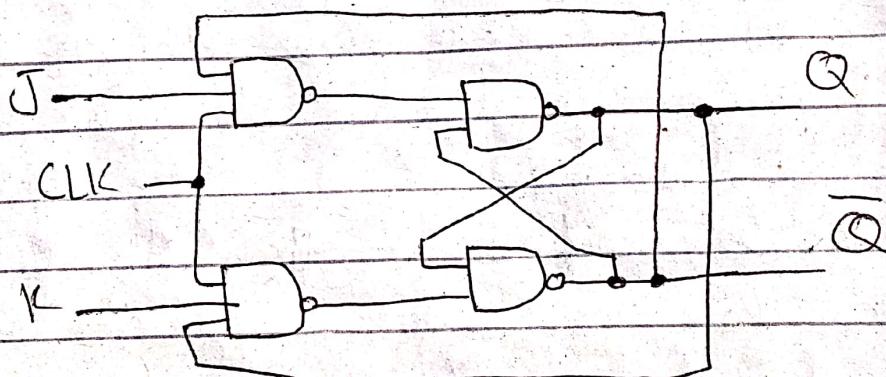
A J-K flip-flop is refinement of the S-R flip-flop to solve the problem (indeterminate state) when both input are '1'.



Symbol for positive edge triggered



Symbol for Negative edge triggered.



Circuit Diagram

J	K	CLK	Q	$\bar{Q}$	Actions
0	0	↑	$Q_0$	$\bar{Q}_0$	No change
0	1	↑	0	1	Reset
1	0	↑	1	0	Set
1	1	↑	$\bar{Q}_0$	$Q_0$	Toggle

Truth table for positive edge-triggered. Negation is same as positive only difference is

that the clock ↓ sign (falling edge) is used in table.

## Difference between latch & flip-flop

### latch

- 1) It is level sensitive device
- 2) Built from logic gates
- 3) Based on enable Signal
- 4) ~~Asynchronous~~

### flip-flop

- It is edge sensitive device.

- Built from latches.
- Based on clocked signal
- Synchronous