

Combinational Circuits

Design Procedure

1. The problem is stated.
2. The number of available inputs and required output variables is determined.
3. The input and output variables are assigned letter symbols.
4. The truth table that defines the required relationships between inputs and outputs is derived.
5. The simplified Boolean functions for each output is obtained.
6. The logic diagram is drawn.

ADDERS

The single addition possible operations consists of 4 namely, $0+0=0$, $0+1=1$, $1+0=1$, and $1+1=10$.

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

Q; Half adder.

- > two binary inputs and two binary outputs.
- > It adds two inputs and produce the sum(s) and the carry(c) bits.

Inputs outputs.

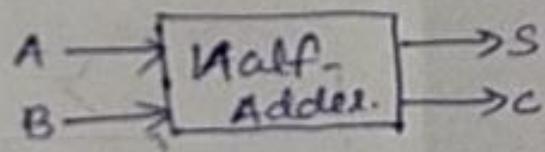
A	B	S	C
---	---	---	---

0	0	0	0
---	---	---	---

0	1	1	0
---	---	---	---

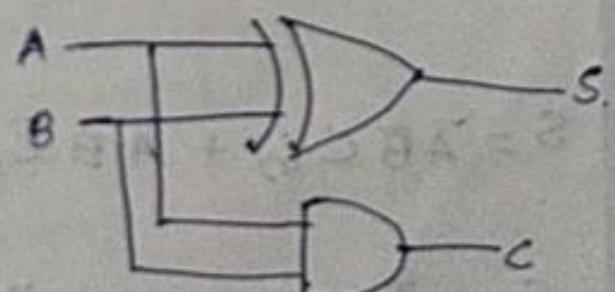
1	0	1	0
---	---	---	---

1	1	0	1
---	---	---	---



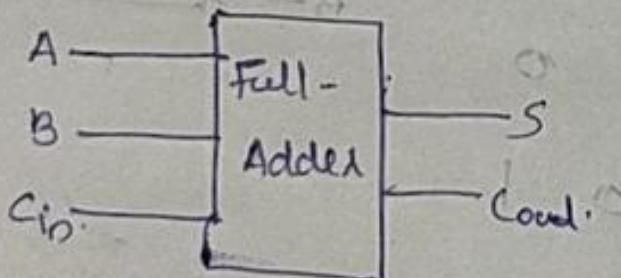
$$S = A\bar{B} + \bar{A}B$$

$$= A \oplus B$$



Full adder

A	B	C _{in}	S	C _{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



$$S = \overline{A}\overline{B}C_{in} + \overline{A}B\overline{C}_{in} + A\overline{B}C_{in} + ABC_{in}$$

$$C_{out} = \overline{A}\overline{B}C_{in} + A\overline{B}C_{in} + AB\overline{C}_{in} + ABC_{in}$$

$$S = A\bar{B}C_{in} + \bar{A}B\bar{C}_{in} + A\bar{B}\bar{C}_{int} + AB\bar{C}_{int}$$

$$= \overline{C_{in}} [\bar{A}B + A\bar{B}] + [\bar{A}\bar{B} + AB] C_{in}$$

$\underbrace{\quad}_{XOR}$

$\underbrace{\quad}_{XWOR}$

$\underbrace{\quad}_{P}$

xor-orb
complemented
xnor.

$$P = A \oplus B$$

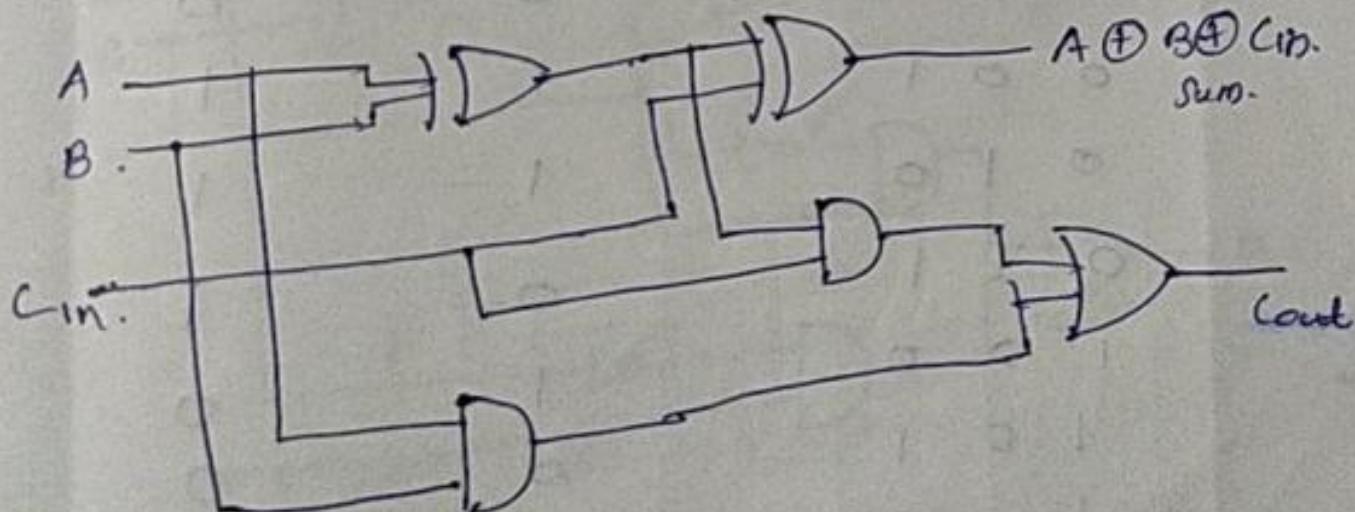
$$= P\overline{C_{in}} + \overline{P}C_{in}$$

$$= P \oplus C_{in}$$

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

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

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

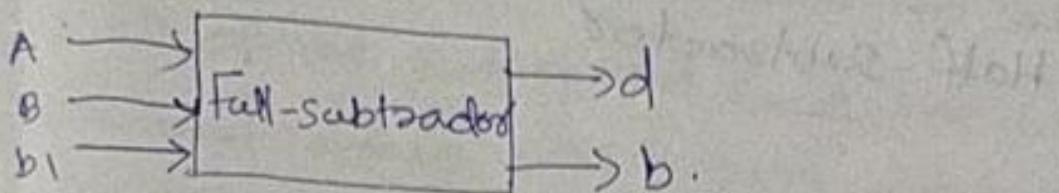


Subtractors

The half-subtractor can be used only for LSB subtraction. If there is a borrow during the subtraction of the LSBs, it affects the subtraction in the next higher column: the subtracted bit is subtracted from the minued bit, considering the borrow from that column used for the subtraction in the preceding column. Such a subtraction is performed by a full-subtractor. It subtracts one bit (b) from another bit (a).
 $(A + B) - (A - B) = 2B$

Inputs Difference Borrow.

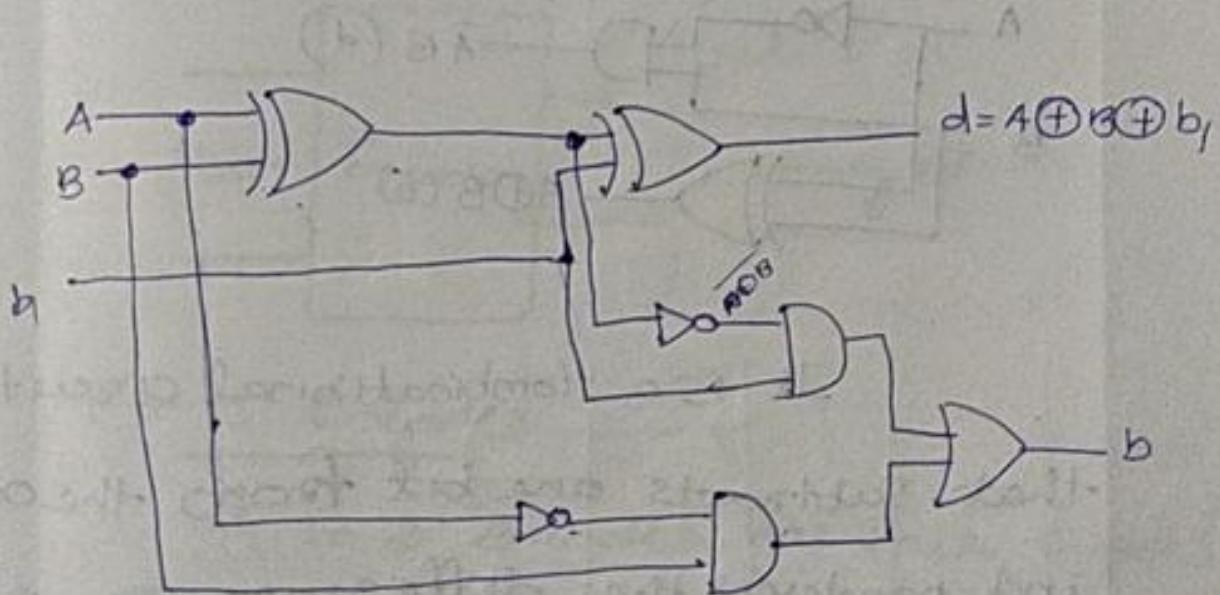
A	B	b_1	d	b
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	0	1
1	0	1	1	0
1	1	0	0	0
1	1	1	1	1



block diagram.

$$\begin{aligned}
 d &= \bar{A} \bar{B} b_1 + \bar{A} B \bar{b}_1 + A \bar{B} \bar{b}_1 + A B b_1 \\
 &= b_1 (A \bar{B} + \bar{A} \bar{B}) + \bar{b}_1 (A \bar{B} + A B) \\
 &= b_1 (\overline{A \oplus B}) + \bar{b}_1 (A \oplus B) = \\
 &= A \oplus B \oplus b_1
 \end{aligned}$$

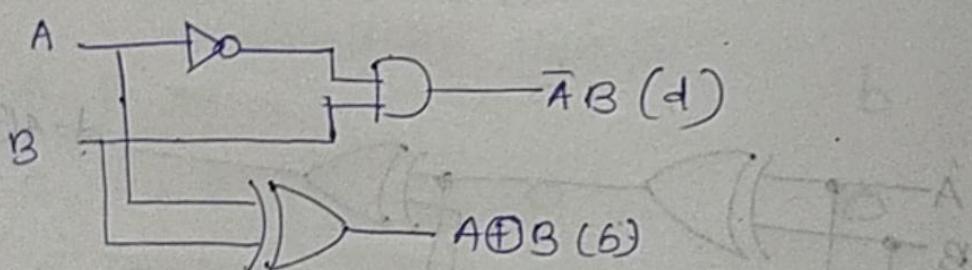
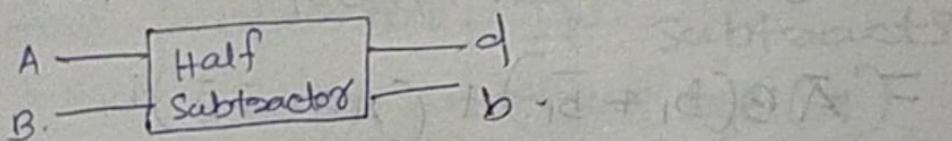
$$\begin{aligned}
 b &= \bar{A} \bar{B} b_1 + \bar{A} B \bar{b}_1 + A \bar{B} b_1 + A B \bar{b}_1 \\
 &= \bar{A} B (b_1 + \bar{b}_1) + (A \bar{B} + A B) \bar{b}_1 \\
 &= \bar{A} B + (\overline{A \oplus B}) b_1
 \end{aligned}$$



Half-Subtractor

Truth-table.

A	B	diff	borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0



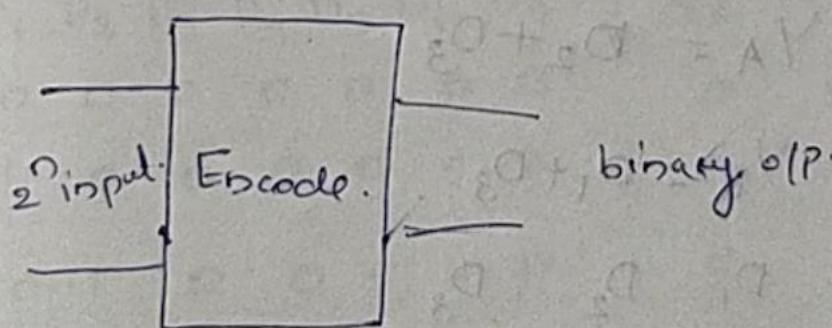
It is a combinational circuit that subtracts one bit from the other and produces the difference. It also has an output to specify if a 1 has

bits borrowed. It is used to subtract the LSB of the subtrahend from the LSB of the minuend when one binary number is subtracted from the other.

Encoder

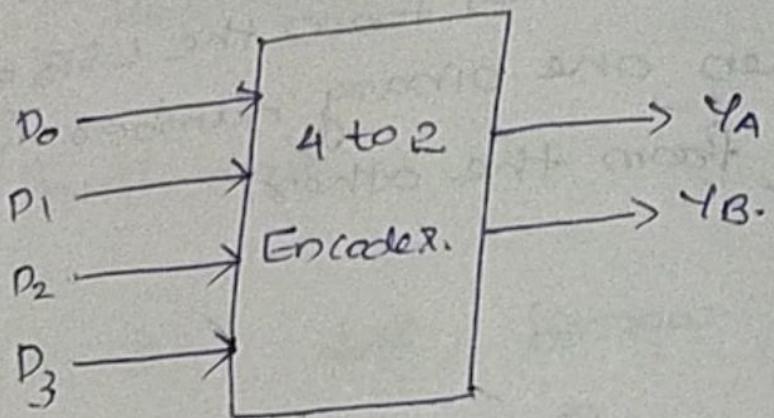
An encoder is a combinational circuit that performs the reverse operation of Decoder. It has m inputs of 2^n input lines and 'n' output lines. It will produce a binary code equivalent to the input, which is active high.

∴ the encoder encodes 2^n input lines with 'n' bits.



4 to 2 Encoder

4 to 2 Encoder has four inputs D_0, D_1, D_2, D_3 and two outputs Y_A and Y_B .

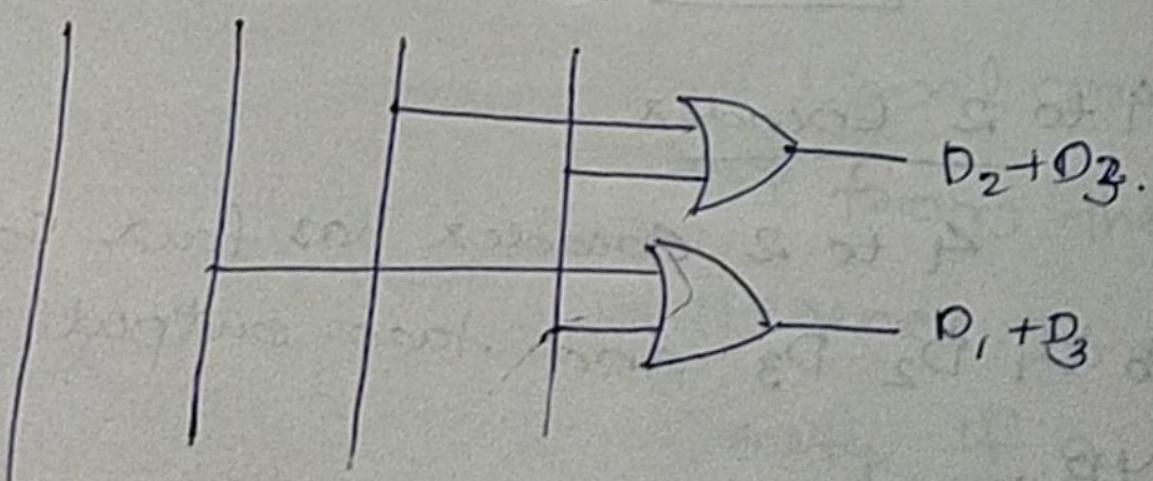
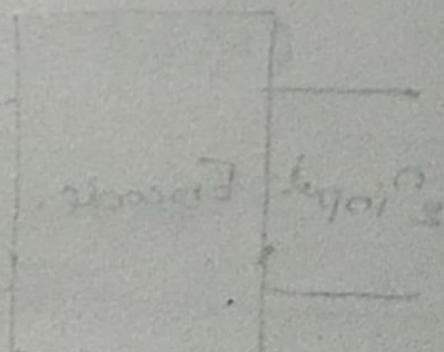


D_0	D_1	D_2	D_3	Y_A	Y_B
0	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1

$$Y_A = D_2 + D_3$$

$$Y_B = D_1 + D_3$$

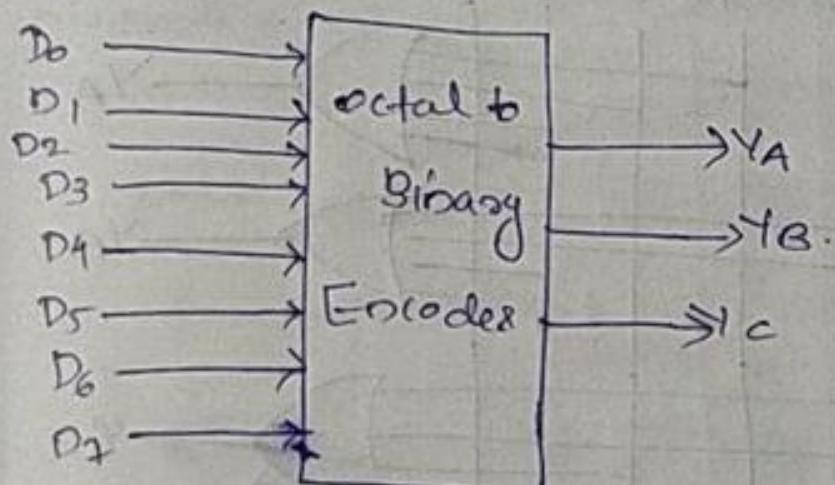
$D_0 \quad D_1 \quad D_2 \quad D_3$.



Octal to binary encoder.

Octal to binary encoder has eight input, D_0 to D_7 . and three outputs Y_A Y_B Y_C . Octal to binary encoder is nothing but 8 to 3 encoder.

The block diagram of octal to binary encoder is below;



D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	Y_A	Y_B	Y_C
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
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	1	0
0	0	0	0	0	0	0	1	1	0	1

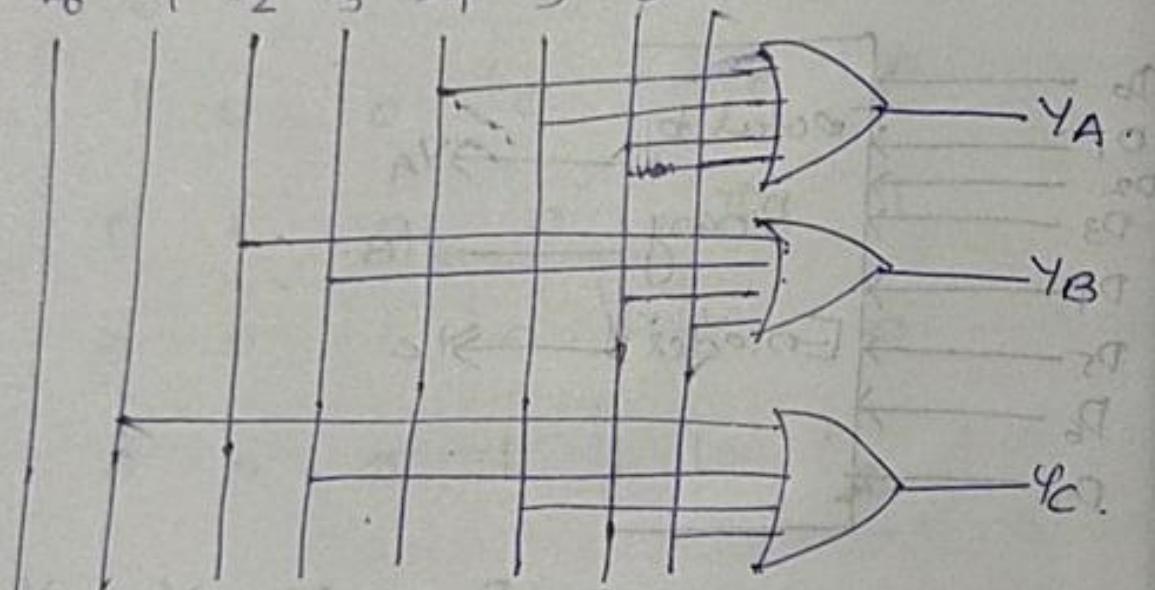
$$Y_A = D_4 + D_5 + D_6 + D_7$$

$$Y_B = D_2 + D_3 + D_6 + D_7$$

$$Y_C = D_1 + D_3 + D_5 + D_7$$

we can implement the above Boolean functions by using four input OR gates.

D₀ D₁ D₂ D₃ D₄ D₅ D₆ D₇.



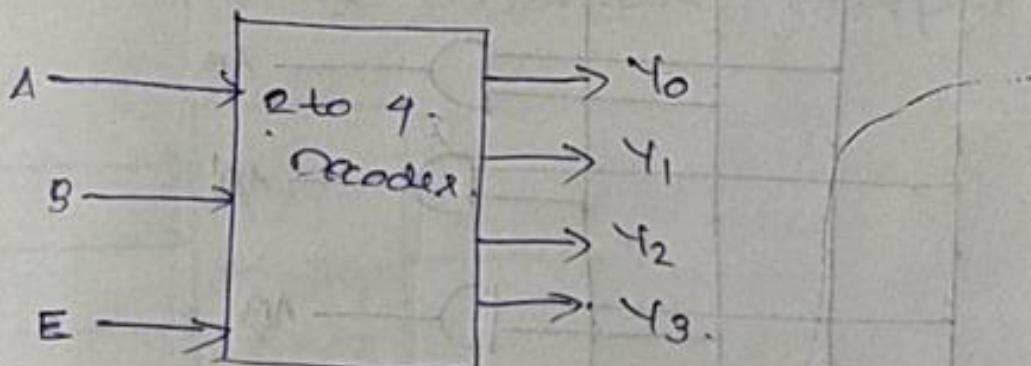
Decoder.

Decoder is a combinational circuit that has 'n' input lines and maximum of 2^n output lines. One of these outputs will be active high based on the combination of inputs present, when the decoder is enabled. That means

decoder detects a particular code. The output of the decoder are nothing but the initial terms of 'n' input variables like., when it's enabled.

2 to 4 decoder.

Let 2 to 4 decoder has two inputs A, B and four outputs y_0, y_1, y_2, y_3 . The block diagram of 2 to 4 decoder is shown below.



One of these four outputs will be '1' for each combination of inputs when enable, E is '1'.

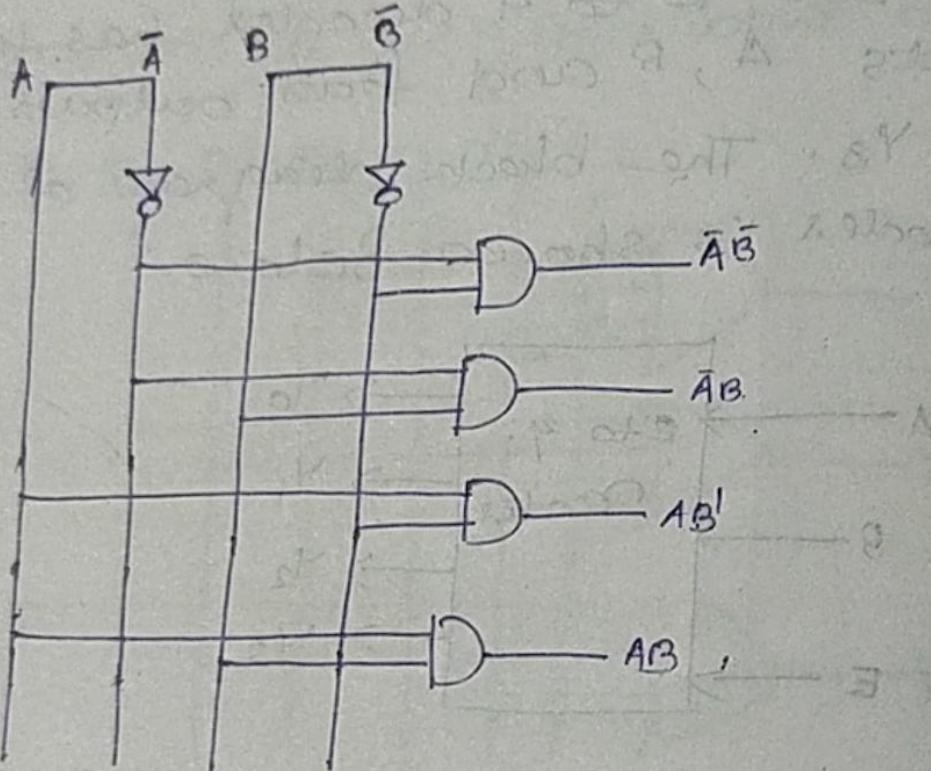
Enable	Inputs	Outputs
E	A B	$y_0 \quad y_1 \quad y_2 \quad y_3$
0	X X	0 0 0 0
1	0 0	1 0 0 0
1	0 1	0 1 0 0
1	1 0	0 0 1 0
1	1 1	0 0 0 1

$$Y_0 = \bar{A}B$$

$$Y_1 = \bar{A}B$$

$$Y_2 = A\bar{B}'$$

$$Y_3 = AB$$



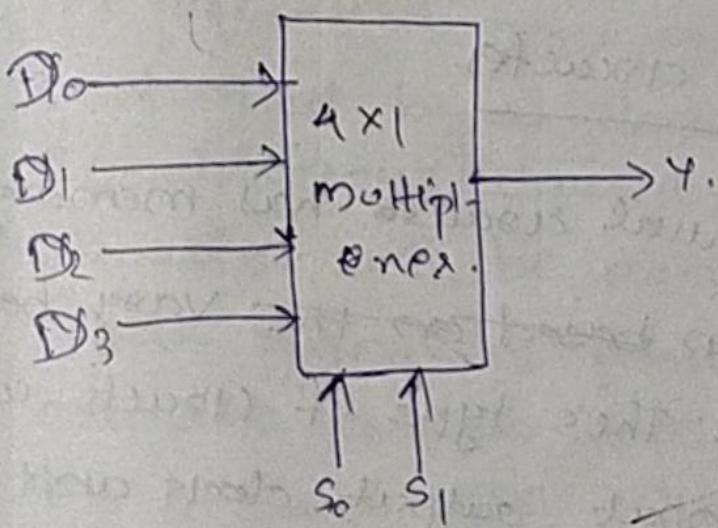
Multiplexer

It is a combinational circuit that has maximum of 2ⁿ data inputs, 'n' Selection lines and single output line. One of these data inputs will be connected to the output based on the values of selection lines.

Since there are 'n' selection lines, there will be 2^n possible combinations of zeros and ones. So, each combination will select only one data input. Multiplexer is also called mux.

4x1 Multiplexer.

4x1 multiplexer has four data inputs D_0, D_1, D_2, D_3 , two selection lines S_1 & S_0 and one output Y .



One of these 4 inputs will be connected to the output based on the combination of inputs present at these two selection lines.

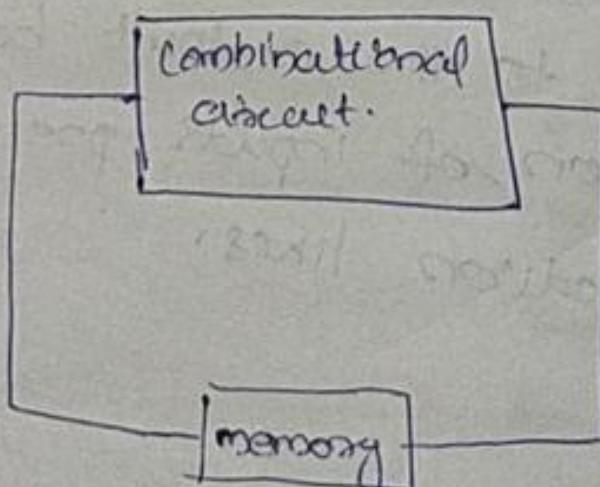
Selection Lines		Output
S_1	S_0	y
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

$$y = D_0 S_1' S_0' + S_1' S_0 D_1 + S_1 S_0' D_2 + S_1 S_0 D_3$$

Module-2

Sequential Circuits

Sequential circuits have memory so output can based on the vary based on input. This type of circuits uses previous input, outputs, clocks and a memory element.



Memory unit is required to store the past history of the input variable in Sequential circuits.

Sequential circuits are slower than combinational circuits.

Sequential circuits may be classified as Synchronous Sequential circuits and as asynchronous Sequential circuits depending on the timing of their signals.

The sequential circuits which are controlled by a clock are called Synchronous Sequential circuits. The circuits will be active only when clock signal is present. The sequential circuits which are not controlled by a clock are called asynchronous Sequential circuits. i.e., the sequential circuits in which events can take place any time the inputs are applied are called asynchronous Sequential circuits. Periodically, recurring pulse is called a clock. It is generated by a pulse generator.

FlipFlop

A flip flop is an electronic circuit with two stable states that can be used to store binary data. The stored data can be changed by applying varying inputs. Flip-flops and latches are fundamental building blocks of digital electronics. Systems used in computers, communications, and many other types of systems. Flip-flops and latches are used as data storage elements. It is the basic storage element in sequential logic.

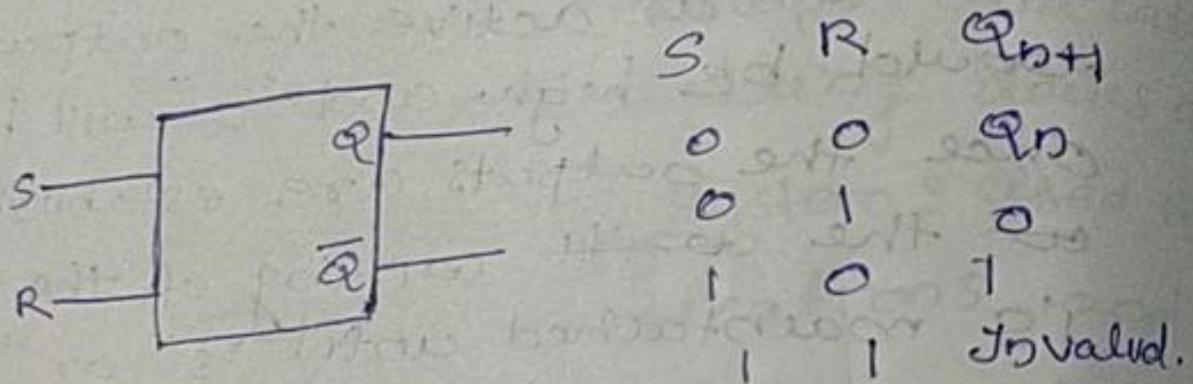
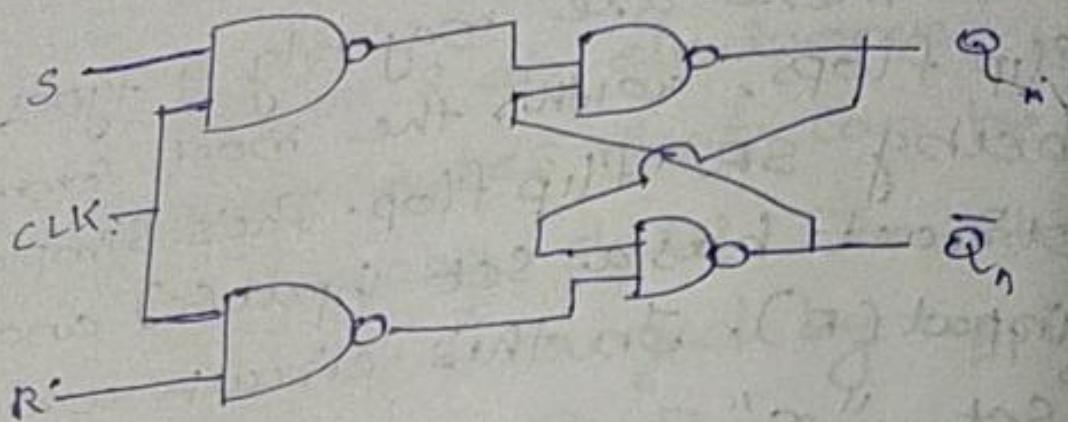
The basic difference between a latch and a flipflop is a gating or clocking mechanism.

SR Flip Flop.

These are majorly 4 types of flip flops, with the most common one being SR flip flop. This simple flip flop circuit has a set input (S) and a reset input (R). In this circuit when you set "S" as active the output "Q" would be high and "Q̄" will be low. Once the outputs are established, the ~~sett~~ coupling of the circuit is maintained until "S" or "R" go high, or power is turned off. As shown above, it is the simplest and easiest to understand. The two outputs, as shown above, are the inverse of each other.

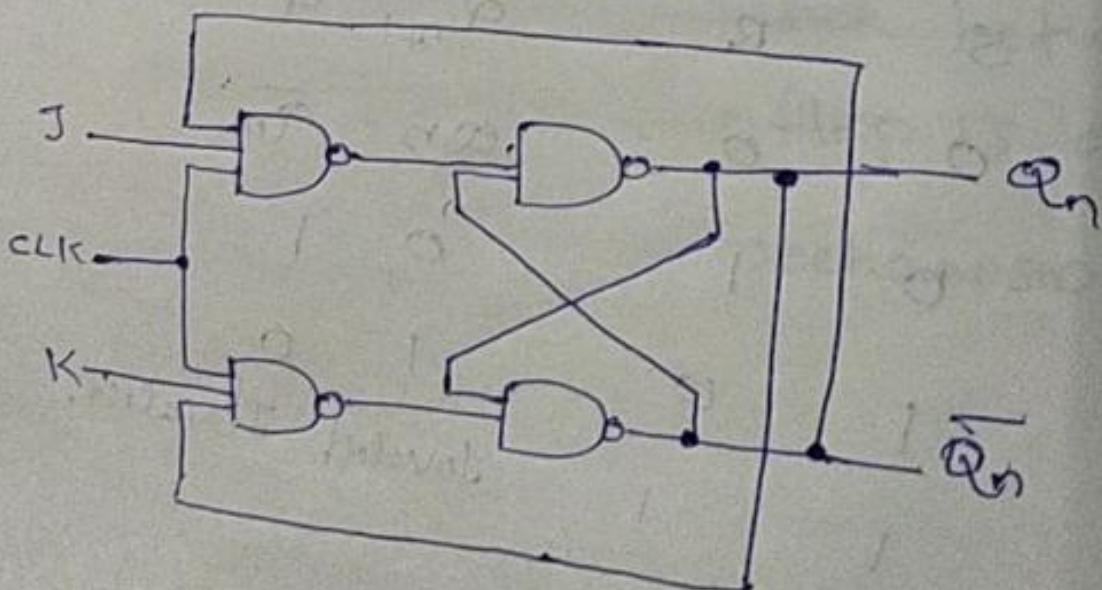
Truth table.

S	R	Q_{n+1}	\bar{Q}_n
0	0	Q _n	\bar{Q}_n
0	1	0	1
1	0	1	0
1	1	Invalid	Invalid



JK flip-flop

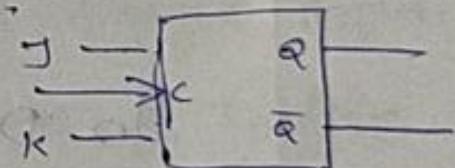
Due to the undefined state in the SR flip flop, another flip flop is required in electronics. The JK flip flop is an improvement on the SR flip flop because $S=R=1$ is not a problem.



The input conditions of $J=K=1$, gives an output inverting the output state. However the outputs are the same unless one tests the circuit practically.

In simple words, if J and K data input are different (i.e. high & low), then the output Q takes the value of J at the next clock edge. If J and K are both low, then no change occurs. If J and K are both high at the clock edge then the output will be toggle from one state to the other. JK flipflop can function as Set or Reset flipflop.

Truth table.



$J = K = 1, Q_n \rightarrow Q_{n+1}$

$0 \quad 0 \quad \rightarrow \quad Q_n$

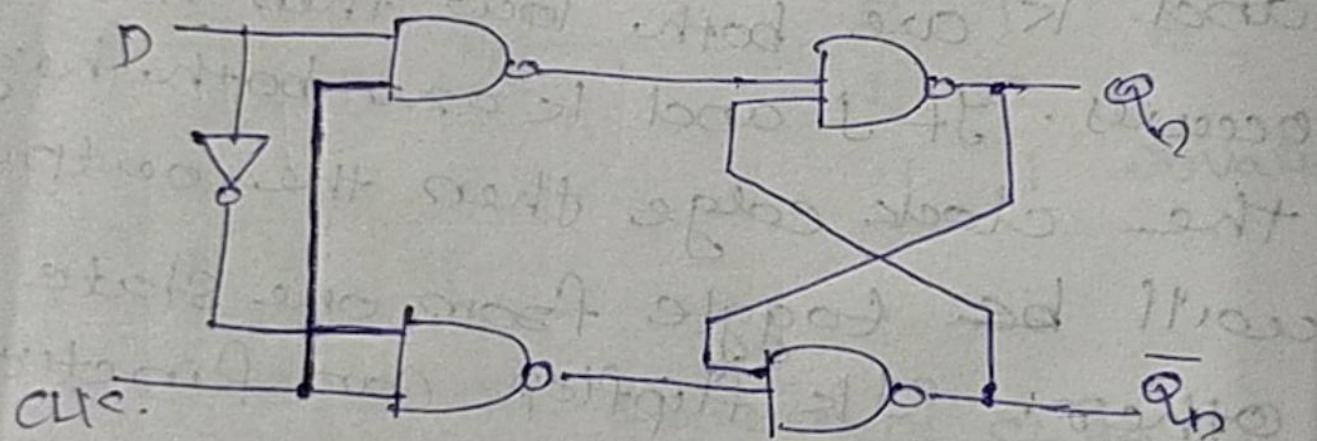
$0 \quad 1 \quad \rightarrow \quad 0 \quad - \text{Reset}$

$1 \quad 0 \quad \rightarrow \quad 1 \quad - \text{Set.}$

$1 \quad 1 \quad \rightarrow \quad \overline{Q_n}$

D Flip flop.

D flip flop is a better alternative that is very popular with digital electronics. They are commonly used for counters and shift registers and input synchronisation.



In D flip flop, the output can be only changed at the clock edge, and if the input changes at other times the output will be unaffected.

D Q₀ Q₁

0

x

0

Reset

1

x

1

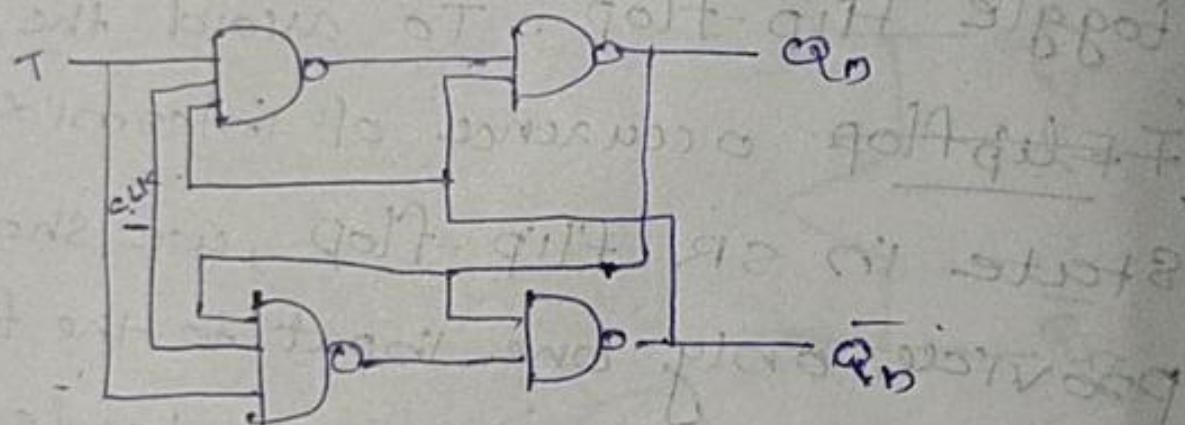
Set

T - Flipflop

T - flip-flop is also known as toggle flip-flop. To avoid the occurrence of intermediate state in SR flip-flop, we should provide only one input to the flipflop called Trigger input or toggle input(T). Then the flip-flop acts as a Toggle Switch. Toggling means changing the next state output to complement of the present state output.

We can design T flip flop by making simple modifications to the JK flip-flop. The T flip-flop is a

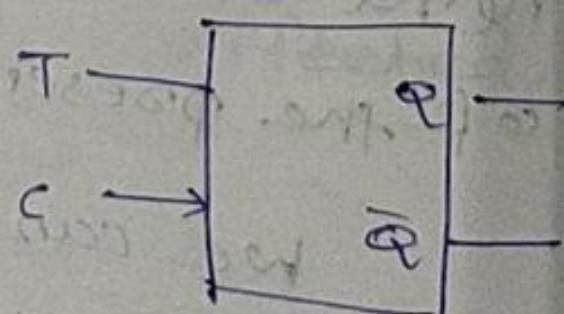
single input device and hence by connecting J and K inputs together and giving them with single input called T we can convert a JK flip-flop into T flip-flop. So a T flip-flop is sometimes called a single input JK flip-flop.



$T \rightarrow Q_0 \rightarrow Q_{n+1}$

$D \times Q_n$ - no charge collector

$\bar{D} \times \bar{Q}_n$



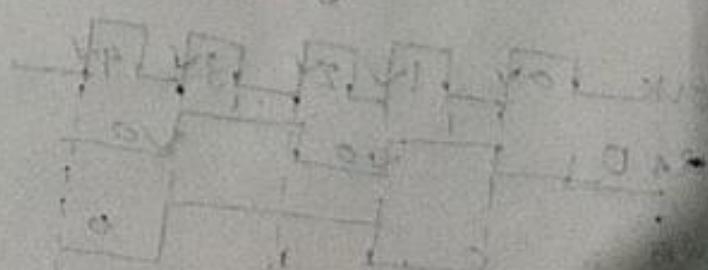
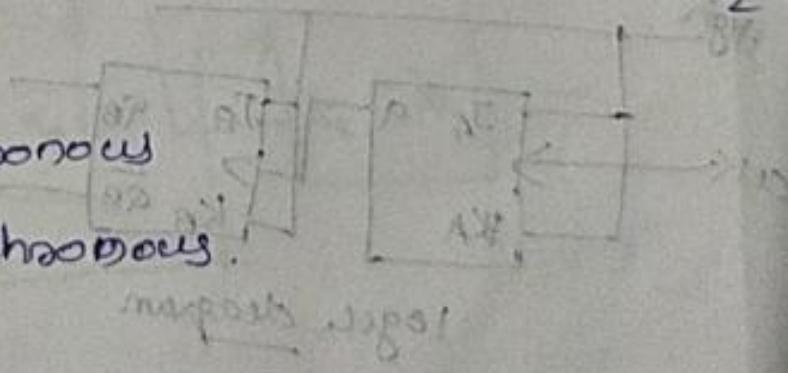
Counters (Synchronous & Asynchronous).

- > Counter is used to count number of pulses & it can also be used as frequency divider.
- > Counter can count in two ways.
 1. Up count ($0, 1, 2, \dots, N$)
 2. Down count ($N, N-1, \dots, 1, 0$)
- > Present present count of the counter represents state of counter.
- > Counter contains set of flip-flops, A n-bit counter will have N flip-flop & 2^n states.
- > Each state frequency = $\frac{\text{total freqency}}{2^n}$

Classification

1, Synchronous

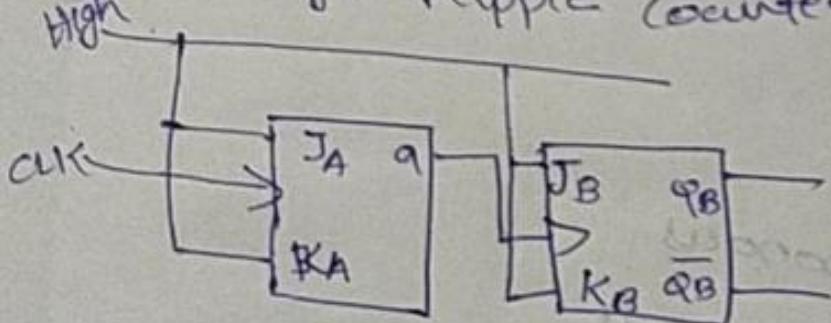
2, Asynchronous.



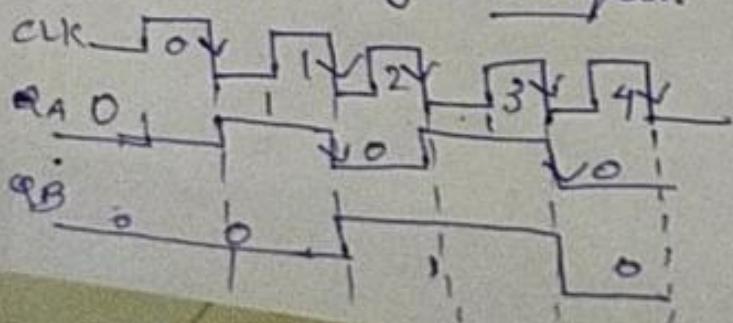
Asynchronous counter

- > It refers to a state that doesn't have a fixed time relationship with each other.
- > In Asynchronous counters flip-flops doesn't have a common clock pulse.
- > So, their states doesn't change exactly at same time.
- > 1st Flip-flop has clock, output of each flip-flop will access clock to the next flip flop in the counter.

Eg. Ripple counter

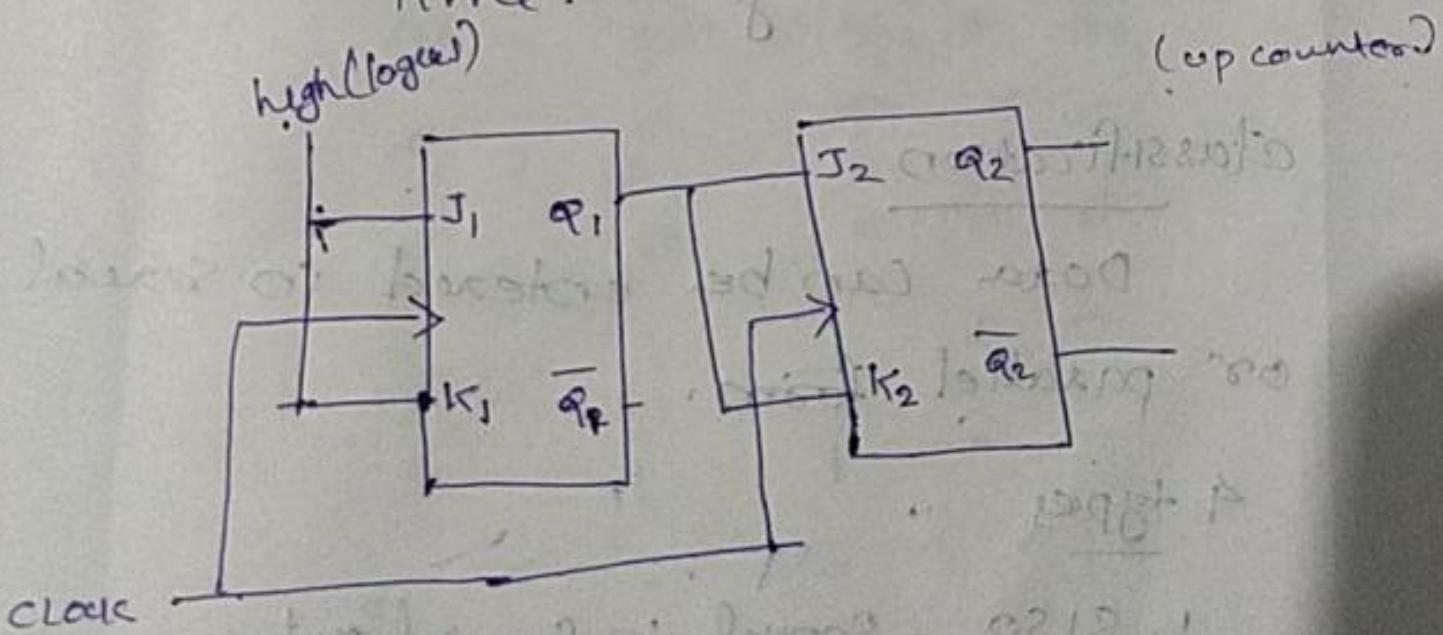


logic diagram



Synchronous counter

- > The counters which use clock signal to change their transition are called Synchronous counters.
- > That means the Synchronous counters depends on their clock input to change state values.
- > In Synchronous counters, all flip flops are connected to the same clock signal and all flip flops will trigger at the same time.



Clock

Two leading 0's inserted - 0010 11

Two leading 0's removed - 0110 11

Two leading 0's removed - 0010 11

Registers

> Data storage is called register.

> flip flop is one bit memory cell.

> To increase the storage capacity, we have to use group of flip flops.

These flip flops is known as Registers.

> Then n-bit register consists of 'n' no: of flipflops & is capable of storing 'n bit' coded.

Classification

Data can be entered in serial or parallel form.

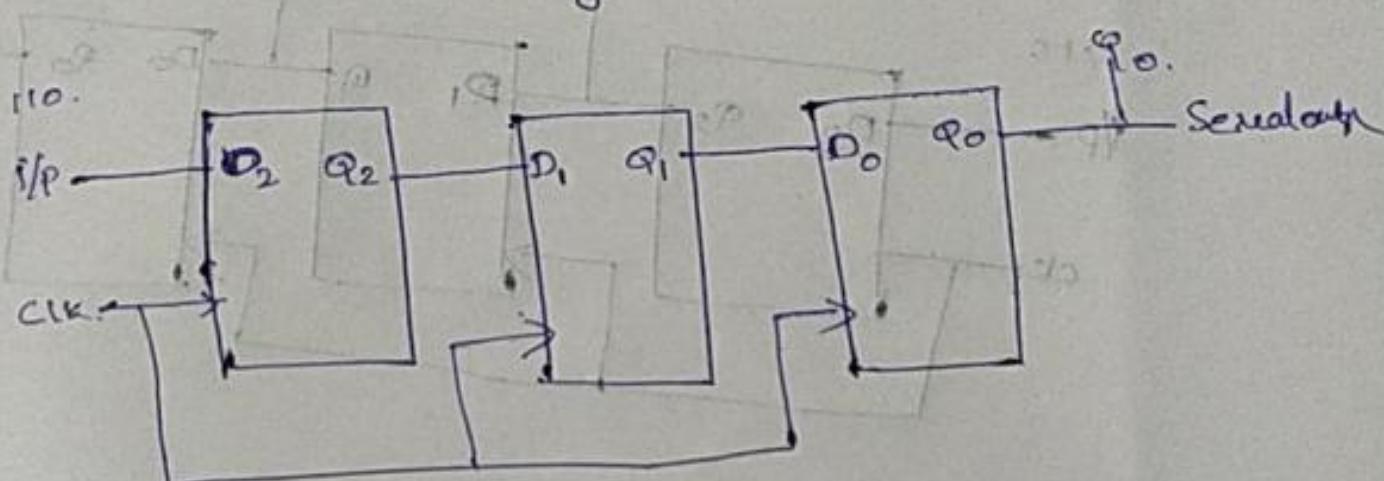
4 types

- 1) SISO - Serial in Serial out
- 2) SIPO - Serial in Parallel out
- 3) PISO - Parallel in Serial out

2) PIPD \rightarrow parallel in parallel out.

3) SISO shift register.

The shift register, which allows serial input and particular serial output. It's known as serial out SISO shift register.



CLK input Q₂ Q₁ Q₀

0 0 0 0 0

1 0 0 0 0

2 1 0 0 0

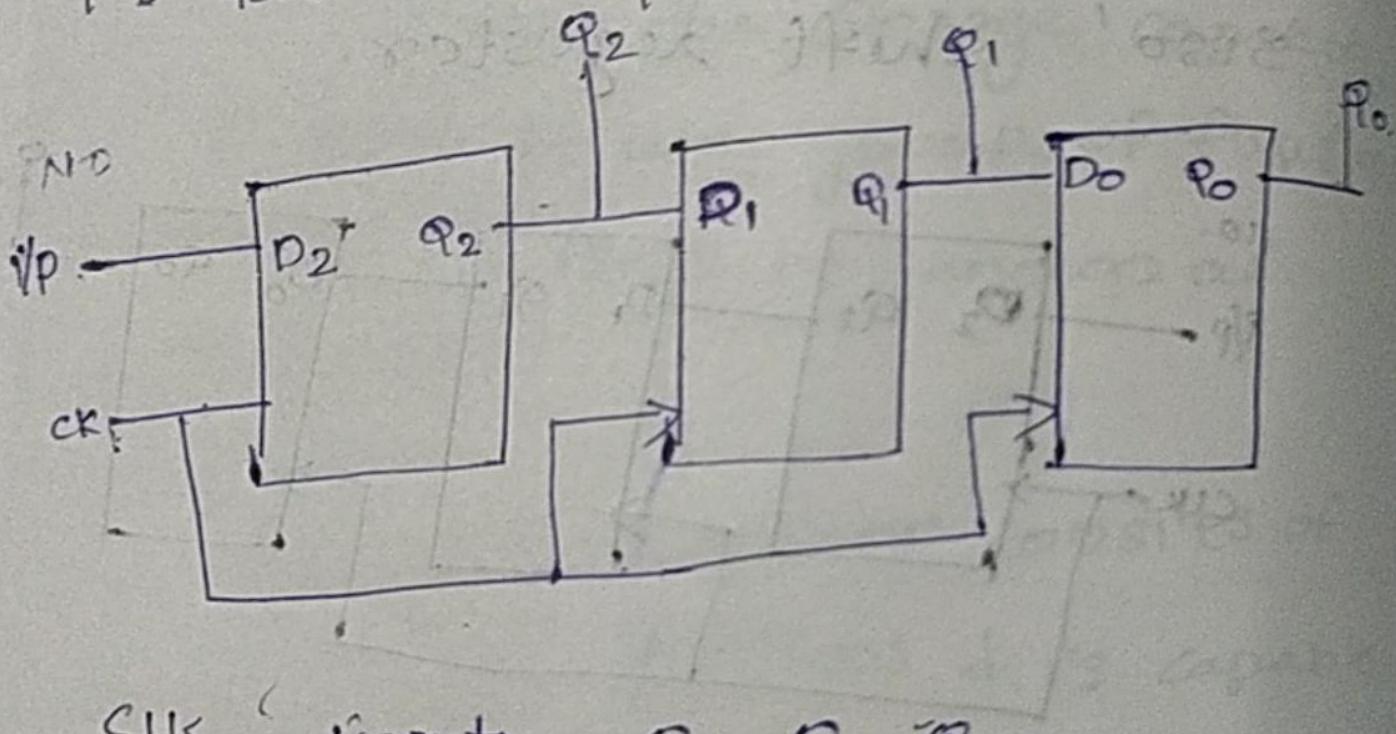
3 1 1 0 0

4 - 0 1

3) SIPO

Input :

Data bits are entered in the register serially but the output is taken in parallel.



CLK Input Q₂ Q₁ Q₀

0 0 0 0 0

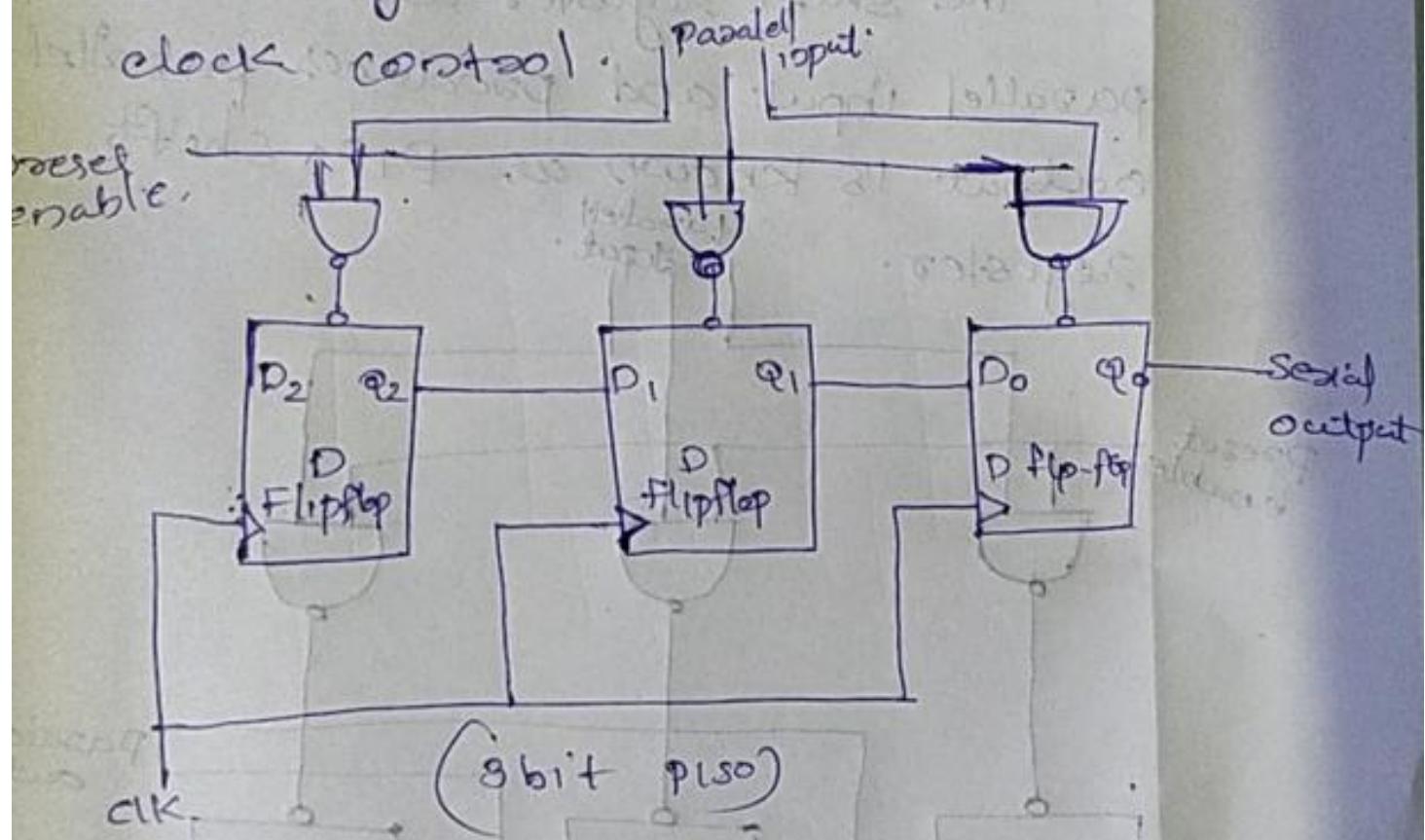
1 0 0 0 0

2 1 1 0 0

3 1 1 1 0

3, PISO (Parallel in Serial out)

the parallel data is loaded into the register serially simultaneously and it's shifted out of the register serially one bit at a time under clock control.

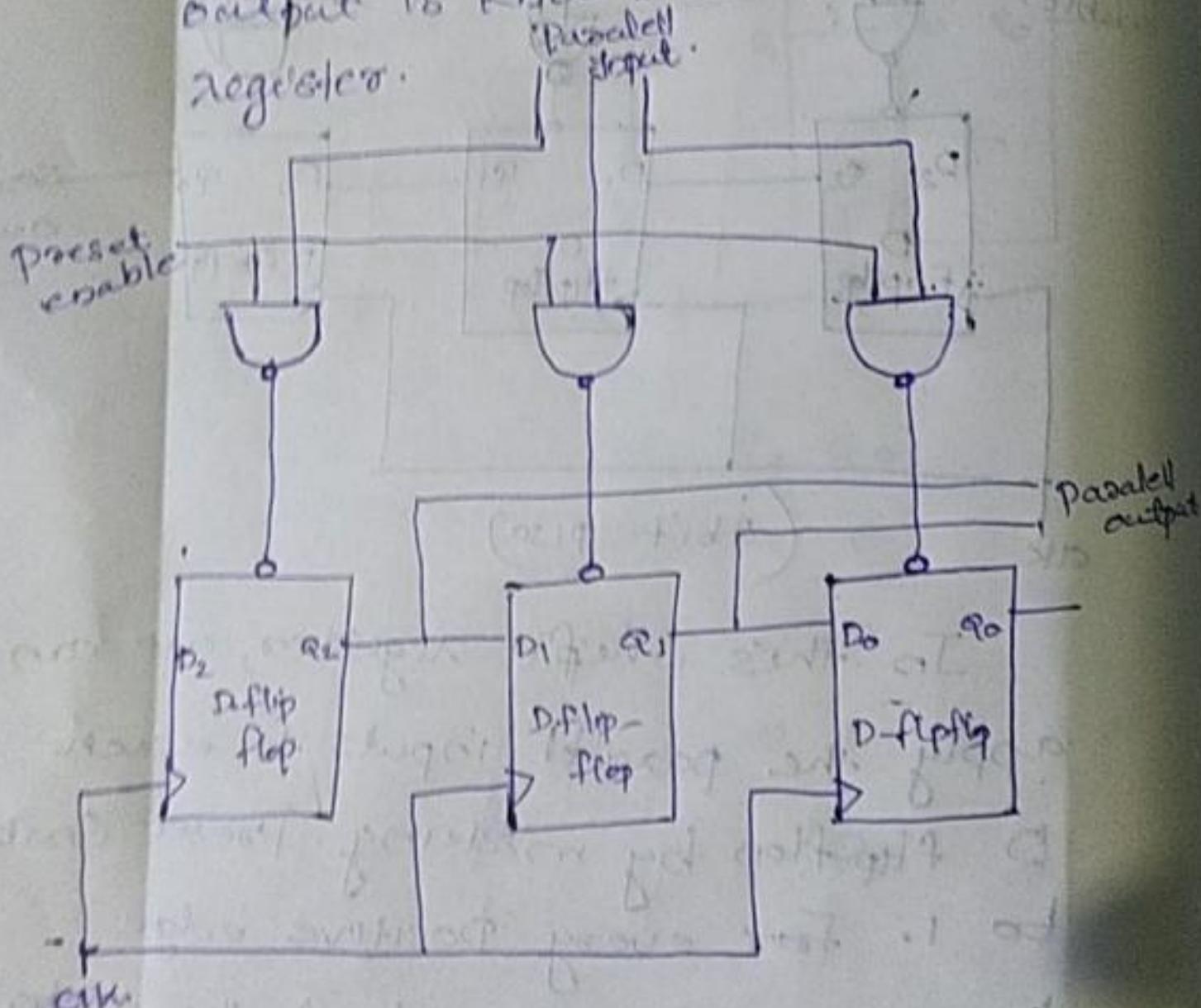


In this shift register, we can apply the parallel inputs to each D flip-flop by making Preset Enable to 1. For every positive edge triggering of clock signal, the data shifts from one stage to the next. So we will get the serial output.

Shows the right most D-Flip-Flop.

1) PIPo (Parallel In - Parallel Out Shift Register)

The shift register which allows parallel input and produces parallel output is known as PIPo Shift Register.



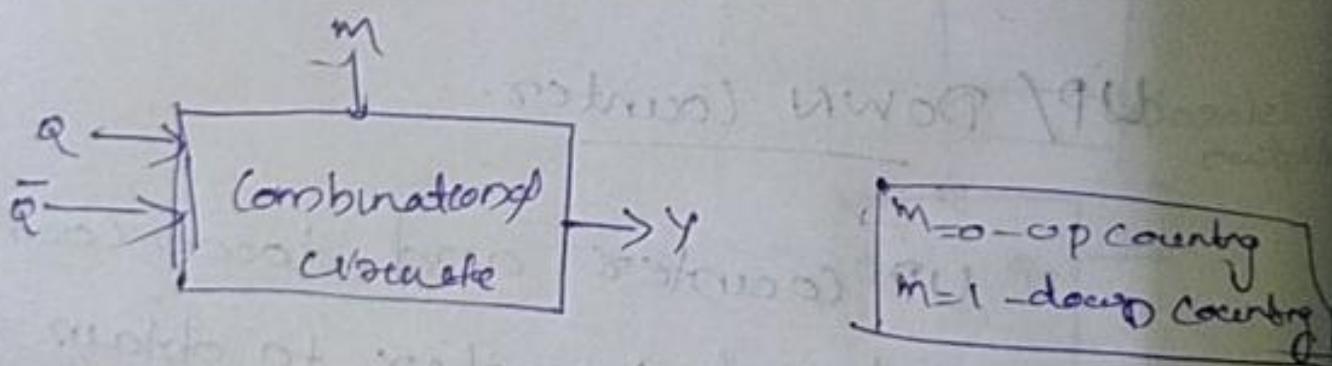
In this shift register, we can apply the parallel inputs to each D flip-flop by making preset Enable to 1. We can apply the parallel inputs through preset or clear. These two are asynchronous inputs. That means, the flip-flops produce the corresponding outputs, based on the values of asynchronous inputs. In this case the effect of outputs is independent of clock transition. So, we will get the parallel outputs from each D flip-flop.

UP/ DOWN Counter

UP counter and down counter is combined together to obtain an up/down counter. A mode control (M) input is also provided to select either up or down mode.

3 bit and 4 bit Up/Down Ripple Counter

- > we have designed the up counters and down counters separately.
- > But in practice, both these modes are combined.
- > A mode control input (m) is used to select either up or down mode.
- > A combinational circuit is required between each pair of flip flops.



M	Q	\bar{Q}	y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
			1

1 1 0 0
 1 1 1 1

K-map

	$\bar{Q}\bar{Q}$	$\bar{Q}Q$	$Q\bar{Q}$	QQ
\bar{m}	0	0	1	1
0	0	0	1	1
1	0	1	1	0

$$Y = I + \underline{II}$$

$$Y = \bar{m}Q + m\bar{Q}$$

