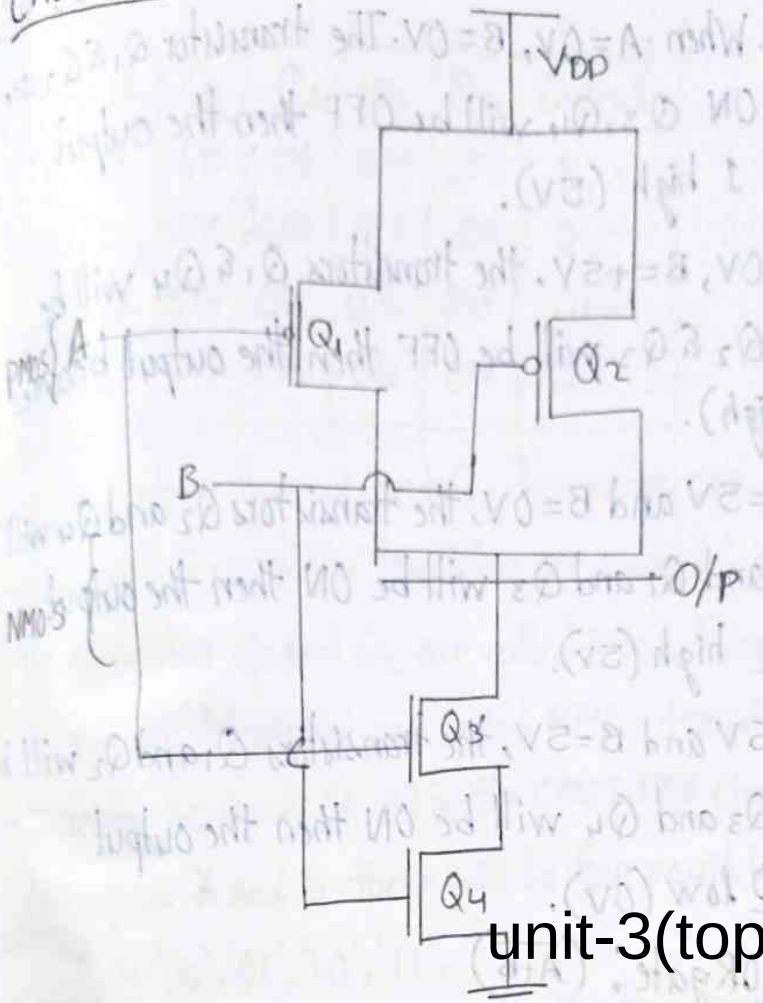
CMOS NAND Gate: $(\overline{A \cdot B})$



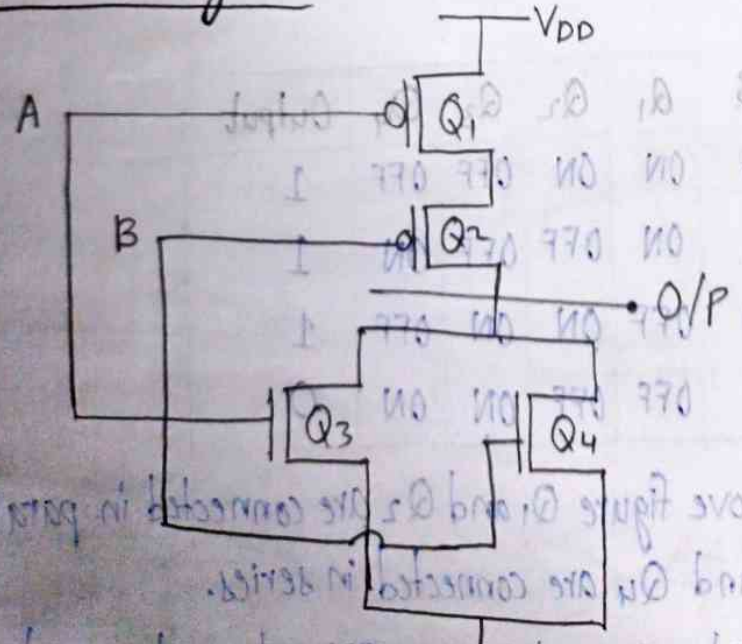
A	B	Q_1	Q_2	Q_3	Q_4	Output
0	0	ON	ON	OFF	OFF	1
0	1	ON	OFF	OFF	ON	1
1	0	OFF	ON	ON	OFF	1
1	1	OFF	OFF	ON	ON	0

- In the above figure Q_1 and Q_2 are connected in parallel and Q_3 and Q_4 are connected in series.
- The transistor Q_1 and Q_2 are PMOS transistors and Q_3 and Q_4 are NMOS transistors. For two inputs A & B.

There will be four possible combinations such as '00', '01', '10', '11'. When A=0V, B=0V. The transistor Q_1 & Q_2 are will be ON Q_3, Q_4 will be OFF then the output becomes 1 high (5V).

- When A=0V, B=+5V, the transistors Q_1 & Q_4 will be ON and Q_2 & Q_3 will be OFF then the output becomes (5V)(high).
- When A=5V and B=0V, the transistors Q_2 and Q_4 will be OFF and Q_1 and Q_3 will be ON then the output becomes 1 high (5V).
- When A=5V and B=5V, the transistors Q_1 and Q_2 will be OFF and Q_3 and Q_4 will be ON then the output becomes 0 low (0V).

CMOS NOR gate: $(\overline{A+B})$



Truth Table:

A	B	Q_1	Q_2	Q_3	Q_4	Output
0	0	ON	ON	OFF	OFF	1
0	1	ON	OFF	OFF	ON	0
1	0	OFF	ON	ON	OFF	0
1	1	OFF	OFF	ON	ON	0

- In the above figure Q_1 and Q_2 transistors are connected in series and Q_3 and Q_4 transistors are connected in parallel.
- The transistors Q_1 and Q_2 are called PMOS transistor and Q_3 and Q_4 transistors are called NMOS transistor.
- There are two inputs (A, B) in the CMOS NOR circuit. For the two inputs A and B there will be four possible combinations. such as '00', '01', '10', '11'.
- When the inputs A=0V, B=0V the transistors Q_1 & Q_2 will be ON and Q_3 & Q_4 will be OFF then the output becomes 1 high (5V).
- When the inputs A=0V, B=5V the transistor Q_1 & Q_4 will be ON and Q_2 & Q_3 will be OFF then the output becomes 0 low (0V).
- When the inputs A=5V, B=0V the transistors Q_1 & Q_4 will be OFF and Q_2 & Q_3 will be ON then output becomes 0 low (0V).
- When the inputs A=5V, B=5V, the transistors Q_1 & Q_2 will be OFF and Q_3 & Q_4 will be ON then the output becomes 0 low (0V).

Unit-4 Sequential Digital Circuits

Flip Flops

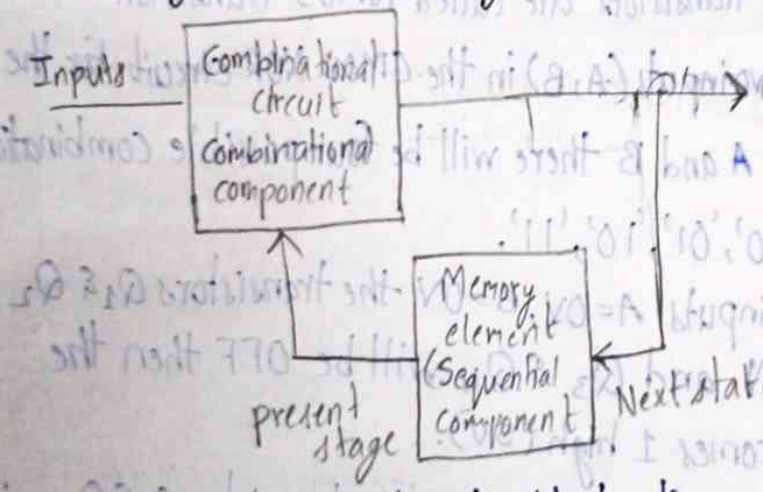
Sequential Circuits:

In sequential circuit the output variables depend not only on the present input variables but they also depend upon past input variables.

- Memory unit is required to store the past inputs in the sequential circuit.

examples are flipflops, serial adder, register, counter

Block diagram: (1 bit storage)



The above diagram shows the block diagram of sequential circuit and memory elements are connected to the combinational circuit as a feedback path.

Triggering: (means changing)

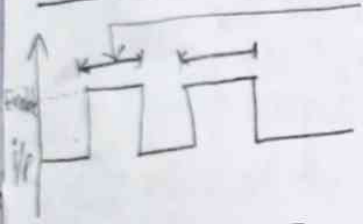
The momentary change (low-high-low) in the output is called trigger.

0 - 1 - 0

There are two types of triggering
(i) Level Triggering (ii) Edge Trigg ering

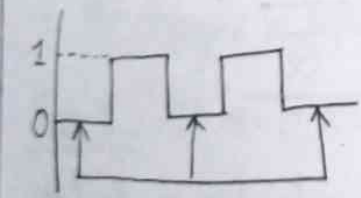
Level Triggering:
In the level triggering the output state is allowed to change according to inputs. When an active level (either +ve or -ve) is maintained at the enable input. There are two types of level triggering.
(a) Positive level triggering (b) Negative level triggering
The output of circuit response to input changes only when its enable input is 1 (high).

Positive Level Triggering:



circuit is enabled only when enable i/p is high

Negative Level Triggering:



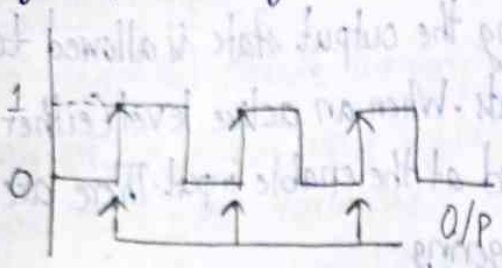
circuit is enabled only when level of t is low.

The output of circuit response to input changes only when it's enable input is 0 (low).

Edge Triggering: In the edge triggering the output response to the changes in the input only at the positive or negative edge of the clock pulse at the clock input.
There are two types of triggering a) Positive edge triggering b) Negative edge triggering

Positive Edge Triggering: (Rising)

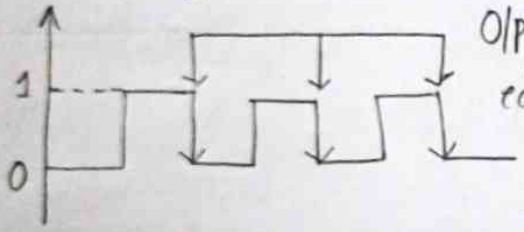
Here the output response to the changes in the input only at positive edge of the clock pulse at clock input



O/p responds only at the +ve edges of the pulses.

Negative Edge Triggering: (falling)

Here the output response to the changes in the input only at negative edge of the clock pulse at clock input.

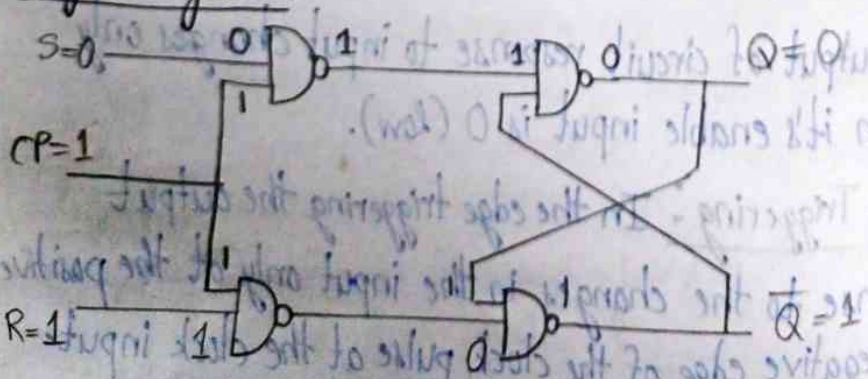


O/p responds only at the -ve edges of the pulse

SR Flipflop:

Block diagram:

Logic diagram:



SR Flipflop using NAND gates

Truth Table: (X - don't care '0' or '1')

Clockpulse	S	R	Q_n (Present State)	Q_{n+1} (Next State)	
0	X	X	0	0	
↑	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:

Flipflop is one bit storage device. positive edge triggered SR flipflop output response to S and R inputs only at positive edges of the clock pulse.

At any other instants of time the SR flipflop will not response to the changes in input.

Case1: If S=0, R=0 and the clock pulse is applied the output do not change i.e., $Q_{n+1} = Q_n$

Case2: If S=0, R=1 and the clock pulse is applied then $Q_{n+1}=0$. This state is called reset state.

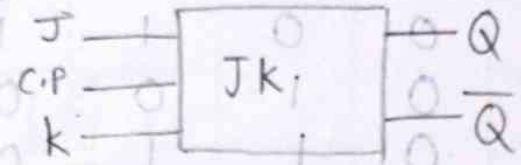
Case3: If S=1, R=0, the clock pulse is applied then $Q_{n+1}=1$. This state is called set state.

Case4: If S=1, R=1, the clock pulse is applied then

it is undeterminate state. So this state is called indeterminate state.

JK Flipflop:

Block diagram:



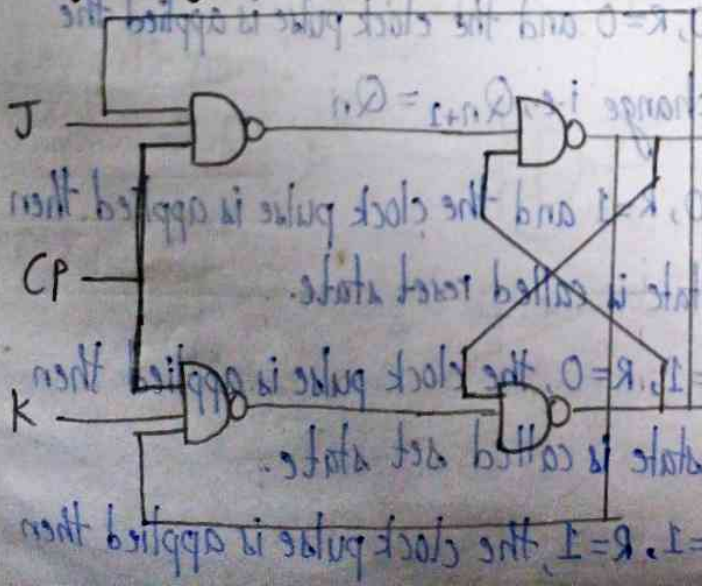
Truth Table:

J	k	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	$\overline{Q_n}$

≡

C.P	J	k	Q_n	Q_{n+1}	
0	×	×	0	0	
↑	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	1	Toggle State
↑	1	1	1	0	

Logic Diagram:

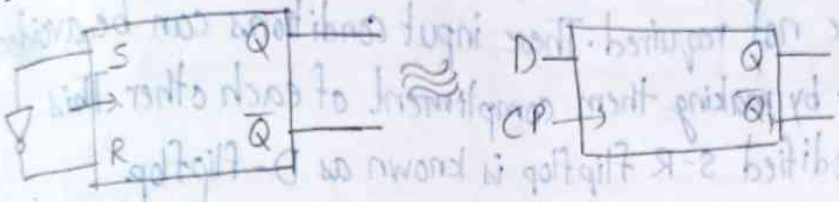


k-map Simplification:

Q_n \ JK	$\bar{J}\bar{K}$	$\bar{J}K$	JK	$J\bar{K}$
$\bar{Q}_n$	0	0	1	1
Q_n	1	0	0	1

$$Q_{n+1} = J\bar{Q}_n + \bar{K}Q_n$$

D-Flip flop (Delay):

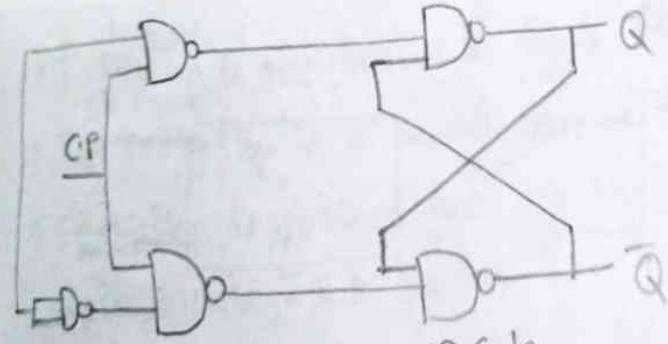


Truth Table:

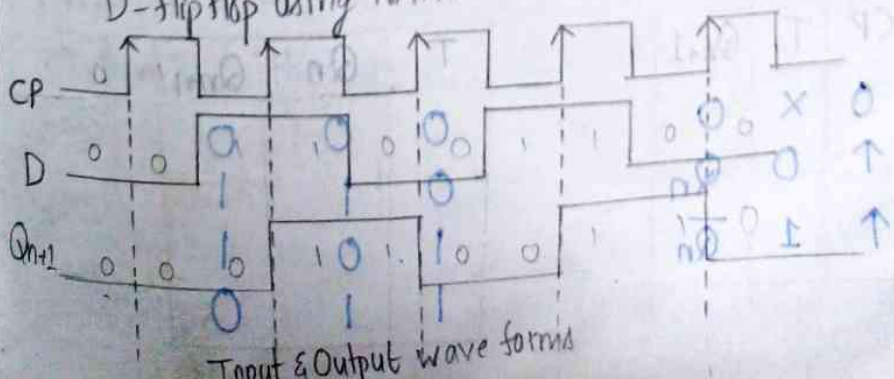
CP	D	Q_{n+1}
0	×	Q_n
↑	0	0
↑	1	1

$D = Q_{n+1}$

Logic Diagram:



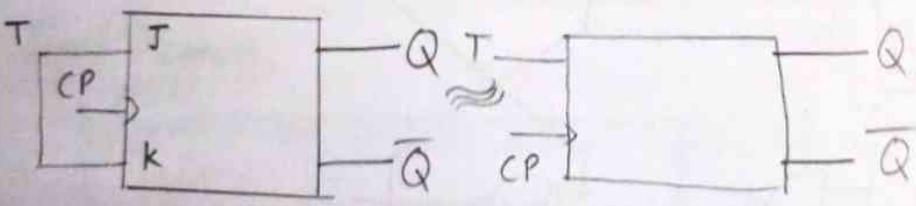
D-flip flop using NAND Gate



Input & Output wave forms

- The basic building block of D-flipflop is S-R flipflop. The S-R flipflop has 2 inputs S and R.
- From the truth table of S-R flipflop we can realise that the both inputs are same the output either does not change or it is invalid.
- In many practical applications these inputs conditions are not required. These input conditions can be avoided for by making them compliment of each other. This modified S-R flipflop is known as D-flipflop.
- From the truth table for D-flipflop Q_{n+1} function follows D-input at the positive going edges of the clock pulses. Hence the characteristic equation of D-flipflop is $Q_{n+1}=D$
- The output is delayed by 1 clock period. So, D-flipflop is known as delay flipflop.

T-flipflop:

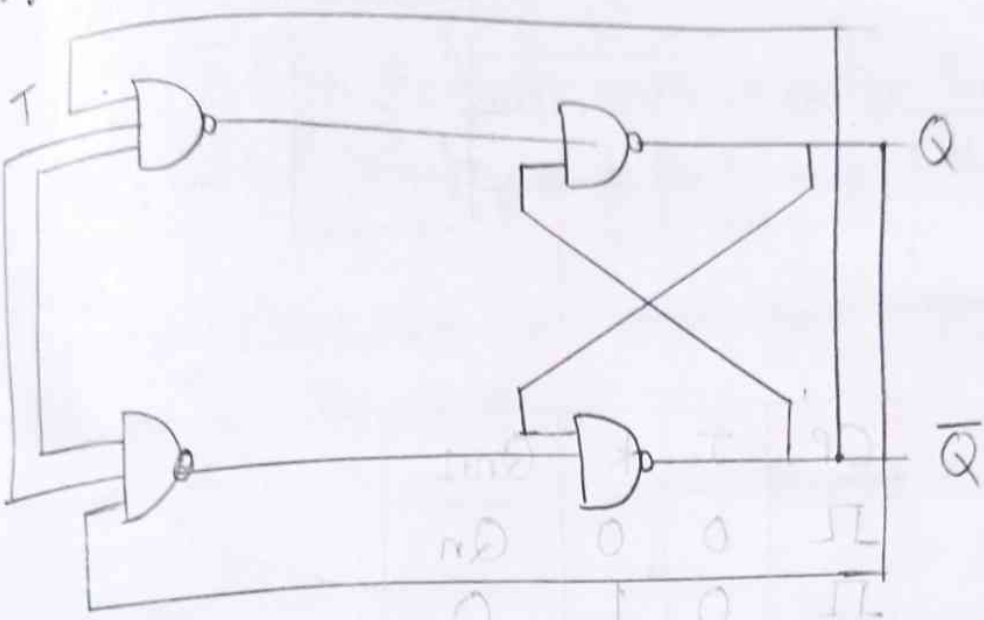


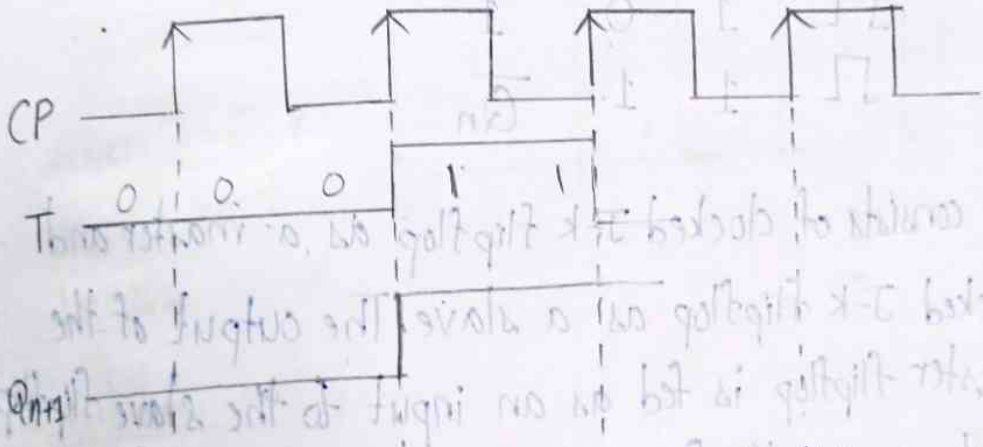
Truth Table:

CP	T	Q_{n+1}
0	X	0
↑	0	Q_n
↑	1	$\overline{Q_n}$

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

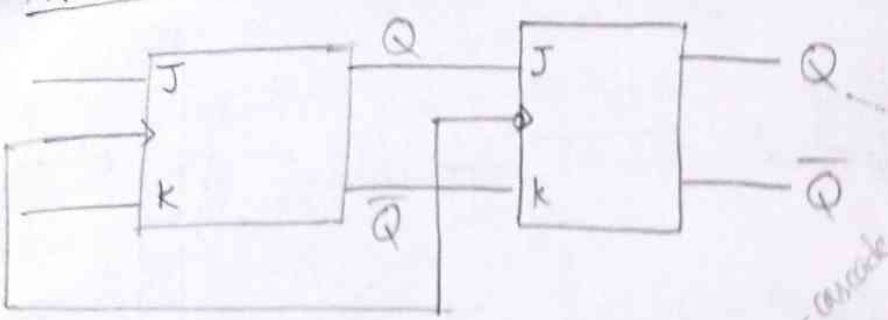
Logic Diagram:





- T flipflop is also known as toggle flipflop.
- The T-flipflop is a modification of the J-k flipflop.
- T-flipflop is obtained from J-k flipflop by connecting both inputs J & k together.
- When T=0 there is no change in the output.
- When T=1 the output is toggles.

Master-Slave J-k flip flop:



CP	J	k	Qn+1
⎍	0	0	Qn
⎍	0	1	0
⎍	1	0	1
⎍	1	1	$\overline{Qn}$

- It consists of clocked J-k flipflop as a master and clocked J-k flipflop as a slave. The output of the master flipflop is fed as an input to the slave flipflop. As shown in the figure the clock signal is connected directly to the master flipflop but it is connected directly to the master flip through inverter to slave flip flop. So, the information present at the J and k inputs is transmitted to the output of master flipflop on the positive clock pulse and it is held their until the negative clock pulse occurs after which it is allowed to pass through to the output of slave.
- When J=1, k=0 the master sets on positive clock the high output of master drives the J input of

of slave so at negative clock slave set copying the action of master.

- When J=0, k=1 the master resets on positive clock, the high output of master goes to the k-input of the slave.
∴ At the negative clock slave resets again copying the action of the master.
- When J=1, k=1 master toggles on the positive clock and slave then copies the output of master on the negative clock.
- When J=0, k=0 the output of master remains same at the positive clock, so the output of slave also remains same at the negative clock.

Excitation Tables:

SR flipflop

S	R	O/P
0	0	No change
0	1	reset
1	0	set
1	1	indeter minate

Excitation Table

Qn	Qn+1	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

J-K flipflop:

Truth Table:

J	K	O/P
0	0	Q_n
0	1	0
1	0	1
1	1	$\overline{Q_n}$

Excitation Table:

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

D-flip flop:

Truth Table:

D	Q_{n+1}
0	0
1	1

Excitation Tables:

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

T-flipflop:

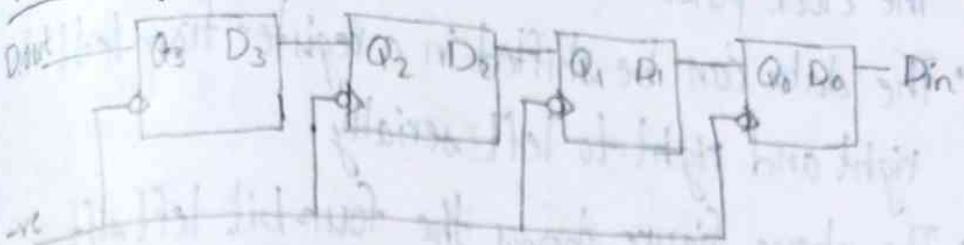
Truth Table:

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

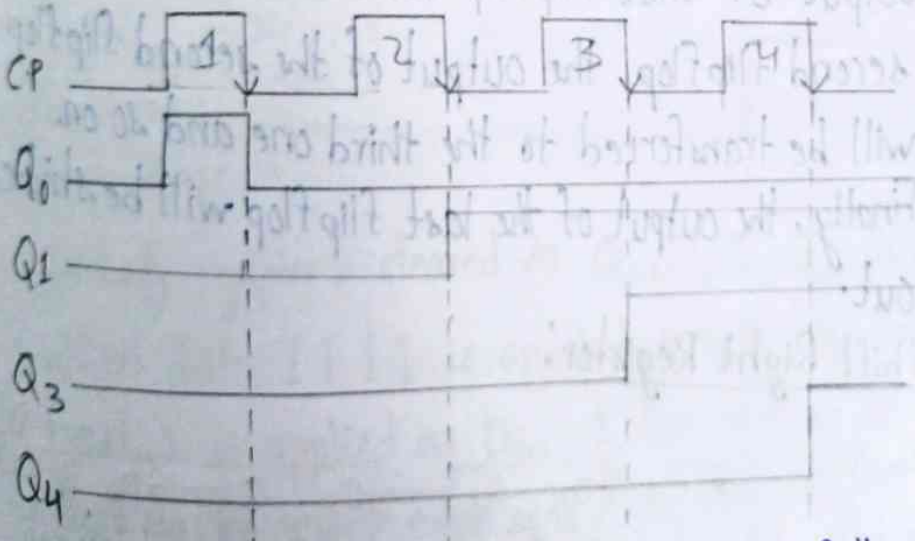
Excitation Table:

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

Left shift register:



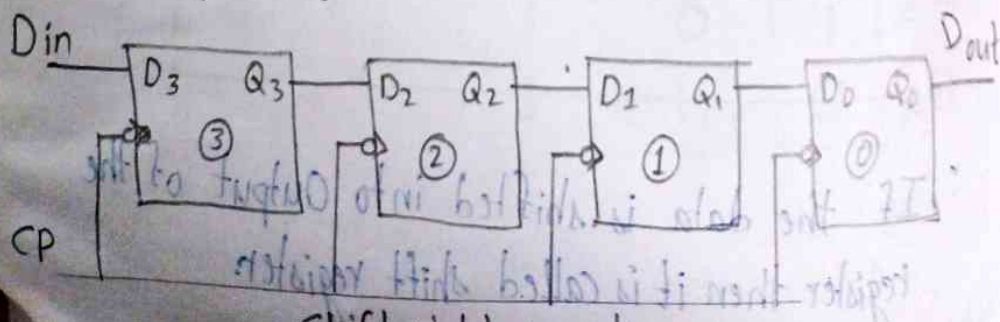
CP	Q_3	Q_2	Q_1	Q_0	D_{in}
initial	0	0	0	0	1
↓	0	0	0	1	1
↓	0	0	1	1	1
↓	0	1	1	1	1
↓	1	1	1	1	1



If the data is shifted into output of the register then it is called shift register.

- The shifting of the data has been done based on the clock pulse.
- The data can be shifted in a register from left to right and right to left serially.
- The above figure shows the four-bit left sift register, in this four flip flops are used to store four bits serially.
- Serial data applied at first flip flop and output of the first flip flop is given as input to the next flipflop.
- At the negative edge of each clock pulse a new bit will be transfered into the first flipflop during this condition already available data at the output of first flipflop will be transfered to the second flipflop, the output of the second flipflop will be transferred to the third one and so on. Finally, the output of the last flip flop will be shifted out.

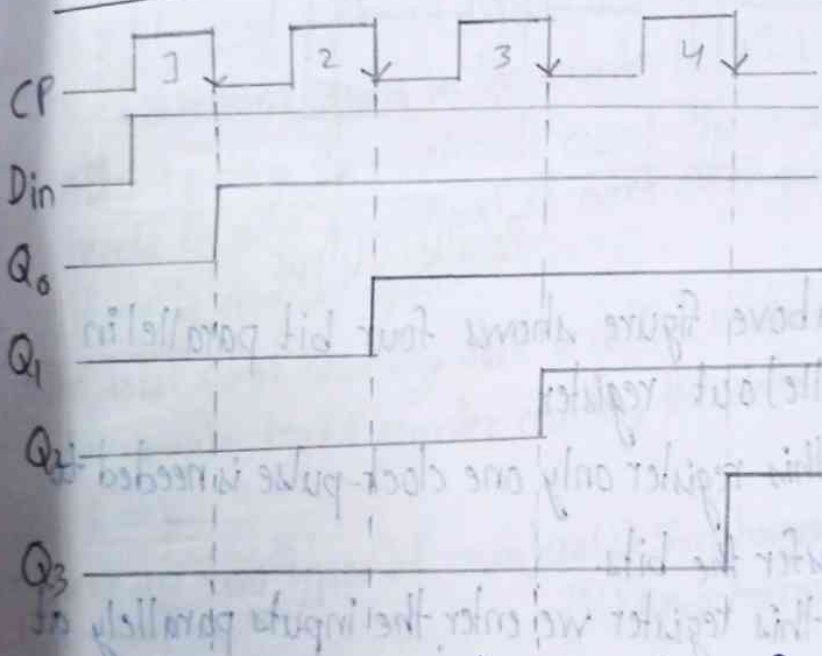
Shift Right Register:



Shift right Operation.

CP	Din	Q_3	Q_2	Q_1	Q_0
Initially	1	0	0	0	0
↓ 1st	1	1	0	0	0
↓ 2nd	1	1	1	0	0
↓ 3rd	1	1	1	1	0
↓ 4th	1	1	1	1	1

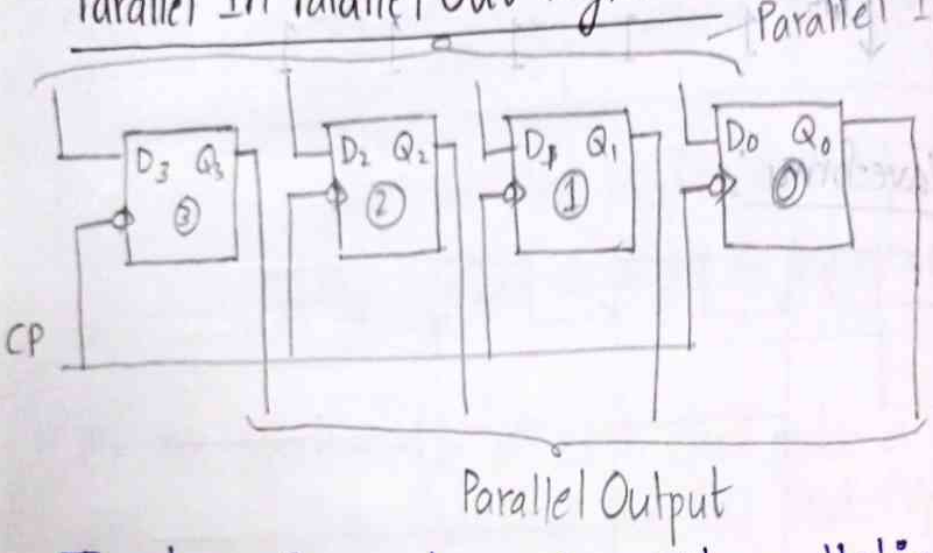
Waveforms:



- Initially register is cleared so $Q_3 Q_2 Q_1 Q_0 = 0000$
- When data 1111 is applied serially the left most 1 is applied as Din = 1 the arrival of first falling clock edge sets the first flipflop output. and the output becomes 1000.
- When second negative clock edge reaches then it sets Q_2 and the register contents becomes 1100.

- When third negative clock-edge reaches then it sets the Q_1 then the register content becomes 1110.
- When fourth negative clock edge reaches sets the Q_0 then register content becomes 1111.

Parallel In Parallel Out Register:



- The above figure shows four bit parallel in parallel out register
- For this register only one clock-pulse is needed to transfer the bits.
- In this register we enter the inputs parallely at a time the output comes parallely.
- These type of registers are used in bi-directional data transfer.

Registers: → They are used to store data in the form of 0's & 1's. → n number of bits
→ It is designed with flip flops

Counters → JK & T flipflops

- A counter is a register capable of counting the no. of clock pulses arriving at its clock input.
- On arrival of each clock pulse the counter is incremented by 1.
- In case of down counter it is decremented by 1.
- The n-bit counter has 'n' number of flip flops and it has 2^n states of outputs.
- For example 2-bit counter has 2 flipflops and it has $2^2 = 4$ no. of different states. (00, 01, 10, 11).
- The maximum count in the counter is $2^n - 1$.
- After reaching the maximum count the counter resets to zero on arrival of next clock pulse and it starts counting again.
- The total no. of counting state is called modules. forexample Mod 6 counter counting the states from 0 to 5.
- There are two types of counters. (i) Synchronous Counter (ii) Asynchronous Counter

Synchronous Counter:

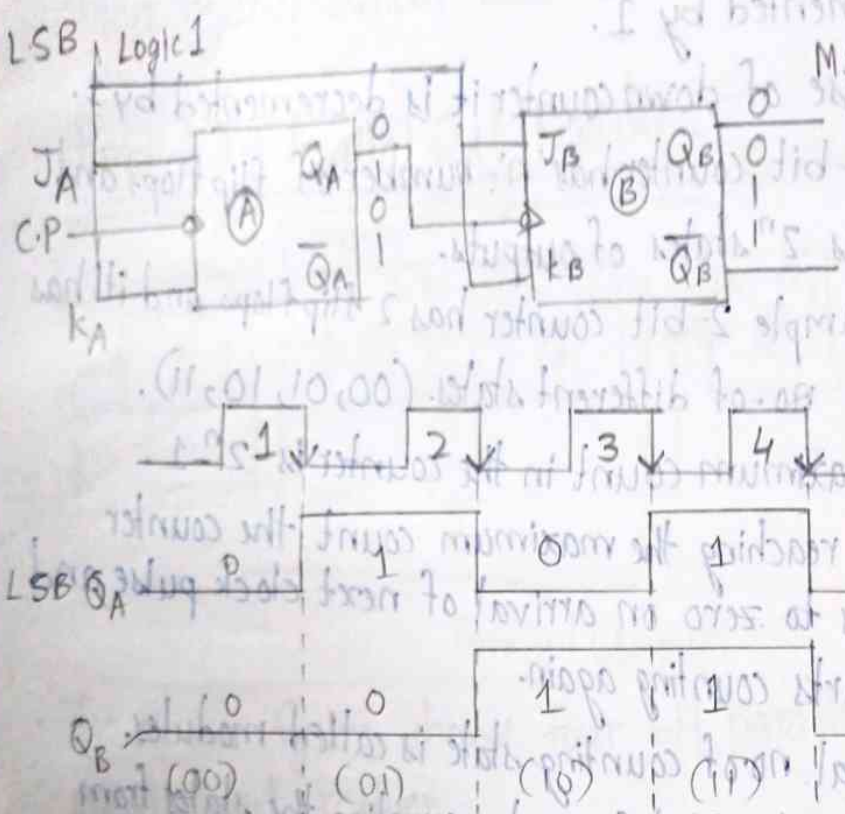
When counter is clocked such that each flip flop in the counter is triggered at the same time, then the counter is called synchronous counter.

Asynchronous Counter: (Ripple Counter)

Asynchronous counter consists of a series connection of complementing flipflop with the output of

each flipflop connected to clock input of next high order flipflop.

2-bit Asynchronous Counter:



Count	00	01	10	11
Stage	0	1	2	3

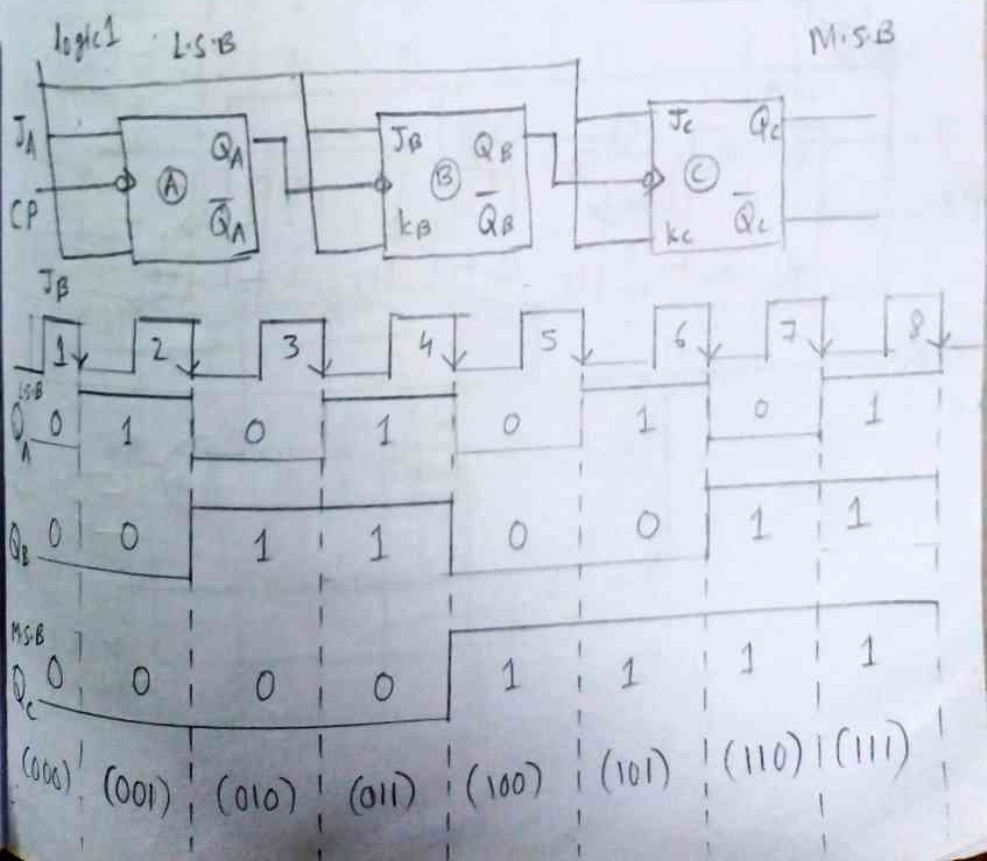
fig: Timing diagram of 2-bit counter

- The figure shows the 2-bit asynchronous counter using J-k flipflops.
- The clock signal is connected to clock input of only first stage flipflop.
- The clock input of second stage F.F is triggered by

Q_A output of 1st stage.

- The transistion (change) of input clock pulse and transition of Q_A output of 1st stage can never occur at the same time. So two flipflop are never triggered simultaneously which results in asynchronous counter operation.
- The second figure shows timing diagram or output wave forms. It illustrates the change in the state of flipflop outputs in response to the clock.
- J and k inputs of each flipflop is always high hence the output will toggle for each negative edge of the clock input.

3 bit (or) Mod 8 Asynchronous Counter.

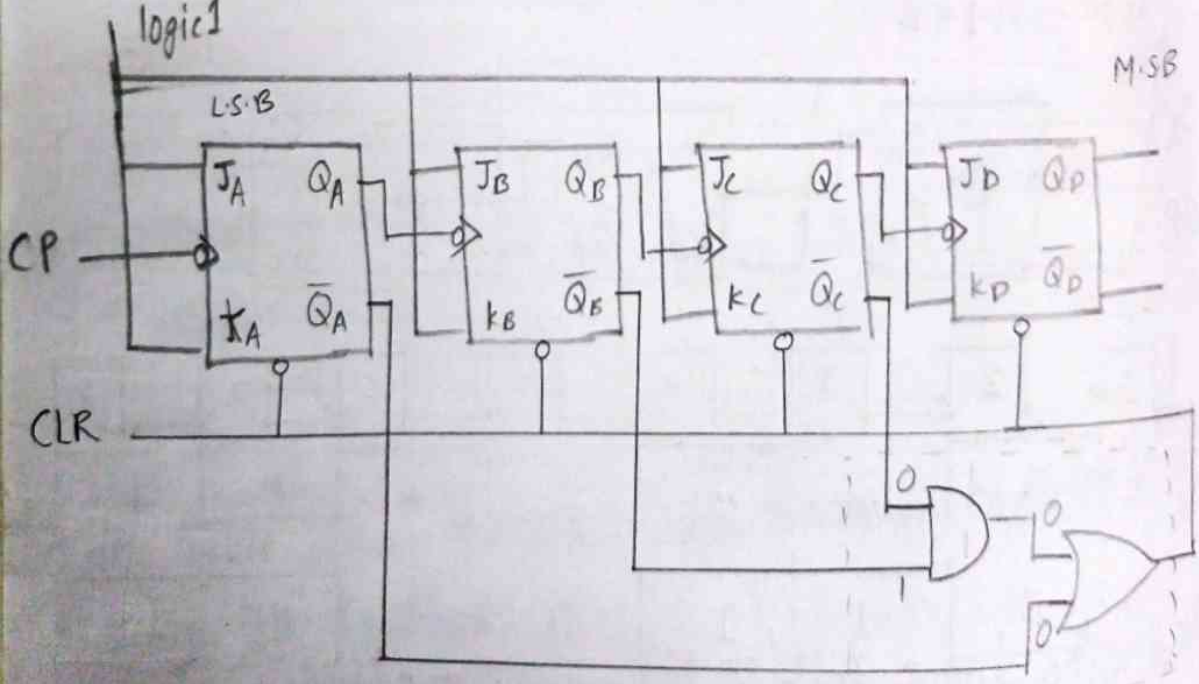


Counting	000	001	010	011	100	101	110	111
Stage	0	1	2	3	4	5	6	7

Design of ripple or Mod-N counter:

- Determine the no. of flipflop needed.
- Choose the type of flipflop to be used: T or J-K.
  If T flipflop are used connect T input of all flipflops to logic 1. If J-k flipflops are used to connect both J and k inputs of all flipflops to logic 1.
- Write the truth table for the counter.
- Derive the reset logic by k-map simplification.
- Draw the logic diagram.

Mod-10 Counter:



Truth Table for reset logic:

Clk	A	B	C	D	Output	
↓	0	0	0	0	1	Valid States
↓	0	0	0	1	1	
↓	0	0	1	0	1	
↓	0	0	1	1	1	
↓	0	1	0	0	1	
↓	0	1	0	1	1	
↓	0	1	1	0	1	
↓	0	1	1	1	1	
	1	0	0	0	1	
	1	0	0	1	1	
	1	0	1	0	0	Invalid States
	1	0	1	1	0	
	1	1	0	0	0	
	1	1	0	1	0	
	1	1	1	0	0	
	1	1	1	1	0	

k-map Simplification:

AB \ CD	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	1	1	1	1
$\bar{A}B$	1	1	1	1
AB				
$A\bar{B}$	1	1		

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

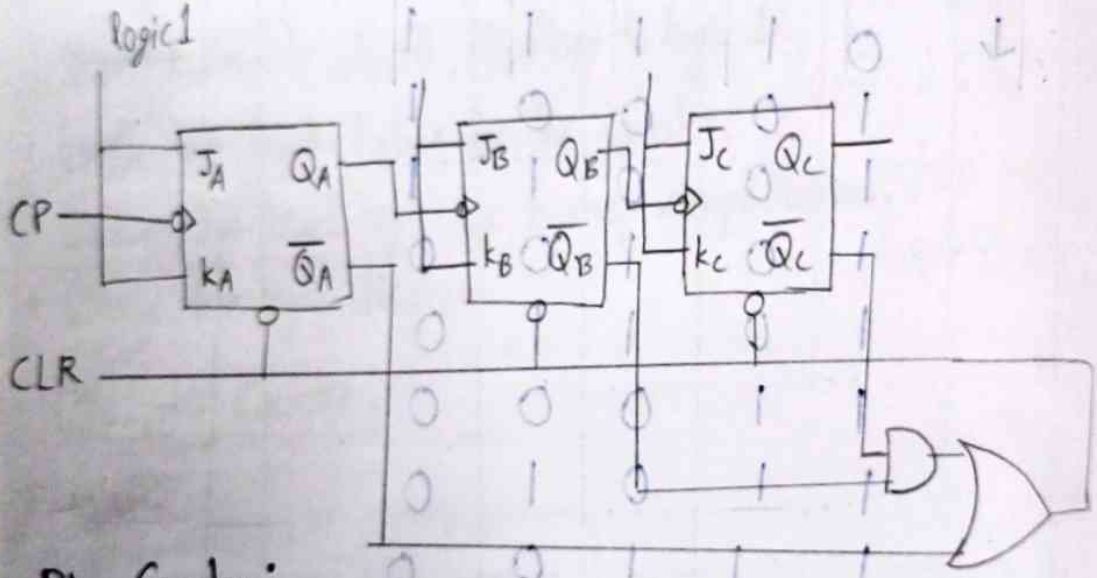
Mod 5 Counter:

C·P	A	B	C	O/P
0	0	0	0	1
	0	0	1	1
	0	1	0	1
	0	1	1	1
	1	0	0	1
	1	0	1	0
	1	1	0	0
	1	1	1	0

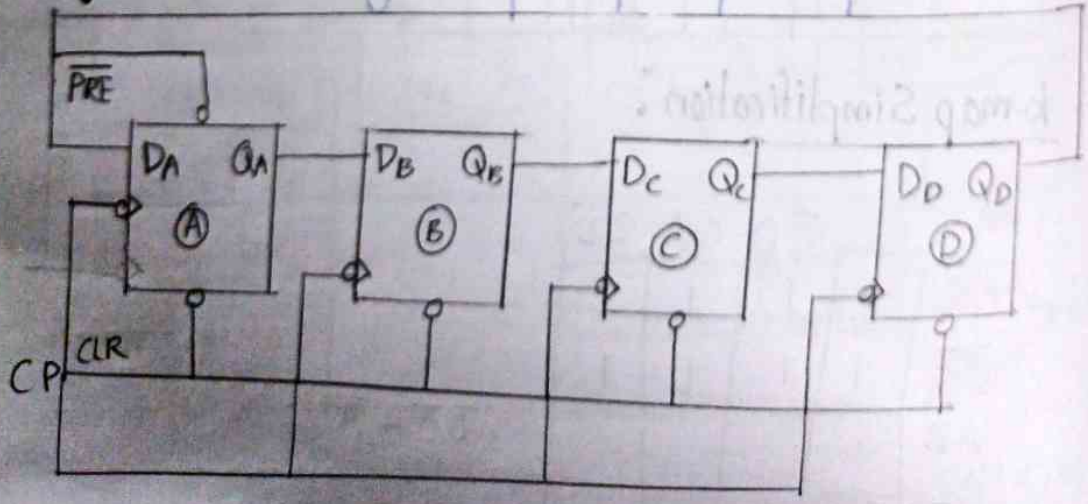
k-mapping

A \ BC	$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
$\bar{A}$	1	1	1	1
A	1	0	0	0

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



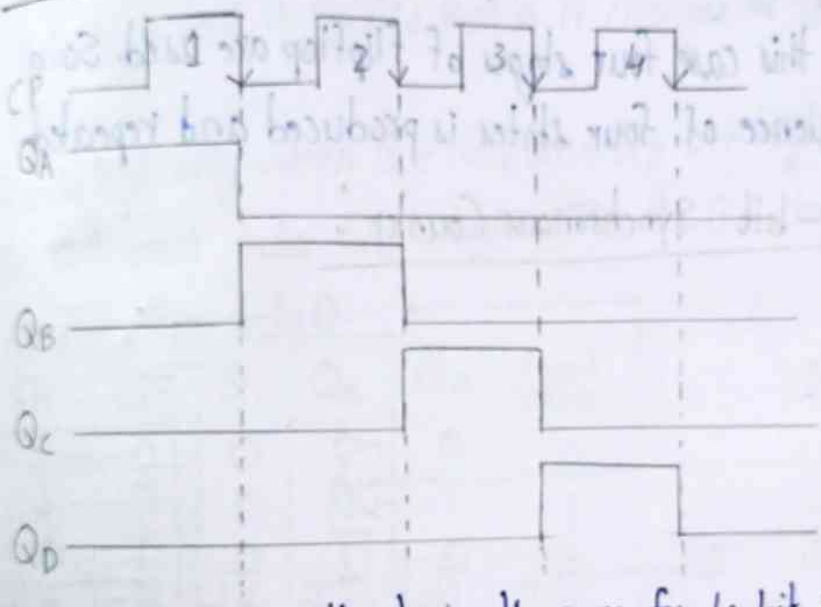
Ring Counter:



Truth Table:

CP	Q_A	Q_B	Q_C	Q_D
0	1	0	0	0
1	0	1	0	0
2	0	0	1	0
3	0	0	0	1
4	1	0	0	0

Wave forms:

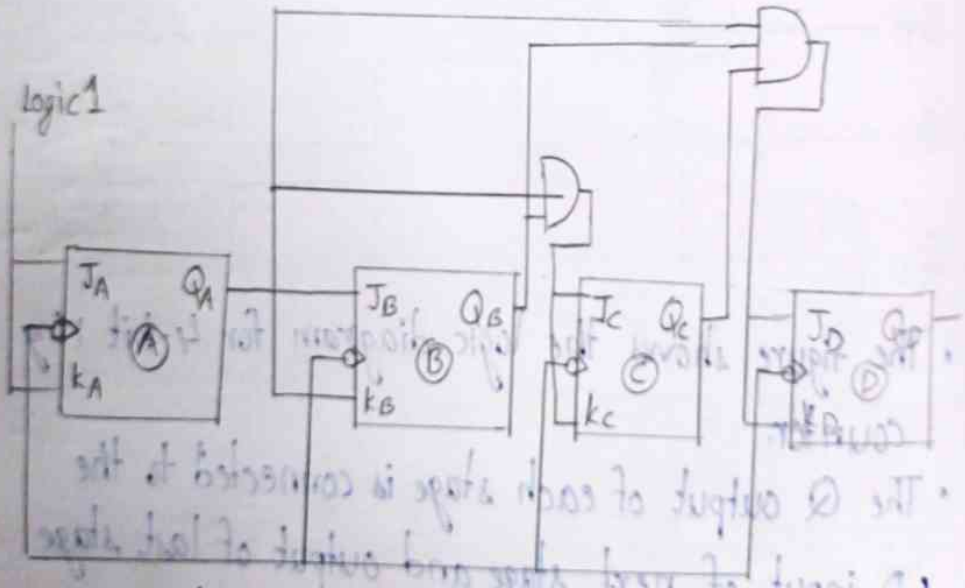


- The figure shows the logic diagram for 4-bit ring counter.
- The Q output of each stage is connected to the D-input of next stage and output of last stage is fed back to input of first stage.
- The $\overline{CLR}$ followed by $\overline{PRE}$ makes the output of first stage to 1. and remaining outputs are D.
  i.e., $Q_A = 1$, Q_B, Q_C, Q_D are zero (0).
- The first clock pulse produces $Q_B = 1$ and remaining

outputs are zero according to the clock pulse applied at the clock input C.P.

- A sequence of four states are produced these states are listed in the truth table.
- In this truth table 1 is always retained in the counter and simply shifted around the ring advancing one stage for each clock pulse.
- In this case four stages of flipflop are used. So a sequence of four states is produced and repeated.

4-bit Synchronous Counter:

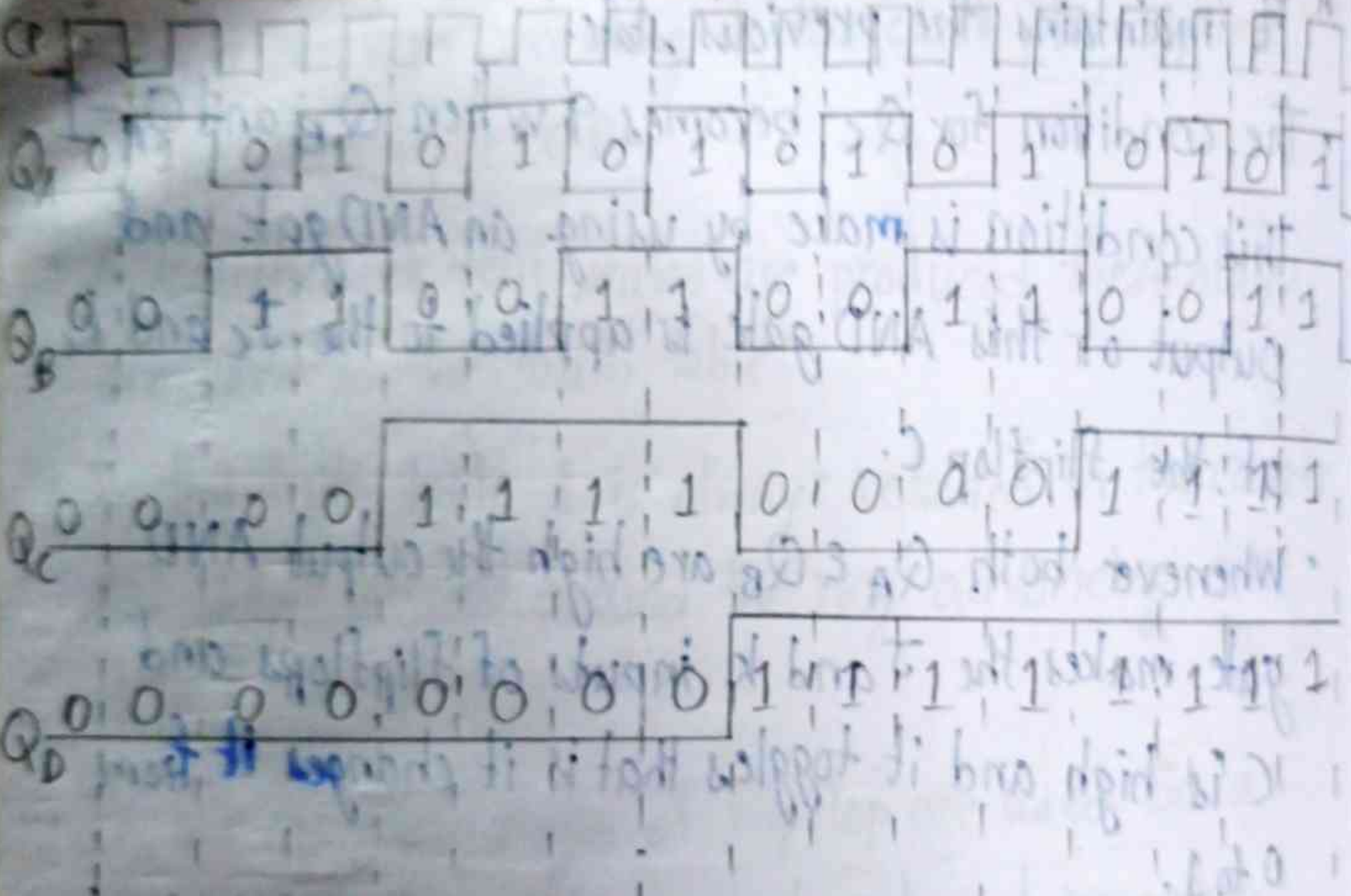


- 4 bit synchronous counter contains 4 flipflops. It counts from 0 to 15 states.
- The logic diagram $J_A = k_A = 1$ and hence Q_A changes state at every negative edge of the clock.
- Q_B changes its state only when $Q_A = 1$ otherwise

it maintains the previous state.

- The condition for Q_C becomes 9 when Q_B and $Q_A = 1$ This condition is make by using an AND gate and output of this AND gate is applied to the J_C and k_C of the flipflop C.
- Whenever both Q_A & Q_B are high the output AND gate makes the J and k inputs of flipflops and C is high and it toggles that is it changes it from 0 to 1.
- Q_D is high only when Q_C, Q_B and Q_A are high at the 15th clock pulse Q_D is high. After its Q_D changes its state from 1 to 0.

Clk	Q_D	Q_C	Q_B	Q_A	Count Value
↓	0	0	0	0	0
↓	0	0	0	1	1
↓	0	0	1	0	2
↓	0	0	1	1	3
↓	0	1	0	0	4
↓	0	1	0	1	5
↓	0	1	1	0	6
↓	0	1	1	1	7
↓	1	0	0	0	8
↓	1	0	0	1	9
↓	1	0	1	0	10
↓	1	0	1	1	11
↓	1	1	0	0	12
↓	1	1	0	1	13
↓	1	1	1	0	14
↓	1	1	1	1	15


1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
CP
Q0 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
Q1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1
Q2 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1
Q3 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1