

SEQUENTIAL LOGIC DESIGN

Sequential logic circuit → whose output depends not only on the present value of its input signals but on the sequence of past inputs.

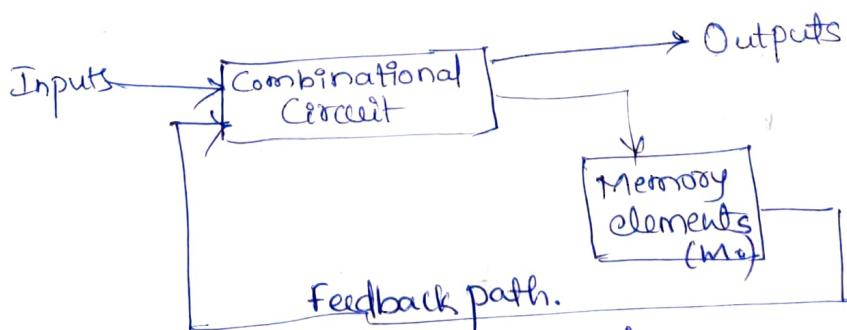
Classification

1) Synchronous → In which the changes in the state of memory elements are synchronized by a clock signal.
Example: Electronic oscillator, Calculators, Cell phones

2) Asynchronous → These circuits depends upon the order in which its input signals change and can be affected at any instant of time.
Example: Time delay devices

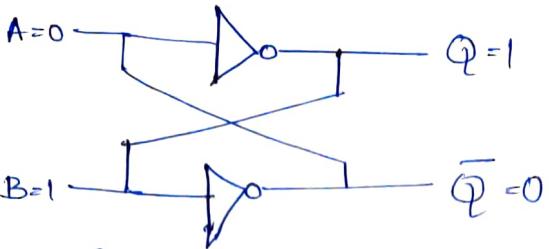
An asynchronous sequential circuit may be regarded as a combinational circuit with feedback. Because of the feedback among logic gates, an asynchronous sequential circuits may at times become unstable. The instability problem imposes many difficulties on the designer. Hence they are not commonly used as synchronous systems.

Block diagram of a sequential circuit



A feedback path is required for the circuit to have memory. The (M₀) memory elements themselves are also sequential circuits.

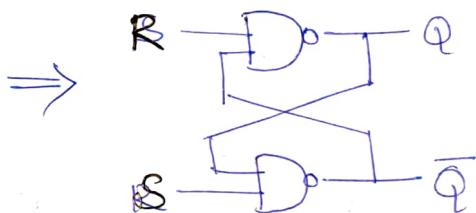
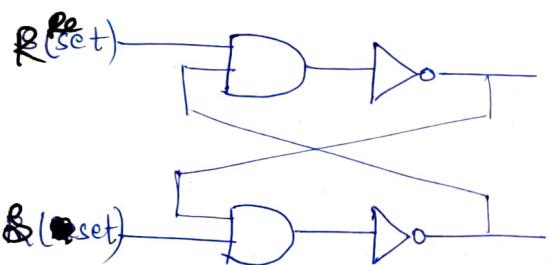
Bistable element



Output value of the each gate is stable, but what value will it be? There is no way to predict. If we could predict or set the value we would have a simple 1-bit memory element.

Latch [Event driven]

- * These are binary cells capable of storing one bit of information. This circuit has two outputs, one for the normal value and one for the complement value of the bit stored in it. Analog equivalence of latch is bistable multivibrator.
- 1. Basic SR latch with NAND gates



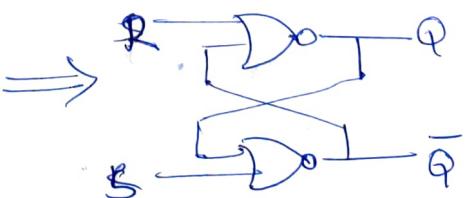
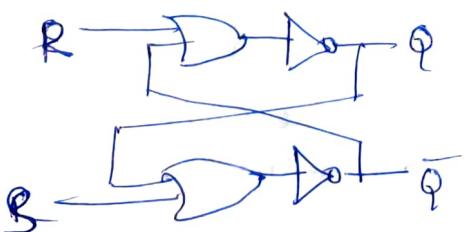
Function table

R	S	Q	\bar{Q}
1	0	0	1
0	0	1	1
0	1	1	0
1	1	1	0
1	0	0	1
1	1	1	0
0	0	0	1

Functional table

R*	S*	Q	\bar{Q}
0	0	Invalid	
0	1	1	0
1	0	0	1
1	1	Memory	

2. Basic SR latch with NOR gates



Function table

S	R	Q	\bar{Q}
1	0	1	0
0	0	1	0
0	1	0	1
0	0	0	1
1	1	0	0



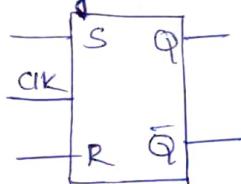
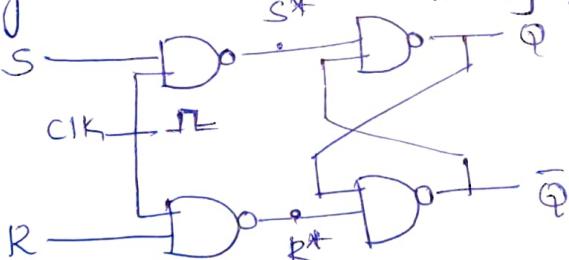
S	R	Q	\bar{Q}
1	0	1	0
0	0		
0	1		
1	1		

Drawbacks

1. Abruptly inputs are changing, there is no way to predict when the input is going to change. Outputs change immediately in response to changes in the inputs. So unable to predict the what output value will we get. So can't construct a simple memory element.

3) Created SR latch [clock driven]

The modification we are going to look at is adding an enable input to our SR latch. A clock signal is acting as gate. It controls the timing of changes in input.



CLK	S	R	Q	\bar{Q}
0	0	0		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

Characteristic table

Q	S	R	Q _{nti}
0	0	0	0
0	0	1	0
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 eq²

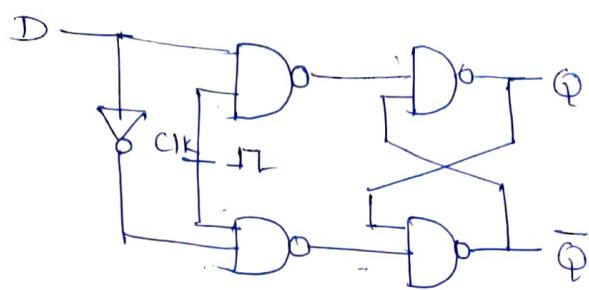
Q	SR	
	00	01
0	X	1
1	1	X

$$Q_{nti} = S + Q\bar{R}$$

Excitation table

Q	Q _{nti}	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

4) Gated D latch



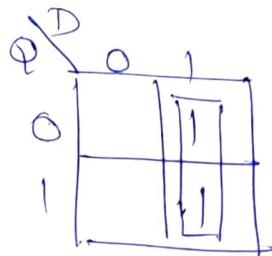
functional table

CLK	D	Q	\bar{Q}
1	0	0	1
1	1	1	0

characteristic table

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

characteristic eq $\frac{Q}{D}$



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

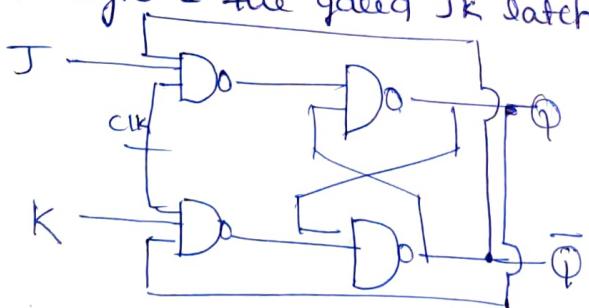
Excitation table

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

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

5) Gated JK latch

To avoid the invalid / forbidden input states of the gated SR latch even when S & R are both at logic "1" the gated JK latch is designed.



* Jack & Kilby
who invented this
latch combination.

J	Q
clk	
K	

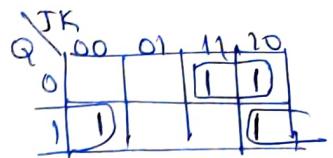
Functional table

C	I	K	Q	\bar{Q}
0	0	0	Memory	
1	0	0	Memory	
1	0	1	0	1
1	1	0	1	0
1	1	1	Toggle	

Characteristic table

Q	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

Characteristic eq?

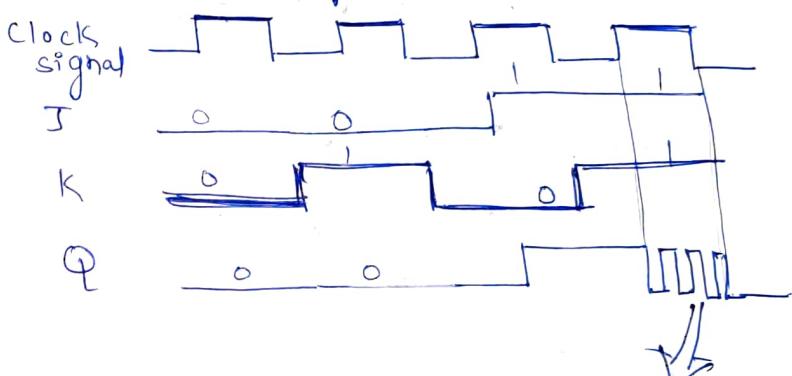


$$Q_{n+1} = \bar{Q}J + Q\bar{K}$$

Excitation table

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

Timing diagram



Drawbacks

1. The problem called race around condition occurs when both the inputs of JK latch are 1.

If the width of the clock pulse t_p is too long, the state of the latch will keep on changing from 0 to 1, 1 to 0, 0 to 1 and so on, and at the end of the clock pulse, its state will be uncertain.

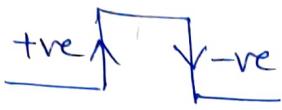
To avoid race around condition

1. $T_{1/2} < \text{Propagation delay of the circuit/gate}$
2. Master - Slave
3. Edge triggered.

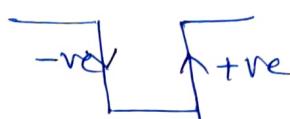
Edge triggering

Sensitive to the pulse transition of the clock function.

① Definition of clock pulse transition

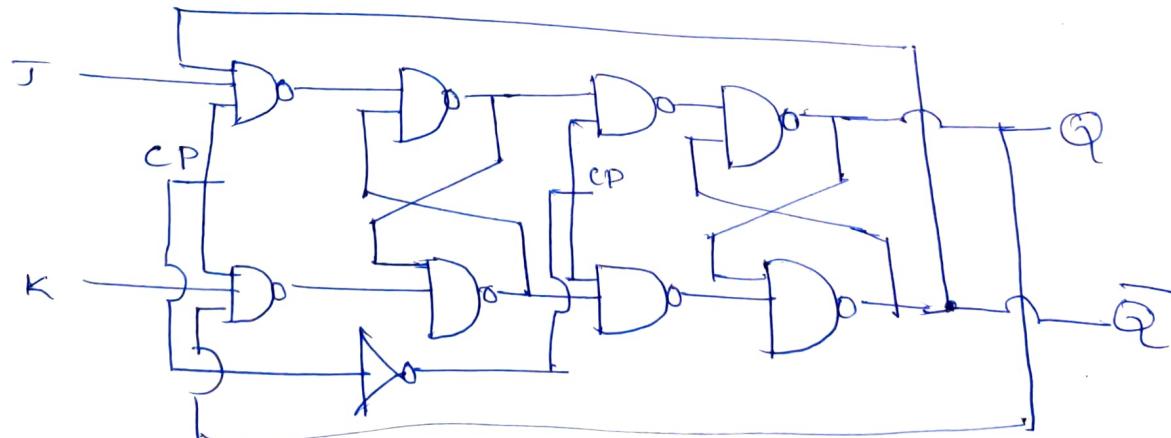
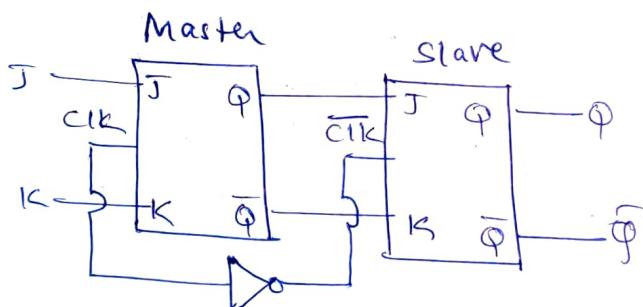
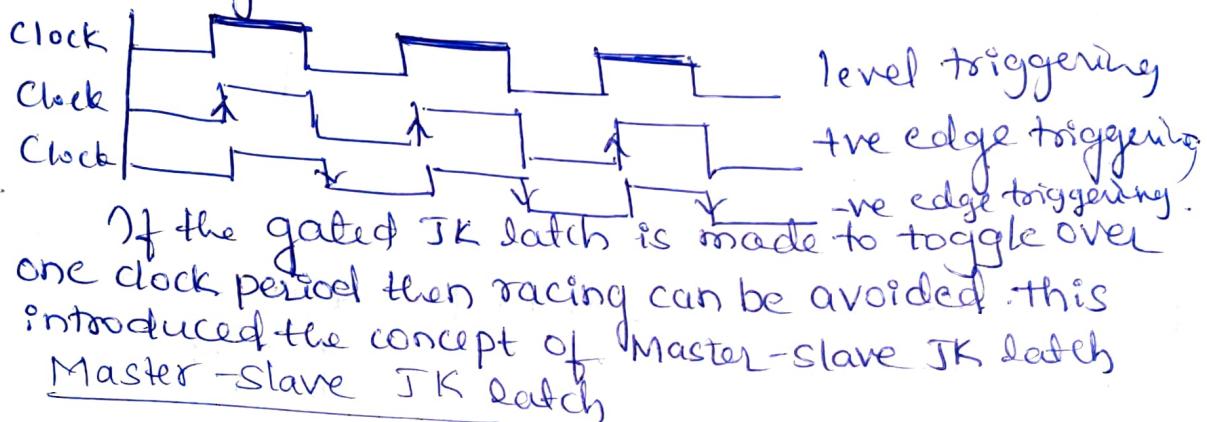


Positive pulse



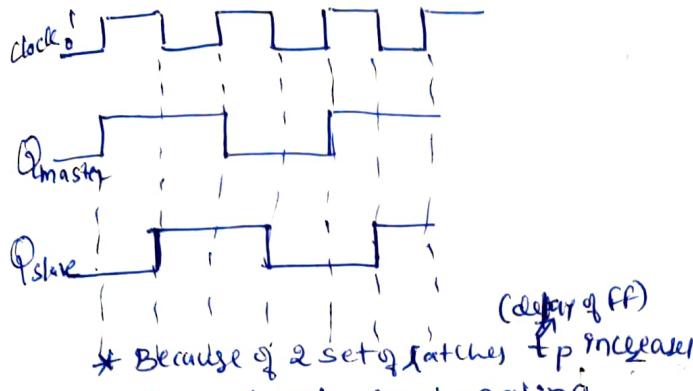
Negative pulse

Any change in the input at the enable clock at high will not effect the output. Only when the input changes occur at the transition will effect the output. Until the next pulse transition comes the output will not change.

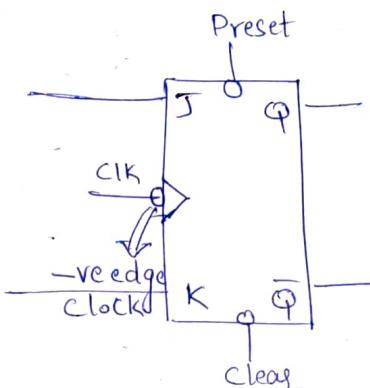
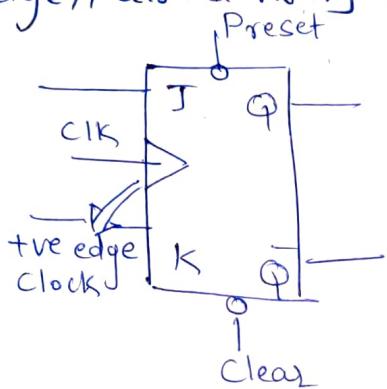


Drawbacks

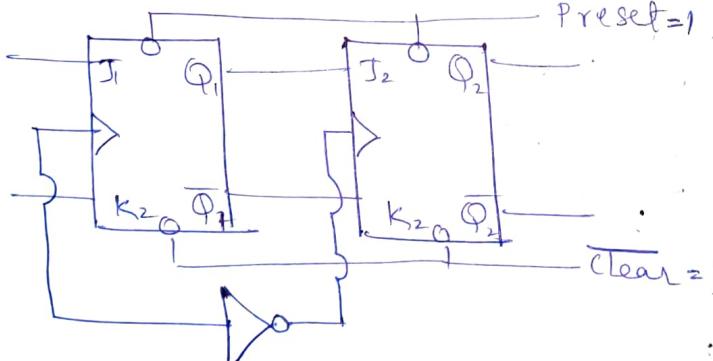
Timing diagram of MS-JK latch.



Flip-flops [Edge triggered] [Edge/Pulse driven]

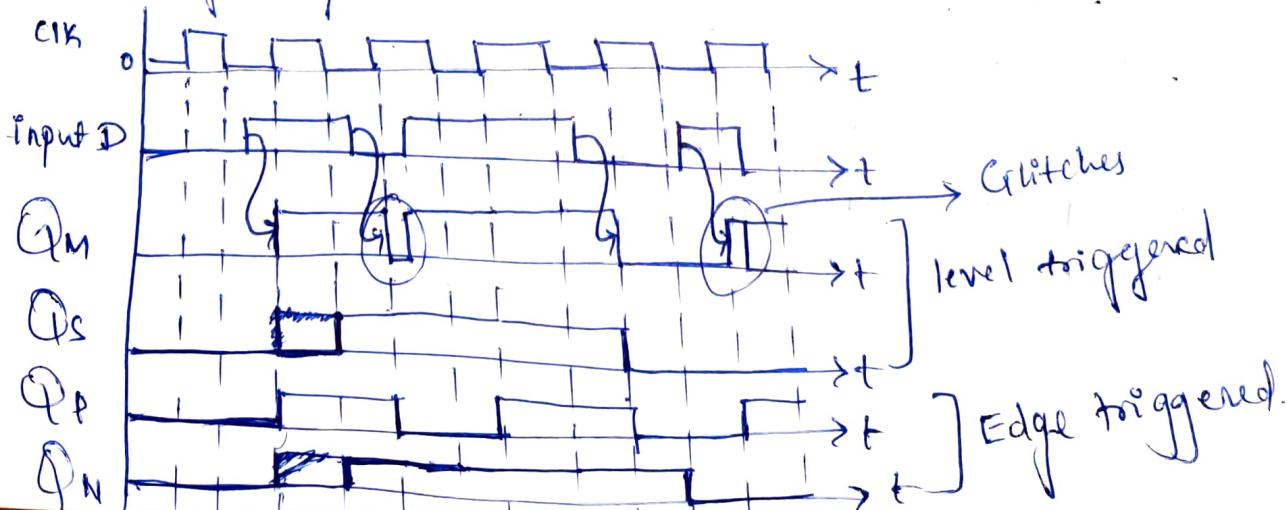


MS-JK Flip-flop



Asynchronous inputs		
Preset	Clear	Output
0	0	Invalid
0	1	1
1	0	0
1	1	Normal

Timing diagram shows the diff. b/w level & edge triggering



feature

1. When the clock pulse goes high the data at the J & K inputs are transmitted through the Master latch as normal. The adjoining slave latch however remains isolated since its clock input, clock pulse is low logic "0" due to the inversion by the inverter.

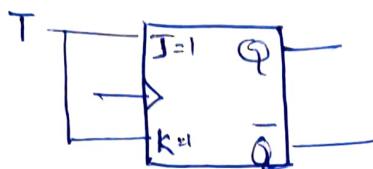
2. When the initial clock pulse returns low to "0", the Master becomes disabled and slave now becomes enabled.

Note:

- * Avoid the extra hardware
- * Once in a clock period the output can change

T flip flop [Toggling flip flop]

function table



T	Q	\bar{Q}
0	Memory	
1	Toggle	

$$J=1, K=1 \Rightarrow T=1$$

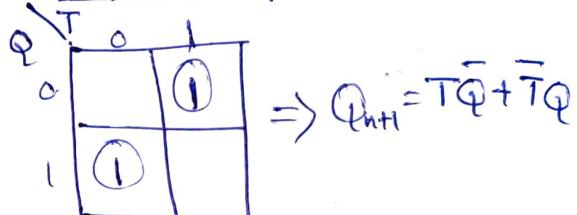
T	Q	\bar{Q}_{+}
0	0	1
1	1	0
0	1	0
1	0	1

Excitation table

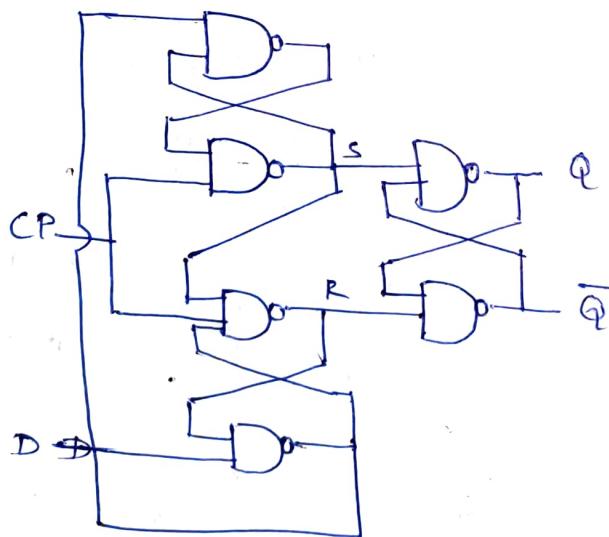
Q	Q_{n+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

transition table

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



Edge triggered D flip flop



CP	D	S	R	Q	\bar{Q}
0	0	1	1	Memory	
0	1	1	1	Memory	
1	0	1	0	0	1
1	1	0	1	1	0

(Hold)

(Reset)

(Set)

When the input clock pulse makes a positive going transition, the value of D is transferred to Q. Changes in D when CP is maintained at a steady value do not affect Q. Moreover, a negative pulse transition does not affect the output, nor does it when $CP = 0$. Hence the edge triggered flip-flop eliminates any feedback problems in sequential circuits just as a master-slave flip-flop does. The set up time & hold time must be taken into consideration.

Counters

Definition

A counter circuit is usually constructed of a number of flip-flops connected in cascade. Counters are a very widely used component in digital circuits, and are manufactured as separate integrated circuits and incorporated as a part of large integrated circuit.

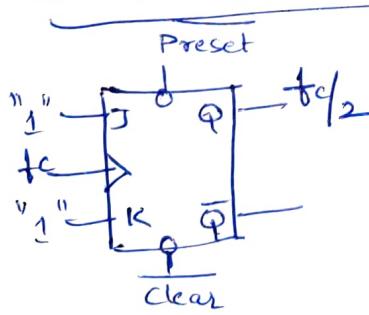
Counters

Asynchronous
: Changing state bits are used as clocks to subsequent state flip-flops.

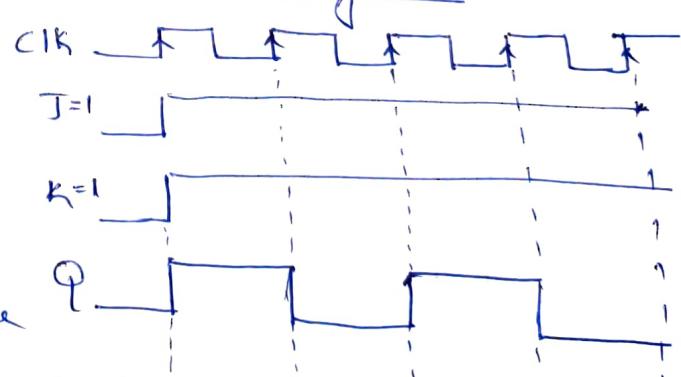
Synchronous
: All state bits change under control of a single clock

Asynchronous Counters / Ripple Counters

Divide by-2 Circuit



Timing diagram

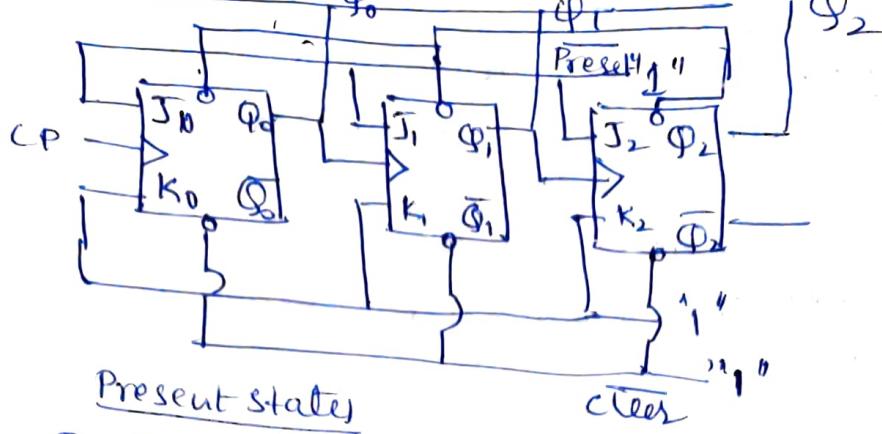


If you observe the

Output Q compare with the input clock signal, the output is exactly dividing the clock frequency by 2.

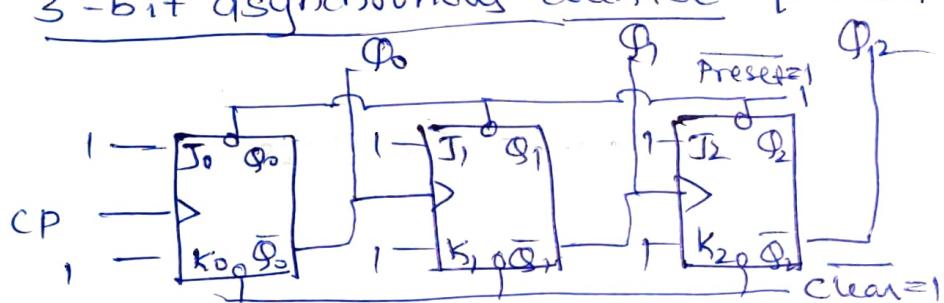
Mod 2^n counter requires n-flipflops.

3 bit asynchronous counter [PUP counter]



Q_0	Q_1	Q_2	clock cycles
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

3-bit asynchronous counter [DOWN Counter]



Q_0	Q_1	Q_2	clock cycles
1	1	1	0
1	1	0	1
1	0	1	2
1	0	0	3
0	1	1	4
0	1	0	5
0	0	1	6
0	0	0	7

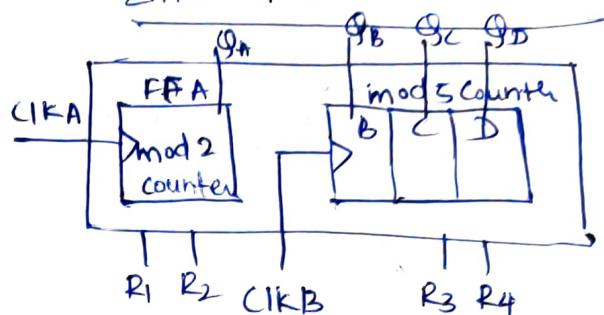
Mod-10 Counter [7490 IC]

BCD Counter

IC 7490

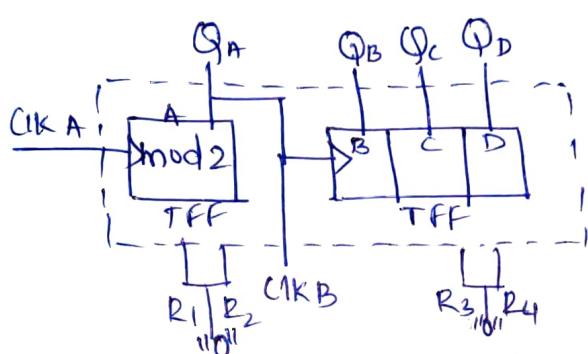


Internal architecture



Mod 2 Counter followed by mod 5 Counter.

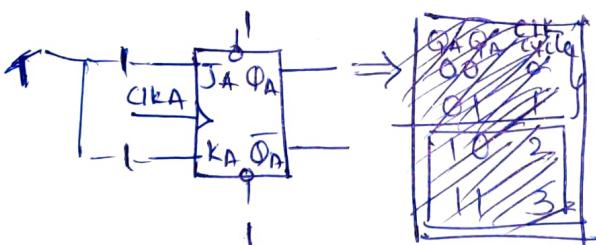
Mod-10 Counter using IC 7490



R₁, R₂ \Rightarrow high] , Reset to "0"
R₃, R₄ \Rightarrow low]

R₃, R₄ \Rightarrow high } Set to max. count
R₁, R₂ \Rightarrow low } 1001

Mod 2 counter i.e. Only one flip flop (FFA) is required.



Mod 2 Counter so only 2 counting states will come into picture.

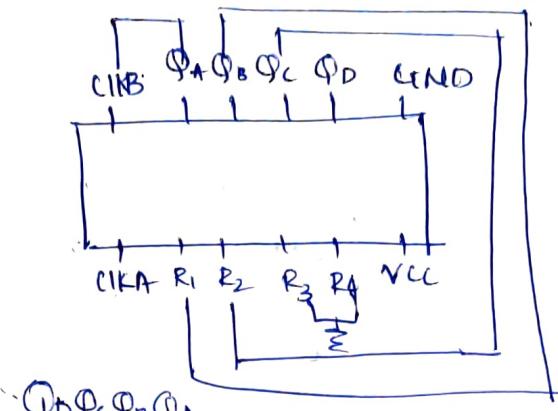
When it is 1st CLK cycle, it gives me 0000.

- * The Output of Mod-2 is externally connected to the input B which is the CLK input of the internal MOD-5 counter.

- * Hence Q_A toggles on every falling edge of CLK input whereas the output Q_D Q_C Q_B of mod 5 Counter will increment from 000 to 100 on low going change of Q_A output.

Q _D	Q _C	Q _B	Q _A	CK cycle
0	0	0	0	0 cycle
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

Divide by -6 Counter using 7490



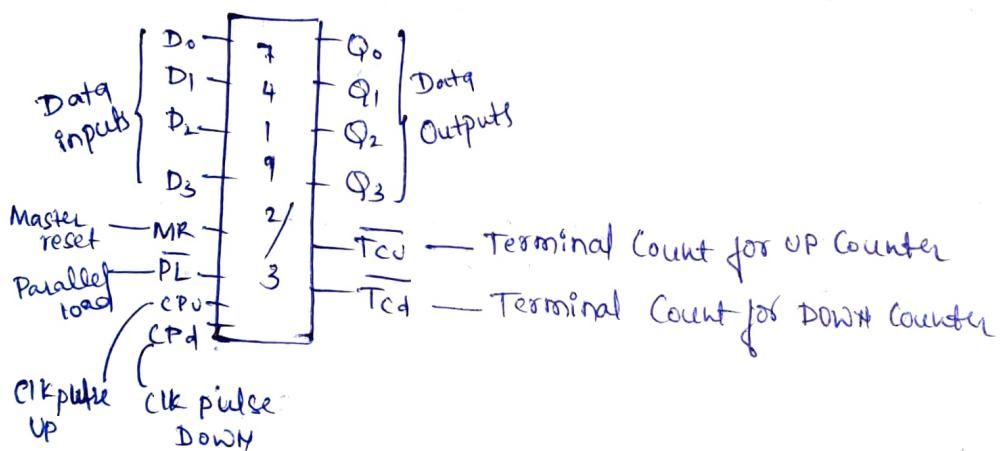
$Q_0 Q_1 Q_2 Q_3 Q_4$
0 1 1 0

Reset $\Rightarrow R_1, R_2 = 1$] Reset to "0"
 R_3, R_4 (Set)

Presetable UP/DOWN Counters [IC 74192/3]

IC 74192 — Decade Counter [0 - 9]

IC 74193 — Binary Counter [0 - 15]



① (3-6) Counter

Load 3 (0011) into D₀ D₁ D₂ D₃ and terminate at Count 6.

CPU \rightarrow Clock

CPD \rightarrow "1"

MR \rightarrow 0

PL \rightarrow 1

When Counter is counting 0 to 8 the $\overline{TCU} = 1$, When it reaches maximum count either 9 or 15 $\overline{TCU} = 0$.

① When $\overline{TCU} = 0$ load D₀ D₁ D₂ D₃ = 0011

$\overline{Q_0 Q_1 Q_2 Q_3}$

0011 3

0100 4

0101 8

0110 6

0111 —



$$\overline{PL} = 0 \quad \overline{TCU} = 0$$

② (6-3) Counter

Load 6 (0110) into $D_0 D_1 D_2 D_3$ and terminate at count 3.

$CP_d \rightarrow C_{IK}$

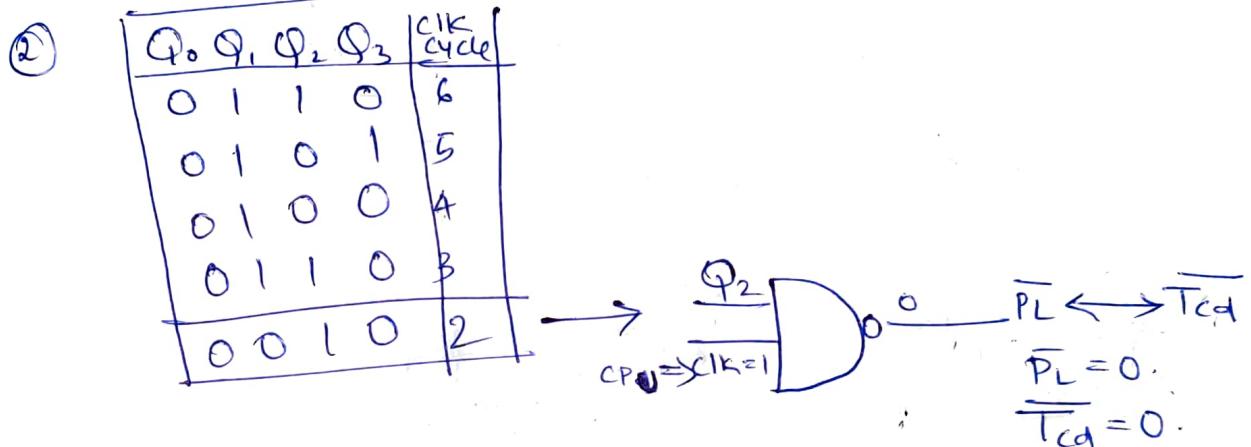
$CP_u \rightarrow "1"$

$MR = 0$

$\overline{PL} = 1$

When counter is counting 9 to 1, the $\overline{T_{cd}} = 1$, when it reaches minimum count 0 (both in 192/193) $\overline{T_{cd}} = 0$.

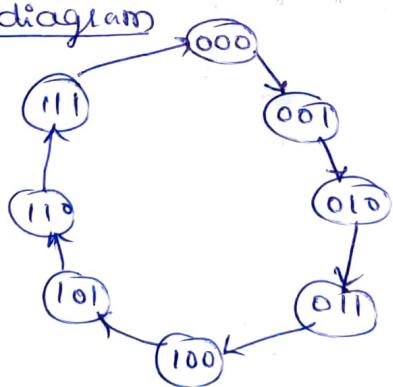
① When $\overline{T_{cd}} = 0$, load $D_0 D_1 D_2 D_3 = 0110$



Synchronous Counter design

3 bit synchronous UP counter

State diagram



No. of flip flops required $\Rightarrow 2^N = 2^3 = 8$ flip flops.

Using JK flip flops.

<u>Present State</u>	<u>Next State</u>	<u>Present Inputs</u>
$Q_2\ Q_1\ Q_0$	$Q_2\ Q_1\ Q_0$	$J_2\ K_2\ J_1\ K_1\ J_0\ K_0$
0 0 0	0 0 1	0X 0X 1X
0 0 1	0 1 0	0X 1X X1
0 1 0	0 1 1	0X X0 1X
0 1 1	1 0 0	1X X1 X1
1 0 0	1 0 1	X0 0X 1X
1 0 1	1 1 0	X0 1X X1
1 1 0	1 1 1	X0 X0 1X
1 1 1	0 0 0	X1 X1 X1

Use K-map to determine state eqⁿ

		$J_0 \Rightarrow$				
		$Q_1\ Q_2$	00	01	11	10
Q_0		0	1	1	1	1
0		1	X	X	X	X
1		1	X	X	X	X

$J_0 = 1$

		$K_0 \Rightarrow$				
		$Q_1\ Q_2$	00	01	11	10
Q_0		0	X	1	1	X
0		1	1	X	X	1
1		1	X	X	1	1

$K_0 = 1$

		$J_1 \Rightarrow$				
		$Q_1\ Q_2$	00	01	11	10
Q_0		0	0	0	X	X
0		1	0	1	X	X
1		1	1	X	X	1

$J_1 = Q_0$

		$K_1 \Rightarrow$				
		$Q_1\ Q_2$	00	01	11	10
Q_0		0	X	X	0	0
0		1	X	X	1	1
1		1	X	X	1	1

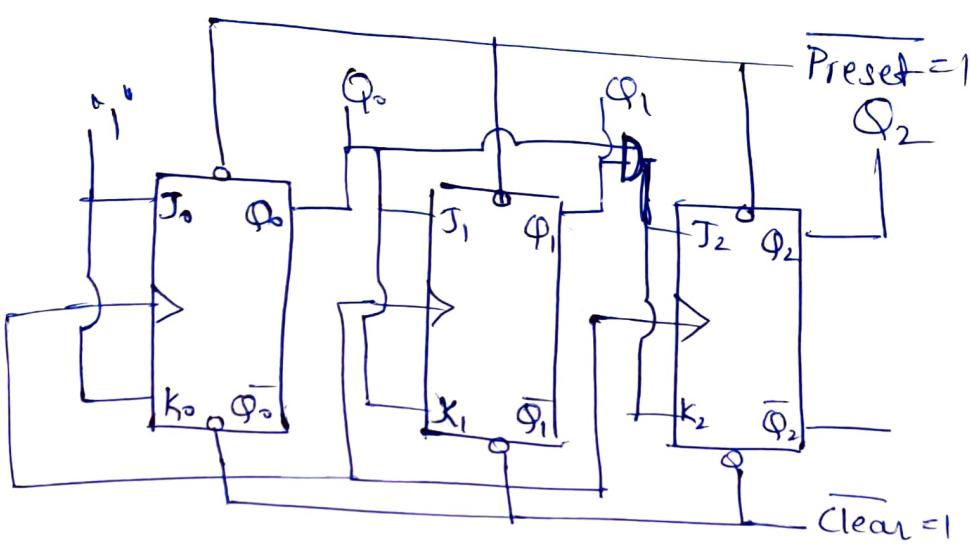
$K_1 = Q_0$

		$J_2 \Rightarrow$				
		$Q_1\ Q_2$	00	01	11	10
Q_0		0	0	X	1	0
0		1	0	X	(X)	0
1		1	X	(X)	1	X

$J_2 = Q_0 Q_1$

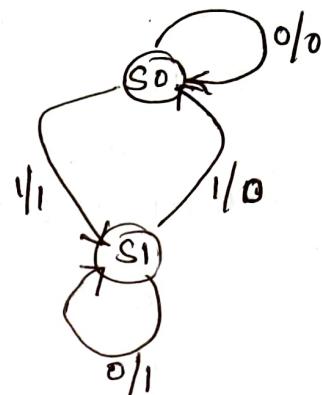
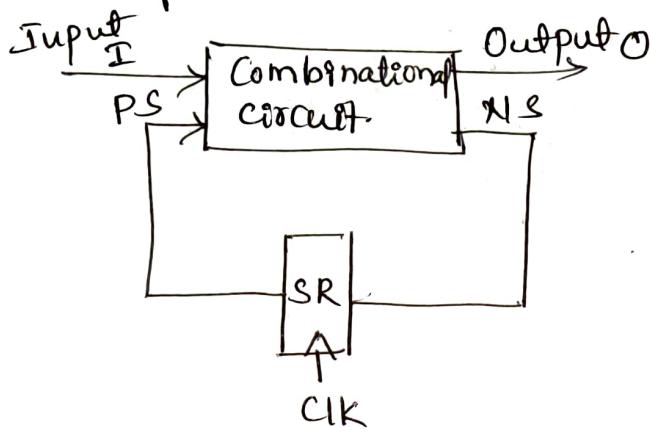
		$K_2 \Rightarrow$				
		$Q_1\ Q_2$	00	01	11	10
Q_0		0	X	0	0	X
0		1	X	0	1	X
1		1	X	0	(1)	X

$K_2 = Q_0 Q_1$



Mealy machines

1. Mealy machine is an FSM, whose output depends on the present state as well as the present input.
(Both output and the next state depends on the present input and the present state)
2. Requires more hardware requirements.



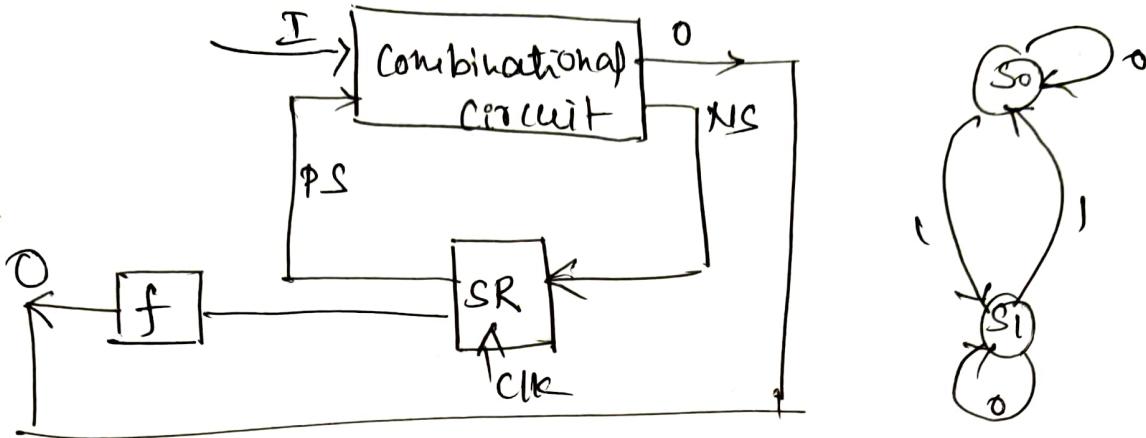
Moore Machines

1. Here present output is not a function of present inputs but is a function of past inputs. The next state is a function of both the present input and the present state.
2. In this case the output is not associated with the transition but are associated with the state.
3. The output 'O' is a function of present state and independent of present input 'I'.

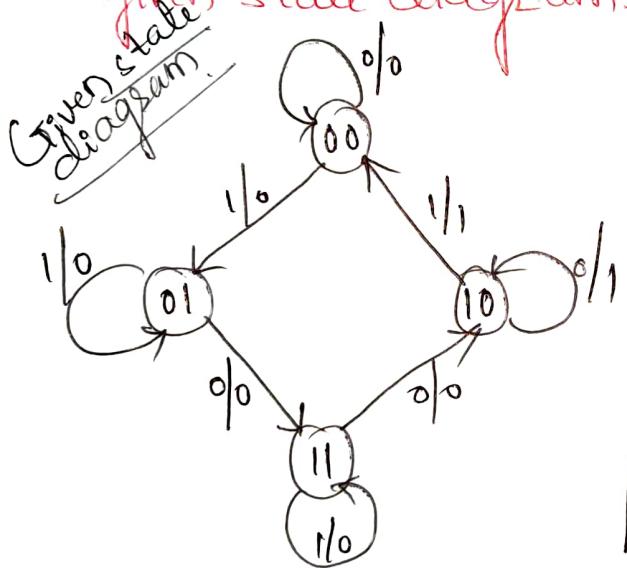
(In fact present input I influences the next state and that's how it is going to influence the output but there is a time lag between the input and output. The present inputs are going to influence the outputs that are going to come after the next clock.)

Finite state machines

1. A finite state machine is a mathematical abstraction used to design algorithms.
2. A state machine will read a series of inputs. When it reads an input, it will switch to a different state. Each state specifies which state to which state to switch to, for a given input.
3. State machines can be used to model a wide variety of systems including;
 - a. User interfaces, with typed input, mouse clicks etc
 - b. The state of a spacecraft, including which valves are open and closed, the levels of fuel and oxygen etc; and
 - c. The sequential patterns in DNA and what they mean.
 - d. conversations, in which, for example, the meaning of a word "it" depends on the history of things that have been said;
4. Clocked sequential circuits are implemented in two different ways.
 - a. Mealy Machine
 - b. Moore machine.



* Construction of logic diagram of a circuit from the given state diagram.



state table

P. S. A B	Input X	N. S		Output Y
		A ⁺	B ^T	
0 0	0	0	0	0
0 0	1	0	1	0
0 1	0	1	1	0
0 1	1	0	1	0
1 0	0	1	0	1
1 0	1	0	0	1
1 1	0	1	0	0
1 1	1	1	1	0

State excitation table

Present state	Present input	Next state	flip flop inputs	Output
A B	X	A ⁺ B ^T	J _A K _A J _B K _B	Y
0 0	0	0 0	0 X 0 X	0
0 0	1	0 1	0 X 1 X	0
0 1	0	1 1	1 X X 0	0
0 1	1	0 1	0 X X 0	0
1 0	0	1 0	X 0 0 X	0
1 0	1	0 0	X 1 0 X	1
1 1	0	1 0	X 0 X 1	0
1 1	1	1 1	X 0 X 0	0

		for J_A			
		00	01	11	10
A	0	0	0	0	1
	1	X	X	X	X

$$J_A = BX'$$

		for K_A			
		00	01	11	10
A	0	X	X	X	X
	1	0	1	0	0

$$K_A = B'x$$

		for Y			
		00	01	11	10
A	0	0	0	0	0
	1	1	1	0	0

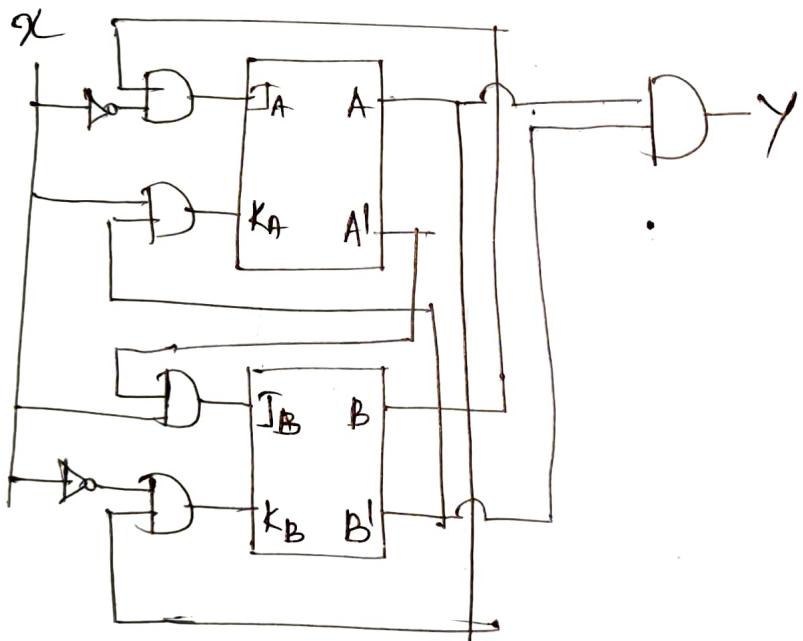
$$Y = AB'$$

		for J_B			
		00	01	11	10
A	0	0	0	X	X
	1	0	0	X	X

$$J_B = A'x$$

		for K_B			
		00	01	11	10
A	0	X	X	0	0
	1	X	X	0	0

$$K_B = AxB'$$



Construct the state diagram & logic diagram from given state eqⁿ.

$$A(t+1) = A(t).X + B(t).x \quad (1)$$

$$B(t+1) = \bar{A}(t).X \quad (2)$$

$$Y = (A(t) + B(t)).\bar{x} \quad (3)$$

Designing using D flipflop

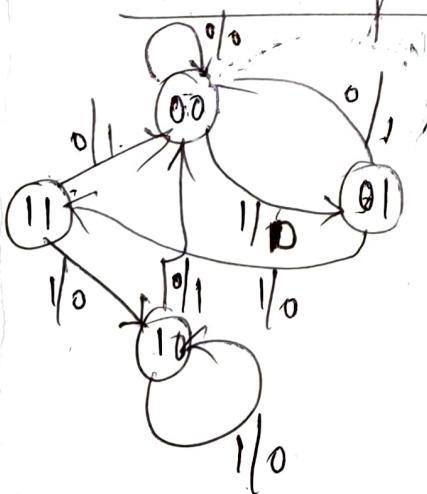
P.S.	P.I.	N.S	O/P	FF I/Ps	
A	B	X	A ^t B ^t	Y	D _A D _B
0	0	0	0 0	0	0 0
0	0	1	0 1	0	0 1
0	1	0	0 0	1	0 0
0	1	1	1 0	0	1 1
1	0	0	0 0	1	0 0
1	0	1	1 0	0	1 0
1	1	0	0 0	1	0 0
1	1	1	1 0	0	1 0

Characteristic eqⁿ

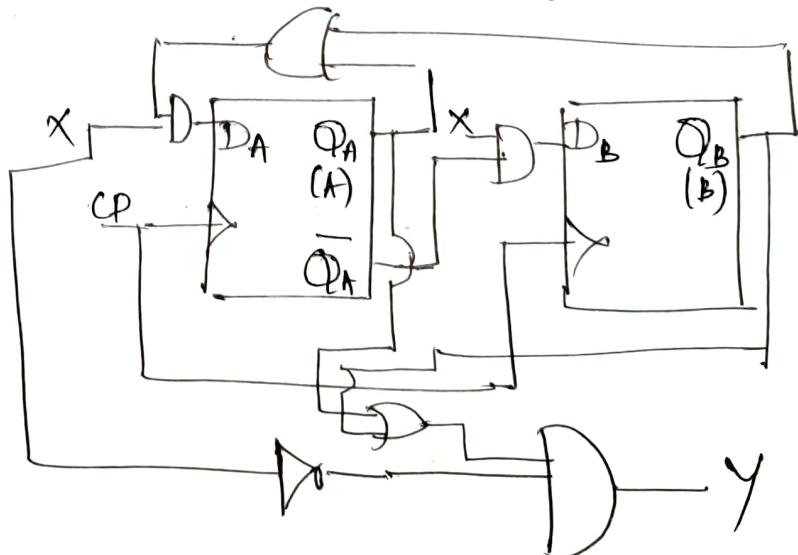
$$D_A = A(t+1) [A^t]$$

$$D_B = B(t+1) [B^t]$$

State diagram



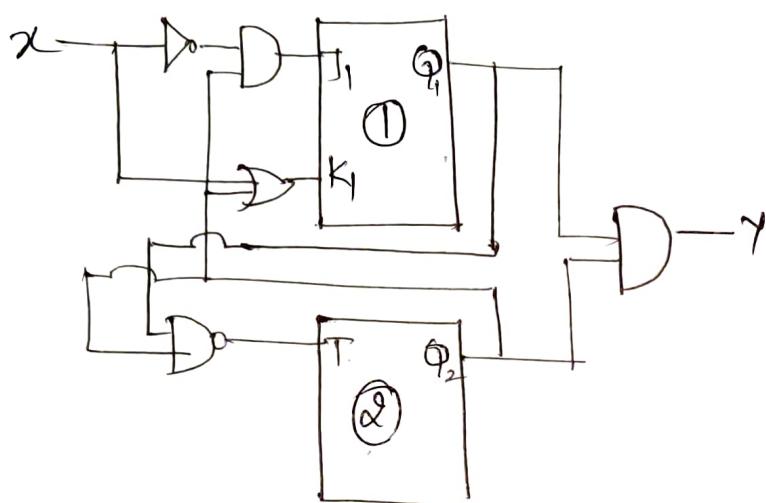
Logic diagram



(4)

Construction of state diagram from given logic diagram

Logic diagram



state eq[?]

$$J_1 = x' Q_2$$

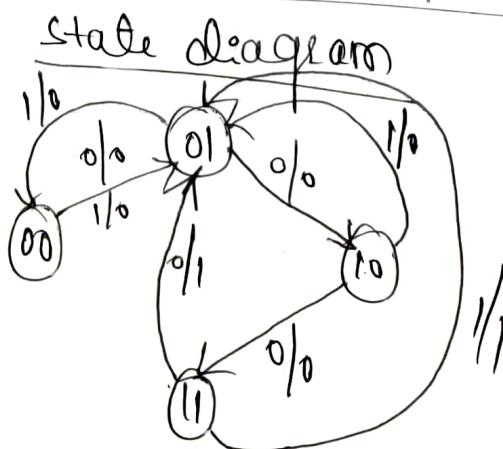
$$K_1 = (x + Q_2)$$

$$T = (Q_2 Q_1)^1$$

$$Y = Q_1 Q_2$$

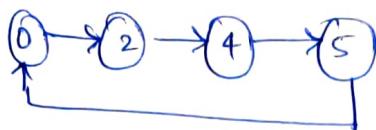
State excitation table

P.S	P-I	F-F i/p's	N.S	F.F i/p's	N.S	O/P
$Q_1 Q_2$	x	$J_1 \ K_1$	Q_1^+		Q_2^+	Y
0 0	0	0 0	0	1	1	0
0 0	1	0 1	0	1	1	0
0 1	0	1 1	1	1	0	0
0 1	1	0 1	0	1	0	0
1 0	0	0 0	1	1	0	0
1 0	1	0 1	0	1	1	0
1 1	0	1 1	0	0	1	1
1 1	1	0 1	0	0	1	1



Sequence generator

Using T-flip flop for the given sequence



It requires 3 flip flops and are named as 0, 1, 2.

Excitation table

Present state	Next state	flip flop inputs
$Q_0 Q_1 Q_2$	$Q_0^+ Q_1^+ Q_2^+$	$T_0 T_1 T_2$
0 0 0	0 1 0	0 1 0
0 0 1	x x x	x x x
0 1 0	1 0 0	1 1 0
0 1 1	x x x	x x x
1 0 0	1 0 1	0 0 1
1 0 1	0 0 0	1 0 1
1 1 0	x x x	x x x
1 1 1	x x x	x x x

Checking for lock out condition

$$Q_0 Q_1 Q_2 = 001$$

$Q_0 Q_1 Q_2$	$Q_0^+ Q_1^+ Q_2^+$	$T_0 T_1 T_2$
0 0 1	1 1 1	1 1 0
1 1 1	0 1 0	1 0 1

Here 010 = state 2
(used state) so no lock out

for T_A

Q_0	Q_1	Q_2	00	01	11	10
0	0	1	x	x	1	
1	1	0	x	x	x	x

$$T_A = Q_2 + Q_1$$

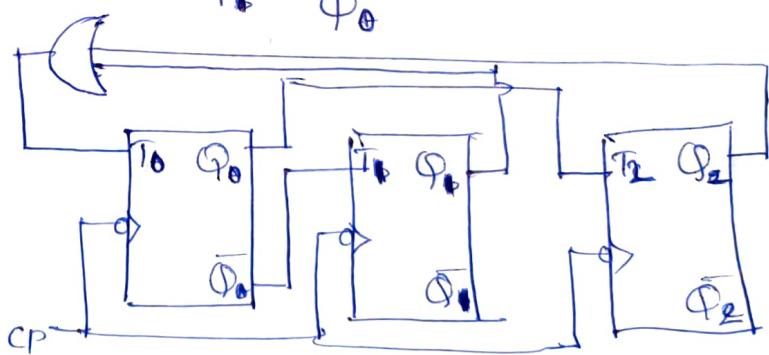
for T_B

Q_0	Q_1	Q_2	00	01	11	10
0	0	1	1	x	x	1
1	1	0	x	x	x	x

$$T_B = \bar{Q}_0$$

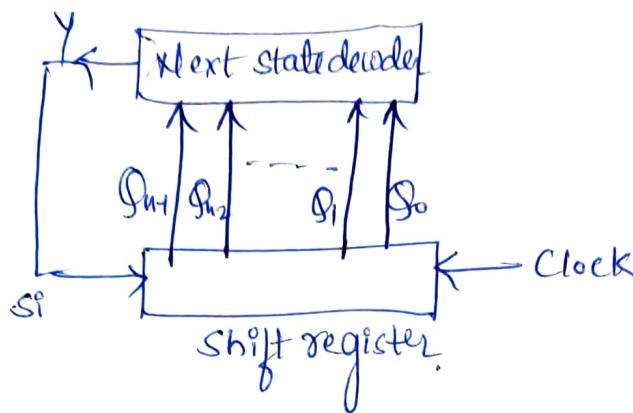
for T_C

Q_0	Q_1	Q_2	00	01	11	10
0	0	1	x	x	x	x
1	1	0	1	x	x	x

$$T_C = Q_0$$


sequence generator

using shift registers



Ex: Design a sequence generator to generate the sequence ---1101011---

step ① No. of flipflop required

$$S \leq 2^N - 1$$

length of sequence

$$7 \leq 2^3 - 1$$

$$7 \leq 7$$

$$N=3$$

However, it is not guaranteed to lead to a solution.

DFF outputs		clk pulse
Q_2	Q_1, Q_0	
1	1 1	1
1	1 1	2
0	1 1	3
1	0 1	4
0	1 0	5
1	0 1	6
1	1 0	7

[None of the states are repeated within 7 clk pulse.]

but here we are not getting 7 distinct states to avoid this problem we are going for 4 flipflop design.

No. of clock pulses	D-FF outputs				Serial input y
	Q_3	Q_2	Q_1	Q_0	
1	1	1	1	0	1
2	1	1	1	1	0
3	0	1	1	1	1
4	1	0	1	1	0
5	0	1	0	1	1
6	1	0	1	0	1
7	1	1	0	1	1
8	1	1	1	0	1
9	1	1	1	1	0
10	0	1	1	1	1
11	1	0	1	1	0
12	0	1	0	1	0
13	0	1	0	0	1
14	0	1	0	0	0

For y

$Q_1 Q_0$	00	01	11	10
$Q_3 Q_2$	x	x	x	x
00	x	1	1	x
01	x	1	0	0
11	x	x	1	0
10	x	x	1	1

$y = \bar{Q}_1 + \bar{Q}_3 + Q_0$

$Q_1 Q_0$	00	01	11	10
$Q_3 Q_2$	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	1	0	0	1
10	0	1	1	1

$$Y = Q_2 \bar{Q}_1 Q_0 + Q_3 Q_2 Q_0 +$$

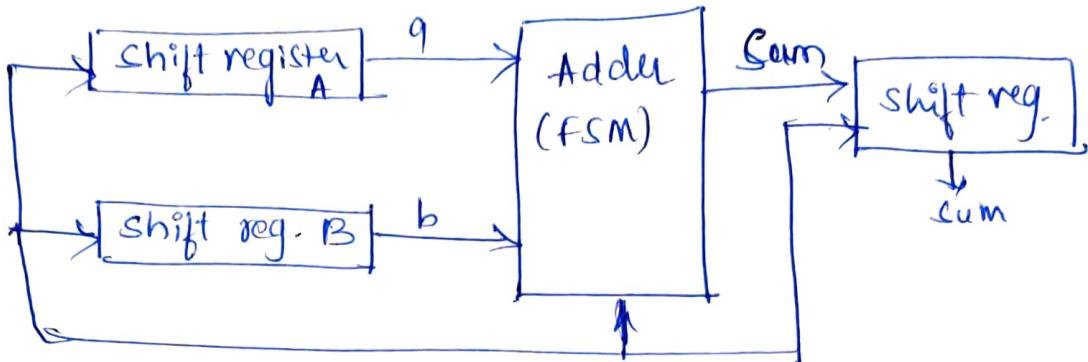
$$Q_3 Q_1 \bar{Q}_0$$

$$= Q_0 (Q_2 \bar{Q}_1 + Q_3 Q_2) +$$

$$Q_3 Q_1 \bar{Q}_0$$

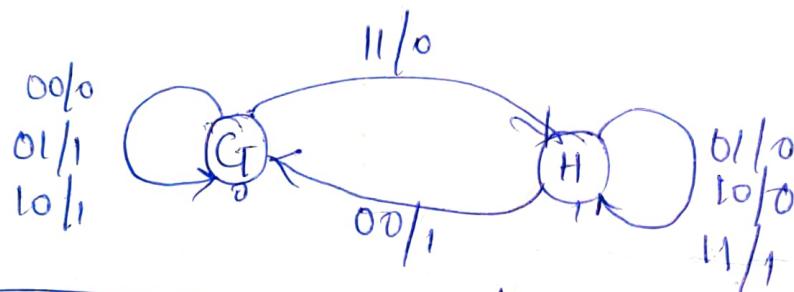
Serial adder

- * Speed is not considerable
- * Cost effective



G → state that the carry-in is '0'

H → state that the carry-in is '1'



Present state	Next (Y) state	Output (S)
G	(ab) 00 01 10 11	(ab) 00 01 10 11
H	G H H H	1 0 0 1

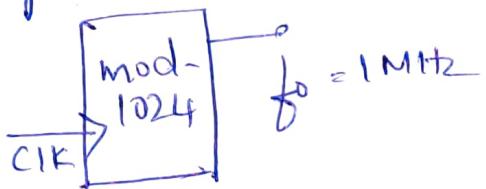
P.S Cin	Present IPs a b		N.S Cout	Out S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

$$Y = ab + aY + bY$$

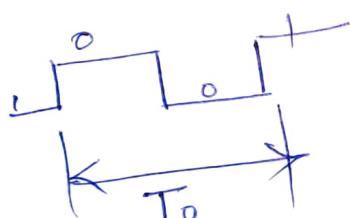
$$S =$$

* 4 bit mod-16 ripple counter uses JK flip flops. If the propagation delay of each flip flop is 50 nsec, the max clock frequency that can be used is equal to

A pulse train with a frequency of 1 MHz is counted using 1024 ripple counter built with JK flip flops. For proper operation of the counter, the maximum permissible propagation delay per flip flop stage is — n sec.



$$2^N = M \text{ (1024 states)} \Rightarrow N = 10.$$



$$T_0 = \frac{1}{f_0} = \frac{1}{1M} = 1 \mu\text{sec}$$

$$\text{One complete cycle} = T_0 = t_{1 \rightarrow 0} + t_{0 \rightarrow 1}$$

$$= 10T + 10T$$

$$= 20T$$

$$20T = 1 \mu\text{sec}$$

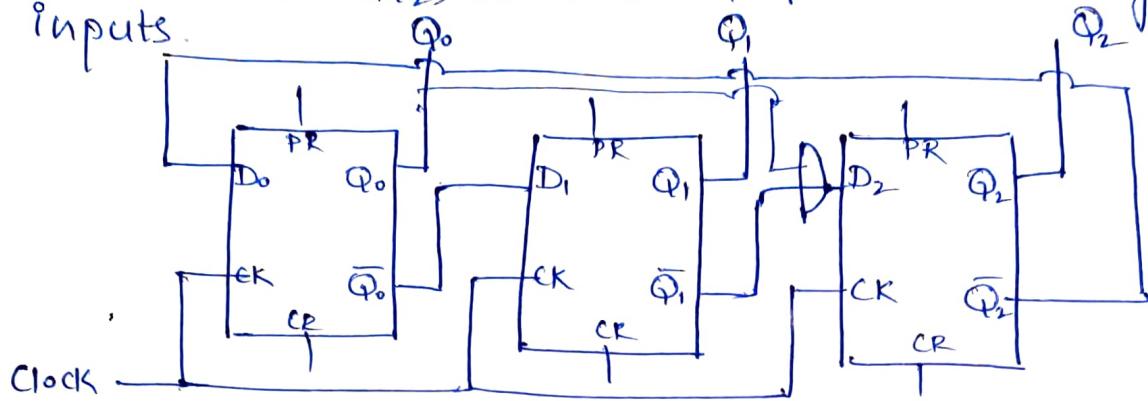
$$T = \frac{1}{20} \mu\text{sec}$$

$$= 0.05 \mu\text{sec}$$

$$\boxed{\underline{T = 50 \text{nsec}}}$$

Example 1.

A sequence generator is shown in figure. The Counter status (Q_0, Q_1, Q_2) is initialized to 010 using preset/clear inputs.



The clock has a period of 50 ns and transitions take place at the rising clock edge.

(a) Give the sequence generated at Q_0 till repeats.

(b) What is the repetition rate of the generated sequence?

Sol:

Present State	Next State	
$Q_0 \ Q_1 \ Q_2$	$Q_0^+ \ Q_1^+ \ Q_2^+$	
0 1 0	1 1 0	$Q_0^+ = D_0 \Rightarrow \bar{Q}_2$
1 1 0	1 0 0	$Q_1^+ = D_1 \Rightarrow \bar{Q}_0$
1 0 0	1 0 1	$Q_2^+ = D_2 \Rightarrow Q_0, \bar{Q}_1$
1 0 1	0 0 1	
0 0 1	0 1 0	
→ 0 1 0	1 1 0	

→ 01110, 01110.....

After every 5 clk pulses sequence repeats.

$$\text{Duration} = ST$$

$$= 5 \times 50 \text{ ns}$$

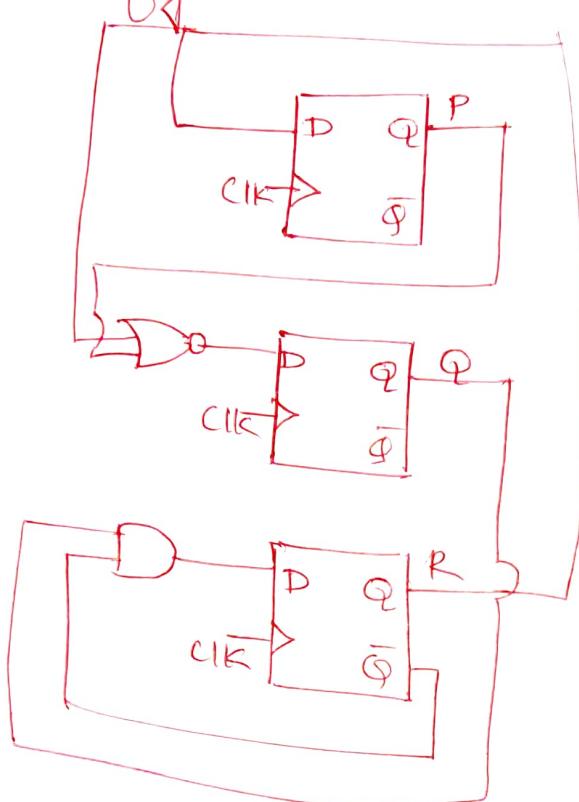
$$= 250 \text{ ns}$$

$$\text{Repetition rate is } \left(\frac{1}{ST} \right) = \left(\frac{1}{250 \text{ ns}} \right) = 4 \text{ MHz}$$

Numericals on sequential Circuits

* Consider the following circuit involving 3 D-type flip flops used in a certain type of counter configuration.

[Gate CS 2011]



If at some instance prior to occurrence of the clock edge P & Q + R have a value 0, 1, 0 respectively what shall be the value of PQR after the clock edge.

$$P^+ (\text{N.S of } P) = R$$

$$Q^+ (\text{N.S of } Q) = (P+R)^1$$

$$R^+ (\text{N.S of } R) = Q \cdot R^1$$

Initial states of P, Q, R are 0, 1, 0

$$P^+ = 0$$

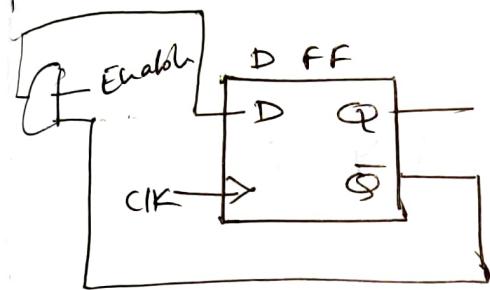
$$Q^+ = (0+0)^1 = 1$$

$$R^+ = 1 \cdot (1) = 1$$

$P^+ Q^+ R^+ = 011$

* An SR flip-flop can be converted into a T flip-flop by connecting — to \bar{Q} and — to Q .
 S to \bar{Q} & R to Q .

* A switch tail ring counter is made by using a single D flip-flop. The resulting circuit is a [Gate 1995 ECE].



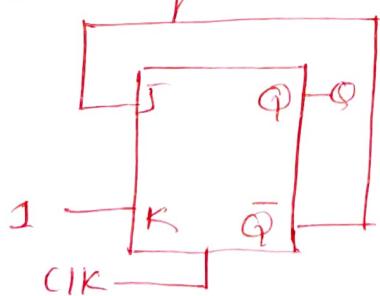
When Enable = 1

P.S	P.I	N.S
Q	$D = \bar{Q}$	Q^+
0	1	1
1	0	0

Toggle when Enable=1

so, Resulting circuit is T-FF

* In a JK flip flop, we have $J = Q'$ and $K = 1$. Assume the flip flop was initially cleared and then clocked for 6 pulses, the sequence at the Q output will be [Gate 1997, ECE]

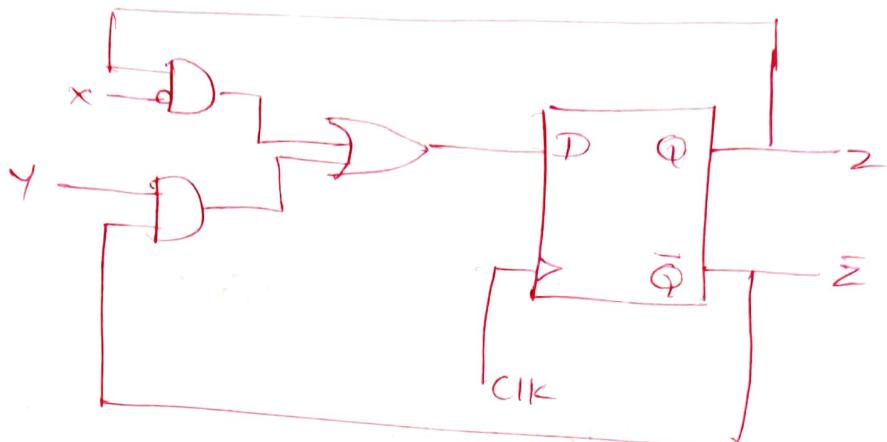


P. S	FF1111	N.S
Q	J K	Q^+
1	Q' 1	Q^+
2	0 1	0
3	1 1	1
4	0 1	0
5	1 1	1
6	0 1	0

"010101"

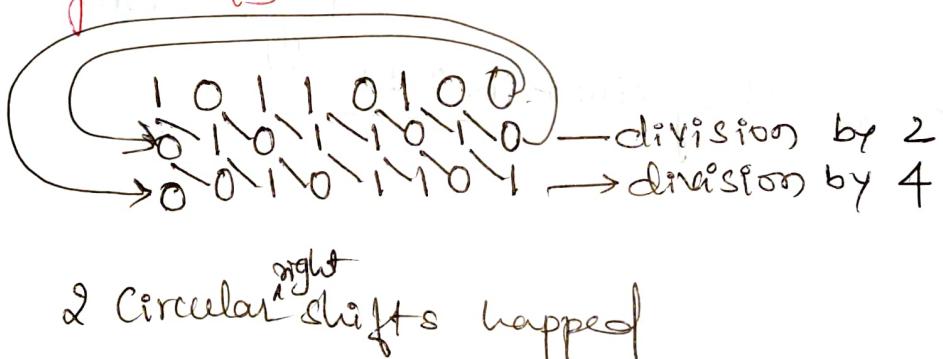
Q sequence after 6 pulses.

* A sequential circuit using D flipflop and logic gate is shown in figure, where X and Y are the inputs and Z is output. The Circuit is

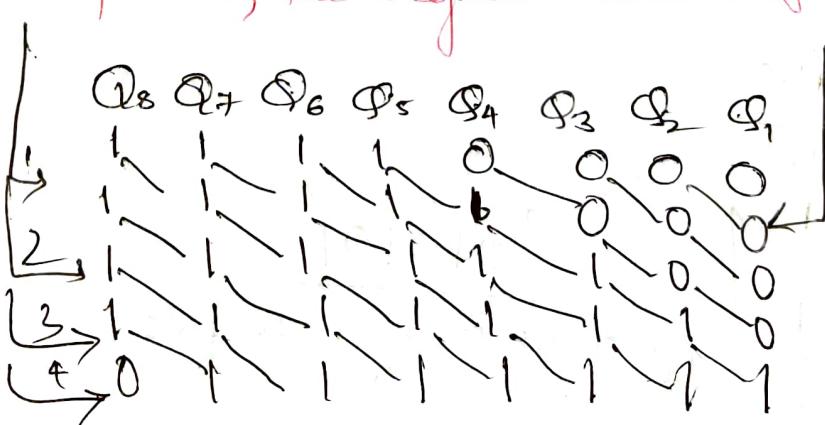


JK Flipflop with inputs $X = K$ & $Y = J$.

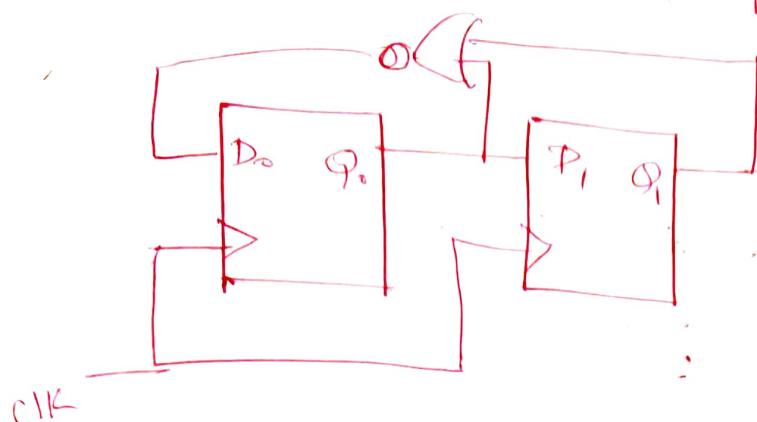
* Number of circular right shift operations required to divide the value 10110100 by 4 using shift register is



* The initial state of 8 bit shift right register is 11110000. The data 10110111 needs to be serially loaded (LSB first) into this register. After 4 clock pulses, the register contains



* For the circuit shown the sequence at Q_1, Q_0 is 0, 1



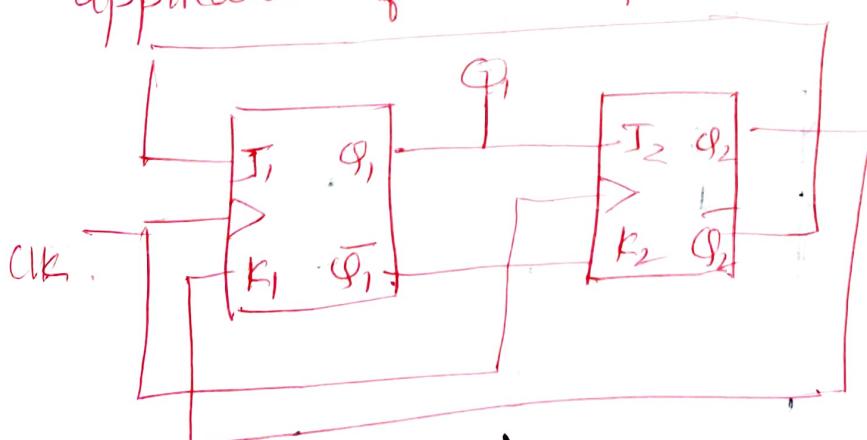
Write the output for 4 clock pulses assuming initially, both the flip flops are cleared. (10, 01, 00, 10)

	Q_1	Q_0	D_0	D_1	Q_1^+	Q_0^+
1.	0	0	0	0	0	0
2.	1	1	0	1	0	1
3.	0	1	1	1	1	1
4.	1	1	0	1	0	1

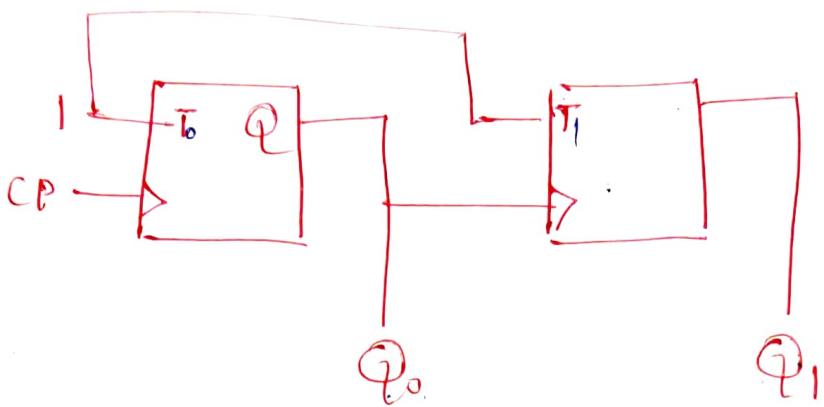
X

* The two flip flops output states Q_1, Q_2 in the figure are initiated to 00. The sequence generated at Q_1 upon application of 5 clock pulses is

01100



* In the sequential circuit shown below, if the initial value of the output Q_1Q_0 is 00, what are the next four values of Q_1Q_0 ?



Truth table of TFF

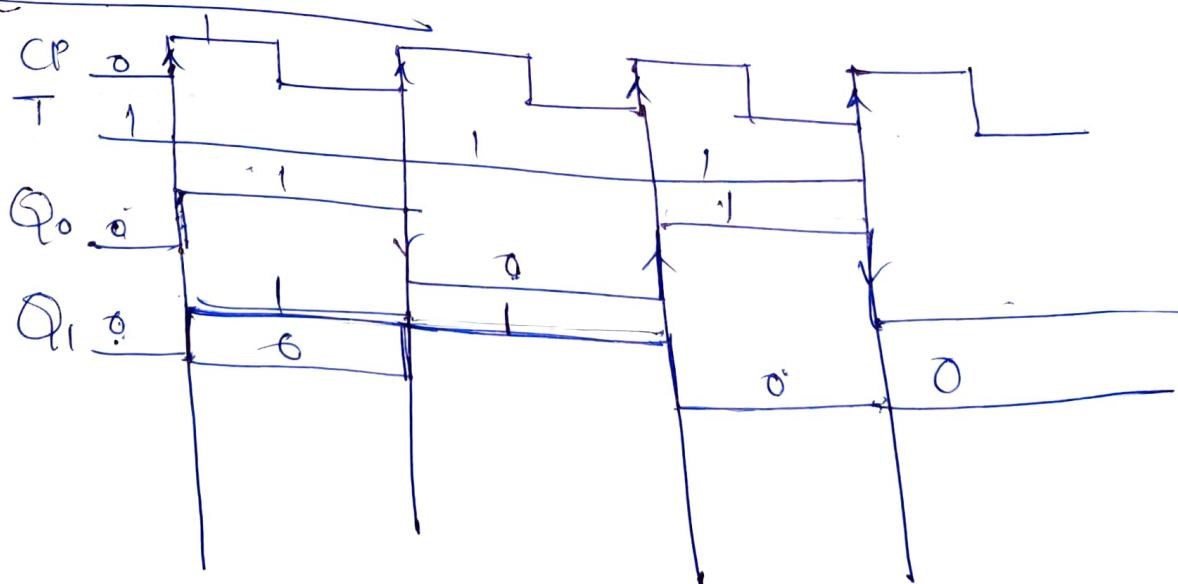
T	Q
0	0 (Memory)
1	0 (Toggle)

$T_0 \ Q_0$	$T_1 \ Q_1$	$Q_1 \ Q_0$
0	0	0 0
1	1	1 1
1	0	1 0
0	1	0 1
0	0	0 0

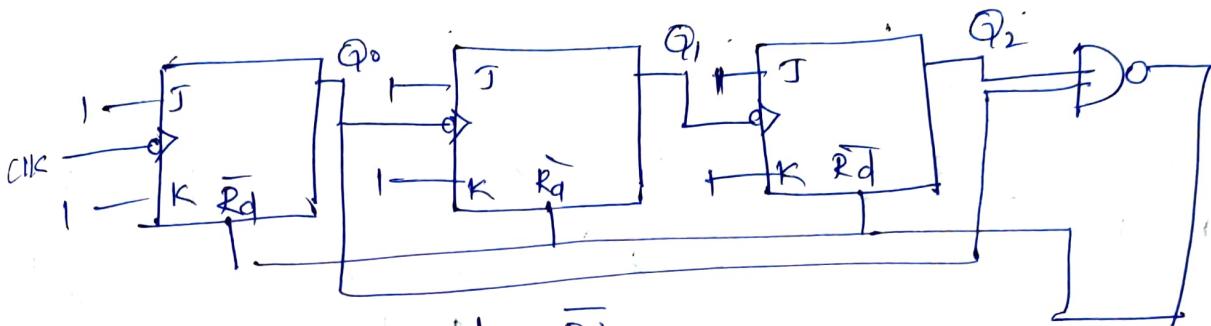
CP	Q_1	Q_0	Q_1
1st	0	0	0
2nd	0	0	1
3rd	0	1	0
4th	1	0	1

$Q_1 \ Q_0$

0 0 $\xrightarrow{1st \ CP}$
 0 1 $\xrightarrow{2nd \ CP}$
 1 0 $\xrightarrow{3rd \ CP}$
 1 1 $\xrightarrow{4th \ CP}$



* The circuit shown consists of JK flip flops, each with an active low asynchronous reset ($\bar{R_d}$ input). The counter corresponding to this circuit is, _____



$Q_2\ Q_1\ Q_0$	$\bar{R_d}$
0 0 0	1
0 0 1	0
0 1 0	0
0 1 1	0
1 0 0	0
0 1 0	1
1 1 0	0
1 1 1	0

when $\bar{R_d} = 0$, $R_d = 1$, it will clear the outputs to 000

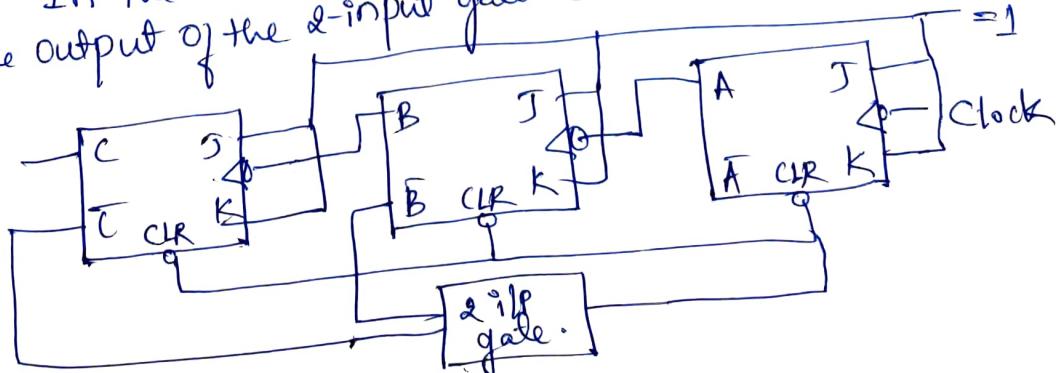
Mod-5 Up Counter

* A 4 bit modulo-16 ripple counter uses JK flip flops. If the propagation delay of each FF is 50 ns, the maximum clock frequency that can be used is equal to,

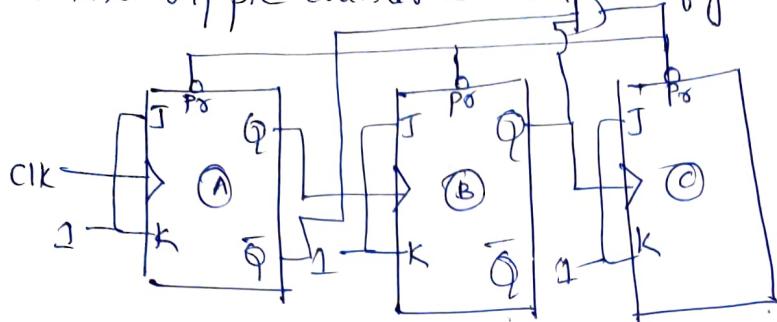
$$\text{No. of FF used} = 4 \\ \text{Min. time period of clock pulse} = 4 \times 50 \text{ ns} \\ = 200 \text{ nsec.}$$

$$\text{Max. Clock frequency} = \frac{1}{200 \times 10^{-9}} \text{ Hz} \\ = 5 \text{ MHz}$$

* In the mod-6 ripple counter shown in the given figure, the output of the &-input gate is used to clear the JK flip flops.



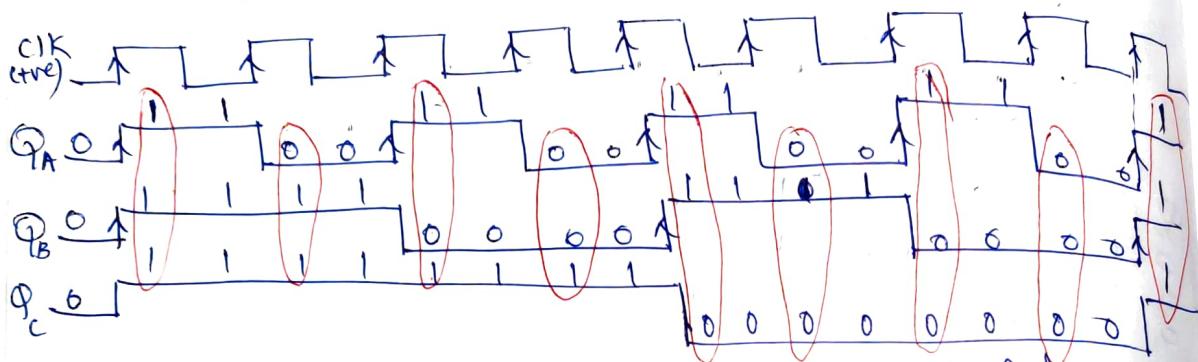
* The ripple counter shown in the figure works as a



P_C	P_B	P_A	D
1	1	1	7
1	1	0	6
1	0	1	5
1	0	0	4 \rightarrow \bar{P}_A
0	1	1	3
0	1	0	2
0	0	1	1
0	0	0	0
1	1	1	-

We
and
the SW
[De
HI
the
ano
pres

* mod-3 down Counter



* A pulse train with a frequency of 1MHz is counted using a modulo-1024 ripple counter built with JK flip flops. For proper operation of the counter, the maximum permissible propagation delay per flip flop stage is

_____ nsec.

$$f = 1 \times 10^6 \text{ Hz}$$

$$\text{no: of flip flops} = n = 10$$

$$(2^n = 1024 = 2^{10})$$

$$t_p \leq \frac{1}{10f}$$

$$\frac{1}{10 \times 10^6}$$

$$= 100 \text{ nsec}$$

* We have a digital quiz game that works on a clock and reads an input from a manual button. We have the switch to transmit only one HIGH pulse to the circuit.

[Design a secondary circuit that will transmit a HIGH pulse with duration of only one cycle when the manual button is pressed and won't transmit another pulse until the button is depressed and pressed again]

Stand by Condition → Where it waits for another button press.

Active

→ Where the button has just been just pressed and our circuit needs to transmit a HIGH pulse

Waiting

→ Where our circuit waits for the button to be released before it returns to the stand by condition.

