COUNTERS

Introduction -

A counter is probably one of the most useful & versatile subsystems in a digital system.

A counter driven by a clock can be used to count the number of clock cycles.

A counter can be used as an instrument for measuring time & therefore period or frequency.

There are two different types of counters.

i) Asynchronous Counters

ii) Synchronous counters.

Asynchronous Counters

Asynchronous counter is simple and straight forward in operation, and the construction usually requires a minimum H/w equipments.

But it has a disadvantage of low speed.

Because each flipflop is triggered by the previous flipflop, the counter has a cumulative settling time. Counters of such type are called as serial or asynchronous.

An increase in the speed of operation can be achieved by the use of a parallel or synchronous counter.

Here, every flipflop is triggered by the clock and thus the settling time is simply equal to the delay time of a single flipflop.

The increase in speed is usually obtained at the price of increased S/W.

Asynchronous Counters [OR] Ripple Counters.

A binary ripple counter can be constructed using clocked JK flipflop's. The figure below shows three -ve edge triggered JK flipflop's connected in cascade.

The system clock drives flip flop A. The output of A drives flipflop B, and the output of B drives flip flop C.

All the J and K inputs are tied to +Vcc. This means that each flipflop will change state [TOGGLE] with a -ve transition at its clock input.

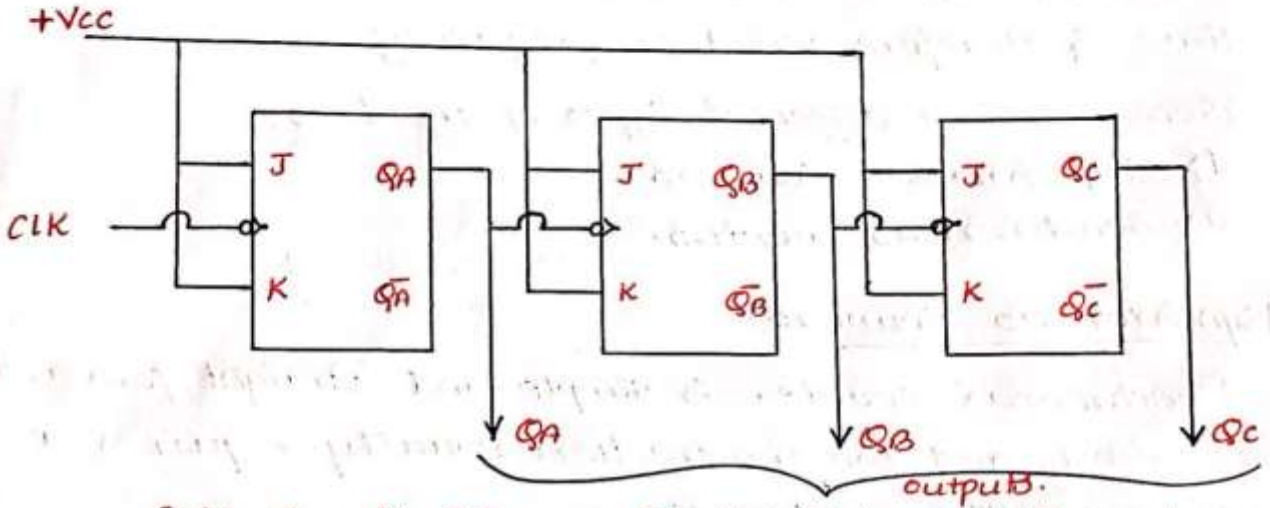


FIG: 3-bit Binary Ripple Counter / Asynchronous UP Counter.

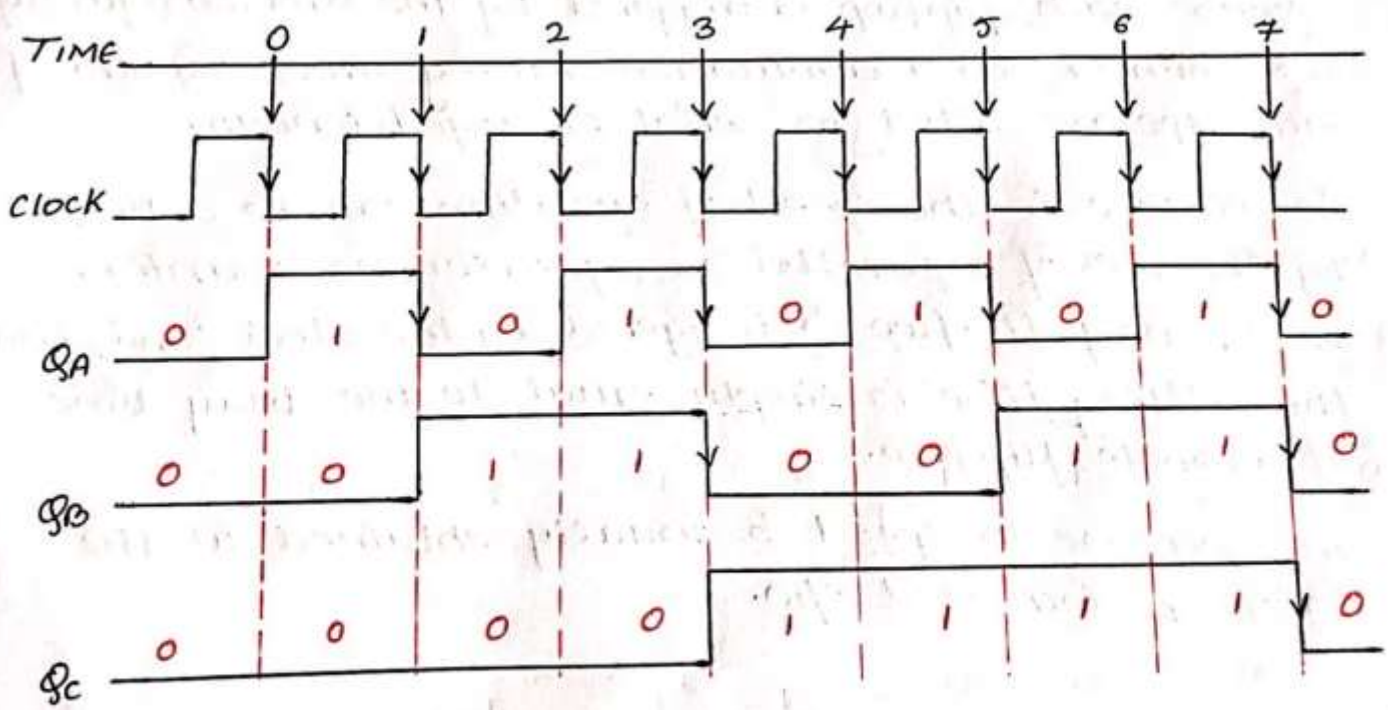


FIG: Timing Diagram of Asynchronous UP Counter or Ripple up counter.

Negative Clock Transition	Q_C	Q_B	Q_A	State or count
0	0	0	0	0
1	0	0	1	1
2	0	1	0	2
3	0	1	1	3
4	1	0	0	4
5	1	0	1	5
6	1	1	0	6
7	1	1	1	7
8	0	0	0	0

Table: Truth Table for a 3-bit Asynchronous up counter.

When the output of a flipflop is used as the clock i/p for the next flipflop, we call the counter a Ripple counter or Asynchronous Counter.

The A flipflop must change state before it can trigger the B flipflop, and the B flipflop has to change state before it can trigger the C flipflop.

The output of each flip flop has to change from $(1 \rightarrow 0)$ only then the next flip flop changes the output state.

Here each and every flipflop is waiting for the previous flipflop o/p. Because of this, the overall propagation delay of the counter is the sum of individual delays of flipflops. For instance, if each flipflop in this three flipflop counter has a propagation delay time of "10 ns", the overall propagation delay time for the counter is 30ns.

At clock 0, $Q_C Q_B Q_A = 000$ where Q_C is MSB & Q_A is LSB.

Every time there is a NT in clock, the flipflop A will change the state. Since Q_A acts as clock for Q_B, each time there is a NT in Q_A, Q_B will toggle. Since Q_B acts as clock for Q_C, each time there is a NT in Q_B, Q_C will toggle.

Asynchronous Down Counter. [OR] Ripple Down Counter.

Asynchronous down counter will work similarly as that of Asynchronous up counter. The only difference b/w them is that, in up counter, the counter value is incremented by 1, but in down counter, it is decremented by 1.

To get the down counter combination, we need to connect the complement of Q_A as the clock of next flipflop ie B. and the complement of ~~Q_A~~ Q_B will be connected as the clock to flipflop C.

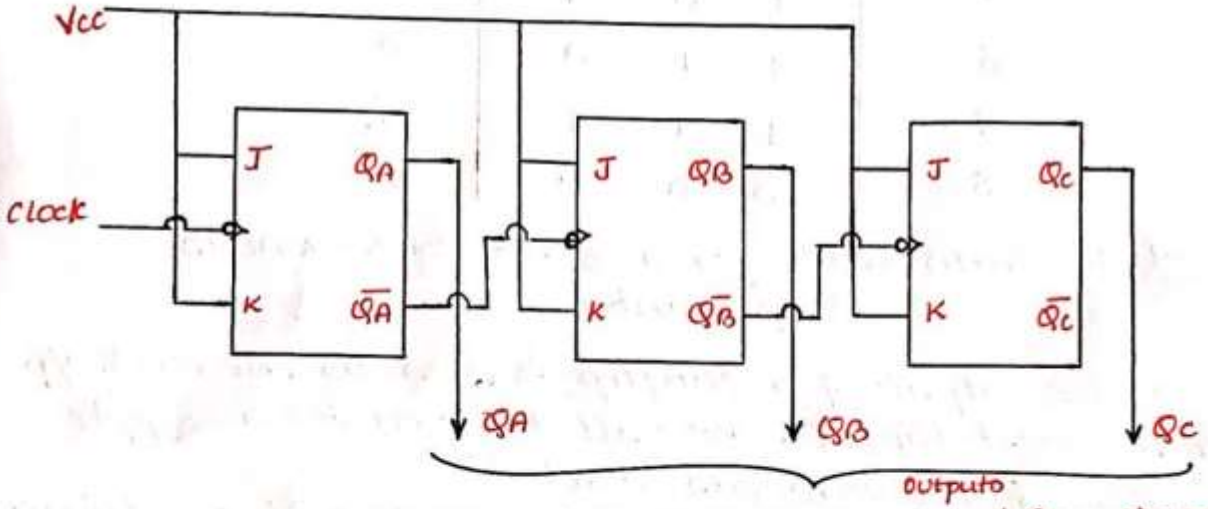


FIG: 3-bit Binary Ripple Down Counter / Asynchronous Down counter.

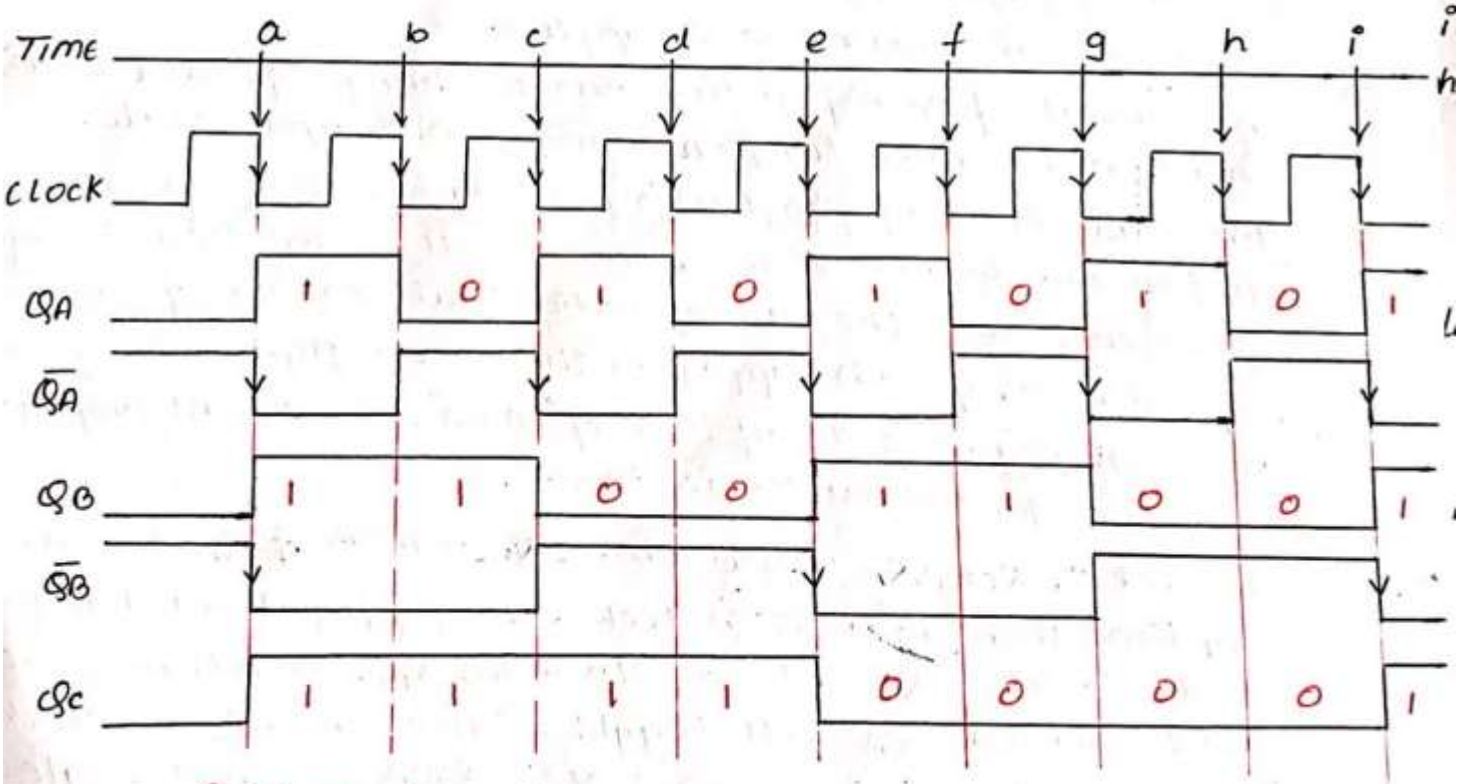


FIG: Timing Diagram of Asynchronous Down Counter or Ripple Down Counter.

Negative clock Transition	Q_C	Q_B	Q_A	State or count.
a	1	1	1	7
b	1	1	0	6
c	1	0	1	5
d	1	0	0	4
e	0	1	1	3
f	0	1	0	2
g	0	0	1	1
h	0	0	0	0
i	1	1	1	7

Table: Truth Table for a 3-bit Asynchronous Decon Counter.

The system clock is still used at the clock i/p to flip flop Q_A, but the complement of Q_A, $\bar{Q_A}$ is used to drive the flipflop Q_B, like wise $\bar{Q_B}$ is used to drive flip flop C. But we have to see the output at Q_A, Q_B & Q_C only.

Flipflop A simply toggles with each negative clock transition as earlier. But flip flop Q_B will toggle each time when $\bar{Q_A}$ changes state from high to low.

Similarly, Q_C will toggle each time when $\bar{Q_B}$ changes state from high to low.

The counter contents become ABC = 111 at point 'a' on timeline it will change to 110 at point 'b' and it will change to 101 at point 'c' and so on.

Notice that the counter contents are reduced by one count with each clock transition. In other words, counter is operating in count decon mode.

Asynchronous / Ripple Up-Down Counter -

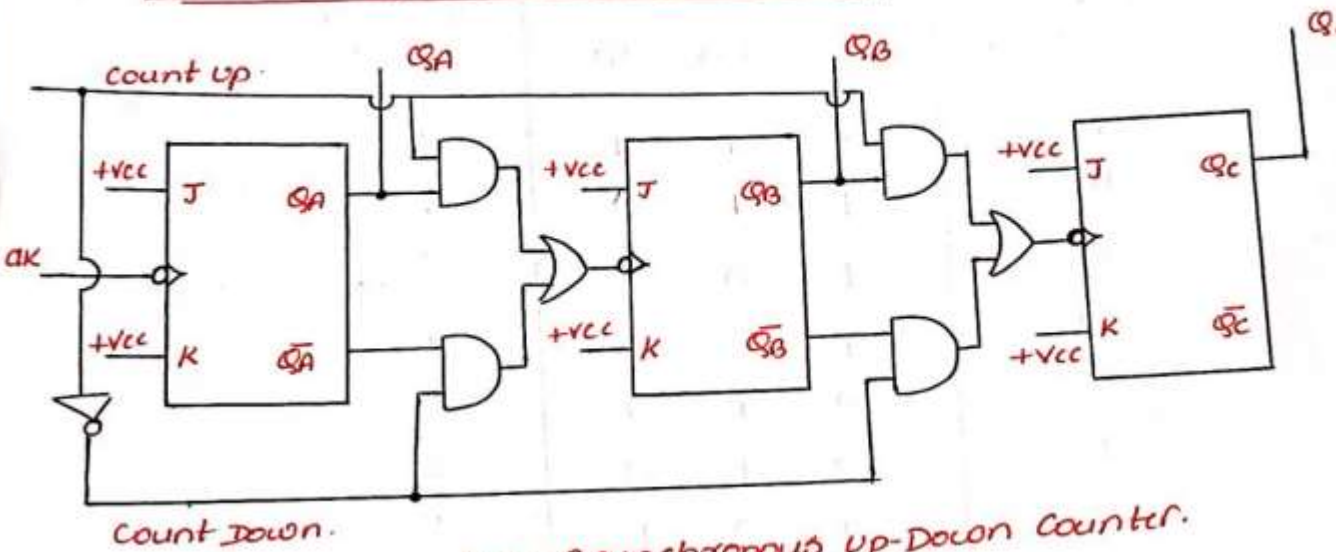


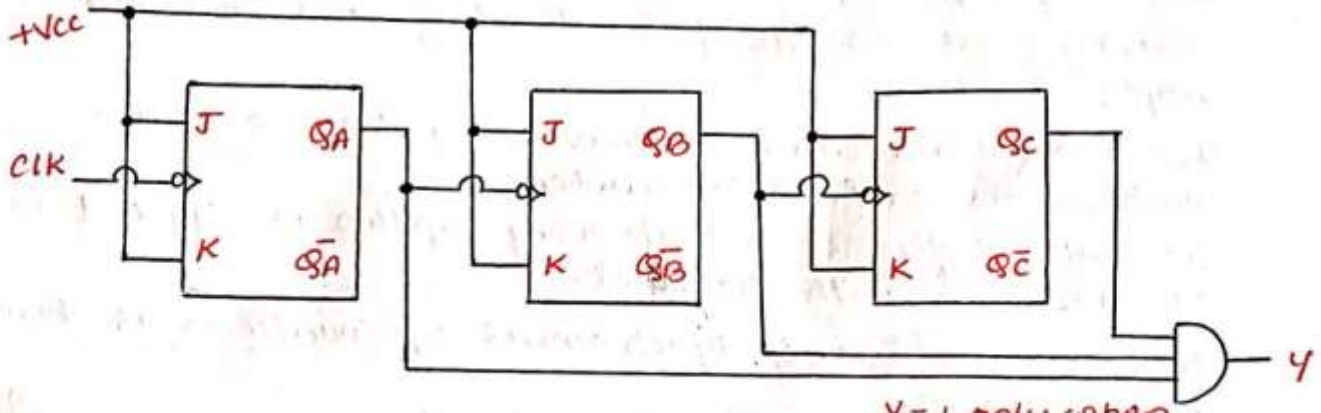
Fig: Asynchronous up-down counter.

A 3-bit Asynchronous up-down counter that counts in a straight binary sequence is as shown in the figure.
For this counter to progress through a count-up sequence, it is necessary to trigger each flipflop with true side of the previous flipflop. If the count up control line is 'HIGH', the clock will follow the upper AND Gate output of each flip flop, and as a result it will work as an up counter.
On the other hand, if the count Down line is high; each flip flop will be triggered from the complement side of previous flip flop. The counter will then be in a count Down mode.

Decoding Gates -

A decoding gate can be connected to the outputs of a counter in such a way that the output of the gate will be high (or low) only when the counter contents are equal to a given state. For instance, the decoding gate connected to the 3-bit ripple counter in the figure will decode state ($Q_C Q_B Q_A = 111$). Thus the gate o/p will be high only when

$Q_A = Q_B = Q_C = 1$.

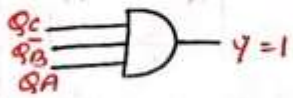


Y = 1 only when $Q_A Q_B Q_C = 111$.

The other seven states of the counter can be developed in a similar fashion.

For instance, to decode state 5, the truth table shows that $Q_C Q_B Q_A = 101$ is the unique state.

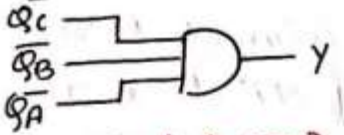
For the gate o/p to be high during this time, we must use $Q_C, \bar{Q}_B, Q_A$ at gate input's.

ie; for state 5 

$Q_C = 1$, $Q_B = 0$, $Q_A = 1$.
$\bar{Q}_B = 1$

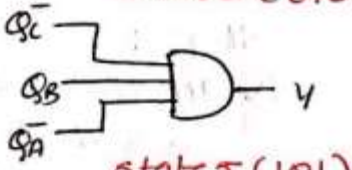
Similarly, for all states.

State 0 (000)



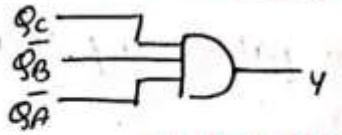
State 1 (001)

State 2 (010)

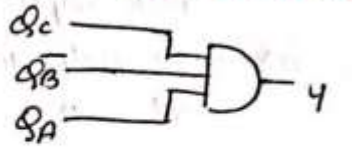


State 3 (011)

State 4 (100)

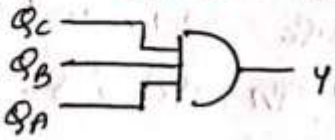


State 5 (101)



State 6 (110)

State 7 (111)



The clock is directly applied to Q_A flip flop. and the JK flip flop used responds to a -ve transition at the clock i/p & Q_A toggles.

Whenever Q_A is high, AND gate X is enabled. & a clock pulse is passed through the gate to the clock i/p. of flip flop Q_B. Thus Q_B changes state with every other clock NT. at points b, d, f, & h on the time line.

Now since AND gate Y is enabled, it will transmit the clock to flipflop Q_C only when both Q_A & Q_B are high, the flipflop Q_C changes state with every fourth clock NT. at points d & h. on the time line.

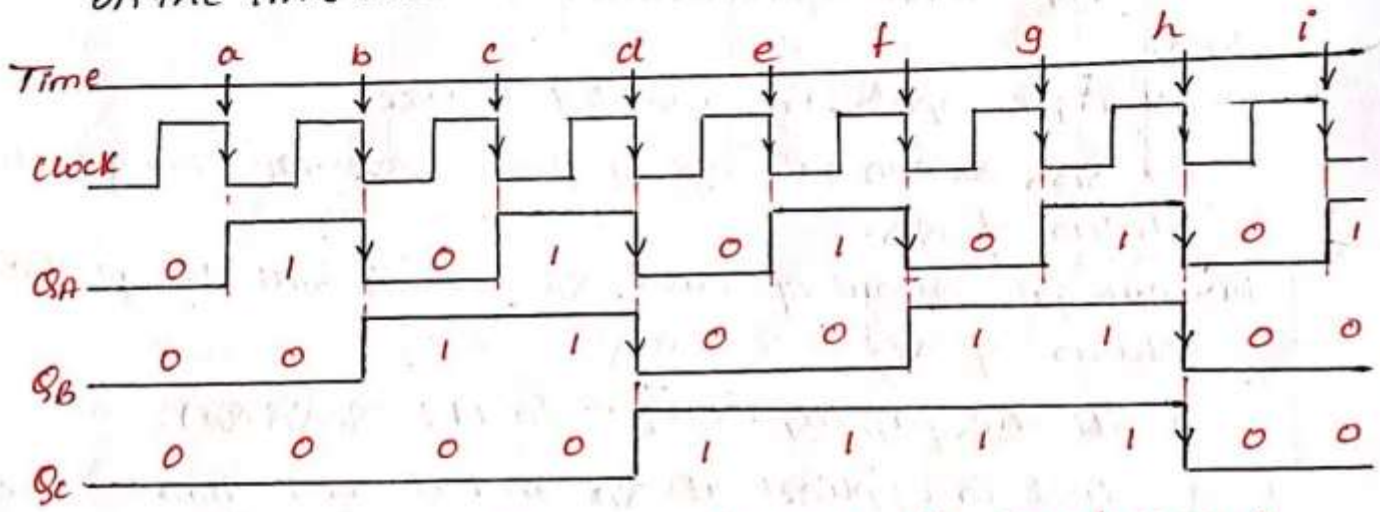


Fig: Timing Diagram for a 3-bit synchronous up counter.

(NT) (clock)	MSB Q_C	Q_B	LSB Q_A	count	CLK AND Q_A AND gate (X)	CLK · Q_A · Q_B AND gate (Y)
—	0	0	0	0	0	0
↓	0	0	1	1	1	0
↓	0	1	0	2	0	0
↓	0	1	1	3	1	1
↓	1	0	0	4	0	0
↓	1	0	1	5	1	0
↓	1	1	0	6	0	0
↓	1	1	1	7	1	1
↓	0	0	0	0	0	0

Truth Table for 3-bit Synchronous up counter.

Synchronous Down Counter -

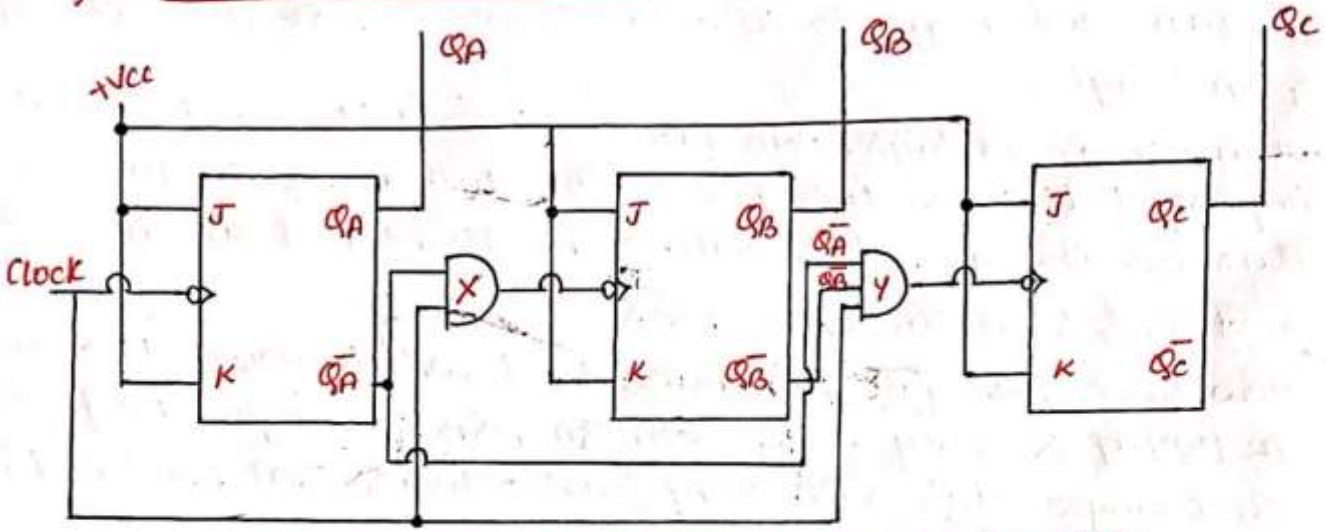


FIG: 3-bit Synchronous Down Counter.

Steps:

i) All J & K inputs are connected to +Vcc.

ii) for Q_B, AND gate o/p of clock & $\bar{Q_A}$ will change the status of Q_B.

iii) AND gate output of clock, $\bar{Q_A}$ and $\bar{Q_B}$ will change the status of Q_C.

Set all the flip flop's output to 1 1 1 ($Q_C Q_B Q_A$).

* The clock is applied at Q_A and it will change the state for every NT of the clock.

* The output Q_B will change the state when the clock and $\bar{Q_A}$ will have the NT.

* The output Q_C will change the state when the clock and $\bar{Q_A}$ and $\bar{Q_B}$ will have the NT.

The truth table as well as the Timing Diagram for a 3-bit Synchronous Down counter is shown in the figure below.

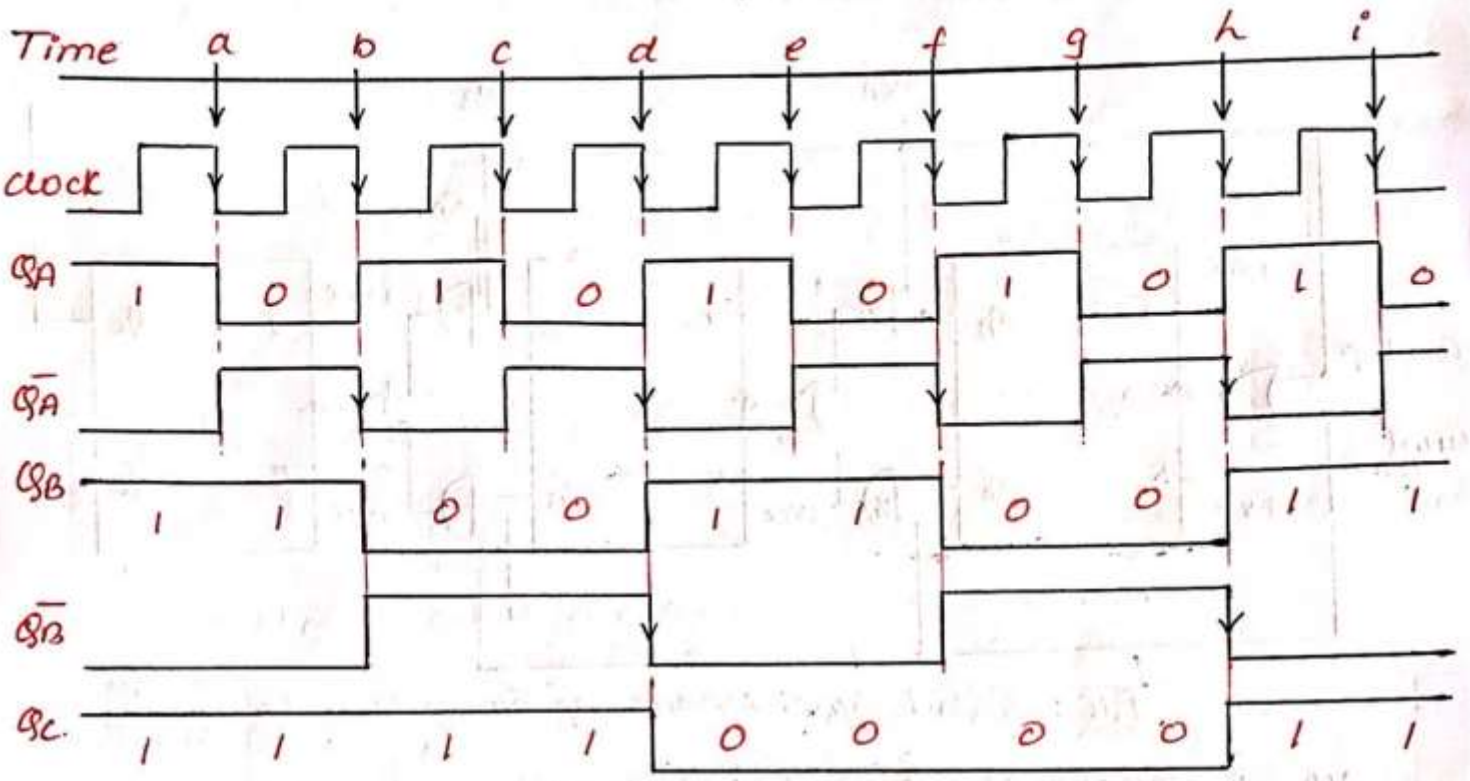


Fig: Timing Diagram for a 3-bit Synchronous Down Counter.

NT clock	count	Q_C	Q_B	Q_A	$\bar{Q}_C$	$\bar{Q}_B$	$\bar{Q}_A$	x gate	Y gate
–	7	1	1	1	0	0	0	–	–
↓	6	1	1	0	0	0	1	↓	0
↓	5	1	0	1	0	1	0	0	0
↓	4	1	0	0	0	1	1	↓	↓
↓	3	0	1	1	1	0	0	0	0
↓	2	0	1	0	1	0	1	↓	0
↓	1	0	0	1	1	1	0	0	0
↓	0	0	0	0	1	1	1	↓	↓
↓	7	1	1	1	0	0	0	0	0

Truth Table for 3-bit Synchronous Down Counter.

Synchronous up Docon Counter-

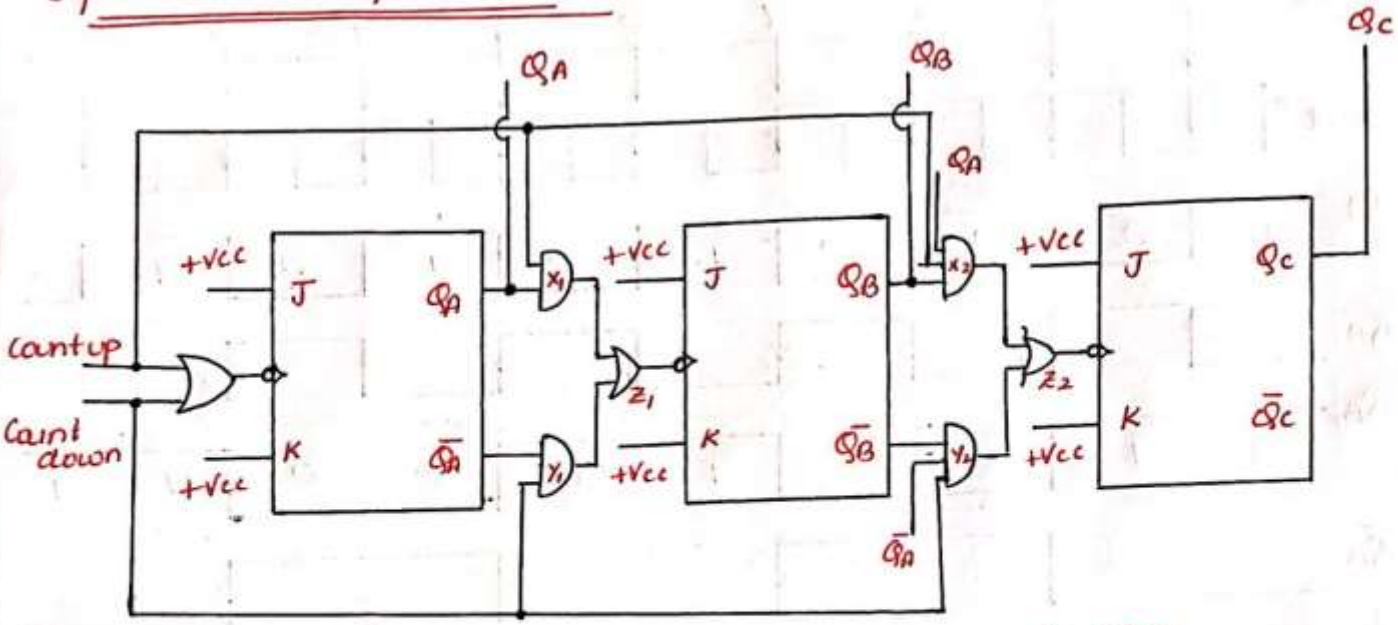


Fig: 3-bit Synchronous up Docon Counter.

All JK inputs are connected to +Vcc.

In any parallel counter (Synchronous Counter), the time at which any flipflop changes state is determined by the states of all previous flipflops in the counter.

In the count up mode, 'A' flip flop must toggle everytime when true value (ie; Q_A, Q_B, Q_C) will change state from 1 to 0.

for count up mode, the clock will follow the result of X_1 & X_2 gates.

In the count docon mode, 'A' flipflop must toggle everytime when complement value (ie; $\bar{Q}_A, \bar{Q}_B, \bar{Q}_C$) will change state from 1 to 0.

for count docon mode, the clock will follow the result of Y_1 & Y_2 gates.

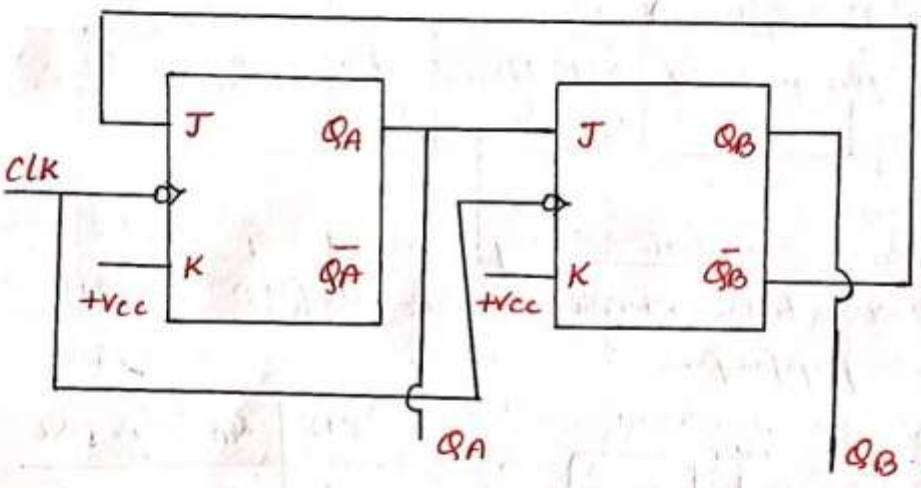
To operate in count-up mode, the system clock is applied at the count up input, while the count docon input is held low.

To operate in count-docon mode, the system clock is applied at the count docon input, while the count-up input is held low.

Changing the counter Modulus -

Q. A Mod-3 counter to design a Mod-6 counter.

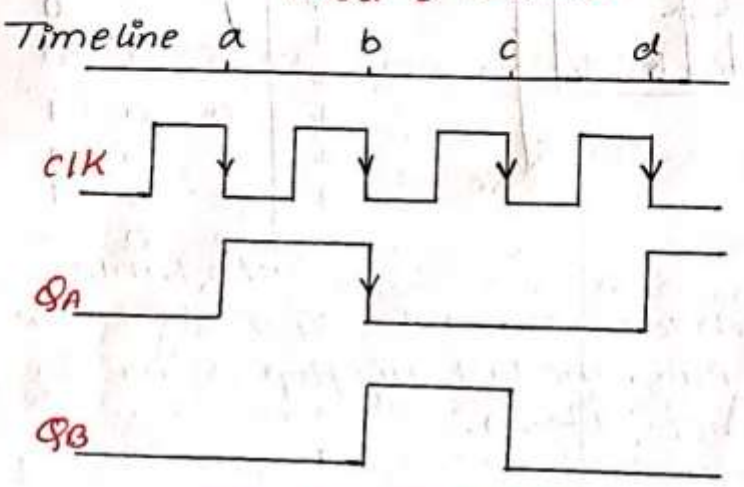
The two flipflop's in the figure have been connected to provide a mod-3 counter. Since two flipflop's have a natural count of 4, this counter skips one state. The waveform & the truth table is as shown in the figure.



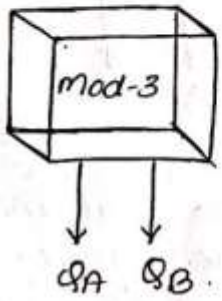
Mod-3 Counter.

Q_B	Q_A	Count
0	0	0
0	1	1
1	0	2
0	0	0

Truth Table.



Timing Diagram.



This counter progresses through count sequence 00, 01, 10. & then back to 00. It clearly skips count 11. Here's how it works.

i) Prior to point 'c' on the timeline, $Q_A = Q_B = 0$ and also $\bar{Q_A} = \bar{Q_B} = 1$. A -ve clock transition at 'a' will cause:

a) Q_A to toggle to 1, since J & K are high.

b) Q_B to reset to 0, bcoz of synchronous clock, both FF's working in parallel. For this reason Q_B f/f will refer previous memory of Q_A (ie, '0') if J=0 & K=1 in second f/f will make $Q_B = 0$ & $\bar{Q_B} = 1$.

ii> Prior to point 'b' on the time line, $Q_A = 1$ & $Q_B = 0$. A -ve clock at point b will cause:

a> Q_A to toggle to 0, since J & K inputs are high.

b> Q_B to toggle to 1, since its J & K inputs are high.

iii> Prior to point 'c' on the time line, A=0 & B=1. A -ve clock at c will cause.

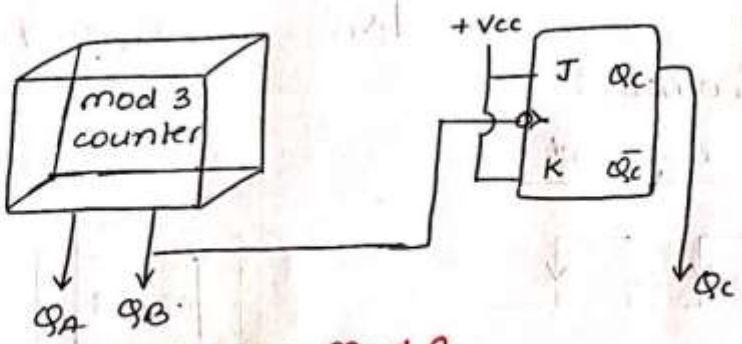
a> A to 0. (Since J=0 & K=1)

b> B to 0. (Since J=0 & K=1)

iv> The counter now progressed all three states & working as Mod-3 counter.

Mod-6 counter

A mod-6 counter can be obtained by a mod 3 & mod 2 counter. To build a Mod-6 counter, connect a Q_B flipflop's output as clock for the next flipflop.

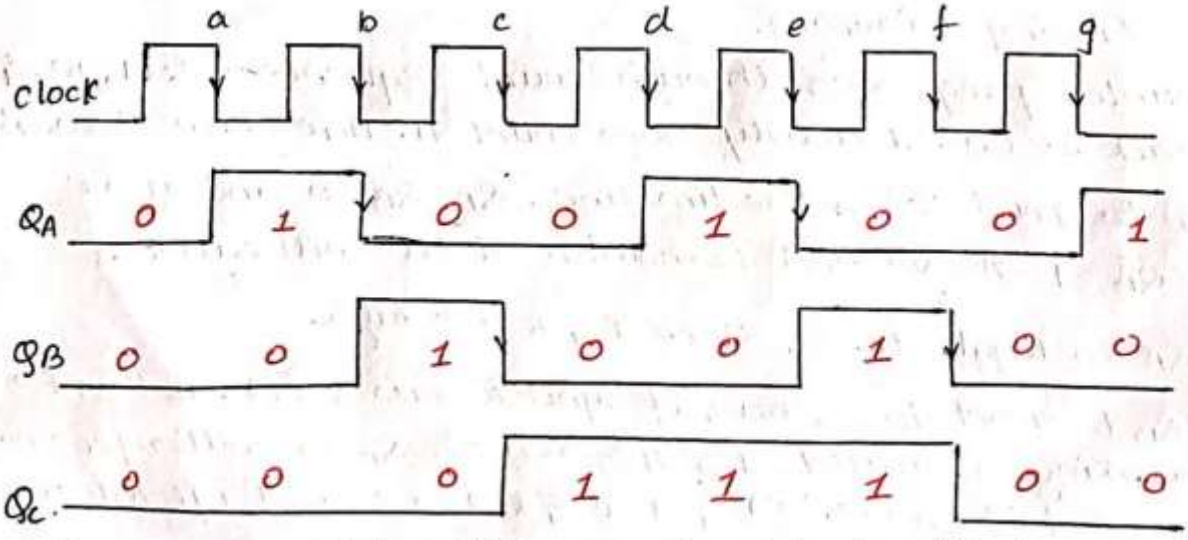


3 × 2 = Mod 6.

CLK	Q_A	Q_B	Q_C
↓	0	0	0
↓	1	0	0
↓	0	1	0
↓	0	0	1
↓	1	0	1
↓	0	1	1
↓	0	0	0

Truth Table.

Notice that mod 3 is synchronous counter & mod 6 is an Asynchronous counter. Bcoz, the last flip flop is waiting for the -ve transition in B. (ie; 1→0).

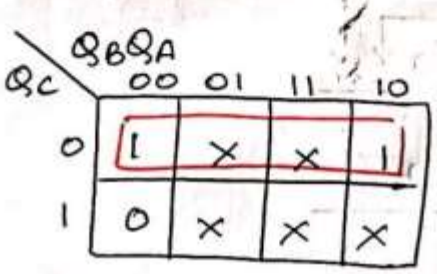


for every -ve transition in Q_B output will change the state of Q_C.

Design a Decade Counter using a Mod 5 x Mod 2 counter.

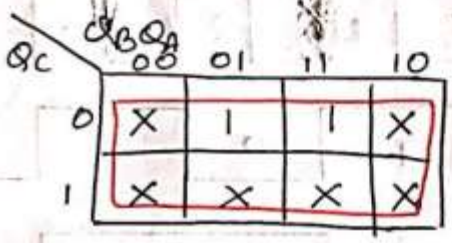
Present state.			Next State			Excitation Table.					
Q_C	Q_B	Q_A	Q_{C+1}	Q_{B+1}	Q_{A+1}	J_C	K_C	J_B	K_B	J_A	K_A
0	0	0	0	0	1	0	X	0	X	1	X
0	0	1	0	1	0	0	X	1	X	X	1
0	1	0	0	1	1	0	X	X	0	1	X
0	1	1	1	0	0	1	X	X	1	X	1
1	0	0	0	0	0	X	1	0	X	0	X

for J_A



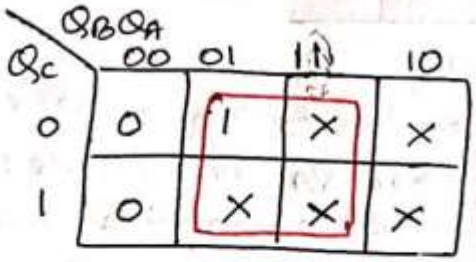
$J_A = \overline{Q_C}$

for K_A



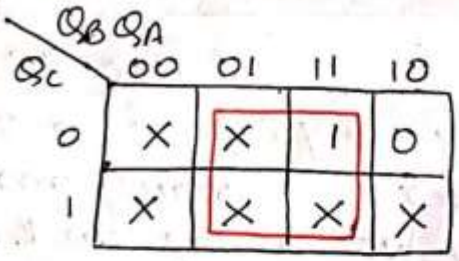
$K_A = 1$

for J_B



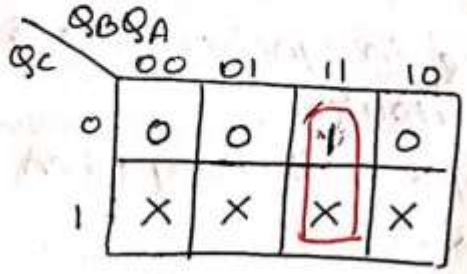
$J_B = Q_A$

for K_B



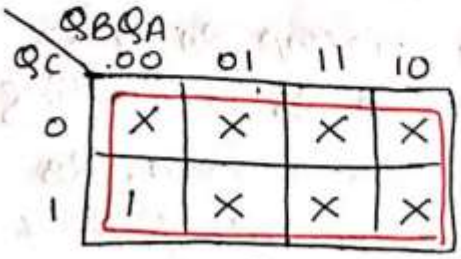
$K_B = Q_A$

for J_C

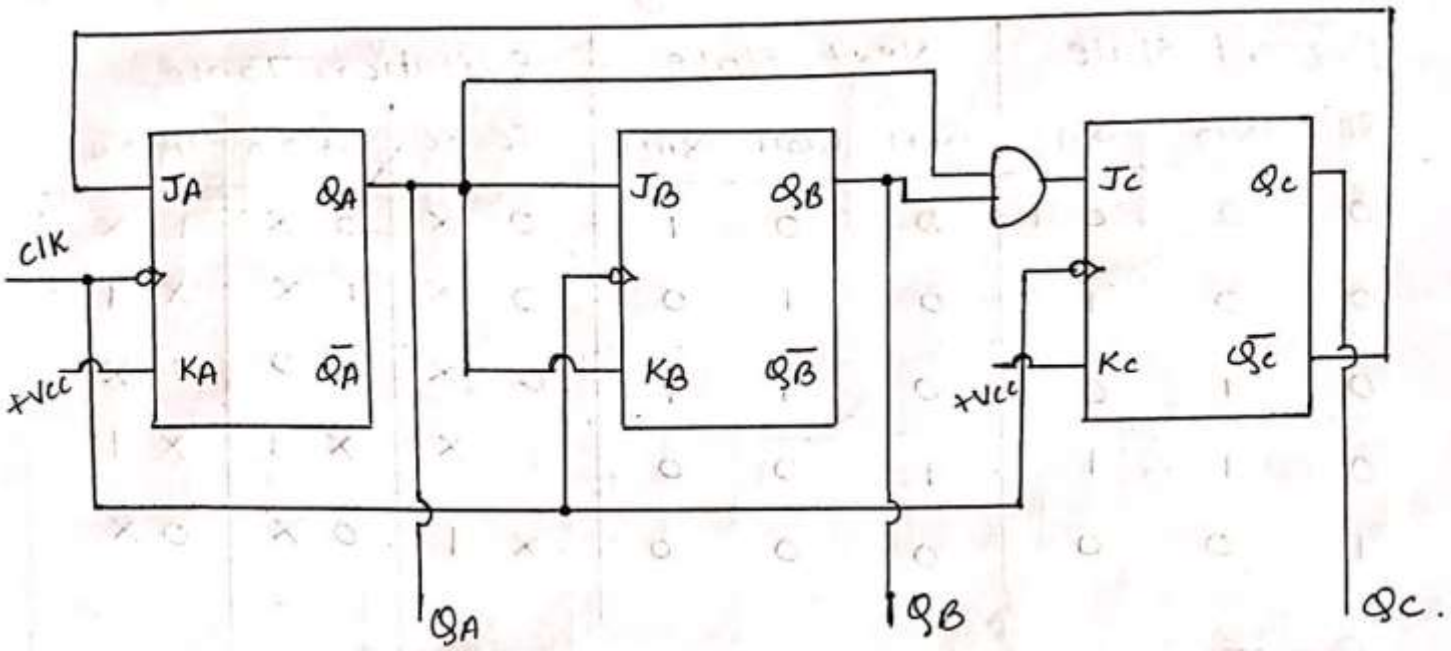


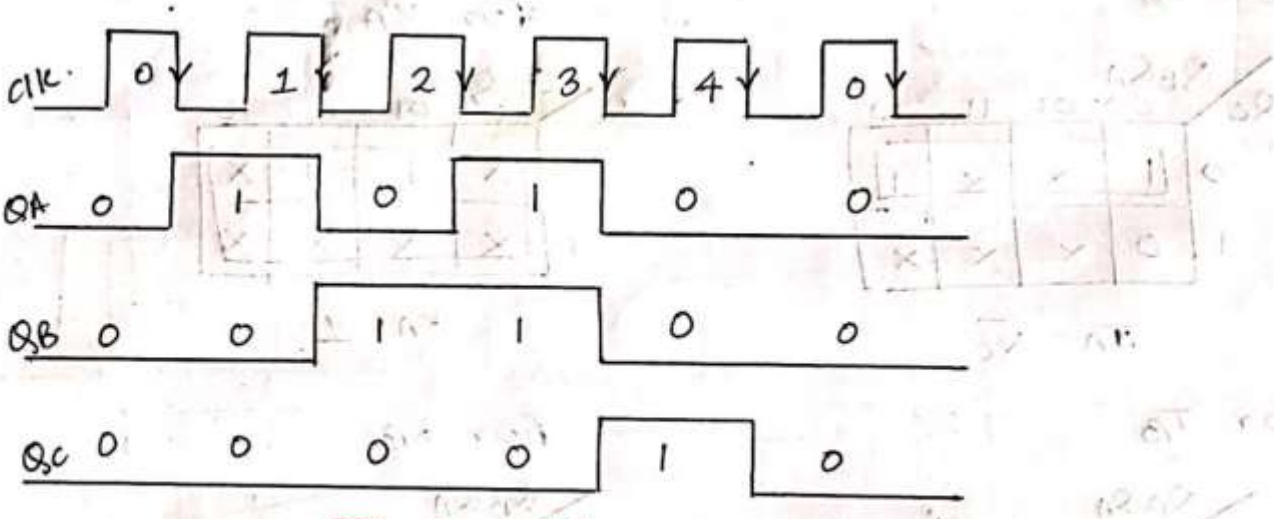
$J_C = Q_A Q_B$

for K_C.



$K_C = 1$





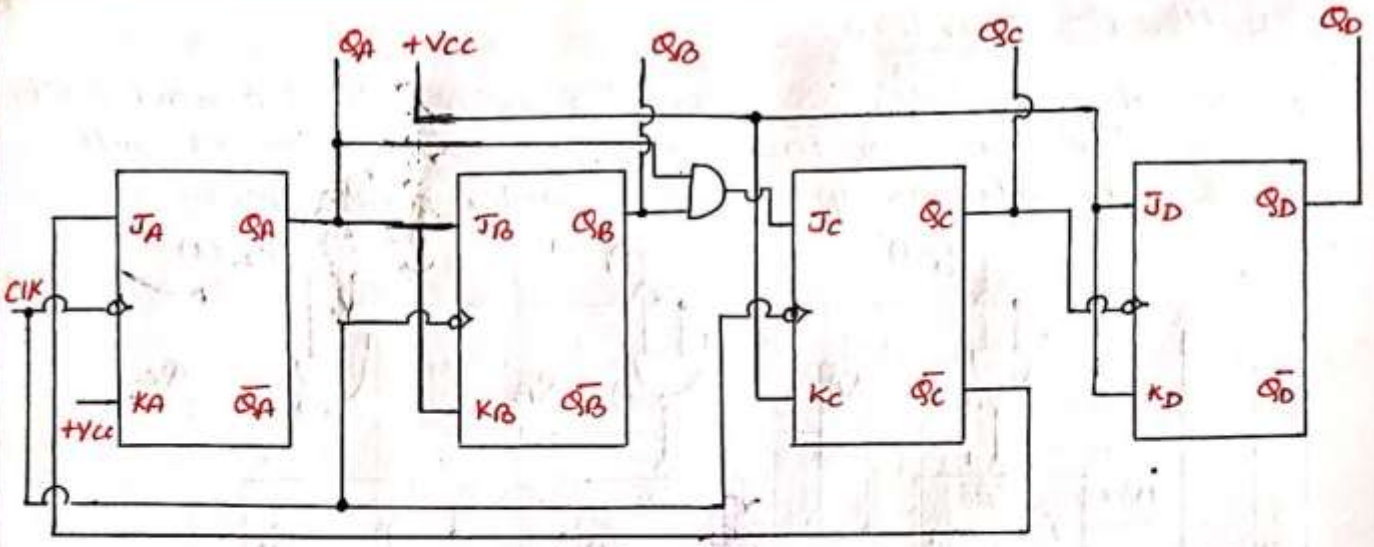
Timing Diagram.

The three flipflops in the figure works as a mod 5 counter. The waveform shows that Q_A will change for every NT in the clock, except the transition from count 4 to count 0.

At count 4, $Q_c = 1$ & $\bar{Q}_c = 0$, This makes the input of JA as 0, which inturn makes Q_A to become 0.

The flipflop Q_A & Q_c is working in synchronous & the flip flop Q_B is waiting for Q_A output.

For this reason, Q_B changes the state only when Q_A changes state from 1 to 0, ie NT.



Decode Counter.

Clock	LSB Q_A	Q_B	Q_C	MSB Q_D	count.
–	0	0	0	0	0
↓	1	0	0	0	1
↓	0	1	0	0	2.
↓	1	1	0	0	3
↓	0	0	1	0	4
↓	0	0	0	1	8
↓	1	0	0	1	9
↓	0	1	0	1	10
↓	1	1	0	1	11
↓	0	0	1	1	12.
↓	0	0	0	0	0

Truth Table.

The Q_D will toggle the output when Q_C is changing from $1 \rightarrow 0$ ($\because Q_C$ is given as a clock to Q_D flipflop).

At value (8), Q_C changes from $1 \rightarrow 0$ which makes Q_D to 1.

At value (12), Q_C changes from $1 \rightarrow 0$ which makes Q_D to 0.

Here, the decade counter doesn't have proper value like 0000, 0001, 0010, - - - - , 0111, 1000, 1001, 000 bcoz the decade counter is constructed using both synchronous & Asynchronous counter concept.

The only identification of decade counter is that, at every difference of 10 clocks, the values will be same.

Presettable Counters.

Presettable counter is a counter in which the user will assign the value to the counter then the counter will work normally as up or down counter as design.

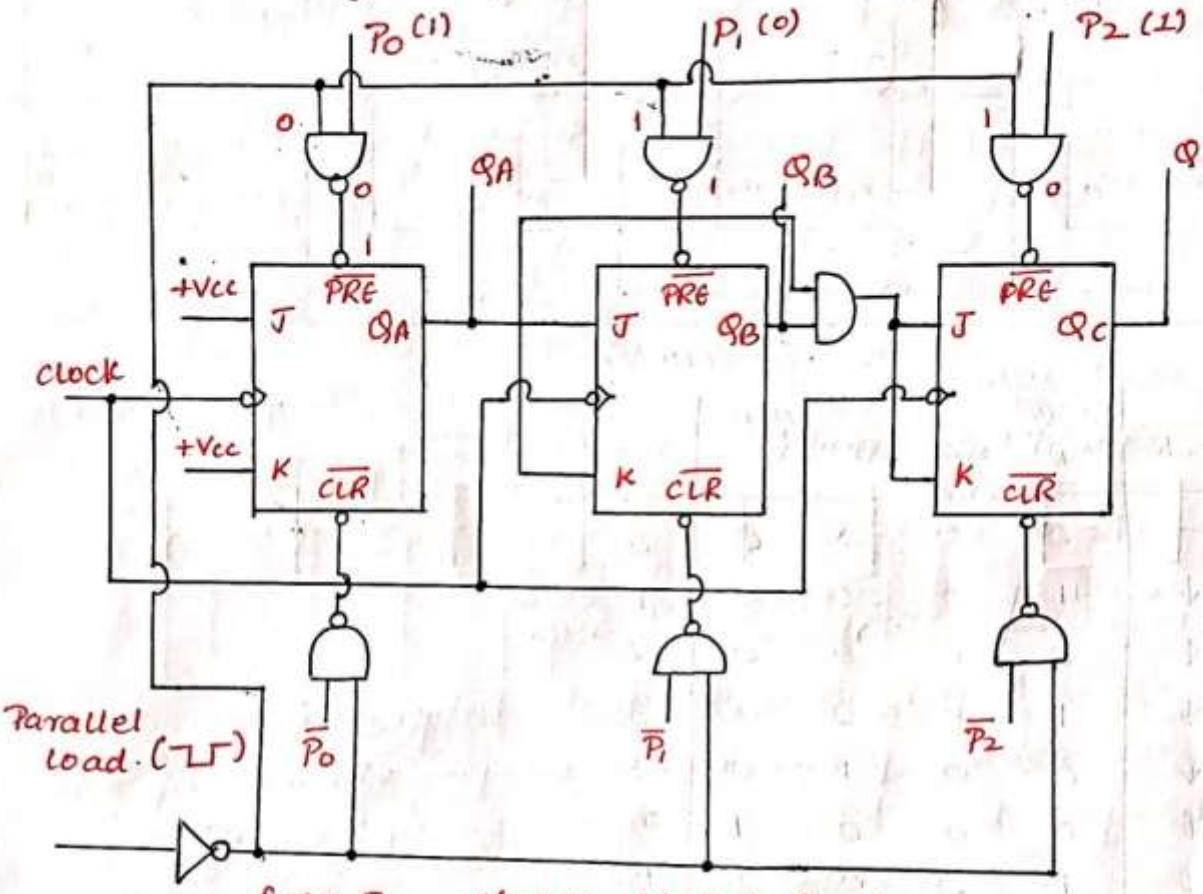


Fig: Presettable Mod-8 Synchronous up-counter.

Many Synchronous counters available as IC's are designed to be presettable. This means that they can be preset to any desired value.

In the diagram, the J, K and clk inputs are wired as a synchronous up-counter. The asynchronous preset & clear inputs are used to perform asynchronous presetting.

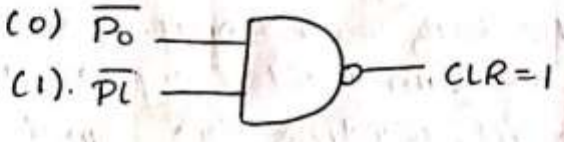
The counter is loaded by applying the desired binary number to the inputs of P_2, P_1 & P_0 and a low pulse is applied to the parallel load input.

If the inputs at $P_L = 0$ makes the output of the inverter to 1 & assigns $P_2 = 1$, $P_1 = 0$, $P_0 = 1$.

Now, At P_0, output of the NAND gate is 0 since inverter o/p value & P_0 value is 1.

If PRE = 0, Q = 1
If PRE = 1, it wont affect Q.

At clr, output of NAND gate is,



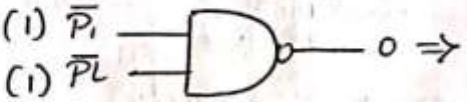
If CLR = 1, it wont affect the status of Q.

If CLR = 0, Q = 0.

At $P_1 = 0$,

wont affect Q.

At clr,

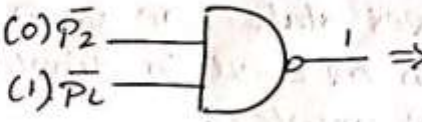


If clr = 0, Q = 0.

At $P_2 = 1$,

makes Q = 1

At clr,



wont affect Q.

If $\overline{PL} = 1$ makes parallel loading & sets the flipflop's output's to 101 bcoz $P_2P_1P_0 = 101$.

If $\overline{PL} = 0$. makes all the and gates to high which is making PRE & CLR = 1.

This wont affect the status of Q, and the counter will normally work as a mod 8 counter.

DIGITAL CLOCK-

A very interesting application of counters & decoding gates arises in the design of a digital clock. Suppose that we want to construct an ordinary clock which will display hours, minutes & seconds. The power supply for this system is usually 60 Hz, 120 Vac commercial power.

In order to obtain pulses occuring at a rate of one second each, it is necessary to divide the 60 Hz Power supply by value 60. If the resulting waveform is again divided by 60, a one per minute waveform is the result. Dividing this signal again by 60 then provides one per hour waveform. This is the basic idea to be used in forming a digital clock.

A block diagram is showing the working of digital clock.

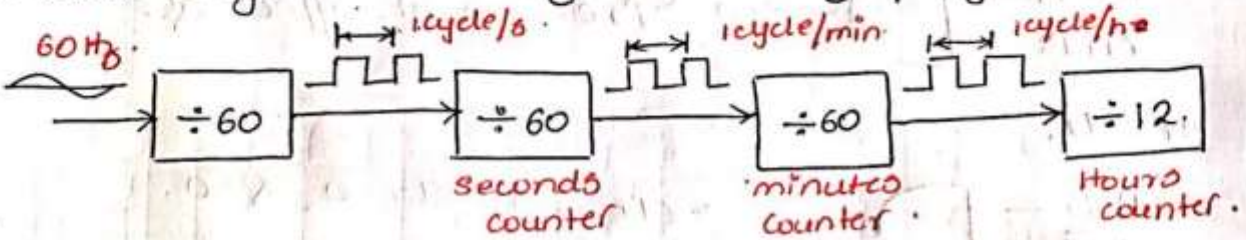


FIG: Digital clock Block Diagram.

The first divide-by-60 counter simply divides 60 Hz power signal to 1 Hz square wave. The 2nd divide-by-60 counter changes state one each second & has discrete 60 states. It can therefore be used to display seconds & hence reffered as seconds counter.

The 3rd divide-by-60 changes state one each minute & has 60 discrete states. It can be used to display minutes & hence referred as minute counter.

The last counter changes state one each 60 minutes. (once each hour). If it is a divide by-12 counter, it will have 12 states that can be decoded to provide signals to display the correct hour. This is referred as hours counter.

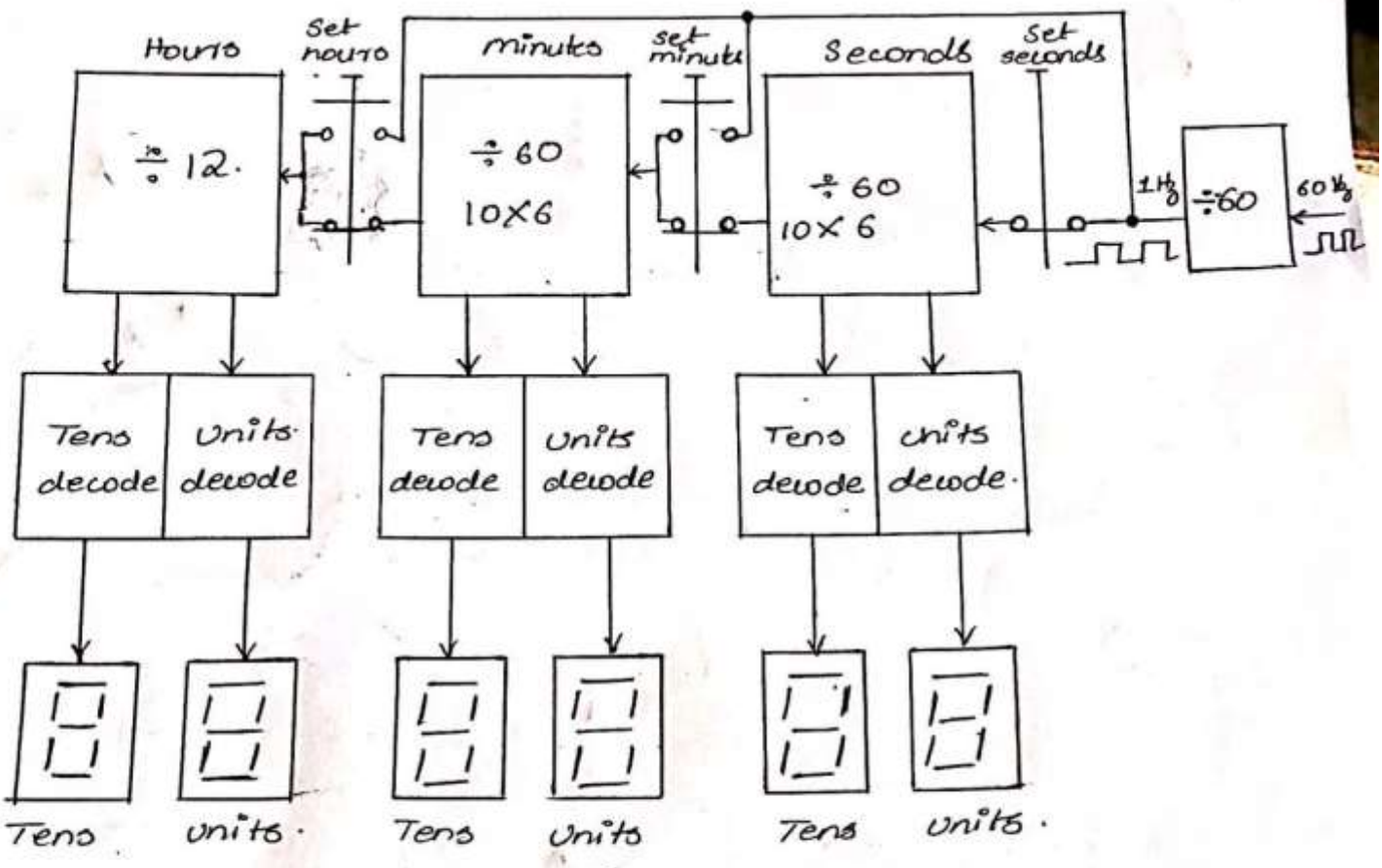


Fig: Digital Clock.

With the help of a neat block diagram, truth table and timing diagram, explain the working of a Mod-16 Ripple counter constructed using POSITIVE edge triggered JK flipflops.

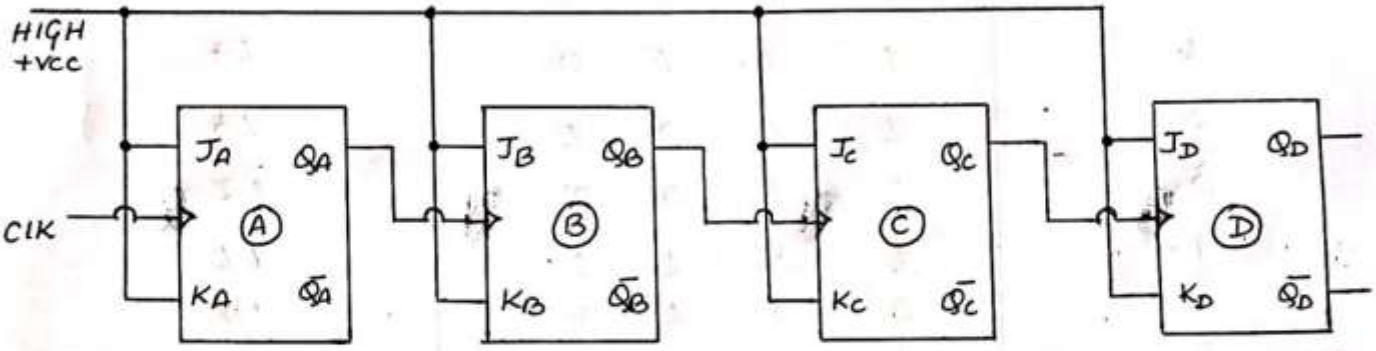


FIG: LOGIC DIAGRAM OF MOD-16 RIPPLE COUNTER.

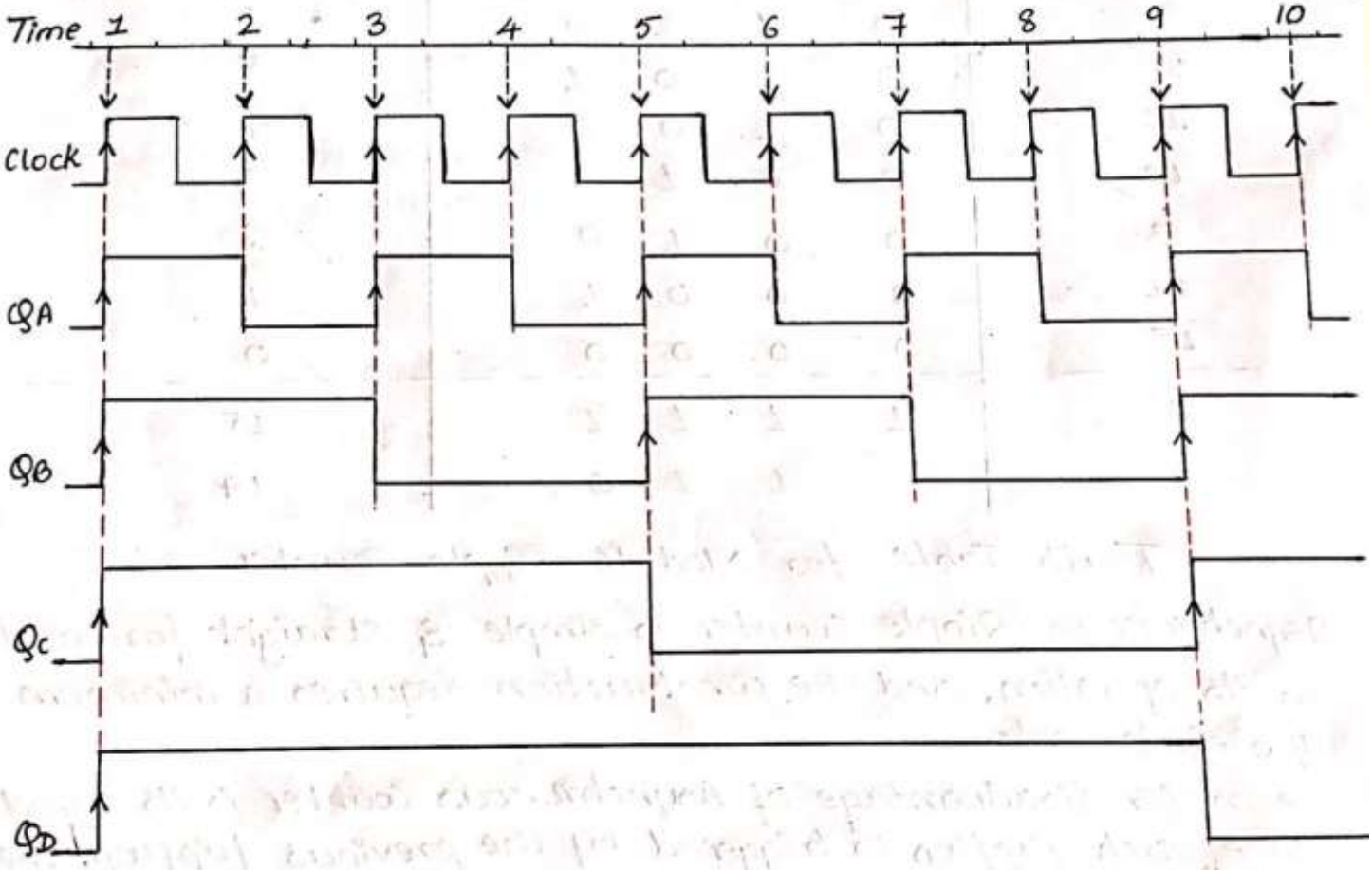


FIG: Timing Diagram of a Mod-16 Ripple Counter.

+ve clock Transitions	Q_D	Q_C	Q_B	Q_A	State or Count
—	0	0	0	0	—
0	1	1	1	1	15
1	1	1	1	0	14
2	1	1	0	1	13
3	1	1	0	0	12
4	1	0	1	1	11
5	1	0	1	0	10
6	1	0	0	1	9
7	1	0	0	0	8
8	0	1	1	1	7
9	0	1	1	0	6
10	0	1	0	1	5
11	0	1	0	0	4
12	0	0	1	1	3
13	0	0	1	0	2
14	0	0	0	1	1
15	0	0	0	0	0
0	1	1	1	1	15
1	1	1	1	0	14

Truth Table for Mod-16 Ripple Counter.

Asynchronous / Ripple Counter is simple & straight forward in its operation, and the construction requires a minimum H/w equipments.

The major disadvantage of Asynchronous Counter is its speed. Since each flipflop is triggered by the previous flipflop, the counter has a cumulative settling time.

A binary ripple counter can be constructed using clocked JK flipflops. The figure above shows 4 positive edge triggered JK flipflops connected in cascade.

The system clock which is a square wave drives the flipflop A

The output of A drives B, the output of B drives C and the output of C drives D. All the J and K inputs are tied to Vcc. This means that each flipflop will change state (TOGGLE) with a positive transition at its clock input.

When the output of the flipflop is used as a clock input for the next flip flop, we call the counter as Ripple Counter.

The A flipflop must change state before it can trigger the B flipflop, and B flipflop has to change state before it can trigger the C flipflop.

The output of each flip flop has to change from (0→1) only then the next flip flop changes the output state.

Here each and every flipflop is waiting for the previous flipflop's output. Because of this, the overall propagation delay of the counter is the sum of individual delays of the flipflops.

For instance, if each flipflop in this 4 flipflops counter has a propagation delay time of 10ns, then the overall propagation delay time for the counter is 40ns.

Each time there is a positive transition in the clock, the flipflop Q_A will change the state. This is indicated by a small arrow in the timeline.

Since Q_A acts as a clock for Q_B, each time the waveform at Q_A goes from low to high, the output of the flipflop B will toggle.

Since Q_B acts as a clock for Q_C, each time the waveform at Q_B goes from low to high, the output of the flipflop C will toggle.

Since Q_C acts as a clock for Q_D, each time the waveform at Q_C goes from low to high, the output of the flipflop D will toggle.

Design a counter for the sequence $0 \to 4 \to 1 \to 2 \to 6 \to 0 \to 4$ using SR flip-flop.

Present State			Next State			Excitation Table					
Q_C	Q_B	Q_A	Q_{C+1}	Q_{B+1}	Q_{A+1}	S_C	R_C	S_B	R_B	S_A	R_A
0	0	0	1	0	0	0	1	x	0	x	0
1	0	0	0	0	1	1	0	x	0	0	1
0	0	1	0	1	0	x	0	0	1	1	0
0	1	0	1	1	0	0	1	0	x	x	0
1	1	0	0	0	0	1	0	1	0	x	0

For S_A

Q_C \ Q_BQ_A	00	01	11	10
0	x	1	x	x
1	0	x	x	x

$S_A = \overline{Q_C}$

For R_A

Q_C \ Q_BQ_A	00	01	11	10
0	0	0	x	0
1	1	x	x	0

$R_A = \overline{Q_B} Q_C$

For S_B

Q_C \ Q_BQ_A	00	01	11	10
0	x	0	x	0
1	x	x	x	1

$S_B = Q_C$

For R_B

Q_C \ Q_BQ_A	00	01	11	10
0	0	1	x	x
1	0	x	x	0

$R_B = Q_A$

For S_C

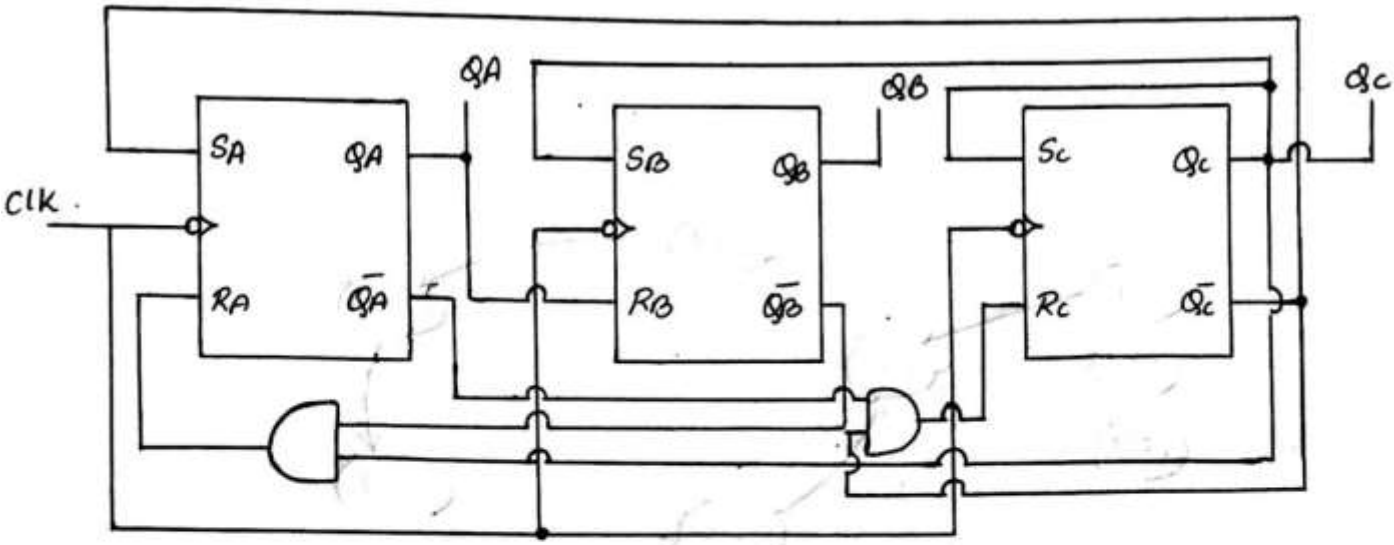
Q_C \ Q_BQ_A	00	01	11	10
0	0	x	x	0
1	1	x	x	1

$S_C = Q_C$

For R_C

Q_C \ Q_BQ_A	00	01	11	10
0	1	0	x	1
1	0	x	x	0

$R_C = \overline{Q_A}\,\overline{Q_C}$

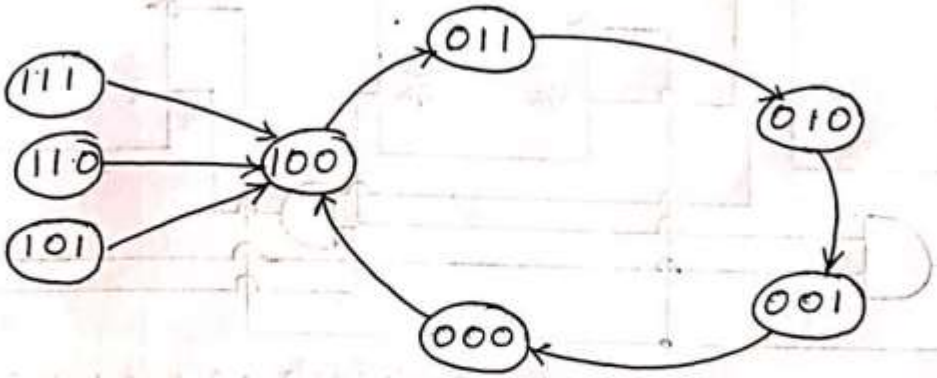


Counter Design for the sequence $0 \to 4 \to 1 \to 2 \to 6 \to 0 \to 4$

Differentiate b/w Asynchronous and Synchronous Counters.

Asynchronous Counters	Synchronous Counters.
i) In this type of counter the flipflops are connected in such a way that the o/p of first flipflop drives the clock for the next flipflop.	i) In this type of counter there is no connection between the o/p of the first flipflop. and the clock input of the next flipflop.
ii) All the flipflops are not clocked simultaneously.	ii) All the flipflops are clocked simultaneously.
iii) Logic circuit is very simple even for more number of states.	iii) Design involves complex logic circuit as the number of states increases.
iv) Main drawback of these counters is their low speed, as the clock is propagated through number of flipflops before it reaches the last flipflop.	iv) As the clock is simultaneo - usly given to all the flip flops, there is no problem of propagation delay. Hence they are high speed counters.

Design a self correcting Mod 5 Synchronous Docon counter using JK flipflops. Assume 100 as the next state for all the unused states.



Present State			Next State							
Q_C	Q_B	Q_A	Q_{C+1}	Q_{B+1}	Q_{A+1}	J_C	K_C	J_B	K_B	J_A	K_A
1	1	1 (7)	1	0	0 (4)	x	0	x	1	x	1
1	1	0 (6)	1	0	0 (4)	x	0	x	1	0	x
1	0	1 (5)	1	0	0 (4)	x	0	0	x	x	1
1	0	0 (4)	0	1	1 (3)	x	1	1	x	1	x
0	1	1 (3)	0	1	0 (2)	0	x	x	0	x	1
0	1	0 (2)	0	0	1 (1)	0	x	x	1	1	x
0	0	1 (1)	0	0	0 (0)	0	x	0	x	x	1
0	0	0 (0)	1	0	0 (4)	1	x	0	x	0	x

For J_A

Q_C \ Q_BQ_A	00	01	11	10
0	0	x	x	1
1	1	x	x	0

$$J_A = Q_B\bar{Q_C} + \bar{Q_B}Q_C$$

For K_A

Q_C \ Q_BQ_A	00	01	11	10
0	x	1	1	x
1	x	1	1	x

$$K_A = 1$$

For J_B

Q_C \ $Q_B Q_A$	00	01	11	10
0	0	0	×	×
1	1	0	×	×

$J_B = \bar{Q_A} Q_C$

For K_B

Q_C \ $Q_B Q_A$	00	01	11	10
0	×	×	0	1
1	×	×	1	1

$K_B = \bar{Q_A} + Q_C$

For J_C

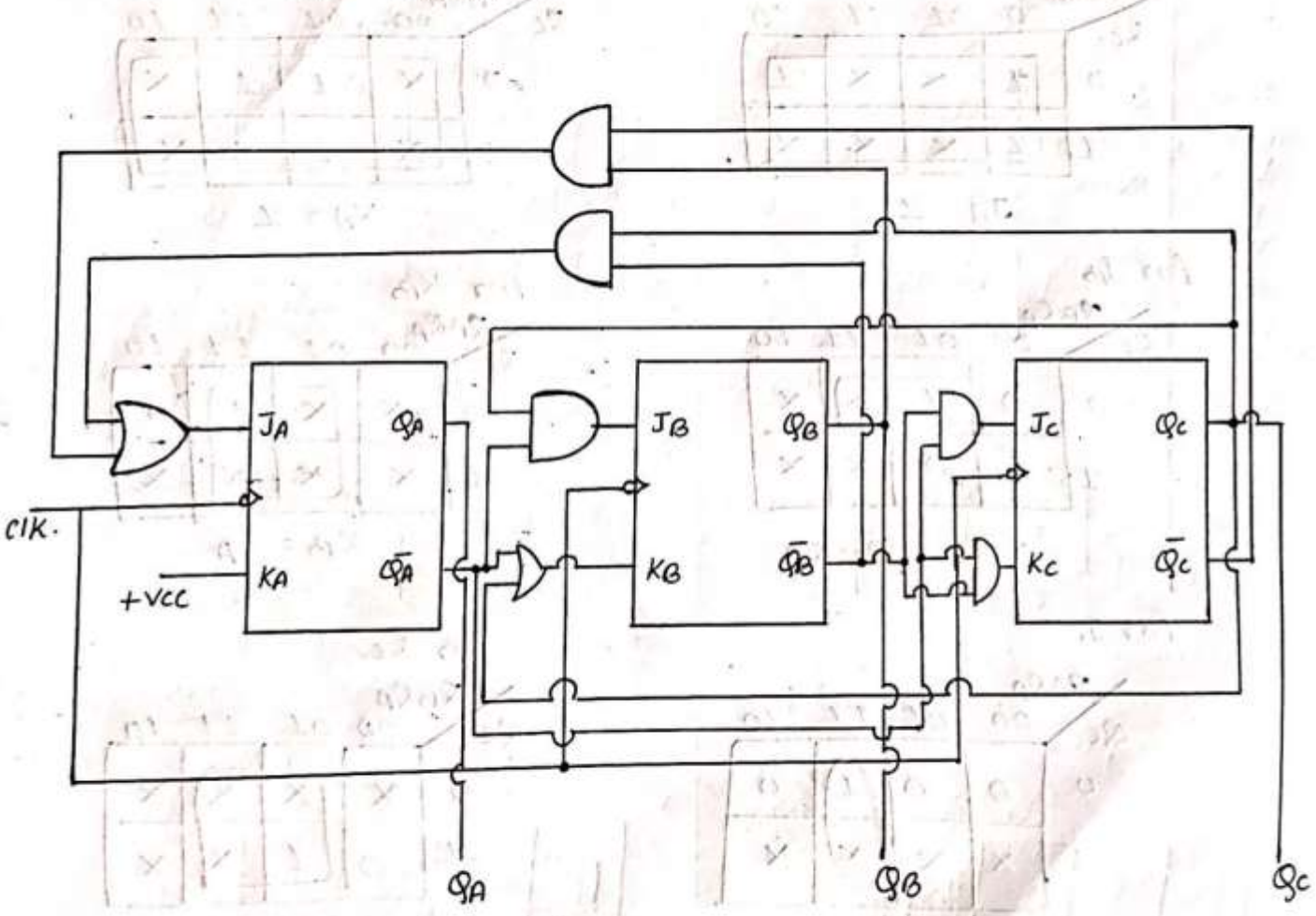
Q_C \ $Q_B Q_A$	00	01	11	10
0	1	0	0	0
1	×	×	×	×

$J_C = \bar{Q_A}\,\bar{Q_B}$

For K_C

Q_C \ $Q_B Q_A$	00	01	11	10
0	×	×	×	×
1	1	0	0	0

$K_C = \bar{Q_A}\,\bar{Q_B}$



Design a Synchronous Mod-6 counter using a JK flipflop

Present State			Next State						
Q_C	Q_B	Q_A	Q_{C+1}	Q_{B+1}	Q_{A+1}	$J_C K_C$	$J_B K_B$	$J_A K_A$
0	0	0	0	0	1	0 x	0 x	1 x
0	0	1	0	1	0	0 x	1 x	x 1
0	1	0	0	1	1	0 x	x 0	1 x
0	1	1	1	0	0	1 x	x 1	x 1
1	0	0	1	0	1	x 0	0 x	1 x
1	0	1	0	0	0	x 1	0 x	x 1

for J_A

Q_C \ Q_BQ_A	00	01	11	10
0	1	x	x	1
1	1	x	x	x

$J_A = 1$

for K_A

Q_C \ Q_BQ_A	00	01	11	10
0	x	1	1	x
1	x	1	x	x

$K_A = 1$

for J_B

Q_C \ Q_BQ_A	00	01	11	10
0	0	1	x	x
1	0	0	x	x

$J_B = \bar{Q_C} Q_A$

for K_B

Q_C \ Q_BQ_A	00	01	11	10
0	x	x	1	0
1	x	x	x	x

$K_B = Q_A$

for J_C

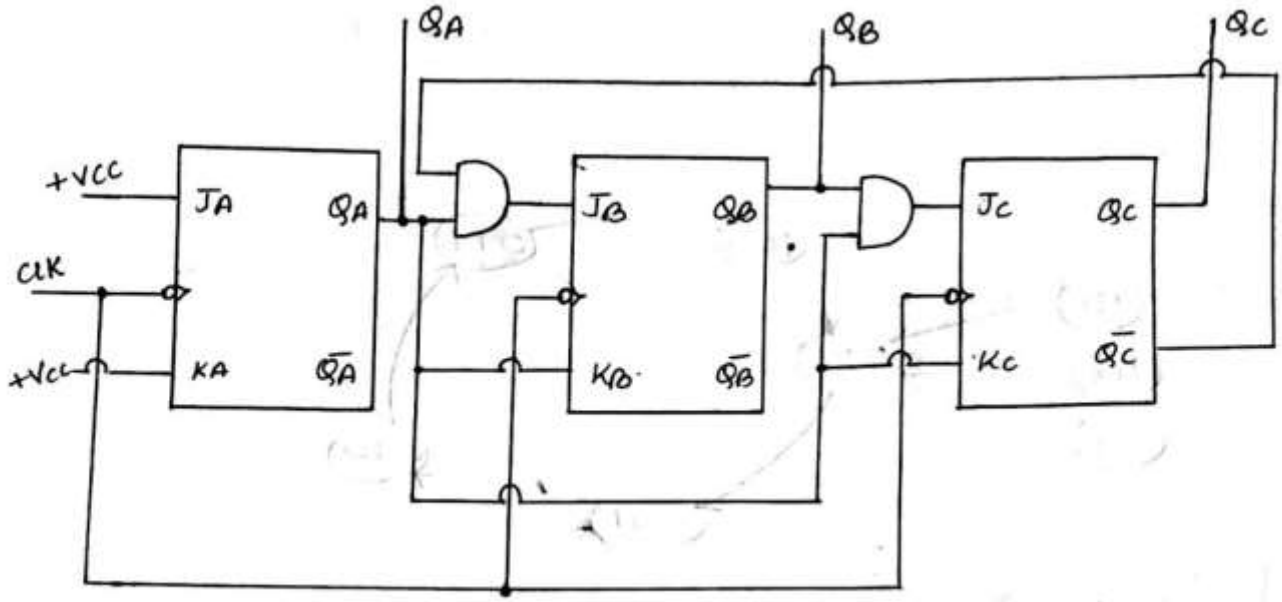
Q_C \ Q_BQ_A	00	01	11	10
0	0	0	1	0
1	x	x	x	x

$J_C = Q_B Q_A$

for K_C

Q_C \ Q_BQ_A	00	01	11	10
0	x	x	x	x
1	0	1	x	x

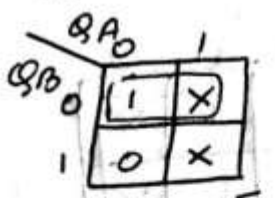
$K_C = Q_A$



Design a Synchronous Mod-3 counter with the following binary sequence using clocked JK flipflops ~~counter~~ counter sequence 0, 1, 2, 0, 1, 2.

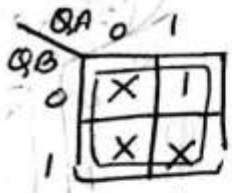
Present State		Next State		JA	KA	JB	KB
Q_B	Q_A	Q_{B+1}	Q_{A+1}				
0	0	0	1	1	X	0	X
0	1	1	0	X	1	1	X
1	0	0	0	0	X	X	1

for JA



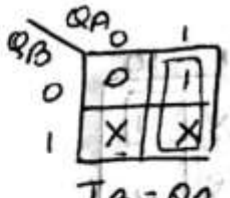
$J_A = \bar{Q}_B$

for KA



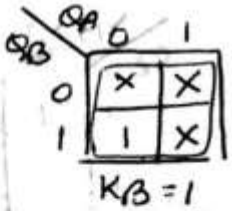
$K_A = 1$

for J_B

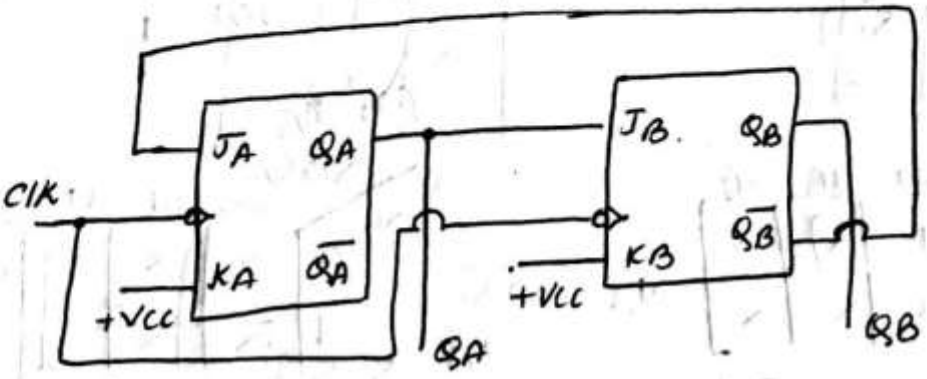


$J_B = Q_A$

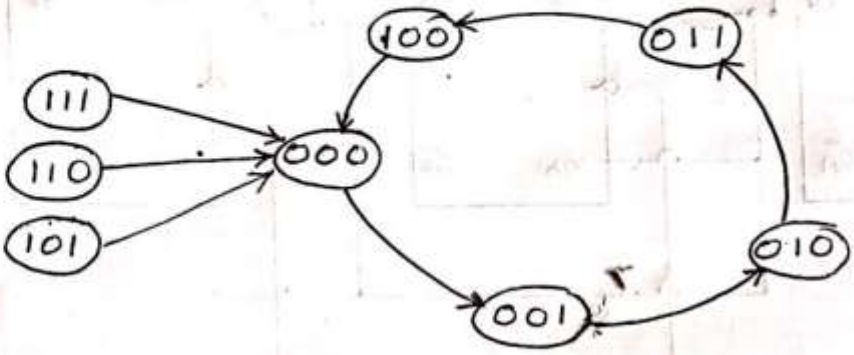
for KB



$K_B = 1$

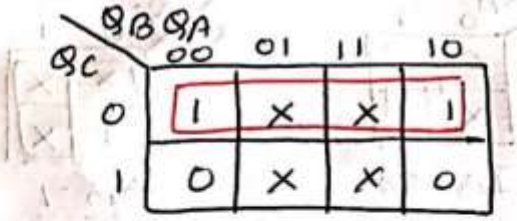


Design a self correcting mod-5 counter using JK flip flops and all the unused states must lead to the state $Q_A Q_B Q_C = 000$.



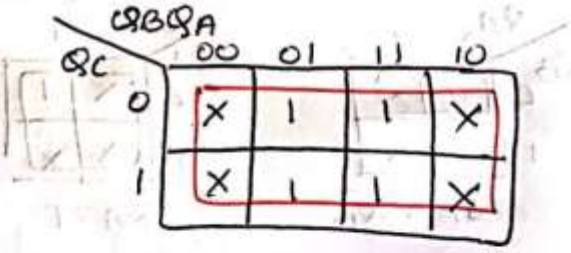
Present State			Next State			Excitation Table.		
Q_C	Q_B	Q_A	Q_{C+1}	Q_{B+1}	Q_{A+1}	J_C K_C	J_B K_B	J_A K_A
0	0	0	0	0	1	0 ×	0 ×	1 ×
0	0	1	0	1	0	0 ×	1 ×	× 1
0	1	0	0	1	1	0 ×	× 0	1 ×
0	1	1	1	0	0	1 ×	× 1	× 1
1	0	0	0	0	0	× 1	0 ×	0 ×
1	0	1	0	0	0	× 1	0 ×	× 1
1	1	0	0	0	0	× 1	× 1	0 ×
1	1	1	0	0	0	× 1	× 1	× 1

For J_A



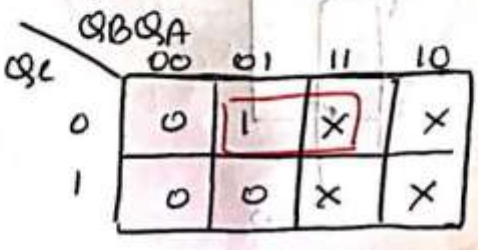
$J_A = \overline{Q_C}$

For K_A



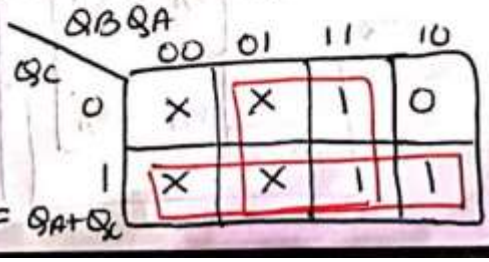
$K_A = 1$

For J_B



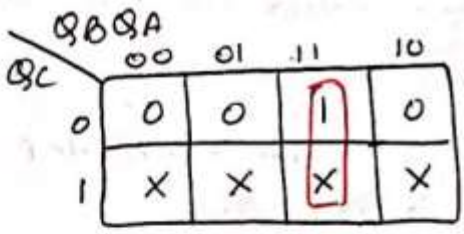
$J_B = Q_A \overline{Q_C}$

For K_B



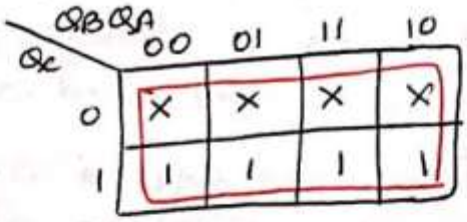
$K_B = Q_A + Q_C$

for Jc

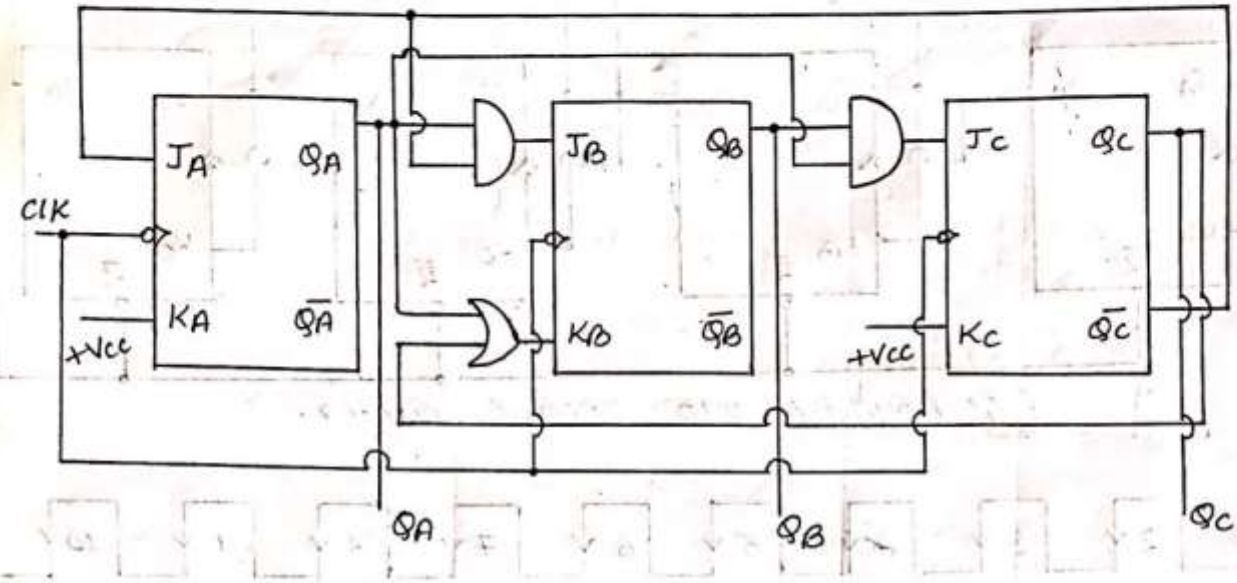


$J_C = Q_A Q_B$

for Kc



$K_C = 1$



Design and Explain the working of a Asynchronous decade counter with a suitable circuit diagram.

ASYNCHRONOUS DECADE COUNTER -

The binary counter has the maximum no. of states equal to 2^n, where n is the no. of flipflops in the counter.

Counters can also be designed to have a no. of states in their ~~seq~~ sequence that is less than 2^n.

In Decade counters, the sequence is truncated upto 10 states, 0000 (0 in decimal) through 1001 (9 in decimal).

The truncation in the count sequence is achieved by resetting the counter at a particular count instead of going through all of its normal states.

In case of Asynchronous Decade counter, the counter is reset back to 0000 state after the 1001 state.

The resetting of the counter is done with the help of reset inputs of each flipflop. These inputs are activated when the desired state is reached.

In the below decade counter, reset input is activated using NAND gates when the 1010 state is reached.

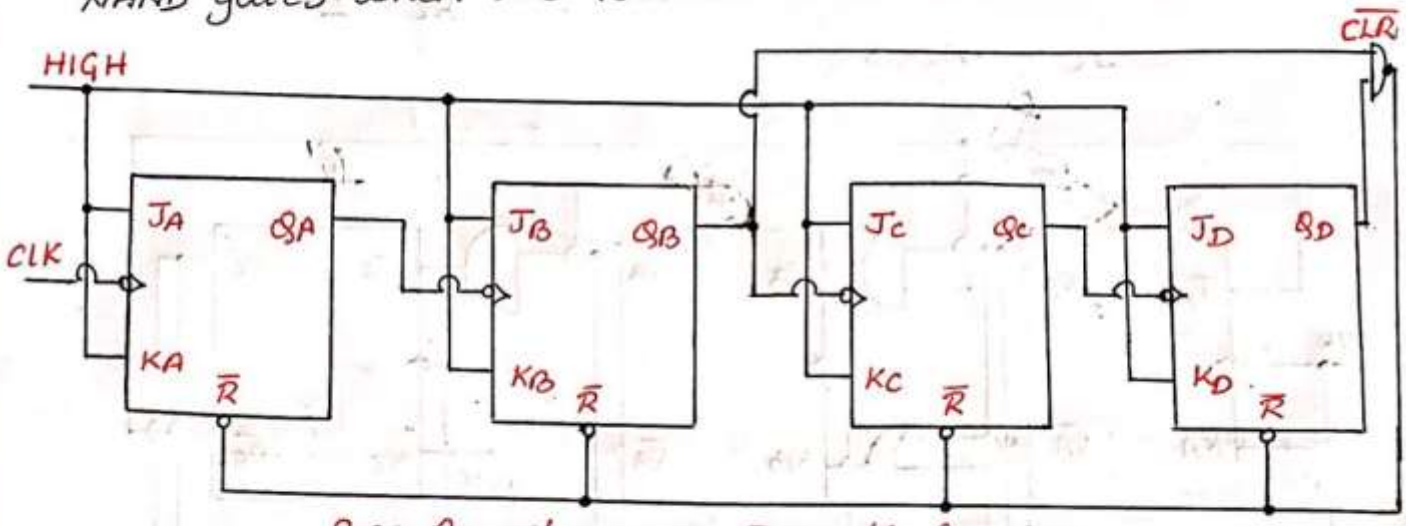


FIG: Asynchronous Decade Counter.

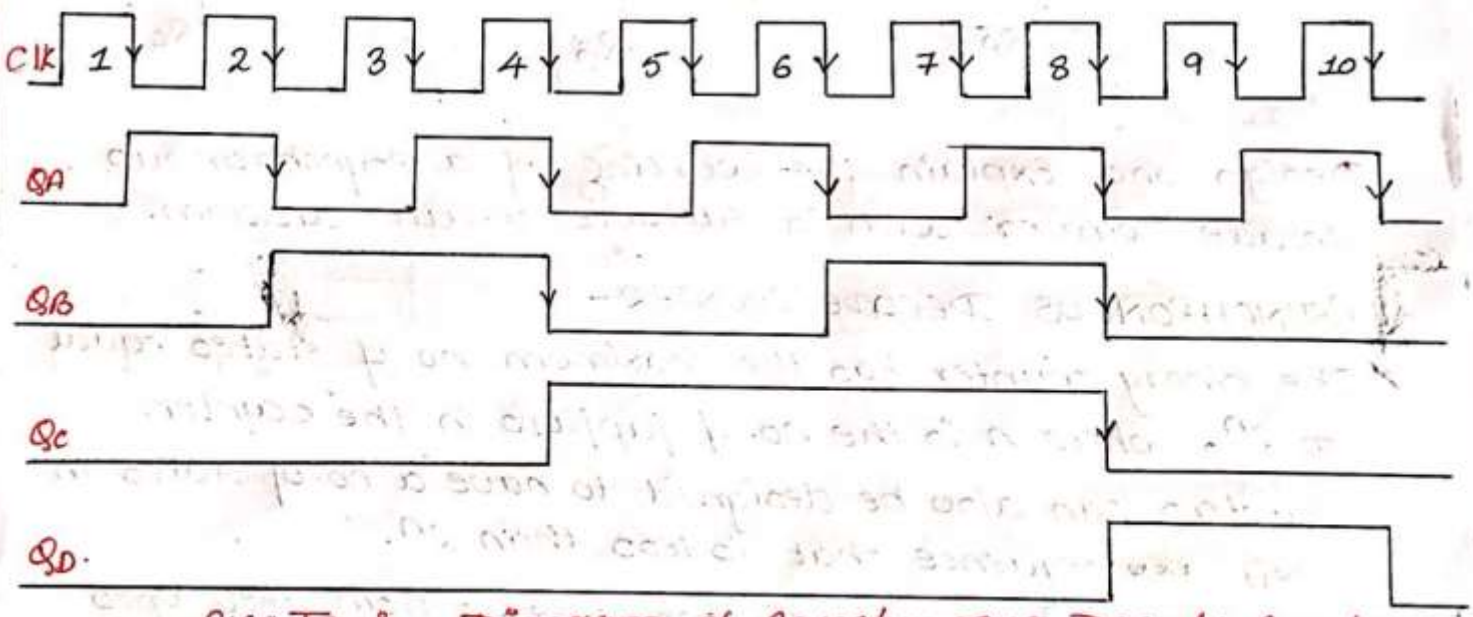


FIG: Timing Diagram of Asynchronous Decade Counter.

Count	Q_D	Q_C	Q_B	Q_A
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

Truth Table of Asynchronous Decade Counter.