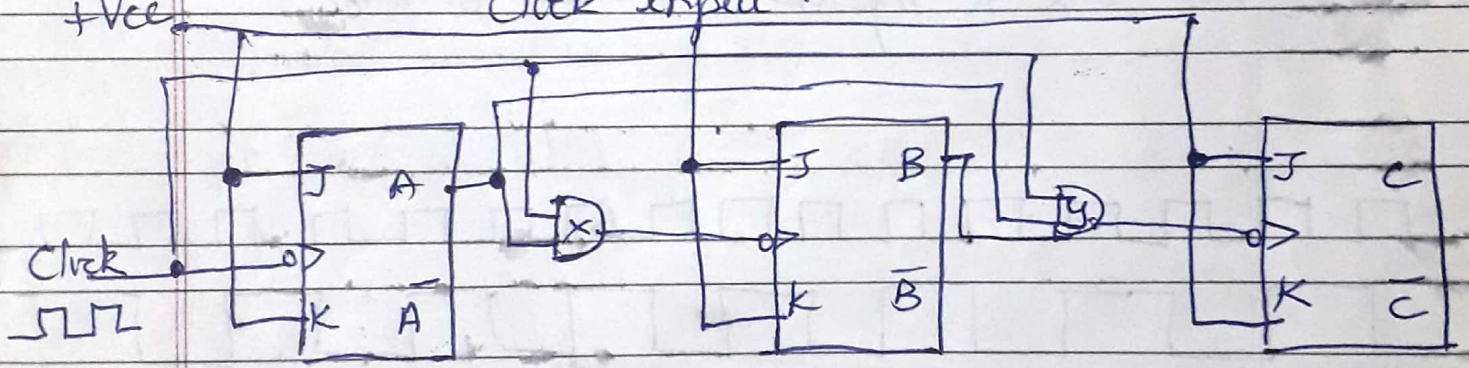
Page No:
Date: / /

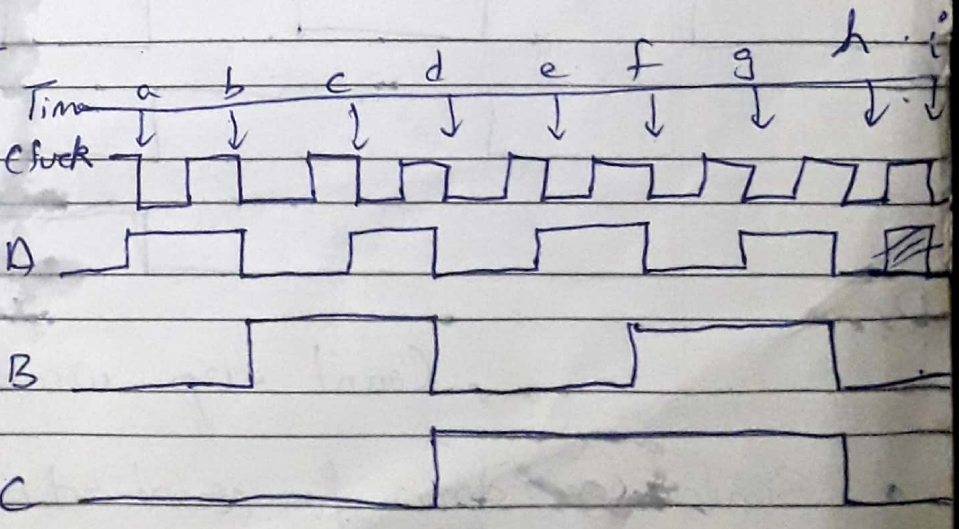
10.3 Synchronous Counters

We have seen in ripple counter, each flip flop has a delay time & settling time. These can be overcome by Synchronous counter where every flip flop is triggered in synchronism with the clock.

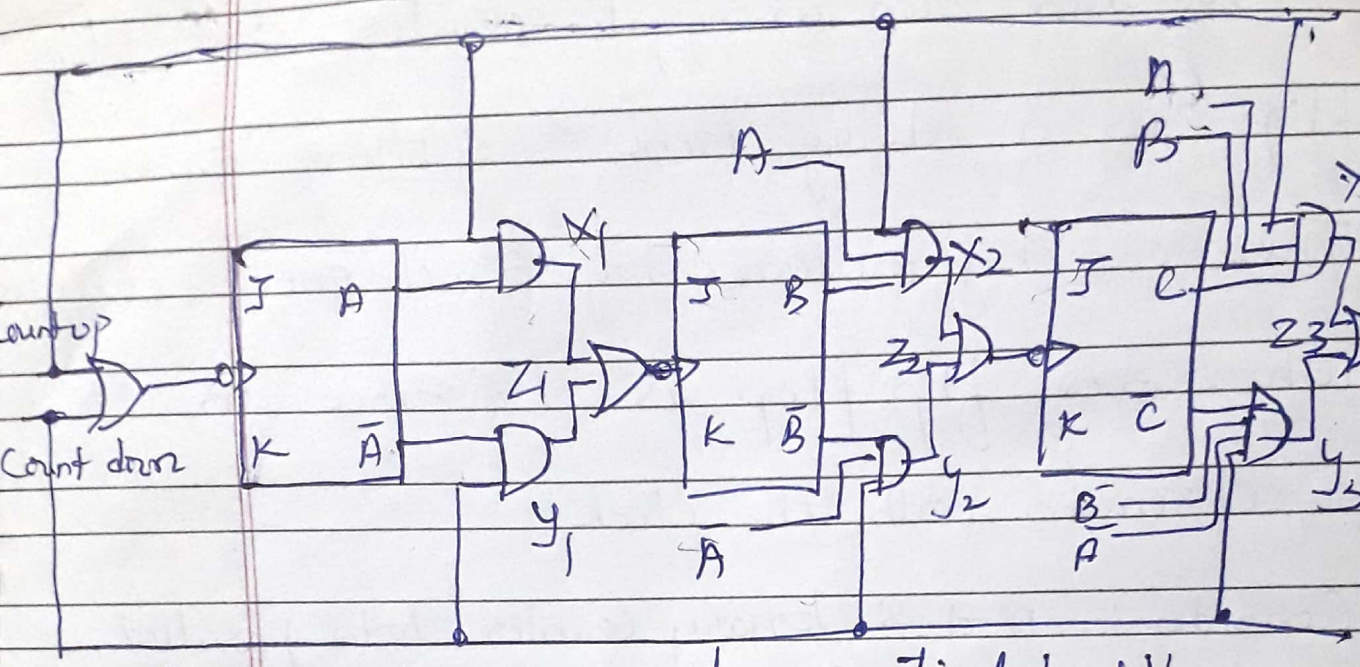
Consider: mod-8 binary counter with parallel clock input.



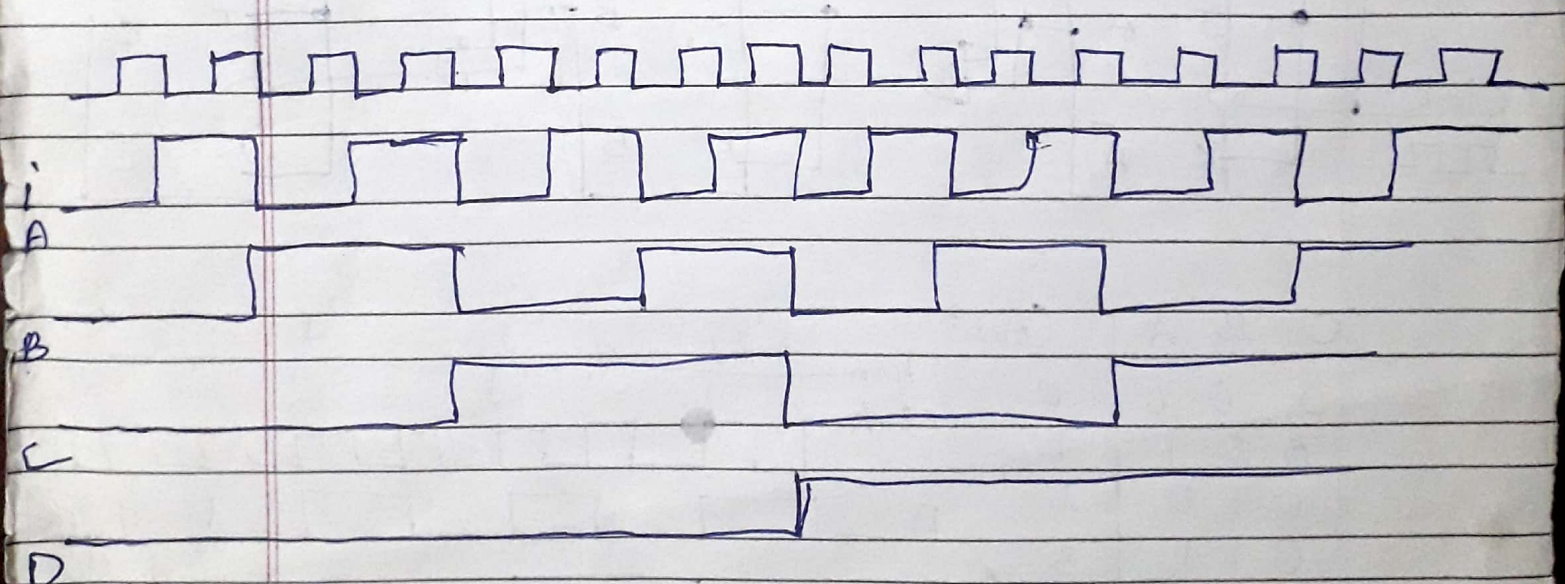
C	B	A	Count
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7
0	0	0	0



A parallel up-down counter



★ All J & K inputs are tied to $+V_{cc}$.



Count -up waveform.

- Similar can draw for count -down

10.4 Changing the Counter modulus.

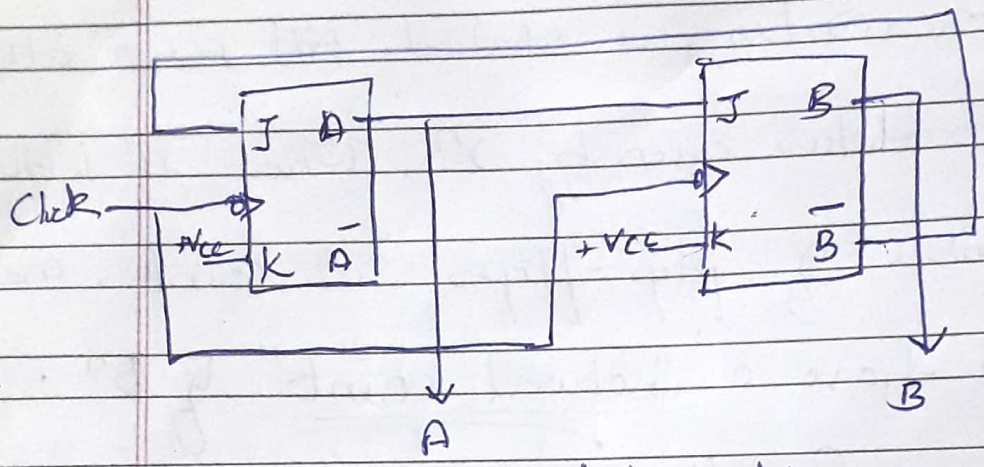
All of the counters we studied till now all have a modulus given by 2^n, where 'n' indicates the number of flip-flops. Such counters are said to have a "natural count" of 2^n.

Sometimes we may not to have counters having a modulus of 3 or 5. These can be constructing using larger modulus counts by skipping states. Such counters are said to have a "modified count".

The number of flip-flops in such counter is determined by choosing the lowest natural count that is greater than the desired modified count.

eg mod-7 counter requires three flip-flops.

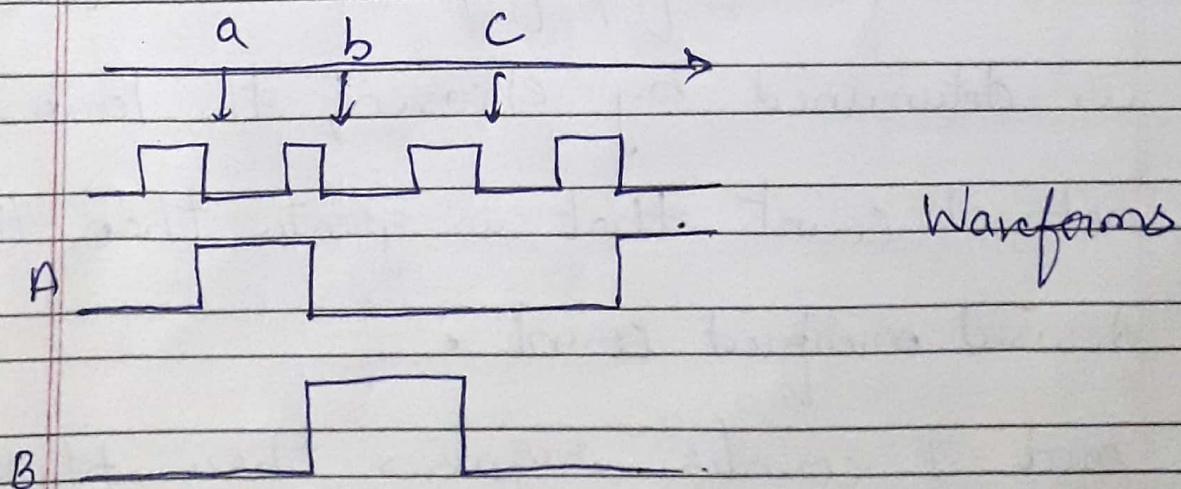
mod-3 counter



logic diagram

B	A	Count
0	0	0
0	1	1
1	0	2
0	0	0

Truth table



Waveforms

Working :-

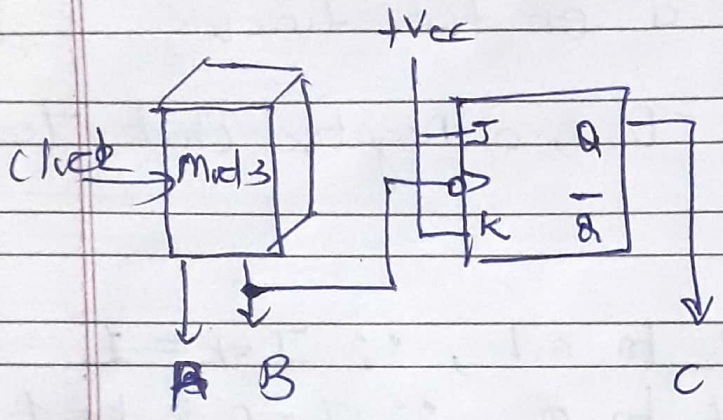
i) Prior to point 'a' on time line,
A = 0 and B = 0, a negative clock transition will cause,
→ A to toggle to a 1, ∵ J = k = 1
→ B to reset to 0, ∵ J = 0 & k = 1

ii) Prior to point 'b' on time line,
A = 1 and B = 0, a negative clock transition will cause,
→ A to toggle to a 0, ∵ J = k = 1
→ B to toggle to 1, ∵ J = k = 1.

iii) Prior to point 'c' on time line,
A = 0 and B = 1, a negative clock transition will cause,
→ A to reset to 0, ∵ J = 0 & k = 1
→ B to reset to 0, ∵ J = 0 & k = 1

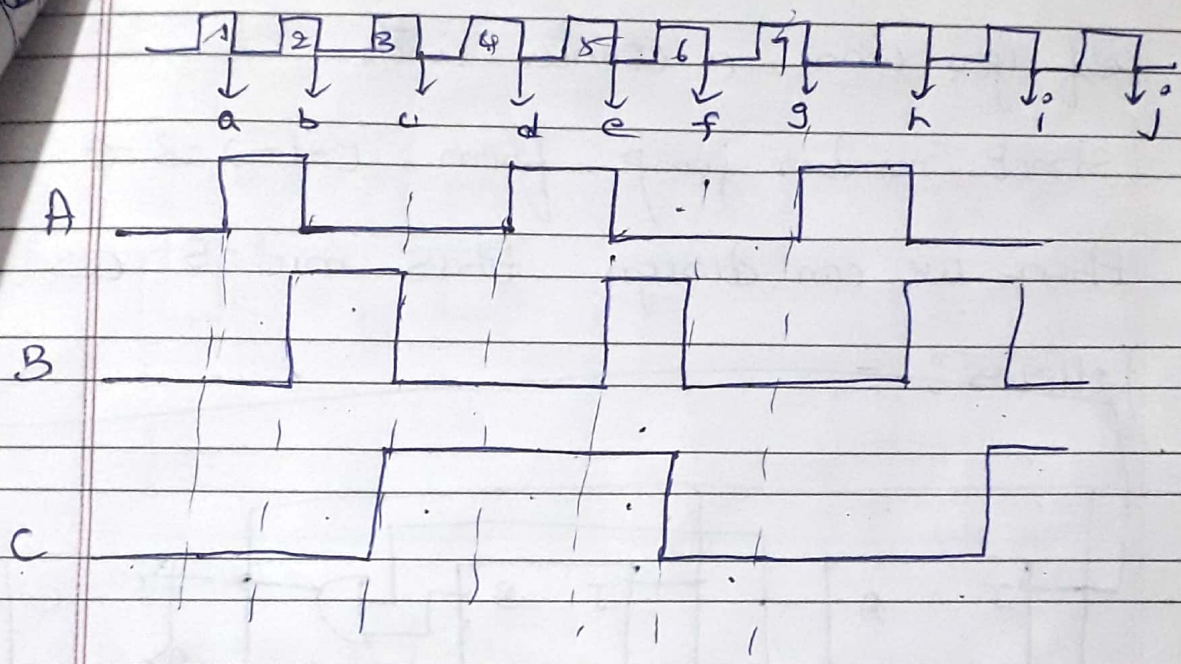
Mod-6 Counter

can be constructed by $(3 \times 2)_{mod} = 6\text{-mod}$.



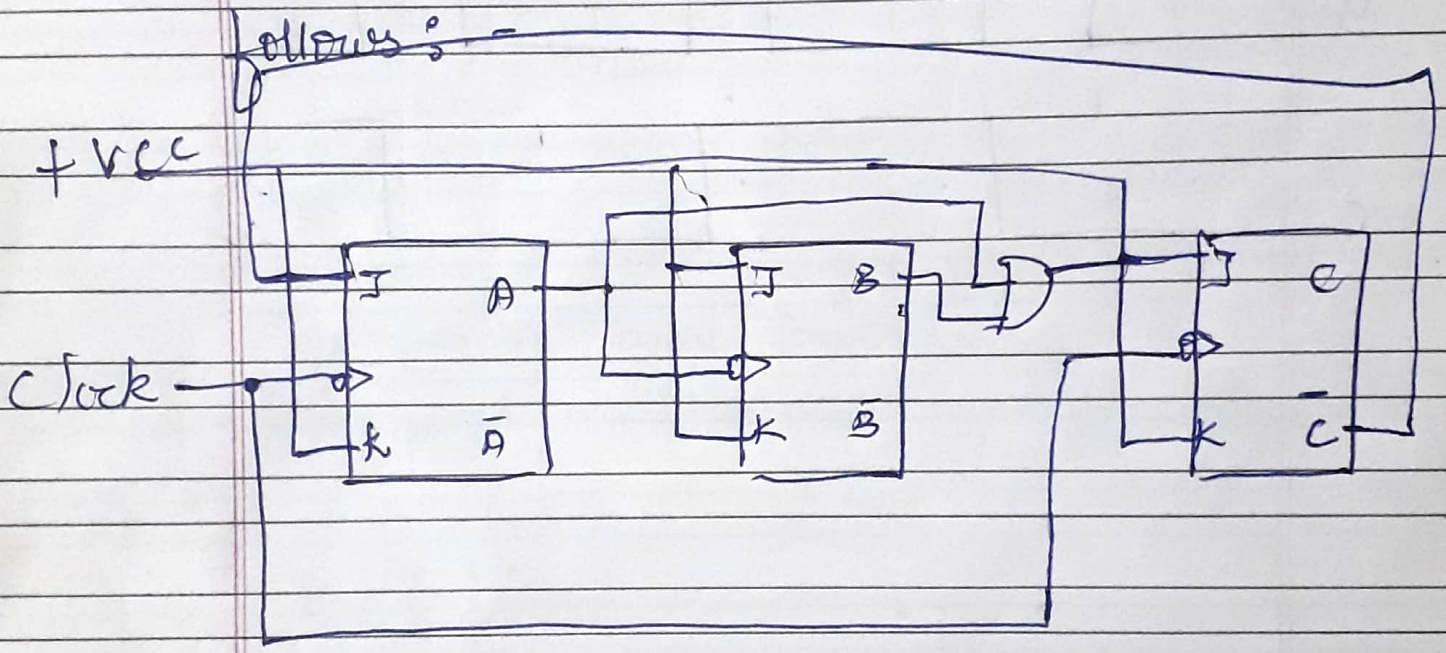
This is no more Synchronous Counter :
Flipflop C is ~~triggered~~ by flip flop B

C	B	A	Count
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~~ ✓
0	0	0	0

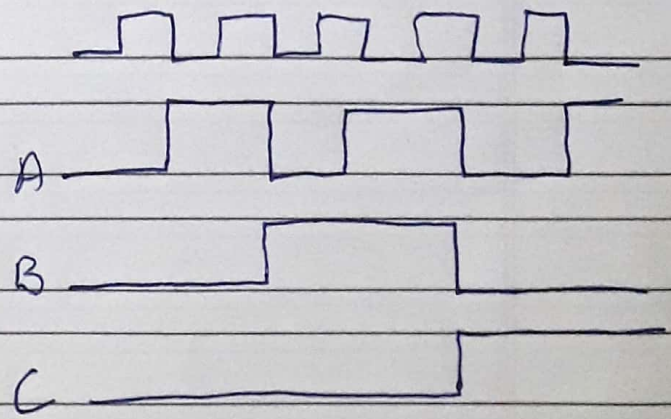


10.5 Decade Counters

If you want a counter with strict number jump from 0-1-2-3-4 then we can design this mod-5 as follows:-



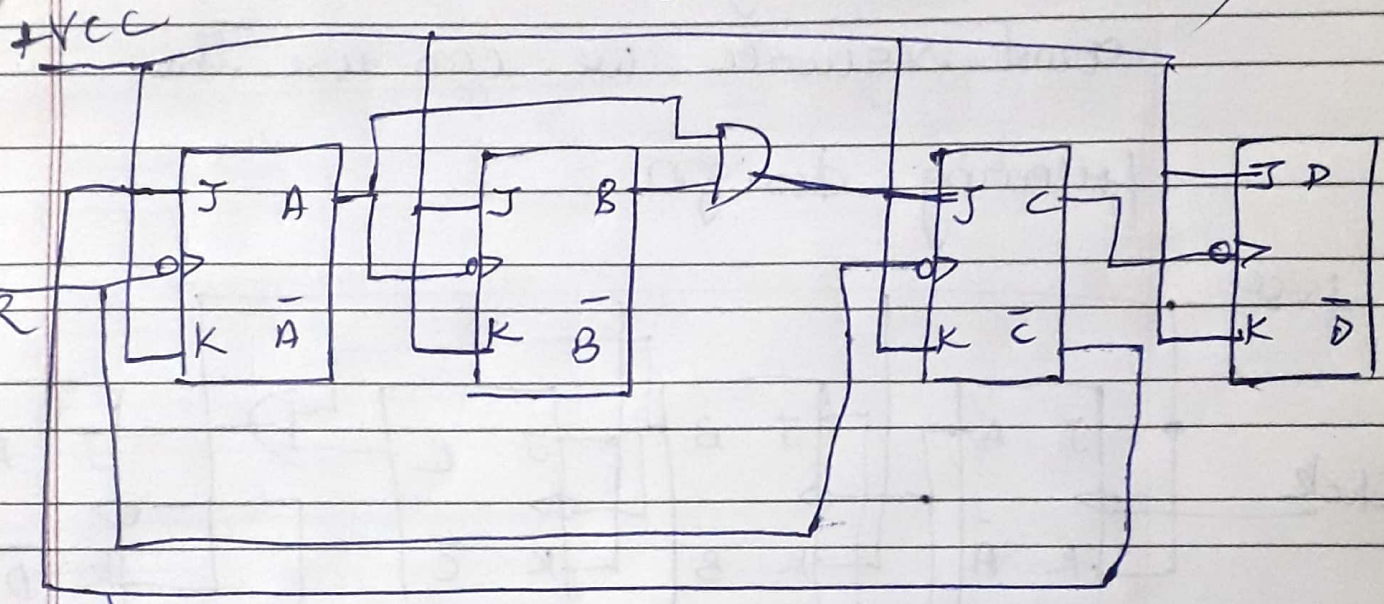
C	B	A	Count
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
0	0	0	0



2-ways

PAGE NO:
DATE: / /

1. Mod-10 Counter (A decade counter)

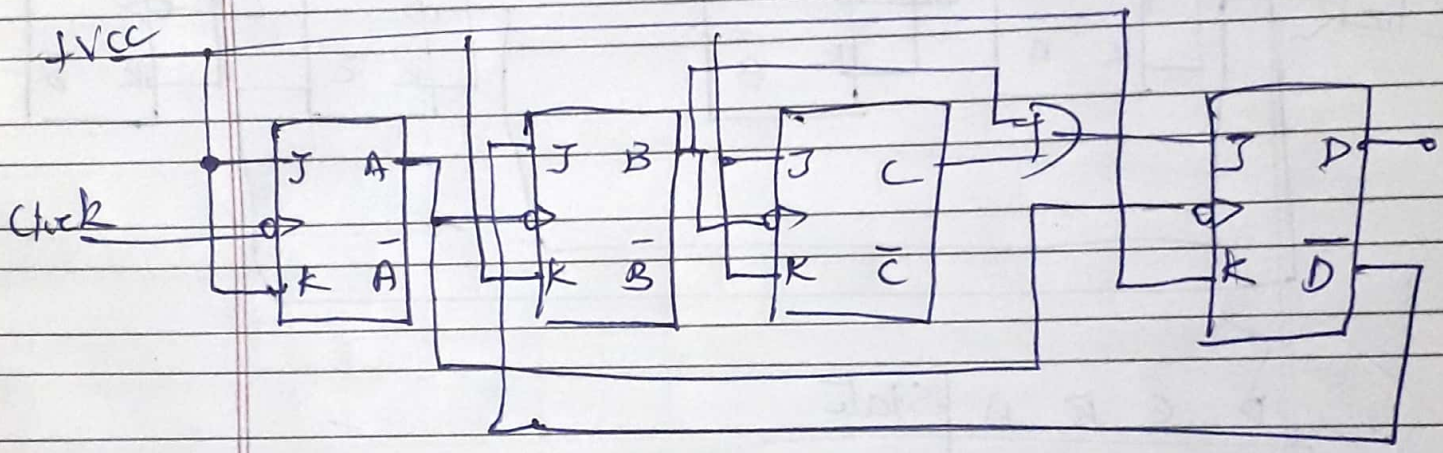


To be analysed by students

D	C	B	A	State
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
1	0	0	0	5
1	0	0	1	6
1	0	1	0	7
1	0	1	1	8
1	1	0	0	9
0	0	0	0	

★ note the change

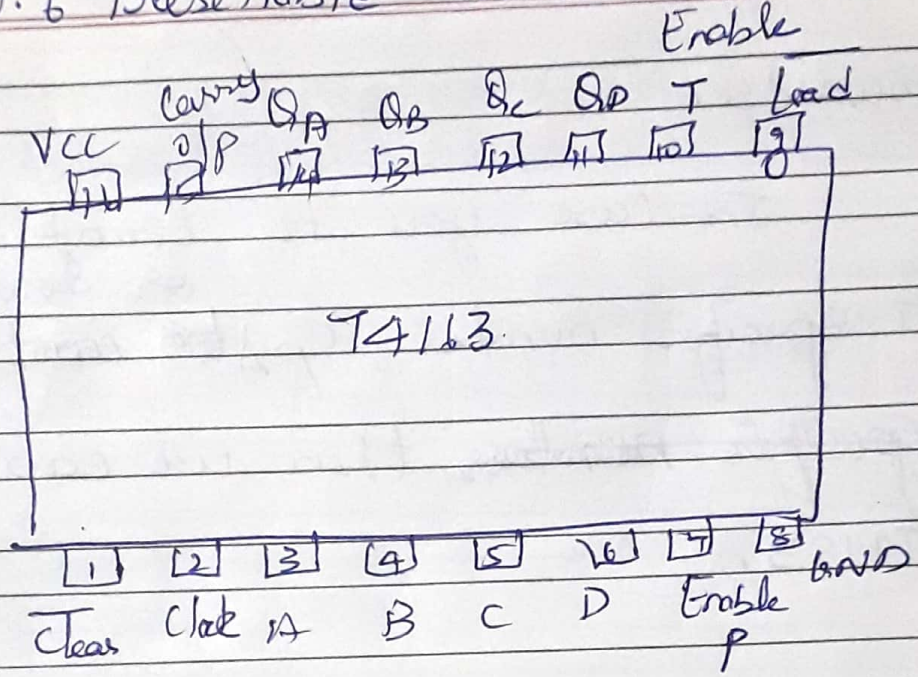
II. By cascading & also to get continuous count sequence we can use the following design.



D	C	B	A	Count
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
0	0	0	0	0

PAGE NO:
DATE: / /

10.6 Presettable Counters



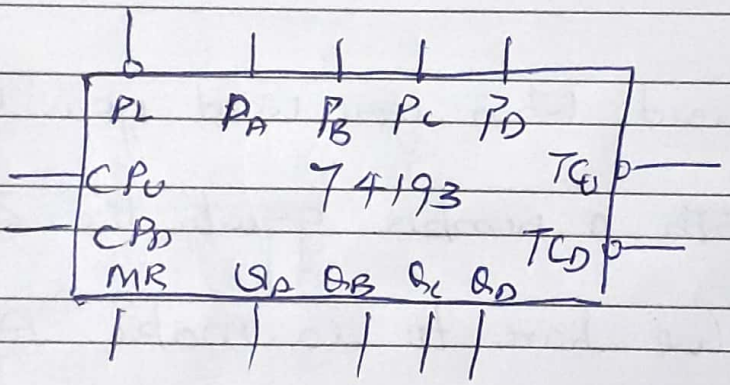
This is used when you want your counter to start with a number greater the Zero. To do so all we have to is make A B C D with binary number we wish to see at & $\overline{Load} = 0$ (as it is active low signal). Suppose A B C D is set to 1010, then counter will be set to 1010 ~~with~~ providing $\overline{Load} = 0$.

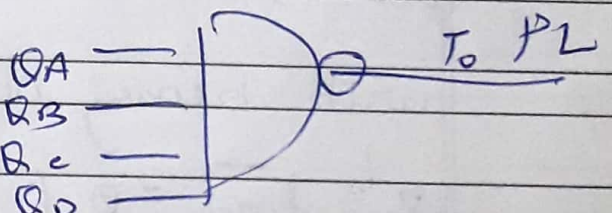
Now if you want counting to continue from here then make $\overline{Load} = 1$, immediately after setting it, so that it counts from 1010

10. Design of sequential

onwards.

In case you to count from or down a specific number upto another specific number then we can use 74193.





MR – Master reset

CPU – Clock Pluse upper

CPD – clock Pluse down

TCU – Cascading arrangement when carry generated while doing up count

TCD – ——————— down count.

PL – Parallel load

PA, PB, PC, PD – Inputs / QA ... QD – Output