

A counter is the most versatile and useful subsystems in the digital system. A counter driven by a clock can be used to count the number of clock cycles. Since clock pulses occur at known intervals, the counter can be used as an instrument for measuring time and therefore period and frequency. There are 2 types of counters: synchronous and asynchronous.

Example: All processors contain a program counter / PC.

- Program consists of a list of instructions that are to be executed one after another.
- PC keeps track of the instruction currently being executed.
- PC increments once on each clock cycle, and the next program instruction is executed.

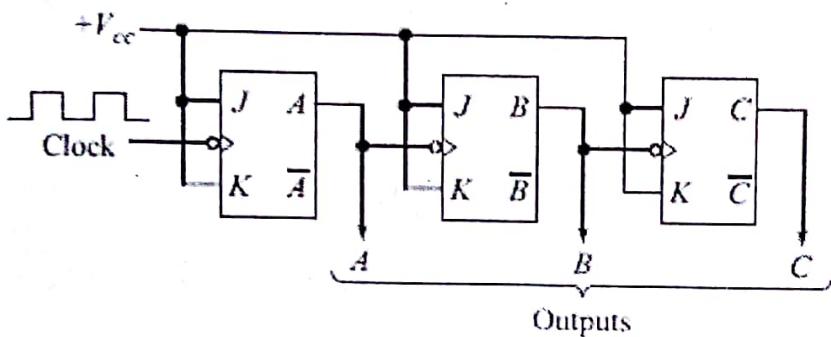
Counters could be serial or parallel counters or combination of both. Counter operates in count up mode, count down mode. When counter are cleared, then all the flip-flops are cleared and they contain zero. Counters are preset such that the contents of the flip-flop represent any desired binary number.

### Asynchronous Counter

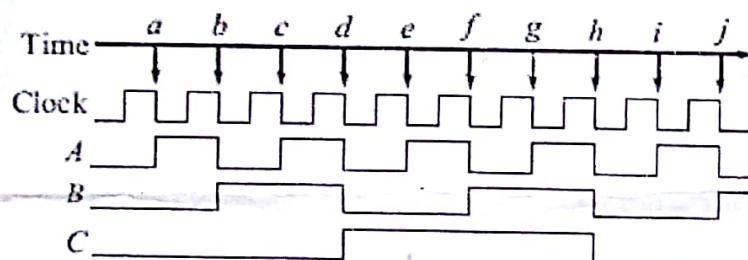
Asynchronous counter is a digital circuit in which flip-flop within the counter do not change states at exactly the same time becoz they do not have common clock pulse.

Events that do not have fixed time relationship with each other and generally do not occur at the same time.

A binary ripple counter can be constructed using clocked JK flip-flops.



(a) Three-bit binary ripple counter



(b) Waveforms

Negative clock transitions	C	B	A	State or count
---	0	0	0	0
a	0	0	1	1
b	0	1	0	2
c	0	1	1	3
d	1	0	0	4
e	1	0	1	5
f	1	1	0	6
g	1	1	1	7
h	0	0	0	0

(c) Truth table

Above fig. shows 3 negative edge triggered JK flip-flop connected to form a 3 bit binary ripple counter. The system clock, a square wave drives flip-flop A. The output of A drives B and the output of B drives flip-flop C. All the J and k inputs are tied to +Vcc. This means that each flip-flop will toggle with a negative transition of its clock. The flip flop A must change state before it can trigger the B flip-flop and the B flip-flop has to change state before it triggers the C flip-flop.

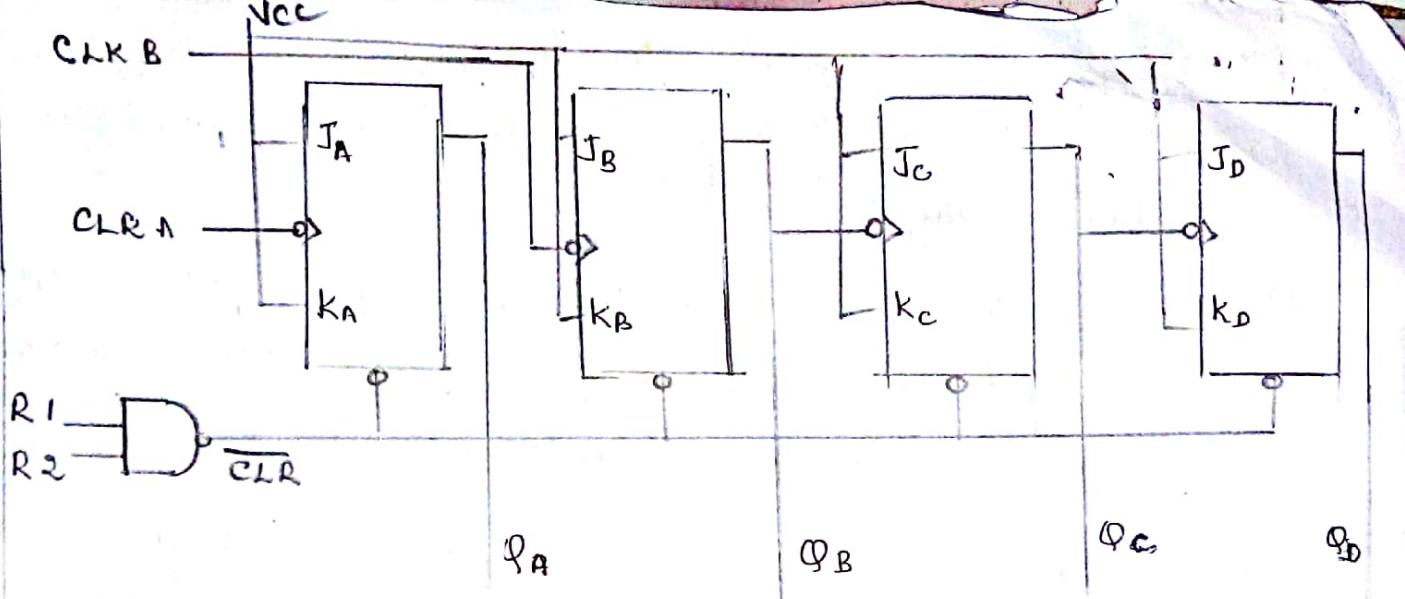
The triggers move through the flip-flops like a ripple in water. So the overall propagation delay time is the sum of individual delays.

Every time there is a clock  $\text{NT}$ , flip-flop A will change states. Since A acts as the clock for B each time the waveform at A goes low, flip-flop B will toggle.

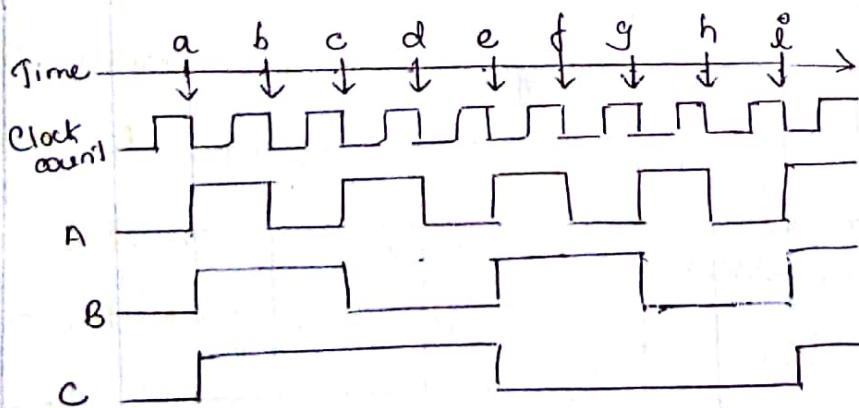
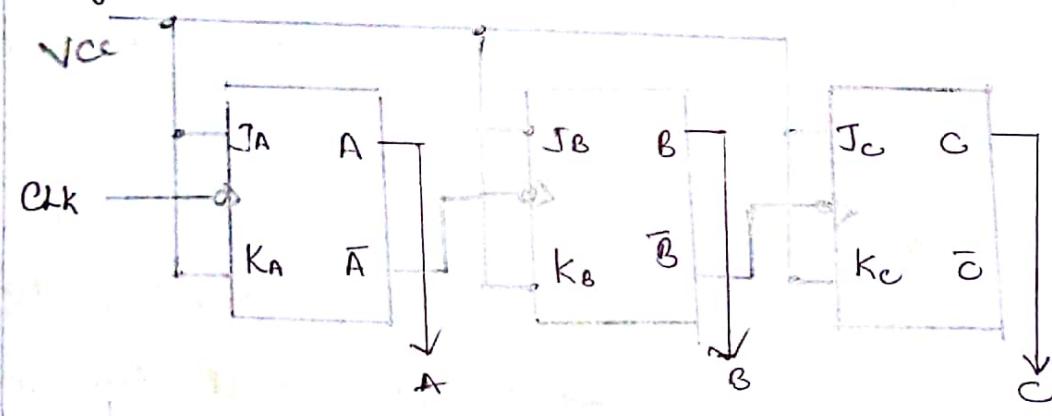
Since a binary ripple counter counts in a straight binary sequence, it is easy to see that a counter having  $n$  flip-flop will have  $2^n$  output conditions. A 3 flip-flop counter is often referred to as a modulus-8 counter since it has 8 states.

#### 4 Bit Asynchronous Binary Counter:

T493A is an 4 bit asynchronous binary counter. An application specific integrated circuit (ASIC) dedicated to 4 bit asynchronous binary counting. It has 4 internal JK FFs. A 2 input NAND gate for clearing all FFs. A 4 bit binary counter can be used in either a mod8/mod16 configuration. If the clock is applied at input CKB the output configuration. If the clock is applied at input CKA and FF QA is connected to input CKB then it is a mod16, 4 bit binary ripple counter. The outputs are  $Q_A, Q_B, Q_C$  and  $Q_D$ . All the FF in T493A have direct reset inputs that are active low. When high at both reset all FFs will reset which is regardless of clock.



Asynchronous / Ripple Down Counter:



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

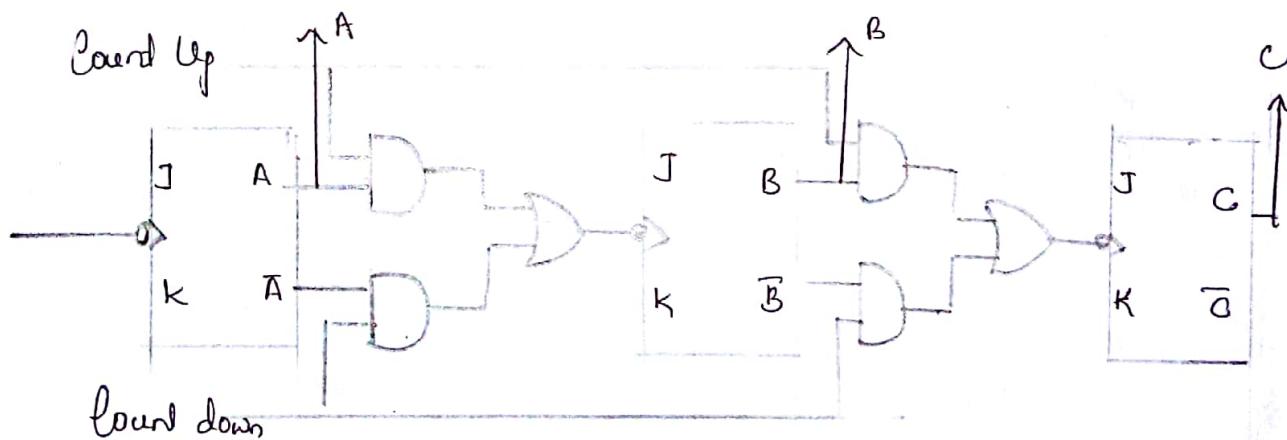
The system clock is used as clock input to FF A, but complement of A i.e  $\bar{A}$  is used as clock for B FF, likely  $\bar{B}$  is used to drive FF C.

FF A toggles with each NT but FF B will toggle when F goes high i.e negative transition on A triggers B. Similarly FF C is triggered by  $\bar{B}$  and so C will toggle.

when B goes high. The counter becomes  $ABC = 111$  at point a and decrease by one count with each clock transition. Hence it is called down counter.

### 3 bit Asynchronous Up Down Counter:

It counts in a straight binary sequence is shown in below fig. It is simply combination of 2 counters up and down counter. If count down is low and count up is high, each flip flop will be triggered from the true side of previous FF. and the counter will <sup>work in</sup> count up mode. If count down is high and count up is low, each flip-flop will be triggered from the complement side of the previous flip flop. and counter will work in count down mode.



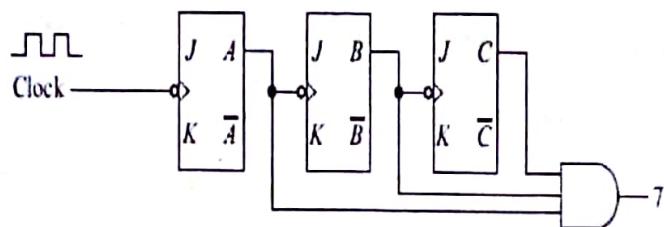
$$(J=K=1)$$

### Decoding Gates:

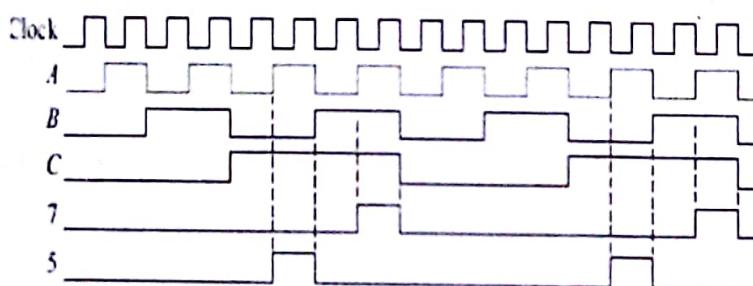
A decoding gate can be connected to the output of a counter in such a way that the output of the gate will be high or low only when the counter contents are equal to a given state.

For instance, the decoding gate connected to the 3 bit

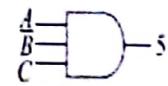
ripple counter as shown fig will decode state  $C_2C_1C_0 = 111$   
 Thus gate output will be high only when  $A=1$ ,  $B=1$ , and  $C=1$   
 The other 7 states of the counter can be decoded in a  
 similar fashion.



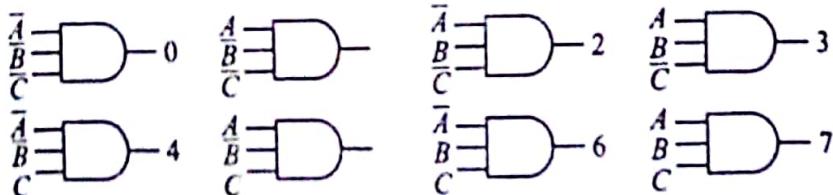
(a) Decoding gate for state 7



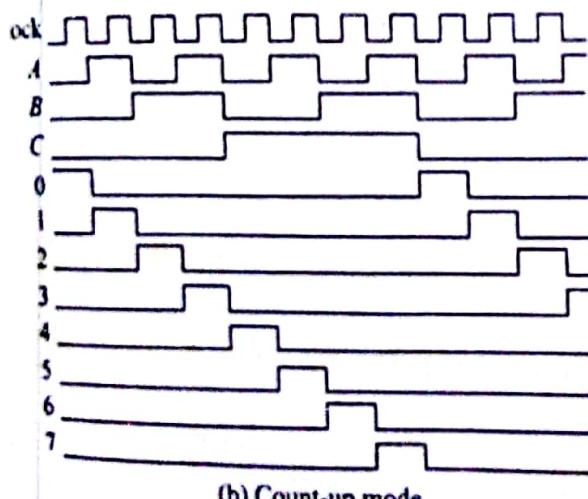
(b) Waveforms



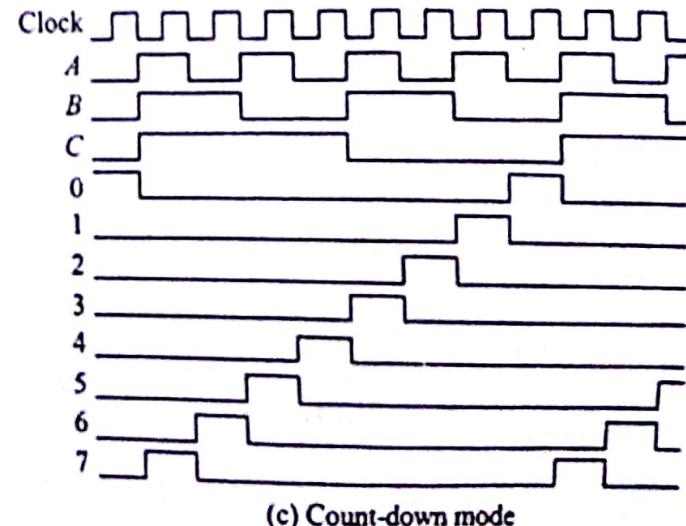
(c) Gate to decode state 5



(a) Gates



(b) Count-up mode



(c) Count-down mode

## Drawbacks In Asynchronous Counter:

1. There is a limit to its highest operating frequency.
2. Each flip-flop has a delay time. These delay times are additive and the total 'settling' time for the counter is approximately the delay time times the total number of flip-flops.
3. There is the possibility of glitches occurring at the output of decoding gates and with a counter

Synchronous Counter: In this all flip-flop change states simultaneously since all clock inputs are driven by the same clock. i.e all the flip-flops change state in synchronism.

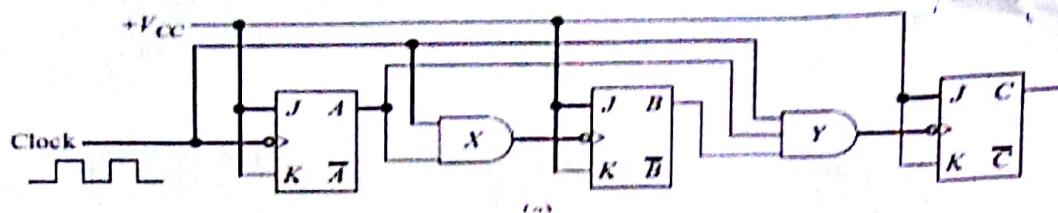
## Synchronous 3 bit Up Counter (Parallel Binary Counter)

In this all flip-flop change states simultaneously since all clock inputs are driven by the same clock i.e all flip-flop change state in synchronism. A 3 bit counter can be used to count the number of clock transitions up to a maximum of 7.

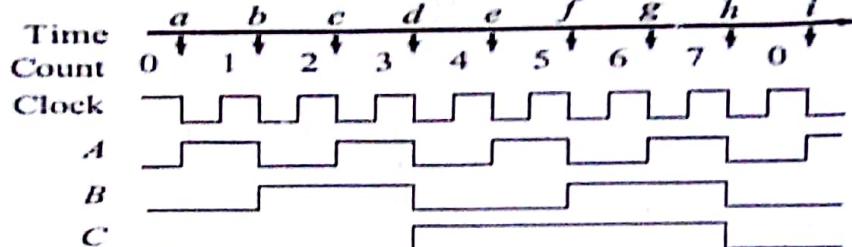
Synchronous 3 bit up-counter is constructed using one negative edge triggered JK-flip-flops. (3 flip-flop). The J and K inputs of all flip-flops are maintained High by connecting them to Vcc, hence all the flip-flop will toggle with a NT.

Then AND gates are used to gate every odd clock to flip-flop B every 2<sup>1</sup>th clock to flip-flop C and so on. This configuration is referred as 'steering logic' since the clock pulses are gated/stored to each individual flip-flop.

In parallel counter the time at which any flip-flop changes state is determined by the states of all previous flip-flops in counter.



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



The clock is applied directly to the flip-flop A, as result flip-flop A changes state with each NT. Whenever A is high AND gate X is enabled and hence it transmits the clock pulse to flip-flop B. Whenever A & B are high, AND gate Y is enabled and hence it transmits the clock pulse to flip-flop C.

#### Synchronous 4 bit Up Down Counter :

To operate in the count-up mode, the system clock is applied to the count-up input, while the count down input is held low

To operate in the count-down mode, the system clock is applied to the count down input while count up is held low

If the count down line is LOW, the lower AND gates  $Y_1, Y_2$  and  $Y_3$  are disabled. The clock applied at count up will then go directly into flip-flop A and will be steered into the other flip-flops by AND gates  $X_1, X_2$ , and  $X_3$ .

If the count up line = LOW, the upper AND gates  $X_1, X_2$  and  $X_3$  are disabled. The clock applied at input count down will go directly into flip-flop A & will be steered into the other flip-flops by AND

gates  $Y_1, Y_2, Y_3$ .

### Module 5

Changing the Counter Modulus: Modulus is defined as the number of states through which a counter can progress.

All the asynchronous/synchronous counters studied so far have the ability to operate in either a count up or count down mode and they all have a modulus given by  $2^n$ , which indicates the number of flip-flop. Such counter are said to have a "natural count" of  $2^n$ .

It is often desirable to construct counters having a modulus other than 2, 4, 8 and so on. For example counter having a modulus of 3 or 5. A small modulus counter can always be constructed from a larger modulus counter by skipping states. Such counters are said to have a modified count.

It is 1st necessary to determine the number of flip-flops required. The correct number of FFs is determined choosing the lowest natural count that is greater than the desired modified count. For example mod 7 requires 3FFs, since 8 is the lowest natural count greater than the desired modified count of 7.

#### Mod 3 Counter:

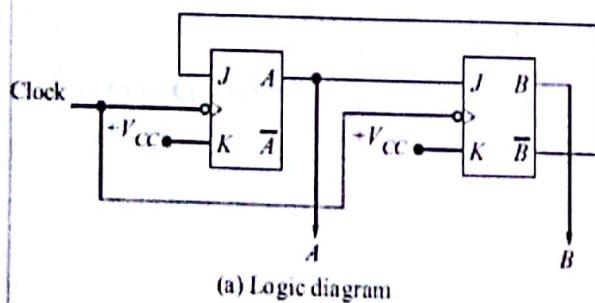
mod 3 requires 2FFs, since 4 is the lowest natural count greater than the desired modified count of 3, this counter skips one state.

#### Working of mod 3 counter.

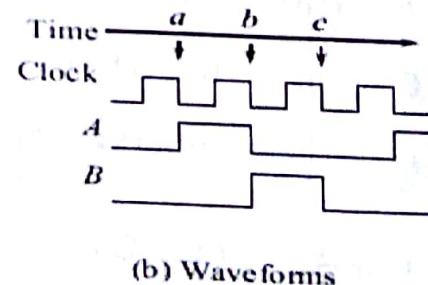
A mod 3 counter can count in sequence from 00 to 11 ( $00 \rightarrow 01 \rightarrow 10$ ) It has a clock input and outputs at A and B.

Steps:

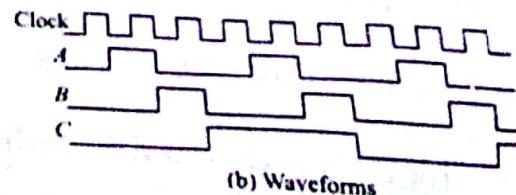
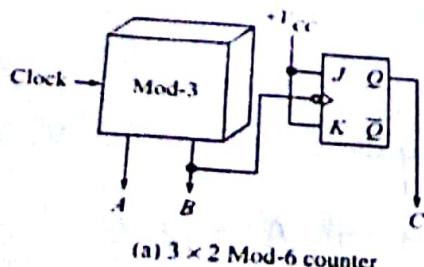
1. Prior to point 'a' on time line A=0 and B=0. A negative clock pulse at 'a' will cause:
- 'A' to toggle to a 1, since its  $J=K=1$
  - 'B' to reset to 0, since its  $J=0 \& K=1$
2. Prior to point 'b' on time line A=1 and B=0, A NT at b will cause:
- 'A' to toggle to a 0, since its  $J=K=1$
  - 'B' to toggle to a 1, since its  $J=K=1$
3. Prior to point 'c' on time line A=0 and B=1, A NT at c will cause:
- 'A' to reset to a 0, since its  $J=0 \& K=1$
  - 'B' to toggle to a 0, since its  $J=0 \& K=1$
4. The counter has now progressed through all 3 of its states advancing one count with each NT.
- It can be considered as a divide by 3 block since the output waveform of B has a period equal to 3 times that of clocks.



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



Mod 6 counter



Mod 6 counter:

A basic FF is a mod 2 counter and

mod 4 counter is 2 FFs in series i.e. 2 mod 2 counter in series.

mod 8 counter is 3 mod 2 counters in series and so on.

Hence a greater number of higher modulus counter can be formed by using the product of any number of lower modulus counter.

Mod 6 counter can be formed by using the mod 3 counter in series with a flip-flop (mod 2 counter).

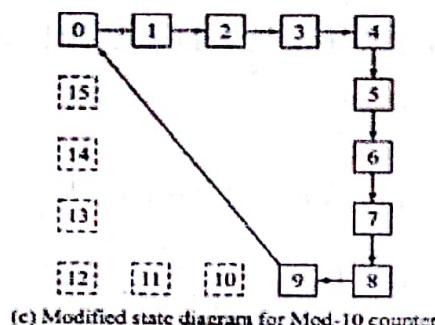
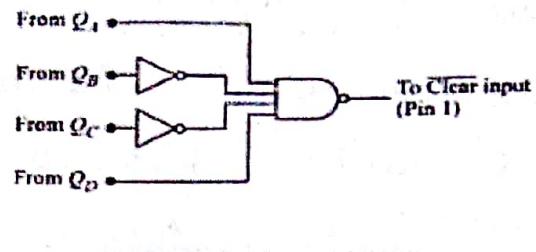
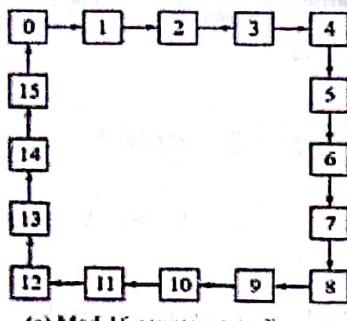
This logic circuit can no longer be considered a synchronous counter since flip-flop C is triggered by flip-flop B. i.e all FFs do not change states in synchronism with the clock.

Presetable Counter:

It is a counter incorporating logic such that it can be preset to any desired state i.e. it can be used to implement a counter that has any modulus.

Nearly all the presetable counters are constructed by using 4 FFs and they are generally referred to as 4-bit counter.

Presetable counters may be either synchronous or asynchronous. When connected such that the count advances in a natural binary sequence from 0000 to 1111, it is simply referred to as a binary counter.



## Counter Design As a Synthesis Problem:

Steps in design of synchronous counters:

- 1) Draw state sequence diagram of a mod counter.
- 2) Obtain the transition table using state sequence diagram
- 3) Find number of flip-flops required to build the counter
- 4) Choose the type of FFs to be used
- 5) Complete transition table using excitation table chosen FF.
- 6) Get the expression for flip-flop inputs using K-map simplification.
- 7) Draw the logic diagram.

Synchronous Mod 6 regular counter using negative edge triggered JK flip-flop.

FFs required :  $2^n \geq N$

$$N=6 \Rightarrow n=3$$

## Difference between combinatorial circuit and sequential circuit

### Combinational Logic Circuits

Output is a function of the present inputs  
(Time Independent Logic)

Do not have the ability to store data  
(state)

It does not require any feedback. It simply outputs the input according to the logic designed

Used mainly for Arithmetic and Boolean operations

Logic gates are the elementary building blocks

Independent of clock and hence does not require triggering to operate

Example: Adder [ $1+0=1$ : Dependency only on present inputs i.e., 1 and 0]

### Sequential Logic Circuits

Output is a function of clock present inputs and the previous states of the system

Have memory to store the present states that is sent as control input (enable) for the next operation

It involves feedback from output to input that is stored in the memory for the next operation

Used for storing data (and hence used in RAM)

Flip flops (binary storage device) are the elementary building unit

Clocked (Triggered for operation with electronic pulses)

Example: Counter [Previous O/P +1=Current O/P. Dependency on present input as well as previous state]

## Difference between asynchronous and synchronous counter

<b>asynchronous counter</b>	<b>synchronous counter</b>
Different flipflops are applied with different clocks	All flipflops are applied with same clock
It is slower in operation	It is faster in operation
fixed count sequence either up or down	any count sequence is possible
produces decoding error	produces no decoding error

## Race around condition of JK Flip Flop

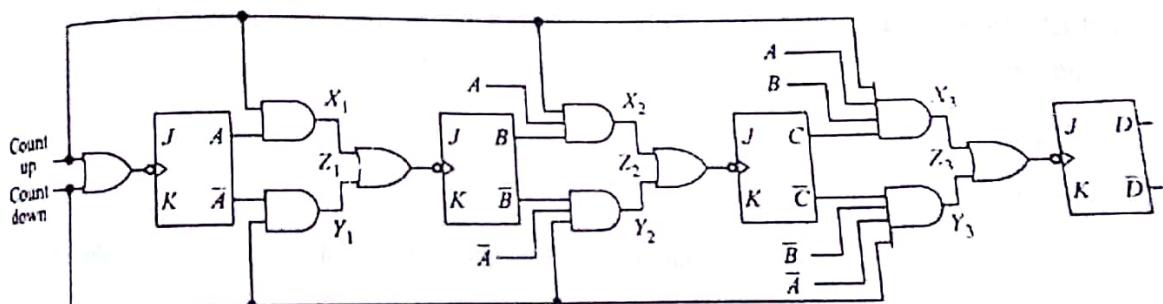
In JK flip flop as long as clock is high for the input conditions J&K equals to the output changes or complements its output from  $1 \rightarrow 0$  and  $0 \rightarrow 1$ . This is called toggling output or uncontrolled changing or racing condition. Consider above J&K circuit diagram as long as clock is high and  $J \& K = 11$  then two upper and lower AND gates are only triggered by the complementary outputs Q and  $Q'$ . I.e. in any

condition according to the propagation delay one gate will be enabled and another gate is disabled. If upper gate is disabled then it sets the output and in the next lower gate will be enabled which resets the flip flop output.

#### Steps to avoid racing condition in JK Flip flop:

1. If the Clock On or High time is less than the propagation delay of the flip flop then racing can be avoided. This is done by using edge triggering rather than level triggering.
2. If the flip flop is made to toggle over one clock period then racing can be avoided. This introduced the concept of Master Slave JK flip flop.

#### 3 bit Synchronous up down counter Logic diagram



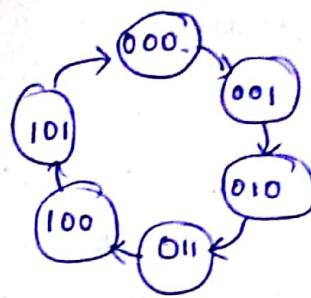
Note : All  $J$  and  $K$  inputs are tied to  $+V_{CC}$

(a) Logic diagram

Excitation for JK FF is

state sequence diagram of mod

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



Transition table:

$C_n$	$B_n$	$A_n$	$C_{n+1}$	$B_{n+1}$	$A_{n+1}$	$J_C$	$K_C$	$J_B$	$K_B$	$J_A$	$K_A$
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
1	1	0	X	X	X	X	X	X	X	X	X
1	1	1	X	X	X	X	X	X	X	X	X

K map Simplification:

	$\bar{B}\bar{A}$	$\bar{B}A$	$BA$	$B\bar{A}$
$\bar{C}$	0	X	X	1
C	4	5	7	6

$$J_A = 1$$

	$\bar{B}\bar{A}$	$\bar{B}A$	$BA$	$B\bar{A}$
$\bar{C}$	0	1	X	X
C	0	0	X	X

$$J_B = \bar{C}A$$

	$\bar{B}\bar{A}$	$\bar{B}A$	$BA$	$B\bar{A}$
$\bar{C}$	0	0	1	0
C	X	X	X	X

$$J_B = \bar{C}A$$

	$\bar{B}\bar{A}$	$\bar{B}A$	$BA$	$B\bar{A}$
$\bar{C}$	X	1	1	X
C	X	1	X	X

$$K_A = 1$$

	$\bar{B}\bar{A}$	$\bar{B}A$	$BA$	$B\bar{A}$
$\bar{C}$	X	X	1	0
C	X	X	X	X

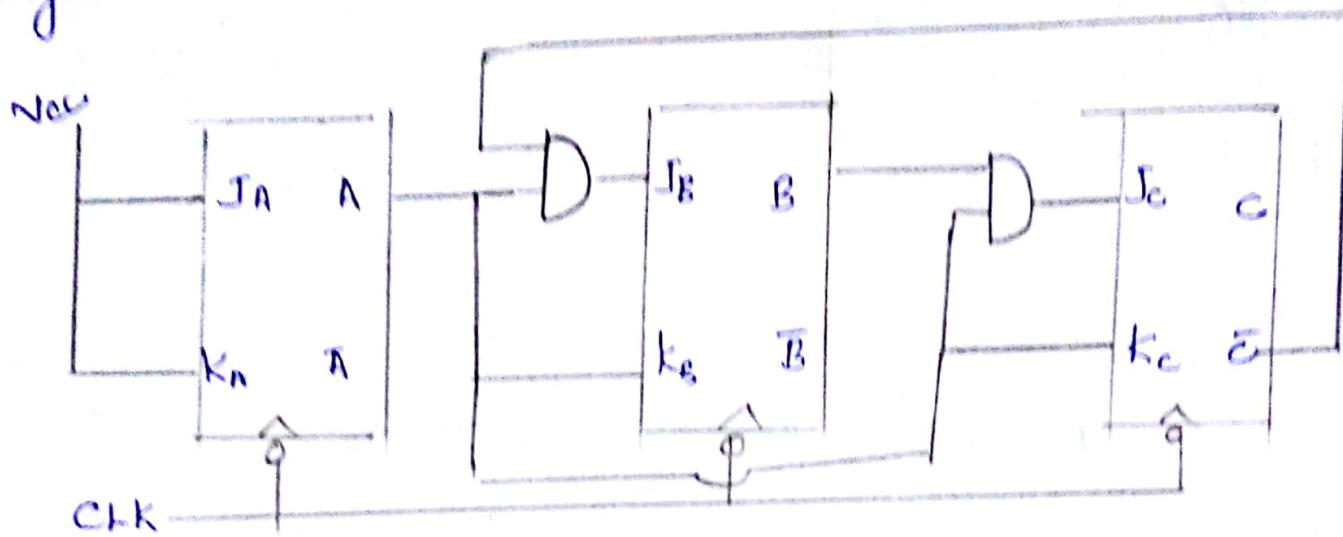
$$K_B = A$$

	$\bar{B}\bar{A}$	$\bar{B}A$	$BA$	$B\bar{A}$
$\bar{C}$	X	X	X	X
C	0	1	X	X

$$J_C = BA$$

$$K_C = A$$

Logic Circuit :

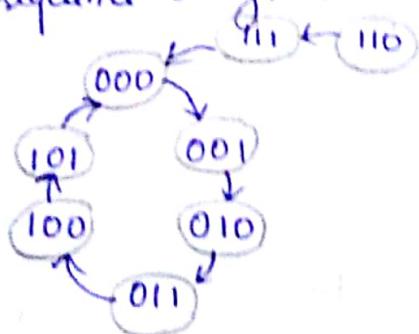


Design self correcting mod 6 counter among JK

$$2^n \geq N$$

$$N = 6 \Rightarrow n = 3$$

state sequence diagram



Excitation Table for JK

$Q_n$	$J_n$	$K_n$	$S$	$K$
0	0	0	0	X
0	1	1	1	X
1	0	1	X	1
1	1	1	X	0

Transition / Synthesis Table:

$C_n$	$B_n$	$A_n$	$en_{n+1}$	$B_{n+1}$	$A_{n+1}$	$J_C$	$K_C$	$J_B$	$K_B$	$J_A$	$K_A$
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
1	1	0	0	0	0	X	1	X	1	0	X
1	1	1	0	0	0	X	1	X	1	X	1

	$\bar{B}A$	$\bar{B}A$	$BA$	$BA$
$\bar{C}$	0	X	X	1
C	4	5	7	6

$$J_A = \bar{C} + \bar{B}$$

	$\bar{B}A$	$\bar{B}A$	$BA$	$BA$
$\bar{C}$	X	1	1	X
C	X	1	1	X

$K_A = 1$

	$\bar{B}A$	$\bar{B}A$	$BA$	$BA$
$\bar{C}$	0	1	X	X
C	0	0	X	X

$$J_B = \bar{C}, A$$

	$\bar{B}A$	$\bar{B}A$	$BA$	$BA$
$\bar{C}$	X	X	1	0
C	X	X	1	1

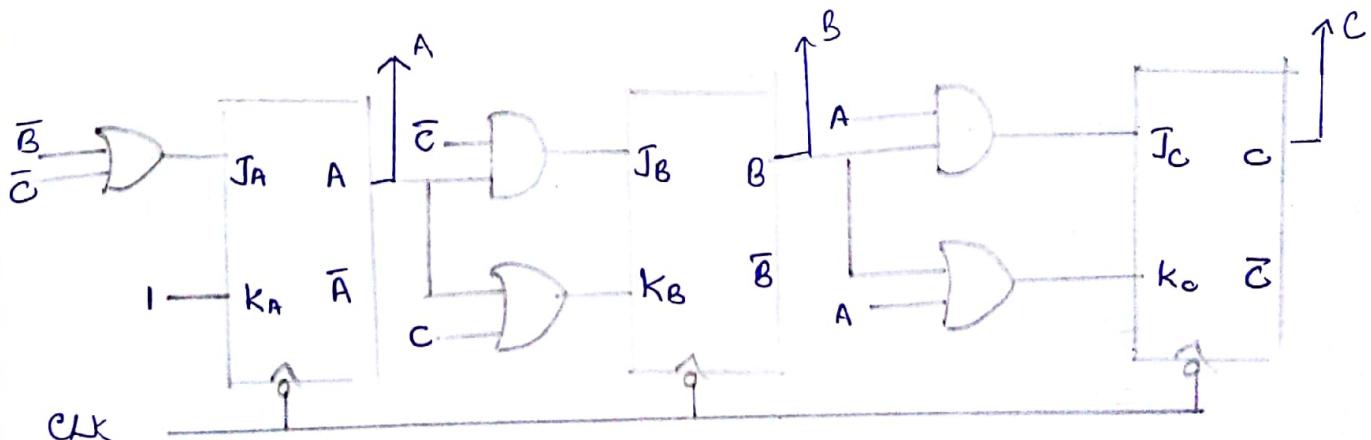
$$K_B = A + C$$

	$\bar{B}A$	$\bar{B}A$	$BA$	$BA$
$\bar{C}$	0	0	1	0
C	X	X	X	X

$$J_C = BA$$

	$\bar{B}A$	$\bar{B}A$	$BA$	$BA$
$\bar{C}$	X	X	X	X
C	0	1	1	1

$$K_C = A + B$$



Design mod 4 irregular counter using D flip-flop:

irregular counter sequence is.

$$00 \rightarrow 10 \rightarrow 11 \rightarrow 01 \rightarrow 00$$

$$2^n \geq N$$

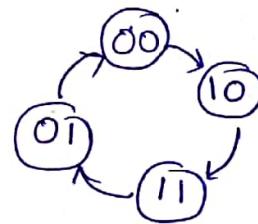
$$2^2 \geq 4$$

$$\underline{n=2}$$

Excitation table for DFF

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

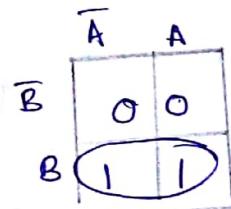
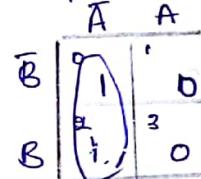
State Sequence diagram



State Table for Design of irregular counter:

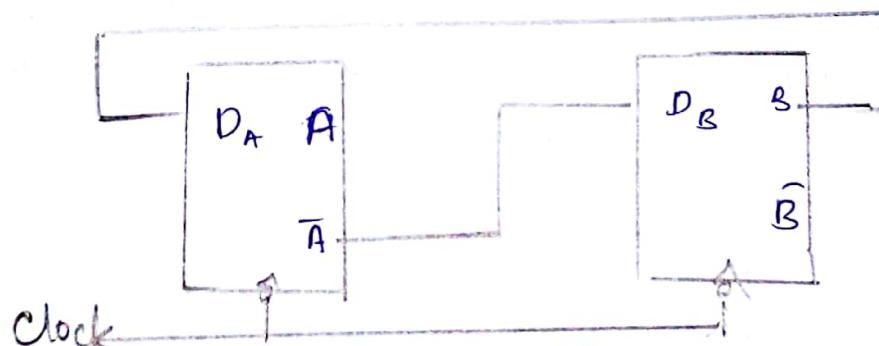
$B_n$	$A_n$	$B_{n+1}$	$A_{n+1}$	$D_B$	$D_A$
0	0	1	0	1	0
0	1	0	0	0	0
1	0	1	1	1	1
1	1	0	1	0	1

K map Simplification



$$D_B = \overline{A}$$

$$D_A = B$$

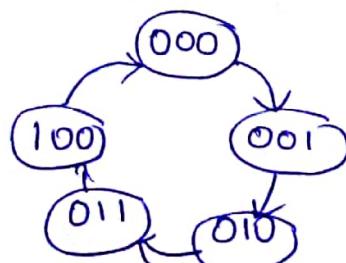


Design mod 5 synchronous regular counter using JK FFs.

$$2^n \geq N$$

$$2^3 \geq 5$$

state diagram



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

$C_n$	$B_n$	$A_n$	$C_{n+1}$	$B_{n+1}$	$A_{n+1}$	$J_C$	$K_C$	$J_B$	$K_B$	$J_A$	$K_A$
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
1	0	1	X	X	X	X	X	X	X	X	X
1	1	0	X	X	X	X	X	X	X	X	X
1	1	1	X	X	X	X	X	X	X	X	X

	$\bar{B}\bar{A}$	$\bar{B}A$	$BA$	$B\bar{A}$
$\bar{B}$	0	0	1	0
$e$	X	X	X	X

$$J_C = BA$$

	$\bar{B}\bar{A}$	$\bar{B}A$	$BA$	$B\bar{A}$
$\bar{B}$	0	1	X	X
$c$	0	X	X	X

$$J_B = A$$

	$\bar{B}\bar{A}$	$\bar{B}A$	$BA$	$B\bar{A}$
$\bar{B}$	1	X	1	X
$c$	0	X	X	X

$$J_A = \bar{C}$$

	$\bar{B}\bar{A}$	$\bar{B}A$	$BA$	$B\bar{A}$
$\bar{B}$	X	X	X	X
$c$	1	X	X	X

$$K_C = 1$$

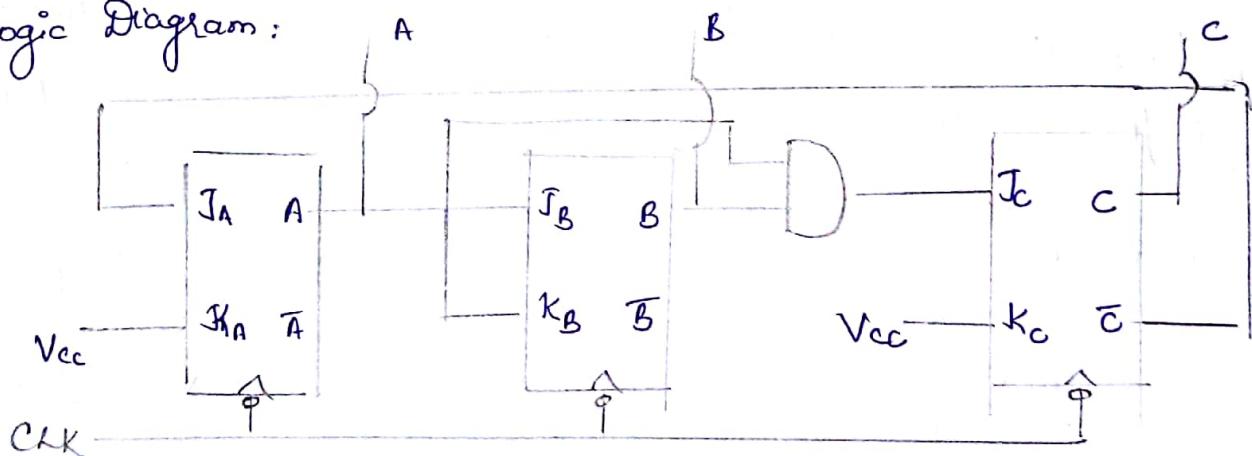
	$\bar{B}\bar{A}$	$\bar{B}A$	$BA$	$B\bar{A}$
$\bar{B}$	X	X	1	0
$c$	X	X	X	X

$$K_B = A$$

	$\bar{B}\bar{A}$	$\bar{B}A$	$BA$	$B\bar{A}$
$\bar{B}$	X	1	X	1
$c$	X	X	X	X

$$K_A = 1$$

Logic Diagram:



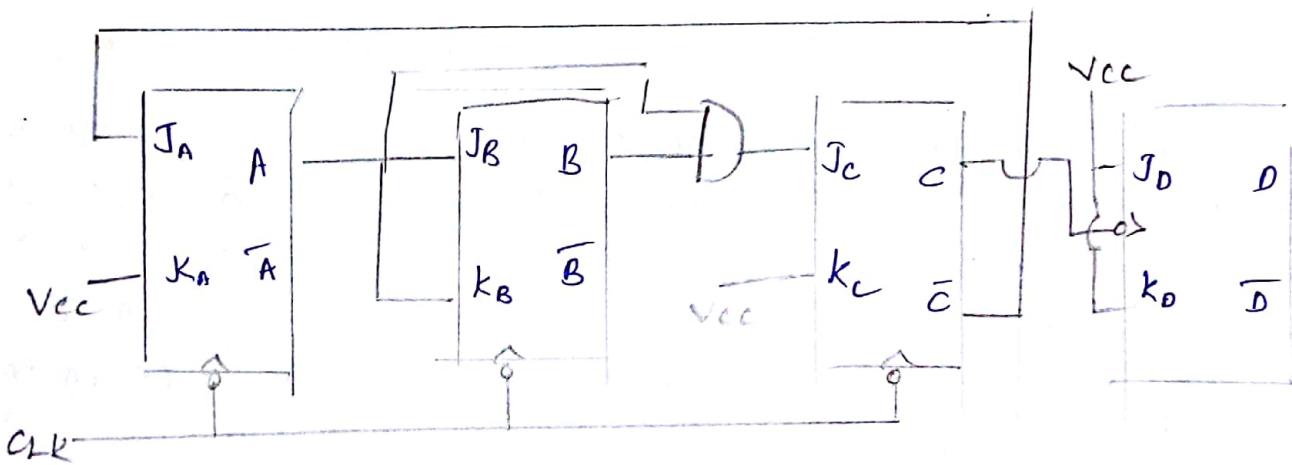
Excitation Table for all the flip-flops:

$Q_n$	$Q_{n+1}$	$J$	$K$	$S$	$R$	$D$	$T$
0	0	0	X	0	X	0	0
0	1	1	X	1	0	1	1
1	0	X	1	0	1	0	1
1	1	X	0	X	0	1	0

## Decade Counter:

Counter which counts 10 different states is called decade counter. It is also called mod 10 counter. Since it can represent digits from 0 to 9 it is called decade counter.

A decade counter can be designed by connecting mod 2 counter in conjunction with mod 5 counter.



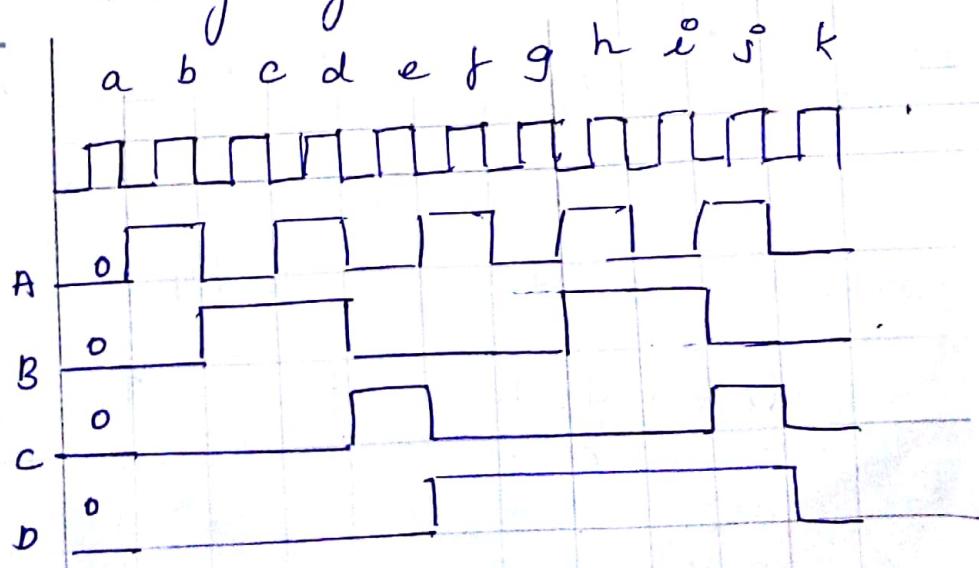
Synchronous mod 5 counter is cascaded asynchronously with mod 2 counter ie output C is the clock for D flip flop.

Since C is the clock for D and D is negative edge triggered D will toggle (since  $J_D = k_D = 1$ ) when C changes from 1 to 0.

Truth Table:

D	C	B	A
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
1	0	0	D
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
0	0	0	0

Timing Diagram:



A digital clock:

Design of a digital clock is an interesting application of counters and decoding.

To obtain pulses occurring at a rate of one each second, divide the 60 Hz power source by 60. If the resulting 1-Hz waveform is again divided by 60, a one per minute waveform is result. Dividing this signal by 60 then provides a one per hour waveform. This is the basic idea to be used in forming a digital clock.

The 1st divide by 60 counter simply divides the 60 Hz power signal down to a 1 Hz square wave. The second divide by 60 counter changes state once each sec and has 60 discrete states. It can therefore be decoded to provide signals to display seconds. This counter is then referred to as the seconds counter.

The 3rd divide by 60 counter changes state once each minute and has 60 discrete states. It can thus be decoded to provide the necessary signals to display minutes. This counter is thus the minute counter.

The last counter changes state once each 60 minutes once each hour. Thus if it is a divide by 12 counter, it will have 12 states that can be decoded to provide signals to display the current hour. This is called hour counter.

Note: modern counter is also called divide by N counter becoz MSB bit waveform has a period equal to N times that of the

The divide by 60 counter can be implemented by cascading counter ( $12 \times 5 = 60$  or  $10 \times 6 = 60$  etc.) 7490 decade counter can be used as a divide by 10 counter and the 7482 can be used as a divider by 6 counter. Similarly minute counter, second counter, can be implemented. The mod 10 counter can then be decoded to represent the unit digit of seconds and mod 6 counter can be decoded to represent the 10's digits of seconds using 7 segment indicator.

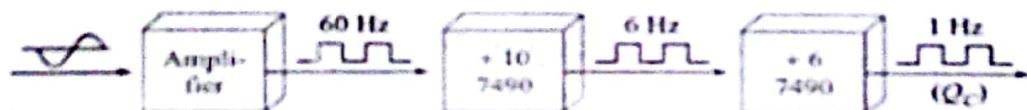
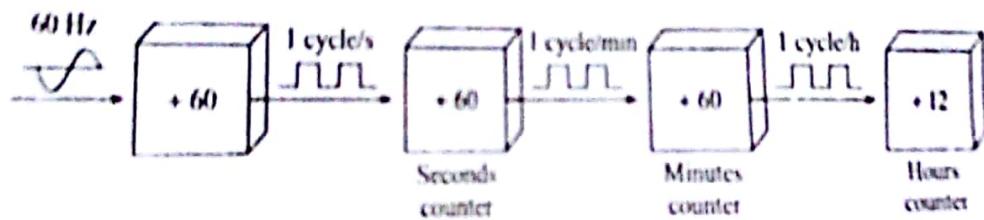
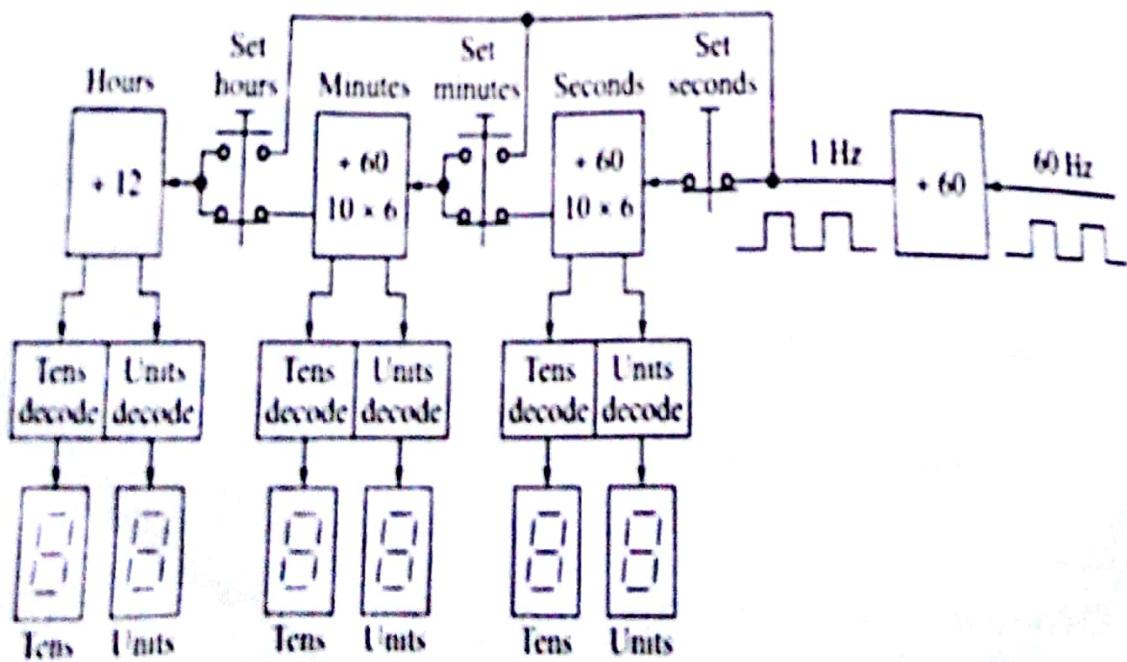


Figure 10.41: Divide-by-60 counter



## Counter Design Using HDL:

mod 8 counter:

```
module mod8uc (Clock, Reset, Q);
    input Clock, Reset;
    output [2:0] Q;
    reg [2:0] q;
    always @ (posedge Clock or negedge Reset)
        if (~Reset) Q = 3'b0;
        else Q = Q + 1;
    endmodule
```

mod 5 Counter:

```
module mod5uc (Clock, Reset, Q);
    input Clock, Reset;
    output [2:0] Q;
    reg [2:0] q;
    always @ (posedge Clock or negedge Reset)
        if (~Reset) Q = 3'b0;
        else
            begin
                Q = Q + 1;
                if (Q == 3'b101)
                    Q = 3'b0;
            end
    endmodule
```

mod 8 Counter using JK flip-flop:

Code for JK flip-flop:

module JKFF (Q, J, K, clock, Rset);

input J, K, Clock, Rset;

output Q;

begin Q;  
always @ (posedge Clock or negedge Rset)  
if (Rset) Q = 1'b0;

else Q <= (J #& Q) | (~K & Q);

endmodule

module mod8Counter (A, B, clock, Rset);

input clock, Rset;

output A, B;

wire JA, JB, KA, KB;

assign JA = ~B;

assign JB = A;

assign KA = 1'b1;

assign KB = 1'b1;

JKFF JK1 (A, JA, KA, clock, Rset);

JKFF JK2 (B, JB, KB, clock, Rset);

endmodule

mod 8 up & down Counter:

(when Mode=0 : counts up and when Mode=1 counts down  
(parallel load when RL=1)

module modupdown(CLK, RL, Mode, Q, Qbar);

input CLK, RL, Mode;

```

input [2:0] D;
output [2:0] Q;
reg [2:0] Q;
always @ (posedge CLK)
begin
    if (CP) Q = D;
    else if (CM) Q = Q - 1;
    else Q = Q + 1;
end
endmodule

```

Problems :

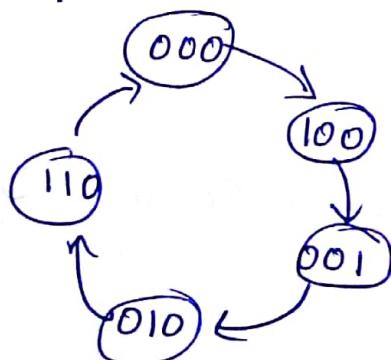
1. Design mod 6 counter using flip-flops.
2. Design mod 6 synchronous counter using JK FFs
3. Design mod 6 synchronous counter using RS FFs.
4. Design mod 5 self correcting counter using JK FFs.
5. Design mod 5 self correcting counter using RS flip-flops.

Design a synchronous counter to sequence  $0 \rightarrow 4 \rightarrow 1 \rightarrow 2$

$\rightarrow 6 \rightarrow 0$  using positive edge triggered SR flip flop.

3 flip-flops of type SR are required.

State diagram:



Excitation table

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

$C_n$	$B_n$	$A_n$	$C_{n+1}$	$B_{n+1}$	$A_{n+1}$	$S_c$	$R_c$	$S_B$	$R_B$	$S_A$	$R_A$
0	0	0	1	0	0	1	0	0	X	0	X
0	0	1	0	1	0	0	X	1	0	0	1
0	1	0	1	1	0	1	0	X	0	0	X
0	1	1	X	X	X	X	X	X	X	X	X
1	0	0	0	0	1	0	1	0	X	1	0
1	0	1	X	X	X	X	X	X	X	X	X
1	1	0	0	0	0	0	1	0	1	0	X
1	1	1	X	X	X	X	X	X	X	X	X

	$\bar{B}\bar{A}$	$\bar{B}A$	$BA$	$B\bar{A}$
$\bar{C}$	1	0	X	1
C	0	X	X	0

$$S_c = \bar{A}\bar{C}$$

	$\bar{B}\bar{A}$	$\bar{B}A$	$BA$	$B\bar{A}$
$\bar{C}$	X	0	X	0
C	X	X	X	1

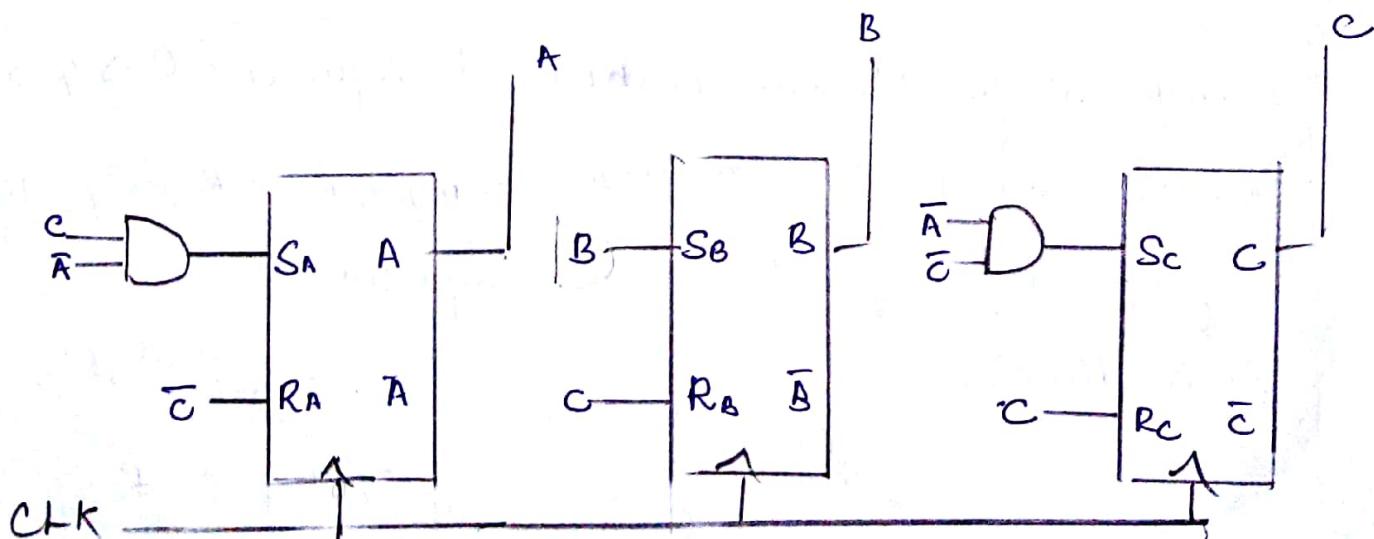
$$R_B = C$$

	$\bar{B}\bar{A}$	$\bar{B}A$	$BA$	$B\bar{A}$
$\bar{C}$	0	0	X	0
C	(1 X) X	X	0	

$$S_A = CA$$

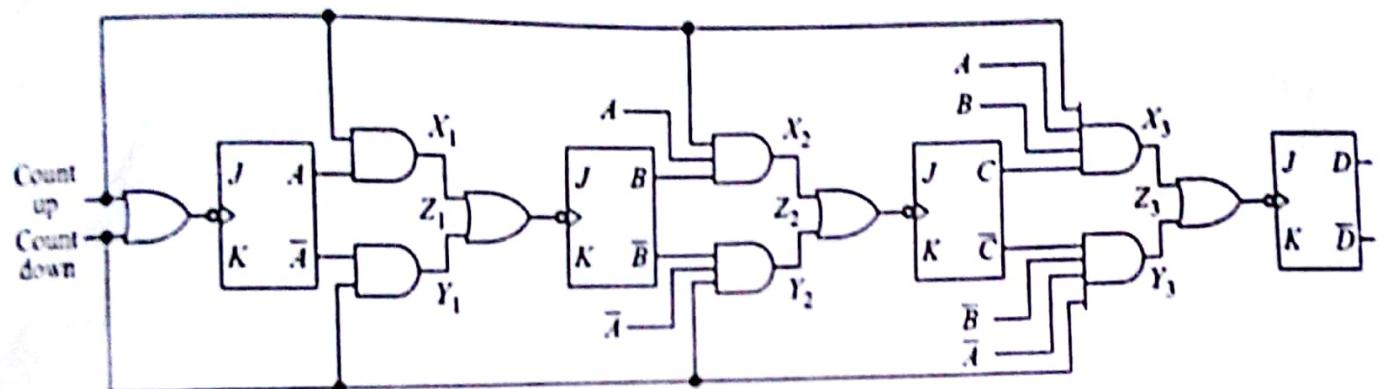
	$\bar{B}\bar{A}$	$\bar{B}A$	$BA$	$B\bar{A}$
$\bar{C}$	X	1	X	(X)
C	0	X	X	X

$$R_A = \bar{C}$$



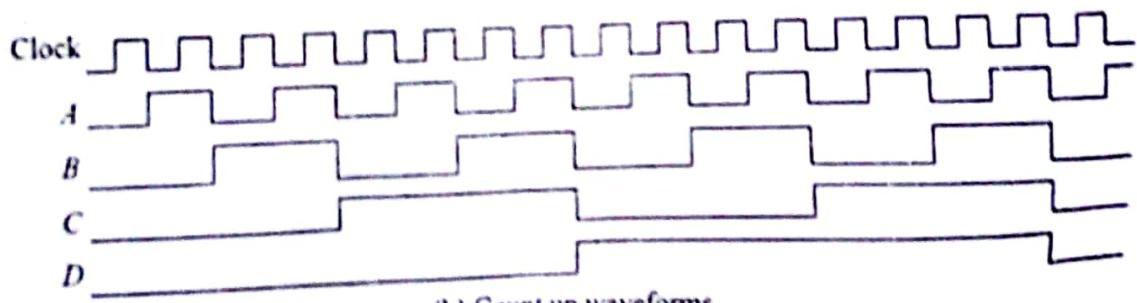
Steps to Design Irregular Counter & Transition Table  
 With regular counter 0 to given modulus in  
 present/current state, while next state arising state diagram.

# Synchronous Up and Down Counter Diagram

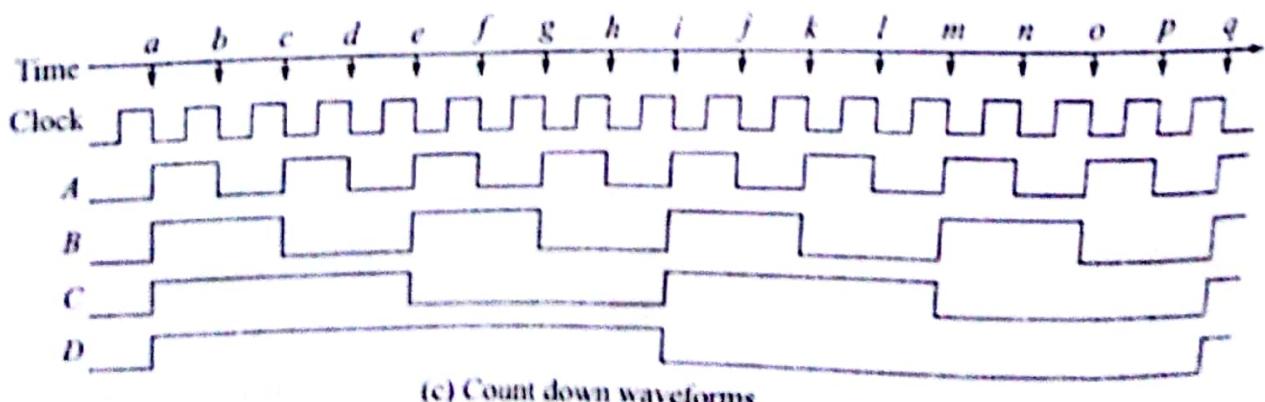


Note : All  $J$  and  $K$  inputs are tied to  $+V_{CC}$

(a) Logic diagram



(b) Count up waveforms



(c) Count down waveforms