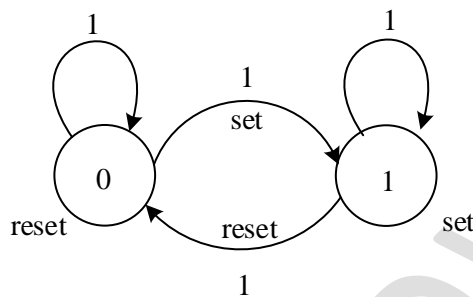


Counters -Sequential Circuits (Part II)

Before starting the applications of FFs we introduce some important feature that will help us to understand better the application part.

State diagram: The design of clocked sequential circuit starts from set of specs that end up in logic diagram. State diagram is a directed graph consisting of vertices (or nodes) and directed arcs between nodes. Nodes are represented by circle with name of state inside. Directed arc represents the state transition. Labels on directed arc specify inputs and outputs separated by slash (Input/output).



State diagram for R-S FF

State diagram of R-S FF is shown in the diagram

Two stable states are 0 and 1.

Apply set input, FF is set to 1

Apply reset input, FF is reset i.e. 0 (irrespective of its previous state)

Flip flop Characteristic and Excitation Tables: Truth tables are very common with combinational circuits. In sequential circuits also we can use truth tables but they do not give the complete information. So they are modified. Truth tables give output for given input combinations. Characteristic table has one more entry at the input i.e. present state and output is next state. To reduce the number of columns in a table this table is modified as inputs and next state is output. Such simplified tables are presented here.

These characteristic table can be tabulated somewhat differently. These tables are shown below:

FF Characteristic Tables

S	R	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	?

(a) RS FF

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

(b) JK FF

D	Q_{n+1}
0	0
1	1

(c) D FF

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

(d) T FF

Characteristic equation is good for defining the operation of FF. But during design we know transition from present state to next to state and wish to find FF input conditions that will cause the required transition. So we need a table that given change of state. Q_n and Q_{n+1} represent present state and next state. Such table is excitation table. Following tables give the excitation tables for the four FFs.

FF Excitation Tables

Q_n	Q_{n+1}	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

(a) RS FF

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

(b) JK FF

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

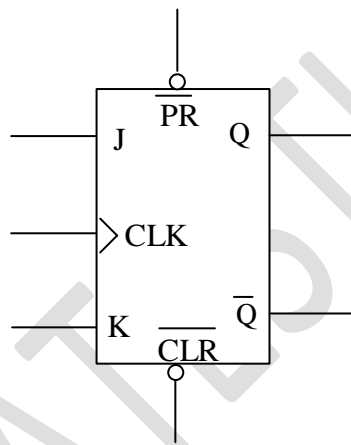
(c) D FF

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

(d) T FF

Applications of FFs: Flip flops find applications in counters, frequency dividers shift and storage registers etc. FFs can also generate one shots. Another important application is transfer of information from one FF or a group of FFs to another group of FFs.

Asynchronous Inputs in FFs: The inputs like S, R, J, K, D and T are called controlled inputs or synchronous inputs.



Other than above there are asynchronous inputs which effect the FF. These are override inputs since they can be used to override other inputs

\overline{PR} & \overline{CLR} are active low preset and clear inputs (also said direct set or direct reset inputs)

Take an example to get the feel of it.

Prob. What necessary conditions are for +ve edge triggered JK FF with active low asynchronous inputs to toggle to its opposite state?

Slon. For FF to toggle $J = K = 1$ there is no role of asynchronous inputs so they should not be active. Since they are active low, so they be kept high i.e. 1

$$So, \overline{PRE} = \overline{CLR} = 1$$

At +ve edge of CLK FF will toggle.

Counters: Counting is fundamental function of digital circuits. A counter consists of collection of FFs that change state (set or reset) in a prescribed sequence.

Some important points:

Number of FF states N

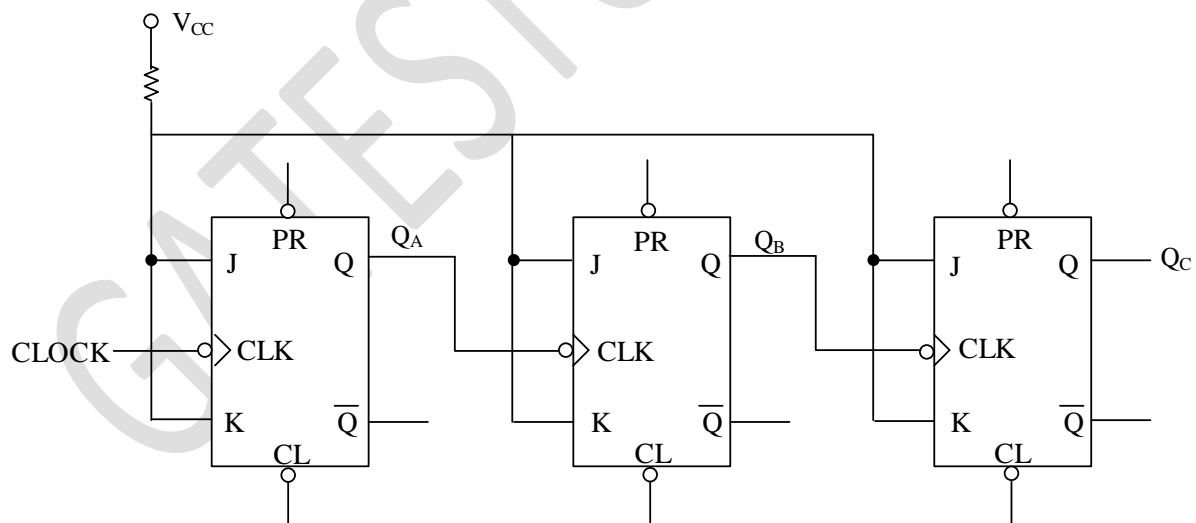
Number of FFs n

$$N \leq 2^n$$

Binary counter will have maximum of 2^n states

If, $N < 2^n$, it is non binary counter

Ripple Counters (Asynchronous): Most straight forward counter is ripple divider. JK FF is used where J and K input are connected together and to high input. The circuit is shown. Any time +ve edge of clock comes the Q output will change. This can be used as ripple divider circuit. You can have divide by 2, 4, 8 counters. They are said asynchronous since no common synchronizing clock is applied to all FFs. **Note that clock is used here as data input.**



Divide by 8 counter

A divide by 8 ripple counter requires three FFs each connected as toggle FF

The counters are said to be rippled since input ripple from one FF to other.

Points to note for ripple counters:

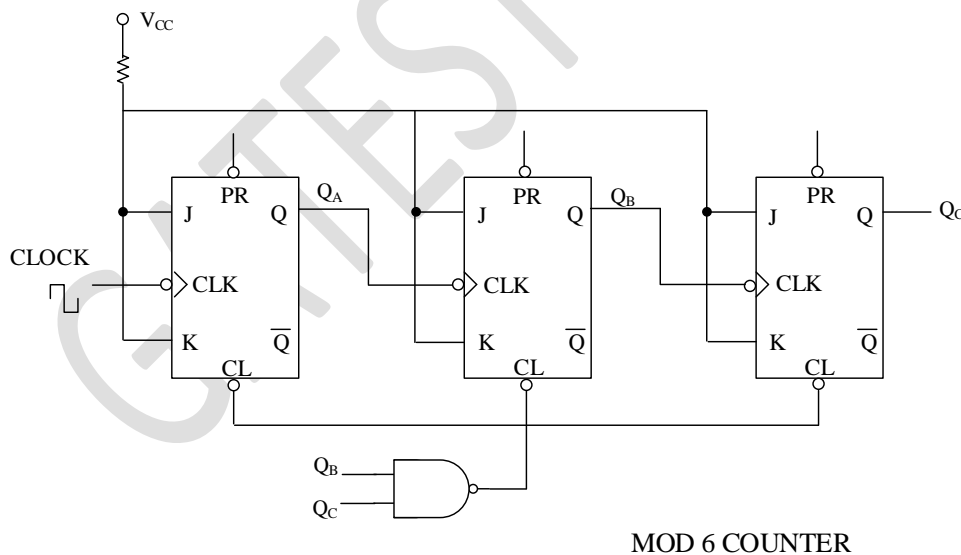
- Asynchronous counters
- FFs operate in toggle mode
- Output of one FF will act as clock to next FF
- Input clock (or data input) is applied to one FF and will act as LSB

Modulus of a Counter

- Mod is number of states in counting sequence.
- If n FFs then 2^n states thus 'Mod 2^n counter'
- Mod gives the frequency division available from the last FF (f/N)
- It counts from 0 to $(2^n - 1)$ states for 4 bit counter 0 to 15
- If MOD M and MOD N counters are cascaded then it acts as Mod $M.N$ counter.
- In asynchronous counters states of FF do not change at the same time

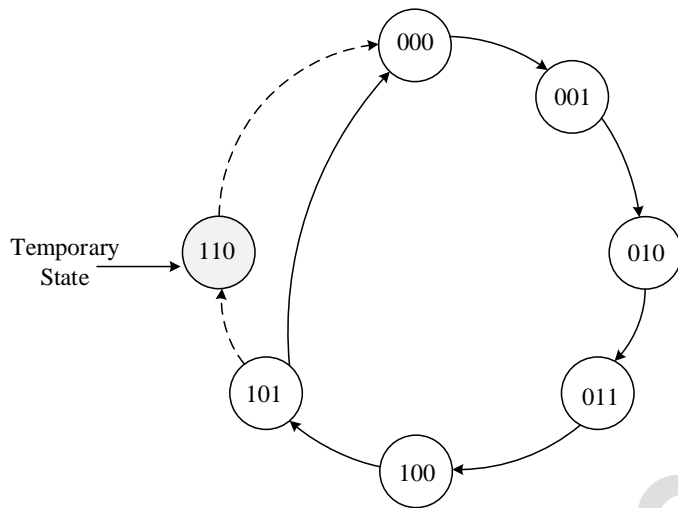
Counters With $MOD < 2^n$

Basic asynchronous counter is limited to MOD numbers that are equal to 2^n where n is number of FFs. This basic counter can be modified to produce MOD numbers less than 2^n by allowing counter to skip the states. One of the common method to do so is shown in a 3 bit counter. It can be seen that it is mod 8 counter, counting form 000 to 111. The presence of NAND gate will alter this sequence.



- (i) NAND output connected to asynchronous clear inputs of each FF. As long as NAND output is high it will have no effect on counter. When low it will clear all FFs.
- (ii) Input to NAND are outputs of B and C FFs so, NAND output low when $Q_B = Q_C = 1$ i.e. it occurs when FF goes to 101 and then to 110 where FF resets.

000 → 001 → 010 → 011 → 100 → 101 → 110 (temporary state to clear the FF) so six states only.

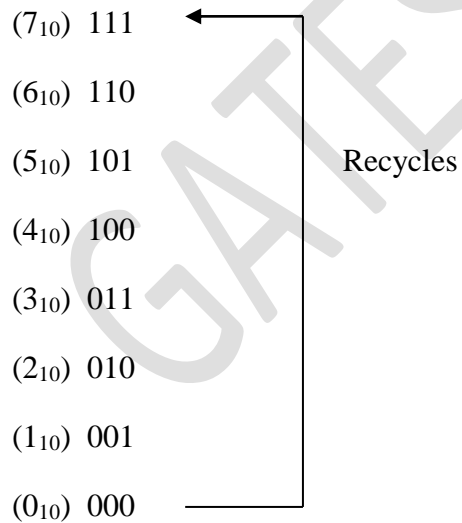


State Transition Diagram

Decade Counters/BCD Counters

MOD 10 is also called decade counter. It counts sequence from 0000 (zero) through 1001 (9_{10}) is also called BCD counter. Note that a decade counter clocked from 50 KHz will have output frequency as $50 \text{ KHz}/10 = 5 \text{ KHz}$

Down Counters: - All the counters which we have seen have been up counters. For a MOD - 8 down counter the count sequence will be



Down counters are not as widely used as up counters. Their major applications is in situations where it must be known when a desired number of input pulses have occurred. So it is preset and allowed to count down.

In down counters the inverted output of each FF is connected to CLK input of the following FF

Propagation delay in Ripple counters: -Ripple counters are the simplest type of counters because they require fewest components. The basic principle of operation of these FFs is that each FF is triggered by the transition at the output of preceding FF (propagation delay). So for an n FF counter the delay will be $n \cdot t_{pd}$.

So $T_{clock} \geq n \cdot t_{pd}$

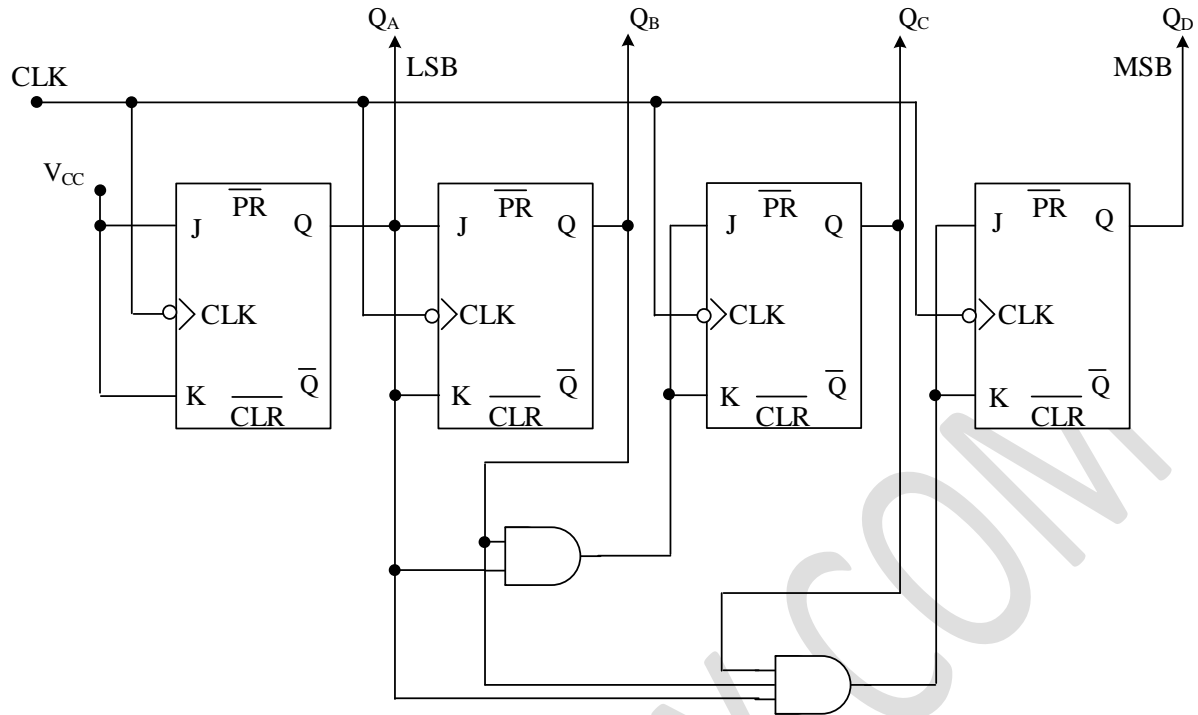
$$f_{max} = \frac{1}{n \times t_{pd}}$$

For $t_{pd} = 24 \text{ ns}$

$$f_{max} = \frac{1}{4 \times 24ns} = 10.4MHz$$

So as the number of FF increases the total propagation delay increases. So f_{max} decreases.

Synchronous (Parallel) Counter: Asynchronous counters discussed so far are the simplest type of binary counter as they require less hardware. But its speed of operation is low because of propagation delay time of FFs. Another problem is of glitches at the decoding gate outputs. Above problems are eliminated by applying the clock pulse to all FFs simultaneously, which is done in synchronous counters or parallel counters (parallel since FF triggered simultaneously) Four bit MOD 16 synchronous counter is shown in the



- Clock is applied simultaneously to all FFs.
- Only LSB FF has $J = K = 1$. J & K inputs of other FFs are driven by some combination of FF outputs.
- Synchronous counter requires more circuitry.

4 bit (MOD 16) synchronous counter with parallel carry is shown

LSB FF $J = K = 1$

JK inputs of FF B connected to Q_A changes state when $Q_A = 1$ & -ve clock transition.

JK inputs of FF C are connected to AND output of Q_A & Q_B , changes state when $Q_A = Q_B = 1$ & -ve clock.

JK inputs of FF D connected to AND output of Q_A , Q_B , & Q_C

It changes state when $Q_A = Q_B = Q_C = 1$ and -ve clock transition.

In parallel counter all FFs change state simultaneously with -ve clock transition. So unlike asynchronous counters in which delay is cumulative, here it is not.

Total propagation delay is

= propagation delay of one FF + propagation delay of AND gate

$$f_{max} = \frac{1}{t_p + t_p}$$

Where

t_p – propagation delay of one FF

t_g – propagation delay of AND gate

Also glitches are avoided due to common clocking

Thus synchronous counters can operate at a much higher input frequency. But the circuitry of synchronous counter is more complex than that of asynchronous.

So the basic principle for constructing a synchronous counter can be stated as.

'Each FF should have its J and K inputs connected so that they are high only when the outputs of all lower order FFs are in high state'

Synchronous Up/Down Counters: - We have seen that ripple counter could be made to count down by using inverted output of each FF to drive next FF. A parallel down counter can be constructed in a similarly way for parallel up/down counter, the nomenclature used for control signal (up/down) was chosen to make it clear how it effects the counter. The count up operation is active high the count- down operation is active low.

Pre-settable Counters: Many synchronous counters available in IC form can be pre-settable i.e. they can be preset to any desired starting count either asynchronously (independent of clock signal) or synchronously (on active transition of clock signal). This presetting is also called parallel loading the counter.

Cascading BCD Counters: BCD counters are often used whenever pulses are to be counted and result displayed in decimal. A BCD counter counts from 0 through 9 and then recycles to 0. To count larger decimal values, BCD counters can be cascaded.