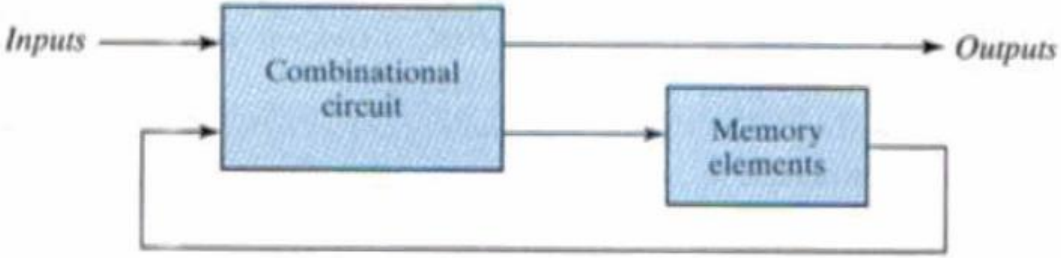
Basic Electronics Engineering (Spring 2024)

Resources of PPT:

- ❑ www.google.com

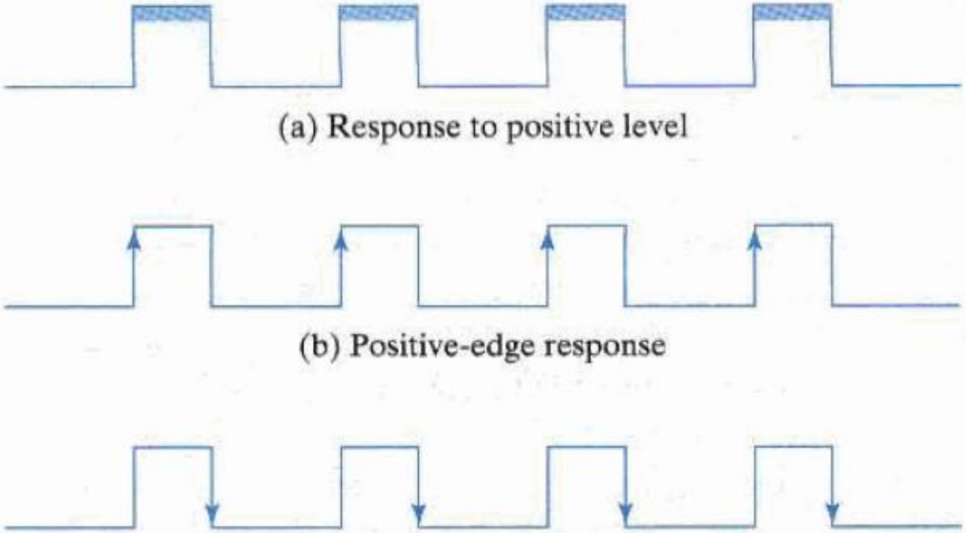
- ❑ Digital Design, 4th Edition
  M. Morris Mano and Michael D. Ciletti

Sequential Circuit



Block diagram of sequential circuit

A sequential circuit is one whose output depends upon both present and past inputs.

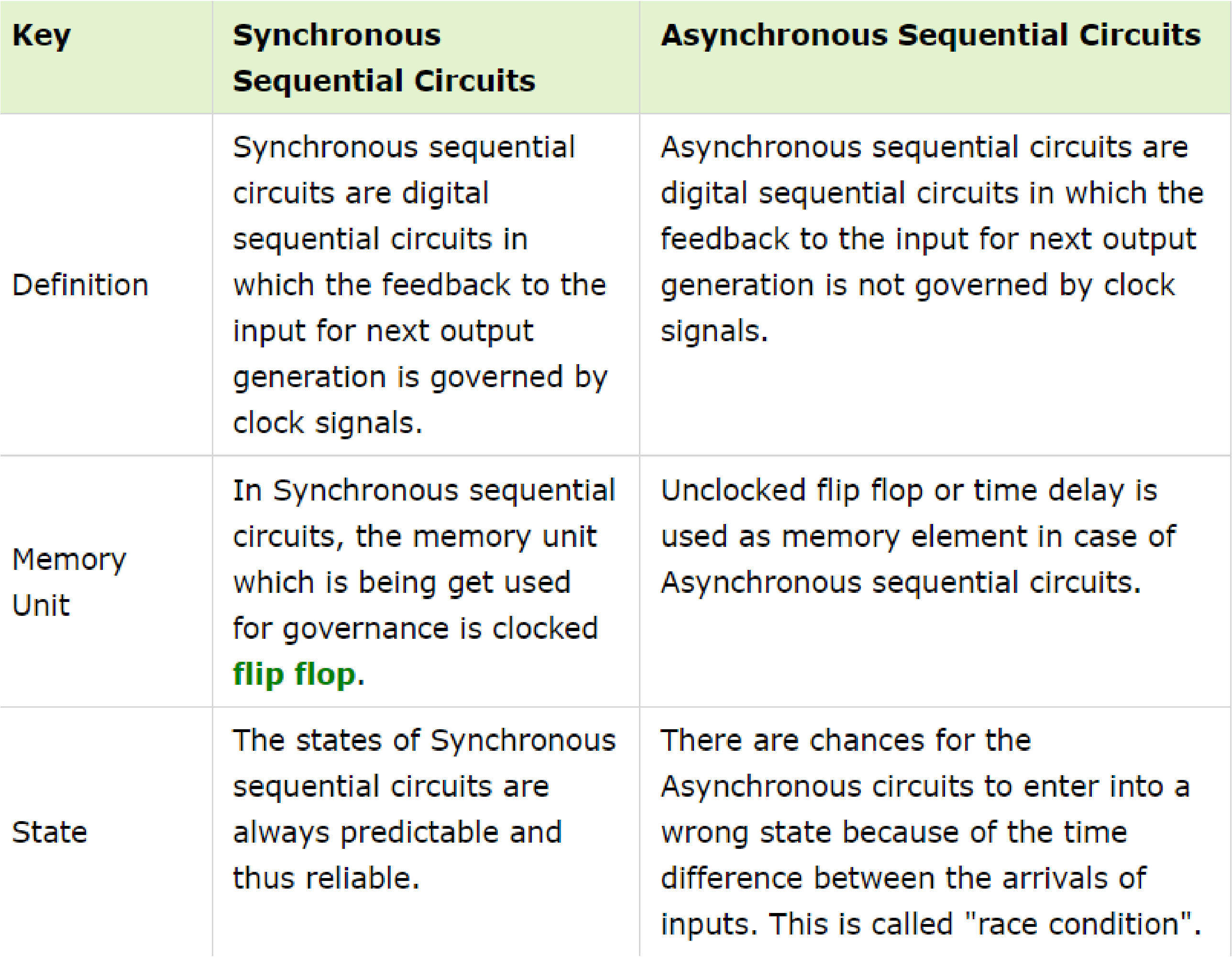


(a) Response to positive level

(b) Positive-edge response

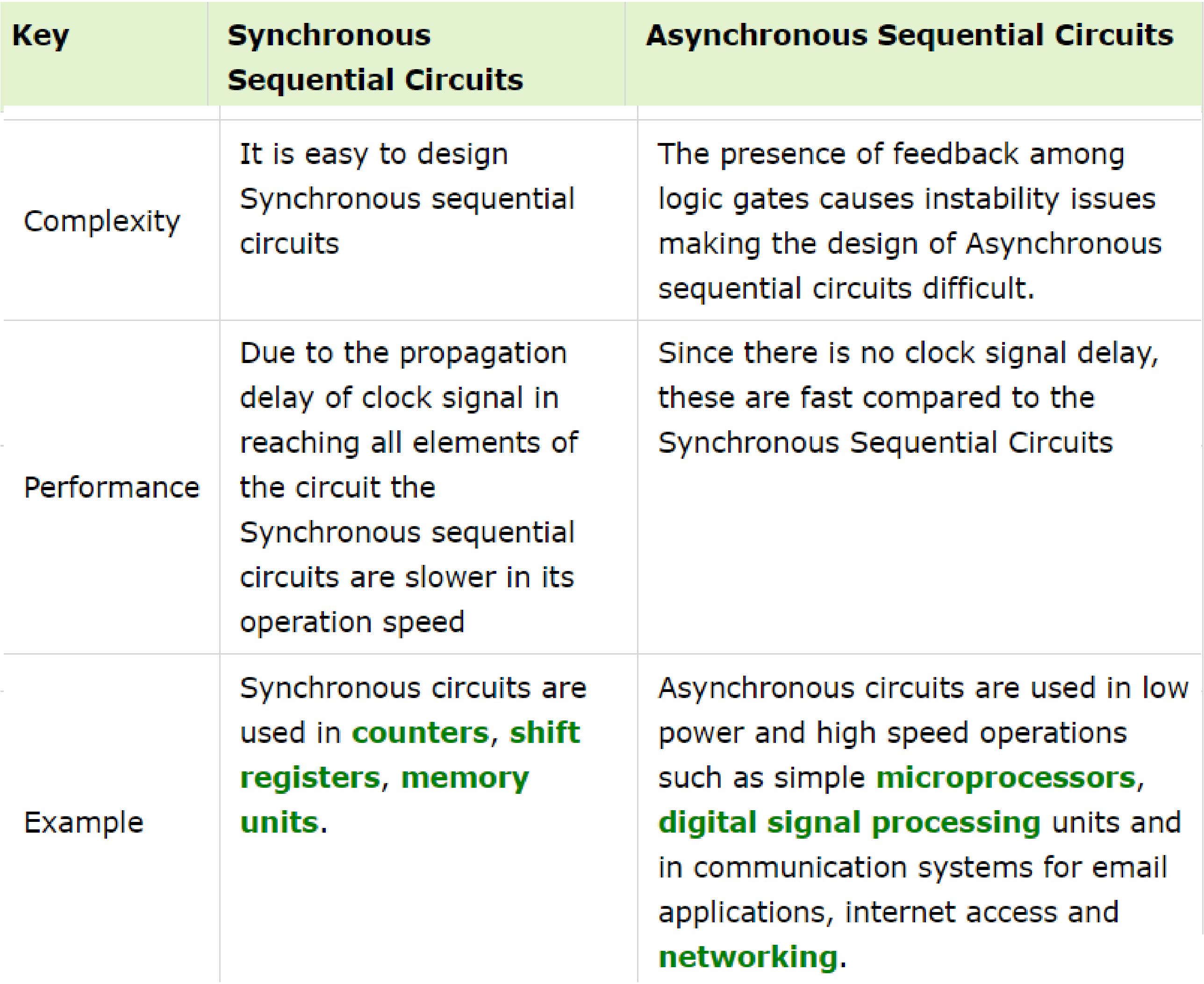
(c) Negative-edge response

Sequential Circuit



Key	Synchronous Sequential Circuits	Asynchronous Sequential Circuits
Definition	Synchronous sequential circuits are digital sequential circuits in which the feedback to the input for next output generation is governed by clock signals.	Asynchronous sequential circuits are digital sequential circuits in which the feedback to the input for next output generation is not governed by clock signals.
Memory Unit	In Synchronous sequential circuits, the memory unit which is being get used for governance is clocked **flip flop.**	Unclocked flip flop or time delay is used as memory element in case of Asynchronous sequential circuits.
State	The states of Synchronous sequential circuits are always predictable and thus reliable.	There are chances for the Asynchronous circuits to enter into a wrong state because of the time difference between the arrivals of inputs. This is called "race condition".

Sequential Circuit



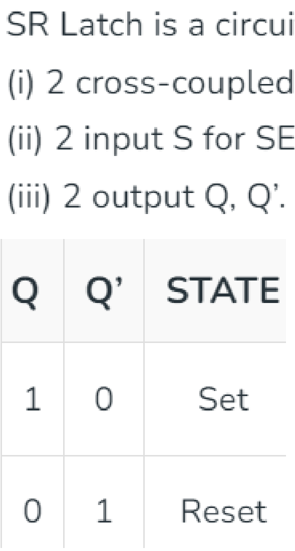
Key	Synchronous Sequential Circuits	Asynchronous Sequential Circuits
Complexity	It is easy to design Synchronous sequential circuits	The presence of feedback among logic gates causes instability issues making the design of Asynchronous sequential circuits difficult.
Performance	Due to the propagation delay of clock signal in reaching all elements of the circuit the Synchronous sequential circuits are slower in its operation speed	Since there is no clock signal delay, these are fast compared to the Synchronous Sequential Circuits
Example	Synchronous circuits are used in **counters**, **shift registers**, **memory units**.	Asynchronous circuits are used in low power and high speed operations such as simple **microprocessors**, **digital signal processing** units and in communication systems for email applications, internet access and **networking**.

Sequential Circuit: Latches

Latches are digital circuits that store a single bit of information and hold its value until it is updated by new input signals. The latches have **low** and **high** two stable states. Due to these states, latches also refer to as **bistable-multivibrators**.

S-R Type Latch:

S-R (Set-Reset) Latches: S-R latches are the simplest form of latches and are implemented using two inputs: S (Set) and R (Reset). The S input sets the output to 1, while the R input resets the output to 0. When both S and R are at 1, the latch is said to be in an “undefined” state.



SR Latch is a circuit with:

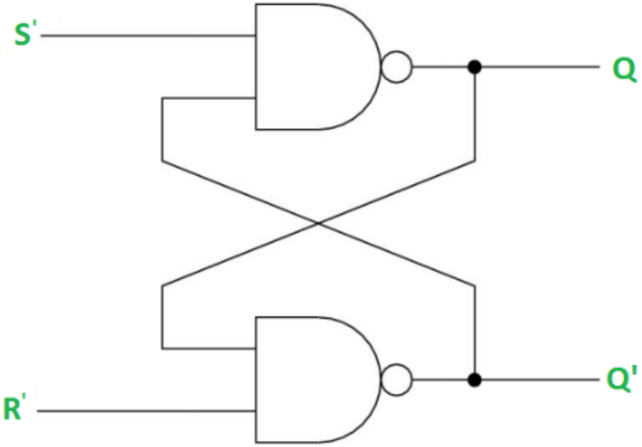
(i) 2 cross-coupled NOR gate or 2 cross-coupled NAND gate.

(ii) 2 input S for SET and R for RESET.

(iii) 2 output Q, Q’.

Q	Q’	STATE
1	0	Set
0	1	Reset

Under normal conditions, both the input remains 0.

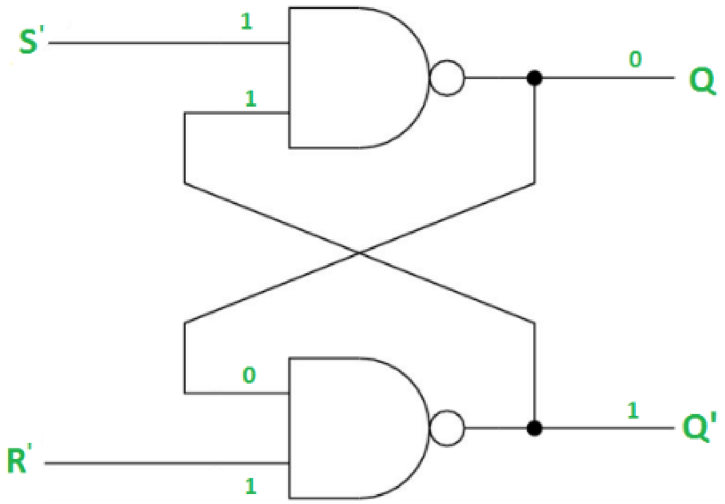


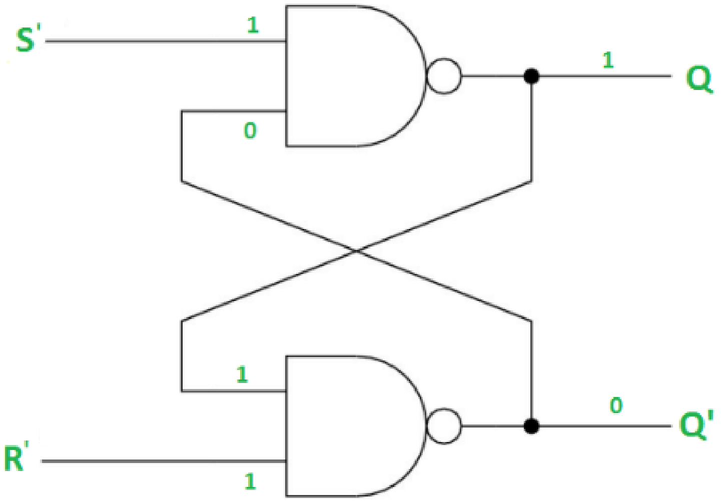
Sequential Circuit: Latches

Case-1: S’=R’=1 (S=R=0) –

If Q = 1, Q and R’ inputs for 2nd NAND gate are both 1.

If Q = 0, Q and R’ inputs for 2nd NAND gate are 0 and 1 respectively.

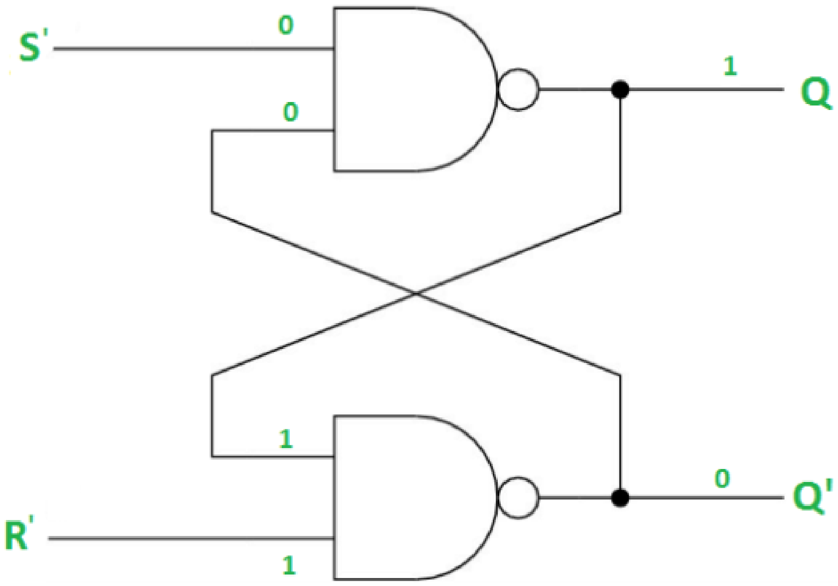




Sequential Circuit: Latches

Case-2: S'=0, R'=1 (S=1, R=0) –

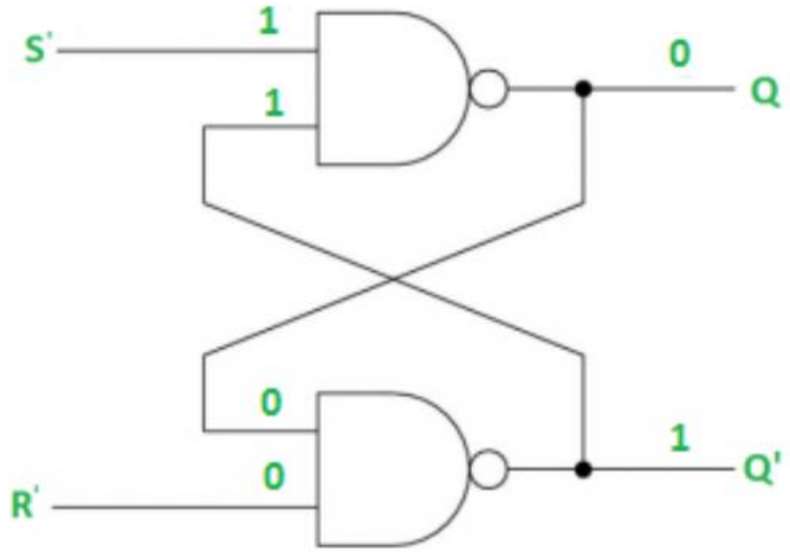
As S'=0, the output of 1st NAND gate, Q = 1(**SET state**). In 2nd NAND gate, as Q and R' inputs are 1, Q'=0.



Sequential Circuit: Latches

Case-3: S'= 1, R'= 0 (S=0, R=1) –

As R'=0, the output of 2nd NAND gate, Q' = 1. In 1st NAND gate, as Q and S' inputs are 1, Q=0(**RESET state**).

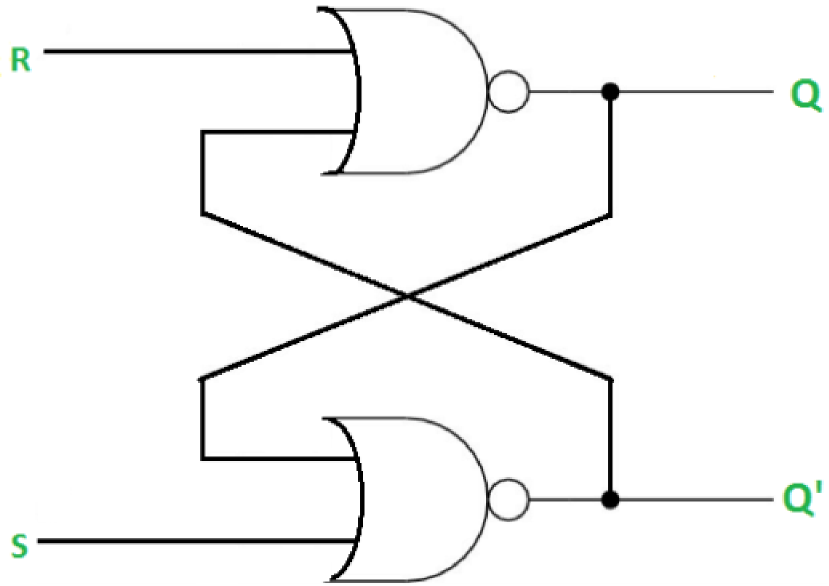


Sequential Circuit: Latches

Case-4: S’= R’= 0 (S=R=1) –

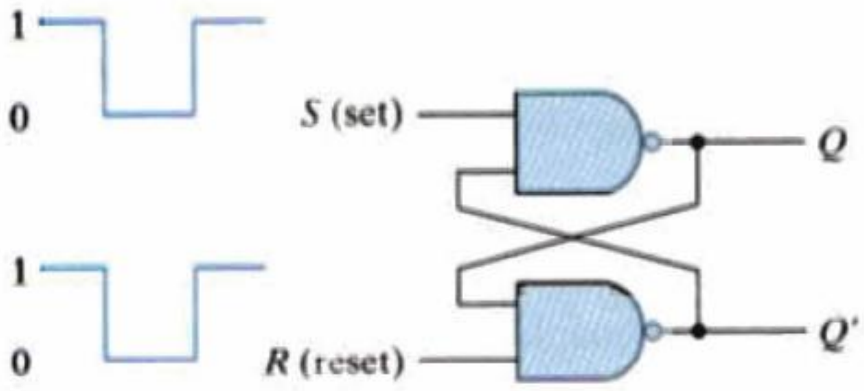
When S=R=1, both Q and Q’ becomes 1 which is not allowed. So, the input condition is prohibited.

The SR Latch using NOR gate is shown below:



Forbidden State!

Sequential Circuit: Latches



(a) Logic diagram

S	R	Q	Q'	
1	0	0	1	
1	1	0	1	(after $S = 1, R = 0$)
0	1	1	0	
1	1	1	0	(after $S = 0, R = 1$)
0	0	1	1	(forbidden)

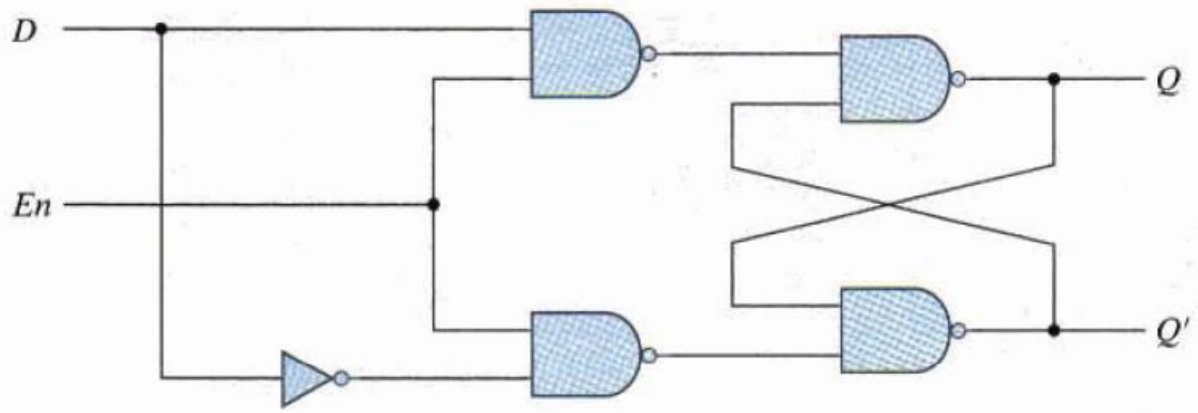
(b) Function table

SR latch with NAND gates

Sequential Circuit: Latches

D Latch:

The **D latch** is the same as **D flip flop**. The only difference between these two is **the ENABLE** input. The output of the latch is the same as the input passed to the **Data** input when the **ENABLE** input set to 1. At that time, the latch is open, and the path is transparent from input to output. If the **ENABLE** input is set to 0, the D latch's output is the last value of the latch, i.e., independent from the input D, and the latch is closed. Below are the circuit diagram and the truth table of the D latch.



(a) Logic diagram

En	*D*	Next state of *Q*
0	X	No change
1	0	$Q = 0$; reset state
1	1	$Q = 1$; set state

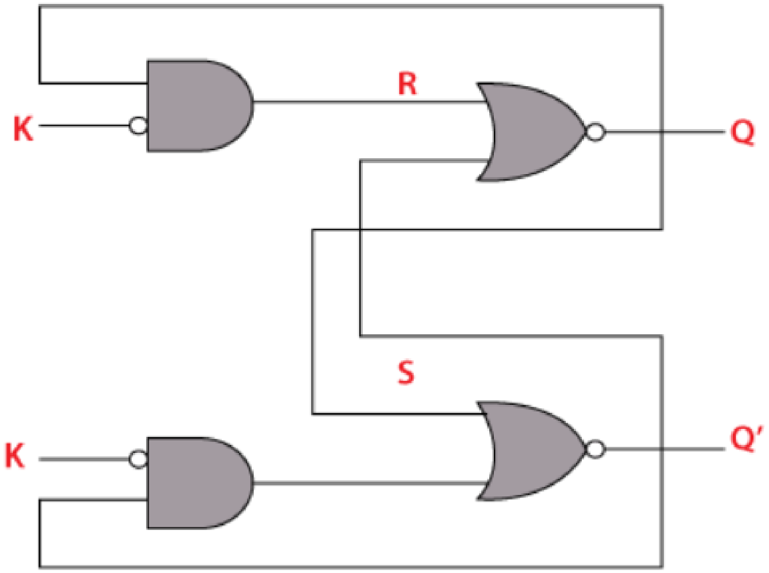
(b) Function table

Sequential Circuit: Latches

JK Latch

The JK Latch is the same as the SR Latch. In JK latch, the unclear states are removed, and the output is toggled when the JK inputs are high. The only difference between SR latch JK latches is that there is no output feedback towards the inputs in the SR latch, but it is present in the JK latch.

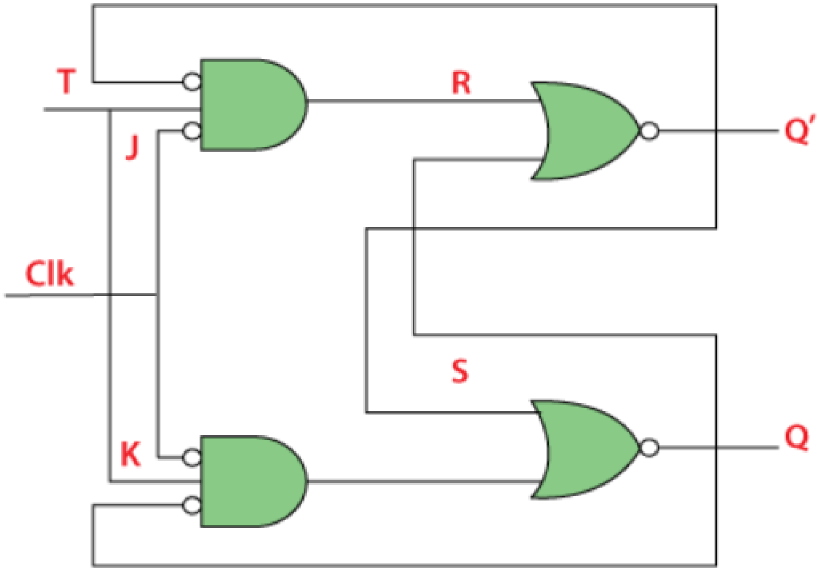
J	K	Q_{next}	Comment
0	0	Q	No change
0	1	0	Reset
1	0	1	Set
1	1	Q'	Toggle



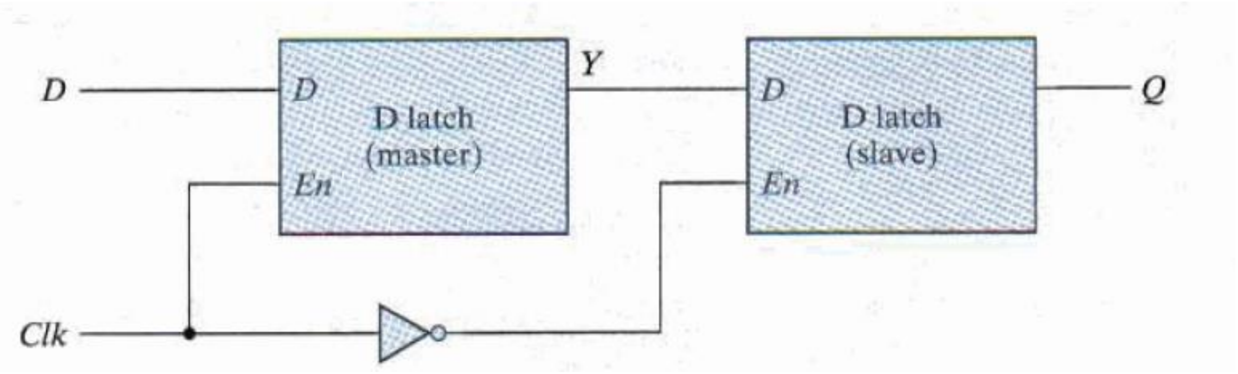
Sequential Circuit: Latches

T Latch

The T latch forms by shorting the JK latch inputs. The output of the T latch toggle when the input set to 1 or high.

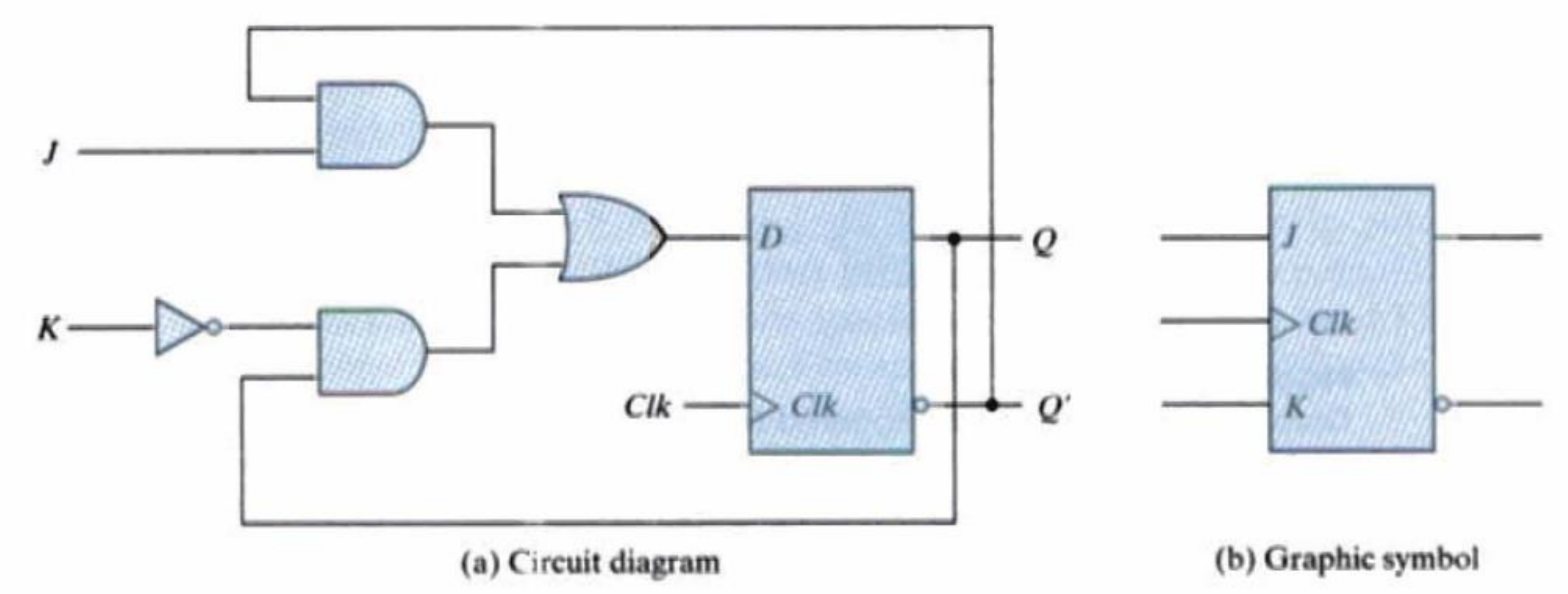


Sequential Circuit: Flipflop



Master–slave *D* flip-flop

Sequential Circuit: Flipflop

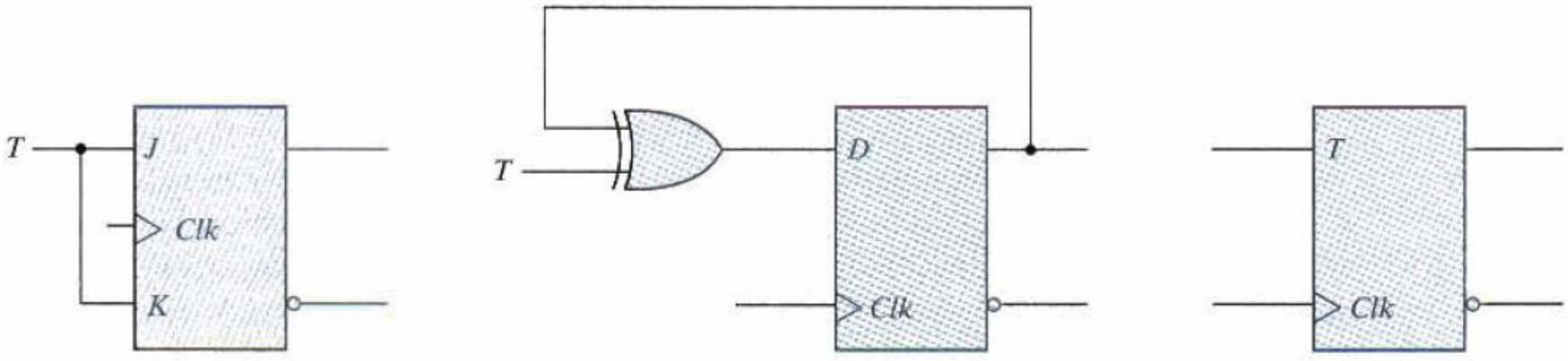


(a) Circuit diagram

(b) Graphic symbol

$$D = JQ' + K'Q$$

Sequential Circuit: Flipflop



(a) From *JK* flip-flop

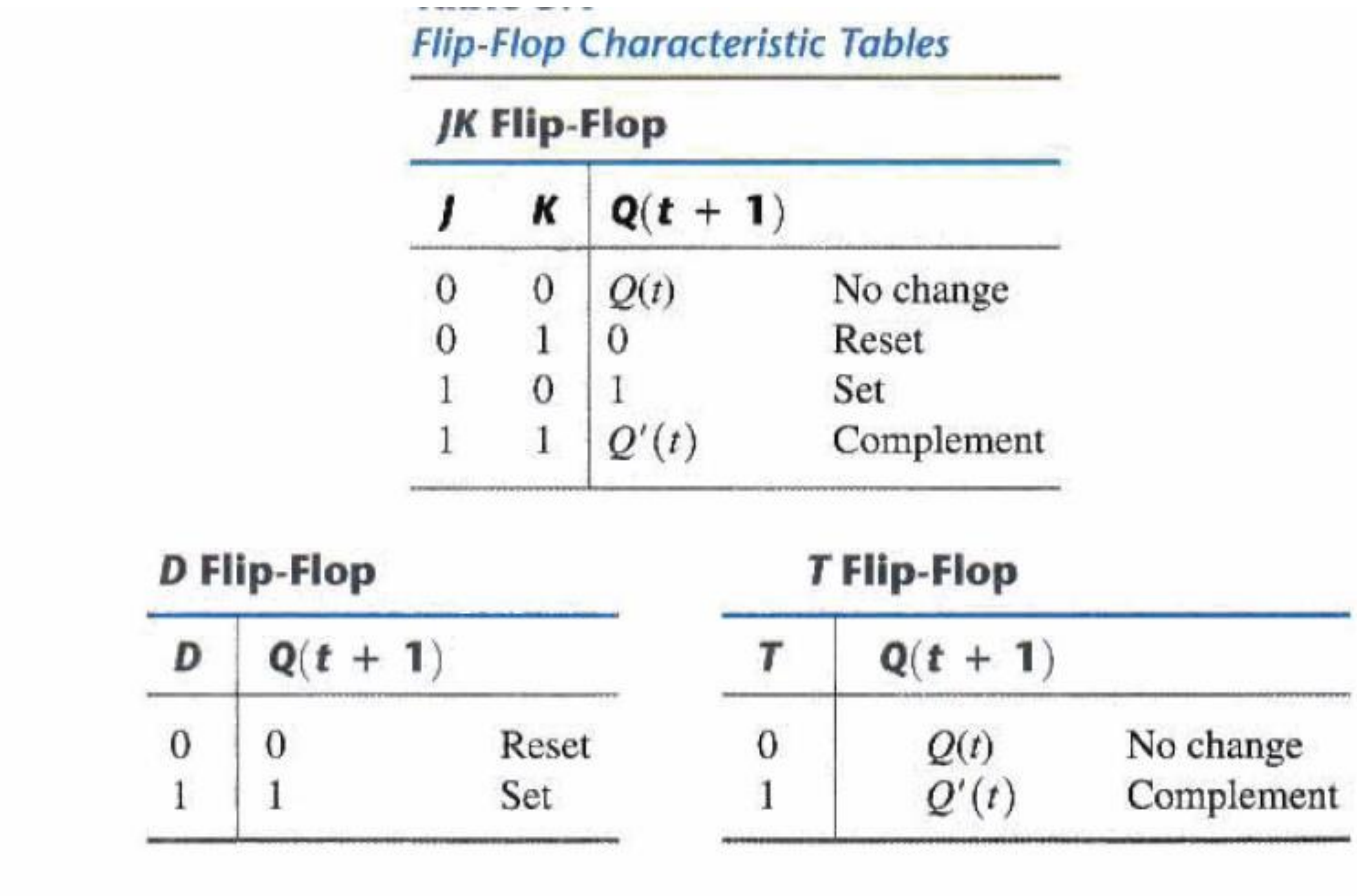
(b) From *D* flip-flop

(c) Graphic symbol

T flip-flop

$$D = T \oplus Q = TQ' + T'Q$$

Sequential Circuit: Flipflop



Flip-Flop Characteristic Tables

JK Flip-Flop

J	***K***	**Q(*t* + 1)**	
0	0	$Q(t)$	No change
0	1	0	Reset
1	0	1	Set
1	1	$Q'(t)$	Complement

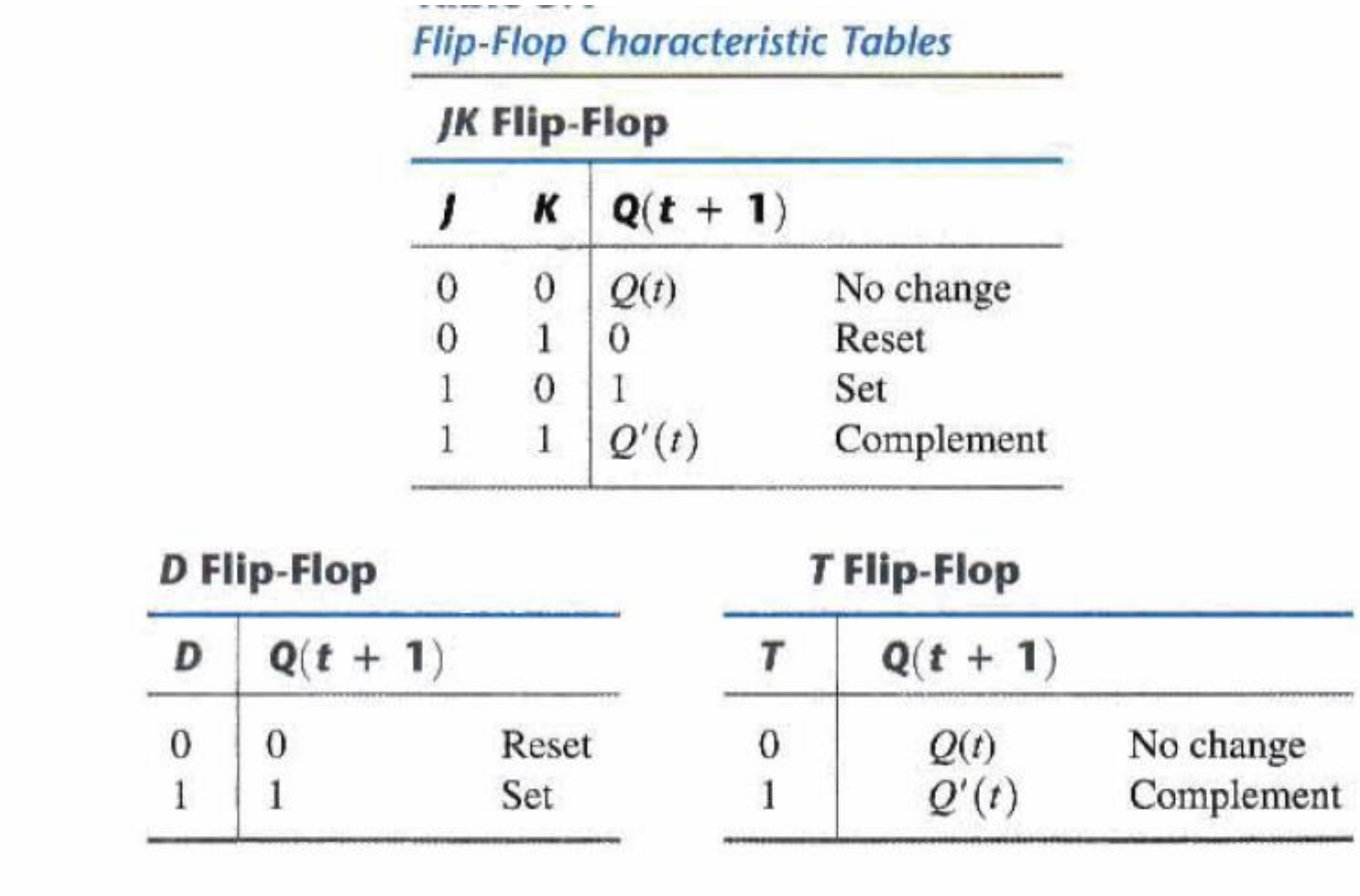
D Flip-Flop

D	**Q(*t* + 1)**	
0	0	Reset
1	1	Set

T Flip-Flop

T	**Q(*t* + 1)**	
0	$Q(t)$	No change
1	$Q'(t)$	Complement

Sequential Circuit: Flipflop



Flip-Flop Characteristic Tables

JK Flip-Flop

J	***K***	**Q(*t* + 1)**	
0	0	$Q(t)$	No change
0	1	0	Reset
1	0	1	Set
1	1	$Q'(t)$	Complement

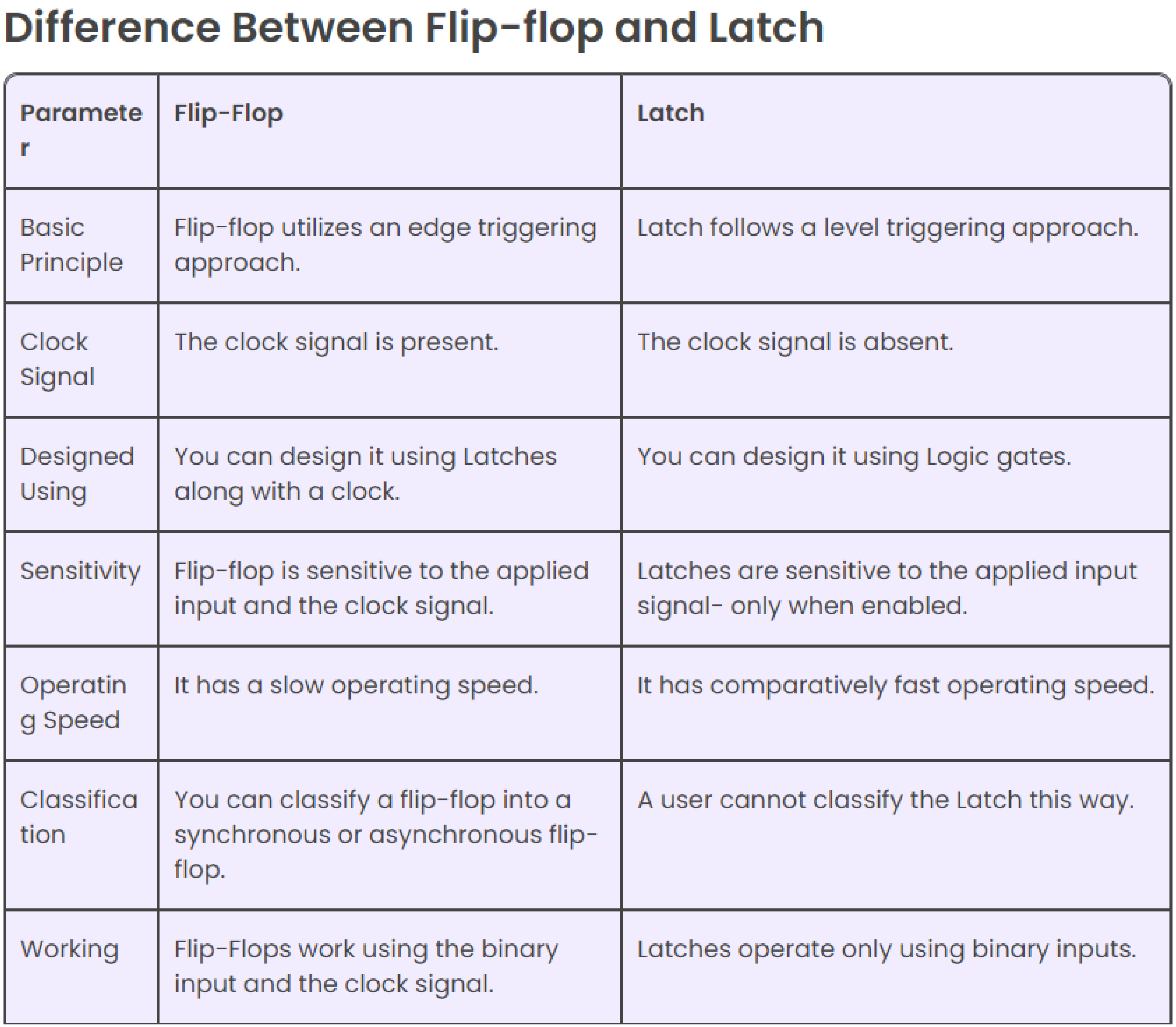
D Flip-Flop

D	**Q(*t* + 1)**	
0	0	Reset
1	1	Set

T Flip-Flop

T	**Q(*t* + 1)**	
0	$Q(t)$	No change
1	$Q'(t)$	Complement

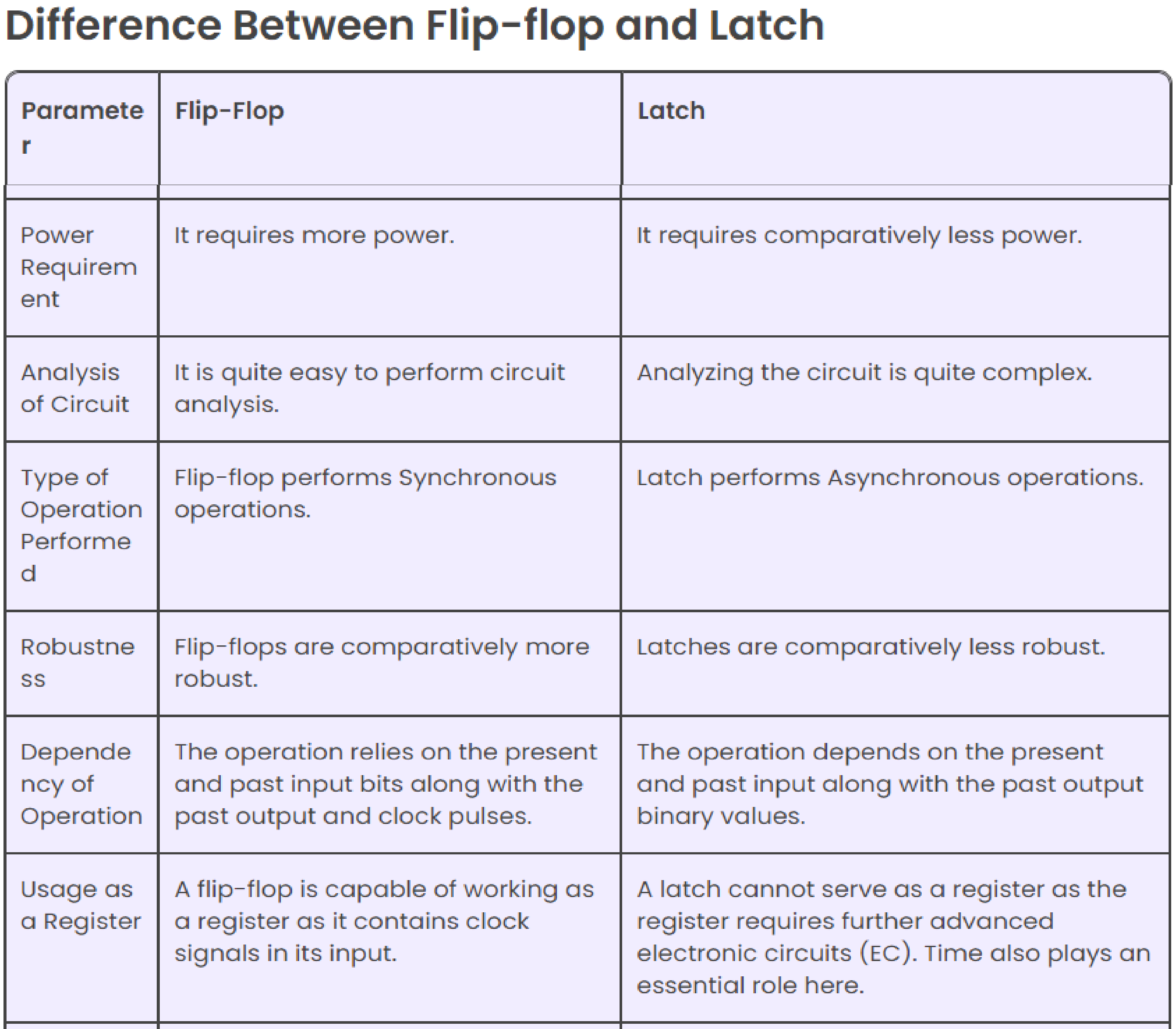
Sequential Circuit: Flipflop



Difference Between Flip-flop and Latch

Parameter	Flip-Flop	Latch
Basic Principle	Flip-flop utilizes an edge triggering approach.	Latch follows a level triggering approach.
Clock Signal	The clock signal is present.	The clock signal is absent.
Designed Using	You can design it using Latches along with a clock.	You can design it using Logic gates.
Sensitivity	Flip-flop is sensitive to the applied input and the clock signal.	Latches are sensitive to the applied input signal- only when enabled.
Operating Speed	It has a slow operating speed.	It has comparatively fast operating speed.
Classification	You can classify a flip-flop into a synchronous or asynchronous flip-flop.	A user cannot classify the Latch this way.
Working	Flip-Flops work using the binary input and the clock signal.	Latches operate only using binary inputs.

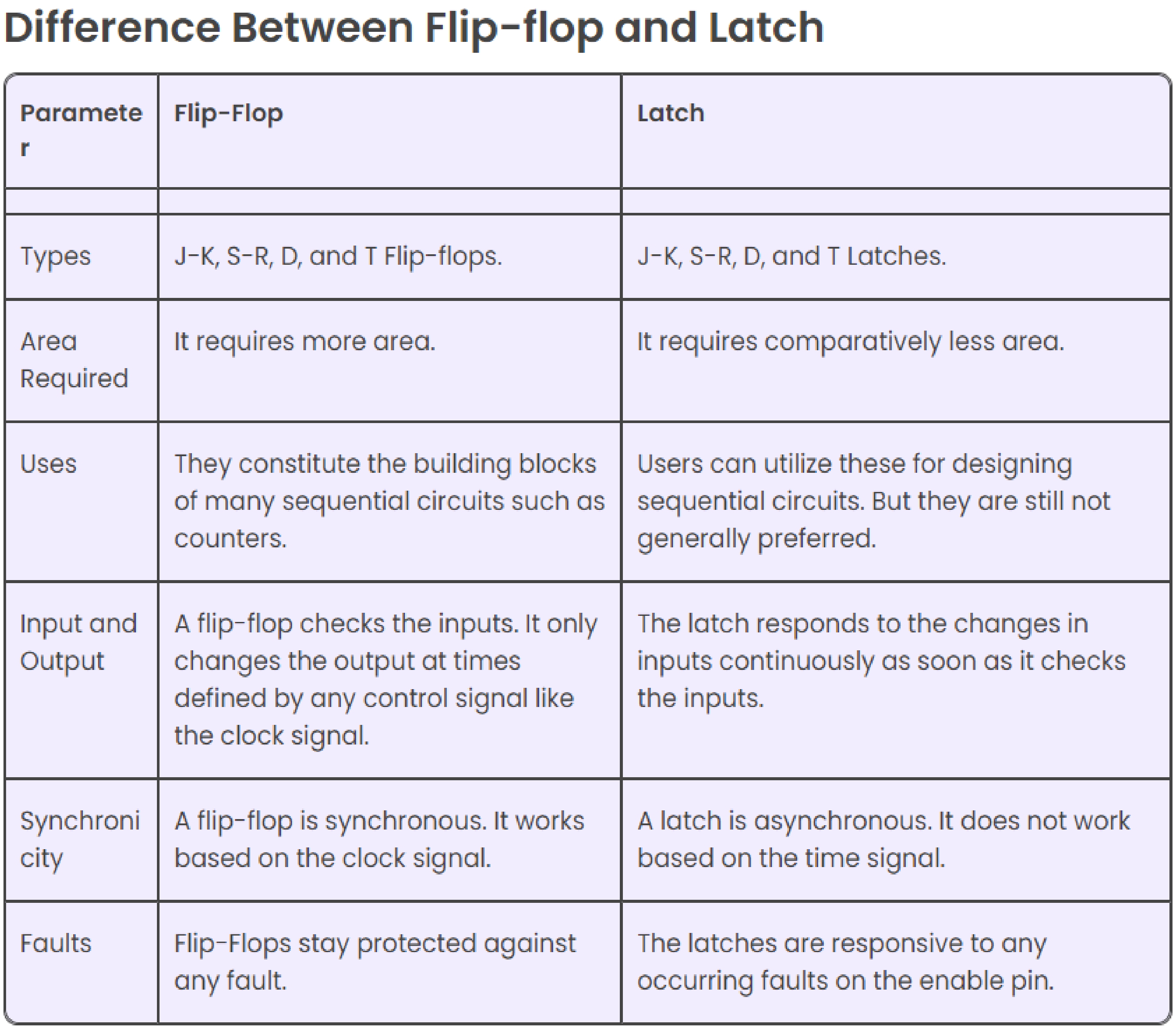
Sequential Circuit: Flipflop



Difference Between Flip-flop and Latch

Parameter	Flip-Flop	Latch
Power Requirement	It requires more power.	It requires comparatively less power.
Analysis of Circuit	It is quite easy to perform circuit analysis.	Analyzing the circuit is quite complex.
Type of Operation Performed	Flip-flop performs Synchronous operations.	Latch performs Asynchronous operations.
Robustness	Flip-flops are comparatively more robust.	Latches are comparatively less robust.
Dependency of Operation	The operation relies on the present and past input bits along with the past output and clock pulses.	The operation depends on the present and past input along with the past output binary values.
Usage as a Register	A flip-flop is capable of working as a register as it contains clock signals in its input.	A latch cannot serve as a register as the register requires further advanced electronic circuits (EC). Time also plays an essential role here.

Sequential Circuit: Flipflop



Difference Between Flip-flop and Latch

Parameter	Flip-Flop	Latch
Types	J-K, S-R, D, and T Flip-flops.	J-K, S-R, D, and T Latches.
Area Required	It requires more area.	It requires comparatively less area.
Uses	They constitute the building blocks of many sequential circuits such as counters.	Users can utilize these for designing sequential circuits. But they are still not generally preferred.
Input and Output	A flip-flop checks the inputs. It only changes the output at times defined by any control signal like the clock signal.	The latch responds to the changes in inputs continuously as soon as it checks the inputs.
Synchronicity	A flip-flop is synchronous. It works based on the clock signal.	A latch is asynchronous. It does not work based on the time signal.
Faults	Flip-Flops stay protected against any fault.	The latches are responsive to any occurring faults on the enable pin.

Conversion of D-Flipflop to T-Flipflop

Step 1: Consider the following **characteristic table** of T flip-flop.

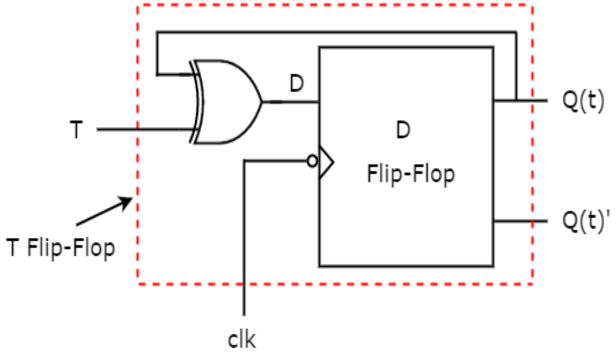
T flip-flop input	Present State	Next State
T	**Q*t***	**Q*t* + 1**
0	0	0
0	1	1
1	0	1
1	1	0

Step 2:

We know that D flip-flop has single input D. So, write down the excitation values of D flip-flop for each combination of present state and next state values. The following table shows the characteristic table of T flip-flop along with the **excitation input** of D flip-flop.

T flip-flop input	Present State	Next State	D flip-flop input
T	**Q*t***	**Q*t* + 1**	**D**
0	0	0	0
0	1	1	1 ←
1	0	1	1 ←
1	1	0	0

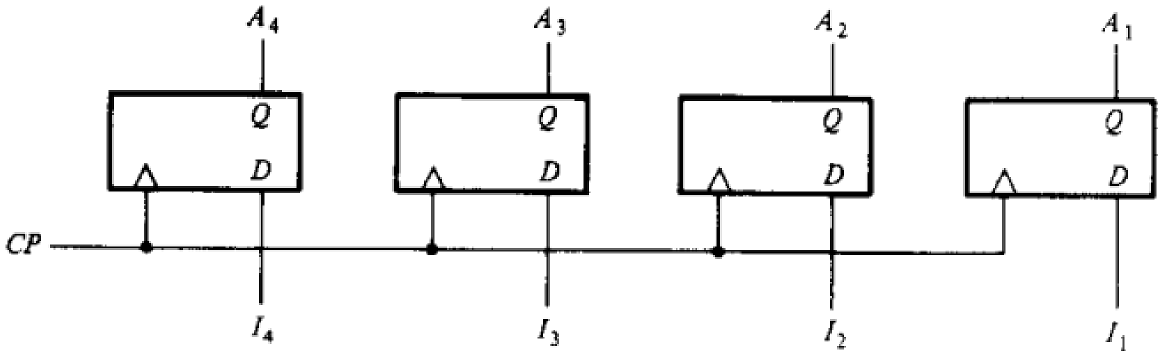
$$D = T'Q(t) + TQ'(t) = T \oplus Q(t)$$



Further resource:
https://www.tutorialspoint.com/digital_circuits/digital_circuits_conversion_of_flip_flops.htm

Sequential Circuit: Register

Flip-flop is a 1 bit memory cell which can be used for storing the digital data. To increase the storage capacity in terms of number of bits, we have to use a group of flip-flop. Such a group of flip-flop is known as a **Register**. The **n-bit register** will consist of **n** number of flip-flop and it is capable of storing an **n-bit** word.



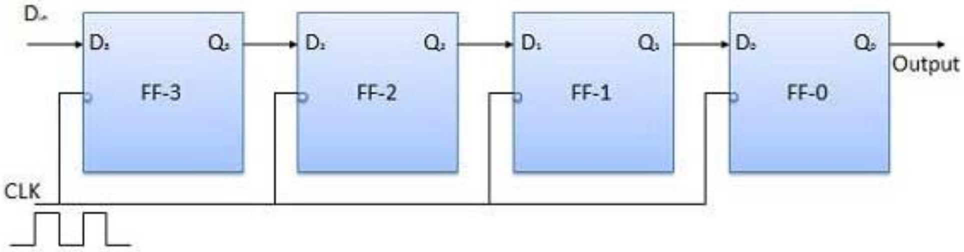
4-bit register

The binary data in a register can be moved within the register from one flip-flop to another. The registers that allow such data transfers are called as **shift registers**. There are four mode of operations of a shift register.

Shift Register:

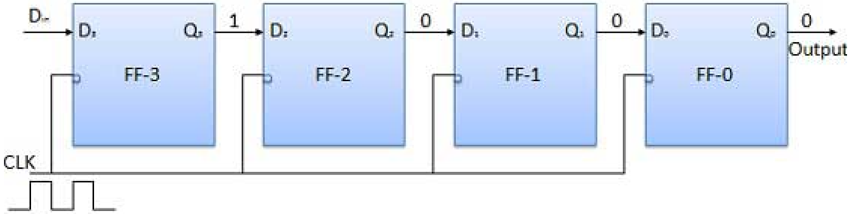
- Serial Input Serial Output
- Serial Input Parallel Output
- Parallel Input Serial Output
- Parallel Input Parallel Output

Serial Input Serial Output Register

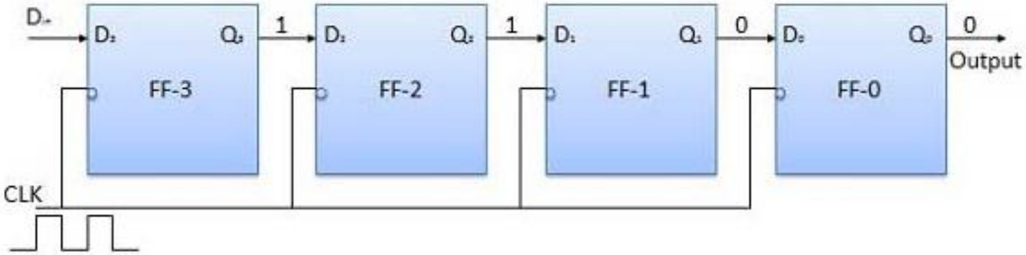


Operation

Before application of clock signal, let Q_3 Q_2 Q_1 Q_0 = 0000 and apply LSB bit of the number to be entered to D_{in}. So D_{in} = D_3 = 1. Apply the clock. On the first falling edge of clock, the FF-3 is set, and stored word in the register is Q_3 Q_2 Q_1 Q_0 = 1000.

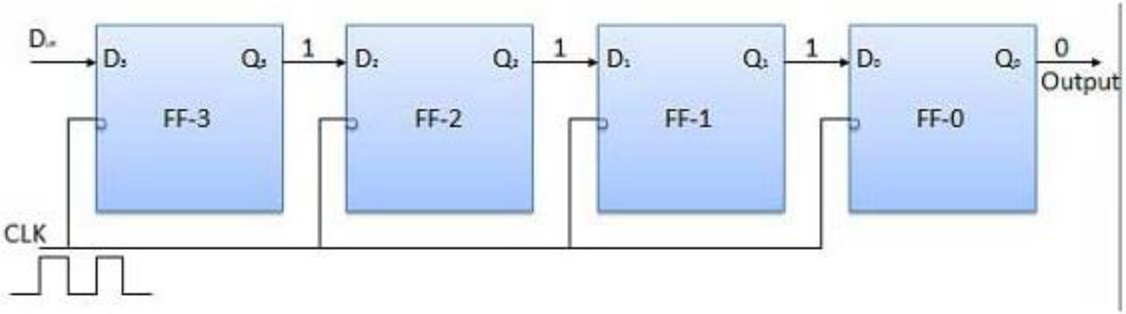


Apply the next bit to D_{in}. So D_{in} = 1. As soon as the next negative edge of the clock hits, FF-2 will set and the stored word change to Q_3 Q_2 Q_1 Q_0 = 1100.

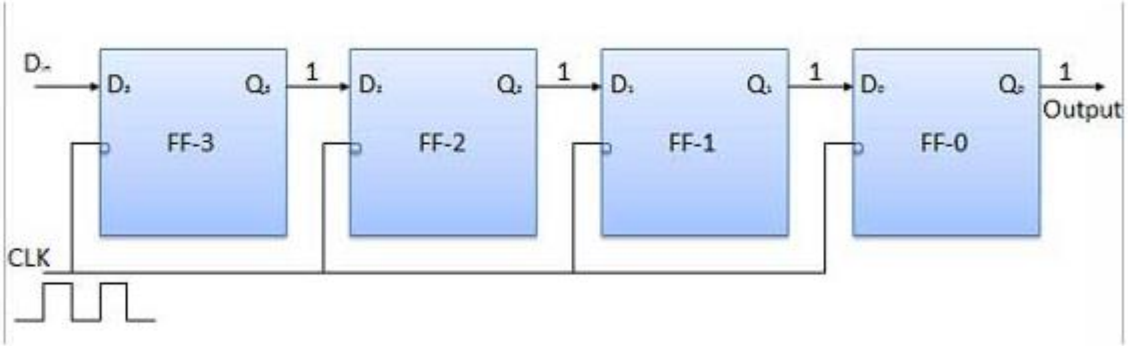


Serial Input Serial Output Register

Apply the next bit to be stored i.e. 1 to D_{in}. Apply the clock pulse. As soon as the third negative clock edge hits, FF-1 will be set and output will be modified to Q_3 Q_2 Q_1 Q_0 = 1110.

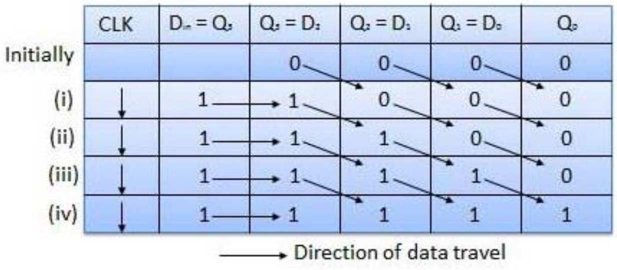


Similarly with D_{in} = 1 and with the fourth negative clock edge arriving, the stored word in the register is Q_3 Q_2 Q_1 Q_0 = 1111.



Serial Input Serial Output Register

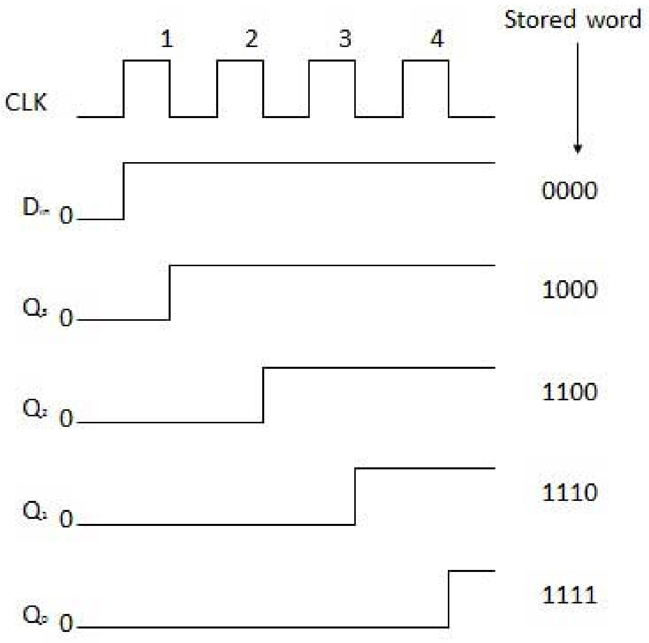
Truth Table



	CLK	$D_{in} = Q_3$	$Q_3 = D_2$	$Q_2 = D_1$	$Q_1 = D_0$	Q_0
Initially			0	0	0	0
(i)	↓	1	1	0	0	0
(ii)	↓	1	1	1	0	0
(iii)	↓	1	1	1	1	0
(iv)	↓	1	1	1	1	1

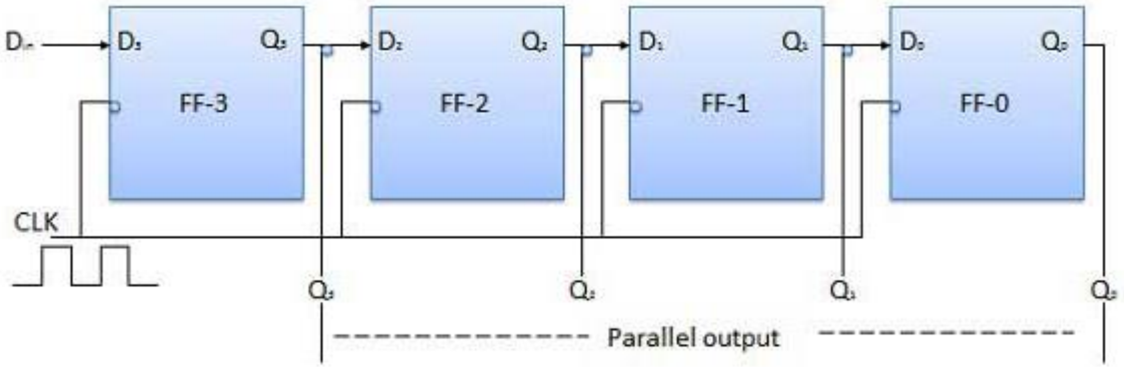
⟶ Direction of data travel

Waveforms



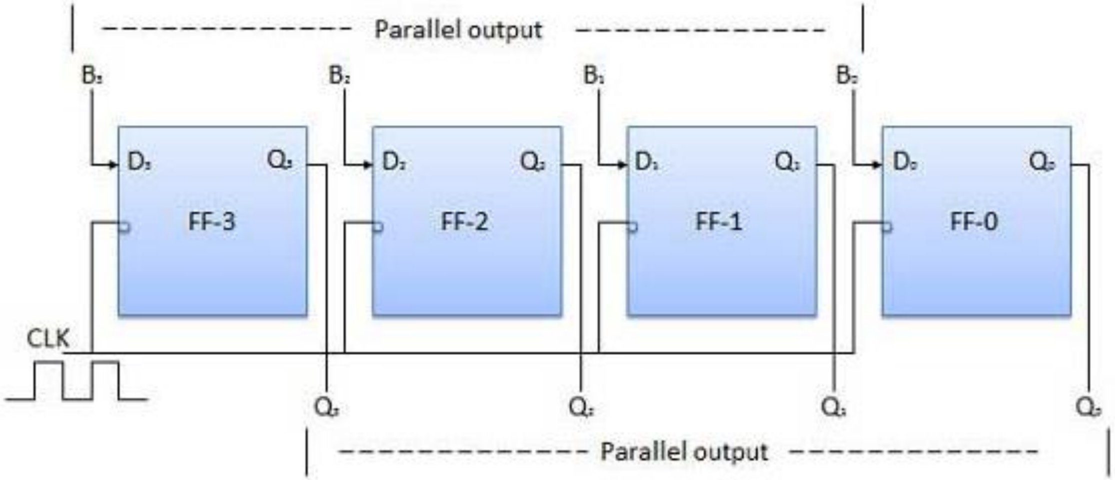
Serial Input Parallel Output Register

- In such types of operations, the data is entered serially and taken out in parallel fashion.
- Data is loaded bit by bit. The outputs are disabled as long as the data is loading.
- As soon as the data loading gets completed, all the flip-flops contain their required data, the outputs are enabled so that all the loaded data is made available over all the output lines at the same time.
- 4 clock cycles are required to load a four bit word. Hence the speed of operation of SIPO mode is same as that of SISO mode.



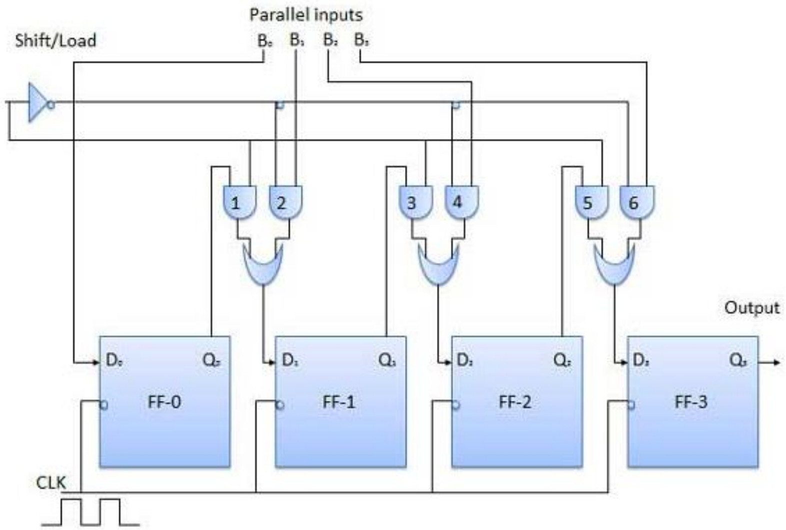
Parallel Input Parallel Output Register

In this mode, the 4 bit binary input B_0, B_1, B_2, B_3 is applied to the data inputs D_0, D_1, D_2, D_3 respectively of the four flip-flops. As soon as a negative clock edge is applied, the input binary bits will be loaded into the flip-flops simultaneously. The loaded bits will appear simultaneously to the output side. Only clock pulse is essential to load all the bits.



Parallel Input Serial Output Register

- Data bits are entered in parallel fashion.
- The circuit shown below is a four bit parallel input serial output register.
- Output of previous Flip Flop is connected to the input of the next one via a combinational circuit.
- The binary input word B_0, B_1, B_2, B_3 is applied though the same combinational circuit.
- There are two modes in which this circuit can work namely - shift mode or load mode.



Load mode

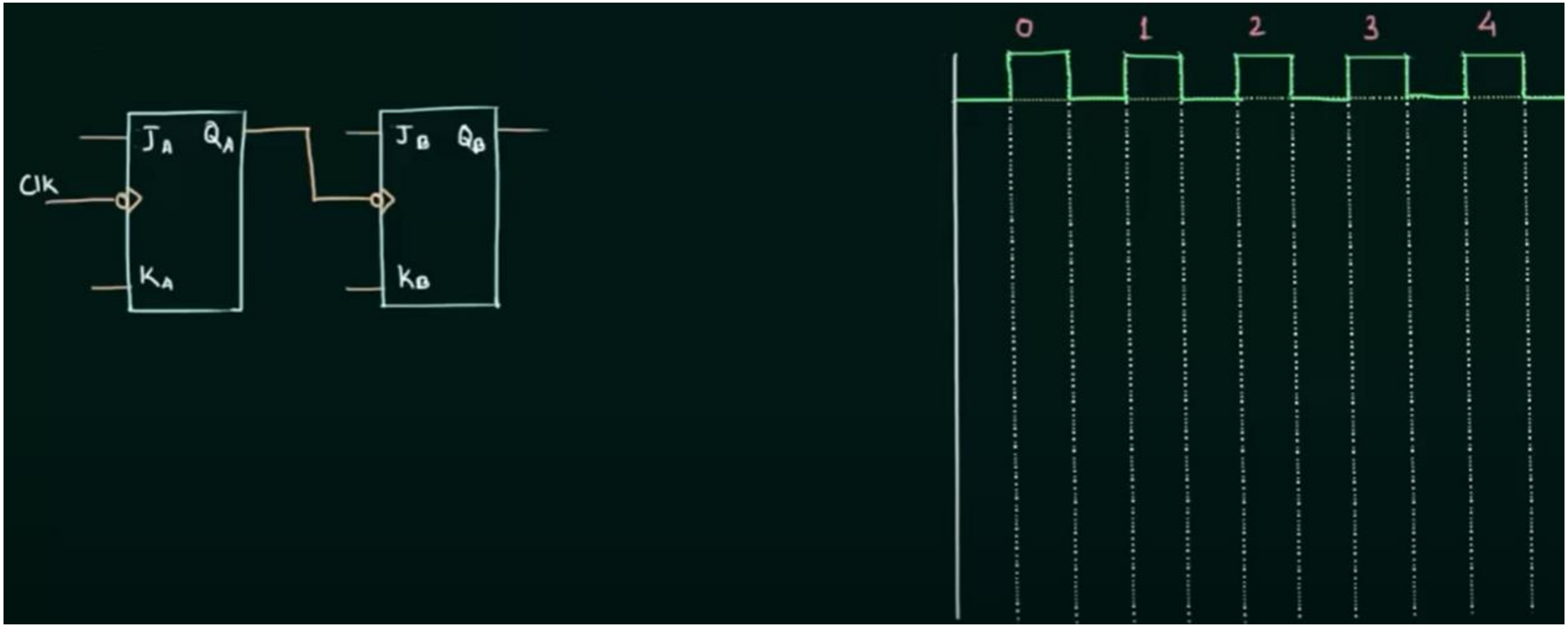
When the shift/load bar line is low (0), the AND gate 2, 4 and 6 become active they will pass B_1, B_2, B_3 bits to the corresponding flip-flops. On the low going edge of clock, the binary input B_0, B_1, B_2, B_3 will get loaded into the corresponding flip-flops. Thus parallel loading takes place.

Shift mode

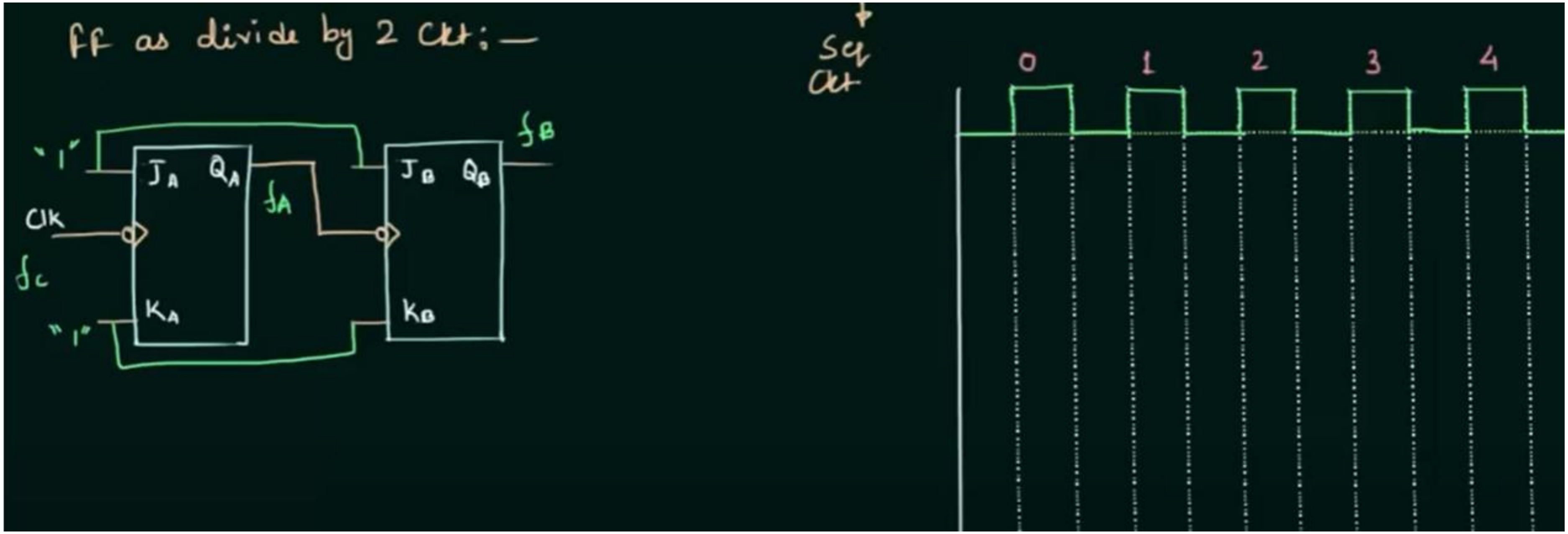
When the shift/load bar line is low (1), the AND gate 2, 4 and 6 become inactive. Hence the parallel loading of the data becomes impossible. But the AND gate 1,3 and 5 become active. Therefore the shifting of data from left to right bit by bit on application of clock pulses. Thus the parallel in serial out operation takes place.

Source: https://www.tutorialspoint.com/computer_logical_organization/digital_registers.htm

Asynchronous Counter



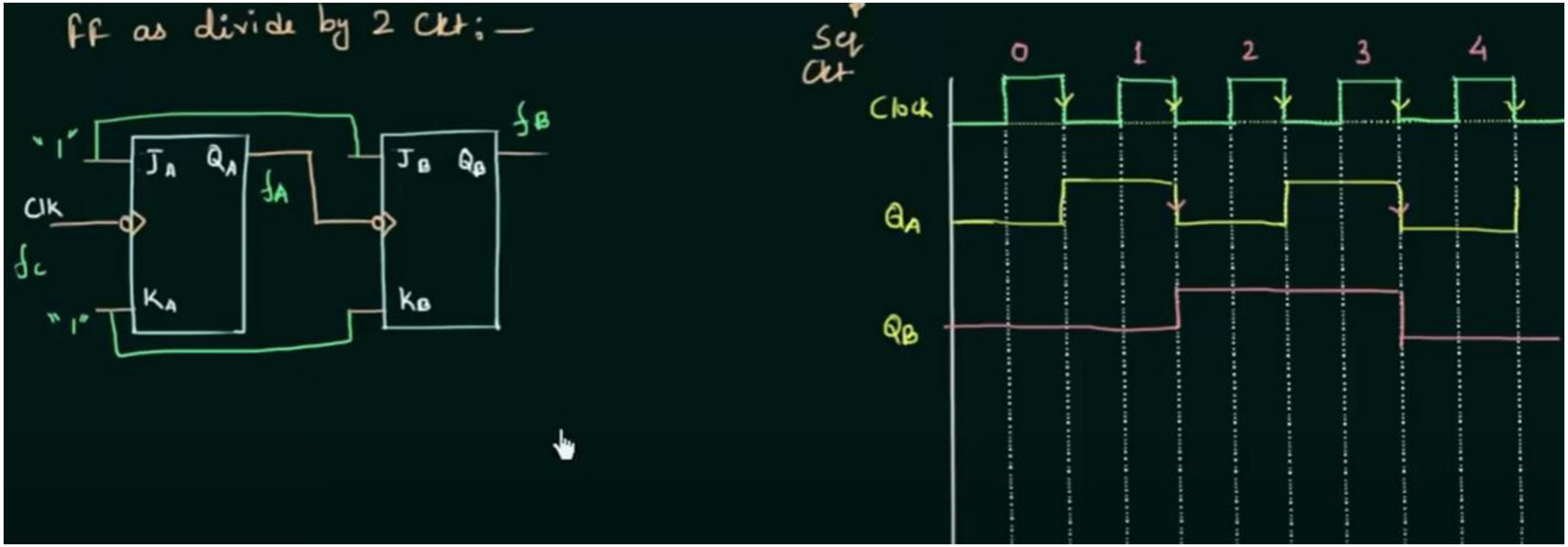
Asynchronous Counter



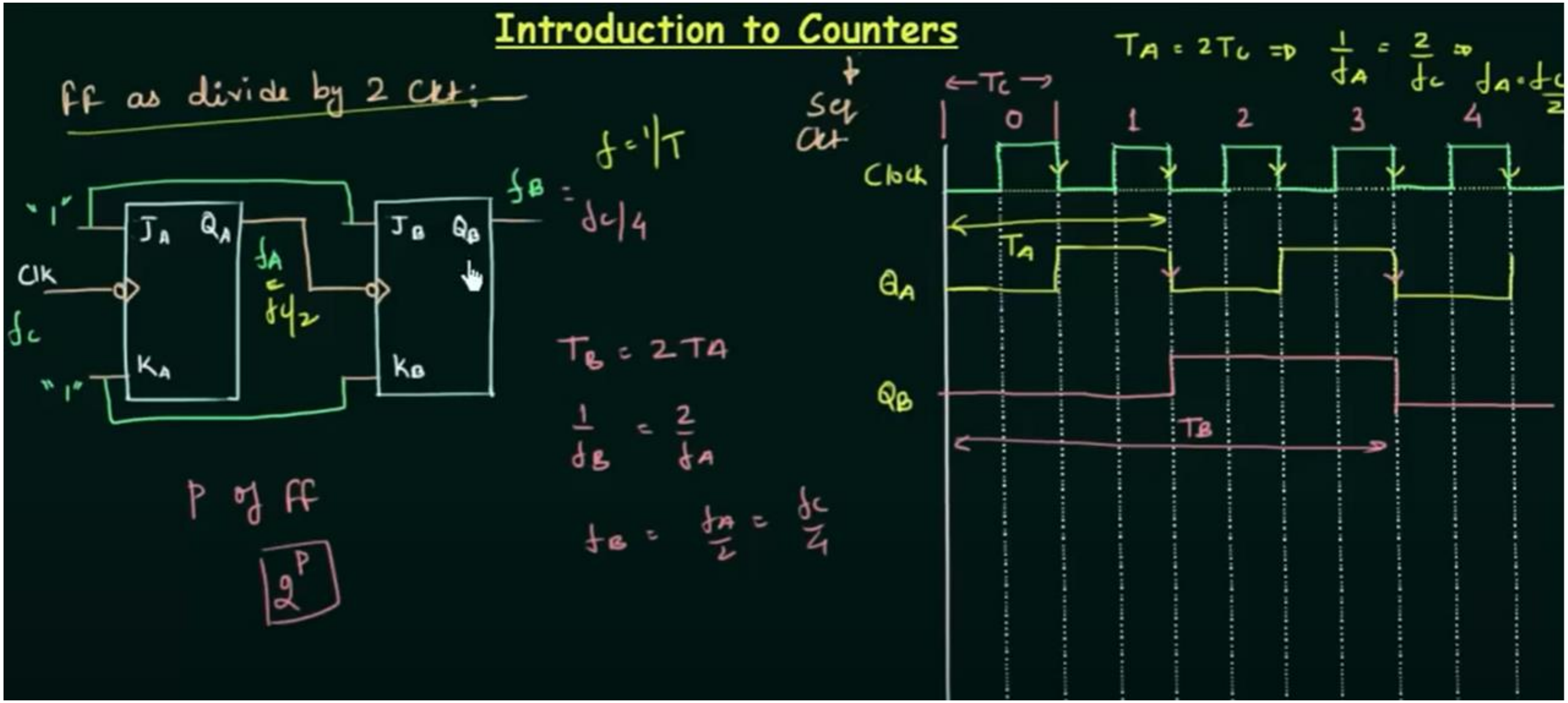
Negative Edge Triggered

Clock frequency in both FFs are not same → Asynchronous Counter

Asynchronous Counter



Asynchronous Counter



Introduction to Counters

ff as divide by 2 ckt:—

$f = 1/T$

seq ckt

$T_A = 2T_c \Rightarrow \frac{1}{f_A} = \frac{2}{f_c} \Rightarrow f_A = \frac{f_c}{2}$

$f_B = f_c/4$

$f_A = f_c/2$

$T_B = 2T_A$

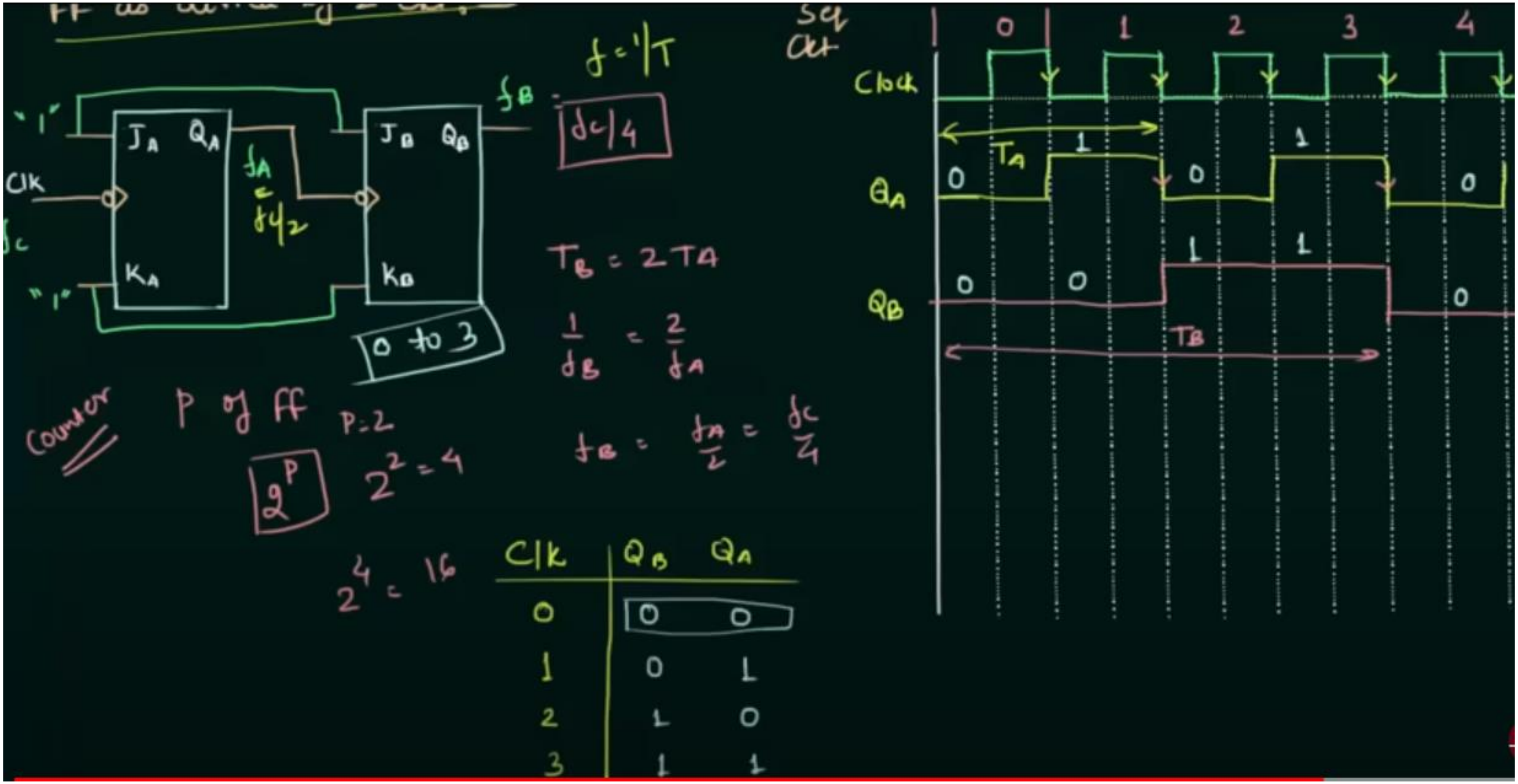
$\frac{1}{f_B} = \frac{2}{f_A}$

$f_B = \frac{f_A}{2} = \frac{f_c}{4}$

P of ff

$\boxed{2^P}$

Asynchronous Counter



$f = 1/T$

$f_B = \boxed{f_c/4}$

$\boxed{0 \text{ to } 3}$

$T_B = 2T_A$

$\frac{1}{f_B} = \frac{2}{f_A}$

$f_B = \frac{f_A}{2} = \frac{f_c}{4}$

counter

P of FF

$\boxed{2^P}$

$P = 2$

$2^2 = 4$

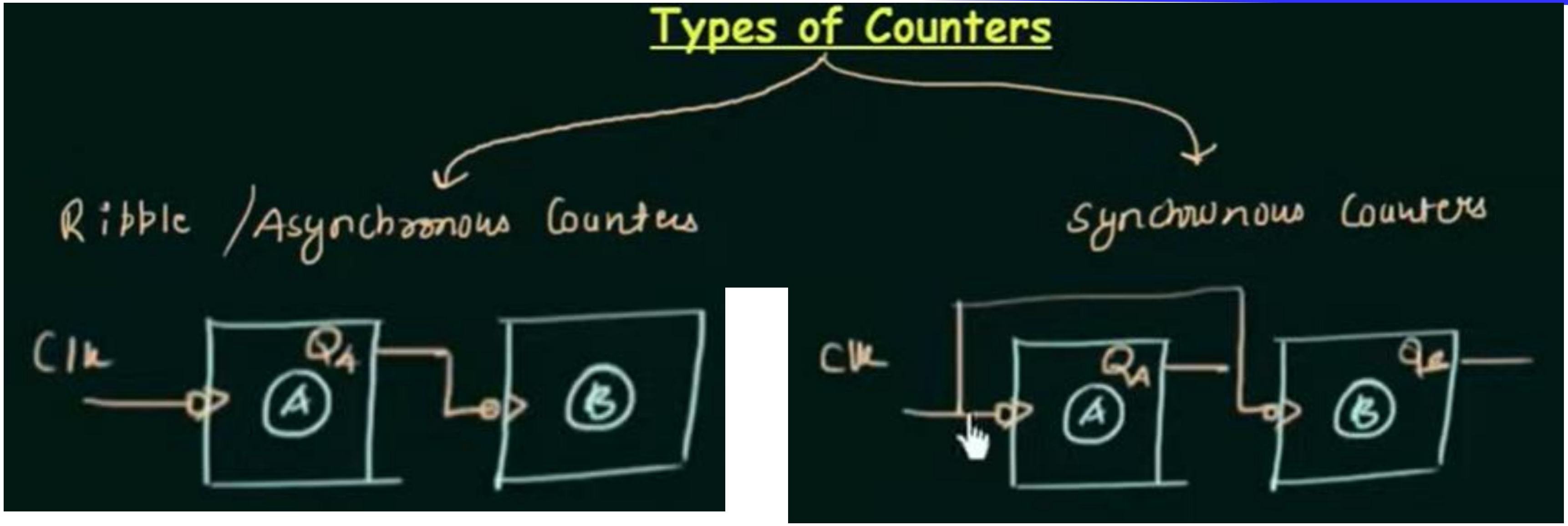
$2^4 = 16$

Clk	Q_B	Q_A
0	0	0
1	0	1
2	1	0
3	1	1

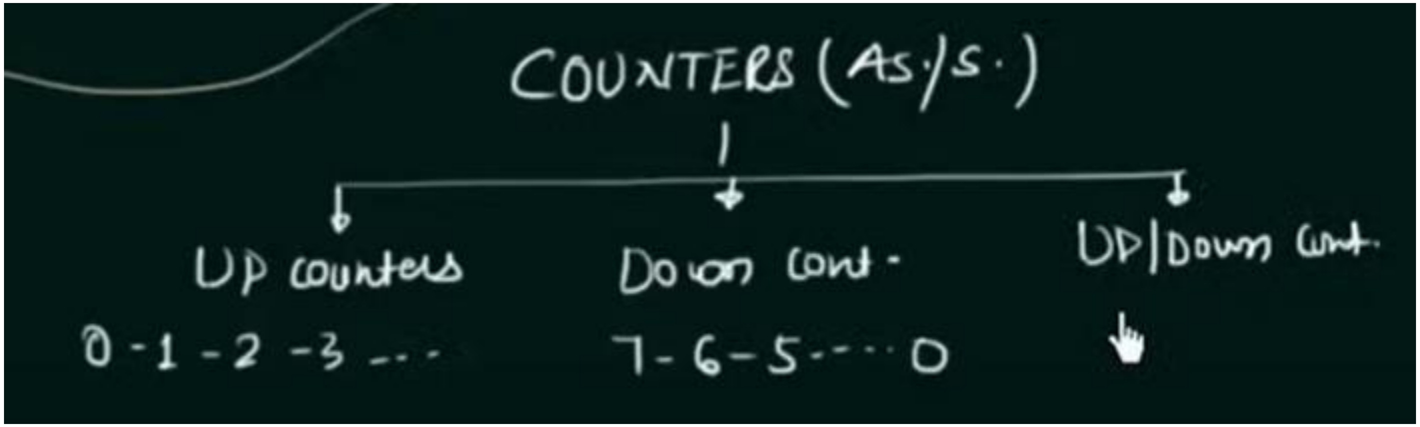
Total number of states to count: 4

Source:
https://www.youtube.com/watch?v=iaIu5SYmWVM&list=PLBlnK6fEyqRjMH3mWf6kwqiTbT798eAOm&index=179

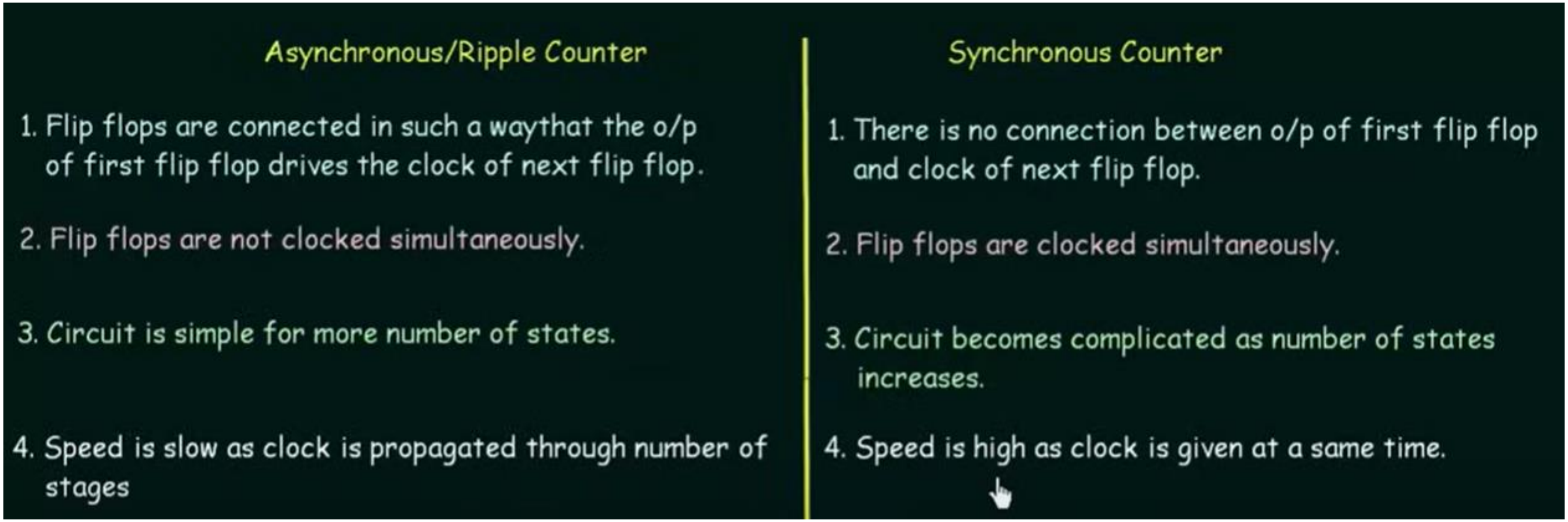
Asynchronous Counter



In synchronous: Clock frequency is same for all flipflops.
In asynchronous: Clock frequency is different for each flipflop.

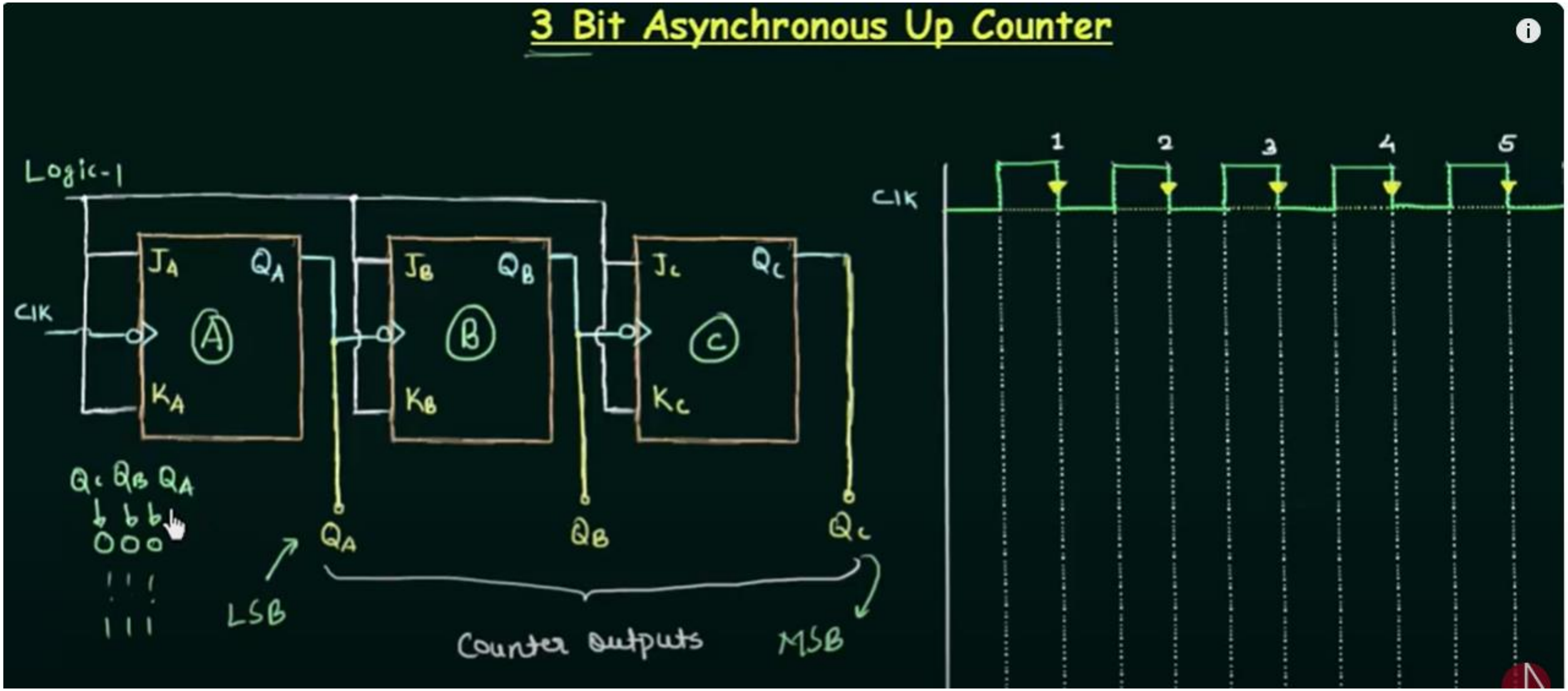


Asynchronous Counter



Asynchronous/Ripple Counter	Synchronous Counter
1. Flip flops are connected in such a waythat the o/p of first flip flop drives the clock of next flip flop.	1. There is no connection between o/p of first flip flop and clock of next flip flop.
2. Flip flops are not clocked simultaneously.	2. Flip flops are clocked simultaneously.
3. Circuit is simple for more number of states.	3. Circuit becomes complicated as number of states increases.
4. Speed is slow as clock is propagated through number of stages	4. Speed is high as clock is given at a same time.

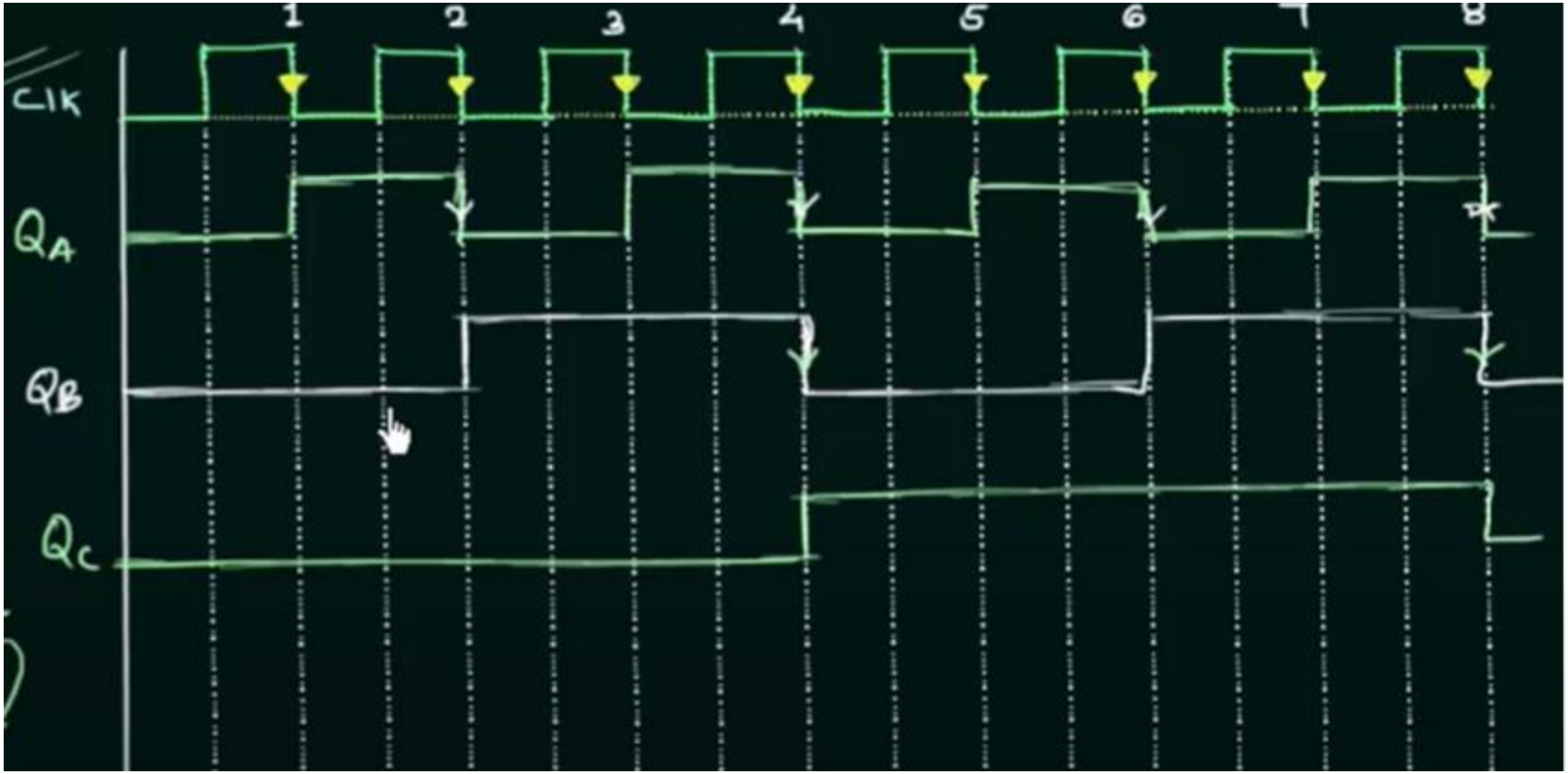
Asynchronous Up Counter



Source:
https://www.youtube.com/watch?v=s1DSZEaCX_g&list=PLBlnK6fEyqRjMH3mWf6kwqiTbT798eAOm&index=180

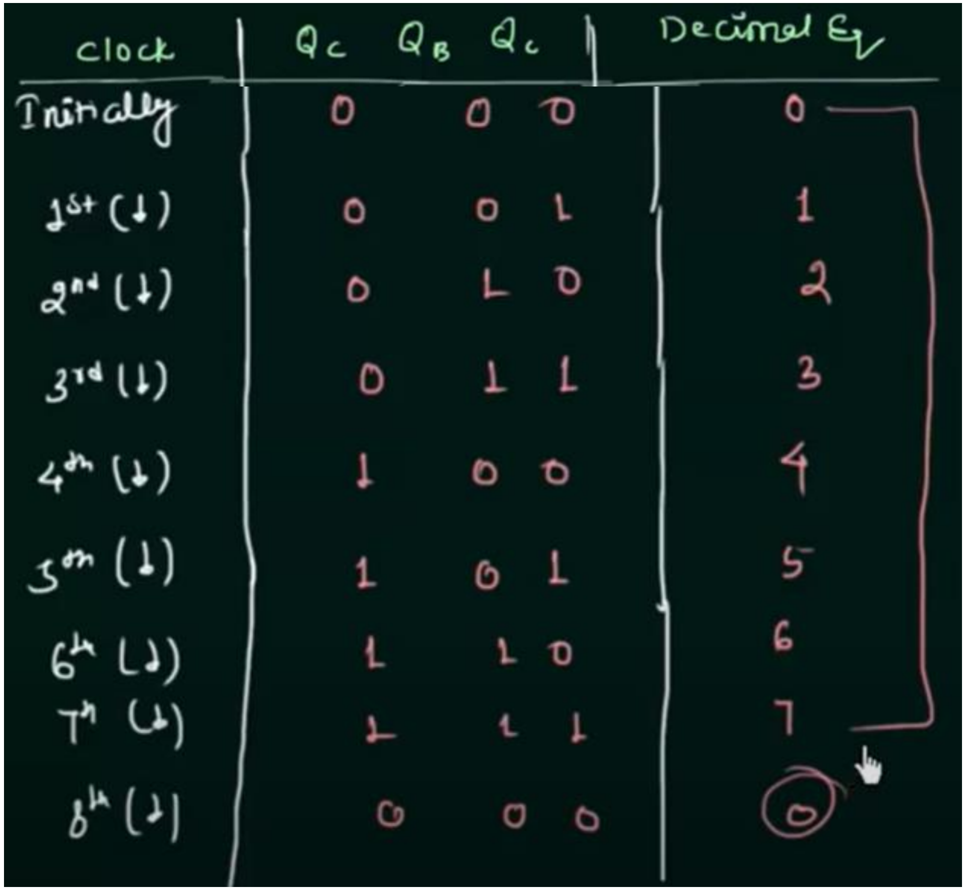
Asynchronous Up Counter

Timing Diagram/ Clock Diagram:

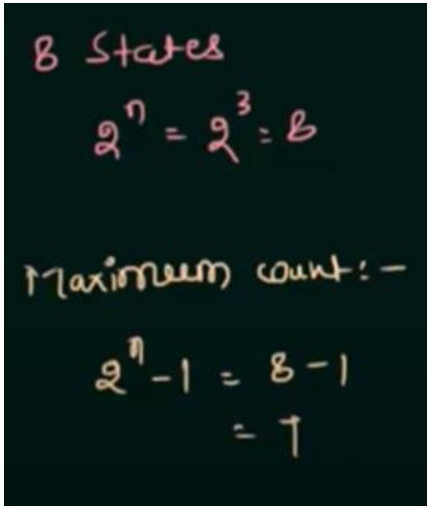


Asynchronous Up Counter

Truth Table Based on the Timing Diagram/ Clock Diagram:



clock	Q_C	Q_B	Q_c	Decimal Eq
Initially	0	0	0	0
1st (↓)	0	0	1	1
2nd (↓)	0	1	0	2
3rd (↓)	0	1	1	3
4th (↓)	1	0	0	4
5th (↓)	1	0	1	5
6th (↓)	1	1	0	6
7th (↓)	1	1	1	7
8th (↓)	0	0	0	0



8 States

$2^n = 2^3 = 8$

Maximum count:-

$2^n - 1 = 8 - 1$

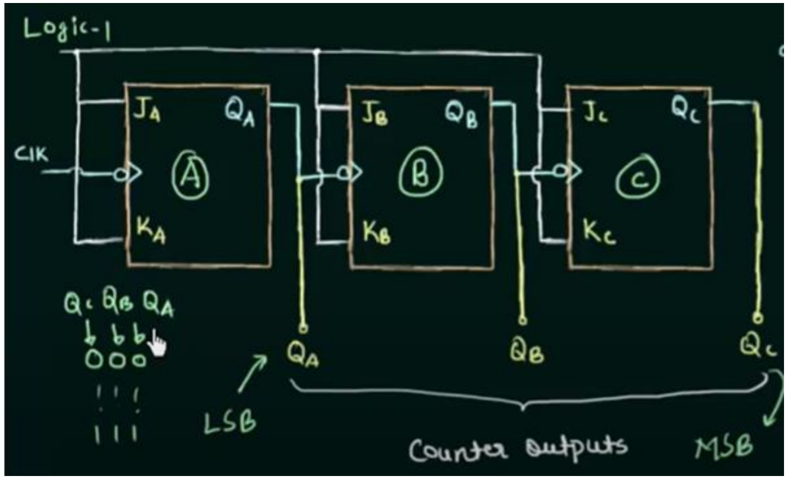
$= 7$

Asynchronous Down Counter

Up Counter

Output is taken from Q

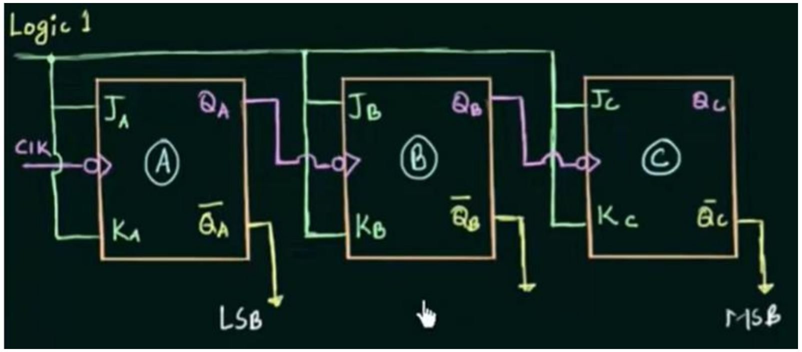
Clock signal is taken from Q



Method 1: Down Counter

Output is taken from Q'

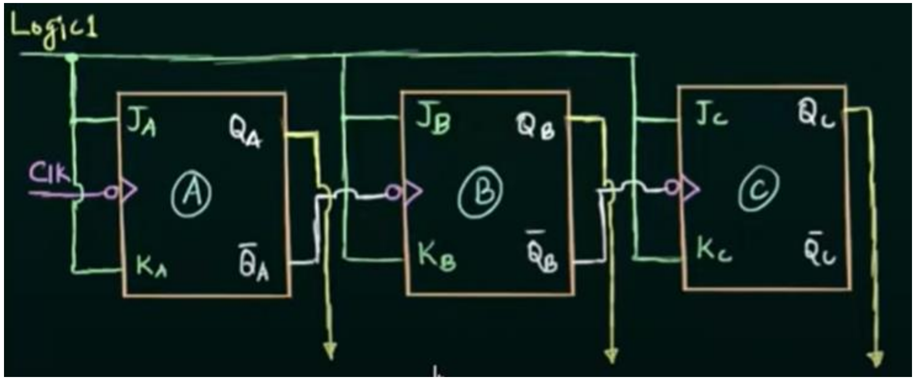
Clock signal is taken from Q



Method 2: Down Counter

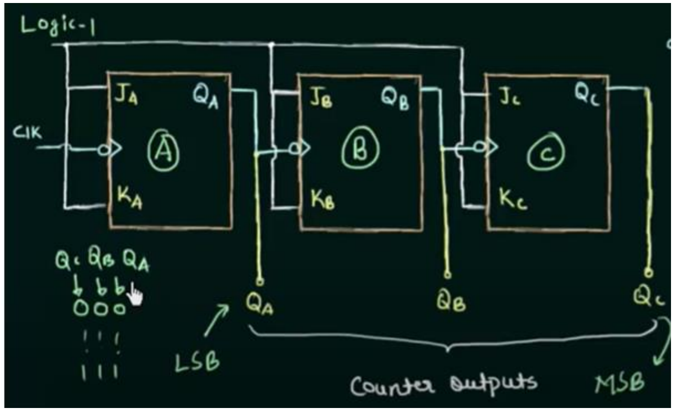
Output is taken from Q

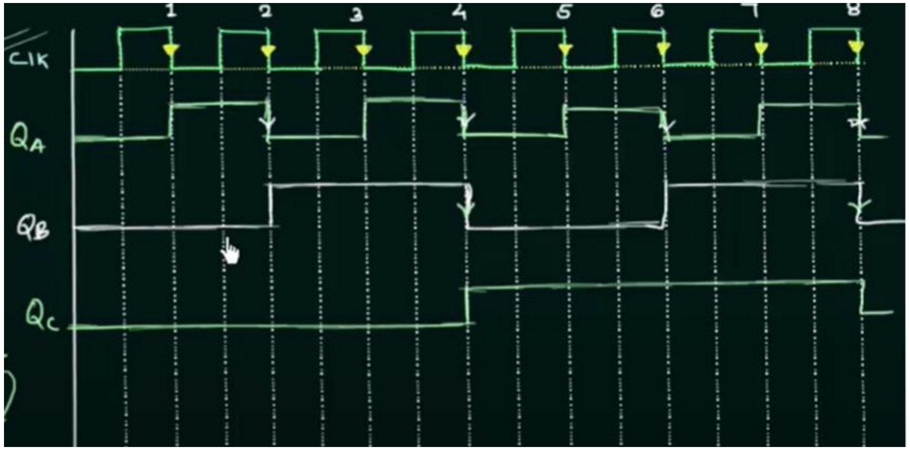
Clock signal is taken from Q'



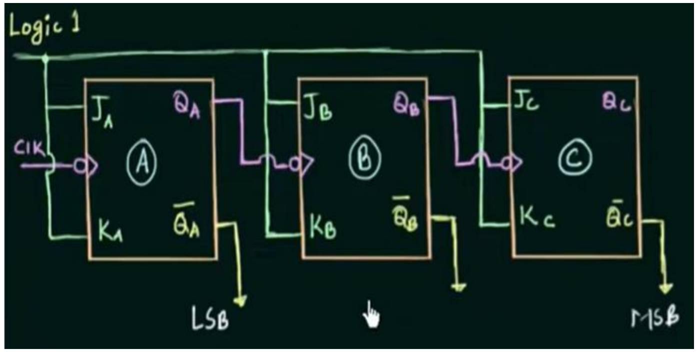
Asynchronous Down Counter

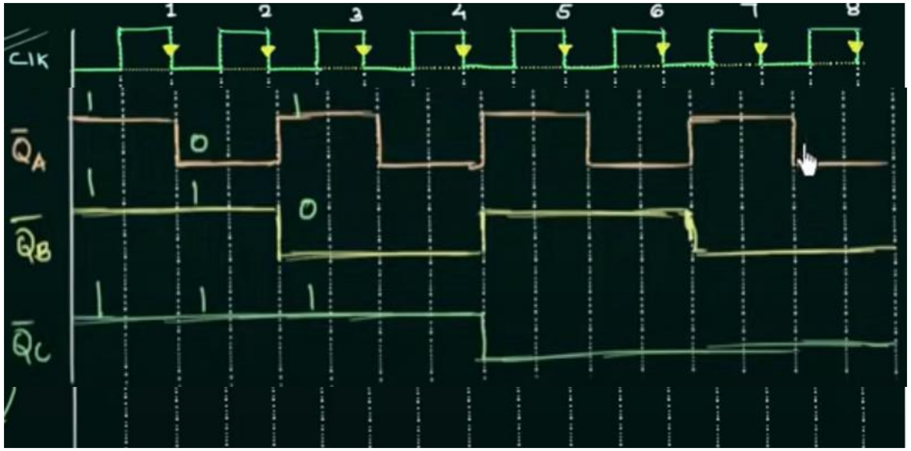
Up Counter





Method 1: Down Counter

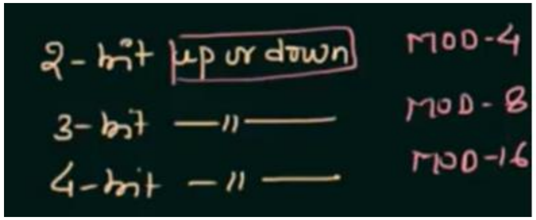




Source: https://www.youtube.com/watch?v=noUcCs2zNaI&list=PLBlnK6fEyqRjMH3mWf6kwqiTbT798eAOm&index=182

Asynchronous Mod Counter

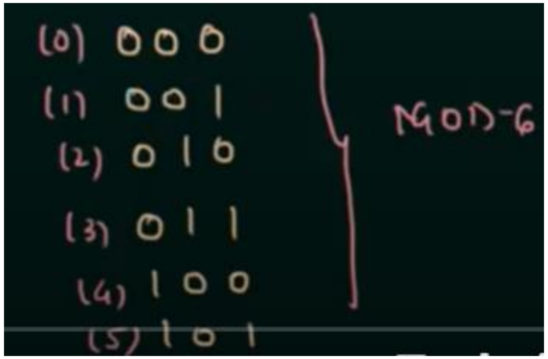
No. of bits: n
Modulus (Mod) number: 2^n



2-bit up or down MOD-4
3-bit —〃—— MOD-8
4-bit —〃—— MOD-16

Design an Mod-6 Counter:

No. of states: 6
Maximum count: 6-1=5



(0) 0 0 0
(1) 0 0 1
(2) 0 1 0
(3) 0 1 1
(4) 1 0 0
(5) 1 0 1

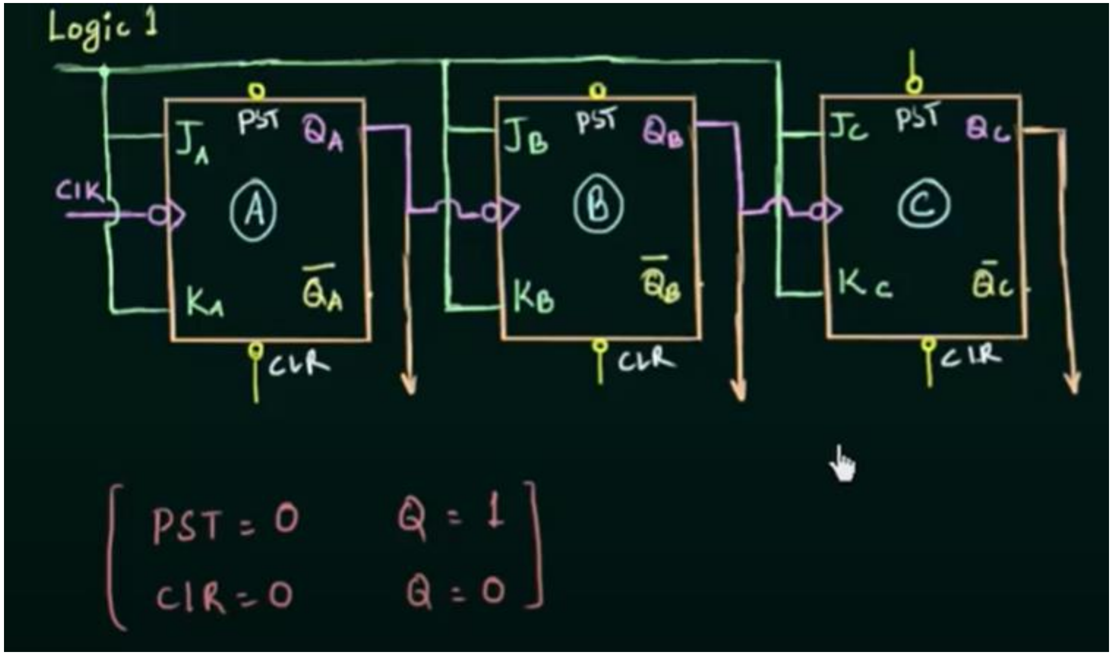
MOD-6

Lowest order Bit number ?

$$2^2 < 6 < 2^3$$

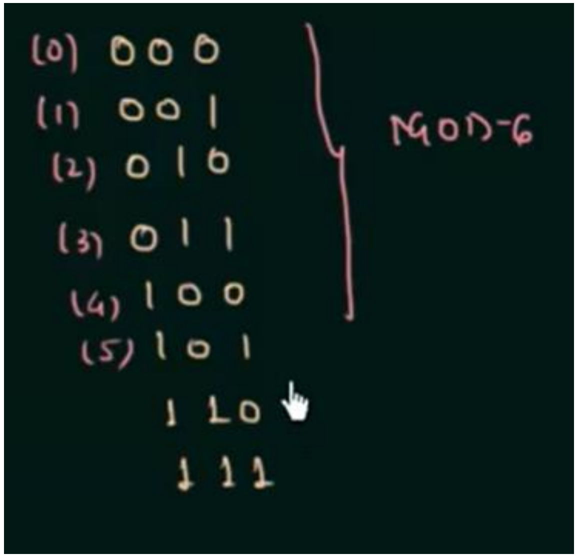
$$n = 3$$

Asynchronous Mod Counter



PST = 0 Q = 1

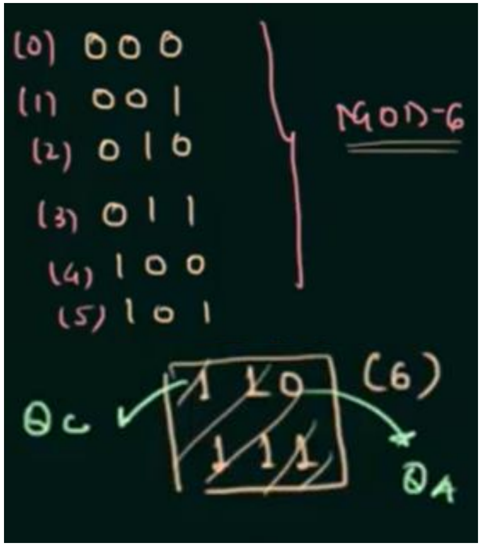
ClR = 0 Q = 0



(0) 0 0 0
(1) 0 0 1
(2) 0 1 0
(3) 0 1 1
(4) 1 0 0
(5) 1 0 1
1 1 0
1 1 1

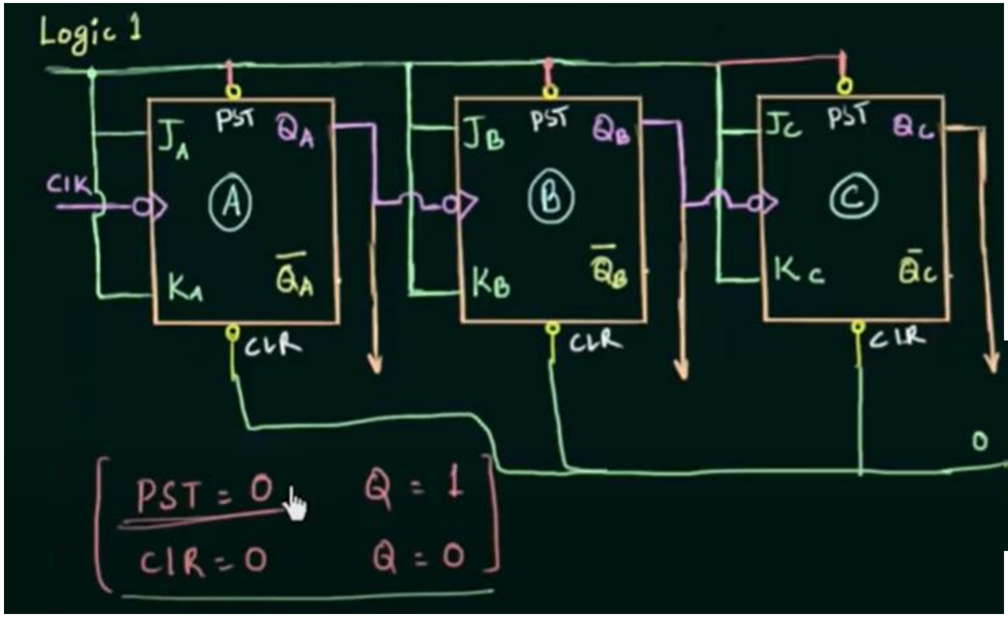
MOD-6

Asynchronous Mod Counter



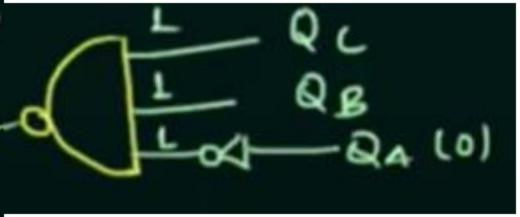
(0)	0	0	0
(1)	0	0	1
(2)	0	1	0
(3)	0	1	1
(4)	1	0	0
(5)	1	0	1
(6)	1	1	0

MOD-6

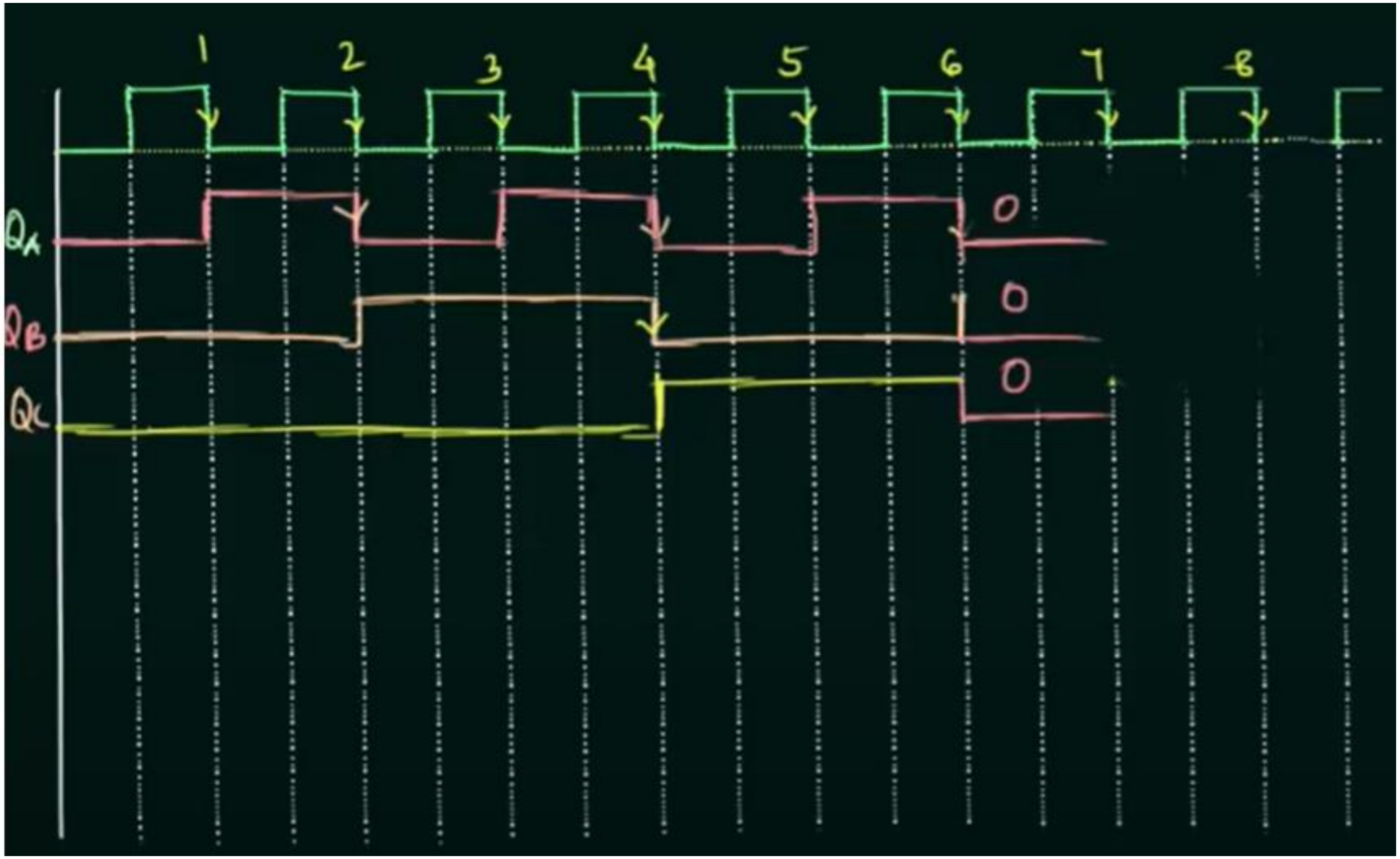


PST = 0 Q = 1

ClR = 0 Q = 0

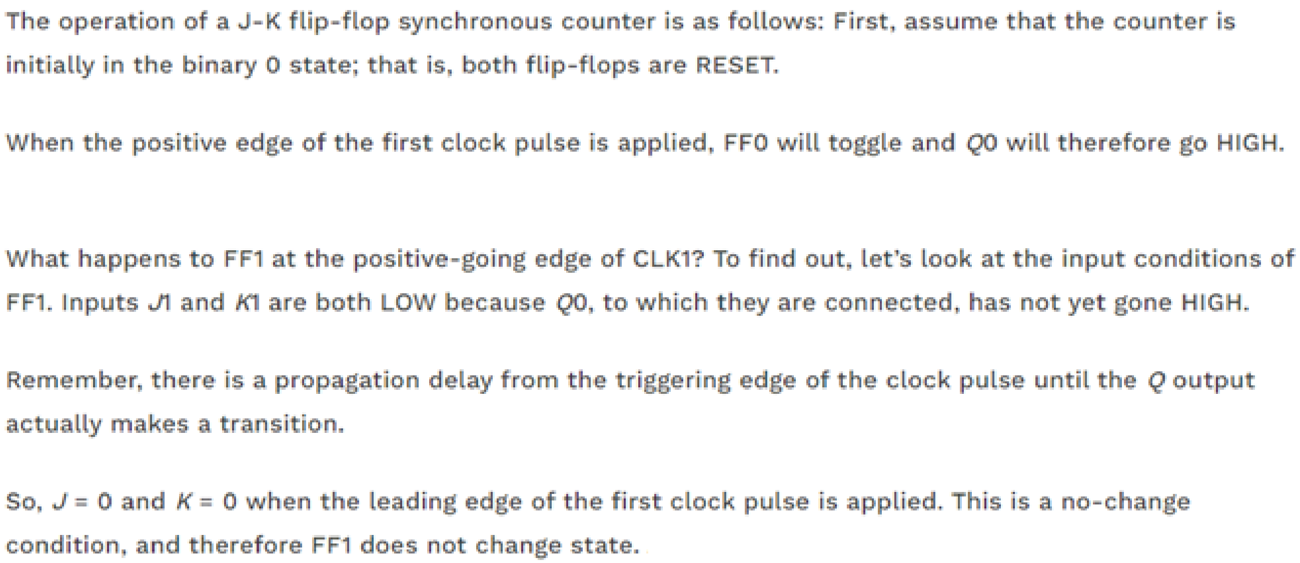


Asynchronous Mod Counter



2-bit Synchronous Up Counter

Unlike the asynchronous counter, synchronous counter has one global clock which drives each flip flop so output changes in parallel. The one advantage of synchronous counter over asynchronous counter is, it can operate on higher frequency than asynchronous counter as it does not have cumulative delay because of same clock is given to each flip flop. It is also called as parallel counter.



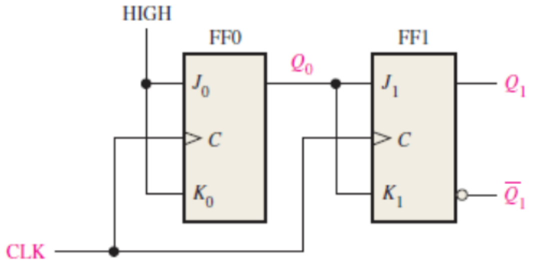
The operation of a J-K flip-flop synchronous counter is as follows: First, assume that the counter is initially in the binary 0 state; that is, both flip-flops are RESET.

When the positive edge of the first clock pulse is applied, FF0 will toggle and *Q*0 will therefore go HIGH.

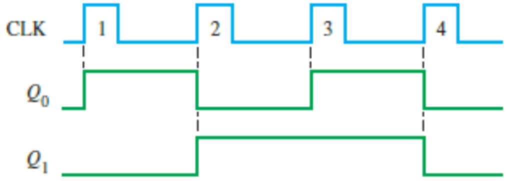
What happens to FF1 at the positive-going edge of CLK1? To find out, let's look at the input conditions of FF1. Inputs *J*1 and *K*1 are both LOW because *Q*0, to which they are connected, has not yet gone HIGH.

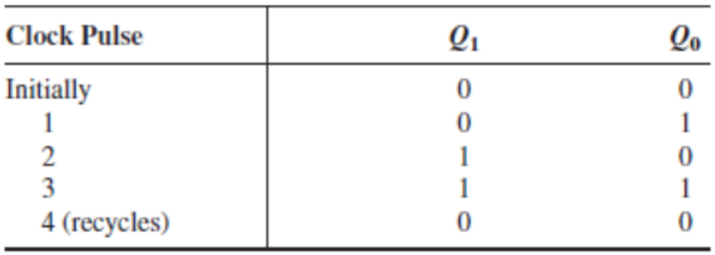
Remember, there is a propagation delay from the triggering edge of the clock pulse until the *Q* output actually makes a transition.

So, *J* = 0 and *K* = 0 when the leading edge of the first clock pulse is applied. This is a no-change condition, and therefore FF1 does not change state.



(a) J-K flip-flop



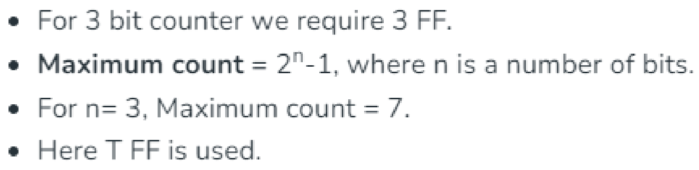


Clock Pulse	Q_1	Q_0
Initially	0	0
1	0	1
2	1	0
3	1	1
4 (recycles)	0	0

3-Bit Synchronous Down Counter

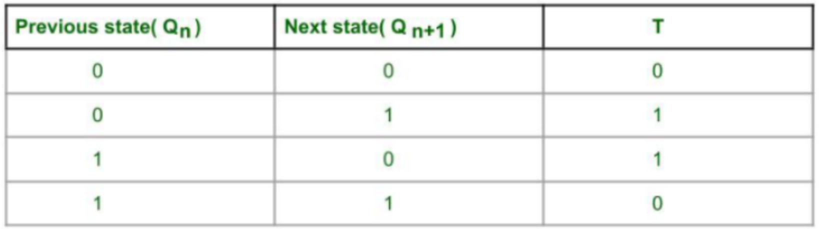
Design : The steps involves in design are

1. Decide the number of Flip flops –



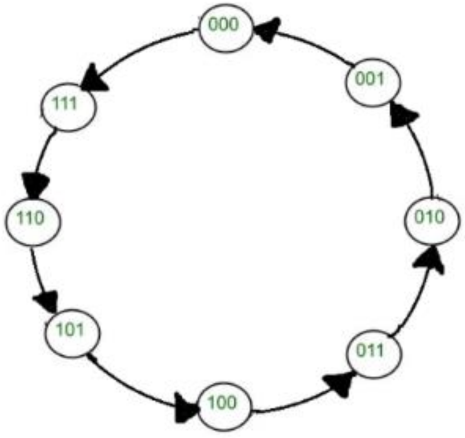
- For 3 bit counter we require 3 FF.
- **Maximum count** = 2^n-1, where n is a number of bits.
- For n= 3, Maximum count = 7.
- Here T FF is used.

2. Write excitation table of FF –

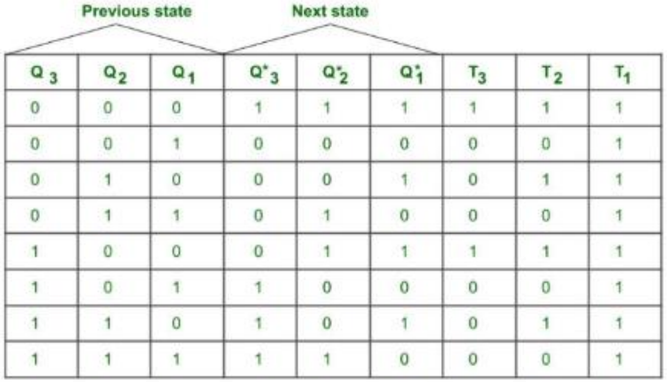


Previous state(Q_n)	Next state(Q_{n+1})	T
0	0	0
0	1	1
1	0	1
1	1	0

3. Draw State diagram and circuit excitation table – Number of states = $2^{n,}$ where n is number of bits.



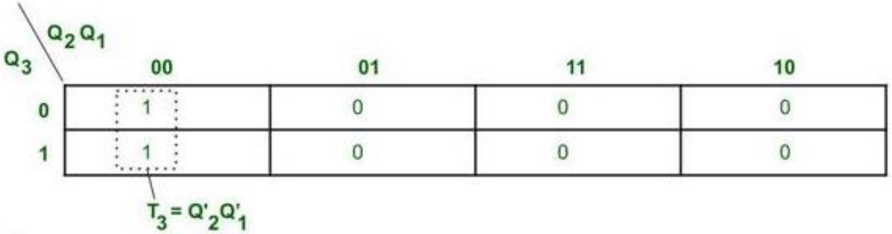
3-Bit Synchronous Down Counter



Previous state			Next state					
Q_3	Q_2	Q_1	Q^*_3	Q^*_2	Q^*_1	T_3	T_2	T_1
0	0	0	1	1	1	1	1	1
0	0	1	0	0	0	0	0	1
0	1	0	0	0	1	0	1	1
0	1	1	0	1	0	0	0	1
1	0	0	0	1	1	1	1	1
1	0	1	1	0	0	0	0	1
1	1	0	1	0	1	0	1	1
1	1	1	1	1	0	0	0	1

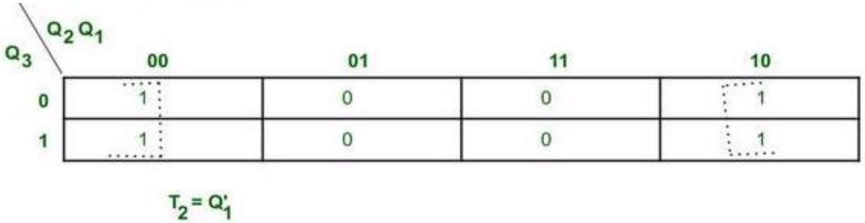
Here T = 1, then there is output state(next state changes from previous state) changes i.e Q changes from 0 to 1 or 1 to 0 T= 0 then, there is no state output state changes i.e Q remains same

4. Find simplified equation using k map –



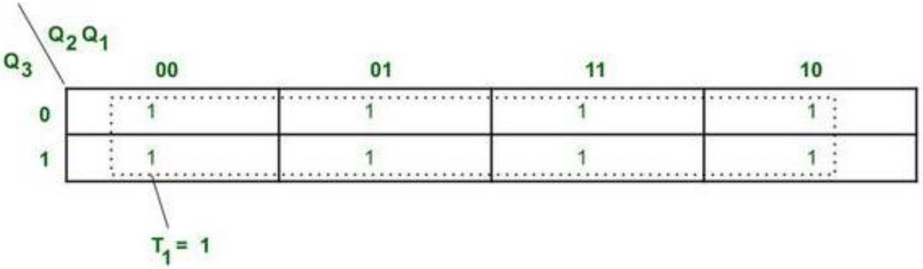
$Q_3 \backslash Q_2Q_1$	00	01	11	10
0	1	0	0	0
1	1	0	0	0

$T_3 = Q'_2Q'_1$



$Q_3 \backslash Q_2Q_1$	00	01	11	10
0	1	0	0	1
1	1	0	0	1

$T_2 = Q'_1$



$Q_3 \backslash Q_2Q_1$	00	01	11	10
0	1	1	1	1
1	1	1	1	1

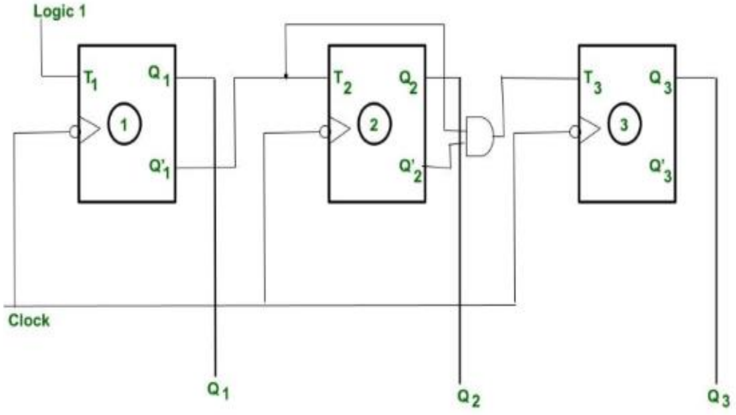
$T_1 = 1$

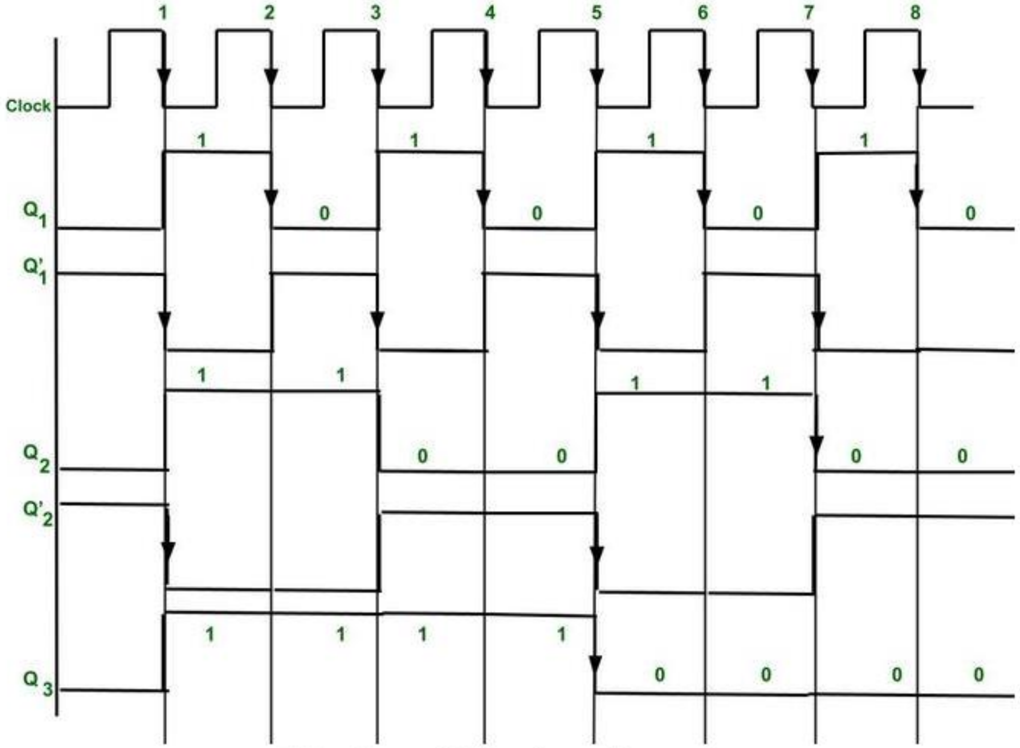
3-Bit Synchronous Down Counter

5. Create circuit diagram –

The clock is provided to every Flip flop at same instant of time.

The toggle(T) input is provided to every Flip flop according to the simplified equation of K map.

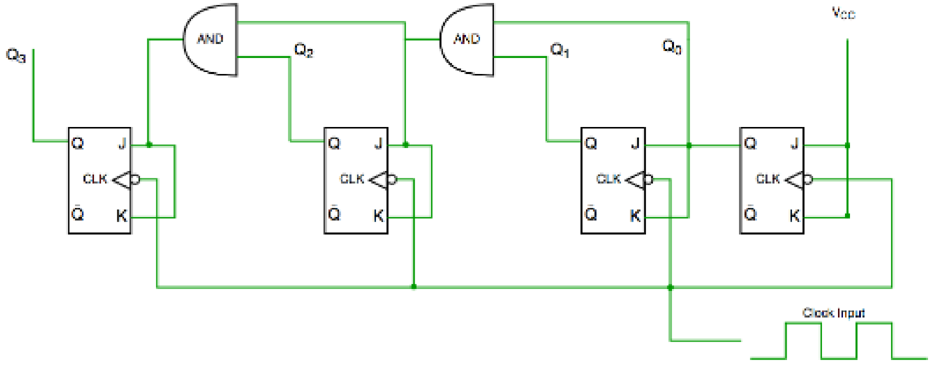


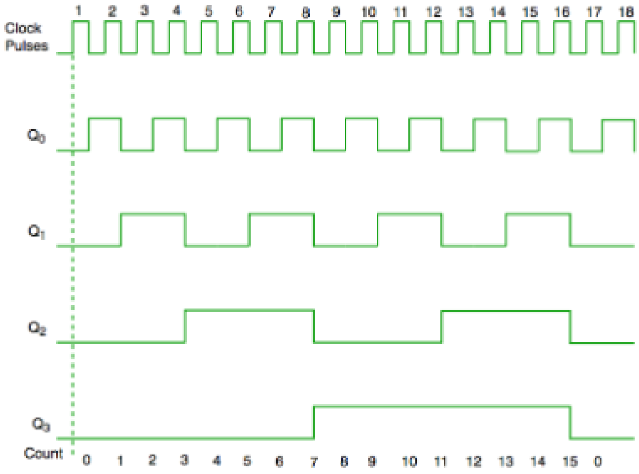


Timing diagram of 3 bit synchronous Down counter.

Source: https://www.geeksforgeeks.org/3-bit-synchronous-down-counter/

4-Bit Synchronous Counter





Timing diagram synchronous counter

From circuit diagram we see that Q0 bit gives response to each falling edge of clock while Q1 is dependent on Q0, Q2 is dependent on Q1 and Q0 , Q3 is dependent on Q2,Q1 and Q0.

Source: https://www.geeksforgeeks.org/counters-in-digital-logic/