Electronics

Module 3: Digital Systems

Flipflops

- They are fundamental building blocks of digital system.
- It is a binary storage device which stores a single bit of data (binary bit either 0 or 1).
- It has two stable states High and Low (0 / 1).
- It has property to remain in one state until it is directed by I/P signal to change that state.
- It is a kind of bi-stable multivibrator.
- It is a sequential circuit and thereby is capable of serving as one bit of memory, bit 1 or 0.
- In the sequential logical circuit, flip flop is the basic storage element.

Uses of Flip-Flop

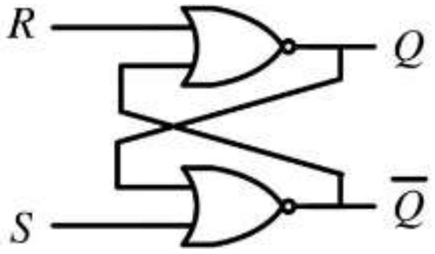
- In memory circuits
- For Logic control Devices
- In Counter Devices
- For Register Device

Types of Flip-flops:

- SR Flip flop
- JK Flip flop
- D flip flop
- T Flip flop

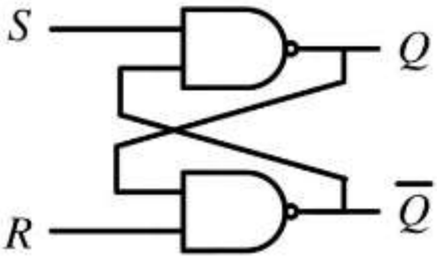
SR Flip Flop

Using NOR gate



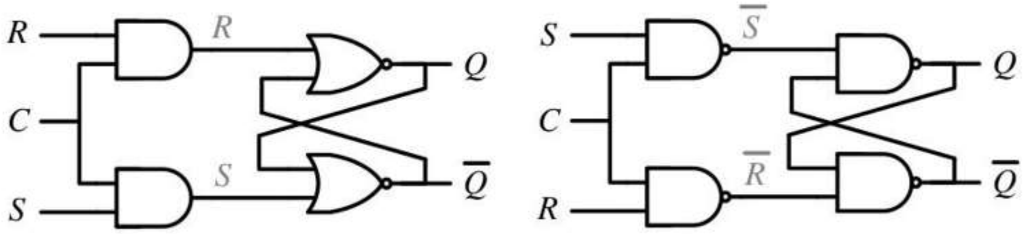
S	R	Q	State
0	0	Last state	No change
0	1	0	Reset
1	0	1	Set
1	1	Forbidden	Invalid

Using NAND gate



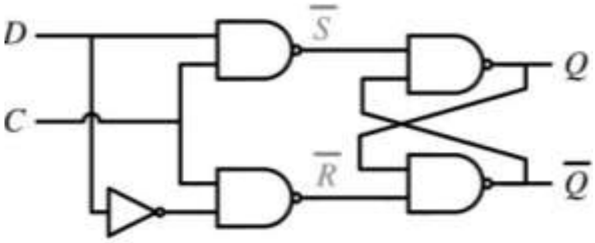
S	R	Q	State
0	0	Forbidden	Invalid
0	1	0	Reset
1	0	1	Set
1	1	Last state	No change

Clocked SR Flip Flop



Truth Table is same as SR flip flop. When clock is high it works as SR flip flop otherwise it will stay in last state only.

D Flip Flop

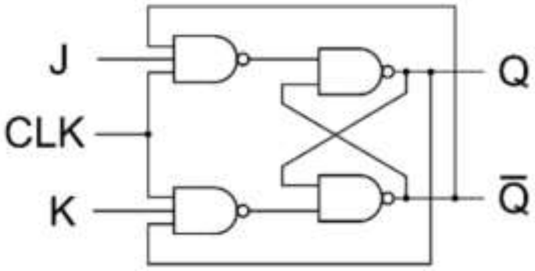


CLK	D	Output
0	X	Last state
1	0	0
1	1	1

- A D type (data or delay flip flop) has a single data input in addition to clock input.
- It can be constructed from SR flip flop or JK flip flop by an addition of an inverter.
- Output changes at clock edge, and if input changes at other times, output will be unaffected.
- The change of state of output is dependent on the rising edge of the clock.
- Output is same as the input and can only change at rising edge of the clock.

JK Flip Flop

- It is used to remove the drawback of the S-R flip flop, i.e., undefined states.
- It is formed by doing modification in SR flip flop.
- The S-R flip flop is improved in order to construct the J-K flip flop.
- When S and R input is set to true, the SR flip flop gives an inaccurate result. But in the case of JK flip flop, it gives the correct output.

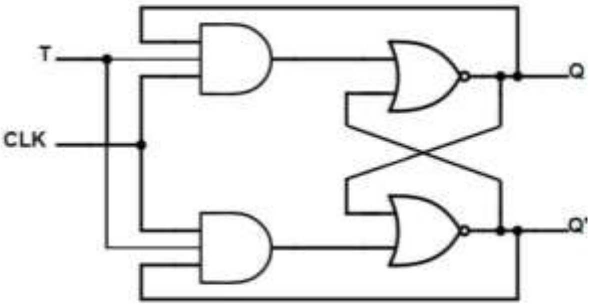


CLK	J	K	Q
↑	0	0	Last state
↑	0	1	Reset (0)
↑	1	0	Set (1)
↑	1	1	Toggle

- If J & K are different, then output Q takes the value of J at next clock edge.
- If J and K are both low, then no change occurs.
- If J and K are both high at clock edge, then the output will toggle from one state to the other.
- JK Flip-Flops can function as Set or Reset Flip-flops.

T Flip Flop

- It is like a JK flip-flop.
- These are basically single-input versions of JK flip-flops.
- This modified form of the JK flip-flop is obtained by connecting inputs J and K together.
- It has only one input along with the clock input.
- Because of their ability to complement their state-Toggle, they are named Toggle flip-flops.

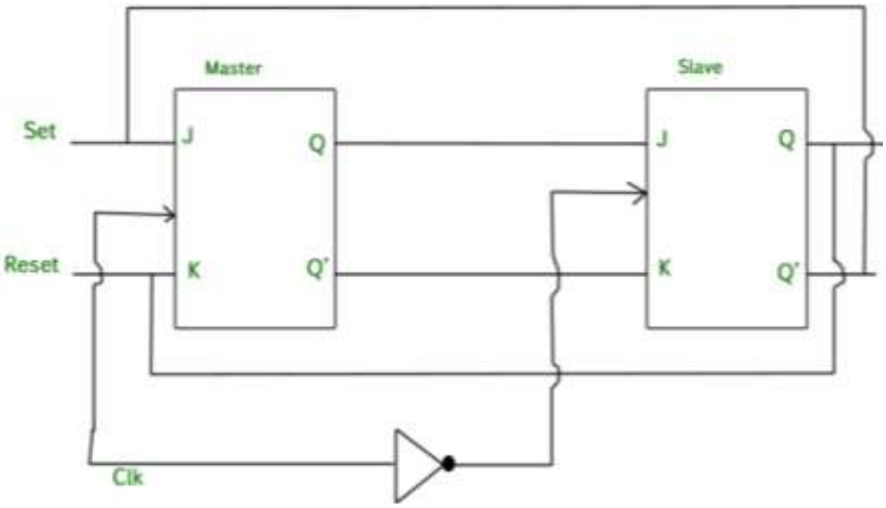


CLK	T	Q
0		
1	0	
1	1	Toggle

Master Slave JK Flip Flop

- For J-K flip-flop, if J=K=1, and if clk=1 for a long period of time, then Q output will toggle as long as CLK is high, which makes the output of the flip-flop unstable or uncertain. This problem is called race around condition in J-K flip-flop. This problem can be avoided by ensuring that clock input is at logic 1 only for a very short time. This introduced concept of Master Slave JK flip flop.

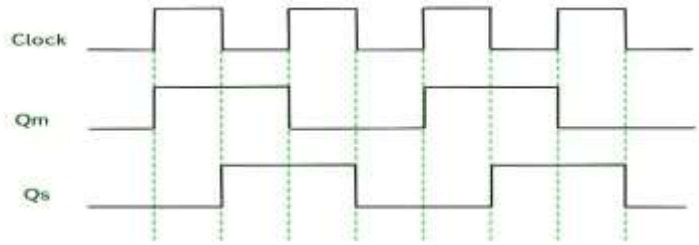
- The master-slave flip flop is constructed by combining two JK flip flops.
- These flip flops are connected in a series configuration.
- One work as master, called the master flip flop, other work as slave, called slave flip flop.
- Master-slave flip flop is designed in such a way that output of master flip flop is passed to both inputs of slave flipflop. Output of slave flip flop is passed to inputs of master flip flop.
- Inverter or NOT gate is also used for passing inverted clock pulse to slave flip flop, inverter is connected to the clock's pulse. When clock's pulse set to 0 for master then clock's pulse is set to 1 for slave, and vice versa.



Working of a master slave flip flop:

1. When the clk pulse goes to 1, slave is isolated; J and K inputs may affect state of system. Slave flip-flop is isolated until clk pulse goes to 0. When clk pulse goes back to 0, information is passed from master flip-flop to slave and output is obtained.
2. Firstly, master FF is +ve level triggered, slave FF is -ve, so master responds before slave.
3. If J=0, K=1, high Q' output of master goes to K input of slave and clk forces slave to reset, thus slave copies master.
4. If J=1, K=0,high Q output of master goes to J input of slave and negative transition of clk sets slave, copying master.
5. If J=1, K=1, it toggles on +ve transition of clk thus slave toggles on -ve transition of clk.
6. If J=0, K=0, the flip flop is disabled and Q remains unchanged.

Timing diagram of master-slave flip flop:



CLK	J	K	State
0	0	0	Last state
1	0	1	Reset
1	1	0	Set
1	1	1	Toggle

1. When Clk pulse is high, output of master is high and remains high till clk is low because state is stored.
2. Now output of master becomes low when clk pulse becomes high again and remains low until clk becomes high again.
3. Thus, toggling takes place for a clk cycle.
4. When clk pulse is high, master is operational but not slave thus output of slave remains low till clk remains high.
5. When clk is low slave becomes operational & remains high until clk again becomes low.
6. Toggling takes place during whole process since output is changing once in a cycle.

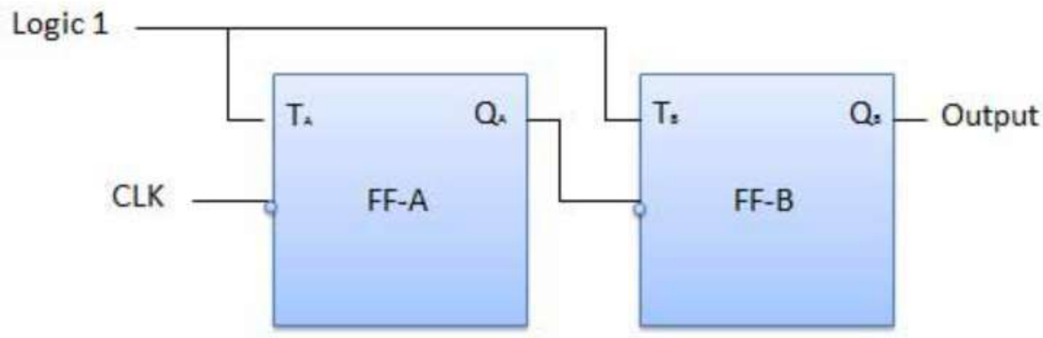
This makes Master-Slave J-K flip flop a Synchronous device as it only passes data with timing of clk signal.

Counters

- Counter is a sequential circuit which is used for a counting pulses.
- It is widest application of flip-flops.
- It is a group of flip-flops with a clock signal applied.
- It is also used for measuring frequency and time.
- It is specially designed synchronous sequential circuits in which state of counter is equal to count held in circuit by flip flops.
- Counters are of two types: Asynchronous/ripple counters and Synchronous counters.

Asynchronous / Ripple Counter

The logic diagram of a 2-bit ripple up counter.
The toggle (T) flip-flops are being used.
But we can use JK flip-flop also with J and K connected permanently to logic 1.
External clk is applied to clk input of FF-A & Q_A output is applied to clk input of next FF-B.



Operation:

1. Initially let both the FFs be in the reset state: Q_BQ_A = 00 initially

2. After 1st negative clock edge:
As soon as first negative clk edge is applied, FF-A will toggle and Q_A will be equal to 1. Q_A is connected to clk input of FF-B. Since Q_A has changed from 0 to 1, it is treated as positive clock edge by FF-B. There is no change in Q_B because FF-B is a negative edge triggered FF.
Q_BQ_A = 01 after the first clock pulse.

3. After 2nd negative clock edge:
On arrival of second negative clock edge, FF-A toggles again and Q_A = 0. Change in Q_A acts as a negative clock edge for FF-B. So, it will also toggle, and Q_B will be 1.
Q_BQ_A = 10 after the second clock pulse.

4. After 3rd negative clock edge:
On the arrival of 3rd negative clock edge, FF-A toggles again and Q_A become 1 from 0.
Since this is a positive going change, FF-B does not respond to it and remains inactive. So Q_B does not change and continues to be equal to 1.
Q_BQ_A = 11 after the third clock pulse.

5. After 4th negative clock edge:
On the arrival of 4th negative clock edge, FF-A toggles again and Q_A becomes 1 from 0.
This negative change in Q_A acts as clk pulse for FF-B. Hence it toggles to change Q_B from 1 to 0. Q_BQ_A = 00 after the fourth clock pulse.

Clock	Q_B	Q_A	State Number	Decimal Output
Initially	0	0	-	0
1st	0	1	1	1
2nd	1	0	2	2
3rd	1	1	3	3
4th	0	0	4	0

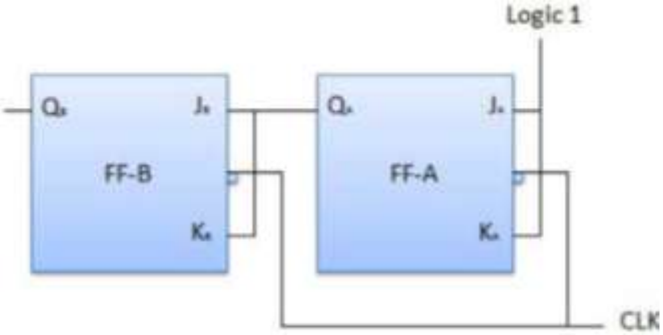
Synchronous Counters

If clk pulses are applied to all flip-flops in a counter simultaneously, then such a counter is called as synchronous counter.

2-bit synchronous up counter:

The J_A and K_A inputs of FF-A are tied to logic 1. So, FF-A will work as a toggle flip-flop. The J_B and K_B inputs are connected to Q_A.

Logical Diagram:



Operation:

1. Initially let both the FFs be in the reset state Q_BQ_A = 00 initially.

2. After 1st negative clock edge

As soon as first negative clk edge is applied, FF-A will toggle & Q_A will change from 0 to 1.
But at instant of application of negative clock edge, Q_A , $J_B = K_B = 0$.
Hence FF-B will not change its state. So Q_B will remain 0.
Q_BQ_A = 01 after the first clock pulse.

3. After 2nd negative clock edge

On the arrival of second negative clock edge, FF-A toggles again and Q_A changes from 1 to 0.
But at this instant Q_A was 1. So, $J_B = K_B = 1$ and FF-B will toggle.
Hence Q_B changes from 0 to 1.
Q_BQ_A = 10 after the second clock pulse.

4. After 3rd negative clock edge

On application of third falling clock edge, FF-A will toggle from 0 to 1 but there is no change of state for FF-B.
Q_BQ_A = 11 after third clock pulse.

5. After 4th negative clock edge

On application of next clk pulse, Q_A will change from 1 to 0 as Q_B will change from 1 to 0.
Q_BQ_A = 00 after the fourth clk pulse.

4-bit Synchronous Up Counter:

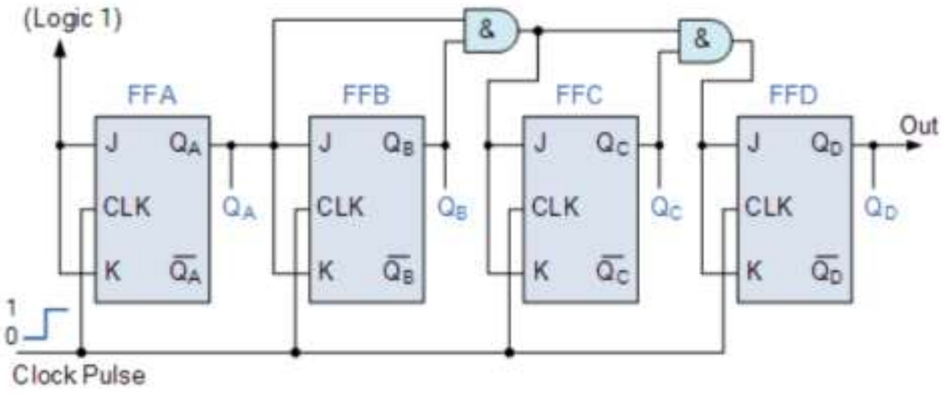
- A 4-bit Synchronous up counter start to count from 0 (0000 in binary) and increment or count upwards to 15 (1111 in binary) and then start new counting cycle by getting reset.
- Its operating frequency is much higher than same range Asynchronous counter.
- Also, there is no propagation delay in synchronous counter just because all flip-flops or counter stage is in parallel clock source and clk triggers all counters at the same time.
- The external clock signal is connected to clock input of every individual flip-flop within counter so that all of flip-flops are clocked together simultaneously (in parallel) at the same time giving a fixed time relationship.
- Changes in the output occur in “synchronisation” with the clock signal.
- Result of this synchronisation is that all individual output bits changing state at exactly same time in response to common clk signal with no ripple effect and hence no propagation delay.

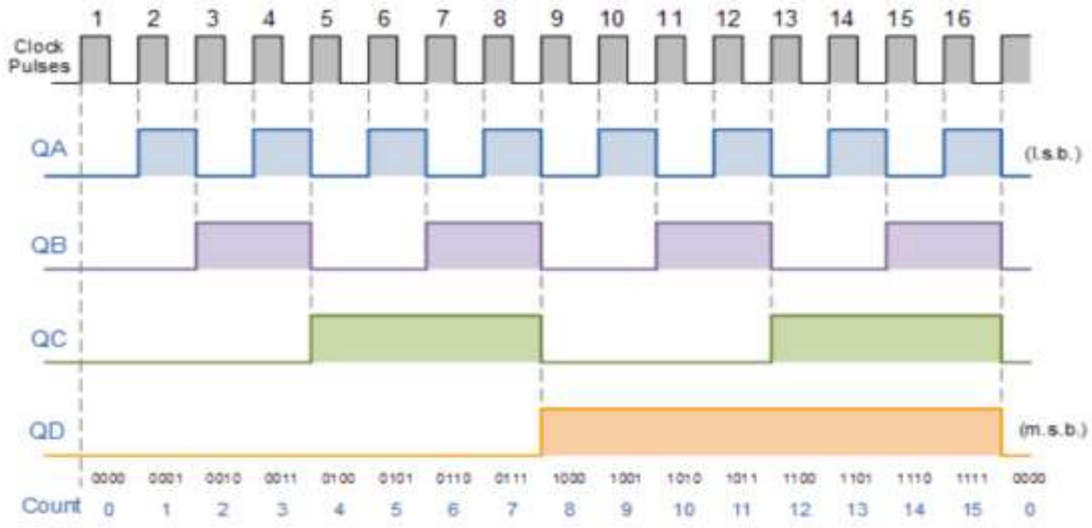
The external clock pulses are fed directly to each of the JK flipflop in the counter chain and that both the J and K inputs are all tied together in toggle mode, but only in first flip-flop, flip-flop FF-A (LSB) are they connected High, logic 1 allowing the flip-flop to toggle on every clock pulse. Then synchronous counter follows a predetermined sequence of states in response to common clock signal, advancing one state for each pulse.

The J, K inputs of flip-flop FF-B are connected directly to output Q_A of flip-flop FFA, but the J, K inputs of flip-flops FF-C and FF-D are driven from separate AND gates which are also supplied with signals from input-output of previous stage. These additional AND gates generate required logic for the JK inputs of the next stage.

If we enable each JK flip-flop to toggle based on whether or not all preceding flip-flop outputs (Q) are High we can obtain same counting sequence as with asynchronous circuit but without ripple effect, since each flip-flop in this circuit will be clocked at exactly the same time.

Then as there is no inherent propagation delay in synchronous counters, because all counter stages are triggered in parallel at the same time, maximum operating frequency of this type of frequency counter is much higher than that for a similar asynchronous counter circuit.



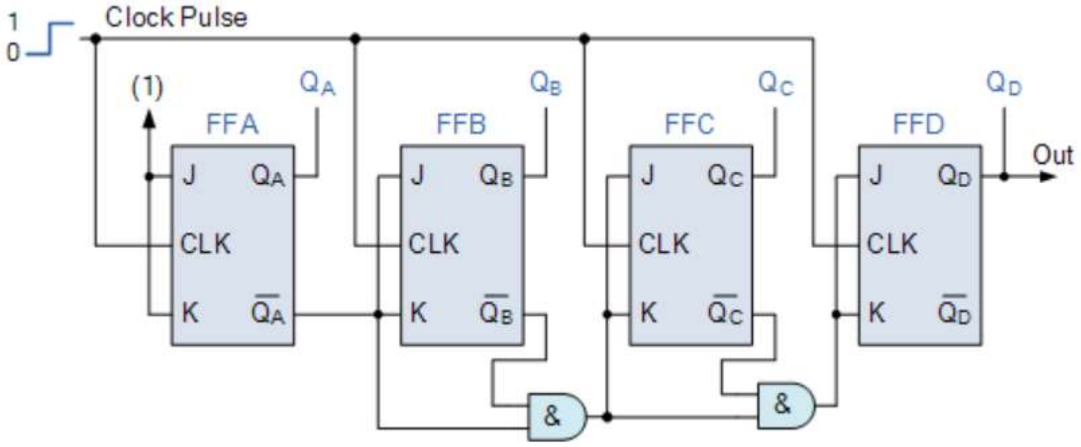


Because this 4-bit synchronous counter counts sequentially on every clock pulse resulting outputs count upwards from 0 (0000) to 15 (1111). Therefore, this type of counter is also known as a 4-bit Synchronous Up Counter.

However, we can easily construct a 4-bit Synchronous Down Counter by connecting AND gates to Q output of flip-flops as shown to produce a waveform timing diagram reverse of the above. Here counter starts with all of its outputs HIGH (1111) and it counts down on the application of each clock pulse to zero, (0000) before repeating again.

Synchronous Down Counter:

- It starts to count from 15 (1111 in binary) and decrement or count downwards to 0 or 0000 and after that it will start new counting cycle by getting reset.
- The AND Gate input is changed. First Flip-flop FFA input is same as we used in previous Synchronous up counter.
- Instead of directly feeding output of first flip-flop to next subsequent flip-flop, we are using inverted output pin which is used to give J and K input across next flip-flop FFB and also used as input pin across AND gate.
- Same as like previous circuit, two AND gates are providing necessary logic to next two Flip-flops FF-C and FF-D.



Modulus-N Counters

- Modulus Counters, or simply MOD counters, are defined based on number of states that counter will sequence through before returning back to its original value.
- For example,
  a 2-bit counter that counts from 00_2 to 11_2 in binary, that is 0 to 3 in decimal, has a modulus value of 4 (00 → 1 → 10 → 11, and return back to 00) so would therefore be called a modulo-4, or mod-4, counter. Note that it has taken four clock pulses to get from 00 to 11.
- As in this example there are only two bits, (n = 2) then the maximum number of possible output states (maximum modulus) for the counter is: $2^n = 2^2$ or 4.
- Therefore, a Mod-N counter will require N number of flip-flops connected together to count a single data bit while providing 2^n different output states, (n is the number of bits).

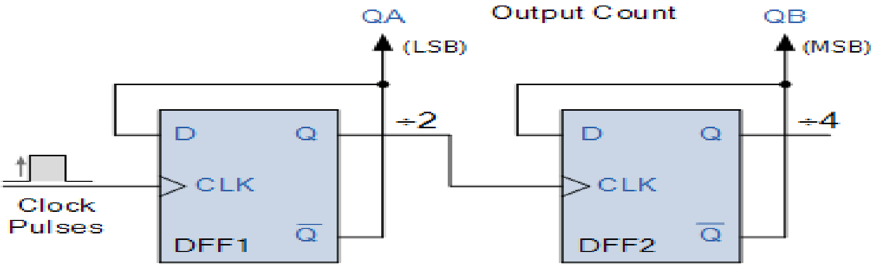
Types of Modulo-N Counter

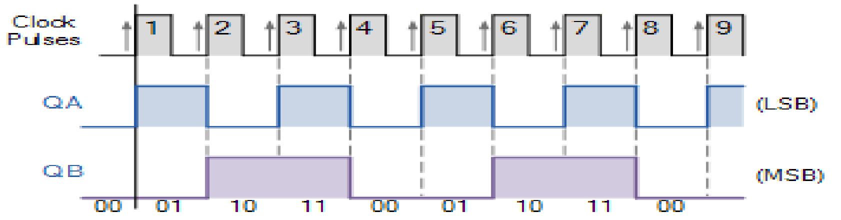
- **Modulo - 2 counter:**

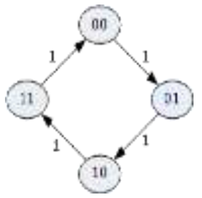
Technically as well as being a 1-bit storage device, a single flip-flop on its own could be thought of as a MOD-2 counter, as it has a single output resulting in a count of two, either 0 or 1, on application of clock signal. But a single flip-flop on its own produces a limited counting sequence, so by connecting together more flip-flops to form a chain, we can increase the counting capacity and construct a MOD counter of any value.

- **Modulo - 4 Counter:**

If a single flip-flop can be considered as MOD-2 counter, then adding a second flip-flop would give us a MOD-4 counter allowing it to count in four discrete steps. Overall effect would be to divide original clock input signal by four. Then binary sequence for this 2-bit MOD-4 counter would be: 00, 01, 10, and 11 as shown.

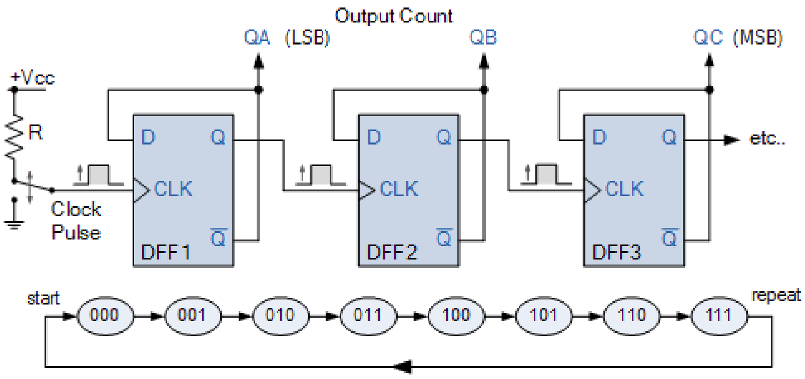




Lock pulse	Present state		Next state		State diagram
0 (start)	0	0	0	1	
1	0	1	1	0	
2	1	0	1	1	
3	1	1	0	0	
4 (repeat)	0	0	0	1	

- **Modulo – 8 Counter:**

By adding another flip-flop onto the end of a MOD-4 counter to produce a MOD-8 counter giving us a 2^3 binary sequence of counting from 000 up to 111, before resetting back to 000. A fourth flip-flop would make a MOD-16 counter and so on, in fact we could go on adding extra flip-flops for as long as we wanted



State transition diagram for a MOD-8 binary counter

- **Modulo - 5 counter:**

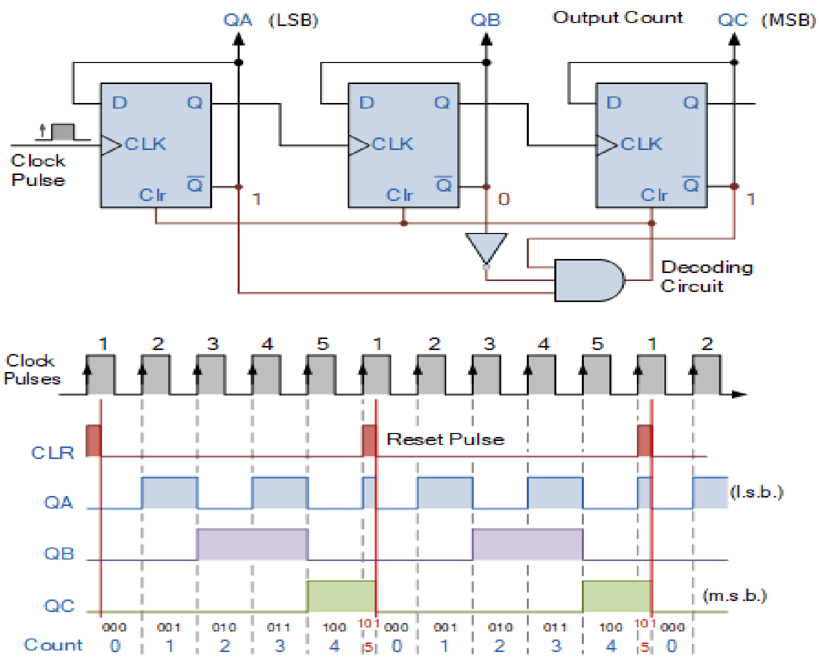
As we want to construct a MOD-5 counter, we need to modify the 3-bit (Mod-8) counter circuit so that it will reset itself back to zero after a count of 5. That is a count sequence of: 1→2→3→4→5→reset, and so on.

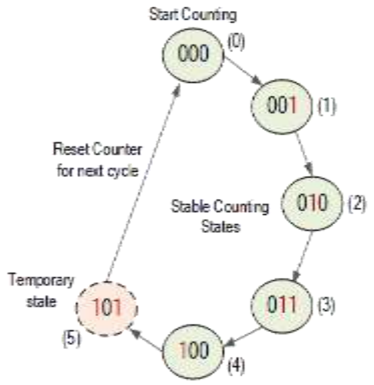
We can decode this output state of 101 (5) to give us a signal to clear (clr) the counter back to zero with the help of a 3-input AND gate and an inverter or NOT gate.

The output of the 3-input AND gate will therefore be at logic level 0 for any combinations of inputs, other than input sequence we want.

While it appears that counter counts up to 101 state, when asynchronous count sequence reaches next binary state of 101 (5), combinational logic decoding circuit will detect this 101 condition, so the AND gate will produce logic level 1 output resetting the counter back to its initial zero state. Thus, counter only remains in this 101 temporary state for only a few nanoseconds before it resets back to 000.

When the output from decoding circuit is LOW, it has no effect on the counting sequence.





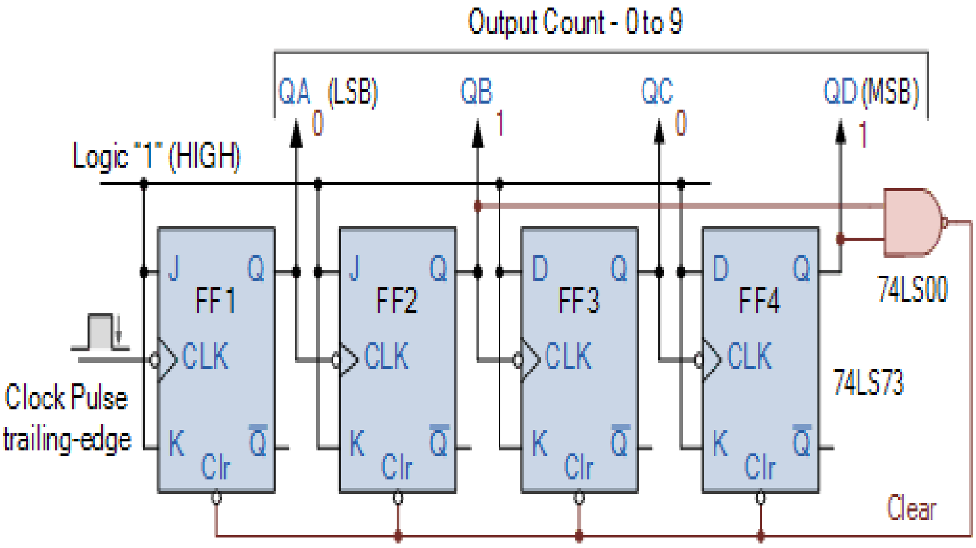
- **Modulo – 10 Counter:**

It is an example of a modulo-m counter circuit which uses external combinational circuits to produce a counter with a modulus of 10 is Decade Counter.

Decade (divide-by-10) counters, have 10 states in its counting sequence making it suitable for human interfacing where a digital display is required.

The decade counter has four outputs producing a 4-bit binary number and by using external AND and OR gates we can detect occurrence of 9th counting state to reset counter back to zero. As with other mod counters, it receives input clock pulse, one by one, and counts up from 0 to 9 repeatedly.

Once it reaches the count 9 (1001 in binary), counter goes back to 0000 instead of continuing on to 1010. The basic circuit of a decade counter can be made from JK flip-flops that switch state on the negative trailing-edge of the clock signal


Output Count - 0 to 9
QA (LSB)
QB
QC
QD (MSB)
0
1
0
1
Logic "1" (HIGH)
J
Q
FF1
CLK
K
Clr
Q̄
FF2
D
FF3
FF4
Clock Pulse
trailing-edge
74LS00
74LS73
Clear