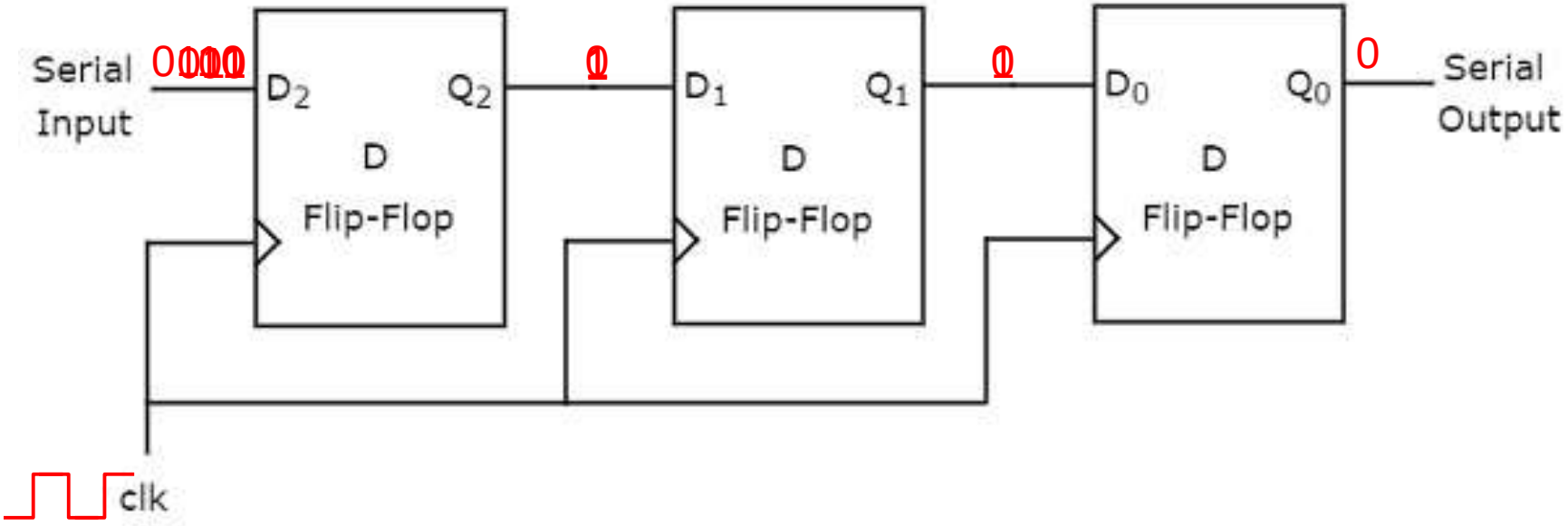
Applications of Flip-flops

By Nilesh Patidar and Shiraz Husain

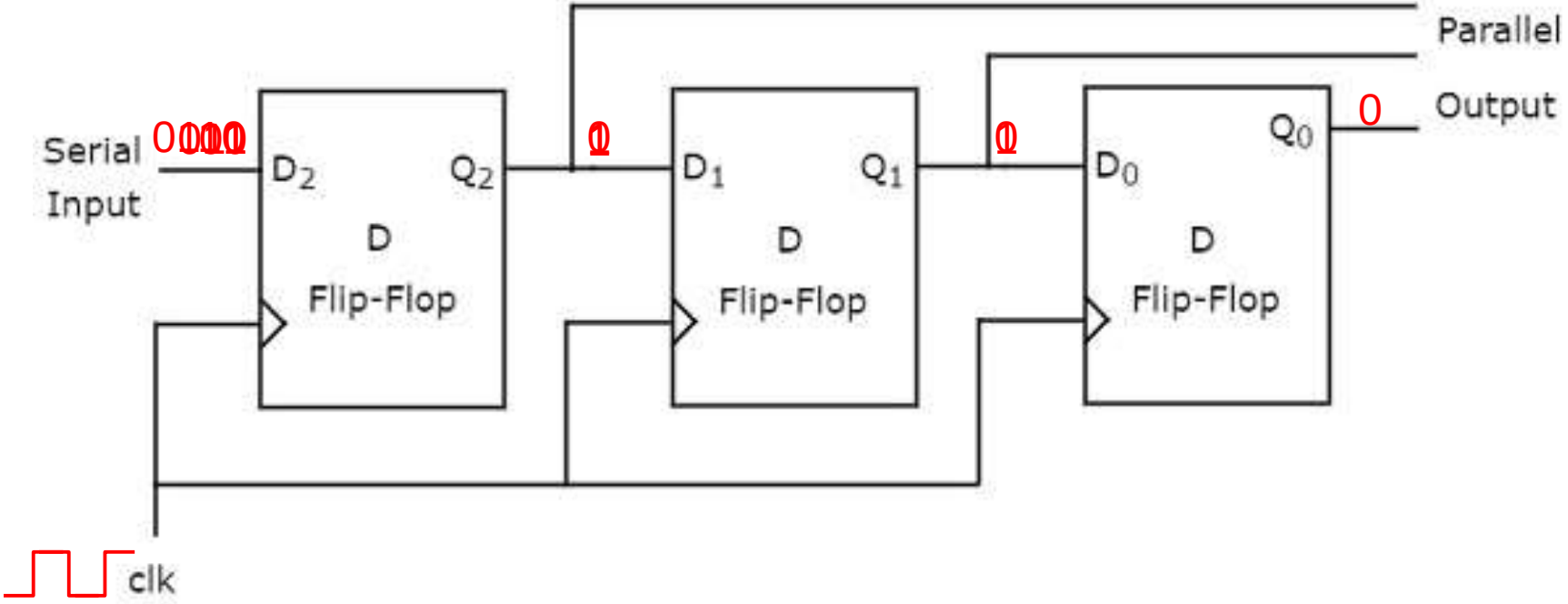
Shift Register

- One flip-flop can store one-bit of information.
- In order to store multiple bits of information, we require multiple flip-flops.
- The group of flip-flops, which are used to hold store the binary data is known as register.
- If the register is capable of shifting bits either towards right hand side or towards left hand side is known as shift register.
- There are the four types of shift registers-
  - Serial In − Serial Out shift register
  - Serial In − Parallel Out shift register
  - Parallel In − Serial Out shift register
  - Parallel In − Parallel Out shift register

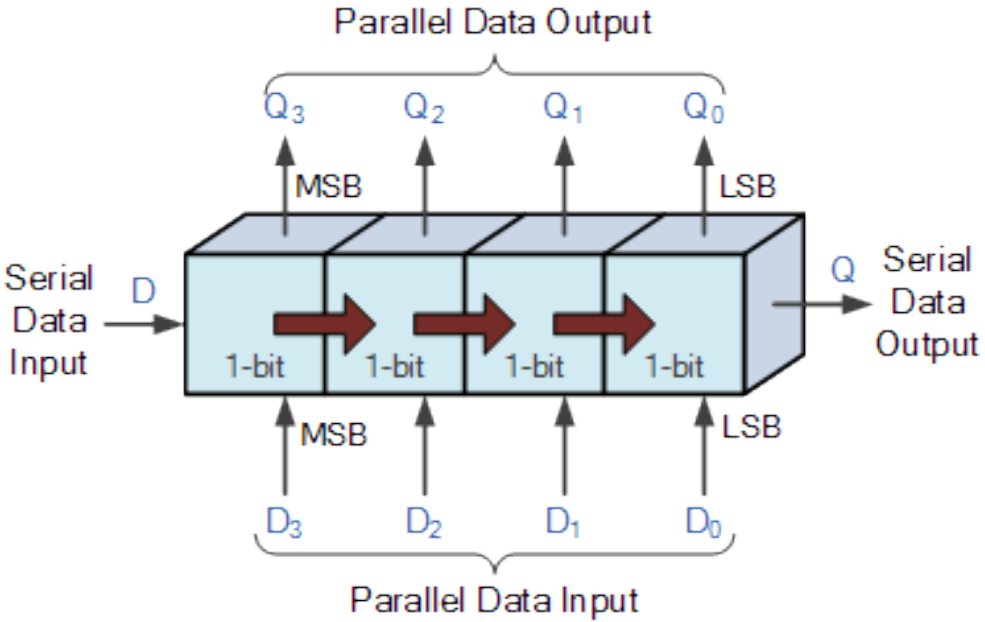
Serial In − Serial Out (SISO)



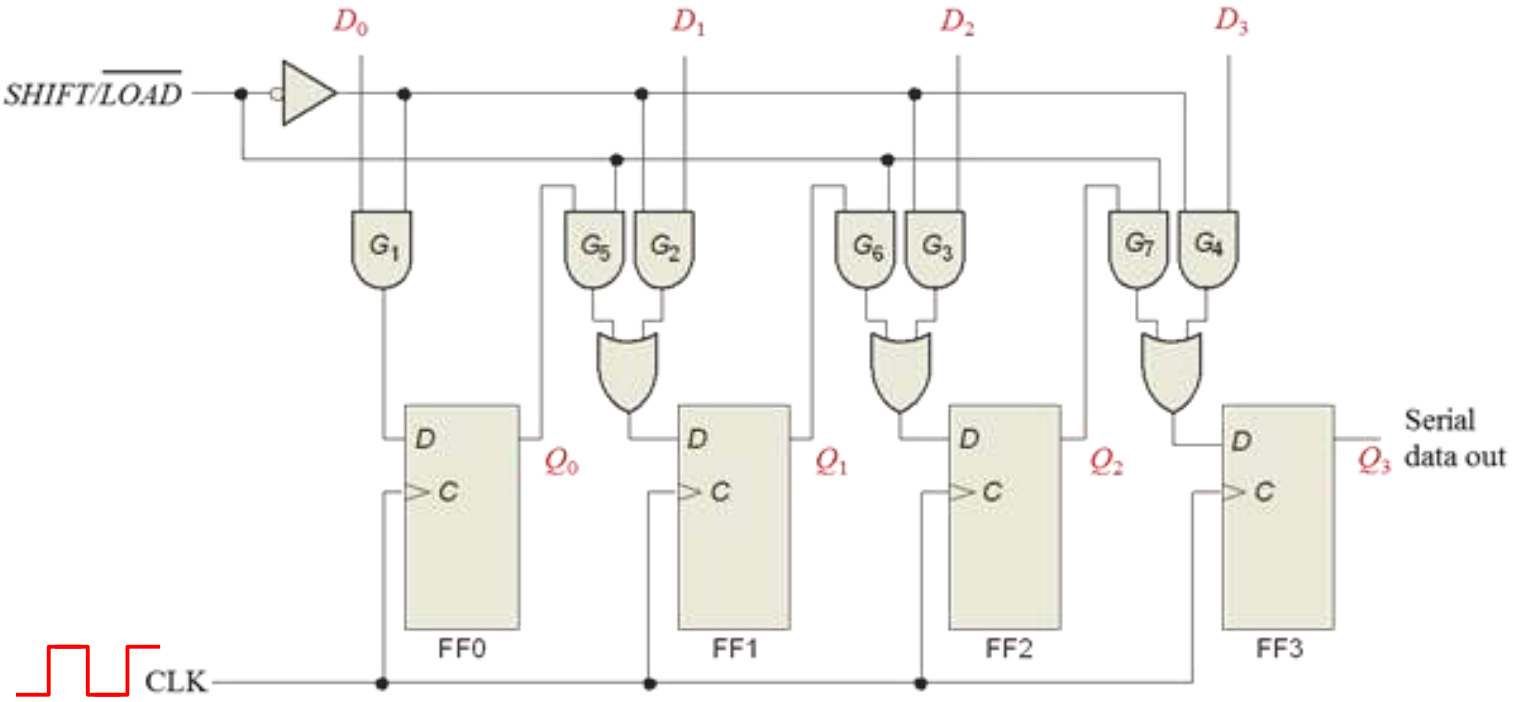
Serial In − Parallel Out (SIPO)



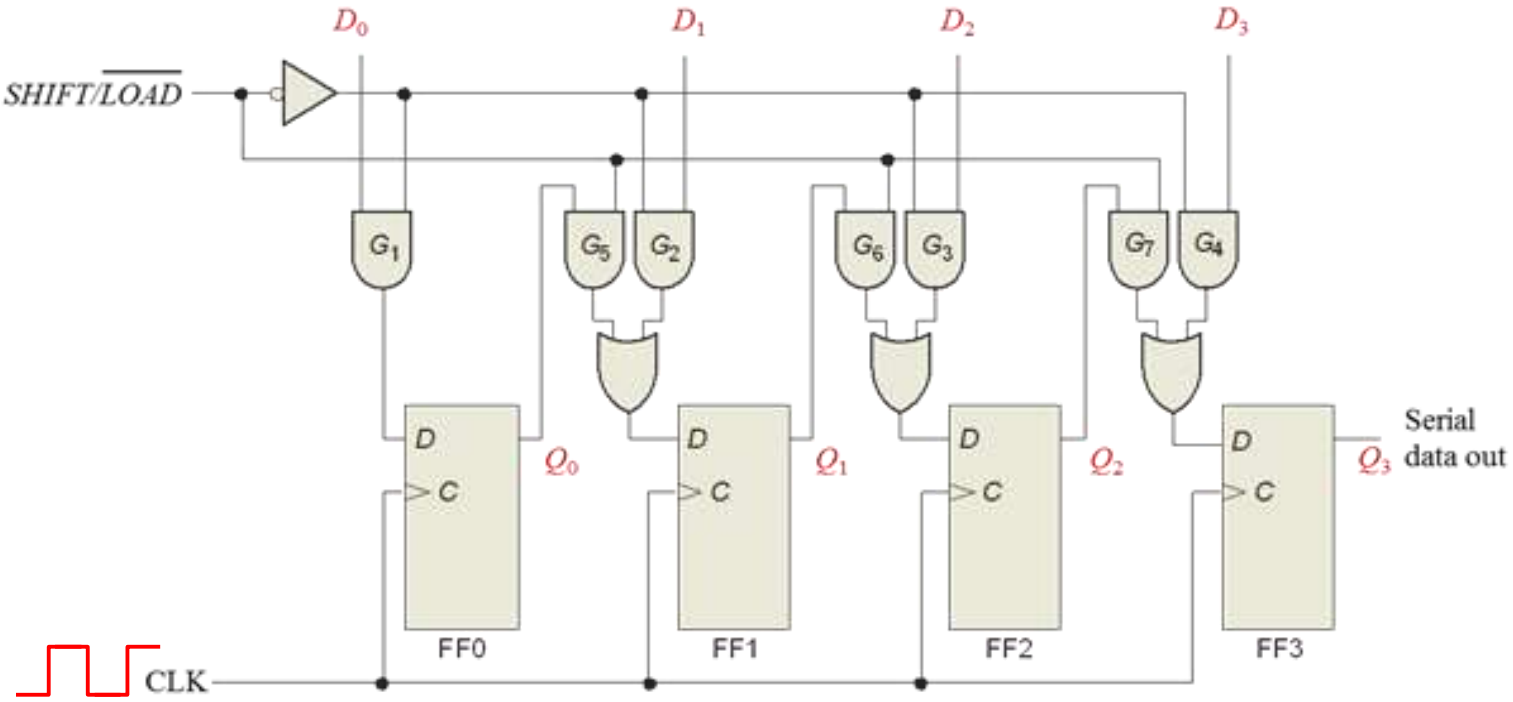
Parallel In Serial Out



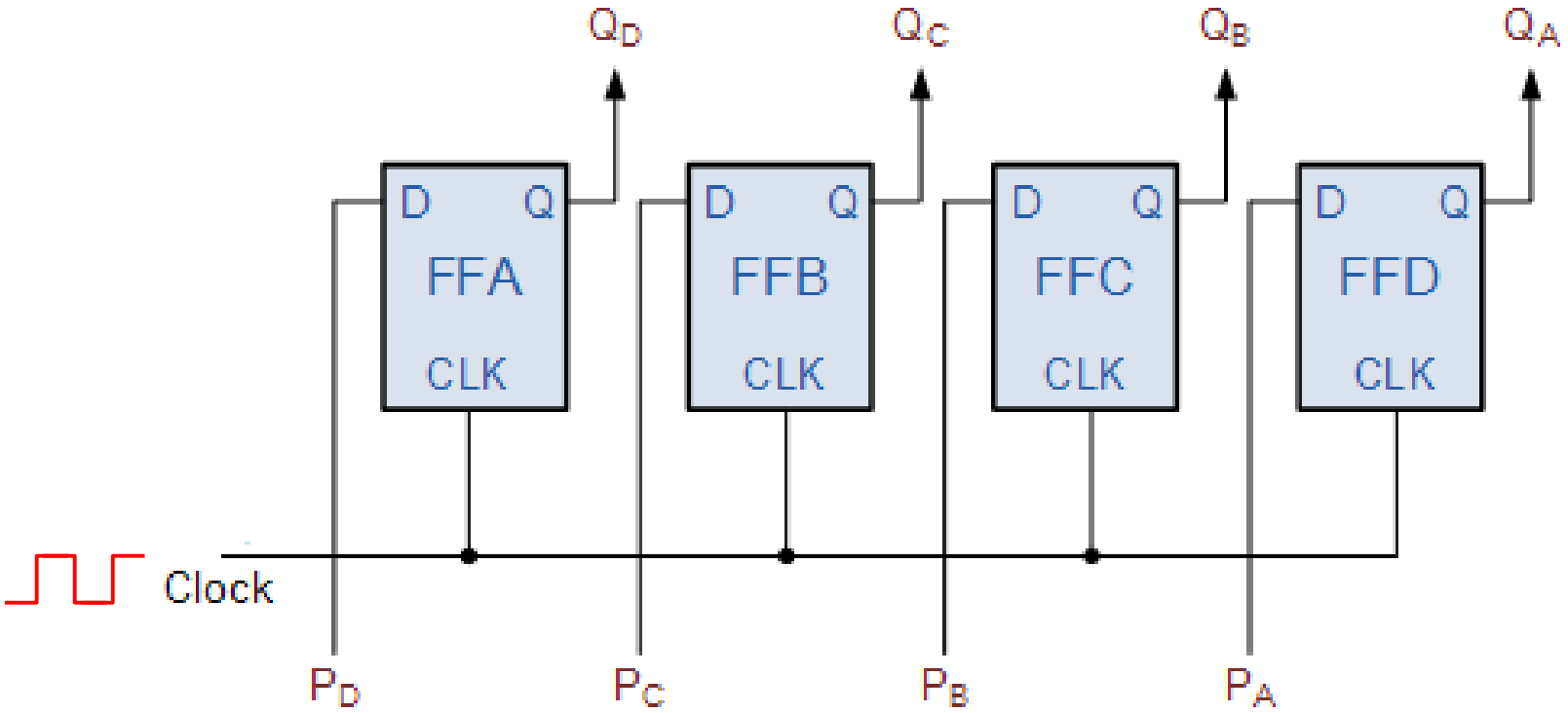
Parallel In − Serial Out (PISO)



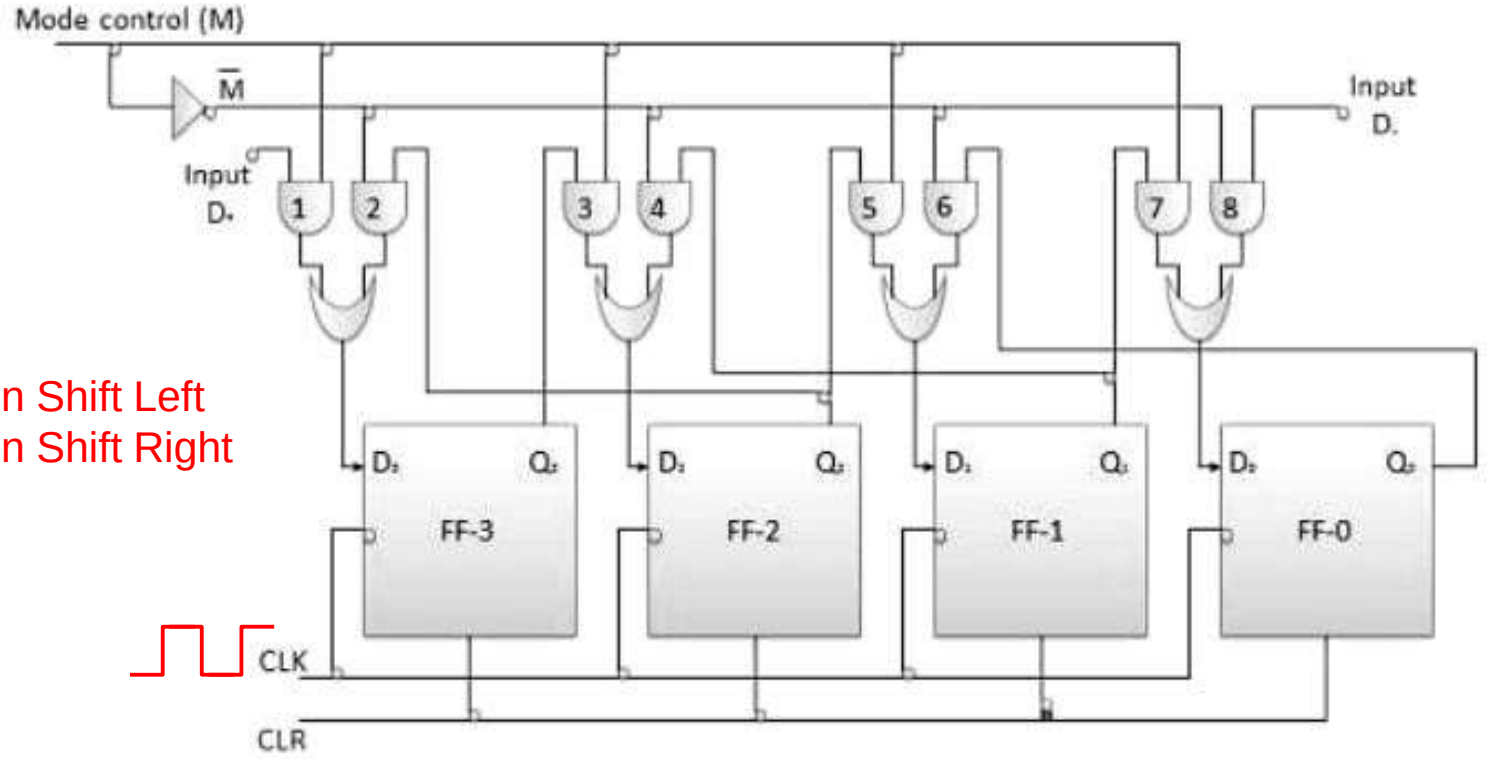
Parallel In − Serial Out (PISO)



Parallel In − Parallel Out (PIPO)



Bidirectional Shift Register



M=0 then Shift Left
M=1 then Shift Right

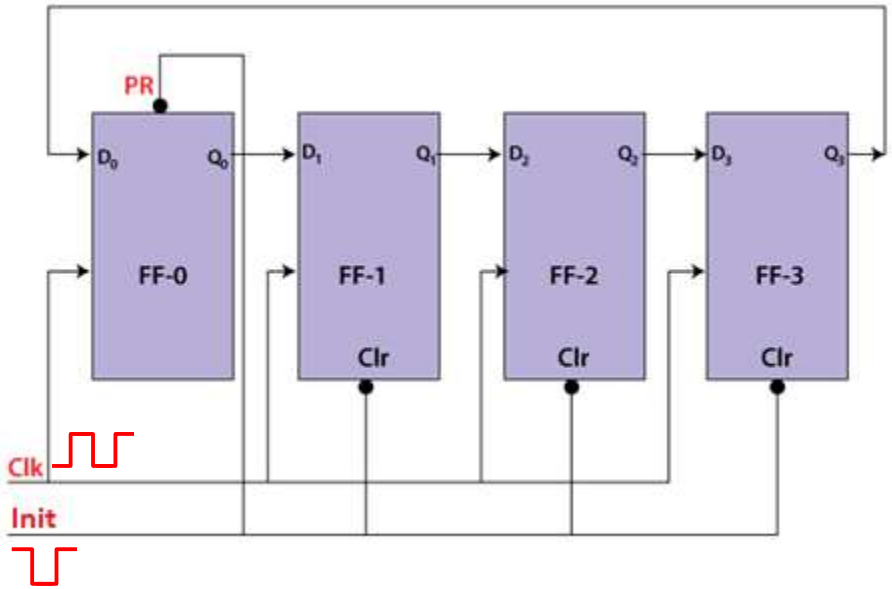
Applications of Shift Register

- Shift register is used as Parallel to serial converter, which converts the parallel data into serial data. It is utilized at the transmitter section after Analog to Digital Converter ADC block.
- Shift register is used as Serial to parallel converter, which converts the serial data into parallel data. It is utilized at the receiver section before Digital to Analog Converter DAC block.
- Shift register along with some additional gates generate the sequence of zeros and ones. Hence, it is used as sequence generator.
- Shift registers are also used as counters.

Ring Counter

- No. of states in Ring counter = No. of flip-flop used

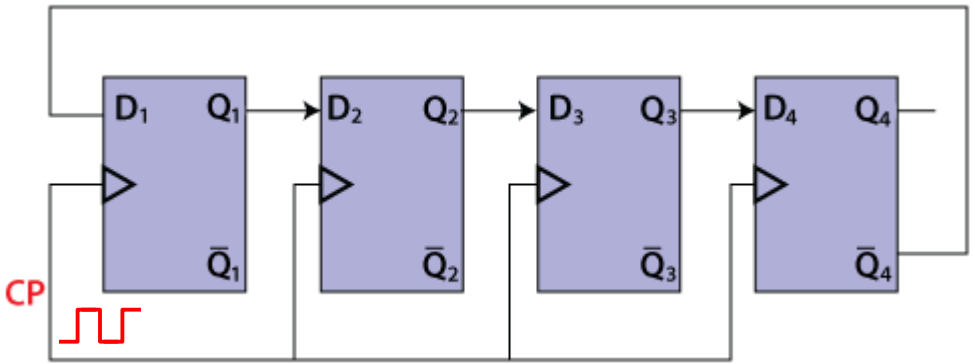
Init	Clk	Q_0	Q_1	Q_2	Q_3
L	X	1	0	0	0
H	↑	0	1	0	0
H	↑	0	0	1	0
H	↑	0	0	0	1
H	↑	1	0	0	0



Johnson Ring Counter

- No. of states in counter = 2*No. of flip-flop used

Clk	Q_1	Q_2	Q_3	Q_4
X	0	0	0	0
↑	1	0	0	0
↑	1	1	0	0
↑	1	1	1	0
↑	1	1	1	1
↑	0	1	1	1
↑	0	0	1	1
↑	0	0	0	1
↑	0	0	0	0

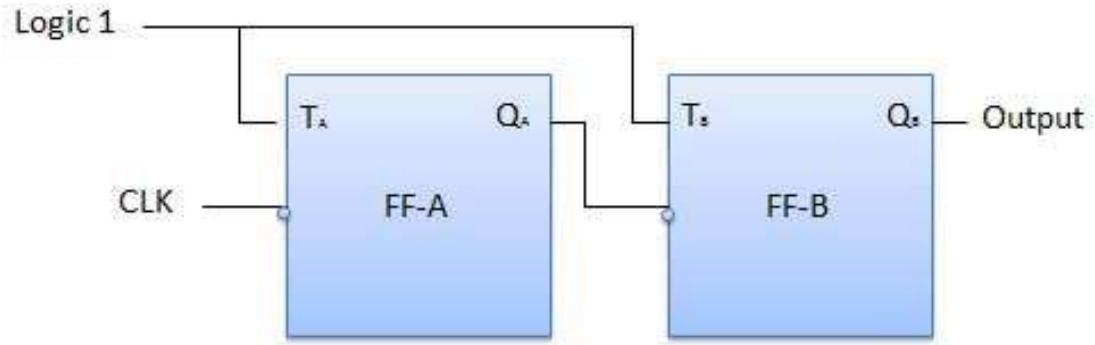


Binary Counters

- An ‘N’ bit binary counter consists of ‘N’ flip-flops and it can count the numbers up to $2^N - 1$ (2^N States)
- If the counter counts from 0 to $2^N - 1$, then it is called as binary **up counter**.
- Similarly, if the counter counts down from $2^N - 1$ to 0, then it is called as binary **down counter**.
- There are two types of counters based on the flip-flops that are connected in synchronous or not.
  - Asynchronous counters
  - Synchronous counters

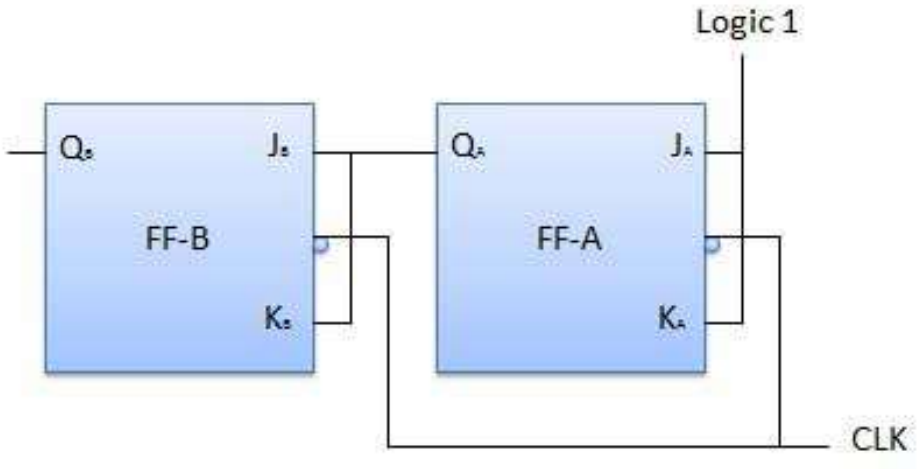
Asynchronous or Ripple Counters

- Asynchronous counters are those whose output is free from the clock signal.
- The first flip-flop is triggered by the clock signal and others successive flip-flops are driven by output of previous flip flops in asynchronous counters.



Synchronous Counters

- Synchronous counters are sometimes called parallel counters as the clock is fed in parallel to all flip-flops.
- The output of all flip-flops are triggered by a single source of clock.



Synchronous v/s Asynchronous

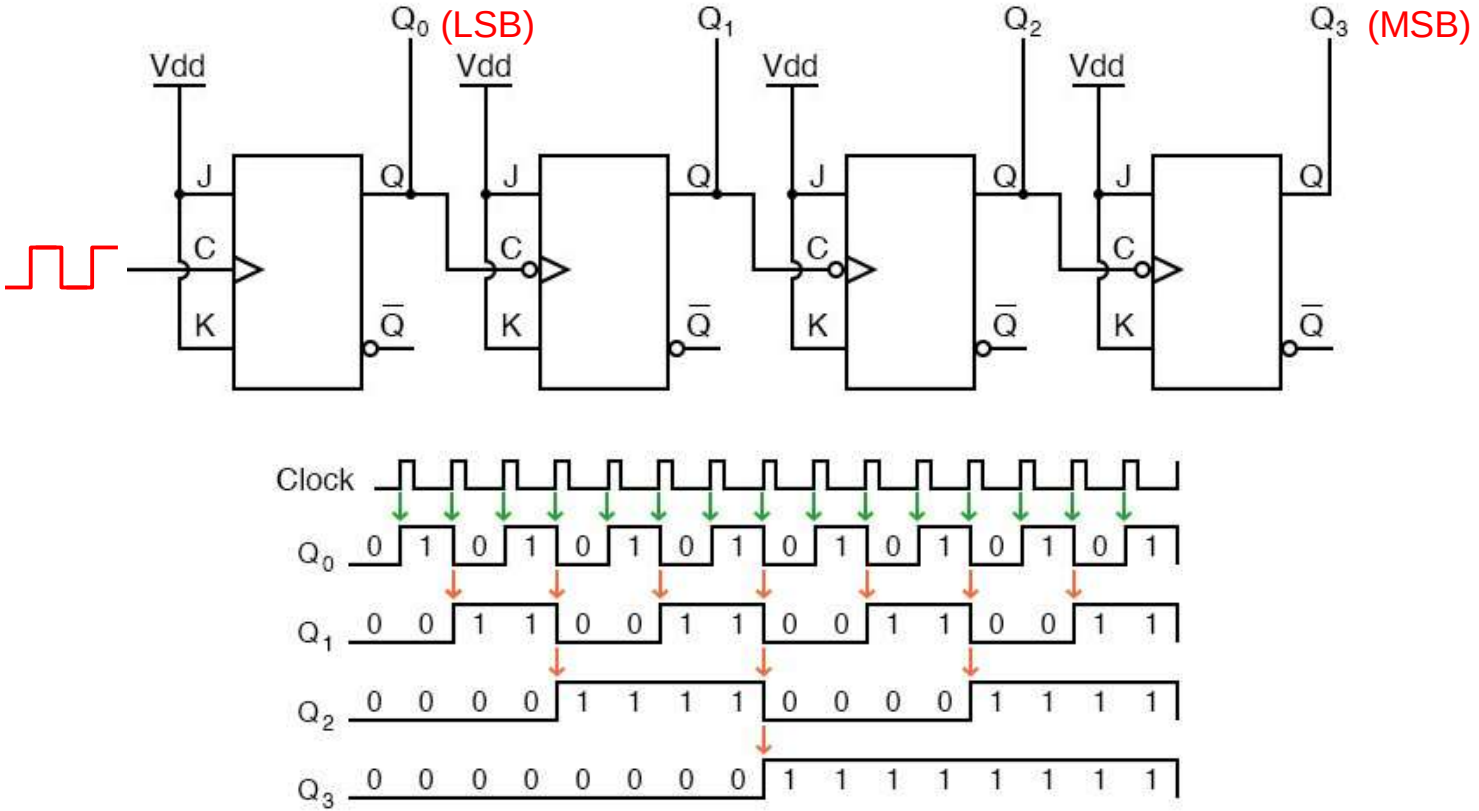
Synchronous

- All flip flops are triggered with same clock simultaneously.
- It is faster in operation
- It does not produce any decoding errors.
- It is also called Parallel Counter.
- The designing as well implementation are complex due to increasing the number of states.
- It will operate in any desired count sequence
- Examples are: Ring counter, Johnson counter.
- Propagation delay is less.

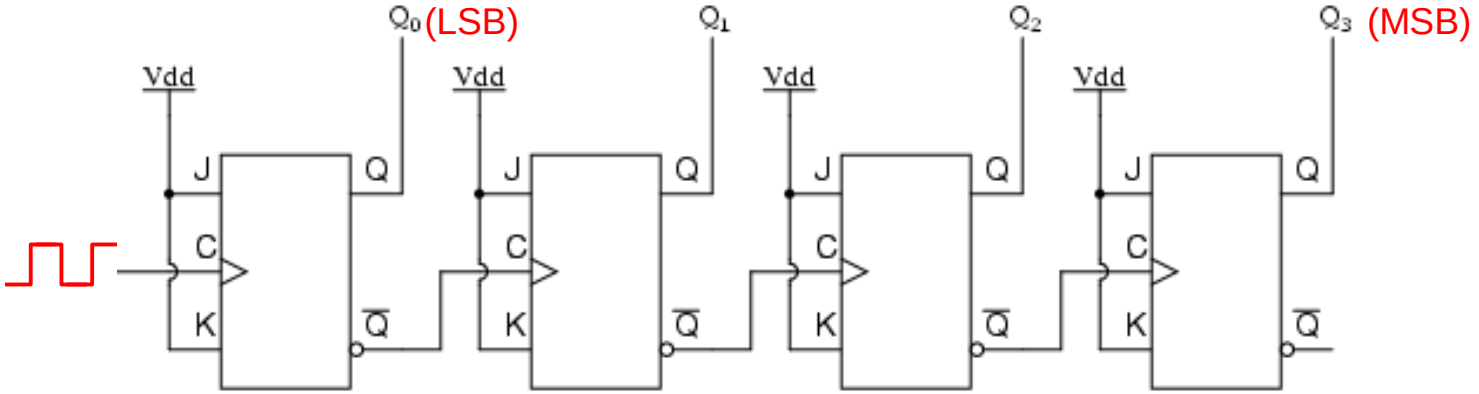
Asynchronous

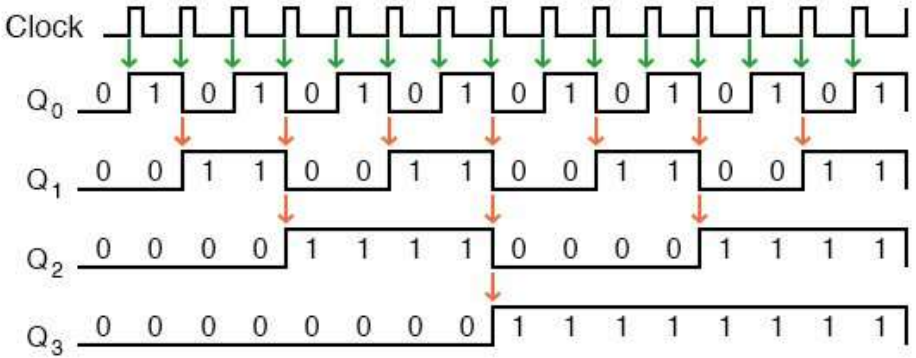
- Different flip flops are triggered with different clock, not simultaneously.
- It is slower in operation.
- It produces decoding error.
- It is also called Serial Counter.
- The designing as well as implementation is very easy.
- It will operate only in fixed count sequence (UP/DOWN).
- Examples are: Ripple UP counter, Ripple DOWN counter.
- High propagation delay.

Asynchronous Up Counters (4-bit)

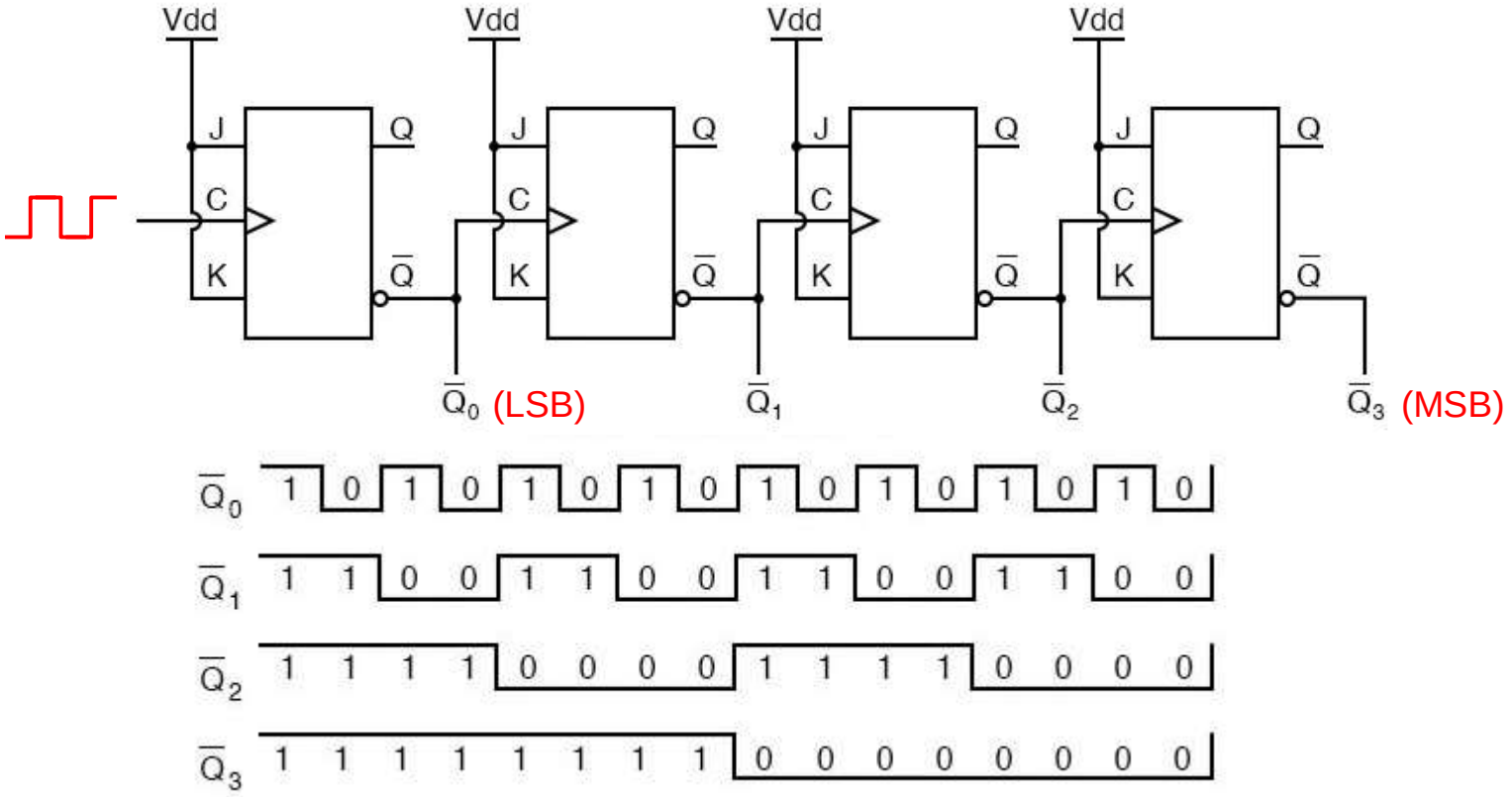


Asynchronous Up Counters (Alternate Configuration)

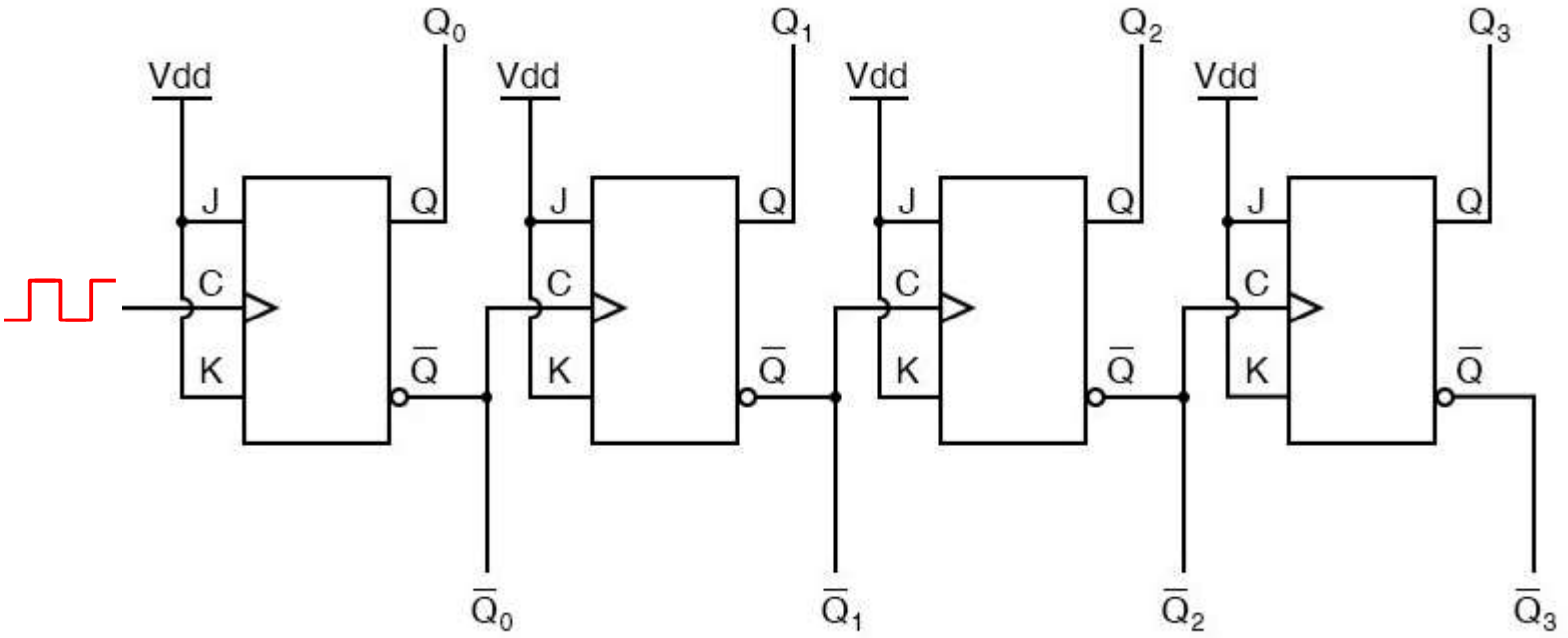




Asynchronous Down Counters (4-bit)



Asynchronous Up-Down Counters (4-bit)



MOD-*N* Counters

- The number of flip flops used in a ripple counter is depends on the number of states of counter.
- The number of output states of counter is called “Modulus” or “MOD” of the counter. (ex: Mod 4, Mod 2 etc).
- For example, if we have 2 flip flops, the maximum number of outputs of the counter is 4 i.e. 2^2. So it is called as “MOD-4 counter” or “Modulus 4 counter”.
- The 2-bit ripple counter is called as MOD-4 counter and 3-bit ripple counter is called as MOD-8 counter. So, an ***n-bit*** ripple counter is called as **modulo-*N*** counter. Where, MOD number $\boldsymbol{N = 2^n}$

Ripple counter with MOD-N ($N \leq 2^n$)

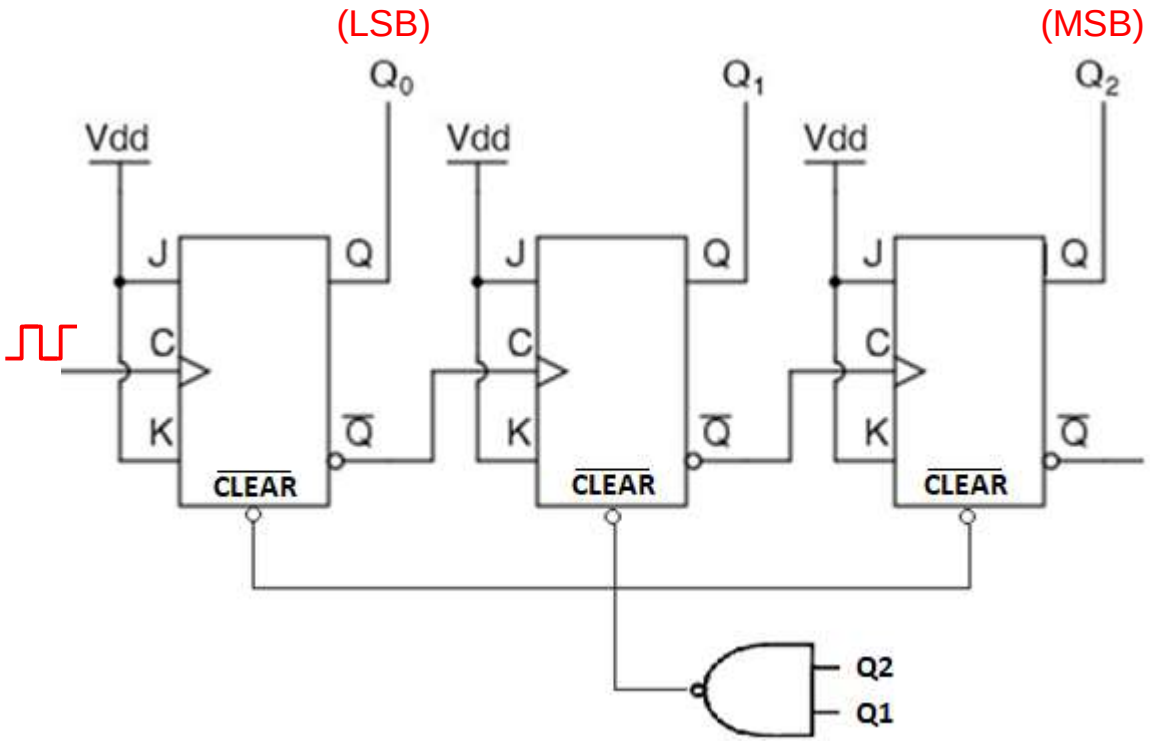
To construct any Mod-N counter, following methods can be used-

1. Find the number of flip-flops (n) required for the desired MOD number (N) using the equation-

$$2^{n-1} \leq N \leq 2^n$$

2. Connect all n flip-flops as a ripple counter.
3. Find the binary number for N.
4. Connect all the flip-flop outputs, for which Q = 1 when the count is N as inputs to NAND gate.
5. Connect the NAND Gates output to clear input of each flip-flop.

Mod-6 counter (Ripple)



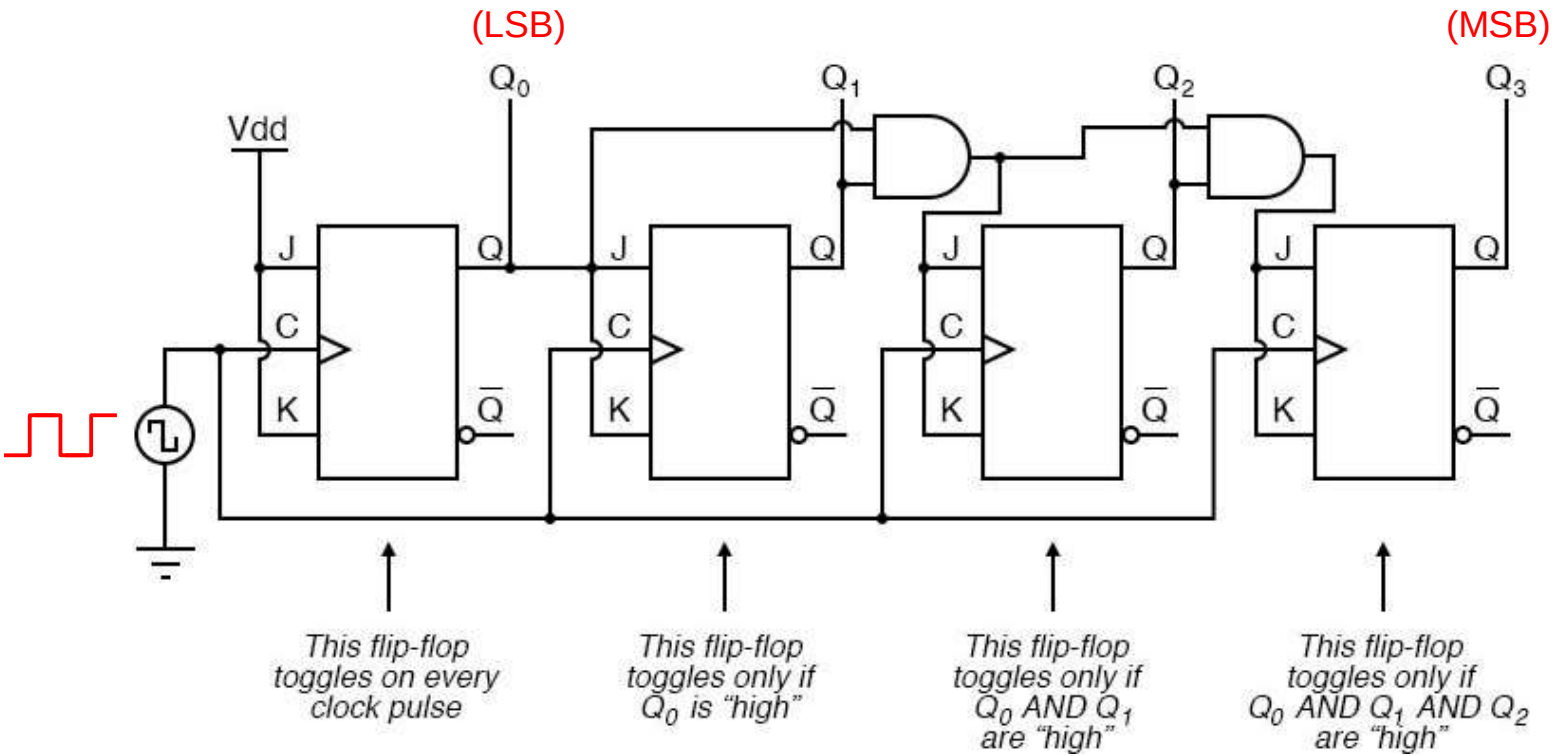
Here N=6
So number flip-flops will be
$2^{n-1} \leq N \leq 2^n$ (so, n=3)

For N=6

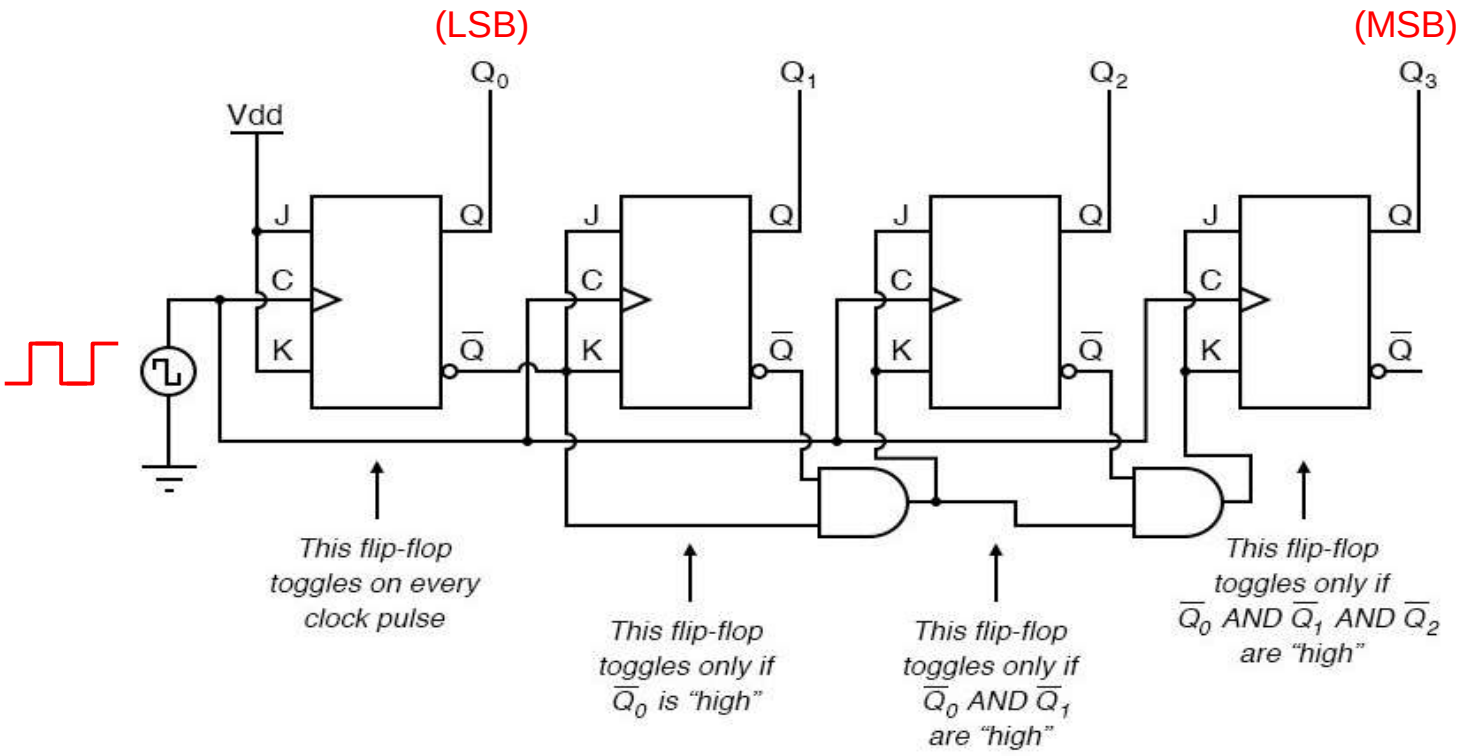
Q2	Q1	Q0	
1	1	0	(Binary of 6)

So Q1 and Q2 are inputs of NAND Gate because Q1=Q2=1

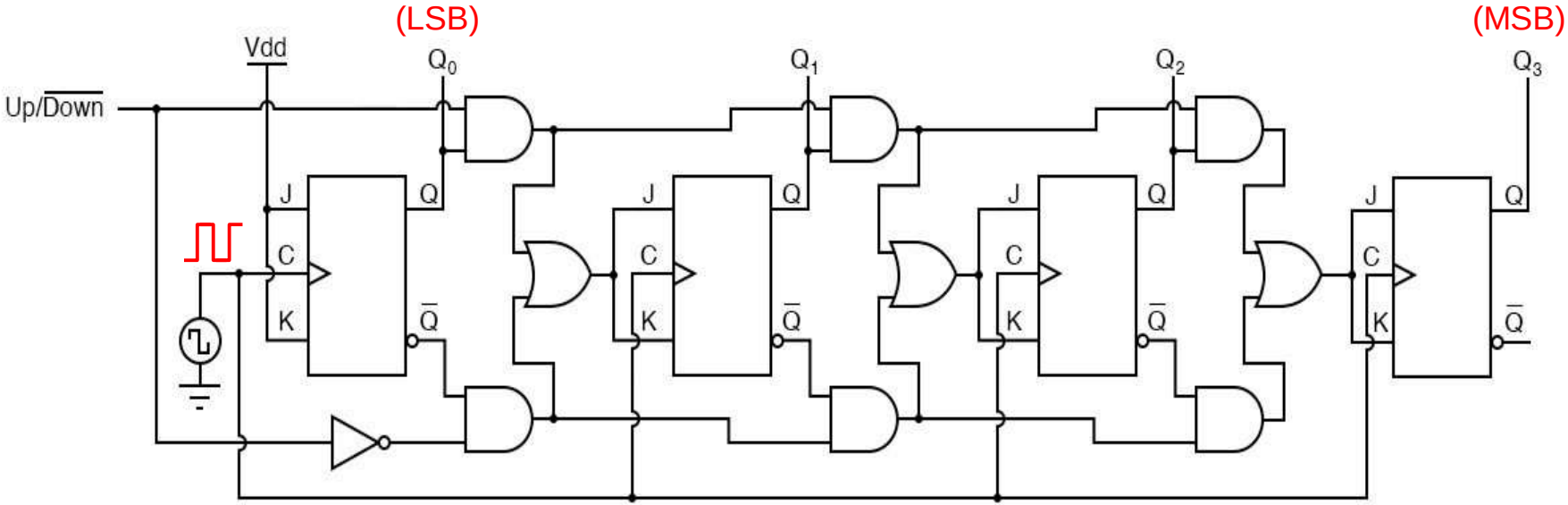
Synchronous Up Counters (4-bit)



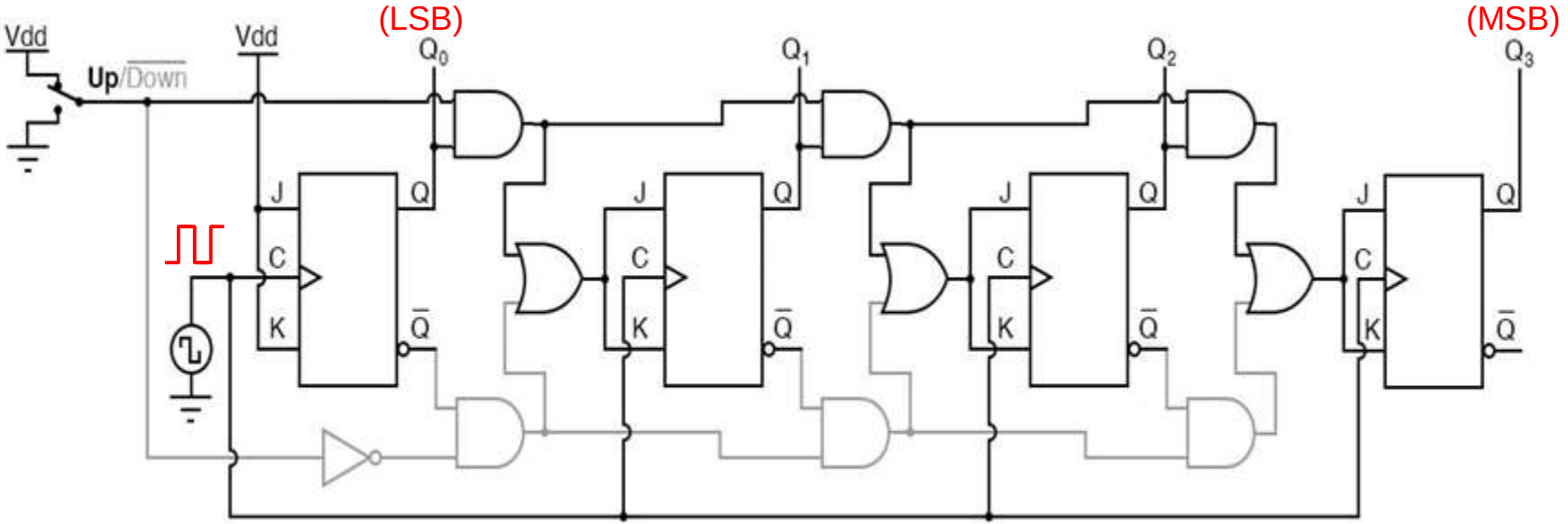
Synchronous Down Counters (4-bit)



Synchronous Up-Down Counters (4-bit)

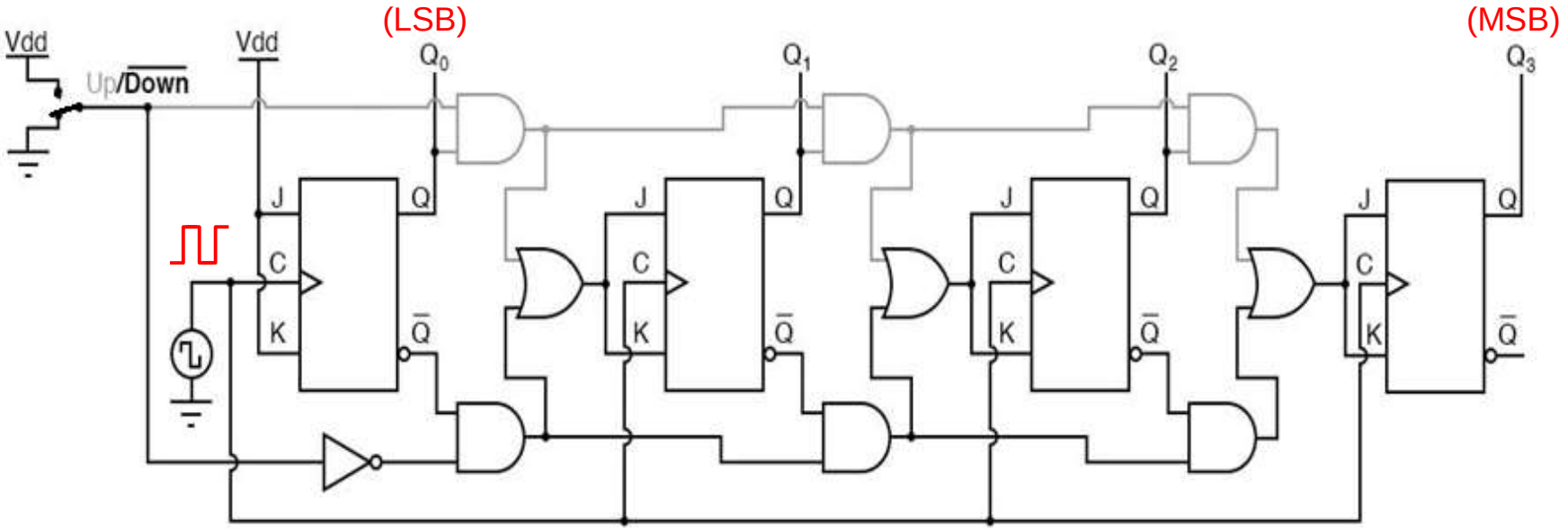


Synchronous Up-Down Counters (4-bit)



When it is acting as Up counter

Synchronous Up-Down Counters (4-bit)



When it is acting as Down counter

Design of Synchronous Counter

1. Determine the number (n) of FFs needed to support the counting sequence's highest number.

   $$2n - 1 \geq \text{Highest number}$$

2. Build a State Transition Diagram. Be sure to include all states.
3. Build a State Table & Excitation Table.
4. Simplify expressions for all inputs for each F/F on K-Maps.
5. Implement the Synchronous Counter/State Machine Circuit.
6. Draw the Timing Diagram (If Needed).

Decade Counter (Synchronous)

- Determine the number (n) of FFs needed to support the counting sequence's highest number. (Count from 0 to 9)

$$2^n - 1 \geq \text{Highest number}$$

$$2^n - 1 \geq 9$$

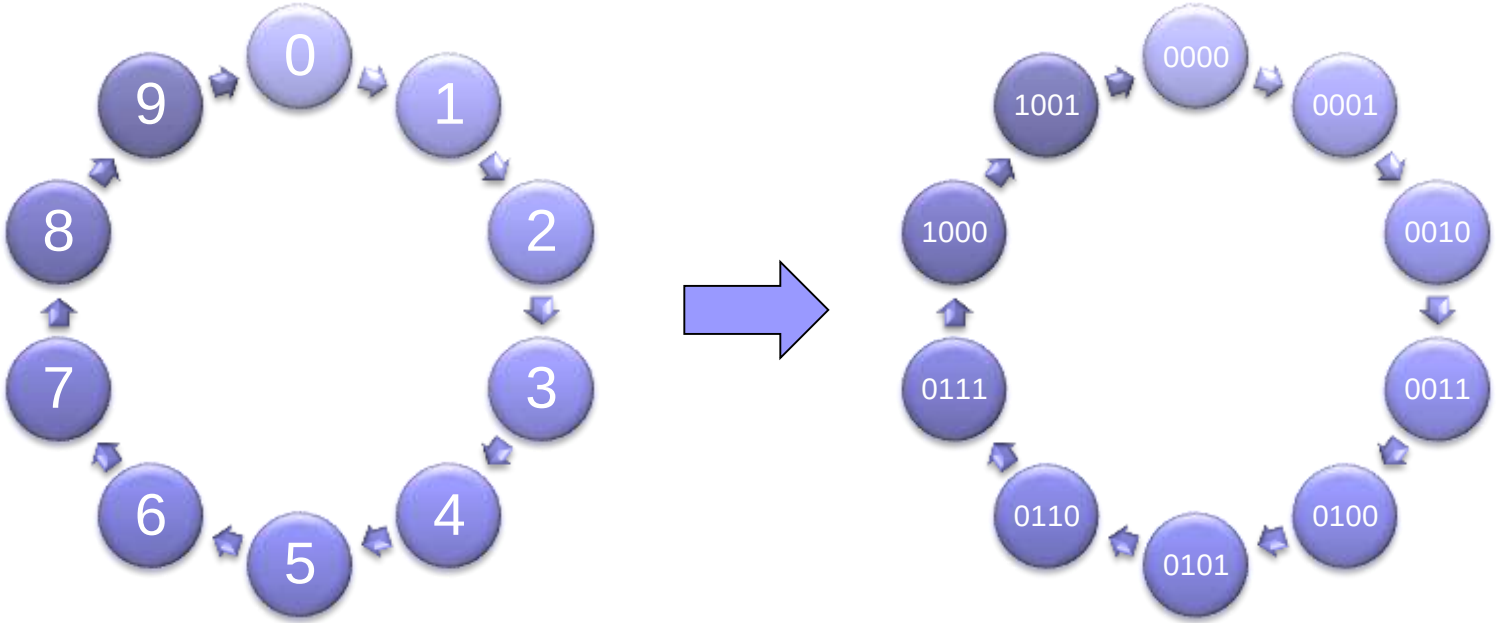
$$2^n \geq 10$$

So, $n = 4$

- It means, there are 4 FFs are required to design Decade Counter (MOD-10 counter that counts from 0 to 9)

Decade Counter (Synchronous)

- Build a State Transition Diagram



Decade Counter

PS	NS	SR-F	PS	NS	JK-FF	
Q_t	Q_{t+1}	S	Q_t	Q_{t+1}	J	K
0	0	0	0	0	0	X
0	1	1	0	1	1	X
1	0	0	1	0	X	1
1	1	X	1	1	X	0

- State Table & Excitation Table

Present State $Q_3\,Q_2\,Q_1\,Q_0$	Next State $Q_3\,Q_2\,Q_1\,Q_0$	Excitation Inputs (For JK FF)							
		J_3	K_3	J_2	K_2	J_1	K_1	J_0	K_0
0 0 0 0	0 0 0 1	0	X	0	X	0	X	1	X
0 0 0 1	0 0 1 0	0	X	0	X	1	X	X	1
0 0 1 0	0 0 1 1	0	X	0	X	X	0	1	X
0 0 1 1	0 1 0 0	0	X	1	X	X	1	X	1
0 1 0 0	0 1 0 1	0	X	X	0	0	X	1	X
0 1 0 1	0 1 1 0	0	X	X	0	1	X	X	1
0 1 1 0	0 1 1 1	0	X	X	0	X	0	1	X
0 1 1 1	1 0 0 0	1	X	X	1	X	1	X	1
1 0 0 0	1 0 0 1	X	0	0	X	0	X	1	X
1 0 0 1	0 0 0 0	X	1	0	X	0	X	X	1

Decade Counter

- Simplify expressions for all inputs for each F/F on K-Maps.

J_3 : Q_3Q_2 \ Q_1Q_0	$Q_1'Q_0'$ (00)	$Q_1'Q_0$ (01)	Q_1Q_0 (11)	Q_1Q_0' (10)
$Q_3'Q_2'$ (00)	0 0	0 1	0 3	0 2
$Q_3'Q_2$ (01)	0 4	0 5	1 7	0 6
Q_3Q_2 (11)	X 12	X 13	X 15	X 14
Q_3Q_2' (10)	X 8	X 9	X 11	X 10

$J_3 = Q_2Q_1Q_0$

K_3 : Q_3Q_2 \ Q_1Q_0	$Q_1'Q_0'$ (00)	$Q_1'Q_0$ (01)	Q_1Q_0 (11)	Q_1Q_0' (10)
$Q_3'Q_2'$ (00)	X 0	X 1	X 3	X 2
$Q_3'Q_2$ (01)	X 4	X 5	X 7	X 6
Q_3Q_2 (11)	X 12	X 13	X 15	X 14
Q_3Q_2' (10)	0 8	1 9	X 11	X 10

$K_3 = Q_0$

Decade Counter

- Simplify expressions for all inputs for each F/F on K-Maps.

J_2 : Q_3Q_2 \ Q_1Q_0	$Q_1'Q_0'$ (00)	$Q_1'Q_0$ (01)	Q_1Q_0 (11)	Q_1Q_0' (10)
$Q_3'Q_2'$ (00)	0 0	0 1	1 3	0 2
$Q_3'Q_2$ (01)	X 4	X 5	X 7	X 6
Q_3Q_2 (11)	X 12	X 13	X 15	X 14
Q_3Q_2' (10)	0 8	0 9	X 11	X 10

$J_2 = Q_1Q_0$

K_2 : Q_3Q_2 \ Q_1Q_0	$Q_1'Q_0'$ (00)	$Q_1'Q_0$ (01)	Q_1Q_0 (11)	Q_1Q_0' (10)
$Q_3'Q_2'$ (00)	X 0	X 1	X 3	X 2
$Q_3'Q_2$ (01)	0 4	0 5	1 7	0 6
Q_3Q_2 (11)	X 12	X 13	X 15	X 14
Q_3Q_2' (10)	X 8	X 9	X 11	X 10

$K_2 = Q_1Q_0$

Decade Counter

- Simplify expressions for all inputs for each F/F on K-Maps.

J_1 : Q_3Q_2 \ Q_1Q_0	$Q_1'Q_0'$ (00)	$Q_1'Q_0$ (01)	Q_1Q_0 (11)	Q_1Q_0' (10)
$Q_3'Q_2'$ (00)	0 0	1 1	X 3	X 2
$Q_3'Q_2$ (01)	0 4	1 5	X 7	X 6
Q_3Q_2 (11)	X 12	X 13	X 15	X 14
Q_3Q_2' (10)	0 8	0 9	X 11	X 10

$J_1 = Q_3'Q_0$

K_1 : Q_3Q_2 \ Q_1Q_0	$Q_1'Q_0'$ (00)	$Q_1'Q_0$ (01)	Q_1Q_0 (11)	Q_1Q_0' (10)
$Q_3'Q_2'$ (00)	X 0	X 1	1 3	0 2
$Q_3'Q_2$ (01)	X 4	X 5	1 7	0 6
Q_3Q_2 (11)	X 12	X 13	X 15	X 14
Q_3Q_2' (10)	X 8	X 9	X 11	X 10

$K_1 = Q_0$

Decade Counter

- Simplify expressions for all inputs for each F/F on K-Maps.

J_0 / Q_3Q_2 \ Q_1Q_0	$Q_1'Q_0'$ (00)	$Q_1'Q_0$ (01)	Q_1Q_0 (11)	Q_1Q_0' (10)
$Q_3'Q_2'$ (00)	1 0	X 1	X 3	1 2
$Q_3'Q_2$ (01)	1 4	X 5	X 7	1 6
Q_3Q_2 (11)	X 12	X 13	X 15	X 14
Q_3Q_2' (10)	1 8	X 9	X 11	X 10

$J_0 = 1$

K_0 / Q_3Q_2 \ Q_1Q_0	$Q_1'Q_0'$ (00)	$Q_1'Q_0$ (01)	Q_1Q_0 (11)	Q_1Q_0' (10)
$Q_3'Q_2'$ (00)	X 0	1 1	1 3	X 2
$Q_3'Q_2$ (01)	X 4	1 5	1 7	X 6
Q_3Q_2 (11)	X 12	X 13	X 15	X 14
Q_3Q_2' (10)	X 8	1 9	X 11	X 10

$K_0 = 1$

Decade Counter (Synchronous)

- Implement the Synchronous Counter/State Machine Circuit.

$J_3 = Q_2Q_1Q_0$

$K_3 = Q_0$

$J_2 = Q_1Q_0$

$K_2 = Q_1Q_0$

$J_1 = Q_3'Q_0$

$K_1 = Q_0$

$J_0 = 1$

$K_0 = 1$

Applications of counters

- Frequency counters
- Digital clock
- Time measurement
- A to D converter
- Frequency divider circuits
- Digital triangular wave generator
- Ramp signal generator

Thank You