Counters

15/10/13

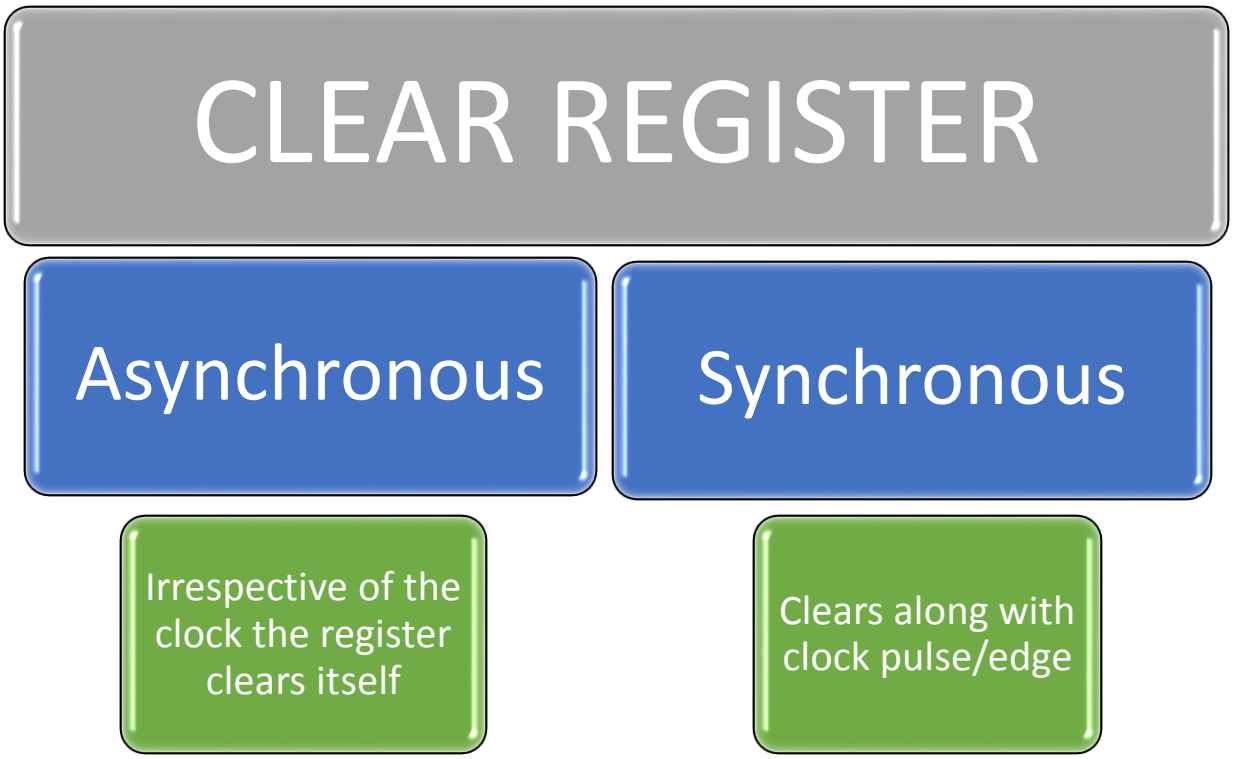
Revanth Gopi Konakanchi
Dept of CSE | 120101036

Anil Kumar Ravuru
Dept of CSE |120101059

Prudhvi Raj Chowhan
Dept of CSE |11010131

Clear Register is used to clear the previous existing values when necessary to do so. These are of two types

1. Asynchronous Register
2. Synchronous Register



COUNTERS:

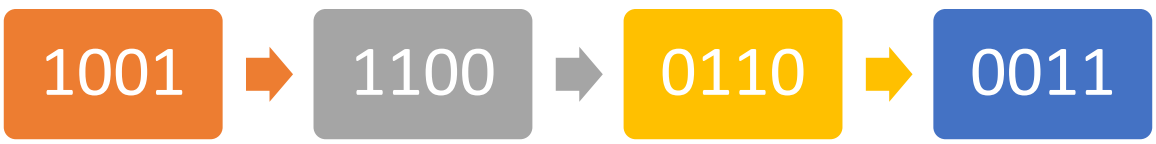
Counters are classified into 3 types

1. Shift Register
2. Ripple Counter
3. Synchronous Counter

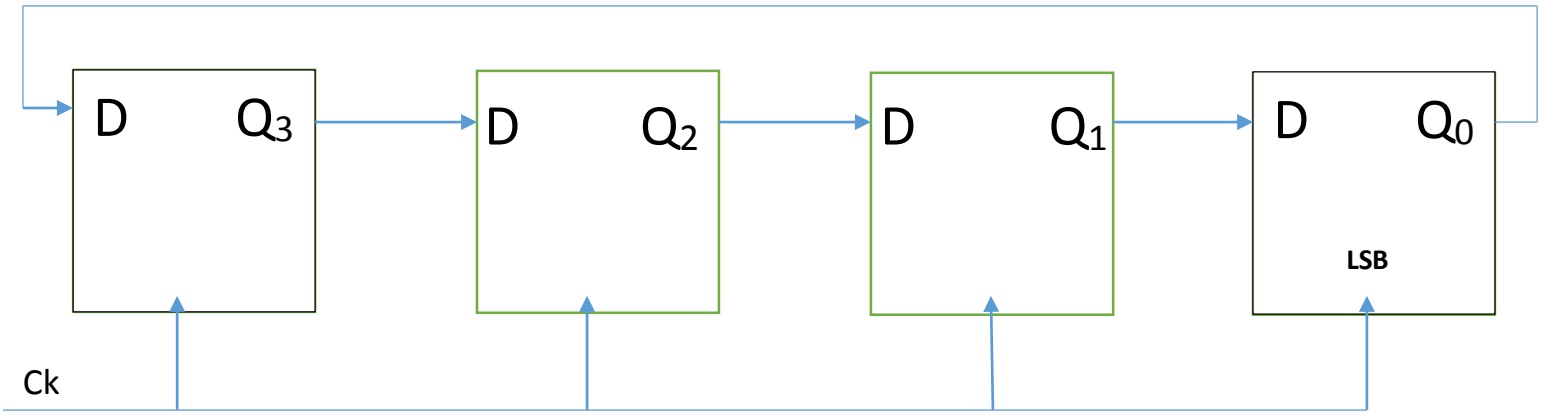
SHIFT REGISTER

shift register is a cascade of flip flops, sharing the same clock, in which the output of each flip-flop is connected to the data input of the next flip-flop in the chain, resulting in a circuit that shifts by one position the bit array stored in it.

Example



➢ Block Diagram for shift register



Consider the above example

Initially $Q_3Q_2Q_1Q_0 = 1001$

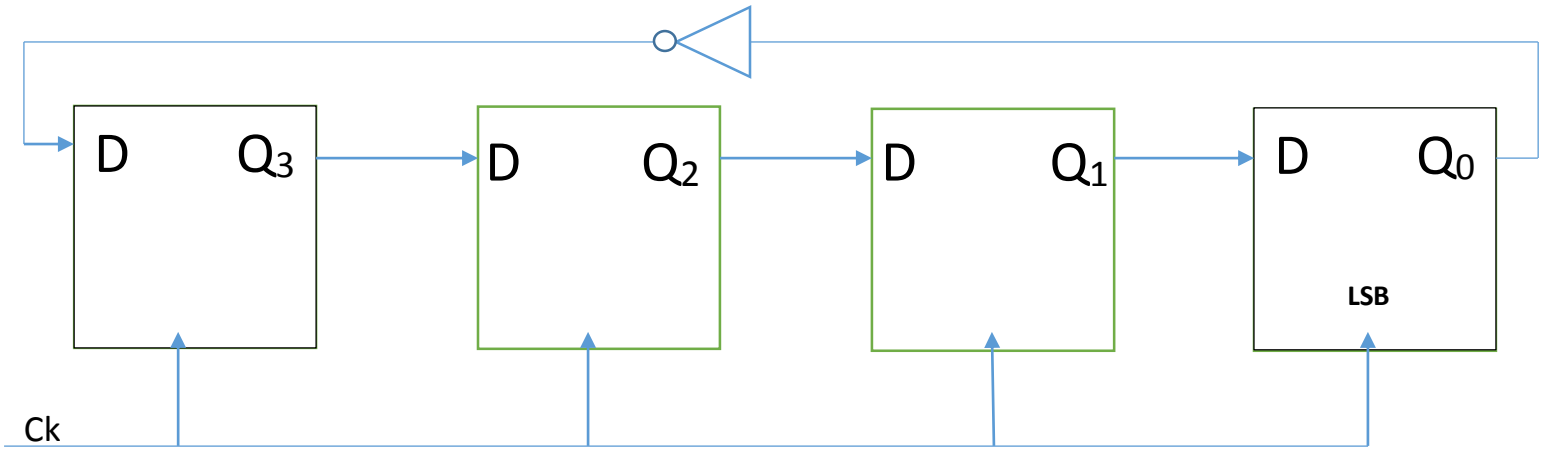
After one cycle $Q_3Q_2Q_1Q_0 = 1100$ (Q_i shifts to Q_{i-1})

After second cycle $Q_3Q_2Q_1Q_0 = 0110$

After third cycle $Q_3Q_2Q_1Q_0 = 0011$

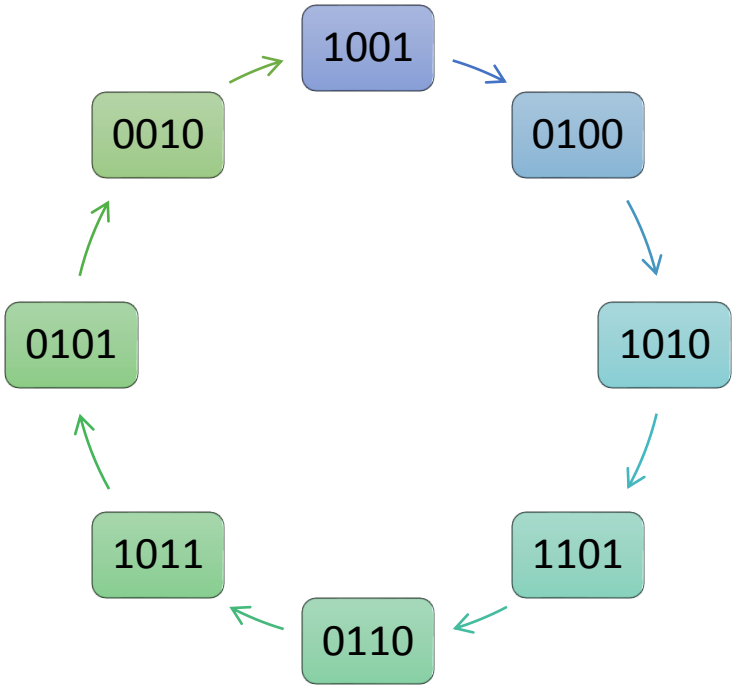
- In normal shift register maximum 4 different outputs are possible.

Johnson Counter (special case of shift register)



In the above example

Initially $Q_3Q_2Q_1Q_0 = 1001$



In this counter maximum 8 different outputs are possible.

RIPPLE COUNTERS

A ripple counter is a counter in which state transitions of one or more flip flops are triggered by the outputs of other flip flops in the circuit.

- Ripple counter is Asynchronous counter

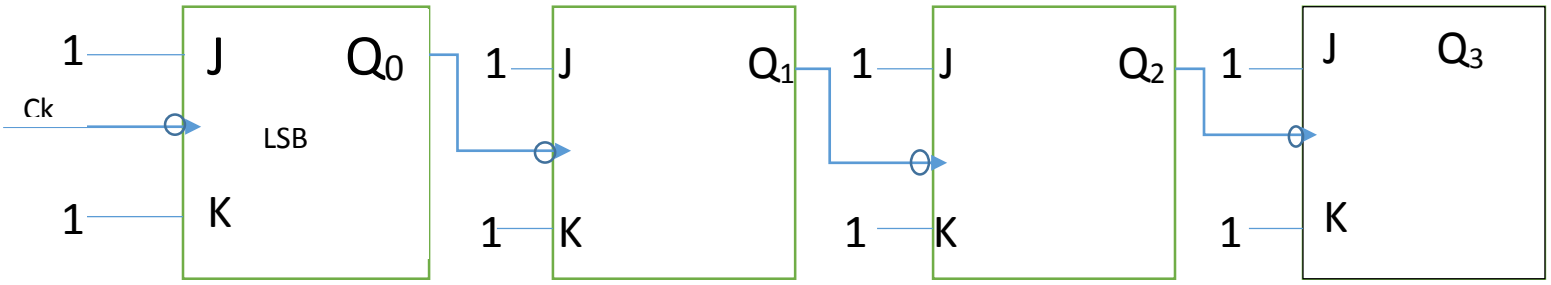
4 BIT BINARY UP COUNTER

Q_3	Q_2	Q_1	Q_0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
.	.	.	.
.	.	.	.
1	1	1	1

- Q_0 always toggles
- Q_1 toggles when Q_0 changes from 1 ➡ 0
- Q_2 toggles when Q_1 changes from 1 ➡ 0
- Q_3 toggles when Q_2 changes from 1 ➡ 0

4 bit binary up counter using J-K flip flop

- Using negative edge triggered clock



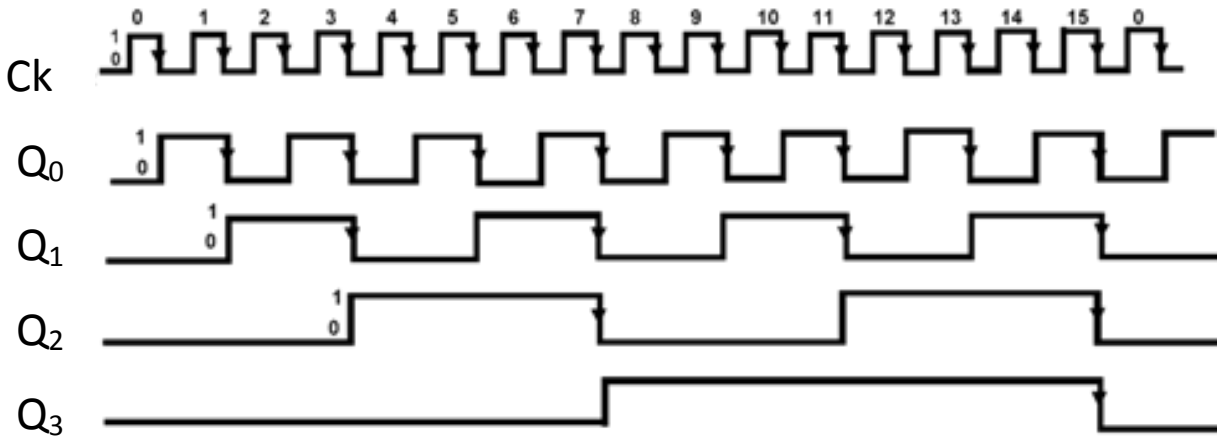
We know that when J=K=1 in J-K Flip-flop Output toggles

So for output Q_0 when clock is in negative edge triggered state the value toggles

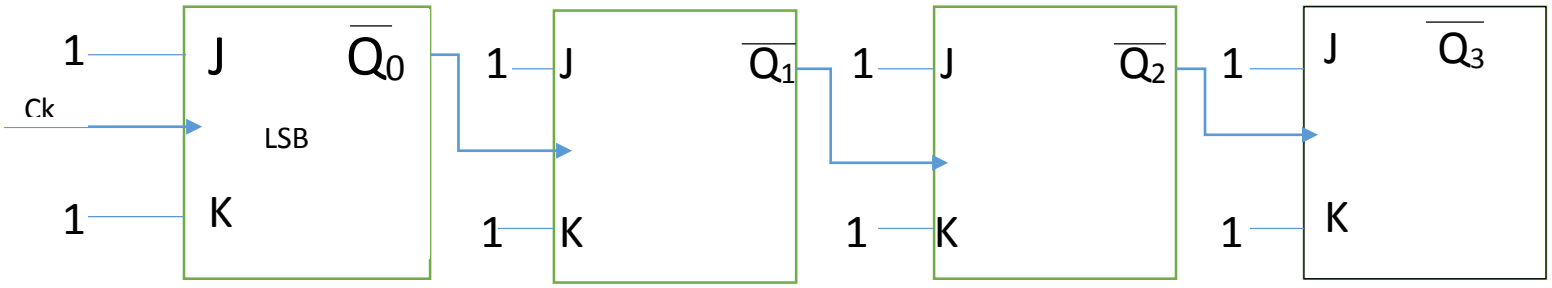
Similarly for Q_1, Q_0 acts as clock signal

Similarly for Q_2, Q_1 acts as clock signal

Similarly for Q_3, Q_2 acts as clock signal



❖ Using positive edge triggered clock



We know that when J=K=1 in J-K Flip-flop Output toggles

So for output Q_0 when clock is in positive edge triggered state the value toggles

Similarly for Q_1, Q_0 acts as clock signal

Similarly for Q_2, Q_1 acts as clock signal

Similarly for Q_3, Q_2 acts as clock signal

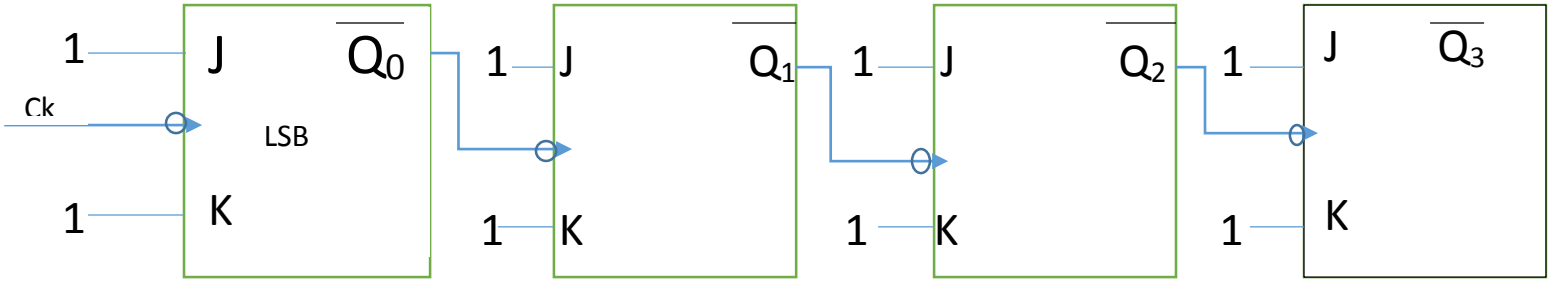
4 BIT BINARY DOWN COUNTER

Q_3	Q_2	Q_1	Q_0
1	1	1	1
1	1	1	0
1	1	0	1
1	1	0	0
1	0	1	1
1	0	1	0
.	.	.	.
.	.	.	.
0	0	0	0

- Q_0 always toggles
- Q_1 toggles when Q_0 changes from 0 ➡ 1
- Q_2 toggles when Q_1 changes from 0 ➡ 1
- Q_3 toggles when Q_2 changes from 0 ➡ 1

4 Bit Binary down Counter using J-K flip-flop

- Using negative edge triggered clock



We know that when J=K=1 in J-K Flip-flop Output toggles

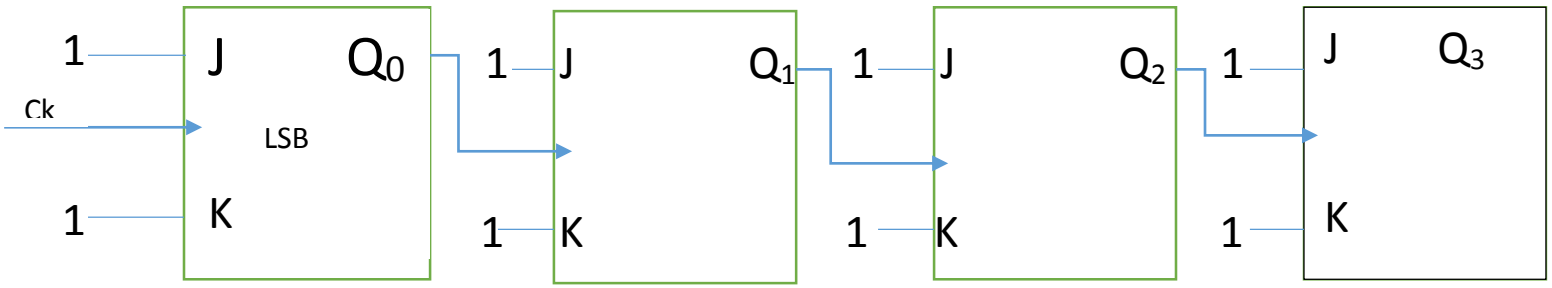
So for output Q_0 when clock is in negative edge triggered state the value toggles

Similarly for Q_1, Q_0 acts as clock signal

Similarly for Q_2, Q_1 acts as clock signal

Similarly for Q_3, Q_2 acts as clock signal

- Using positive edge triggered clock



We know that when J=K=1 in J-K Flip-flop Output toggles

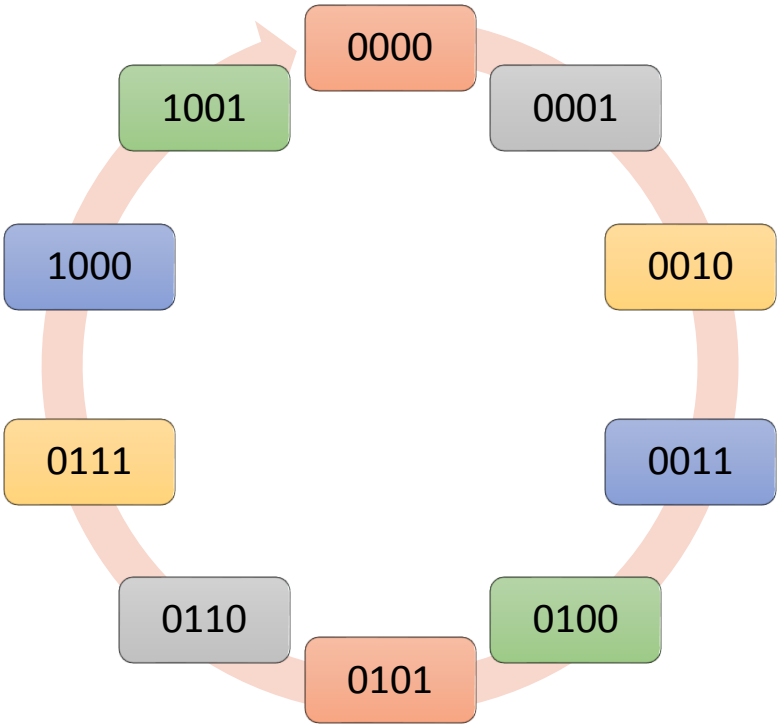
So for output Q_0 when clock is in positive edge triggered state the value toggles

Similarly for Q_1, Q_0 acts as clock signal

Similarly for Q_2, Q_1 acts as clock signal

Similarly for Q_3, Q_2 acts as clock signal

Decade up counter

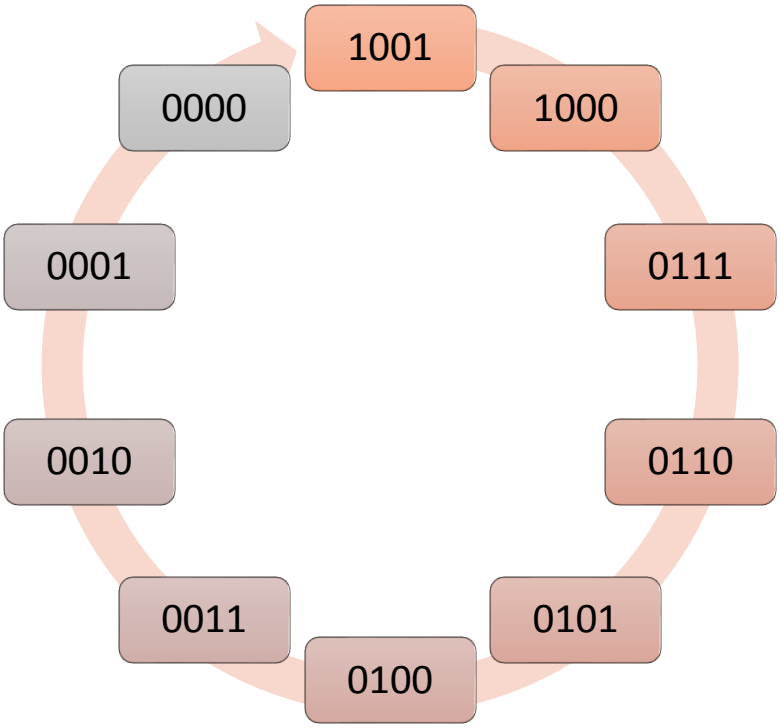


After 1001 $Q_3Q_2Q_1Q_0$ goes to 0000 instead of 1010.

So we reset the flip-flop using 1010 value

Reset = $Q_3\overline{Q_2}Q_1\overline{Q_0}$

Decode down counter



After 0000 $Q_3Q_2Q_1Q_0$ goes to 1001 instead of 1111.

So we reset the flip-flop using 1111 value

Reset = Q_3 Q_2 Q_1 Q_0

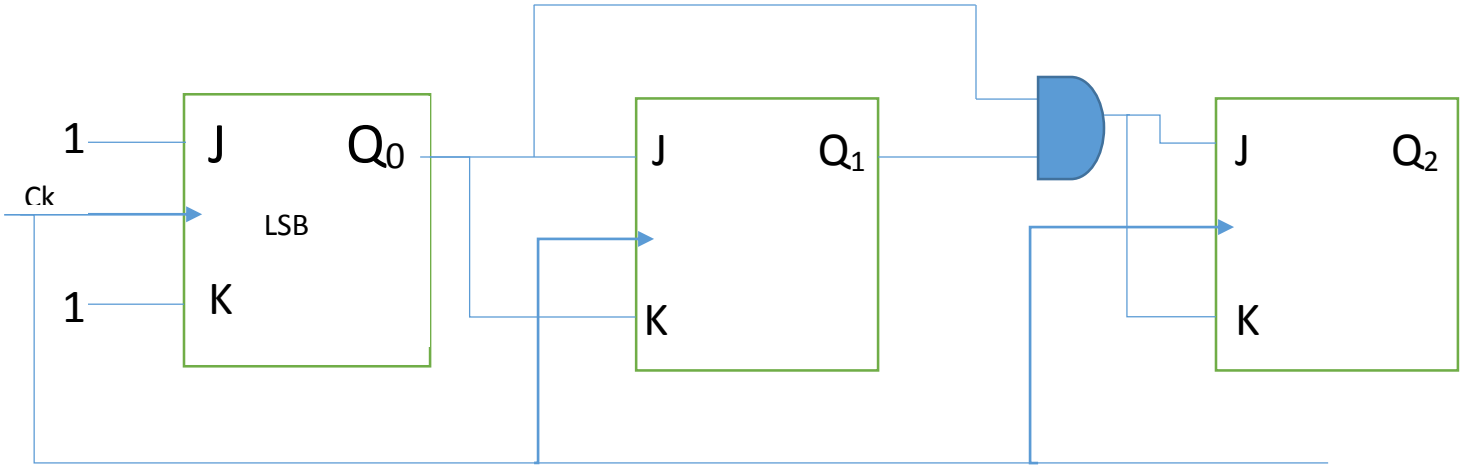
Synchronous counter

Every flip-flop receives the exact same clock pulse at the exact same time (Global Clock)

3 Bit Binary up Counter

Q_2	Q_1	Q_0
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1
0	0	0

- Q_1 toggles in next cycle when $Q_0 = 1$
- Q_2 toggles in next cycle when $Q_0=Q_1=1$


1
J
Q_0
Ck
LSB
1
K
J
Q_1
K
J
Q_2
K