# EXPT.-07

**AIM: -** Design and Simulate Up Counter using Verilog HDL.

**Software: -** Xilinx Vivado 2019.1

**Theory: -**

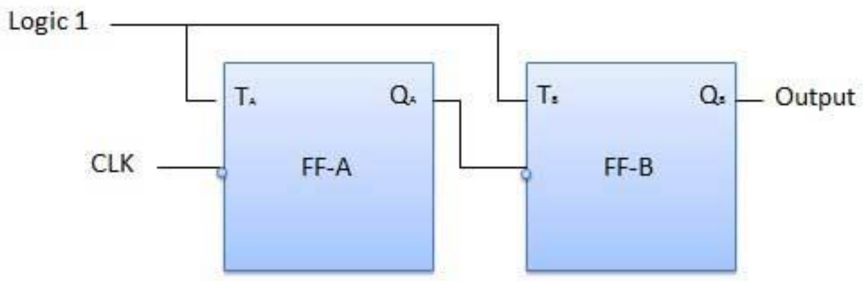
Counter is a sequential circuit. A digital circuit which is used for a counting pulses is known counter. Counter is the widest application of flip-flops. It is a group of flip-flops with a clock signal applied. Counters are of two types.

- Asynchronous or ripple counters.
- Synchronous counters.

## Asynchronous or ripple counters

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

### Logical Diagram



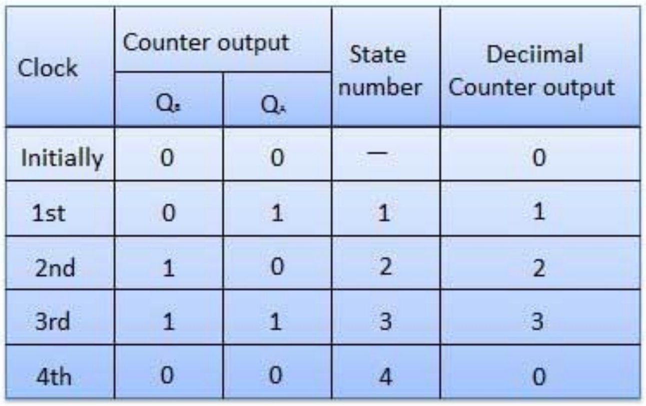
### Operation

| S.N. | Condition | 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 the first negative clock edge is applied, FF-A will toggle and $Q_A$ will be equal to 1. |

| | | |
|---|---|---|
| | | $Q_A$ is connected to clock input of FF-B. Since $Q_A$ has changed from 0 to 1, it is treated as the positive clock edge by FF-B. There is no change in $Q_B$ because FF-B is a negative edge triggered FF.<br><br>$Q_BQ_A$ = 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 $Q_A$ = 0.<br><br>The change in $Q_A$ acts as a negative clock edge for FF-B. So it will also toggle, and $Q_B$ will be 1.<br><br>$Q_BQ_A$ = 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.<br><br>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.<br><br>$Q_BQ_A$ = 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.<br><br>This negative change in $Q_A$ acts as clock pulse for |

| | | |
|---|---|---|
| | | FF-B. Hence it toggles to change $Q_B$ from 1 to 0. $Q_BQ_A$ = 00 after the fourth clock pulse. |

## Truth Table



| Clock | Counter output | | State number | Deciimal Counter output |
|---|---|---|---|---|
| | $Q_B$ | $Q_A$ | | |
| Initially | 0 | 0 | — | 0 |
| 1st | 0 | 1 | 1 | 1 |
| 2nd | 1 | 0 | 2 | 2 |
| 3rd | 1 | 1 | 3 | 3 |
| 4th | 0 | 0 | 4 | 0 |

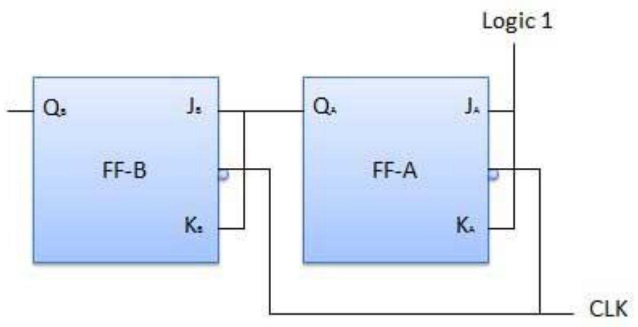
# Synchronous counters

If the "clock" pulses are applied to all the 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

| S.N. | Condition | 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 the first negative clock edge is applied, FF-A will toggle and $Q_A$ will change from 0 to 1.<br><br>But at the 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.<br><br>$Q_BQ_A$ = 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.<br><br>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.<br><br>$Q_BQ_A$ = 10 after the second clock pulse. |
| 4 | **After 3rd negative clock edge** | On application of the third falling clock edge, FF-A will toggle from 0 to 1 but there is no change of state for FF-B.<br><br>$Q_BQ_A$ = 11 after the third clock pulse. |
| 5 | **After 4th negative clock edge** | On application of the next clock pulse, $Q_A$ will change |

| | | |
|---|---|---|
| | | from 1 to 0 as $Q_B$ will also change from 1 to 0. $Q_BQ_A$ = 00 after the fourth clock pulse. |

## Classification of counters

Depending on the way in which the counting progresses, the synchronous or asynchronous counters are classified as follows −

- Up counters
- Down counters
- Up/Down counters

## UP/DOWN Counter

Up counter and down counter is combined together to obtain an UP/DOWN counter. A mode control (M) input is also provided to select either up or down mode. A combinational circuit is required to be designed and used between each pair of flip-flop in order to achieve the up/down operation.

- Type of up/down counters
- UP/DOWN ripple counters
- UP/DOWN synchronous counter

## UP/DOWN Ripple Counters

In the UP/DOWN ripple counter all the FFs operate in the toggle mode. So either T flip-flops or JK flip-flops are to be used. The LSB flip-flop receives clock directly. But the clock to every other FF is obtained from (Q = Q bar) output of the previous FF.

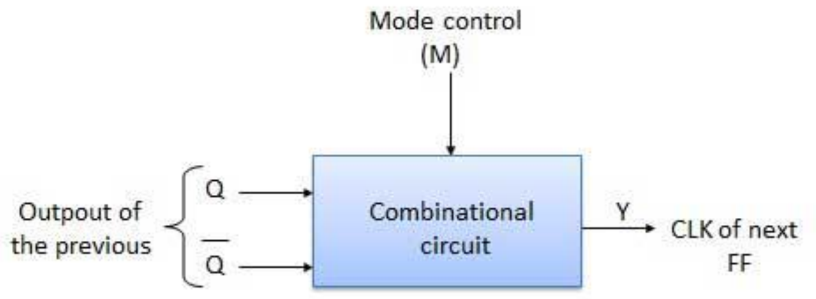
- **UP counting mode (M=0) −** The Q output of the preceding FF is connected to the clock of the next stage if up counting is to be achieved. For this mode, the mode select input M is at logic 0 (M=0).
- **DOWN counting mode (M=1) −** If M = 1, then the Q bar output of the preceding FF is connected to the next FF. This will operate the counter in the counting mode.

## Example

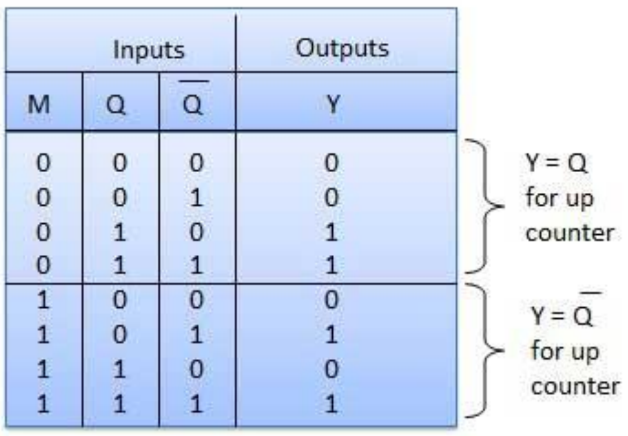
3-bit binary up/down ripple counter.

- 3-bit − hence three FFs are required.
- UP/DOWN − So a mode control input is essential.
- For a ripple up counter, the Q output of preceding FF is connected to the clock input of the next one.
- For a ripple up counter, the Q output of preceding FF is connected to the clock input of the next one.
- For a ripple down counter, the Q bar output of preceding FF is connected to the clock input of the next one.
- Let the selection of Q and Q bar output of the preceding FF be controlled by the mode control input M such that, If M = 0, UP counting. So connect Q to CLK. If M = 1, DOWN counting. So connect Q bar to CLK.

## Block Diagram



## Truth Table



| Inputs | | | Outputs |
|---|---|---|---|
| M | Q | $\overline{Q}$ | Y |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

Y = Q for up counter

Y = $\overline{Q}$ for up counter

## Operation

| S.N. | Condition | Operation |
|---|---|---|
| 1 | **Case 1 − With M = 0 (Up counting mode)** | If M = 0 and M bar = 1, then the AND gates 1 and 3 in fig. will be enabled whereas the AND gates 2 and 4 will be disabled. Hence $Q_A$ gets connected to the clock input of FF-B and $Q_B$ gets connected to the clock input of FF-C. These connections are same as those for the normal up counter. Thus with M = 0 the circuit work as an up counter. |
| 2 | **Case 2: With M = 1 (Down counting mode)** | If M = 1, then AND gates 2 and 4 in fig. are enabled whereas the AND gates 1 and 3 are disabled. Hence $Q_A$ bar gets connected to the clock input of FF-B and $Q_B$ bar gets connected to the clock input of FF-C. These connections will produce a down counter. Thus with M = 1 the circuit works as a down counter. |

## Modulus Counter (MOD-N Counter)

The 2-bit ripple counter is called as MOD-4 counter and 3-bit ripple counter is called as MOD-8 counter. So in general, an n-bit ripple counter is called as modulo-N counter. Where, MOD number = $2^n$.

### Type of modulus

- 2-bit up or down (MOD-4)
- 3-bit up or down (MOD-8)
- 4-bit up or down (MOD-16)

## Application of counters

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

**<u>Code:</u> -**

```
//JKFF
`timescale 1ns / 1ps

module JKFF(clk,J,K,clr,Q);
input clk,J,K,clr;
output reg Q;

always @(posedge clk)
begin
case({J,K})
2'b00:Q<=Q;
2'b01:Q<=0;
2'b10:Q<=1;
2'b11:Q<=~Q;
endcase
end

always @(posedge clr)
Q<=1'b0;

endmoduler
```

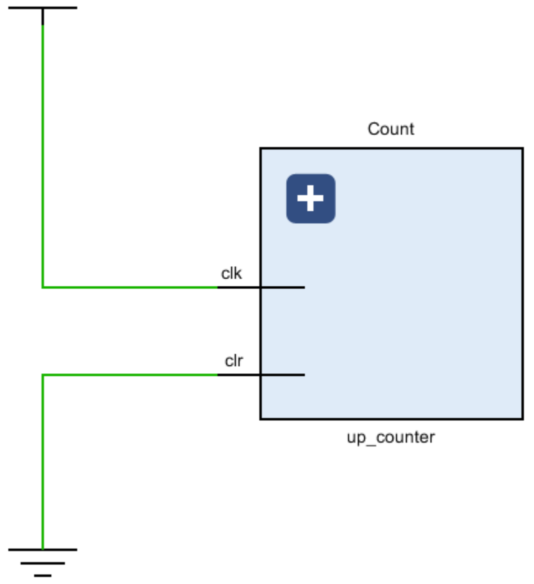
```
//tb
`timescale 1ns / 1ps

module tb_counter();

reg clk,clr;
wire [3:0] q;

up_counter Count(.clk(clk),.clr(clr),.q(q));
//JKFF FF(clk,j,k,clr,q);
initial
begin
clk=1'b0;
forever #3 clk=~clk;
end

initial
begin
clr=1'b1;
#10 clr=1'b0;
#50 clr=1'b1;
#10 clr=1'b0;
end

initial
$monitor($time,"q=%b,clk=%b,clr=%b",q,clk,clr);

initial
#150 $finish;

endmodule
```

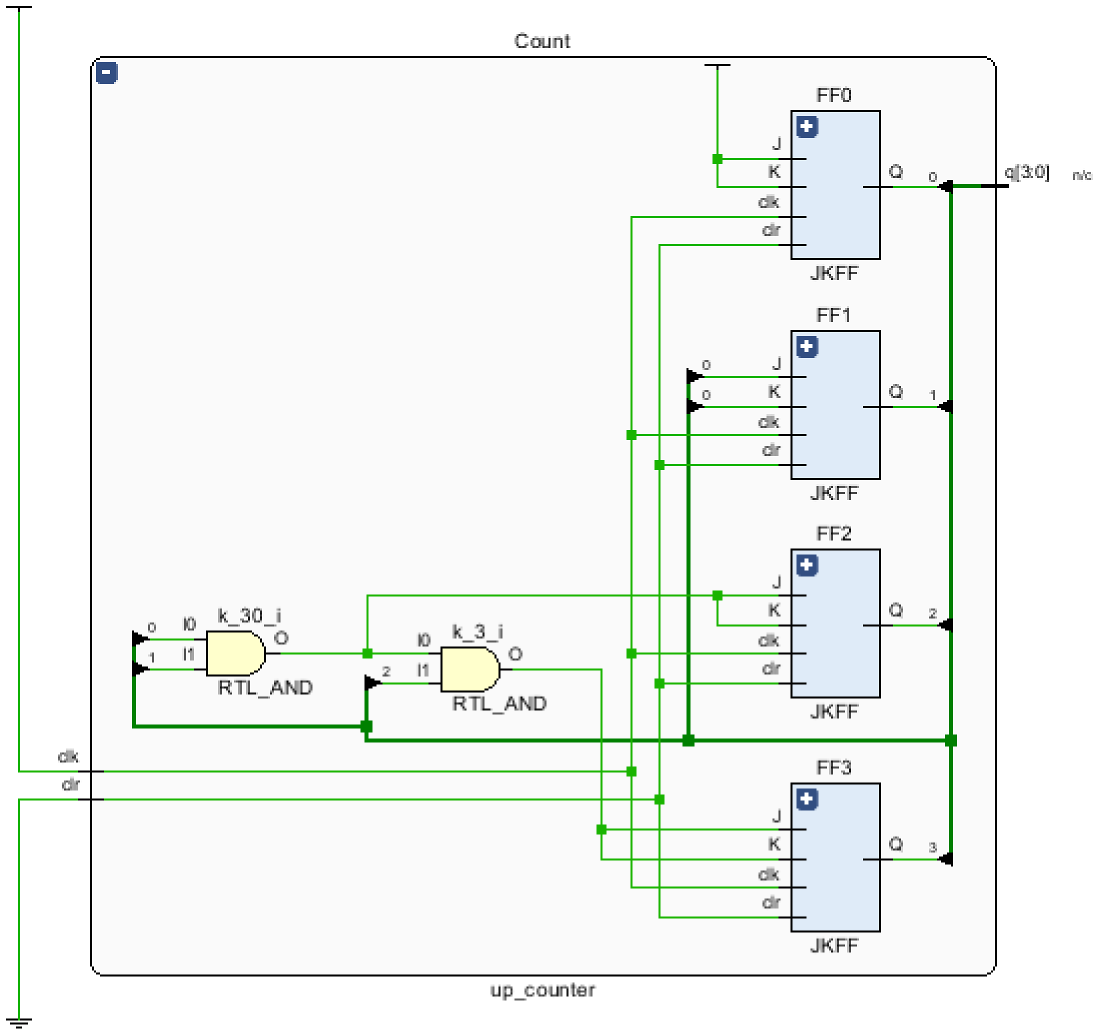
```
//up counter

`timescale 1ns / 1ps

module up_counter(clk,clr,q);

input clk,clr;
output [3:0] q;
wire [3:0] j,k;

assign j[0]=1'b1;
assign k[0]=1'b1;
JKFF FF0(clk,j[0],k[0],clr,q[0]);

assign j[1]=q[0];
assign k[1]=q[0];
JKFF FF1(clk,j[1],k[1],clr,q[1]);

assign j[2]=q[0]&q[1];
assign k[2]=q[0]&q[1];
JKFF FF2(clk,j[2],k[2],clr,q[2]);

assign j[3]=q[0]&q[1]&q[2];
assign k[3]=q[0]&q[1]&q[2];
JKFF FF3(clk,j[3],k[3],clr,q[3]);

endmodule
```

## Schematics and Simulated Waveforms: -

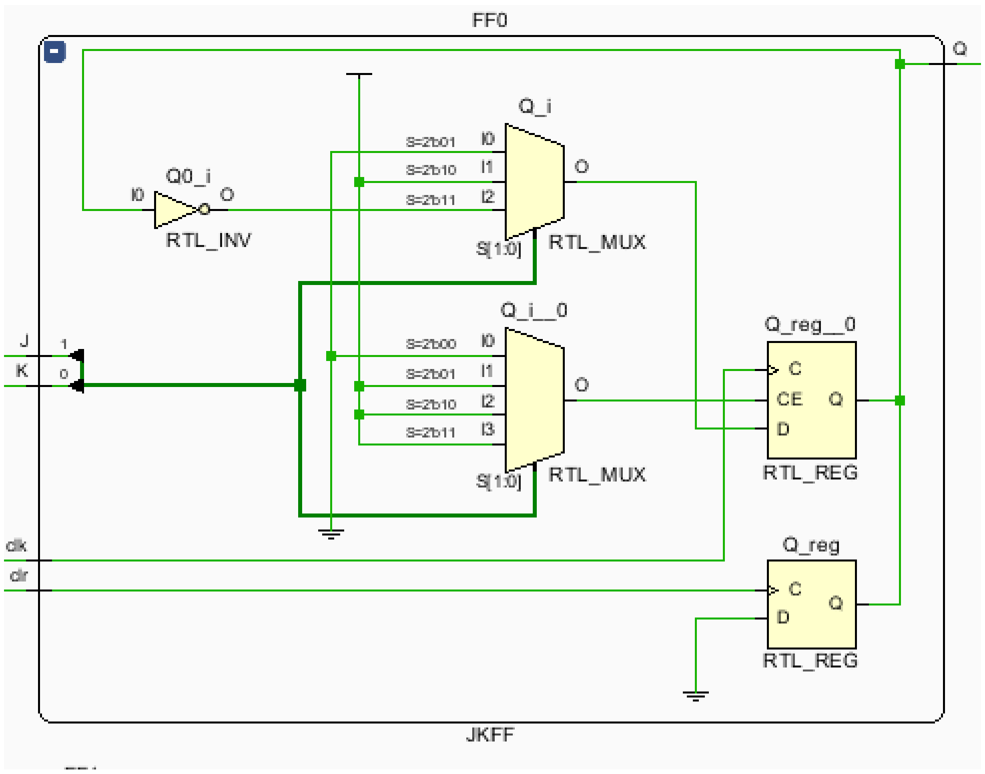
## Schematic:

Up Counter

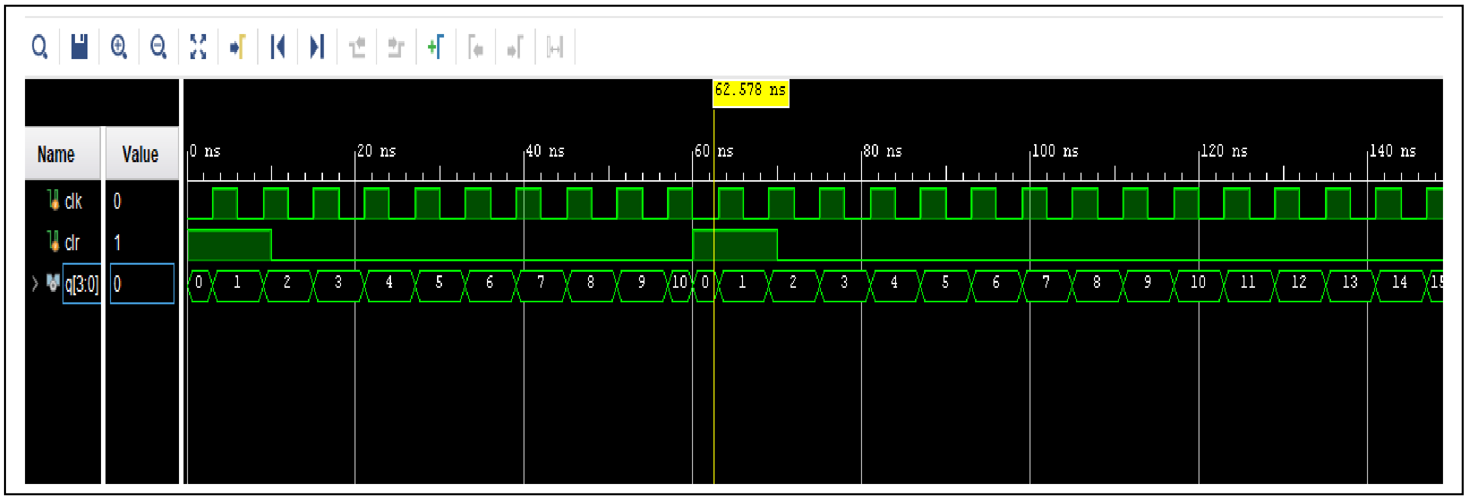

Count
clk
clr
up_counter


Count
FF0
J
K
clk
clr
Q
JKFF
q[3:0]
n/c
FF1
FF2
FF3
k_30_i
I0
I1
O
RTL_AND
k_3_i
clk
clr
up_counter

JKFF



## Waveform:



## Conclusion: -

Hence , we have successfully designed and simulated Up counter using behavioral modeling style in Verilog HDL. Also verified simulated waveforms with actual truth tables using Xilinx Vivado 2019.1