

# 10<sup>th</sup> Day

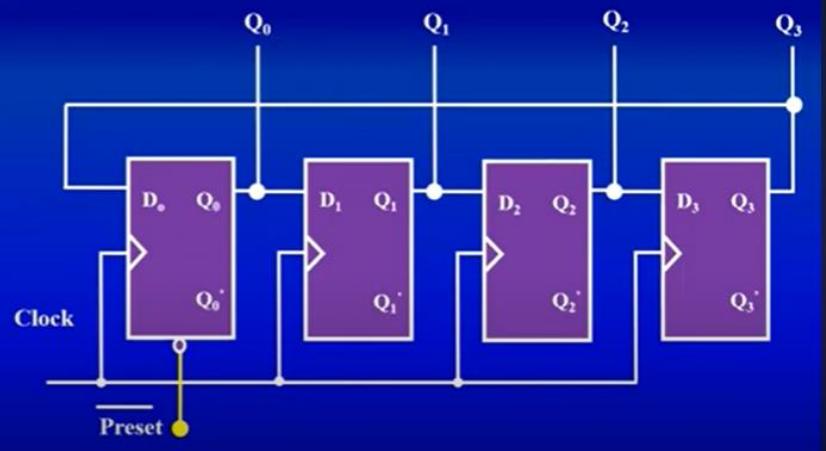
## Ring Counter

### **Ring Counter**

- **Ring counter** is an application of **Shift register**.
- The **last flip-flop output is connected to first flip-flop input**.
- The number of states produced by the Ring Counter is depends on the number of Flip-flops present in the counter.
- For a Ring Counter have **n-flip-flops**, it can produce **n-states**.

#### ◊ 4-bit Ring Counter

### **4-bit Ring Counter**



Q0	Q1	Q2	Q3
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1
1	0	0	0

## Design

```
module ring_counter(input clk,rst,output reg[0:3]q);

always @(posedge clk)
begin

if(rst)
q<=4'b1000;
else
begin
q<={q[3],q[0:2]};
end
end
endmodule
```

## Testbench

```
module ring_counter_tb();

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

ring_counter dut(clk,rst,q);

always #5 clk=~clk;

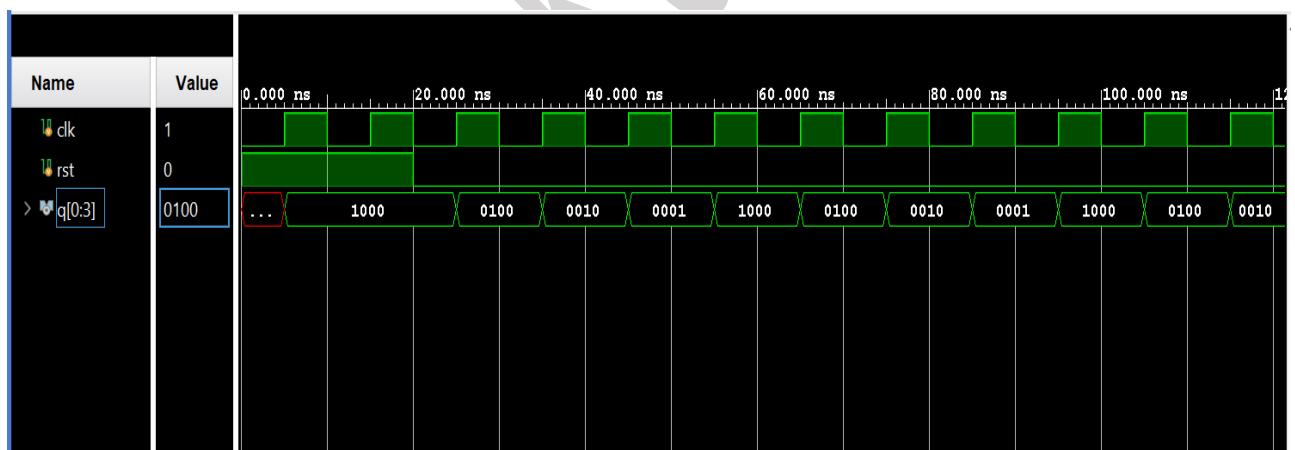
initial
begin
clk=1'b0;
rst=1'b1;
#20;
rst=0;
end

initial
begin
$monitor("time=%0t,clk=%b rst=%b q=%b",$time,clk,rst,q);
#150;
$finish;
end

endmodule
```

## Simulation

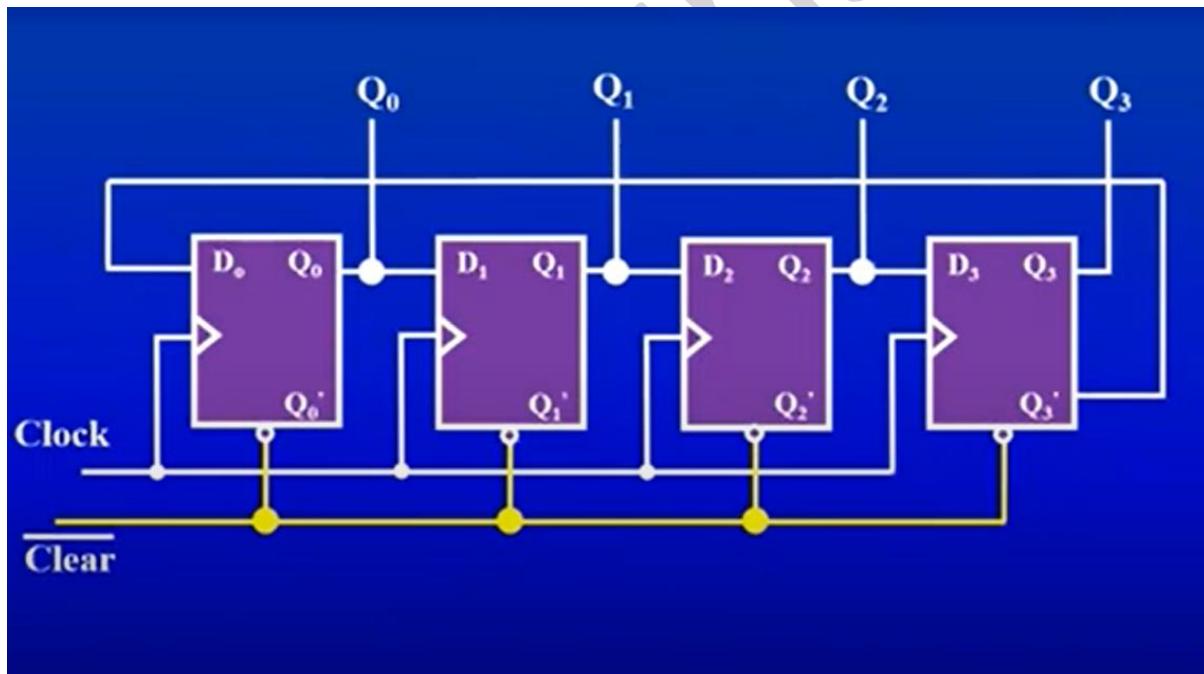
```
# run 1000ns
time=0,clk=0 rst=1 q=xxxx
time=5000,clk=1 rst=1 q=1000
time=10000,clk=0 rst=1 q=1000
time=15000,clk=1 rst=1 q=1000
time=20000,clk=0 rst=0 q=1000
time=25000,clk=1 rst=0 q=0100
time=30000,clk=0 rst=0 q=0100
time=35000,clk=1 rst=0 q=0010
time=40000,clk=0 rst=0 q=0010
time=45000,clk=1 rst=0 q=0001
time=50000,clk=0 rst=0 q=0001
time=55000,clk=1 rst=0 q=1000
time=60000,clk=0 rst=0 q=1000
```



## Johnson Ring Counter

### Johnson counter

- Johnson counter is also called **Twisted Ring counter or Switch-tail ring counter**.
- This counter is obtained from a **Serial-In Serial-Out (SISO)** shift register by providing feedback from the **inverted output ( $Q'$ ) of the last FF to the D input of the first FF**.
- The Q output of each DFF is connected to the DFF input of the next stage, but the  $Q'$  output of the last stage is connected to the D input of the first stage, therefore, the name twisted ring counter.
- This feedback arrangement produces a unique sequence of states.



Q0	Q1	Q2	Q3
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

## Design

```
module john_ring(input clk,rst,output reg[0:3]q);

always @(posedge clk)
begin

if(rst)
q<=4'b0000;
else
begin
q<={~q[3],q[0:2]};
end
end

endmodule
```

## Testbench

```
module john_ring_tb();

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

john_ring dut(clk,rst,q);

always #5 clk=~clk;

initial
begin
clk=1'b0;
rst=1'b1;
#20;
rst=0;
end
initial
begin
$monitor("time=%0t,clk=%b rst=%b q=%b",$time,clk,rst,q);
#150;
$finish;
end

endmodule
```

## Simulation

```
# run 1000ns
time=0,clk=0 rst=1 q=xxxx
time=5000,clk=1 rst=1 q=0000
time=10000,clk=0 rst=1 q=0000
time=15000,clk=1 rst=1 q=0000
time=20000,clk=0 rst=0 q=0000
time=25000,clk=1 rst=0 q=1000
time=30000,clk=0 rst=0 q=1000
time=35000,clk=1 rst=0 q=1100
time=40000,clk=0 rst=0 q=1100
time=45000,clk=1 rst=0 q=1110
time=50000,clk=0 rst=0 q=1110
time=55000,clk=1 rst=0 q=1111
time=60000,clk=0 rst=0 q=1111
time=65000,clk=1 rst=0 q=0111
time=70000,clk=0 rst=0 q=0111
time=75000,clk=1 rst=0 q=0011
time=80000,clk=0 rst=0 q=0011
time=85000,clk=1 rst=0 q=0001
time=90000,clk=0 rst=0 q=0001
time=95000,clk=1 rst=0 q=0000
time=100000,clk=0 rst=0 q=0000
```

