30 Days of RTL Coding

-By T Tushar Shenoy

Day 10

Problem Statement: Implementing Johnson Counter using Structural Style.

Theory:

Johnson counters, also known twisted ring counter, are a fundamental class of digital sequential circuits widely used in various electronic applications. These counters are particularly significant in digital circuit design due to their ability to generate a continuous cyclic sequence of binary values, making them valuable components in signal generation, pattern recognition, and control systems.

A Johnson counter circulates a sequence of ones and zeros in a continuous loop, with only one bit changing its state at a time. This sequential nature and cyclic behaviour make Johnson counters well-suited for applications where controlled and predictable sequencing is essential. The motivation behind this endeavour lies in the importance of sequential circuits in modern digital systems. Johnson counters offer unique advantages, such as simplicity, and the generation of cyclic patterns, making them suitable for a range of applications where repetitive sequences are required.

Johnson Counters can be easily constructed using flip-flops or shift registers, making them popular choices in many electronic projects, such as frequency dividers, LED chasers, and other control and automation systems. Compared to traditional binary counters, Johnson counters can produce twice as many unique count states using the same number of D Flip-Flops. This makes them highly efficient in generating a larger number of distinct sequences within a limited number of bits, its unique properties and efficient operation make it a versatile choice in various digital system designs and applications

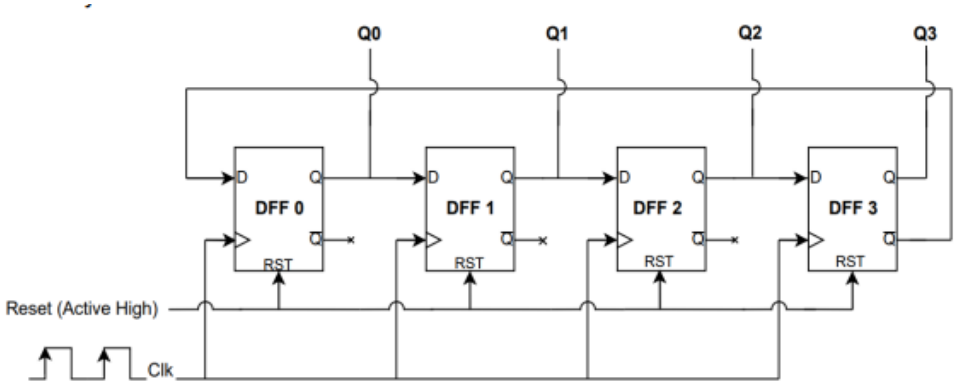


FIG: Johnson Counter Block Diagram

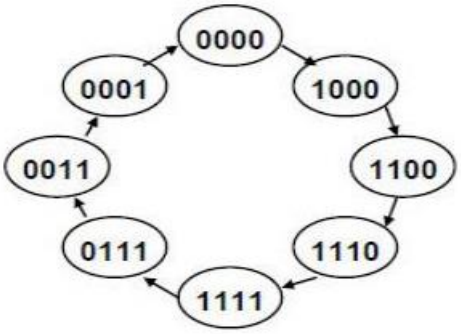


FIG Johnson Counter state diagram

- **Truth Table of D Flip Flop**

Clock	D	Q+	$\bar{Q}$+
0	X	Q	$\bar{Q}$
↑	0	0	1
↑	1	1	0

- **Johnson Counter truth table**

Reset	Clock	Q0	Q1	Q2	Q3
1	X	0	0	0	0
0	↑	1	0	0	0
0	↑	1	1	0	0
0	↑	1	1	1	0
0	↑	1	1	1	1
0	↑	0	1	1	1
0	↑	0	0	1	1
0	↑	0	0	0	1
0	↑	0	0	0	0

- **Johnson Counter excitation table**

Present State				Next State				D Flip-Flop Inputs			
Q0	Q1	Q2	Q3	$Q0^+$	$Q1^+$	$Q2^+$	$Q3^+$	D0	D1	D2	D3
0	0	0	0	1	0	0	0	1	0	0	0
1	0	0	0	1	1	0	0	1	1	0	0
1	1	0	0	1	1	1	0	1	1	1	0
1	1	1	0	1	1	1	1	1	1	1	1
1	1	1	1	0	1	1	1	0	1	1	1
0	1	1	1	0	0	1	1	0	0	1	1
0	0	1	1	0	0	0	1	0	0	0	1
0	0	0	1	0	0	0	0	0	0	0	0

Verilog Code:

```
//Verilog Code for D Flip Flop
module D_Flip_Flop(D, reset, clk, Q, Qb);
 input D, reset, clk;
 output Q, Qb;
 reg Q, Qb;
 always @(posedge clk,posedge reset)
 begin
 if (reset)
 Q = 0;
 else
 Q <= D;
 Qb <= ~Q;
 end
endmodule
```

```
//Verilog Code for Johnson Counter
module Johnson_Counter_4bit(clk, reset, Q);

 input clk;
 input reset;

 output [3:0] Q;

 wire Db3, Db2, Db1, Db0;

 D_Flip_Flop DFF_1(.D(Db0), .reset(reset), .clk(clk), .Q(Q[3]), .Qb(Db3));
 D_Flip_Flop DFF_2(.D(Q[3]), .reset(reset), .clk(clk), .Q(Q[2]), .Qb(Db2));
 D_Flip_Flop DFF_3(.D(Q[2]), .reset(reset), .clk(clk), .Q(Q[1]), .Qb(Db1));
 D_Flip_Flop DFF_4(.D(Q[1]), .reset(reset), .clk(clk), .Q(Q[0]), .Qb(Db0));
endmodule
```

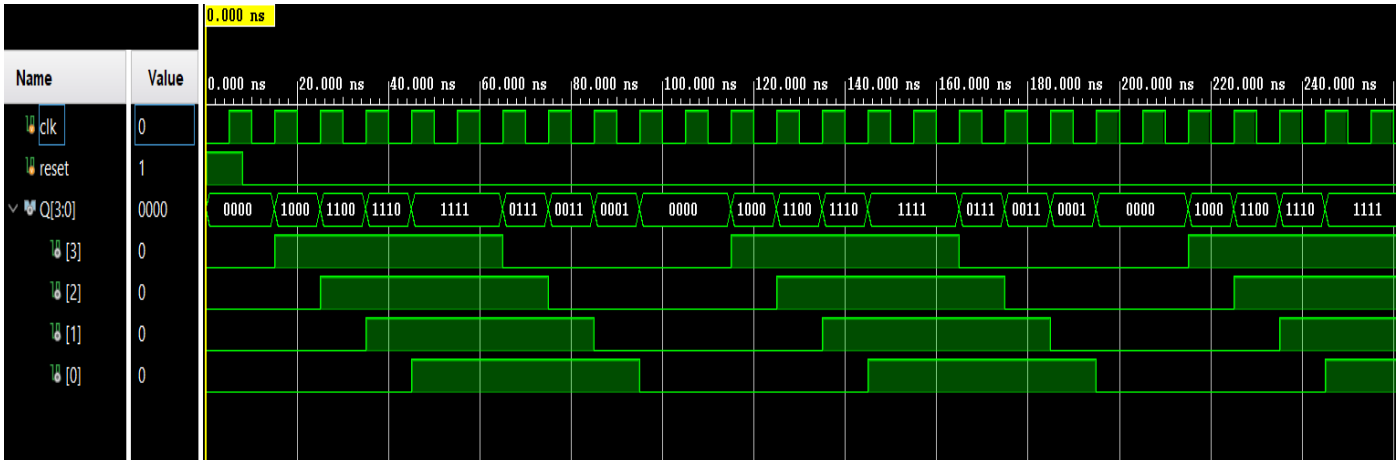
Testbench Code:

```
module Johnson_Counter_4bit_tb;
reg clk,reset;
wire [3:0]Q;

Johnson_Counter_4bit dut(clk, reset, Q);

initial begin
 clk=1'b0;
 reset=1;
 #8 reset=0;
end
always #5 clk=~clk;
endmodule
```

Simulation Output:



GitHub Repository URL: - https://github.com/tusharshenoy/RTL-Day-10-Johnson-Counter