https://www.pes.edu/wp-content/uploads/2019/09/pes_logo.png

**END SEMESTER ASSESSMENT (ESA) B.TECH. (CSE)**

**III SEMESTER**

**UE18CS206 – DIGITAL DESIGN & COMPUTER ORGANIZATION LABORATORY**

**PROJECT REPORT**

**ON**

“DESIGN AND IMPLEMENT A RING AND JOHNSON COUNTER WITH CONTROL LOGIC”

SUBMITTED BY

**NAME SRN**

1. **AchyutJagini PES2UG19CS013**
2. **Adithya M PES2UG19CS015**
3. **Aditya Mahesh Hegde PES2UG19CS016**

**4)Achyuta Bhimsen Mudhol PES2UG19CS014**

**AUGUST – DECEMBER 2020**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**ELECTRONIC CITY CAMPUS,**

**BENGALURU – 560100, KARNATAKA, INDIA**

|  |  |  |
| --- | --- | --- |
| TABLE OF CONTENTS | | |
| Sl.No | TOPIC | PAGE No |
|  | ABSTRACT OF THE PROJECT | 1 |
|  | CIRCUIT DIAGRAM | 2 |
|  | MAIN VERILOG CODE | 4 |
|  | TEST BENCH FILE | 5 |
|  | SCREEN SHOTS OF THE OUTPUT | 6 |

**ABSTRACT OF THE PROJECT:**

Ring counter is a sequential logic circuit which is constructed using shift registers, same data recirculates in the counter depending on the clock pulse.

It is of two types: -

1. Ordinary Ring Counter
2. Johnson Counter

**ORDINARY RING COUNTER: -**

In ring counter the output of last flip flop is connected to the input of the first flip flop. It transfers the same output throughout the circuit and if the output of any stage is 1 then its remainder is 0.By this what we mean is if the output of the first flip flop is 1, then this is transferred to the next stage (2nd flip flop) and by transferring this output to the next stage the output of the first flip flop becomes 0 and this process continues for all stages of a ring counter.

If we use n flip flop in a ring counter, then 1 is circulated for every n clock cycles. Initially all flip flop in the counter is set to 0 and before applying the clock pulse we assign the value 1 to the ring counter circuit.

Can be implemented using D and JK flip flops. It is a self-decoding circuit.

**JOHNSON COUNTER: -**

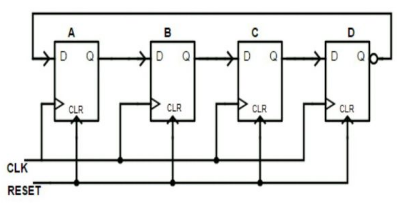
The Johnson Counter is a modification of the ring counter. In this the inverted output of the last stage flip flop is connected to the first stage flip flop.If we use n flip flop to design the Johnson Counter, then it is called a 2n bit Johnson Counter or |2n| Johnson Counter.

This is one of its advantage that it requires only half the amount of flip flop that that the ring counter uses (This is one of the limitations of the Ring counter taken care of). It is also known as a Twisted Ring Counter, with a feedback. Like Ring counter, the clock signal in Johnson counter is connected to the clock input of each flip flop simultaneously.

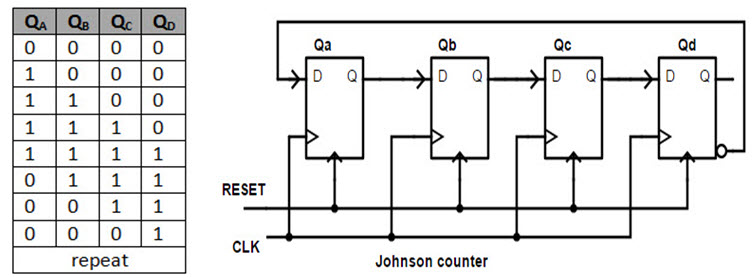
Initially Zero/NULL is fed to the Johnson counter and on applying the clock signal the output will change in a sequence and this sequence will repeat for next clock signal. Johnson counters are used as frequency dividers, by varying their feedback connection, they are also used to count the data in a continuous loop.

**CIRCUIT DIAGRAM:**

**4 BIT RING COUNTER**



**JOHNSON COUNTER**



MAIN VERILOG CODE:ring.v

module ring\_counter(

Clock,

Reset,

Count\_out

);

//what are the input ports and their sizes.

input Clock;

input Reset;

//what are the output ports and their sizes.

output [3:0] Count\_out;

//Internal variables

reg [3:0] Count\_temp;

//Whenever the Clock changes from 0 to 1(positive edge) or

//a change in Reset, execute the always block.

always @(posedge(Clock),Reset)

begin

if(Reset == 1'b1) begin//when Reset is high

Count\_temp = 4'b0001; end//The Count value is reset to "0001".

elseif(Clock == 1'b1) begin//When the Clock is high

//Left shift the Count value.

Count\_temp = {Count\_temp[2:0],Count\_temp[3]}; end

end

//The Count value is assigned to final output port.

assignCount\_out = Count\_temp;

endmodule

**TEST BENCH FILE:tb\_ring.v**

module tb\_ring;

// Inputs

reg Clock;

reg Reset;

// Outputs

wire [3:0] Count\_out;

// Instantiate the Unit Under Test (UUT)

ring\_counteruut (

.Clock(Clock),

.Reset(Reset),

.Count\_out(Count\_out)

);

///////////////////Clock generation ///////////////////////////

initial Clock = 0;

//Simulation inputs.

initialbegin $dumpfile("dump.vcd");

$dumpvars(0,Count\_out);

//Apply Reset for 50 ns.

Reset = 1; //Reset is high

#50; //Wait for 50 ns

Reset = 0; //Reset is low.

end

endmodule

**MAIN VERILOG CODE:johnson.v**

module johnson\_counter(

Clock,

Reset,

Count\_out

);

//what are the input ports and their sizes.

input Clock;

input Reset;

//what are the output ports and their sizes.

output [3:0] Count\_out;

//Internal variables

reg [3:0] Count\_temp;

//Whenever the Clock changes from 0 to 1(positive edge) or

//a change in Reset, execute the always block.

always @(posedge(Clock) or Reset)

begin

if(Reset == 1'b1) begin//when Reset is high

Count\_temp = 4'b0000; end//The Count value is reset to "0000".

elseif(Clock == 1'b1) begin//When the Clock is high

//Left shift the Count value and at the same time

//negate the least significant bit.

Count\_temp = {Count\_temp[2:0],~Count\_temp[3]}; end

end

//The Count value is assigned to final output port.

assignCount\_out = Count\_temp;

endmodule

TEST BENCH FILE:tb\_johnson.v

module tb\_johnson;

// Inputs

reg Clock;

reg Reset;

// Outputs

wire [3:0] Count\_out;

// Instantiate the Unit Under Test (UUT)

johnson\_counteruut (

.Clock(Clock),

.Reset(Reset),

.Count\_out(Count\_out)

);

///////////////////Clock generation ///////////////////////////

initial Clock = 0;

//Simulation inputs.

initialbegin $dumpfile("dump.vcd");

$dumpvars(0,Count\_out);

//Apply Reset for 50 ns.

Reset = 1; //Reset is high

#50; //Wait for 50 ns

Reset = 0; //Reset is low.

end

endmodule

**SCREEN SHOT OF THE OUTPUT:ring.v**

**iverilog -o aout ring.v tb\_ring.v**

A picture containing graphical user interface

Description automatically generated

**SCREEN SHOT OF THE OUTPUT:johnson.v**

**iverilog -o aout johnson.v tb\_johnson.v**

A picture containing graphical user interface

Description automatically generated