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

SL NO.	NAME	SRN
1.	PAVITRA K	PES2UG19CS279
2.	PAVAN N	PES2UG19CS277
3.	PHANI KUMAR VEDURUMUDI	PES2UG19CS281
4.	PAWAN PRASAD P	PES2UG19CS280

AUGUST – DECEMBER 2020

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

ELECTRONIC CITY CAMPUS,

BENGALURU – 560100, KARNATAKA, INDIA

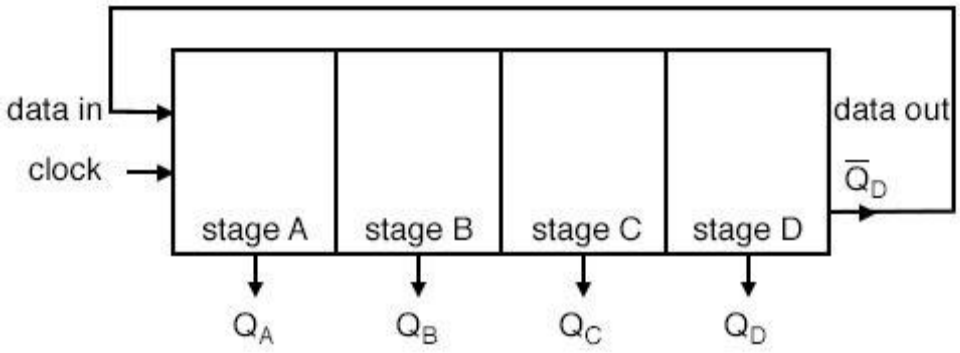
TABLE OF CONTENTS

ABSTRACT OF THE PROJECT

INTRODUCTION

Counters are sequential circuits whose function is to count pulse, frequency and time of the signal using a single clock signal. They are designed by grouping various flip flops together. Counters operate in different modes of modules, which are represented by the number of states of the cycle. There are two types of counter, they are synchronous and asynchronous counters. The synchronous counter depends on the input clock signal whereas the asynchronous counter does not. The synchronous counter is a shift register counter which is further classified as a ring-type and twisted type ring counter (Johnson counter).

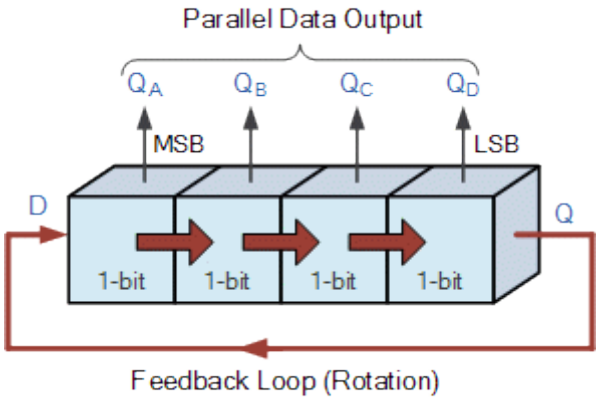
RING COUNTER



Ring Counter: shift register output fed back to input

A ring counter is also known as SISO (serial in serial out) shift register counter where the output of the flip flop is connected to the input of the flip flop. The designing of the ring counter can be done by using four D-Flip Flops with a common clock signal and overriding input can be connected to pre-set and clear. The main function of pre-set and clear is to change the output value when the input clock signal changes. Based on these signals they operate in ring format hence it is called a ring counter. The number of states is the number of flip flops used. Ring counter is a typical application of Shift resister. Ring counter is almost same as the shift counter. The only change is that the output of the last flip-flop is connected to the input of the first flip-flop in case of ring counter but in case of shift resister it is taken as output. Except this all the other things are same.

JOHNSON RING COUNTER



The Johnson Ring Counter, also called the Twisted Ring Counter is a shift register with feedback exactly the same as the standard Ring Counter, except that the **inverted** output of the last flip-flop is connected back to the input of the first flip-flop. Unlike the standard ring counter, it only needs half the number of flip-flops and its modulo number is halved. So a n-stage Johnson counter will circulate a single bit of data giving a sequence of 2n different states and can therefore be considered as a mod-2n counter. Johnson Ring Counter is nothing but Johnson Counter. The Johnson Ring Counter consists of a number of counters connected together with the output fed back to the input.

Difference between Ring Type Counter and Johnson Type Counter:

Ring Counter	Johnson Counter
The output of the last flip flop is given as input to the starting flip flop.	The output of the last flip flop is inverted and given as input to the starting flip flop.
Number of states is the number of flip flops used.	Number of states is half the number of flip flops used.
Input frequency = n Output frequency = f/n	Input frequency = f Output frequency = f/2n
Total unused states = (2n – n)	Total unused states = (2n – 2n)

There are multiple applications of these counters such as Frequency counter, ADC, Digital clocks, Generating staircase voltage, Washing machine, Alarm clocks, Measure timers and rate, etc.

In this project we explore the Design & implementation of a Ring and Johnson Counter with control logic using verilog along with the circuit diagram and waveform.

WORKING

RING COUNTER

Consider QA, QB, QC, QD as the 4 bits of the ring counter. The truth table for the 4-bit ring counter is given below.

States	QA	QB	QC	QD
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1

In the ring counter, logic '1' flows through all stages of the counter. In each state, it flows one bit to the right. When it reaches stage 4, it circulates back to stage 1 of the counter. Ring counter's state needs to be set before the operation. Since the ring counter circulates 1 through all stages, and there are no external inputs except the clock signal. So we need to set its state to initial state 1000 manually. We set the first stage flip-flop and clear the rest of the stages to obtain the state 1000. The preset input pin is designed to do this function.

Whenever the first clock edge hits the counter the outputs of each stage shifts to the next succeeding stage. And the output of the last will shift to the first stage making the state 0100. Upon the next clock cycle, each stage will update its state according to its input. So the '1' will be shifted to the third stage making the state 0010.

Upon another clock cycle, the '1' will reach the last stage making the 0001.

Now upon the next clock cycle, '1' from the last stage (flip-flop) will shift back to the first stage making the initial state 1000.

And it starts again from the first state repeating itself considering the clock signal is provided. This is how the data inside the ring counter circulates in the ring. Ring counter divides the frequency of the clock signal by 'n' where n is the bit size of the ring counter. So the ring counter can be used as a frequency divider.

JOHNSON COUNTER

Consider a 4-bit Johnson counter with QA, QB, QC, QD as the output of 4 stages of the counter. The truth table of the 4-bit Johnson counter is given below.

States	QA	QB	QC	QD
1	0	0	0	0
2	1	0	0	0
3	1	1	0	0
4	1	1	1	0
5	1	1	1	1
6	0	1	1	1
7	0	0	1	1
8	0	0	0	1

The output of each flip-flop is connected with the input of the succeeding flip-flop. The complemented output of the last flip-flop is connected with the input of the first flip-flop. It is also called the Inverse Feedback Counter or Twisted Ring Counter. The Same clock input is connected with all flip-flops. There is clear input for resetting the state to default 0000.

The default state of Johnson counter is 0000 thus before starting the clock input we need to clear the counter using clear input.

Whenever a clock edge hits the counter the output of each flip-flop will transfer to the next stage (flip-flop) but the inverted output of the last flip-flop will shift to the first stage making the state 1000.

Upon the next clock cycle, another '1' will stack in from the left side as the inverted output of the last stage will be shifted to the first stage.

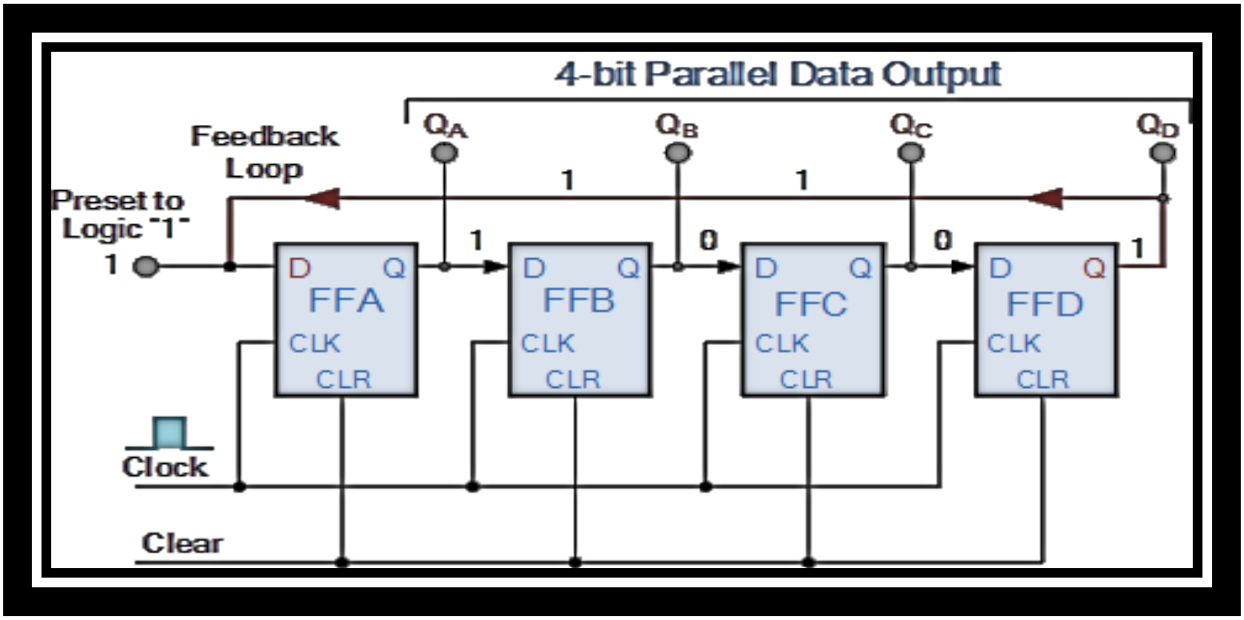
On the next clock cycle, another '1' will add in from the left until the state becomes 1111.
Now that the last flip-flop's output is '1', the next clock cycle will shift the invert of the last flip-flop which is '0' into the first flip-flop. It will result in stacking '0' from the left side. This stacking of the first 0 will make the state 1111 into 0111.

The next coming clock cycles will stack in the 0's from the left making the states 0011, 0001 & 0000 respectively with each clock cycle.

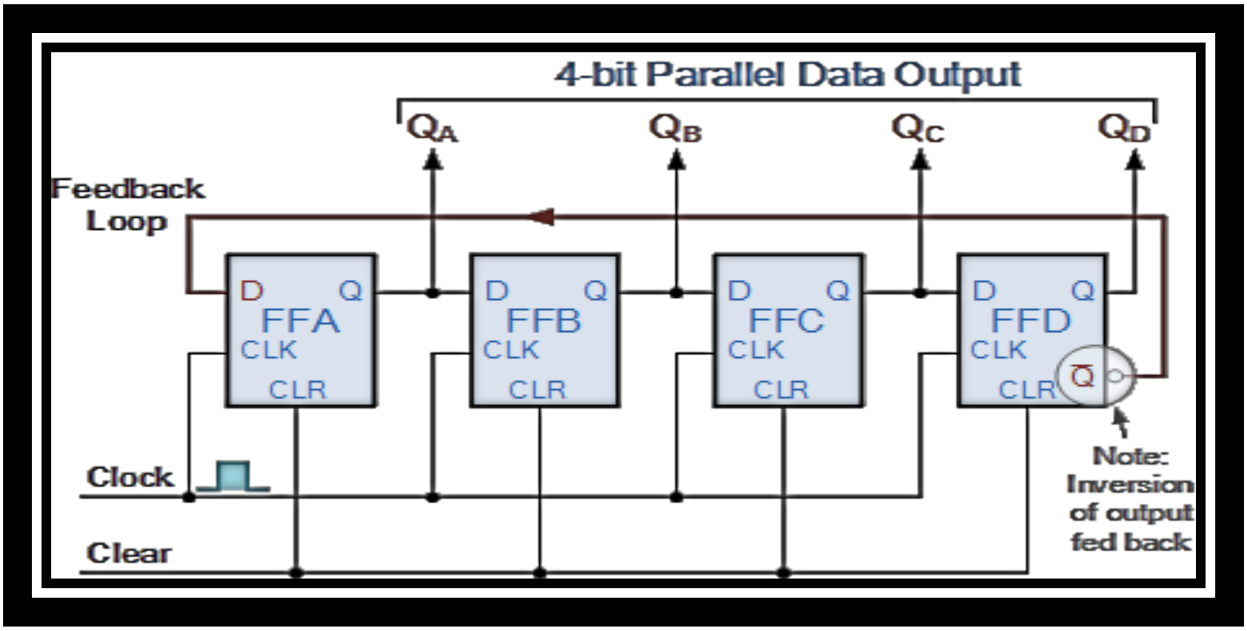
Eventually, it reaches its default state and it starts from the beginning again. Hence in a four-bit counter, with initial register values of 0000, the repeating pattern is: 0000, 1000. 1100, 1110, 1111, 0111, 0011, 0001, 0000….

CIRCUIT DIAGRAM

RING COUNTER



JOHNSON COUNTER



MAIN VERILOG CODE

RING COUNTER

```
module ring_counter (input clk, clr, output [3:0] out);
reg[3:0] cnt;
always @(posedge clk)
        if (clr)
                cnt = 4'b0001;
        else
        begin
                cnt <= cnt<<1;
                cnt[0] <= cnt[3];
        end
        assign out=cnt;
endmodule
```

JOHNSON COUNTER

```
module johnson_counter(clk, clr, out);
input clk,clr;
output [3:0] out;
reg [3:0] df;
always @(posedge clk)
begin
                if(clr)
                        dff=4'b0000;
                else
                begin
                        dff[3]<=dff[2];
                        dff[2]<=dff[1];
                        dff[1]<=dff[0];
                        dff[0]<=(~dff[3]);
                end
end
assign out = dff;
endmodule
```

TEST BENCH FILE

RING COUNTER

```
`timescale 1ns / 1ps
module tb_ring_counter;
reg clk;
reg clr;
wire[3:0] out;

initial
begin
$dumpfile("ring_counter.vcd");
$dumpvars(0,tb_ring_counter);
end

ring_counter r1 ( .clk(clk), .clr(clr), .out(out));
always #10 clk = ~clk;
initial
begin
clk = 0;
clr = 0;
#5 clr = 1;
#20 clr = 0;
#250 $finish;
end

initial
begin
$monitor($time,"clear=%1b,clock=%1b,count=%4b",clr,clk,out);
end

endmodule
```

JOHNSON COUNTER

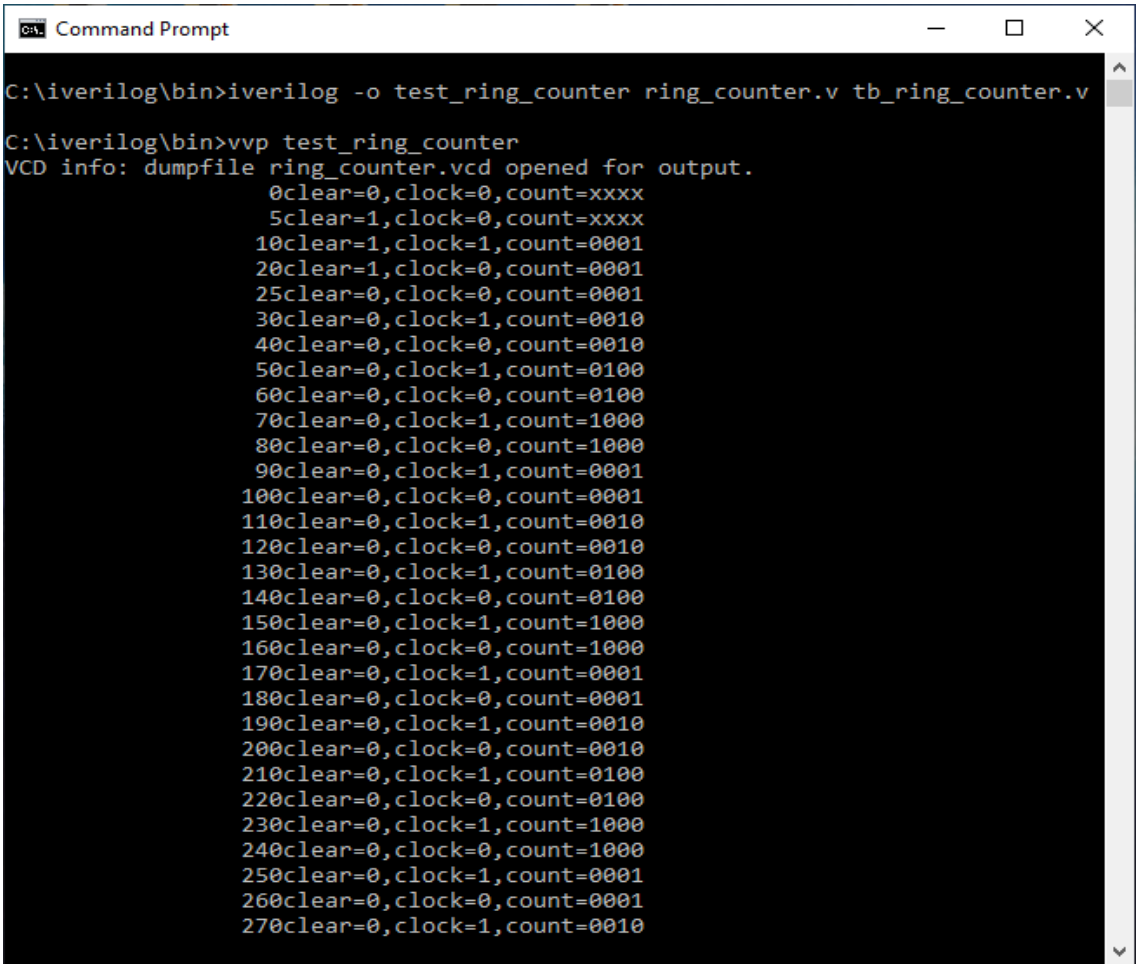
```
module tb_johnson_counter;
reg clk,clr;
wire [3:0] out;

initial
begin
$dumpfile("johnson_counter.vcd");
$dumpvars(0,tb_johnson_counter);
end

johnson_ctr j (.out(out), .clr(clr), .clk(clk));
always
```

```
#5 clk =~clk;
initial
begin
clr=1'b1; clk=1'b0;
#20 clr= 1'b0;
end

initial
begin
$monitor( $time, "clr=%b, clk=%b, out= %b", clr,clk,out);
#105 $finish;
end

endmodule
```

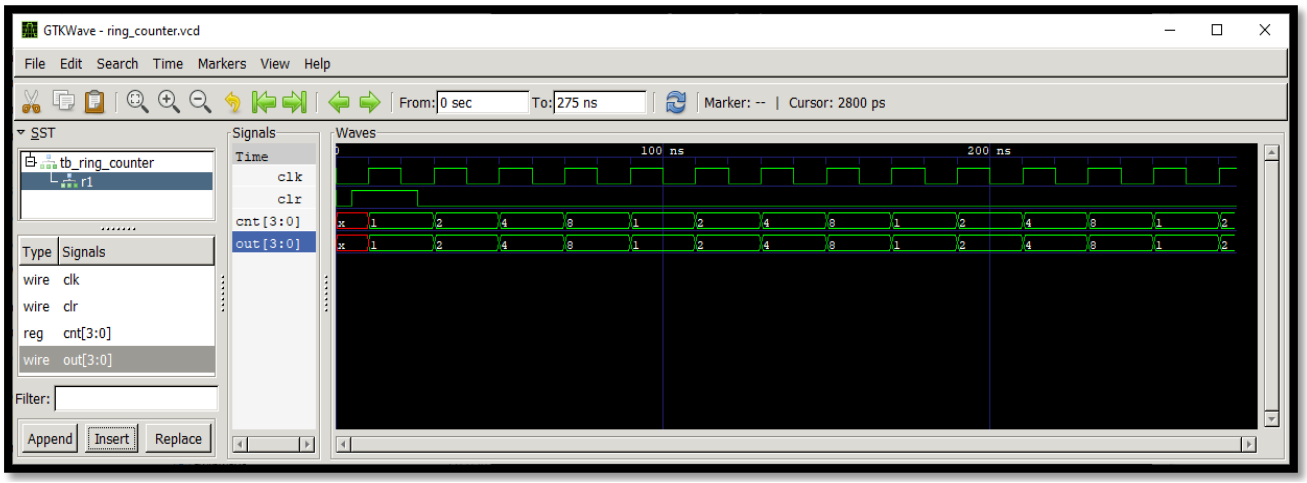
SCREEN SHOT OF THE OUTPUT

RING COUNTER

COMMAND PROMPT:

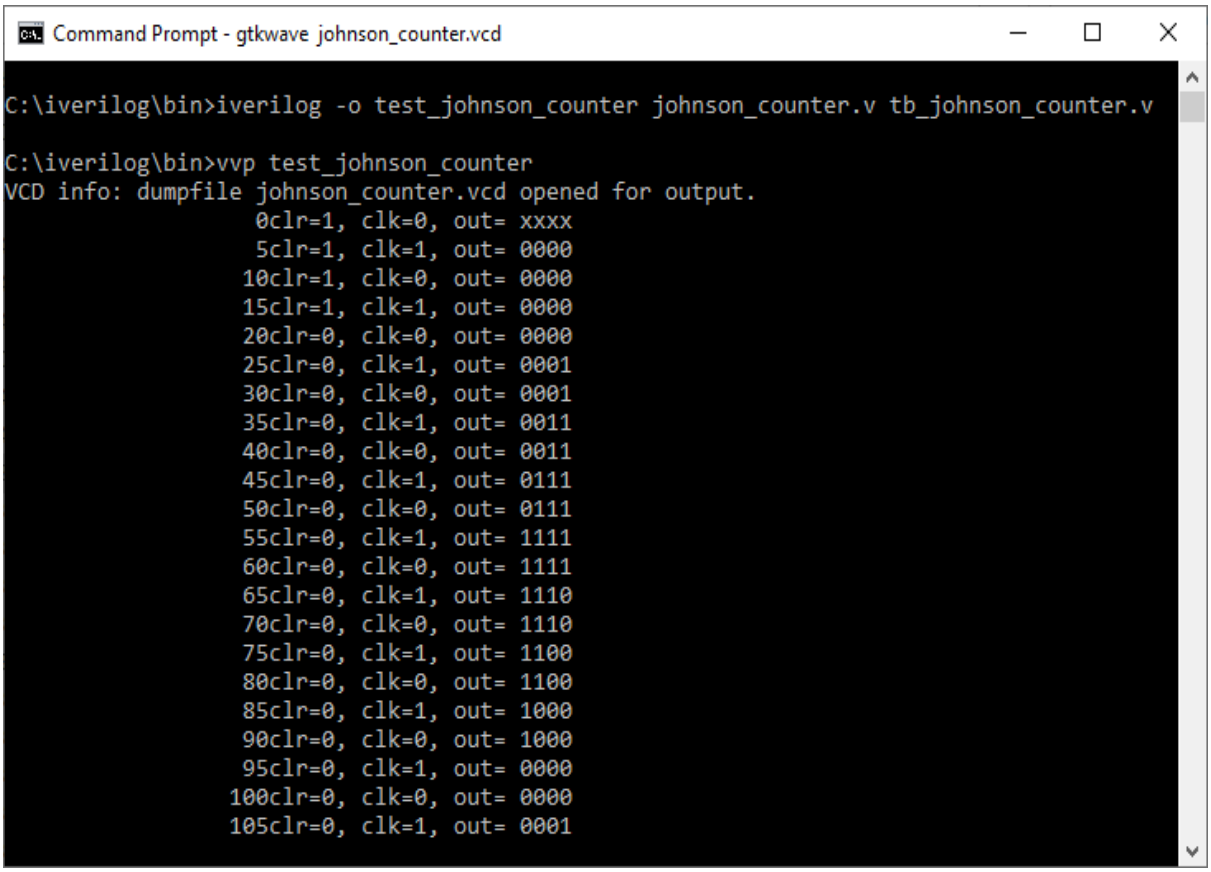


GTKWAVEFORM:



JOHNSON COUNTER

COMMAND PROMPT:



GTKWAVEFORM: