



Ring Counter Verilog Code:

```
module Ring_Count #(parameter WIDTH=4)
(
    input clk,
    input rstn,
    output reg [WIDTH-1:0] out
);
integer i = 0;
always @(posedge clk) begin
    if (!rstn)
        out <= 1;
    else
        begin
            out[WIDTH-1] <= out[0];
            for (i = 0; i < WIDTH-1; i=i+1)
                begin
                    out[i] <= out[i+1];
                end
            end
        end
end
endmodule
```



Test Bench:

```
module Ring_Count_TB;
```

```
    parameter WIDTH = 4;
```

```
    reg clk;
```

```
    reg rstn;
```

```
    wire [WIDTH-1:0] out;
```

```
    Ring_Count u0 (.clk (clk),
```

```
                  .rstn (rstn),
```

```
                  .out (out));
```

```
    always #10 clk = ~clk;
```

```
    initial begin
```

```
        {clk, rstn} <= 0;
```

```
        $monitor ("T=%0t out=%b", $time, out);
```

```
        repeat (2) @(posedge clk);
```

```
        rstn <= 1;
```

```
        repeat (15) @(posedge clk);
```

```
        $finish;
```

```
    end
```

```
endmodule
```