ES204 Digital Systems
LAB Assignment - 3

Indian Institute of Technology, Gandhinagar
January 24, 2024
Time & Venue: Tuesday 8:30-9:50am [7/108]

Submission deadline: Jan 24, 2024 (9:50 am)
Marks : 20

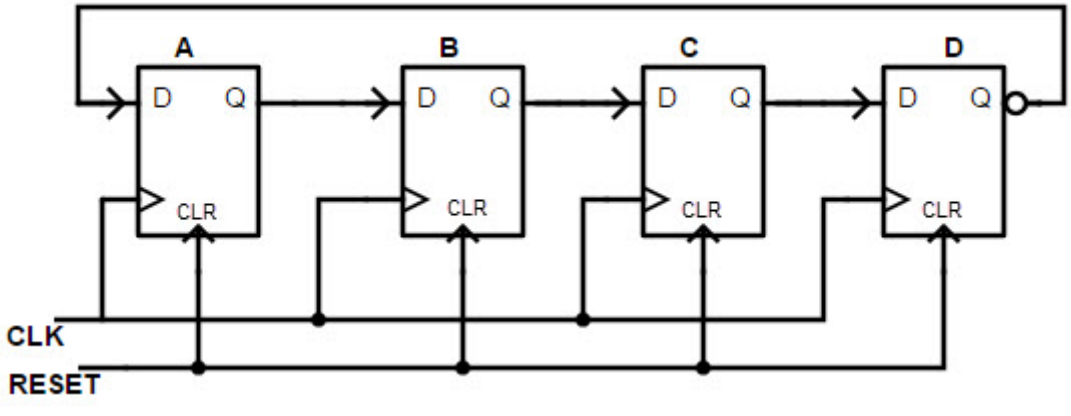
For each of the questions, write a Verilog code. You also need to create a **testbench** and show the simulation results.

[20 Marks]

Implement at 4-bit Johnson Ring Counter. Use procedural way of writing the code (and not structural.)

Observe the circuit that will result if you use blocking statements (=) versus non-blocking (<=) statements. Show the schematic of both of the designs for comparison.

The working of the Johnson ring counter can be understood from the circuit below.



Lab 3 - ES 204

Member 1: Birudugadda Srivibhav (22110050)
Member 2: Reddybathuni Venkat (22110220)

Implementation of Johnson Ring Counter with non-blocking (<=) statements.

1. Code:

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 24.01.2024 03:09:51
// Design Name:
// Module Name: johnson_ring_4bit
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module johnson_ring_4bit(
    input clk, reset,
    output reg [3:0] Q
    );
    always@(posedge clk)
        begin
            if(!reset)
                begin
                    Q[3] <=0;
                    Q[2] <=0;
                    Q[1] <=0;
                    Q[0] <=0;
                end
            else begin
                Q[3]<=Q[2];
                Q[2]<=Q[1];
```

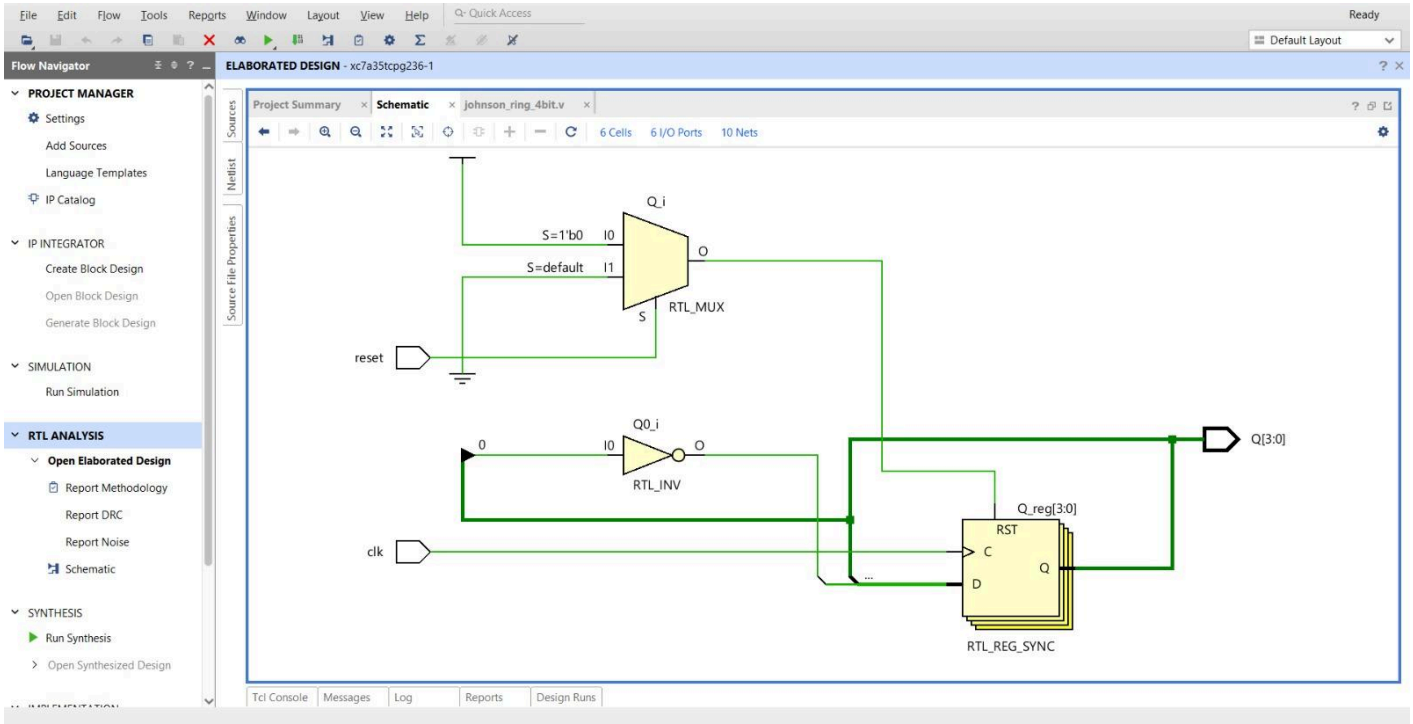
```
            Q[1]<=Q[0];
            Q[0]<=(~Q[3]);
            end
        end
endmodule
```

2. Test Bench:

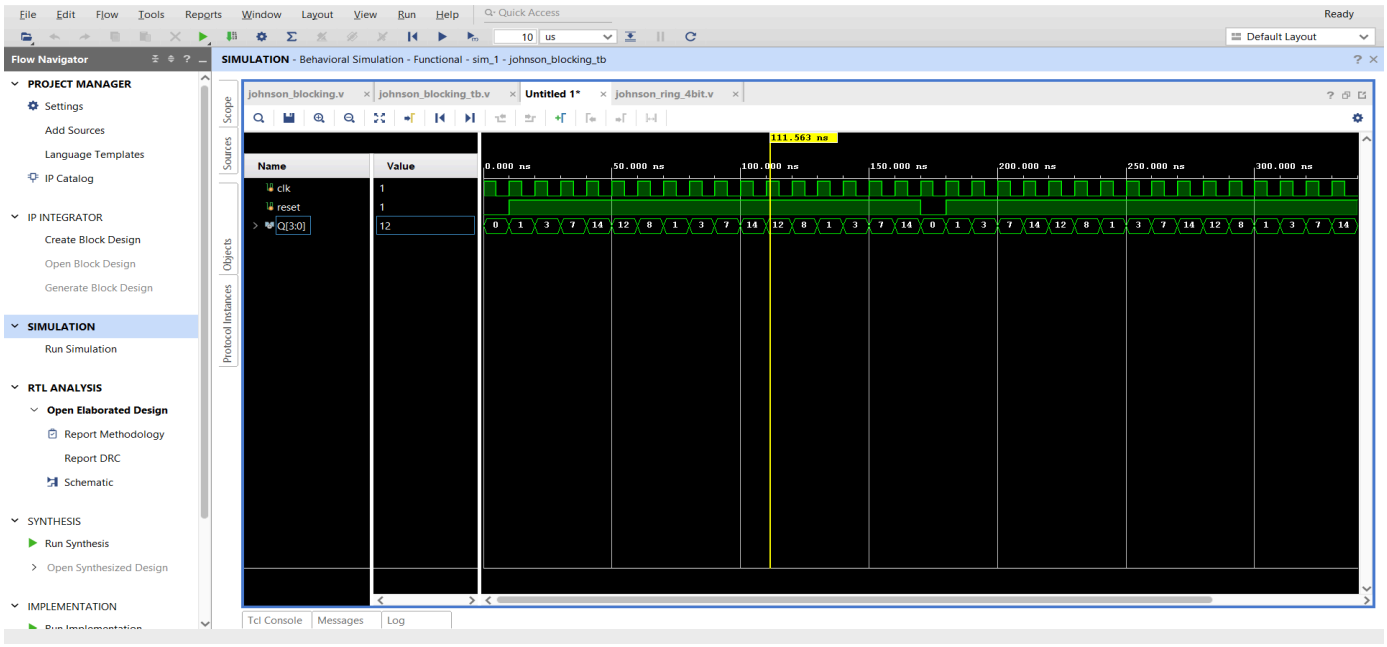
```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 24.01.2024 03:30:00
// Design Name:
// Module Name: johnson_ring_tb
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module johnson_ring_tb();
    reg clk;
    reg reset;
    wire [3:0] Q;
    johnson_ring_4bit uut(.clk(clk), .reset(reset), .Q(Q));
    initial
    begin
    clk = 1;
        forever #5 clk = ~clk;
    end
    initial
    begin
        reset = 0;
```

```
        #10;
        reset = 1;
        #160;
        reset = 0;
        #10;
        reset = 1;
        #160;
        $finish();
    end
Endmodule
```

3. Circuit Diagram:



4. Simulation Results:



Implementation of Johnson Ring Counter with blocking (=) statements.

1. Code:

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 24.01.2024 03:48:42
// Design Name:
// Module Name: johnson_blocking
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
```

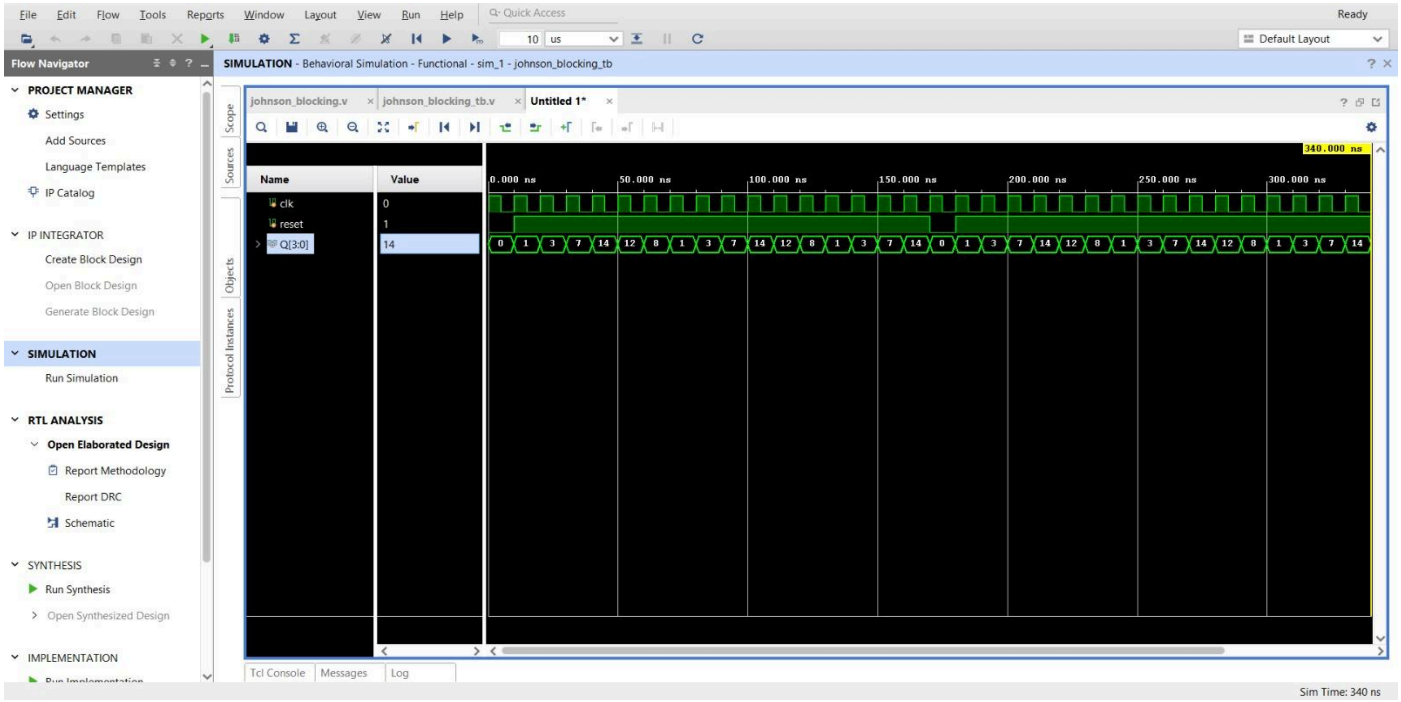
```
//
//////////////////////////////////////////////////////////////////////////////////


module johnson_blocking(
    input clk, reset,
    output reg [3:0] Q
    );
        always@(posedge clk)
        begin
            if(!reset)
                begin
                    Q[3] =0;
                    Q[2] =0;
                    Q[1] =0;
                    Q[0] =0;
                end
            else begin
                Q[3]=Q[2];
                Q[2]=Q[1];
                Q[1]=Q[0];
                Q[0]=(~Q[3]);
                end
        end
endmodule
```

2. Test Bench:

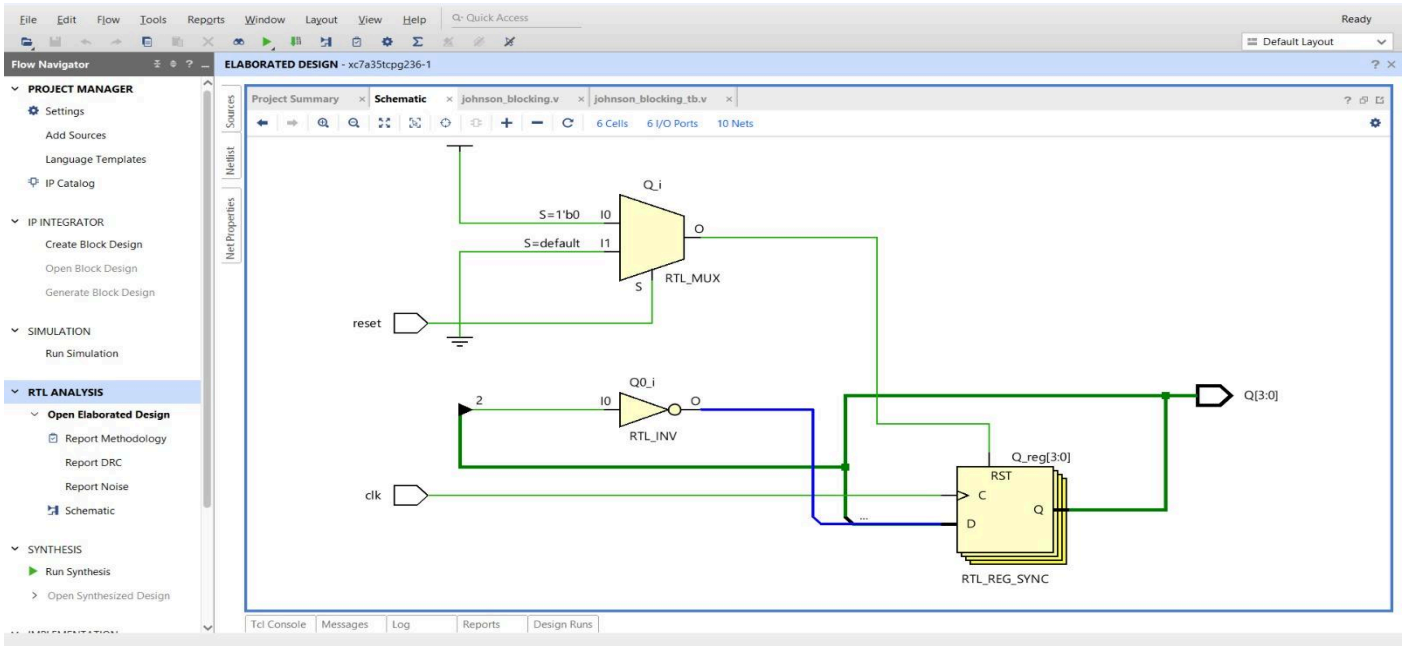
```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 24.01.2024 03:58:43
// Design Name:
// Module Name: johnson_blocking_tb
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
```

```
module johnson_blocking_tb();
    reg clk;
    reg reset;
    wire [3:0] Q;
    johnson_blocking uut(.clk(clk), .reset(reset), .Q(Q));
    initial
    begin
    clk = 1;
        forever #5 clk = ~clk;
    end
    initial
    begin
        reset = 0;
        #10;
        reset = 1;
        #160;
        reset = 0;
        #10;
        reset = 1;
        #160;
        $finish();
    end
endmodule
```

3. Simulation Results:



4. Circuit Diagram:



Observation:

For blocking assignment the order in which we took Q mattered and disrupted the counting which is not the case in non blocking.