Assignment_3_extra

==========

Name: Ahmed Nour

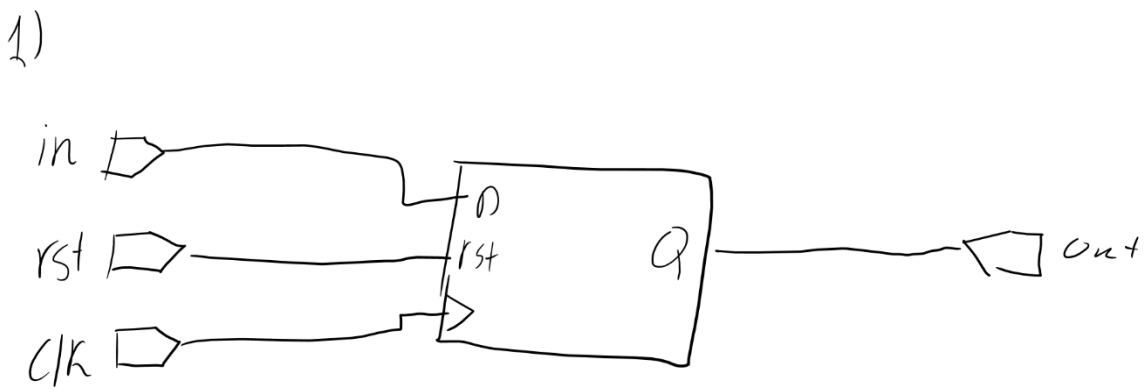
Sheet ID: 16

Email: ahmedmohamednoour@gmail.com

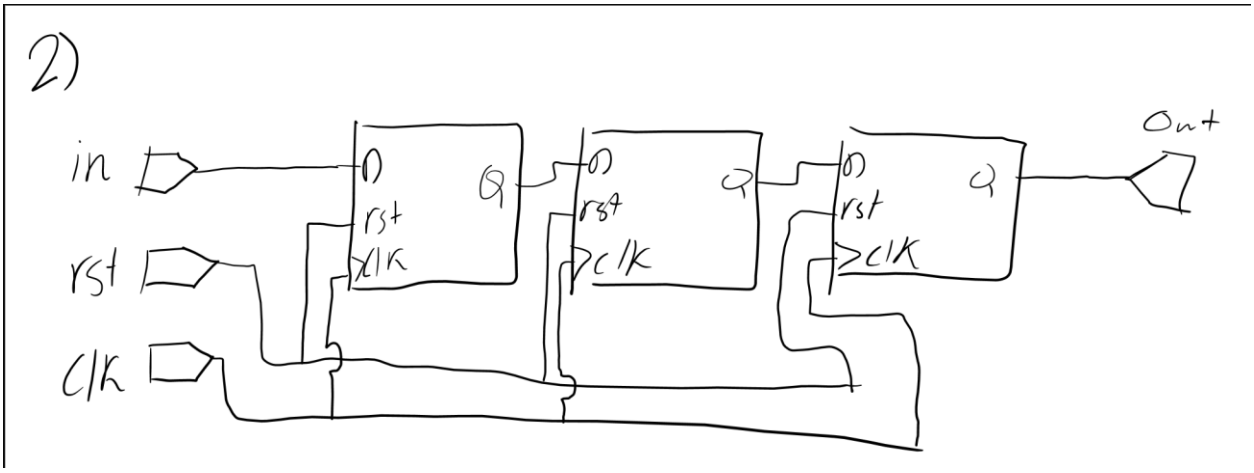
==========

Q1)

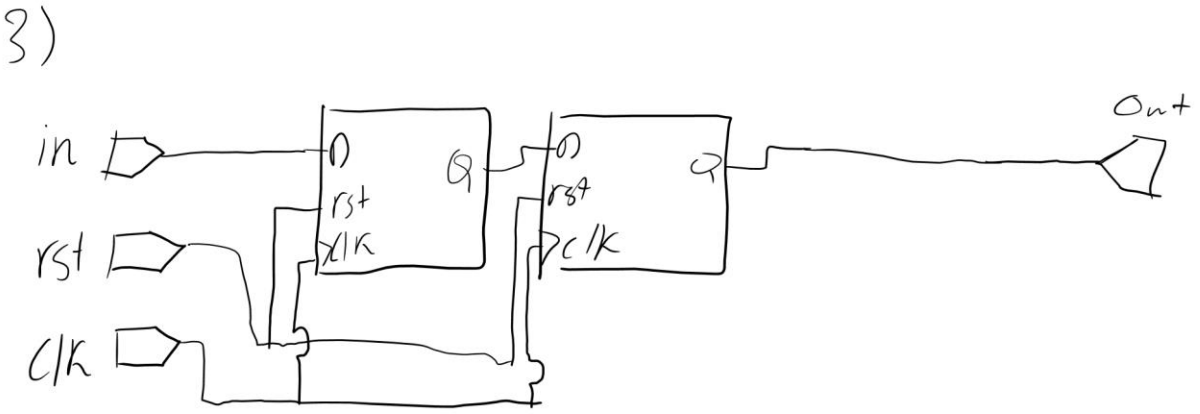
1)



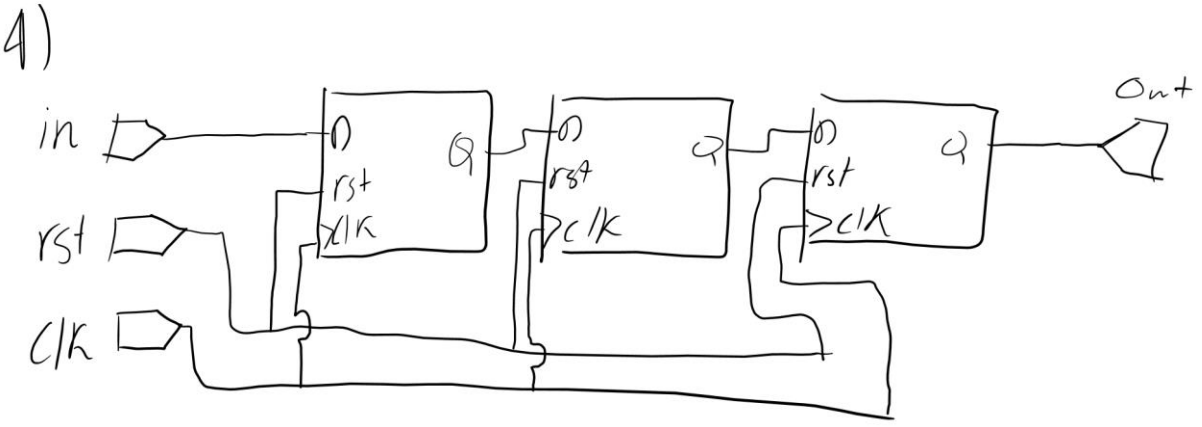
2)



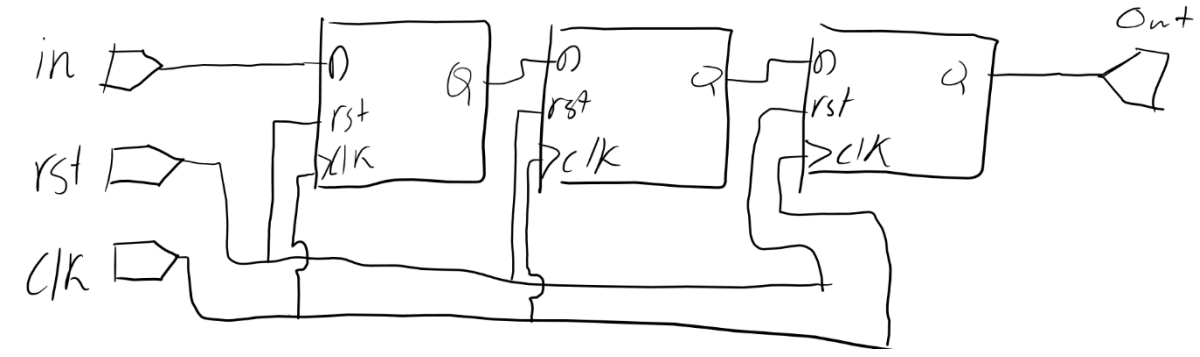
3)


3)
in
rst
clk
D
Q
rst
clk
D
Q
rst
clk
Out

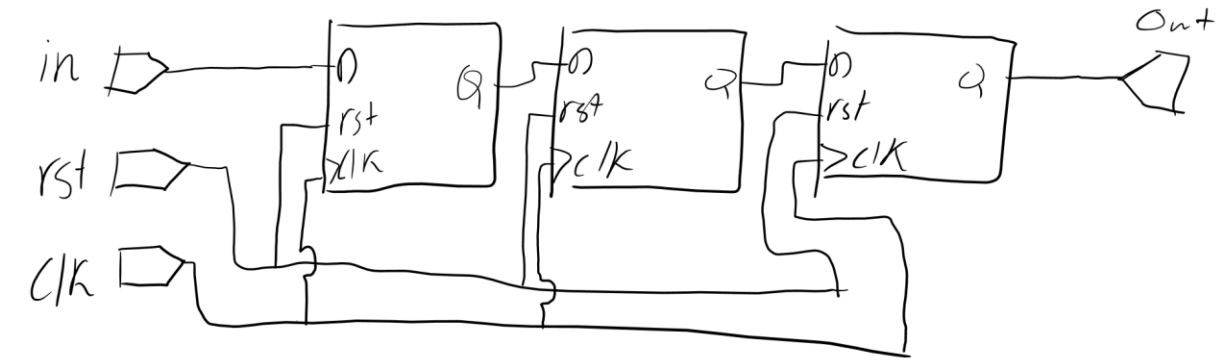
4)


4)
in
rst
clk
D
Q
rst
clk
D
Q
rst
clk
D
Q
rst
clk
Out

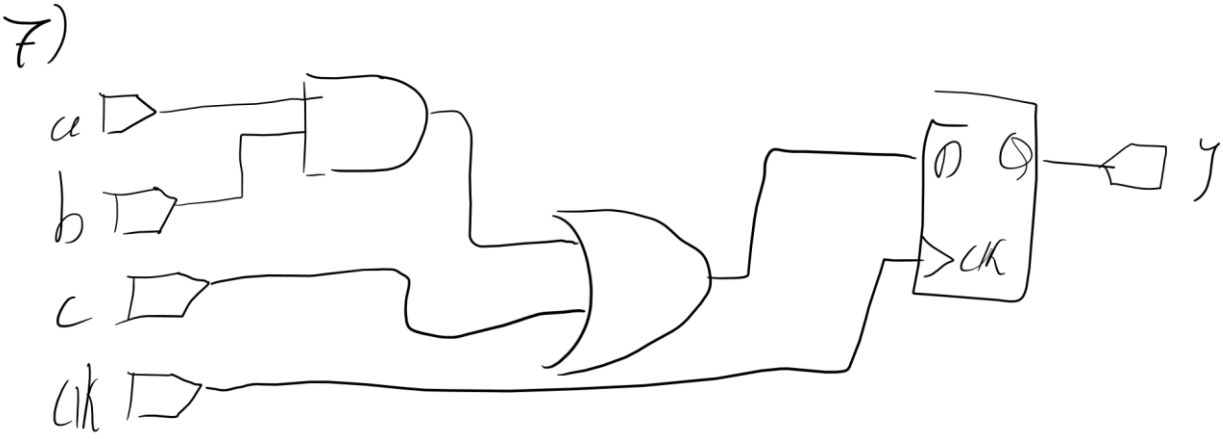
5)


5)
in
rst
clk
D
Q
rst
clk
D
Q
rst
clk
D
Q
rst
clk
Out

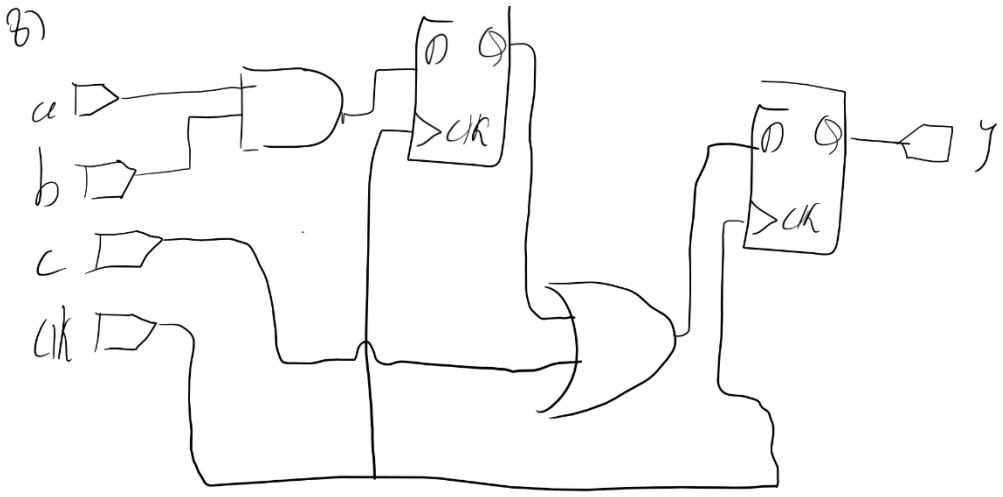
6)



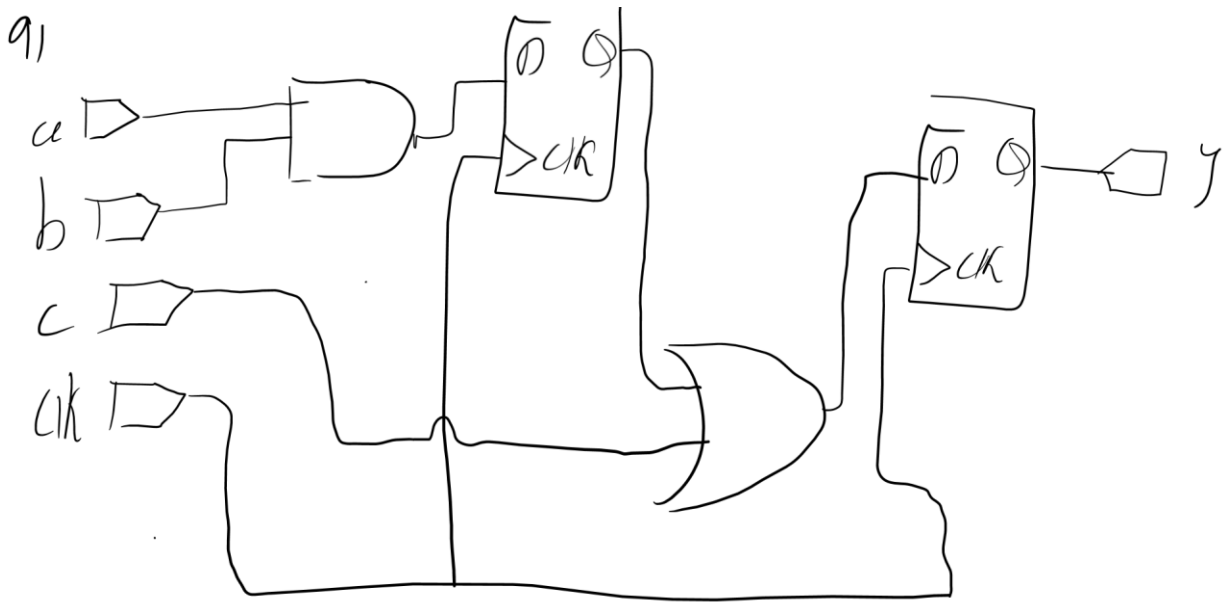
7)



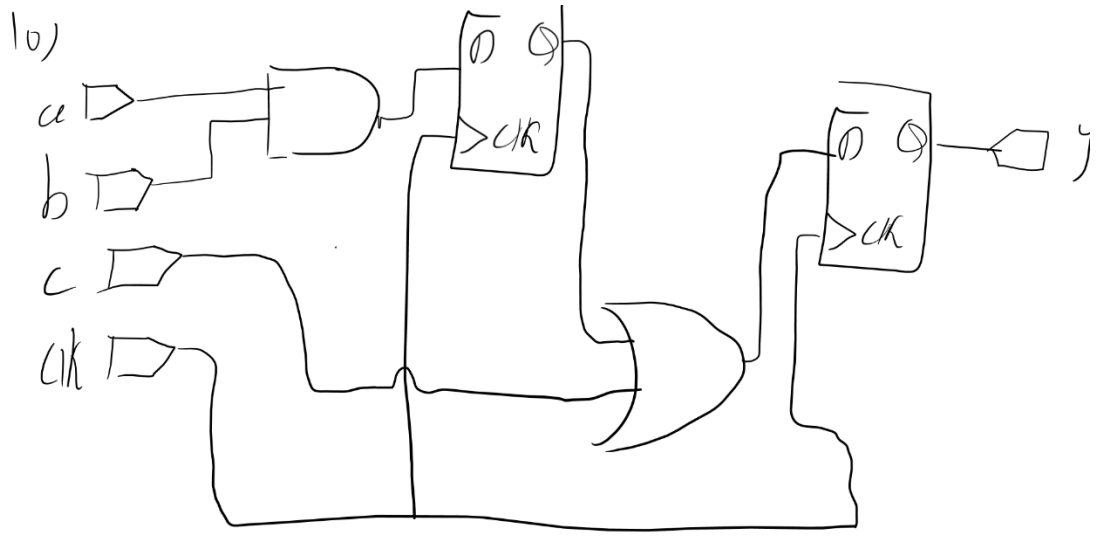
8)



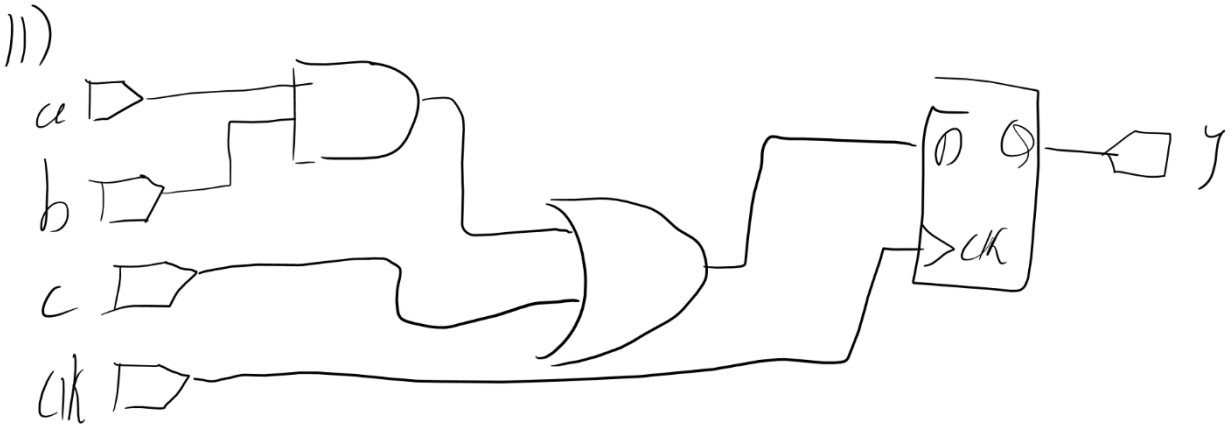
9)


9)
a
b
c
clk
D
Q
>clk
D
Q
>clk
y

10)


10)
a
b
c
clk
D
Q
>clk
D
Q
>clk
y

11)


11)
a
b
c
clk
D
Q
>clk
y

Q2)

Verilog code:

a) RTL design:

```verilog
module behavioral_counter (out, clk, set);
    input clk , set;
    output reg [3:0] out;

    always @(posedge clk or negedge set) begin
        if(!set)begin
            out <= 4'b1111;
        end
        else begin
            out <= out + 1;
        end
    end
endmodule
```

```verilog
module DFF(q ,qbar, d, rstn, clk);
    input d, rstn, clk;
    output reg q;
    output qbar;

    assign qbar = ~q;

    always @(posedge clk or negedge rstn) begin
        if(!rstn)
            q <= 0;
        else begin
            q <= d;
        end
    end

endmodule
```

```verilog
module ripple_counter_4bit(out, clk, rstn);
    input clk, rstn;
    output [3:0] out;

    wire qn0, qn1, qn2, qn3;
    wire q0, q1, q2, q3;

    DFF dff0 (q0 ,qn0, qn0, rstn, clk);
    DFF dff1 (q1 ,qn1, qn1, rstn, q0);
    DFF dff2 (q2 ,qn2, qn2, rstn, q1);
    DFF dff3 (q3 ,qn3, qn3, rstn, q2);

    assign out[3:0] = {qn3, qn2, qn1, qn0};

endmodule
```

b) Testbench code:

```
module behavioral_counter_tb ();
    reg clk_tb , set_tb;
    wire [3:0] out_tb;
    wire [3:0] out_GR;

    initial begin
        clk_tb = 0;
        forever begin
            #1 clk_tb = ~ clk_tb;
        end
    end

    behavioral_counter DUT (out_tb, clk_tb, set_tb);

    ripple_counter_4bit golden_ref (out_GR, clk_tb, set_tb);

    initial begin
        set_tb = 0;
        #2;
        @(negedge clk_tb);
        if (out_tb != out_GR) begin
            $display("error at out DUT: %d , out golden ratio: %d", out_tb, out_GR);
        end

        set_tb = 1;
        repeat(17)begin
            @(negedge clk_tb);
            if (out_tb != out_GR) begin
                $display("error at out DUT: %d , out golden ratio: %d", out_tb, out_GR);
            end
        end
        $display("test done");
        $finish;
    end

endmodule
```

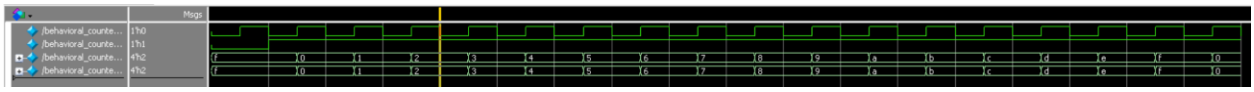
Simulation Tool

a) Do file

```
vlib work
vlog q2.v q2_tb.v ripcounter.v DFF.v
vsim -voptargs=+acc work.behavioral_counter_tb
add wave *
run -all
#quit -sim
```

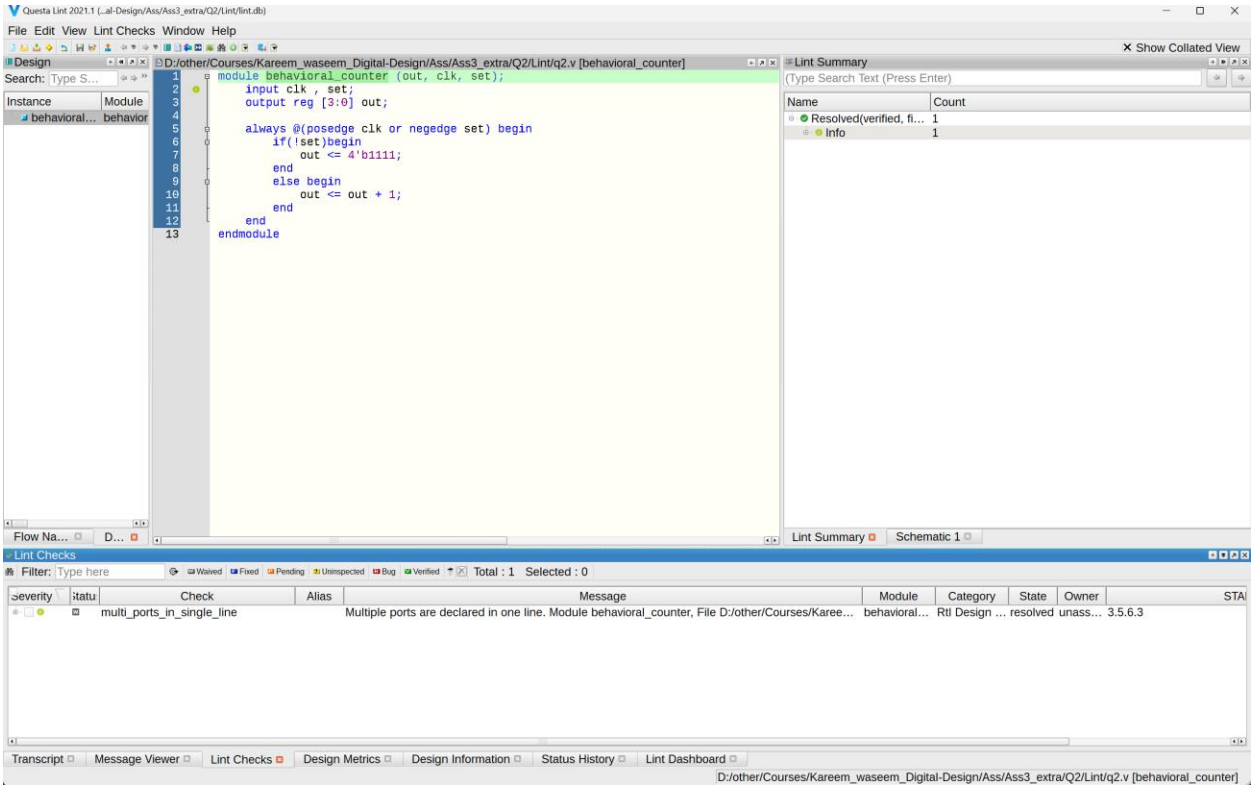
b) Questasim

```
# End time: 23:28:28 on Jul 25,2025, Elapsed time: 0:00:00
# Errors: 0, Warnings: 0
# vsim -voptargs="+acc" work.behavioral_counter_tb
# Start time: 23:28:28 on Jul 25,2025
# ** Note: (vsim-3813) Design is being optimized due to module recompilation...
# Loading work.behavioral_counter_tb(fast)
# Loading work.behavioral_counter(fast)
# Loading work.ripple_counter_4bit(fast)
# Loading work.DFF(fast)
# Loading work.DFF(fast__1)
# test done
# ** Note: $finish    : q2_tb.v(33)
#    Time: 36 ns  Iteration: 1  Instance: /behavioral_counter_tb
```

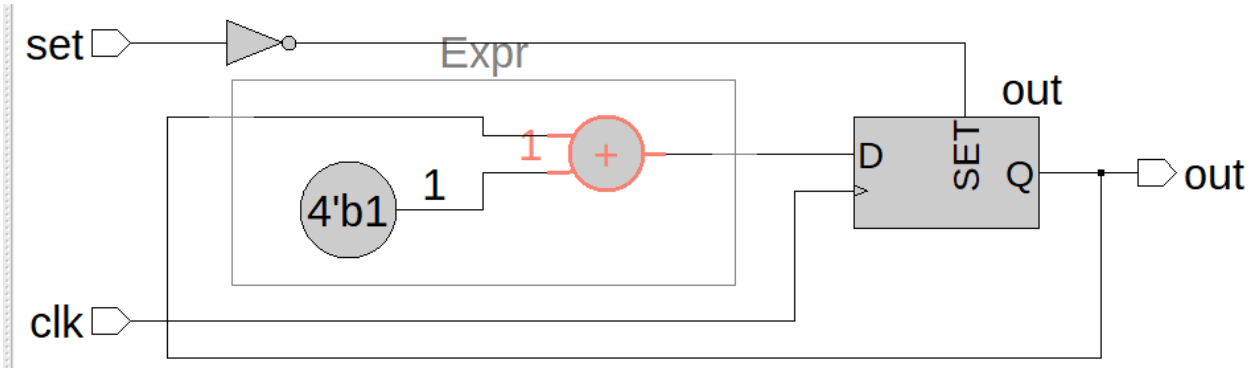


Lint tool

a) Linting



b) Schematic



Q3)

Verilog code:

c) RTL design:

```
module behavioral_counter_div (out, div_2, div_4, clk, set);
    input clk , set;
    output reg [3:0] out;
    output div_2, div_4;

    assign div_2 = (out[0] == 1)? 1 : 0;
    assign div_4 = (out[1] == 1)? 1: 0;

    always @(posedge clk or negedge set) begin
        if(!set)begin
            out <= 4'b1111;
        end
        else begin
            out <= out + 1;
        end
    end
endmodule
```

d) Testbench code:

```
module behavioral_counter_div_tb ();
    reg clk_tb , set_tb;
    wire [3:0] out_tb;
    wire div_2_tb, div_4_tb;
    reg div_2_exp, div_4_exp;


    behavioral_counter_div DUT (out_tb, div_2_tb, div_4_tb, clk_tb, set_tb);

    initial begin
        clk_tb = 0;
        forever begin
            #1 clk_tb = ~ clk_tb;
        end
    end


    initial begin
        div_2_exp = 0;
        div_4_exp = 0;
        set_tb = 0;
        @(negedge clk_tb);
        set_tb = 1;

        repeat(50)begin
            @(negedge clk_tb);
            div_2_exp = ~div_2_exp;
            #2;
            if (div_2_exp != div_2_tb) begin
                $display("error in div 2");
            end
        end

        $display("test done");
        $finish;
    end

    initial begin
        @(negedge clk_tb);
         repeat(50)begin
            @(negedge clk_tb);
            div_4_exp = ~div_4_exp;
            #4;
            if (div_4_exp != div_4_tb) begin
                $display("error in div 4");
            end
        end
    end

endmodule
```

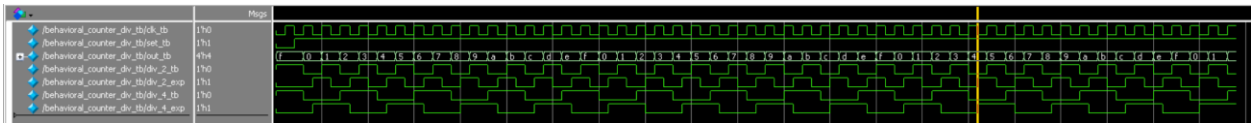
Simulation Tool

c) Do file

```
vlib work
vlog q3.v q3_tb.v
vsim -voptargs=+acc work.behavioral_counter_div_tb
add wave *
run -all
#quit -sim
```

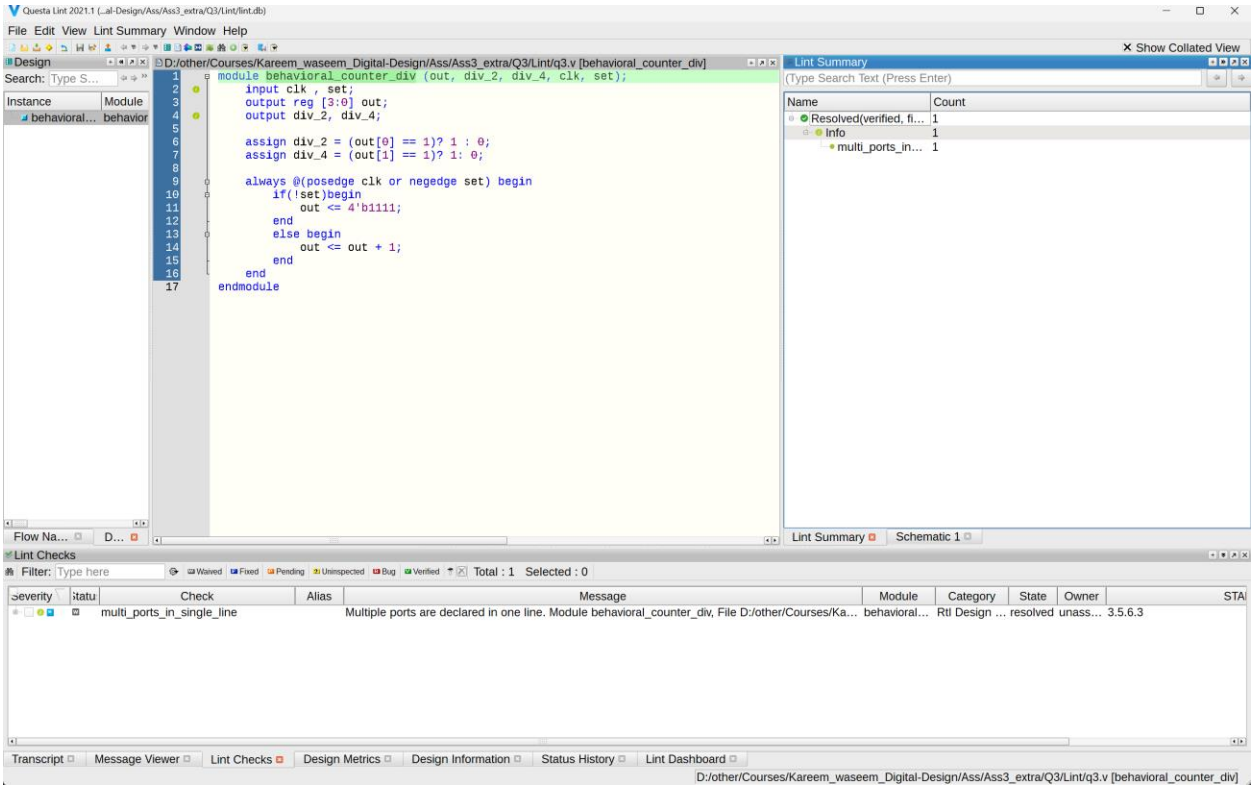
d) Questasim

```
# Top level modules:
#       behavioral_counter_div_tb
# End time: 16:28:05 on Jul 27,2025, Elapsed time: 0:00:00
# Errors: 0, Warnings: 0
# vsim -voptargs="+acc" work.behavioral_counter_div_tb
# Start time: 16:28:05 on Jul 27,2025
# ** Note: (vsim-3813) Design is being optimized due to module recompilation...
# Loading work.behavioral_counter_div_tb(fast)
# Loading work.behavioral_counter_div(fast)
# test done
# ** Note: $finish    : q3_tb.v(35)
#    Time: 104 ns  Iteration: 0  Instance: /behavioral_counter_div_tb
```

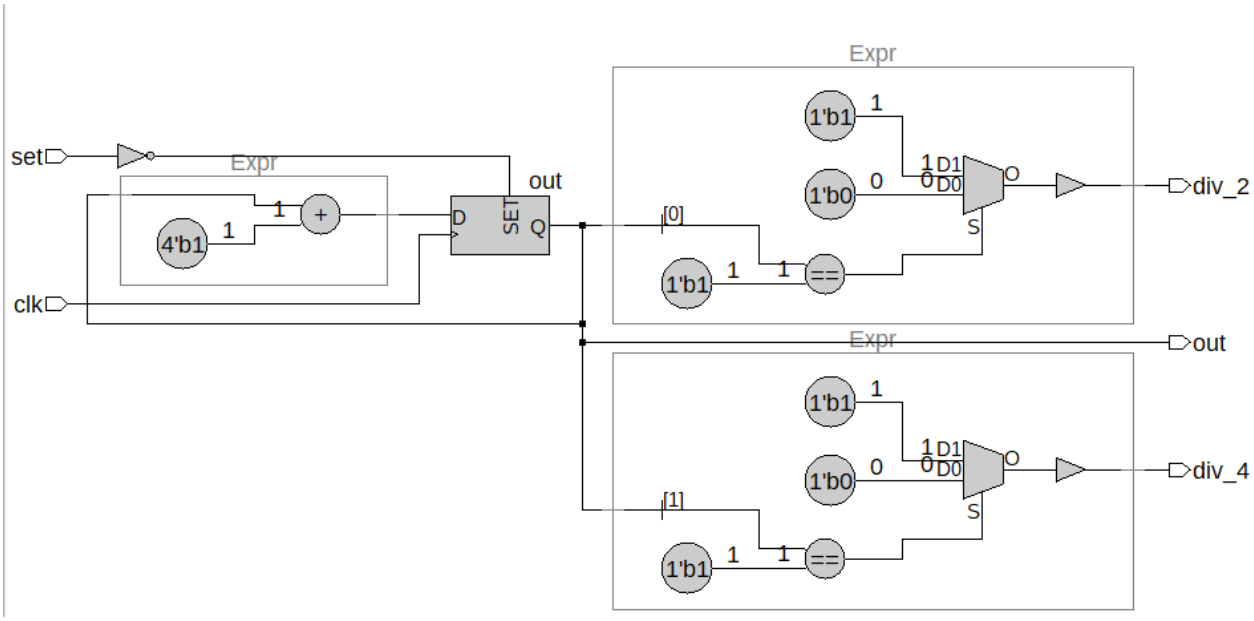


Lint tool

c) Linting



d) Schematic



Q4

Verilog code:

e) **RTL design:**

```
module gray_counter (gray_out, clk, rst);
    input clk, rst;
    output [1:0] gray_out;

    reg [1:0] counter;

    assign gray_out[1] = counter[1];
    assign gray_out[0] = ^counter;

    always @(posedge clk)begin
        if (rst)
            counter <= 0;
        else
            counter <= counter + 1;
    end

endmodule
```

f) Testbench code:

```
module gray_counter_tb ();
    reg clk_tb, rst_tb;
    wire [1:0] gray_out_tb;

    gray_counter DUT (gray_out_tb, clk_tb, rst_tb);

    initial begin
        clk_tb = 0;
        forever begin
            #1 clk_tb = ~clk_tb;
        end
    end

    initial begin
        rst_tb = 1;
        #3;
        rst_tb = 0;
        repeat(10)begin
            @(negedge clk_tb);
        end
        #5;
        rst_tb = 1;
        repeat(10)begin
            @(negedge clk_tb);
        end
        #5;
        rst_tb = 0;
        repeat(50)begin
            @(negedge clk_tb);
        end

        $finish;
    end


endmodule
```

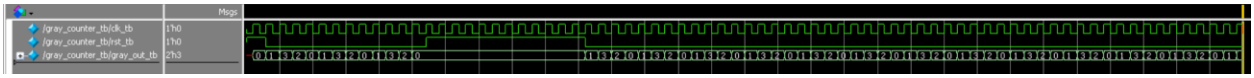
Simulation Tool

e) Do file

```
vlib work
vlog q4.v q4_tb.v
vsim -voptargs=+acc work.gray_counter_tb
add wave *
run -all
#quit -sim
```

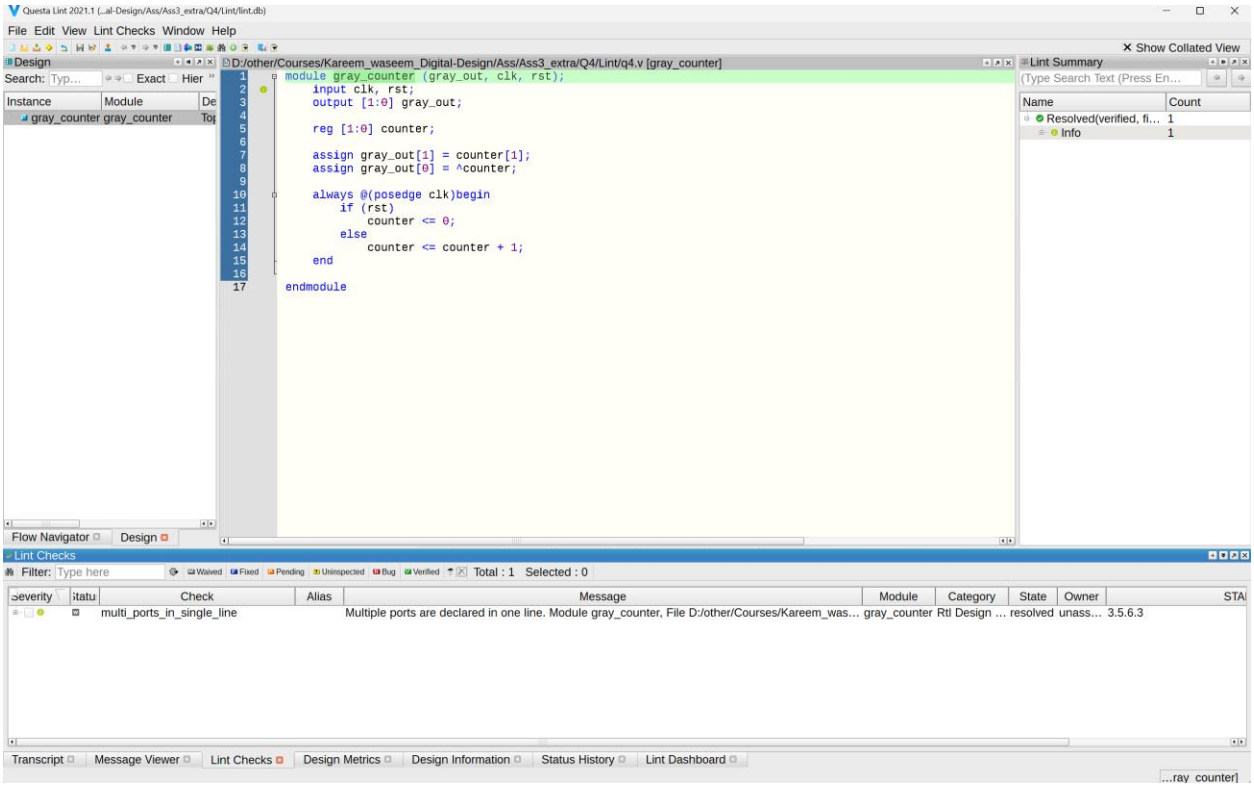
f) Questasim

```
# Top level modules:
#       gray_counter_tb
# End time: 16:49:58 on Jul 27,2025, Elapsed time: 0:00:00
# Errors: 0, Warnings: 0
# vsim -voptargs="+acc" work.gray_counter_tb
# Start time: 16:49:58 on Jul 27,2025
# ** Note: (vsim-3813) Design is being optimized due to module recompilation...
# Loading work.gray_counter_tb(fast)
# Loading work.gray_counter(fast)
# ** Note: $finish    : q4_tb.v(32)
#    Time: 150 ns  Iteration: 1  Instance: /gray_counter_tb
```



Lint tool

e) Linting



f) Schematic