

Assignment_3_extra

Name: Ahmed Nour

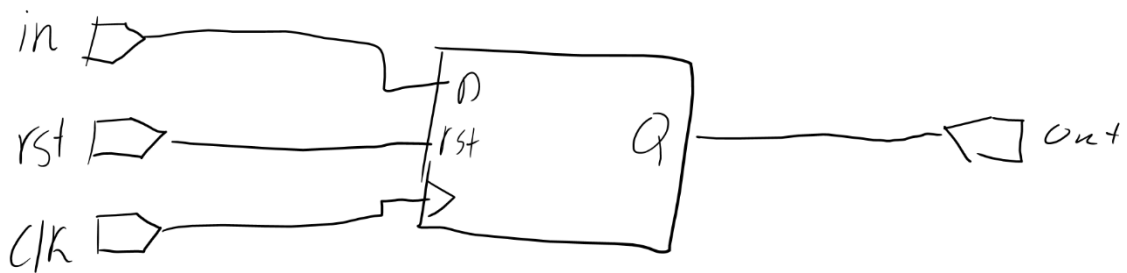
Sheet ID: 16

Email: ahmedmohamednoor@gmail.com

Q1)

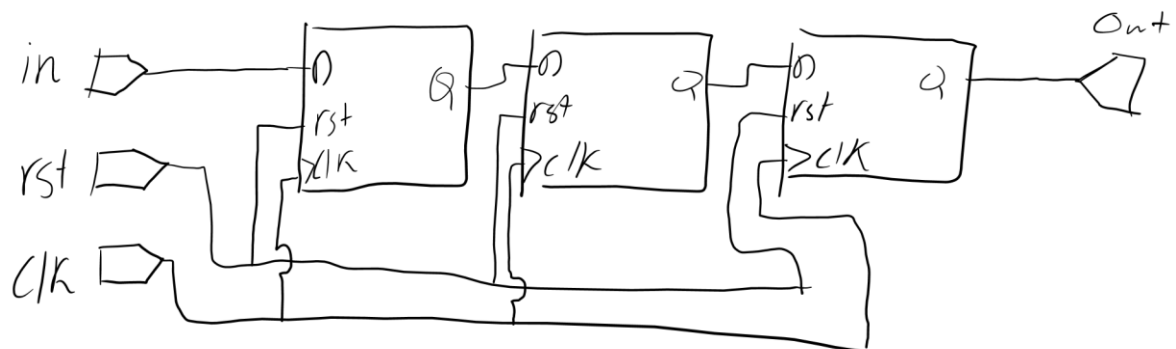
1)

1)



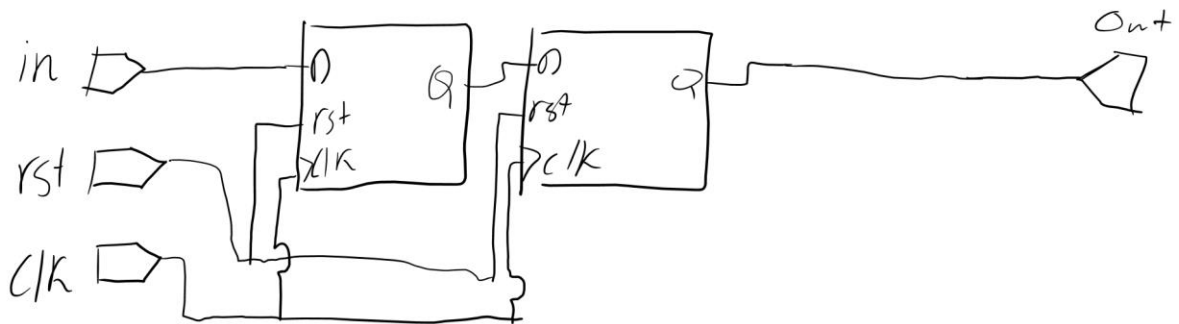
2)

2)



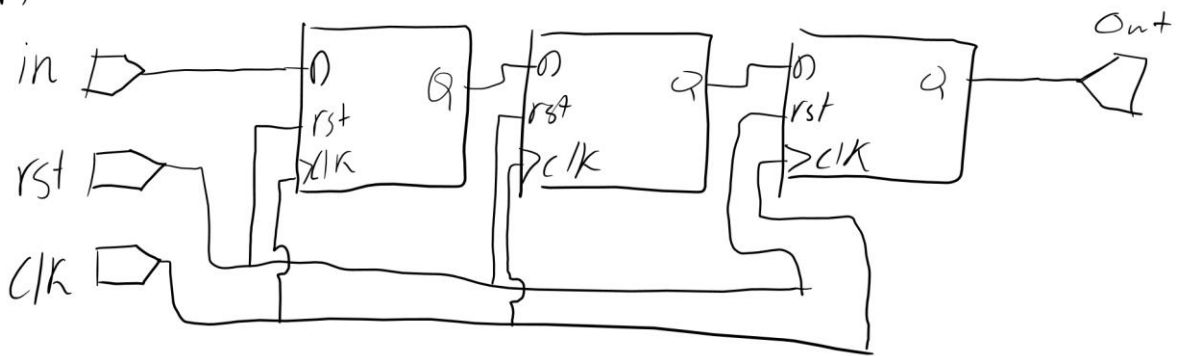
3)

3)



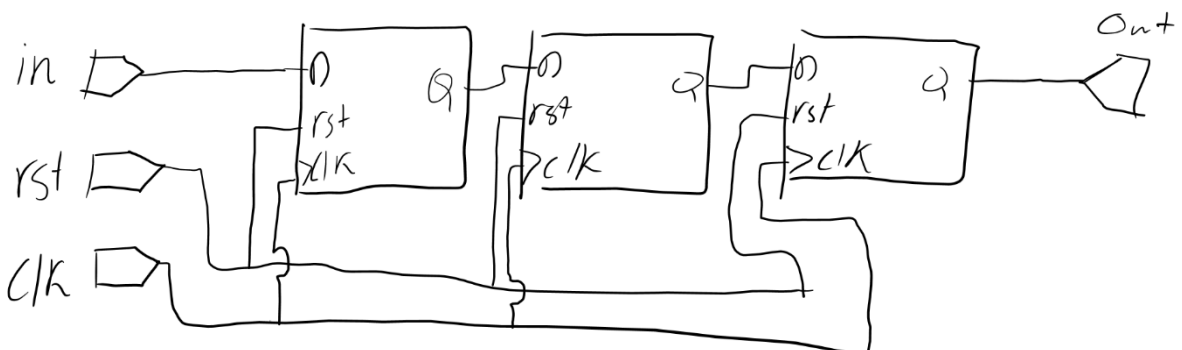
4)

4)



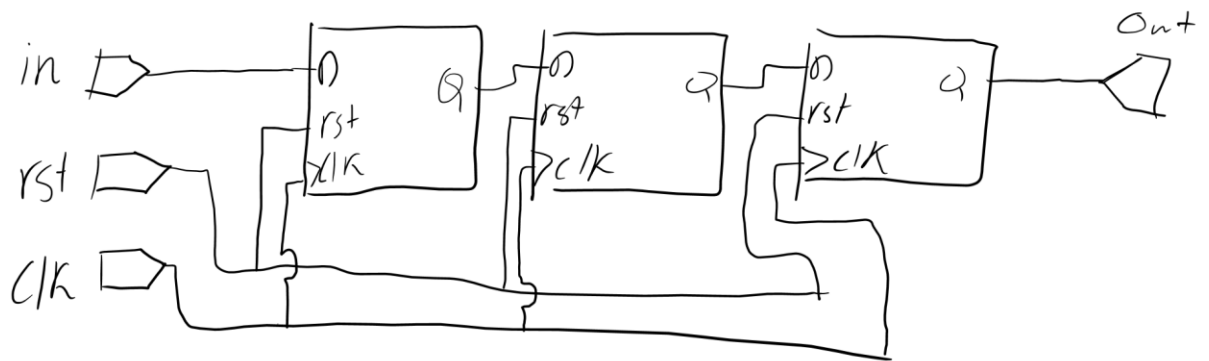
5)

5)



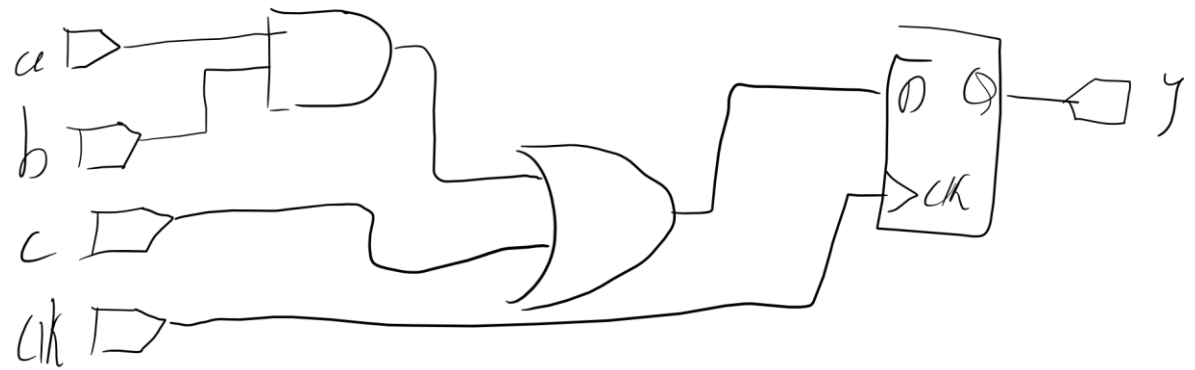
6)

6)



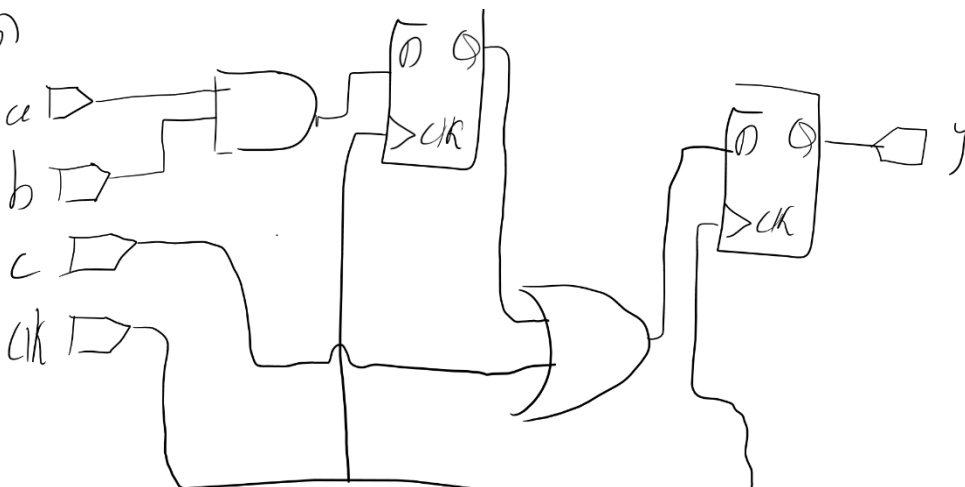
7)

7)

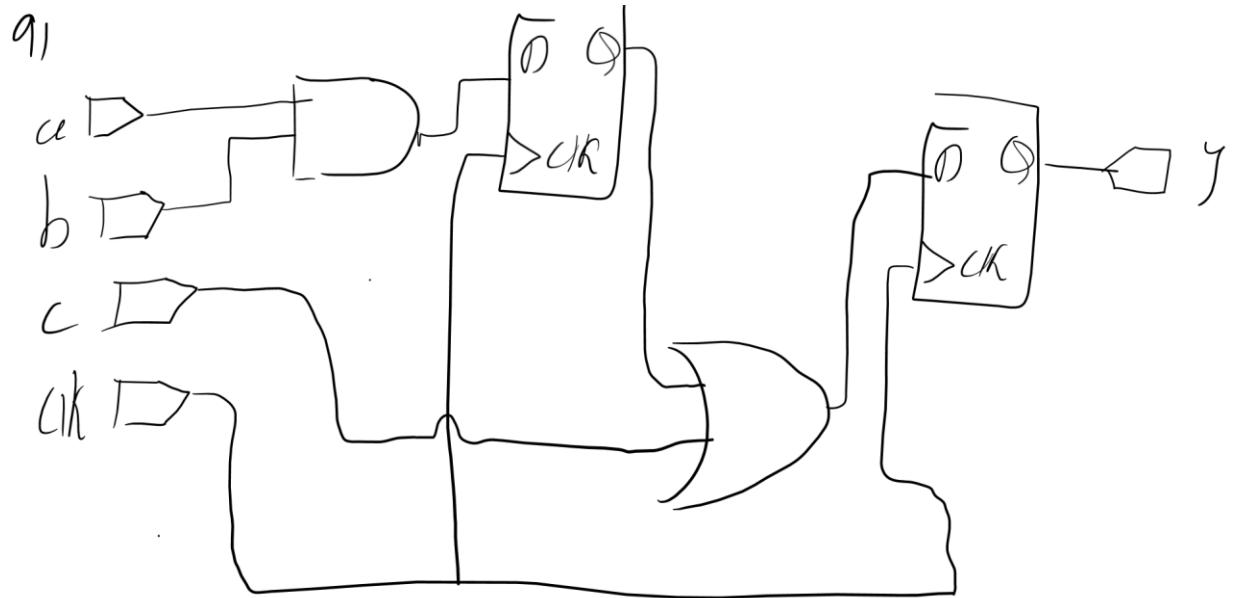


8)

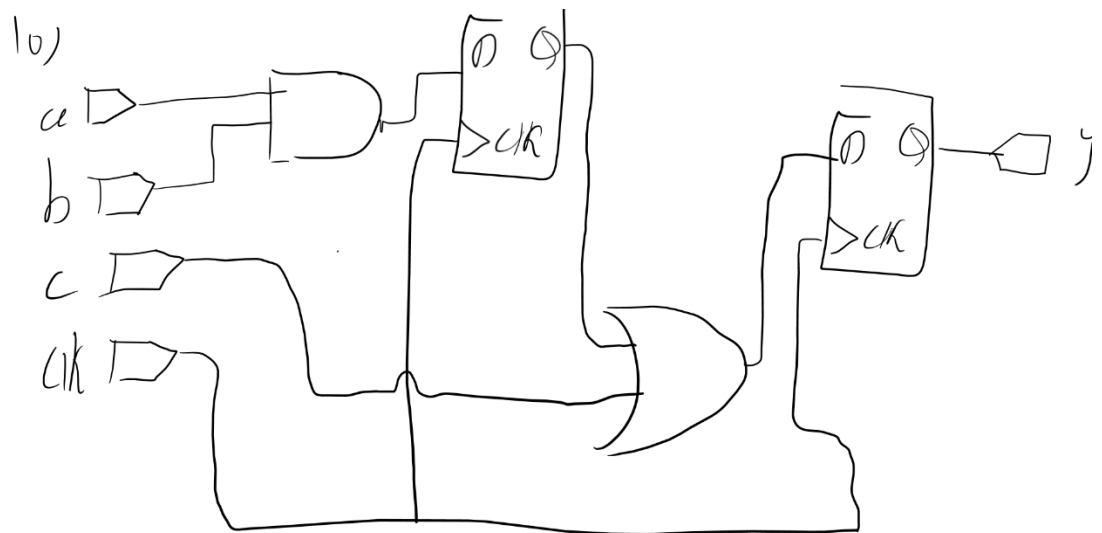
8)



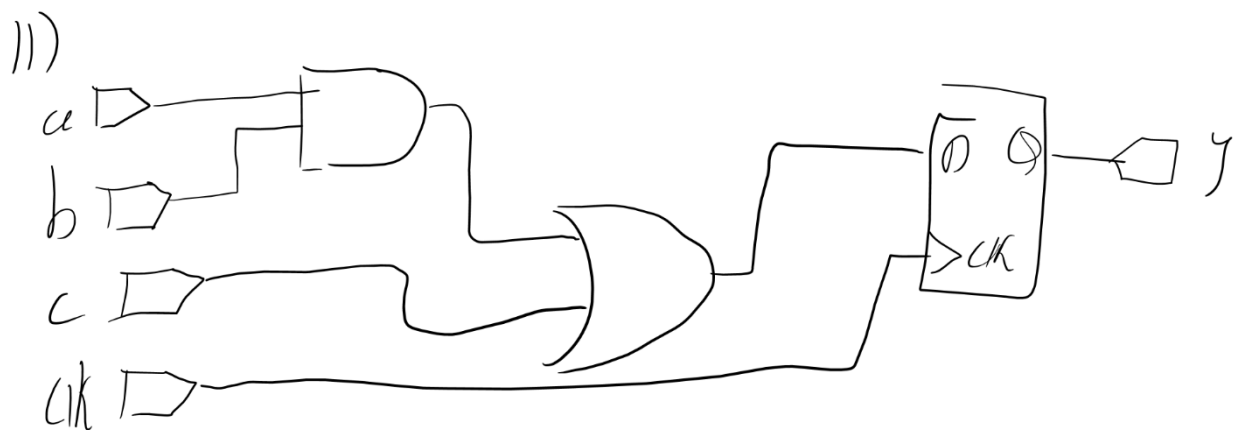
9)



10)



11)



Q2)

Verilog code:

a) RTL design:

```
1 module behavioral_counter (out, clk, set);
2     input clk , set;
3     output reg [3:0] out;
4
5     always @(posedge clk or negedge set) begin
6         if(!set)begin
7             out <= 4'b1111;
8         end
9         else begin
10            out <= out + 1;
11        end
12    end
13 endmodule
```

```
1 module DFF(q ,qbar, d, rstn, clk);
2     input d, rstn, clk;
3     output reg q;
4     output qbar;
5
6     assign qbar = ~q;
7
8     always @(posedge clk or negedge rstn) begin
9         if(!rstn)
10            q <= 0;
11        else begin
12            q <= d;
13        end
14    end
15
16 endmodule
```

```
1 module ripple_counter_4bit(out, clk, rstn);
2     input clk, rstn;
3     output [3:0] out;
4
5     wire qn0, qn1, qn2, qn3;
6     wire q0, q1, q2, q3;
7
8     DFF dff0 (q0 ,qn0, qn0, rstn, clk);
9     DFF dff1 (q1 ,qn1, qn1, rstn, q0);
10    DFF dff2 (q2 ,qn2, qn2, rstn, q1);
11    DFF dff3 (q3 ,qn3, qn3, rstn, q2);
12
13    assign out[3:0] = {qn3, qn2, qn1, qn0};
14
15 endmodule
```

b) Testbench code:

```
1 module behavioral_counter_tb ();
2     reg clk_tb , set_tb;
3     wire [3:0] out_tb;
4     wire [3:0] out_GR;
5
6     initial begin
7         clk_tb = 0;
8         forever begin
9             #1 clk_tb = ~ clk_tb;
10        end
11    end
12
13    behavioral_counter DUT (out_tb, clk_tb, set_tb);
14
15    ripple_counter_4bit golden_ref (out_GR, clk_tb, set_tb);
16
17    initial begin
18        set_tb = 0;
19        #2;
20        @(negedge clk_tb);
21        if (out_tb != out_GR) begin
22            $display("error at out DUT: %d , out golden ratio: %d", out_tb, out_GR);
23        end
24
25        set_tb = 1;
26        repeat(17)begin
27            @(negedge clk_tb);
28            if (out_tb != out_GR) begin
29                $display("error at out DUT: %d , out golden ratio: %d", out_tb, out_GR);
30            end
31        end
32        $display("test done");
33        $finish;
34    end
35
36 endmodule
```

Simulation Tool

a) Do file

```
1 vlib work
2 vlog q2.v q2_tb.v ripcounter.v DFF.v
3 vsim -voptargs=+acc work.behavioral_counter_tb
4 add wave *
5 run -all
6 #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: ifinish : q2_tb.v(33)
# Time: 36 ns Iteration: 1 Instance: /behavioral_counter_tb
```

		Haps																	
/behavioral_counte...	th0																		
/behavioral_counte...	th1																		
/behavioral_counte...	th2																		
/behavioral_counte...	th2																		
		t	10	11	12	13	14	15	16	17	18	19	1a	1b	1c	1d	1e	1f	1a
		t	10	11	12	13	14	15	16	17	18	19	1a	1b	1c	1d	1e	1f	1a

Lint tool

a) Linting

The screenshot shows the Questa Lint 2021.1 interface. The main window displays the source code for a behavioral counter module. The code is as follows:

```
1 module behavioral_counter (out, clk, set);
2   input clk, set;
3   output reg [3:0] out;
4
5   always @(posedge clk or negedge set) begin
6     if (!set) begin
7       out <= 4'b1111;
8     end
9     else begin
10      out <= out + 1;
11    end
12  end
13 endmodule
```

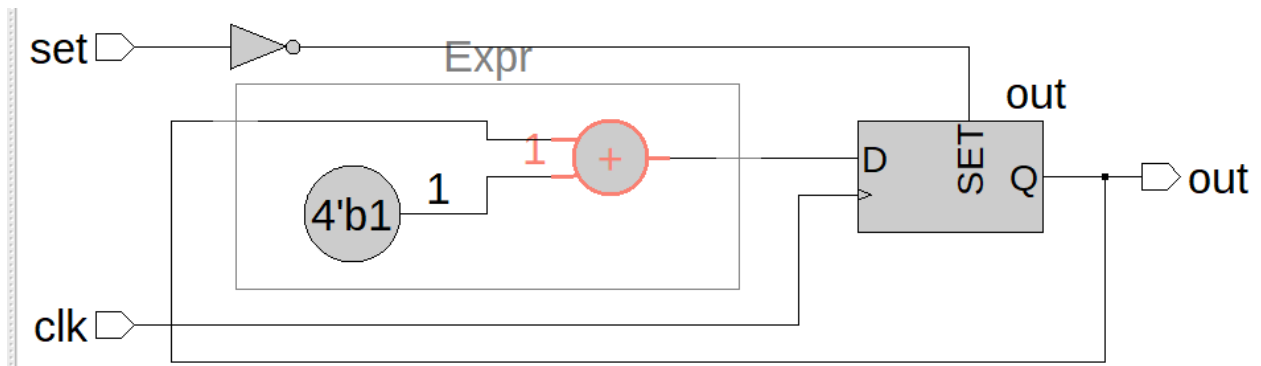
The Lint Summary window on the right shows the following results:

Name	Count
Resolved(verified, fl...	1
Info	1

The Lint Checks window at the bottom shows a single lint check:

Severity	Status	Check	Alias	Message	Module	Category	State	Owner	STA
Info	Resolved	multi_ports_in_single_line		Multiple ports are declared in one line. Module behavioral_counter, File D:/other/Courses/Kareem...	behavioral...	Rtl Design ... resolved unass...	3.5.6.3		

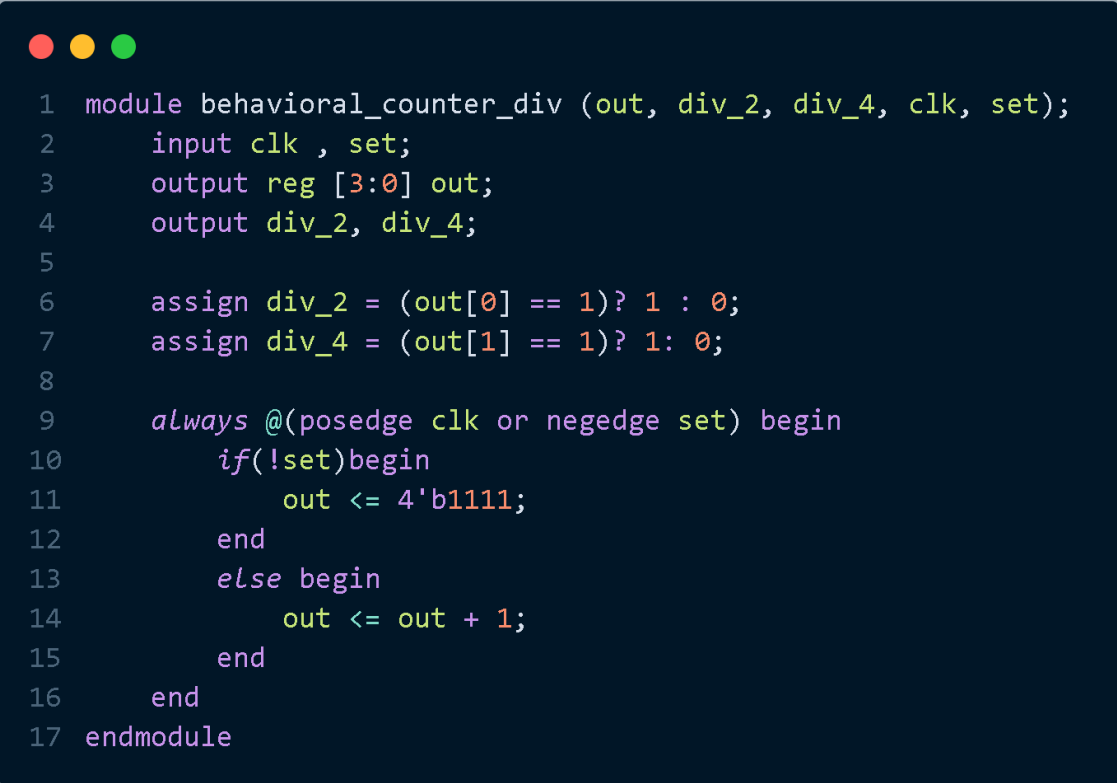
b) Schematic



Q3)

Verilog code:

c) RTL design:

A screenshot of a Verilog code editor window with a dark blue background and light-colored text. The code is a Verilog module named 'behavioral_counter_div'. It has four inputs: 'out', 'div_2', 'div_4', 'clk', and 'set'. The 'out' input is a 4-bit register. The 'div_2' and 'div_4' outputs are 1-bit signals. The code uses an 'always' block triggered by the positive edge of 'clk' or the negative edge of 'set'. Inside the 'always' block, there is an 'if' statement that checks if 'set' is 1. If 'set' is 1, 'out' is assigned the value 4'b1111. If 'set' is 0, 'out' is incremented by 1. The code is as follows:

```
1 module behavioral_counter_div (out, div_2, div_4, clk, set);
2     input clk , set;
3     output reg [3:0] out;
4     output div_2, div_4;
5
6     assign div_2 = (out[0] == 1)? 1 : 0;
7     assign div_4 = (out[1] == 1)? 1: 0;
8
9     always @(posedge clk or negedge set) begin
10         if(!set)begin
11             out <= 4'b1111;
12         end
13         else begin
14             out <= out + 1;
15         end
16     end
17 endmodule
```

d) Testbench code:

```
1 module behavioral_counter_div_tb ();
2     reg clk_tb , set_tb;
3     wire [3:0] out_tb;
4     wire div_2_tb, div_4_tb;
5     reg div_2_exp, div_4_exp;
6
7
8     behavioral_counter_div DUT (out_tb, div_2_tb, div_4_tb, clk_tb, set_tb);
9
10    initial begin
11        clk_tb = 0;
12        forever begin
13            #1 clk_tb = ~ clk_tb;
14        end
15    end
16
17
18    initial begin
19        div_2_exp = 0;
20        div_4_exp = 0;
21        set_tb = 0;
22        @(negedge clk_tb);
23        set_tb = 1;
24
25        repeat(50)begin
26            @(negedge clk_tb);
27            div_2_exp = ~div_2_exp;
28            #2;
29            if (div_2_exp != div_2_tb) begin
30                $display("error in div 2");
31            end
32        end
33
34        $display("test done");
35        $finish;
36    end
37
38    initial begin
39        @(negedge clk_tb);
40        repeat(50)begin
41            @(negedge clk_tb);
42            div_4_exp = ~div_4_exp;
43            #4;
44            if (div_4_exp != div_4_tb) begin
45                $display("error in div 4");
46            end
47        end
48    end
49
50 endmodule
```

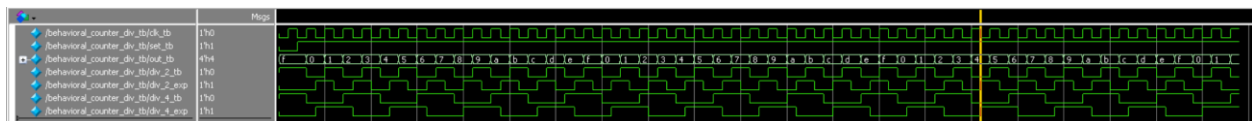
Simulation Tool

c) Do file

```
1 vlib work
2 vlog q3.v q3_tb.v
3 vsim -voptargs="+acc" work.behavioral_counter_div_tb
4 add wave *
5 run -all
6 #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

The screenshot shows the Questa Lint 2021.1 interface. The main window displays a Verilog module named `behavioral_counter_div` with the following code:

```
1 module behavioral_counter_div (out, div_2, div_4, clk, set);
2   input clk, set;
3   output reg [3:0] out;
4   output div_2, div_4;
5
6   assign div_2 = (out[0] == 1)? 1 : 0;
7   assign div_4 = (out[1] == 1)? 1 : 0;
8
9   always @(posedge clk or negedge set) begin
10     if(!set)begin
11       out <= 4'b1111;
12     end
13     else begin
14       out <= out + 1;
15     end
16   end
17 endmodule
```

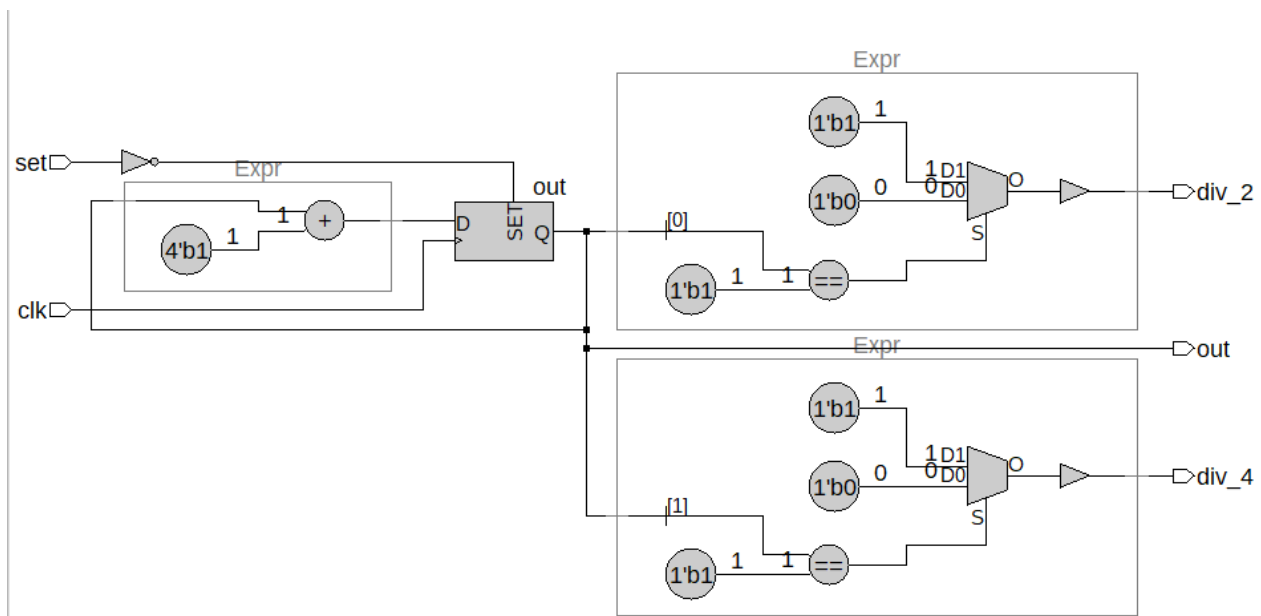
The Lint Summary window on the right shows the following results:

Name	Count
Resolved(verified, fl...	1
Info	1
multi_ports_in...	1

The Lint Checks window at the bottom shows a single lint message:

Severity	Check	Alias	Message	Module	Category	State	Owner	STA
Info	multi_ports_in_single_line		Multiple ports are declared in one line. Module behavioral_counter_div, File D:/other/Courses/Kareem_waseem_Digital-Design/Ass3_extra/Q3/Lint/q3.v [behavioral_counter_div]	behavioral...	Rtl Design ... resolved unass...	3.5.6.3		

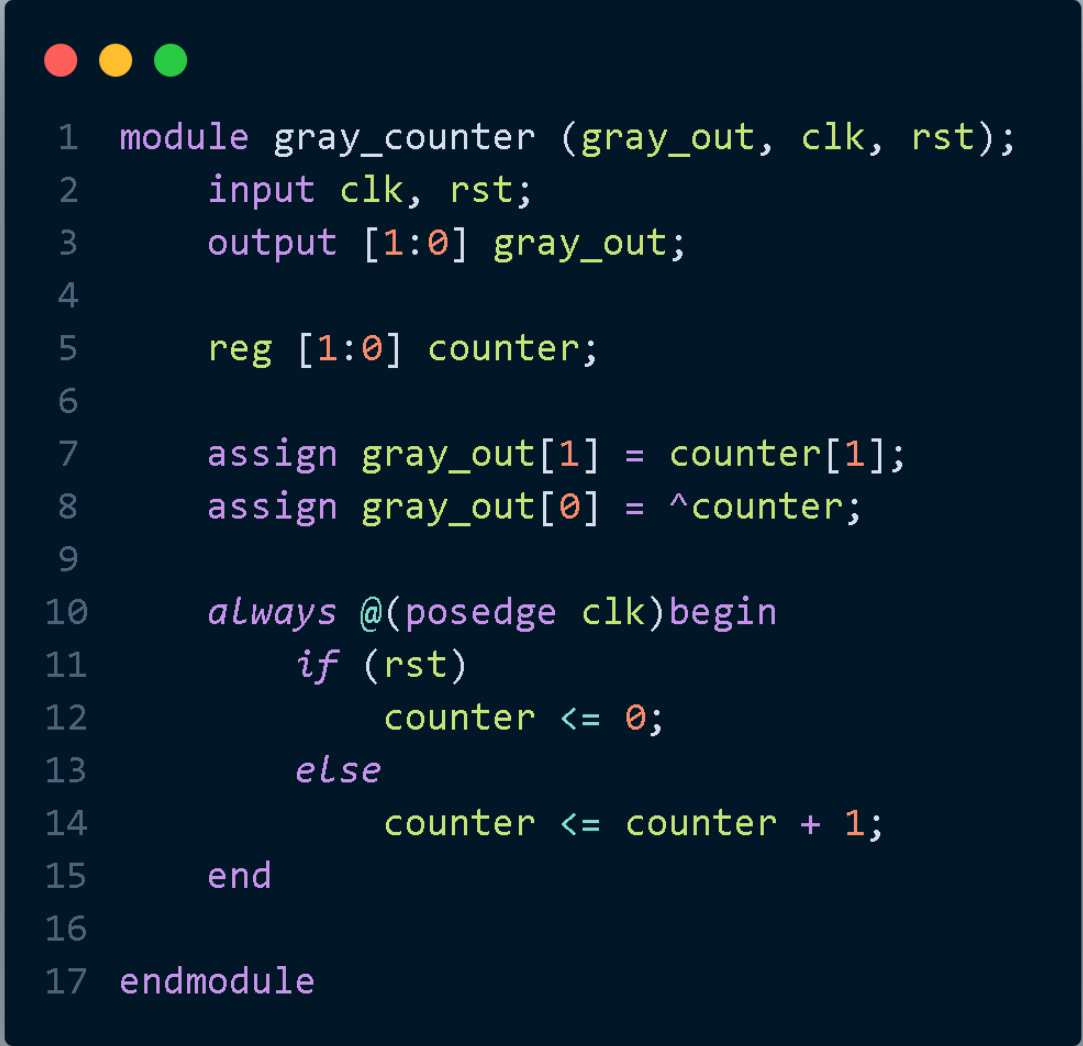
d) Schematic



Q4

Verilog code:

e) RTL design:



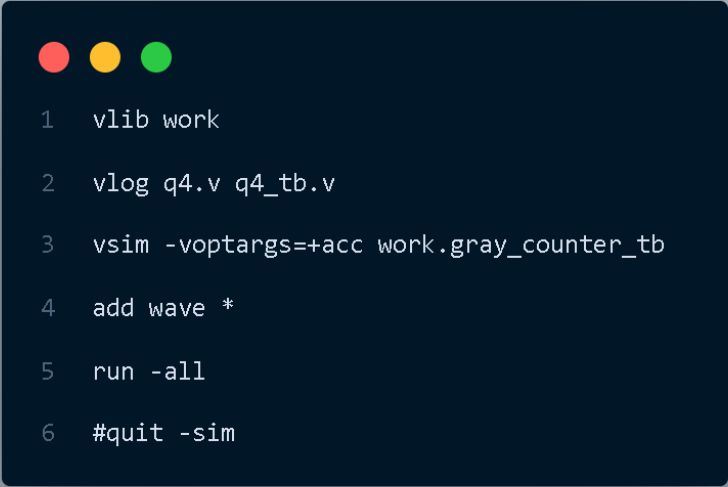
```
1  module gray_counter (gray_out, clk, rst);
2      input clk, rst;
3      output [1:0] gray_out;
4
5      reg [1:0] counter;
6
7      assign gray_out[1] = counter[1];
8      assign gray_out[0] = ^counter;
9
10     always @(posedge clk)begin
11         if (rst)
12             counter <= 0;
13         else
14             counter <= counter + 1;
15     end
16
17 endmodule
```

f) Testbench code:

```
1  module gray_counter_tb ();
2      reg clk_tb, rst_tb;
3      wire [1:0] gray_out_tb;
4
5      gray_counter DUT (gray_out_tb, clk_tb, rst_tb);
6
7      initial begin
8          clk_tb = 0;
9          forever begin
10             #1 clk_tb = ~clk_tb;
11         end
12     end
13
14     initial begin
15         rst_tb = 1;
16         #3;
17         rst_tb = 0;
18         repeat(10)begin
19             @(negedge clk_tb);
20         end
21         #5;
22         rst_tb = 1;
23         repeat(10)begin
24             @(negedge clk_tb);
25         end
26         #5;
27         rst_tb = 0;
28         repeat(50)begin
29             @(negedge clk_tb);
30         end
31
32         $finish;
33     end
34
35
36 endmodule
```

Simulation Tool

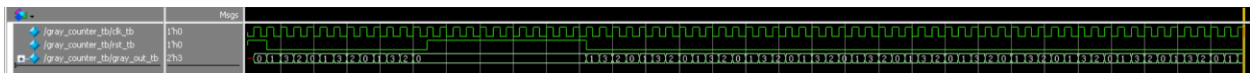
e) Do file



```
1 vlib work
2 vlog q4.v q4_tb.v
3 vsim -voptargs="+acc" work.gray_counter_tb
4 add wave *
5 run -all
6 #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

The screenshot shows the Questa Lint 2021.1 interface. The main window displays a Verilog module named `gray_counter` with the following code:

```
1 module gray_counter (gray_out, clk, rst);
2   input clk, rst;
3   output [1:0] gray_out;
4
5   reg [1:0] counter;
6
7   assign gray_out[1] = counter[1];
8   assign gray_out[0] = ^counter;
9
10  always @(posedge clk) begin
11    if (rst)
12      counter <= 0;
13    else
14      counter <= counter + 1;
15  end
16
17 endmodule
```

The right-hand pane shows the Lint Summary:

Name	Count
Resolved/verified, fl...	1
Info	1

The bottom pane shows the Lint Checks table:

Severity	Status	Check	Alias	Message	Module	Category	State	Owner	STA
Info	Resolved	multi_ports_in_single_line		Multiple ports are declared in one line. Module gray_counter, File D:/other/Courses/Kareem_was... gray_counter Rtl Design ... resolved unass... 3.5.6.3	gray_counter	Rtl Design	Resolved		

f) Schematic

