Assignment_3

==

Name: Ahmed Nour

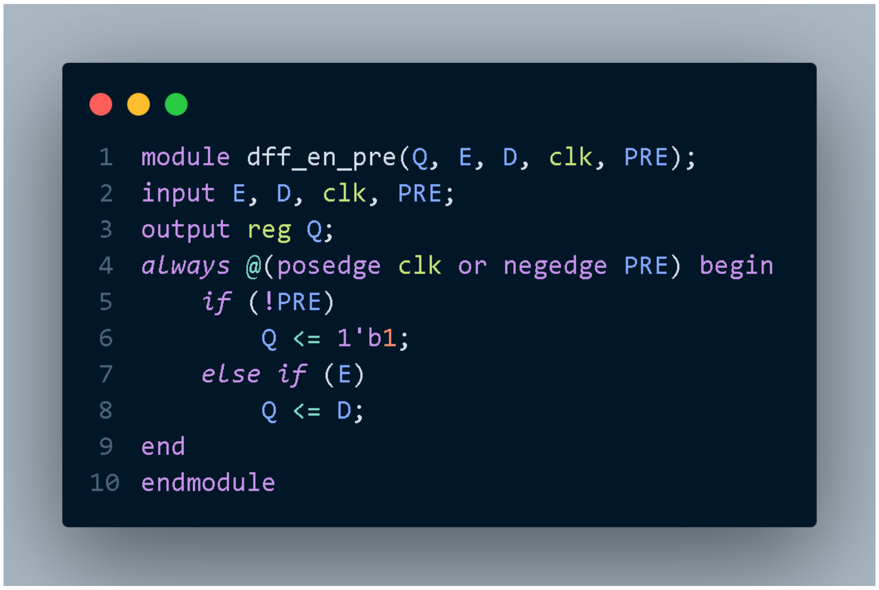
Sheet ID: 16

Email: ahmedmohamednoour@gmail.com

==

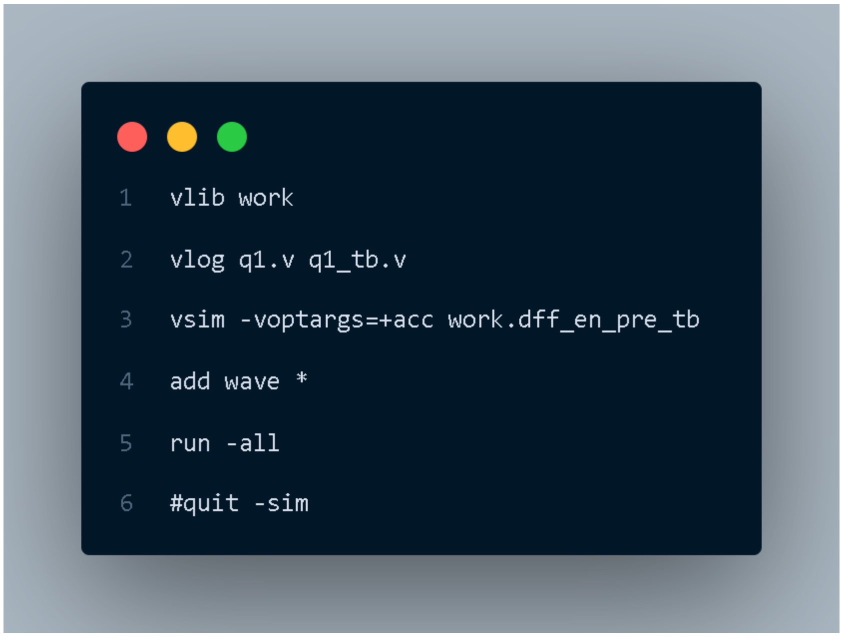
Q1)

Main Code:



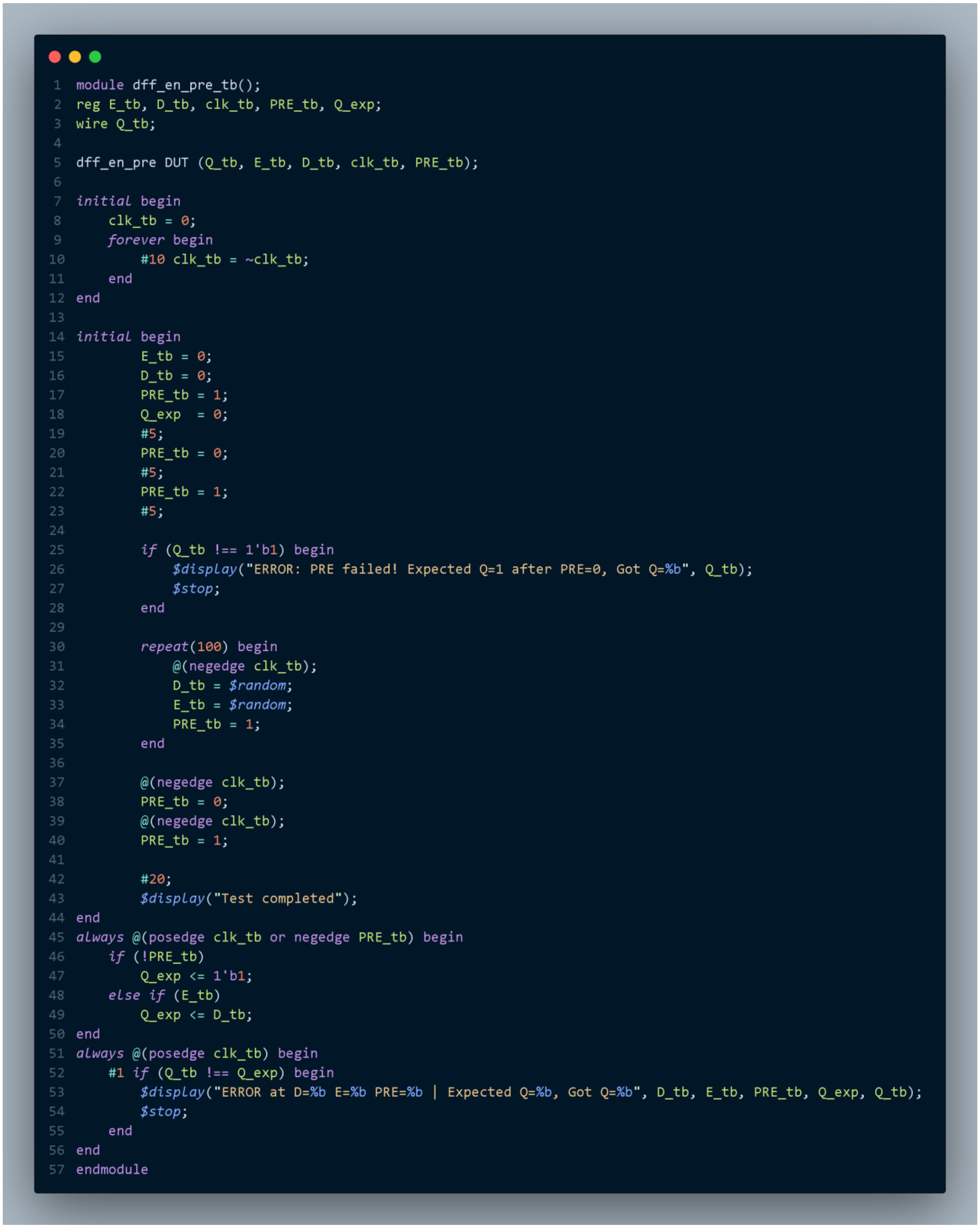
```
module dff_en_pre(Q, E, D, clk, PRE);
input E, D, clk, PRE;
output reg Q;
always @(posedge clk or negedge PRE) begin
    if (!PRE)
        Q <= 1'b1;
    else if (E)
        Q <= D;
end
endmodule
```

Do file:



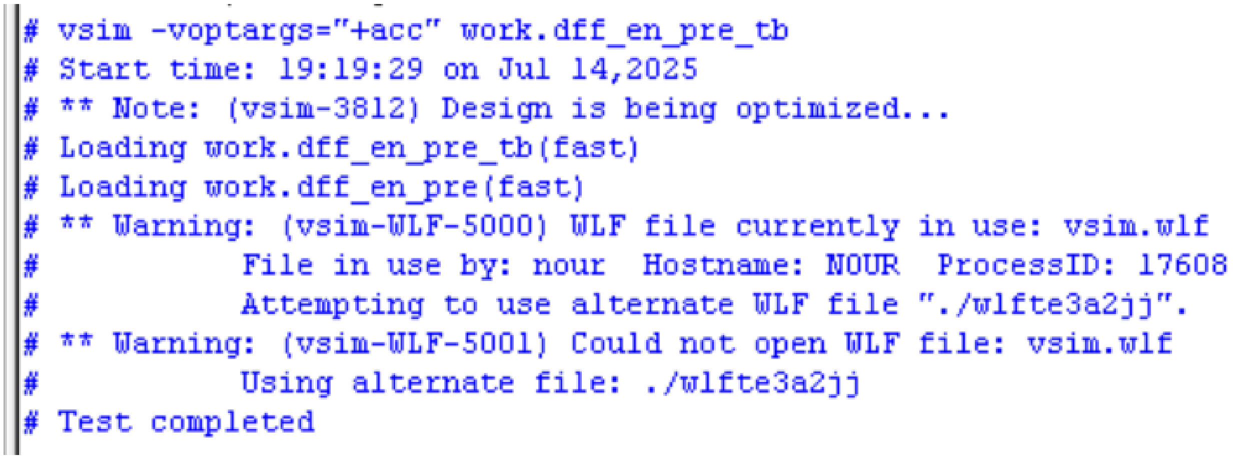
```
vlib work
vlog q1.v q1_tb.v
vsim -voptargs=+acc work.dff_en_pre_tb
add wave *
run -all
#quit -sim
```

Testbench code:

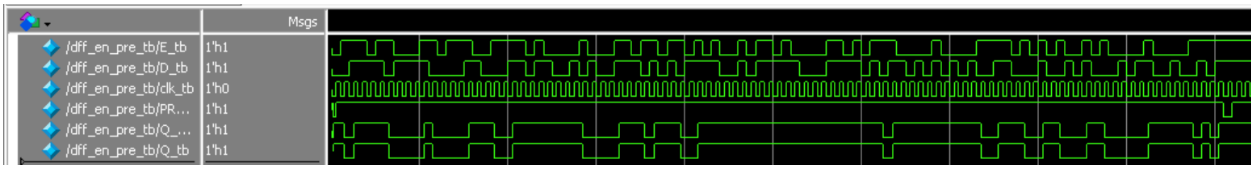


```
module dff_en_pre_tb();
reg E_tb, D_tb, clk_tb, PRE_tb, Q_exp;
wire Q_tb;

dff_en_pre DUT (Q_tb, E_tb, D_tb, clk_tb, PRE_tb);

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

initial begin
        E_tb = 0;
        D_tb = 0;
        PRE_tb = 1;
        Q_exp  = 0;
        #5;
        PRE_tb = 0;
        #5;
        PRE_tb = 1;
        #5;

        if (Q_tb !== 1'b1) begin
            $display("ERROR: PRE failed! Expected Q=1 after PRE=0, Got Q=%b", Q_tb);
            $stop;
        end

        repeat(100) begin
            @(negedge clk_tb);
            D_tb = $random;
            E_tb = $random;
            PRE_tb = 1;
        end

        @(negedge clk_tb);
        PRE_tb = 0;
        @(negedge clk_tb);
        PRE_tb = 1;

        #20;
        $display("Test completed");
end
always @(posedge clk_tb or negedge PRE_tb) begin
    if (!PRE_tb)
        Q_exp <= 1'b1;
    else if (E_tb)
        Q_exp <= D_tb;
end
always @(posedge clk_tb) begin
    #1 if (Q_tb !== Q_exp) begin
        $display("ERROR at D=%b E=%b PRE=%b | Expected Q=%b, Got Q=%b", D_tb, E_tb, PRE_tb, Q_exp, Q_tb);
        $stop;
    end
end
endmodule
```

Simulation:

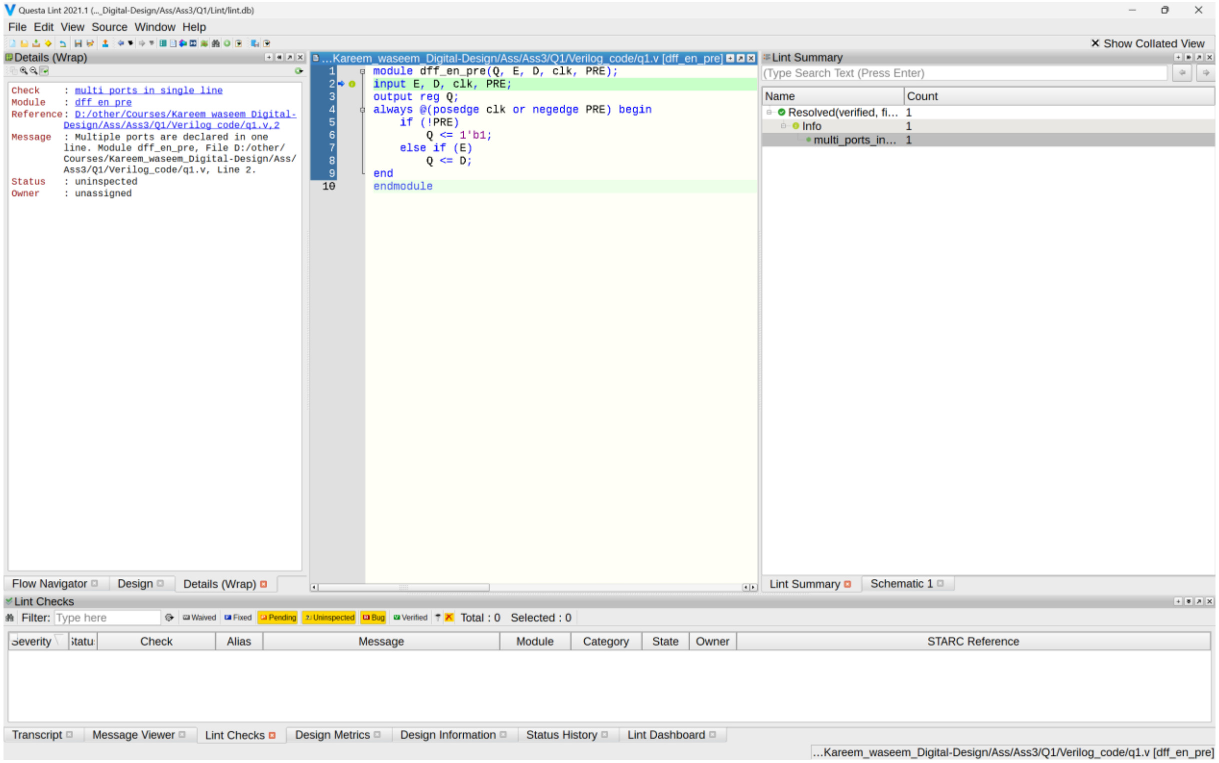


```
# vsim -voptargs="+acc" work.dff_en_pre_tb
# Start time: 19:19:29 on Jul 14,2025
# ** Note: (vsim-3812) Design is being optimized...
# Loading work.dff_en_pre_tb(fast)
# Loading work.dff_en_pre(fast)
# ** Warning: (vsim-WLF-5000) WLF file currently in use: vsim.wlf
#           File in use by: nour  Hostname: NOUR  ProcessID: 17608
#           Attempting to use alternate WLF file "./wlfte3a2jj".
# ** Warning: (vsim-WLF-5001) Could not open WLF file: vsim.wlf
#           Using alternate file: ./wlfte3a2jj
# Test completed
```



Msgs
/dff_en_pre_tb/E_tb 1'h1
/dff_en_pre_tb/D_tb 1'h1
/dff_en_pre_tb/clk_tb 1'h0
/dff_en_pre_tb/PR... 1'h1
/dff_en_pre_tb/Q_... 1'h1
/dff_en_pre_tb/Q_tb 1'h1

Linting:



Questa Lint 2021.1 (..._Digital-Design/Ass/Ass3/Q1/Lint/lint.db)
File Edit View Source Window Help
✕ Show Collated View
Details (Wrap)
...Kareem_waseem_Digital-Design/Ass/Ass3/Q1/Verilog_code/q1.v [dff_en_pre]
Lint Summary
(Type Search Text (Press Enter)

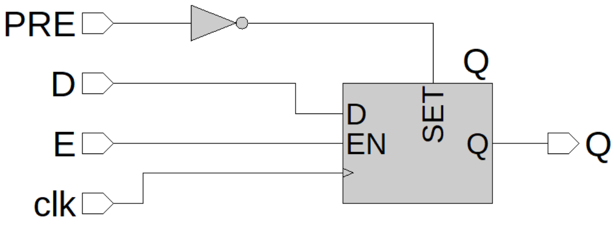
```
Check     : multi_ports_in_single_line
Module    : dff_en_pre
Reference : D:/other/Courses/Kareem_waseem_Digital-
            Design/Ass/Ass3/Q1/Verilog_code/q1.v,2
Message   : Multiple ports are declared in one
            line. Module dff_en_pre, File D:/other/
            Courses/Kareem_waseem_Digital-Design/Ass/
            Ass3/Q1/Verilog_code/q1.v, Line 2.
Status    : uninspected
Owner     : unassigned
```

```
module dff_en_pre(Q, E, D, clk, PRE);
input E, D, clk, PRE;
output reg Q;
always @(posedge clk or negedge PRE) begin
    if (!PRE)
        Q <= 1'b1;
    else if (E)
        Q <= D;
end
endmodule
```

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

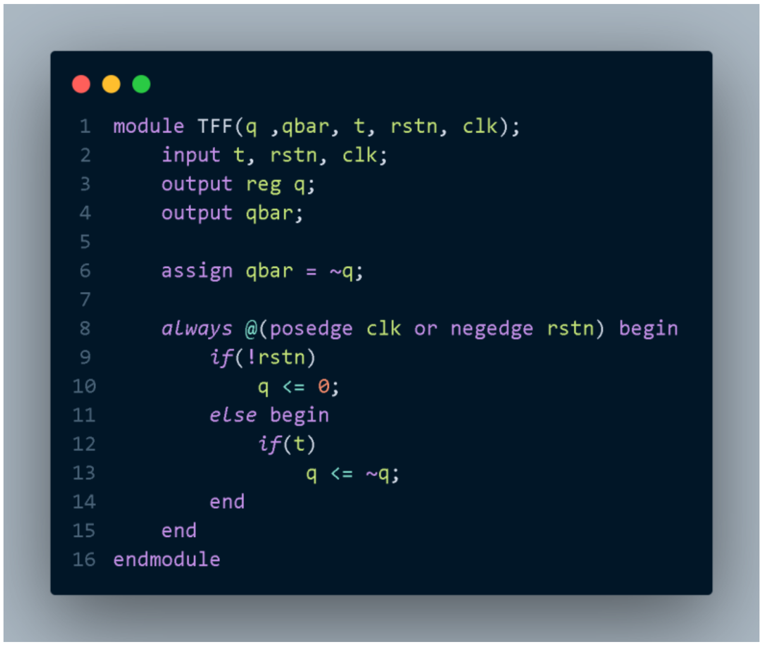
Total : 0 Selected : 0

Severity	Status	Check	Alias	Message	Module	Category	State	Owner	STARC Reference



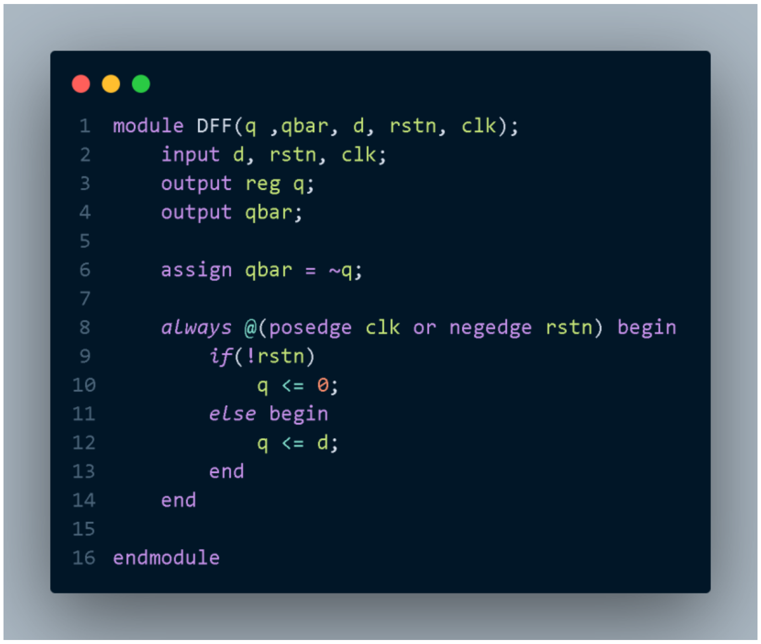
Q2)

TFF Golden model code:



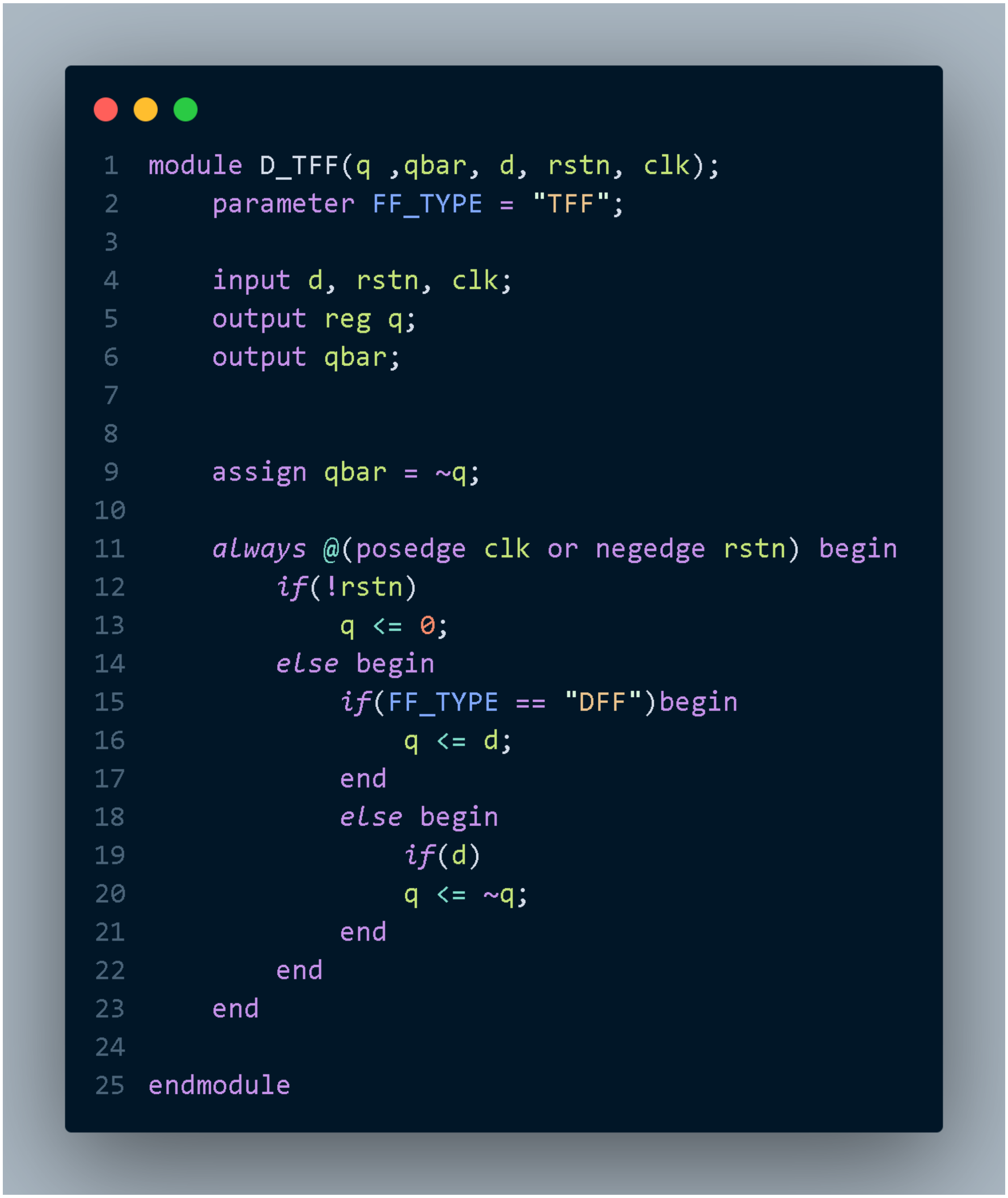
```
module TFF(q ,qbar, t, rstn, clk);
    input t, rstn, clk;
    output reg q;
    output qbar;

    assign qbar = ~q;

    always @(posedge clk or negedge rstn) begin
        if(!rstn)
            q <= 0;
        else begin
            if(t)
                q <= ~q;
        end
    end
endmodule
```

DFF Golden model code:



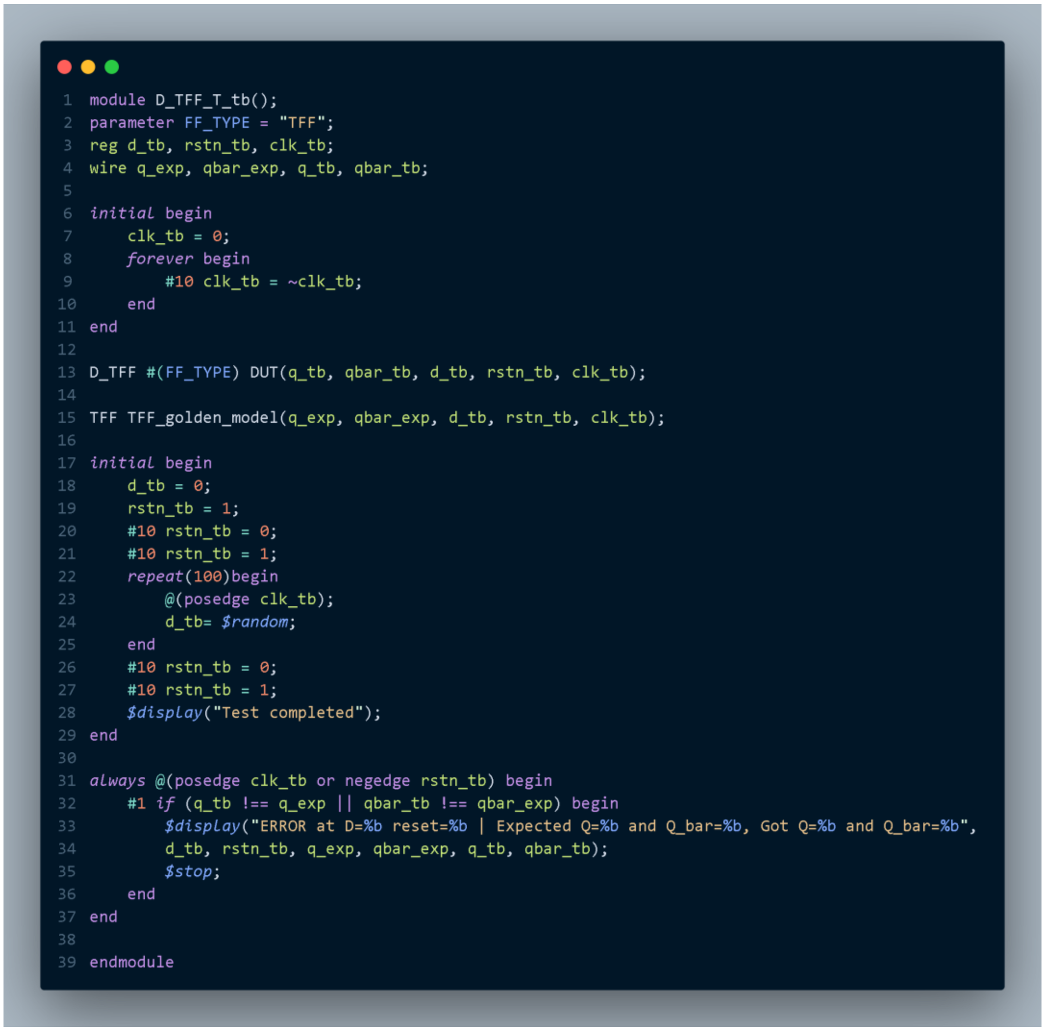
```
module DFF(q ,qbar, d, rstn, clk);
    input d, rstn, clk;
    output reg q;
    output qbar;

    assign qbar = ~q;

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

endmodule
```

Main code:



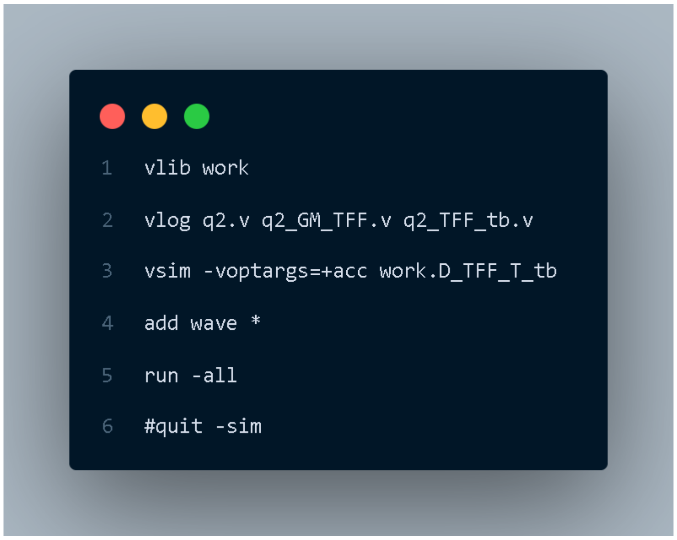
```verilog
module D_TFF(q ,qbar, d, rstn, clk);
    parameter FF_TYPE = "TFF";

    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
            if(FF_TYPE == "DFF")begin
                q <= d;
            end
            else begin
                if(d)
                q <= ~q;
            end
        end
    end

endmodule
```

TFF testbench:



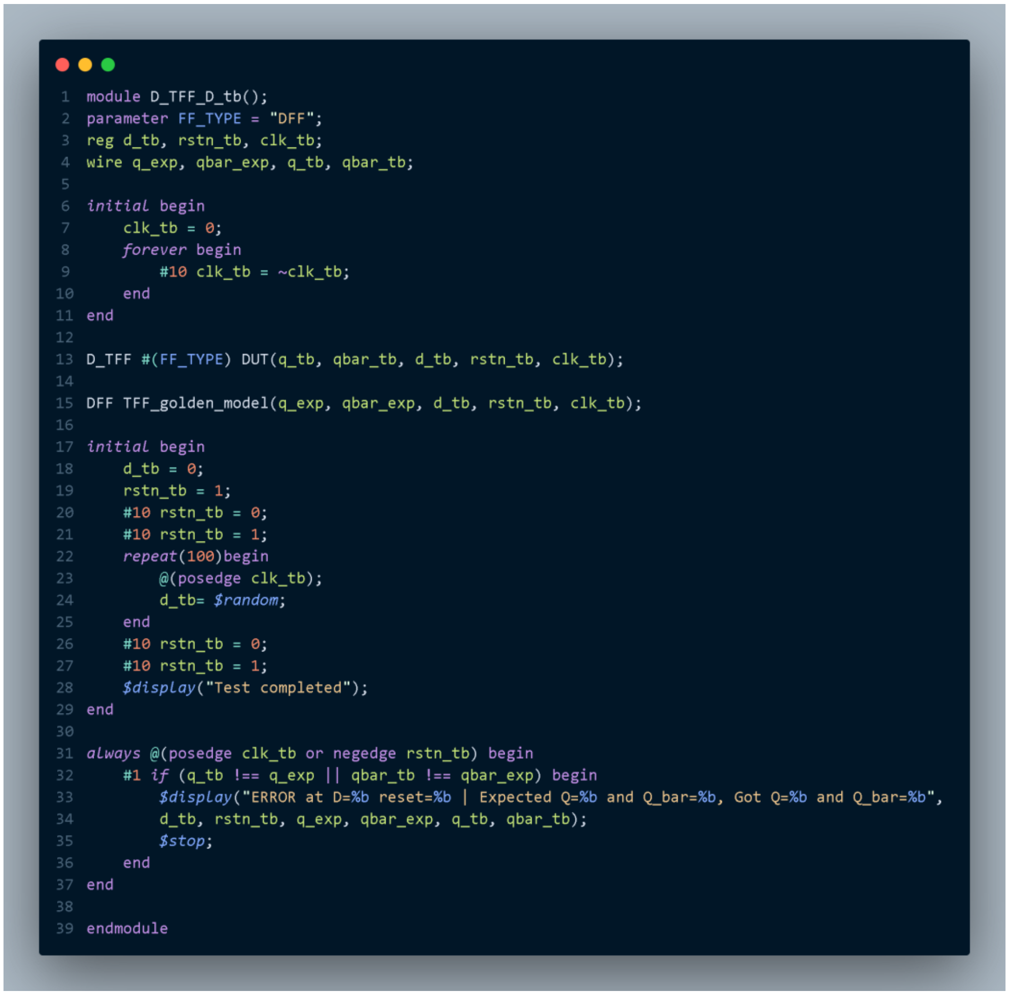
```verilog
module D_TFF_T_tb();
parameter FF_TYPE = "TFF";
reg d_tb, rstn_tb, clk_tb;
wire q_exp, qbar_exp, q_tb, qbar_tb;

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

D_TFF #(FF_TYPE) DUT(q_tb, qbar_tb, d_tb, rstn_tb, clk_tb);

TFF TFF_golden_model(q_exp, qbar_exp, d_tb, rstn_tb, clk_tb);

initial begin
    d_tb = 0;
    rstn_tb = 1;
    #10 rstn_tb = 0;
    #10 rstn_tb = 1;
    repeat(100)begin
        @(posedge clk_tb);
        d_tb= $random;
    end
    #10 rstn_tb = 0;
    #10 rstn_tb = 1;
    $display("Test completed");
end

always @(posedge clk_tb or negedge rstn_tb) begin
    #1 if (q_tb !== q_exp || qbar_tb !== qbar_exp) begin
        $display("ERROR at D=%b reset=%b | Expected Q=%b and Q_bar=%b, Got Q=%b and Q_bar=%b",
        d_tb, rstn_tb, q_exp, qbar_exp, q_tb, qbar_tb);
        $stop;
    end
end

endmodule
```

TFF do file:



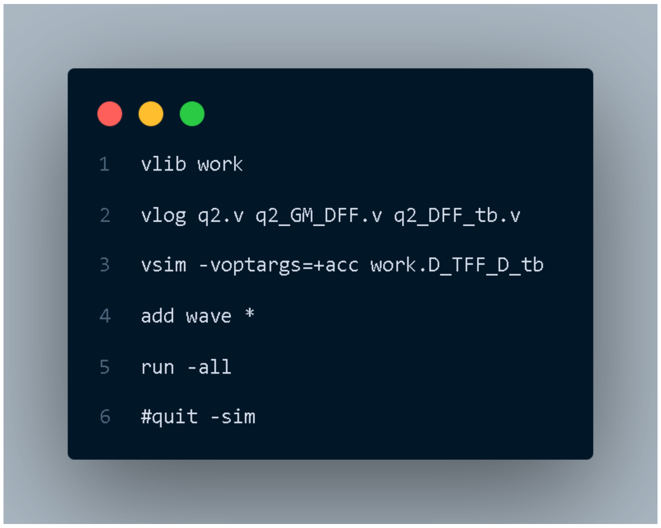
```
vlib work

vlog q2.v q2_GM_TFF.v q2_TFF_tb.v

vsim -voptargs=+acc work.D_TFF_T_tb

add wave *

run -all

#quit -sim
```

DFF testbench:



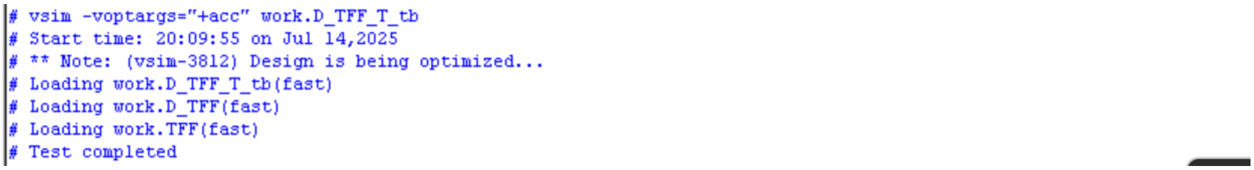
```verilog
module D_TFF_D_tb();
parameter FF_TYPE = "DFF";
reg d_tb, rstn_tb, clk_tb;
wire q_exp, qbar_exp, q_tb, qbar_tb;

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

D_TFF #(FF_TYPE) DUT(q_tb, qbar_tb, d_tb, rstn_tb, clk_tb);

DFF TFF_golden_model(q_exp, qbar_exp, d_tb, rstn_tb, clk_tb);

initial begin
    d_tb = 0;
    rstn_tb = 1;
    #10 rstn_tb = 0;
    #10 rstn_tb = 1;
    repeat(100)begin
        @(posedge clk_tb);
        d_tb= $random;
    end
    #10 rstn_tb = 0;
    #10 rstn_tb = 1;
    $display("Test completed");
end

always @(posedge clk_tb or negedge rstn_tb) begin
    #1 if (q_tb !== q_exp || qbar_tb !== qbar_exp) begin
        $display("ERROR at D=%b reset=%b | Expected Q=%b and Q_bar=%b, Got Q=%b and Q_bar=%b",
        d_tb, rstn_tb, q_exp, qbar_exp, q_tb, qbar_tb);
        $stop;
    end
end

endmodule
```

DFF do file:

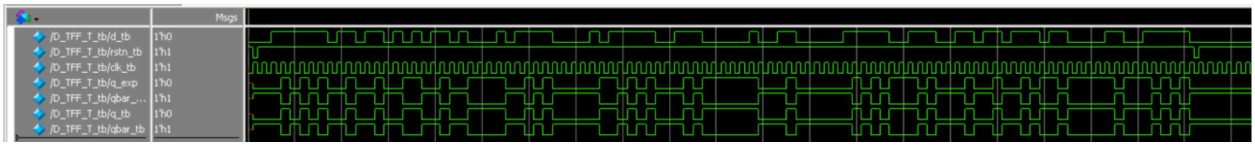


```
vlib work
vlog q2.v q2_GM_DFF.v q2_DFF_tb.v
vsim -voptargs=+acc work.D_TFF_D_tb
add wave *
run -all
#quit -sim
```

TFF simulation:

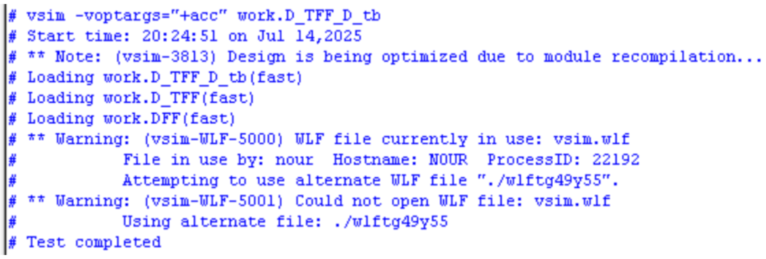


```
# vsim -voptargs="+acc" work.D_TFF_T_tb
# Start time: 20:09:55 on Jul 14,2025
# ** Note: (vsim-3812) Design is being optimized...
# Loading work.D_TFF_T_tb(fast)
# Loading work.D_TFF(fast)
# Loading work.TFF(fast)
# Test completed
```

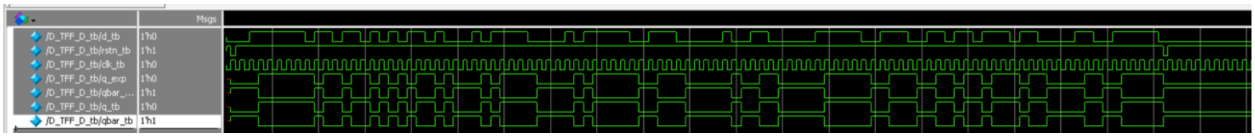


```
/D_TFF_T_tb/d_tb        1'h0
/D_TFF_T_tb/rstn_tb     1'h1
/D_TFF_T_tb/clk_tb      1'h1
/D_TFF_T_tb/q_exp       1'h0
/D_TFF_T_tb/qbar_...    1'h1
/D_TFF_T_tb/q_tb        1'h0
/D_TFF_T_tb/qbar_tb     1'h1
```

DFF simulation

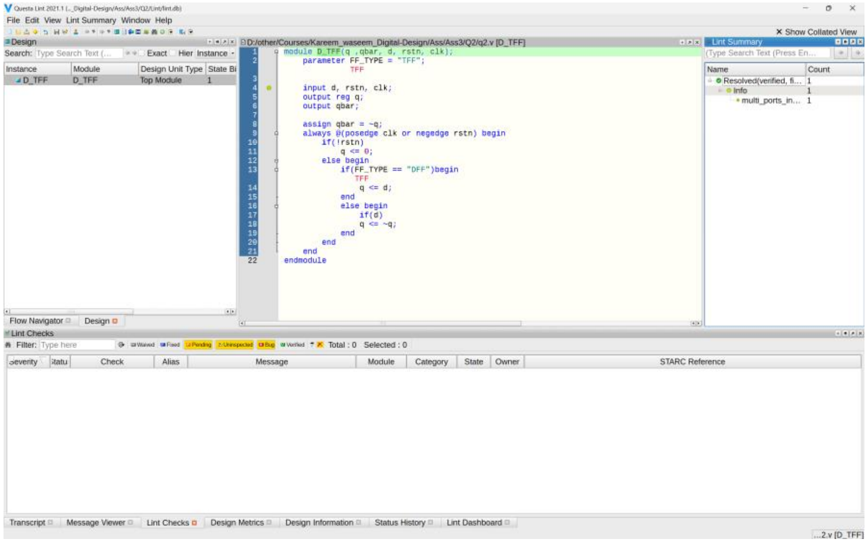


```
# vsim -voptargs="+acc" work.D_TFF_D_tb
# Start time: 20:24:51 on Jul 14,2025
# ** Note: (vsim-3813) Design is being optimized due to module recompilation...
# Loading work.D_TFF_D_tb(fast)
# Loading work.D_TFF(fast)
# Loading work.DFF(fast)
# ** Warning: (vsim-WLF-5000) WLF file currently in use: vsim.wlf
#           File in use by: nour  Hostname: NOUR  ProcessID: 22192
#           Attempting to use alternate WLF file "./wlftg49y55".
# ** Warning: (vsim-WLF-5001) Could not open WLF file: vsim.wlf
#           Using alternate file: ./wlftg49y55
# Test completed
```



```
/D_TFF_D_tb/d_tb        1'h0
/D_TFF_D_tb/rstn_tb     1'h1
/D_TFF_D_tb/clk_tb      1'h0
/D_TFF_D_tb/q_exp       1'h0
/D_TFF_D_tb/qbar_...    1'h1
/D_TFF_D_tb/q_tb        1'h0
/D_TFF_D_tb/qbar_tb     1'h1
```

TFF lint:



Questa Lint 2021.1 (..._Digital-Design/Ass/Ass3/Q2/Lint/lint.db)

D:/other/Courses/Kareem_waseem_Digital-Design/Ass/Ass3/Q2/q2.v [D_TFF]

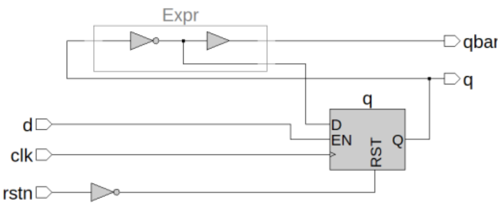
```
module D_TFF(q ,qbar, d, rstn, clk);
    parameter FF_TYPE = "TFF";
              TFF

    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
            if(FF_TYPE == "DFF")begin
              TFF
                q <= d;
            end
            else begin
                if(d)
                q <= ~q;
            end
        end
    end
endmodule
```

Instance	Module	Design Unit Type	State Bi
D_TFF	D_TFF	Top Module	1

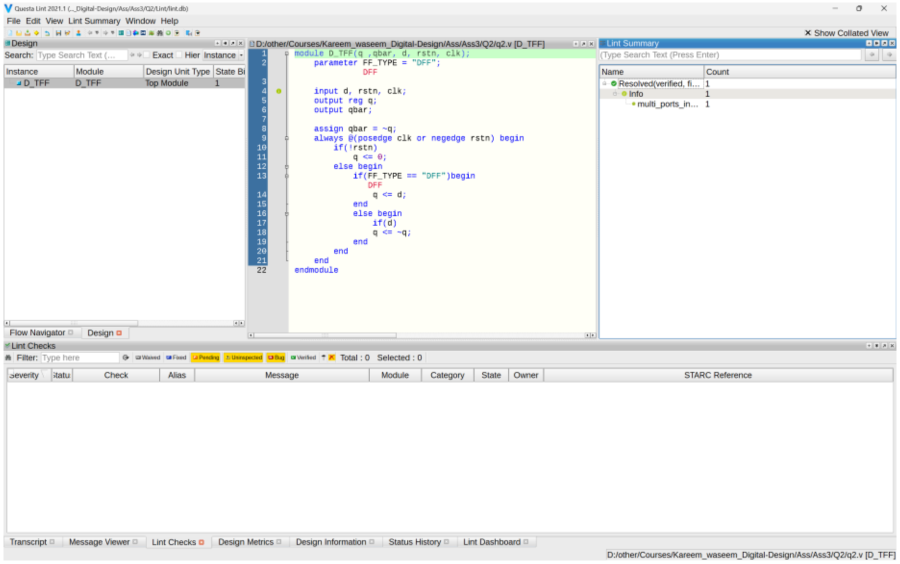
Name	Count
Resolved(verified, fi...	1
Info	1
multi_ports_in...	1

Total : 0 Selected : 0

Severity	Statu	Check	Alias	Message	Module	Category	State	Owner	STARC Reference



DFF lint:



Questa Lint 2021.1 (..._Digital-Design/Ass/Ass3/Q2/Lint/lint.db)

D:/other/Courses/Kareem_waseem_Digital-Design/Ass/Ass3/Q2/q2.v [D_TFF]

```
module D_TFF(q ,qbar, d, rstn, clk);
    parameter FF_TYPE = "DFF";
              DFF

    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
            if(FF_TYPE == "DFF")begin
              DFF
                q <= d;
            end
            else begin
                if(d)
                q <= ~q;
            end
        end
    end
endmodule
```

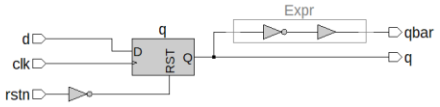
Instance	Module	Design Unit Type	State Bi
D_TFF	D_TFF	Top Module	1

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

Total : 0 Selected : 0

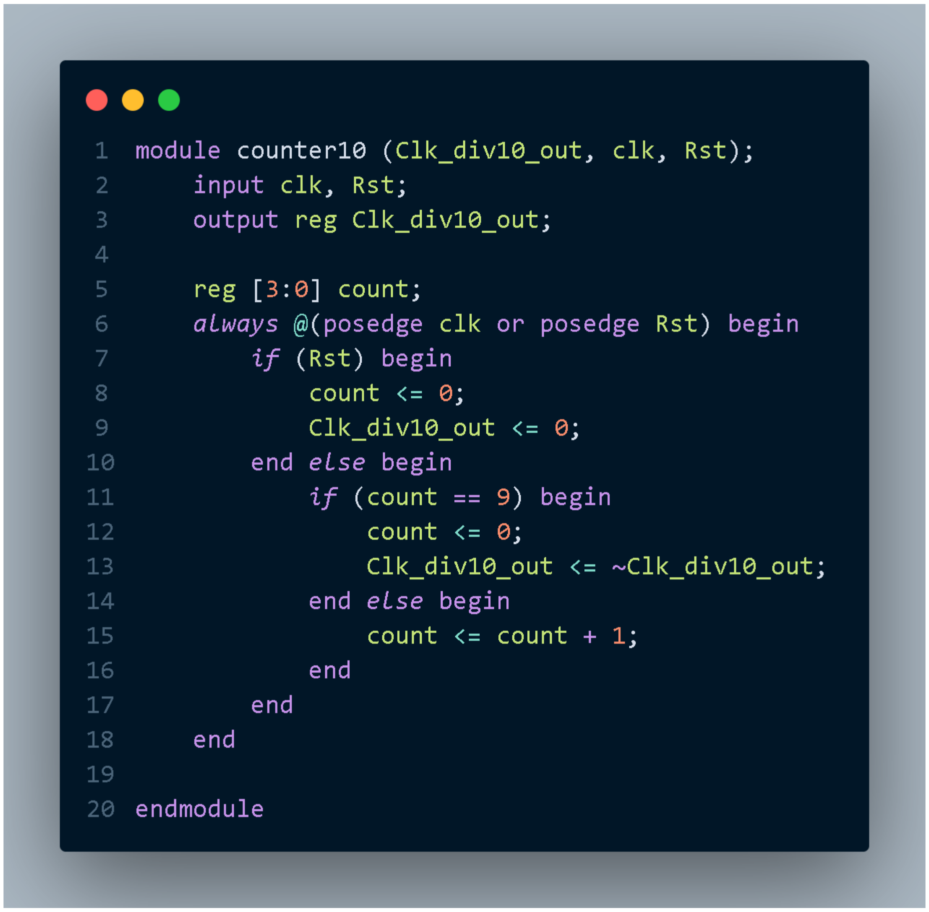
Severity	Statu	Check	Alias	Message	Module	Category	State	Owner	STARC Reference

D:/other/Courses/Kareem_waseem_Digital-Design/Ass/Ass3/Q2/q2.v [D_TFF]



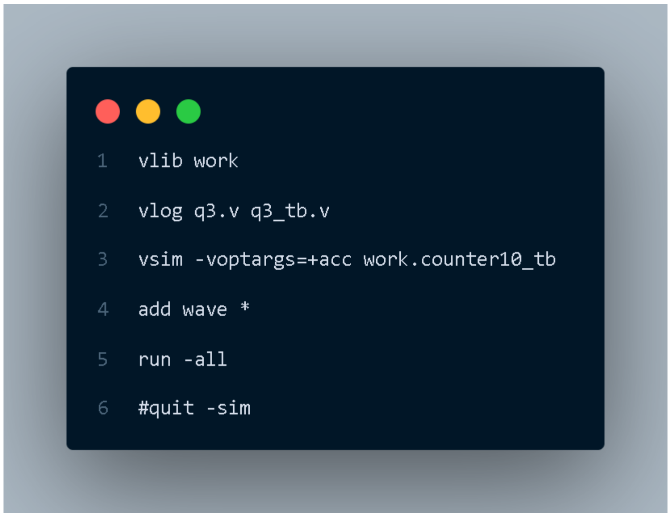
Q3)

Main code:



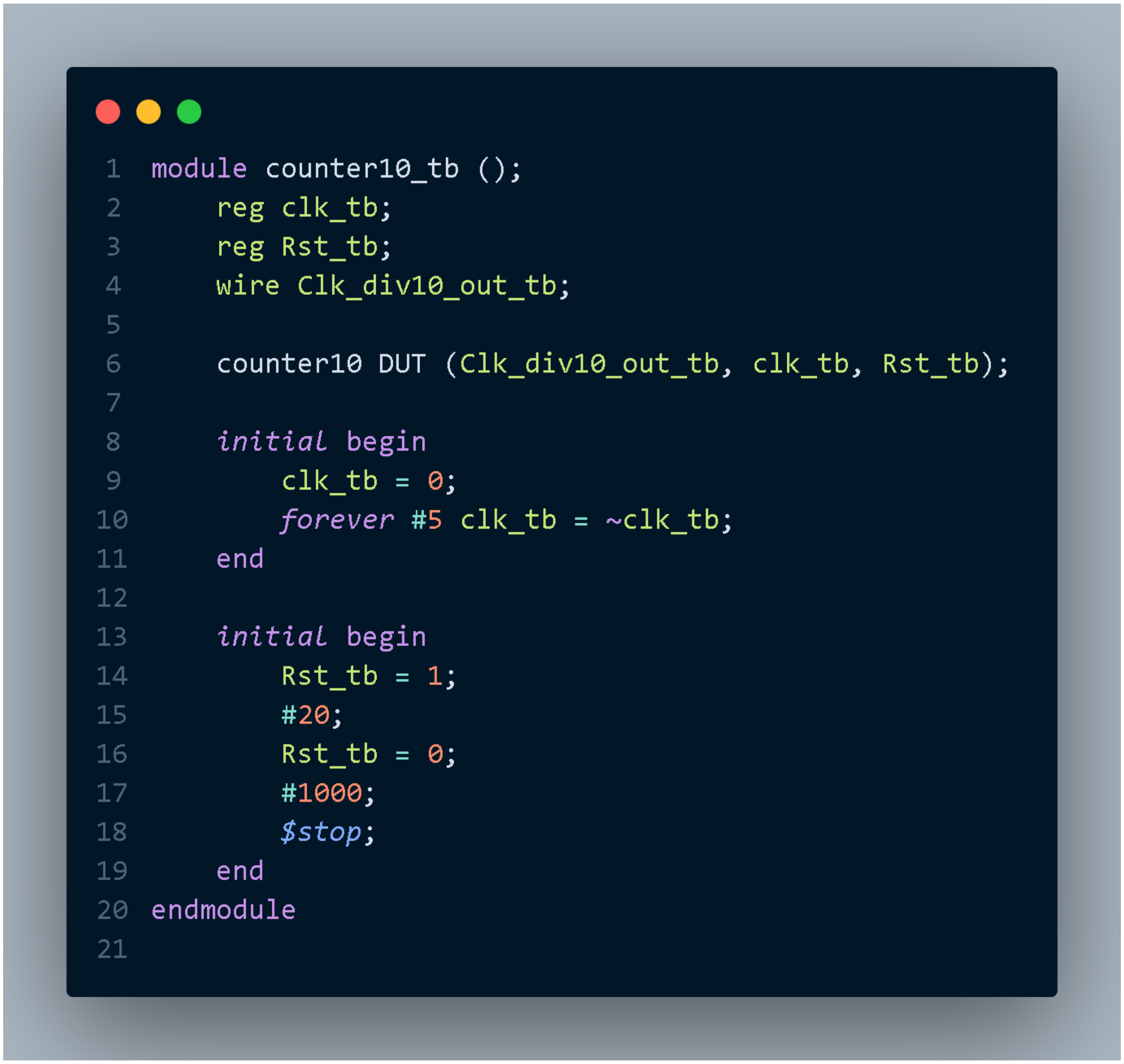
```
module counter10 (Clk_div10_out, clk, Rst);
    input clk, Rst;
    output reg Clk_div10_out;

    reg [3:0] count;
    always @(posedge clk or posedge Rst) begin
        if (Rst) begin
            count <= 0;
            Clk_div10_out <= 0;
        end else begin
            if (count == 9) begin
                count <= 0;
                Clk_div10_out <= ~Clk_div10_out;
            end else begin
                count <= count + 1;
            end
        end
    end

endmodule
```

Do file:



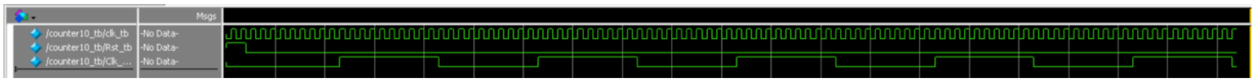
```
vlib work

vlog q3.v q3_tb.v

vsim -voptargs=+acc work.counter10_tb

add wave *

run -all

#quit -sim
```

Testbench code:



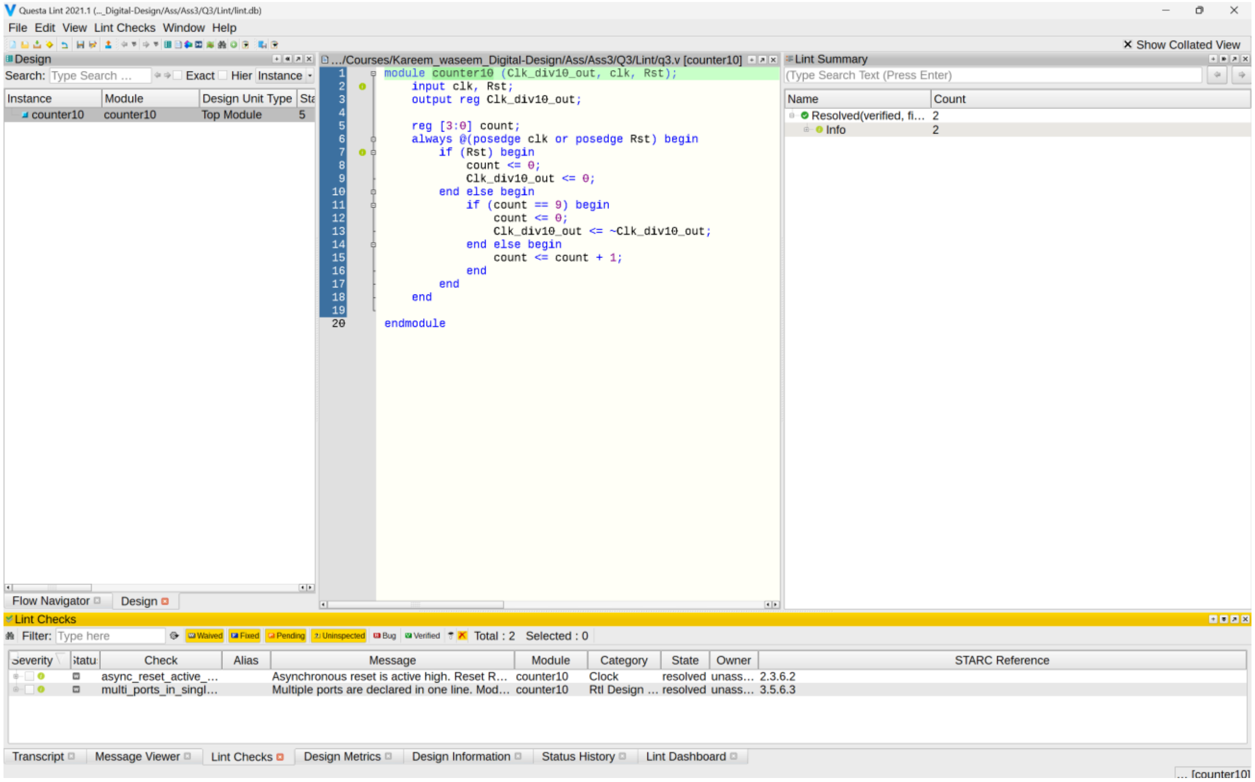
```
module counter10_tb ();
    reg clk_tb;
    reg Rst_tb;
    wire Clk_div10_out_tb;

    counter10 DUT (Clk_div10_out_tb, clk_tb, Rst_tb);

    initial begin
        clk_tb = 0;
        forever #5 clk_tb = ~clk_tb;
    end

    initial begin
        Rst_tb = 1;
        #20;
        Rst_tb = 0;
        #1000;
        $stop;
    end
endmodule

```

Simulation:



```
/counter10_tb/clk_tb    -No Data-
/counter10_tb/Rst_tb    -No Data-
/counter10_tb/Clk_...   -No Data-
```

Lint:



Questa Lint 2021.1 (..._Digital-Design/Ass/Ass3/Q3/Lint/lint.db)

Design — Search: Type Search ... Exact Hier Instance

Instance	Module	Design Unit Type	Sta
counter10	counter10	Top Module	5

.../Courses/Kareem_waseem_Digital-Design/Ass/Ass3/Q3/Lint/q3.v [counter10]

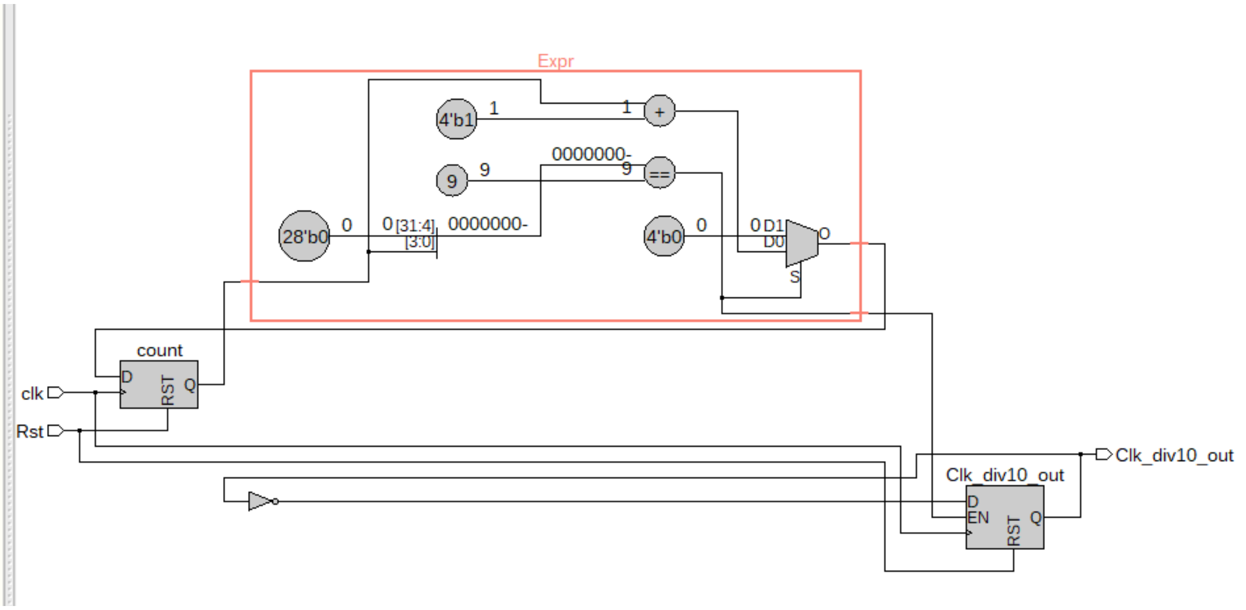
```verilog
module counter10 (Clk_div10_out, clk, Rst);
    input clk, Rst;
    output reg Clk_div10_out;

    reg [3:0] count;
    always @(posedge clk or posedge Rst) begin
        if (Rst) begin
            count <= 0;
            Clk_div10_out <= 0;
        end else begin
            if (count == 9) begin
                count <= 0;
                Clk_div10_out <= ~Clk_div10_out;
            end else begin
                count <= count + 1;
            end
        end
    end

endmodule
```

Lint Summary

Name	Count
Resolved(verified, fi...	2
Info	2

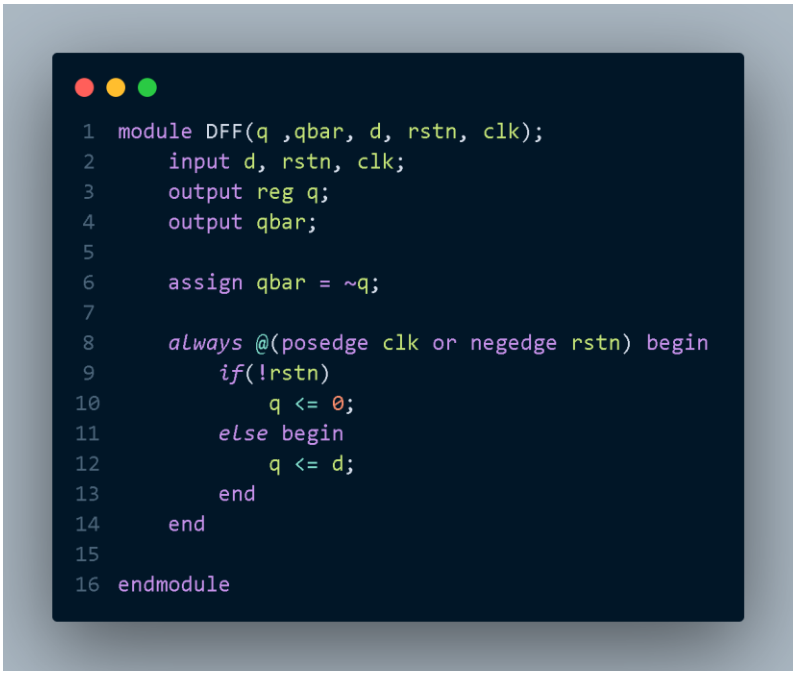
Lint Checks — Filter: Type here — Waived Fixed Pending Uninspected Bug Verified — Total : 2 Selected : 0

Severity	Status	Check	Alias	Message	Module	Category	State	Owner	STARC Reference
●		async_reset_active_...		Asynchronous reset is active high. Reset R...	counter10	Clock	resolved	unass...	2.3.6.2
●		multi_ports_in_singl...		Multiple ports are declared in one line. Mod...	counter10	Rtl Design ...	resolved	unass...	3.5.6.3



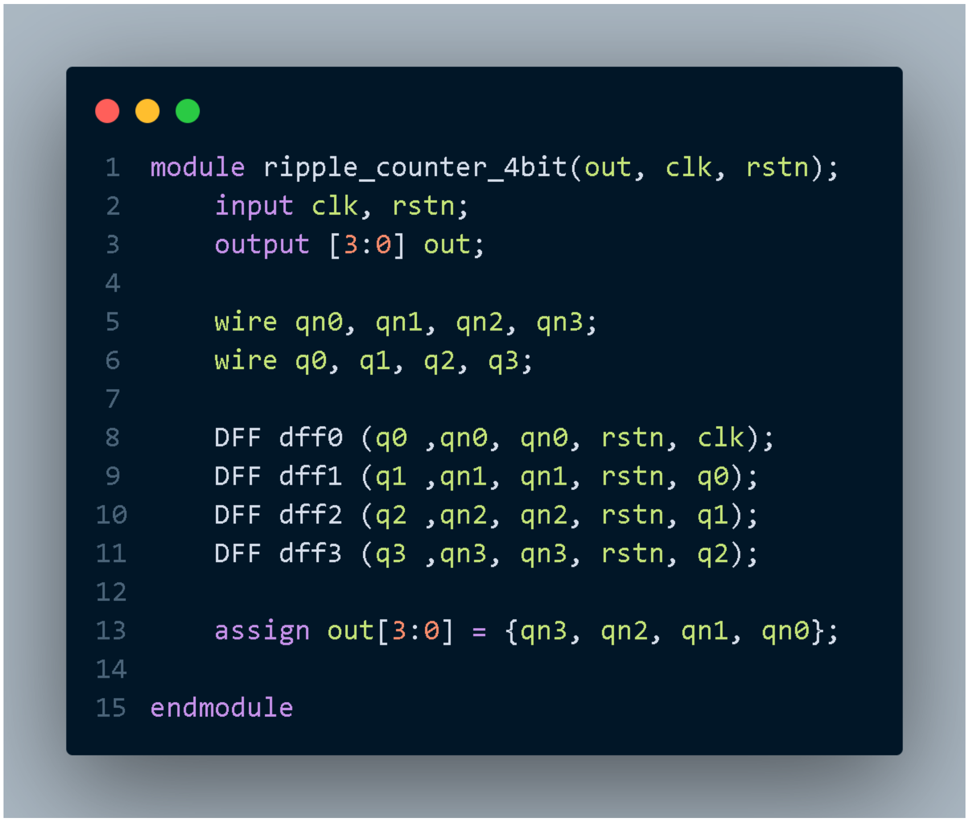
Q4)

DFF code:



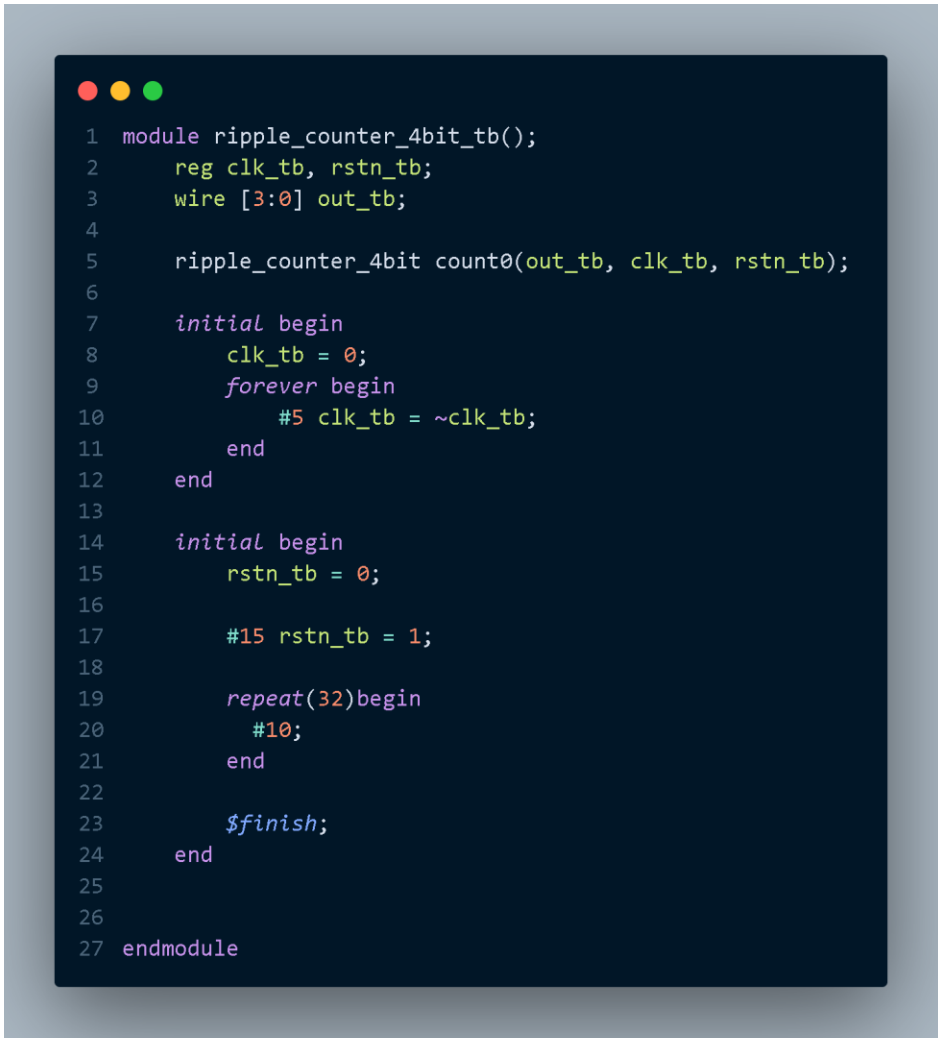
```
module DFF(q ,qbar, d, rstn, clk);
    input d, rstn, clk;
    output reg q;
    output qbar;

    assign qbar = ~q;

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

endmodule
```

Counter code:



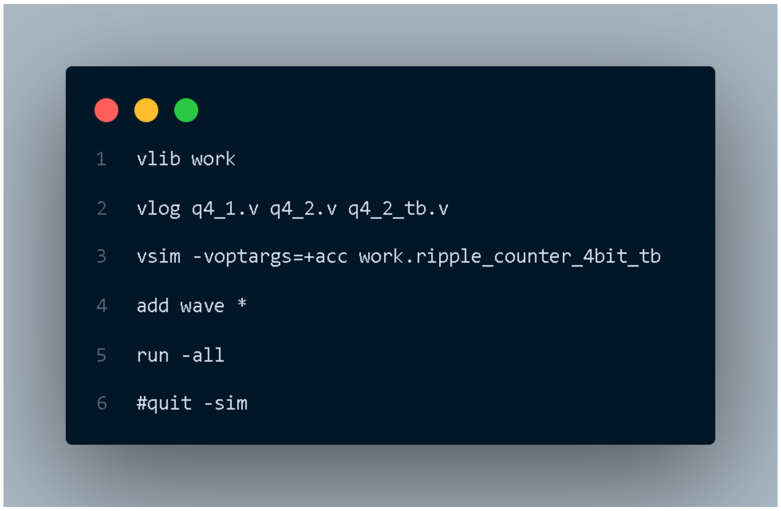
```
module ripple_counter_4bit(out, clk, rstn);
    input clk, rstn;
    output [3:0] out;

    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
```

Counter testbench code:



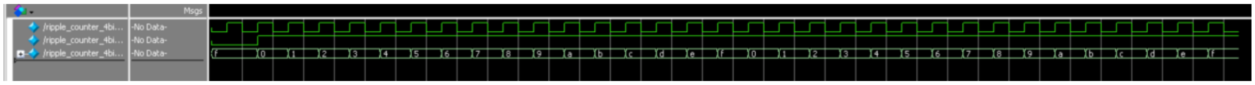
```verilog
module ripple_counter_4bit_tb();
    reg clk_tb, rstn_tb;
    wire [3:0] out_tb;

    ripple_counter_4bit count0(out_tb, clk_tb, rstn_tb);

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

    initial begin
        rstn_tb = 0;

        #15 rstn_tb = 1;

        repeat(32)begin
          #10;
        end

        $finish;
    end


endmodule
```

Do file:

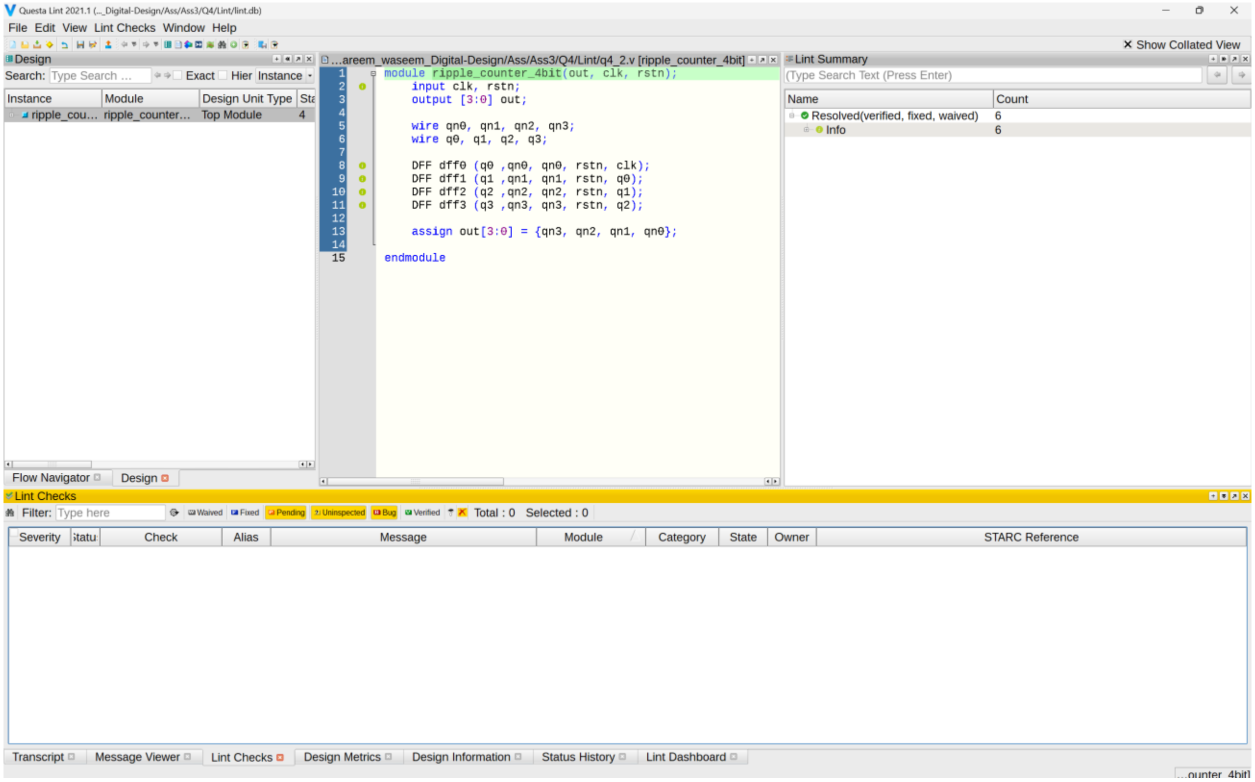


```
vlib work

vlog q4_1.v q4_2.v q4_2_tb.v

vsim -voptargs=+acc work.ripple_counter_4bit_tb

add wave *

run -all

#quit -sim
```

Simulation:



Lint:


Questa Lint 2021.1 (..._Digital-Design/Ass/Ass3/Q4/Lint/lint.db)
File Edit View Lint Checks Window Help
Design
Search: Type Search ... Exact Hier Instance
Show Collated View
Lint Summary
(Type Search Text (Press Enter)
Flow Navigator
Design
Lint Checks
Filter: Type here
Waived Fixed Pending Uninspected Bug Verified
Transcript
Message Viewer
Lint Checks
Design Metrics
Design Information
Status History
Lint Dashboard
...ounter_4bit]

Instance	Module	Design Unit Type	Sta
ripple_cou…	ripple_counter…	Top Module	4

...areem_waseem_Digital-Design/Ass/Ass3/Q4/Lint/q4_2.v [ripple_counter_4bit]

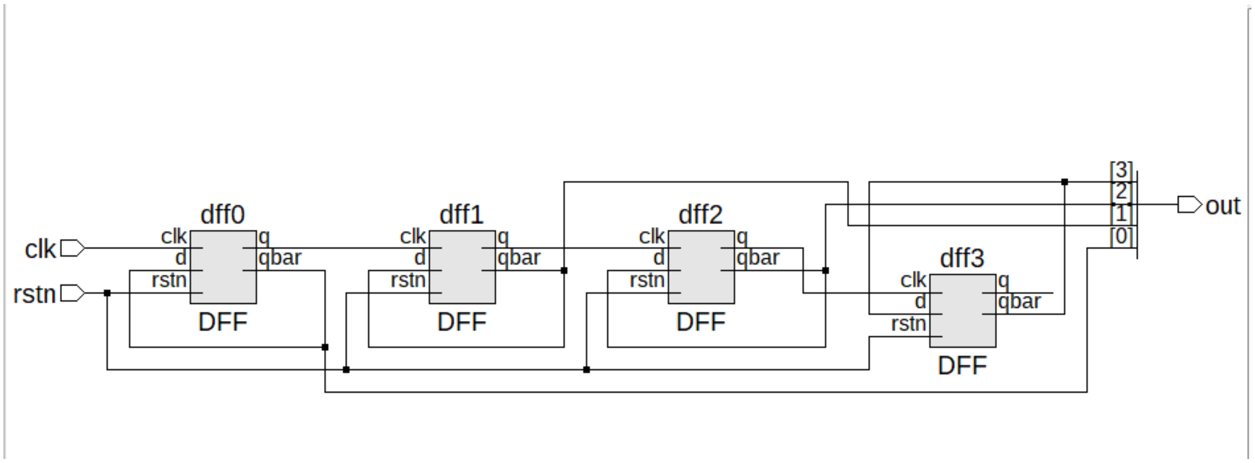
```
module ripple_counter_4bit(out, clk, rstn);
    input clk, rstn;
    output [3:0] out;

    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
```

Name	Count
Resolved(verified, fixed, waived)	6
Info	6

Total : 0 Selected : 0

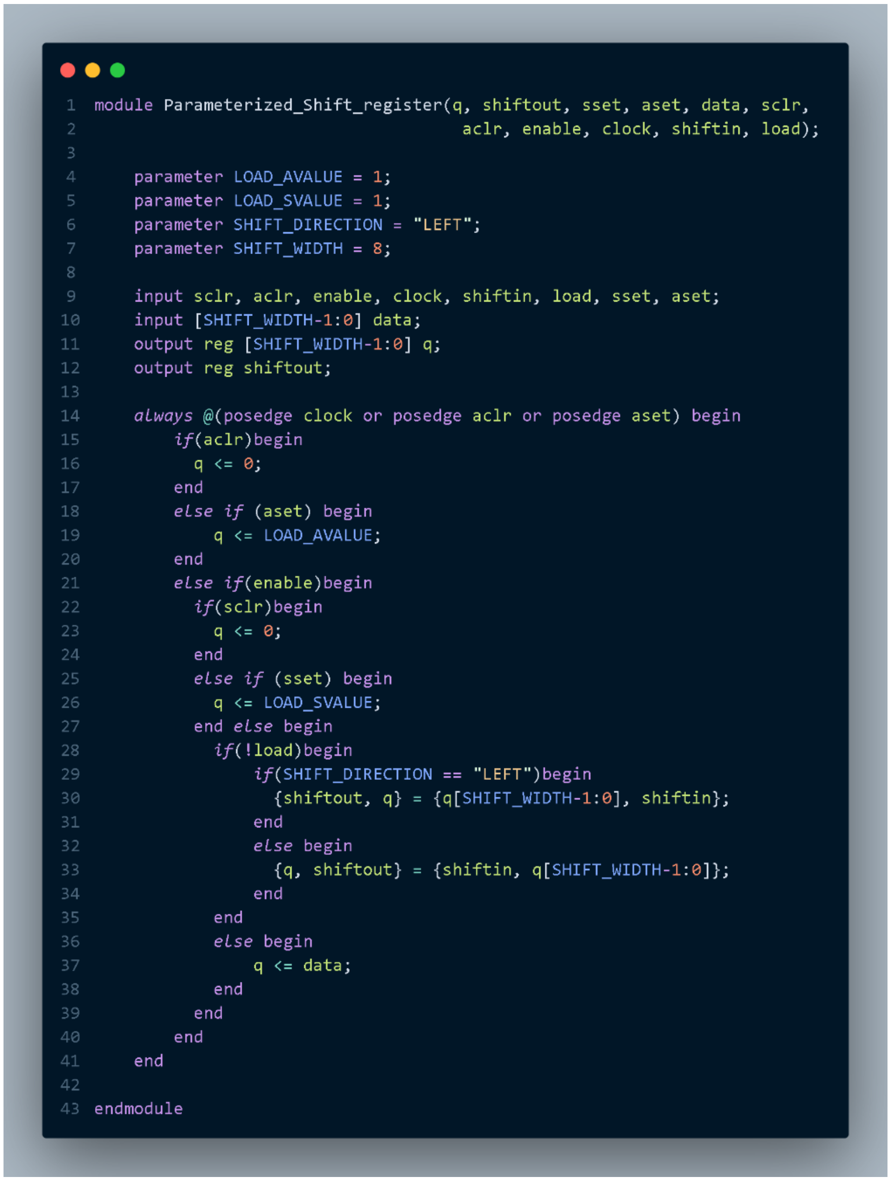
Severity	Statu	Check	Alias	Message	Module	Category	State	Owner	STARC Reference

Schematic:



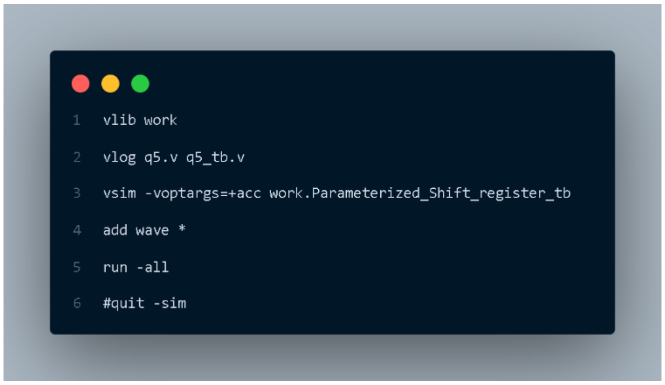
Q5)

Register code:



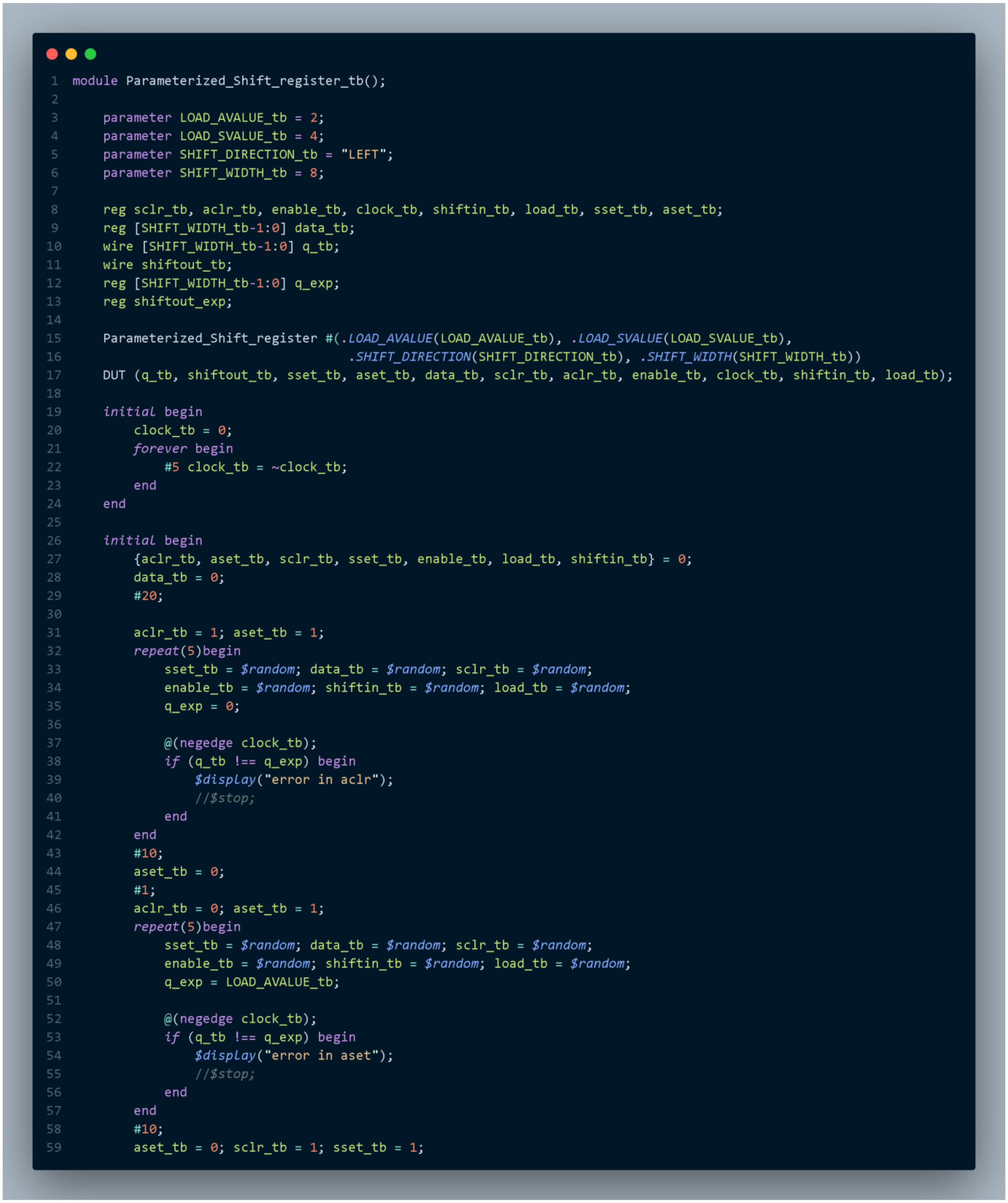
```
module Parameterized_Shift_register(q, shiftout, sset, aset, data, sclr,
                                    aclr, enable, clock, shiftin, load);

    parameter LOAD_AVALUE = 1;
    parameter LOAD_SVALUE = 1;
    parameter SHIFT_DIRECTION = "LEFT";
    parameter SHIFT_WIDTH = 8;

    input sclr, aclr, enable, clock, shiftin, load, sset, aset;
    input [SHIFT_WIDTH-1:0] data;
    output reg [SHIFT_WIDTH-1:0] q;
    output reg shiftout;

    always @(posedge clock or posedge aclr or posedge aset) begin
        if(aclr)begin
          q <= 0;
        end
        else if (aset) begin
            q <= LOAD_AVALUE;
        end
        else if(enable)begin
          if(sclr)begin
            q <= 0;
          end
          else if (sset) begin
            q <= LOAD_SVALUE;
          end else begin
            if(!load)begin
                if(SHIFT_DIRECTION == "LEFT")begin
                  {shiftout, q} = {q[SHIFT_WIDTH-1:0], shiftin};
                end
                else begin
                  {q, shiftout} = {shiftin, q[SHIFT_WIDTH-1:0]};
                end
            end
            else begin
                q <= data;
            end
          end
        end
    end

endmodule
```

Do file:



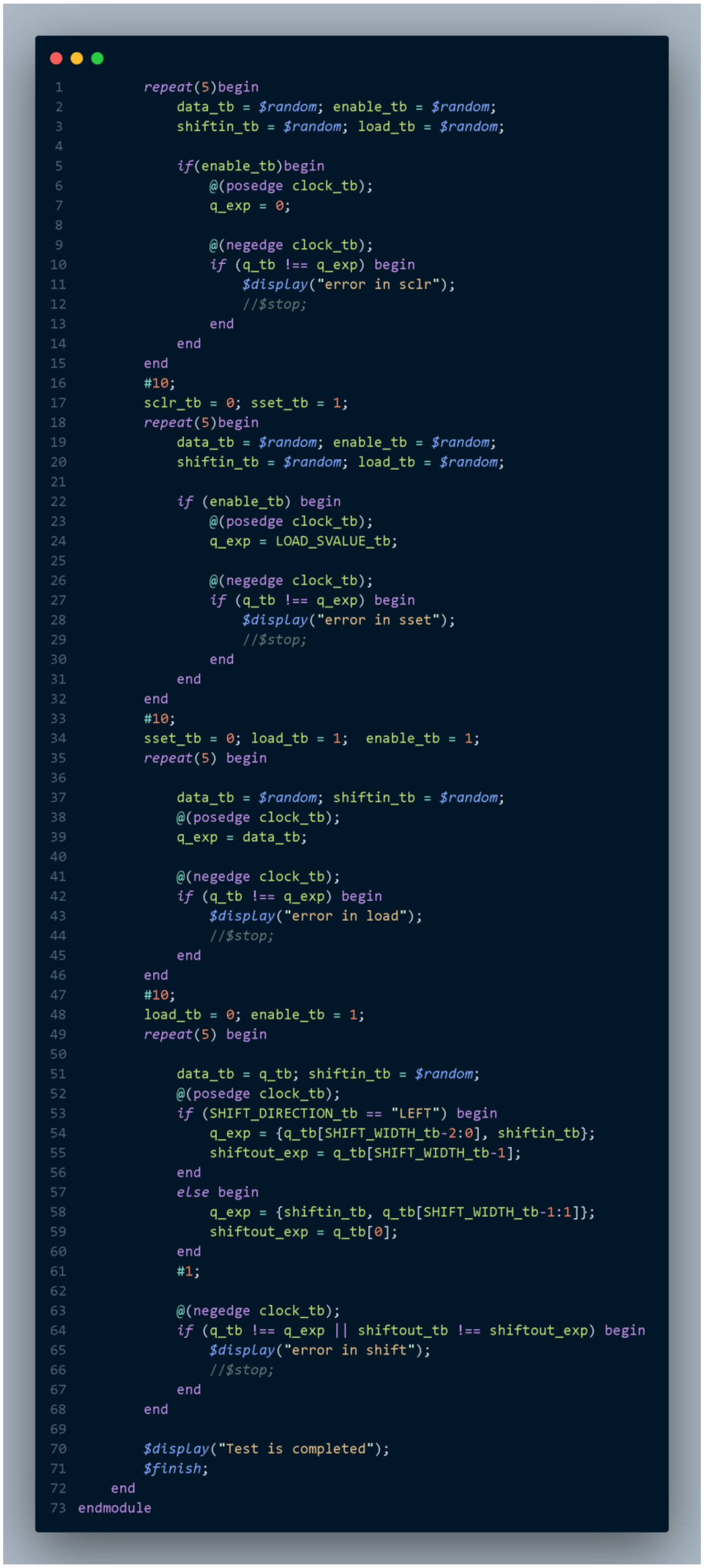
```
vlib work
vlog q5.v q5_tb.v
vsim -voptargs=+acc work.Parameterized_Shift_register_tb
add wave *
run -all
#quit -sim
```

Testbench code:



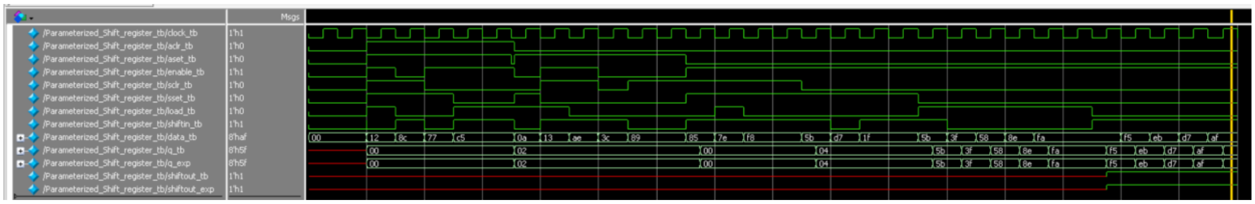
```
module Parameterized_Shift_register_tb();

    parameter LOAD_AVALUE_tb = 2;
    parameter LOAD_SVALUE_tb = 4;
    parameter SHIFT_DIRECTION_tb = "LEFT";
    parameter SHIFT_WIDTH_tb = 8;

    reg sclr_tb, aclr_tb, enable_tb, clock_tb, shiftin_tb, load_tb, sset_tb, aset_tb;
    reg [SHIFT_WIDTH_tb-1:0] data_tb;
    wire [SHIFT_WIDTH_tb-1:0] q_tb;
    wire shiftout_tb;
    reg [SHIFT_WIDTH_tb-1:0] q_exp;
    reg shiftout_exp;

    Parameterized_Shift_register #(.LOAD_AVALUE(LOAD_AVALUE_tb), .LOAD_SVALUE(LOAD_SVALUE_tb),
                                    .SHIFT_DIRECTION(SHIFT_DIRECTION_tb), .SHIFT_WIDTH(SHIFT_WIDTH_tb))
    DUT (q_tb, shiftout_tb, sset_tb, aset_tb, data_tb, sclr_tb, aclr_tb, enable_tb, clock_tb, shiftin_tb, load_tb);

    initial begin
        clock_tb = 0;
        forever begin
            #5 clock_tb = ~clock_tb;
        end
    end

    initial begin
        {aclr_tb, aset_tb, sclr_tb, sset_tb, enable_tb, load_tb, shiftin_tb} = 0;
        data_tb = 0;
        #20;

        aclr_tb = 1; aset_tb = 1;
        repeat(5)begin
            sset_tb = $random; data_tb = $random; sclr_tb = $random;
            enable_tb = $random; shiftin_tb = $random; load_tb = $random;
            q_exp = 0;

            @(negedge clock_tb);
            if (q_tb !== q_exp) begin
                $display("error in aclr");
                //$stop;
            end
        end
        #10;
        aset_tb = 0;
        #1;
        aclr_tb = 0; aset_tb = 1;
        repeat(5)begin
            sset_tb = $random; data_tb = $random; sclr_tb = $random;
            enable_tb = $random; shiftin_tb = $random; load_tb = $random;
            q_exp = LOAD_AVALUE_tb;

            @(negedge clock_tb);
            if (q_tb !== q_exp) begin
                $display("error in aset");
                //$stop;
            end
        end
        #10;
        aset_tb = 0; sclr_tb = 1; sset_tb = 1;
```

//rest of code in next page:



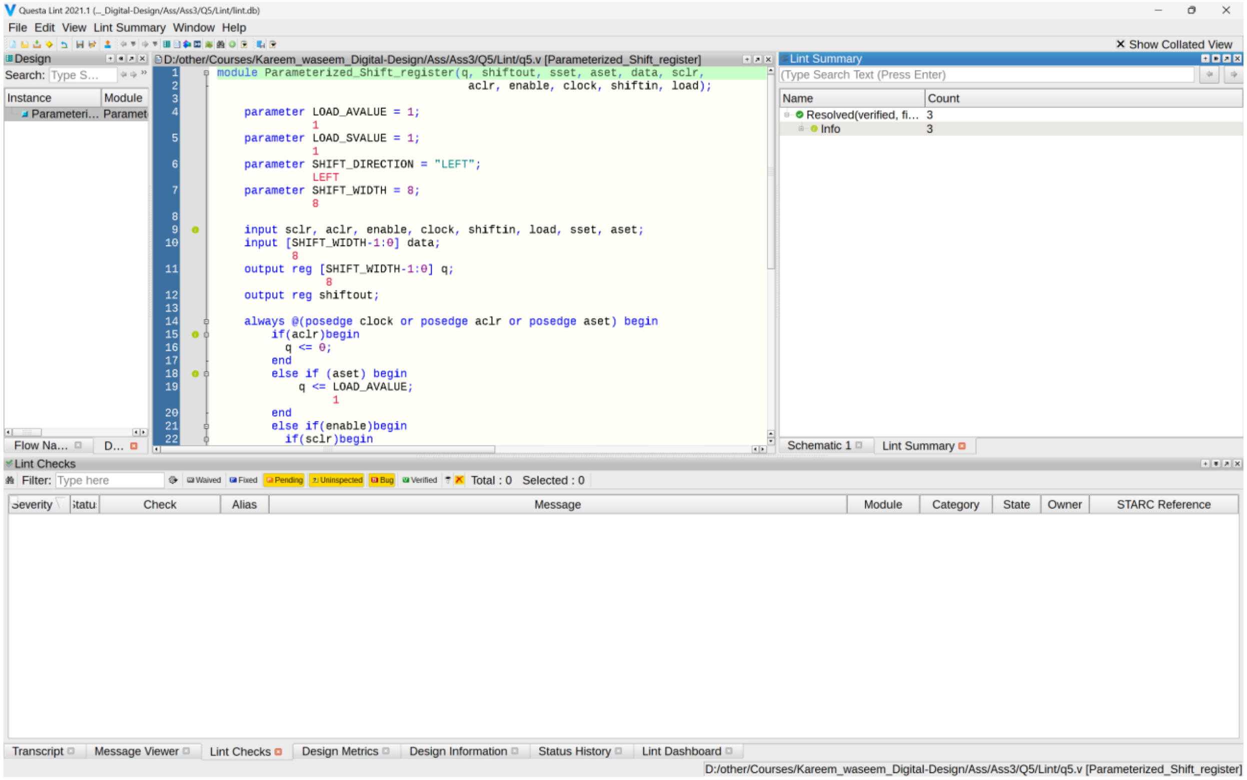
```
        repeat(5)begin
            data_tb = $random; enable_tb = $random;
            shiftin_tb = $random; load_tb = $random;

            if(enable_tb)begin
                @(posedge clock_tb);
                q_exp = 0;

                @(negedge clock_tb);
                if (q_tb !== q_exp) begin
                    $display("error in sclr");
                    //$stop;
                end
            end
        end
        #10;
        sclr_tb = 0; sset_tb = 1;
        repeat(5)begin
            data_tb = $random; enable_tb = $random;
            shiftin_tb = $random; load_tb = $random;

            if (enable_tb) begin
                @(posedge clock_tb);
                q_exp = LOAD_SVALUE_tb;

                @(negedge clock_tb);
                if (q_tb !== q_exp) begin
                    $display("error in sset");
                    //$stop;
                end
            end
        end
        #10;
        sset_tb = 0; load_tb = 1;  enable_tb = 1;
        repeat(5) begin

            data_tb = $random; shiftin_tb = $random;
            @(posedge clock_tb);
            q_exp = data_tb;

            @(negedge clock_tb);
            if (q_tb !== q_exp) begin
                $display("error in load");
                //$stop;
            end
        end
        #10;
        load_tb = 0; enable_tb = 1;
        repeat(5) begin

            data_tb = q_tb; shiftin_tb = $random;
            @(posedge clock_tb);
            if (SHIFT_DIRECTION_tb == "LEFT") begin
                q_exp = {q_tb[SHIFT_WIDTH_tb-2:0], shiftin_tb};
                shiftout_exp = q_tb[SHIFT_WIDTH_tb-1];
            end
            else begin
                q_exp = {shiftin_tb, q_tb[SHIFT_WIDTH_tb-1:1]};
                shiftout_exp = q_tb[0];
            end
            #1;

            @(negedge clock_tb);
            if (q_tb !== q_exp || shiftout_tb !== shiftout_exp) begin
                $display("error in shift");
                //$stop;
            end
        end

        $display("Test is completed");
        $finish;
    end
endmodule
```

Simulation:



```
/Parameterized_Shift_register_tb/clock_tb      1'h1
/Parameterized_Shift_register_tb/aclr_tb       1'h0
/Parameterized_Shift_register_tb/aset_tb       1'h0
/Parameterized_Shift_register_tb/enable_tb     1'h1
/Parameterized_Shift_register_tb/sclr_tb       1'h0
/Parameterized_Shift_register_tb/sset_tb       1'h0
/Parameterized_Shift_register_tb/load_tb       1'h0
/Parameterized_Shift_register_tb/shiftin_tb    1'h1
/Parameterized_Shift_register_tb/data_tb       8'haf
/Parameterized_Shift_register_tb/q_tb          8'h5f
/Parameterized_Shift_register_tb/q_exp         8'h5f
/Parameterized_Shift_register_tb/shiftout_tb   1'h1
/Parameterized_Shift_register_tb/shiftout_exp  1'h1
```

Lint:



Instance	Module
Parameteri…	Paramet

D:/other/Courses/Kareem_waseem_Digital-Design/Ass/Ass3/Q5/Lint/q5.v [Parameterized_Shift_register]

```
module Parameterized_Shift_register(q, shiftout, sset, aset, data, sclr,
                                    aclr, enable, clock, shiftin, load);

    parameter LOAD_AVALUE = 1;
              1
    parameter LOAD_SVALUE = 1;
              1
    parameter SHIFT_DIRECTION = "LEFT";
              LEFT
    parameter SHIFT_WIDTH = 8;
              8

    input sclr, aclr, enable, clock, shiftin, load, sset, aset;
    input [SHIFT_WIDTH-1:0] data;
           8
    output reg [SHIFT_WIDTH-1:0] q;
                8
    output reg shiftout;

    always @(posedge clock or posedge aclr or posedge aset) begin
        if(aclr)begin
          q <= 0;
        end
        else if (aset) begin
            q <= LOAD_AVALUE;
                 1
        end
        else if(enable)begin
          if(sclr)begin
```

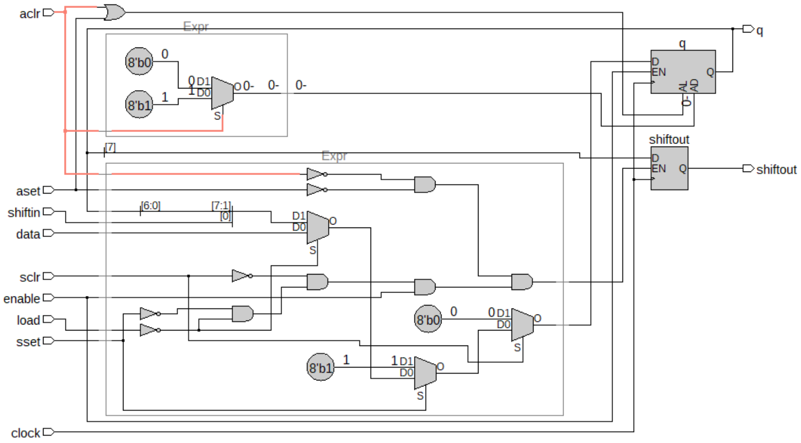
Name	Count
Resolved(verified, fi…	3
Info	3

Total : 0 Selected : 0

Severity	Status	Check	Alias	Message	Module	Category	State	Owner	STARC Reference

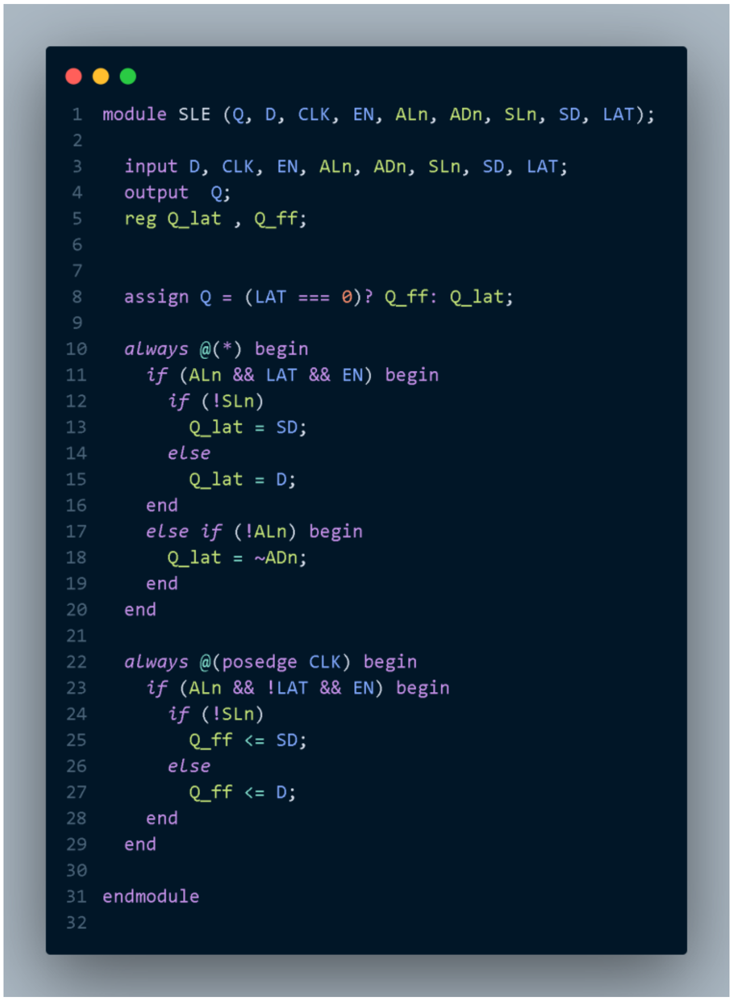
D:/other/Courses/Kareem_waseem_Digital-Design/Ass/Ass3/Q5/Lint/q5.v [Parameterized_Shift_register]

Schematic:



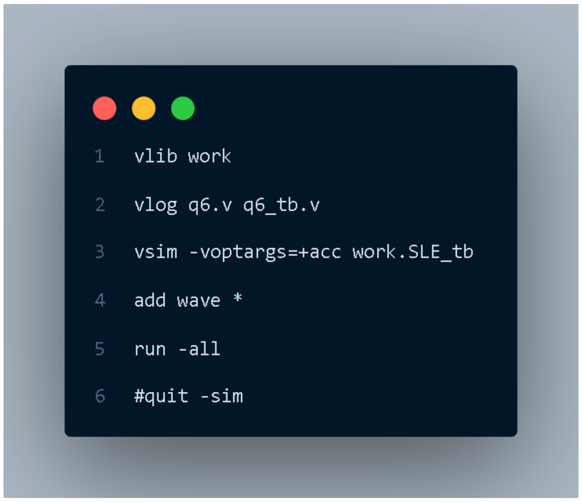
Q6)

SLE code:



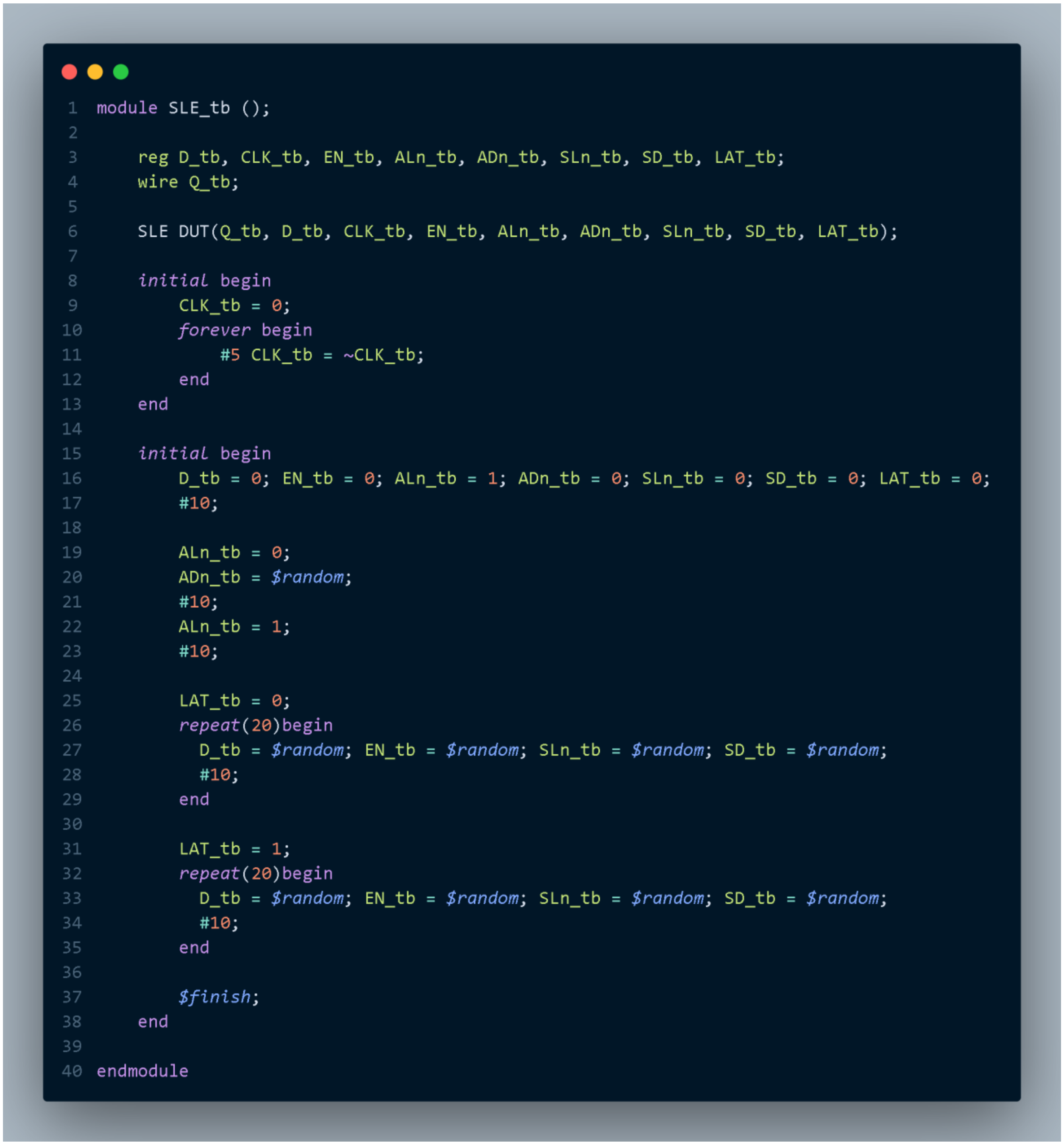
```
module SLE (Q, D, CLK, EN, ALn, ADn, SLn, SD, LAT);

  input D, CLK, EN, ALn, ADn, SLn, SD, LAT;
  output  Q;
  reg Q_lat , Q_ff;


  assign Q = (LAT === 0)? Q_ff: Q_lat;

  always @(*) begin
    if (ALn && LAT && EN) begin
      if (!SLn)
        Q_lat = SD;
      else
        Q_lat = D;
    end
    else if (!ALn) begin
      Q_lat = ~ADn;
    end
  end

  always @(posedge CLK) begin
    if (ALn && !LAT && EN) begin
      if (!SLn)
        Q_ff <= SD;
      else
        Q_ff <= D;
    end
  end

endmodule

```

Do file:



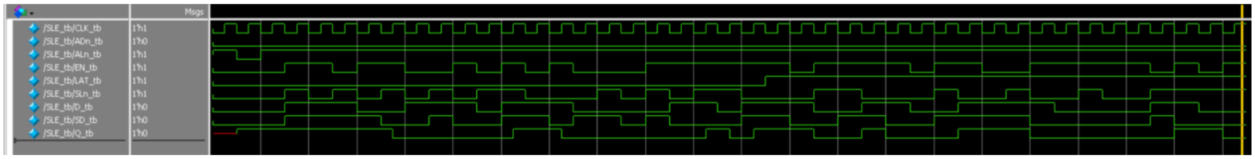
```
vlib work

vlog q6.v q6_tb.v

vsim -voptargs=+acc work.SLE_tb

add wave *

run -all

#quit -sim
```

Testbench code:



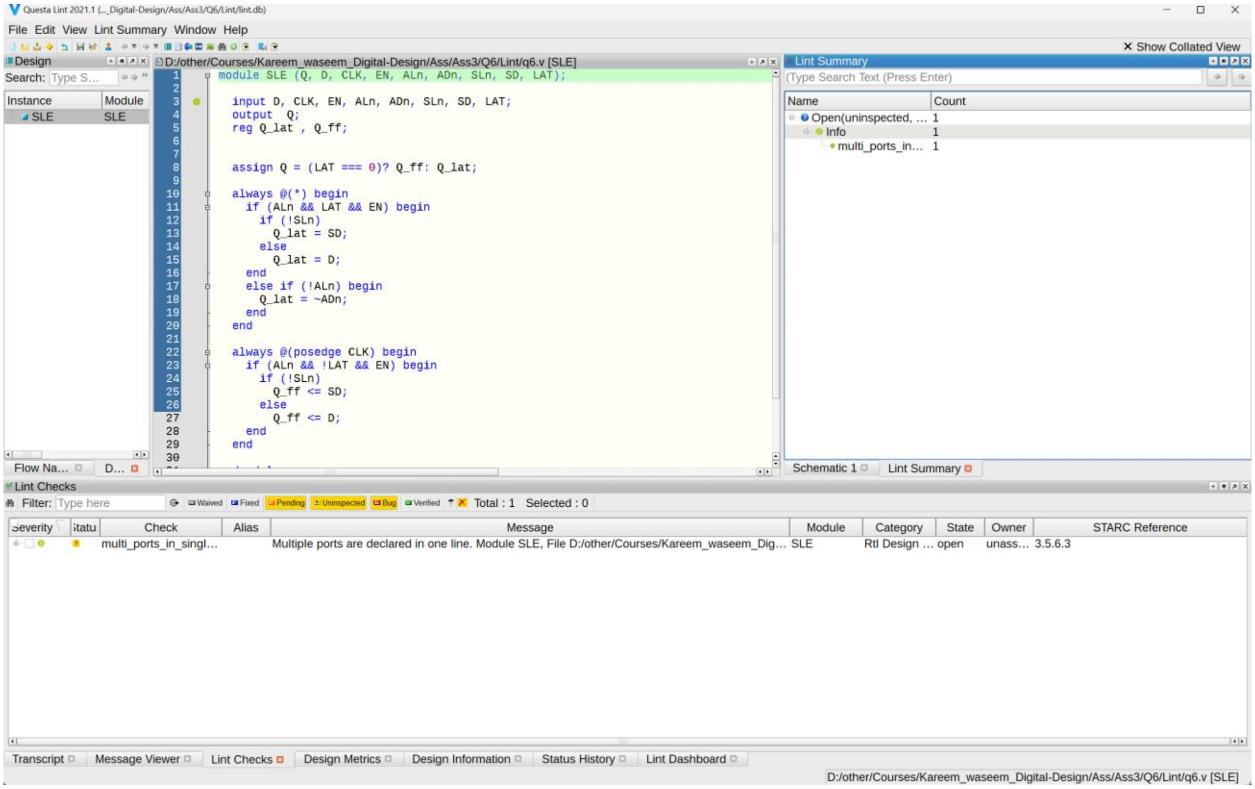
```verilog
module SLE_tb ();

    reg D_tb, CLK_tb, EN_tb, ALn_tb, ADn_tb, SLn_tb, SD_tb, LAT_tb;
    wire Q_tb;

    SLE DUT(Q_tb, D_tb, CLK_tb, EN_tb, ALn_tb, ADn_tb, SLn_tb, SD_tb, LAT_tb);

    initial begin
        CLK_tb = 0;
        forever begin
            #5 CLK_tb = ~CLK_tb;
        end
    end

    initial begin
        D_tb = 0; EN_tb = 0; ALn_tb = 1; ADn_tb = 0; SLn_tb = 0; SD_tb = 0; LAT_tb = 0;
        #10;

        ALn_tb = 0;
        ADn_tb = $random;
        #10;
        ALn_tb = 1;
        #10;

        LAT_tb = 0;
        repeat(20)begin
          D_tb = $random; EN_tb = $random; SLn_tb = $random; SD_tb = $random;
          #10;
        end

        LAT_tb = 1;
        repeat(20)begin
          D_tb = $random; EN_tb = $random; SLn_tb = $random; SD_tb = $random;
          #10;
        end

        $finish;
    end

endmodule
```

Simulation:



```
/SLE_tb/CLK_tb   1'h1
/SLE_tb/ADn_tb   1'h0
/SLE_tb/ALn_tb   1'h1
/SLE_tb/EN_tb    1'h1
/SLE_tb/LAT_tb   1'h1
/SLE_tb/SLn_tb   1'h1
/SLE_tb/D_tb     1'h0
/SLE_tb/SD_tb    1'h0
/SLE_tb/Q_tb     1'h0
```

Lint:



Questa Lint 2021.1 (..._Digital-Design/Ass/Ass3/Q6/Lint/lint.db)
File Edit View Lint Summary Window Help
Show Collated View
Design
Search: Type S...
Instance Module
SLE SLE
Flow Na... D...
D:/other/Courses/Kareem_waseem_Digital-Design/Ass/Ass3/Q6/Lint/q6.v [SLE]
Lint Summary
(Type Search Text (Press Enter)
Schematic 1
Lint Summary
Lint Checks
Filter: Type here
Waived Fixed Pending Uninspected Bug Verified
Transcript Message Viewer Lint Checks Design Metrics Design Information Status History Lint Dashboard
D:/other/Courses/Kareem_waseem_Digital-Design/Ass/Ass3/Q6/Lint/q6.v [SLE]

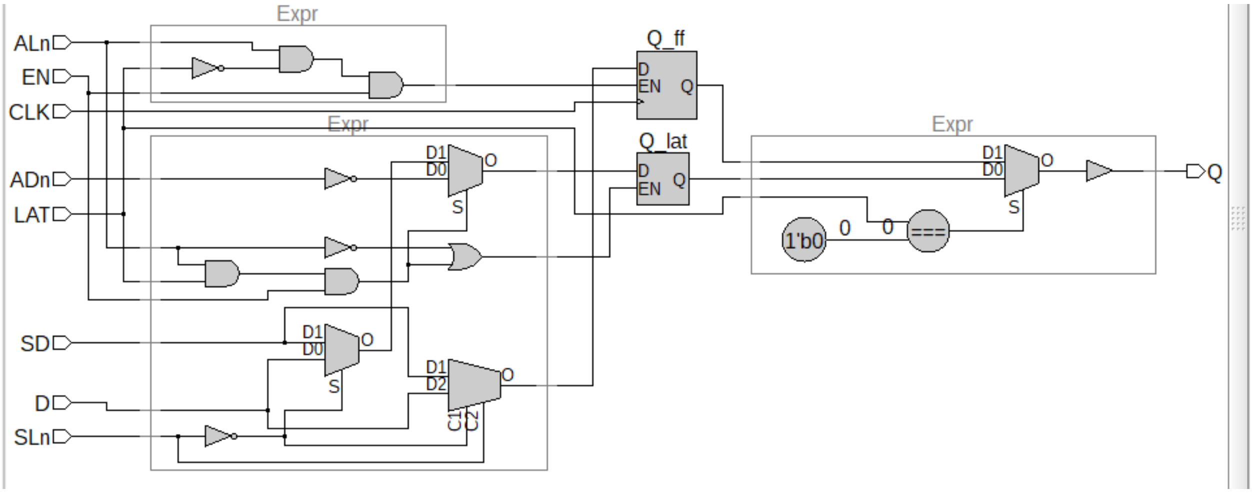
```
module SLE (Q, D, CLK, EN, ALn, ADn, SLn, SD, LAT);

  input D, CLK, EN, ALn, ADn, SLn, SD, LAT;
  output  Q;
  reg Q_lat , Q_ff;


  assign Q = (LAT === 0)? Q_ff: Q_lat;

  always @(*) begin
    if (ALn && LAT && EN) begin
      if (!SLn)
        Q_lat = SD;
      else
        Q_lat = D;
    end
    else if (!ALn) begin
      Q_lat = ~ADn;
    end
  end

  always @(posedge CLK) begin
    if (ALn && !LAT && EN) begin
      if (!SLn)
        Q_ff <= SD;
      else
        Q_ff <= D;
    end
  end

```

Name	Count
Open(uninspected, ...	1
Info	1
multi_ports_in...	1

Total : 1 Selected : 0

Severity	Status	Check	Alias	Message	Module	Category	State	Owner	STARC Reference
		multi_ports_in_singl...		Multiple ports are declared in one line. Module SLE, File D:/other/Courses/Kareem_waseem_Dig...	SLE	Rtl Design ...	open	unass...	3.5.6.3

Schematic:



Expr
Expr
Expr
ALn
EN
CLK
ADn
LAT
SD
D
SLn
Q_ff
D
EN
Q
Q_lat
D
EN
Q
D1
D0
O
S
D1
D0
O
S
D1
D2
O
C1
C2
D1
D0
O
S
1'b0
0
0
===
Q