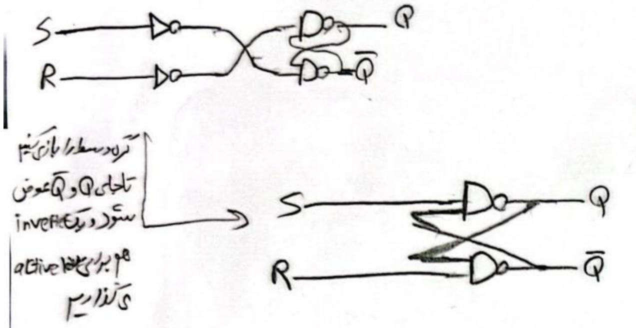
Amin Aghakasiri 810101381

CA4 Report

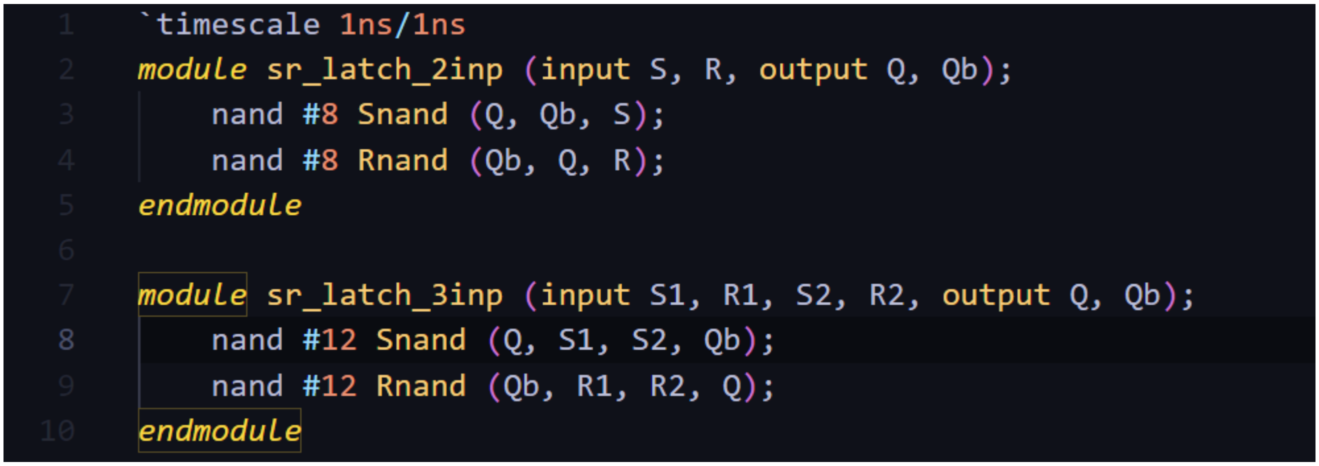
Question1:

Part a , b:



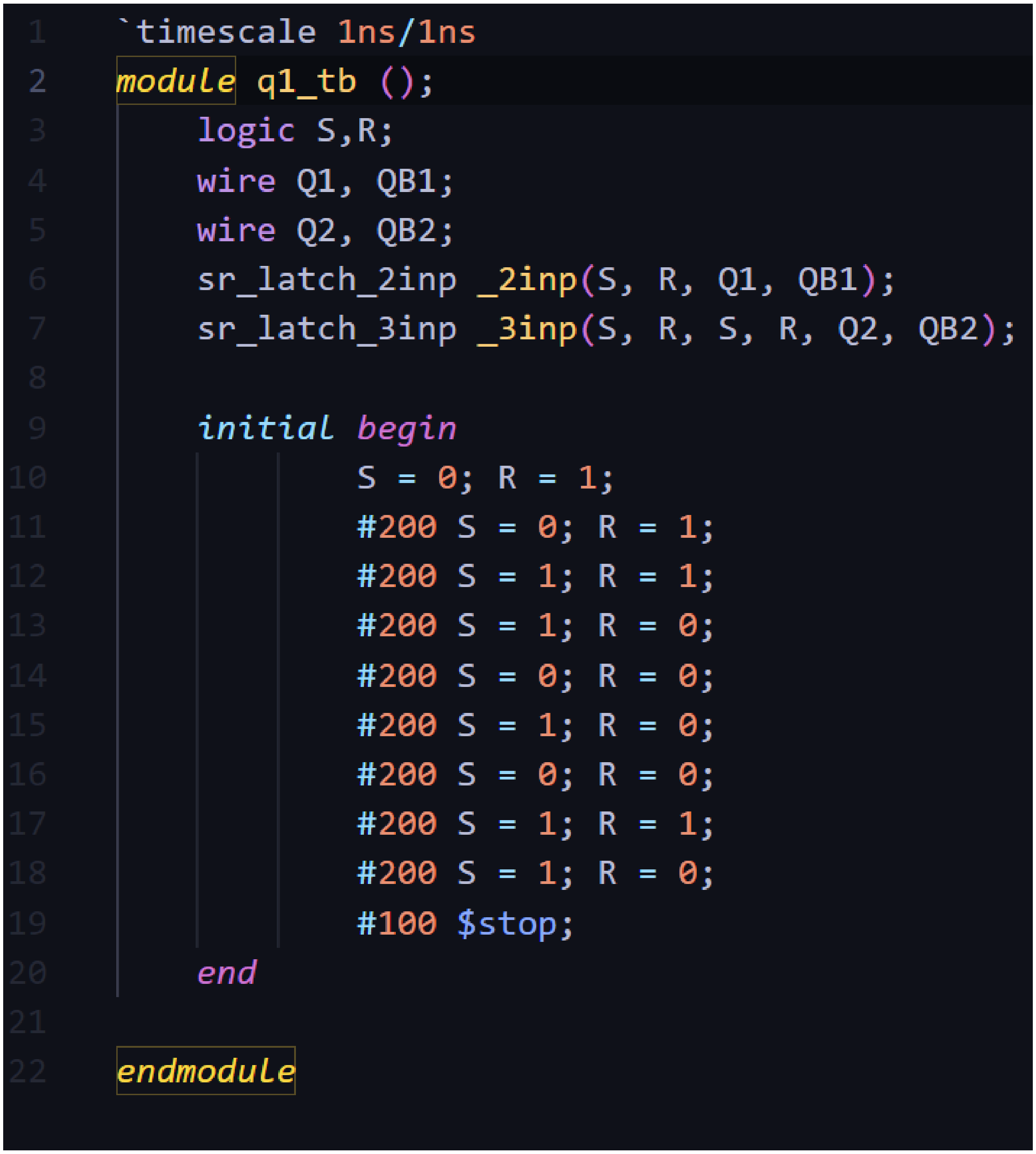
S	R	Q^+
1	1	Q
0	1	1
1	0	0
0	0	memory loss

Code:

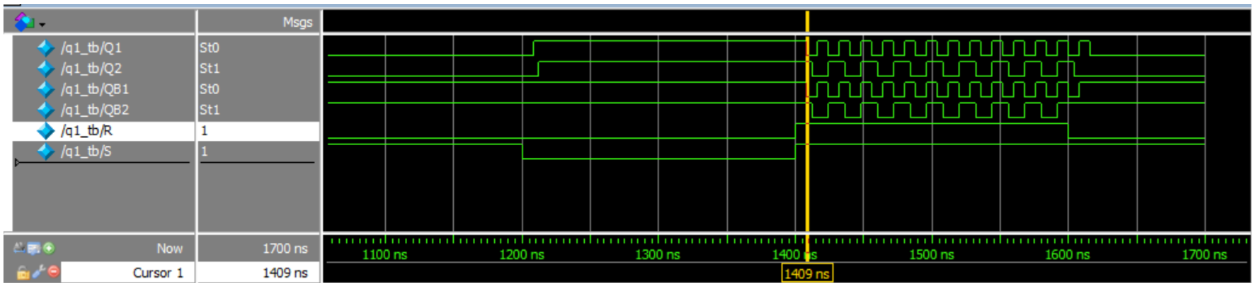


```
`timescale 1ns/1ns
module sr_latch_2inp (input S, R, output Q, Qb);
    nand #8 Snand (Q, Qb, S);
    nand #8 Rnand (Qb, Q, R);
endmodule

module sr_latch_3inp (input S1, R1, S2, R2, output Q, Qb);
    nand #12 Snand (Q, S1, S2, Qb);
    nand #12 Rnand (Qb, R1, R2, Q);
endmodule
```

Test bench:



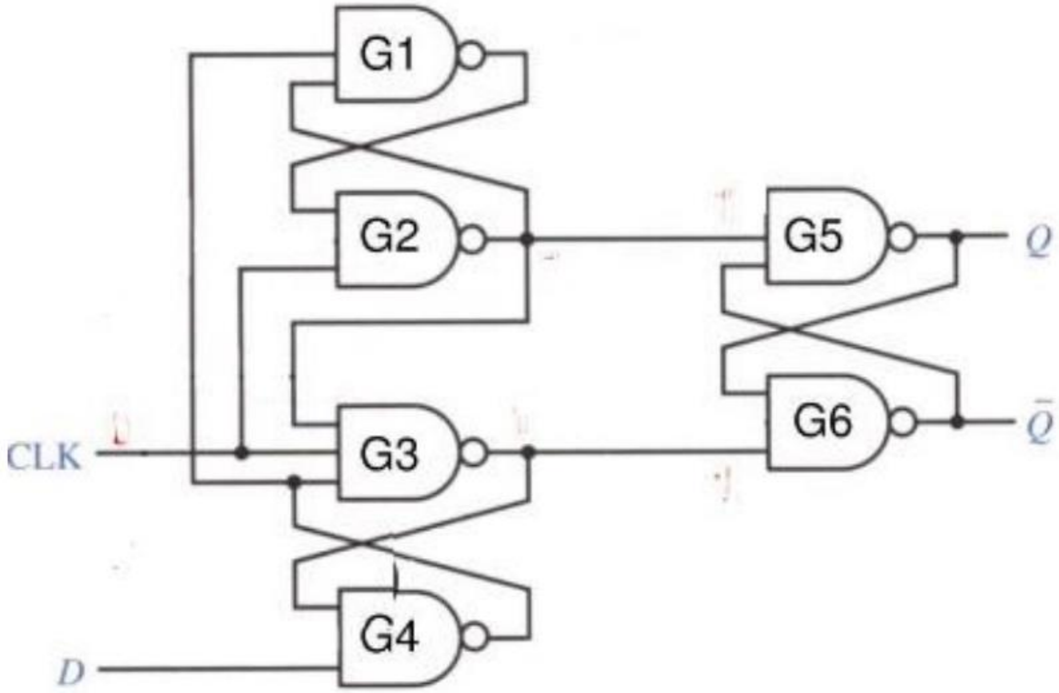
```
`timescale 1ns/1ns
module q1_tb ();
    logic S,R;
    wire Q1, QB1;
    wire Q2, QB2;
    sr_latch_2inp _2inp(S, R, Q1, QB1);
    sr_latch_3inp _3inp(S, R, S, R, Q2, QB2);

    initial begin
            S = 0; R = 1;
            #200 S = 0; R = 1;
            #200 S = 1; R = 1;
            #200 S = 1; R = 0;
            #200 S = 0; R = 0;
            #200 S = 1; R = 0;
            #200 S = 0; R = 0;
            #200 S = 1; R = 1;
            #200 S = 1; R = 0;
            #100 $stop;
    end

endmodule
```



Part c:

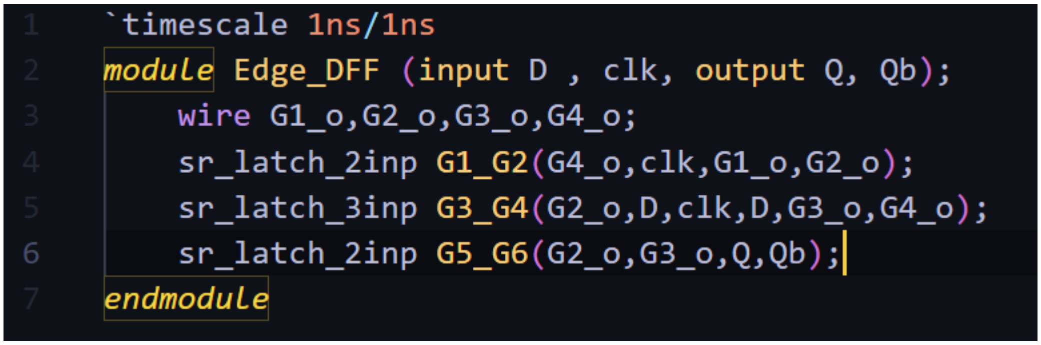
If S and R are simultaneously active (00 in this case because they are active low), there will be memory loss. For example if the input after 00 is 11, the output must be Q but because of the loss of memory, the output is unknown(goes into a loop of 0 and 1’s)

Question 2:



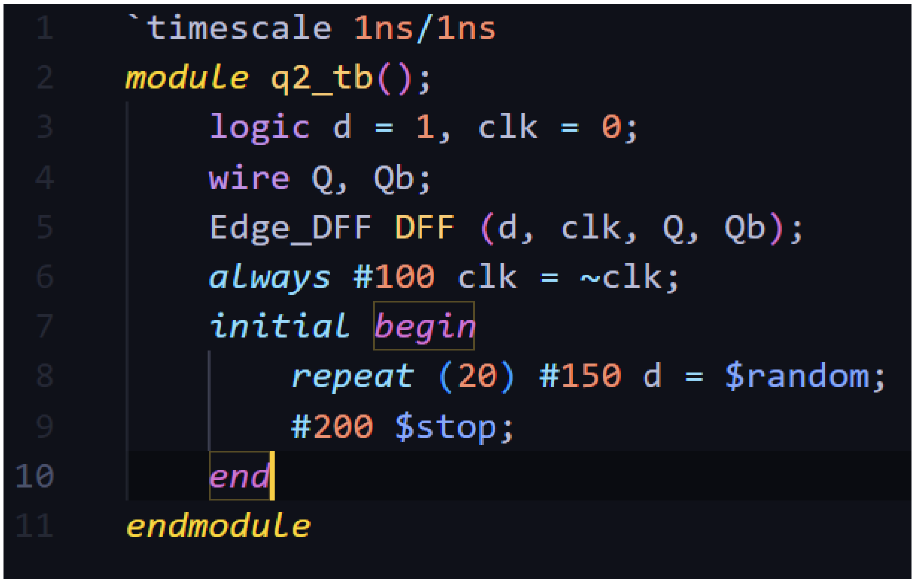
Part a:

Code:



```
`timescale 1ns/1ns
module Edge_DFF (input D , clk, output Q, Qb);
    wire G1_o,G2_o,G3_o,G4_o;
    sr_latch_2inp G1_G2(G4_o,clk,G1_o,G2_o);
    sr_latch_3inp G3_G4(G2_o,D,clk,D,G3_o,G4_o);
    sr_latch_2inp G5_G6(G2_o,G3_o,Q,Qb);
endmodule
```

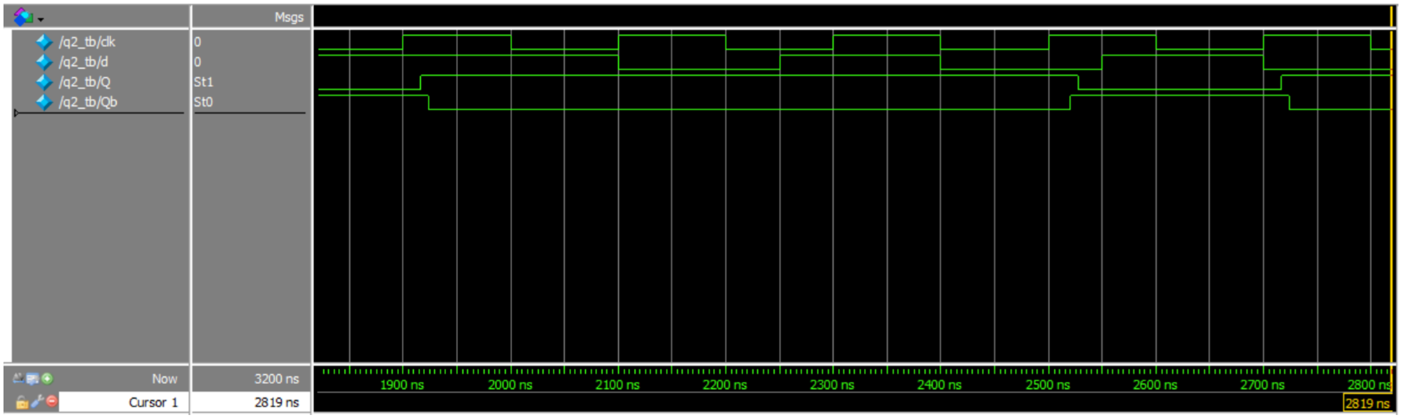
Testbench:



```
`timescale 1ns/1ns
module q2_tb();
    logic d = 1, clk = 0;
    wire Q, Qb;
    Edge_DFF DFF (d, clk, Q, Qb);
    always #100 clk = ~clk;
    initial begin
        repeat (20) #150 d = $random;
        #200 $stop;
    end
endmodule
```

Part b:

Testbench:

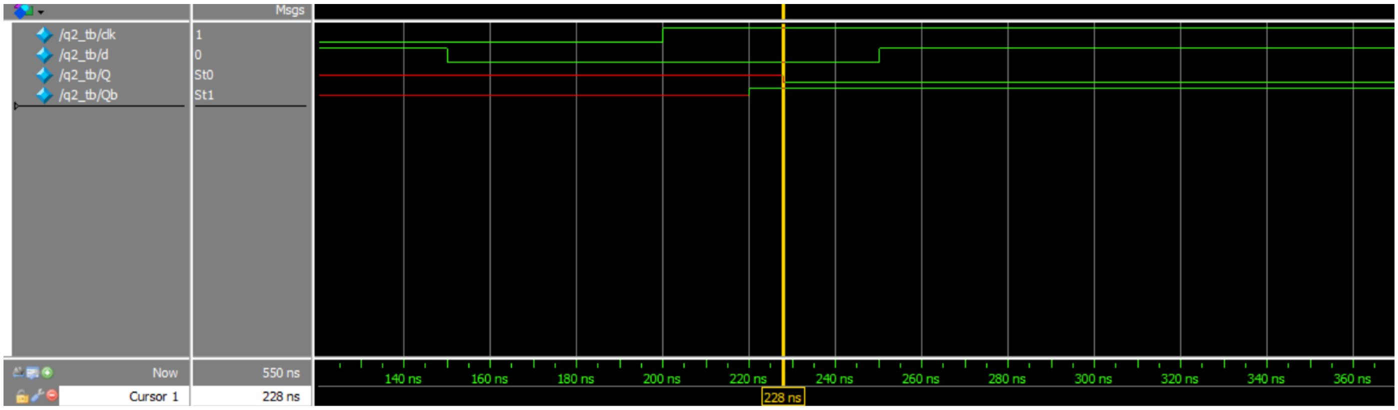


At the positive edge of clock, the output Q becomes the d value.

When d changes from 1 to 0, the delay for changing Q and Qb are:

Q : 28ns

Qb: 20ns



	Msgs
/q2_tb/clk	1
/q2_tb/d	0
/q2_tb/Q	St0
/q2_tb/Qb	St1

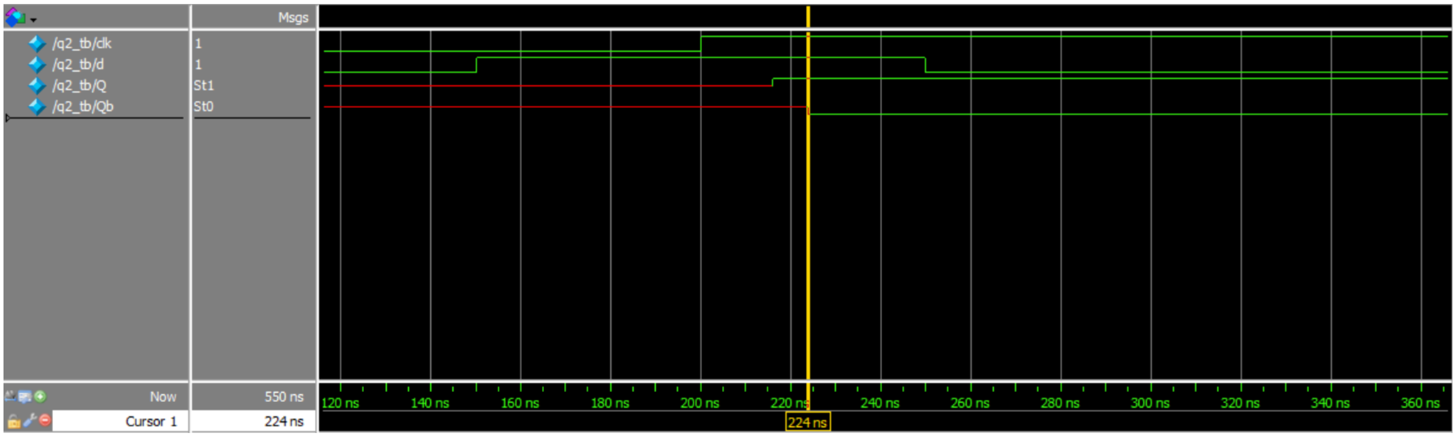
Now 550 ns

Cursor 1 228 ns

When d changes from 0 to 1, the delay for changing Q and Qb are:

Q : 16ns

Qb: 24ns



	Msgs
/q2_tb/clk	1
/q2_tb/d	1
/q2_tb/Q	St1
/q2_tb/Qb	St0

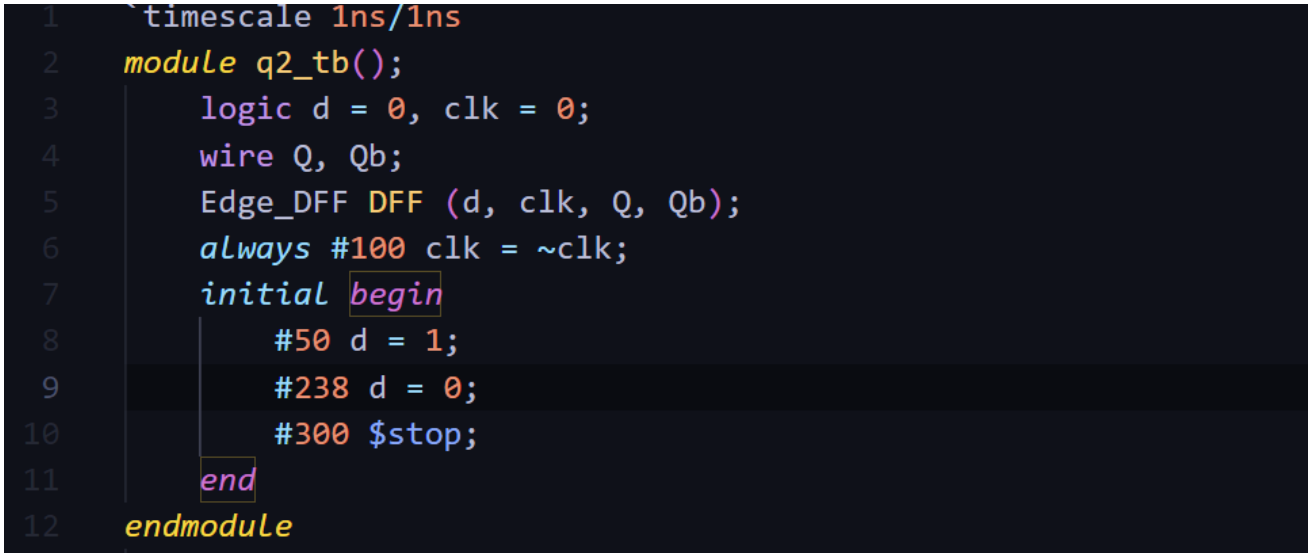
Now 550 ns

Cursor 1 224 ns

Part c:

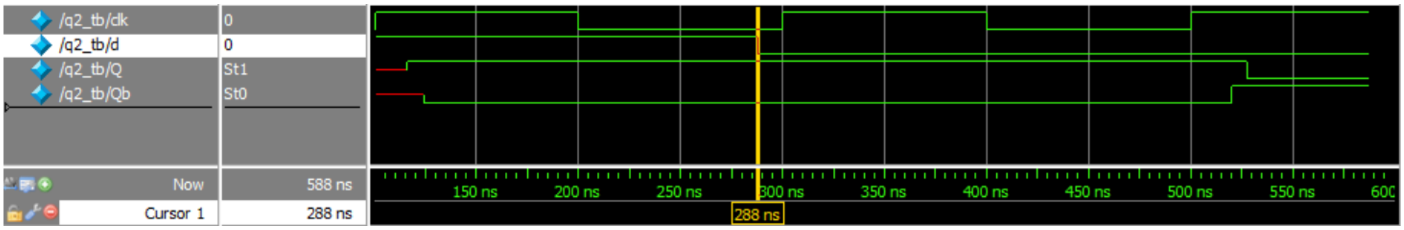
Tsetup = (12 + epsilon)ns → 13ns

Testbench for calculating Tsetup:



```
`timescale 1ns/1ns
module q2_tb();
    logic d = 0, clk = 0;
    wire Q, Qb;
    Edge_DFF DFF (d, clk, Q, Qb);
    always #100 clk = ~clk;
    initial begin
        #50 d = 1;
        #238 d = 0;
        #300 $stop;
    end
endmodule
```

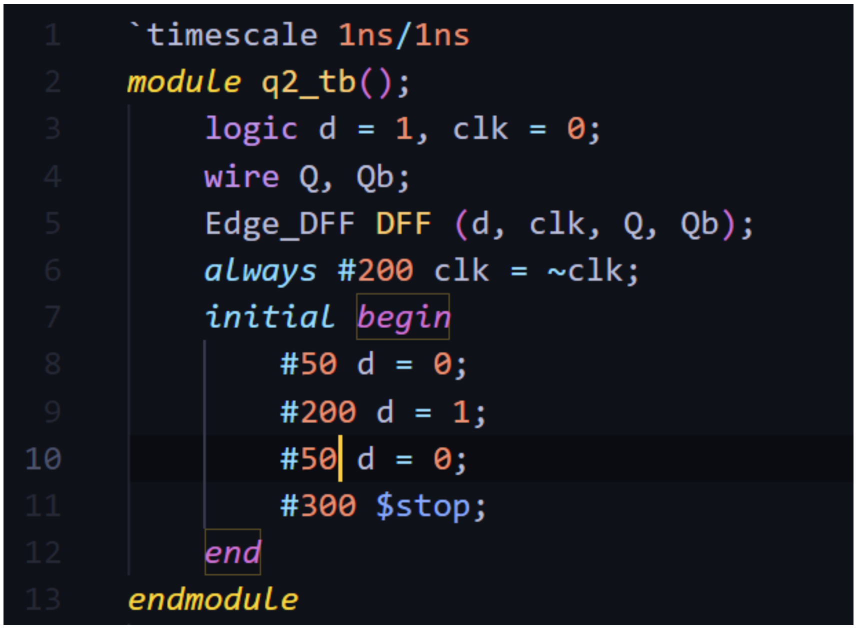
Waveform for calculating Tsetup:



Part d:

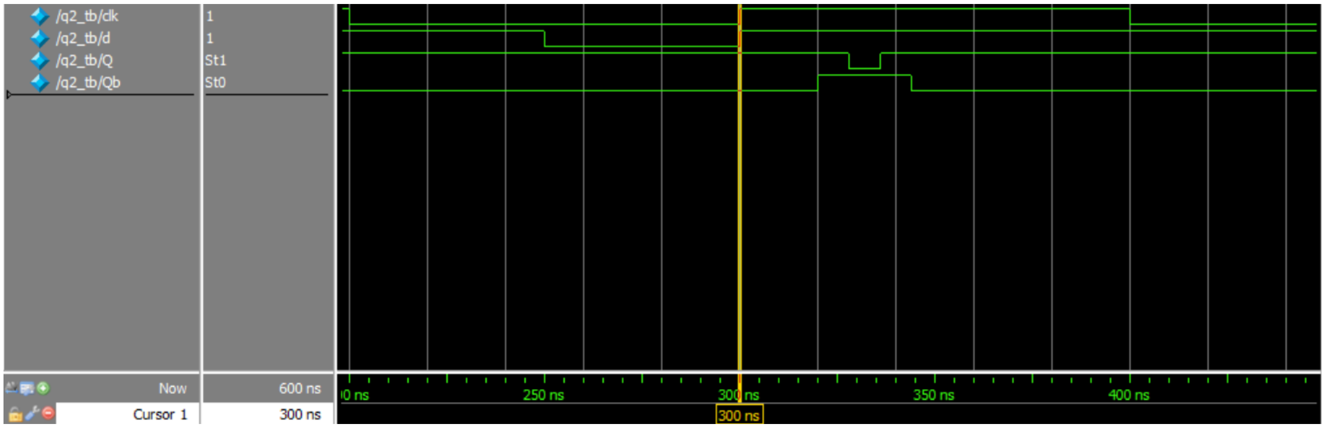
Thold = (0 + epsilon)ns → 1ns

Testbench for calculating Thold:

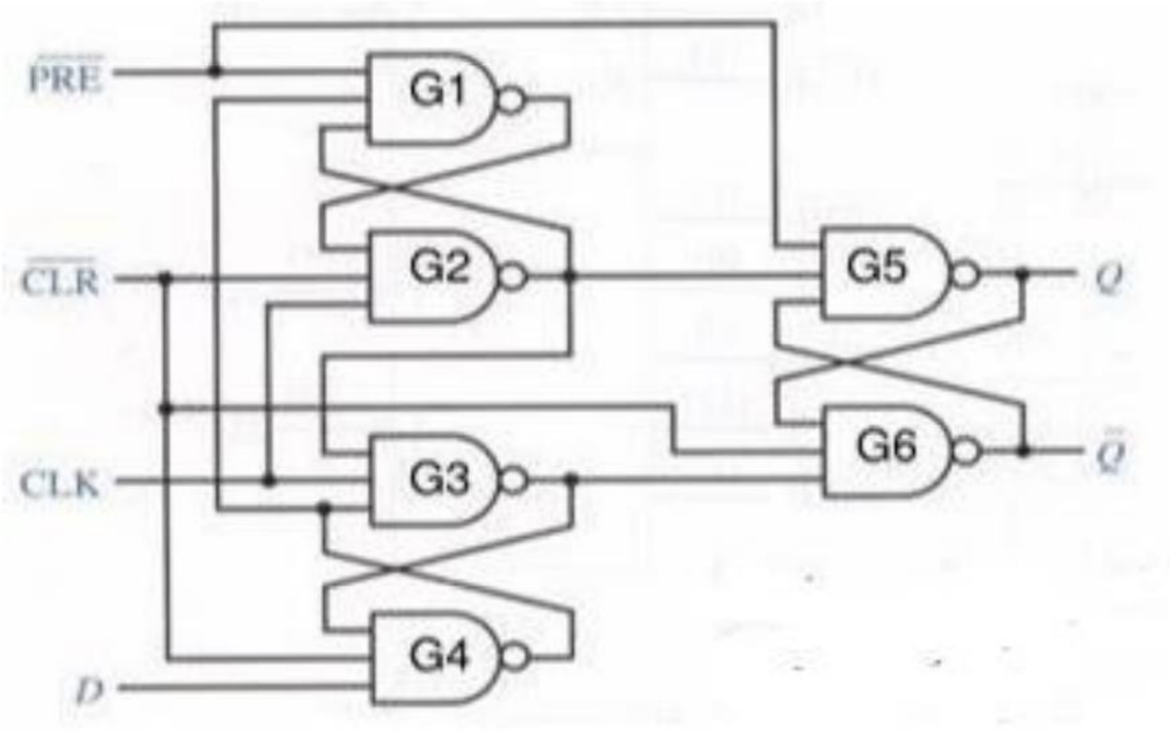


```
`timescale 1ns/1ns
module q2_tb();
    logic d = 1, clk = 0;
    wire Q, Qb;
    Edge_DFF DFF (d, clk, Q, Qb);
    always #200 clk = ~clk;
    initial begin
        #50 d = 0;
        #200 d = 1;
        #50 d = 0;
        #300 $stop;
    end
endmodule
```

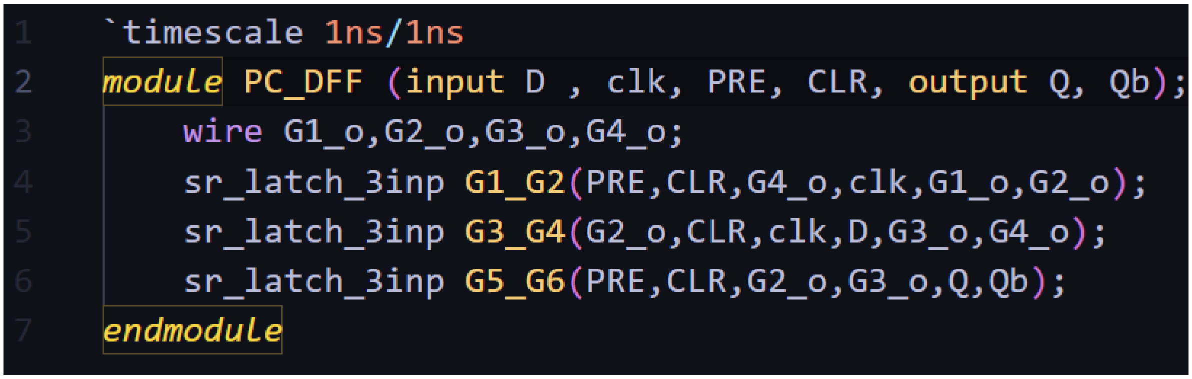
Waveform for calculating Thold:



Question 3:

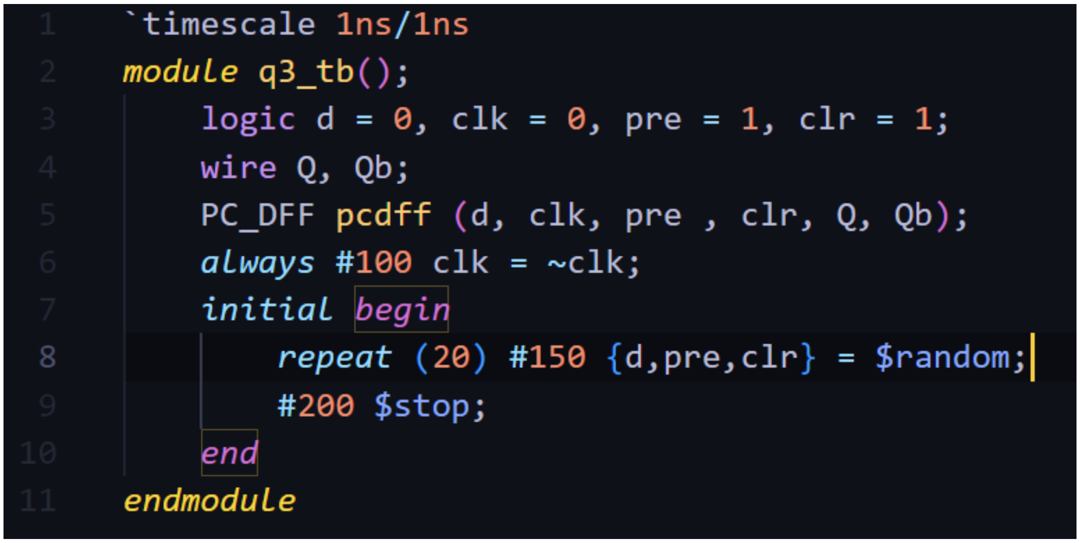


Part e:



```
`timescale 1ns/1ns
module PC_DFF (input D , clk, PRE, CLR, output Q, Qb);
    wire G1_o,G2_o,G3_o,G4_o;
    sr_latch_3inp G1_G2(PRE,CLR,G4_o,clk,G1_o,G2_o);
    sr_latch_3inp G3_G4(G2_o,CLR,clk,D,G3_o,G4_o);
    sr_latch_3inp G5_G6(PRE,CLR,G2_o,G3_o,Q,Qb);
endmodule
```

Testbench:



```
`timescale 1ns/1ns
module q3_tb();
    logic d = 0, clk = 0, pre = 1, clr = 1;
    wire Q, Qb;
    PC_DFF pcdff (d, clk, pre , clr, Q, Qb);
    always #100 clk = ~clk;
    initial begin
        repeat (20) #150 {d,pre,clr} = $random;
        #200 $stop;
    end
endmodule
```

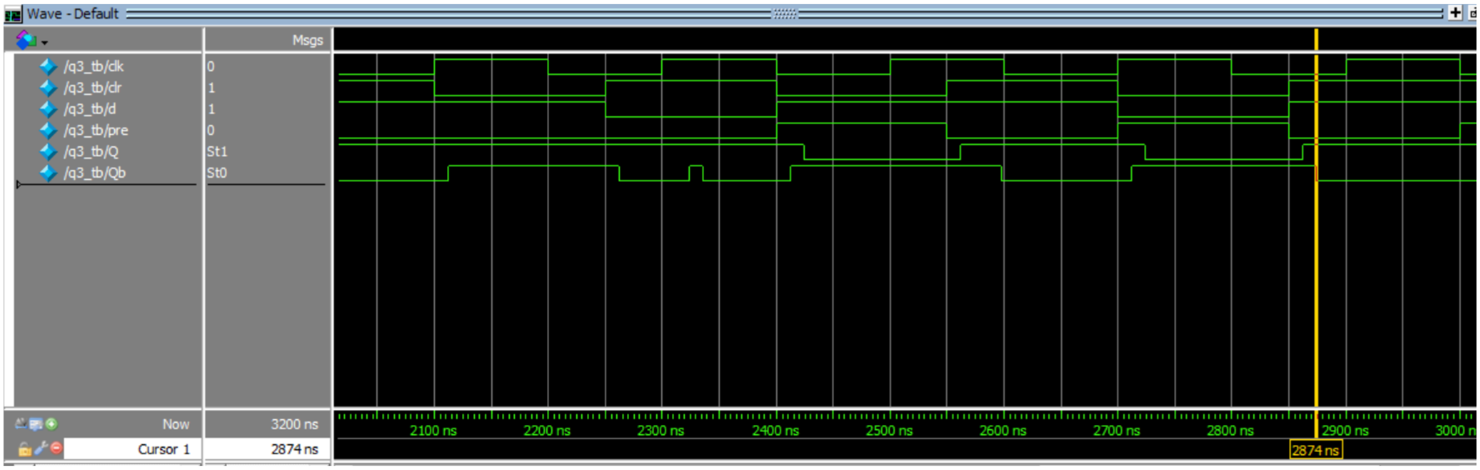
Part f:

change D from 0 to 1: Q = 24ns and Qb = 36ns

change D from 1 to 0: Q = 36ns and Qb = 24ns

PRE active: Q = 12ns and Qb = 48ns

CLR active: Q = 24ns and Qb = 12ns



Wave - Default

	Msgs
/q3_tb/clk	0
/q3_tb/clr	1
/q3_tb/d	1
/q3_tb/pre	0
/q3_tb/Q	St1
/q3_tb/Qb	St0
Now	3200 ns
Cursor 1	2874 ns

Part g:

Nothing happens if clock activates while preset or clear are already active because preset and clear are asynchronous in this flip flop and they have priority to clock.

Part h:

Active Preset: Q = 1

Active CLR : Q = 0

Active CLR and Preset simultaneously: Memory loss(Q = Qb)