In LAB: 1. Implement R-S Latch in Verilog and simulate

Code: `timescale 1ns / 1ps

```
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date:    18:48:05 05/01/2018
// Design Name:
// Module Name:    RSLatch
// Project Name:
// Target Devices:
// Tool versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
module RSLatch(
    input R,
    input S,
    output Q,
    output NQ
```

```verilog
    );

        nor(Q, R, NQ);

        nor(NQ, S, Q);


endmodule
```

Test Code:

```verilog
`timescale 1ns / 1ps
```

```verilog
//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date:    18:49:49 05/01/2018

// Design Name:    RSLatch

// Module Name:    /tmp/tmp.TbiQs2mH3q/RSLatch/RSLatch_test.v

// Project Name:   RSLatch

// Target Device:

// Tool versions:

// Description:

//

// Verilog Test Fixture created by ISE for module: RSLatch

//

// Dependencies:

//
```

```verilog
// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////

module RSLatch_test;


        // Inputs

        reg R;

        reg S;


        // Outputs

        wire Q;

        wire NQ;


        // Instantiate the Unit Under Test (UUT)

        RSLatch uut (

                .R(R),

                .S(S),

                .Q(Q),

                .NQ(NQ)

        );


        initial begin

                // Initialize Inputs
```

```verilog
      R = 0;

      S = 0;


      // Wait 100 ns for global reset to finish

      #100;


      // Add stimulus here

      R = 1; S = 0;


      #100 R = 0; S = 1;


      #100 R = 0; S = 0;


      #100 R = 1; S = 1;



   end


endmodule
```
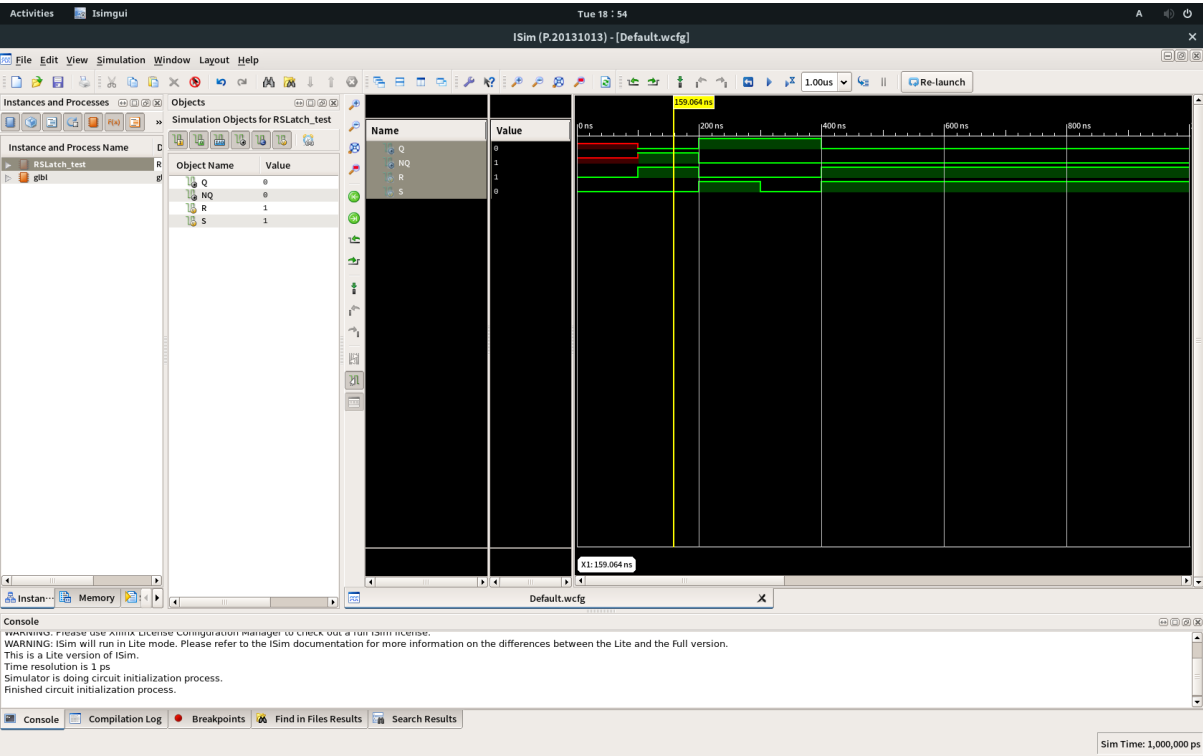
실행 결과:

2. Implement gated R-S Latch in Verilog and simulate

Code: `timescale 1ns / 1ps

```verilog
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date:    18:55:28 05/01/2018
// Design Name:
// Module Name:    GatedRSLatch
// Project Name:
// Target Devices:
// Tool versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
module GatedRSLatch(
    input R,
    input E,
    input S,
    output Q,
```

```verilog
    output NQ

    );


        wire TR, TS;


        and(TR, R, E);

        and(TS, S, E);


        nor(Q, TR, NQ);

        nor(NQ, TS, Q);


endmodule
```

Test Code:

```verilog
`timescale 1ns / 1ps


//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date:    18:56:51 05/01/2018
// Design Name:    GatedRSLatch
// Module Name:    /tmp/tmp.TbiQs2mH3q/RSLatch/GatedRSLatch_test.v
// Project Name:   RSLatch
// Target Device:
// Tool versions:
```

```verilog
// Description:

//

// Verilog Test Fixture created by ISE for module: GatedRSLatch

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////

module GatedRSLatch_test;

        // Inputs

        reg R;

        reg E;

        reg S;


        // Outputs

        wire Q;

        wire NQ;


        // Instantiate the Unit Under Test (UUT)

        GatedRSLatch uut (

                .R(R),
```

```verilog
        .E(E),

        .S(S),

        .Q(Q),

        .NQ(NQ)

);


initial begin

        // Initialize Inputs

        R = 0;

        E = 0;

        S = 0;


        // Wait 100 ns for global reset to finish

        #100;


        // Add stimulus here

        R = 1; S = 0;


        #100 R = 0; S = 1;


        #100 R = 0; S = 0;


        #100 R = 1; S = 1;


        #100 E = 1; R = 0; S = 0;
```

```
          #100 R = 1; S = 0;




          #100 R = 0; S = 1;




          #100 R = 0; S = 0;




          #100 R = 1; S = 1;




     end


endmodule
```
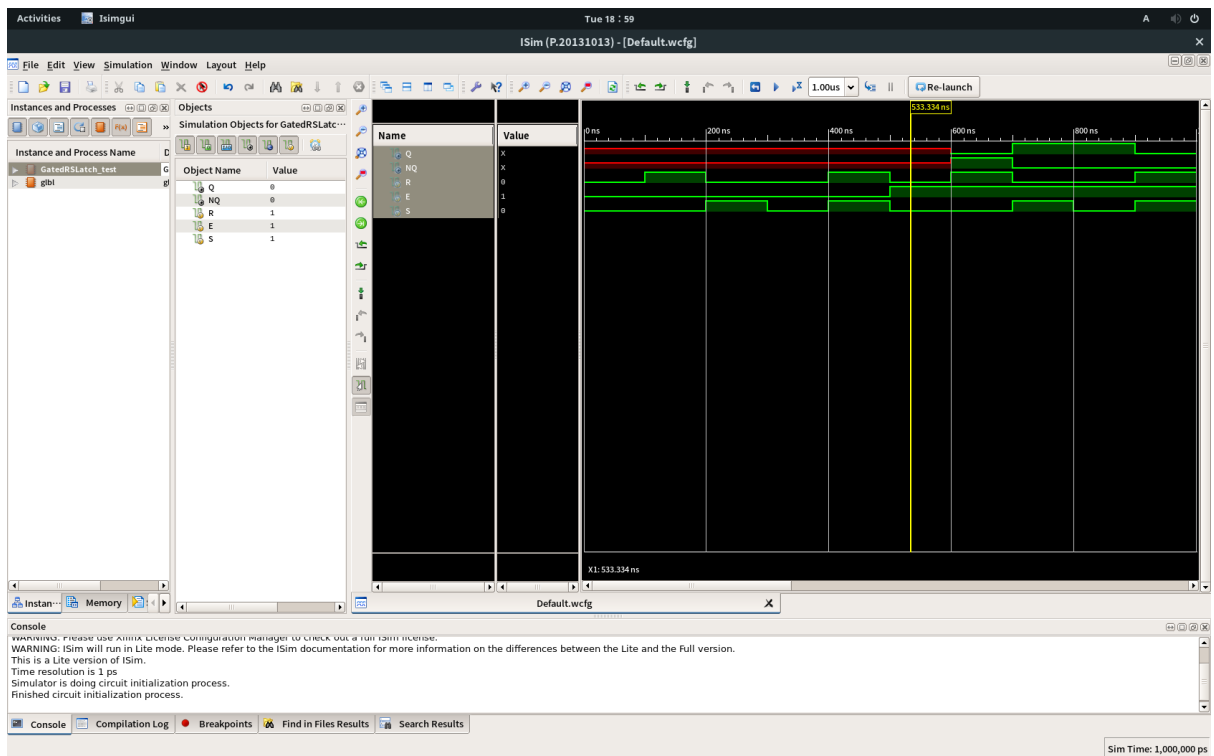
실행 결과:

3. Implement simple oscillator in Verilog and simulate

Code:

```verilog
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date:     19:12:49 05/01/2018
// Design Name:
// Module Name:     Oscillator
// Project Name:
// Target Devices:
// Tool versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
module Oscillator(
    input E,
    output F
    );
```

```verilog
        wire tmp;

        assign tmp = E&F;

        assign #5 F = !tmp;


endmodule
```

Test Code:

```verilog
`timescale 1ns / 1ps


//////////////////////////////////////////////////////////////////////
// Company:

// Engineer:

//

// Create Date:    19:14:50 05/01/2018

// Design Name:    Oscillator

// Module Name:    /tmp/tmp.TbiQs2mH3q/RSLatch/Oscillator_test.v

// Project Name:   RSLatch

// Target Device:

// Tool versions:

// Description:

//

// Verilog Test Fixture created by ISE for module: Oscillator

//

// Dependencies:

//

// Revision:
```

```verilog
// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////


module Oscillator_test;


        // Inputs

        reg E;


        // Outputs

        wire F;


        // Instantiate the Unit Under Test (UUT)

        Oscillator uut (

                .E(E),

                .F(F)

        );


        initial begin

                // Initialize Inputs

                E = 0;


                // Wait 100 ns for global reset to finish

                #100;
```

```
                    // Add stimulus here


                    E = 1;



            end



endmodule
```

실행 결과:

Homework: No Homework!