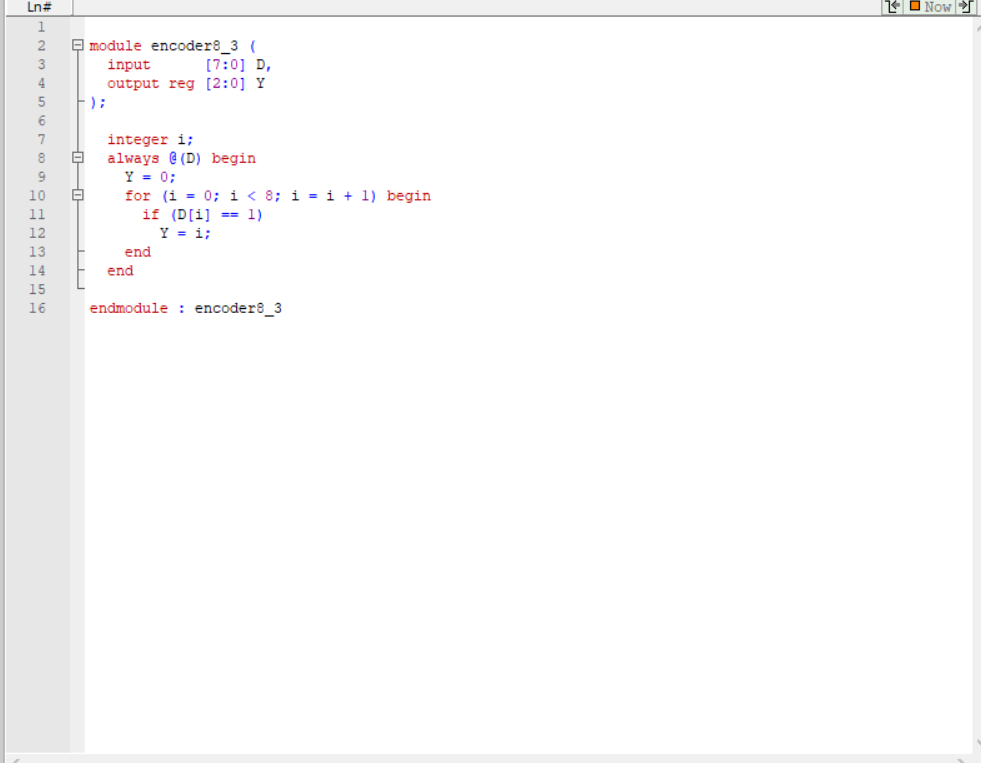
VHDL Week 7

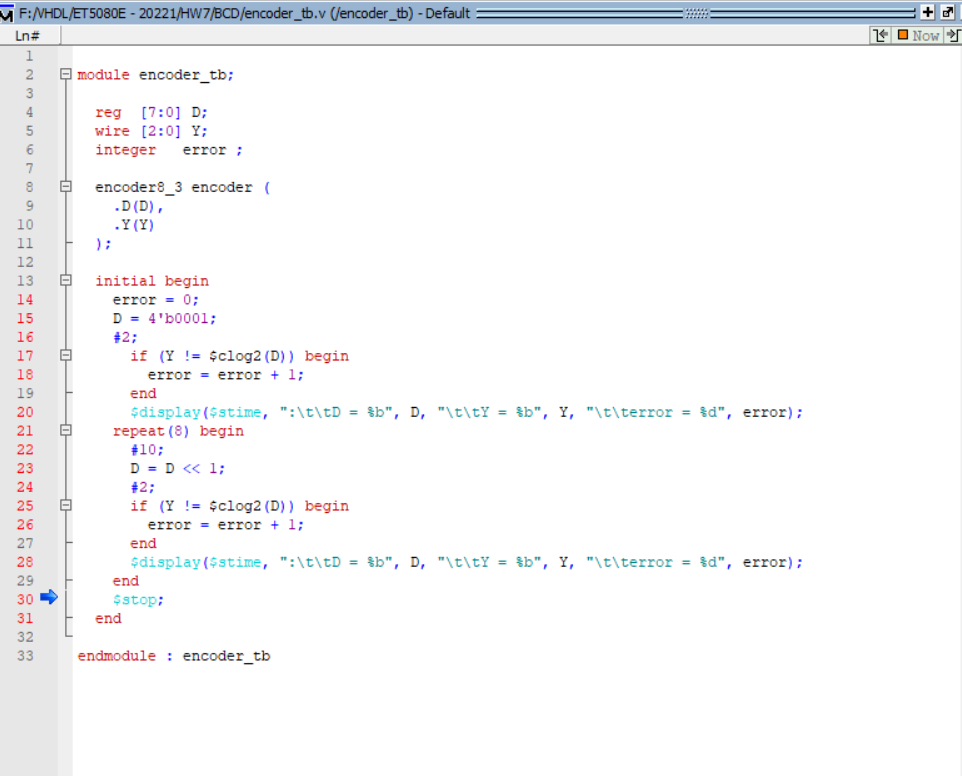
**Ex1**

**1 Encoder 8x3**

Verilog



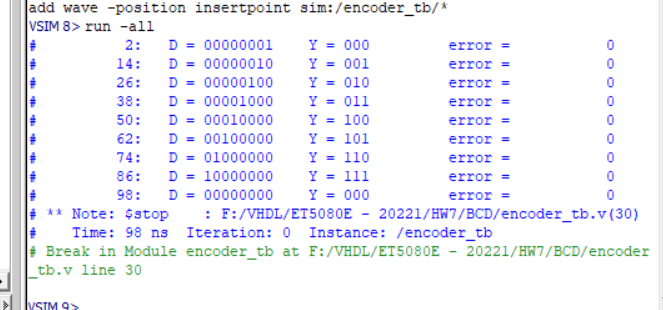
Testbench



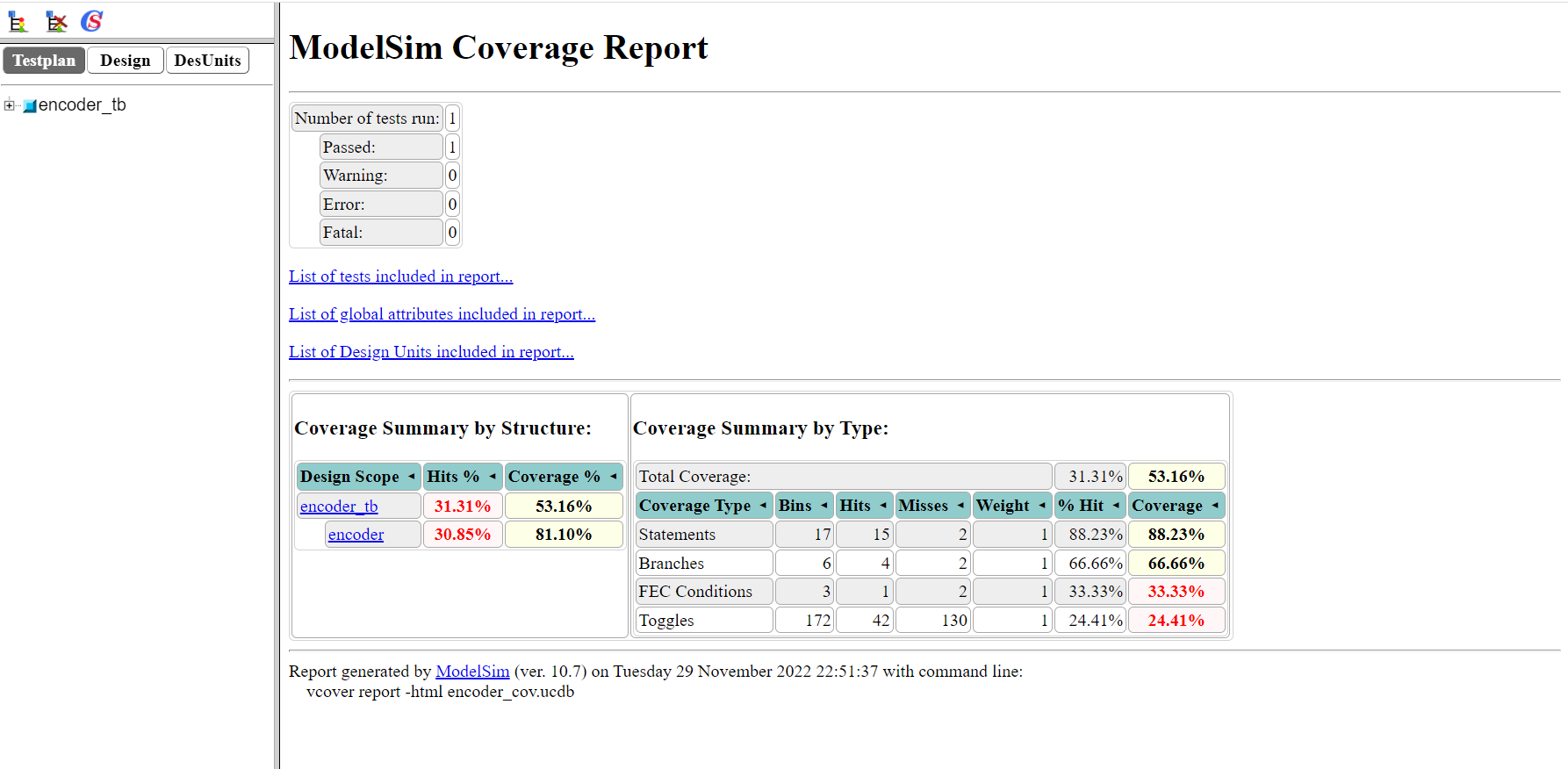
Wave



Script

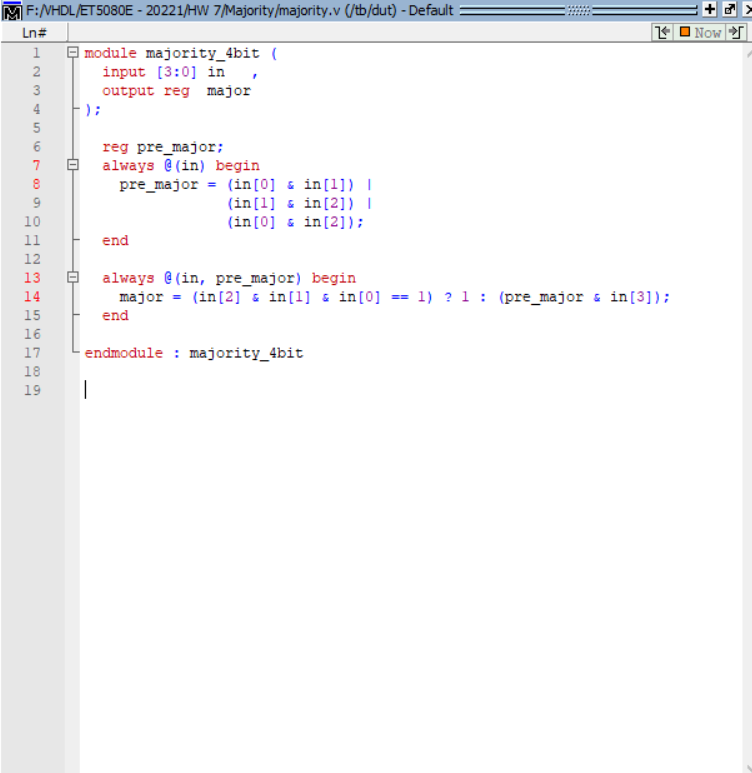


Report



**2 Majority 4 bit**

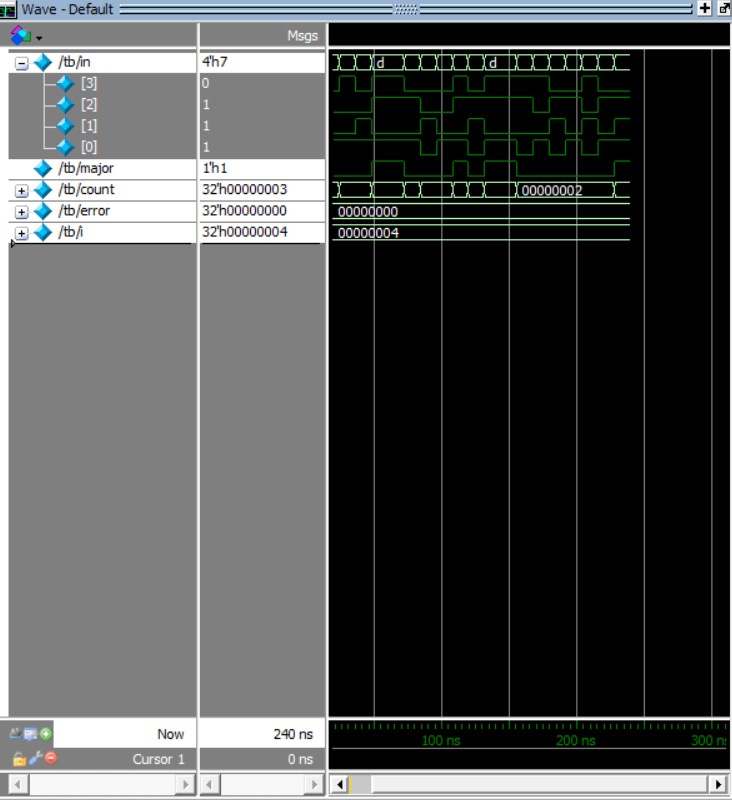
Verilog



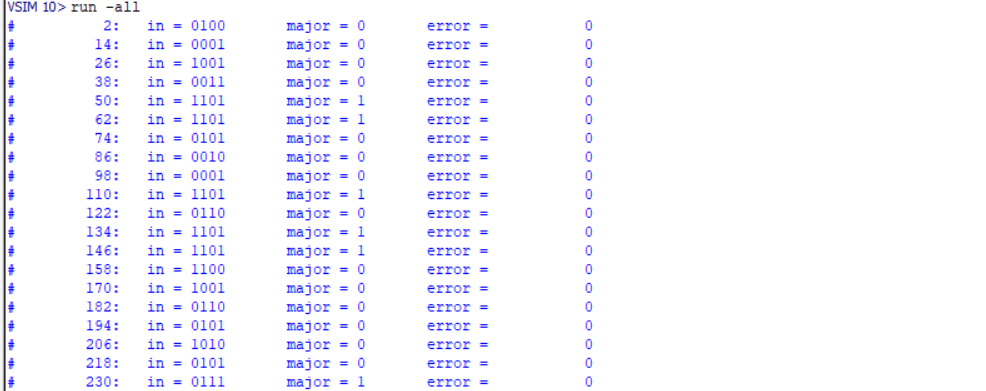
Testbench



Wave



Script



Report



**3 BCD 7-segment**

Verilog

module encoder\_7segment (

input [3:0] in ,

output reg [6:0] out

);

always @(in) begin

case (in)

4'b0000: begin

out = 7'b1111110;

end

4'b0001: begin

out = 7'b0110000;

end

4'b0010: begin

out = 7'b1101101;

end

4'b0011: begin

out = 7'b1111001;

end

4'b0100: begin

out = 7'b0110011;

end

4'b0101: begin

out = 7'b1011011;

end

4'b0110: begin

out = 7'b1011111;

end

4'b0111: begin

out = 7'b1110000;

end

4'b1000: begin

out = 7'b1111111;

end

4'b1001: begin

out = 7'b1111011;

end

default : begin

out = 7'b1111111;

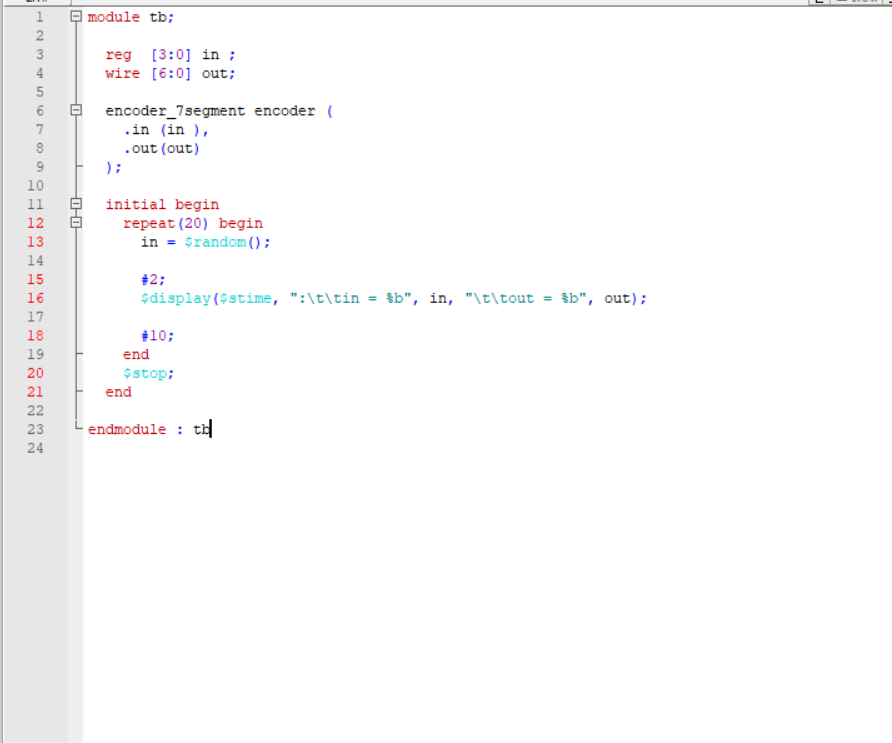
end

endcase

end

endmodule : encoder\_7segment

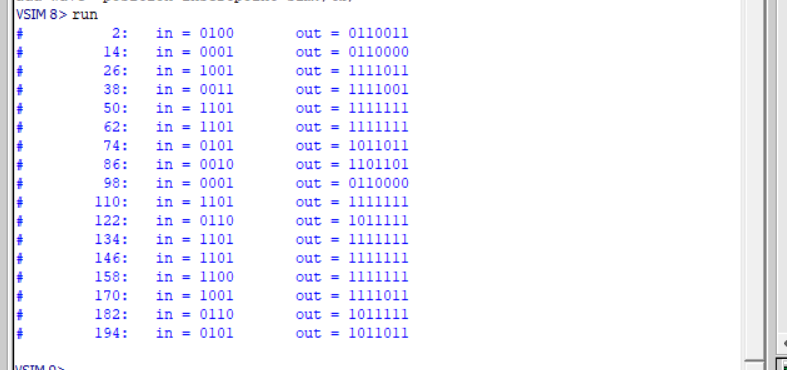
Testbench



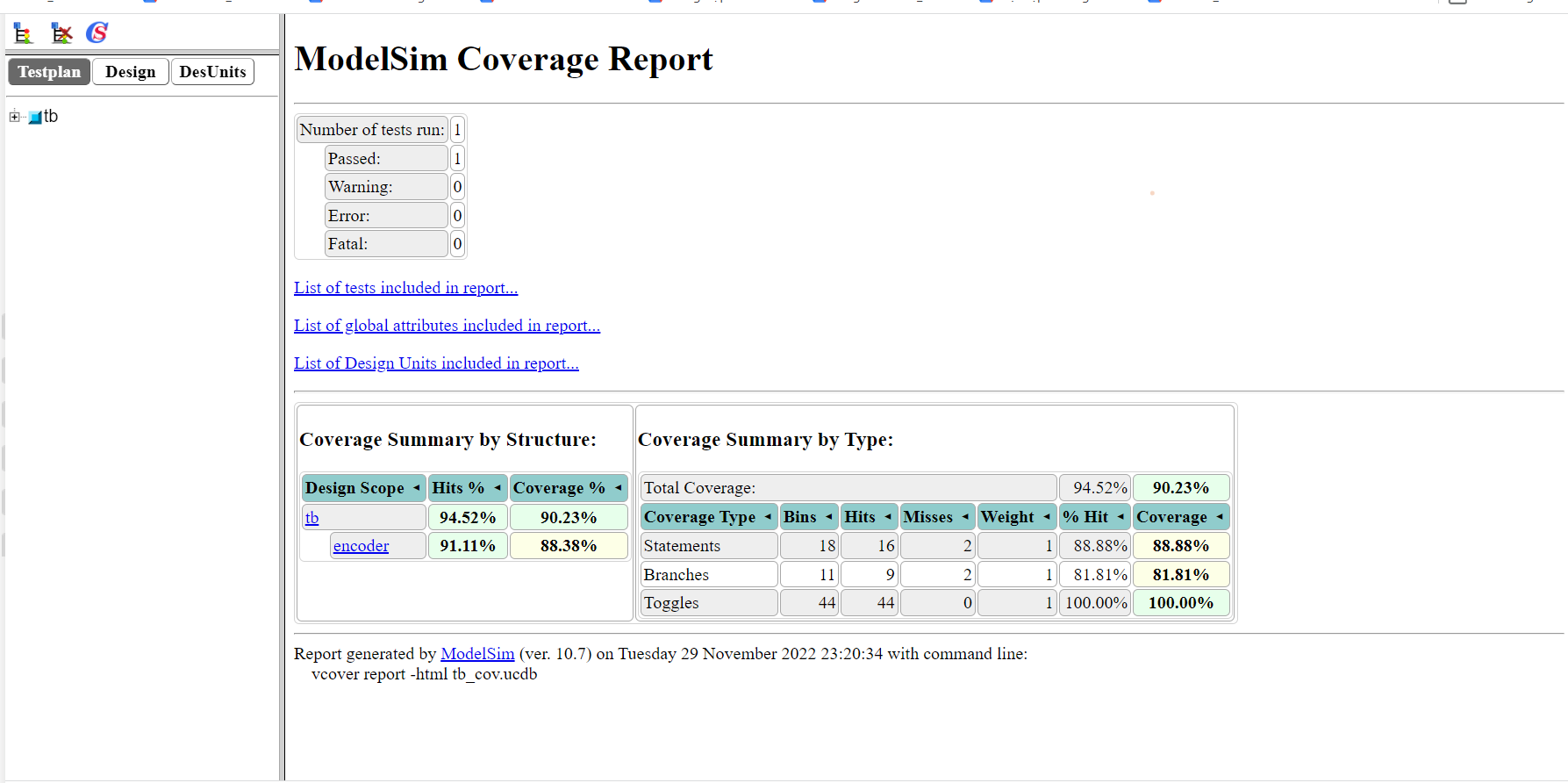
Wave

****

Script

****

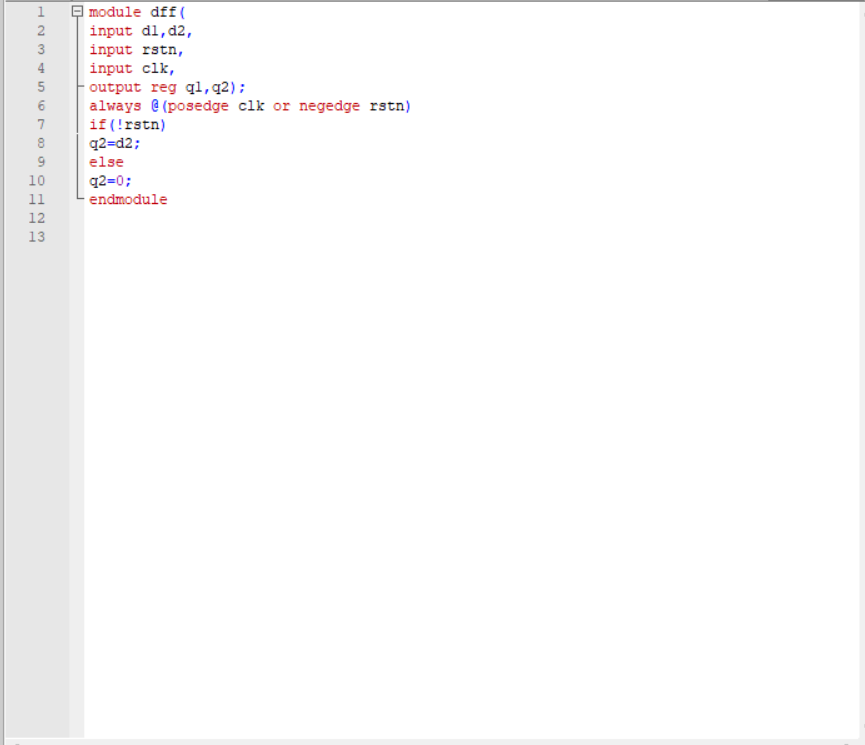
Report

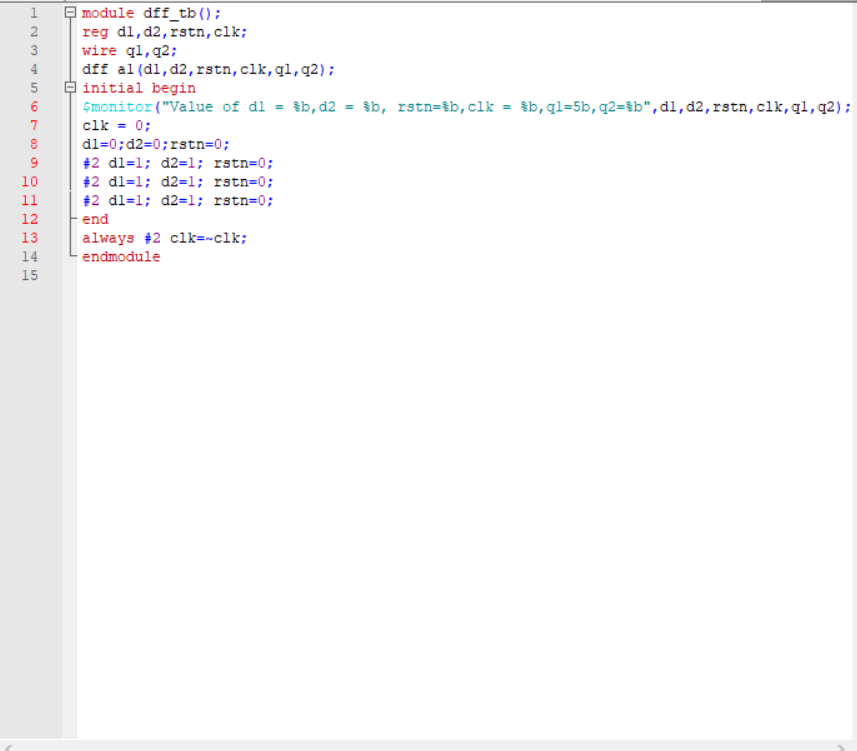
****

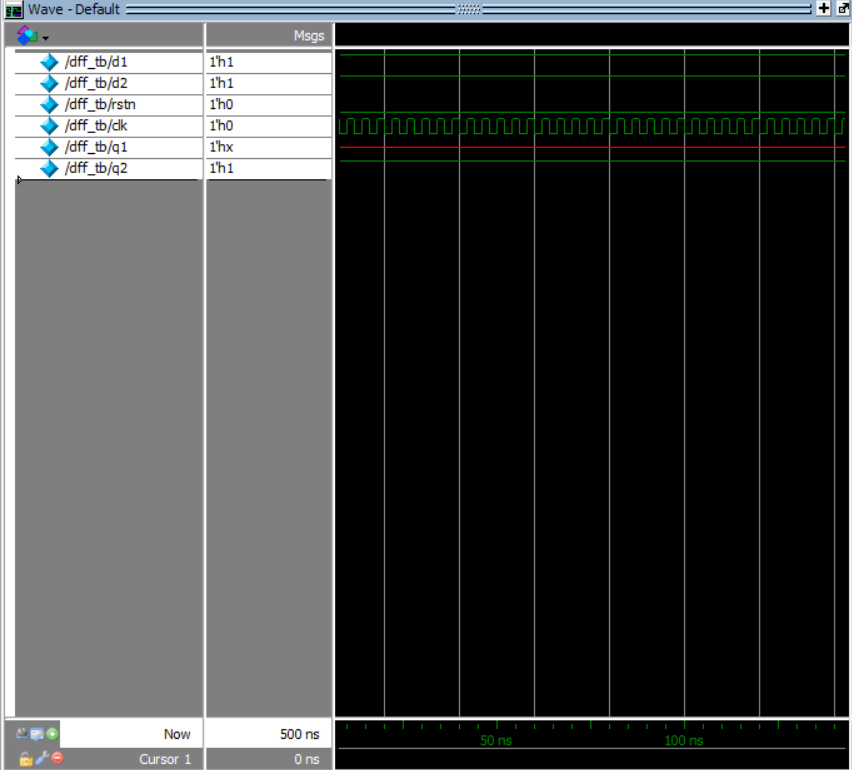
4

DFF

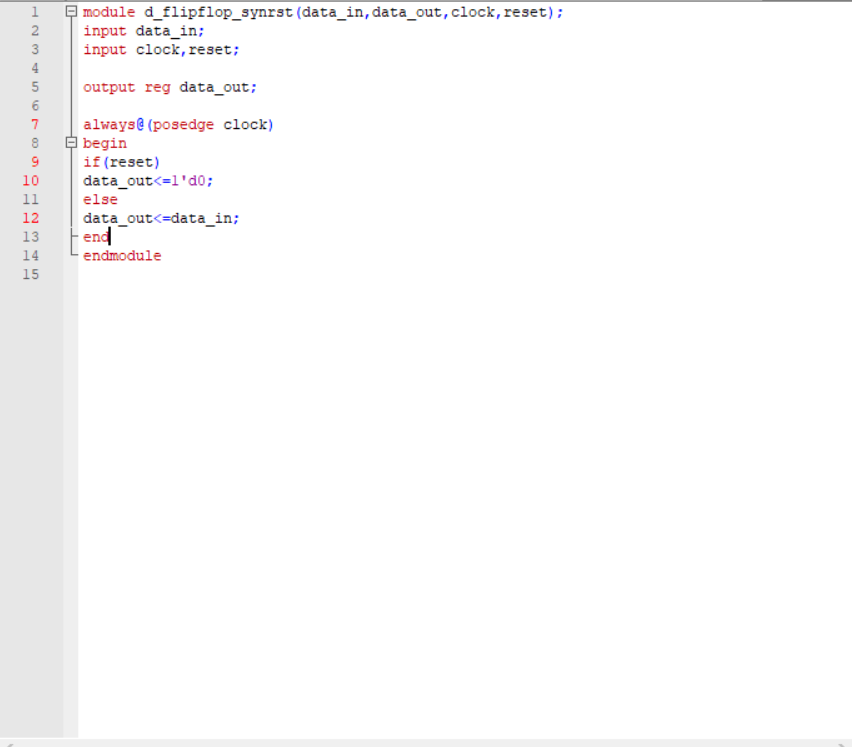
Asynchronous

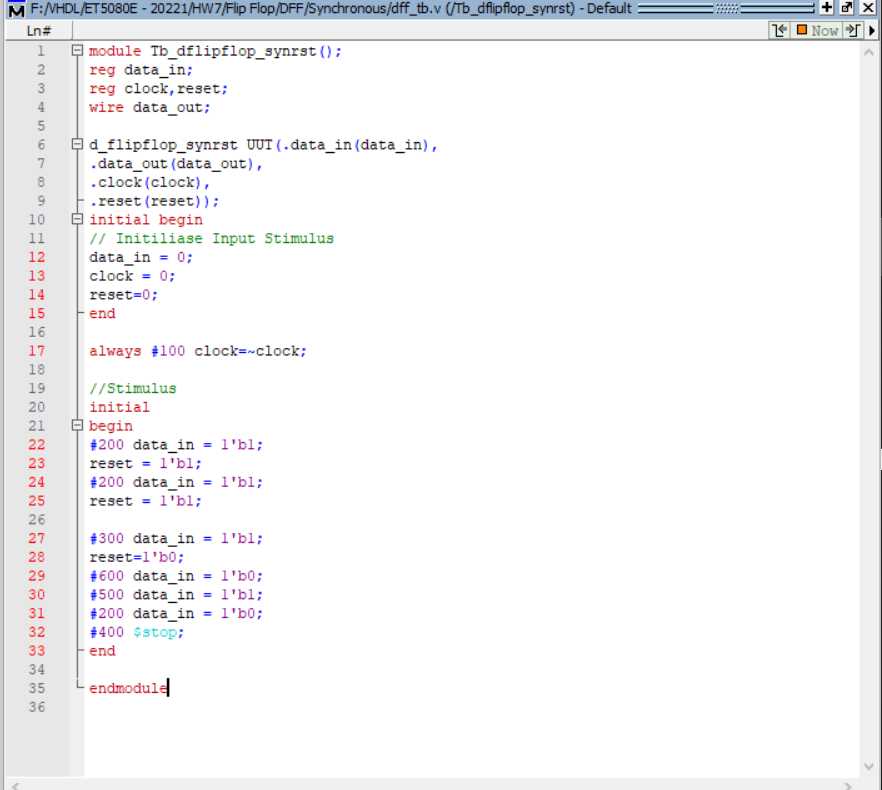


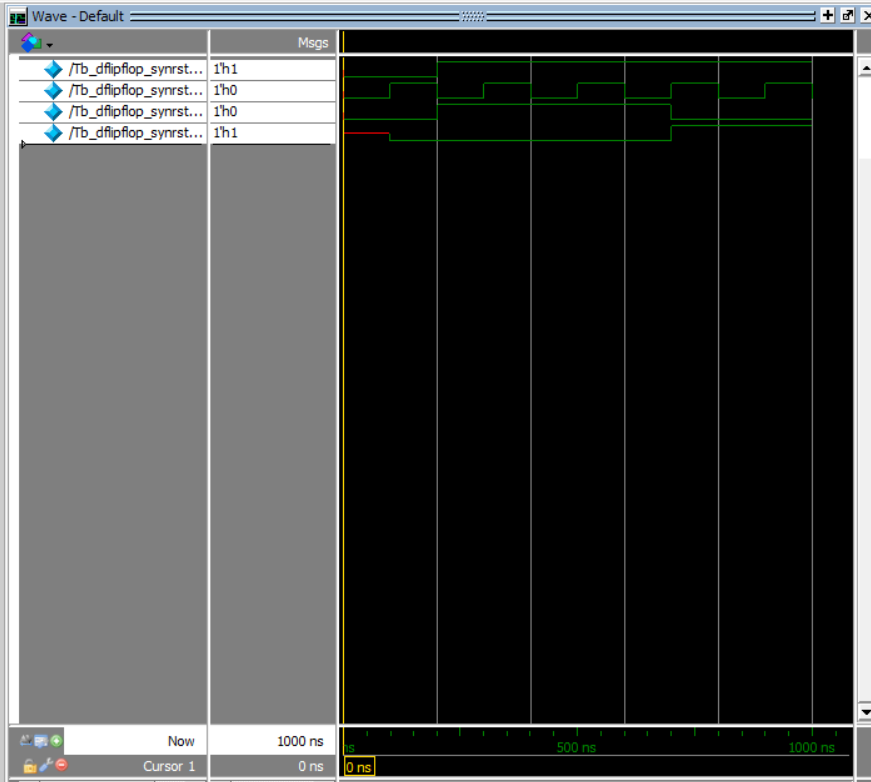


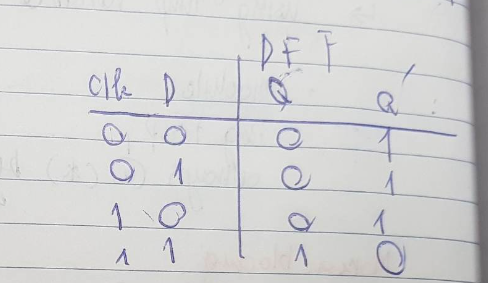


Synchronous



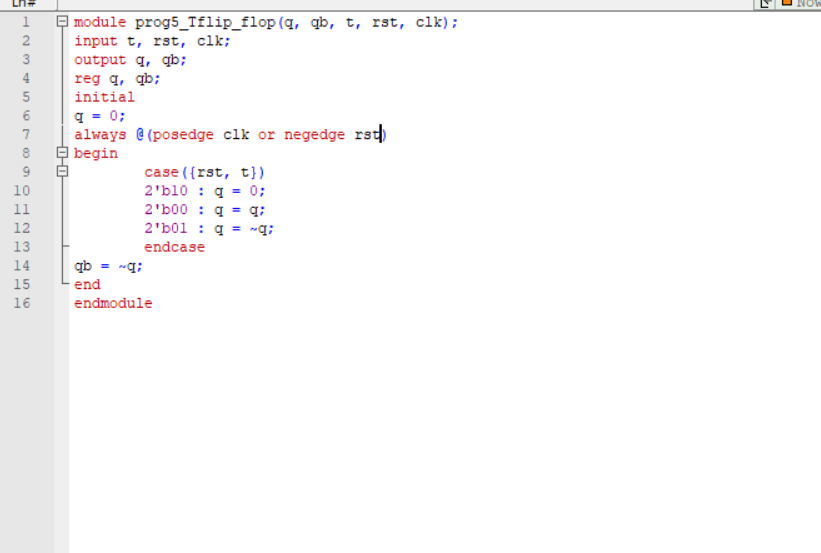


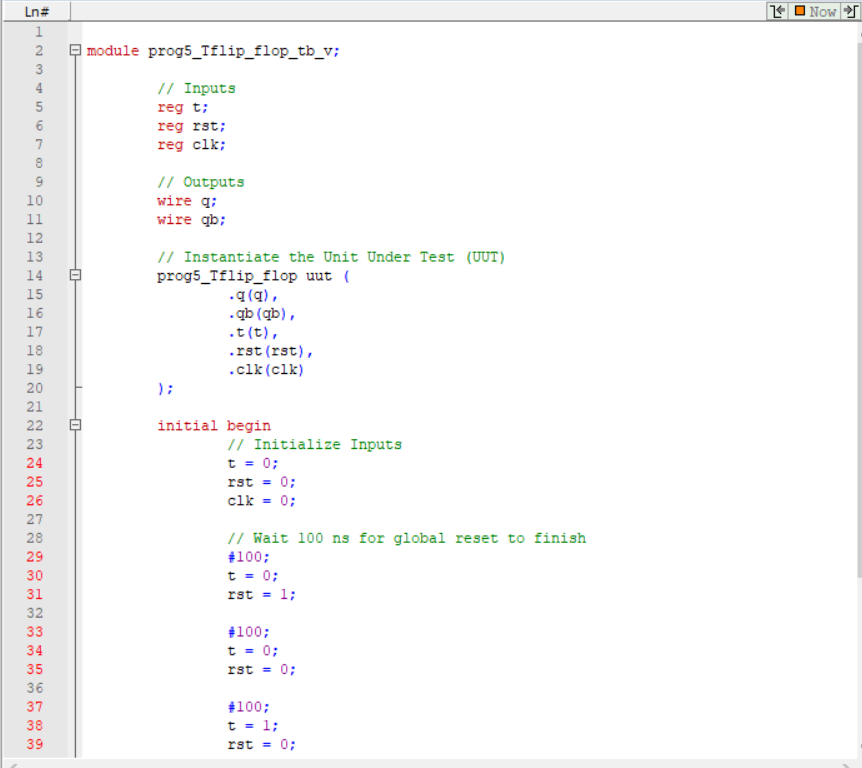


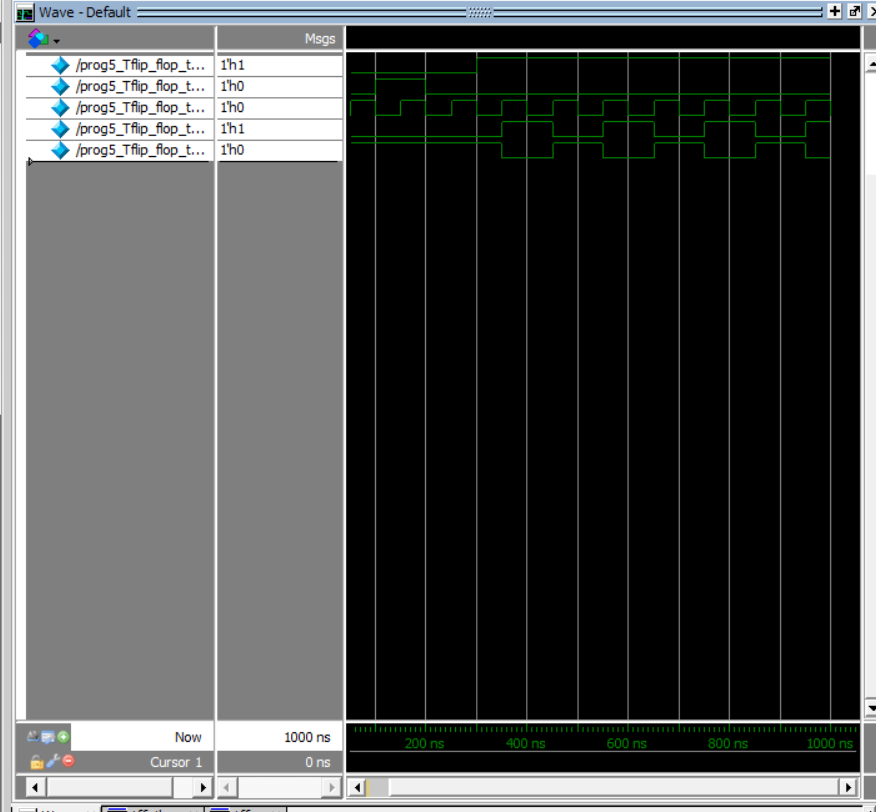


**TFF**

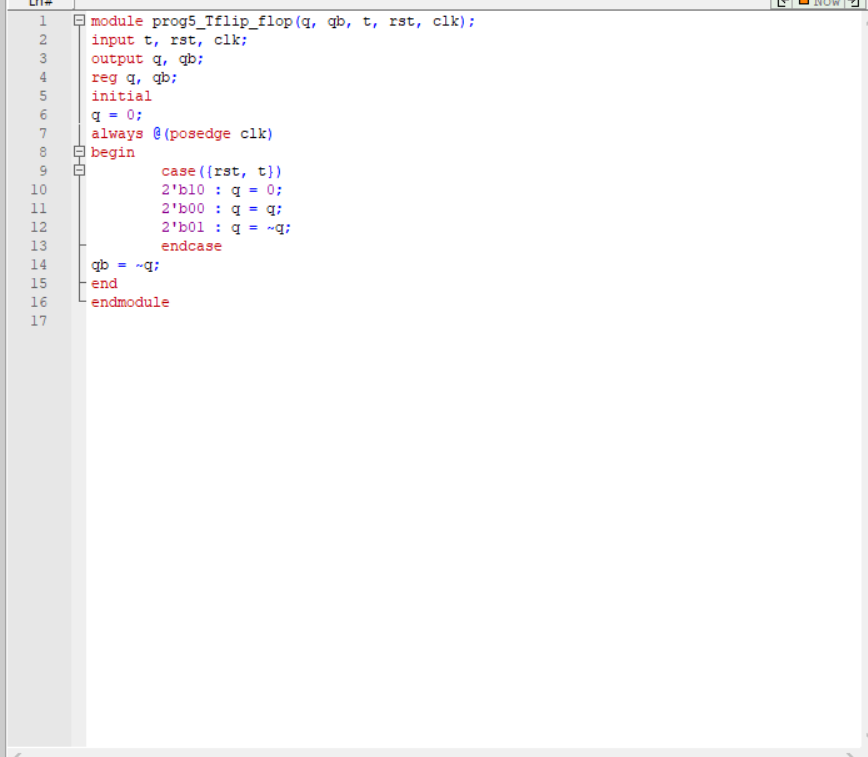
Asynchronous

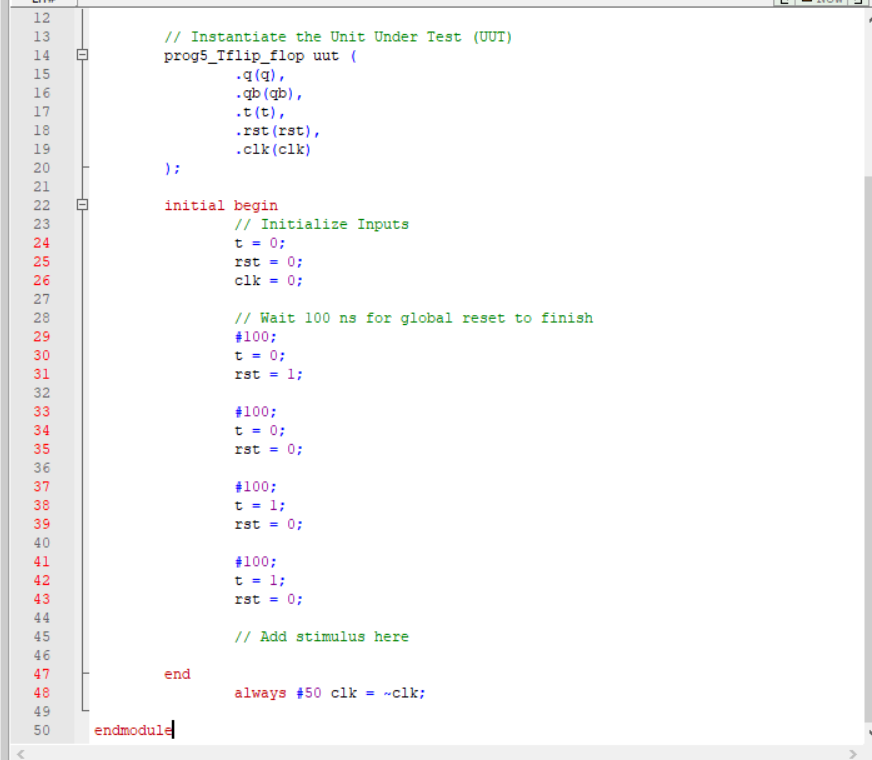


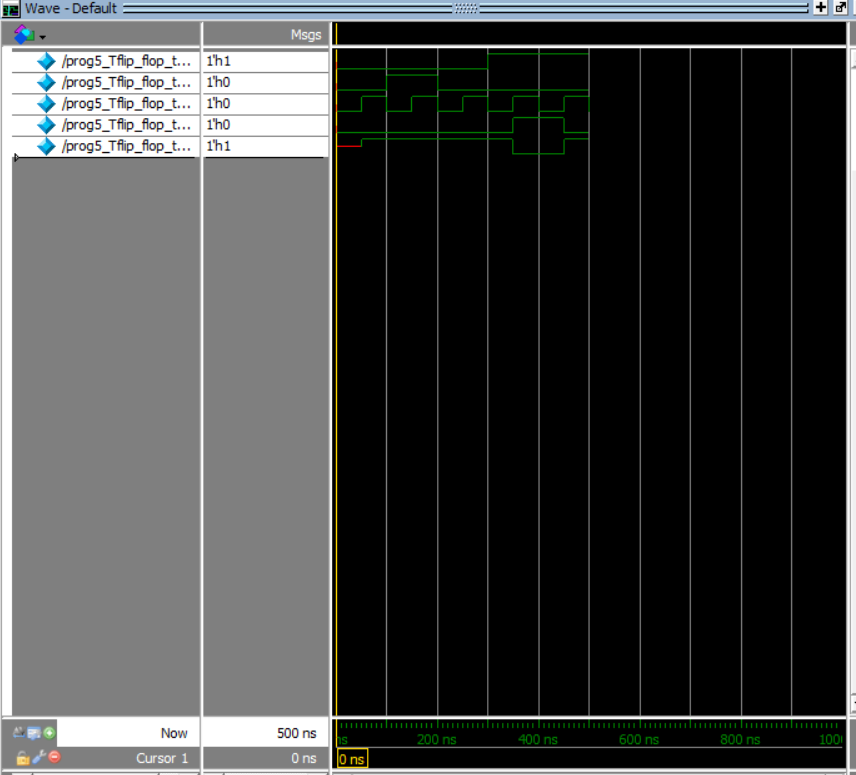


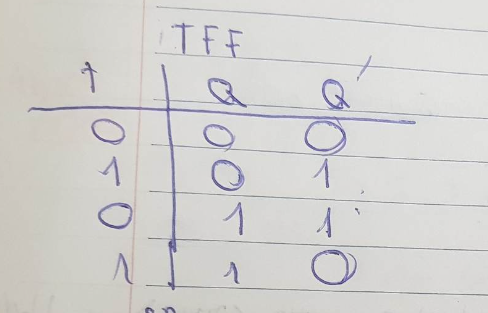


Synchronous



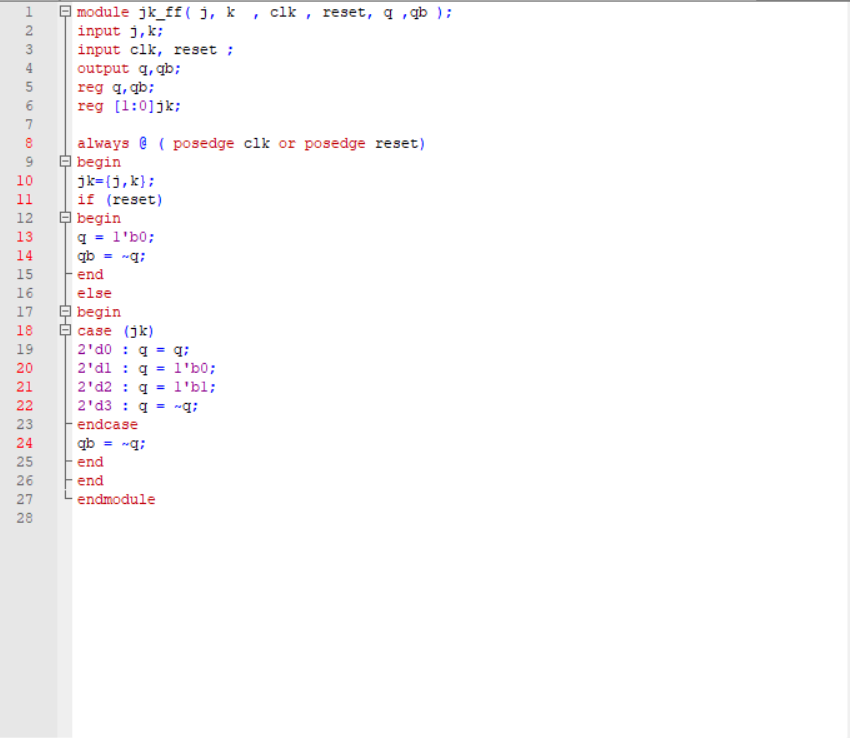


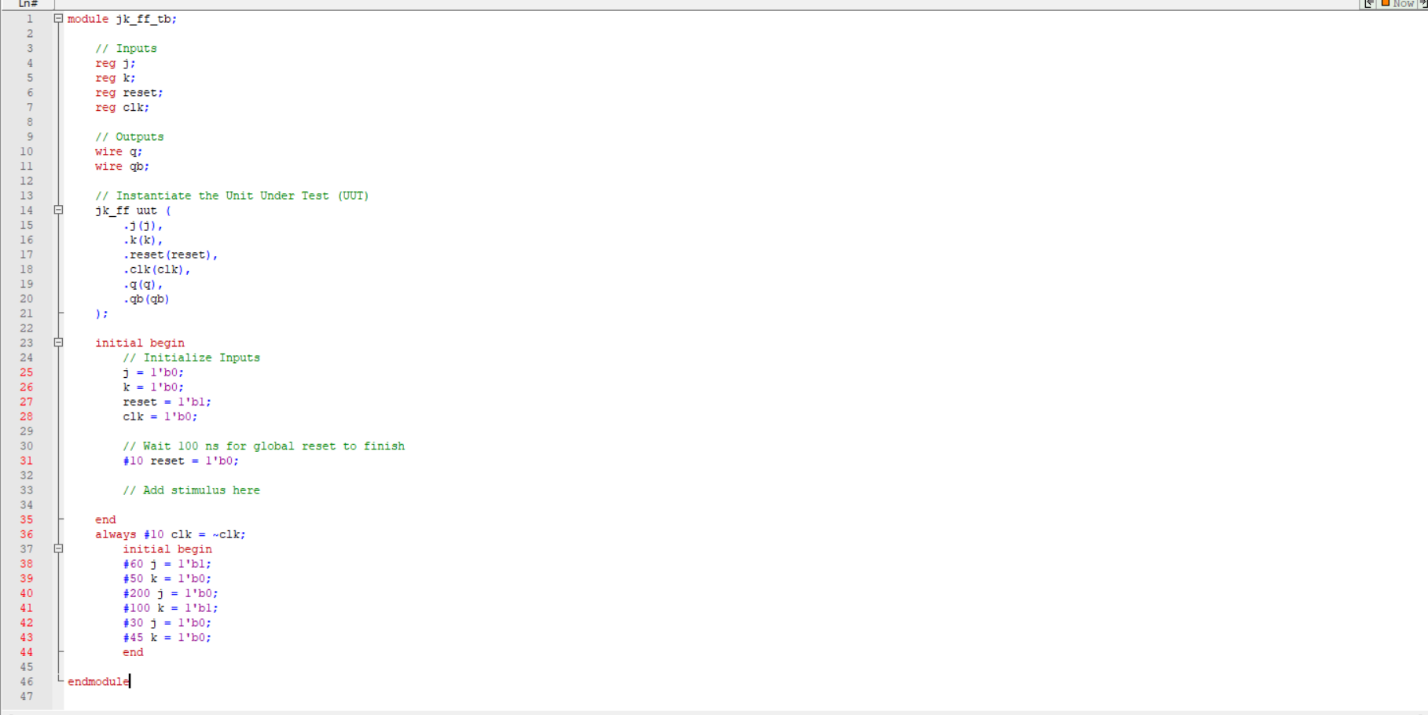


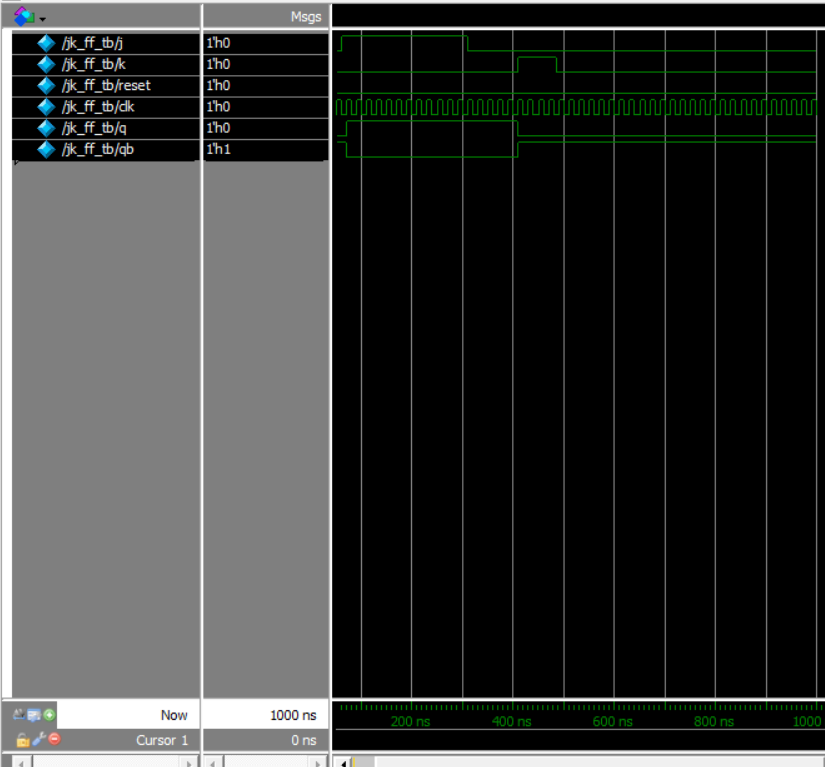


**JKFF**

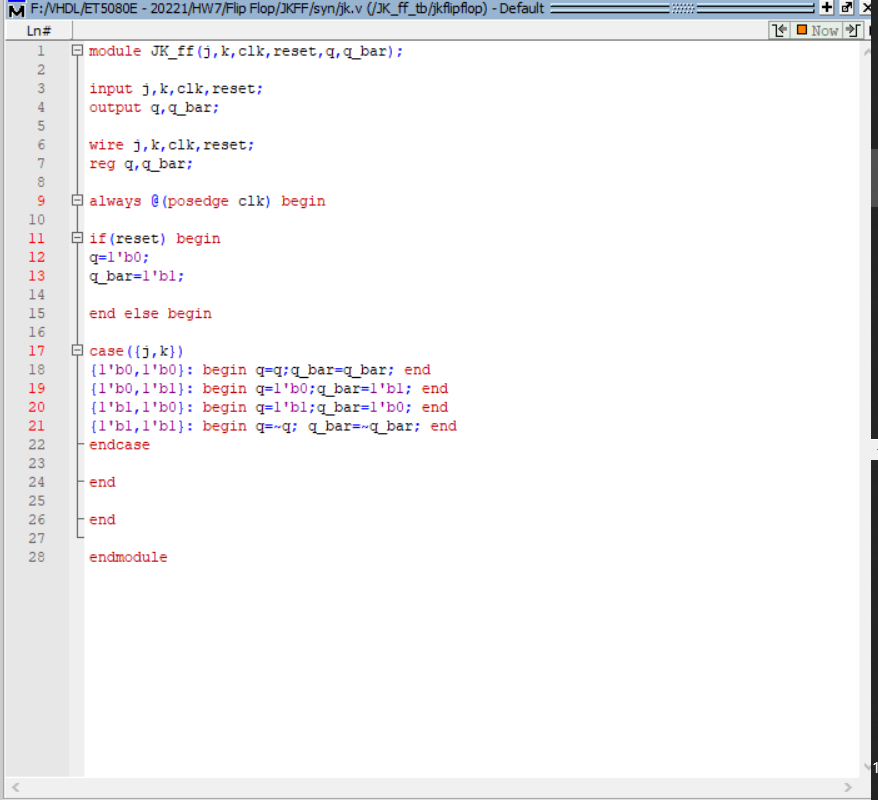
Asynchronous

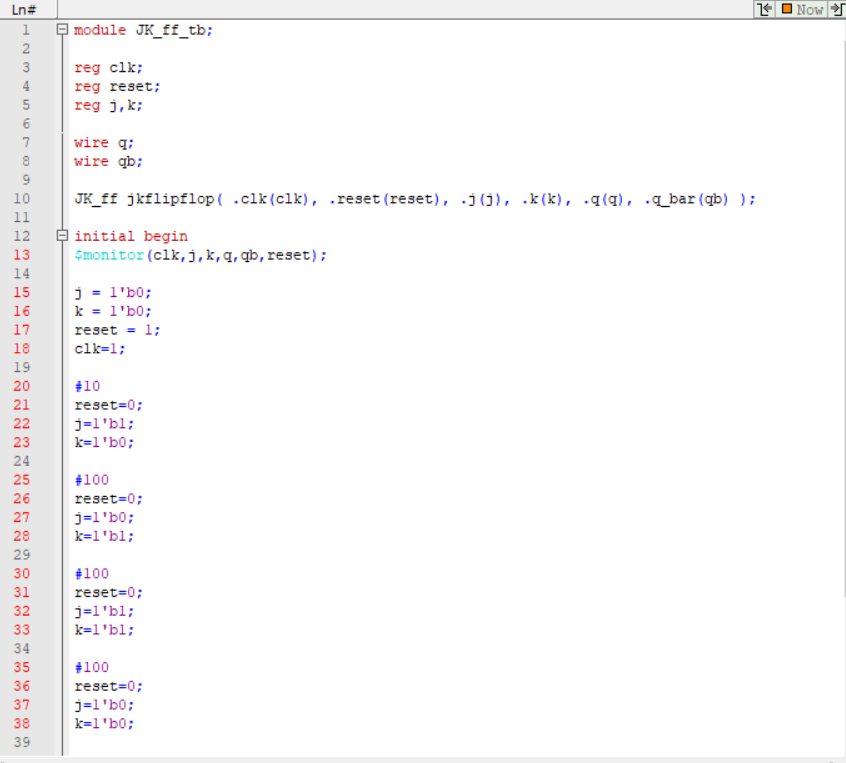


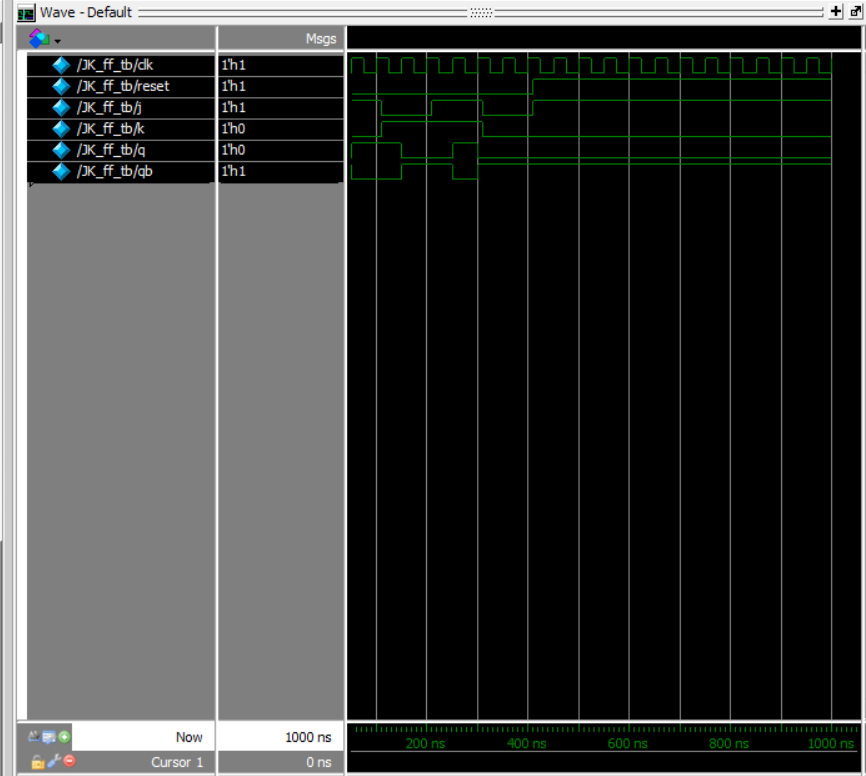


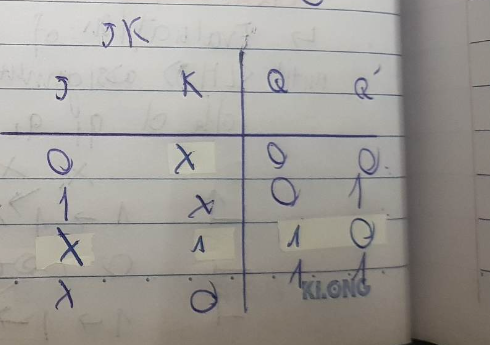


Synchronous



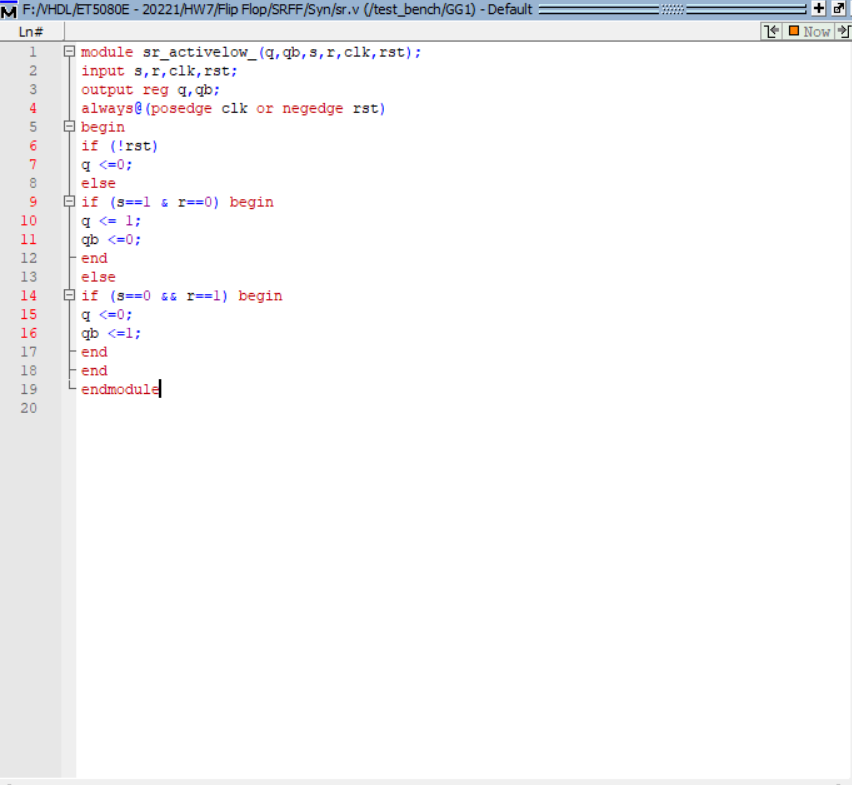


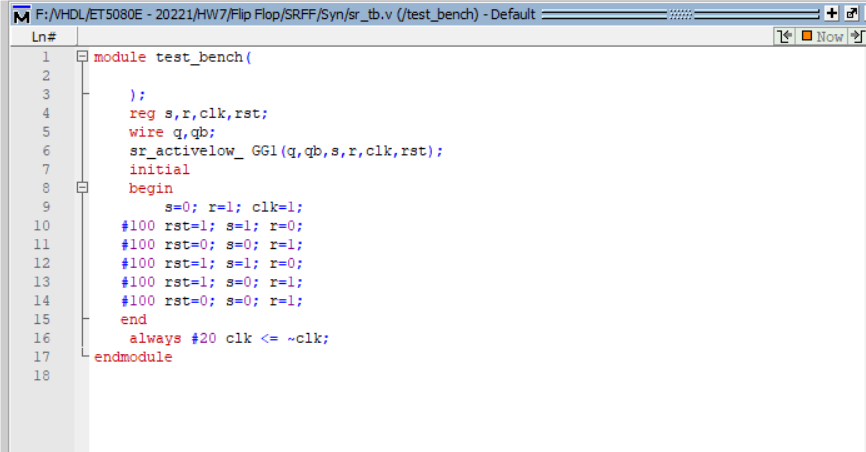


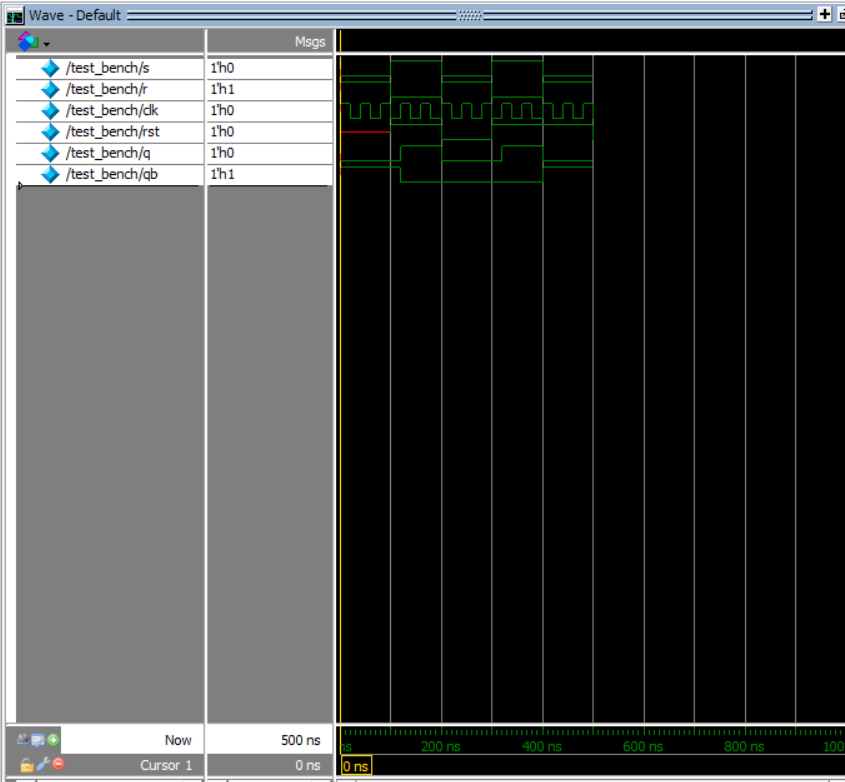


**SRFF**

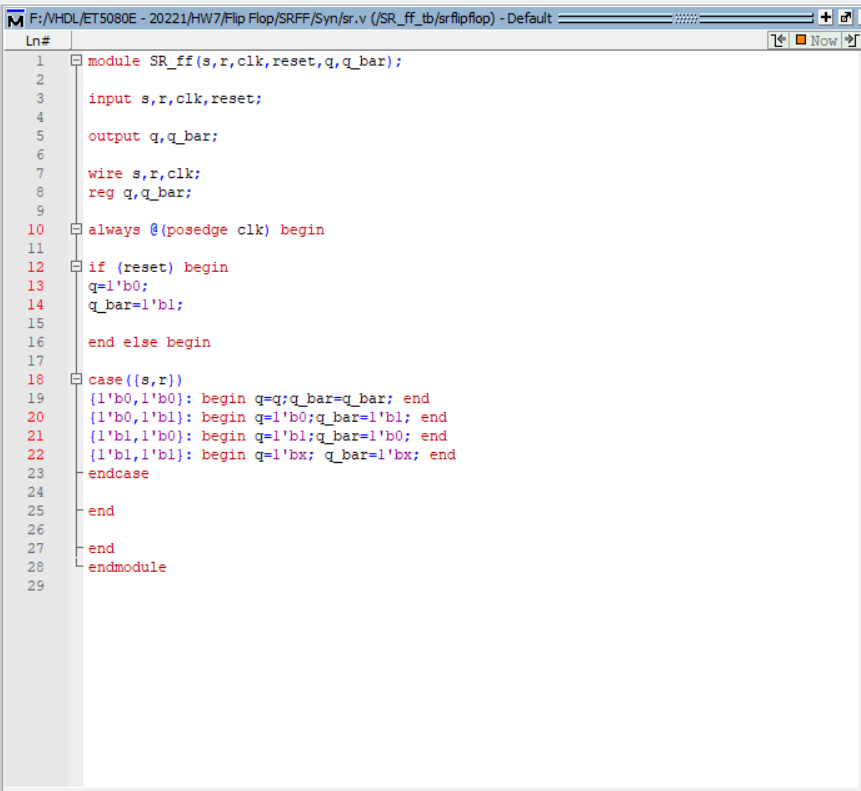
Asynchronous

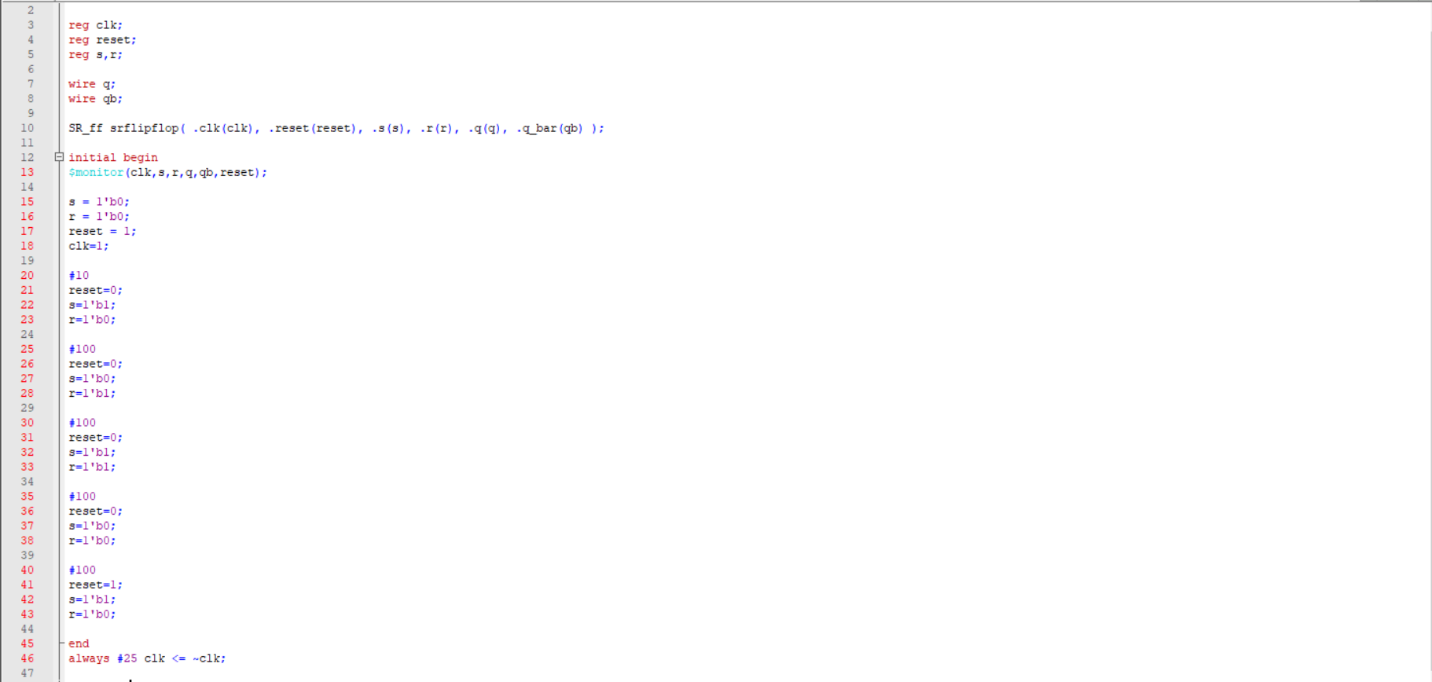


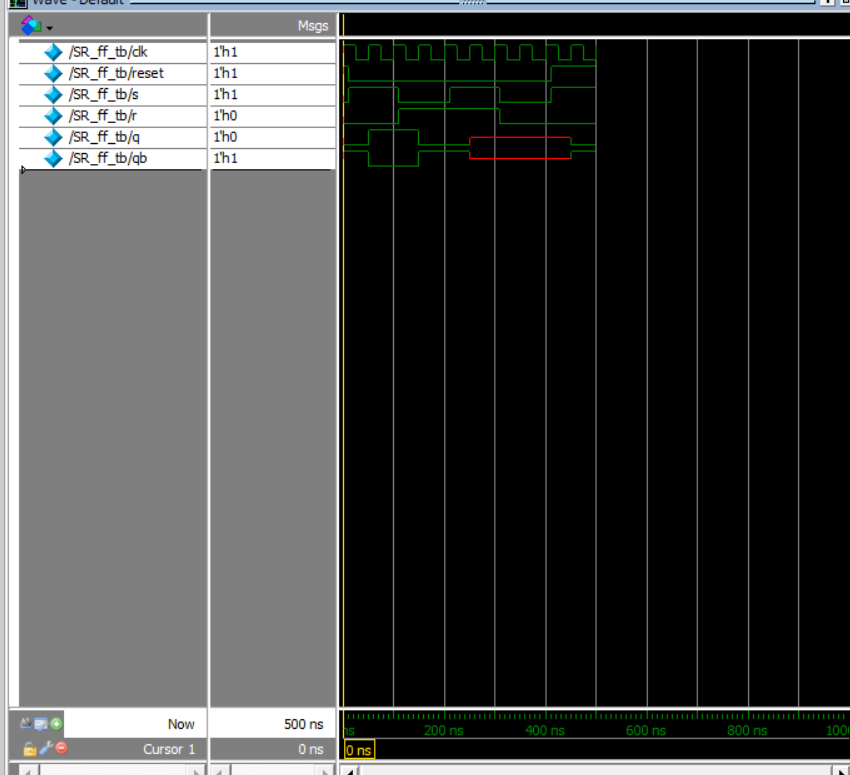


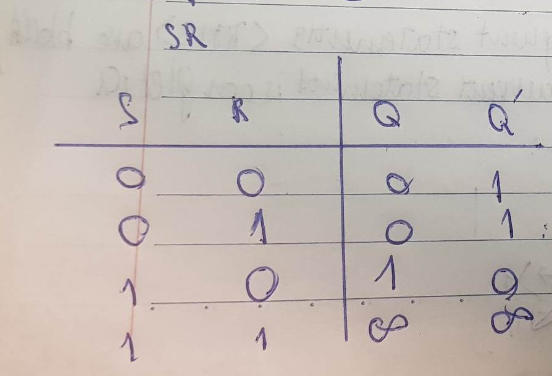


Synchronous









**Ex2**

**Always and initial blocks**

Similarity: All starts at t =0;

Differences:

**Initial**

Syntax for initial statement cam be indicated as below

<initial\_statement>

::= initial <statement>

\

The instruction executes only once in the whole process. It begins its execution at the start of the simulation at the time t =0. If there exists more than 1 initial block, then all the initial blocks are executed concurrently

**Always**

Always statement executes repeatedly, although the execution starts at time t = 0 and keep on executing all the simulation time. It works like an infinite loop. It is generally used to model a functionality that ‘s continuously repeated

Syntax:

always [timing\_control] procedural\_statement

To control the always statement we can use the trigger depending on what you are choosing to control