Laboratory Report #6

Name: **German E Felisarta III** **Group Number: 3**

Laboratory Exercise Title: **Behavioral Modeling of Sequential** **Date Completed: 11/17/2020**

Target Course Outcomes:

CO1: Create descriptions of digital hardware components such as in combinational and sequential circuits using synthesizable Verilog HDL constructs.
CO2: Verify the functionality of HDL-based components through design verification tools.

Exercise 6A:

In this exercise, a T Flipflop was to be made. Based on the Truth table of the T FlipFlop, the Design Entry was made. Since it is a positive-edge triggered latch, then the nReset test condition should be inverted with a not symbol (!). Apparently, after the Analysis and Synthesis, registers in quartus only has 1 Q output, so to output Q_bar, another register will be generated.

Fig 1a. Truth Table of T Flip Flop

	PREV		**NEXT**	
T	**Qprev**	**Q'prev**	**Qnext**	**Q'next**
0	0	1	0	1
0	1	0	1	0
1	0	1	1	0
1	1	0	0	1

Fig 1b. Design Entry of T Flip Flop

TFlipFlop.v Compilation Report - TFlipFlop

```
//GERMAN E FELISARTA III   16101002    CpE3101L Grp3

module TFlipFlop(T, nReset, Clk, Q, Q_bar);

   input wire T, nReset, Clk;
   output reg Q, Q_bar;

   always @ (posedge Clk) begin

      if(!nReset)
         Q <= 1'b0;

      else
            if (T == Q) begin
               Q <= 1'b0;
               Q_bar <= ~Q;
            end
            else begin
               Q <= 1'b1;
               Q_bar <= ~Q;
            end
   end

endmodule
```

Fig 1c. Flow Summary of T Flip Flop

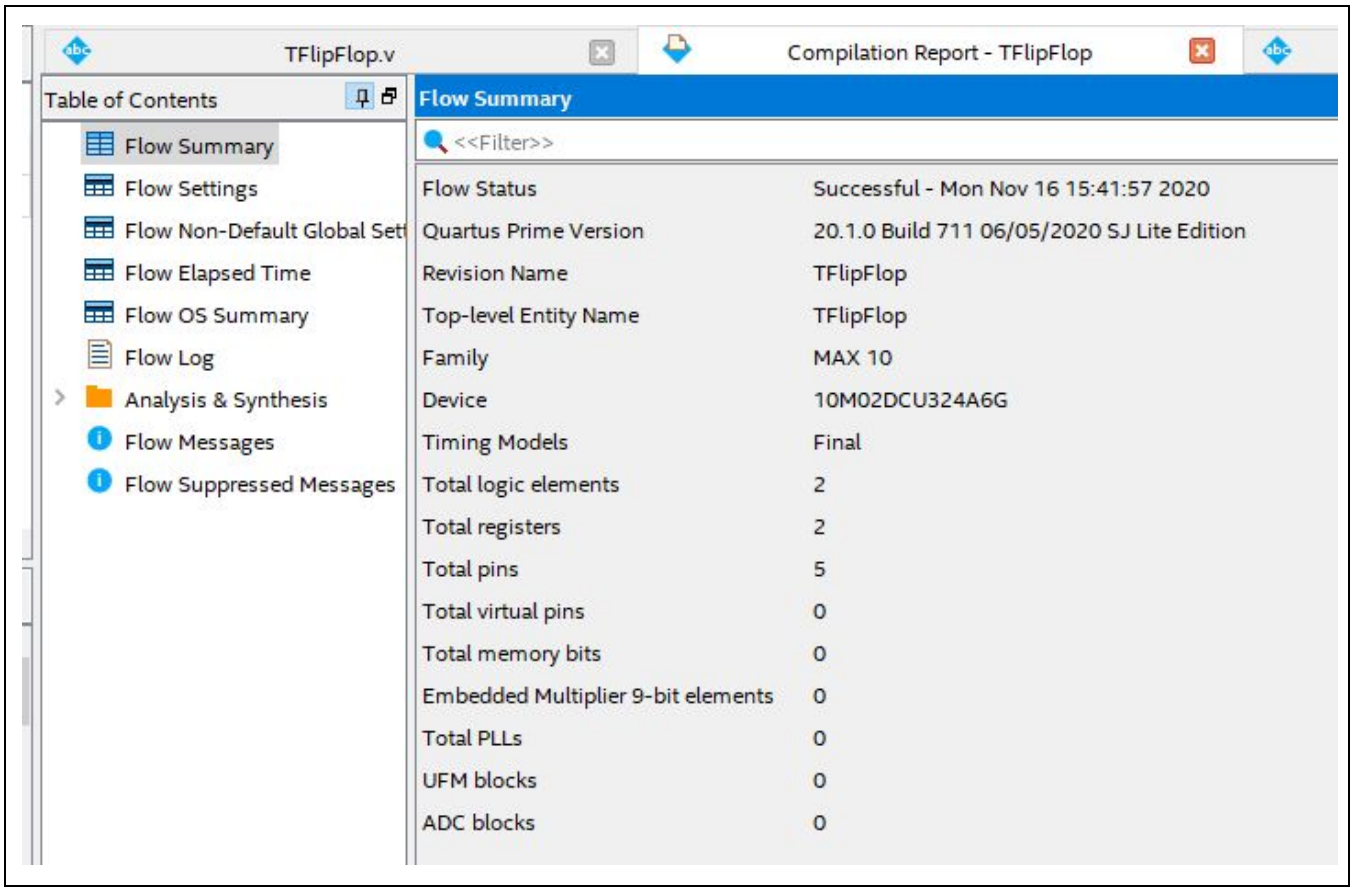


Fig 1d. RTL View of T Flip Flop

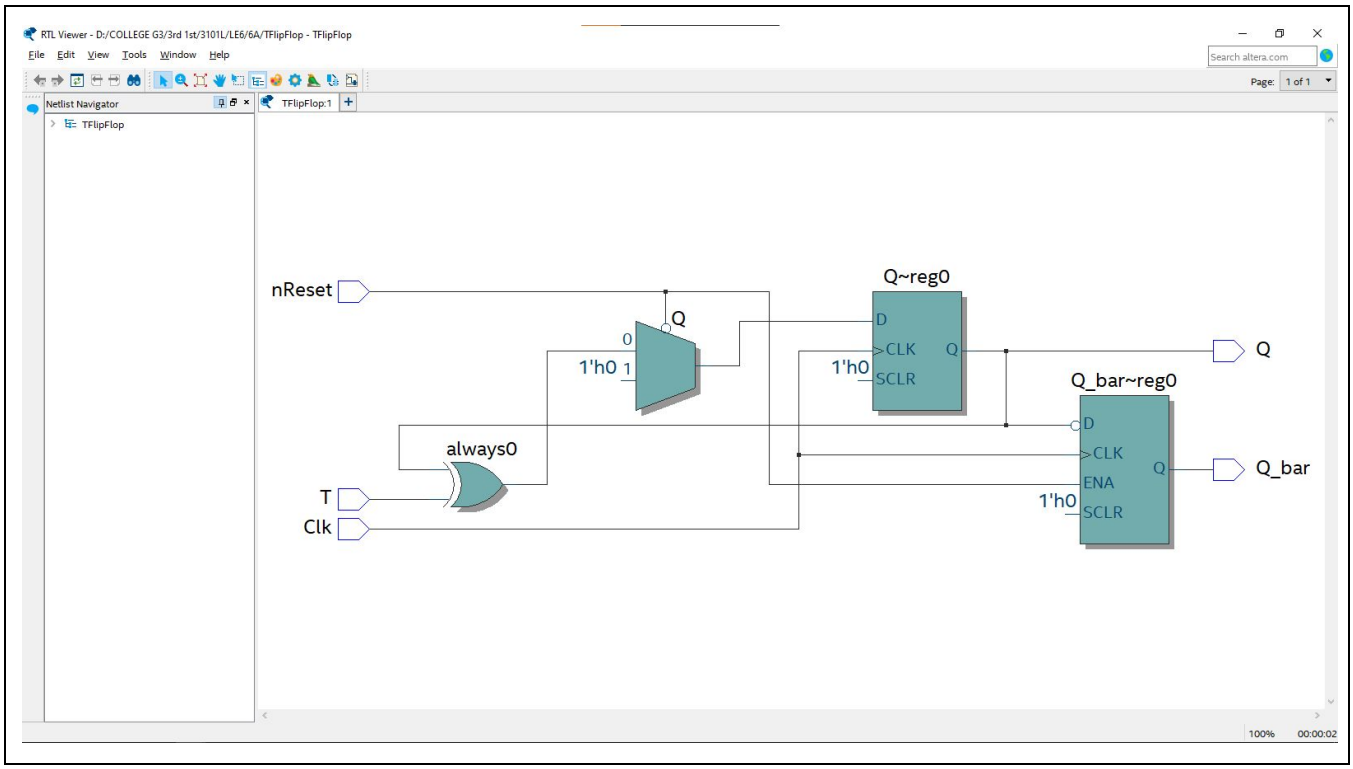
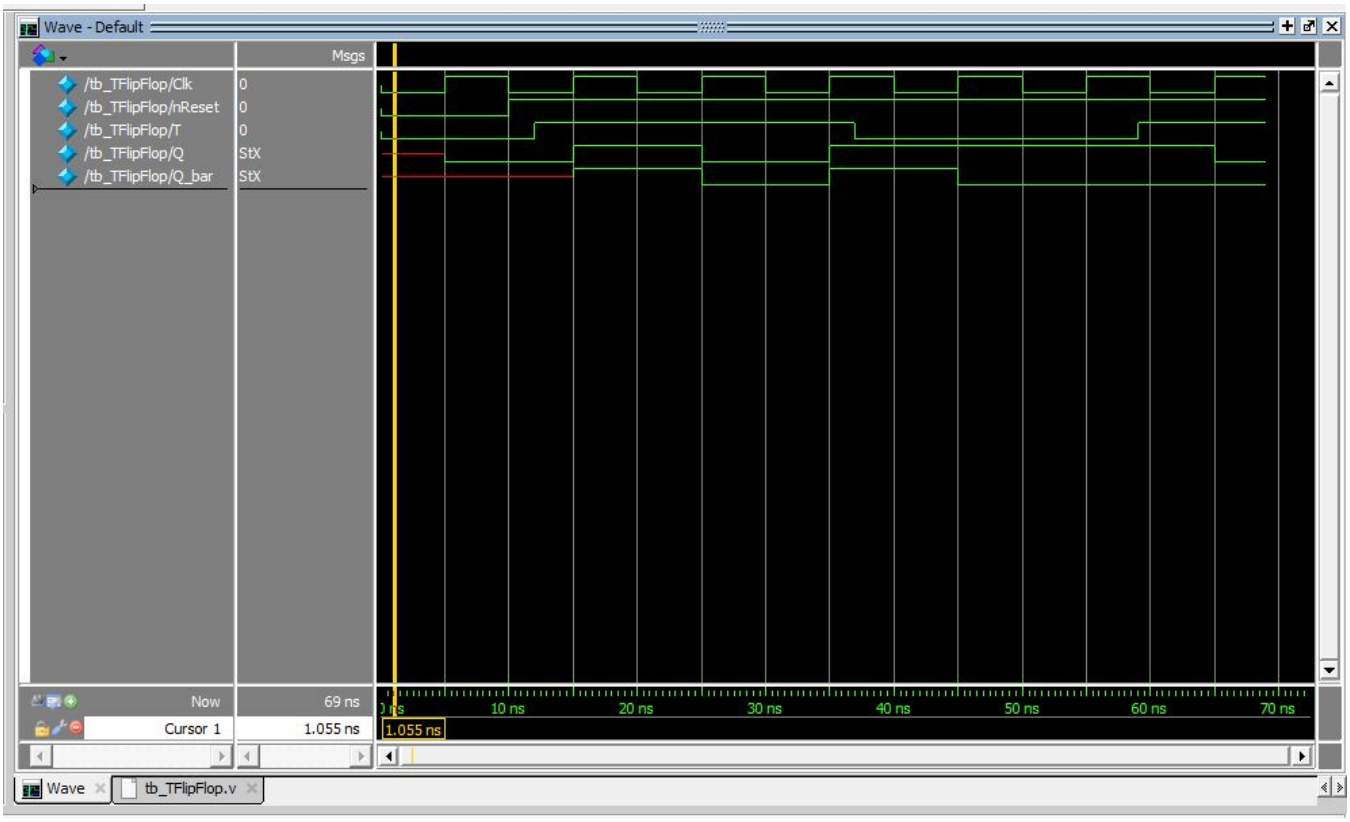


Fig 1e. Design Entry of T Flip Flop Testbench

TFlipFlop.v | Compilation Report - TFlipFlop | tb_TFlipFlop.v

```
//German E Felisarta III   16101002    CpE3101L Grp3

`timescale 1 ns / 1 ps
module tb_TFlipFlop();

   reg Clk, nReset, T;
   wire Q, Q_bar;

   TFlipFlop UUT (T, nReset, Clk, Q, Q_bar);

   initial
      Clk = 1'b0;

   always
      #5 Clk = ~Clk;

   initial begin
      nReset = 1'b0; #10
      nReset = 1'b1;
   end

   initial begin
      $display("Starting Simulation at %0d ns..", $time);
      T = 1'b0; #12
      T = 1'b1; #25
      T = 1'b0; #10
      T = 1'b0; #12
      T = 1'b1; #10
      $display("Finished simulation at %0d ns.", $time);
      $stop;
   end

endmodule
```

Fig 1f. RTL Simulation of T Flip Flop



Exercise 6B:

In this exercise, a JK FlipFlop was to be made. The same as Exercise 6A, the design entry for the JK flipflop is modeled after its truth table. Instead of positive-edge, the JK flip flop, as mentioned in the Lab Guide is negative-edged. It still has the same Register behavior where there is only one Q register output.

Fig 2a. Truth Table of JK Flip Flop

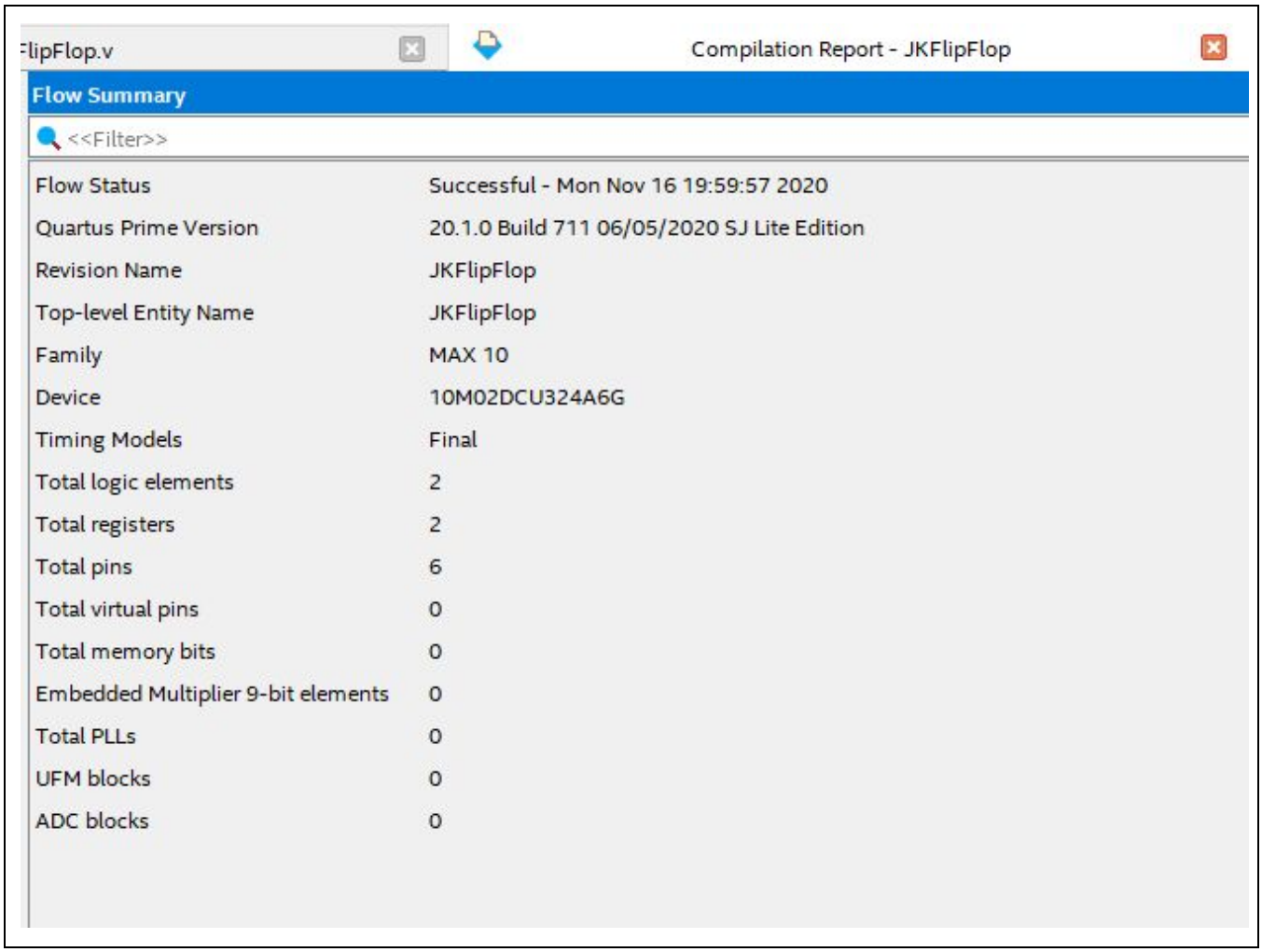
Clk	J	K	Q	Q’
down	0	0	Q	Q’
down	1	0	1	0
down	0	1	0	1
down	1	1	Q’	Q

Fig 2b. Design Entry of JK Flip Flop

JKFlipFlop.v Compilation Report - JKFlipFlop

```
//GERMAN E FELISARTA III    16101002   CpE 3101L Grp 3

module JKFlipFlop(J, K, Reset, Clk, Q, Q_bar);

   input wire J, K, Reset, Clk;
   output reg Q, Q_bar;


   always @ (negedge Clk) begin

      if(Reset) begin

         Q <= 1'b0;

      end

      else begin

         case ({J, K})

            2'b00 : Q <= Q;
            2'b01 : Q <= 1;
            2'b10 : Q <= 0;
            2'b11 : begin
                     if(Q == 1)
                        Q <= 0;
                     else
                        Q <= 1;
                    end
         endcase

      end

      Q_bar <= ~Q;

   end

endmodule
```

Fig 2c. Flow Summary of JK Flip Flop



FlipFlop.v

Compilation Report - JKFlipFlop

Flow Summary

<<Filter>>

Flow Status	Successful - Mon Nov 16 19:59:57 2020
Quartus Prime Version	20.1.0 Build 711 06/05/2020 SJ Lite Edition
Revision Name	JKFlipFlop
Top-level Entity Name	JKFlipFlop
Family	MAX 10
Device	10M02DCU324A6G
Timing Models	Final
Total logic elements	2
Total registers	2
Total pins	6
Total virtual pins	0
Total memory bits	0
Embedded Multiplier 9-bit elements	0
Total PLLs	0
UFM blocks	0
ADC blocks	0

Fig 2d. RTL View of JK Flip Flop

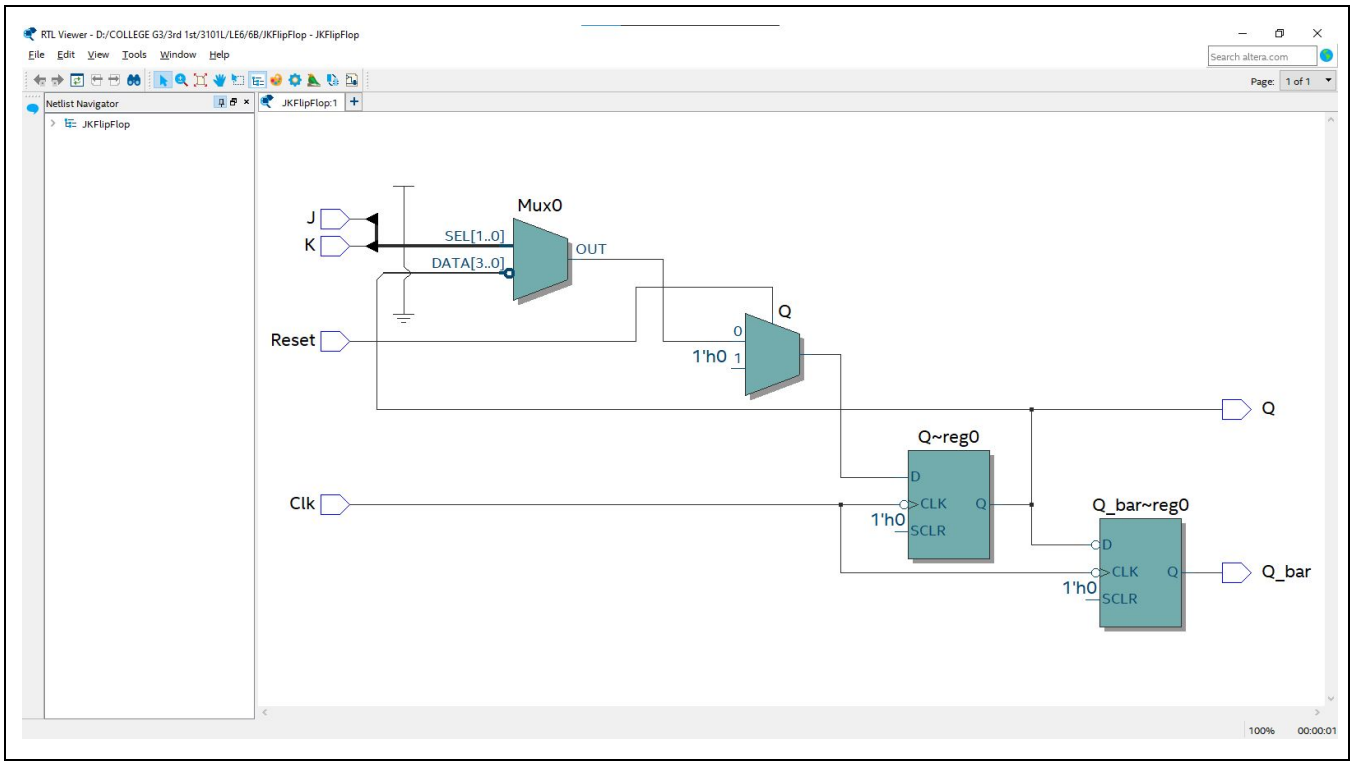
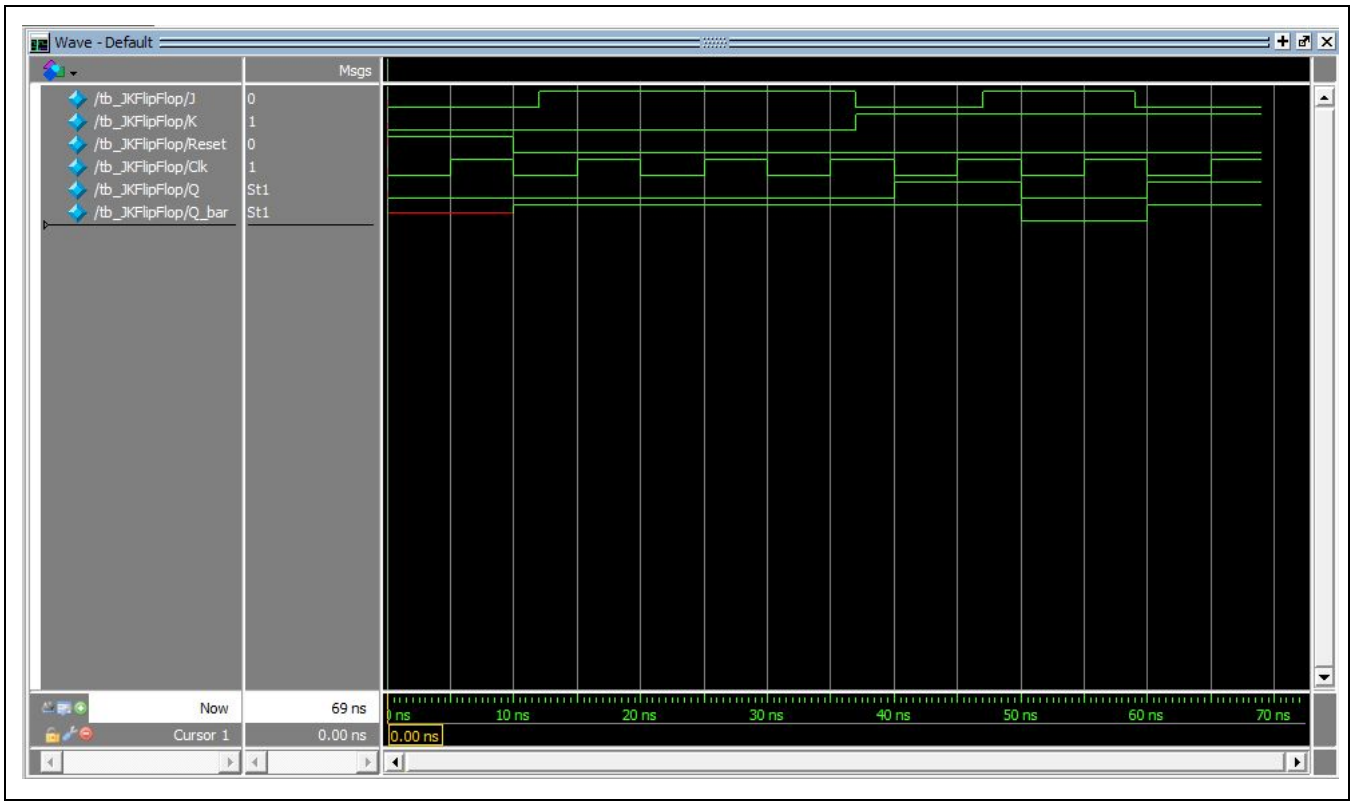


Fig 2e. Design Entry of JK Flip Flop Testbench

JKFlipFlop.v | Compilation Report - JKFlipFlop | tb_JKFlipFlop.v

```
//German E Felisarta III   16101002    CpE 3101LGrp3

`timescale 1 ns / 1 ps
module tb_JKFlipFlop();

   reg J, K, Reset, Clk;
   wire Q, Q_bar;

   JKFlipFlop UUT (J, K, Reset, Clk, Q, Q_bar);

   initial
      Clk = 1'b0;

   always
      #5 Clk = ~Clk;

   initial begin
      Reset = 1'b1; #10
      Reset = 1'b0;
   end

   initial begin
      $display("Starting simulation at %0d ns...", $time);

      {J,K} = 2'b00; #12
      {J,K} = 2'b10; #25
      {J,K} = 2'b01; #10
      {J,K} = 2'b11; #12
      {J,K} = 2'b01; #10


      $display("Finished simulation at %0d ns.", $time);
      $stop;
   end

endmodule
```

Fig 2f. RTL Simulation of JK Flip Flop



Exercise 6C:

In this exercise, ang nBit Register was to be made. From what was analyzed in the Lab Guide, whenever read mode is active, or RW = 0 and Enable = 1, then the output in Data_out will be high impedance (z). If it is in write mode, RW = 1, Enable = 1, then it would copy the input from Data_in to the output Data_out.

Fig 3a. Design Entry of nBitRegister

nBitRegisterSet.v | Compilation Report - nBitRegisterSet

```
//German E Felisarta III 16101002   CpE 3101L Grp 3

module nBitRegisterSet(Clk, nReset, Enable, RW, Address, Data_in, Data_out);

   localparam n = 8;

   input wire Clk, nReset, Enable, RW;
   input wire [2:0]Address;
   input wire [n-1:0]Data_in;

   output reg [n-1:0]Data_out;

   always @ (negedge Clk)

      if(!nReset) begin
         Data_out <= 8'b00000000;
      end
      else begin

         if(RW == 0) //read mode
            Data_out <= 8'bzzzzzzzz;
         else begin
            if (Enable == 1) //write mode
            Data_out <= Data_in;

         end
   end

endmodule
```

Fig 3b. Flow Summary of nBitRegister

Compilation Report - nBitRegisterSet

Flow Summary

<<Filter>>

Flow Status	Successful - Mon Nov 16 20:54:20 2020
Quartus Prime Version	20.1.0 Build 711 06/05/2020 SJ Lite Edition
Revision Name	nBitRegisterSet
Top-level Entity Name	nBitRegisterSet
Family	MAX 10
Device	10M02DCU324A6G
Timing Models	Final
Total logic elements	17
Total registers	16
Total pins	23
Total virtual pins	0
Total memory bits	0
Embedded Multiplier 9-bit elements	0
Total PLLs	0
UFM blocks	0
ADC blocks	0

Fig 3c. RTL View of nBitRegister

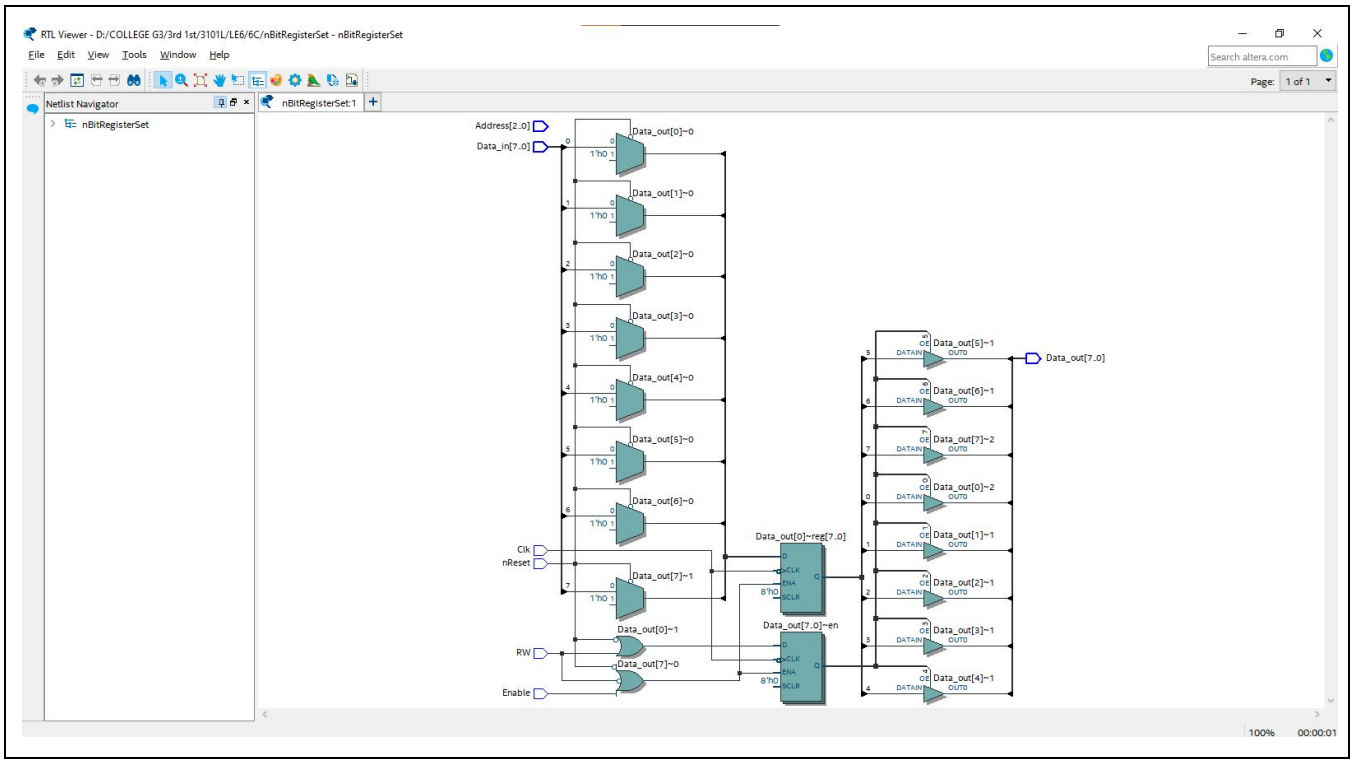


Fig 3d. Design Entry of nBitRegister Testbench

```
//German E Felisarta III 16101002          CpE 3101L Grp 3

`timescale 1 ns / 1 ps
module tb_nBitRegisterSet();

        localparam n = 8;

        reg Clk, nReset, Enable, RW;
        reg [2:0]Address;
        reg [n-1:0]Data_in;

        wire [n-1:0]Data_out;

        nBitRegisterSet UUT (Clk, nReset, Enable, RW, Address, Data_in, Data_out);

        initial
                Clk = 1'b0;

        always
                #5 Clk = ~Clk;

        initial begin
                nReset = 1'b0; #10
                nReset = 1'b1;
        end
```

```
	initial begin
		$display("Starting simulation at %0d ns... ", $time);

		{RW, Enable} = 2'b10; #10
		{RW, Enable, Address} = 5'b11000;
		Data_in = 8'b01010101; #10
		{RW, Enable, Address} = 5'b11001;
		Data_in = 8'b00001111; #10
		{RW, Enable, Address} = 5'b11010;
		Data_in = 8'b00110011; #10
		{RW, Enable, Address} = 5'b11011;
		Data_in = 8'b00011101; #10
		{RW, Enable, Address} = 5'b11100;
		Data_in = 8'b10101010; #10
		{RW, Enable, Address} = 5'b11101;
		Data_in = 8'b11001100; #10
		{RW, Enable, Address} = 5'b11110;
		Data_in = 8'b11100010; #10
		{RW, Enable, Address} = 5'b11111;
		Data_in = 8'b11111111; #10

		$display("Finished simulation at %0d ns.", $time);
		$stop;
	end

endmodule
```

Fig 3e. RTL Simulation of nBitRegister