CME341 Assignment 2

1. The purpose of this question is to familiarize the students with the Quartus II and Modelsim software tools. This is done by constructing a set/clear latch using two NOR gates. Note that the **set** and **clear** inputs are active high and that the **Q** output comes from the NOR gate with the clear input. Follow the instructions below:

   (a) Draw a schematic diagram of a set_clear_latch based on two NOR gates.

   (b) Create an appropriate directory on your hard drive for this Quartus project.

   (c) Open Quartus II as per instructions in the class notes.

   (d) Name the project "set_clear_latch". Give the top-level entity the same name, which is "set_clear_latch".

   (e) Select the appropriate FPGA. Select Altera Cylcone IV E FPGA with part number "**EP4CE115F29C7**".

   (f) From the Quartus II window select "File" from the top menu, then "New". Under "Device Design Files", "SystemVerilog HDL File". Click OK. In the editor window, write an explicit structural description (i.e. use primitives for the NOR gates) of your set/clear latch. (NOTE: You must name your module the same name as you entered in the top level entity box i.e. "set_clear_latch"). Save the file as "set_clear_latch.sv". The file name is always the same as the top level module name (the name and appropriate folder should come up automatically).

   (g) Compile the project by selecting "Processing" from the top menu. Click on "Start Compilation". If there are any errors, alter your Verilog code appropriately. Otherwise, click OK in the information box telling you the compilation was successful.

   (h) While the compiler is running it will print information (green print), warnings (blue print) and errors (red print). Most of the of the time a warning means something is wrong. In this case there

will be a few warnings which will be explained at a later time. One of the warnings indicates that the circuit that was biult has feedback and may be a latch. There is always the potential for circuits with feedback to oscillate so when this is detected by the compiler it prints a warning.

(i) Once the project is successfully compiled, open Modelsim by following the instructions in the class notes.

(j) Construct a testbench for "set_clear_latch.sv". Generate waveforms for the **set** and **clear** inputs. Make both high to start. Then, at 1 microsecond, make clear low. Make **set** low at 2 microseconds. Put a 0.5 microsecond pulse on **clear** starting at 3 microseconds. Put a 0.5 microsecond pulse on **set** starting at 4 microseconds.

(k) Simulate the circuit by following the procedure described in the class notes.

(l) Print out or save screenshots of the timing diagrams that demonstrate the circuit works as intended. These should be included with your assignment submission.

(m) By inspecting the waveforms using Modelsim, find the propagation time from **set** to **Q**, **set** to **n_Q**, **clear** to **Q** and **clear** to **n_Q**.

2. The purpose of this question is to learn how to design and compile a digital circuit by instantiating one digital circuit inside another. Follow the instructions below.

(a) Draw the schematic diagram of the transparent latch described by the Verilog HDL below:

```
module transparent_latch (D, gate, Q, Q_bar);
input D,gate;
output Q, Q_bar;
wire set_bar, clear_bar, D_bar;

nand nand_1 (set_bar, gate, D);
nand nand_2 (clear_bar, gate, D_bar);
not not_1 (D_bar, D);
```

```
// instantiate set/clear latch
set_clear_latch sc_latch_1 (.n_set(set_bar),
    .n_clear(clear_bar),
    .Q(Q),
    .n_Q(Q_bar));
endmodule
```

(b) The set/clear latch instantiated in the description above has active low inputs while a set/clear latch based on cross coupled NOR gates has active high inputs. Modify the schematic to work with a set/clear latch designed with NOR gates.

(c) Create a new project called "transparent_latch" (follow the same steps as in Question 1) and write a Verilog description of a transparent latch based on the modified schematic diagram, i.e. the one that has a set/clear latch with active high inputs. Name the file "transparent_latch.sv". In your description instantiate the set/clear latch that you created earlier in question 1f. The set/clear latch should be described in a separate Verilog file (copy "set_clear_latch.sv" from the project folder for Question 1 into the project folder for Question 2).

(d) Before compiling the project, select "Project" from the top menu, then click on "Add/Remove Files in Project". Make sure both Verilog files have been added to the project. If not, add them.

(e) Compile your transparent latch.

(f) Create a testbench. Make the simulation 10 microseconds long. Put three signals in the wave window: **D**, **gate**, and **Q**. Put **D** on top, **gate** in the middle and **Q** on the bottom. Make **D** a clock signal with 500 ns period and make the clock high to start. A clock signal is generated in a simulator with the following commands:

```
initial D=1'b1;
always
#250 D = ~D;
```

Start **gate** high, make it go low at 850 ns and then make it go high again at 4.1 microsecond and then low again at 5.1 microseconds and remain low for the rest of the simulation. Simulate and hand in a printout or screenshot of the three signals.

3. There are several ways to make a D flip flop. One way is to use two transparent latches in series. The clock for the D flip flop is connected to the gate of the second latch. It is also connected to the gate of the first latch through an inverter. Create a new project and write a Verilog description for a D flip flop that has only three pins: **clk**, **D**, and **Q**. Your description should consist of three statements: two instantiation statements that build two transparent latches, and an assign statement that builds an inverter. Note: nothing is connected to **Q_bar** of either transparent latch. The connection list of the instantiation should only have three connections. The ".Q_bar( )" entry should be omitted.

4. Debug the D flip flop and then demonstrate it works by including a printout or screenshot containing the results of a simulation. Place the **D** input on top, the clock in the middle, and Q on the bottom. Make the simulation 2 microseconds long. Make **clk** a clock signal with period 200 ns and start the clock high. Make **D** start high and toggle at times 300 ns, 420 ns, 650 ns and 1720 ns.