

# Project 2: Analysis and Simulation of a Simple Sequential Machine

Instructor Prof. Joanne E. Degroat

This project assignment assumes you are familiar (at a basic level) with the elements of the *Xilinx ISE* software. If you are not familiar, review the steps in Project 1 regarding opening a new project, creating schematics in the schematic editor, setting up inputs for functional simulation and doing the simulation. You may use departmental laboratory facilities or your own computer to complete this project.

## Create a Project and invoke the Schematic Editor

Invoke the Project Navigator, and create a new project by selecting **File** → **New Project** with the name **561Proj2**. In the New Project Wizard window, specify *Schematic* as the *Top-Level Module Type* and *VHDL* as *Preferred Language*. After clicking *Next*, specify *XC9500 CPLDs* as the *Device Family* and *XST (VHDL/Verilog)* as your *Synthesis Tool* from the pull-down menus.

We will not create new sources for this project at this point. Continue clicking *Next*, *Next*, *Next* and then *Finish* to end the new project setup process.

To create new source, click on **Project** → **New Source**. In the pop-up window, select the entry *Schematic* and give the name *top* in the *File Name* field. Click *Next* and in the next window click *Finish*. Once you do that a blank schematic *Xilinx - ISE - [top.sch]* will open in the Schematic Editor window.

## Add Inputs and Outputs


Start by adding six I/O Markers to the design. Click on **Tools** → **Create IOMarkers**. A window will pop up, requesting you to specify the input and output port names. Specify inputs as **INPUT**, **PRE**, and **CLK** and outputs as **Q0**, **Q1**, and **RCO**. Separate the IOMarker names by commas. This will add 3 inputs and 3 outputs with wires attached to them in your schematic.

Now add input buffers corresponding to each input. In the left side of your schematic editor window, under the symbol sub-window called *Categories*, click on the entry *IO*. If you don't see the sub-windows, click on **Add** → **Symbol**. Now click on the entry *ibuf* under the *Symbols* sub-window. This will select an input buffer. To place the



toolbar on the left edge of the window. It allows you to select one item or a group of them.

## Add some wires

Now you will 'wire up' the circuit as shown in Figure 2. Click the tool button  to enter the wiring mode. Begin a wire by clicking on one of the ends of a device in the sheet, and terminate the wire by double-clicking on its destination. Wire the logic using this technique. To route a single signal to multiple destinations, complete one connection, and then click on this wire to start the other connections. You can delete a wire by selecting it and pressing the delete key. You can reroute a wire (sometimes with difficulty) by clicking and dragging it around. The final design layout is in Figure 3.

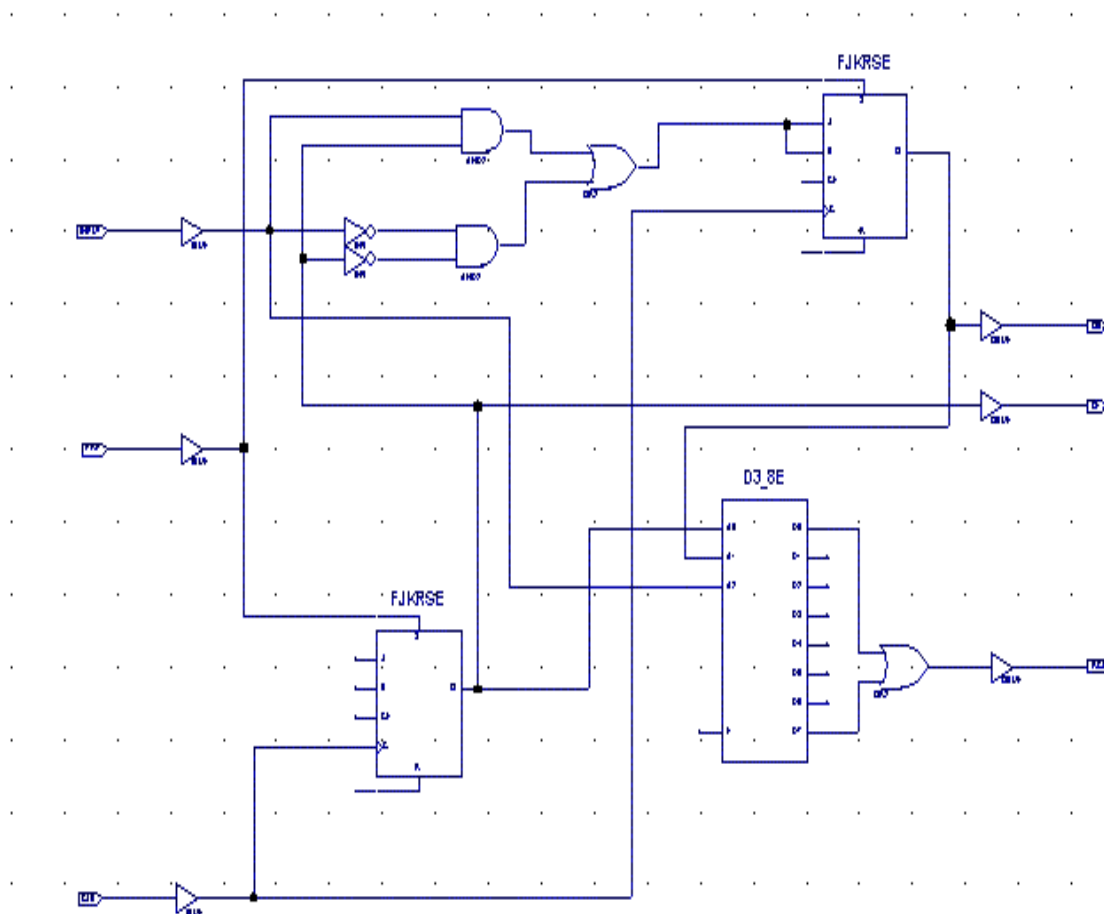


Figure 2: Schematic Editor window with partially-wired circuit

## Note

In the schematic shown above, only at the positions where there is a blue dot, is a connection between wires. Otherwise, wires that cross each other don't have any connection. For example, the input **CLK** is connected only to the **C** inputs of the flip-flops *ffkrse*. Even though this connection passes through the wires from the **A0** and **A2** inputs of *d3\_8e*, there is no connection between them. The input named **INPUT** is actually an input in your circuit; don't change that.

## Add constant signals

The remaining dangling inputs need to be connected to ground (logic 0) or high (logic 1) signals. This is done by clicking on the *General* item in the *Categories* in your Schematic Editor window. Now click on the entry named *gnd* in the Symbols sub-window. This selects ground (logic 0). You can click to place the symbol wherever you like. To place high (logic 1) select the entry named *vcc* from the device list and click on the required location in the schematic to place it. Be careful to attach all dangling inputs. The **R** (reset) inputs of flip-flops should be tied to *gnd*. The **E** (enable) input of the decoder, the **CE** (clock enable) lines on the flip-flops, and the **J K** inputs on the lower left flip-flops should be tied high. The final schematic is given in Figure 3.

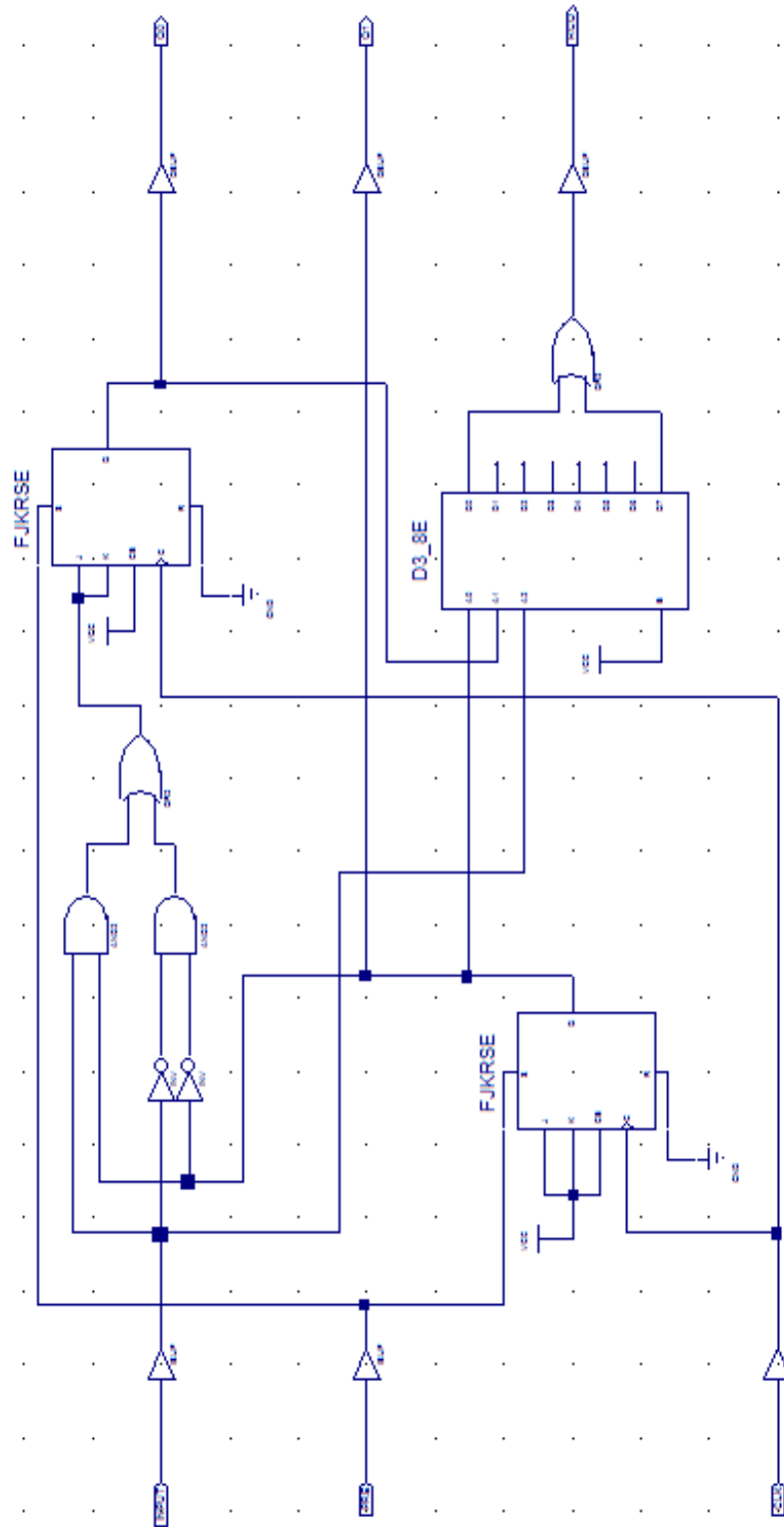


Figure 3: Final design

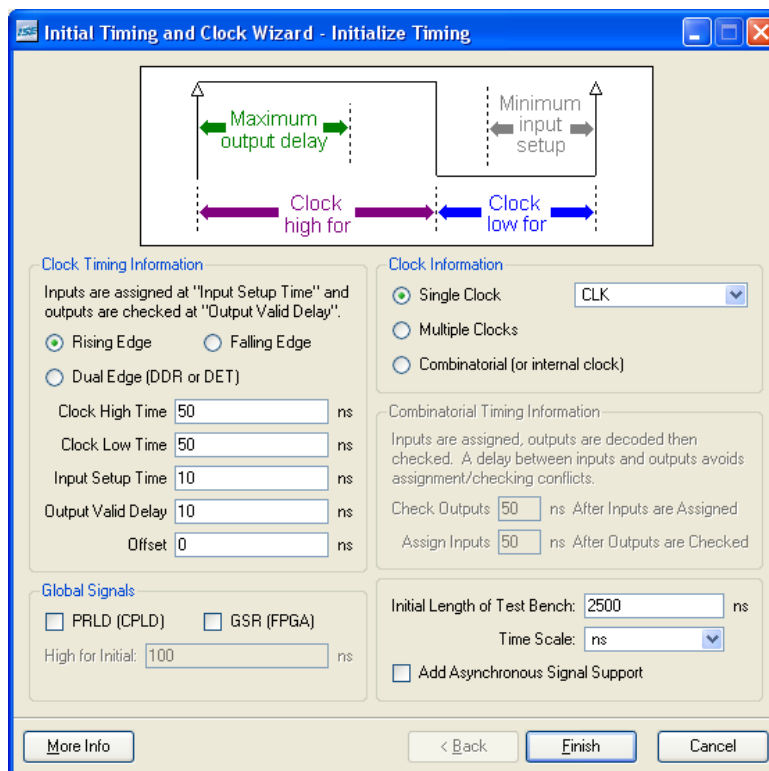
## Checking Errors

Click on **Tools** → **Check Schematic**. If there are errors, they will be displayed in the Transcripts window at the bottom of Project Navigator. If there are no errors, then the Console tab will display *No error or warning is detected*. Note: successful passage of the integrity test does not mean the circuit is correct!

## Functional Simulation and Analysis


Now you can invoke a functional simulation of the design. This simulation assumes ideal timing and does not reflect the realities of finite propagation delays and setup/hold times. In your Project Navigator window, click on your schematic file *top* (*top.sch*) to make it active. Now select **Project** → **New Source**. In the window that opens up select the option *Test Bench Waveform*. Specify a name for the waveform in the *File Name* field and click on *Next*, *Next* and then in the following window click *Finish*.

This will open the HDL Bencher window. In the Initialize Timing window, select the option *Single Clock*. Set *Clock High Time* and *Clock Low Time* to 50 ns, *Input Setup Time* and *Output Valid Delay* to 10 ns, *Initial Length of Test Bench* to 2500 ns, and deselect *GSR (FPGA)*. After checking the values, click on *Finish*.



**Figure 4: Initialize Timing window**

## Assigning Input Values

Now when the HDL Benchner opens up in the Project Navigator window, you will find that the **CLK** input is assigned a clock waveform with a 100ns period. Now assign a zero logic level to **PRE** and one logic level to **INPUT** at 1150 ns. The digit 0 should appear to the right of the signal names in the simulator window. Save your waveform configuration by clicking on the Save Waveform icon  in the HDL Benchner window.

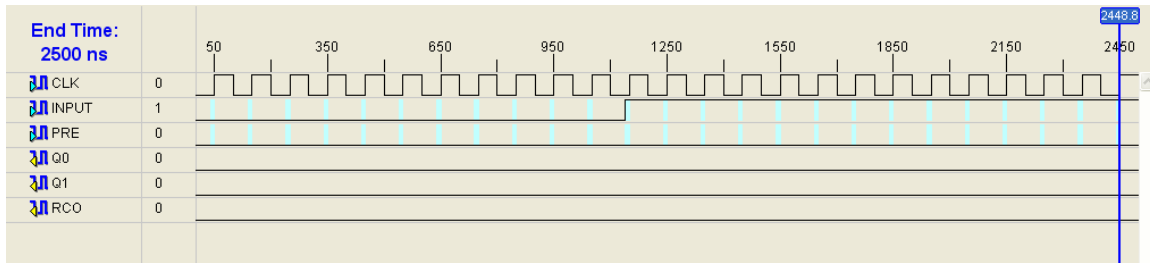


Figure 5: HDL Benchner window

In the Sources sub-window and the Sources tab, select *Sources for Behavioral Simulation* and click on the waveform file. In the Processes sub-window and the Processes tab, click on the plus sign next to *ModelSim Simulator*. Right click on *Simulate Behavioral Model* and select *Properties*. In Process Properties window, change *Simulation Run Time* to 2500 ns and click OK. Double-click on the option *Simulate Behavioral Model*. This will run ModelSim and open up the waveform with the outputs in the ModelSim and Wave-Default window.

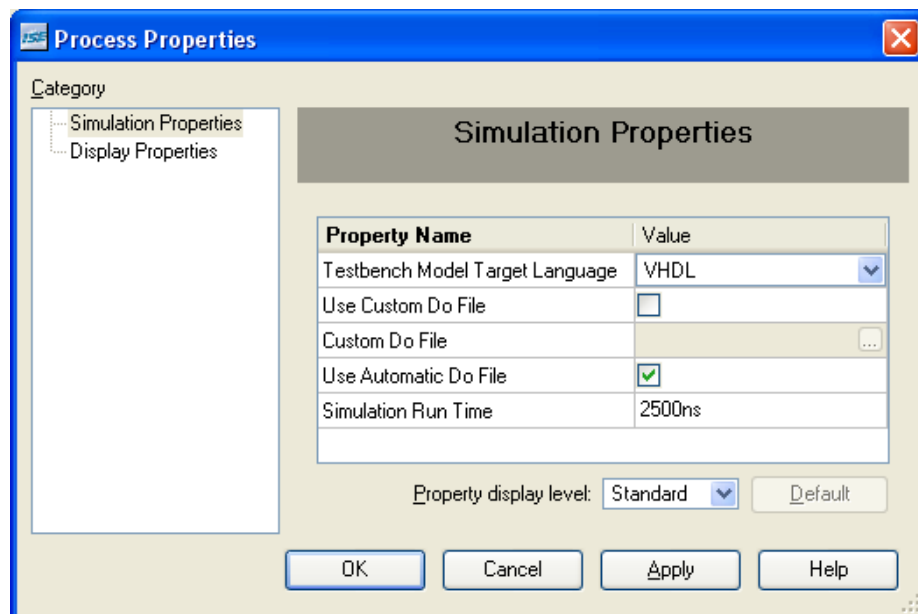


Figure 6: Process Properties window

You might be able to figure out what this machine does from this single simulation. Consider the two outputs **Q0** and **Q1** as a single 2-bit bus with **Q0** forming the MSB and **Q1** forming the LSB.

**Problem 1: Run a simulation containing ten clock periods with INPUT = 0, followed by ten clock periods with INPUT = 1. What does this circuit do? Make sure you describe what the PRE and INPUT inputs do, and what the RCO output does. Attach a printout of the simulation waveforms and write on the printout as necessary to document your conclusions.**

**Problem 2: Based on your answer to Problem 1, suggest a technique to make the circuit operate "in reverse" given the same inputs. Modify the schematic to implement your technique, simulate the resulting circuit, and attach a simulation printout as in Problem 1. Did your modification work?**

**Note: If the definition of "in reverse" operation is unclear. Please clearly specify your definition of "in reverse" operation and the modification should follow your specification.**

**Problem 3: Using the analysis techniques covered in class, obtain a state/output table and a state diagram for this circuit. Is this a Mealy or a Moore machine? Why?**

## Timing Simulation

*Remove the modification you added for Problem 2 before continuing.*

Now you will use default parameters for the PLD devices to determine the maximum possible clock speed for this device. To do this in the *Xilinx* environment, you must take your design all the way through the "implementation" process and then run a timing simulation.

In the Sources sub-window and the Source tab, select *Source for Synthesis/Implementation* and the schematic file *top (top.sch)*. You can analyze your design by looking at two analysis reports as follows.

In the Processes sub-window and the Processes tab, click on the (+) sign in front of *Implement Design*, Click on the (+) in front of *Synthesize – XST*. You can view the synthesis report by double-clicking the option *View Synthesis Report*.

Under *Implement Design*, Click on the (+) in front of *Optional Implementation Tools*, and click on the (+) in front of *Generate Timing*. You can view the timing report by clicking the option *Timing Report*.

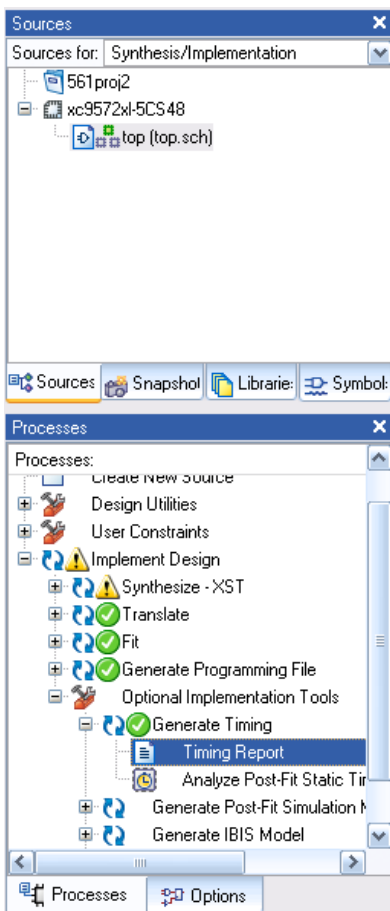


Figure 7: Process for Source window with Timing Report selection

**Problem 4: According to the Timing Report analysis, what is the maximum clock speed for this circuit? Attach the Timing Report.**

## Report

The report should be typed. Be sure to include:

- Answers to the questions in Problems 1-4
- A printout of the schematic of your circuit (original and modified)
- A trace output from each simulation (waveforms)
- State/output table and state diagram for Problem 3
- Timing Report in Problem 4