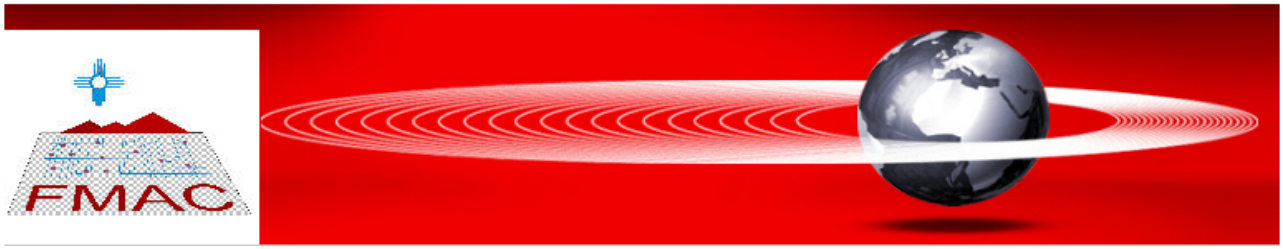

Tutorial 4
Sequence Detector, ISE 10.1 on the Digilent Spartan-3E board

Introduction

In this lab, we will implement a sequence detector on the Spartan-3E starter board. The sequence detector will look for the input series "10010." The LED's will show how much of the series has been detected and when the entire series has been entered an additional LED will come on. Circuit input will be controlled by a reset button, another button that sends a clock pulse, and a switch that will enter a '1' or a '0'.

Objective

The objective of this tutorial is to introduce the use of **sequential logic**. The sequence detector we will build is a sequential circuit or more specifically a *clocked synchronous state machine*.
Up to this point we have been working with **combinational logic**. With combinational logic the output of the circuit depends only on the current input values. In sequential logic the output depends on the current input values and also the previous inputs.

When describing the behavior of a sequential logic circuit we talk about the **state** of the circuit. The state of a sequential circuit is a result of all previous inputs and determines the circuit's output and future behavior. This is why sequential circuits are often referred to as *state machines*.

Most sequential circuits (including our sequence detector) use a clock signal to control when the circuit changes states. The inputs of the circuit along with the circuit's current state provide the information to determine the circuit's *next state*. The clock signal then controls the passing of this information to the *state memory*. The output depends only on the circuit's state, this is known as a *Moore Machine*. Figure 1 on the next page shows the schematic of a Moore Machine.

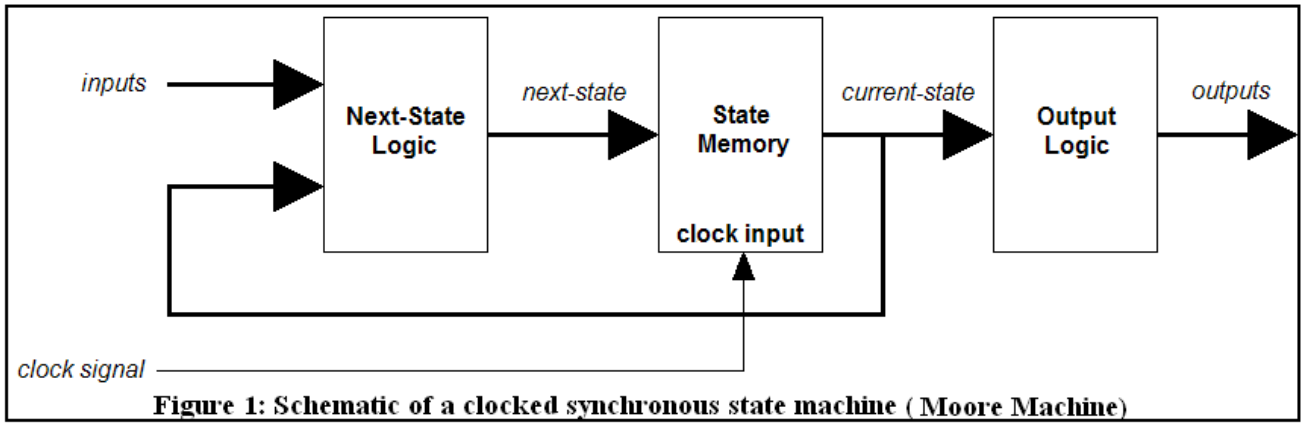

Figure 1: Schematic of a clocked synchronous state machine (Moore Machine)

A sequential circuit's behavior can be shown in a *state diagram*. The state diagram for our sequence detector is shown in figure 2. Each circle represents a state and the arrows show the transitions to the next state. Inside each circle are the state name and the value of the output. Along each arrow is the input value that corresponds to that transition.

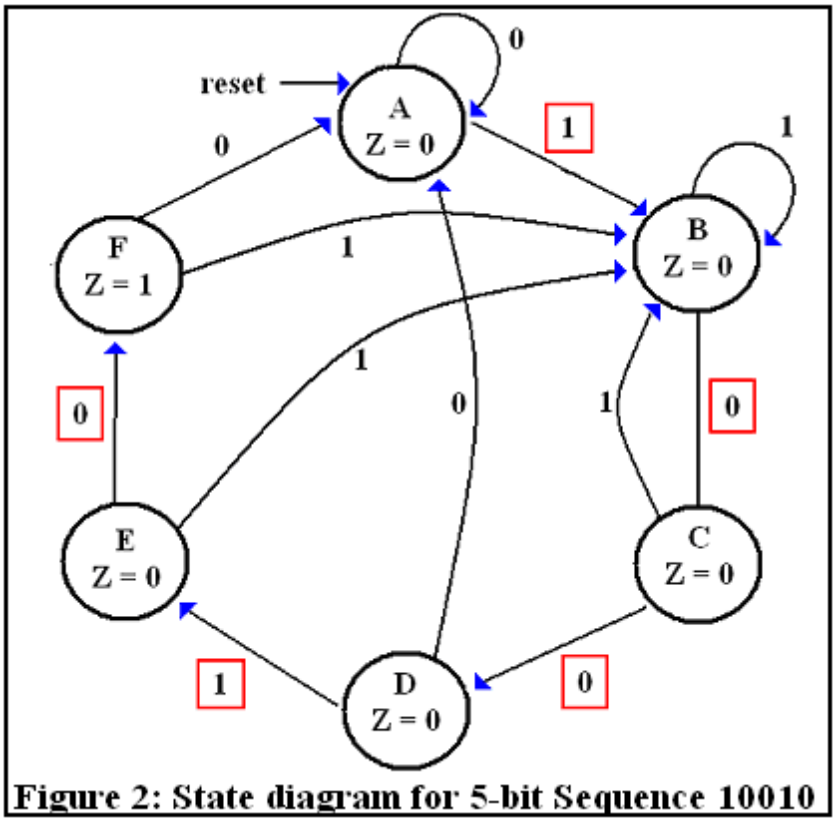

Figure 2: State diagram for 5-bit Sequence 10010

Process

1. Create project using ISE 10.1
2. Test behavior of the sequence detector using ISE 10.1.
3. Configure FPGA with the sequence detector
4. Test behavior of sequence detector on the Spartan 3E starter board

Implementation

1. Go to www.fpgamac.com and download the following files into a temporary folder.

    a. top_sequence.vhd
    b. sequence.vhd
    c. sequence_tb.vhd
    d. clockbuffer.vhd
    e. top_sequence.ucf

2. Open ISE Project Navigator. If a project is already open, go to the File menu and select "Close Project". Now under the File menu select "New Project…." ISE will launch the New Project Wizard. In the **Create New Project** window under Project Name: enter your project name. Under Project Location click the button with the three dots and navigate to where you want the project to be located. Under Top-Level Source Type: make sure "HDL" is selected and then click "Next>".

3. In the **Device Properties** window copy the settings from figure 3 and then click "Next>".

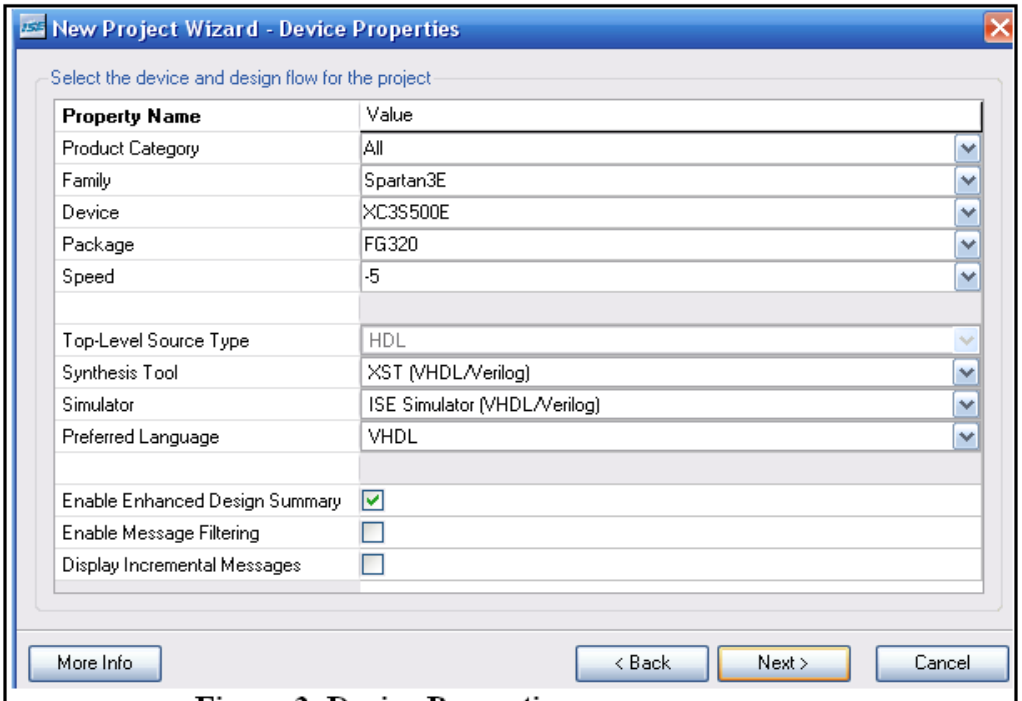

Figure 3: Device Properties

4. Click “Next>” on the **Create New Source** window. Click “Add Source” on the **Add Existing Sources** window. Select the five files that you downloaded and click “Open”. When you get the window in figure 4, click “Next>”.

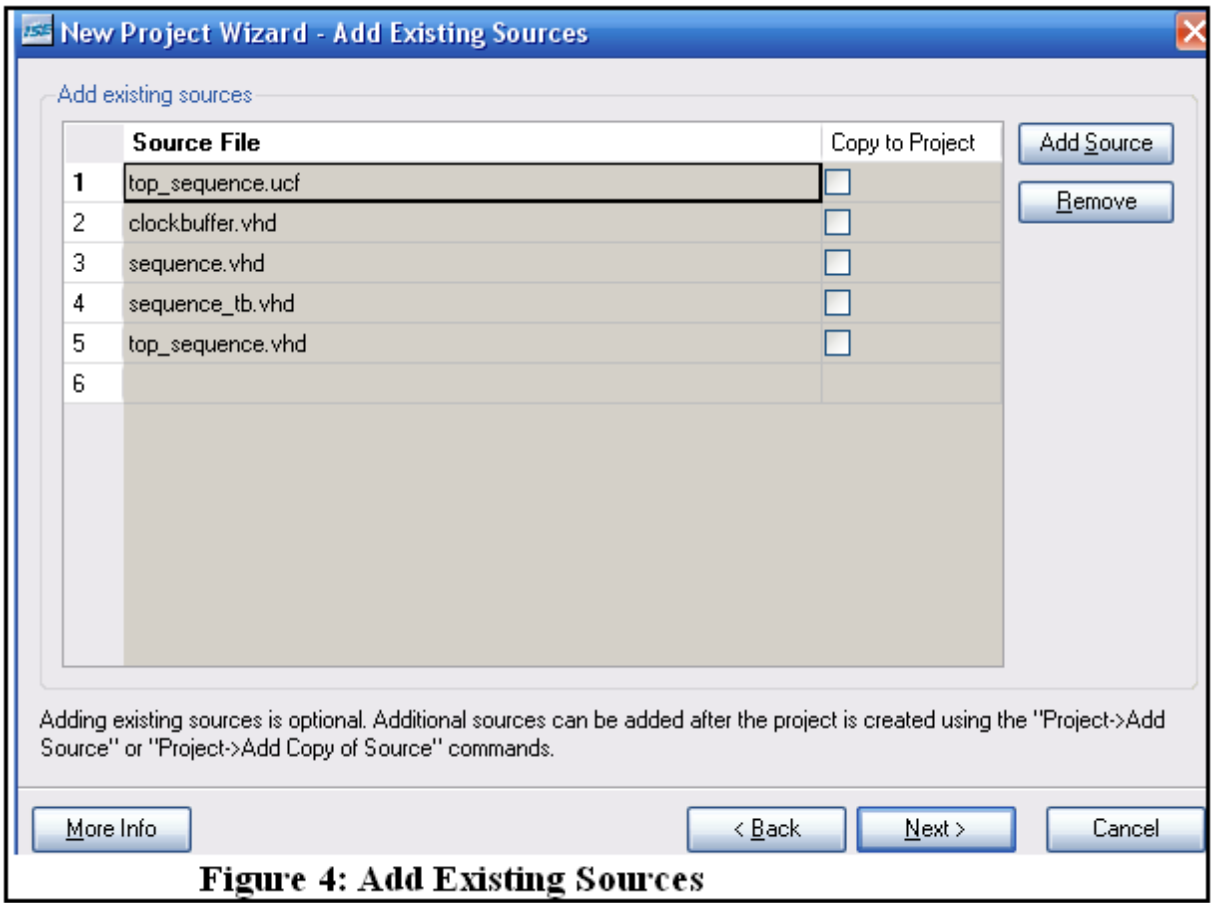

Figure 4: Add Existing Sources

5. Click “Finish” on the **Project Summary** window. Click “Ok” on the **Adding Source Files…** You have created the project and in the workspace window of ISE you should see a project summary. You can close the project summary by going under the File menu and selecting “Close”.

6. In the **Sources** window, expand the file hierarchy by clicking on the small box with the “+” symbol that is next to top_sequence.vhd (see figure 5). Now open top_sequence.vhd, sequence.vhd, and clockbuffer.vhd in the ISE workspace by double clicking on the file names.

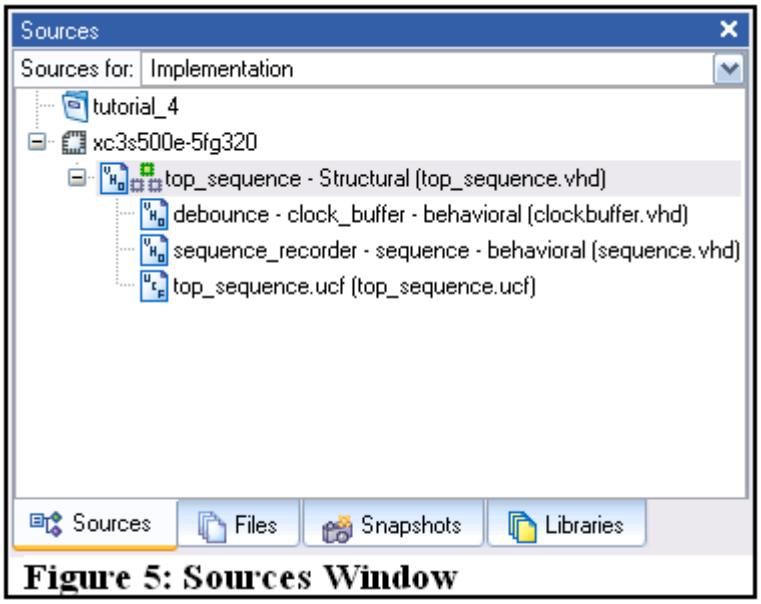

Figure 5: Sources Window

Go to the pull down menu in the **Sources** window and select “Behavioral Simulation”. Now you can open sequence_tb.vhd in the ISE workspace (see figure 6).

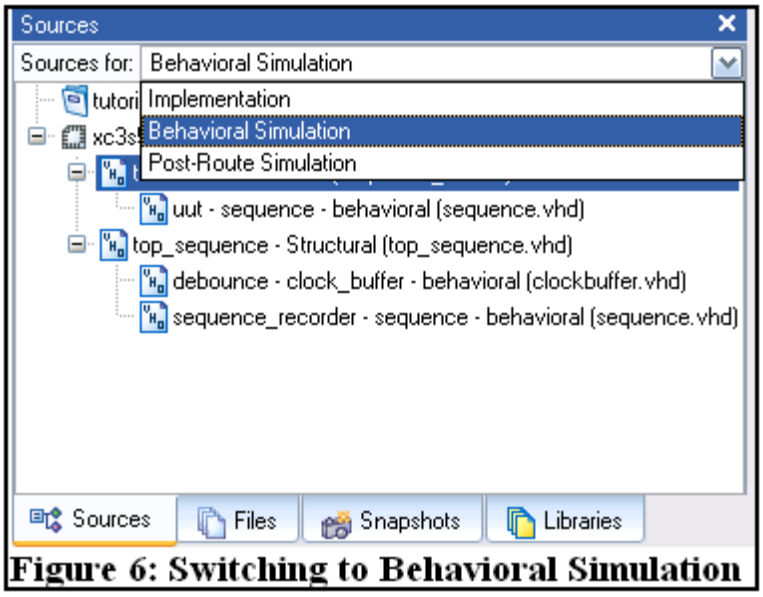

Figure 6: Switching to Behavioral Simulation

7. Take some time to look through the *.vhd files. They have been notated to help you understand the VHDL code. The layout of the three components is shown in figure 7.

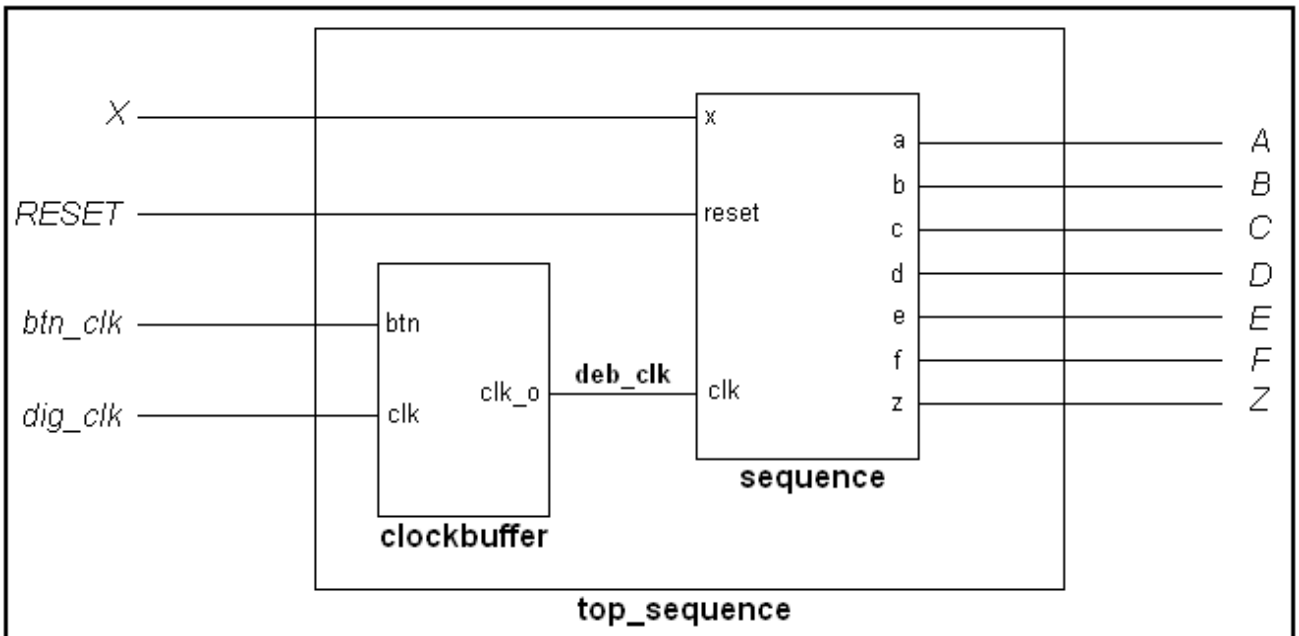

Figure 7: Structure of top_sequence

Highlight the testbench file in the **Sources** window by clicking on it. In the **Processes** window, click on the small box with the "+" symbol that is next to the "Xilinx ISE Simulator" toolbox and then double click "Simulate Behavioral Model" to start the simulation. See figure 8.

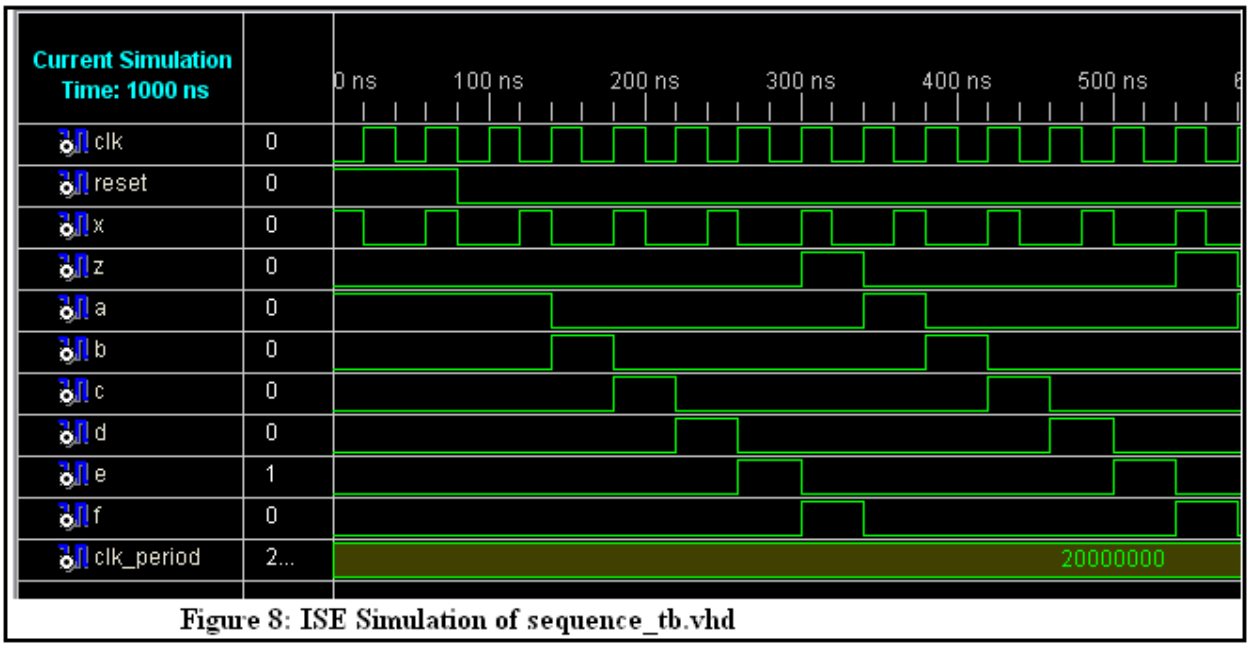

Figure 8: ISE Simulation of sequence_tb.vhd

8. Go to the **Source** window pulldown menu and select “Implementation” and then click on top_sequence.ucf to highlight it. In the **Processes** window expand the “User Constraints” toolbox and double click “Edit Constraints (Text)”. This will open top_sequence.ucf in the ISE workspace. See figure 9.

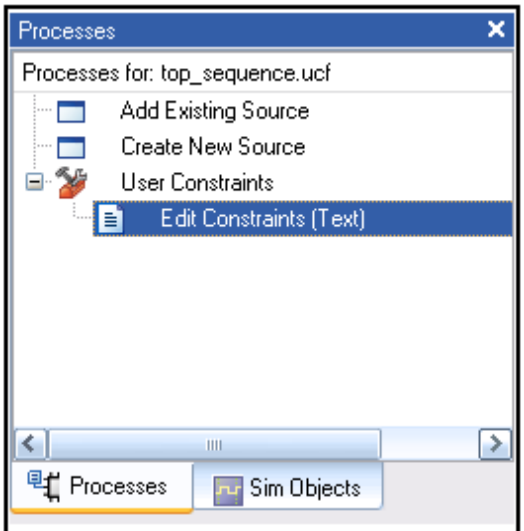

Figure 9: Edit .ucf

The user constraints file has been notated to show what board features have been connected to the inputs and outputs of top_sequence.vhd. See figure 10

```
########## Pin assignments for top_sequence ###########

## The Spartan 3E's 50 MHz clock is used in the clock buffer.

NET "dig_clk"  LOC = "C9"  ;              # CLK_50MHZ

## 'RESET' and 'btn_clk' are tied to buttons
## The 'PULLDOWN' constraint makes the button return a
## low when it is released, otherwise it will float.

NET "RESET"    LOC = "K17" | PULLDOWN ;   # BTN_SOUTH
NET "btn_clk"  LOC = "V4"  | PULLDOWN ;   # BTN_NORTH

## The data input will be controlled with a switch

NET "X"        LOC = "L13" ;              # SW<0>

## Outputs are routed to the LED's

NET "A"        LOC = "F12" ;              # LED<0>
NET "B"        LOC = "E12" ;              # LED<1>
NET "C"        LOC = "E11" ;              # LED<2>
NET "D"        LOC = "F11" ;              # LED<3>
NET "E"        LOC = "C11" ;              # LED<4>
NET "F"        LOC = "D11" ;              # LED<5>
NET "Z"        LOC = "F9"  ;              # LED<7>
```

Figure 10: User Constraint File

9. It is time to program the Spartan 3E board. In the **Processes** window you have to run the "Synthesize - XST", "Implement Design", and "Generate Programming File" processes. Instead of doing each one separately, you can double click on "Generate Programming File". This will run all the processes. See figure 11.

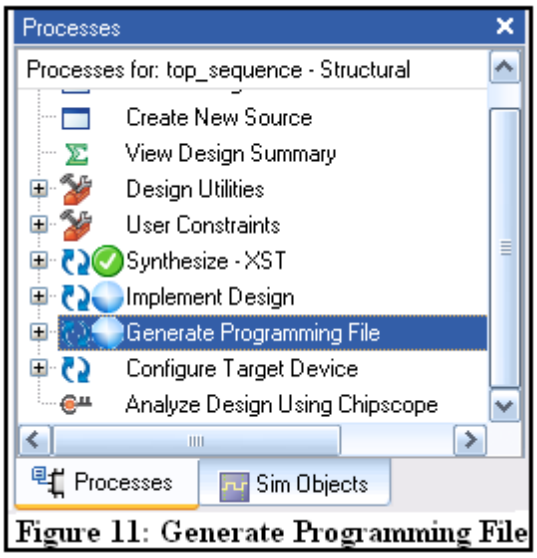

Figure 11: Generate Programming File

As the processes finish running they will be marked with a green checkmark to indicate no problems were encountered. The "Implement Design" process may generate a warning (yellow triangle with an exclamation point) about excessive skew of the clock buffer output. This warning can be ignored.

10. Connect the Spartan 3E board to the computer and turn the board's power on. Expand the "Configure Programming File" process and double click on "Manage Configuration Project (iMPACT)". This will launch the iMPACT program. Click "Finish" on the **Welcome to iMPACT** window.

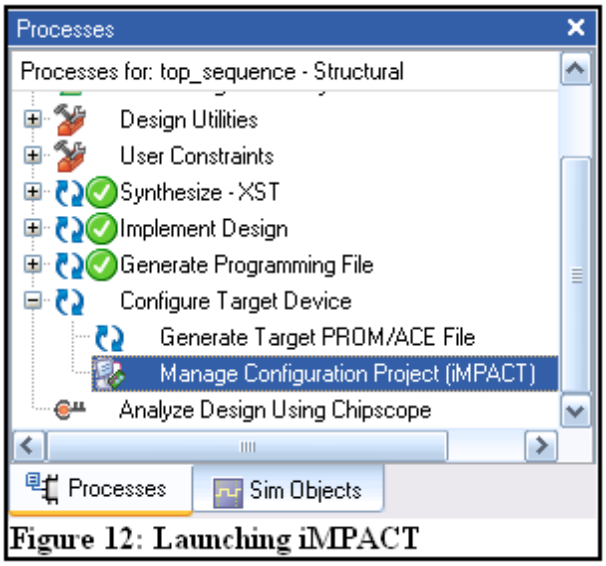

Figure 12: Launching iMPACT

iMPACT will perform a boundary scan and will display three devices in the ISE workspace. Pictures of Xilinx IC packages represent the devices.

We are going to assign the top_sequence.bit file to the Spartan 3E's FPGA. The FPGA is represented in the workspace by the picture of the Xilinx package labeled "xc3s500e". The package should already be highlighted.

Click on top_sequence.bit in the **Assign New Configuration File** window and then click the "Open" button. The file is now associated with the FPGA and the next device is highlighted. See figure 13.

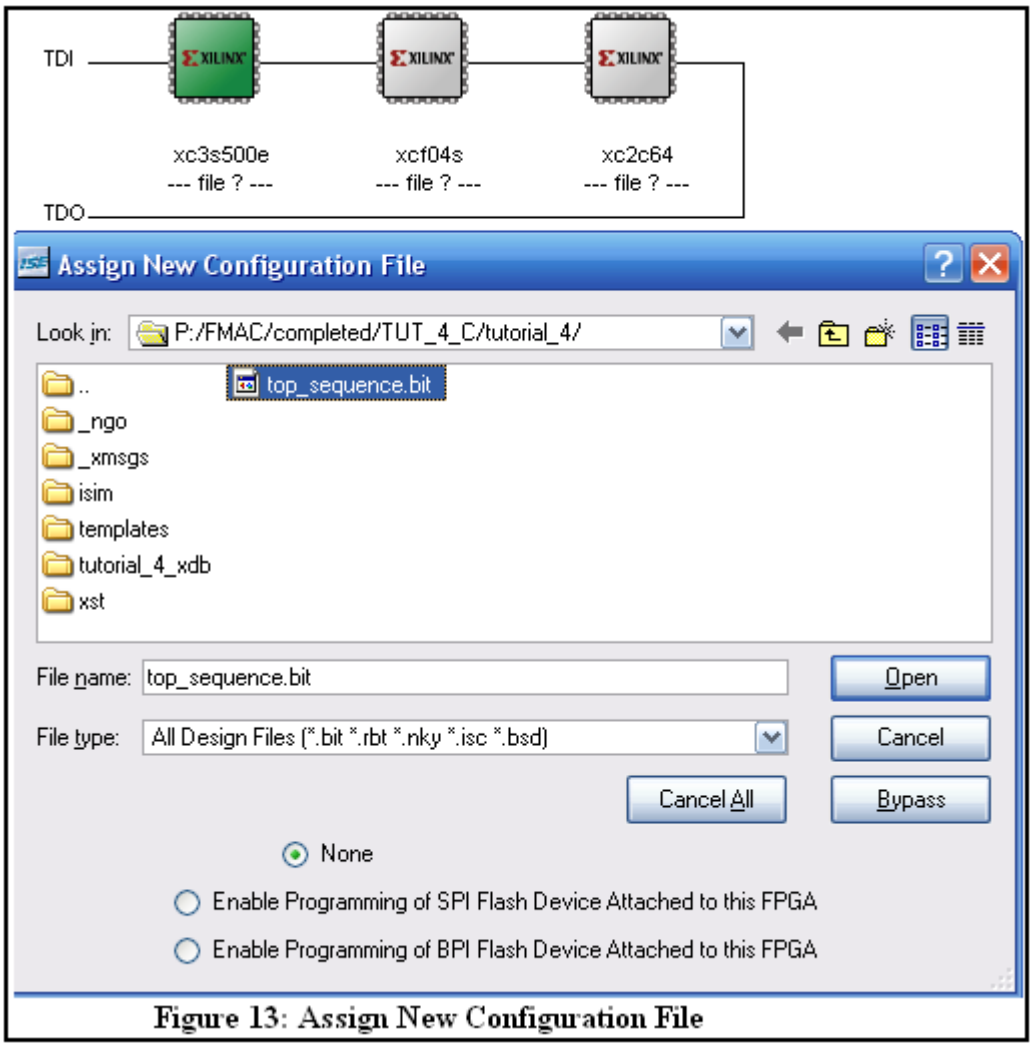

Figure 13: Assign New Configuration File

Click “Bypass” for the next two devices since we are not programming them and then click “OK” on the next screen. Now right click on the xc3s500e and select “Program…” from the drop down menu. See figure 14.

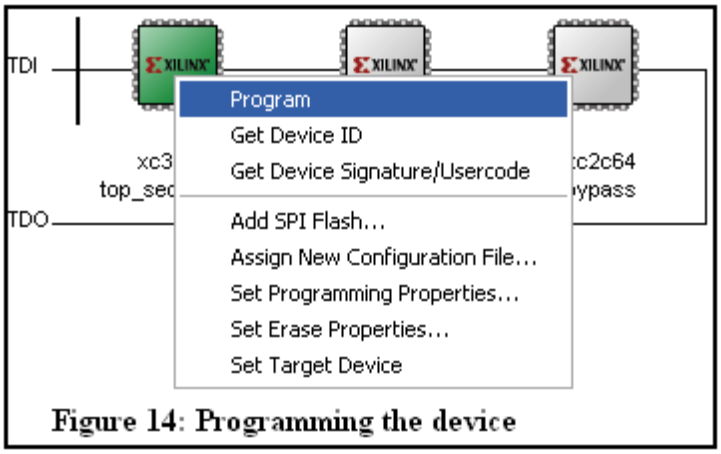

Figure 14: Programming the device

After the FPGA is programmed a “Program Succeeded” message will be displayed in the ISE workspace and a yellow LED will show the Spartan 3E has been configured. See Figure 15.

Figure 15: Program Succeeded

11. The Spartan 3E is programmed as a sequence detector. The board will hold this program until the power is turned off, the reset button near the yellow LED is pressed, or you reprogram the board.

    The inputs and outputs are labeled in figure 16 on the next page. To reset the state machine press the reset button (BTN_SOUTH). To enter a ‘1’, slide the switch (SW0) to the high position and press the clock button (BTN_NORTH). To enter a ‘0’, slide the switch to the low position and press the clock button. The LED’s will light to show the state. When the entire sequence has been detected an additional LED will come on. If you press and hold the clock button, a clock signal will be sent approximately every 0.23 seconds. This is the time the buffer delays the button signal from reaching the state machine.

This tutorial was authored by Stephen Tomany. Stephen is a Junior in the Electrical Engineering Department at The University of New Mexico in Albuquerque. Questions or comments can be sent to stomany@unm.edu.

Rev. 11/25/08

Revision to Xilinx 10.1 completed by Brian Zufelt. Brian is a Junior in the Electrical Engineering Department at The University of New Mexico in Albuquerque.

Acknowledgment:

Wakerly, John F. (2006). *Digital Design: Principles and Practices*. 4th edition. New Jersey: Pearson Prentice Hall.

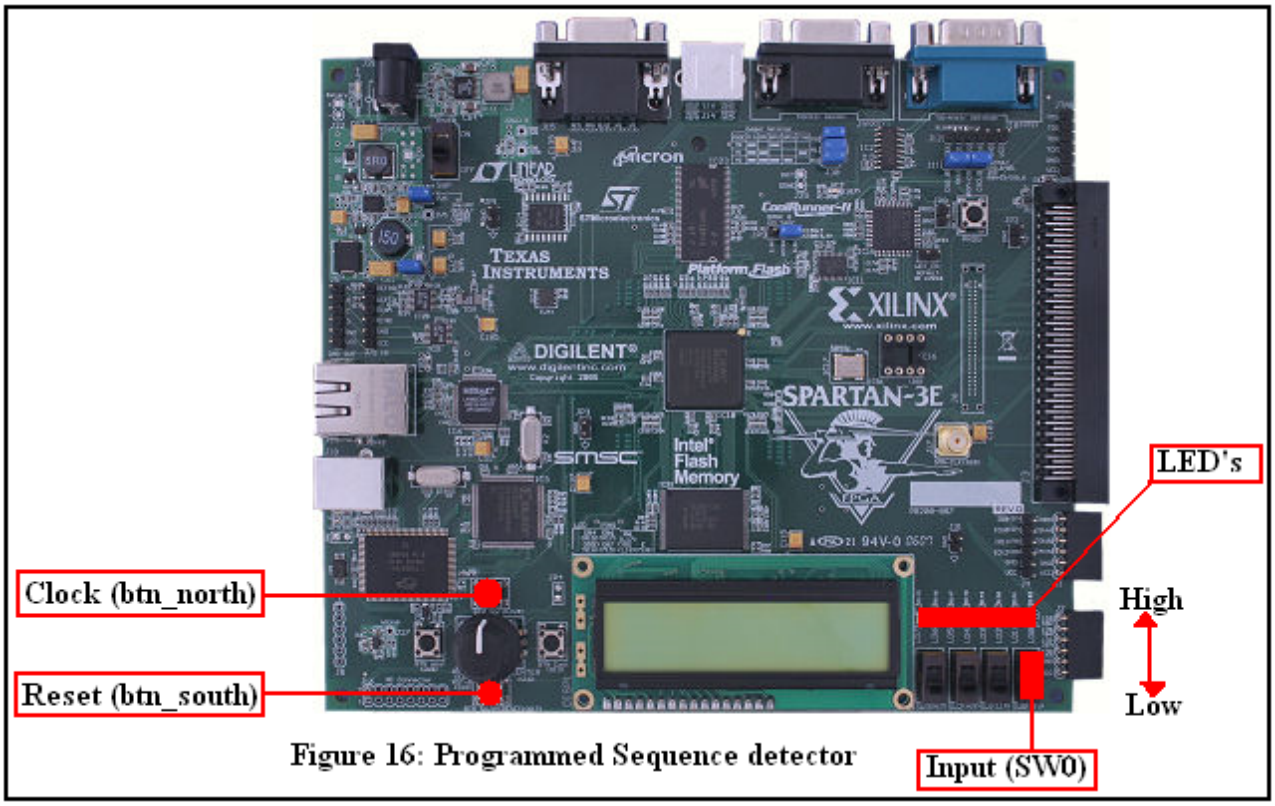

Figure 16: Programmed Sequence detector