Lab 1 – Introduction to Development Environment

CS2052 Computer Architecture

Dept. of Computer Science and Engineering, University of Moratuwa

Learning Outcomes

In this lab we will learn how to develop logic circuits using an FPGA (Field-Programmable Gate Array). After completing the lab you will be able to:

- design a simple logic circuit using schematics
- connect the inputs and outputs of your circuit to switches and LEDs
- generate the bitstream
- configure the FPGA using the generated bitstream
- verify the functionality on the development board

Introduction

The laboratory assignments will use the Digilent BASYS 2 development board shown in Fig. 1. BASYS 2 is a circuit design and implementation platform that can be used to gain experience in building real digital circuits. Built around a Xilinx Spartan-3E FPGA, the BASYS 2 provides complete, ready-to-use hardware suitable for hosting circuits ranging from basic logic devices to complex controllers. The board comes with a large collection of on-board Input/output (I/O) devices such as switches, push buttons, LEDs, 7-segements, PS/2 port, VGA port, etc.

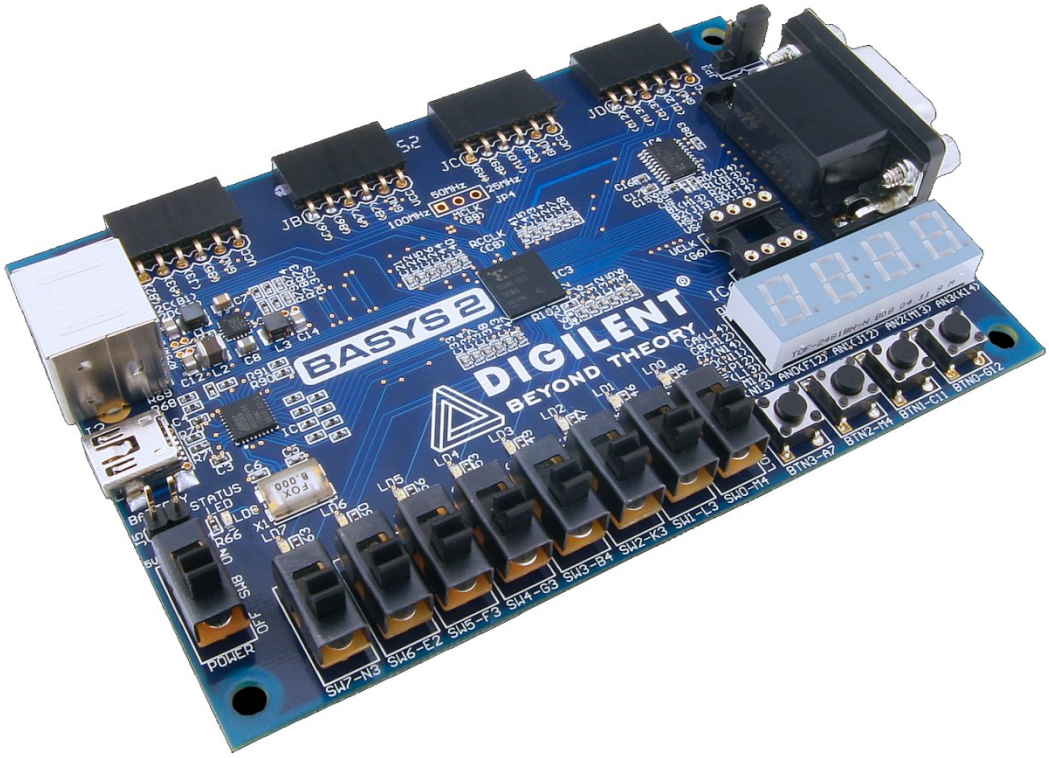

Figure 1 – Digilent BASYS 2 Spartan-3E FPGA development board.

An FPGA is an integrated circuit designed to be configured by a designer after manufacturing – hence "field-programmable". The FPGA configuration is generally specified using schematics or Hardware Description Language (HDL). In our labs, we will only use the schematics to configure/program the FPGA, as it is relatively easier to lean and use. Alternatively, HDL take more time to lean but very useful while building more complex circuits.

We use Xilinx ISE Design Suite software to program the FPGA using schematics. The laboratory assignments are based on version 14.7 of the Xilinx ISE WebPack edition. As the BASYS 2 board is not developed by Xilinx, we have to follow an additional step to program the FPGA. Instead of directly programming the BASYS 2, we use Digilent Adept software to download the bitstream generated by the Xilinx ISE.

Note As you are new to digital design, do not be surprised if it appears intimidating at first. Do not despair. We will selectively introduce the features as needed while ignoring others that we do not need. As the semester progresses, you will understand more and more features, and by the end of the semester, you will be able to design complex circuits using many features of the ISE Design Suite.

Tips In addition to using the computers in the lab, you can also install the ISE WebPack edition in your own computer. Installation files are 6.5 GB (installation requires ~ 17 GB). Hence, obtain the installation files from the instructor rather than downloading them from the web. Once installed, you will be asked to register the product to obtain a license file.

Design Flow

Typical steps involved in programming an FPGA are depicted in Fig. 2. Following is a brief description about the various steps involved:

- Create Design – We first create an ISE Design Suite project and then, create or add source files to that project. Projects can contain many types of source files and design modules, including schematic, HDL, embedded processor, and digital signal processing modules. We will use only the schematics for our laboratory assignments.
- Simulate Design – At various points during the design flow, we can verify the functionality of the design using a simulation tool. We use ISim, which is delivered with the ISE. This step will be covered in a later lab.
- Synthesize Design – During synthesis, the synthesis engine compiles the design to transform schematic or HDL sources into an architecture-specific design *netlist* (i.e., connectivity of an electronic design). The ISE supports the use of Xilinx Synthesis Technology (XST), which is delivered with the ISE.
- Implement Design – After synthesis, we run design implementation, which converts

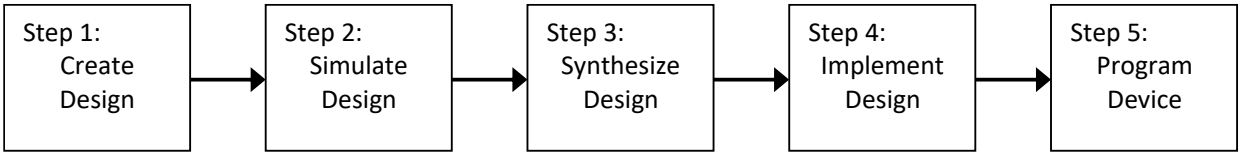

Figure 2 – Design flow overview.

the logical design into a physical file format (i.e., *bitstream*) that can be downloaded to the selected target FPGA (a.k.a. device). Bitstream tells how to configure the given FPGA to build the desired circuit.

- Program Device – After generating a programming file (i.e., bitstream), we configure the device. To download the bitstream from a host computer to a device, we use Digilent Adept software.

My First Circuit

Step 1: Starting ISE Design Suite

Start **ISE Design Suite 14.7** by locating the icon on the Windows **Desktop** or Windows **Start** menu.

You should see a display like the one in Fig. 3. This display consists of several windows that provide access to various features of the ISE software. Tip of the Day pop-up window shows various tips every time you load the software. Click **OK** to close the **Tip of the Day** window.

Most of the commands provided by the ISE can be accessed using a set of menus that are located below the title bar.

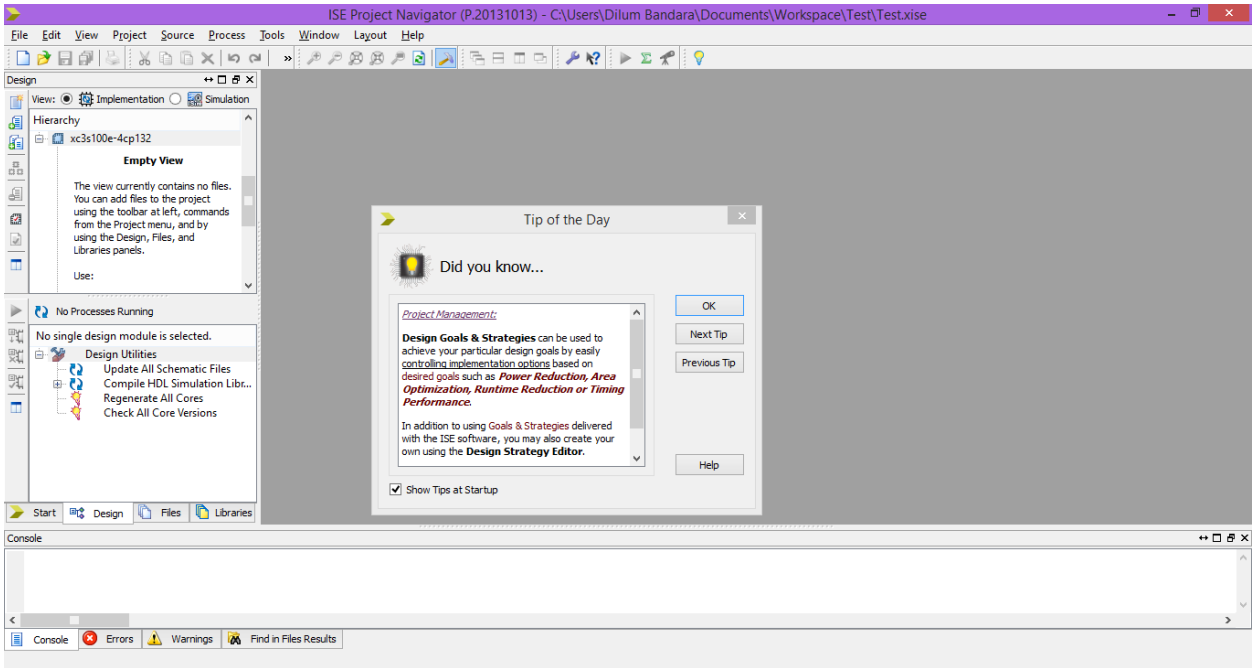

Figure 3 – ISE Project Navigator window.

Step 2: Starting a New Project

To create a new project either click on the **New Project...** button on the **Start** panel or **File → New Project...** from the menu.

This opens up New Project Wizard dialog box. Enter project **Name:** as **Lab 1**. Use **Location** textbox (or **...** button) to a set a suitable location to store your project files.

Set **Top-level source type:** list box to **Schematic**. The completed dialogue box should be similar to Fig. 4.

Click **Next >** button to continue.

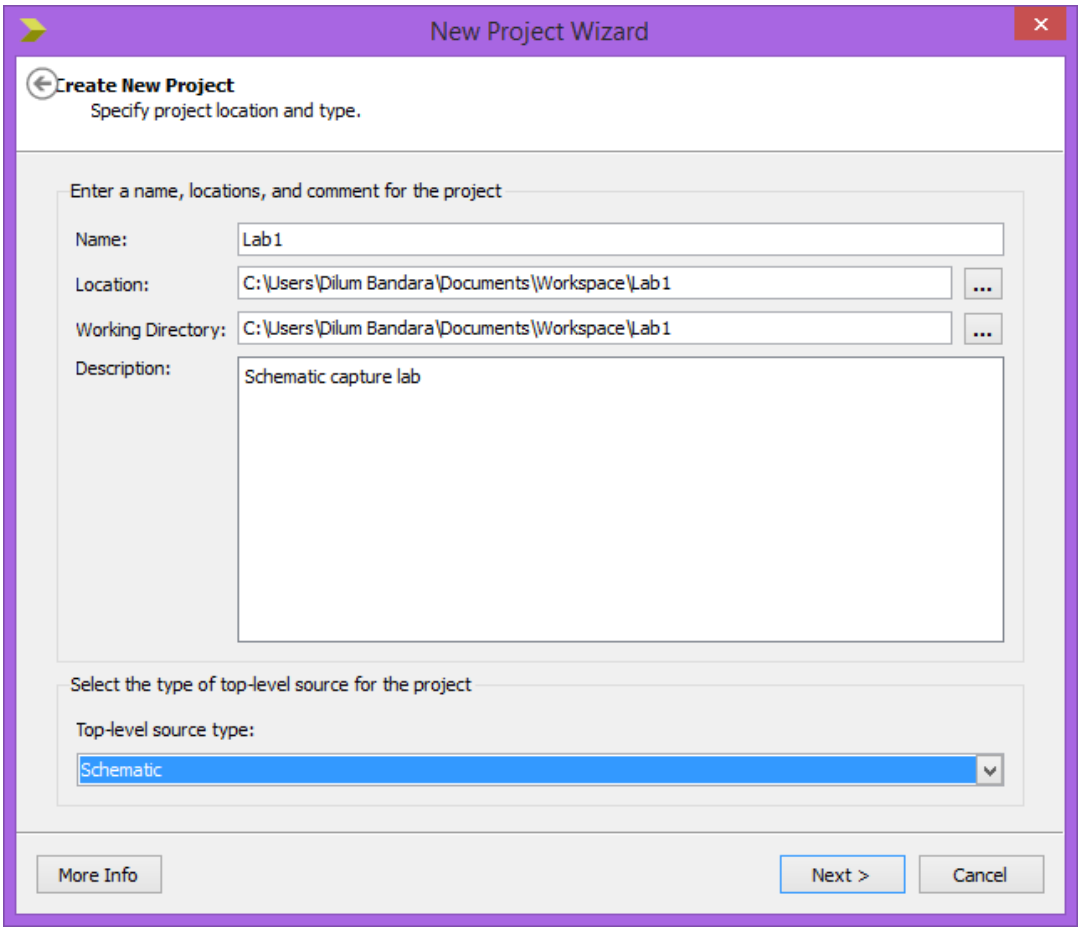

Figure 4 – New Project Wizard dialog box.

Step 3: Project Settings

Next, we configure the properties of the device and design flow. Set the properties as given in Table 1. These parameters depend on the FPGA you are targeting for the project. The completed dialog box should be like Fig. 5.

Click **Next >** button to continue.

Then you will see a summary of the project. Double check and make sure all the settings are correct. Any modifications can be made by clicking the **Back** button.

Click **Finish** button.

Table 1 – Device and design flow properties.

Property Name	Value
Evaluation Development Board	None Specified
Product Category	All
Family	Spartan 3E
Device	XC3S100E
Package	CP132
Speed	-4
Top-Level Source Type	Schematic
Synthesis Tool	XST (VHDL/Verilog)
Simulator	ISim (VHDL/Verilog)
Preferred Language	VHDL

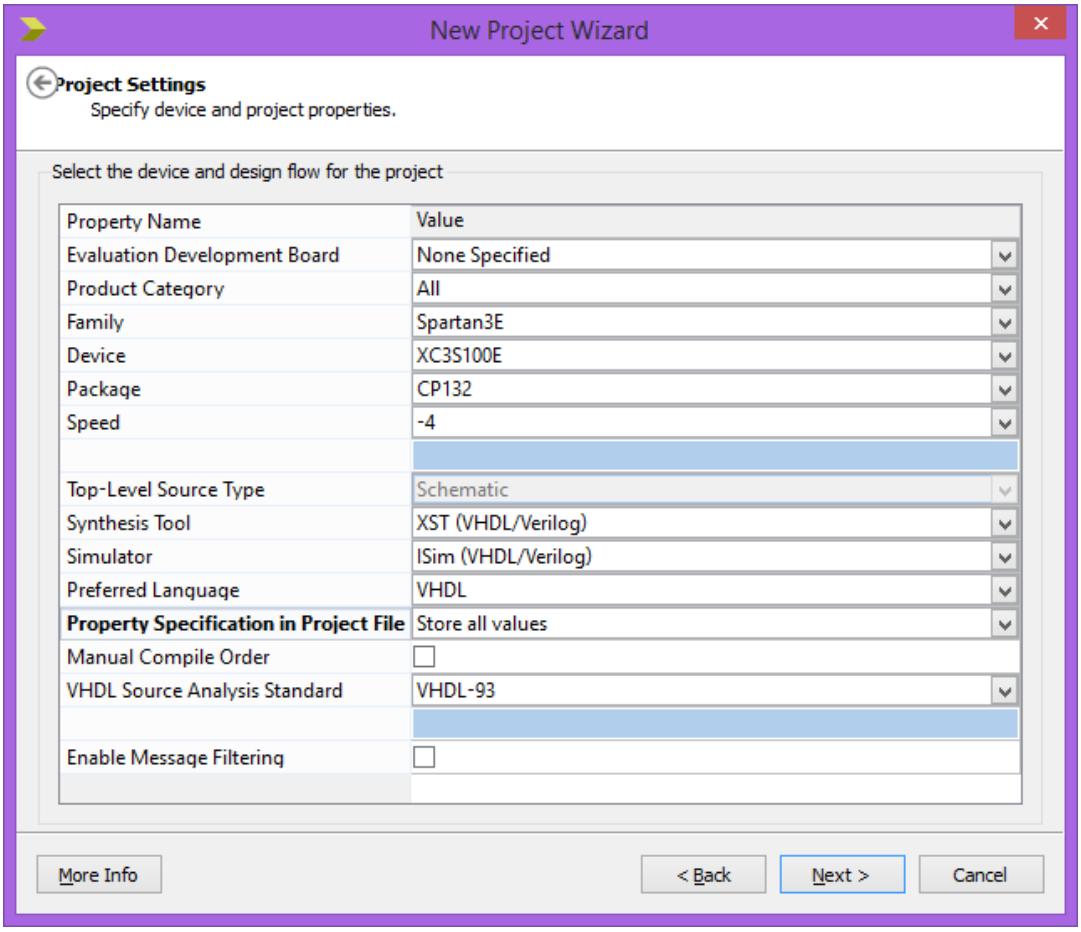

Figure 5 – New Project Wizard dialog box with device and design flow parameters.

Once the new project has been created, ISE opens the project in Project Navigator. Navigator can be split into three areas as Design panel (located on the left), Console panel (on the bottom), and HDL/Schematic Editor (on the right). The design panel consists of two sub-panels, namely Sources window (on top

left) and Processes window (on the bottom left). Sources window displays all source files associated with the current design and Processes window displays all available processes that can be run on a selected source file. Console panel displays status messages, including error and warning messages. HDL/Schematic Editor window displays source code or the schematic from files selected in the Design panel.

Step 4: Adding New Source Files

Once the new project is created, two sources are listed in the Sources window on the Design panel. It should list the Project file name and the Device targeted for the design.

To add a new or existing source files to a project, right click on the target device **xc3s100e-4cp132** and select **New Source...** from pop-up menu (see Fig. 6).

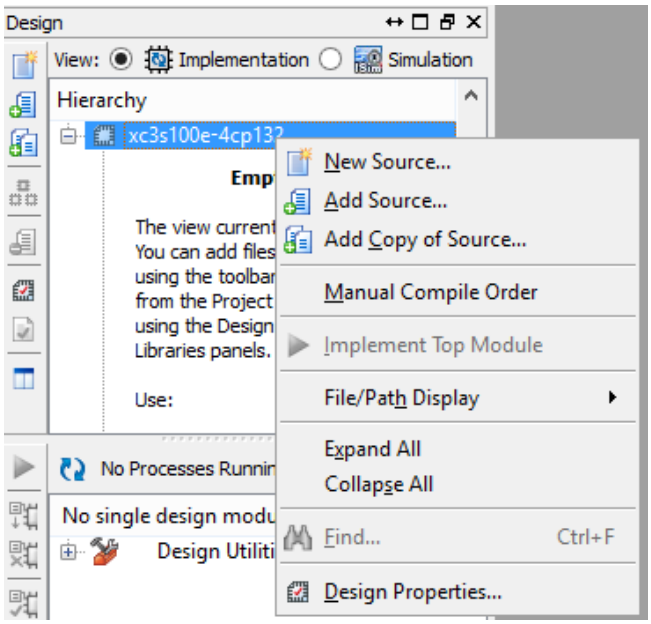

Figure 6 – Adding a new source file.

This will open up the New Source Wizard dialog box (see Fig. 7).

Select **Schematic** from the list and set a suitable **File name:** (e.g., Circuit 1).

Click **Next >** button. Once the summary appears, click on the **Finish** button.

Once you have created the new schematic file, you can see it in the Sources panel. If the Schematic Editor window (see Fig. 8) does not open up automatically, double click on the source file name to open it.

Step 5: Building the Circuit

Schematic Editor can be used to build circuits by adding symbols and shapes representing logic gates or logic circuits and then by interconnecting them using lines that represent wires.

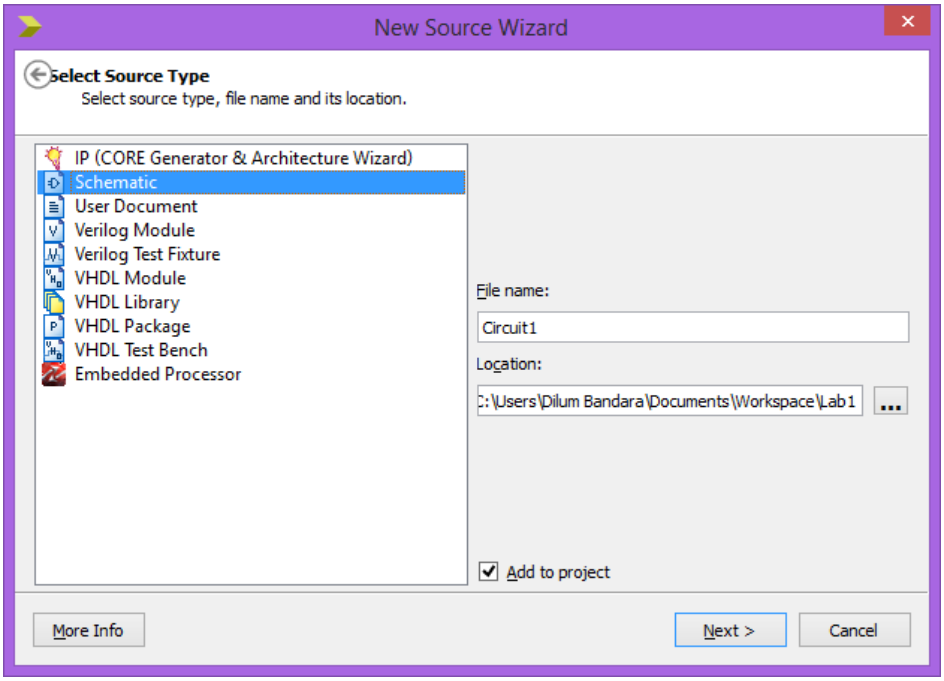

Figure 7 – New Source Wizard dialog box.

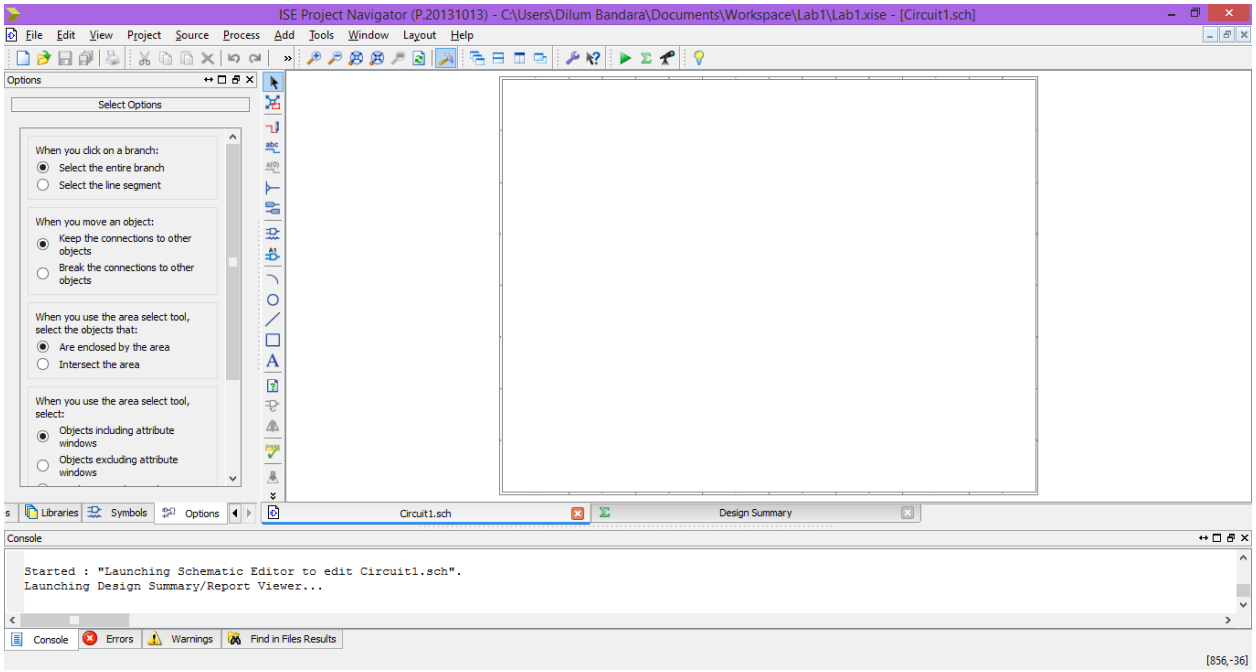

Figure 8 – Schematic editor window.

Click on the **Symbols** tab in the Design panel.

You will see two list boxes labelled Symbols and Categories. The Symbols list shows all the symbols in selected category in the Categories list. For example, the “<--All symbols-->” category displays all symbols in the current library.

Using suitable symbols build a digital circuit that represents the following logic equation.

```
X = (A . B) + C
```

Tips You can use the **Symbol Name Filter** textbox to search for symbols as well.

Use the Zoom icons on the main toolbar (under the menu bar) to resize the display so that the circuit appears clearly.

To add wires, click on the **Add Wire** icon on the toolbar (located to the right of Design panel).

I/O markers are used to connect the inputs and outputs to external components outside the FPGA. Adding I/O markers to your circuit also tells the synthesizer and simulator tools which ports to regard as overall inputs and outputs.

Add four I/O marks (for A, B, C, and X) using the **Add I/O Marker** icon on the toolbar.

Tips Frequently used functions/options can also be accessed by right clicking on the Schematic Editor window.

Go back to regular cursor mode and double click on an I/O marker. When the I/O marker's object properties dialog box appears, select the **Nets** category and change the labels of input and outputs as **A**, **B**, **C**, and **X** using the **Name** field of the I/O marker.

Make sure to save all changes to the project using **Save All** icon or menu option. Make sure to frequently save your work.

The completed circuit should be similar to Fig. 9.

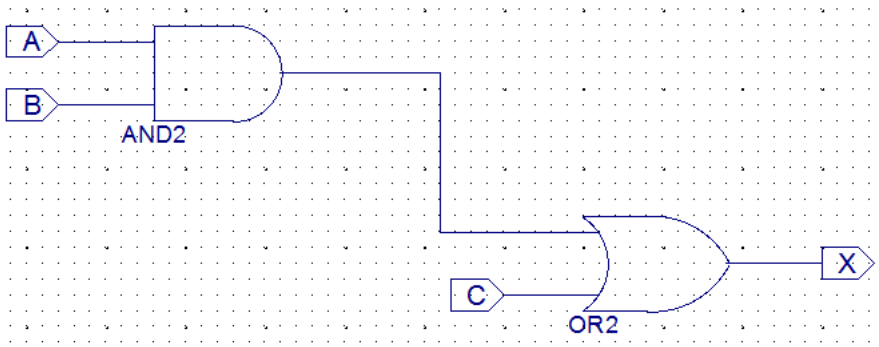

Figure 9 – Schematic view of the circuit.

Step 6: Creating User Constraints File

A user constraints file (.ucf file) defines user constraints like physical *pin* to circuit *net* mappings (i.e., mapping the buttons and LEDs on the BASYS 2 board to inputs/outputs in the circuit). This is sometimes referred to as an implementation constraints file. The .ucf file can be modified inside ISE using a text editor.

To add an .ucf file to your design, go to the **Sources** window and right click on the source file that requires user constraints. Select **New Source...** option from

the drop-down menu. Select **Implementation Constraints File** from the New Source Wizard dialog box. Name the file as **PinAssignment**.

To edit the .ucf file, select **PinAssignment.ucf** file listed in the Sources window. Then expand the **User Constraints** option in the Processes window below, and double-click on the **Edit Constraints (Text)** option. This should open up a blank text editor.

Each entry in the file should have the following format:

```
NET "net_name" LOC ="xxx";
```

In the statement, "net_name" is the name of the *net* (i.e., input or output on our circuit) to attach to the *pin* (connection on BASYS 2 board) number xxx. Make sure to include quotes. For our project, the three inputs are assigned to switches 0 through 2 (SW0 – SW2) and the output is assigned to LED0 (LD0) on the BASYS 2 board. A partly filled .ucf file is as given below.

```
NET "A" LOC ="P11";
NET "B" LOC ="L3";
NET "C" LOC ="  ";
NET "X" LOC ="  ";
```

Complete the entries for pins C and X by finding the correct pin assignment given in Digilent Basys2 Reference Manual (this file can be found in Moodle).

Tips Don't copy and paste above lines from PDF to ISE text editor. PDF may introduce hidden characters which ISE is not able to interpret.

Step 7: Generating the Programming File

Now we are ready to create a programming file (a.k.a. bitstream, .bit file) for the FPGA in BASYS 2.

Go to the **Sources** window and select the schematic file. Next, go to the **Processes** window. You should see the following 3 processes listed:

1. Synthesize - XST
2. Implement Design
3. Generate Programming file

Run the synthesis process either by double clicking on the **Synthesize – XST** or clicking and selecting the **Run** option.

This process analyses the circuit you have created, checking for valid connections, syntax, and structure, to verify that the circuit is valid and synthesizable. This process may take some time depending the complexity of your circuit and speed of the computer.

If the synthesis process is successful, you will see a message similar to the following on the Console panel (tick will also appear on the Processes window):

```
Process "Synthesize - XST" completed successfully
```

If the process returns any error messages go and double check the circuit. Do necessary changes, and then rerun the **Implement Design** process as above.

This process uses various algorithms to map out the digital circuit and then creates place and route information so that it can be placed on the physical FPGA. It may also take a while to complete. Once successful you should see a message similar to the following:

```
Process "Generate Post-Place & Route Static Timing"
completed successfully
```

If the Implement Design process does not return any errors, you can run the Generate Programming File process. Before we do this, we need to specify the signal that will be used to clock the startup sequence at the end of the FPGA configuration process. This option allows a configuration file to configure a board straight from the computer, load a configuration from platform Flash memory on the board, or use an external clock source connected to the clock pin on the BASYS 2.

Right click on the **Generate Programming File** process and select **Process Properties...**.

In the **Category** pane of the Process Properties dialog box, select **Startup Options**. The first item in the right panel is the **FPGA Start-Up Clock** property. To configure the board from the computer, the start-up clock value should be set to **JTAG Clock**. Complete window should be like Fig. 10.

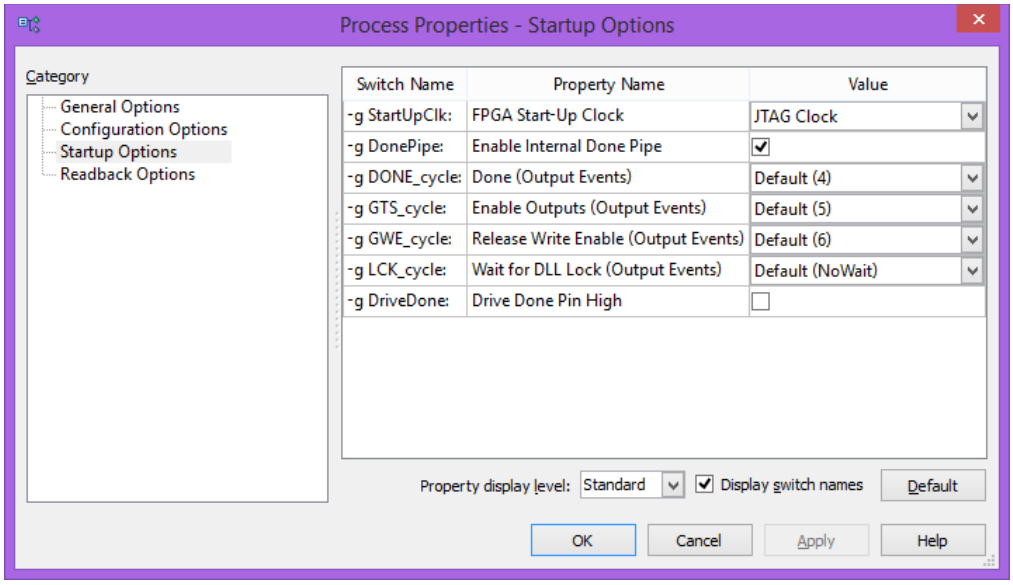

Figure 10 – Process Properties dialog box.

Then click **OK** button. Run the **Generate Programming File** process. After this process completes, a configuration .bit file (**circuit1.bit** in our case) should appear in the directory where your project is located.

Step 8: Configuring the Board

As the BASYS 2 board is not developed by Xilinx, we have to follow an additional step to program the FPGA. Instead of directly programming the BASYS 2, we

use Digilent Adept software to download the .bit file to the board. If the computer that you are using is not connected to a BASYS 2, talk to the instructor to get access to a computer that is connected to a BASYS 2. Following steps should be carried out only on a computer with BASYS 2 attached. You need to copy .bit file to the new computer.

Warning	Never touch/hold the development board from the top, bottom, or middle. Static charges can ruin the small electronic components. Always hold the board from the sides as shown in Fig 11.

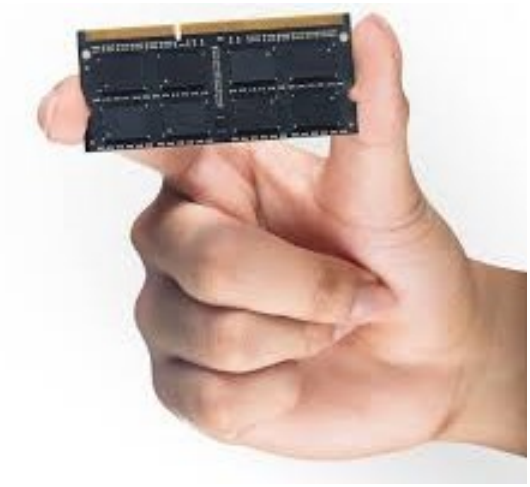

Figure 11 – How to hold a circuit board.

Make sure BASYS 2 is connected to the computer using a USB cable. Also make sure **JP3** jumper on the board is set to **PC**. This tells the program to be loaded from the PC/computer not from ROM. Set the **POWER** switch to **ON**.

Start **Digilent Adept** by locating the icon on the Windows **Desktop** or Windows **Start** menu.

Adept should automatically detect the BASYS 2 board. If not, select it from the **Connect**: dropdown list.

Adept tries to initialize itself for device configuration. It can also be manually initialized by clicking the Initialize button. After initializing, the Adept interface shows the available configuration options for the attached device in the dropdown list next to the device icon. If Adept is unable to connect to the product, or is unable to initialize, the window displays “No devices identified.” If this happens, try disconnecting and then reconnecting the board again.

Click the **Browse** button next to the FPGA box in the window. Using the Open dialog box open the appropriate configuration (**circuit1.bit**) file. The completed window should be like Fig. 12.

Adept displays a history of configuration files in the drop-down list box next to the device.

Click the **Program** button or right click on the FPGA box and click on **Program Device** button. This will program the device. Any error messages will be indicated at the bottom of Adept window.

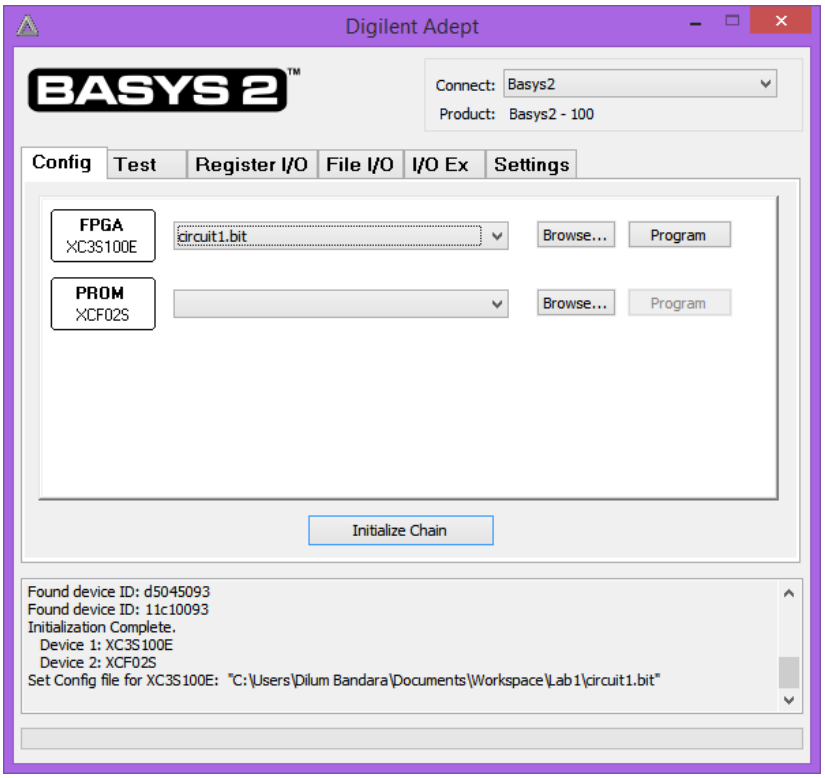

Figure 12 – Adept configuration window.

Step 9: Change the switches (SW0 – SW2) on the BASYS 2 and verify the functionality of your circuit (check the output on LED LD0).

Note As we did not program the ROM, the board will lose the FPGA configuration once the power is switched off. For the labs, we do not need to permanently program the board. Hence, even in future labs, we will not program the ROM.

Demonstrate the circuit to the instructor and get the Lab Completion Log singed.

Congratulations!!! for successfully completing the first lab.

If you finish early, develop a circuit to detect whether a given 3-bit number is prime.

Bibliography

- Digilent Inc., "Xilinx ISE WebPACK Schematic Capture Tutorial," Feb. 27, 2010.
- Digilent Inc., "Adept Application User's Manual," May 28, 2010.
- Digilent Inc., "Digilent Basys2 Board Reference Manual," Nov. 11, 2010.
- Xilinx, "Vivado Design Flow – Lab Workbook," 2013.

Prepared By

- Dilum Bandara, PhD – Jan 26, 2014
- Updated on Aug 24, 2017