Using XUP Basic Elements in IPI Design

Introduction

This lab guides you through using basic elements, provided by XUP, in basic digital design course.

Objectives

After completing this lab, you will be able to:

- Use provided XUP basic elements in IP Intergrator design
- Simulate and verify IP functionality
- Generate the bitstream and verify the functionality in hardware

Procedure

This lab is separated into steps that consist of general overview statements that provide information on the detailed instructions that follow. Follow these detailed instructions to progress through the lab.

This lab comprises 3 primary steps: You will create a project to use the provided IP, simulate the design, and verify the functionality in the hardware.

Create the Project for Using the XUP Created IP — Step 1

1-1. Create an empty Vivado project calling it as my_and2_test in the c:\xup\digital directory targeting the xc7a100tcsg324-1 device. Setup the IP Repository to point to xup_lib directory. Add the my_and2 to the IP catalog.

1-1-1. Open Vivado by selecting **Start > All Programs > Xilinx Design Tools > Vivado 2013.3 > Vivado 2013.3**

1-1-2. Click the *Create New Project* link.

1-1-3. Set the directory path to *c:\xup\digital* and the project name as **my_and2_test**.

1-1-4. Click **Next** and make sure that the *RTL Project* type is selected.

1-1-5. Click **Next** and make sure that **Verilog** is selected as the *Target language* and *Simulation language*.

1-1-6. Click **Next** three times until *Default Part* form is displayed.

1-1-7. Using the appropriate filters, select *xc7a100tcsg324-1* part, then click **Next**, and then **Finish**.

1-1-8. Click **Project Settings** in the *Flow Navigator* pane. If it is not available then select **Tools > Project Settings**.

1-1-9. Select **IP** in the left pane of the *Project Settings* form.

1-1-10. Click on the **Add Repository…** button, browse to **c:\xup\digita\xup_lib** and click **Select**.

1-1-11. Click the **Add IP…** button, and select **xup_my_and2_1.0** entry, and click **OK**.

The *my_led_ip_v1_0* IP entry will appear in the **Selected Repository** window.

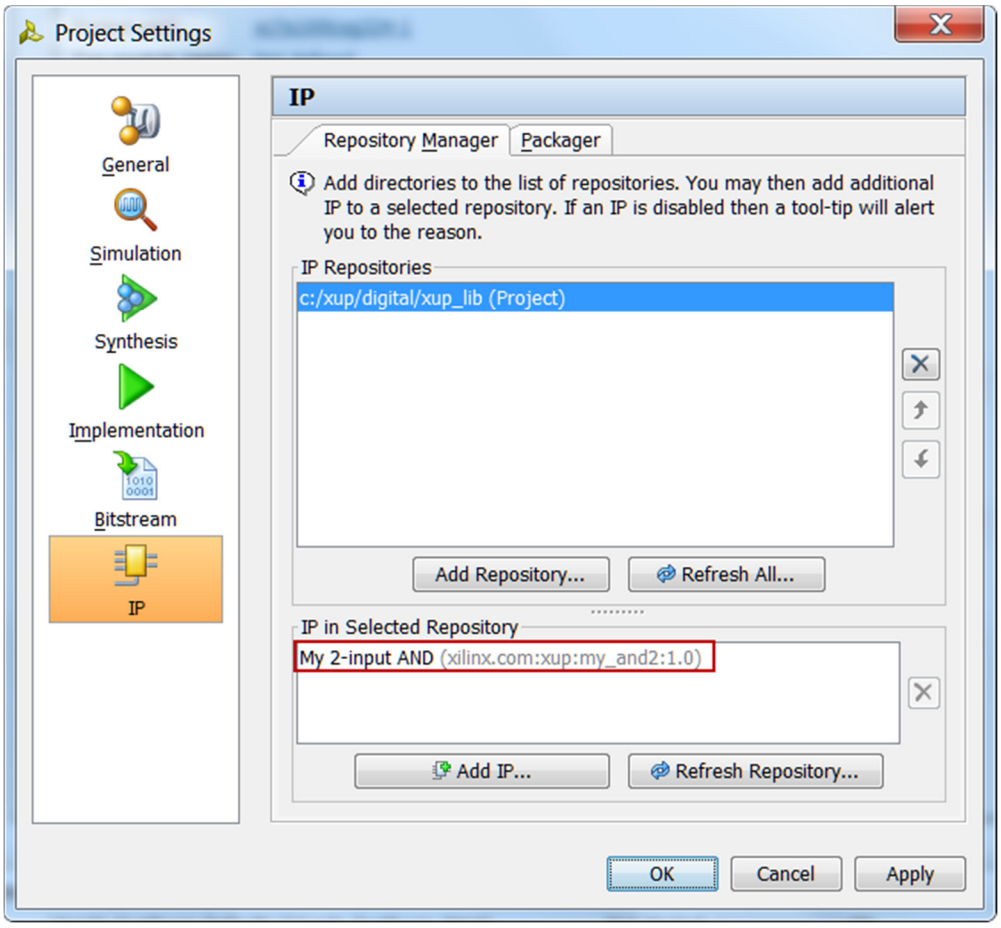

Figure 1. Specify IP Repository

1-1-12. Click **OK**.

1-2. Create the block design instantiating the my_and2 block.

1-2-1. Click on the **Create Block Design** in the *Flow Navigator* pane.

1-2-2. Set the design name to **system** and click **OK**.

1-2-3. IP from the catalog can be added in different ways. Click on *Add IP* in the message at the top of the *Diagram* panel.

1-2-4. Once the IP Catalog is open, type “my” into the Search bar, find and double click on **My 2-input AND** entry, or click on the entry and hit the Enter key to add it to the design.

1-2-5. Double-click on the *my_and2_0* instance to open the configuration form.

1-2-6. Change the *delay* value to **5** and click **OK**.

1-2-7. Right-click on the *y* port and select **Make External**.

1-2-8. Similarly, select the *ain* and *bin* ports and make them external.

1-2-9. The block diagram should look like below.

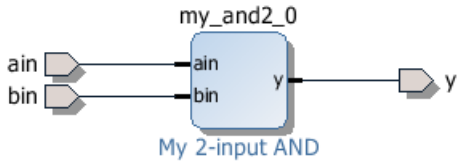

Figure 2. The block design

1-2-10. Click **Tools > Validate design** and correct any errors if necessary

1-2-11. Right Click on *system.bd* in the **Sources** tab and select **Create HDL Wrapper**.

1-2-12. Select **Let Vivado Manage wrapper and auto-update** option and click **OK** when prompted.

1-3. Look at the source file (model of the IP).

1-3-1. Select **IP Sources** tab and expand the hierarchy.

Observe the my_and2.v entry under the synthesis and simulation categories. In this simple example, it is the same. In complex design, they may be different depending on how the IP was created.

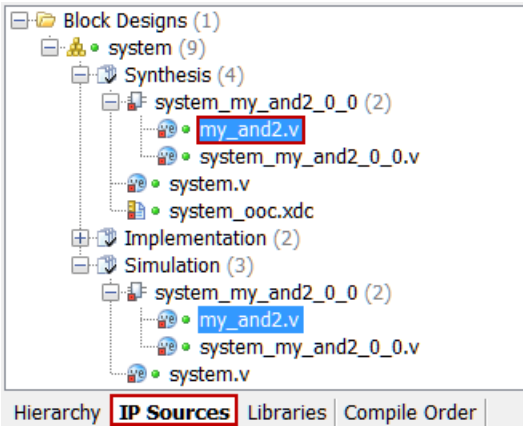

Figure 3. The IP hierarchy

1-3-2. Double-click the *my_and2.v* entry and look at the model used to create the IP and the functionality it is providing.

```
1 `timescale 1ns / 1ps
2 //////////////////////////////////////
3 // Module Name: my_and2
4 //////////////////////////////////////
5 module my_and2 #(parameter delay=3)(
6     input ain,
7     input bin,
8     output y
9     );
10
11    and #delay (y, ain, bin);
12
13 endmodule
```

Figure 4. The IP model

Simulate the Design Step 2

2-1. Set simulation time to 100 ns.

2-1-1. Select **Simulation Settings** > **preferences…**

2-1-2. Select **Simulation** in the left pane.

2-1-3. In the **Simulation** tab, set the simulation time to **100** ns and click **OK** to close the window.

2-2. Add provided my_and2_tb.v testbench, simulate and examine the output

2-2-1. Click *Add Sources.*

2-2-2. Select *Add or Create Simulation Sources* and click **Next.**

2-2-3. Click *Add Files*, and browse to the *c:/xup/digital* directory and select **my_and2_tb.v** and click **Finish.**

2-2-4. Expand the hierarchy and notice how the design is setup for the simulation.

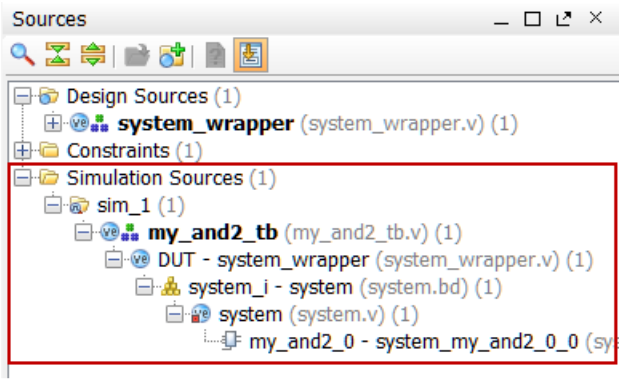

Figure 5. Hierarchy for simulation

2-3. Launch the simulator which will automatically elaborate the model source file, load the simulation model, and run the simulation.

2-3-1. In Vivado, select **Run Simulation** > **Run Behavioral Simulation** to launch the simulator.

When done, the waveform window will show up.

2-3-2. Click on the zoom full button ( ) to see the entire simulation waveform. It should look similar to the one shown below.

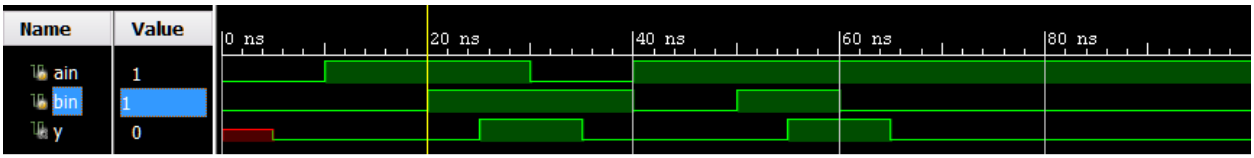

Figure 6. Full Simulation Output

2-3-3. Click at 20 ns to show a marker.

2-3-4. Add another marker by clicking ( ).

A blue ribbon will appear.

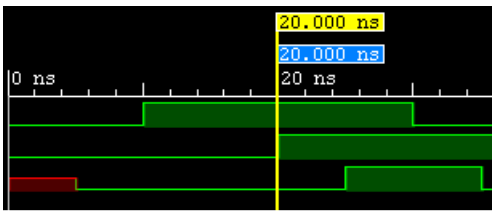

Figure 7. Another marker added

2-3-5. Drag the blue ribbon to the where y makes transition from 0 to 1. Verify that the time difference is 5 ns (you may have to zoom in for accurate measurement).

2-3-6. Close the simulation by selecting **File > Close Simulation** without saving the changes to the waveform.

2-4. Change the delay in the block diagram to 3, create the hdl wrapper, simulate the design and verify that the delay has changed to 3.

2-4-1. Select *Open Block Design* in the *Flow Navigator* pane.

2-4-2. In the *Diagram*, tab, double-click the *my_and2_0* instance to open the configuration form.

2-4-3. Change the *delay* value to **3** and click **OK**.

2-4-4. Right Click on *system.bd* (expand the hierarchy if necessary) in the **Sources** tab and select **Create HDL Wrapper**.

2-4-5. Select **Let Vivado Manage wrapper and auto-update** option and click OK when prompted

2-4-6. In Vivado, select **Run Simulation** > **Run Behavioral Simulation** to launch the simulator.

2-4-7. Verify that the delay has changed to 3.

Verify the Design in Hardware Step 3

3-1. Add the provided design constraint file.

3-1-1. Click *Add Sources.*

3-1-2. Select *Add or Create Constraints* and click **Next.**

3-1-3. Click *Add Files*, and browse to the *c:/xup/digital* directory and select **my_and2.xdc** and click **Finish.**

3-2. Connect the board and power it ON. Generate the bitstream, open a hardware session, program the FPGA and verify the functionality.

3-2-1. Make sure that the power supply source is jumper to *USB* and the provided Micro-USB cable is connected between the board and the PC. Note that you do not need to connect the power jack and the board can be powered and configured via USB alone

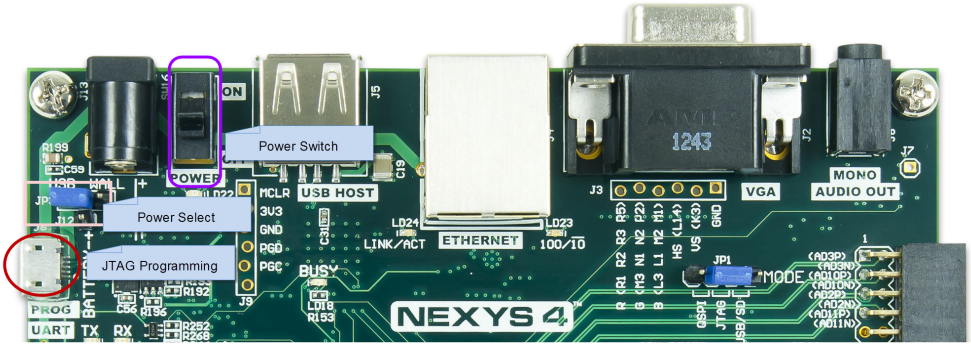

Figure 8. Board settings

3-2-2. Power **ON** the switch on the board.

3-2-3. Click the *Generate Bitstream* in the *Flow Navigator* pane.

3-2-4. Click **Yes** to run the necessary processes and wait for the *Bitstream Generation Completed* form to appear.

3-2-5. Click on the **Open Hardware Manager** option and select **OK.**

You can also click on the Open Recent Hardware Target link if the board was already targeted before.

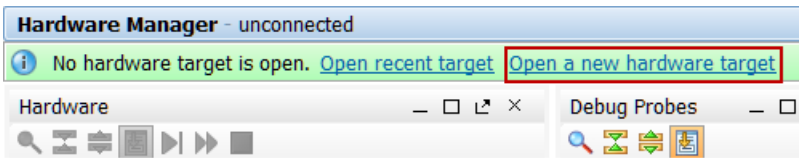

Figure 9. Opening new hardware target

3-2-6. Click **Next** to see the Vivado CSE Server Name form.

3-2-7. Click **Next** with the localhost port selected.

The JTAG cable will be searched and the Xilinx_tcf should be detected and identified as e hardware target. It will also show the hardware devices detected in the chain.

3-2-8. Click **Next** twice and **Finish**.

The Hardware Session status changes from Unconnected to the server name and the device is highlighted. Also notice that the Status indicates that it is not programmed.

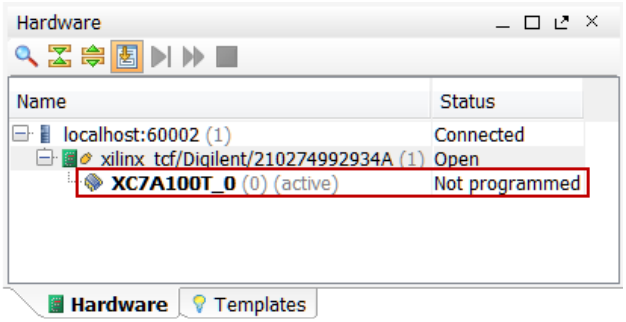

Figure 10. Opened hardware session

3-2-9. **Right-click** on the device and select *Program Device…* to program the target FPGA device. Click OK to program the board with the system_wrapper.bit file.

The DONE light will lit when the device is programmed. You may see LED[0] lit depending on the switches position.

3-2-10. Verify the functionality by flipping switches and observing the output on the LEDs.

3-2-11. Close the hardware session by selecting **File > Close Hardware Manager.**

3-2-12. Click **OK** to close the session.

3-2-13. Power **OFF** the board.

3-2-14. Close the **Vivado** program by selecting **File > Exit** and click **OK**.

Conclusion

This lab guided you through using the custom IP, provided by XUP, in the IP Integrator, simulating it, and verifying the functionality in the hardware.