### Acknowledgements

Developed by Craig Kief, Alonzo Vera, Alexandria Haddad, and Quinlan Cao, at the Configurable Space Microsystems Innovations & Applications Center (COSMIAC). Based on original tutorial developed by Bassam Matar, Engineering Faculty at Chandler-Gilbert Community College, Chandler, Arizona. *Funded by the National Science Foundation (NSF).*

### Lab Summary

This is a voluntary lab for those of you who lack experience in the use of Xilinx ISE®. We will walk you through a small version of a lab session and produce a fictitious lab report. In the process you will use all of the main steps that you will use in later lab work.

This lab teaches design entry, simulation, and prototyping using tools provided by Xilinx ISE® for this purpose. We will show how a simple design circuit of a **2-**input AND gate can be directly entered into Xilinx ISE® for synthesis, post synthesis simulation, and timing analysis. We will show the implementation of more complex designs in future labs by running them through the design flow illustrated in this lab.

### Lab Goal

The goal of this lab is to learn how to use the Xilinx ISE® software by implementing inputs and outputs for a simple 2-input AND gate.

### Learning Objectives

1. Use the Xilinx ISE® Schematic Editor to create a 2-input AND gate project using the free ISE® WebPACK™.
2. Compile and simulate the 2-input AND design.
3. Compile and simulate the 4-input XOR design.

### Grading Criteria

Your grade is determined by your instructor.

### Time Required

2-3 hours

### Lab Preparation

* Read this document completely before you start on this experiment.
* Print out the laboratory experiment procedure that follows.

### Equipment and Materials

Access to Xilinx software

|  |  |
| --- | --- |
| Software needed | Quantity |
| Download the ISE® WebPACK™ software from the Xilinx website, www.xilinx.com. | 1 |

### Additional References

Current Xilinx ISE Software manuals found on Xilinx web site: [www.xilinx.com](http://www.xilinx.com).

# Lab Procedure 1: Download and Install ISE® WebPACK™

In this lab, you will first load and license the current version of the free ISE® WebPACK™. Next, you will create a schematic of a small circuit of limited functionality. If you already have a copy of Xilinx ISE® or the WebPACK™ you may skip these steps.

You might be using a different version of ISE than the one used in this lab. However, there should be minimal changes in the instructions.

Installing the ISE® WebPACK™ Software

Step 1:

Register with Xilinx at [www.xilinx.com](http://www.xilinx.com). Navigate to the <http://www.xilinx.com/products/design-tools/ise-design-suite/ise-webpack.htm> page and download the free ISE® WebPACK™.

Step 2:

Install ISE® WebPACK™.

System Requirements:

Microsoft Windows XP Professional7, Vista or Linux.

Step 3:

Have Fun!

# Lab Procedure 2: Design a 2-input AND Circuit

|  |  |
| --- | --- |
|  | Xilinx Design Process overview:  Step 1: Design Entry   * Two design methods:  1. HDL (Verilog or VHDL)  or 2. Schematic drawings.   *For the simulation part of our class, we will use the schematic method and VHDL.*  Step 2: Design Synthesis   * Translate VHDL and schematic files into an industry standard format EDIF file.   Step 3: Design Implementation   * Translate Map, Place and Route. This process will generate a configuration file (.BIT) for FPGA programming.   Step 4: Xilinx Device Programming   * Download JED file into FPGA |

In this portion of the lab, you will use Xilinx ISE software to build a simple 2-input AND circuit. This lab focuses on Steps 1 and 2 of the Xilinx Design Process. We will implement Steps 3 and 4 in upcoming labs.

NOTE: At the conclusion of the lab, you will be asked to comment on why certain steps are required. Be sure to take notes on these questions as they appear in the lab procedure.

## Step 1: Design Entry

|  |  |
| --- | --- |
| 1. Open Xilinx ISE Design Tool Project Navigator.   Your system might have slightly different Start Menu options. |  |
| 1. The ISE Project Navigator window opens, with the Tip of the Day displayed. Click OK to close the Tip of the Day. |  |
| 1. Start a new project by selecting File 🡪 New Project from the menu. The New Project Wizard starts. 2. Type AND\_2 in the Name text box. 3. Select a location on your computer to save your project files by clicking the ellipsis (…) button to the right of the Location text box. 4. Under Top-level source type, select HDL. 5. Click Next.   The Project Settings dialog box opens. |  |
| NOTE: File names must start with a letter. Use underscores ( \_ ) for readability. Do not use hyphens (-); although the file name will work, the entity name will not. More on this later. | |
| 1. Select Spartan-6 SP601 Evaluation Platform from the Evaluation Development Board drop down menu.   The Product Category, Family, Device, Package, and Speed should all automatically populate (top half of the screen). You will need to set some options in the lower half of the dialog box. 2. Select VHDL from the Preferred Language drop down menu.   The Project Settings should resemble the figure to the right. 3. Click Next. |  |
| NOTE: The options specified are for the Spartan 6 LX FPGA. Your board might be different than the board used for these instructions. Board specifications are printed on the FPGA chip in the middle of the board as shown in the figure to the right. The board information is also listed on the box it came in. |  |
| NOTE: You can also leave the Evaluation Development Board drop down set to None Specified, and change each of the options manually.  NOTE: Press F1 or select Help 🡪 Help Topics from the menu to access context sensitive Help about any ISE option.  The Help menu has many other support options available also, e.g., Xilinx on the Web Tutorials. Selecting this option opens the Xilinx support page on a separate workspace tab inside the ISE window. |  |
| 1. The Project Summary displays. Verify the file type and name are correct then click Finish to complete the New Project creation process. |  |
| 1. The Project Wizard window closes and a new project is created. The Project Navigator window provides all the options you need to create and/or include project files.   Take a minute to familiarize yourself with the Project Navigator window Menu, Toolbar, Panels, and Work Space area. |  |
| 1. We will be building a simple 2-input AND gate as shown on the right. First we will use the schematic option, and then we will create a VHDL code file. This provides an introduction to the VHDL code and syntax. |  |
| 1. Add a New Source Schematic to your project by selecting Project🡪New source from the menu. The New Source Wizard dialog box opens. 2. Select Schematic from the list of sources and type AND\_2 for the file name and click the Next button. The Source Summary dialog box opens. Click Finish.   NOTE: AND-2 is just a name for your schematic. You can name the schematic file any name you desire. |  |
| 1. The new AND\_2.sch schematic is displayed in the workspace. You might also see a workspace tab for the Design Summary.   The left panel display area now shows the Options panel and additional panel tabs for Options and Symbols. Note there is also a left pointing arrow to navigate the panel tabs. Also notice the schematic toolbar on the left of the workspace. |  |
| 1. Click the Symbols panel tab to display the Symbols panel. 2. Under Categories select Logic. 3. Under Symbols select and2, and then click the mouse on the schematic workspace to place the and2 symbol. |  |
| Using the symbols tools  All the symbols tools are either on or off. Selecting a tool, either from the symbols panel, menu, or the toolbar turns the tool on. The mouse pointer will reflect the symbol that is currently on. Clicking on the workspace will place the current symbol at that location. The tool remains on, and multiple symbol placements can be made, until the tool is turned off by pressing the ESC key. When a tool is active the options panel provides context sensitive options for that tool. | |
| 1. Zoom in or out, using the Zoom buttons on the toolbar, to resize your view. 2. Add wires to the AND gate to enable future connection of pins and gates to the schematic.   Click the Add Wires button on the toolbar to turn on the Add Wires tool.   Add a wire to each input and the output of the AND gate by Clicking once on the end of the AND gate to start the wire then double-clicking to end the wire. Repeat for each wire.  Press the esc key to turn off the Add Wires tool.   NOTE: For this simple schematic it would not normally be necessary to add wires. This step is for practice purposes only. |  |
| Adding I/O Markers  An I/O marker is an input, output, or bidirectional signal. This establishes net polarity (direction of signal flow) and shows that the net is externally accessible. Without pins, the device is meaningless. All primary inputs and outputs must be marked with I/O markers. You can add an I/O Marker from the menu or the toolbar:   * Select Add 🡪 I/O Marker from the menu. * Click the Add I/O Marker toolbar button tools-iomarker. | |
| 1. Click the Add I/O Markers button on the toolbar to turn on the Add I/O Markers tool.  Add an I/O marker by placing the mouse pointer over the end of a wire and clicking once. Click on each wire end to add an I/O marker.   Press the esc key to turn off the Add I/O Markers tool. |  |
| 1. Change the name of the I/O marker by right-clicking on the marker and selecting Rename Port from the shortcut menu.   Type the new name for the port and click OK. |  |
| 1. Once the schematic is complete, test for wiring errors by selecting Tools 🡪 Check Schematic from the menu. |  |
| 1. The Console Panel will display any errors or warnings. If you have errors, fix them and run Check Schematic again. 2. Save your file. |  |

## Step 2: Design Simulation

|  |  |
| --- | --- |
| 1. Create a test bench for your file. Right-click in the Hierarchy Pane and select New Source from the short cut menu. 2. Select VHDL Test Bench as the source type. 3. Enter AND\_2\_TB as the file name.   NOTE: It is good practice to name your test bench file with the same name as the original file and append \_TB to the end of the file name. |  |
| 1. Click the Next button. The Associate Source dialog box displays with AND\_2 selected. Click the Next Button.   NOTE: When you have a project with multiple .vhd files, every file is listed in the Associate Source dialog box. Before you click Next, make sure you have the correct file selected for the test bench you are creating.   1. The New Source Summary dialog box displays. 2. Click the Finish button. |  |
| 1. The test bench file will display in the workspace. Make the changes highlighted on the right.   NOTE: The PROCESS section of the simulation sets the values of A and B from 0 to 1 and then waits for a small period of time (25 and 50 nanoseconds, respectively).   1. Click the Save All button on the toolbar. | -- initialize inputs to 0  SIGNAL A : STD\_LOGIC:='0';  SIGNAL B : STD\_LOGIC:='0';  SIGNAL OUTPUT : STD\_LOGIC;  -- \*\*\* Test Bench - User Defined Section \*\*\*  ~~tb : PROCESS~~  ~~BEGIN~~  ~~WAIT; -- will wait forever~~  ~~END PROCESS;~~  -- offset timing of inputs simulate all input options.  -- output = 1 only when A = B = 1  A\_process : PROCESS  BEGIN  A <= NOT A;  WAIT FOR 25 ns;  END PROCESS;  B\_process: PROCESS  BEGIN  B <= NOT B;  WAIT FOR 50 ns;  END PROCESS;  -- \*\*\* End Test Bench - User Defined Section \*\*\* |
| 1. On the View Pane, select Simulation. The Simulation Hierarchy and Process options display. 2. Expand the ISim Simulator tool if it isn’t already expanded. 3. Double-click the Simulate Behavioral Model tool. Click Yes to save any changes. |  |
| 1. The Simulation Elaboration will begin to run. When it is finished the ISim window will open displaying the simulation. |  |
| 1. Use the Zoom to Full View and the Zoom In buttons to display the simulation waveform clearer.   Review the waveform to ensure the testbench is working as designed.   The waveform should match the following logic table for this circuit, A•B=C.  |  |  |  |  | | --- | --- | --- | --- | | A | B | = | C | | 0 | 0 |  | 0 | | 0 | 1 |  | 0 | | 1 | 0 |  | 0 | | 1 | 1 |  | 1 |   When you are finished, close the ISim application and return to ISE. |  |

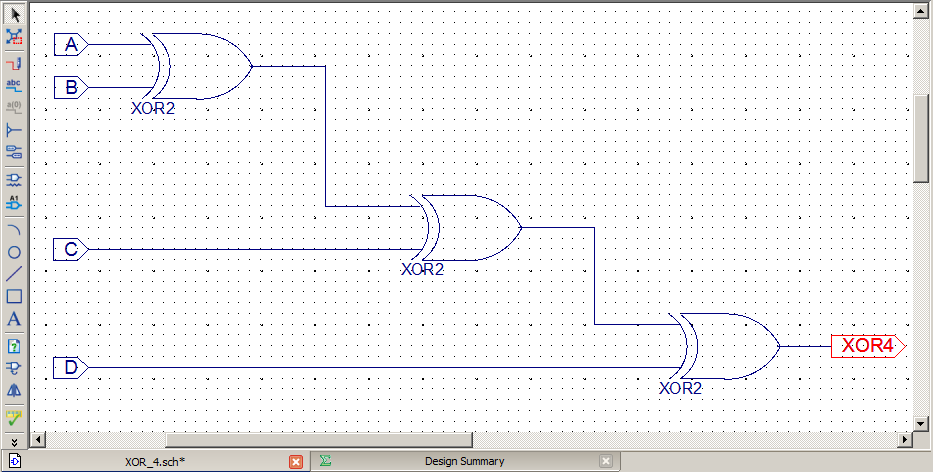
# Lab Procedure 3: Create VHDL file for 2-input AND circuit

|  |  |  |
| --- | --- | --- |
| 1. Click the panel tab’s left triangle button to display the Design tab. |  | |
| 1. Right-click in the Design Panel and Click New Source… from the shortcut menu. The New Source Wizard displays. 2. Select VHDL Module from the source type list. 3. Type AND\_2 as the file name. 4. Click the Next button. The Define Module dialog displays. |  |  |
| 1. Type the port names and select the direction of each port as shown in the figure to the right. 2. Click the Next button. The Summary dialog displays. 3. Click the Finish button to accept the New Source Wizards options. |  | |
| 1. ISE will generate source code for you and it will display in the workspace area. |  | |
| Basic VHDL code has three sections: 1) library, 2) entity, and 3) architecture. The library section holds all the libraries and packages needed for the code. All the code’s inputs and outputs are declared in the entity section. The actual circuit’s instruction code is in the architecture section. | | |
| 1. Scroll down to the architecture section of the code and modify the code as highlighted:     begin  OUTPUT <= A and B;  end Behavioral;   1. Save your files by clicking on the Save All button. |  | |

# Lab Procedure 4: Test your Knowledge

The more you practice, the more you will be proficient with the software.

Build the following circuit (4-input XOR) using three 2-input XOR gates.



Create a test bench. Make the following changes as highlighted, and then run the test bench.

SIGNAL D : STD\_LOGIC:='0';

SIGNAL C : STD\_LOGIC:='0';

SIGNAL A : STD\_LOGIC:='0';

SIGNAL B : STD\_LOGIC:='0';

SIGNAL XOR4 : STD\_LOGIC;

-- \*\*\* Test Bench - User Defined Section \*\*\*

~~tb : PROCESS~~

~~BEGIN~~

~~WAIT; -- will wait forever~~

~~END PROCESS;~~

A\_process : PROCESS

BEGIN

A <= NOT A;

WAIT FOR 25 ns;

END PROCESS;

B\_process: PROCESS

BEGIN

B <= NOT B;

WAIT FOR 50 ns;

END PROCESS;

C\_process: PROCESS

BEGIN

C <= NOT C;

WAIT FOR 75 ns;

END PROCESS;

D\_process: PROCESS

BEGIN

D <= NOT D;

WAIT FOR 100 ns;

END PROCESS;

-- \*\*\* End Test Bench - User Defined Section \*\*\*

END;