

ECE480 Application Note on Xilinx ISE

Xilinx ISE is a software tool created by Xilinx for synthesis and analysis of HDL(Hardware Description Language) or VHDL(Virtual Hardware Description Language). The reason why it could be either the two is because users can choose to program and test the program virtually using test benches(simulated files used to imitate real testing devices) and simulators or create a programming files and load them to real devices. Unfortunately, Xilinx ISE is a design environment for FPGA(Field Programmable Gate Array) products from Xilinx only and not from other vendors which makes it not a very suitable program to be used in general projects.

To start with the tutorial for using Xilinx ISE, download the Xilinx ISE WebPack from www.xilinx.com and install it to your computer. I am assuming every engineer users know how to install this program and I would like to go directly to how to start and use the program.

Launching Xilinx Project Navigator

Select Start > All Programs > Xilinx ISE WebPack > ISE Design Suite > ISE Design Tools > Project Navigator

This is the main program where you create VHDL projects, simulate/debug codes, and generate programming files that can be downloaded to the microcontroller embedded devices to test your program.

Create a New Project

1. Select File > New Project... The New Project Wizard should appear.
2. Type tutorial in the Project Name field.
3. Enter or browse to a directory for this project.
4. Verify that HDL is selected from the Top-Level Source Type list.
5. Click Next to move to the device properties page.
6. Fill in the properties show in Figure1.

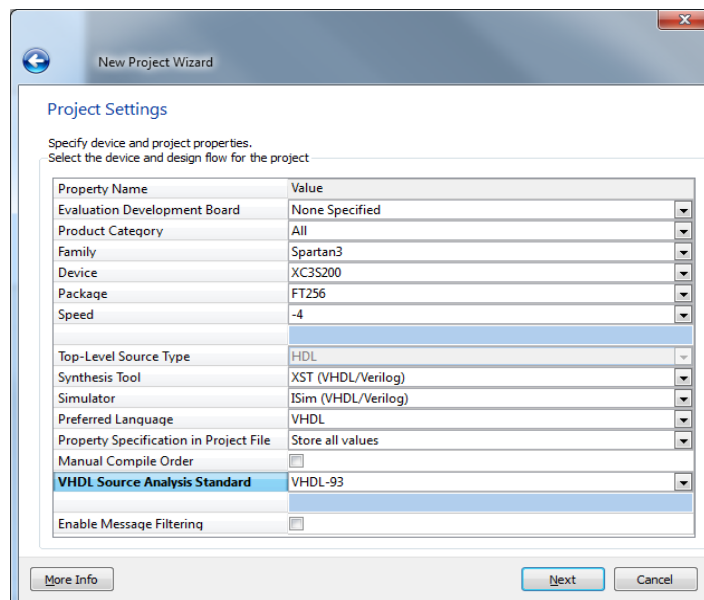


Figure 1.

7. Click Next to proceed to the Create New Source window. At the end of the next section, new project will be complete.

Creating a VHDL Source

1. Click the New Source button in the New Project Wizard.
2. Select VHDL Module as the source type.
3. Type in the file name fulladder since we are going to make a fulladder as a tutorial.
4. Click Next.
5. Now we can initialize the ports in the dialogue box and determine their directions which mean that whether they are inputs, outputs, or both. If you check mark Bus, this means that the port is not only a single bit value and you can specify the bits to MSB(Most Significant Bit) and LSB(Least Significant Bit). See Figure 2.

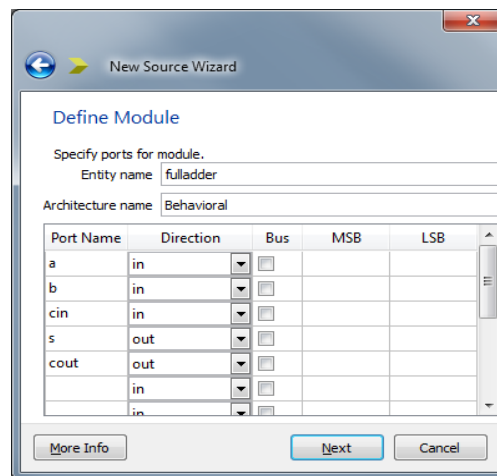


Figure 2.

6. Click Next, then Finish. Click Next, then Next, then Finish.

Input VHDL Code

1. Make sure that the fulladder.vhd tab is selected. You should see the code that Xilinx has already generated.
2. Since Xilinx already generated the port declaration statement, all you have to do is fill in the body.
3. Put this code between begin and end under architecture of the behavioral

```
s <= a xor b xor cin;
```

```
cout <= (a and b) or (a and cin) or (b and cin);
```

4. Check Syntax of your VHDL code and verify if it is valid. You should be able to find Check Syntax under the Synthesize-XST header in Processes window.
5. The window at the bottom should not report any error. If so, check you codes again.

6. Save the file.

Simulating the Full Adder Code Using Force Commands

1. Simulate the fulladder using ISim. To do this, select View window, select Simulation and verify that Behavioral is selected in the drop down box just below the Simulation radio button. Click on fulladder – Behavioral(fulladder.vhd) file icon under the xc3s200-4ft256 header in the View window. Then in the Processes window, double-click on Simulate Behavioral Model under the ISim Simulator header
2. In the ISim window, we will input 'force' commands to force our inputs to 0's and 1's. We can then verify the correct output operation in the wave window.
3. In the Console window, type the following series of force commands.

```
restart
isim force add /fulladder/a 0
isim force add /fulladder/b 0
isim force add /fulladder/cin 0
run 50ns
isim force add /fulladder/b 1
run 50ns
isim force add /fulladder/a 1
run 50ns
isim force add /fulladder/b 0
run 50ns
isim force add /fulladder/cin 1
run 50ns
isim force add /fulladder/b 1
run 50ns
isim force add /fulladder/a 0
run 50ns
isim force add /fulladder/b 0
run 50ns
```

4. Select Zoom > To Full View or press F6 to see the entire waveform.
5. All lines should be green. Red lines indicate undefined values.
6. Check whether the waveforms produce the correct values you should expect.

Generating *.bit File(Programmable File)

1. Make sure Implementation is selected in the View window. In the Processes window, right-click Generate Programming File, then select Process Properties.
2. In the dialog box, select the Startup Options tab, then select JTAG clock for FPGA Start-up Clock.
3. Select OK.
4. Once we have specified that we're using the JTAG clock, we need to assign switches to the input pins and LEDs to the output pins. Notice that near each switch, pushbutton, and LED on the

Spartan-3 board there is a letter/number in parenthesis. This (i.e., P11 for LD7) is the pin name. Let's assign the following inputs and outputs for our full adder:

Signal	Pin
a	K13
b	K14
C _{in}	J13
S	P12
C _{out}	P11

5. In order to do this, double-click I/O Pin Planning (PlanAhead) - Pre-Synthesis under User Constraints in the Processes for window.
6. Select Yes if asked if you would like Xilinx to create your UCF file.
7. PlanAhead window has by default four sub windows. In the bottom window, verify that the I/O ports tab is the active one.
8. In this tab, the I/O pins should appear by expanding the header nodes All ports and Scalar ports.
9. Specify the pins (in the Site column) as indicated in the table above.
10. Now select File->Save Constraints.
11. Close the PlanAhead window.
12. Back to the Project Navigator window, make sure that fulladder.vhd is selected in the View in Project window.
13. In the Processes window, double-click the Generate Programming File header. This will generate your *.bit file.
14. This step may take a few minutes, for the computer is mapping out your circuit. Once the *.bit file is generated, A message may appear warning that no iMPACT project file exists... etc, click OK to proceed to iMPACT.
15. In the iMPACT Flows upper left window double-click on the Boundary Scan icon. This will show a white area in the bigger window at the right. Right-click in that window and click Initialize Chains.
16. This step should show two chip icons that represent the FPGA and the EPROM modules.

Downloading and Running the Diagnostic Program(*.bit file)

1. Launch iMPACT by selecting Start > All Programs > Xilinx ISE Webpack > ISE Design Suite > ISE Design Tools > iMPACT.
2. Click OK in the iMPACT Access Permission message.
3. Notice that iMPACT detects 2 devices. The first one is xc3s200 which is the FPGA. The second one is the EPROM module. We will only be downloading to the FPGA only, but not EPROM for this tutorial.
4. Right Click and select Initialize Chain. A window for the first FPGA programming file assignment should pop up.

5. Select the Programming file you generated above, then select Open.
6. Bypass the second window prompt.
7. When done, you should come back to see the 2 devices again. Click on the first device, and select Program.
8. Done! Now you should be able to play around with your fulladder on the board with your selected keys set.