Introduction to Discrete Digital Logic and Programmable Logic

Introduction

This lab will be and introduction to design techniques using VHDL. ISE is the tool provided by Xilinx to this purpose. Through the lab you will became familiar with basics operation using ISE.

VHDL has three styles of implementing designs -- data flow, structural or behavioral. Behavioral was chosen for this particular lab. There are actually **two** sections of code for this lab. The first is the behavioral description of what the project is to do. The second set of code is the test bench. As its name implies, it is used to test the project. It is used to drive the inputs so that you can verify that your design is working correctly by observing the outputs. The complete behavioral description is given for this lab. However the test bench is not complete, it will only test some of the possible inputs. You will have to add a few lines of code in order to fully test the project. Figure 1 shows the schematic to be implemented.

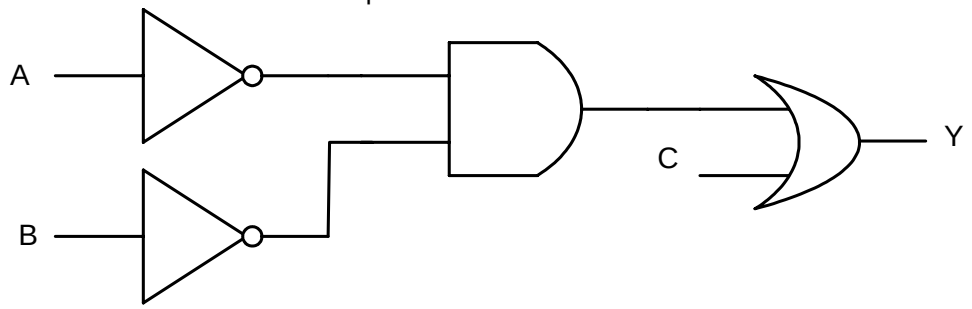


Figure 1. Schematic of the circuit to be implemented

Goals

After completing this lab, you will be able to:

- Create a project using ISE tools
- Identify main parts of a VHDL code
- Write test benches and simulate a simple circuit.

Start Project Navigator and Create the Project Step 1

Almost all the code for this project is provided. On the computer create a new directory for your project with the following format: ***C:\ece238\fall2005\student_name\lab1***. ISE tools does not allow for spaces in the directory name, so you will not be able to save your project on the Desktop or in My Documents.

1. Acquire files for the project; Download the files ***fewgates.vhd*** and ***fewgates_tb.vhd*** provided at the lab website and save them in the directory

C:\ece238\fall05\student_name\lab02. Make sure you are saving them without a ".txt" extension but a ".vhd" extension.

2. Start the Xilinx Project Navigator; Go to ***Start Menu*** → ***Programs*** → ***Xilinx ISE 7.1*** →***Project Navigator.*** (or you can also look for the ISE icon on your desktop)
3. Create a new project; In the Project Navigator, select ***File*** → ***New Project***. This will bring up a window like the one shown in figure 2. Setup ***fewgates*** as the name for your project and then click on the ***...*** button to browse to your directory. Choose ***HDL*** for ***Top-Level Module Type***; because we will be implementing the project in VHDL, which is a Hardware Descriptive Language (HDL). Once the correct options have been chosen, click ***Next***.

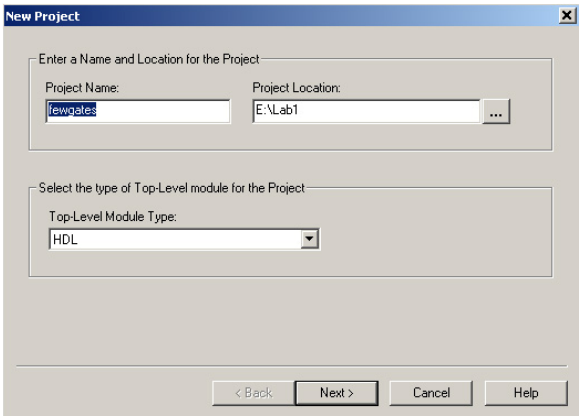


Figure 2. New Project options

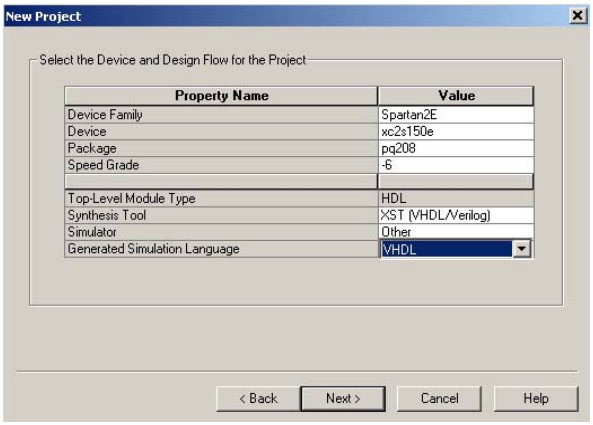


Figure 3. Device Options

4. Setting a device; The next window allows us to choose the type of device we will be using for the project. However, we are only going to simulate this project not download it to a device. The device type does not really matter this time. We just need to pick a device that is big enough to store the project. Choose the settings like the ones shown in figure 3, click ***Next***.
5. Adding or creating new source files; The next window allows you to create a new source. We need to add existing sources to the project so click ***Next***. Now we can add our existing sources to the project. Click ***Add Source***. Browse to your directory and choose ***fewgates.vhd*** and click ***Open***. In the next window, make sure ***VHDL Design File*** is highlighted (as shown in figure 4), click ***OK***.

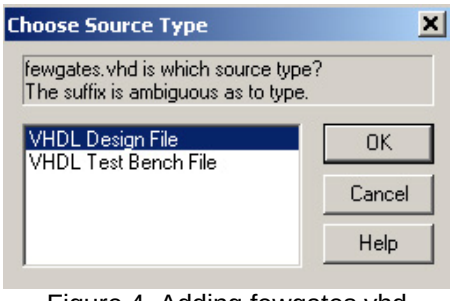


Figure 4. Adding fewgates.vhd

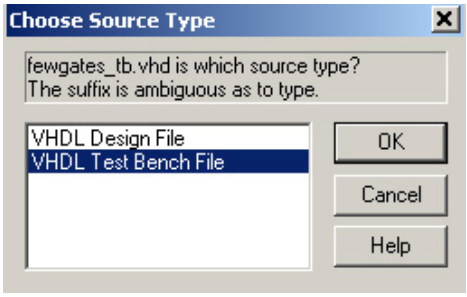


Figure 5. Adding fewgates_tb.vhd

6. Now add the ***fewgates_tb.vhd*** file to your project, but make sure ***VHDL Test Bench File*** is selected as the source type as shown in figure 5. Now the final window should look like the one shown in figure 6. When you are done adding your files click ***Next***.

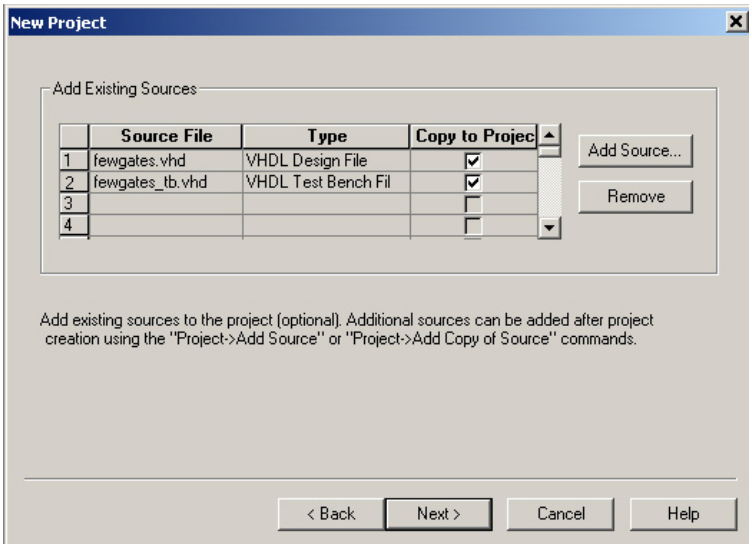


Figure 6. Files added to the project

The next window is a summary of all of the settings of you project. Verify that they are correct and click ***Finish***. Project Navigator is now showing your newly created project. Spend some time analyzing the code. Determine what it is actually doing.

Analyze and Compile the Code — Step 2

Look for the ***Sources in Project*** box in the upper left corner of the window. Notice how the test bench code falls below the main code, ***fewgates.vhd***, in a hierarchical level as shown in figure 7. This is based on the concept that the test bench would serve no purpose without the main code. The test bench serves only to test the main code.

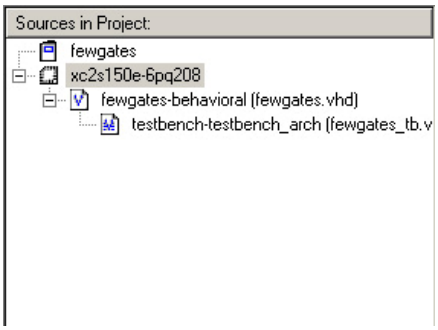


Figure 7. Hierarchical level for files

Double click on ***fewgates-behavioral (fewgates.vhd)***. This will open the file so that it can be edited; note the way the software color-codes the code (green for comments, blue for keyword/reserved words, etc. as shown in figure 8). This feature can be used as an error detection technique. It can sometimes help find code problems. Close the ***fewgates.vhd*** code.

```
1  --*********************************************
2  -- File: fewgates.vhd
3  --
4  -- Purpose: This file is used to introduce students to
5  --using VHDL. It uses three inputs and one output.
6  --
7  -- Created:5/1/02 CK
8  -- Modified:5/15/02 CK
9  --Fixed comments
10 --
11 --*********************************************
12 --Defining the library packages to be used
13 library IEEE;
14 use IEEE.STD_LOGIC_1164.ALL;
15 use IEEE.STD_LOGIC_ARITH.ALL;
16 use IEEE.STD_LOGIC_UNSIGNED.ALL;
17
18 --Declaration of the module's inputs and outputs
19 --The modules name is "fewgates"
20 entity FEWGATES is port (
21 a: in std_logic;
22 b: in std_logic;
23 c: in std_logic;
24 y: out std_logic
25 );
26 end FEWGATES;
```

Figure 8. Comments and color-codes in a VHDL file.

1. Synthesize the project; Highlight the ***fewgates.vhd*** code (in the ***Sources in Project*** box) and then expand the ***Synthesize*** section within the ***Processes for Source*** box and double click ***Check Syntax***. The syntax check should indicate no errors. To see how it looks with errors, change some letters in the ***fewgates.vhd*** code and do the process again. A green check mark is conformation that the syntax check was successful as shown in figure 9.

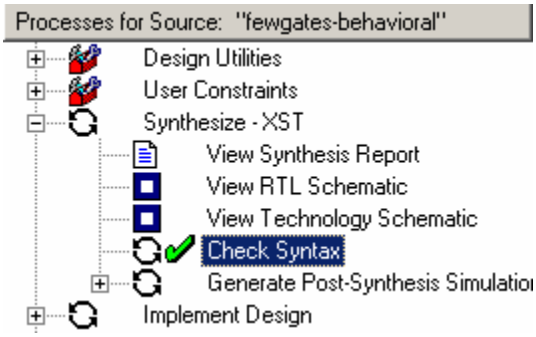


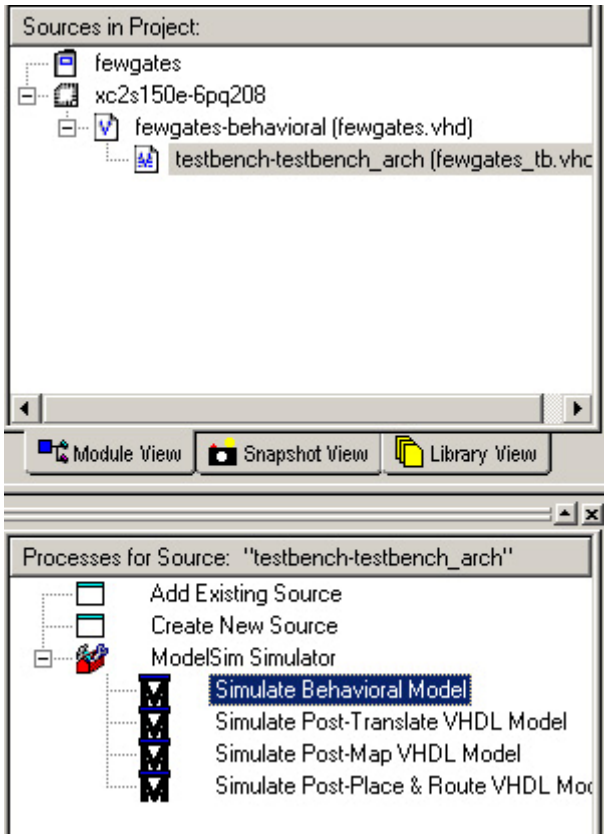
Figure 9. Green check mark showing the process was successful.

Simulate the system — Step 3

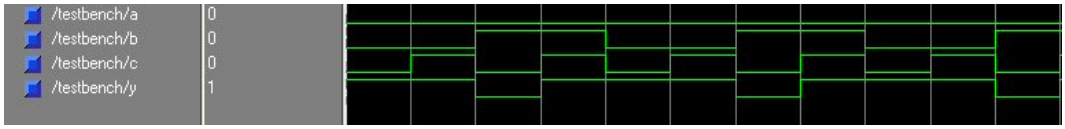
The next step is to simulate this design. Simulation gives the ability to ensure a project does what is expected

1. Run behavioral simulation; Highlight the file ***fewgates_tb.vhd*** in the ***Sources in Project*** window and double click on ***Simulate Behavioral Model*** in the ***Processes for current***

sources window. This tells the Xilinx software to launch ModelSim and use the test bench file to test the main code.



Four new ModelSim windows will appear. For this lab, only the one titled wave - default (a black screen with green lines) will be utilized. Expand the waveform by clicking on the magnifying glass that says Zoom Full. The waveform should look like the following.



Now it can be seen that anytime ***a*** and ***b*** are low or ***c*** is high, ***y*** will be high. This could be thought of in terms of voltages with the lower line being 0 volts and the elevated line being 5 volts. Click anywhere on the waveform. This will create a cursor that can be moved across the waveform. This will show the input and output status at various times.