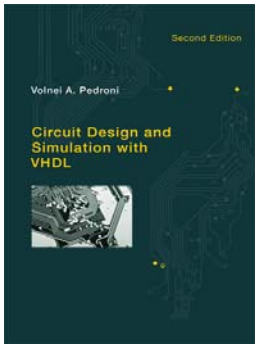


Circuit Design and Simulation with VHDL
2nd edition
Volnei A. Pedroni
MIT Press, 2010
Book web: www.vhdl.us

Appendix C

Xilinx ISE Tutorial

(ISE 11.1)

This tutorial is based on **ISE 11.1 WebPack** (free at www.xilinx.com). The circuit used in the tutorial is the registered multiplexer of figure 1.

1. Introduction

ISE 11.1 allows integrated synthesis and simulation as follows.
Synthesis: With XST (Xilinx Synthesis Technology), Precision RTL (from Mentor Graphics), or Synplify (from Synopsys).
Simulation: With ISim (ISE Simulator, from Xilinx), ModelSim (from Mentor Graphics), NC-Sim (from Cadence), or VCS (from Synopsys)

The registered multiplexer studied in Section 10.5 (repeated in figure 1 below) will be used in this tutorial. Its design and test files are the following:
Design file: reg_mux.vhd, seen in Section 10.5.
Testbench file: reg_mux_tb.vhd, with one option (without automated verification) seen in Example 10.4 and another (with automated verification) seen in Example 10.6. The former will be employed here.

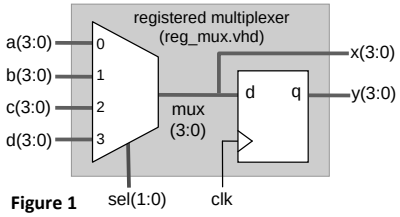


Figure 1

2. Starting a New Project

a) Create a directory where all the design files should be located (work library). Copy the files *reg_mux.vhd* and *reg_mux_tb.vhd* mentioned above to that directory. If they were not typed yet, the ISE text editor can be used (step 2f ahead).

b) Launch ISE.

c) Select **File > New Project**, which will open the Create New Project dialog of figure 2a. Enter the project name and location (note that the project name is automatically copied to the project location). Click **Next**, which opens the dialog of figure 2b.

d) In the Device Properties dialog (figure 2b), select the device, the synthesizer (XST), the simulator (ISim – we will deal with ModelSim too later), and the language (VHDL). Click **Next** and **Finish** until the Project Navigator (figure 3) is opened.

e) Observe the Project Navigator's four windows: Sources, Processes, Transcript, and Workspace. Make sure that Sources for is set to **Implementation**.

f) Open the design (*reg_mux.vhd*) and testbench (*reg_mux_tb.vhd*) files (they will be displayed in the Workspace window). If they were not typed yet, click or select **File > New > Text File** to enter each file.

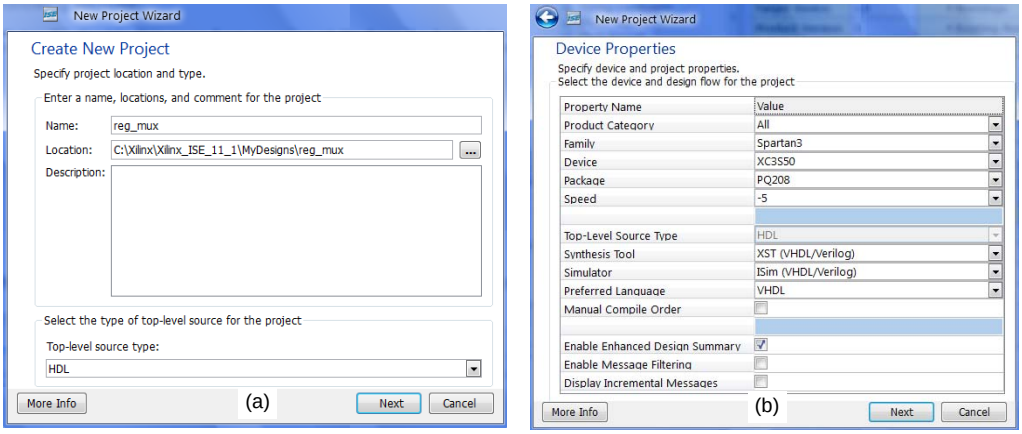


Figure 2

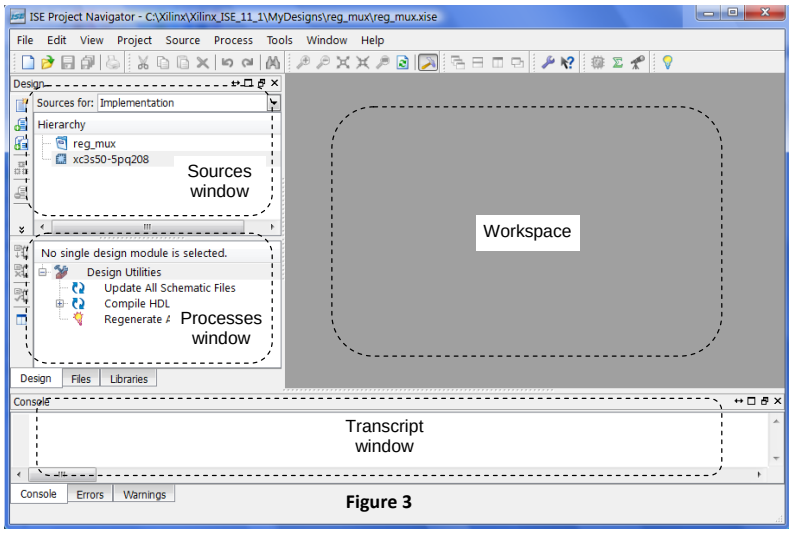


Figure 3

3. Synthesizing the Design

a) First, we must define the project "sources", which are the two files prepared above. Select **Project > Add Source**, highlight *reg_mux.vhd* and click **Open**. This opens the dialog of figure 4a. In the Association field, select **All**. Do the same for *reg_mux_tb.vhd*, now selecting **Simulation** in the Association field, as shown in figure 4b.

Note 1: The first of the files above is the source for synthesis, while the second is the source for simulation. Consequently, if one prefers, the latter can be included later (in Section 5 or 6).

Note 2: If at any point you need to remove sources from the project, proceed as follows.
To remove a design file: In the Source for field of the Sources window, select **Implementation**. Right-click the file name and select **Remove**.
To remove a testbench file: In the Source for field of the Sources window, select **Behavioral Simulation**. Right-click the file name and select **Remove**.

b) Next, set the synthesis effort. Select **Project > Design Goals & Strategies** and choose **Balanced**.

c) The design is now ready to be synthesized. At this point, the Processes window will look like that in figure 5. Note that under the synthesis process there are four possible actions. For example, if the input files are still being debugged, one

might prefer to run just the Check Syntax module. To synthesize the circuit, double-click **Synthesize – XST**. Some of the results from the synthesis process will be examined in the next section.

Note: At any point you can restart the whole process by selecting **Project > Cleanup Project Files**, which will delete the files created by ISE (but not the design and testbench files).

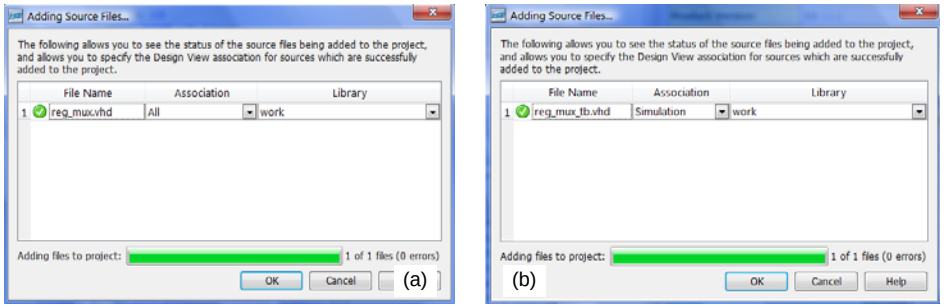


Figure 4

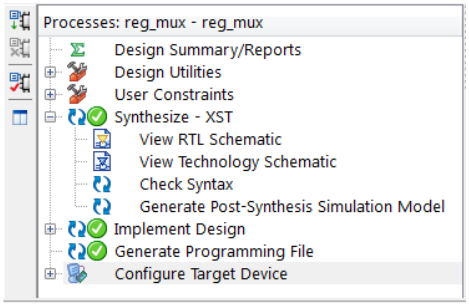


Figure 5

4. Inspecting Synthesis Results

The synthesis reports contain several pieces of valuable information, some of which are examined below.

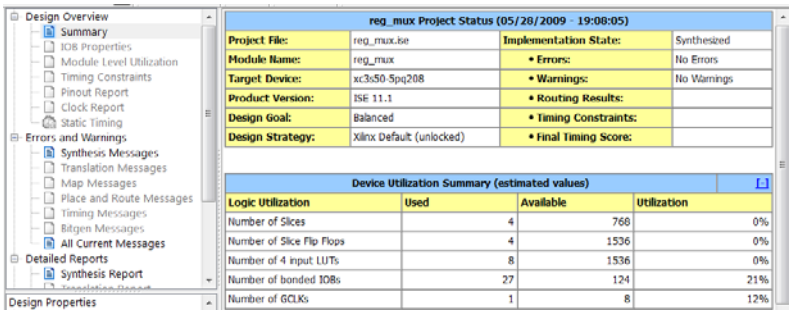
a) *Design parameters:* On the upper left corner of the Workspace window select **Design Overview > Summary**. The result is partially shown in figure 6. Observe in the Project Status table the name of the design, the device used, and the design goal (Balanced).

b) *Device utilization:* In the Device Utilization Summary table of figure 6 observe that 4 slices were used (out of 768 available in the chosen device), 4 flip-flops were inferred (as expected), and that the circuit requires 27 pins (expected: 4x4+2+1 inputs + 2x4 outputs = 27).

c) *RTL view:* Double-click **View RTL Schematic** in figure 5 or select **Tools > Schematic Viewer > RTL**. The Create RTL Schematic dialog will be opened. Click the "+" icon next to Signal, select all signals in the list, then click **Add** to copy the signals to the column on the right. Finally, click **Create Schematic**. The result is shown in figure 7a. Note that this circuit coincides with that in figure 1. This is how our circuit was understood by the synthesizer (after optimization and place & route it might look a little different, but obviously still with the same functionalities).

d) *Technology view:* Double-click **View Technology Schematic** in figure 5 or select **Tools > Schematic Viewer > Technology**, then proceed as in step 4c above, which will cause the final circuit to be exhibited (figure 7b).

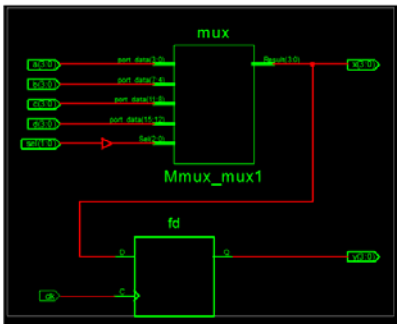
e) *Timing analysis:* Select **Tools > Timing Analyzer > Post-Place & Route**. After the process is concluded, check in the Report Navigation the time values in **Setup/Hold to clock clk** and in **Pad to pad** (a few ns are generally reported), which help give an initial rough idea about the design's maximum speed.



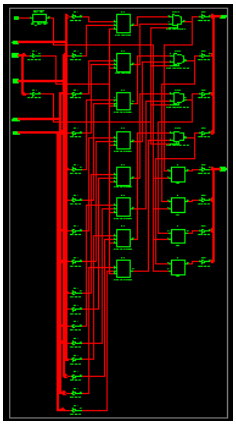
reg_mux Project Status (05/28/2009 - 19:08:05)			
Project File:	reg_mux.ise	**Implementation State:**	Synthesized
Module Name:	reg_mux	**• Errors:**	No Errors
Target Device:	xc3s50-5pq208	**• Warnings:**	No Warnings
Product Version:	ISE 11.1	**• Routing Results:**	
Design Goal:	Balanced	**• Timing Constraints:**	
Design Strategy:	Xilinx Default (unlocked)	**• Final Timing Score:**	

Device Utilization Summary (estimated values)			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slices	4	768	0%
Number of Slice Flip Flops	4	1536	0%
Number of 4 input LUTs	8	1536	0%
Number of bonded IOBs	27	124	21%
Number of GCLKs	1	8	12%

Figure 6



(a)



(b)

Figure 7

5. Simulating the Circuit with ISim

a) If not done yet, enter the source file for simulation (in this tutorial, it was done in step 3a).

b) We must choose between *functional* or *timing* simulation. The former checks only the design functionalities, while the latter includes also the device's internal propagation delays, thus representing the actual circuit.
- For functional simulation, continue in step 5c below.
- For timing simulation, go to step 5d.

c) *Functional simulation:* In the Sources window of the Project Navigator, select **Behavioral Simulation** in the Sources for field (as shown in the upper part of figure 8a). Then highlight the testbench file, which will cause the ISim menu to be exhibited in the Processes window (lower part of figure 8a). Double-click **Simulate Behavioral Model** in the Processes window. This will start ISim, which displays the waveforms window of figure 8b. When done, go to step 5e.

d) *Timingsimulation:* In the Sources window of the Project Navigator, select **Post-Route Simulation** in the Sources for field (as shown in the upper part of figure 8c). Then highlight the testbench file, which will cause the ISim menu to be exhibited in the Processes window (lower part of figure 8c). Double-click **Simulate Post-Place & Route Model** in the Processes window. This will start ISim, which displays the waveform window of figure 8d.

e) In figure 8b or 8d (depending on the simulation type), select all signals but *clk* and change their radix to unsigned decimal (right-click a signal name and select **Radix > Unsigned Decimal**).

f) Click the Zoom to Full View icon  to see the whole plot. Note that the simulation time interval (0.8 us in this example) can be changed to any other value.

g) Examine the results in figure 8b. Note particularly the following:
- They coincide with the results in Example 10.4.
- If it is a functional simulation (figure 8b), there are no time delays between input and output transitions.
- If it is a timing simulation (figure C8d), then there are time delays between input and output transitions. For example, the marker and cursor in the figure show a 6.2 ns time delay between the clock transition and the change in *y*.

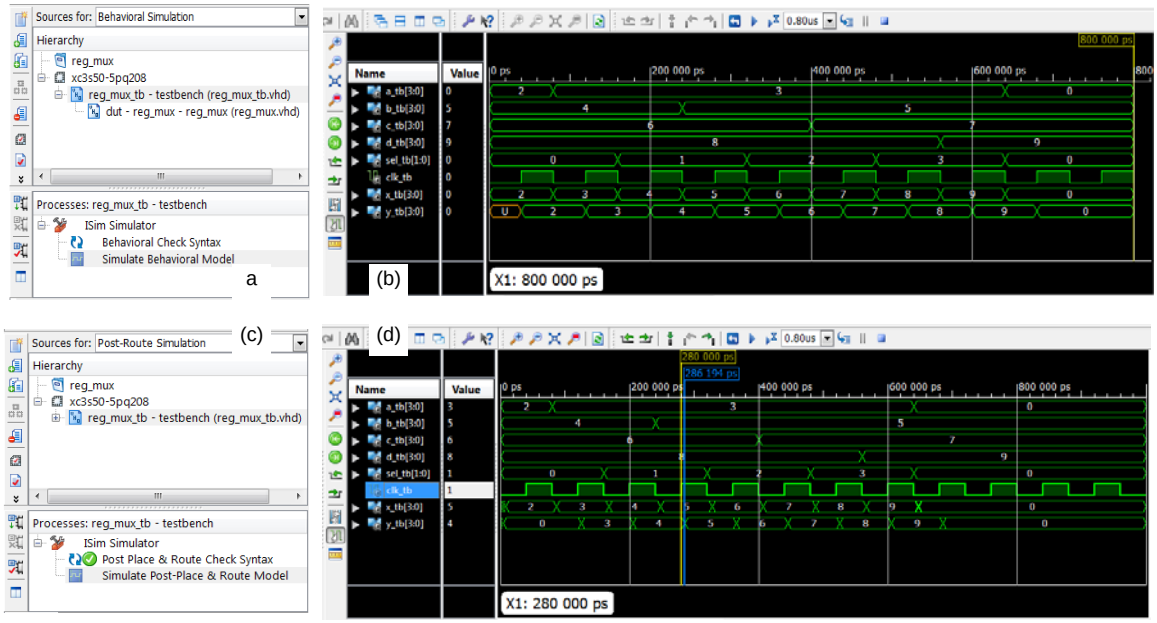


Figure 8

h) To get acquainted with ISim, practice with the Run and Zoom controllers. For example, click the Run for the Time Specified icon , followed by the Zoom to Full View icon , to see that the simulation advances another 0.8 us. Now click the Restart icon , followed by and .

i) Practice also with:

- Previous/Next Transition icons (for example, select *x* and then click one of these icons and observe the corresponding time value at the cursor foot).
- Go to Time 0 and Go to the Latest Time icons .
- Cursor and time markers .

6. Simulating the Circuit with ModelSim

a) If not done yet, enter the source file for simulation (in this tutorial, it was done in step 3a).

b) In the simulation above, ISim was used. To change to ModelSim, right-click anywhere in the Sources window and select **Design Properties**, which opens a dialog similar to that in figure 2b. In the Simulator field, select the ModelSim version available.

c) Now proceed as in Section 5 above to run either functional or timing simulation. When ModelSim is started, follow the ModelSim tutorial of Appendix D.

Note: If ISE is unable to find the ModelSim executable file, select **Edit > Preferences** and enter the full path to the executable file in the Model Tech Simulator field.

7. Making Pin Assignments

a) Start the PlanAhead component of ISE by selecting **Tools > PlanAhead > I/O Pin Planning (PlanAhead) Post-Synthesis**, which opens the PlanAhead navigator of figure 9.

b) To have the I/O pins placed automatically, select **Tools > Auto-Place I/O Ports**. Click **Next** and **Finish**. Observe that device pins are assigned to all 27 circuit ports.

Note: To remove all pin assignments, select **Tools > Clear Placement Constraints** and just follow the dialogs.

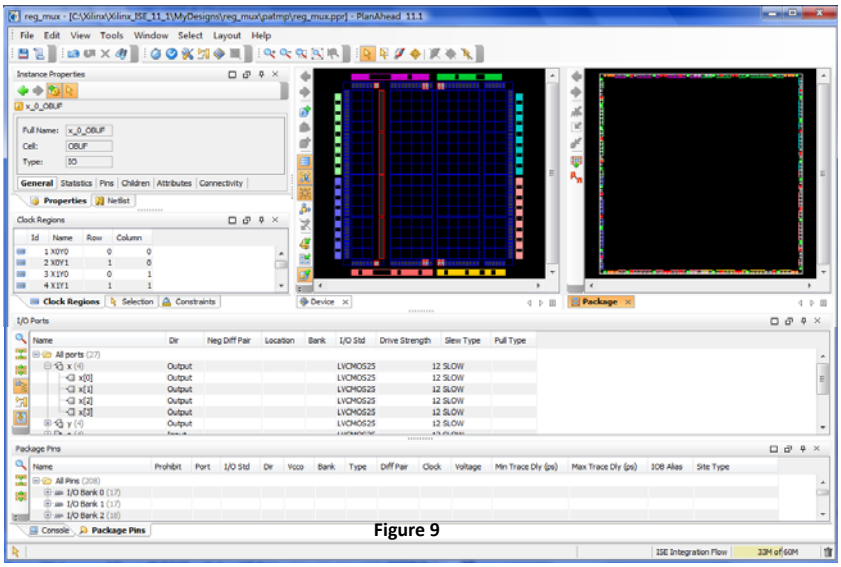


Figure 9

c) In the I/O Ports window of the PlanAhead navigator (figure 9), click any port name and observe that it is highlighted in both the Device and the Package floorplan windows.

d) To change pin assignments, use again the I/O Ports window. Double-click the port name (say, a[0]) or any other position in that line, which will cause that specific port to be displayed in the I/O Port Properties window. In the Site field, enter the desired pin name, then click **Apply**. If that pin is available, the assignment will be accepted.

e) When done making manual pin assignments, run **Tools > DRC** to check (fix) any inconsistencies, then recompile the design (step 3c).

8. Physical Implementation

a) Connect the development board containing the target CPLD or FPGA device to your computer and turn the power on the board on.

b) In the Sources for field of the Sources window select **Implementation**.

c) Still in the Sources window, highlight the design file (*reg_mux.vhd*), which will cause the menu shown in the Processes window of figure 5 to be displayed.

d) In the Processes window, double-click **Implement Design**.

e) Next, double-click **Generate Programming File**.

f) Finally, double-click **Configure Target Device**, which starts the iMPACT tool, responsible for configuring the device (iMPACT requires some setups, like the chain type –Boundary Scan, for example; details about iMPACT are available at the Xilinx website).