Tutorial on FPGA Design Flow based on Aldec Active HDL ver 1.6

Fall 2011
Preparing the Input:

Go to the link given above and download following files.

1. **Synthesizable VHDL Codes:**
   - clock_divider.vhd
   - counter.vhd
   - SSegCtrl.vhd
   - lab3_demo_package.vhd
   - lab3_demo.vhd

2. **Testbench:** lab3_demo_tb.vhd
3. **User Constraints File:** lab3_demo.ucf
4. **Bitstream:** lab3_demo_bitstream.bit (used only if you work with the FPGA board).

**Current Version of Tools:** This tutorial has been tested using the following tools

- **CAD Tools**
  - **ActiveHDL**  
    Versions: 7.2, 8.2, 8.3
  - **Synthesis Tools**
    - Xilinx ISE Webpack XST  
      Versions: 9.1, 10.1, 12.2, 12.4, 13.2
    - Synplify Premier DP  
      Version: D-2010-03
  - **Implementation Tool**
    - Xilinx ISE/WebPack  
      Versions: 9.1, 10.1, 12.2, 12.4, 13.2

- **FPGA Board**
  - Digilent Basys2

**The combinations of tools supported as of Fall 2011 are as follows:**

**At home:**
- Aldec Active-HDL Student Edition ver. 7.2
- Xilinx ISE/Webpack 9.1 SP3

**At GMU:**
- Aldec Active-HDL ver. 8.3 SP1
- Xilinx ISE/Webpack 13.2
- Synplify Premier DP D-2010-03
# Table of Contents

1  **Project Settings**  
   4

2  **Synthesis**  
   2.1 Synthesis using Xilinx XST  
       2.2.1 Synthesis Options  
       2.2.2 Synthesis Report  
   16
   2.2 Synthesis using Synplify Premier DP  
       2.1.1 Synthesis Options  
       2.1.2 Synthesis Report  
   21
   2.3 Post-Synthesis Simulation  
   33

3  **Implementation**  
   3.1 Implementation Options  
   34
   3.2 Implementation Reports  
   37
   3.3 Timing Simulation  
   44

4  **Uploading Bitstream to FPGA Board**  
   46
1. Project Settings
New Workspace

Specify basic information about the new workspace.

Type the workspace name:
lab3_aws

Select the location of the workspace folder:
H:\lab3\valdec_workspace

- Add New Design to Workspace

[OK] [Cancel]
Create new workspace and choose *Create an Empty Design with Design Flow*.

Then press **Next**. You will see a picture similar to the one shown below.
Verify that Flow Configuration Settings are defined as follows:

Synthesis Tool:
- Xilinx ISE/WebPack <version number> XST

Implementation Tool:
- Xilinx ISE/WebPack <version number>

Default Family:
- Xilinx<version number>x SPARTAN 3

If not, click at the Flow Configuration Settings button and adjust appropriately.
Also choose, Block Diagram Configuration

- Default HDL Language
  Default HDL Language
- VHDL

Once done, select Next → Finish
Now you should see a screen divided into several parts, with a Flow panel on the right side. If you do not see the Flow panel on the right side as shown in the picture, you can press Alt+3 or ViewÆFlow from the top menu bar to open the panel.

Go to ToolsÆPreferences. In the category expand Tools--> Flow--> integrated tools. Browse to the latest synthesis and implementation tools and select them appropriately.

Below example refers to the versions of tools available in the GMU Labs. For home use, you may need to use different versions of tools and program paths.

Synthesis tools:
1. Xilinx XST 13.2 (Browse to Xilinx/13.2/ISE/bin/nt/xst.exe)
2. Synplify Premier DP (Select to Synplify Premier with DP D-2010-03 under category “synopsis”)
Implementation tool:
Xilinx XST 13.2   (Browse to Xilinx/13.2/ISE/bin/nt/xst.exe)

If the tools for HDL synthesis and implementation are already selected then click on OK. If not, choose the tools as shown above and the path accordingly. Please note that you will be able to select only one synthesis tool at a time.

Specify the new design name. Download to your hard drive all VHDL files provided to you at the website for lab3 demo.

Add and compile all files from lab3 demo. Then, test your design if it works correctly in the functional simulation as you would normally do. If you are following the tutorial by using lab3demo, make sure you change the \textit{slow\_clock\_period} located inside \textit{lab3\_demo\_package.vhd} to a number suitable for simulation (5). It will take a long time to simulate otherwise.
New Design Wizard

The new design will have the following specifications:

Design name: lab3_demo

Design directory:
H:\lab3\aldec_workspace\lab3_allocated

☐ Compile source files after creation

Back Finish Cancel
Make sure to exclude the .ucf file from compilation.
2. Synthesis

Synthesis can be done either by using Xilinx XST or Synplify Premier DP. Xilinx XST can be used both at school and at home. Synplify Premier DP is available only at school.

2.1 Synthesis using Xilinx XST

2.1.1 Synthesis Options

Click at the options button next to the synthesis icon. Under Synthesis Options select Update synthesis order. Arrange your files in the order from the bottom to the top of the design hierarchy. Exclude your non-synthesizable files, such as testbench. Also select a correct Top-level Unit, which is lab3_demo in the example you follow.

Make sure that your settings under General tab are as follows:

- Family : Xilinx<version_number>x SPARTAN3
- Device  : 3s50pq208
- Speed Grade : -4.

Under Std Synthesis and Adv Synthesis tabs, you can adjust optimization goal of the synthesis tool for various results. Most notably, you can tell the synthesis tool to optimize for either area or speed. To select either one of them, choose Std Synthesis ➔ Optimization Goal ➔ select Speed or Area. Click on OK when you are done with option settings.
Click on the **synthesis** button and wait until synthesis is completed.
2.1.2 Synthesis Report

Minimum clock period, critical path, and resource utilization can be found from the report file generated after synthesis. To view this file, click on the reports button next to the synthesis icon.

Minimum clock period, maximum frequency, and critical path can be found under Timing Summary section. Looking at the critical path can give you an idea which portions of your code to change in order to improve the circuit performance.

Resource utilization is located in the Final Report section.

Example Report: Resource Utilization

![Resource Utilization Report](image-url)
Example Report: Minimum Clock Period and Critical Path

<table>
<thead>
<tr>
<th>CellRef</th>
<th>Propagate</th>
<th>Propagate Delay</th>
<th>Destination Logical Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>slow_clock_gen/counter_1</td>
<td>slow_clock_gen/counter_1</td>
<td>1.216</td>
<td>slow_clock_gen/counter_1</td>
</tr>
<tr>
<td>slow_clock_gen/counter_2</td>
<td>slow_clock_gen/counter_2</td>
<td>0.000</td>
<td>slow_clock_gen/counter_2</td>
</tr>
<tr>
<td>slow_clock_gen/counter_3</td>
<td>slow_clock_gen/counter_3</td>
<td>0.000</td>
<td>slow_clock_gen/counter_3</td>
</tr>
<tr>
<td>slow_clock_gen/counter_4</td>
<td>slow_clock_gen/counter_4</td>
<td>0.000</td>
<td>slow_clock_gen/counter_4</td>
</tr>
<tr>
<td>slow_clock_gen/counter_5</td>
<td>slow_clock_gen/counter_5</td>
<td>0.000</td>
<td>slow_clock_gen/counter_5</td>
</tr>
<tr>
<td>slow_clock_gen/counter_6</td>
<td>slow_clock_gen/counter_6</td>
<td>0.000</td>
<td>slow_clock_gen/counter_6</td>
</tr>
<tr>
<td>slow_clock_gen/counter_7</td>
<td>slow_clock_gen/counter_7</td>
<td>0.000</td>
<td>slow_clock_gen/counter_7</td>
</tr>
<tr>
<td>slow_clock_gen/counter_8</td>
<td>slow_clock_gen/counter_8</td>
<td>0.000</td>
<td>slow_clock_gen/counter_8</td>
</tr>
<tr>
<td>slow_clock_gen/counter_9</td>
<td>slow_clock_gen/counter_9</td>
<td>0.000</td>
<td>slow_clock_gen/counter_9</td>
</tr>
<tr>
<td>slow_clock_gen/counter_10</td>
<td>slow_clock_gen/counter_10</td>
<td>0.000</td>
<td>slow_clock_gen/counter_10</td>
</tr>
<tr>
<td>slow_clock_gen/counter_11</td>
<td>slow_clock_gen/counter_11</td>
<td>0.000</td>
<td>slow_clock_gen/counter_11</td>
</tr>
<tr>
<td>slow_clock_gen/counter_12</td>
<td>slow_clock_gen/counter_12</td>
<td>0.000</td>
<td>slow_clock_gen/counter_12</td>
</tr>
<tr>
<td>slow_clock_gen/counter_13</td>
<td>slow_clock_gen/counter_13</td>
<td>0.000</td>
<td>slow_clock_gen/counter_13</td>
</tr>
<tr>
<td>slow_clock_gen/counter_14</td>
<td>slow_clock_gen/counter_14</td>
<td>0.000</td>
<td>slow_clock_gen/counter_14</td>
</tr>
<tr>
<td>slow_clock_gen/counter_15</td>
<td>slow_clock_gen/counter_15</td>
<td>0.000</td>
<td>slow_clock_gen/counter_15</td>
</tr>
<tr>
<td>slow_clock_gen/counter_16</td>
<td>slow_clock_gen/counter_16</td>
<td>0.000</td>
<td>slow_clock_gen/counter_16</td>
</tr>
<tr>
<td>slow_clock_gen/counter_17</td>
<td>slow_clock_gen/counter_17</td>
<td>0.000</td>
<td>slow_clock_gen/counter_17</td>
</tr>
<tr>
<td>slow_clock_gen/counter_18</td>
<td>slow_clock_gen/counter_18</td>
<td>0.000</td>
<td>slow_clock_gen/counter_18</td>
</tr>
<tr>
<td>slow_clock_gen/counter_19</td>
<td>slow_clock_gen/counter_19</td>
<td>0.000</td>
<td>slow_clock_gen/counter_19</td>
</tr>
<tr>
<td>slow_clock_gen/counter_20</td>
<td>slow_clock_gen/counter_20</td>
<td>0.000</td>
<td>slow_clock_gen/counter_20</td>
</tr>
</tbody>
</table>

Timing Summary:

- **Minimum period**: 7.497ns (Minimum Frequency: 133.29MHz)
- **Maximum input arrival time before clock**: No path found
- **Maximum output required time after clock**: 9.186ns
- **Maximum combinational path delay**: No path found

Timing Details:

All values displayed in nanoseconds (ns)
2.2 Synthesis using Synplify Premier DP

2.2.1 Synthesis Options:

Change the Flow settings in order to select Synplify Premier DP as the primary tool for synthesis.
Click on options button to select the synthesis options
Choose lab3_demo as Top-level Unit from the drop down menu.

Click OK and then click the “synthesis” button right next to the option button in the Design flow manager. Choose “synplifypremierdp” as your default license type.
After applying appropriate settings, click on **synthesis** in the design flow. You should see the following new window.
There is no need to create a project and adding VHDL source files. Tool will automatically select lab3_demo as default project.

Choose the “Fast Synthesis” option and deselect “Physical Synthesis” option. Alternatively, double-click lab3_demo to select the appropriate synthesis options.
Click on “Run” in the Button panel or select “Run” from the Run menu. Alternatively, press F8 as the shortcut to this menu.
After synthesis, Synplify Premier DP will generate the netlist file lab3_demo.edf in the synthesis folder under the path ..\synthesis\lab3_demo. This file is required by Aldec Active-HDL 8.3 for post-synthesis simulation and by Xilinx ISE for implementation.
2.2.2 Synthesis Report

Analyze the results, using report file, the HDL Analyst schematic views, the Message window and the Log Watch window.

Log Watch window:

Select **Compiler Report** to analyze the performance summary, timing information, critical path and resource utilization of the design.
Timing Information:

Device: START OF TIMING REPORT

Mapper Report:

Performance Summary

Worst slack in design: -0.018

<table>
<thead>
<tr>
<th>Starting Clock</th>
<th>Requested Frequency</th>
<th>Estimated Frequency</th>
<th>Period</th>
<th>Requested Period</th>
<th>Estimated Slack</th>
<th>Clock Type</th>
<th>Clock Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>lab3_demo clock</td>
<td>511.5 MHz</td>
<td>500.2 MHz</td>
<td>3.211</td>
<td>3.729</td>
<td>-0.518</td>
<td>inferred</td>
<td>Autoconstruct_clkgroup_0</td>
</tr>
</tbody>
</table>

Resource Usage Report for lab3_demo

Mapping to part: xc3vlx20ctff320-2

Cell usage:

- 7U: 5 uses
- 7IC: 5 uses
- 7DCE: 43 uses
- 7BE: 6 uses
- 7OHD: 8 uses
- 7MBCI_L: 77 uses
- 7MDM: 2 uses
- 7WCC: 4 uses
- 7MBC: 1 use
- 7LUTS: 17 uses
- 7DR: 1 use
- 7LUT4: 24 uses
- 7LUT5: 4 uses
- 7LUT6: 28 uses

I/O ports: 14

L/O primitives: 14

- 1LBUF: 1 use
- 1LBUF_L: 1 use
- 1DBUF: 12 uses
- 1BUF: 1 use

L/Z register bits: 0

Global Clock Buffers: 1 of 32 (0%)

Total load per clock:

lab3_demo clock: 61

Mapping Summary:

Total LUTs: 63 (100%)

Number of unique control sets: 6

Mapper successful!

Process took 0h:00m:00s realtime, 0h:00m:00s optime
Critical Path Information:

Note: Critical path information is also available in HDL Analyst view for better visibility.

HDL Analyst schematic views:

Select HDL Analyst-> RTL->Hierarchical View or Flattened View to view the design graphically.

RTL Hierarchical view: most designs are hierarchical so interactive hierarchical viewing helps to better analyze the design.
**RTL Flattened view:** Flattening removes hierarchy so you can view the logic without hierarchy levels.
Critical path: lab3_demo

To generate a view of critical path with Physical analyst tool, click on show critical path (stopwatch icon) or select the command from the menu.

Example report: Critical path
2.3 Post-Synthesis Simulation

Click at the options button next to the post-synthesis simulation icon. Remove the default input file, and select netlist file lab3_demo.edf from synthesis folder and your testbench as an input file by clicking at the button close to the cross sign (marked by a dot). Then, select Recompile Files. Once done, choose the appropriate top-level unit, which is lab3_demo_tb.vhd in this example.

Press OK, and then select post-synthesis simulation. Now you should see timing waveforms similar to the ones obtained during functional simulation. The difference is that the components and signals are now mapped into appropriate FPGA hardware.
3 Implementation

3.1 Implementation Options

Click at the options button next to the implementation icon. Select the correct Netlist File which is a file with the same name as your top level VHDL file and the extension .edf. It is normally located in the synthesis folder of your workspace. Use this file to implement your design. Choose the correct FPGA Family, Device and Speed Grade, the same as used during the Synthesis phase:

Browse to synthesis folder and find Netlist file lab3_demo.edf generated after synthesis

Browse to user constraint file lab3_demo.ucf, provided in the example Lab 3
In our example these should be (please ignore values given in the screen shot above):

- **Family**: Xilinx<version_number>x SPARTAN3
- **Device**: 3s50pq208
- **Speed Grade**: -4.

Under **Constraint File**, select **Custom constraint file**. Browse to your .ucf for the lab, lab3_demo.ucf in our example. Then, navigate to the **BitStream** tab by clicking at the right arrow at the top right hand corner. Under **General** tab of **BitStream** deselect **Do Not Run Bitgen**. This will create bitstream, .bit, which you can upload to FPGA.

Additionally, under **Post-Map STR**, **Post-PAR STR**, and **Simulation** tabs make sure that your device speed grade is set to 4.
You can also specify the implementation tool to use a certain optimization goal. To do this, go to **Advanced Map → Optimization Goal** → select either **Area** or **Speed**.

Press **OK**, and then select **implementation**.
3.2 Implementation Reports Analysis

Similarly to synthesis, you can access the generated reports by clicking the reports button, near the implementation icon. Unlike synthesis log, implementation report is divided into several smaller reports, which are named differently. Below is a list of reports in which you can find the most useful information about your design after implementation, such as resource utilization, maximum clock frequency, and critical path:
Resource Utilization:

- **Map**: See Design Summary
- **Place & Route**: See Device Utilization Summary

Note: Place & Route provides overall information about the design after placing and routing. Map provides a more detailed summary of resource utilization.

Minimum Clock Period (Maximum Frequency):

- **Post-Place & Route Static Timing Report**

This file describes the worst case scenario in terms of minimum clock period. However, since the implementation tools do not provide complete information, please refer to **Timing Analysis** below for a more detailed report.

Note: Post-Map Static Timing Report can be ignored because it provides timing report before placing & routing, and thus cannot correctly predict interconnect delays.
Timing Analysis (Clock Period, Maximum Frequency and Critical Path)

For the detailed analysis of critical path and minimum clock period (or maximum frequency) a separate timing analyzer provided by Xilinx should be used. To generate the report, select **Analysis → Static Timing Analyzer** from the **Flow** panel. This will open Xilinx Timing Analyzer. You can also navigate to the program from the Windows menu. The path used in the ECE labs at GMU is by **Start → All Programs → VLSI Tools → Xilinx ISE → Accessories → Timing Analyzer**.

Once the program is opened, select **Open**, choose netlist file located in /implement/ver1/rev1 of your workspace, * .ncd, and press **OK**. Selecting **Analyze against Auto Generated Design Constraints** will generate a static timing report.
Release 13.2 Trace Inc.
Copyright (C) 1994-2001 Xilinx, Inc. All rights reserved.

Design files: lab3_demo.ncf, lab3_demo_postpar.tw

Open Design
Specify design and constraints file(s) to open.

Design:
- lab3 Demo.ncf

Report:
- Only physical design (.ncd) and constraints (.pdf and .jdf) files will be opened immediately. Translated report (.ngd) will be opened only when more probing to other applications.

- Physical design file (.ncd):
  - lab3 Demo.ncf

- Physical constraints file (.pdf):
  - lab3 Demo_postpar.pcf

- Translated report file (.ngd):
  - lab3 Demo_postpar.ngd

- User constraints file (.jdf):
  - lab3 Demo_postpar.jdf

Data:
- Automatically fill in the files specified above for the next session.

OK  Cancel  Help

Getting Started  lab3_demo_postpar.tw
Example Report: Clock period, Maximum Frequency and Critical Path.
3.3 Timing Simulation

Click at the options button next to the timing simulation icon. Select your testbench as the Top-Level Unit. Afterwards, select timing simulation, which will generate timing waveforms based on your netlist after implementation. You should notice slight timing delays compared to the waveforms from your post-synthesis simulation & functional simulation.
4. Uploading Bitstream to FPGA Board

Before uploading Bit file, make sure that you change your constant values in all your files to proper values, and re-synthesize/re-implement all the files. In particular, in our example, please change the value of the constant slow_clock_period in the Lab3Demo_package.vhd.

Select the Adept program as shown in the picture above. When the program is opened, a device will be shown if it is connected and recognized. Select the bit file by clicking Browse and finding the appropriate file. Click Program to program the device.
Good luck! Have fun debugging =)