Tutorial on FPGA Design Flow based on Aldec Active HDL ver 1.7

Fall 2012
Preparing the Input:

Go to the link given above and download following files.

1. **Synthesizable VHDL Codes:**
   a. clock_divider.vhd
   b. counter.vhd
   c. SSegCtrl.vhd
   d. lab3_demo_package.vhd
   e. lab3_demo.vhd

2. **Testbench:** lab3_demo_tb.vhd
3. **User Constraints File:** lab3_demo_ucf.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**
- **Simulator** (Aldec Active-HDL ver. 8.3, 9.1)
- **Synthesis Tools** (Xilinx ISE Webpack XST ver. 13.2, 14.2 and Synplify Premier DP ver. D-2010-03)
- **Implementation Tools** (Xilinx ISE/WebPack ver. 13.2, 14.2)
- **FPGA Board** (Digilent Basys2)

**The combinations of tools supported as of Fall 2012 are as follows:**

**At home:**
Aldec Active-HDL Student Edition
Xilinx ISE/Webpack 14.2

**At GMU:**
Aldec Active-HDL ver. 9.1
Xilinx ISE/Webpack 14.2
Synplify Premier DP D-2010-03
# Table of Contents

1  Project Settings  

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

3  Implementation  
   3.1  Implementation Options  
   3.2  Implementation Reports  
   3.3  Timing Simulation  

4  Uploading Bitstream to FPGA Board
1. Project Settings
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** and then **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 go to View menu and click on Flow from the top menu bar to open the panel.

Go to Tool→Preferences. 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 <version number> (Browse to Xilinx/<version number>/ISE/bin/nt/xst.exe)
2. Synplify Premier DP (Select to Synplify Premier with DP <version number> under category “synopsis”)
Implementation tool:
Xilinx XST <version number> (Browse to Xilinx/<version number>/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 slow_clock_period located inside lab3_demo_package.vhd to a number suitable for simulation (5). It will take a long time to simulate otherwise.
The new design will have the following specifications:

**Design name:** lab3_demo

**Design directory:**
H:\lab3\lab3_ws\lab3_aws

- [ ] Compile source files after creation

[Back] [Finish] [Cancel]
Compile all files except .ucf file from compilation. Go to Design menu and choose “Compile All” option.
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, click on Std Synthesis tab and chose Optimization Goal to be either 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
Example Report: Minimum Clock Period and Critical Path

Timing Summary:

- Speed Grade: -4
- Minimum period: 7.497ns (Maximun Frequency: 131.107MHz)
- Minimum input arrival time before clock: No path found
- Maximum output required time after clock: 0.100ns
- Maximum combinational path delay: No path found

Timing details:

- All values displayed in nanoseconds (ns)

==================================================================================================================================================================================================================
Timing constraint: Default period analysis for Clock 'clock'
Clock periods: 7.497ns (frequency: 131.107MHz)
Total number of paths / destination gates: 1959 / 44

==================================================================================================================================================================================================================
Delay: 7.497ns (44 value of longest = 33)
Source: slow_clock_gen/counter_1 (FF)
Destination: slow_clock_gen/counter_11 (FF)
Source clock: clock
Destination Clock: clock

Data Path: slow_clock_gen/counter_1 to slow_clock_gen/counter_11

<table>
<thead>
<tr>
<th>Source</th>
<th>Delay</th>
<th>Destination</th>
<th>Logical Name</th>
<th>Net Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>PDC1&lt;0</td>
<td>2.720</td>
<td>slow_clock_gen/counter_1 (slow_clock_gen/counter_1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LUTT16&lt;0</td>
<td>0.500</td>
<td>slow_clock_gen/counter_1 (slow_clock_gen/counter_1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LUTT16&lt;0</td>
<td>0.064</td>
<td>slow_clock_gen/counter_1 (slow_clock_gen/counter_1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LUTT16&lt;0</td>
<td>0.064</td>
<td>slow_clock_gen/counter_1 (slow_clock_gen/counter_1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LUTT16&lt;0</td>
<td>0.064</td>
<td>slow_clock_gen/counter_1 (slow_clock_gen/counter_1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LUTT16&lt;0</td>
<td>0.064</td>
<td>slow_clock_gen/counter_1 (slow_clock_gen/counter_1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LUTT16&lt;0</td>
<td>0.064</td>
<td>slow_clock_gen/counter_1 (slow_clock_gen/counter_1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LUTT16&lt;0</td>
<td>0.064</td>
<td>slow_clock_gen/counter_1 (slow_clock_gen/counter_1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LUTT16&lt;0</td>
<td>0.064</td>
<td>slow_clock_gen/counter_1 (slow_clock_gen/counter_1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LUTT16&lt;0</td>
<td>0.064</td>
<td>slow_clock_gen/counter_1 (slow_clock_gen/counter_1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LUTT16&lt;0</td>
<td>0.064</td>
<td>slow_clock_gen/counter_1 (slow_clock_gen/counter_1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LUTT16&lt;0</td>
<td>0.064</td>
<td>slow_clock_gen/counter_1 (slow_clock_gen/counter_1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LUTT16&lt;0</td>
<td>0.064</td>
<td>slow_clock_gen/counter_1 (slow_clock_gen/counter_1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LUTT16&lt;0</td>
<td>0.064</td>
<td>slow_clock_gen/counter_1 (slow_clock_gen/counter_1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LUTT16&lt;0</td>
<td>0.064</td>
<td>slow_clock_gen/counter_1 (slow_clock_gen/counter_1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LUTT16&lt;0</td>
<td>0.064</td>
<td>slow_clock_gen/counter_1 (slow_clock_gen/counter_1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LUTT16&lt;0</td>
<td>0.064</td>
<td>slow_clock_gen/counter_1 (slow_clock_gen/counter_1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LUTT16&lt;0</td>
<td>0.064</td>
<td>slow_clock_gen/counter_1 (slow_clock_gen/counter_1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LUTT16&lt;0</td>
<td>0.064</td>
<td>slow_clock_gen/counter_1 (slow_clock_gen/counter_1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LUTT16&lt;0</td>
<td>0.064</td>
<td>slow_clock_gen/counter_1 (slow_clock_gen/counter_1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LUTT16&lt;0</td>
<td>0.064</td>
<td>slow_clock_gen/counter_1 (slow_clock_gen/counter_1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LUTT16&lt;0</td>
<td>0.064</td>
<td>slow_clock_gen/counter_1 (slow_clock_gen/counter_1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LUTT16&lt;0</td>
<td>0.064</td>
<td>slow_clock_gen/counter_1 (slow_clock_gen/counter_1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LUTT16&lt;0</td>
<td>0.064</td>
<td>slow_clock_gen/counter_1 (slow_clock_gen/counter_1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LUTT16&lt;0</td>
<td>0.064</td>
<td>slow_clock_gen/counter_1 (slow_clock_gen/counter_1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LUTT16&lt;0</td>
<td>0.064</td>
<td>slow_clock_gen/counter_1 (slow_clock_gen/counter_1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LUTT16&lt;0</td>
<td>0.064</td>
<td>slow_clock_gen/counter_1 (slow_clock_gen/counter_1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LUTT16&lt;0</td>
<td>0.064</td>
<td>slow_clock_gen/counter_1 (slow_clock_gen/counter_1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LUTT16&lt;0</td>
<td>0.064</td>
<td>slow_clock_gen/counter_1 (slow_clock_gen/counter_1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LUTT16&lt;0</td>
<td>0.064</td>
<td>slow_clock_gen/counter_1 (slow_clock_gen/counter_1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LUTT16&lt;0</td>
<td>0.064</td>
<td>slow_clock_gen/counter_1 (slow_clock_gen/counter_1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LUTT16&lt;0</td>
<td>0.064</td>
<td>slow_clock_gen/counter_1 (slow_clock_gen/counter_1)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
2.2 Synthesis using Synplify Premier DP

2.2.1 Synthesis Options:

Icon to select Design Flow Manager

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 <version number> 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:

# TIMING REPORT #
# Timing Report written on Sat Sep 10 19:54:48 2011 #

Top view: lab3_demo
Requested Frequency: 311.5 MHz
Core load node: t9
Inputs requested: 6
Constraints File(s):

Ob: M1829 : This timing report estimates place and route data. Please look at the place and route timing report for final
QO: M1829 : Clock constraints cover only FF-to-FF paths associated with the clock.

Performance Summary

Worst slack in design: -0.518

<table>
<thead>
<tr>
<th>Starting Clock</th>
<th>Requested Frequency</th>
<th>Estimated Frequency</th>
<th>Requested Period</th>
<th>Estimated Period</th>
<th>Slack</th>
<th>Clock Type</th>
<th>Clock Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>lab3_demo.clock</td>
<td>311.5 MHz</td>
<td>311.5 MHz</td>
<td>0.211</td>
<td>0.739</td>
<td>0.528</td>
<td>inferred</td>
<td></td>
</tr>
</tbody>
</table>

Mapper Report:

1. command (lab3_demo)
   - Compiler Report
   - Pre-mapping Report (up-to-date)
   - Pre-mapping Report
   - Mapper Report
   - Timing Report
     - Performance Summary
     - Clock Relationships
     - Interface Information
     - Detailed Report for Clock lab3_demo.clock
     - Starting Points with Worst Slack
     - Ending Points with Worst Slack
     - Worst Path Information
   - Resource Utilization

Log File Links:

- lab3_demo
  - Hierarchical Area Report
    - lab3_demo (19.34 10: Sep)
- Session Log

Resource Usage Report for lab3_demo

Mapping to part: xef6v120xf0232-2
Cell usage:
- FO: 5 uses
- FOCE: 26 uses
- FOCE: 40 uses
- FOR: 6 uses
- GEN: 8 uses
- MGRY_L: 77 uses
- MGRY_S: 2 uses
- TCO: 2 uses
- TCO: 2 uses
- TMCY: 27 uses
- LUT1: 1 use
- LUT2: 17 uses
- LUT3: 1 use
- LUT4: 24 uses
- LUT5: 4 uses
- LUT6: 28 uses

I/O ports: 14
I/O primitives: 14
IBUF: 1 use
IBUF: 1 use
IBUF: 12 uses

IOCE: 1 use
I/O Register bits: 0
Register bits not including I/Os: 56 (01)
Global Clock Buffers: 1 at 32 (32)
Total load per clock:
- lab3_demo.clock: 51

Mapping Summary:
Total LUTs: 63 (01)

- Number of unique control sets: 6
- Mapper successful!
Process took 5:00m 01s realtime, 0h:00m:01s cputime
Critical Path Information:

<table>
<thead>
<tr>
<th>Instance</th>
<th>Net</th>
<th>Type</th>
<th>Pin</th>
<th>Pin</th>
<th>Delay</th>
<th>Arrival</th>
<th>No. of Fanout</th>
</tr>
</thead>
<tbody>
<tr>
<td>clockclk_gen.counter[1]</td>
<td>clockclk_gen.counter[1]</td>
<td>REC</td>
<td>0</td>
<td>0</td>
<td>0.076</td>
<td>0.056</td>
<td>-</td>
</tr>
<tr>
<td>counter[1]</td>
<td>clockclk_gen.counter[1]</td>
<td>REC</td>
<td>0</td>
<td>0</td>
<td>0.376</td>
<td>4.056</td>
<td>-</td>
</tr>
<tr>
<td>clockclk_gen.clk_div, u01_counter[0,5]</td>
<td>clockclk_gen.clk_div, u01_counter[0,5]</td>
<td>ADD</td>
<td>10</td>
<td>1</td>
<td>4.77</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>clockclk_gen.clk_div, u01_counter[0,5]</td>
<td>clockclk_gen.clk_div, u01_counter[0,5]</td>
<td>ADD</td>
<td>0</td>
<td>0</td>
<td>0.692</td>
<td>5.079</td>
<td>-</td>
</tr>
<tr>
<td>u01_counter[0]</td>
<td>clockclk_gen.clk_div, u01_counter[0,5]</td>
<td>ADD</td>
<td>0</td>
<td>0</td>
<td>0.266</td>
<td>-</td>
<td>1</td>
</tr>
<tr>
<td>clockclk_gen.clk_div, u01_counter[0,5]</td>
<td>clockclk_gen.clk_div, u01_counter[0,5]</td>
<td>ADD</td>
<td>0</td>
<td>0</td>
<td>0.16</td>
<td>4.801</td>
<td>-</td>
</tr>
<tr>
<td>u01_counter[1]</td>
<td>clockclk_gen.clk_div, u01_counter[0,5]</td>
<td>ADD</td>
<td>0</td>
<td>0</td>
<td>0.266</td>
<td>-</td>
<td>1</td>
</tr>
<tr>
<td>clockclk_gen.clk_div, u01_counter[0,5]</td>
<td>clockclk_gen.clk_div, u01_counter[0,5]</td>
<td>ADD</td>
<td>0</td>
<td>0</td>
<td>0.266</td>
<td>-</td>
<td>13</td>
</tr>
<tr>
<td>clockclk_gen.clk_div, u01_counter[0,5]</td>
<td>clockclk_gen.clk_div, u01_counter[0,5]</td>
<td>ADD</td>
<td>0</td>
<td>0</td>
<td>0.266</td>
<td>-</td>
<td>14</td>
</tr>
<tr>
<td>clockclk_gen.clk_div, u01_counter[0,5]</td>
<td>clockclk_gen.clk_div, u01_counter[0,5]</td>
<td>ADD</td>
<td>0</td>
<td>0</td>
<td>0.266</td>
<td>-</td>
<td>40</td>
</tr>
</tbody>
</table>

Total path delay (propagation time + setup) of 2.292 is 1.78% (47.4%) logic and 1.98% (52.4%) route.
Path delay compensated for clock skew. Clock skew is added to clock-to-out value, and is subtracted from setup time value.

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.
2.3 Post-Synthesis Simulation

Click on the option button next to **post-synthesis simulation** to select the input files. The input file is automatically generated and is included in the post-synthesis simulation option based on your selection for synthesis tool (Xilinx XST or Synplify premier DP).

**Input file after synthesis using Xilinx XST**

**Input file (.edf) after synthesis using Synplify premier DP**

Click on this icon to add the test bench file here
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 on the option button next to timing simulation to perform setting for timing simulations.
Browse to synthesis folder and find Netlist file lab3_demo.edf generated after synthesis (Synplify premier DP) or lab3_demo.ngc after synthesis (Xilinx XST).

Browse to user constraint file lab3_demo.ucf, provided in the example Lab 3.
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:

In our example these should be (please ignore values given in the screen shot above):

- **Family**: Xilinx<version_number>xSPARTAN3
- **Device**: 3s50pq208
- **Speed Grade**: -4.

### 3.1 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.
Release 13.2 Map 0.61xd (st)
Silicon Mapping Report file for Design 'lab3_demos'

Design Information

Command line: map -p SVLZDSTFFS39-2 -o map.aid -pc off -ol high -om area -it off -l global,cs off -lc off -m off lab3_demos.mbf lab3_demos.pcf
Target device: xeSvlx300
Target Package: FF329
Target Speed: -2
Mapper Version: vesrtis -- $Revision: 1.55 $
Mapped Date: Sat Sep 10 23:12:31 2011

Design Summary

Number of errors: 0
Number of warnings: 0

Slice Logic Utilization:

Number of Slice Registers: 96 out of 12,490 1%
Number used as flip-flops: 96
Number of Slice LUTs: 106 out of 12,490 1%
Number used as slices: 133 out of 12,490 1%
Number using O6 output only: 96
Number using O5 output only: 77
Number using O4 and O5: 12
Number using an exclusive route-thru: 3
Number of route-thrus: 80
Number using O6 output only: 80

Slice Logic Distribution:

Number of occupied slices: 42 out of 3,120 1%
Number of LUT flip-flop pairs used: 107
Number with an unused flip-flop: 91 out of 197 29%
Number with an unused LUT: 107 out of 197 1%
Number of fully used LUT-FP pairs: 95 out of 197 60%
Number of unique control sets: 5
Number of slice register site pairs to control set restrictions: 4 out of 12,490 1%
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.
Clock to Pad

Destination: to Pad | Internal Clock(s) | Place

- E_Seq0
- E_Seq1
- E_Seq2
- E_Seq3
- E_Seq4
- E_Seq5

Clock to Setup on Destination clock clock

Source Clock: (Sett: Rise) (Sett: Fall) (Sink: Rise) (Sink: Fall)

Clock: 5.762
Example Report: Clock period, Maximum Frequency and Critical Path.
3.2 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.
Specifying the frequency/ Time Period: Open any text editor and create two files (one with extension .xcf and the other with extension .ucf). Specify constraints regarding Synthesis in the xcf file and constraints related to Implementation in the ucf file. Edit both xcf and ucf files to specify timing constraints (clock period of 10 ns) on the clock in the file.

e.g., NET "clk" PERIOD = 10 ns;
Optimization strategies: Click on the options icon beside implementation to select a complete list of optimization strategies (Translate, map, place & route optimizations)
Print the output waveform in PDF format: In order to view all the important details of the behavioral, post-synthesis, post-translate, post-map and post-place & route simulations in waveform in PDF format, use 3rd part tools
Go to file menu and click on the print preview.

Show page setup dialog (to choose time range that is visible in print and choose full color to see colored figure)

A pop up will appear when you click Show page setup dialog icon. It will give you the option to select time range (full range or for a specific range) and an option to have colored or black and white figure.

Click OK and chose PDF Lite or any other tool to do PDF conversion. After completing the print setup, a new window pops up and ask you to save the file in PDF format.

Here is the output waveform of post-translate simulation in PDF format.
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.