Implementation and Timing Analysis

PURPOSE - This lab will continue to walk you through typical design steps. Particularly, you will continue with Implementation and Timing Analysis steps of your design, which you started during the first lab.

----

1. Introduction

A short description of all the design steps was given in the Introduction section of the previous lab. You may want to revisit it.

You will continue your design (which you Entered, Behaviorally simulated, and Synthesized during the previous lab) with the Implementation and Timing Simulation steps.

2. Start ISE by clicking on the Project Navigator icon on your desktop or by:

`Start ->Programs ->Xilinx ISE 8.2i ->Project Navigator`

Create a new adder project in a new directory lab2 under labs and load the adder project.

Design Implementation is the process of translating, mapping, placing, routing, and generating a Bit file for your design:

In what follows a short description of the steps performed during Implementation is presented:

i) ***Translate*** -- All the input netlists are merged. This is done by NgdBuild.

ii) ***MAP (FPGAs)*** -- Maps a logical design to a Xilinx FPGA. The input to a mapping program is an NGD file. MAP first performs a logical DRC (Design Rule Check) on the design in the NGD file. MAP then maps the logic to the components (logic cells, I/O cells, and other components) in the target Xilinx FPGA. The output design is an NCD (Native Circuit Description) file physically representing the design mapped to the components in the Xilinx FPGA.

iii) ***Place and Route (FPGAs)*** -- Done by PAR (Xilinx's Place and Route program). PAR takes an NCD file, places and routes the design, and produces an NCD file, which is used by the bitstream generator.

PAR places and routes a design using a combination of two methods.

-- Cost-based - This means that placement and routing are performed using various cost tables which assign weighted values to relevant factors such as constraints, length of connection and available routing resources.

-- Timing-Driven - PAR places and routes a design based upon your timing constraints.

iv) ***Generating the Programming File*** -- After the design has been completely routed, you must configure the device so that it can execute the desired function. Xilinx's bitstream generation program, BitGen, takes a fully routed NCD (Circuit Description) file as its input and produces a configuration bitstream - a binary file with a .bit extension.

The BIT file contains all of the configuration information from the NCD file defining the internal logic and interconnections of the FPGA, plus device-specific information from other files associated with the target device. The binary data in the BIT file can then be downloaded into the FPGA's memory cells, or it can be used to create a PROM file.

3. Running the implementation

To run the implementation step, inside Process View panel of Project Navigator, right click and select Run. This runs all processes.
A check mark in the Processes for Current Source denotes a process that was run successfully. An exclamation mark indicates that the process was run and that there is a warning for the process. More information about warnings can be obtained in the Transcript window.

Browse any output files that may be created by the implementation process. Specifically see the Map Report, Place & Route Report and the Pad Report.

4. Viewing the Design in Floorplanner

Now, you can view the implemented design in the Floorplanner:

- Select **fourbit_adder** in the Sources in Project window.
- In the Processes for Current Source window, click the **+** sign beside Implement Design and the **+** sign beside Place & Route.
- Double-click View/Edit Placed Design (Floorplanner).

Floorplanner is launched and displays the placement of the design for the project.
The Floorplanner is an interactive graphical tool that allows you to view and edit location constraints in your designs. The Floorplanner provides information on design connectivity and resource requirements, target FPGA resource layout, and design mapping via location constraints. You can manually or automatically place logic into a floorplan for a selected FPGA. The graphical user interface includes pull-down menus and toolbar buttons that contain all of the necessary commands for changing the design hierarchy, floorplanning, and performing design rule checks. Dialog boxes allow you to quickly set parameters and options for command execution.

NOTE: Take some time and get familiar with the Floorplanner. For those who would like to master this tool it is **highly recommended** to go thorough the entire Help of it.

To view the implemented design results in a more meaningful way, you can display and zoom in on the input/output signals.

- In the fourbit_adder.fnf Design Hierarchy window (`View->Hierarchy`), select the top-level hierarchy, fourbit_adder, to show the signals in the Placement window.
- **Note** Alternatively, you can draw a rectangle around the design area in the Placement window to show the signals.
- Select **`View-> Zoom-> To Selected`** or click the Zoom to Selected icon in the Floorplanner toolbar.
- Verify that all the I/Os are accounted for by holding the cursor over each of the pads and reading the pad name in the lower left corner of the Floorplanner window.
- **Note:** Alternatively, you can view isolated signals in the placement window by selecting individual signals from the list in the fourbit_adder.fnf Design Hierarchy window.
- You can also see the IOs connected to a specific element in the placement window by clicking on it. If the nets do not appear enable this option from the menu Edit->Preferences and Ratsnest tab. Put a check mark in the “Display nets connected to selected logic” check box.

When you have finished viewing the implemented design, save the Floorplanner design view using **`File ->Save`** and exit Floorplanner.

5. Performing timing analysis

Timing Analyzer provides a powerful, flexible, yet easy way to perform static timing analysis on FPGA and CPLD designs. Analysis may be performed immediately after mapping, placing or routing an FPGA design or after fitting and routing a CPLD design. Timing Analyzer may be controlled through GUI features or its comprehensive macro facility.

The Timing Analyzer verifies that the delay along a given path or paths meets your specified timing requirements. It organizes and displays data that allows you to analyze the critical paths in your circuit, the cycle time of the circuit, the delay along any specified paths, and the paths with the greatest delay. It also provides a quick analysis of the effect of different speed grades on the same design.

The Timing Analyzer works with synchronous systems composed of synchronous elements and combinatorial logic. In synchronous design, the Timing Analyzer takes into account all path delays, including clock-to-out and setup requirements, while calculating the worst-case timing of the design. The Timing Analyzer performs setup and hold checks.

The Timing Analyzer creates timing analysis reports based on existing timing constraints or user specified paths within Timing Analyzer.

There are several ways to issue commands to the Timing Analyzer. You can select menus, click toolbar buttons, type keyboard commands in the console window, and run macros. For descriptions of Timing Analyzer commands, see Help Topics.

To start using the Timing Analyzer:

- Select **fourbit_adder** in the Sources in Project window.
- In the Processes for Current Source window, click the + sign beside Implement Design and the + sign beside Place & Route and then Generate Post-Place & Route Static Timing.
- Double-click Analyze Post-Place & Route Static Timing (Timing Analyzer).

The Timing Analyzer opens and automatically loads the project's NGD file.

Xilinx software tools support two different methodologies of implementing timing analysis. For FPGAs, timing is analyzed using existing timing constraints or through user-defined constraints specified with Timing Analyzer commands and filters.

Select `Analyze->Against Timing Constraints`. Here you may customize your analysis. Leave as it is and click Ok. Browse any report files!

NOTE: Take some time and get familiar with the Timing Analyzer menu options. For those who would like to master this tool it is **highly recommended** to go thorough the entire Help of it.

After playing a while and seeing the timing of your placed and routed design, save and close the Timing Analyzer.

Submit a one page description for first 2 lab sessions. Also attach Modelsim graphs for lab1 & Timing Report for lab2.

SUMMARY -- This laboratory wanted to teach you how to perform the Implementation design step, how to see the floorplan, and how to perform timing analysis of the placed and routed design.

Extra Credit:

Find out how to specify timing constraints for your design. Now, go back to the Synthesis step and with tighter timing constraints (try different settings & after each new timing constraint is set, rerun the Implementation step and check whether the design was placed and routed successfully with the timing constraints met). To do that, select (and double click) **`Design Entry Utilities->User Constraints->Edit Implementation Constraints`** (**`Constraints Editor`**) and set **Pad to Pad** to a smaller clock period. Save and close. Rerun implementation. By doing this, you should be able to find out the smallest clock for which your design can be implemented. Report this min clock to your TA at the end of the lab for extra credit!