ECEN3763 - Homework Week 7

Spring, 2022

Due Wednesday February 28th, end of day – 15 Points

The purpose of this lab is to use additional tools in order to report and understand FPGA timing. Timing Analyzer is a GUI based tool in Quartus that allows the designer to view and interact with timing and timing constraints in a design.

This lab will analyze the reset timing for a simple design, and learn one technique for improving reset timing.

Quartus is a complex set of tools that provides multiple approaches to understanding the fitter results. Learning these tools requires practice, and this homework assignment will take you through use of these tools, and how to move between tools as needed.

You will **not** be using your DE10-Standard board for this homework.

Part 1: Register Optimization

1) Open the supplied design in Quartus, and compile the project. You will need to add the files in the source directory to your project first.

    a) How many registers are in the design? ___________________________

2) Review the source files.

    a) How many **counter** registers would you expect to see? _____________

Quartus is optimizing three of the counters away, since the 4 counters are identical and are routed to the same output pins. In order to eliminate the optimization, you will need to modify a default setting in the Quartus project.

3) In the Tasks pane, under Compile Design, expand Analysis & Synthesis, click on Edit Settings.

    a) Select the Advanced Settings (Synthesis) button

    b) Change the Allow Register Merging setting to Off. Be certain to click the Apply button when closing the Compiler Settings window.

    c) In Quartus, open the Tcl console (View > Utility Windows >Tcl Console), and run the command **export_assignments**. Now look in the project .qsf file. What command was written into the .qsf file?

____

3) Recompile the project. How many registers now? ___________________________

4) Since you will not be using your board, you do not need Quartus to run the Assembly step each time. You can disable this by going to the Task Pane, Compile Design > Fitter > Edit Settings. When the Compiler Settings window opens, select Compilation Process Settings in the left pane, and uncheck Run Assembler during compilation. Don't forget to hit the Apply button.

Note: When changes are made in Quartus setting, Quartus stores these settings in memory. The **export_assignments** command forces Quartus to write all current project settings into the project .qsf file.

Part 2: Reset Timing Analysis and Component Locations

1) Launch Timing Analyzer (Tools > Timing Analyzer)

a) In the Tasks pane, double click on Update Timing Netlist. Notice that Create Timing Netlist and Read SDC File are also run. At the time a timing netlist of the complete design has been created, any availble SDC constraints are read in, and the timing netlist is rewritten to contain the SDC constraints.

b) Go to Reports > Custom Reports > Report Path. When the Report Path window opens, do the following :

i) Clock on 3 dots in Targets From:, then in the Name Finder window change Set Collection to get_registers. Clock the List button, and add reset:U4|reset_n to the match list.

ii) Set the Report number of paths to 32, and click the Report Path button

2) What is the range of delay reported on the 32 reset_n paths?

____

The skew on reset_n path is small. The design is very small, and the fitter is able to pack the 4 counters close together.

You can see the locations of the FFs in the Location column of the Data Path. The position is given by X, Y coordinates, where X0 denotes the left side of the die, and Y0 denotes the bottom of the die. For example, FF_X2_Y40 is a location on the left side of the die, about half way down.

3) What different X,Y coordinates are reported for the 8 counter:U0 flip flops?

____

Close Timing Analyzer

4) Launch Chip Planner (Tools > Chip Planner)

Notice that 4 LAB locations are darker blue, indicating logic has been placed there. Ignore the LAB in the center of the die and look at the LABs in the lower right corner. Zoom in enough that you can place the cursor over each location. What are the reported locations of these LABs?

____

The Chip planner provides a graphical view of where logic is placed. This view confirms that all logic was placed very close together. Note that the LAB in the center is only generating a ground signal. You can confirm this by zooming into the lab and double clicking on the rectangle. This will bring up the Resource Property Editor that shows the details of how the specific ALM.

Close the Chip Planner, and close Quartus.

Part 3: Separate the Logic

FPGA tools, including Quartus, provide low level control over component placement. In general, hand placing logic is not recommended. However, for the sake of this homework assignment, we will lock the locations of the counters at different portions of the die, to represent the effects of component placement in a design containing more logic.

1) Find the file location_constraints.txt. This file contains all the assignments necessary to spread the four counters around the FPGA die. These constraints were created by creating Logic Lock areas. If you are interested in how these constraints were generated let me know.

Paste the contents of location_constraints.txt file to the end of the .qsf file, save the .qsf file.

2) Open Quartus and recompile the project.

3) Open Chip Planner and identify the X,Y locations of the counters. The view is a little different, in that the counters appear with a dark blue outline.

You can see that the logic is more spread out. In the Find window (Edit > Find if not already visible), search for *U0*. In the results, find counter:U0|countout[0]. Right click and select Zoom Into Selections. What is the X, Y location of this register?

Counter 0 Bit 0 Location: ______________________

4) Close Chip Planner

5) In Quartus, Tools > Netlist Viewers > Technology Map Viewer (Post-Fitting).

  a) In the Netlist Navigator, find the reset_n register, and double click.

  b) In the graphic view, select the reset_n flip flop, right click, Locate Node > Locate in Chip Planner

  c) You will see the register listed in the Locate History pane in the lower right corner of the screen (you may need to expand where it says Located 1 nodes.

This is another method to determine the physical locations of components. Notice that Quartus provides multiple interconnected ways to find specific placement and configuration details of specific logic components. In most cases you will never need to work at such a granular level, but you may find situations where understanding the details is important.

Close Chip Planner and the Technology Map Viewer.

6) Launch Timing Analyzer.

  a) Run Report Path starting at the reset:U4|reset_n register and ending at all destination registers. Set the Report number of paths to 100 so all paths can be seen.

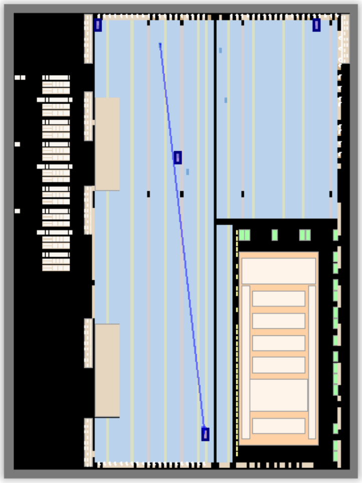
  b) What is the range of skew seen now?

  ____

c) For the longest path, what is the percentage of delay that is interconnect (indicated as IC)? You can find this in the Statistics tab in the Path #1 pane.

____

d) Highlight the longest path, right click, and select Locate Path > Locate in Chip Planner. The net runs almost the entire vertical length of the die, which explains the long delay reported.



Since the logic is now more spread out, a design with a single reset net will result in much higher skew in the reset net. If the clock frequency for this design was 300Mhz instead of 50Mhz, this amount of skew could result in flip flops not coming out of reset on the same clock cycle, which is a design flaw that is very difficult to debug.

How can we fix this?

Close Chip Planner and Timing Analyzer.

Part 4: Create multiple reset_n sources

One approach to improving skew on the reset net is to create multiple reset nets. It is important that there is only one base reset source, but then you can safely duplicate reset nets by creating multiple registered versions of reset.

1) Comment out the 4 counters on lines 27 – 30 in ResetTiming.v. Uncomment out the code in lines 34 – 46.

The code changes create four unique registered reset output signals, one for each counter.

2) Compile the design in Quartus.

3) What is the worst case reset net delay for each of the 4 counters? You will need to use the new net source registers (reset_n_0, reset_n_1, reset_n_2, and reset_n_3).

Counter 0: ____

Counter 1: ____

Counter 2: ____

Counter 3: ____

The long reset net to Counter 2 is now greatly reduced. By separating a single reset net into multiple equivalent nets, the fitter can move the reset source register nearer to the destination registers.

This is one technique that is widely used for designs that use a high percentage of FPGA resources, and/or run at high frequencies, and/or are compiled into large FPGAs.