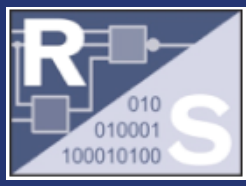
Placement with Irregularities

Mini Task Report

Florian Prott and Julian Käuser
Supervisor: Jakob Wenzel M.Sc.
Submission: 09/25/2017
Institute for Computer Engineering | Computer Systems Group | Prof. Dr.-Ing. Christian Hochberger

TECHNISCHE
UNIVERSITÄT
DARMSTADT



1 Introduction

Routing and Placement are two necessary and time-consuming steps in the FPGA synthesis toolchain. Optimizations for module design in these steps speed up or enhance the FPGA design process significantly. RapidSmith [**?**] allows the implementation of these steps in a Java framework. It offers an API to read in Xilinx designs and FPGA descriptions, and offers researchers the possibility to approach placement and routing problems in an object-oriented way. In RapidSoC, a project based on RapidSmith, a re-placement method for already placed and routed modules is integrated. Although this method works for a majority of modules, some modules routing elements cannot be re-placed due to irregular structures in the FPGA fabric. In this project, the cause for these errors is examined, and an approach to fix them is introduced.

The report is structured as following. First, the exact problem description is presented. The designed approach to the problem using isoelectric planes is described. Then, the insights gained are presented. Finally, future work is proposed, and the report is concluded.

1.1 Task Description

Since the fabric of FPGAs is in general regular, the expected error source for designs at different locations are irregularities regarding the routing elements in the fabric. In the current state, the nets of the design have to be re-routed, which is time-consuming. Since we guess that only 1-2 programmable interconnect points ("PIPs") are set falsely, a detection and repair of the conflicting nets promises faster processing.

1.2 Framework and Environment

This project is embedded into the RapidSmith java environment, version a.b.c. Classes of the RapidSoC framework, especially the class `MoveModulesEverywhere`, are the basis of our work. `MoveModulesEverywhere` parses existing designs from `.xdl` files into the RapidSmith environment, places the included modules on any feasible destination, and generates the corresponding `.xdl` output files. Further processing is accomplished with Xilinx ISE [**?**]. Based on the output of the ISE processing, the re-placed designs can be evaluated to be functional or conflicting.

All processing developed in this project is performed in a separate Java package. Every re-placed design is extracted before the output step, so that the developed methods can be applied to a `Design` object.

2 Task Description

2.1 Is that even necessary?

3 Approaches to the Problem

3.1 Isoelectric Plane Search

here we tell about the potential stuff

3.2 Hand Routing

handrouting tells us that bullshit is happening

4 Insights

In the process of the task, we found out that... X Y Z doesn't work as expected

4.1 Possible Solutions

For the encountered problems, we propose the following approaches

5 Conclusion

the task was to - examine re-placed designs' net lists

- find out if something does not work

- expected: only one or two PIPs are missing due to irreguar structure of FPGA

- if problem is encountered, find rule/mask based method to fix these missing pips

what we did was:

- handle net by net

- for each Pin, determine all elements of the isoelectric it is connected to (class potential)

- checking method: if (potential(source)!=potential(sink) -> broken

- for broken nets:

- search isoelectric of source, sink for adjacent, non-set PIPs (interesction of sets). if !=empty set, activate this pip

- why it did not work:

- even for correct (in terms of make file) nets, the method fails

- hand routing (function of RapidSmith framework) neither does

- we suspect missing information in fabric;

- according to the hand router, the method of "isoelectric search"

results in correct sub nets; might be useful for later work