Microelectronics Journal 45 (2014) 311–324

Contents lists available at ScienceDirect

Microelectronics Journal

journal homepage: www.elsevier.com/locate/mejo

Timing characterization and constraining tool

Piotr Amrozik*, Andrzej Napieralski

Department of Microelectronics and Computer Science, Lodz University of Technology, Wólczanska 221/223, 90-924 Łódź, Poland

ARTICLE INFO

Article history:
Received 7 February 2013
Received in revised form
19 November 2013
Accepted 27 November 2013
Available online 15 December 2013

Keywords:
Design automation
Design methodology
Integrated circuit synthesis
Digital integrated circuits
Reconfigurable logic

ABSTRACT

This paper presents Timing Characterization and constraining Tool (TCT) that facilitates designing of modular reconfigurable Integrated Circuits (ICs) by supporting early constraint-based design space exploration and timing constraining. These steps of the design methodology are crucial from the perspective of quality of results and are not directly addressed by the synthesis tools used nowadays.

Although the idea of TCT is presented here using one of the currently available logic synthesis tools as an example, it can be easily adapted for other ones. Such flexibility increase usability of TCT and makes it very helpful for scientists who look for new integrated architectures that utilize dynamically reconfigurable resources.

1. Introduction

Unique features of Dynamically Reconfigurable (DR) circuits give opportunity to optimize performance (execution time of complex tasks and power dissipation) of integrated computational architectures due to better utilization of available hardware resources. For that reason integrated computational systems based on DR hardware are very attractive field of study for researchers [1]. Many of scientific publications show that such systems have great potential to compete with or even replace general-purpose processors in diverse implementations [2–11].

Most of the above-referenced systems employ off-the-shelf modular DR ICs, commonly known as Field-Programmable Gate Array (FPGA). It is reported that non-conventional use of these devices requires an effective solution for dealing with their reconfiguration time bottleneck. This goal can be achieved by utilizing sophisticated reconfiguration techniques, like multi-context dynamic reconfiguration [12], and different granularities of reconfigurable matrices. The problem is that the devices equipped with such advanced reconfiguration solutions and diverse granularities have never been available on the market. Therefore, researchers who work on new computational architectures based on DR hardware have to look for possibilities to realize their ideas as custom ICs [1,10,13].

Ubichip [13] constitutes an example of a modular DR IC developed for special scientific purposes. It was designed and fabricated for the PERPLEXUS project carried out under the European Commission's Sixth Framework Programme [14]. During implementation of Ubichip, the design team was forced to develop their own design flow which employs hierarchical approach along with cloning techniques [15]. This approach solves the critical problems related to Static Timing Analysis (STA) [16] and it enabled the designers to lead the project to the successful end. But one of the specific features of the developed flow is that it makes the design very sensitive to the quality of a floorplan. This is a typical problem for the hierarchical designs [17,18].

In case of FPGA-like architectures most part of a chip floorplan is composed of a matrix of identical Reconfigurable Module (RM). Since RMs constitute partitions to be cloned in the chip layout [15], the size and shape of these logic blocks significantly affect the design – its timings, size and power consumption. Therefore, the designers need something more than general early chip estimations in such cases. They have to explore the design space of RMs before the start of the floorplan preparation in order to carry out the whole process effectively.

The early constraint-based design space exploration of a logic block is an iterative process, which consists in determining design space points by manipulating timing constraints [16,19]. Each point is composed of three values: worst path delay, silicon area utilization and power consumption of the circuit. The result of this process is a set of design space points with timing constraints correlated to them.

Determination of the worst path delay (timing characterization) of an RM requires analysis of path delays in relation to each of its configuration modes. It can be realized with every kind of simulations which allow collecting data on module timings, for example with back-annotated functional simulations or STA. The first approach seems to be a natural candidate for this purpose.

* Corresponding author.
E-mail addresses: pamrozik@dmcs.p.lodz.pl (P. Amrozik), napier@dmcs.p.lodz.pl (A. Napieralski).

0026-2692/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.mejo.2013.11.014

However, it requires additional effort from designers (i.a. preparation of appropriate testbenches) and it has to be done using dedicated software, outside of the synthesis tools. Therefore, back-annotated functional simulations are not suitable for early constraint-based design space exploration of the design. Fortunately, STA, which is used by every synthesis tool, can be employed for this purpose as well. Such approach gives significant advantage since it allows performing timing characterization of a logic block during its synthesis. In consequence, it enables to explore a design space of the module in one logic synthesis session.

Timing characterization with STA of an RM requires detailed analysis of reports produced by STA what is a quite laborious task. Moreover, the architecture of reconfigurable logic blocks causes serious problems for STA engine [15]. These obstacles can be overcame by appropriate timing constraining of the circuit. Therefore, the matter of timing constraining of RMs for their timing characterization with STA is very important and requires careful consideration – it will be discussed in detail further in this paper.

Concluding, an early constraint-based design space exploration and timing constraining of RMs are crucial moments during the logic synthesis stage of modular reconfigurable IC design. These steps of the methodology are very time-consuming and quite difficult even for experienced designers. Unfortunately, up to now, there have not been any Electronic Design Automation (EDA) tools that support designers in that field. Constant interest of researchers in integrated DR systems and authors' experience gained during the design of Ubichip became the motivation for starting the work on Timing Characterization and constraining Tool (TCT) – the tool which is intended to make the designers' life easier and encourage researchers to implement custom modular DR ICs.

In the following part of this paper the idea of TCT is presented. In Section 2 the general assumptions regarding the tool are listed and discussed. This section is an introduction to the description of TCT. Section 3 addresses the timing constraining issues of RMs. It presents single-mode and multi-mode timing constraining techniques and their influence on results produced by STA from the perspective of RM timing characterization. Section 4 contains detailed description of TCT and its influence on the design methodology of modular reconfigurable ICs. In Section 5 the results of TCT verification are presented and discussed. Finally, at the end the conclusions are enclosed.

2. TCT – general assumptions

Taking into consideration problems related to the modular reconfigurable IC design methodology and the state-of-the-art EDA tools, the authors assumed that TCT should:

1. combine logic synthesis with design space exploration of an RM;
2. not be a stand-alone application, but extend capabilities of the existing logic synthesis tools, by cooperating with them to cover the early stage of the design methodology;
3. operate relying only on data carried by standard technology libraries and input data provided by designers;
4. automatically generate a set of timing constraints in a commonly used industrial format like Synopsys Design Constraints (SDCs) [19] for each of determined design space points;
5. automatically explore the design space by manipulating of the timing constraints;
6. provide reports on the design space exploration results including timing characterization;
7. generate results in a reasonable period of time.

Commonly used today logic synthesis tools, like CADENCE Encounter RTL Compiler [20], provide commands giving access to the design database created during synthesis session. Therefore, development of a new logic synthesis tool in order to enable constraint-based design space exploration of RMs is not necessary. It is better to implement a custom tool based on Tool Command Language (TCL) that utilizes access to the design database and in this manner support reconfigurable circuit design (as it is stated in points 1 and 2).

Regarding the 3rd point, the input data for the tool should be simpler and easier-to-prepare than a set of timing constraints in e.g. SDC standard. Thus, it should rely on designer's knowledge about RM architecture and do not require detailed knowledge of timing constraints.

Timing constraints have direct impact on synthesis results [16], so the designer must be sure of their reliability. Furthermore, the timing constraints have to be set using appropriate technique which enables exploitation of STA for the early constraint-based design space exploration (particularly timing characterization). In order to perform this task, the tool should automatically operate the timing constraints. Moreover, it should save them for each of designated design space points using commonly used format (according to the points 4 and 5). SDC format is most suitable since it is a well-known industrial standard exploited by IC EDA tools of most vendors.

According to the point 6, the results of the design space exploration should be reported using format which is readable by human and which can be easily converted into graphs. It is important since TCT is intended to give the designer a clear view on the design from the perspective of a floorplan.

The tool should also generate the results in reasonable time. Manually performed design space exploration can take up to a few person months (depending on design team experience and complication level of the design). Therefore, shortening the time needed for this operation to several hours would be a great achievement. Moreover, if the results can be produced quite fast and the input file is not complicated to prepare, the tool can be used for early predictions which simplifies the choice of a fabrication process and facilitate fulfillment of the design specification.

3. Constraining methods of RMs

To start the discussion, let us analyze the simple RM depicted in Fig. 1. It can be considered as a typical reconfigurable logic block similar in its structure to ones used in available FPGA devices. This simplification is introduced to facilitate the discussion by limiting the number of timing paths to be taken into account. It does not omit any important issues from the perspective of timing constraining.

The considered RM has two input ports: IN and MODE, and one output port called OUT. From the module behavior point of view, IN and OUT ports belong to data-flow paths. Through the MODE port, the configuration bit can be stored in the module

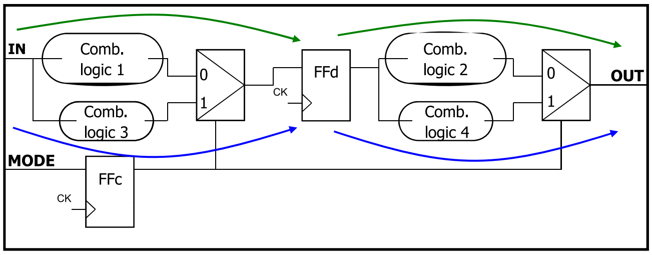

Fig. 1. Example of a Reconfigurable Module (RM).