Partial Run-Time Reconfiguration of a Bit-Serial Architecture

Achim Rettberg[1], Florian Dittmann[2], Stefan Frank[2], Elmar Weber[2], Raphael Weber[1]

achim@c-lab.de, roichen@upb.de

[1]*C-LAB – University Paderborn – Germany*

http://www.c-lab.de

[2]*Heinz Nixdorf Institute – University Paderborn – Germany*

http://www.upb.de

Abstract

The growing need for application class specific but still flexible data processing leads to a demand of new computer architectures. Reorganziation and combination of proven design paradigms are promising ways to reach these goals. The fully re-configurable self-timed bit-serial and fully interlocked MACT architecture is one of those new architectures. In combination with a partial run-time reconfiguration design tool PART-E, the MACT High-Level Synthesis Tool (MHLS) is extended. PART-E automates the generation of partial bit streams. It provides several views of the design that are specified to visualize the temporal and spatial domains of reconfigurable systems..

1. Introduction

The combination of the fully re-configurable self-timed bit-serial and fully interlocked MACT architecture with the partial reconfiguration tool PART-E can be used to implement systems that adapt their execution area over time. However, systems that exploit run-time reconfiguration are still rarely found. One reason is the missing tool support, which is targeted by the PART-E tool.

2. MACT High-Level Synthesis

The The MHLS tool starts with the scheduling phase of the dataflow graph. Due to the bit-serial characteristic of MACT (see [1], [2], [3] and [4]) the allocation and binding phase is straight forwards, because all operations within the dataflow graph are directly mapped to real resources. Furthermore, all additional control wires of MACT are generated during the synthesis. The MHLS user can draw the dataflow graph with the help of basic operations provided by the tool. This basic operations stored in a library can easily be extended by the user.

Besides the graphical input it is also possible to use the textual interface. Therefore, a grammar is defined. Additional information during the synthesis process of the design are displayed in the console window. An extension to generate the dataflow graph from a hardware description language (HDL) like SystemC is right now under development. The output of MHLS was synthesizable VHDL code. A screenshot of the MHLS tool is depicted in Figure 1.

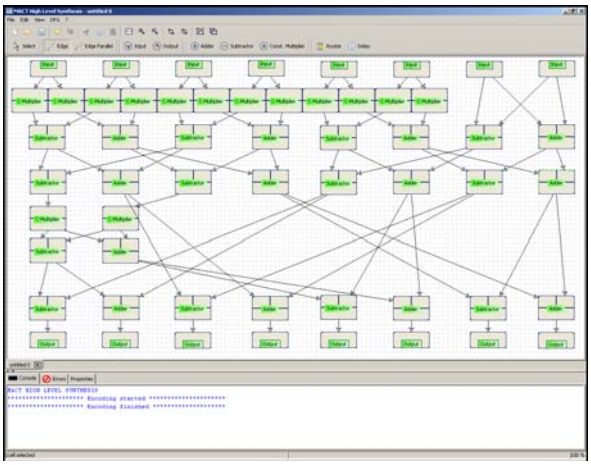

Figure 1: The MHLS Tool

3. The Tool PART-E

PART-E was developed to provide an integrated means of designing partially reconfigurable systems, including the generation of partial bitstreams (see [5] and [6]). The tool was implemented relying on one single model as backbone. All design explorations are performed on this single model. Thus, we can ease the design process despite the complexity of partially reconfigurable systems.

The tool enables to combine HDL-Modules, which each represent a task to be executed dynamically on the FPGA, to so-called Tops, which assemble and structure the design including global logic, etc. Each Top comprises the set of Modules that can be present on an FPGA at the same time. Additionally, physical constraints like the position of the tasks on the FPGA are necessary. The tool eases the generation of those physical constraints, including the placement of inter-module connections. The properties are displayed in the several views, all referring to the same model. A screenshot from the tool is shown in Figure 2.

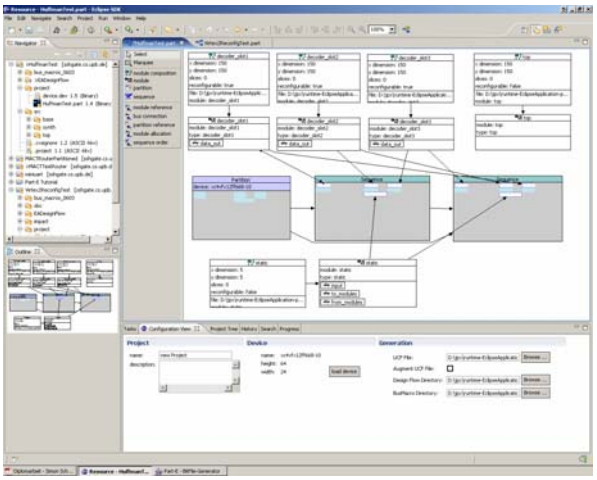

Figure 2: The PART-E Tool

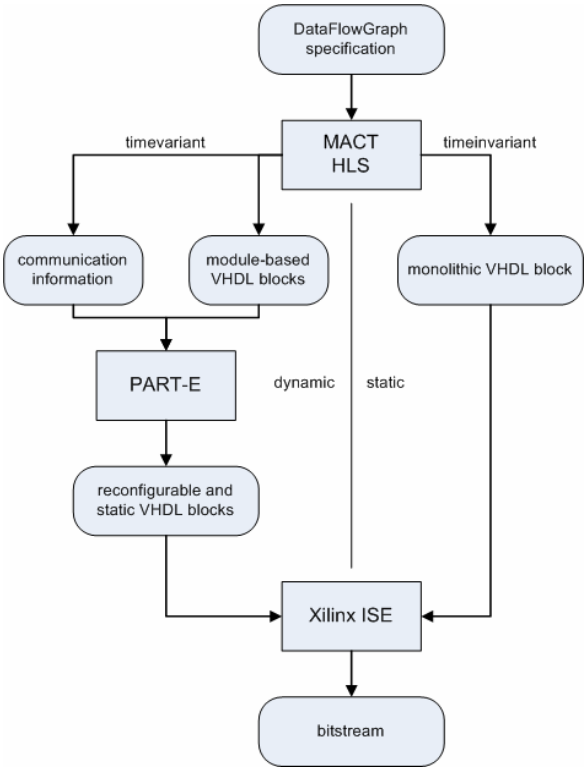

Figure 3: Design Flow

4. Coupling MHLS and PART-E

In MHLS schedules the dataflow graph in time-invariant mode into one monolithic VHDL block, see Figure 3. From this VHDL block Xilinx ISE generates a bitstream, but without the ability with dynamic reconfiguration. In timevariant mode, MHLS divides the design into module blocks also in VHDL code. In this case communication information is generated that describes the interconnection between the different modules.

The modules together with the communication information are handed over to PART-E. PART-E identifies static and reconfigurable blocks, and allows for an automated generation of partial bitstreams by activating the Xilinx ISE tools based on the EarlyAccess design flow. Thus, a seamless integration was realized.

5. References

[1] Dittmann, Florian; Rettberg, Achim; Lehmann, Thomas; Zanella, Mauro C.: Invariants for Distributed Local Control Elements of a New Synchronous Bit-Serial Architecture. In: Second IEEE International Workshop on Electronic Desing, Test and Applications (DELTA 2004), pp. 245-250, Perth, Western Australia, 28 - 30 Jan 2004.

[2] Dittmann, Florian; Rettberg, Achim; Weber, Raphael: Path Concepts for a Reconfigurable Bit-Serial Synchronous Architecture. In: Proceedings of the 2005 IFIP International Conference on Embedded And Ubiquitous Computing (EUC'2005), Nagasaki, Japan, 6 - 9 Dec 2005

[3] Rettberg, Achim; Lehmann, Thomas; Zanella, Mauro C.; Bobda, Christophe: Selbststeuernde rekonfigurierbare bit-serielle Pipelinearchitektur. Dec 2004.

[4] Rettberg, Achim; Zanella, Mauro C.; Bobda, Christophe; Lehmann, Thomas: A Fully Self-Timed Bit-Serial Pipeline Architecture for Embedded Systems. In: Proceedings of the Design Automation and Test Conference (DATE), Messe Munich, Munich, Germany, 3 - 7 Mar 2003.

[5] Dittmann, Florian; Rettberg, Achim: Design of Partially Reconfigurable Systems: From Abstract Modeling to Practical Realization. In: Proceedings of the 1st International Workshop on Reconfigurable Computing Education, Karlsruhe, Germany, 1 Jan 2006.

[6] Dittmann, Florian; Rettberg, Achim; Schulte, Fabian: A Y-Chart Based Tool for Reconfigurable System Design. In: Workshop on Dynamically Reconfigurable Systems (DRS), pp. 67-73, Innsbruck, Austria, 17 Mar 2005, VDE Verlag.

Acknowledgement

This work was partly funded by the Deutsche Forschungsgemeinschaft (DFG) in SPP 1148-