technology opportunity

Method for fast hardware specialization at run-time: improving the overall area and clock speed of the design

Introduction

The inputs of a parameterizable circuit can be divided into two groups: data inputs and parameter inputs. The former are used to supply the circuit with data; the latter specify how data should be processed. A parameterizable circuit can solve a set of specific problems, one problem for each set of legal parameter values. On the other hand a circuit optimized for one specific problem is often smaller, faster and more energy efficient. Thanks to the inherent reconfigurability of SRAM (Static RAM) -based FPGAs (Field Programmable Gate Arrays) it is possible to generate a specific circuit on the fly each time the parameter values change, thus combining the advantages of the specific circuits with the flexibility of the parameterizable circuit. This technique is called run-time hardware generation. Unfortunately, conventional tools are very computationally expensive which leads to large generation times and makes run-time hardware generation useless (inefficient) for most applications.

Technology

Our dual approach solves this problem by separating the hardware generation in two phases: a compile-time preparation step and a very fast run-time finalization step. The compile-time synthesis generates a tunable circuit that can be specialized at run-time to solve every specific problem by merely changing the LUT (Look Up Table) functionalities. This hugely speeds up the run-time generation in two ways. First, since the structural circuit information is fixed for all specific circuits, placement and routing can be done at compile-time. Second, at run-time only the new LUT functionalities have to be calculated for a specific parameter set. This is done in linear time with a very fast constant propagation algorithm.

Applications

Our technology can be used for optimizing all kinds of video processing hardware, routing switches, etc., basically any streaming application for which hardware acceleration is needed and for which the hardware implementation depends on a set of parameters that change their value infrequently. Ultimately, this technique is very useful for applications with a lot of user settings that largely influence the way the application data is treated. We then develop a single generic approach for all possible user settings that is still optimized for specific user settings and is automatically updated to the specific user implementation.

Advantages

- The use of run-time hardware generation reduces the chip area needed to execute a task where part of the input is only dependent on parameters. Theoretical results show an area reduction of 50% for a

multiplier and 66% for a multiplexer. The cost of a system executing a task is proportional to the area needed to execute the task. Therefore run-time hardware generation can greatly reduce system cost.

- The technology only requires conventional design skills, as opposed to the manual design methods which require FPGA and run-time reconfiguration specialists. This brings the design cost of a system using run-time reconfiguration to a level comparable to that of the same system designed without run-time reconfiguration.
- Reusing the placement and routing information that is generated at compile time makes the run-time specialization of the tunable LUT circuit very fast. This was the main disadvantage of previous automated methods.
- Previous methods for run-time hardware generation partially give up place and route quality (speed and power efficiency) to reduce run-time generation overhead. Because in our method placement and routing is done at compile-time we can use the tools to their full extent. This leads to circuits with a very good quality
- The invention can be built into an existing FPGA toolchain. This is relatively simple because only a slight modification of the existing mapping tools is necessary.

Status of development

The technique has been implemented in the lab for such applications as FIR filtering, AES decoding, Ternary Content Addressable Memories, and Network Intrusion Detection.

Intellectual property

European patent application EP2304622, filed on 15 May 2009
Granted US patent US8347243, filed on 15 May 2009, granted on 1 January 2013

Figure

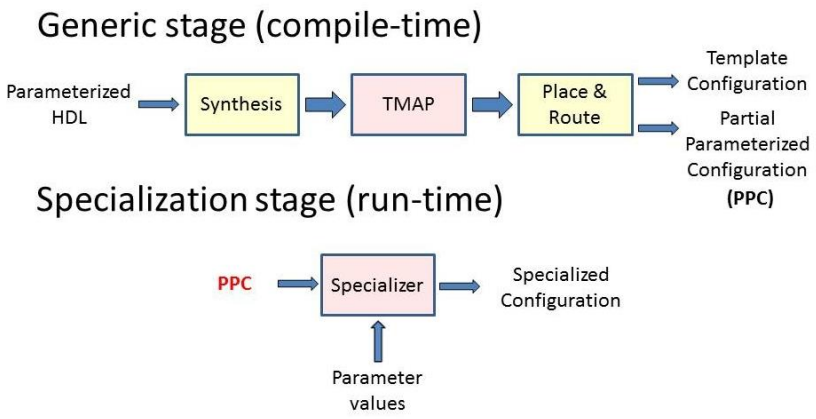

Figure 1: A Partial Parameterized Configuration (PPC) acts as the intermediate layer between the generic stage and specialization stage.

technology opportunity

The Inventor

Karel Bruneel, ELIS Department, Ghent University

References

Karel Bruneel, Wim Heirman and Dirk Stroobandt Dynamic data folding with parameterizable FPGA configurations ACM TRANSACTIONS ON DESIGN AUTOMATION OF ELECTRONIC SYSTEMS, Vol. 16(4), pp. 29 (2011).

Keywords

FPGA, cost-efficient implementation, automatic run-time reconfiguration

Contact

Prof. Dirk Stroobandt, tel.: +32 9 264 34 01, Dirk.Stroobandt@UGent.be
Piet De Vos, PhD, Licensing Manager, +32 9 264 34 01. Piet.Devos@UGent.be