An integrated environment for HW/SW co-design based on a CAL specification and HW/SW code generators
Ghislain Roquier, Christophe Lucarz, Marco Mattavelli, Matthieu Wipliez, Mickaël Raulet, Jörn W. Janneck, Ian D. Miller, David B. Parlour

To cite this version:

HAL Id: hal-00440497
https://hal.archives-ouvertes.fr/hal-00440497
Submitted on 11 Dec 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers. L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Abstract

The possibility of specifying both SW and HW components using the same language is a very attractive design approach. However, despite the efforts spent for implementing such approach using common programming languages such as C and C++, it has not yet shown to be viable and efficient for complex design. The main reason is the difficulty of expressing architectural properties at the level of a common description, difficulty that finally results into the failure of synthesizing efficient HW and efficient parallel SW. This is not the case for dataflow programming that is well-suited to the description of complex real-world computational systems that may contain appreciable amounts of concurrency. A CAL [1, 2] dataflow program is a collection of actors connected by lossless first-in/first-out (FIFO) communication channels, through which they exchange packets of data called tokens. Each actor executes in a sequence of steps, during which it can (1) consume input tokens, (2) produce output tokens, (3) modify its own local state. Actors strictly encapsulate their state, i.e. they cannot directly use or modify the state of other actors. Consequently, executing actors concurrently does not cause race conditions on any state variables, which makes dataflow an excellent (and scalable) parallel programming model. This in turn is key to the generation of efficient HW and SW implementations. This demonstration presents an integrated environment that embeds the synthesis tool that translates a CAL-based dataflow specification into a heterogeneous implementation, composed by HDL and C codes. The demonstration focuses on the capability of the co-design environment to automatically build an executable heterogeneous system implementation running on a platform composed by a processor and a FPGA platform just by annotating the CAL specification. The possibility of direct synthesis from a high level specification is a crucial issue for enabling efficient re-design cycles that include rapid prototyping and validation of performances of the final implementation. The design approach enabled by such integrated environment is particularly suited for development of complex processing systems such as video codecs. As a case study, the demonstration provides the analysis and validation of different SW and HW partitioning of a MPEG-4 Simple Profile decoder.

Introduction

When a high-level CAL system specification is available, such as in the case of a MPEG RVC description [1], the possibility of generating heterogeneous system implementations from such unique specification is a very attractive feature for any system designer. In fact, the decision of partitioning the processing on SW and HW (such as C and VHDL) at a very early stage of the design is simple, but may result not a good option on the final platform. When the development of final SW and HW implementations requires large efforts, and those cycles that result from a new high level SW and HW partitioning are prohibitive. However when synthesis tools for both HW and SW such as Cal2C [3] and CAL2HDL [4] can directly generate an integrated implementation that can be profiled and validated, successful redesign cycles can be efficiently iterated.

The Hardware Platform

The platform with an FPGA and embedded memory providing the APIs from e to the SW components is a PCMCIA WILDCARD™II card. It delivers the processing power of the Xilinx® Virtex™II-P series FPGA (XC2V5000-1I). The Wildcard is provided with a built-in API in order to establish the communication with the FPGA. The API is written in C and has been used to implement the channels of the CAL model that connects actors mapped on the FPGA with the one mapped on the SW on the PC. Inside the FPGA the following components establish the communication:

- INPUT blocks which implement the connections between the card bus and the FPGA (LAD bus) and the generated RTL code.
- OUTPUT blocks which implement the connections between the generated RTL code and the LAD bus.
- The generated RTL code mapped on the FPGA CLBs and the RTL code generated implementing the actors mapped on the FPGA can execute their tasks.

An input block for each input port of the corresponding CAL model actor is instantiated on the FPGA as well as an OUTPUT block for each output port of the corresponding CAL model. The function of the INPUT block is to feed the RTL with the data coming from the local bus which is driven by the function WC_PeRegWrite() in order to send data tokens to the card. The function of the OUTPUT block is to send the data resulting from the execution of the actions to the local bus (LAD Bus) so that the API can transmit the data back into the C program using the function WC_PeRegRead().

The Demonstration

The demonstration will show examples of partitions into SW and HW components with associated synthesis to SW and HW using the OpenDF integrated environment for the design case of a MPEG-4 SP decoder.

REFERENCES