Holistic Approach to Design Heterogeneous Reconfigurable Systems

Muhammad Rashid*† and Bernard Pottier† (PhD Advisor)
* Thomson Research and Development, Compression Lab, Cesson Sevigne, France.
Email: muhamad.rashid@thomson.net,
Estimated Graduation Date: June 2009.
† Université de Bretagne Occidentale, CNRS, UMR 3192, Lab-STICC, Brest, France.

I. Introduction

Existing hardware software (HW-SW) co-design methodologies mainly focus on HW-SW co-simulation to build a virtual prototyping environment that enables software design and system verification without need of making a hardware prototype. However, it fails to reduce the gape between design productivity and system complexity. It is required to target additional design problems such as design space exploration, performance estimation and system synthesis. One commercial example of such a co-design environment is LISATeK from CoWare. The evaluation results of LISATeK design environment for multimedia application are presented in [1] and [2].

In this research work, we first evaluate an holistic co-design environment called hArtes [3]. It provides a seamless design flow from functional simulation to system synthesis. It consists of three phases: algorithm exploration and translation (AET) phase, design space exploration (DSE) phase and system synthesis (SS) phase. After evaluating the hArtes design flow with H.264 video encoder application [4], we highlight some bottlenecks and propose methodologies in each phase of the design flow. In AET phase, we propose a methodology to transform a sequential C code to a block-diagram data-flow specification [5]. In DSE phase, we propose a performance estimation methodology by performing simulation at instruction level [5] [6]. In SS phase, we present an analytical performance analysis methodology to estimate the expected performance of parallel execution on the target architecture [7] [8].

In addition to our contributions in hArtes design flow, we also present a high level generic application analysis methodology for early design space exploration [9]. Application specification written in a high-level language is translated into the trace tree representation by dynamic analysis of the application. Analysis operations are performed on the trace tree representation to get analysis results. We exploit obtained analysis results and re-formulate the application in form of parallel process networks expressed in AVEL framework [10]. MPEG-2 video decoding application serves as a case study.

II. Evaluation of the hArtes Design Flow

The hArtes -Holistic Approach to Reconfigurable real Time Embedded Systems- design flow [3] addresses the development of an holistic tool-chain for reconfigurable heterogeneous platforms. We evaluate the tools in the DSE phase and the SS phase. The tools in the DSE phase facilitate task partitioning, task optimization and assignment of the tasks to the appropriate hardware element of the target architecture. The tools in the SS phase facilitate the hardware/software co-design of embedded applications and perform compilation and HDL generation. C pragma notations and XML architecture description file are used to share the information between different tools. H.264 video encoder application serves as a case study. The experimental results suggested that the hArtes design flow provided overall application speedup of 3.8 times. The hArtes design flow and evaluation results are described in [4].

III. Proposed Transformation Methodology and Performance Estimation Technique

The AET phase of the hArtes design flow provides a graphical algorithms exploration (GAE) tool to support algorithm exploration and validation through simulation and graphical visualization. Despite of hArtes improved design productivity, the bottleneck lies in capturing the initial algorithm specification in the specified GAE tool. Similarly hArtes DSE phase lacks simulation based design space exploration.

A. Contributions in the hArtes AET phase

We propose a transformation methodology which is based on functions re-organization and variables definitions [5]. In functions re-organization, we perform function sequencing, function combination and function copy techniques such that function blocks in sequential code can be transformed into NUTS (basic elementary unit) of graphical data flow environment. We define three types of variables to communicate data between NUTS: port variables, local variables and state variables. We experiment with H.264 encoding application to demonstrate how the proposed methodology can be applied to a complicated real-life multimedia application [5].

B. Contributions in the hArtes DSE phase

Once reference specification is transformed into data flow specification, we present a design space exploration framework that consists of two design loops: computational architecture selection loop and communication architecture selection loop [6]. Before entering into these loops, it is critical to estimate the performance of function blocks. We propose a performance estimation technique for software function blocks by performing simulation at instruction level [5]. The proposed technique considers the effects of architecture features, compiler optimizations and data dependent behavior of the application. Performance of each function block is measured and stored in a performance estimation library. It is a database to store the performance estimation results of individual function blocks. The information in this library is used in two different ways. First, it is used for component selection and mapping decision for each function block. Second, Once component selection and mapping decision is made, the performance of the entire application is estimated as a linear combination of block performances on the mapped components. Experimental results with H.264 encoding application are provided in [5].

IV. Analytical Performance Estimation for Parallel Execution

In SS phase of the hArtes design flow, the hybrid application execution is performed on the Delft Workbench (an heterogeneous reconfigurable architecture) with PowerPC as GPP [4] In this section, we present an analytical performance analysis methodology for video encoding applications to estimate the expected performance of parallel execution on heterogeneous reconfigurable architectures. We analyze the application and present an application parallelization approach for the target platform. H.264 video encoding application serves as a case study and while the Delft workbench is taken as target platform [7]. We perform quantitative analysis of the H.264 encoding application and formulate the performance estimation with an equation to find speed gain by parallel execution on the target architecture. The equation shows the slow down factors that hinder the speed-up of parallel execution. Considering the communication architecture of the target architecture, we parallelize the H.264 encoding algorithm at macroblock level. The details are presented in [7]. We propose a high level design for H.264 encoding application based on performance estimation results [8].

V. Analysis based Design Space Exploration

This section presents a generic application analysis approach for early design space exploration [9]. The objective is to characterizes the source specification at a higher abstraction level without any architectural directives. The input specification is transformed into internal trace tree representation by dynamic analysis of source specification and analysis operations are performed on trace tree representation. Trace tree contains information about execution of application at run time representing implementation independent application characteristics. Once source specification is transformed into trace tree representation, we perform operations on trace tree for different types of analysis. The application analysis framework can be extended to fulfill multiple requirements of design space exploration by simply defining new operations on trace tree representation of source specification. Experimental results for MPEG-2 decoder are shown in [9].

We exploit analysis results and describe source specification of application in AVEL framework [10]. We propose AVEL framework to describe the application in form of process networks to perform parallel & distributed computations. It specifies three kinds of processes which are composed hierarchically. The first type is a *Primitive Process* which is the leaf of hierarchy and implements an atomic behavior. The second type is *Node Process* which is a composition of other processes and behaviors. It allows a hierarchical description of the network. The third type is an *Alias Process* which can be declared outside the main process and further reused just by a link to its name. MPEG-2 decoder is described in AVEL and presented in [10].

VI. Conclusions

We have evaluated the hArtes design flow by taking H.264 video encoder as a case study. We highlighted bottlenecks and proposed solutions accordingly in each step of the hArtes design flow. In AET phase, a transformation methodology was proposed to convert reference sequential C code to data flow specifications. In DSE phase, we proposed a performance estimation technique for fast design space exploration by instruction set simulation. In SS phase, an analytical approach of performance estimation for parallel execution on reconfigurable architectures was proposed. In addition to hArtes design methodology and associated contributions in each phase of the design flow, we also proposed a high level generic application analysis methodology for early design space exploration. MPEG-2 video decoder served as a case study.

References

[1] Muhammad Rashid, Ludovic Apvrille and Renaud Pacalet, “Application Specific Processors for Multimedia Applications.”, In *11th IEEE International Conference on CSE*, pp. 109-116, Sao Paulo, Brazil, 2008.

[2] Muhammad Rashid, Ludovic Apvrille and Renaud Pacalet, “Evaluation of ASIPs Design with LISATek”, In *SAMOS 2008*, LNCS, Vol. 5114/2008, pp: 177-186, 2008.

[3] hArtes Design Methodology: http://www.hartes.org/

[4] Muhammad Rashid, Fabrizio Ferrandi and Koen Bertel, “Design Space Exploration and System Synthesis in the hArtes Design Flow,” Submitted to *RTAS, 2009*.

[5] Muhammad Rashid, Fabrice Urban and Bernard Pottier, “Capturing Data Flow Specifications and Performance Estimation for Fast Design Space Exploration“, Submitted to *ACM Transactions: Special Issue on Model Driven Embedded Systems Design, 2009*.

[6] Muhammad Rashid, Fabrice Urban and Bernard Pottier, “Performance Estimation by Instruction Set Simulation for Fast Design Space Exploration“, Submitted to *ARC 2009*.

[7] Muhammad Rashid, Jean-Christophe Le Lann and Koen Bertels, “Video Encoding Analysis for Parallel Execution on Reconfigurable Architectures”, In *6th Symposium on Design, Analysis and Simulation of Distributed Systems*, Edinburgh UK, 2008.

[8] Muhammad Rashid, “A High Level Design based on Performance Estimation Methodology for Reconfigurable Architectures”, In *4th International Workshop on Reconfigurable Communication Centric System-on-Chips*, Spain 2008.

[9] Muhammad Rashid, Thiery Goubier and Bernard Pottier, “A High Level Generic Application Analysis Mthodology for Early Design Space Exploration”, In *Workshop on Design and Architectures for Signal and Image Processing*, France, 2007.

[10] Muhammad Rashid, Damien Picard, Bernard Pottier “Application Analysis for Parallel Processing”, In *11th EuroMicro Conference on Digital System Design, Architectures, Methods and Tools*, Parma, Italy, 2008.