2011 19th International Euromicro Conference on Parallel, Distributed and Network-Based Processing

Issues and Challenges in Development of Massively-Parallel Heterogeneous MPSoCs Based on Adaptable ASIPs

Lech Jóźwiak
Eindhoven University of Technology
Eindhoven, The Netherlands
L.Jozwiak@tue.nl

Menno Lindwer
Silicon Hive B.V.
Eindhoven, The Netherlands
Menno.Lindwer@siliconhive.com

***Abstract*—The recent spectacular progress in modern nanoelectronic technology enabled implementation of very complex multiprocessor systems on single chips (MPSoCs) and created a big stimulus towards development of MPSoCs for embedded applications. The increasingly complex MPSoCs are required to perform real-time computations to extremely tight schedules and to satisfy high demands regarding adaptability, as well as energy, area and cost efficiency. This results in serious design and development challenges. The opportunities created can effectively be exploited only through use of more adequate system architectures and more integrated system IP modules, supported by new effective design methods and electronic design automation tools. This paper focuses on mastering the automatic architecture synthesis and application mapping for heterogeneous massively-parallel MPSoCs based on customizable application-specific instruction-set processors (ASIPs). It is related to a European project ASAM being currently executed in the framework of the ARTEMIS program. It presents the results of our analysis of the main problems that have to be solved and challenges to be faced in design of such heterogeneous customizable MPSoCs for modern demanding applications.**

Keywords- embedded systems, heterogeneous multi-processor system-on-chip (MPSoC), customizable ASIPs, architecture synthesis, MPSoC and ASIP design automation;

I. INTRODUCTION

The recent spectacular progress in modern nano-electronic technology enabled implementation of very complex multiprocessor systems on single chips (MPSoCs) and created a big push towards development of various kinds of high-performance embedded systems. An embedded system serves a specific aim in a certain larger embedding system through repeatedly executing specific computation processes required by its application. It has to be especially designed to adequately serve the execution of these specific computation processes and satisfy application requirements related to such attributes as functional behavior, reaction speed or throughput, energy consumption, geometrical dimensions, price etc. The recent progress in nano-electronic technology and new applications it enabled create a situation in which increasingly complex and sophisticated embedded systems are required to perform real-time computations to extremely tight schedules and to satisfy high demands regarding energy, power, area, and cost efficiency. Moreover, many embedded systems are required to be flexible enough to enable adequate reuse among different product versions, reaction to market shifts, adherence to evolving standards or user requirements, and easy modification during their development or even their field use. This all results in serious design and development challenges, such as: multi-objective MPSoC optimization, adequate resolution of numerous complex design tradeoffs, reduction of the design productivity gap for the increasingly complex and sophisticated systems, reduction of the time-to market and development costs without compromising the system quality, etc. The opportunities created can effectively be exploited only through use of more adequate application-specific system architectures and more integrated system IP modules, supported by new design methods and electronic design automation (EDA) tools for an adequate system-level design exploration, rapid development of high-quality hardware platforms, and efficient automatic mapping of applications on the platforms.

This paper addresses the development of heterogeneous MPSoCs based on configurable and extensible application-specific instruction-set processors (ASIPs). The new MPSoC design technology based on adaptable ASIPs addresses several fundamental development challenges of electronic systems for modern highly-demanding applications. It is able to deliver a high performance, high flexibility and low energy consumption at the same time. It is relevant for a very broad range of applications and applicable to several implementation technologies. MPSoCs based on this technology can be built at substantially lower costs and with much shorter times to market than hardwired ASICs, and can also provide the flexibility required for engineering of robust, context-aware and adaptive systems for many application domains. However, despite a decade of research in the field of SoC architecture synthesis, an architecture synthesis methodology and tools for heterogeneous customizable MPSoCs based on adaptable ASIPs have not been built. This paper focuses on mastering the automatic architecture synthesis and application mapping for the customizable ASIP-based MPSoCs. It presents the results of our analysis of the main problems that have to be solved and challenges to be faced in design of such MPSoCs for modern demanding applications, and

1066-6192/11 $26.00 © 2011 IEEE
DOI 10.1109/PDP.2011.55

briefly discusses the approach proposed by us in the recently started European research project ASAM of the ARTEMIS Program.

II. ARCHITECTURE PLATFORM AND APPLICATIONS

The **architecture platform** targeted in the ASAM project is a ***configurable and extensible for specific applications heterogeneous multi-ASIP platform***. In particular, the project targets the MPSoC platforms of its industrial partners involving optimized for various application fields generic customizable ASIPs. Each ASIP is composed of an actual processor core (core) and core I/O (coreio) that together form a VLIW machine capable of executing parallel software with a single thread of control (see Fig.1). It includes a VLIW datapath controlled by a sequencer that uses status and control registers and executes programs from the local program memory. The data path contains functional units organized in several parallel scalar and/or vector issue slots connected via programmable argument and result interconnect networks to registers organized in several register files. The functional units perform computation operations on intermediate data stored in the register files. The coreio provides the local memory and I/O subsystem enabling easy integration of the ASIP in any larger system which has access to the devices in coreio via slave interfaces. The local memories collaboration with particular issue slots enable scalar access for the scalar slots and vector or block access for the vector slots. Both SIMD and MIMD processing can be realized. The ASIPs are configurable and extensible. The numbers, kinds and parameters of functional units, issue slots, register files, memories, interfaces, etc. can freely be selected. Moreover, new functional units, issue slots, etc., specific for a particular application, can be developed and added.

Several such different ASIPs, each customized for a particular part of a complex application, can be interconnected with global memories and other sub-systems using a configurable hierarchical interconnection network, and implemented on one chip together with possible hardware accelerators and other digital or analog sub-systems. Several hundreds of such ASIPs with up to 5000 issue slots in total can be placed on a single chip

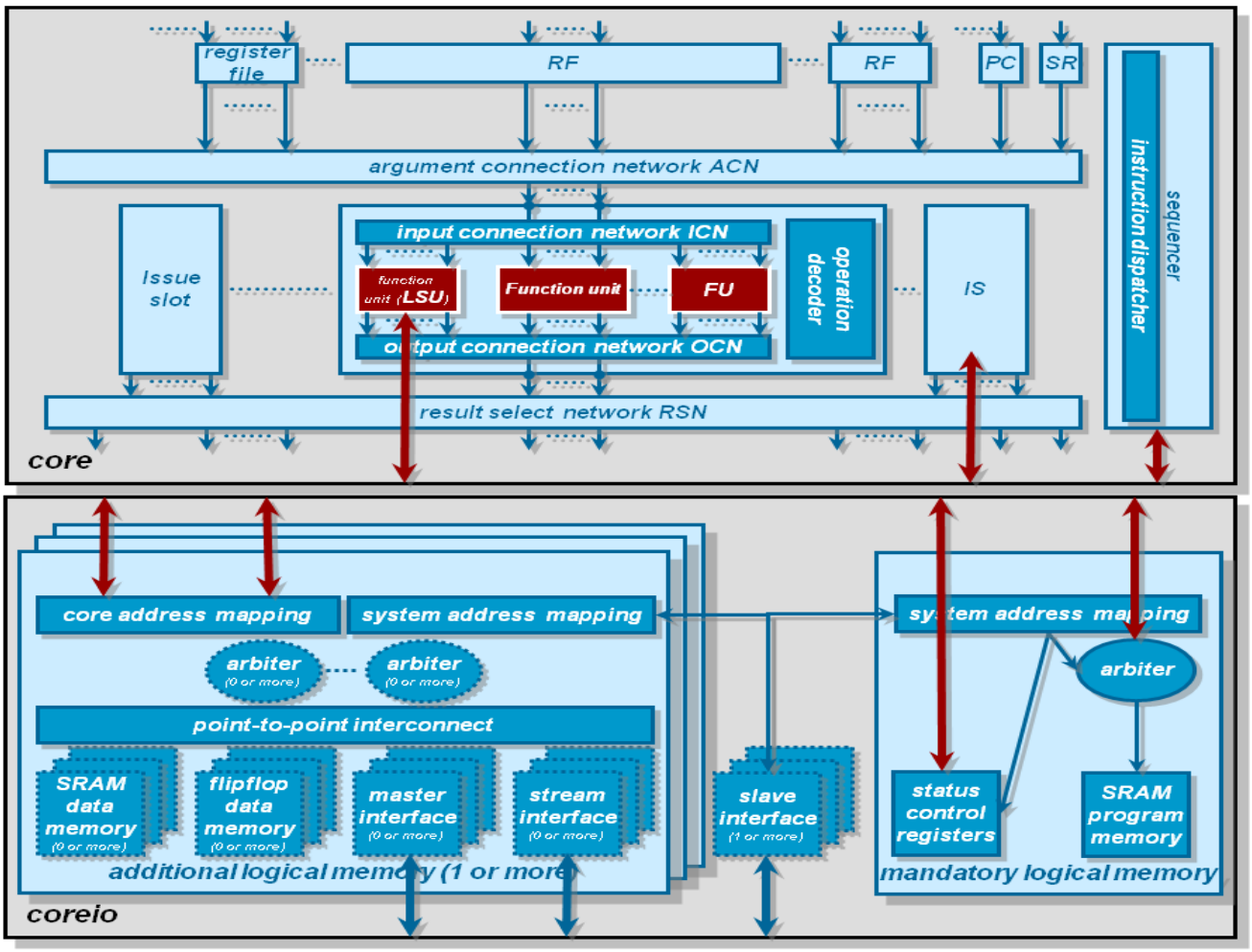

Fig.1.Generic ASIP architecture of the targeted MPSoC platform.

implemented in 28 nm CMOS technology. When operated at 400-600MHz, these ASIPs can deliver up to 2.5 Tops/s with power consumption below 1W.

The customizable for specific applications heterogeneous massively-parallel MPSoC platform based on the configurable and extensible ASIPs enables efficient application-specific exploitation of various kinds of parallelism. While the MPSoC's multiple processors serve coarse-grain parallelism exploitation at the task or sub-program level, the ASIP's parallel issue slots and custom instructions serve the fine-grained acceleration.

The **MPSoC design technology based on adaptable ASIPs** is ***relevant for a very broad range of application domains*** (e.g. telecommunications, networking, multimedia, entertainment, consumer electronics, medical instrumentation, advanced machinery, military applications, automotive applications, etc.) and for the ***cross-domain convergence products***. It is ***applicable to several implementation technologies*** (e.g. SOC or ASIC, structured ASIC, and FPGA). It is especially suitable for complex applications involving different kinds of processing with various kinds of parallelism and being highly demanding regarding physical and economic characteristics. In particular, many of the modern multimedia and communications applications involve sophisticated algorithms of divergent character and require extremely high performances that can only be delivered by application-specific hardware solutions or sophisticated heterogeneous multi-ASIP platforms. At the same time, they often demand adaptable programmable platforms, because of a plethora of existing and emerging standards, and demand for scalable quality levels in transmission and decoding performance.

III. Issues and Challenges in Development of MPSoCs Based on Customizable ASIPs

In the traditional embedded system development approaches, the major development activities are usually largely disjoint and performed by different teams using different supporting tools. This leads to inefficiencies, errors, and costly reiterations in the design process. Moreover, the traditional algorithm and software development approaches require an existing and stable computation platform (HW platform, compilers etc.), while for the modern embedded systems the architecture, algorithms, hardware and software have to be application-specific, and must be developed largely in parallel. In particular, the hardware platform of the application-specific MPSoCs based on adaptable ASIPs has to be developed in parallel with development and mapping of the MPSoC software on this platform. Unfortunately, the efficiency of the required parallel development is much too low with the currently available development technology due to lack of effective automated methods of industrial strength for many MPSoC design problems, and weak interoperability of the architecture, algorithm, HW/SW, and hardware synthesis tools. The inefficiencies identified above result in a substantially lower than attainable quality of the resulting systems, much longer than necessary development time, and much higher development costs.

However, the systemic realizations of complex and highly demanding applications, for which the customizable multi-ASIP MPSoC technology is especially suitable, demand the performance and energy usage levels comparable to those of ASICs, while remaining small-size and cost-effective. Satisfaction of these stringent and often conflicting demands requires efficient exploitation of different kinds of parallelism involved in these applications, implementation of critical parts of their information processing in application-specific hardware and efficient trade-off exploitation among various design characteristics.

To facilitate design of satisfactory systems for the highly demanding applications, the research of the ASAM project aims to develop a complete coherent system-level synthesis flow for MPSoCs based on adaptable ASIPs, involving the application-specific system-level architecture synthesis, module-level micro-architecture synthesis, as well as application and technology-specific hardware and software implementation and optimization, all based on the application analysis. While most of the existing methods and tools for ASIP construction are devoted to a single processor design, in a customizable multi-ASIP MPSoC its various ASIPs have to be customized in combination, and in a strict relation to the selection of the number of ASIPs, as well as to the scheduling and mapping of the application's required computations on the particular ASIPs. The MPSoC macro-architecture and the micro-architectures of its particular ASIPs are strictly interrelated. Important trade-offs have to be resolved regarding the granularity of individual processing cells, and between the amount of parallelism and resources at each of the two architecture levels (e.g. similar performance can be achieved with fewer processors, each being more parallel and better targeted to particular part of application, as with more processors, each being less parallel and less application-specific). A lot of communication is required during the multi-ASIP MPSoC synthesis between the macro- and micro-architecture synthesis tasks to arrive at a satisfactory MPSoC architecture. In consequence, optimization of the performance/resources trade-off required by a particular application can only be achieved through a careful construction of a certain application-specific macro-/micro-architecture combination. The aim here is thus to find an adequate balance between the number of parallel processors, the complexity of the inter-processor communication, and the intra-processor parallelism and complexity. To achieve this aim several promising macro-architecture/micro-architecture combinations have to be considered in an iterative way, and finally, the best of them has to be selected for an actual realization.

Most of the existing methods and tools of custom instruction-set construction are devoted to optimization of a single processor and are very simplistic. In most cases, the construction is based on some proxy attributes and optimization objectives, on a kind of simplified application

analysis, and performed without actually accounting for the related data-path and control-path implementation. For instance, it is usually performed without accounting for the effects and trade-offs of hardware sharing by various instructions. In result, the proxy formulations of the custom instruction set construction problems and their suggested solutions do not reflect well the actual problems to be solved and their required solutions. Additionally to the instruction-set selection, there is a plethora of other architectural choices on the micro-architectural level, as the number of parallel instructions and issue slots, grouping of issue slots into clusters, number of register files, ports and registers, interconnects between clusters, to name some of them.

Moreover, many modern applications (e.g. various communication, multimedia, networking or encryption applications) require hardware implementation of algorithms that involve complex interrelationships between the data and computing operations. This can result in complex memory accesses and complex communication between the memories and computing units in the related hardware. For applications of this kind, the main design problems are related to an adequate resolution of memory and communication bottlenecks and to decreasing the memory and communication hardware complexity, which has to be achieved through an adequate memory and communication structure design. Moreover, for applications of this kind, complex interrelationships exist between the computing unit design and the corresponding memory and communication structure design. The memory and communication structure design, and micro-architecture design for computing units cannot be performed independently, because they substantially influence each other. For example, exploitation of more data parallelism in a computing unit micro-architecture usually requires getting the data in parallel for processing, i.e. having simultaneous access to memories in which the data reside and simultaneous transmission of the data.

IV. AIM AND APPROACH OF THE PROJECT

The research of the ASAM project builds on the platform-based design of heterogeneous multi-processor embedded systems [1][4] (being researched only in the recent years), as well as, hardware compilation techniques [4] (being researched for at least the last two decades, but still requiring much more research), and software analysis, re-structuring and compilation techniques [2][4] (being researched much longer, but still demanding more effective methods, a. o. for application parallelism exploitation).

The main aim of the ASAM project is to much enhance the MPSoC design efficiency, while substantially improving the result quality, through an adequate system and sub-system level design exploration and uniform automatic HW/SW compilation of application specifications into their highly optimized application-specific heterogeneous multi-ASIP implementations. The uniform design flow will provide efficient exploration of the architecture and application mapping alternatives and tradeoffs. The compilation process will efficiently select the most appropriate ASIP processor types for different parts of a given application, reuse and instantiate the selected generic ASIPs, extend them with new hardware implemented as parts of their application-specific data- or control-paths, correspondingly restructure the application's software and implement the software on the so constructed application-specific multi-processor platform.

This aim will be realized through development of **a unified synthesis and prototyping environment** enabling**:**

- ***multi-objective design space exploration for customizable heterogeneous multi-ASIP SoCs and construction of the application-tailored and technology-aware system-level architectures***, including decision on the kind of processors, communication resources and memory hierarchy, their composition, as well as scheduling and mapping of the application computation processes on the hardware platform;
- ***architecture customization of particular generic ASIPs, communication resources and memory structures***, including extension of processors with new application-specific hardware (e.g. new instructions);
- ***adequate hardware synthesis*** of the so designed hardware platform;
- ***software compiler retargeting and (semi-)automatic software restructuring and implementation on the resulting multi-processor platform***;
- ***feedback creation for both architecture design levels*** on the satisfaction of the functional and extra-functional requirements through simulation and FPGA emulation.

This way a consistent highly efficient automatic synthesis flow will be created from the algorithmic specification down to its hardware/software implementation at the circuit/code level. The new coherent automated design environment will enable the system and algorithm designers to perform rapid exploration of the high-level algorithm and architecture solution spaces, and in consequence, quickly develop high-quality designs.

With respect to the MPSoC macro-architecture synthesis, the project builds on the quality-driven model-based system-level design exploration and architecture synthesis approach [3][4], and modelling, emulation, estimation and design exploration concepts [3][4][5][6], earlier developed by some of the project partners. It will develop a new architecture methodology and supporting tools that will enable an effective and efficient reuse of a generic architecture platform and platform modeling in the form of an abstract architecture template, generic architecture template instantiation, abstract requirement modeling, application process scheduling and mapping on the generic architecture template instance when observing the constraints objectives and tradeoffs of the requirement model, and final architecture refinement and optimization (processing, interfacing, and memory abstraction refinement and optimization).

The ASIP micro-architecture synthesis will perform the actual processor data-path and control-path construction when re-using the existing application-class specific generic ASIP IP cores customizable for a specific application. It is perhaps the most difficult task in the design of a system based on customizable ASIPs. It has to identify parts of the application that require specialized hardware through finding operation patterns to be replaced by new instructions, define sequences of operations that can operate independently on separated hardware clusters and local register files, and construct the corresponding instruction set hardware clusters, register files, memories, and other processor parts, when adequately accounting for hardware reuse and satisfying the application's constraints, objectives and trade-offs. The existing commercial and academic developments in this field do not provide adequate support for this critical part of the designers' work (see e.g. [4]). Therefore, new methods will be developed in the scope of the project to much better reflect the actual problems to be solved and their required solutions in the proxy formulations and solutions of the new methods than in it is the case in the existing methods.

As explained in the previous section, there are very strong interrelations between the macro- and micro-architecture synthesis of heterogeneous MPSoCs based on configurable and extendable ASIPs. Therefore, in our architecture synthesis method the macro-architecture and micro-architecture synthesis will represent one coherent complex system architecture synthesis task, and not two separate tasks, as in the state-of-the-art methods. There will be common aims and a strong consistent communication between the two sub-tasks. The macro-architecture synthesis will propose a certain number of customizable ASIP cores of several types to be used and assign a part of the application to be executed on each of the proposed ASIP cores. The micro-architecture synthesis will customize each of the ASIP cores together with its local memories and other blocks to execute the assigned application part as effective and efficient as possible. Subsequently, the RTL-level HDL description of the customized ASIP cores will be automatically generated and synthesized to an actual hardware design. Based on this actual hardware implementation, the micro-architecture synthesis will provide feedback on the physical characteristics of each particular sub-system implemented with each ASIP core to the macro-architecture synthesis. This way the micro-/macro-architecture trade-off exploitation will be enabled, and after several iterations an optimized MPSoC architecture will be constructed.

A complete MPSoC platform architecture involves of course the adequately selected and instantiated processor cores, as well as, adequate memory and interconnection structures. Since many modern applications require hardware implementation of algorithms that involve complex interrelationships between the data and computing operations, our new design environment will account for this through new methods and tools for the application-tailored and technology aware interconnects and memory structures construction. While current research in this area is mainly focused on the design of interconnection systems, this project will also consider the mutual relationships between interconnects, memories and processor IPs. The interconnect and memory structures will be optimized in an iterative refinement process, when accounting for the application-specific memory-processor communication and the technology related interconnect and memory features, such as power dissipation or area. Regarding the interconnect and memory structures the project builds on recent results of one of the project partners [7][8].

V. CONCLUSION

This paper presented the results of our analysis of the main problems and challenges to be faced in the design of heterogeneous customizable ASIP-based MPSoCs for modern demanding applications, and briefly discussed the approach proposed in the recently started European research project ASAM. Further problem analysis and workout of the approach proposed into actual effective and efficient methods and tools will make it possible to improve the MPSoC design efficiency for actual industrial designs.

ACKNOWLEDGMENT

The authors of this paper are indebted to all partners of the ASAM project.

REFERENCES

[1] D. Densmore, A.Sangiovanni-Vincentelli, R. Passerone: A platform-based taxonomy for ESL design, IEEE Des. Test Comput, Vol. 23, No 5, 2006, pp. 359-374

[2] R. Leupers and Peter Marwedel: Retargetable compiler technology for embedded systems - tools and applications, Kluwer, 2001

[3] L. Jóźwiak and S.-A. Ong: Quality-driven Model-based Architecture Synthesis for Real-time Embedded SoCs, Journal of Systems Architecture, Vol. 54, No 3-4, 2008, pp. 349-368

[4] L. Jóźwiak, N. Nedjah and M. Figueroa: Modern Development Methods and Tools for Embedded Reconfigurable Systems - A Survey, Integration - the VLSI Journal, Vol. 43, No 1, 2010, pp. 1-33

[5] S. Mahadevan, F. Angiolini, J. Sparso, L. Benini, J. Madsen: A Reactive and Cycle-True IP Emulator for MPSoC Exploration, IEEE Trans. on CAD, Vol. 27, No 1, 2008, pp. 109-122

[6] A. Tranberg-Hansen, J. Madsen: A Service Based Component Model for Composing and Exploring MPSoC Platforms, Proc. of the First International Symposium on Applied Science in Biomedical and Communication Technologies (ISABEL), October 2008

[7] S.Murali, D.Atienza, P.Meloni, S.Carta, L. Benini, G. De Micheli, L.Raffo. "Synthesis of Predictable Networks-on-Chip Based Interconnect Architectures for Chip Multi-Processors". IEEE Transactions On VLSI, Vol. 15, No 8, 2008, pp. 869 – 880

[8] F.Angiolini, P.Meloni, S.Carta, L.Benini, L.Raffo. "A Layout-Aware Analysis of Networks-on-Chip and Traditional Interconnects for MPSoCs", IEEE Transactions On Computer Aided Design, Vol. 26, 2007, pp. 421-434.