



Activity Report 2013

Team DREAMPAL

Dynamic Reconfigurable Massively Parallel
Architectures and Languages

RESEARCH CENTER
Lille - Nord Europe

THEME
Architecture, Languages and Compila-
tion

Table of contents

1. Members	1
2. Overall Objectives	2
3. Research Program	2
3.1. New Models for New Technologies	2
3.2. Multi-softcore on 3D FPGA	3
3.3. When Hardware Meets Software	4
4. Application Domains	4
4.1. Reflective Camera Networks	4
4.2. Set-top Boxes	5
4.3. Safe and Intelligent Transportation	5
5. Software and Platforms	5
5.1.1. HoMade	5
5.1.2. JHomade	6
5.1.3. \mathbb{K} -based language-independent symbolic execution and verification tool	6
6. New Results	6
6.1. Language-Independent Symbolic Execution, Program Equivalence, and Program Verification	6
6.1.1. Symbolic Execution	7
6.1.2. Program Equivalence	7
6.1.3. Program Verification	7
6.2. Master-Slave Control Structure for MP-SoC Architectures	7
6.3. Toward a Massively Parallel and Reflective Execution Model	8
6.4. Power Estimation at System-Level for MPSoC Based Platforms	8
6.5. Dynamically reconfigurable CPU/FPGA architecture for the testing and simulation of avionic systems	9
6.6. A custom reconfiguration controller for partial and dynamic reconfiguration in HoMade based systems	9
6.7. Hardware control for partially and dynamically reconfigurable systems: from modelling to implementation	9
6.8. A model-based approach for dynamically reconfigurable systems design: from MARTE to RecoMARTE	10
7. Bilateral Contracts and Grants with Industry	10
7.1. Bilateral Contracts with Industry	10
7.2. Bilateral Grants with Industry	11
8. Partnerships and Cooperations	11
8.1. Regional Initiatives	11
8.2. International Initiatives	11
8.3. International Research Visitors	12
8.3.1. Visits of International Scientists	12
8.3.2. Internships	12
8.3.3. Visits to International Teams	12
9. Dissemination	12
9.1. Scientific Animation	12
9.2. Teaching - Supervision - Juries	12
9.2.1. Teaching	12
9.2.2. Committees	13
9.3. Popularization	13
10. Bibliography	13

Team DREAMPAL

Keywords: Embedded Systems, Reconfigurable Hardware, Programming Languages, Formal Methods

Creation of the Team: 2013 January 01.

1. Members

Research Scientist

Vlad Rusu [Team leader, Inria, Researcher, from Jan 2013, HdR]

Faculty Members

Jean-Luc Dekeyser [Univ. Lille I, Professor, from Jan 2013, HdR]

Frédéric Guyomarch [Univ. Lille I, Associate Professor, from Jan 2013]

Philippe Marquet [Univ. Lille I, Associate Professor, from Jan 2013]

Samy Meftali [Univ. Lille I, Associate Professor, from Jan 2013, HdR]

External Collaborator

Rabie Ben Atitallah [Univ. Valenciennes, Associate Professor, from Jan 2013]

PhD Students

George Afonso [Univ. Lille I, CIFRE, from Jan 2013 until Jul 2013]

Karim Ali [Univ. Valenciennes, from Sep 2013]

Andrei Arusoaie [Univ. Iasi Romania (co-supervision), from Jan 2013]

Mokhtar Bouain [Univ. Valenciennes, from Sep 2013]

Sana Cherif [Univ. Lille III, from Jan 2013 until Dec 2013]

Wissem Chouchene [Univ. Lille I, Erasmus Mundus AL Fihri grant, from Sept 2013]

Hana Krichene [Univ. Lille I and ENIS, Sfax, Tunisia, from Jan 2013 until Dec 2013]

Paméla Meftali [Univ. Lille I, from Jan 2013 until June 2013]

Santhosh Kumar Rethinagiri [Univ. Valenciennes, from Jan 2013 until Feb 2013]

Venkatasubramanian Viswanathan [Univ. Lille I, CIFRE, from Jan 2013]

Post-Doctoral Fellows

Chiraz Trabelsi [Univ. Valenciennes, since Sep 2013]

Chadi Aljundi [Univ. Lille1, since Sep 2013]

Visiting Scientists

Stefan Ciobaca [[Univ. Lille I, from Jul 2013 until Jul 2013]

Dorel Lucanu [Univ. Lille I, from Jul 2013 until Jul 2013]

Administrative Assistant

Corinne Jamroz [Inria]

Other

Kanwarjeet Dhaliwal [Inria, intern, from May 2013 until Aug 2013]

2. Overall Objectives

2.1. Executive Summary

Standard Integrated Circuits are reaching their limits and need to evolve in order to meet the requirements of next-generation computing. We anticipate that FPGAs (Field Programmable Gate Arrays) will play a major role in this evolution: FPGAs are currently only one or two generations behind the most advanced technologies for standard processors, and their application-specific hardware is an order of magnitude faster than software solutions on standard processors. One of the most promising evolutions are next-generation 3D-FPGAs, which, thanks to their fast reconfiguration and inherent parallelism, will enable users to build dynamically reconfigurable, massively parallel hardware architectures around them. This new paradigm opens many opportunities for research, since, to our best knowledge, there are no methodologies for building such architectures, and there are no dedicated languages for programming on them.

We shall thus address the following topics: proposing an execution model and a design environment, in which users can build customized massively parallel dynamically reconfigurable hardware architectures, benefiting from the reconfiguration speed and parallelism of 3D-FPGAs; proposing dedicated languages for programming applications on such architectures; and designing software engineering tools for those languages: compilers, simulators, and formal verifiers. The overall objective is to enable an efficient and safe programming on the customized architectures. Our target application domain are embedded systems performing intensive signal/image processing (e.g., smart cameras, radars, and set-top boxes)

3. Research Program

3.1. New Models for New Technologies

Over the past 25 years there have been several hardware-architecture generations dedicated to massively parallel computing. We have contributed to them in the past, and shall continue doing so in the Dreampal project. The three generations, chronologically ordered, are:

- Supercomputers from the 80s and 90s, based on massively parallel architectures that are more or less distributed (from the Cray T3D or Connection Machine CM2 to GRID 5000). Computer scientists have proposed methods and tools for mapping sequential algorithms to those parallel architectures in order to extract maximum power from them. We have contributed in this area in the past: <http://www.lifl.fr/west/team.html>.
- Parallelism pervades the chips! A new challenge appears: hardware/software co-design, in order to obtain performance gains by designing algorithms together with the parallel architectures of chips adapted to the algorithms. During the previous decade many studies, including ours in the Inria DaRT team, were dedicated to this type of co-design. DaRT has contributed to the development of the OMG MARTE standard (<http://www.omgmarTE.org>) and to its implementation on several parallel platforms. Gaspard2, our implementation of this concept, was identified as one of the key software tools developed at Inria: <http://www.inria.fr/en/centre/lille/research/platforms-and-flagship-software/flagship-software>.
- The new challenge of the 2010s is, in our opinion, the integration of dynamic reconfiguration and massive parallelism. New circuits with high-density integration and supporting dynamic hardware reconfiguration have been proposed. In such architectures one can dynamically change the architecture while an algorithm is running on it. The Dynamic Partial Reconfiguration (DPR) feature offered by recent FPGA boards even allows, in theory, to generate optimized hardware at runtime, by adding, removing, and replacing components on a by-need basis. This integration of dynamic reconfiguration and massive parallelism induces a new degree of complexity, which we, as computer scientists, need to understand and deal with in order to make possible the design of applications running on such architectures. This is the main challenge that we address in the Dreampal project. We note that we address these problems as computer scientists; we do, however, collaborate with electronics specialists in order to benefit from their expertise in 3-D FPGAs.

Excerpt from the HiPEAC vision 2011/12

“The advent of 3D stacking enables higher levels of integration and reduced costs for off-chip communications. The overall complexity is managed due to the separation in different dies, independently designed.”

FPGAs (Field Programmable Gate Arrays) are configurable circuits that have emerged as a privileged target platform for intensive signal processing applications. FPGAs take advantage of the latest technological developments in circuits. For example, the Virtex7 from Xilinx offers a 28-nanometer integration, which is only one or two generations behind the latest general-purpose processors. 3D-Stacked Integrated Circuits (3D SICs) consist of two or more conventional 2D circuits stacked on the top of each other and built into the same IC. Recently, 3D SICs have been released by Xilinx for the Virtex 7 FPGA family. 3D integration will vastly increase the integration capabilities of FPGA circuits. The convergence of massive parallelism and dynamic reconfiguration is inevitable: we believe it is one of the main challenges in computing for the current decade.

By incorporating the configuration and/or data/program memory on the top of the FPGA fabric, with fast and numerous connections between memory and elementary logic blocks (~10000 connections between dies), it will be possible to obtain dynamically reconfigurable computing platforms with a very high reconfiguration rate. Such a rate was not possible before, due to the serial nature of the interface between the configuration memory and the FPGA fabric itself. The FPGA technology also enables massively parallel architectures due to the large number of programmable logic fabrics available on the chip. For instance, Xilinx demonstrated 3600 8-bit picoBlaze softcore processors running simultaneously on the Virtex-7 2000T FPGA. For specific applications, picoBlaze can be replaced by specialized hardware accelerators or other IPs (Intellectual Property) components. This opens the possibility of creating massively parallel IP-based machines.

3.2. Multi-softcore on 3D FPGA

From the 2010 Xilinx white paper on FPGAs:

“Unlike a processor, in which architecture of the ALU is fixed and designed in a general-purpose manner to execute various operations, the CLBs (configurable logic blocks) can be programmed with just the operations needed by the application... The FPGA architecture provides the flexibility to create a massive array of application-specific ALUs..The new solution enables high-bandwidth connectivity between multiple die by providing a much greater number of connections... enabling the integration of massive quantities of interconnect logic resources within a single package”

Softcore processors are processors implemented using hardware synthesis. Proprietary solutions include PicoBlaze, MicroBlaze, Nios, and Nios II; open-source solutions include Leon, OpenRisk, and FC16. The choice is wide and many new solutions emerge, including multi-softcore implementations on FPGAs. An alternative to softcores are hardware accelerators on FPGAs, which are dedicated circuits that are an order of magnitude faster than softcores. Between these two approaches, there are other various approaches that connect IPs to softcores, in which, the processor’s machine-code language is extended, and IP invocations become new instructions. We envisage a new class of softcores (we call them reflective softcores ¹), where almost everything is implemented in IPs; only the control flow is assigned to the softcore itself. The partial dynamic reconfiguration of next-generation FPGAs makes such dynamic IP management possible in practice. We believe that efficient reflective softcores on the new 3D-FPGAs should be as small as possible: low-performance generic hardware components (ALU, registers, memory, I/O...) should be replaced by dedicated high-performance IPs.

¹ Hereafter, by reflective system, we mean a system that is able to modify its own structure and behaviour while it is running. A reflective softcore thus dynamically adds, removes, and replaces IPs in the application running on it, and is able to dynamically modify its own program memory, thereby dynamically altering the program it is executing.

The reflective softcore HoMade that we have started developing in 2012 (<http://www.lifl.fr/~dekeyser/Homade>) follows these ideas. The current version is statically configurable; dynamically reconfigurable multi-processors are the next steps. Users have to add to it the functionality they need in their applications via IPs. We have also developed a library of IPs for the most common processor functions (ALU, registers, ...).

In the multi-reflective softcores that we shall develop, some softcores will be slaves and others will be masters. Massively parallel dynamically reconfigurable architectures of softcores can thus be envisaged. This requires, additionally, a parallel management of the partial dynamic reconfiguration system. This can be done, for example, on a given subset of softcores: a massively parallel reconfiguration will replace the current replication of a given IP with the replication of a new IP. Thanks to the new 3D-FPGAs this task can be performed efficiently and in parallel using the large number of 3D communication links (Through-Silicon-Vias). Our roadmap for HoMade is to evolve towards this multi-reflective softcore model.

3.3. When Hardware Meets Software

HIPEAC vision 2011/12: *"The number of cores and instruction set extensions increases with every new generation, requiring changes in the software to effectively exploit the new features."*

When the new massively parallel dynamically reconfigurable architectures become reality users will need languages for programming software applications on them. The languages will be themselves dynamic and parallel, in order to reflect and to fully exploit the dynamicity and parallelism of the architectures. Thus, developers will be able to invoke reconfiguration and call parallel instructions in their programs. This expressiveness comes with a cost, however, because new classes of bugs can be induced by the interaction between dynamic reconfiguration and parallelism; for example, deadlocks due to waiting for output from an IP that does not exist any more due to a reconfiguration. The detection and elimination of such bugs before deployment is paramount for cost-effectiveness and safety reasons.

Thus, we shall build an environment for developing software on parallel, dynamically reconfigurable architectures that will include languages and adequate formal analyses and verification tools for them, in addition to more traditional tools (emulators, compilers, etc). To this end we shall be using formal-semantics frameworks associated with easy-to-use formal verification tools in order to formally define our languages of interest and allow users to formally verify their programs. The K semantic framework (<http://k-framework.org>), developed jointly by Univs. Urbana Champaign, USA, and Iasi, Romania) is one such framework, which is mature enough (it has allowed defining a formal semantics of the largest subset of the C language to date, as well as many other languages from essentially all programming paradigms) and is familiar to us from previous work. In K, one can rapidly prototype a language definition and try several versions of the syntax and semantics of instructions. This is important in our project, where the proposed programming languages (in particular, the HoMade assembly language) will go through several versions before being stabilized. Moreover, once a language is defined in K one gets an interpreter of the language and one gains access to formal verification tools for free. We are also developing new analysis verification tools for K (in collaboration with the K team), which will be adapted and used in the Dreampal project.

We shall focus on embedded systems performing intensive computations, in particular smart-camera systems and set-up-boxes. Some of the targeted classes of applications are safety-critical, and formal verification is essential for them. For the other ones, formal verification provides an added value in terms of quality of service.

4. Application Domains

4.1. Reflective Camera Networks

HIPEAC vision 2011-2012:

"reconfiguration, customization and runtime adaptation techniques will facilitate switching between tasks during the deployment of smart camera networks"

A Smart Camera (SC) is a vision system which, in addition to image-capturing capabilities, is able to extract application-specific information from the captured images and to automatically make intelligent decisions based on them. Dynamicity is inherent in SCs: processing may change depending on the specific observations they make and on the context. For example, an SC may use a low-quality face recognition IP while observing an office during the day, but switch to a high-quality one if it detects an intrusion during the night. Moreover, image processing requires high-performance computing, which is achieved by using parallelism. Thus, the integration of dynamic reconfiguration and parallelism, which is addressed by our project, is naturally present in SCs. Previous work in the DaRT team has already explored efficient uses of FPGAs in an SC network deployed in a retail store. A new proposal concerns embedded reflective camera in the Smart Cities multidisciplinary project developed on the University Lille 1 campus.

4.2. Set-top Boxes

Television sets and set-top-boxes are forming a symbiotic connection, which relies on common standards and protocols such as DLNA, Web standards, Web 2.0, H264, HEVC... As a result, the hardware platform on which applications run is becoming less important: commonly used ISAs like x86 are not mandatory any more. Dedicated pieces of hardware could efficiently provide specific services according to user requests. End-users expect platforms supporting many services with maximum performance, but do not require all of them at the same time. Dynamic reconfiguration is here too a good compromise, and it is efficient enough to support high performance algorithms like H264 or HEVC. It could also provide a ground for supporting on-the-fly codec switching. This may occur because the broadcaster decides to change the encoding of its video signal for safety reasons. Nowadays this operation is performed by a software because changing a hard codec still means flashing the set-top boxes to update it. Dreampal has started a collaboration with Kalray (<http://www.kalray.eu>) to develop a massively parallel language (without dynamic reconfiguration facilities, for now) on their MPSoC. H264 will be tested on this chip and on special FPGA boards with dedicated extensions for multimedia applications like the Xilinx Zynq.

4.3. Safe and Intelligent Transportation

Safety issues are today a key differentiator in the transportation industrial sector. The supervision and the detection of dangerous situations is a key technological challenge for future transportation systems at the infrastructure and vehicle levels. As an example, various obstacles can be detected on the road or in a Level Crossing (LC) using embedded systems. The proposed system will be based on stereo-vision technology (high definition cameras) and embedded reconfigurable computing and can be integrated either in vehicles or in the rail network. Also, Dreampal has started a collaboration with INDUCT (<http://induct-technology.com/>) to develop reconfigurable parallel architecture for the detection and the identification of obstacles in the frame of the NAVIA (autonomous electrical vehicle) project. This application will be implemented on Xilinx Zynq-based boards equipped with video processing features.

5. Software and Platforms

5.1. Software and Platforms

Download page : https://gforge.inria.fr/frs/?group_id=3646

5.1.1. HoMade

HoMade V4 is available and was used by 140 students this year on a Xilinx Nexys3 board. The Xilinx Virtex6 and Virtex7 are also supporting this new release. All the design is in VHDL except some ISE schematic specifications.

The main novelty of this release concerns :

- three stage pipe-lining of the HoMade core,
- new execution stack to improve frequency,
- instruction memory loading via UART port,
- MSPMD support
- reflexive features: Write-In-Program-Memory (WIM) instruction
- development of new IPs.

A test with 56 HoMade Slaves on a ring topology was running on a Virtex6 on a parallel matrix-vector multiplication example.

A low-level stack-based assembler supports binary generation from a Forth-like post-fixed syntax. It is written in Forth and can automatically generate binary code for the UART Port loading. This assembler will be merged with the JHomade software.

5.1.2. *JHomade*

JHomade is a software suite including a compiler for the HoMade processor. It allows us to compile HiHope programs (or HoMade assembly) and load the binary on the FPGA board. Its first release was in december 2013.

5.1.3. *\mathbb{K} -based language-independent symbolic execution and verification tool*

The results in [9], [14] were implemented in the \mathbb{K} framework and are distributed with it. The implementation allows users to symbolically execute programs in arbitrary languages defined in \mathbb{K} , with the only restriction that data cannot become code (and reciprocally). It also allows users to formally verify programs against specifications written in Reachability Logic, a specification formalism that can be seen as a language-independent Hoare logic. These language-independent tools will be specialized to the languages of interest in the project (HiHope, HoMade assembly and machine code).

6. New Results

6.1. Language-Independent Symbolic Execution, Program Equivalence, and Program Verification

A significant part of our research project consists in applying formal techniques for symbolically executing and formally verifying HiHope programs, as well as for formally proving the equivalence of HiHope programs with the corresponding HoMade assembly and machine-code programs obtained by compilation of HiHope.

- Symbolic execution will detect bugs (e.g., stack underflow) in HiHope programs. Additionally, symbolic execution is the natural execution manner of HiHope programs as soon as they contain (typically, underspecified) hardware IPs;
- program verification will guarantee the absence of bugs (with respect to specified properties, e.g., no stack underflow, no invocation of unavailable IPs, ...);
- program equivalence will guarantee that such above-mentioned bugs are also absent from the HoMade assembly and machine-code programs obtained by compilation of HiHope source code.

Since these languages (especially HiHope) are not completely defined yet, we decided to work (together with our colleagues from Univ. Iasi, Romania) on language-independent symbolic execution, program-equivalence, and program-verification techniques. In this way, when all the languages in our project become stable, we will be readily able to instantiate the above generic techniques on (the \mathbb{K} formal definitions of) the languages in question. We note that all the techniques described below are also independent of \mathbb{K} : they are applicable to other language-definition frameworks that use similar rewriting-based formal operational semantics.

6.1.1. Symbolic Execution

In [9] we propose a language-independent symbolic execution framework for languages endowed with a formal operational semantics based on term rewriting. Starting from a given definition of a language, a new language definition is automatically generated, which has the same syntax as the original one but whose semantics extends data domains with symbolic values and adapts semantical rules to deal with these values. Then, the symbolic execution of concrete programs is, by definition, the execution of programs with the new symbolic semantics, on symbolic input data. We prove that the symbolic execution thus defined has the properties naturally expected from it. A prototype implementation of our approach was developed in the K framework. We demonstrate the genericity of our tool by instantiating it on several languages, and show how it can be used for the symbolic execution and model checking of several programs.

6.1.2. Program Equivalence

In [12] we propose a logic and a deductive system for stating and automatically proving the equivalence of programs in deterministic languages having a rewriting-based operational semantics. The deductive system is circular in nature and is proved sound and weakly complete; together, these results say that, when it terminates, our system correctly solves the program-equivalence problem as we state it. We show that our approach is suitable for proving the equivalence of both terminating and non-terminating programs, and also the equivalence of both concrete and symbolic programs. The latter are programs in which some statements or expressions are symbolic variables. By proving the equivalence between symbolic programs, one proves in one shot the equivalence of (possibly, infinitely) many concrete programs obtained by replacing the variables by concrete statements or expressions. We also report on a prototype implementation of the proposed deductive system in the K framework.

6.1.3. Program Verification

In [14] we present an automatic and language-independent program verification approach based on symbolic execution. The specification formalism we consider is Reachability Logic, a language-independent logic that constitutes an alternative to Hoare logics. Reachability Logic has a sound and relatively complete deduction system, which offers a lot of freedom (but no guidelines) for constructing proofs. Hence, we propose symbolic execution as a strategy for proof construction. We show that, under reasonable conditions on the semantics of programming languages, our symbolic-execution based Reachability-Logic formula verification is sound. We present a prototype implementation of the resulting language-independent verifier as an extension of a generic symbolic execution engine that we are developing in the K framework. The verifier is illustrated on programs written in languages also formally defined in K.

6.2. Master-Slave Control Structure for MP-SoC Architectures

Our Synchronous Communication Asynchronous Computation (SCAC) model is a data-parallel execution model dedicated to the Massively Parallel System-on-Chip. This model proposes a novel control structure, referred to as master-slave control [11]. Its concept departs from the centralized configuration. However, instead of a uni-processor master controlling a set of parallel processing elements (PE), the master cooperates with a grid of parallel slave controllers which supervises the activities of cluster of PEs.

The control structure in SCAC model is presented by two hierarchical control levels:

- The Master Control Unit (MCU), which controls the order execution in the whole system. It is a simple processor, which fetches and decodes program instruction and broadcasts execution orders to Slave Control Unit. It controls the end execution to establish synchronous communication.
- The Slave Control Unit (SCU), which controls: local node and PEs activities, parallel instructions execution and synchronous communication. It is a crucial component in the master-slave control structure. The SCUs grid allows independent parallel execution.

The hardware architecture is composed of a single MCU and multiple Slave controllers (SCUs) combined with local processing element (PE) (or a cluster of 16 PEs), known collectively as Nodes. The MCU and SCU array are connected through single level hierarchical bus and the SCUs are connected together through X-net interconnection network [2]. This network is clocked synchronously with the SCUs and respectively with the PEs. SCU controllers in the grid care for the instruction execution activities that involve a large degree of parallelism and the communication activities that need to coordinate all the PEs in the grid. The structure of master-slave control should be distinguished from other hierarchical or clustered approaches proposed for parallel computing. Such proposals are usually motivated by memory latency considerations and the desire to build a scalable system. The use of two control levels is therefore visible to the user in its effect on the communication between various processors. With master-slave control structure, the PEs in massively parallel system can execute independently and then can communicate synchronously. Such a construction has the advantage of allowing the designer to optimize distinct processors for their intended tasks and to implement simple interconnection network without additionally buffers and complex routing algorithms.

The aim of these last works is to design a master-slaves control structure for SCAC architecture to allow autonomous processing with simple and regular communication. This control structure based on IP blocks which offers good flexibility and scalability was implemented in synthesizable VHDL code. It is simulated and synthesized for Xilinx Virtex q6 (XC6VLX240T) board. The difficulty of designing a master-slave structure is a compromise between an optimal execution time and high flexibility, while reducing power consumption and silicon area.

6.3. Toward a Massively Parallel and Reflective Execution Model

FPGAs are undoubtedly suited to the definition of what could be called a DSHA (Domain Specific Hardware Architecture). Similarity with the DSSA (Domain Specific Software Architecture) an assembly of functional components performs basic transformations on data, while a software / hardware infrastructure ensures the ordering of these transformations. The HoMade processor is designed with this in mind: it can be seen as an IP integrator offering a mechanism for interprocess communication IPs via a battery and a scheduler of IPs via dedicated instructions for flow control. In this control we find two particular instructions for flow control designed for a massively parallel execution model for SPMD, and a new instruction can make HoMade reflexive . With this instruction, you can at runtime change the behavior of a virtual component by dynamically associating it to a particular HoMade instruction sentence and in particular IP triggering instructions. Same components can successively after applying this instruction, trigger a hardware IP, a software function which itself can trigger a flow of execution of hardware IPs. This intercession² feature , parts of HoMade core, is valid for one processor or for all HoMade slave components in a massively parallel architecture. We demonstrated on a FPGA board which computes the Fibonacci sequence with three different methods, but always through a single call to a unique Virtual Component.

6.4. Power Estimation at System-Level for MPSoC Based Platforms

Shifting the design entry point up to the system level is the most important countermeasure adopted to manage the increasing complexity of Multiprocessor System on Chip (MPSoC). The reason is that decisions taken at this level, early in the design cycle, have the greatest impact on the final design in terms of power and energy efficiency. However, taking decisions at this level is very difficult, since the design space is extremely wide and it has so far been mostly a manual activity. Efficient system-level power estimation tools are therefore necessary to enable proper Design Space Exploration (DSE) based on power/energy and timing. We propose a tool based on efficient hybrid system level power estimation methodology for MPSoC. In this methodology, a combination of Functional Level Power Analysis (FLPA) and system level simulation technique are used to compute the power of the whole system. Basically, the FLPA concept is proposed for processor architecture in order to obtain parameterized arithmetic power models depending on the consumption of the main functional blocks. In this work, FLPA is extended to set up generic power models for the different parts of the platform.

²Wikipedia definition: intercession is the ability of a program to modify its own execution state or alter its own interpretation or meaning.

In addition, a simulation framework is developed at the transactional level to evaluate accurately the activities used in the related power models. The combination of the above two parts leads to a hybrid power estimation, that gives a better trade-off between accuracy and speed. The proposed methodology has several benefits: It considers the power consumption of the embedded system in its entirety; and Leads to accurate estimates without a costly and complex material. The proposed methodology is also scalable for exploring complex embedded architectures. Based on the proposed methodology, our Power Estimation Tool at System-Level (PETS) is developed. The usefulness and effectiveness of our PETS tool is validated through a typical mono-processor and multiprocessor embedded system designed around the TI OMAP (3530 and 5912) and the Xilinx Virtex II Pro FPGA boards. This methodology is demonstrated and evaluated by using a variety of basic programs to complete media benchmarks. Estimated power values are compared to real board measurements for both simple and multiprocessor architectures. Our obtained power estimation results provide less than 3% of error for mono-processor, 3.8% for homogeneous multiprocessor system and 4.3% for heterogeneous multiprocessor system and 70x faster compared to the state-of-the-art power estimation tools. These results have been presented in the PhD of Santhosh Kumar Rethinagiri [2] and published in [4].

6.5. Dynamically reconfigurable CPU/FPGA architecture for the testing and simulation of avionic systems

Real-time computing systems are increasingly used in aerospace and avionic industries. In the face of power wall and real-time requirements, hardware designers are directed towards reconfigurable computing with the usage of heterogeneous CPU/FPGA systems. However, there is a lack of real-time environments able to deal with the execution of applications on such heterogeneous systems dedicated to avionic Testing and Simulation (T&S). This year, we addressed the problem of soft real-time environments for CPU/FPGA systems and we proposed first a high-performance hardware architecture used to implement intimately coupled hardware and software avionic models. Second, we developed an efficient real-time software environment for the model's execution, the multi-core CPU monitoring and the runtime task re-allocation to avoid the timing constraint violation. Experimental results underpin the industrial relevance of the presented approach for avionic T&S systems with real-time support. These results are presented in the PhD of George Afonso [1] and in different publications [7] [10] [8].

6.6. A custom reconfiguration controller for partial and dynamic reconfiguration in HoMade based systems

In all Xilinx devices supporting dynamic reconfiguration, such a functionality is realized using a hardware reconfiguration port called ICAP, that moves bitstreams from the reconfiguration memory to the programmable logic. ICAP is initialized by a Xilinx HW controller driven exclusively by a Microblaze processor and thus connected to a PLB or AXI bus.

This makes the partial and dynamic reconfiguration a very tedious task, as it implies using several Xilinx tools (XPS, ISE, PlanAhead,..etc). PDR becomes also resources and time consuming due to the fact that it uses very large interfaces and a static Xilinx architecture (in addition to the system that we want to design) including specific processors, buses, controllers,..etc.

Our contribution is the design of a custom ICAP controller, driven only by a HoMade processor, without any additional processors, buses or controllers. This ensures that our HoMade reconfigurable systems consumes fewer resources on the FPGA and does not require other tools than the standard ISE and PlanAhead tools in order to be designed.

6.7. Hardware control for partially and dynamically reconfigurable systems: from modelling to implementation

This work proposes a control design methodology for FPGA-based reconfigurable systems aiming at increasing control design productivity and guaranteeing implementation efficiency. This methodology is based on a

semi-distributed control model [5] composed of a set of modular distributed controllers executing each observation, decision-making and reconfiguration tasks for a reconfigurable region of the system, and a coordinator between the distributed controllers decisions in order to respect global systems constraints and objectives. This semi-distributed decision-making is based on the mode-automata formalism. The proposed combination between modularity, control splitting and formalism-based design allows to enhance the flexibility, reusability and scalability of the control design. Another point that can be added to this combination, to enhance design productivity, is design automation. For this, the proposed methodology is based on Model-Driven Engineering approach [5] allowing to automate code generation from high-level models. This approach makes use of the UML MARTE (Modeling and Analysis of Real-Time and Embedded Systems) standard profile, allowing to make low-level technical details transparent to designers and to automate the VHDL code generation for hardware implementation of the modeled control systems in order to guarantee their performance. The generated control systems were validated using simulation. Synthesis results showed an acceptable time and resource overhead for systems having different numbers of controllers. A control system composed of four controllers and a coordinator was also validated through physical implementation in an FPGA system for an image processing application.

6.8. A model-based approach for dynamically reconfigurable systems design: from MARTE to RecoMARTE

This work is done in the context of the ANR FAMOUS project. It proposes a co-design methodology of dynamically reconfigurable systems based on FPGA. Our methodology is based on the Engineering Model Driven approach (MDE) and the models specification is done in the UML MARTE profile. It aims at ensuring flexibility, reusability and automation to facilitate the work of the designer and improve his productivity. The first contribution related is identifying parts of dynamically reconfigurable FPGA that can be modeled at the high abstraction levels. So, we defined a design flow based on the MDE to ensure the automation of code generation. According to this flow, several models are created mainly through MARTE profile concepts.

However, the modeling concepts of dynamic reconfiguration on FPGAs required extensions in MARTE. Thus, we identified the missing concepts to be integrated in a new profile that extends MARTE called RECOMARTE. The second contribution allows the chain automation and experimental validation. To integrate our design flow and to automate code generation, a processing chain was used. The final model resulting from MARTE proposed design flow is given as input to this chain.

We thereby move from MARTE to RECOMARTE models via an intermediate description according to the IP-XACT standard to finally generate files describing the complete system in the Xilinx XPS environment. This automation will accelerate the design phase and avoid errors due to the direct manipulation of these details. Finally, an example of application of image processing has been developed to demonstrate and validate our methodology.

7. Bilateral Contracts and Grants with Industry

7.1. Bilateral Contracts with Industry

Collaboration contract with Nolam Embedded Systems: In conjunction with the CIFRE grant of Venkatasubramanian Viswanathan, a collaboration contract is established with Nolam ES. The objective is to design an innovative embedded computing platform supporting massively parallel dynamically reconfigurable execution model. The use-cases of this platform cover several application domains such as medical, transportation and aerospace.

7.2. Bilateral Grants with Industry

The FAMOUS project aims at introducing a complete methodology that takes the reconfigurability of the hardware as an essential design concept and proposes the necessary mechanisms to fully exploit those capabilities at runtime. The project covers research in system models, compile time and run time methods, and analysis and verification techniques. These tools will provide high-quality designs with improved designer productivity, while guaranteeing consistency with the initial requirements for adaptability and the final implementation.

Thus FAMOUS is a research project with an immediate industrial impact. Actually, it will make reconfigurable systems design easier and faster. The obtained tool in this project is expected to be used by both companies designers and academic researchers, especially for modern applications system specific design as smart camera, image and video processing, FAMOUS tools will be based on well established standards in design community. In fact, modeling will start from very high abstraction level using an extended version of MARTE. Simulation and synthesizable models will be obtained by automatic model to model transformations, using MDE approach. These techniques will contribute to shorten drastically time-to-market.

FAMOUS ended in December 2013. Its main result is a complete MDE tool for modeling, transforming and generating dynamically reconfigurable systems targeting Xilinx devices. This tool has been validated on a video processing application as a demonstrator.

8. Partnerships and Cooperations

8.1. Regional Initiatives

8.1.1. IRCICA project " Smart Cities "

Smart Cities is an interdisciplinary project, internal of IRCICA (<http://www.ircica.univ-lille1.fr/>), in collaboration with the laboratory of Civil Engineering of Lille I. It builds on the expertise of several teams hosted by IRCICA (RF networks, sensors, high-performance and real-time embedded systems computing, pattern recognition). The scientific problem, we tackle within this project, is to develop an intelligent platform for managing accidents and incidents in the drinking water and wastewater. In this platform, a permanent dialogue M2M (machine to machine) between servers, embedded systems (laptops, smartphones, tablets, ...), smart cameras, and sensors, will detect and solve problems in real time.

Scientific problems relate to the study of the possibility of linking objects (cameras , sensors, servers ...) all together, with a standardized mixed network (radio frequency wifi and internet). DreamPal is responsible for implementing the part of the hardware platform for high performance dedicated to intelligent video applications using the HoMade softcore. This work involves the processing of data , analysis of video images, the use of these data, and the integration of embedded reconfigurable components (on Xilinx Zynq 7000 board) as well as the existing RF network cards. It uses the video data acquisition to apply algorithms to detect such an anomaly on the water in a part of the building, or abnormal number of people in a given area, or any information about a specific person such as the recognition of face, the nature of motion. The work done during this year usefully supplements our platform by developing video modules dedicated to intelligent surveillance

8.2. International Initiatives

We have a strong ongoing collaboration with Univ. Iasi, Romania, which includes (but is not limited to) the co-supervision of the PhD of Andrei Arusoae. Collaboration topics include language-independent techniques for analysis of programs, and their specialization to the languages designed in the Dreampal project (HiHope, HoMade assembler and machine code).

8.3. International Research Visitors

8.3.1. Visits of International Scientists

Prof. Dorel Lucanu, Assist. Prof./ Stefan Ciobaca, and PhD student Andrei Arusoai from Univ. Iasi (Romania) visited us in July 2013. We initiated work on language-independent program-verification techniques and on the formal definitions of the HiHope and HoMade assembler languages, as well as on the formally proved correctness of compilation between these languages.

8.3.2. Internships

Kanwarjeet Dhaliwal made his internship in the Dreampal team from May to July 2013. He worked on the formal semantics of the parallel version of Hihope, and also made a preliminary work to compile Hihope to the Kalray's MPPA platform. This work was partially funded by Kalray (<http://www.kalray.eu>).

8.3.3. Visits to International Teams

In June 2013, Rabie Ben Atitallah and Wissem Chouchene visited Michael Huebner, Professor and Chair for Embedded Systems in Information Technique (ESIT) at the Ruhr-University of Bochum. The objective is to establish a new collaboration in the field of 3D FPGA next generation.

In October 2013, Andrei Arusoai visited the team of Prof. Grigore Roşu at the University of Illinois at Urbana Champaign, where he worked on implementing the symbolic domains used in our language-independent symbolic execution and verification tool. He benefitted from the guest team's expertise on symbolic domains.

9. Dissemination

9.1. Scientific Animation

F. Guyomarch is a member of the CompAS program committee. Jean-luc Dekeyser is PC Member of DSD, Reconfig, Recosoc and Sympa.

9.2. Teaching - Supervision - Juries

9.2.1. Teaching

Licence : F. Guyomarch, Algorithmique et programmation, 144h, L1, IUT-A (Université de Lille 1), France

Licence : F. Guyomarch, Modélisation et théorie des langages, 64h, L2, IUT-A (Université de Lille 1), France

Licence : Philippe Marquet, Introduction to Computer Science, 15h, Secondary Education Teacher Training, Université Lille 1, France

Licence : Philippe Marquet, System Programming, 60h, L3, Université Lille 1, France

Master: Philippe Marquet, Design of Operating System, 60h, M1, Université Lille 1, France

Master: Philippe Marquet, Web of Things: Embedded System Programming, 20h, M1, Université Lille 1, France

Master: Philippe Marquet, Parallel and Distributed Programming, 24h, M1, Université Lille 1, France

Master: Philippe Marquet, Introduction to Innovation and Research, 15h, M2, Université Lille 1, France

Licence : Jean-Luc Dekeyser, Architecture élémentaire, 85h, L2, Université Lille 1, France

Master: Jean-Luc Dekeyser, Architecture évoluée, 90h, M1, Université Lille 1, France

Licence : Rabie Ben Atitallah, Introduction to Computer Architecture and Operating System, 36h, L2, Université de Valenciennes et du Hainaut-Cambrésis, France

Licence : Rabie Ben Atitallah, Algorithms and Language C Programming, 48h, L2, Université de Valenciennes et du Hainaut-Cambrésis, France

Master: Rabie Ben Atitallah, Tools for Embedded System Design, 32h, M2, Université de Valenciennes et du Hainaut-Cambrésis, France

Master: Rabie Ben Atitallah, Development and Compilation of Embedded Application, 32h, M2, Université de Valenciennes et du Hainaut-Cambrésis, France

Master: Vlad Rusu, Software Specification and Verification, 27 h, Univ. Lille 1, France

Master: Vlad Rusu, Advanced Software Architecture, 42 h, Univ. Lille 1, France.

9.2.2. Committees

Vlad Rusu participated as a reviewer in the PhD committees of Rouwaida Ben-Abdallah (Univ. Rennes) and Pierre-Nicolas Tolle (Conservatoire National des Arts et Métiers, Paris).

9.3. Popularization

Philippe Marquet is vice-president of the *Société informatique de France*, the French professional society in computer science.

Philippe Marquet is involved in scientific popularization, mostly within the context of a partnership of between the Inria Lille - Nord Europe Research Center, the University Lille 1, and the Académie of Lille. He organizes and participates to the visit of classrooms on the Inria Plateau at EuraTechnologies, promoting interactions between the scientific community and secondary school students and their teachers. This year, 30 “provisseurs”, 30 teachers, and about 170 students spend half a day on the Plateau. He has designed the [isnlilleacademie.fr](http://www.isnlilleacademie.fr) web site (<http://www.isnlilleacademie.fr>), an information site on the *spécialité ISN* (computer and digital science in french high schools) for pupils, students, and their parents’.

Philippe Marquet is a member of the editorial board of 1024, the new bulletin of the *Société informatique de France* that aims at showing informatics, science and technology, in all its dimensions. 1024 targets a wide audience, from high school students to researcher, including anyone interested in computer science.

10. Bibliography

Publications of the year

Doctoral Dissertations and Habilitation Theses

- [1] G. AFONSO. , *Vers une nouvelle génération de systèmes de test et de simulation avionique dynamiquement reconfigurables*, Université des Sciences et Technologie de Lille - Lille I, July 2013, <http://hal.inria.fr/tel-00921874>
- [2] S. KUMAR RETHINAGIRI. , *Une approche système pour l'estimation de la consommation de puissance des plateformes MPSoC*, Université de Valenciennes et du Hainaut-Cambresis, March 2013, <http://hal.inria.fr/tel-00921894>
- [3] C. TRABELSI. , *Contrôle matériel des systèmes partiellement reconfigurables sur FPGA : de la modélisation à l'implémentation*, Université des Sciences et Technologie de Lille - Lille I, July 2013, <http://hal.inria.fr/tel-00852361>

Articles in International Peer-Reviewed Journals

- [4] R. BEN ATITALLAH, E. SENN, D. CHILLET, M. LANOE, D. BLOUIN. *An Efficient Framework for Power-Aware Design of Heterogeneous MPSoC*, in "IEEE Transactions on Industrial Informatics", February 2013, vol. 9, n^o 1, pp. 487-501 [DOI : 10.1109/TII.2012.2198657], <http://hal.inria.fr/hal-00921900>
- [5] S. MEFTALI, J.-L. DEKEYSER, C. TRABELSI. *Decentralized control for dynamically reconfigurable FPGA systems*, in "Microprocessors and Microsystems", 2013, <http://hal.inria.fr/hal-00922290>
- [6] V. RUSU. *Embedding domain-specific modeling languages into Maude specifications*, in "Software and Systems Modeling", 2013, vol. 12, n^o 4, pp. 847-869 [DOI : 10.1007/s10270-012-0232-5], <http://hal.inria.fr/hal-00660104>

International Conferences with Proceedings

- [7] G. AFONSO, Z. BAKLOUTI, D. DUVIVIER, R. BEN ATITALLAH, E. BILLAUER, S. STILKERICH. *Heterogeneous CPU/FPGA reconfigurable computing system for avionic test application*, in "IEEE 27th International Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW)", Cambridge, United Kingdom, May 2013, <http://hal.inria.fr/hal-00922004>
- [8] A. AIT EL CADI, R. BEN ATITALLAH, A. ARTIBA. *Mathematical Programming Models for Scheduling in a CPU/FPGA Architecture with Communication Delay*, in "IESM'2013 - International Conference on Industrial Engineering and Systems Management - 2013", Rabat, Morocco, October 2013, <http://hal.inria.fr/hal-00922014>
- [9] A. ARUSOAI, D. LUCANU, V. RUSU. *A Generic Framework for Symbolic Execution*, in "6th International Conference on Software Language Engineering", Indianapolis, United States, M. ERWIG, R. PAIGE, E. VAN WYK (editors), Lecture Notes in Computer Science, Springer Verlag, August 2013, vol. LNCS 8225, pp. 281-301, <http://hal.inria.fr/hal-00853588>
- [10] Z. BAKLOUTI, D. DUVIVIER, R. BEN ATITALLAH, A. ARTIBA, N. BELANGER. *Real-Time Simulator supporting Heterogeneous CPU/FPGA Architecture*, in "International Conference on Industrial Engineering and Systems Management", Rabat, Morocco, October 2013, <http://hal.inria.fr/hal-00922009>
- [11] H. KRICHENE, M. BAKLOUTI, J.-L. DEKEYSER, P. MARQUET, M. ABID. *Master-Slave Control structure for massively parallel System on Chip*, in "DSD SEAA - 16th Euromicro Conference on Digital System Design", Santander, Spain, 2013, <http://hal.inria.fr/hal-00906906>
- [12] D. LUCANU, V. RUSU. *Program Equivalence by Circular Reasoning*, in "Integrated Formal Methods", Turku, Finland, Lecture Notes in Computer Science, Springer, June 2013, vol. LNCS 7940, pp. 362-377, <http://hal.inria.fr/hal-00820871>

National Conferences with Proceedings

- [13] P. WATTEBLED, J.-P. DIGUET. *Méthodologie basée sur des membranes pour la gestion de la reconfiguration dynamique dans les systèmes embarqués parallèles*, in "CompAS", Grenoble, France, January 2013, <http://hal.inria.fr/hal-00922289>

Research Reports

- [14] A. ARUSOAIE, D. LUCANU, V. RUSU. , *Language-Independent Program Verification Using Symbolic Execution*, Inria, September 2013, n^o RR-8369, 22 p. , <http://hal.inria.fr/hal-00864341>

- [15] D. LUCANU, V. RUSU. , *Program Equivalence by Circular Reasoning*, Inria, 2013, n^o RR-8116, 33 p. , <http://hal.inria.fr/hal-00744374>