Activity Report 2014

Team TEMPO

Trustworthy EMbedded PlatfOrms

RESEARCH CENTER
Paris - Rocquencourt

THEME
Proofs and Verification

Table of contents

1. Members **1**

2. Overall Objectives **1**

3. Research Program **2**

3.1. Cyber Physical Systems 2

3.2. Verification of Embedded Systems Properties 3

4. Application Domains **5**

4.1. Cyber Physical Systems 5

4.2. Simulation 5

4.3. Automated Test Generation 5

5. New Software and Platforms **5**

6. New Results **5**

6.1. Highlights of the Year 5

6.2. Approximately Timed Simulation 5

6.3. Automated generation of simulator 6

6.4. Automated Test 6

6.5. SAT based bounded model checking 7

7. Partnerships and Cooperations **7**

7.1.1. Inria International Labs 7

7.1.2. Inria International Partners 7

7.1.3. Participation In other International Programs 7

8. Dissemination **7**

8.1. Promoting Scientific Activities 7

8.1.1. Scientific events 7

8.1.1.1. Invitationr 7

8.1.1.2. Member of the organizing committee 8

8.1.1.3. Member of the conference program committee 8

8.1.1.4. Reviewer 8

8.1.2. Journal 8

8.2. Teaching - Supervision - Juries 8

8.2.1. Teaching 8

8.2.2. Supervision 8

8.2.3. Juries 8

9. Bibliography **8**

Team TEMPO

Keywords: Embedded Systems, Virtualization, Simulation, Verification, Program Testing

This project is an international collaboration project. Although it is adminstratively attached to Inria Rocquencourt, the project is mostly run at East China Normal University, in Shanghai.

Creation of the Team: 2014 January 01.

1. Members

Research Scientists

Vania Joloboff [Team leader, Inria]
Mingsong Chen [Team leader, ECNU]

PhD Students

Haifeng Gu [ECNU]
Xinqian Zhang [ECNU]

Others

Shenpeng Wang [Master Student]
Shengpeng Liu [Master Student]

2. Overall Objectives

2.1. Objectives

Cyber physical systems have become ubiquitous in our everyday life, ranging from simple sensors to complex systems such as consumer electronics, communication devices, process control, etc. Cyber physical systems consist of a network of heterogeneous devices. Each devices is itself composed of increasingly sophisticated hardware and software, and their development and reliability has become a key to economic success.. In the context of this project, we are particularly interested in systems that have a specific application running on dedicated hardware. Moreover we want to focus on devices that are meant to important safety constraints, such as those used in transportation industry, whether automotive, trains or aircrafts. This project is aiming at making research advances towards the *development of safer and more reliable heterogeneous cyber physical systems* in the selected target domain, and at *reducing time to market* for the development of such systems, considering *both the application and the hardware parts*.

The project assumes a Model Driven Engineering (MDE) approach, accompanied with Virtual Prototyping. Because a model is a formal abstraction, it is easier to apply formal methods to verify its properties. Model verification tools, reachability checks, deadlocks detection, theorem proving techniques, can be used on the models to prove system properties. However, in the end, the modeling tools usually cannot generate the entire embedded software. Proven software typically has to be integrated with libraries that are not proven. Also most applications do not run on bare hardware, but use a real time operating system that has not been certified either. Hence, in fine, it is not always clear that properties proven on the model are implemented on the final system and it is still necessary to use more traditional validation techniques. The project aims at building also executable models of the target embedded platform providing a *virtual prototype* of the platform. Virtual prototypes can run the application software, thus the software engineering team can develop and test the entire system.

Models also make it possible to generate conventional tests instead of manual coding. An advantage of the virtual prototyping approach is that the embedded application software can be run onto the virtual prototype and immediately tested with conventional tests.

Within the context described above, our project aims at addressing the challenges of embedded systems design with a new approach, combining modeling and formal methods, possibly code generation of application code and/or tests code, next run and validate the application code on a approximately timed virtual prototype in order to verify qualitative and quantitative, functional or non functional properties of the final system.

This approach requires the constructive combination of a virtual prototyping environment, surrounded by tools for the analysis of simulation models or simulation output, or analysis of the embedded software to make proofs of properties of the target system. We therefore need to connect modeling tools and formal methods tools with simulation tools running the real application code. We propose for that to work collaboratively towards a development platform that consists of a set of complementary components forming a tool chain in the development process, by associating technologies and tools developed both in Europe and China by three institutions, namely **CWI** in Netherlands, **ECNU SEI** in China and **Inria** in France.

One can distinguish two orthogonal research directions:

- developing fast and powerful simulators that can simulate platforms and offer convenient interface to the users,
- connected to specific tools, the goal of which is to verify required properties of the system.

The first action is mostly relevant to simulation research to accelerate or parallelize the simulation. The second is related to usage of formal methods, as adjunct tools to the previous one.

3. Research Program

3.1. Cyber Physical Systems

The development of complex embedded systems platforms requires putting together many hardware components, processor cores, application specific co-processors, bus architectures, peripherals, etc. The hardware platform of a project is seldom entirely new. In fact, in most cases, 80 percent of the hardware components are re-used from previous projects or simply are COTS (Commercial Off-The-Shelf) components. There is no need to simulate in great detail these already proven components, whereas there is a need to run fast simulation of the software using these components.

These requirements call for an integrated, modular simulation environment where already proven components can be simulated quickly, (possibly including real hardware in the loop), new components under design can be tested more thoroughly, and the software can be tested on the complete platform with reasonable speed.

Modularity and fast prototyping also have become important aspects of simulation frameworks, for investigating alternative designs with easier re-use and integration of third party components. The project aims at developing such a rapid prototyping, modular simulation platform, combining new hardware components modeling, verification techniques, fast software simulation for proven components, capable of running the real embedded software application without any change.

To fully simulate a complete hardware platform, one must simulate the processors and co-processors, together with the peripherals such as network controllers, graphics controllers, USB controllers, etc. A commonly used solution is the combination of some ISS (Instruction Set Simulator) connected to a Hardware Description Language (HDL) simulator, in a co-simulation environment such as [12], [13]. Some communication and synchronization must be designed and maintained between the two using some inter-process communication (IPC), which slows down the process.

The idea we pursue is to combine hardware modeling and fast simulation into a fully integrated, software based simulation environment, which uses a single simulation loop thanks to Transaction Level Modeling (TLM) [3] combined with a new ISS technology designed specifically to fit within the TLM environment.

The most challenging way to enhance simulation speed is to simulate the processors. Processor simulation is achieved with Instruction Set Simulation (ISS). There are several alternatives to achieve such simulation. In *interpretive simulation*, each instruction of the target program is fetched from memory, decoded, and executed. This method is flexible and easy to implement, but the simulation speed is slow as it wastes a lot of time in decoding. Interpretive simulation is used in Simplescalar [2]. Another technique to implement a fast ISS is *dynamic translation* [8], [4] which has been favored by many implementors [18], [19], [20], [14] in the past decade.

There are many ways of translating binary code into cached data, which each come at a price, with different trade-offs between the translation time and the obtained speed up on cache execution. Also, simulation speed-ups usually don't come for free: most of time there is a trade-off between accuracy and speed. There are two well known variants of the dynamic translation technology: the target code is translated either directly into machine code for the simulation host, or into an intermediate representation, independent from the host machine, that makes it possible to execute the code with faster speed. A challenge in the development of high performance simulators is to maintain simultaneously fast speed and simulation accuracy. In the TEMPO project, we expect to develop a dynamic translation technology satisfying the following additional objectives:

- provide different levels of translation with different degrees of accuracy so that users can choose between accurate and slow (for debugging) or less accurate but fast simulation.
- to take advantage of multi-processor simulation hosts to parallelize the simulation;
- to define intermediate representations of programs that optimize the simulation speed and possibly provide a more convenient format for studying properties of the simulated programs.

Another objective of the TEMPO simulation is to extract information from the simulated applications in order to prove system properties. One can use model based tools to generate tests that can be run on the simulator to check whether the test fails or not on the real application. The project is considering an approach as illustrated in Figure 1

Thus, it is also a goal of TEMPO activities to use such formal methods tools to detect failures, either by generating tests, or by using formal methods tools to analyze the results of simulation sessions.

3.2. Verification of Embedded Systems Properties

Since last decade, we have witnessed rapid development in embedded system domain. More and more state-of-the-art embedded systems adopt the heterogeneous multi-processor platform rather than the platform with single core. To achieve better quality and performance, the design paradigm shift from simple control system to complex heterogeneous Cyber-Physical Systems (CPS) is gaining more interests. Increasing complexity coupled with time-to-market pressure create a critical need to validate heterogeneous embedded system designs. The functional validation is thus widely acknowledged as a major bottleneck in embedded system design. To guarantee the reliability of heterogeneous embedded systems, up to 70% of the overall design time and resources are spent on functional validation.

From the verification point of view, the major objective of this project is to reduce the overall validation efforts in the top-down design flow of embedded system design using the high-level specifications. In this project, we plan to address the following three major problems:

- **Formal modeling of high-level specifications**. We want to investigate how to model heterogeneous systems with multiple models of computation (MoC) and how to extract the formal models from system-level specifications to enable automated analysis.
- **Efficient validation of system-level specifications with minimum effort**. The idea here is to investigate the automated directed test generation from high-level specification validation and explore various approaches and techniques to further reduce the directed test generation time (eliminate redudant tests).
- **Consistency checking between different abstraction layers**. We also want to explore the possibility of reusing high-level validation efforts for low-level implementation validation as well as to check the consistency between different abstraction layers.

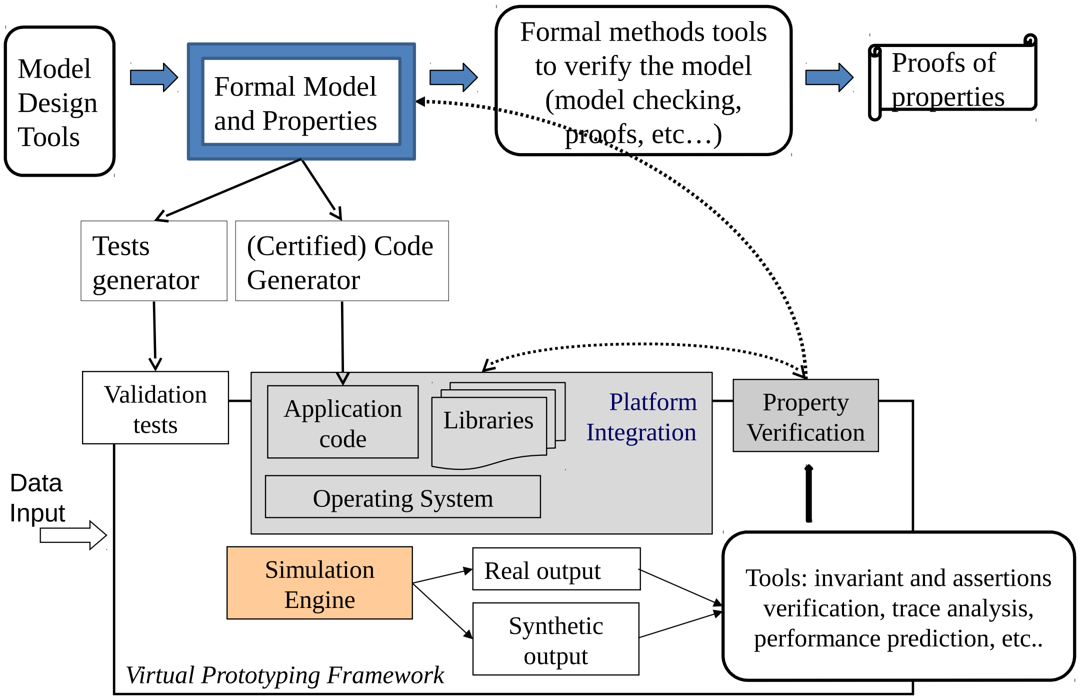
Model Design Tools
Formal Model and Properties
Formal methods tools to verify the model (model checking, proofs, etc…)
Proofs of properties
Tests generator
(Certified) Code Generator
Validation tests
Application code
Libraries
Platform Integration
Operating System
Property Verification
Data Input
Simulation Engine
Real output
Synthetic output
Tools: invariant and assertions verification, trace analysis, performance prediction, etc..
Virtual Prototyping Framework

Figure 1. Proposed development methodology

In conclusion, this project targets to improve the effectiveness and efficiency of functional validation of heterogeneous embedded systems. We believe that our approaches can not only enhance the reliability of heterogeneous embedded systems, but also reduce the time-to-market.

4. Application Domains

4.1. Cyber Physical Systems

The overall project is geared towards the development of complex heterogeneous cyber physical systems that require high reliability such as nuclear power, energy distribution, industry automation and transportation, where formal verification methods are necessary.

4.2. Simulation

Simulation is relevant to most areas where complex embedded systems are used, not only to the semiconductor industry for System-on-Chip modeling, but also to any application where a complex hardware platform must be assembled to run the application software. It has applications for example in industry automation, digital TV, telecommunications and transportation.

4.3. Automated Test Generation

Manual testing is tedious. Automated testing makes it possible to increase test coverage while also minimizing the amount of redundancy created by manual testing.

5. New Software and Platforms

5.1. SimSoC

We have continued to work on the SimSoC virtual prototyping framework distributed by Inria. Because of issues in the design of the Power Architecture simulator, we did a redesign of the Power simulator and a new implementation, so that we can simulate in the future both the Power Classic and Power Extended architectures in both 32 bits or 64 bits. We also contributed new extensions as described below.

6. New Results

6.1. Highlights of the Year

The project was created.

6.2. Approximately Timed Simulation

Participants: Vania Joloboff, Shenpeng Wang.

Existing fast simulators such as SimSoC are Loosely Timed. They evaluate the time taken by instructions executed based on an average model. Typically, the clock value is increased by a constant K every N instructions. This is sufficient to test application software with time-outs or to synchronize multicore applications, but it cannot provide a reasonable performance estimate of the embedded software.

To obtain precise parformance estimate, a common practice is to run the software on Cycle Accurate simulators, which provides a performance measure absolutely correct, but take a very long time. This is becoming a bottleneck. In fact, in many cases the software developers need some performance estimate, but do not require cycle precision. The idea of "Approximately Timed" simulation is to provide a fast simulation that can be used by software developers, and yet provide performance estimate. The goal of approximately timed simulation is to provide estimates that are within a small margin error from the real hardware, but at a simulation speed that is an order of magnitude faster than a cycle accurate one.

It is possible to maintain fast simulation, (though slower than Loosely Timed) whereas predicting reasonably accurate performance. The challenge is to come up with an abstract model of the processor that does not simulate the processor at cycle level but simulate enough to measure elapsed time with good precision. The approach is the following: a modern processor in nominal mode executes at least one instruction per clock cycle. If it does not do so, it is because there is a delay, whether a cache miss, a pipe line stall, etc. If one can simulate enough of the system so that the cause of the delays can be reproduced in the simulation and the delays evaluated, although the details of the system are not reproduced exactly, then the delays estimate may be accurate enough to provide an acceptable margin error. Moreover some of these computation can be done only once, not for each iteration of a loop.

In our work, we are considering only the processor model and we rely upon TLM interface to the interconnect for peripheral access to provides us with timing delays. We estimate the performance by using static analysis of the application control flow graph combined with a minimum of dynamic computation in order to maintain a reasonable simulation speed. We have developed such a fast Approximately Timed ISS, that does not fully simulate the hardware, yet provides good precision estimates, and does not use stastistical methods. Our approach consists in developing a higher abstraction model of the processor (than the CA models) that still executes instructions using fast SystemC/TLM code, but in parallel maintains some architecture state to measure the delays introduces by cache misses and pipe line stalls, although the pipe line is not really simulated. This work will be published in 2015 in volume 68 of the WIT Transactions on Information and Communication Technologies (ISBN 978-1-78466-054-3) [10].

6.3. Automated generation of simulator

Participants: Vania Joloboff, Shengpeng Liu.

Developing a simulator for a complete processor represents a lot of work when it is manual coding, and it is error-prone. Several efforts have been made to generate partly or entirely simulators. The dominant approach in the past years has been to use a high level description language of the processor and to generate code with the language compiler, such as LISA [18], MIMOLA [15], EXPRESSION-ADL [17]. But still, the architecture is described manually into the high level language. It is interesting to explore new architectures, but has the same issues as manual coding for simulation of commercial off-the-shelf processors such as ARM an Power architectures. Of course this approach only makes sense if the vendor has at least some semi-formal description of the architecture, which is not the case for Intel, but is the case for ARM, PowerPC and SH.

In order to automatically generate simulators from the vendor specification, we have initiated a new approach: generating the simulator from the specification of the hardware vendor as available from their web site as .pdf document. After a relatively successful initiative using ad-hoc tools, we wanted to pursue this work in a more robust and industrial context, using XML to generate an XML model of the instruction set from the vendor specification in .pdf, formalize some XML model transformations and finally generate directly the simulator code in C++. In addition, we wanted the translator to be architecture independent, not making any assumption during the translation process. However, this work is hitting difficult issues due to the fact that the vendor specification is incomplete and we have to do more manual architecture specific complements to the specification that we anticipated, which seriously weakens the project objective.

6.4. Automated Test

Participants: Mingsong Chen, Haifeng Gu, Xinqian Zhang.

Under the increasing complexity together with the time-to-market pressure, functional validation is becoming a major bottleneck of smart applications running on mobile platforms (e.g., Android, iOS).Unlike traditional software, smartphone applications are reactive and GUI (Graphical User Interface) intensive. The execution of smartphone applications heavily relies on the interactions with users. Manual GUI testing is extremely slow and unacceptably expensive in practice. However, the lack of formal models of user behaviors in the design phase hinders the automation of GUI testing (i.e., test case generation and test evaluation). While thorough test efforts are required to ensure the consistency between user behavior specifications and GUI implementations, few of existing testing approaches can automatically utilize the design phase information to test complex smartphone applications. Based on UML activity diagrams, we propose an automated GUI testing framework called ADAutomation, which supports user behavior modeling, GUI test case generation, and post-test analysis and debugging. The experiments using two industrial smartphone virtual prototypes demonstrate that our approach [16] can not only drastically reduce overall testing time, but also showed to improve the quality of designs.

6.5. SAT based bounded model checking

Participants: Mingsong Chen, Haifeng Gu, Xinqian Zhang.

SAT-based Bounded Model Checking (BMC) is promising for automated generation of directed tests. Due to the state space explosion problem, SAT-based BMC is unsuitable to handle complex properties with large SAT instances or large bounds. In this work, we propose a framework to automatically scale down the SAT falsification complexity by utilizing the decision ordering based learning from decomposed sub-properties. Our framework makes three important contributions: i) it proposes learning-oriented decomposition techniques for complex property falsification, ii) it proposes an efficient approach to accelerate the complex property falsification using the learning from decomposed sub-properties, and iii) it combines the advantages of both property decomposition and property clustering to reduce the overall test generation time. The experimental results [11] using both software and hardware benchmarks demonstrate the effectiveness of our framework.

7. Partnerships and Cooperations

7.1. International Initiatives

7.1.1. Inria International Labs

The TEMPO project belongs to the LIAMA laboratory in China. The project is hosted by East China Normal University Software Engineering Institute.

7.1.2. Inria International Partners

7.1.2.1. Declared Inria International Partners

The projects is run in collaboration with East China Normal University Software Engineering Institute and Netherlands CWI.

7.1.3. Participation In other International Programs

The project is run within the context of China LIAMA laboratory.

8. Dissemination

8.1. Promoting Scientific Activities

8.1.1. Scientific events

8.1.1.1. Invitationr

Vania Joloboff was invited for a short stay at Academia Sinica in Taipei Taiwan, in October 2014.

8.1.1.2. Member of the organizing committee

Vania Joloboff jointly organized with ECNU staff the LIAMA Open Day in Shanghai in July 2014

Vania Joloboff jointly organized and chaired a session at the LIAMA Workshop for collaboration between Europe and China in May 2014.

Mingsong Chen was invited to served as a Session Chair of the conference SERE 2015.

8.1.1.3. Member of the conference program committee

Vania Joloboff is program committee member of DATE conference workshop on Model Implementation Fidelity.

Mingsong Chen served as a Program Committee member of International Conference on VLSI Design.

Mingsong Chen served as a Program Committee member of the conference SAC 2015.

Mingsong Chen served as a Program Committee member of the conference DATE 2015.

8.1.1.4. Reviewer

Mingsong Chen was a reviewer for the conferences VLSID 2015, SAC 2015, DATE 2015, DAC 2015.

8.1.2. Journal

8.1.2.1. Member of the editorial board

Mingsong Chen serves as an associate editor of Journal of Circuits, Systems and Computers.

8.2. Teaching - Supervision - Juries

8.2.1. Teaching

Virtual Prototyping

Vania Joloboff : Virtual Prototyping, ECNU SEI

Master Students

8.2.2. Supervision

Mingsong Cheng and Vania Joloboff started supervision of several Master and PhD. students at East China Normal University.

8.2.3. Juries

Vania Joloboff was jury member of PhD defence for Yanwen Chen, a double PhD degree of East China Normal University and University of Nice.

9. Bibliography

Major publications by the team in recent years

[1] F. Blanqui, C. Helmstetter, V. Joloboff, J.-F. Monin, X. Shi. *Designing a CPU model: from a pseudo-formal document to fast code*, in "3rd Workshop on: Rapid Simulation and Performance Evaluation: Methods and Tools", Heraklion, Greece, January 2011, https://hal.inria.fr/inria-00546228

[2] D. Burger, T. M. Austin. *The SimpleScalar Tool Set, Version 2.0*, University of Wisconsin-Madison, 1997, n^{o} TR-1342

[3] L. Cai, D. Gajski. *Transaction level modeling: an overview*, in "CODES+ISSS '03: Proceedings of the 1st IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis", New York, NY, USA, ACM Press, 2003, pp. 19–24, http://doi.acm.org/10.1145/944645.944651

[4] B. Cmelik, D. Keppel. *Shade: a fast instruction-set simulator for execution profiling*, in "SIGMETRICS '94: Proceedings of the 1994 ACM SIGMETRICS conference on Measurement and modeling of computer systems", New York, NY, USA, ACM, 1994, pp. 128–137, http://doi.acm.org/10.1145/183018.183032

[5] C. Helmstetter, V. Joloboff. *SimSoC: A SystemC TLM integrated ISS for full system simulation*, in "APCCAS - IEEE Asia-Pacific Conference on Circuits and Systems - 2008", Macau, SAR China, IEEE (editor), IEEE, November 2008 [*DOI :* 10.1109/APCCAS.2008.4746381], https://hal.inria.fr/hal-00777158

[6] V. Joloboff, X. Zhou, C. Helmstetter, X. Gao. *Fast Instruction Set Simulation Using LLVM-based Dynamic Translation*, in "International MultiConference of Engineers and Computer Scientists 2011", Hong Kong, China, Lecture Notes in Engineering and Computer Science, Springer, March 2011, vol. 2188, pp. 212-216, https://hal.inria.fr/hal-00646947

[7] J. Song, C. Helmstetter, V. Joloboff, H. Hao. *Generation of Executable Representation for Processor Simulation with Dynamic Translation*, in "2008 International Conference on Computer Science and Software Engineering", Wuhan, China, IEEE (editor), IEEE, December 2008 [*DOI :* 10.1109/CSSE.2008.635], https://hal.inria.fr/hal-00777157

[8] E. Witchel, M. Rosenblum. *Embra: fast and flexible machine simulation*, in "SIGMETRICS '96: Proceedings of the 1996 ACM SIGMETRICS international conference on Measurement and modeling of computer systems", New York, NY, USA, ACM, 1996, pp. 68–79, http://doi.acm.org/10.1145/233013.233025

[9] Z. Zuyu, V. Joloboff, X. Zhou, C. Helmstetter. *Fast Dynamic Translation Using LLVM On Multi-Core Hosts*, in "5th Workshop on Architectural and Microarchitectural Support for Binary Translation (AMAS-BT)", Portland, Oregon, United States, ACM (editor), Intel Corporation, June 2012, https://hal.inria.fr/hal-00777156

Publications of the year

International Conferences with Proceedings

[10] V. Joloboff, S. Wang, Y. Deng. *Fast approximately timed simulation*, in "2014 International Conference on Computer Science and Systems Engineering (CSSE2014)", Hong Kong, SAR China, WIT Transactions on Information and Communication Technologies, WIT Press, September 2014, https://hal.archives-ouvertes.fr/hal-01081104

References in notes

[11] M. Chen, X. Qin, M. Prabhat. *Learning-oriented Property Decomposition for Automated Generation of Directed Tests*, in "Journal of Electronic Testing", 2014, vol. 30, n^{o} 3, pp. 287–306, http://dx.doi.org/10.1007/s10836-014-5452-x

[12] F. Fummi, G. Perbellini, M. Loghi, M. Poncino. *ISS-centric modular HW/SW co-simulation*, in "GLSVLSI '06: Proceedings of the 16th ACM Great Lakes symposium on VLSI", New York, NY, USA, ACM, 2006, pp. 31–36, http://doi.acm.org/10.1145/1127908.1127918

[13] P. Gerin, S. Yoo, G. Nicolescu, A. Jerraya. *Scalable and flexible cosimulation of SoC designs with heterogeneous multi-processor target architectures*, in "ASP-DAC '01: Proceedings of the 2001 conference on Asia South Pacific design automation", New York, NY, USA, ACM, 2001, pp. 63–68, http://doi.acm.org/10.1145/370155.370276

[14] C. Helmstetter, V. Joloboff, H. Xiao. *SimSoC: A full system simulation software for embedded systems*, in "OSSC'09", IEEE (editor), 2009

[15] R. Leupers, J. Elste, B. Landwehr. *Generation of Interpretive and Compiled Instruction Set Simulators*, in "Asia and South Pacific Design Automation Conference (ASP-DAC)", 1999, pp. 339–342

[16] A. Li, Z. Qin, M. Chen, J. Liu. *ADAutomation: An Activity Diagram Based Automated GUI Testing Framework for Smartphone Applications*, in "Eighth International Conference on Software Security and Reliability, SERE 2014", San Francisco, California, USA, June 2014, pp. 68–77, http://dx.doi.org/10.1109/SERE.2014.20

[17] W. S. Mong, J. Zhu. *A Retargetable Micro-architecture Simulator*, in "Design Automation Conference", 2003, vol. 0, 752 p. , http://doi.ieeecomputersociety.org/10.1109/DAC.2003.1219120

[18] A. Nohl, G. Braun, O. Schliebusch, R. Leupers, H. Meyr, A. Hoffmann. *A universal technique for fast and flexible instruction-set architecture simulation*, in "DAC '02: Proceedings of the 39th conference on Design automation", New York, NY, USA, DAC '02, ACM, 2002, pp. 22–27, http://doi.acm.org/10.1145/513918.513927

[19] M. Poncino, J. Zhu. *DynamoSim: a trace-based dynamically compiled instruction set simulator*, in "ICCAD '04: Proceedings of the 2004 IEEE/ACM International conference on Computer-aided design", Washington, DC, USA, IEEE Computer Society, 2004, pp. 131–136, http://dx.doi.org/10.1109/ICCAD.2004.1382557

[20] M. Reshadi, P. Mishra, N. Dutt. *Instruction set compiled simulation: a technique for fast and flexible instruction set simulation*, in "Design Automation Conference, 2003. Proceedings", 2003, pp. 758-763