Quantitative Evaluation in Embedded System Design: Validation of Multiprocessor Multithreaded Architectures

Nicolas Coste, Hubert Garavel, Holger Hermanns, Richard Hersemeule, Yvain Thonnart, Meriem Zidouni

Quantitative Evaluation in Embedded System Design: Validation of Multiprocessor Multithreaded Architectures*

Nicolas Coste
STMicroelectronics, Grenoble, France
nicolas.coste@st.com

Hubert Garavel
INRIA, Grenoble, France
hubert.garavel@inria.fr

Holger Hermanns
Saarland University, Saarbrücken, Germany
hermanns@cs.uni-sb.de

Richard Hersemeule
STMicroelectronics, Grenoble, France
richard.hersemeule@st.com

Yvain Thonnart
CEA/Leti, Grenoble, France
yvain.thonnart@cea.fr

Meriem Zidouni
Bull, Les Clayes-sous-Bois, France
meriem.zidouni@bull.net

Abstract

As levels of parallelism are becoming increasingly complex in multiprocessor architectures, GALS, and asynchronous circuits, methodologies and software tools are needed to verify their functional behavior (qualitative properties) and to predict their performance (quantitative properties). This paper presents the work currently done in the Multival project (pôle de compétitivité mondial Minalogic), in which verification and performance evaluation tools developed at INRIA and Saarland University are applied to three industrial architectures designed by Bull, CEA/Leti and STMicroelectronics.

1. Introduction

Asynchronous computing tends to be pervasive at many levels: grids, clusters, multiprocessor architectures, multicore processors, GALS, asynchronous logic. The absence of global synchronization (no clock or several clocks) is a major departure from the classical synchronous design approach. Due to their functional complexity, there is a need to prove correctness of asynchronous architectures and circuits, but no industrial methodology is available to date. Moreover, a high degree of concurrency may introduce latencies, thus decreasing the expected performance. Consequently, correctness of functional behavior may not be sufficient for the targeted applications: performance properties (such as latencies, throughputs, resource usage, etc.) should also be considered at design time.

Launched in December 2006 after successful collaborations between its partners [1, 4, 9, 8], the Multival project[1] aims at using formal methods for functional verification and performance evaluation of asynchronous hardware. This project focuses on three different architectures that share massive asynchronous parallelism and complex communication protocols, together with strong industrial potential:

- FAME2 (*Flexible Architecture for Multiple Environments*) is a CC-NUMA multiprocessor architecture developed at Bull for teraflops mainframes;
- FAUST (*Flexible Architecture of Unified System for Telecom*) is a Network-on-Chip (NoC) based platform designed at CEA/Leti for wireless telecom applications (4G, MIMO, etc.);
- xSTream is a multiprocessor dataflow architecture for high performance embedded multimedia streaming applications designed at STMicroelectronics.

2. Formal Modeling Flow in Multival

A first step in the Multival approach is to model the architectures under study using formal languages suitable for representing asynchronous behaviors. We use either the LOTOS language (ISO standard 8807) or the CHP (*Communicating Hardware Processes*) language that can be translated automatically into LOTOS [6].

*This work has been partially funded by the French government and by Conseil Général de l'Isère as part of the pôle de compétitivité Minalogic (Multival project).

[1] http://www.inrialpes.fr/vasy/multival

Multival already produced a CHP model of the asynchronous network router of FAUST, and LOTOS models for the network queues of xSTream, the higher level protocols of FAUST, the FAME2 circuits implementing routing/cache-coherency protocols, and the MPI software layer and MPI benchmark applications to be run over FAME2 mainframes. These models are structural (in a bottom-up way using composition of sub-modules) or functional (in a top-down way using successive refinements).

3. Functional Verification Flow in Multival

The second step is to use the CADP toolbox [2] to translate these LOTOS models into Labeled Transition Systems (LTSs), whose vertices and edges correspond respectively to the states and the internal/external communications in the model. Those LTSs, which enumerate the state space of the model, can be verified using the model checking tools (based on temporal logics) and/or the equivalence checking tools (based on bisimulations) included in CADP. To avoid state space explosion, refined approaches based on compositional verification and/or co-simulation are also used.

Up to now, several significant results have been obtained: two functional issues in xSTream have been highlighted [5]; the FAUST NoC router has been verified formally [7], and theoretical results on isochronous forks in asynchronous circuits have been demonstrated automatically.

4. Performance Evaluation Flow in Multival

Performance evaluation in the Multival project is based on the IMC (*Interactive Markov Chains*) formalism [3], which combines the concepts of concurrency theory (LOTOS, LTSs, bisimulations, etc.) with the theory of Markov chains, and which is supported by the CADP toolset [1, 4].

In this approach, the LOTOS functional models for the architectures are decorated with timing information describing the stochastic behavior. This can be done either *directly*, by inserting stochastic transitions into the LOTOS model, or *compositionally*, in the following steps: (1) localization of the relevant delays in the functional model; (2) exposition (using LOTOS gates) of the start and end of each such delay; (3) instantiation of each delay by synchronizing these LOTOS gates with an auxiliary LOTOS process expressing the delay as a phase-type distribution. The decorated model is then turned into an IMC using a compositional approach (which alternates state space generation and stochastic state space minimization). This IMC is then transformed into a Markov chain, for which the Markov solvers included in CADP can compute steady-state or time-dependent state probabilities and transition throughputs.

Using this flow, early results have been obtained in the Multival project. Bull was able to predict the latency of an MPI benchmark in different topologies, different software implementations of the MPI primitives, and different cache coherency protocols. STMicroelectronics explores this flow to predict latency, throughputs in the communication architecture, and occupancy within xSTream queues.

5. Conclusion

The joint study of functional verification and performance evaluation is a key point in the Multival project; so far, these two aspects have been addressed independently, using different modeling languages and software tools, thus increasing costs and delays. Although formal modeling of asynchronous hardware differs significantly from the standard synchronous design methodologies, and despite its steep learning curve, at the current development stage of Multival, LOTOS was sufficient to model the three architectures. These functional models are advanced enough to enable formal verification using CADP. Compositional verification (which requires expertise) is used to address state space explosion issues. Work on performance evaluation is currently exploring two issues: new algorithms to handle nondeterminism (currently not accepted by the Markov solvers of CADP), and representations of fixed-time delays, for which there is a space-accuracy tradeoff when approximating them in the IMC formalism.

References

[1] H. Garavel and H. Hermanns. On combining functional verification and performance evaluation using CADP. In *Proc. FME'2002*, LNCS 2391, pages 410–429, July 2002.

[2] H. Garavel, F. Lang, R. Mateescu, and W. Serwe. CADP 2006: A toolbox for the construction and analysis of distributed processes. In *Proc. CAV'2007*, LNCS 4590, pages 158–163, Jul 2007.

[3] H. Hermanns. *Interactive Markov Chains and the Quest for Quantified Quality*, LNCS 2428, 2002.

[4] H. Hermanns and C. Joubert. A set of performance and dependability analysis components for CADP. In *Proc. TACAS'2003*, LNCS 2619, pages 425–430, Apr. 2003.

[5] E. Lantreibecq. Verification of the xSTream communication model using CADP. Tech. report, STMicroelectronics, 2007.

[6] G. Salaün and W. Serwe. Translating hardware process algebras into standard process algebras — illustration with CHP and LOTOS. In *Proc. IFM'2005*, LNCS 3771, Nov. 2005.

[7] G. Salaün, W. Serwe, Y. Thonnart, and P. Vivet. Formal verification of CHP specifications with CADP — illustration on an asynchronous Network-on-Chip. In *Proc. 13th IEEE Int. Symp. on Asynchronous Circuits and Systems ASYNC 2007*, pages 73–82, Mar. 2007.

[8] `http://www.inrialpes.fr/vasy/dyade/formal fame.html`

[9] P. Wodey, G. Camarroque, F. Baray, R. Hersemeule, and J.-P. Cousin. LOTOS code generation for model checking of STbus-based SoC: The STbus interconnect. In *Proc. 1st ACM/IEEE Int. Conf. on Formal Methods and Models for Codesign MEMOCODE'03*, pages 204–213, June 2003.