FVDAM
Formal Verification of Deadlock Avoidance Mechanisms

Julien Schmaltz and Frits Vaandrager
Institute for Computing and Information Sciences
Radboud University Nijmegen
PO Box 9010 6500 GL Nijmegen, The Netherlands
{ julien, F.Vaandrager}@cs.ru.nl

January 15, 2008

1 Project Information

1a. Project Title **F**ormal **V**erification of **D**eadlock **A**voidance **M**echanisms

1b. Project Acronym FVDAM

1c. Principal Investigator Dr. Julien Schmaltz

1d. Renewed Application N/A

2 Summary

2a. English Summary Embedded systems are extremely complex, and, as demonstrated many times (*e.g.*, the Intel FDIV bug, Ariane V crash), it is difficult to build them correctly. One of the reasons is that their design is supported by *ad hoc* methodologies lacking a formal basis. Formal methods contribute general mathematical theories that support the development of reliable systems. Processing elements have been the topic of most of the verification efforts. Communication modules have received little attention. There is no general formal verification method for this class of systems.

To handle their complexity, *Systems on a Chip* result from the integration of parameterized components described at a high-level of abstraction. Communications become critical for the functionality and the performance of the overall system. Networks on a chip (NoCs) emerge as a promising solution that could meet future system requirements. Their restricted environment in terms of area, power consumption, and heat dissipation brings new challenges. Restrictions on the area reduce router buffer capacity. NoCs are more prone to deadlock. Routing hops and protocols have a strong influence on the power consumption and heat dissipation.

The objective of FVDAM is to develop the *first* specification models and validation methodologies for *high-level* and *parameterized* descriptions of NoCs. We aim at a general theory dealing with the essential properties and structures common to a wide variety of architectures. FVDAM

will allow the formal specification and validation of the absence of message loss, deadlock, livelock, and the quantification of the routing performance of an NoC.

2b. Abstract for laymen Computersystemen behoren tot de meest complexe artefacten die de mens heeft geconstrueerd. Talloze voorbeelden (zoals de beroemde Intel FDIV bug die Intel naar schatting 500 M$ heeft gekost) tonen aan dat het buitengewoon moeilijk is om dit soort systemen correct te krijgen. Een van de Grand Challenges binnen de informatica betreft het correct bewijzen — ondersteund door een theorem prover — van een *compleet computersysteem* via een zogenaamde "stack proof". Een uiterst belangrijke, recente ontwikkeling binnen de hardware-industrie is de opkomst van de zogenaamde *Systems on Chip (SoC)* en *Networks on Chip (NoC)*. In een SoC worden een aantal componenten, die tradioneel op afzonderlijke chips werden ondergebracht (processoren, geheugen, bus), samen op één chip gezet. NoCs vormen een veelbelovende ontwikkeling waarbij tevens een compleet packet switching network op de chip wordt geplaatst ten behoeve van de communicatie tussen componenten. Voorziene toepassingen van deze technologie zijn alles-in-een mobiele telefoons, pace-makers en auto's. Omdat de buffercapaciteit in een NoC beperkt is en - in tegenstelling tot bijv. het internet - berichten in een NoC niet verloren mogen raken, moet men bij NoCs enorm oppassen voor het vastlopen van het netwerk, zogenaamde deadlocks. Het doel van het FVDAM project is om een algemene wiskundige theorie te ontwikkelen waarmee we - ondersteund door theorem proving software - als eerste gewenste eigenschappen — zoals het ontbreken van deadlocks — kunnen bewijzen voor een algemene klasse van NoCs. Een dergelijke theorie zou een eerste stap zijn op weg naar een stack proof voor SoCs.

3 Classification

Computer Science. The research is relevant for the NOAG-ICT theme "Methoden voor Ontwerpen en Bouwen" (Methods for Designing and Building).

4 Composition of the Research Team

This project will be carried out within the Model Based System Development (MBSD) research theme of the Institute for Computing and Information Sciences at the Radboud University Nijmegen. Dr. J. Schmaltz will act as co-promotor of the prospective PhD student. Prof. F.W. Vaandrager will act as the official promotor. The research will be performed in close collaboration with Prof. L. Pierre from the TIMA Laboratory, Grenoble, France.

name	Specialism	hrs/wk
Dr. J. Schmaltz	Theorem proving, NoCs,	6
N.N. (PhD student)	Theorem proving, NoCs	40
Prof. F.W. Vaandrager	Model Checking, Concurrency Theory	1
Prof. L. Pierre	Hardware Verification, Design Languages	p.m.

Schmaltz has submitted a Veni proposal in the round of January 2008 titled "PRENOC: Proven Refinements in Effect for Networks On a Chip". The Veni proposal is complementary to this FVDAM proposal. FVDAM will develop *specification* models and validation methods for high-level descriptions of NoCs, whereas PRENOC will develop refinement methodologies to formally verify *hardware implementations* against their specifications.

5 Research School

The research of this project will be carried out in the context of research school IPA (Institute for Programming and Algorithmic).

6 Research Proposal

6a Description of the proposed research

Research question, motivation, and expected results

Computer systems are the most complex artifacts we build everyday. The number of states of a simple digital device is several order of magnitudes larger than the number of atoms in the universe[1]. Moreover, digital systems are discrete; a single bit inversion produces a complete different behavior. As it has been demonstrated many times (*e.g.*, the Intel FDIV bug [34], the cost of which amounted to $\$\ 500,000,000$; Ariane V crash), it is difficult to build them right. In particular, because their design is supported by *ad hoc* methodologies lacking a formal basis.

The pervasive verification of computing systems means the proof of *one* correctness theorem for an *entire* system. This vision, also called *stack proof*, produces formal models of the different abstraction layers of system models and formal relations between these layers. It was first demonstrated by Bevier et al. [5]. The different components - software applications, compilers, operating systems, processors, devices and communication architectures - of computer systems form the layers of a stack. The top layer and the most abstract one is occupied by software applications. Going down in abstraction, software applications together with an operating system are compiled into machine code and run on top of processing units and memories. Their gate level description constitutes the lowest layer of the stack. The extension of this cross-layer approach to distributed systems is the mission of the Verisoft[2] project, where we contributed the formal proof of FlexRay-like time-triggered hardware at the lowest abstraction layer [52]. Verified stacks constitute a "grand challenge" for formal methods [39].

These models and relations constitute the underlying theories that enable formal verification techniques to prove properties *for all* inputs and states of a design. Algorithmic techniques *e.g.*, equivalence and property checking, offer a high degree of automation, but are restricted to fixed size models. Interactive theorem proving techniques offer powerful reasoning engines that apply to parameterized models, but their application requires human expertise. Algorithmic techniques are routinely used in industry. Only major companies can afford theorem proving experts (*e.g.*, AMD, Intel).

The results of pervasive verification efforts contribute formal theories about computer systems. This provides a deeper understanding of their mechanisms, and therefore makes them more reliable. Processors have been the focus of earlier work, where informal textbooks (*e.g.*, [42]) have been mathematically formalized [40]. With respect to communication modules, only informal descriptions are available (*e.g.*, [10]). Moreover, the sparse specific efforts dedicated to communication architectures were performed at the RTL, on very specific designs. There is *no general verification approach* for

[1] There are between 10^{69} to 10^{81} atoms in the universe. A simple digital device with 10MB has more than $10^{20,000,000}$ states.

[2] www.verisoft.de

communication hardware modules. In particular, there is *no general high-level specification and validation method* that can be used as a reference against which gate-level or RTL designs can be formally verified.

Multi Processors Systems on a Chip (MPSoCs) denote the integration on a single die of complete computer systems *i.e.*, storage and processing elements (*e.g.*, multi-cores), as well as I/O interfaces and peripherals. The progress of chip technologies (32 and 22nm) will enable the design of large SoCs, that will populate our every day environment (*e.g.*, all-in-one mobile phones, pacemakers, cars). To handle this complexity, the trend in the design community is to raise the level of abstraction of the initial design phases, and to build a new SoC as the interconnection of pre-designed *parameterized* modules, called *Intellectual Properties (IPs)*, as part of a generic design platform [58]. The challenges lie in (1) the link between the initial abstract specification and the final Register Transfer Level (RTL) implementation, and (2) the design and the validation of the communication architecture [55], which plays a crucial role in the functionality and performance of the overall system. Networks on a chip (NoCs) emerge as a promising solution that could meet future system requirements [4]. This new paradigm brings new research challenges. NoCs work in a very constrained environment *e.g.*, area, power consumption, and heat dissipation. The area restriction induces a strong limitation on the buffer capacity of intermediate nodes. Therefore, the network is likely to be overloaded, and more prone to deadlock. Due to the restricted power and heat budget, routing hops must be optimized. Livelocks have dramatic consequences with respect to power consumption and heat dissipation. Moreover, losing and re-sending messages is unacceptable in an NoC, while it is current on the Internet.

The objective of this FVDAM proposal is to develop the *first* formal specification and validation method for high-level descriptions of NoCs. We aim at a general model and verification methodology that encompasses the essential constituents of NoCs communication architectures - *i.e.*, protocols *and* topologies, routing algorithms, and scheduling policies - and applies to a wide variety of architectures. Our approach will tackle the specification of properties about the absence of message loss, deadlock, livelock, and will allow to quantify the routing performance of an NoC. In particular, FVDAM focuses on the analysis of deadlock avoidance mechanisms (DAMs). Deadlocks may be generated at the protocol or at the structural (*e.g.*, routing algorithms) level. The challenge of FVDAM lies in supporting the analysis of these *two* kinds of deadlocks for *parameterized*, or *unbounded*, models.

Methodology

GeNoC [51, 53] is a function representing a generic network model (see Fig. 1). It formalizes the interaction between three key constituents: interfaces, routing algorithms, and scheduling policies. The correctness of GeNoC is expressed by a theorem stating that messages reach their expected destination without modification of their content. Interfaces represent the protocols used to format the information sent over the networks. Routing algorithms represent the computation of message routes. Finally, scheduling policies consist of all mechanisms related to the scheduling of messages *e.g.*, bus arbitration, circuit or packet switching techniques, etc. All these modules are represented by generic functions. Their essential properties, called *proof obligations* are formalized, but not their explicit definition. The proof of the main theorem about GeNoC follows from these proof obligations only. Consequently, the GeNoC model and its correctness theorem are a *meta-model*

and a *meta-theorem* of all networks satisfying the proof obligations. For a concrete network, the corresponding proof obligations are automatically generated. Once they have been proven to be satisfied by the concrete network, it automatically follows that this network satisfies the global correctness of GeNoC.

GeNoC has been formalized in the logic of the ACL2 theorem prover [29]. ACL2 comprises a logic, a mechanized theorem prover for that logic, and a programming language. The latter is a subset of the language Common LISP. Consequently, ACL2 models are executable. The simulation of large designs is feasible, and has been proven a key aspect in industrial verification efforts [22]. This efficient execution capability makes ACL2 unique. For the applicability of our results, it is important that the same model is used for both simulation and verification.

As most initial formalizations, the original GeNoC models makes some simplifying assumptions, in order to focus on the essential aspects and to exhibit a methodology. Such a restriction is that a communication action is not considered to be the move from one node to its neighbor, but the move from the source to the destination. In this context, deadlocks are not possible. Indeed, the exact formulation of the global correctness of GeNoC states that "when message m is received at node n, message m was actually emitted by another node and destined to n". The goal of FVDAM is to elevate the GeNoC approach to the analysis of performance, deadlock, and livelock. Our objective is to formalize the correctness theorem of GeNoC to mean "every message emitted on the network *eventually* reaches its *expected* destination without modification of its content". More precisely, the goals of FVDAM are (1) to modify the GeNoC definition to represent potential deadlocks and livelocks, (2) to rephrase the GeNoC correctness theorem, (3) to identify which proof obligations must be added on every module to prove this new property, and (4) to demonstrate the applicability of the results to industrial designs.

Enabling GeNoC to analyze deadlocks and livelocks will amount to the introduction of a set of communications that are using the network but have not reached their destination yet. If this set is empty when GeNoC terminates (from a finite set of input communications), then all messages that have left their source node have eventually reached their expected destination. Thus, there was no deadlock and no livelock. The traditional way of proving such properties is to define an ordinal measure (*e.g.*, a function returning a natural number) and prove that it is decreasing over a well-founded ordering (*e.g.*, the relation $<$ over the naturals) [45] . The purpose of FVDAM is to define such a measure for GeNoC and to identify for each module the proof obligations necessary to show that this measure is decreasing. This ordinal measure will be used as a basis for a quantitative analysis of the performance of an NoC.

Importance and urgency of the research

NoCs become a hot topic in the SoC design community. From early papers on principles [4, 59], concrete implementations have been proposed. The most popular ones are Æthereal [46, 19], Octagon [27], Nostrum [36], aSoC [31], Hermes [2], Proteo [50], XPipes [9], FAUST [14]. Some of these systems have been synthesized on FPGA, on ASICS or full custom technologies (100-180nm). The overall functionality and performance of an SoC will depend in a large extend on the communication architecture *i.e.*, on the NoC. As SoCs become ubiquitous, especially in safety critical applications (*e.g.*, pacemakers, fly-by-wire, brake-by-wire), it is necessary to formally verify their design. Our research objective is a general model and a verification methodology that applies to most of the

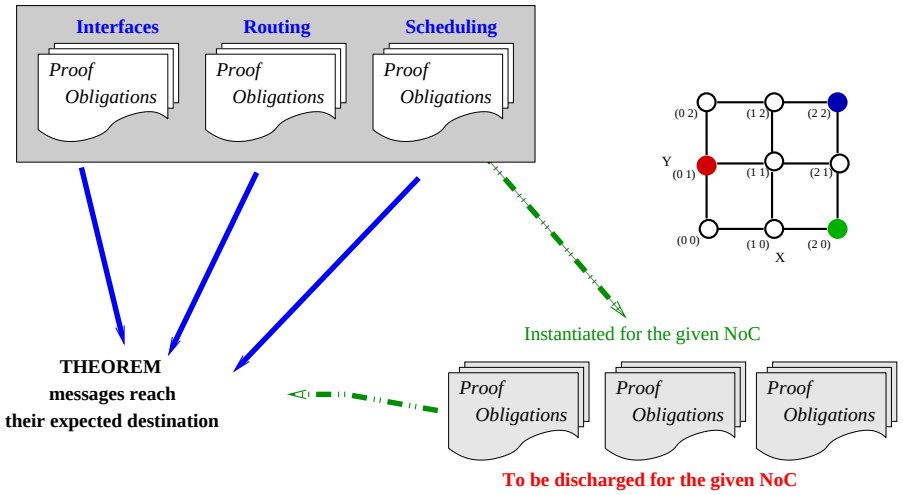

Figure 1: The GeNoC Approach

above implementations.

All mentioned systems – except NOSTRUM – implement deterministic routing algorithms. The future generation of NoCs will be built on 32nm or 22nm technologies and will face soft errors due to Deep Sub-micron issues. Not all of these soft errors will be corrected. Hence, this generation of NoCs will have to be adaptive and fault-tolerant *i.e.*, much more complex than the currently proposed solutions. In this adaptive scenario, deadlocks are even more likely to occur.

Related Work

Formal Verification of Communication Architectures Bus architectures, and their protocols, have been the subject of the earlier works on that topic. Roychoudhury *et al.* use the SMV model checker [35] to debug an academic implementation of the AMBA AHB protocol [48]. Their model is written at the register transfer level and without any parameter. Roychoudhury *et al.* detect a live lock scenario that was caused by the implementation of their arbiter rather than by the protocol itself. More recently, Amjad [1] used a model checker, implemented in the HOL [21] theorem prover, to verify the AMBA APB and AHB protocols, and their composition in a single system. Using model checking, safety properties are verified on each protocol individually. The HOL tool is used to verify their composition. In this work also, the model is at a low level of abstraction, and without any parameter.

NoCs are a more recent design paradigm, and little work has been done about their formal verification outside straightforward model checking on fixed structures. A notable exception is the work of Gebremichael *et al.* [16], who recently specified the Æthereal protocol [18] of Philips in the PVS logic [41]. The main property they verified is the absence of deadlock for an arbitrary number of masters and slaves.

Formal Theories for Communication Architectures. It is worth noting that the above mentioned formal verification efforts, devoted to communication architectures and protocols, were per-

formed at the RTL, on very specific designs. In contrast, the research results that we now review were more influential to our own research, as they tackle the formalization from a generic perspective. As such, they laid important landmarks on the path to the definition of a general theory to reason about communications.

Moore [38] defines a formal model of asynchrony by a function in the Boyer-Moore logic [7], and shows how to use this general model to verify a biphase mark protocol. More recently, Herzberg and Broy [25] presents a formal model of stacked communication protocols, in the sense of the OSI reference model. In a relational framework supporting a component-oriented view, they define operators and conditions to navigate between protocol layers. Herzberg and Broy's framework considers all OSI layers. Thus, it is more general than Moore's work, which is targeted at the lowest layer. In contrast, Moore provides mechanized support. Both studies focus on protocols and do not consider the underlying interconnection structure explicitly.

In the context of time-triggered architectures, the seminal work of Rushby [49] proposes a general model of timed-triggered implementations and their synchronous specifications. The simulation relation between these two models is proven for a large class of algorithms using an axiomatic theory, in a similar spirit of our generic model and its proof obligations. Pike recently improves the application domain of this theory [44, 43]. Miner *et al.* [37] define a unified fault-tolerant protocol acting as a generic specification framework that can be instantiated for particular applications. These studies focus on time-triggered protocols. Our work aims at a more general network model, and concentrates on the actual interconnect rather than the protocols based on top of this structure.

Deadlock Avoidance in NoCs. A huge body of work has been devoted to the study of deadlocks in routing algorithms [11, 12, 13, 15, 54, 56]. The principle of these techniques is to build a dependency graph (*e.g.*, channel or buffer dependency graph) from the global routing function. Then, there is no deadlock if there is no cycle in this graph. These techniques are efficient for *fixed* size systems. They do not apply to *parameterized* (or *unbounded*) system. Regarding this class of system, Gebremichael *et al.* [16] have proven in PVS [41] deadlock avoidance of Æthereal for an arbitrary number of masters and slaves. Their work applies to a very specific design described at the RTL. Our project will explore techniques for checking deadlock avoidance for a general class of parameterized systems and at a higher-level of abstraction, following the current trend in the SoCs design community.

With respect to protocol deadlocks, control flow mechanisms (*e.g.*, credits or Time Division Multiple Access [10]) developed for parallel computer systems are applicable to NoCs. Nevertheless, because of the limitation on storage and computation resources in NoCs and the fact that the protocol stack is mostly implemented in hardware, the design constraints and optimization goals are fundamentally different. A recent study explores issues and solutions regarding protocol deadlock avoidance in the context of NoCs [23]. The FVDAM project will complement this *design* solution with the capability to formally check that the actual NoC meets the expected performance and functionalities.

Program Termination. GeNoC is a function, which provides a computational model for NoCs. More precisely, GeNoC is recursive and can be proved to terminate. Our goal is to formalize our global property as a decreasing measure over a well-formed ordering. This is exactly the formulation of termination proof in program analysis. Once we have formalized the global measure and identified how each component of the networks contributes to decrease it – maybe by defining specific measures

for each one of them – we will be able to link our model and methodology to powerful techniques dedicated to termination proof, like Context Calling Graphs [33] or the Terminator tool [8].

Embedding of the Research

The project will be carried out within the chair "Informatics for Technical Applications" (ITA), which is part of the newly formed section Model-Based System Development within the Institute of Computing and Information Sciences (ICIS) at the Radboud University Nijmegen.

The research mission of ITA is to carry out fundamental research on formal methods and tools for the specification, design, analysis and testing of computer systems (with focus on embedded systems, distributed algorithms and protocols), and to demonstrate and assess the effectiveness of using these methods and tools in the industrial software development process. Main scientific achievements include the development of the hybrid and timed I/O automata modeling framework [32, 30] (together with the team of Nancy Lynch at MIT), the work on model based testing, contributions to the timed automata model checker UPPAAL, and the application of (timed) model checking and theorem proving technology to dozens of complex, industrial problems (see *e.g.*, [3, 17, 57, 24]). In May 2004, an International Review Committee rated the research program of ITA as "Excellent". The ITA team is involved in a number of international research projects, in particular the EU FP7 project QUASIMODO, the EU Marie Curie network TAROT, the EU IST network of excellence Artist2, and the NWO/DFG project VOSS2. The group has close ties with the Dutch Embedded Systems Institute (ESI) and has been / is involved in larger industrial collaboration projects such as BODERC and OCTOPUS (with OCE Technologies) and TANGRAM (with ASML). From the current ITA research projects, the one that is most closely related to this proposal is the NWO project FRAAI (Fault-tolerant Real-time Algorithms Analyzed Incrementally), which aims at establishing links between different abstraction layers for analysis of distributed algorithms.

This project will be carried out in close collaboration with the "Verification and modeling of Digital Designs" (VDS) group of Prof. Laurence Pierre at TIMA Laboratory in Grenoble, France. The VDS group is a pioneer and leading institute in the application of formal methods to hardware systems. They have experienced in a wide range of techniques: symbolic simulation, model checking, theorem proving, etc.

We have excellent contacts with industry partners. STMicroelectronics and NXP Semiconductors expressed strong interest in the project, and offered to provide relevant case studies. STMicrolectronics and NXP are leading chip manufacturers and contributed pioneered design solutions [28, 20]. Our utmost goal is to demonstrate our results on realistic case studies. Therefore, these contacts with industry are extremely valuable for our project.

We received strong support from J Strother Moore of the University of Texas at Austin, USA. Moore is a co-author, with Boyer and Kaufmann, of the ACL2 theorem proving system. The ACL2 authors were awarded the prestigious 2005 ACM Software Systems Awards.

6b Application perspective

As defended by Tony Hoare [26], computer science *is* a science, in the sense that one can develop general mathematical theories about computer systems, and one can carry out experiments to assess the validity of these theories. As mentioned earlier in this proposal, processors have been the focus of earlier work, but communication hardware modules have received little attention so far. This FVDAM project will contribute the first general model and verification methodology that applies

to a large class of NoCs. This will constitute an important step towards a general mathematical theory of communication modules.

In a long term prospective, FVDAM will contribute to show that formal methods can be applied *cost effectively* to realistic systems. Theorem proving techniques are capable to verify extremely large and parameterized systems, but are today too expensive to be widely used. One innovation of FVDAM is to attach proof obligations to a generic model and to reduce the verification of instances of this model to the proof of instances of the proof obligations. This lowers the verification of a complete system to the verification of properties about local components. Moreover, FVDAM will provide methodologies to develop instances of the generic model, as well as guidelines to prove the corresponding instances of the proof obligations. Altogether, this will make proofs easier to develop, reduce the level of expertise required to use theorem provers, and, finally, improve the productivity of verification engineers.

7 Project Planning

Fasering

Year 1 Bibliography & first case study

- 1 – 6 Detailed bibliography study about NoCs
- 7 – 12 Analysis of a simple case study

Year 2 More case studies and first generalization

- 1 – 6 Analysis of different case studies with different routing algorithms, scheduling policies and protocols
- 7 – 12 Extract a general methodology to define and prove the measures of the case studies done so far

Year 3 Systematic approach & realistic case study

Application of the extracted systematic approach to realistic case studies, *e.g.*, provided by industry partners.

Year 4 Writing up the manuscript

The training and education of the PhD student will be mostly fulfilled by attending basic courses and spring/fall days organized by the IPA research school. The PhD student will also visit at least one international research school.

Opleidingsaspecten

- As an initial case study, we plan to start with the analysis of a simplified version or sub-parts of the NoC architecture of NXP [47], or with the academic achitecure Hermes [2], for which we already built a partial specification [6].

- Currently, GeNoC has been implemented in ACL2, which seems to be an adequate tool. It provides an interesting logic and an efficient execution environment. The logic is first order, and it might be that we need higher order idioms to express our deadlock free conditions. If this is the case, we will switch to theorem provers for higher order logics. For instance, we have experience with the Isabelle [52] and PVS systems [57].

8 Expected Use of Instrumentation

Not Applicable

9 Literature

The following papers are the 5 most important publications of the research team relevant to the proposed research:

1. J. Schmaltz and D. Borrione.
   A Functional Formalization of On Chip Communications
   Formal Aspects of Computing, Accepted for publication, Springer, 2007
   DOI: 10.1007/s00165-007-0049-0

2. D. Borrione, A. Helmy, L. Pierre and J. Schmaltz.
   A Generic Model for Formally Verifying NoC Communication Architectures: A Case Study
   1st International Symposium on Networks on Chip (NoCS'07), pp 127-137, Princeton, New Jersey, May 7-9, USA, IEEE Press Society 2007.

3. J. Schmaltz and D. Borrione. [3]
   A Functional Approach to the Formal Specification of Networks on Chip
   Formal Methods in Computer-Aided Design (FMCAD'04), LNCS 3312, pp 52-66, Springer-Verlag, Austin, Texas, USA, November 14-17, 2004.

4. F. W. Vaandrager and A. de Groot.
   Analysis of a biphase mark protocol with Uppaal and PVS
   Formal Aspects of Computing, 18(4):433–458, Springer 2006.

5. B. Gebremichael, F. W. Vaandrager, M. Zhang, K. Goossens, E. Rijpkema and A. Radulescu.
   Deadlock Prevention in the Æthereal Protocol.
   Correct Hardware Design and Verification Methods (CHARME'05), pp. 345–348, LNCS 3725, Springer 2005

10 Requested Budget

The following numbers are communicated by the financial department of the ICIS at Radboud University.

[3] First published result on the formal verification of NoCs.

a)	PhD Student	=	194,781 kEuros
b)	personal benchfee	=	5 kEuros
c)	additional traveling budget	=	0 kEuros
d)	project related apparatus/software	=	0
	Total PhD Student	=	199,871 kEuros

References

[1] H. Amjad. Model Checking the AMBA Protocol in HOL. Technical report, University of Cambridge, Computer Laboratory, September 2004.

[2] A. Bartic, J-Y. Mignolet, V. Nollet, T. Marescaux, D. Verkest, S. Vernalde, and R. Lauwereins. Highly Scalable Network on Chip for Reconfigurable Systems. In *Proceedings of the International Conference on System-On-Chip 2003 (SoC'03)*, pages 79–82, 2003.

[3] J. Bengtsson, W.O.D. Griffioen, K.J. Kristoffersen, K.G. Larsen, F. Larsson, P. Pettersson, and Wang Yi. Verification of an audio protocol with bus collision using UPPAAL. pages 244–256.

[4] L. Benini and G. De Micheli. Networks on Chips: A New SoC Paradigm. *Computer*, 35(1):70–78, 2002.

[5] W.R. Bevier, W.A. Hunt, J Strother Moore, and W.D. Young. An approach to systems verification. *Journal of Automated Reasoning*, 5(4):411–428, 1989.

[6] D. Borrione, A. Helmy, L. Pierre, and J. Schmaltz. A Generic Model for Formally Verifying NoC Communication Architectures: A Case Study. In *Proc. of First International Symposium on Networks-on-Chip (NOCS'07)*, pages 127–136, Princeton, NJ, USA, 7-9 May 2007. IEEE.

[7] R. S. Boyer and J Strother Moore. *A Computation Logic Handbook*. Academic Press, 1988.

[8] B. Cook, A. Podelski, and A. Rybalchenko. Terminator: Beyond safety. In *CAV*, pages 415–418, 2006.

[9] M. Dall'Osso, G. Biccari, L. Giovannini, D. Bertozzi, and L. Benini. Xpipes: a Latency Insensitive Parameterized Network-on-chip Architecture for Multi-Processor SoCs. In *ICCD*, pages 536–, 2003.

[10] W.J. Dally and B. Towles. *Principles and Practices of Interconnection Networks*. Morgan-Kaufmann Publisher, 2004.

[11] J. Duato. A New Theory of Deadlock-Free Adaptive Routing in Wormhole Networks. *IEEE Transactions on Parallel and Distributed Systems*, 4(12):1320–1331, 1993.

[12] J. Duato. A Necessary and Sufficient Condition for Deadlock-Free Adaptive Routing in Wormhole Networks. In *International Conference on Parallel Processing*, pages 142–149, 1994.

[13] J. Duato. A Necessary and Sufficient Condition for Deadlock-Free Routing in Cut-Through and Store-and-Forward Networks. *IEEE Transactions on Parallel and Distributed Systems*, 7(8):841–854, 1996.

[14] Y. Durand, C. Bernard, and D. Lattard. FAUST: On-chip distributed architecture for a 4G baseband modem SoC. In *IP-SoC*, 2005.

[15] E. Fleury and P. Fraigniaud. A General Theory for Deadlock Avoidance in Wormhole-Routed Networks. *IEEE Transactions on Parallel and Distributed Systems*, 9(7):626–??, 1998.

[16] B. Gebremichael, F. Vaandrager, M. Zhang, K. Goossens, E. Rijpkema, and A. Rădulescu. Deadlock Prevention in the Æthereal protocol. In D. Borrione and W.J. Paul, editors, *Correct Hardware Design and Verification Methods (CHARME'05)*, volume 3725 of *LNCS*, pages 345–348, 2005.

[17] B. Gebremichael, F.W. Vaandrager, and M. Zhang. Analysis of the Zeroconf protocol using Uppaal. In *Proceedings 6th Annual ACM & IEEE Conference on Embedded Software (EMSOFT 2006),* Seoul, South Korea, October 22-25, 2006, pages 242–251. ACM Press, 2006.

[18] K. Goossens, J. Dielissen, and A. Rădulescu. Æthereal Network on Chip: Concepts, Architectures, and Implementations. *IEEE Design and Test of Computers*, 22(5):414–421, September-October 2005.

[19] K. Goossens, J. Dielissen, and A. Rădulescu. The Æthereal network on chip: Concepts, architectures, and implementations. *IEEE Design and Test of Computers*, 22(5):21–31, September-October 2005.

[20] K. Goossens, J. van Meerbergen, A. Peeters, and P. Wielage. Networks on Silicon: Combining Best-Effort and Guaranteed Services. In *Proc. of Design Automation and Test in Europe Conference and Exhibition (DATE'02)*, pages 423–425, 2002.

[21] M.J.C. Gordon. HOL: A Proof Generating System for Higher-Order Logic. In G. Birthwislte and P.A. Subrahmanyam, editors, *VLSI Specification, Verification and Synthesis*, pages 73–128, Boston, 1987. Kluwer Academic Publishers.

[22] D. Greve. Symbolic Simulation of the JEM1 Microprocessor. In Ganesh Gopalakrishnan and Phillip Windley, editors, *Formal Methods in Computer-Aided Design (FMCAD '98)*, volume 1522, pages 321–333, Palo Alto, CA, 1998. Springer-Verlag.

[23] A. Hansson, K. Goossens, and A. Rădulescu. Avoiding message-dependent deadlock in network-based systems on chip. *VLSI Design*, 2007:Article ID 95859, 10 pages, May 2007. Hindawi Publishing Corporation.

[24] M. Hendriks, N. J. M. van den Nieuwelaar, and F. W. Vaandrager. Model checker aided design of a controller for a wafer scanner. *Software Tools for Technology Transfer*, pages 1–15, 2006. Special Section on Quantitative Analysis of Real-time Embedded Systems.

[25] D. Herzberg and M. Broy. Modeling Layered Distributed Communication Systems. *Formal Aspects of Computing*, 17(1):1–18, 2005.

[26] Tony Hoare. The ideal of verified software. In *ACL2 '06: Proceedings of the sixth international workshop on the ACL2 theorem prover and its applications*, pages 61–62, New York, NY, USA, 2006. ACM.

[27] F. Karim, A. Nguyen, and S. Dey. An Interconnect Architecture for Networking Systems On Chip. *IEEE Micro*, pages 36–45, September-October 2002.

[28] F. Karim, A. Nguyen, S. Dey, and R. Rao. On-Chip Communication Architecture for OC-768 Network Processor. In 38^{th} *Design Automation Conference (DAC'01)*, pages 678–683, 2001.

[29] M. Kaufmann, P. Manolios, and J Strother Moore. *ACL2 Computer Aided Reasoning: An Approach*. Klulwer Academic Press, 2000.

[30] D.K. Kaynar, N.A. Lynch, R. Segala, and F.W. Vaandrager. *The Theory of Timed I/O Automata*. Morgan & Claypool Publishers, 2006. Synthesis Lecture on Computer Science, 101pp, ISBN 159829010X.

[31] J. Liang, S. Swaminathan, and R. Tessier. asoc: A Scalable, Single-Chip Communications Architecture. In *IEEE PACT*, pages 37–46, 2000.

[32] N.A. Lynch, R. Segala, and F.W. Vaandrager. Hybrid I/O automata. *Information and Computation*, 185(1):105–157, 2003.

[33] P. Manolios and D. Vroon. Termination analysis with calling context graphs. In *CAV*, pages 401–414, 2006.

[34] J. Markoff. Circuit Flaw Causes Pentium to Miscalculate: Intel Admits. *New York Times*, 24th November 1994. See also http://mathworks.com/company/pentium/index.html.

[35] K. L. McMillan. *Symbolic Model Checking*. Kluwer Academic Press, 1993.

[36] M. Millberg, E. Nilsson, R. Thid, S. Kumar, and A. Jantsch. The Nostrum Backbone - a Communication Protocol Stack for Networks on Chip. In *VLSI Design*, pages 693–696, 2004.

[37] P. S. Miner, A. Geser, L. Pike, and J. Maddalon. A Unified Fault-Tolerance Protocol. In Y. Lakhnech and S. Yovine, editors, *Formal Techniques, Modeling and Analysis of Timed and Fault-Tolerant Systems (FORMATS-FTRTFT)*, volume 3253 of *LNCS*, pages 167–182. Springer, 2004.

[38] J Strother Moore. A Formal Model of Asynchronous Communications and Its Use in Mechanically Verifying a Biphase Mark Protocol. *Formal Aspects of Computing*, 6(1):60–91, 1993.

[39] J Strother Moore. A Grand Challenge Proposol for Formal Methods: A Verified Stack. In B. K. Aichernig and T. S. E. Maibaum, editors, 10^{th} *Anniversary Colloquium of UNI/IIST*, volume 2757 of *LNCS*, pages 161–172. Springer, 2003.

[40] S.M. Müller and W.J. Paul. *Computer Architecture, Complexity and Correctness*. Springer Verlag, 2000.

[41] S. Owre, J.M. Rushby, and N. Shankar. PVS: A Prototype Verification System. In D. Kapur, editor, *Eleventh International Conference on Automated Deduction (CADE'92)*, volume 607 of *LNAI*, pages 748–752, Saragota, NY, June 1992. Springer-Verlag.

[42] D. A. Patterson and J. L. Hennessy. *Computer architecture: a quantitative approach*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990.

[43] L. Pike. Modeling time-triggered protocols and verifying their real-time schedules. In *Proceedings of Formal Methods in Computer Aided Design (FMCAD'07)*. IEEE, 2007.

[44] Lee Pike. A note on inconsistent axioms in rushby's "systematic formal verification for fault-tolerant time-triggered algorithms". *IEEE Transactions on Software Engineering*, 32(5):347–348, May 2006.

[45] S. Ray and J Strother Moore. Proof styles in operational semantics. In *FMCAD*, pages 67–81, 2004.

[46] A. Rădulescu, J. Dielissen, S. González Pestana, O.P. Gangwal, E. Rijpkema, P. Wielage, and K. Goossens. An Efficient On-Chip Network Offering Guaranteed Services, Shared-Memory Abstraction, and Flexible Network Programming. *IEEE Transactions on CAD of Integrated Circuits and Systems*, 24(1), 2005.

[47] E. Rijpkema, K. G. W. Goossens, A. Radulescu, J. Dielissen, J. van Meerbergen, P. Wielage, and E. Waterlander. Trade offs in the design of a router with both guaranteed and best-effort services for networks on chip. In *DATE '03: Proceedings of the conference on Design, Automation and Test in Europe*, page 10350, Washington, DC, USA, 2003. IEEE Computer Society.

[48] A. Roychoudhury, T. Mitra, and S.R. Karri. Using Formal Techniques to Debug the AMBA System-on-Chip Bus Protocol. In *Design Automation and Test Europe (DATE'03)*, pages 828–833, 2003.

[49] John Rushby. Systematic formal verification for fault-tolerant time-triggered algorithms. *IEEE Transactions on Software Engineering*, 25(5):651–660, sep 1999.

[50] I. Saastamoinen, M. Alho, and J. Nurmi. Buffer implementation for Proteo networks-on-chip. In *International Symposium on Circuits and Systems (ISCAS'03)*, pages 113–116, 2003.

[51] J. Schmaltz. *Une formalisation fonctionnelle des communications sur la puce.* PhD thesis, Joseph Fourier University, Grenoble, France, January 2006. In French. Available at www.cs.ru.nl/∼julien/. A partial translation is available upon request to the author.

[52] J. Schmaltz. A Formal Model of Clock Domain Crossing and Automated Verification of Time-Triggered Hardware. In J. Baumgartner and M. Sheeran, editors, *Formal Methods in Computer-Aided Design (FMCAD'07)*, Austin, TX, USA, 11-14 November 2007. IEEE/ACM.

[53] J. Schmaltz and D. Borrione. A functional formalization of on chip communications. *Formal Aspects of Computing*, 2007. DOI: 10.1007/s00165-007-0049-0.

[54] L. Schwiebert and D.N. Jayasimha. A Universal Proof Technique for Deadlock-Free Routing in Interconnection Networks. In 7^{th} *Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA'95)*, pages 175–184, 1995.

[55] G. Spirakis. Beyond Verification: Formal Methods in Design. In A. Hu and A.K. Martin, editors, *Formal Methods in Computer-Aided Design (FMCAD'04)*, volume 3312 of *LNCS*, Austin, Texas, USA, November 2004. Springer-Verlag. Invited Speaker.

[56] S. Taktak, E. Encrenaz, and J.-L. Desbarbieux. A tool for automatic detection of deadlock in whormhole networks on chip. In *11th IEEE International Workshop on High-Level Design Verification and Test (HLDVT'2006)*, 2006.

[57] Frits W. Vaandrager and Adriaan de Groot. Analysis of a biphase mark protocol with uppaal and pvs. *Formal Asp. Comput.*, 18(4):433–458, 2006.

[58] J. van Meerbergen. Networks on chip: A communication-centric approach to platform-based design. In *PROGRESS White Papers 2006*. STW, The Netherlands.

[59] T. Tao Ye, L. Benini, and G. De Micheli. Packetized On-Chip Interconnect Communication Analysis for MPSoC. In *DATE '03: Proceedings of the conference on Design, Automation and Test in Europe*, pages 344–349. IEEE Computer Society, 2003.