Software-Defined Hardware (SDH)

Applicant name: Martin Schöberl
Created: Tuesday, September 29, 2020 kl. 11:40 PM
Number of appendices: 15
Funding body: Danmarks Frie Forskningsfond | Teknologi og Produktion
Instrument: DFF-Forskningsprojekt2/DFF-Research Project 2

Table of contents

APPLICANT

READING GUIDE

The information provided in the digital application form is incorporated in this PDF document.
The sign [x] indicates that a positive answer has been provided in the application form.
The sign [ ] indicates that either a negative answer or no answer has been provided in the application form

PERSONAL INFORMATION AND APPLIED COUNCIL

First name:	Martin
Surname:	Schöberl
E-mail address:	masca@dtu.dk
Highest attained academic degree:	Other
Academic degree:	Habilitation
Gender:	Male
Council:	Independent Research Fund Denmark | Technology and Production
Date of birth:	02-11-1966

CITIZENSHIP

Citizenship:	Austria
Do you hold a dual citizenship?:	No

OPEN RESEARCHER AND CONTRIBUTOR ID

ORCID:	0000-0003-2366-382X

PHD DEGREE

Have you obtained a PhD degree?:	[X]
State the date of obtaining the degree, as stated on your PhD diploma:	11-04-2005
Have you had any periods of leave since obtaining your PhD degree?:	[]

CONTRIBUTION TO SCIENCE

Give an account of your most significant contributions to science :	Defining the research area of time-predictable computer architecture, as part of my habilitation at TU Vienna. Publishing relevant papers in this area.

PLACE OF EMPLOYMENT, ETC.

CURRENT PLACE OF EMPLOYMENT

Are you currently employed?:	[X]
Current position:	Associate professor
Choose place of employment:	Technical University of Denmark
Department:	DTU Compute

OTHER

DANISH

CVR number:	30060946

PLACE OF EMPLOYMENT DURING THE PROJECT PERIOD

Is the place of employment during the project period the same as your current place of employment?:	[X]

PLACE FOR CARRYING OUT THE PROJECT ACTIVITIES

Will the research activities be carried out at the place of employment (the project period)?:	[X]

RESEARCH ACTIVITIES ABROAD

Does the project involve research activities outside Denmark (excl. conference participation)? :	Yes
Total number of months for applicant:	3
Total number of months for all PhDs:	6
Total number of months for all Postdocs :	0
Total number of months for other scientific/academic persons:	0

TITEL AND SCIENTIFIC CONTENT

APPLICATION TITLE

Application title:	Software-Defined Hardware (SDH)

BRIEF SCIENTIFIC SUMMARY

The brief scientific summary should be written in English. The summary is used in the recruitment of peer reviewers.:

Performance increase with general-purpose processors has come to a halt. We can no longer depend on Moore's Law to increase computing performance. The only way to achieve higher performance or lower energy consumption is by building domain-specific hardware accelerators. These accelerators can be built in ASICs or in FPGAs in the cloud. To efficiently design and verify those domain-specific accelerators, we need agile hardware development.

This project aims to develop a method and concrete tools for agile hardware development. We will use tools, languages, development, and testing methods from the last decades in software development and apply them to hardware design. We aim to raise the tooling level for a digital design to increase productivity. Time for verifying (testing) of digital systems is about double the time of developing them in the first place. Therefore, this project's central focus is on applying software development testing methods to hardware development.

POPULARISED DESCRIPTION OF THE SCIENTIFIC CONTENT

Popularised description of the scientific content:

Digitale systemer er en central del af vores nuværende og fremtidige digitalt forbedrede liv. Danmark har en betydelig industri inden for design og udvikling af digitale systemer. Dog er design og verifikation af sådanne systemer en stadigt voksende udfordring. Hovedproblemet er brugen af gamle design og verifikations værktøjer i forhold til programmerings værktøjer til disse enheder. Dette projekt sigter mod at anvende værktøjer og metoder fra softwareudvikling til hardwareudvikling for at øge produktiviteten, og danske virksomheder vil drage fordel af resultaterne af dette projekt.

SCIENTIFIC KEYWORDS

Keyword 1:	digital design
Keyword 2:	digital hardware
Keyword 3:	testing
Keyword 4:	verification

MAIN AREAS OF SCIENCE

Technical sciences:	100
Sum:	100%

CLASSIFICATION CODES (OECD)

CLASSIFICATION CODE

Classification code 1:	Computer hardware and architecture
Classification code 2:	Communication engineering and systems
Classification code 3:	Computer sciences

CLASSIFICATION CODE 1

ID:	20203
Classification code :	Computer hardware and architecture

Classification code English:	Computer hardware and architecture

CLASSIFICATION CODE 2

ID:	20204
Classification code:	Communication engineering and systems
Classification code English:	Communication engineering and systems

CLASSIFICATION CODE 3

ID:	10201
Classification code:	Computer sciences
Classification code English:	Computer sciences

BUDGET

Uploaded by:	Martin Schöberl
Uploaded, date and time:	9/25/2020 1:18 PM

BUDGET INFORMATION

DURATION OF THE PROJECT

Starting date:	01-09-2021
End Date:	31-08-2025
Duration in months:	48

SALARY EXPENSES

Will you be salaried by the grant applied for?:	Not salaried by the grant applied for

AMOUNT APPLIED FOR

Amount for scientific/academic salaries:	3.353.902 kr.
Amount for technical/administrative salaries:	0 kr.
Amount for equipment:	10.000 kr.
Amount for operating expenses:	840.000 kr.
Amount applied for excl. overhead/administration expenses:	4.203.902 kr.
Amount for overhead/administration expenses:	1.849.715 kr.
Amount applied for incl. overhead/administration expenses:	6.053.617 kr.

ADMINISTRATOR

CVR number:	30060946
P number:	1003403265
Administrator:	Technical University of Denmark
Organisation type:	Dansk: Universitet
Department/institute:	DTU Compute
Address:	Anker Engelunds Vej 101A
City:	Kgs. Lyngby
Postcode:	2800
Email address:	projektrevision@adm.dtu.dk
Country:	Denmark

PERSONAL INFORMATION

PERSONAL INFORMATION

Number of persons:	5

PERSON 1

First name:	Martin
Last name:	Schoeberl
E-mail:	masca@dtu.dk

PERSON 2

First name:	Jan
Last name:	Madsen
E-mail:	jama@dtu.dk

PERSON 3

First name:	Zhoulai
Last name:	Fu
E-mail:	zhfu@itu.dk

PERSON 4

First name:	Peter
Last name:	Sestoft
E-mail:	sestoft@itu.dk

PERSON 5

First name:	Kasper
Last name:	Hesse
E-mail:	s183735@win.dtu.dk

CONSENT TO SUBMITTING YOUR APPLICATION FOR PEER REVIEW IN THIRD COUNTRIES

If applicable, do you consent to submitting your application for external peer review in third countries?:	Yes

ETHICAL ISSUES AND GATHERING DATA

ETHICAL ISSUES

Does your research raise any ethical issues that should be dealt with?:	No

GATHERING DATA

Does your project involve gathering or purchase of quantitative data within the research areas of social sciences, medical sciences or the humanities?:	No

GENDER COMPOSITION

GENDER COMPOSITION

Gender composition :	As the already named researchers are all male, we will actively search for female researchers for the PhD positions. However, the PhD position will be announced openly and men and women will have equal opportunities for applying.

SUBMISSION TO SEVERAL COUNCILS

SUBMISSION TO SEVERAL COUNCILS

Do you request your application also to be assessed by another of Independent Research Fund Denmark's research councils?:	[]

REFERRAL TO OTHER RESEARCH COUNCILS

Do you accept that Independent Research Fund Denmark may refer your application to a different research council than the one you originally applied to, if Independent Research Fund Denmark considers it to be scientifically relevant?:	[X]

OTHER APPLICATIONS

PREVIOUS APPLICATION(S)

Is this application a resubmission of one or several previous applications, whether in the same or in a revised form?:	[]

OTHER SOURCES

Have you applied for funding for activities covered by this application from other sources, including foundations, councils or programme committees?:	[]

PREVIOUS GRANTS

PREVIOUS GRANTS

Have you previously received any funding for activities related to the present application?:	[]
Have you previously received any funding for activities not related to the present application?:	[X]

PREVIOUS GRANTS TO OTHER PROJECT ACTIVITIES THAN THE PROPOSED PROJECT ACTIVITIES

Year:	2015
Amount awarded:	5.170.177 kr.
Granted by:	DFF
Project title:	Time-predictable Control Systems (PREDICT)
Please describe the scientific content and the results of the grant.:	The project targets computer-based control systems such as robot control systems, factory automation systems, and fly-by-wire systems in airplanes. One common aspect of such systems is that at least some fraction of the functionality must be time-predictable in order to guarantee real-time requirements. The aim of the proposed project is to combine the expertise of the Drone Research Laboratory at AAU in guidance and navigation of drones, and in obstacle detection and avoidance systems with DTU Compute's expertise on time-predictable multi- core processors and programming of multi-core processors. The project has no relation to the proposed project.

PROJECT MANAGEMENT

If you are PI of one or more running grants of at least 1.000.000 DKK::	I am PI for the currently running PREDICT project. However, the PREDICT project will finish June 2021 and there will be no management overlap between PREDICT and this application.

EXCLUDING REVIEWERS FROM EXTERNAL PEER REVIEW

EXCLUDE REVIEWER

Do you want to request not to involve named researchers from being involved in a possible external peer review of your application?:	[]

DFF Project Description
Software-Defined Hardware (SDH)

Martin Schoeberl, DTU Compute

1 Introduction and Objectives

We can no longer depend on Moore's Law to increase computing performance [16]. Performance increase with general-purpose processors has come to a halt. The only way to achieve higher performance or lower energy consumption is by building domain-specific hardware accelerators [13]. These accelerators can be built in chips or in FPGAs in the cloud. The production of a chip is costly. Therefore, it is essential to get the design right at the first tape-out. Thorough testing and verification of the design is mandatory.

To efficiently develop and verify those accelerators, we can learn from software development trends such as agile software development [10]. We believe that **we need to adapt to agile hardware development** [20].

Furthermore, as accelerators become part of the cloud service, i.e., FPGAs in the cloud, software developers will increasingly need to adapt critical algorithms to FPGAs to enhance performance. Hence, it is imperative to make accelerator design accessible for software developers. By adapting hardware accelerator design to the methods and tools of contemporary software design, it is possible to bridge both domains catering for a more uniform hardware/software development process.

Until a few years, the two main design languages Verilog and VHDL dominated the design and testing of digital circuits. However, compared to software development and testing, digital design and testing methods and tools lack several decades of development. Within this project, we plan to **leverage software development and testing methods for digital design.** This project explores the hardware construction language Chisel [8] with Scala and the Universal Verification Method (UVM) with SystemVerilog [4] for the design and test of digital systems.

This project aims to develop a method and concrete tools for agile hardware development. We will use tools, languages, development, and testing methods from the last decades in software development and apply them to hardware design. We aim to **raise the tooling level for a digital design to increase productivity**. Time for verifying (testing) of digital systems is about double the time of developing them in the first place. Therefore, this project's central focus is on **applying software testing methods for hardware testing**.

We will build a combination of open-source tools for verifying circuits described in mixed languages (VHDL, SystemVerilog, and Chisel). It builds on top of the Chisel hardware construction language and uses Scala to drive the verification.

2 Background and State-of-the-Art

Digital Design. VHDL and Verilog are the classic hardware description languages, first appeared in the 1980s. SystemVerilog [4], as an extension to Verilog, adds features from VHDL for the hardware description and object-oriented features for verification. Recent advances with SystemVerilog and Chisel [8, 25] have brought object-oriented programming into the digital design and verification process.

Chisel is a "Hardware Construction Language" embedded in Scala, to describe digital circuits [8]. Scala/Chisel brings object-oriented and functional programming into the world of digital design. For hardware generation and testing, the full Scala language and Scala and Java libraries are available. As Scala and Java's full power is available to the verification engineer, the verification process is also made more efficient.

SystemVerilog has become a complex language with more than 250 keywords, and it is unclear which tools support which language constructs for hardware description. In contrast with Chisel, when the program compiles, it is synthesizable hardware. Chisel is a small language, where the cheat sheet fits on two pages. The power of Chisel comes from the embedding in Scala. Furthermore, as classic hardware description languages are niche products, not many tools or libraries are available. With Chisel on Scala we have the choice of different integrated development environments (IDE), testing infrastructure (e.g., ScalaTest), and many free libraries.

Testing Methods. Testing is crucial for making software and hardware reliable. The higher expectation of software quality and shrinking development cycle have driven programming language researchers and software engineers to develop a spectrum of *automated testing* techniques. For example, continuous integration [15] implements an agile method that automatically runs manually written

SystemVerilog adds object-oriented concepts for the non-synthesizable verification code. The SystemVerilog direct programming interface [14] allows the programmer to call C functions inside a SystemVerilog (UVM) testbench. This enables co-simulation with a "golden model" written in C, and the testbench verifying the device under test (DUT). With ChiselTest we can co-simulate with Java and Scala models and use the Java Native Interface to co-simulate with models in C.

The digital design described in Chisel can be tested and verified with ChiselTest [22], a non-synthesizable testing framework for Chisel. ChiselTest emphasizes usability and simplicity while providing ways to scale up complexity. Fundamentally, ChiselTest is a Scala library that provides access into the simulator through operations like poke (write value into circuit), peek (read value from circuit, into the test framework), and step (advance time). As such, tests written in ChiselTest are just Scala programs, imperative code that runs one line after the next. This structure uses the latest programming language developments that have been implemented into Scala and provides a clean and concise interface, unlike approaches that attempt to reinvent the wheel like UVM.

Furthermore, ChiselTest tries to enable testing best practices from software engineering. Its lightweight syntax encourages writing targeted unit tests by making small tests easy. A clear and clean test code also enables the test-as-documentation pattern, demonstrating a module's behavior from a temporal perspective.

Determining Specification. The first challenge is how to determine the expected code outcome, or called *specification*. Software developers often document specifications within the software itself. *Property-based testing* provides a convenient way of formulating specification as constraints expressed in a domain-specific language [12]. For example, developers in the Scala language can specify "$n^2 > 0$ for all integer n" in the property-based testing library ScalaCheck [24] as follows:

```
val propSquare = forAll {(n: Int) => n * n > 0}
```

An efficient approach to obtaining such specifications is to learn from historical bug patterns. Such patterns are gathered in pattern-based software analysis tools, e.g., in FindBug [7] or Scalafix [3]. Recently, a specification for robotic software is obtained by studying bugs fixed on the Robot Operating System [17].

Generating Test Inputs. The second challenge is how to generate interesting test inputs. The state-of-the-practice uses *fuzzing*, which has emerged as one of the most effective testing techniques for discovering reliability issues in software [23]. Google's OSS-Fuzz project, for example, has filed over 20 000 bugs in 300 open-source projects (as of June 2020) [2]. Fuzzing techniques generate random inputs and improve them based on observed code status, e.g., crashes or code coverage [11, 18]. Property-based testing [12], which is initially designed for testing functional programming languages, uses random data generation, a form of plain fuzzing. For example, by invoking `propSquare.check` with ScalaCheck (mentioned above), where `propSquare` is the constraint $n^2 > 0$ for all integers n, we can immediately get an input n that falsifies the constraint:

```
scala> propSquare.check
      ! Falsified after 1 passed tests.
     > ARG_0: 0
     > ARG_0_ORIGINAL: 1083860448
```

ScalaCheck first finds that n = 1083860448 produces an overflow wrapped to a negative (thus $n^2 > 0$ fails). The input is then "shrunk", in the terminology of property-based testing, to a smaller one, namely 0 in this case.

3 Research Plan

As prerequisites, all researchers involved in this project will need to learn languages and tools involved in the project and related work. They will learn about the Scala programming language and ScalaCheck (a Scala implementation of the property-based testing) on the software side. On the hardware side, the researchers need to get familiar with Chisel, SystemVerilog, and UVM.

Hardware Generators. The productivity of hardware design can be greatly increased by developing so-called hardware generators. A hardware generator is a program that can generate a configurable hardware description. Scala with functional programming is an excellent basis for developing a methodology for the development of such hardware generators.

Property-based Testing. As a starting point, we will use ScalaCheck, an implementation of property-based testing in Scala, for hardware testing. We will invite hardware developers to write properties as constraints (like propSquare above), and then we will use ScalaCheck to validate or refute those constraints. Property-based testing has seen notable successes previously, such as in locating a long-standing concurrency bug in the Erlang database server [21]; it was also used by Ericsson to test its media proxy [5], by Volvo to test car communications protocols [6].

Constraint Random Testing and Fuzzing Techniques. Constraint random testing in hardware is similar to fuzzing in software testing. We will add support for constrain random testing to ChiselTest [22]. Once we have tools to instrument the hardware under test with a user-written specification or automatically generated specification, we will be able to use fuzzing techniques to validate those specifications. To this end, we will work with our industrial collaborators to get access use cases serving as the DUT.

Test Coverage. Code coverage is a useful tool for verifying digital designs since it allows one to see which parts of their design have actually been tested correctly. We will implement coverage inside of the execution engine of the Chisel simulator using a technique presented by Ira. D. Baxter [9]. We will add a method to specify *functional coverage points*, also known as *coverage groups* in SystemVerilog.

Verification Framework. We will develop an object-oriented and functional framework for verification in Scala. This framework is inspired by UVM, but will leverage Scala's conciseness with the combination of object-oriented programming with functional programming. Within our verification framework, we will support mixed language verification. Verilog can easily be combined with Chisel, as Chisel generates Verilog, and we will use ChiselTest as a driver for the open-source Verilog simulator Verlator. We plan the open-source Yosys synthesis suite [26] with the GHDL [19] to translate VHDL into Verilog.

Learning Specification from Historical Bug Patterns. We will consider generating specifications from previously known bug patterns. We will collaborate with our hardware engineers to explore a history of hardware issues triggered by software defects and get patterns from which we generate specification. Such bug patterns in Java, for example, can be found in [1]. This step will generate specification in a syntax-driven way, relating to the expected functional behavior of the hardware.

Dissemination and Publication Scientific results will be published and presented at international conferences (e.g., DATE, DAC, CAV, FPL, ISCAS, FPGA) and in relevant scientific journals and two PhD theses. We aim to publish in open access, to a large extent, in the gold open access model. However, publishers such as ACM also allow publishing in green open access at no additional cost, where a pre-print version of a paper can be uploaded, for example, to ArXiv.

The results from the project will be available as open-source under the industry-friendly BSD license. A project web site will host the project documentation, the published papers, and the design's source code. We will provide unrestricted and cost-free digital access to all research and development results. We will use the developed method and tools to train a new generation of HW/SW engineers at DTU and ITU. In the middle of the project, we will have a coordination workshop with the project partners. In the end, we will organize a design and verification workshop, including a hands-on tutorial, open to all interested companies, and students from DTU and ITU.

4 Practical Feasibility

Industrial Cooperation Danish industry in digital design for ASICs and FPGAs is currently in transition from using traditional test benches written in VHDL and Verilog to a verification method based on constraint random test generation with tools such as UVM. Therefore, the SDH project is just-in-time to support this transition.

Several companies are interested in the SDH project and support it with non-trivial use cases. Teledyne is interested in exploring Chisel for development and testing with the compatibility of VHDL and Verilog. Microchip is interested in using Chisel to improve the design throughput and provides a hardware use case to sort Ethernet packets according to their deadlines. Synopsys provides tool support for the project. WSA provides a decimation filter use case for our verification framework. Napatech will follow the project and offer challenges to the project. Syosil will provide consulting and training in connection with any questions related to functional verification and UVM.

Human Resources For the SDH project, we request funding of two PhD students and a student researcher. Each of the senior researchers will contribute to the SDH research project. As the already named researchers are all male, we will actively search for female researchers for the PhD positions.

Martin Schoeberl (MS) is associate professor at DTU Compute and is the PI. During his stay at UCB in 2012 he picked up Chisel and brought it to DTU in research and teaching. Martin has written the Chisel textbook [25], which has been translated into Chinese and Japanese. Martin is a member of the Technical Advisory Committee for Chisel and therefore keeps the work of SDH in sync with the Chisel main development. Martin is part of the regular Chisel developer meeting and a meeting on verification with UC Berkeley researchers and developers from SiFive, a startup in silicon valley.

Jan Madsen (JM) is professor and deputy director of DTU Compute. His research spans methods and tools for systems engineering of computing systems, embedded systems-on-a-chip, Cyber-Physical Systems (Internet-of-Things), microfluidic biochips (Lab-on-Chip), and synthetic biology (molecular computing).

Kasper Hesse (KH) is student at DTU with a strong background in digital design. He has already some knowledge in UVM and can therefore help the project to be productive from day one.

Zhoulai Fu (ZF) is an assistant professor at ITU. His research spans programming language theory and software engineering techniques. Related to this project, he worked on developing constraint solving and automated testing techniques in achieving high coverage and detecting bugs in numerical programs.

Peter Sestoft (PS) is professor and head of department at the IT University of Copenhagen. His research focus is programming languages, and especially functional, parallel, domain-specific and declarative languages, their description, formal modeling, analysis, transformation, and implementation. Most of his books have appeared with leading international academic publishers.

PhD1/2 we are looking for two PhD candidates with a background in computer engineering with interest in digital design and programming languages.

Tasks, Milestones, and Timetable The project is divided into several tasks. For each task, the time is given in person-months (PM). The Gantt chart shows the project schedule. For an assessment of the project's success, we plan the following milestones:

M1 (Month 6): The PhD students are selected and employed.

M2 (Month 15): Tools have been learned, and first artifacts have been developed.

M3 (Month 30): All development has been finished, and the different components can be used for exploration and evaluation of the results with the industrial use cases.

M4 (Month 48): The project has finished, and two PhD theses have been handed in.

Task	PM	Person	Description
Recruiting	1	MS and PS	Recruiting of the two PhDs.
UVM	3	KH	Develop a UVM verification example on a complex multicore device.
Prepare1	3	PhD1	Exploring related work and learning Chisel, SystemVerilog, and UVM.
Cover	6	PhD1	Adding constraint random testing and coverage to ChiselTest.
Generate	6	PhD1	Developing a hardware generator methodology by using functional programming in Scala.
Framework	9	PhD1	Development of an object-oriented and functional testing framework.
Prepare2	3	PhD2	Explore related work on property-based testing and ScalaCheck.
Spec	6	PhD2	Collaborate with hardware developers to manually inject both valid and faulty specifications.
Testing	6	PhD2	Develop property-based testing the specifications with ScalaCheck.
Bugbase	3	PhD2	Create a database from historical bug patterns from hardware development.
Implement	6	PhD2	Implement a program transformer to convert the bug patterns into additional, meaningful specification in code (à la FindBugs or ScalaFix).
Explore	2x4	PhD1/2	Detect and report bugs with the implemented tool and incorporate their feedback into a detailed empirical bug study.
Eval	2x4	PhD1/2	Evaluation of the results collected with the industry use cases.
Thesis	4	PhD1/2	Thesis writing und submission.
TeachSW	2	JM	Make Chisel and its design methods accessible for software designers.
Chapter	2	MS	Adding a chapter on verification to the Chisel book [25].

Table 1: Tasks for SDH

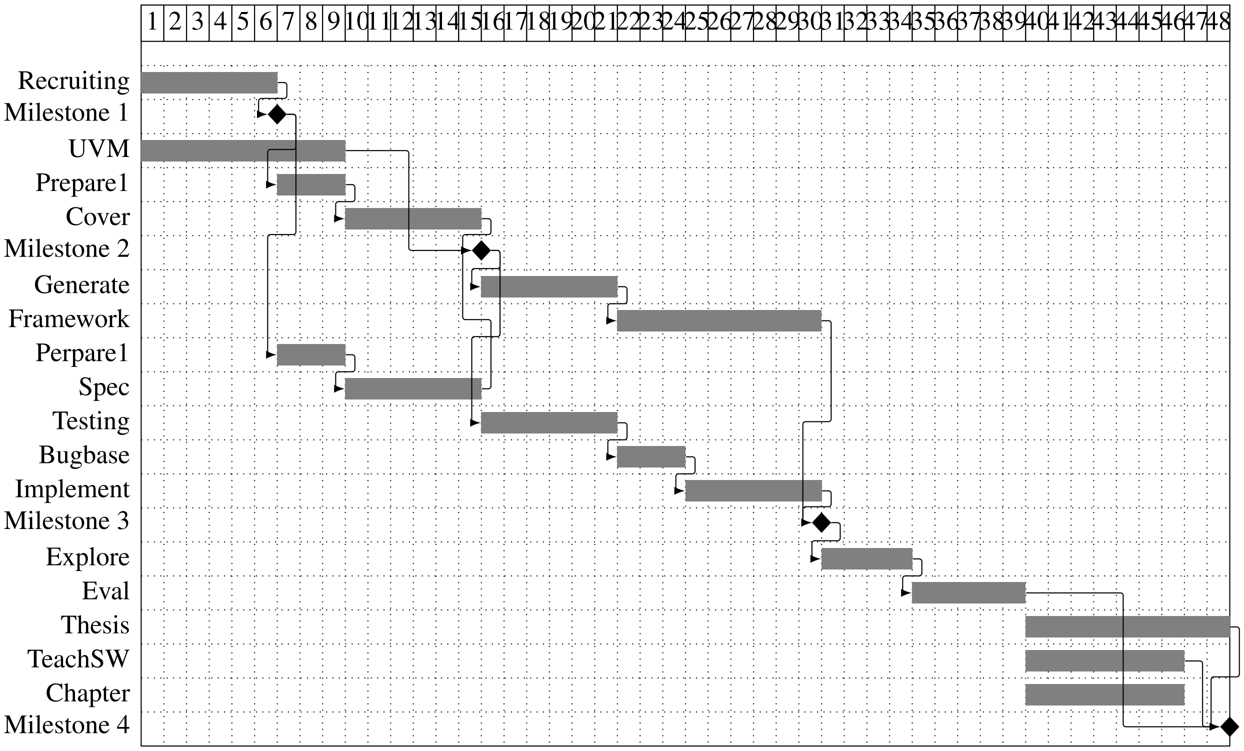

Figure 1: The Gantt chart of SDH

References

[1] FindBugs bug descriptions. http://findbugs.sourceforge.net/bugDescriptions.html. Accessed: 2020-09-12.

[2] OSS-Fuzz: Continuous fuzzing for open source software. https://github.com/google/oss-fuzz. Accessed: 2020-09-12.

[3] Scalafix: Refactoring and linting tool for scala. https://github.com/scalacenter/scalafix. Accessed: 2020-09-12.

[4] IEEE standard for SystemVerilog – unified hardware design, specification, and verification language, 2018.

[5] T. Arts, J. Hughes, J. Johansson, and U. T. Wiger. Testing telecoms software with quviq quickcheck. In M. Feeley and P. W. Trinder, editors, *Proceedings of the 2006 ACM SIGPLAN Workshop on Erlang, Portland, Oregon, USA, September 16, 2006*, pages 2–10. ACM, 2006.

[6] T. Arts, J. Hughes, U. Norell, and H. Svensson. Testing AUTOSAR software with quickcheck. In *Eighth IEEE International Conference on Software Testing, Verification and Validation, ICST 2015 Workshops, Graz, Austria, April 13-17, 2015*, pages 1–4. IEEE Computer Society, 2015.

[7] N. Ayewah, W. Pugh, J. D. Morgenthaler, J. Penix, and Y. Zhou. Evaluating static analysis defect warnings on production software. In M. Das and D. Grossman, editors, *Proceedings of the 7th ACM SIGPLAN-SIGSOFT Workshop on Program Analysis for Software Tools and Engineering, PASTE'07, San Diego, California, USA, June 13-14, 2007*, pages 1–8. ACM, 2007.

[8] J. Bachrach, H. Vo, B. Richards, Y. Lee, A. Waterman, R. Avizienis, J. Wawrzynek, and K. Asanovic. Chisel: constructing hardware in a scala embedded language. In *The 49th Annual Design Automation Conference (DAC 2012)*, pages 1216–1225, San Francisco, CA, USA, June 2012. ACM.

[9] I. D. Baxter. Branch coverage for arbitrary languages made easy. http://www.semdesigns.com/Company/Publications/TestCoverage.pdf, 2002.

[10] K. Beck, M. Beedle, A. van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R. C. Martin, S. Mellor, K. Schwaber, J. Sutherland, and D. Thomas. Manifesto for agile software development. https://agilemanifesto.org/, 2001.

[11] M. Böhme, V. Pham, and A. Roychoudhury. Coverage-based greybox fuzzing as markov chain. *IEEE Trans. Software Eng.*, 45(5):489–506, 2019.

[12] K. Claessen and J. Hughes. Quickcheck: a lightweight tool for random testing of haskell programs. In M. Odersky and P. Wadler, editors, *Proceedings of the Fifth ACM SIGPLAN International Conference on Functional Programming (ICFP '00), Montreal, Canada, September 18-21, 2000*, pages 268–279. ACM, 2000.

[13] W. J. Dally, Y. Turakhia, and S. Han. Domain-specific hardware accelerators. *Commun. ACM*, 63(7):48–57, June 2020.

[14] Doulos. Doulos | SystemVerilog DPI Tutorial. http://www.doulos.com/knowhow/systemverilog/systemverilog-tutorials/systemverilog-dpi-tutorial/.

[15] S. G. A. Duvall, Paul; Matyas. *Continuous integration : improving software quality and reducing risk.* Addison-Wesley, Upper Saddle River, N.J, 2007.

[16] H. Esmaeilzadeh, E. Blem, R. St. Amant, K. Sankaralingam, and D. Burger. Dark silicon and the end of multicore scaling. In *Proceedings of the 38th Annual International Symposium on Computer Architecture*, ISCA '11, pages 365–376, New York, NY, USA, 2011. Association for Computing Machinery.

[17] A. Fischer-Nielsen, Z. Fu, T. Su, and A. Wąsowski. The forgotten case of the dependency bugs: On the example of the robot operating system. In *The 42nd International Conference on Software Engineering: Software Engineering in Practice ((ICSE-SEIP)*, 2020, to appear.

[18] Z. Fu and Z. Su. Achieving high coverage for floating-point code via unconstrained programming. In A. Cohen and M. T. Vechev, editors, *Proceedings of the 38th ACM SIGPLAN Conference on Programming Language Design and Implementation, PLDI 2017, Barcelona, Spain, June 18-23, 2017*, pages 306–319. ACM, 2017.

[19] T. Gingold. Ghdl. https://github.com/ghdl/ghdl.

[20] J. L. Hennessy and D. A. Patterson. A new golden age for computer architecture. *Commun. ACM*, 62(2):48–60, Jan. 2019.

[21] J. M. Hughes and H. Bolinder. Testing a database for race conditions with quickcheck: none. In K. Rikitake and E. Stenman, editors, *Proceedings of the 10th ACM SIGPLAN workshop on Erlang, Tokyo, Japan, September 23, 2011*, pages 72–77. ACM, 2011.

[22] R. Lin. ChiselTest. https://github.com/ucb-bar/chisel-testers2.

[23] A. T. D. D. Miller. *Fuzzing for software security testing and quality assurance*. Artech House, Boston, MA, 2018.

[24] R. Nilsson. *ScalaCheck : the definitive guide*. Artima Press, Walnut Creek, California, 2014.

[25] M. Schoeberl. *Digital Design with Chisel*. Kindle Direct Publishing, 2019. available at `https://github.com/schoeberl/chisel-book`.

[26] C. Wolf. Yosys open synthesis suite. `http://www.clifford.at/yosys/`.

Participants

The budget figures are collected from the uploaded spreadsheet file

			Project participation				
			Total		Funded by DFF		
Type	Institution	Position, Name	Months	Amount excl. overheads (DKK)	Months	Amount excl. overheads (DKK)	Description (tasks and salary expenses, max. 300 characters)
Applicant	Technical University of Denmark	Martin Schoeberl					
PhD student	Technical University of Denmark	PhD student, N N	36	1.407.248	36	1.407.248	
PhD student	IT University of Copenhagen	PhD student, N N	36	1.373.796	36	1.373.796	
Scientific/academic staff	Technical University of Denmark	Student, N N	22	572.857	22	572.857	
			94	3.353.901	94	3.353.901	

Expenses

The budget figures are collected from the uploaded spreadsheet file

	DFF-financing	Co-financing	Other sources	Total (DKK)
Applicant				
Scientific/academic staff excluding postdocs and PhD students	572.857			572.857
Postdoc(s)				
PhD-student(s)	2.781.045			2.781.045
Total scientific/academic	3.353.902			3.353.902
Technical/administrative staff				
Equipment expenses	10.000			10.000
Operating expenses	840.000			840.000
Total - excl. overhead	4.203.902			4.203.902
Overheads	1.849.715	*n/a*	*n/a*	1.849.715
Total (DKK)	**6.053.617**			**6.053.617**

Expenses excl. payroll costs - details (DFF financing only)

Expense type	Institution	Amount (excl. overheads) DKK	Description, max. 300 characters
Operating expenses	Technical University of Denmark	180.000	Travel project participants
Operating expenses	Technical University of Denmark	240.000	Tuition fee
Equipment expenses	Technical University of Denmark	10.000	Minor equipment (FPGA board)
Operating expenses	IT University of Copenhagen	180.000	Travel project participants
Operating expenses	IT University of Copenhagen	240.000	Tuition fee
Total		**DKK 850.000**	

Curriculum Vitae

Assoc. Prof. DI Dr. Martin Schoeberl
Mariendalsvej 25, 1
2000 Frederiksberg
Denmark
`mailto:masca@dtu.dk`
`http://www.imm.dtu.dk/~masca/`

Education

December 2010	Habilitation at TU Vienna
April 2005	PhD Degree in Computer Engineering with distinction from TU Vienna, date: 11.4.2005
2000 – 2004	PhD Studies of Computer Engineering at the TU Vienna
June 2000	Conservatory Diploma in Jazz guitar at the Gustav Mahler Conservatory, Vienna
Summer 1999	Studies of Jazz guitar at the Berklee College of Music, Boston, USA
November 1994	Master's Degree in Computer Science from TU Vienna
1993 – 2000	Studies of Jazz guitar at the Prayner and Gustav Mahler Conservatory, Vienna
1986 – 1994	Studies in Computer Science at the TU Vienna
May 1986	School leaving examination, with distinction
1980 – 1986	Engineering School for Communications Engineering and Electronics in St. Pölten

Employment

Since 2010	Associate Professor at the Department of Applied Mathematics and Computer Science, Technical University of Denmark
2005 – 2009	Assistant Professor at the Institute of Computer Engineering, TU Vienna
1996	Civilian Service in Vienna
Since 1994	Self-employed with projects in automation and supervision for the Lower Austrian energy provider (EVN), Balfour Beatty, and the Austrian railway company (ÖBB)
1992 – 1994	Software engineer at Wirtschafts- und Sozialwissenschaftliches Rechenzentrum
1987 – 1991	Software engineer at COIN Computerentwicklungen GmbH
1986 – 1987	Software engineer at SYSGRAPH Computergraphik GmbH

Currently I am associate professor at the Technical University of Denmark (DTU), at DTU Compute with a tenured position. From 2005 to 2010 I was assistant professor at the Institute of Computer Engineering, Vienna University of Technology. I completed my PhD at the Vienna University of Technology in 2005 and received the Habilitation in Computer engineering in 2010.

My research focus is on time-predictable computer architectures, multi-core system-on-chip for real-time systems, and on Java for hard real-time systems. During my PhD, I developed the time-predictable Java processor JOP. This processor is currently being used in industrial projects and is the basis for further research on chip-multiprocessors for real-time systems. The Java processor is not only open-source, it is also being used at other universities as a basis for real-time Java research, and, in addition, is the subject of an active online community (i.e., a mailing list with 716 members). JOP is the only Java processor for which the execution time can be easily analyzed and a worst-case execution time (WCET) analysis tool is available.

The result of my PhD thesis, the real-time Java processor JOP, enabled my participation in the EU project JEOPARD (Java Environment for Parallel Realtime Development).[1] The JEOPARD project started in January 2008 and the overall project budget is EUR 3.2 million. I was responsible for the Austrian part of the proposal, representing the TU Vienna in the project, and I lead the "Architecture" work-package in JEOPARD.

My work on time-predictable architectures led to the **EC funded project T-CREST** (Time-predictable Multi-Core Architecture for Embedded Systems).[2] T-CREST started September 2011 and the overall project budget is EUR 3.8 million. **I led the funding proposal and was technical lead of T-CREST.**

Furthermore, my expertise in real-time Java and the implementation of the time-predictable JOP led to my membership of the Expert Group for the standard on Safety Critical Java (JSR 302).[3]

Industrial Experience

Since 1986, I have been working in the area of computer engineering and in the automation industry on distributed soft real-time systems. Since 1994, I have been self-employed and occupied on projects in embedded systems for automation and supervision. Four of the projects were based on the Java processor JOP:[4] (1) A distributed motor control system for Balfour Beatty; (2) An industrial lift controller (Ankara, Turkey); (3) A remote control system for the Lower Austrian energy provider EVN; and (4) A railway control support system for single track lines for the Austrian railway company ÖBB. All those systems are embedded systems where I designed the hardware with an FPGA, for three of the four, the external hardware (electronics), and wrote the embedded software. I consider all those tasks at the center of applied computer engineering. Those projects are all embedded real-time systems, systems that interact with the physical world, i.e., cyber physical systems.

Moreover, I have supported the Austrian SME DECOMSYS in the chip design for the real-time communication controller for the FlexRAY bus during a half year project. I designed the bus frontend with the synchronization unit and the interface to the PowerPC.

[1] http://www.jeopard.org/

[2] http://www.t-crest.org/

[3] http://jcp.org/en/jsr/detail?id=302

[4] Three of the projects are described in an invited paper for the IFAC 2008, available at http://www.jopdesign.com/doc/jop_app.pdf

Benefit for Academia

My extended experience in research and development for industry provides a good background for research in the area of embedded real-time systems and cyber-physical systems. I returned to academia due to personal interest and to solve some of the fundamental problems I had seen during my industrial work. Finally, my industrial experience and leading development projects provides profound knowledge to establish and lead an independent research group.

Summary of Scientific Work

- 27 journal articles
- 150 original, peer reviewed conference and workshop publications
- 4 books
- 1 patent
- 4176 citations, h-index of 36
- several invited talks
- Reviewer for several journals and PC member of several conferences
- Member of the Expert Group for the Safety Critical Java Specification

Research Visits

- Fall 2019, University of California, Berkeley
- Winter 2018/19, University of California, Berkeley
- Winter 2015/16, University of California, Berkeley
- Winter 2012/13, University of California, Berkeley
- Fall 2009, University of California, Berkeley
- April 2008, Aalborg University, Denmark
- August 2007, Aalborg University, Denmark
- February (+ spring part time) 2006, CBS, Copenhagen, Denmark

Funding

- 2016–2021: Time-predictable Control Systems (PREDICT), Danish Council for Independent Research — Technology and Production Sciences under contract 6111-00363; DKK 5,170,177.-
- 2013–2016: Hard Real-Time Embedded Multiprocessor Platform - RTEMP (Co-applicant, PI is Jens Sparsø), Danish Research Council for Technology and Production Sciences under contract 12-127600; DKK 4,977,862.-

- 2012–2016: ICT COST Action IC1202 Timing Analysis on Code-Level (TACLe), Co-applicant and work package 2 lead.
- 2011–2014: FP7 EC project T-CREST (Time-predictable Multi-Core Architecture for Embedded Systems) under grant agreement number 288008; EUR 3,807,000.- (total), EUR 703,000.- for DTU. I **led the funding proposal** and was **technical coordinator of T-CREST**. The proposal was **ranked at position 4 out of 59** submissions.
- 2011–2014: Certifiable Java for Embedded Systems (*Principal Investigator* and funding applicant), Danish Research Council for Technology and Production Sciences under contract 10-083159; DKK 5,058,000.-.
- 2008–2010: FP7 EC project JEOPARD (Java Environment for Parallel Realtime Development) under grant agreement number 216682; EUR 3,170,000.- (total), EUR 167,000.- for the TU Vienna. I wrote the proposal part for TU Vienna and I am lead the work package on architectures. JEOPARD deals with the issues for chip-multiprocessor solutions for real-time Java.
- 2004–2007: *Principal Investigator* and funding applicant (self employed) for the national SME funding project: Implementation of the CLDC standard for real-time systems on a Java processor, EUR 80,000.-

Invited Talks and Tutorials

- September 20, 2020, “Software-Defined Hardware: Digital Design with Chisel”, one day tutorial at ESWEEK, on-line
- September 18, 2020, “Digital Design in Chisel”, RISC-V Day Vietnam 2020
- September 4, 2020, “Software-Defined Hardware: Digital Design with Chisel”, one day tutorial at FPL, on-line
- January 29, 2020, “Is Chisel Ready for Class?”, Chisel Community Conference, Western Digital, Milpitas, CA
- October 13, 2019, “Hardware Design in the 21st Century with the Object Oriented and Functional Language Chisel”, one day tutorial at ESWEEK, New York, USA
- September 13, 2019, “Hardware Design in the 21st Century with the Object Oriented and Functional Language Chisel”, one day tutorial at FPL, Barcelona, Spain
- February 11, 2019, “A Time-predictable Multicore Processor Architecture for Real-Time Systems”, DREAM seminar, University of California, Berkeley, EECS
- November 13, 2018, “Fast Prototyping for Computer Architecture Research with Chisel”, Chisel Community Conference, University of California, Berkeley, CA
- August 30, 2018, “Hardware Design in the 21st Century with the Object Oriented and Functional Language Chisel”, one day tutorial at FPL, Dublin, Ireland

- June 11, 2018, “Time-predictable Computer Architecture with Time Division Multiplexing”, Optimizing Real-Time Systems Working Group Meeting, Paris, France
- November 3, 2016, “T-CREST: Time-predictable Multi-Core Architecture for Embedded Systems”, Dagstuhl seminar, Dagstuhl, Germany
- June 15-16, 2016, “Chisel Tutorial”, University of Augsburg, Germany.
- June 14, 2016, “T-CREST: Time-predictable Multi-Core Architecture for Embedded Systems”, University of Augsburg, Germany
- November 16, 2012, “T-CREST: Time-predictable Multi-Core Architecture for Embedded Systems”, DREAM seminar, University of California, Berkeley, EECS
- July 6, 2011, “A Time-predictable Microprocessor: the Patmos Approach”, 11th International Forum on Embedded MPSoC and Multicore (MPSoC), 2011, Beaune, France
- September 19–23, 2010, “The Java Optimized Processor: Java in a Field-Programmable Gate Array”, JavaOne, San Francisco, California, USA
- June 29, 2010, “Schedule Memory Access, not Threads”, 10th International Forum on Embedded MPSoC and Multicore (MPSoC), 2010, Gifu city, Gifu, Japan
- May 7, 2010, “Real time JAVA in FPGA based systems”, Prevas 25 year celebration, Copenhagen, Denmark
- December 10, 2009, “A Java Processor in an FPGA for Real-Time Systems”, Sun Labs, Menlo Park CA, USA
- March 5, 2009, “JOP, a Java Processor for Embedded Real-time Systems”, University of Lugano, Switzerland
- February 3, 2009, “JEOPARD, Multi-Core and Safety Critical Java”, The Open Group, Real-Time Embedded Systems Forum, San Diego, California
- September 24, 2008, JTRES 2008 Panel: Approaches to Multi-Core Processing
- April 2008, Aalborg University, Denmark
- February 2008, University of Augsburg, Germany
- September 2006, University College Vitus Bering, Denmark
- September 2006, Workshop on Java in Embedded Systems, CISS, ITU Copenhagen
- June 2006, Dept. of Informatics, CBS, Copenhagen
- March 2006, Workshop on Java in Embedded Systems, CISS, Aalborg
- August 2004, Dept. of Computer Science, Lund Institute of Technology, Sweden

Community Services

- General chair ARCS 2019 in Copenhagen
- EU projects track chair DATE 2019
- Program chair JTRES 2009, 2016
- Program chair WCET 2016
- Program Co-Chair ISORC 2015
- Workshop chair JTRES 2012 in Copenhagen
- Track Co-Chair ACM SAC 2011
- Editorial Board Member of Journal of Systems Architecture (JSA)
- Associate editor: EURASIP JES, IJERTCS
- Guest editor for the special issue on *Java Technologies for Real-Time Distributed and Embedded Systems* in Concurrency and Computation: Practice and Experience
- Workshop chair JTRES 2007 in Vienna
- Expert Group member of the Safety Critical Java Technology Specification (JSR 302)
- PC member JTRES 2006–2016
- PC member FPL 2007–2015
- PC member ACM SAC 2010–2021
- PC member ISORC 2010, 2012–2020
- PC member RTNS 2012, 2017
- PC member WCET 2012, 2015–2019
- PC member SIES 2016–2018
- PC member SEUS 2010, 2013–2016
- PC member Transact 2010
- PC member PPPJ 2013, 2014, ECRTS WiP 2013
- PC member SSV 2014
- PC member HiRES 2014–2016
- PC member Ada-Europe 2016
- PC member RTSS 2018

- PC member RTAS 2019
- PC member DESTION 2019, 2020
- Reviewer for embedded systems journals: RTS, JSA, EURASIP JES, TECS, TCAD, TII, TPDS, TCAS, SP&E, IET Software, TPDS, S P&E, IJERTCS, ESL, TAES,...

Teaching

- Verification of Digital Designs (2020)
- Computer Architecture and Engineering (since 2010)
- Advanced Computer Architecture (since 2011)
- Digital Electronics (since 2011)
- Computer Architecture VO (2006–2008)
- Computer Architecture Lab (2006–2008)
- The Java Virtual Machine in Hardware VL (2005–2009)
- Very Small Information Systems (2006 at CBS, Copenhagen)
- Distributed Systems (2006 at CBS, Copenhagen)
- Electrical Engineering Lab (2005–2006)
- Digital Signal Processor Lab (2005–2008)
- Bachelor Project and Seminar (since 2006)

Supervised Theses

Guiding PostDocs

- Torur Biskopsto Strom
- Wolfgang Puffitsch
- Florian Brandner

PhD Theses

- Eleftherios Kyriakakis, FORA Open Source Fog Node: Hardware support for virtualization, current
- Torur Biskopsto Strom, Real-Time Multi-Core Communication and Synchronization, 2019
- Luca Pezzarossa, Dynamic Partial Reconfiguration in FPGA based Multi-core Real-time Embedded Systems, 2018 (Co-supervision)

- Rasmus Bo Sørensen, Hardware/Software tradeoffs in Real-Time Multiprocessor Platforms, 2016 (Co-supervision)
- Sahar Abbaspourseyedi, Time-predictable VLIW Processor, 2015
- Evangelia Kasapaki, Asynchronous Network-on-Chip for Time-Predictable Multi-Core Embedded Systems, 2015 (Co-supervision)
- Juan Rios, Certifiable Java for Embedded Systems, 2014
- Alexander Jordan, Restricted Global Scheduling and Cache Optimization Techniques for VLIW Architectures, (Co-supervision with TU Vienna), 2014
- Wolfgang Puffitsch, Real-Time Garbage Collection for Multiprocessor Systems, 2012
- Christof Pitter, Time-Predictable Java Chip-Multiprocessor, 2009 (Co-supervision)

Master Theses

- Emad Jacob Maroun, Instruction Scheduler for the Dual-Issue Patmos Pipeline, 2019
- Roman Birca, SystemC Tracing for Observability and Profiling, 2018
- Daniel Sanz Ausin, Audio Processing on a Multicore Platform, 2017
- Philipp Degasperi, Method Cache for Patmos, 2014
- Marco Ziccardi, A Time-Composable Operating System for the Patmos Processor, 2014
- David VH Chong, Memory Access Arbitration for the Patmos Chip-Multicore Processor, 2014
- Edgar Lakis, FPGA implementation of time predictable memory controller for a chip-multiprocessor system, 2013
- Tórur Biskopstø Strøm, Safety-Critical Java 3D Desktop Printer, 2013
- Rasmus Bo Sørensen, Programming of the T-CREST real-time multi-processor platform, 2012
- Stefan Hepp, Worst-Case Execution Time Driven Method Inlining for Embedded Java Processors, 2011
- Thomas Bowley, Migrating a backbone network product platform for a wireless sensor system, from assembler/C to a higher level embedded Java paradigm, 2010
- Peter Hilber, Hardware Transactional Memory for a Real-Time Chip Multiprocessor, 2010
- Benedikt Huber, Worst-Case Execution Time Analysis for Real-Time Java, 2009
- Christian Stoif, Implementation and Performance of Synchronization Methods for Dual-Core Engine Control-Systems, 2008
- Wolfgang Puffitsch, picoJava-II in an FPGA, 2007

- Mikael Lundsgaard and Jens Kritian Rasmussen, JOPSPEECH, Embedded Java Speech Recognition SDK, 2007, CBS Copenhagen
- Kasper Hansen, Bluetooth API for JOP, 2007, CBS Copenhagen

Links

- Personal page at DTU: `http://www.imm.dtu.dk/~masca/`
- Google Scholar profile: `http://scholar.google.com/citations?user=wiRNmwUAAAAJ&hl=en&oi=ao`
- GitHub page: `https://github.com/schoeberl`
- PhD thesis: `http://www.jopdesign.com/thesis/thesis.pdf`
- JOP web site: `http://www.jopdesign.com/`

Publications Martin Schoeberl

I have published four books, 27 journal articles, one book chapter, one patent, and 150 papers in peer reviewed conferences and workshops; 74 of the publications as main author (38 of the 74 as single author). According to Google Scholar,[1] my papers are cited 4176 times and my h index is 36.

Five Most Important Publications

- Martin Schoeberl. A Java processor architecture for embedded real-time systems. *Journal of Systems Architecture*, 54/1–2:265–286, 2008.

  This article summarizes my PhD work on a time-predictable Java processor (called JOP). This article was the **most cited article at JSA** of articles published in the last five years for several years (it is now out of the 5 year window.) Its citation count is 248.

- Martin Schoeberl, Wolfgang Puffitsch, Rasmus Ulslev Pedersen, and Benedikt Huber. Worst-case execution time analysis for a Java processor. *Software: Practice and Experience*, 40/6:507–542, 2010.

  This article presents the worst-cases execution time (WCET) analysis tool for the Java processor JOP. The hardware design of JOP together with this tool is currently the only solution for embedded Java where the worst-case execution time can be predicted statically.

- Martin Schoeberl. Time-predictable computer architecture. *EURASIP Journal on Embedded Systems*, vol. 2009, Article ID 758480:17 pages, 2009.

  This article take the basic ideas from the JOP design on time-predictable computer architecture towards general purpose processors. This article was the main article for my habilitation thesis, where I **establish time-predictable computer architecture as a research direction**. Furthermore, the T-CREST project was based on the ideas presented in this article.

- Christof Pitter and Martin Schoeberl. A real-time Java chip-multiprocessor. *ACM Trans. Embed. Comput. Syst.*, 10(1):9:1–34, 2010.

  Together with my first PhD student Christof Pitter we built a time-predictable chip-multiprocessor based on time-predictable memory arbitration and several Java processors. To the best of my knowledge this was the first multicore processor that was WCET analyzable and supported by a WCET analysis tool.

- Martin Schoeberl, Sahar Abbaspour, Benny Akesson, Neil Audsley, Raffaele Capasso, Jamie Garside, Kees Goossens, Sven Goossens, Scott Hansen, Reinhold Heckmann, Stefan Hepp, Benedikt Huber, Alexander Jordan, Evangelia Kasapaki, Jens Knoop, Yonghui Li, Daniel Prokesch, Wolfgang Puffitsch, Peter Puschner, André Rocha, Cláudio Silva, Jens Sparsø, and Alessandro Tocchi. T-CREST: Time-predictable Multi-Core Architecture for Embedded Systems. *Journal of Systems Architecture* 61(9):449–471, 2015.

  This article summarizes the research and development work within the EC funded project T-CREST.

[1] Google profile at `http://scholar.google.com/citations?user=wiRNmwUAAAAJ&hl=en&oi=ao`

Books

1. Martin Schoeberl. *Digital Design with Chisel*. Number ISBN 9781689336031. Kindle Direct Publishing, August 2019.

2. Martin Schoeberl. *JOP Reference Handbook: Building Embedded Systems with a Java Processor*. Number ISBN 978-1438239699. CreateSpace, August 2009.

3. Martin Schoeberl. *JOP: A Java Optimized Processor for Embedded Real-Time Systems*. Number ISBN 978-3-8364-8086-4. VDM Verlag Dr. Müller, July 2008.

4. Doug Locke, B. Scott Andersen, Ben Brosgol, Mike Fulton, Thomas Henties, James J. Hunt, Johan Olmütz Nielsen, Kelvin Nilsen, Martin Schoeberl, Joyce Tokar, Jan Vitek, and Andy Wellings. Safety-critical Java technology specification, public draft, 2011.

Book Chapter

1. Martin Schoeberl. Hardware support for embedded Java. In M. Teresa Higuera-Toledano and Andy J. Wellings, editors, *Distributed, Embedded and Real-time Java Systems*, pages 159–176. Springer US, 2012.

Patent

1. Martin Schoeberl. Instruction Cache für Echtzeitsysteme, April 2006. Austrian patent AT 500.858.

Theses

1. Martin Schoeberl. Time-predictable computer architecture. Habilitation thesis, Institut for Computer Engineering, Vienna University of Technology, September 2009.

2. Martin Schoeberl. *JOP: A Java Optimized Processor for Embedded Real-Time Systems*. PhD thesis, Vienna University of Technology, 2005.

Journal Articles

2020

1. Eleftherios Kyriakakis, Maja Lund, Luca Pezzarossa, Jens Sparsø, and **Martin Schoeberl** . A time-predictable open-source TTEthernet end-system. *Journal of Systems Architecture* 108:101744, 2020.

2019

2. Morten B. Petersen, Anthon V. Riber, Simon T. Andersen, and **Martin Schoeberl** . Time-predictable Distributed Shared On-Chip Memory. *Microprocessors and Microsystems* 2019.

3. Tórur Biskopstø Strøm, Jens Sparsø, and **Martin Schoeberl** . Hardlock: Real-time multicore locking. *Journal of Systems Architecture* 97:467–476, 2019.

4. Emad Jacob Maroun, Henrik Enggaard Hansen, Andreas Toftegaard Kristensen, and **Martin Schoeberl** . Time-predictable synchronization support with a shared scratchpad memory. *Microprocessors and Microsystems* 64:34–42, 2019.

2018

5. **Martin Schoeberl** , Wolfgang Puffitsch, Stefan Hepp, Benedikt Huber, and Daniel Prokesch. Patmos: A Time-predictable Microprocessor. *Real-Time Systems* 54(2):389–423, 2018.

6. Luca Pezzarossa, Andreas Toftegaard Kristensen, **Martin Schoeberl** , and Jens Sparsø. Using Dynamic Partial Reconfiguration of FPGAs in Real-Time Systems. *Microprocessors and Microsystems* 61:198–206, 2018.

7. **Martin Schoeberl** , Luca Pezzarossa, and Jens Sparsø. A Multicore Processor for Time-Critical Applications. *IEEE Design Test* 35:38–47, 2018.

8. Rasmus Ulslev Pedersen and **Martin Schoeberl** . Direct garbage collection: two-fold speedup for managed language embedded systems. *International Journal of Embedded Systems* 10(5):394–405, 2018.

2017

9. **Martin Schoeberl** , Andreas Engelbredt Dalsgaard, Rene Rydhof Hansen, Stephan E. Korsholm, Anders P. Ravn, Juan Ricardo Rios Rivas, Torur Biskopstø Strøm, Hans Søndergaard, Andy Wellings, and Shuai Zhao. Safety-critical Java for embedded systems. *Concurrency and Computation: Practice and Experience* 29(22), 2017.

10. Tórur Biskopstø Strøm, Wolfgang Puffitsch, and **Martin Schoeberl** . Hardware Locks for a Real-Time Java Chip-Multiprocessor. *Concurrency and Computation: Practice and Experience* 29(6):e3950–n/a, 2017.

11. Rasmus Bo Sørensen, Luca Pezzarossa, **Martin Schoeberl** , and Jens Sparsø. A resource-efficient network interface supporting low latency reconfiguration of virtual circuits in time-division multiplexing networks-on-chip. *Journal of Systems Architecture* 74(Supplement C):1–13, 2017.

2016

12. Evangelia Kasapaki, **Martin Schoeberl** , Rasmus Bo Sørensen, Christian T. Müller, Kees Goossens, and Jens Sparsø. Argo: A Real-Time Network-on-Chip Architecture with an Efficient GALS Implementation. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 24:479–492, 2016.

2015

13. **Martin Schoeberl** , Sahar Abbaspour, Benny Akesson, Neil Audsley, Raffaele Capasso, Jamie Garside, Kees Goossens, Sven Goossens, Scott Hansen, Reinhold Heckmann, Stefan Hepp, Benedikt Huber, Alexander Jordan, Evangelia Kasapaki, Jens Knoop, Yonghui Li, Daniel Prokesch, Wolfgang Puffitsch, Peter Puschner, André Rocha, Cláudio Silva, Jens Sparsø, and Alessandro Tocchi. T-CREST: Time-predictable Multi-Core Architecture for Embedded Systems. *Journal of Systems Architecture* 61(9):449–471, 2015.

2014

2013

14. **Martin Schoeberl** , Benedikt Huber, and Wolfgang Puffitsch. Data cache organization for accurate timing analysis. *Real-Time Systems* 49(1):1–28, 2013.

15. Flavius Gruian and **Martin Schoeberl** . Hardware Support for CSP on a Java Chip-Multiprocessor. *Microprocessors and Microsystems* 37(4–5):472–481, 2013.

2012

16. Trevor Harmon, **Martin Schoeberl** , Raimund Kirner, Raymond Klefstad, K.H. (Kane) Kim, and Michael R. Lowry. Fast, Interactive Worst-Case Execution Time Analysis with Back-Annotation. *IEEE Transactions on Industrial Informatics* 8:366–377, 2012.

17. Fadi Meawad, Karthik Iyer, **Martin Schoeberl** , and Jan Vitek. Micro-transactions for concurrent data structures. *Concurrency and Computation: Practice and Experience* 2012.

18. Benedikt Huber, Wolfgang Puffitsch, and **Martin Schoeberl** . Worst-case execution time analysis driven object cache design. *Concurrency and Computation: Practice and Experience* 24(8):753–771, 2012.

19. Anders P. Ravn and **Martin Schoeberl** . Safety-Critical Java with Cyclic Executives on Chip-Multiprocessors. *Concurrency and Computation: Practice and Experience* 24:772–788, 2012.

2011

20. **Martin Schoeberl** , Stephan Korsholm, Tomas Kalibera, and Anders P. Ravn. A Hardware Abstraction Layer in Java. *ACM Trans. Embed. Comput. Syst.* 10(4):42:1–42:40, 2011.

2010

21. Christof Pitter and **Martin Schoeberl** . A Real-Time Java Chip-Multiprocessor. *ACM Trans. Embed. Comput. Syst.* 10(1):9:1–34, 2010.

22. **Martin Schoeberl** and Wolfgang Puffitsch. Nonblocking real-time garbage collection. *ACM Trans. Embed. Comput. Syst.* 10(1):6:1–28, 2010.

23. **Martin Schoeberl** . Scheduling of Hard Real-Time Garbage Collection. *Real-Time Systems* 45(3):176–213, 2010.

24. **Martin Schoeberl** , Wolfgang Puffitsch, Rasmus Ulslev Pedersen, and Benedikt Huber. Worst-case execution time analysis for a Java processor. *Software: Practice and Experience* 40/6:507–542, 2010.

2009

25. **Martin Schoeberl** . Time-predictable Computer Architecture. *EURASIP Journal on Embedded Systems* vol. 2009, Article ID 758480:17 pages, 2009.

26. Walter Binder, **Martin Schoeberl** , Philippe Moret, and Alex Villazon. Cross-profiling for Java processors. *Software: Practice and Experience* 39/18:1439–1465, 2009.

2008

27. **Martin Schoeberl** . A Java Processor Architecture for Embedded Real-Time Systems. *Journal of Systems Architecture* 54/1–2:265–286, 2008.

Reviewed Conference and Workshop Papers

2020

1. Mathieu Jan, Mihail Asavoae, **Martin Schoeberl** , and Edward A. Lee. Formal Semantics of Predictable Pipelines: a Comparative Study. *TODO: ASP-DAC* January, 2020.

2. Emad Jacob Maroun, **Martin Schoeberl** , and Peter Puschner. Towards Dual-Issue Single-Path Code. *2020 IEEE 23rd International Symposium on Real-Time Distributed Computing (ISORC)* 176–183, 2020.

2019

3. Eleftherios Kyriakakis, Jens Sparsø, and **Martin Schoeberl** . InterNoC: Unified Deterministic Communication For Distributed NoC-based Many-Core. *13th Junior Researcher Workshop on Real-Time Computing* November, 2019.

4. Mihail Asavoae, Imane Haur, Mathieu Jan, Belgacem Ben Hedia, and **Martin Schoeberl** . Towards Formal co-Validation of Hardware and Software Timing Models of CPS. *Model-Based Design of Cyber Physical Systems (CyPhy'19)* October, 2019.

5. **Martin Schoeberl** . Multicore Models of Communication for Cyber-Physical Systems. *Model-Based Design of Cyber Physical Systems (CyPhy'19)* October, 2019.

6. **Martin Schoeberl** , Luca Pezzarossa, and Jens Sparsø. S4NOC: a Minimalistic Network-on-Chip for Real-Time Multicores. *Proceedings of the 12th International Workshop on Network on Chip Architectures* 8:1–8:6, October, 2019.

7. Marten Lohstroh, **Martin Schoeberl** , Mathieu Jan, Edward Wang, and Edward A. Lee. Programs with Ironclad Timing Guarantees: Work-in-progress. *Proceedings of the International Conference on Embedded Software Companion* 1:1–1:2, New York, New York, October, 2019.

8. Marten Lohstroh, **Martin Schoeberl** , Andrés Goens, Armin Wasicek, Christopher Gill, Marjan Sirjani, and Edward A. Lee. Actors Revisited for Time-Critical Systems. *Proceedings of the 56th Annual Design Automation Conference 2019* 152:1–152:4, Las Vegas, NV, USA, June, 2019.

9. **Martin Schoeberl** , Luca Pezzarossa, and Jens Sparsø. A Minimal Network Interface for a Simple Network-on-Chip. *Architecture of Computing Systems - ARCS 2019* 295–307, May, 2019.

10. Oktay Baris, Shibarchi Majumder, Tórur Biskopstø Strøm, Anders la Cour-Harbo, Jens Sparsø, Thomas Bak, and **Martin Schoeberl** . Demonstration of a Time-predictable Flight Controller on a Multicore Processor. *Proceedings of the 22nd IEEE International Symposium on Real-time Computing (ISORC)* 95–96, May, 2019.

11. **Martin Schoeberl** and Morten Borup Petersen. Leros: The return of the accumulator machine. *Architecture of Computing Systems - ARCS 2019 - 32nd International Conference, Proceedings* 115–127, May, 2019.

12. Maja Lund, Luca Pezzarossa, Jens Sparsø, and **Martin Schoeberl** . A Time-predictable TTEthernet Node. *2019 IEEE 22nd International Symposium on Real-Time Computing (ISORC)* 229–233, May, 2019.

13. **Martin Schoeberl** , Benjamin Rouxel, and Isabelle Puaut. A Time-predictable Branch Predictor. *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing* 607–616, Limassol, Cyprus, April, 2019.

14. **Martin Schoeberl** , Tórur Biskopstø Strøm, Oktay Baris, and Jens Sparsø. Scratchpad Memories with Ownership. *2019 Design, Automation and Test in Europe Conference Exhibition (DATE)* 1216–1221, March, 2019.

15. Christos Gkiokas and **Martin Schoeberl** . A Fault-Tolerant Time-Predictable Processor. *2019 IEEE Nordic Circuits and Systems Conference (NORCAS): NORCHIP and International Symposium of System-on-Chip (SoC)* 1–6, 2019.

16. Eleftherios Kyriakakis, Jens Sparsø, and **Martin Schoeberl** . Implementing time-triggered communication over a standard ethernet switch. *Proceedings of the Fog-IoT Workshop 2019* 21–25, 2019.

2018

17. Morten B. Petersen, Anthon V. Riber, Simon T. Andersen, and **Martin Schoeberl** . Time-Predictable Distributed Shared Memory for Multi-Core Processors. *2018 IEEE Nordic Circuits and Systems Conference (NORCAS): NORCHIP and International Symposium of System-on-Chip (SoC)* 1-7, October, 2018.

18. Hammond Pearce, Partha Roop, Morteza Biglari-Abhari, and **Martin Schoeberl** . Faster Function Blocks for Precision Timed Industrial Automation. *2018 IEEE 21st International Symposium on Real-Time Distributed Computing (ISORC)* 67-74, May, 2018.

19. **Martin Schoeberl** and Rasmus Ulslev Pedersen. tpIP: A Time-Predictable TCP/IP Stack for Cyber-Physical Systems. *2018 IEEE 21st International Symposium on Real-Time Distributed Computing (ISORC)* 75–82, May, 2018.

20. Tórur Biskopstø Strøm and **Martin Schoeberl** . Hardlock: A Concurrent Real-Time Multicore Locking Unit. *2018 IEEE 21st International Symposium on Real-Time Distributed Computing (ISORC)* 9–16, May, 2018.

21. **Martin Schoeberl** . Design of a Time-predictable Multicore Processor: The T-CREST Project. *2018 Design, Automation Test in Europe Conference Exhibition (DATE)* 909–912, March, 2018.

22. **Martin Schoeberl** . One-Way Shared Memory. *2018 Design, Automation and Test in Europe Conference Exhibition (DATE)* 269–272, March, 2018.

23. Eleftherios Kyriakakis, Jens Sparsø, and **Martin Schoeberl** . Hardware Assisted Clock Synchronization with the IEEE 1588-2008 Precision Time Protocol. *Proceedings of the 26th International Conference on Real-Time Networks and Systems* 51–60, 2018.

24. **Martin Schoeberl** . Lipsi: Probably the Smallest Processor in the World. *Architecture of Computing Systems – ARCS 2018* 18–30, 2018.

2017

25. Henrik Enggaard Hansen, Emad Jacob Maroun, Andreas Toftegaard Kristensen, Jimmi Marquart, and **Martin Schoeberl** . A Shared Scratchpad Memory with Synchronization Support. *2017 IEEE Nordic Circuits and Systems Conference (NORCAS): NORCHIP and International Symposium of System-on-Chip (SoC)* 1-6, October, 2017.

26. **Martin Schoeberl** and Jens Sparsø. Timing Organization of a Real-Time Multicore Processor. *2017 New Generation of CAS (NGCAS)* 89–92, September, 2017.

27. Bekim Cilku, Wolfgang Puffitsch, Daniel Prokesch, **Martin Schoeberl** , and Peter Puschner. Improving Performance of Single-path Code Through a Time-predictable Memory Hierarchy. *Proceedings of the 20th IEEE International Symposium on Real-Time Distributed Computing (ISORC 2017)* 76–83, May, 2017.

28. Luca Pezzarossa, **Martin Schoeberl** , and Jens Sparsø. A Controller for Dynamic Partial Reconfiguration in FPGA-Based Real-Time Systems. *2017 IEEE 20th International Symposium on Real-Time Distributed Computing (ISORC)* 92–100, May, 2017.

29. Daniel Sanz Ausin, Luca Pezzarossa, and **Martin Schoeberl** . Real-Time Audio Processing on the T-CREST Multicore Platform. *2017 IEEE 11th International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSoC)* 2017.

30. Luca Pezzarossa, Andreas Toftegaard Kristensen, **Martin Schoeberl** , and Jens Sparsø. Can Real-Time Systems Benefit from Dynamic Partial Reconfiguration?. *2017 IEEE Nordic Circuits and Systems Conference (NORCAS): NORCHIP and International Symposium of System-on-Chip (SoC)* 2017.

31. **Martin Schoeberl** , Bekim Cilku, Daniel Prokesch, and Peter Puschner. Best Practice for Caching of Single-Path Code. *17th International Workshop on Worst-Case Execution Time Analysis (WCET 2017)* 1–12, 2017.

2016

32. Alexander Jordan, Sahar Abbaspour, and **Martin Schoeberl** . A Software Managed Stack Cache for Real-Time Systems. *Proceedings of the 24th International Conference on Real-Time Networks and Systems (RTNS 2016)* 319–326, October, 2016.

33. Rasmus Bo Sørensen, **Martin Schoeberl** , and Jens Sparsø. State-based Communication on Time-predictable Multicore Processors. *Proceedings of the 24th International Conference on Real-Time Networks and Systems (RTNS 2016)* 225–234, October, 2016.

34. Florian Kluge, **Martin Schoeberl** , and Theo Ungerer. Support for the Logical Execution Time Model on a Time-predictable Multicore Processor. *14th International Workshop on Real-Time Networks* July, 2016.

35. Luca Pezzarossa, **Martin Schoeberl** , and Jens Sparsø. Reconfiguration in FPGA-based multi-core platforms for hard real-time applications. *11th International Symposium on Reconfigurable Communication-centric Systems-on-Chip (ReCoSoC 2016)* 1–8, June, 2016.

36. **Martin Schoeberl** and Carsten Nielsen. A Stack Cache for Real-Time Systems. *Proceedings of the 19th IEEE Symposium on Real-time Distributed Computing (ISORC 2016)* 150–157, May, 2016.

37. Wolfgang Puffitsch and **Martin Schoeberl** . Time-Predictable Virtual Memory. *Proceedings of the 19th IEEE Symposium on Real-time Distributed Computing (ISORC 2016)* 158–165, May, 2016.

38. **Martin Schoeberl** . Lessons learned from the EU project T-CREST. *Design, Automation Test in Europe Conference Exhibition (DATE 2016)* 870–875, March, 2016.

39. Heiko Falk, Sebastian Altmeyer, Peter Hellinckx, Björn Lisper, Wolfgang Puffitsch, Christine Rochange, **Martin Schoeberl** , Rasmus Bo Sørensen, Peter Wägemann, and Simon Wegener. TACLeBench: A Benchmark Collection to Support Worst-Case Execution Time Research. *16th International Workshop on Worst-Case Execution Time Analysis (WCET 2016)* 2:1–2:10, 2016.

2015

40. Wolfgang Puffitsch, Rasmus Bo Sørensen, and **Martin Schoeberl** . Time-Division Multiplexing vs Network Calculus: A Comparison. *Proceedings of the 23th International Conference on Real-Time and Network Systems (RTNS 2015)* November, 2015.

41. Luca Pezzarossa, Rasmus Bo Sørensen, **Martin Schoeberl** , and Jens Sparsø. Interfacing Hardware Accelerators to a Time-Division Multiplexing Network-on-Chip. *Proc. of the 1st Nordic Circuits and Systems Conference (NORCAS 2015)* October, 2015.

42. Stephan E. Korsholm, **Martin Schoeberl** , and Wolfgang Puffitsch. Safety-Critical Java on a Time-Predictable Processor. *Proceedings of the 13th International Workshop on Java Technologies for Real-Time and Embedded Systems (JTRES 2015)* October, 2015.

43. Tórur Biskopstø Strøm and **Martin Schoeberl** . Multiprocessor Priority Ceiling Emulation for Safety-Critical Java. *Proceedings of the 13th International Workshop on Java Technologies for Real-Time and Embedded Systems (JTRES 2015)* October, 2015.

44. **Martin Schoeberl** . Scala for Real-Time Systems?. *Proceedings of the 13th International Workshop on Java Technologies for Real-Time and Embedded Systems (JTRES 2015)* October, 2015.

45. Carsten Nielsen and **Martin Schoeberl** . Stack Caching Using Split Data Caches. *Proceedings of the 11th Workshop on Software Technologies for Embedded and Ubiquitous Systems (SEUS 2015)* 36–43, April, 2015.

46. Tórur Biskopstø Strøm and **Martin Schoeberl** . Hardware Locks with Priority Ceiling Emulation for a Java Chip-Multiprocessor. *Proceedings of the 17th IEEE Symposium on Real-time Distributed Computing (ISORC 2015)* 268–271, April, 2015.

47. Marco Ziccardi, **Martin Schoeberl** , and Tullio Vardanega. A Time-Composable Operating System for the Patmos Processor. *The 30th ACM/SIGAPP Symposium On Applied Computing, Embedded Systems Track* April, 2015.

48. Rasmus Bo Sørensen, Wolfgang Puffitsch, **Martin Schoeberl** , and Jens Sparsø. Message Passing on a Time-predictable Multicore Processor. *Proceedings of the 17th IEEE Symposium on Real-time Distributed Computing (ISORC 2015)* 51–59, April, 2015.

49. **Martin Schoeberl** , Rasmus Bo Sørensen, and Jens Sparsø. Models of Communication for Multicore Processors. *Proceedings of the 11th Workshop on Software Technologies for Embedded and Ubiquitous Systems (SEUS 2015)* 44–51, April, 2015.

50. Luca Pezzarossa, **Martin Schoeberl** , and Jens Sparsø. Towards Utilizing Reconfigurable Shared Resources in Multi-Core Hard Real-Time Systems. *9th Junior Researcher Workshop on Real-Time Computing JRWRTC 2015* 21–24, 2015.

2014

51. **Martin Schoeberl** , Andreas Engelbredt Dalsgaard, René Rydhof Hansen, Stephan E. Korsholm, Anders P. Ravn, Juan Ricardo Rios Rivas, Tórur Biskopstø Strøm, and Hans Søndergaard. Certifiable Java for Embedded Systems. *Proceedings of the 12th International Workshop on Java Technologies for Real-Time and Embedded Systems (JTRES 2014)* 10–19, October, 2014.

52. Benedikt Huber, Stefan Hepp, and **Martin Schoeberl** . Scope-based Method Cache Analysis. *Proceedings of the 14th International Workshop on Worst-Case Execution Time Analysis (WCET 2014)* 73–82, July, 2014.

53. **Martin Schoeberl** , David VH Chong, Wolfgang Puffitsch, and Jens Sparsø. A Time-predictable Memory Network-on-Chip. *Proceedings of the 14th International Workshop on Worst-Case Execution Time Analysis (WCET 2014)* 53–62, July, 2014.

54. Jack Whitham and **Martin Schoeberl** . WCET-Based Comparison of an Instruction Scratchpad and a Method Cache. *Proceedings of the 10th Workshop on Software Technologies for Embedded and Ubiquitous Systems (SEUS 2014)* June, 2014.

55. Juan Ricardo Rios and **Martin Schoeberl** . An Evaluation of Safety-Critical Java on a Java Processor. *Proceedings of the 10th Workshop on Software Technologies for Embedded and Ubiquitous Systems (SEUS 2014)* June, 2014.

56. Philipp Degasperi, Stefan Hepp, Wolfgang Puffitsch, and **Martin Schoeberl** . A Method Cache for Patmos. *Proceedings of the 17th IEEE Symposium on Object/Component/Service-oriented Real-time Distributed Computing (ISORC 2014)* 100–108, June, 2014.

57. **Martin Schoeberl** , Cláudio Silva, and André Rocha. T-CREST: A Time-predictable Multi-Core Platform for Aerospace Applications. *Proceedings of Data Systems In Aerospace (DASIA 2014)* June, 2014.

58. Juan Ricardo Rios and **Martin Schoeberl** . Reusable Libraries for Safety-Critical Java. *Proceedings of the 17th IEEE Symposium on Object/Component/Service-oriented Real-time Distributed Computing (ISORC 2014)* 188–197, June, 2014.

2013

59. Tórur Biskopstø Strøm, Wolfgang Puffitsch, and **Martin Schoeberl** . Chip-Multiprocessor Hardware Locks for Safety-Critical Java. *Proceedings of the 11th International Workshop on Java Technologies for Real-Time and Embedded Systems (JTRES 2013)* 38–46, October, 2013.

60. Alexander Jordan, Florian Brandner, and **Martin Schoeberl** . Static Analysis of Worst-case Stack Cache Behavior. *Proceedings of the 21st International Conference on Real-Time Networks and Systems (RTNS 2013)* 55–64, Sophia Antipolis, France, 2013.

61. Edgar Lakis and **Martin Schoeberl** . An SDRAM Controller for Real-Time Systems. *Proceedings of the 9th Workshop on Software Technologies for Embedded and Ubiquitous Systems* 2013.

62. Jens Sparsø, Evangelia Kasapaki, and **Martin Schoeberl** . An Area-efficient Network Interface for a TDM-based Network-on-Chip. *Proceedings of the Conference on Design, Automation and Test in Europe* 1044–1047, Grenoble, France, 2013.

63. Sahar Abbaspour, Florian Brandner, and **Martin Schoeberl** . A Time-predictable Stack Cache. *Proceedings of the 9th Workshop on Software Technologies for Embedded and Ubiquitous Systems* 2013.

2012

64. Rasmus Bo Sørensen, **Martin Schoeberl** , and Jens Sparsø. A Light-Weight Statically Scheduled Network-on-Chip. *Proceedings of the 29th Norchip Conference* November, 2012.

65. Florian Brandner and **Martin Schoeberl** . Static Routing in Symmetric Real-Time Network-on-Chips. *Proceedings of the 20th International Conference on Real-Time and Network Systems (RTNS 2012)* 61–70, November, 2012.

66. Wolfgang Puffitsch and **Martin Schoeberl** . On the Scalability of Time-predictable Chip-Multiprocessing. *Proceedings of the 10th International Workshop on Java Technologies for Real-Time and Embedded Systems (JTRES 2012)* 98–104, October, 2012.

67. Andreas E. Dalsgaard, René Rydhof Hansen, and **Martin Schoeberl** . Private Memory Allocation Analysis for Safety-Critical Java. *Proceedings of the 10th International Workshop on Java Technologies for Real-Time and Embedded Systems (JTRES 2012)* 9–17, October, 2012.

68. Juan Ricardo Rios, Kelvin Nilsen, and **Martin Schoeberl** . Patterns for Safety-Critical Java Memory Usage. *Proceedings of the 10th International Workshop on Java Technologies for Real-Time and Embedded Systems (JTRES 2012)* 1–8, October, 2012.

69. Tórur Biskopstø Strøm and **Martin Schoeberl** . A Desktop 3D Printer in Safety-Critical Java. *Proceedings of the 10th International Workshop on Java Technologies for Real-Time and Embedded Systems (JTRES 2012)* 72–79, October, 2012.

70. **Martin Schoeberl** and Juan Ricardo Rios. Safety-Critical Java on a Java Processor. *Proceedings of the 10th International Workshop on Java Technologies for Real-Time and Embedded Systems (JTRES 2012)* 54–61, October, 2012.

71. **Martin Schoeberl** . Is Time Predictability Quantifiable?. *International Conference on Embedded Computer Systems (SAMOS 2012)* July, 2012.

72. **Martin Schoeberl** , Florian Brandner, Jens Sparsø, and Evangelia Kasapaki. A Statically Scheduled Time-Division-Multiplexed Network-on-Chip for Real-Time Systems. *Proceedings of the 6th International Symposium on Networks-on-Chip (NOCS)* 152–160, May, 2012.

73. Juan Ricardo Rios and **Martin Schoeberl** . Hardware Support for Safety-Critical Java Scope Checks. *Proceedings of the 15th IEEE International Symposium on Object/component/service-oriented Real-time distributed Computing (ISORC 2012)* 31–38, April, 2012.

74. Stefan Hepp and **Martin Schoeberl** . Worst-Case Execution Time Based Optimization of Real-Time Java Programs. *Proceedings of the 15th IEEE International Symposium on Object/component/service-oriented Real-time distributed Computing (ISORC 2012)* 64–70, April, 2012.

2011

75. **Martin Schoeberl** . Leros: A Tiny Microcontroller for FPGAs. *Proceedings of the 21st International Conference on Field Programmable Logic and Applications (FPL 2011)* 10–14, September, 2011.

76. James Caska and **Martin Schoeberl** . Java Dust: How Small Can Embedded Java Be?. *Proceedings of the 9th International Workshop on Java Technologies for Real-Time and Embedded Systems (JTRES 2011)* 125–129, September, 2011.

77. Fadi Meawad, Karthik Iyer, **Martin Schoeberl** , and Jan Vitek. Real-Time Wait-free Queues using Micro-Transactions. *Proceedings of the 9th International Workshop on Java Technologies for Real-Time and Embedded Systems (JTRES 2011)* 1–10, September, 2011.

78. Andy Wellings and **Martin Schoeberl** . User-Defined Clocks in the Real-Time Specification for Java. *Proceedings of the 9th International Workshop on Java Technologies for Real-Time and Embedded Systems (JTRES 2011)* 74–81, September, 2011.

79. **Martin Schoeberl** . Memory Management for Safety-Critical Java. *Proceedings of the 9th International Workshop on Java Technologies for Real-Time and Embedded Systems (JTRES 2011)* 47–53, September, 2011.

80. **Martin Schoeberl** . ejIP: A TCP/IP Stack for Embedded Java. *Proceedings of the 9th International Conference on the Principles and Practice of Programming in Java (PPPJ 2011)* August, 2011.

81. Aibek Sarimbekov, Andreas Sewe, Walter Binder, Philippe Moret, **Martin Schoeberl** , and Mira Mezini. Portable and Accurate Collection of Calling-Context-Sensitive Bytecode Metrics for the Java Virtual Machine. *Proceedings of the 9th International Conference on the Principles and Practice of Programming in Java (PPPJ 2011)* August, 2011.

82. Reinhard von Hanxleden, Niklas Holsti, Björn Lisper, Erhard Ploedereder, Reinhard Wilhelm, Armelle Bonenfant, Hugues Casse, Sven Bünte, Wolfgang Fellger, Sebastian Gepperth, Jan Gustafsson, Benedikt

Huber, Nazrul Mohammad Islam, Daniel Kästner, Raimund Kirner, Laura Kovacs, Felix Krause, Marianne de Michiel, Mads Christian Olesen, Adrian Prantl, Wolfgang Puffitsch, Christine Rochange, **Martin Schoeberl** , Simon Wegener, Michael Zolda, and Jakob Zwirchmayr. WCET Tool Challenge 2011: Report. *Proceedings of the 11th International Workshop on Worst-Case Execution Time (WCET) Analysis* July, 2011.

83. Christian Stoif, **Martin Schoeberl** , Benito Liccardi, and Jan Haase. Hardware Synchronization for Embedded Multi-Core Processors. *Proceedings of the 2011 IEEE International Symposium on Circuits and Systems (ISCAS 2011)* May, 2011.

84. **Martin Schoeberl** , Walter Binder, and Alex Villazon. Design Space Exploration of Object Caches with Cross-Profiling. *Proceedings of the 14th IEEE International Symposium on Object/component/service-oriented Real-time distributed Computing (ISORC 2011)* 213–221, March, 2011.

85. **Martin Schoeberl** , Pascal Schleuniger, Wolfgang Puffitsch, Florian Brandner, Christian W. Probst, Sven Karlsson, and Tommy Thorn. Towards a Time-predictable Dual-Issue Microprocessor: The Patmos Approach. *First Workshop on Bringing Theory to Practice: Predictability and Performance in Embedded Systems (PPES 2011)* 11–20, March, 2011.

86. **Martin Schoeberl** . A Time-predictable Object Cache. *Proceedings of the 14th IEEE International Symposium on Object/component/service-oriented Real-time distributed Computing (ISORC 2011)* 99–105, March, 2011.

2010

87. Flavius Gruian and **Martin Schoeberl** . NoC-based CSP Support for a Java Chip Multiprocessor. *Proceedings of the 28th Norchip Conference* November, 2010.

88. **Martin Schoeberl** . Time-predictable Chip-Multiprocessor Design. *Asilomar Conference on Signals, Systems, and Computers* November, 2010.

89. **Martin Schoeberl** , Christopher Brooks, and Edward A. Lee. Code Generation for Embedded Java with Ptolemy. *Proceedings of the 8th IFIP Workshop on Software Technologies for Future Embedded and Ubiquitous Systems (SEUS 2010)* October, 2010.

90. **Martin Schoeberl** , Thomas B. Preusser, and Sascha Uhrig. The Embedded Java Benchmark Suite JemBench. *Proceedings of the 8th International Workshop on Java Technologies for Real-Time and Embedded Systems (JTRES 2010)* 120–127, Prague, Czech Republic, August, 2010.

91. **Martin Schoeberl** and Peter Hilber. Design and Implementation of Real-Time Transactional Memory. *Proceedings of the 20th International Conference on Field Programmable Logic and Applications (FPL 2010)* 279–284, August, 2010.

92. **Martin Schoeberl** , Florian Brandner, and Jan Vitek. RTTM: Real-Time Transactional Memory. *Proceedings of the 25th ACM Symposium on Applied Computing (SAC 2010)* 326–333, March, 2010.

93. Rasmus Ulslev Pedersen and **Martin Schoeberl** . Object oriented machine learning with a multi-core real-time Java processor: short paper. *Proceedings of the 8th International Workshop on Java Technologies for Real-time and Embedded Systems (JTRES 2010)* 76–78, Prague, Czech Republic, 2010.

94. Tomas Kalibera, Pavel Parizek, Michal Malohlava, and **Martin Schoeberl** . Exhaustive testing of safety critical Java. *Proceedings of the 8th International Workshop on Java Technologies for Real-time and Embedded Systems (JTRES 2010)* 164–174, Prague, Czech Republic, 2010.

95. Benedikt Huber, Wolfgang Puffitsch, and **Martin Schoeberl** . WCET Driven Design Space Exploration of an Object Cache. *Proceedings of the 8th International Workshop on Java Technologies for Real-time and Embedded Systems (JTRES 2010)* 26–35, Prague, Czech Republic, 2010.

96. Anders P. Ravn and **Martin Schoeberl** . Cyclic executive for safety-critical Java on chip-multiprocessors. *Proceedings of the 8th International Workshop on Java Technologies for Real-time and Embedded Systems (JTRES 2010)* 63–69, Prague, Czech Republic, 2010.

97. Wolfgang Puffitsch, Benedikt Huber, and **Martin Schoeberl** . Worst-Case Analysis of Heap Allocations. *Proceedings of the 4th International Symposium On Leveraging Applications of Formal Methods, Verification and Validation (ISoLA 2010)* 464–478, 2010.

2009

98. **Martin Schoeberl** , Wolfgang Puffitsch, and Benedikt Huber. Towards Time-predictable Data Caches for Chip-Multiprocessors. *Proceedings of the Seventh IFIP Workshop on Software Technologies for Future Embedded and Ubiquitous Systems (SEUS 2009)* 180–191, November, 2009.

99. **Martin Schoeberl** , Peter Puschner, and Raimund Kirner. A Single-Path Chip-Multiprocessor System. *Proceedings of the Seventh IFIP Workshop on Software Technologies for Future Embedded and Ubiquitous Systems (SEUS 2009)* 47–57, November, 2009.

100. Rasmus Ulslev Pedersen and **Martin Schoeberl** . Educational Case Studies with an Open Source Embedded Real-Time Java Processor. *Proceedings of the 2009 Workshop on Embedded Systems Education (WESE 2009)* 71–77, Grenoble, France, October, 2009.

101. Stephen A. Edwards, Sungjun Kim, Edward A. Lee, Isaac Liu, Hiren D. Patel, and **Martin Schoeberl** . A Disruptive Computer Design Idea: Architectures with Repeatable Timing. *Proceedings of IEEE International Conference on Computer Design (ICCD 2009)* October, 2009.

102. **Martin Schoeberl** , Peter Puschner, and Raimund Kirner. Single-Path Programming on a Chip-Multiprocessor System. *Workshop on Reconciling Performance with Predictability (RePP)* October, 2009.

103. Jack Whitham, Neil Audsley, and **Martin Schoeberl** . Using Hardware Methods to Improve Time-predictable Performance in Real-time Java Systems. *Proceedings of the 7th International Workshop on Java Technologies for Real-time and Embedded Systems (JTRES 2009)* 130–139, September, 2009.

104. Philippe Moret, Walter Binder, Alex Villazon, Danilo Ansaloni, and **Martin Schoeberl** . Locating Performance Bottlenecks in Embedded Java Software with Calling-Context Cross-Profiling. *Proceedings of the 6th International Conference on the Quantitative Evaluation of SysTems (QEST 2009)* 107–108, September, 2009.

105. **Martin Schoeberl** , Walter Binder, Philippe Moret, and Alex Villazon. Design Space Exploration for Java Processors with Cross-Profiling. *Proceedings of the 6th International Conference on the Quantitative Evaluation of SysTems (QEST 2009)* 109–118, September, 2009.

106. Philippe Moret, Walter Binder, **Martin Schoeberl** , Alex Villazon, and Danilo Ansaloni. Analyzing Performance and Dynamic Behavior of Embedded Java Software with Calling-Context Cross-Profiling. *Proceedings of the 7th International Conference on the Principles and Practice of Programming in Java (PPPJ 2009)* 121–124, August, 2009.

107. **Martin Schoeberl** and Peter Puschner. Is Chip-Multiprocessing the End of Real-Time Scheduling?. *Proceedings of the 9th International Workshop on Worst-Case Execution Time (WCET) Analysis* July, 2009.

108. Benedikt Huber and **Martin Schoeberl** . Comparison of Implicit Path Enumeration and Model Checking based WCET Analysis. *Proceedings of the 9th International Workshop on Worst-Case Execution Time (WCET) Analysis* 23–34, July, 2009.

109. **Martin Schoeberl** . Time-predictable Cache Organization. *Proceedings of the First International Workshop on Software Technologies for Future Dependable Distributed Systems (STFSSD 2009)* 11–16, March, 2009.

110. Andy Wellings and **Martin Schoeberl** . Thread-local Scope Caching for Real-time Java. *Proceedings of the 12th IEEE International Symposium on Object/component/service-oriented Real-time distributed Computing (ISORC 2009)* 275–282, March, 2009.

111. Thomas Henties, James J. Hunt, Doug Locke, Kelvin Nilsen, **Martin Schoeberl** , and Jan Vitek. Java for Safety-Critical Applications. *2nd International Workshop on the Certification of Safety-Critical Software Controlled Systems (SafeCert 2009)* March, 2009.

112. Florian Brandner, Tommy Thorn, and **Martin Schoeberl** . Embedded JIT Compilation with CACAO on YARI. *Proceedings of the 12th IEEE International Symposium on Object/component/service-oriented Real-time distributed Computing (ISORC 2009)* 63–70, March, 2009.

113. Stephen A. Edwards, Sungjun Kim, Edward A. Lee, Hiren D. Patel, and **Martin Schoeberl** . Reconciling Repeatable Timing with Pipelining and Memory Hierarchy. *Workshop on Reconciling Performance with Predictability (RePP)* 2009.

2008

114. Walter Binder, Alex Villazon, **Martin Schoeberl** , and Philippe Moret. Cache-aware Cross-profiling for Java Processors. *Proceedings of the 2008 international conference on Compilers, architecture, and synthesis for embedded systems (CASES 2008)* 127–136, October, 2008.

115. Walter Binder, **Martin Schoeberl** , Philippe Moret, and Alex Villazon. Cross-Profiling for Embedded Java Processors. *Proceedings of the 5th International Conference on the Quantitative Evaluation of SysTems (QEST 2008)* 287–296, September, 2008.

116. Wolfgang Puffitsch and **Martin Schoeberl** . Non-Blocking Root Scanning for Real-Time Garbage Collection. *Proceedings of the 6th International Workshop on Java Technologies for Real-time and Embedded Systems (JTRES 2008)* 68–76, September, 2008.

117. **Martin Schoeberl** and Wolfgang Puffitsch. Non-blocking Object Copy for Real-Time Garbage Collection. *Proceedings of the 6th International Workshop on Java Technologies for Real-time and Embedded Systems (JTRES 2008)* 77–84, September, 2008.

118. **Martin Schoeberl** . Application Experiences with a Real-Time Java Processor. *Proceedings of the 17th IFAC World Congress* 9320–9325, July, 2008.

119. Peter Puschner and **Martin Schoeberl** . On Composable System Timing, Task Timing, and WCET Analysis. *Proceedings of the 8th International Workshop on Worst-Case Execution Time (WCET) Analysis* 91–101, July, 2008.

120. Christof Pitter and **Martin Schoeberl** . Performance Evaluation of a Java Chip-Multiprocessor. *Proceedings of the 3rd IEEE Symposium on Industrial Embedded Systems (SIES 2008)* 34–42, June, 2008.

121. Trevor Harmon, **Martin Schoeberl** , Raimund Kirner, and Raymond Klefstad. Toward Libraries for Real-time Java. *Proceedings of the 11th IEEE International Symposium on Object/component/service-oriented Real-time distributed Computing (ISORC 2008)* 458–462, May, 2008.

122. **Martin Schoeberl** , Stephan Korsholm, Christian Thalinger, and Anders P. Ravn. Hardware Objects for Java. *Proceedings of the 11th IEEE International Symposium on Object/component/service-oriented Real-time distributed Computing (ISORC 2008)* 445–452, May, 2008.

123. Stephan Korsholm, **Martin Schoeberl** , and Anders P. Ravn. Interrupt Handlers in Java. *Proceedings of the 11th IEEE International Symposium on Object/component/service-oriented Real-time distributed Computing (ISORC 2008)* 453–457, May, 2008.

124. Trevor Harmon, **Martin Schoeberl** , Raimund Kirner, and Raymond Klefstad. A Modular Worst-case Execution Time Analysis Tool for Java Processors. *Proceedings of the 14th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS 2008)* 47–57, April, 2008.

2007

125. **Martin Schoeberl** . SimpCon - a Simple and Efficient SoC Interconnect. *Proceedings of the 15th Austrian Workshop on Microelectronics, Austrochip 2007* October, 2007.

126. Christof Pitter and **Martin Schoeberl** . Towards a Java Multiprocessor. *Proceedings of the 5th International Workshop on Java Technologies for Real-time and Embedded Systems (JTRES 2007)* 144–151, September, 2007.

127. **Martin Schoeberl** . Architecture for Object Oriented Programming Languages. *Proceedings of the 5th International Workshop on Java Technologies for Real-time and Embedded Systems (JTRES 2007)* 57–62, September, 2007.

128. **Martin Schoeberl** and Jan Vitek. Garbage Collection for Safety Critical Java. *Proceedings of the 5th International Workshop on Java Technologies for Real-time and Embedded Systems (JTRES 2007)* 85–93, September, 2007.

129. Wolfgang Puffitsch and **Martin Schoeberl** . picoJava-II in an FPGA. *Proceedings of the 5th International Workshop on Java Technologies for Real-time and Embedded Systems (JTRES 2007)* 213–221, September, 2007.

130. **Martin Schoeberl** . A Time-Triggered Network-on-Chip. *International Conference on Field-Programmable Logic and its Applications (FPL 2007)* 377–382, August, 2007.

131. Christof Pitter and **Martin Schoeberl** . Time Predictable CPU and DMA Shared Memory Access. *International Conference on Field-Programmable Logic and its Applications (FPL 2007)* 317–322, August, 2007.

132. Raimund Kirner and **Martin Schoeberl** . Modeling the Function Cache for Worst-Case Execution Time Analysis. *Proceedings of the 44rd Design Automation Conference (DAC 2007)* 471–476, June, 2007.

133. **Martin Schoeberl** , Hans Sondergaard, Bent Thomsen, and Anders P. Ravn. A Profile for Safety Critical Java. *10th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC'07)* 94–101, May, 2007.

134. **Martin Schoeberl** . Mission Modes for Safety Critical Java. *Software Technologies for Embedded and Ubiquitous Systems, 5th IFIP WG 10.2 International Workshop (SEUS 2007)* 105–113, May, 2007.

2006

135. Rasmus Pedersen and **Martin Schoeberl** . An Embedded Support Vector Machine. *Proceedings of the Fourth Workshop on Intelligent Solutions in Embedded Systems (WISES 2006)* 79–89, June, 2006.

136. **Martin Schoeberl** . Real-Time Garbage Collection for Java. *Proceedings of the 9th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC 2006)* 424–432, April, 2006.

137. **Martin Schoeberl** . A Time Predictable Java Processor. *Proceedings of the Design, Automation and Test in Europe Conference (DATE 2006)* 800–805, March, 2006.

138. Rasmus Pedersen and **Martin Schoeberl** . Exact Roots for a Real-Time Garbage Collector. *Proceedings of the 4th International Workshop on Java Technologies for Real-time and Embedded Systems (JTRES 2006)* 77–84, Paris, France, 2006.

139. **Martin Schoeberl** and Rasmus Pedersen. WCET Analysis for a Java Processor. *Proceedings of the 4th International Workshop on Java Technologies for Real-time and Embedded Systems (JTRES 2006)* 202–211, Paris, France, 2006.

2005

140. **Martin Schoeberl** . Evaluation of a Java Processor. *Tagungsband Austrochip 2005* 127–134, October, 2005.

141. **Martin Schoeberl** . Design and Implementation of an Efficient Stack Machine. *Proceedings of the 12th IEEE Reconfigurable Architecture Workshop (RAW2005)* April, 2005.

142. Flavius Gruian, Per Andersson, Krzysztof Kuchcinski, and **Martin Schoeberl** . Automatic Generation of Application-Specific Systems Based on a Micro-programmed Java Core. *Proceedings of the 20th ACM Symposium on Applied Computing, Embedded Systems track* 879–884, March, 2005.

2004

143. **Martin Schoeberl** . A Time Predictable Instruction Cache for a Java Processor. *On the Move to Meaningful Internet Systems 2004: Workshop on Java Technologies for Real-Time and Embedded Systems (JTRES 2004)* 371–382, October, 2004.

144. **Martin Schoeberl** . Design Rationale of a Processor Architecture for Predictable Real-Time Execution of Java Programs. *Proceedings of the 10th International Conference on Real-Time and Embedded Computing Systems and Applications (RTCSA 2004)* August, 2004.

145. **Martin Schoeberl** . Java Technology in an FPGA. *Proceedings of the International Conference on Field-Programmable Logic and its Applications (FPL 2004)* 917–921, August, 2004.

146. **Martin Schoeberl** . Real-Time Scheduling on a Java Processor. *Proceedings of the 10th International Conference on Real-Time and Embedded Computing Systems and Applications (RTCSA 2004)* August, 2004.

147. **Martin Schoeberl** . Restrictions of Java for Embedded Real-Time Systems. *Proceedings of the 7th IEEE International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC 2004)* 93–100, May, 2004.

2003

148. **Martin Schoeberl** . JOP: A Java Optimized Processor. *On the Move to Meaningful Internet Systems 2003: Workshop on Java Technologies for Real-Time and Embedded Systems (JTRES 2003)* 346–359, November, 2003.

149. **Martin Schoeberl** . Design Decisions for a Java Processor. *Tagungsband Austrochip 2003* 115–118, October, 2003.

150. **Martin Schoeberl** . Using a Java Optimized Processor in a Real World Application. *Proceedings of the First Workshop on Intelligent Solutions in Embedded Systems (WISES 2003)* 165–176, June, 2003.

Technical Reports

1. Jack Whitham and Martin Schoeberl. The limits of TDMA based memory access scheduling. Technical Report YCS-2011-470, University of York, 2011.

2. Martin Schoeberl, Benedikt Huber, Walter Binder, Wolfgang Puffitsch, and Alex Villazon. Object cache evaluation. Technical report, Technical University of Denmark, 2010.

3. Martin Schoeberl, Hiren D. Patel, and Edward A. Lee. Fun with a deadline instruction. Technical Report UCB/EECS-2009-149, EECS Department, University of California, Berkeley, Oct 2009.

CV for Jan Madsen, March 2020

A N Hansens Alle 31A, st, DK-2900 Hellerup; +45 60 17 10 97; jama@dtu.dk; 1501-1963; Married, two children

Current position:
Full Professor, Deputy Director of DTU Compute (400 people), Head of Embedded Systems Engineering (30 people).

Academic Degrees:

- 1992 *Ph.D. in Computer Science*, Technical University of Denmark, Denmark.
- 1986 *M.Sc. in Electrical Engineering*, Technical University of Denmark, Denmark.

Employment track & research affiliations:

- 2002 – Full Professor, Dept. of Applied Mathematics and Computer Science, Technical University of Denmark.
- 2010 – Deputy Director, DTU Compute.
- 2008 – Head of section for Embedded Systems Engineering (ESE), DTU Computes.
- 1996 – 2002 Associate Professor, Dept. of Informatics and Mathematical Modelling, Tech. University of Denmark.
- 1992 – 1996 Assistant Professor, Dept. of Information Technology, Tech. University of Denmark.
- 1991 – 1992 Postdoc, Department of Computer Science, Technical University of Denmark.

Honors, awards & funding

- EDAA DATE Fellow, 2019
- IEEE-CEDA Outstanding Contribution Award, 2019
- DTU's Award for Scientific Advise (the DSB IC4 train braking incident investigation), 2013.
- Best Paper Awards, MECO 2013, CASES 2009
- Nominated for Research Project of the Year 2012, Biologically Inspired Hardware Cell Architecture.
- Jorck's Foundation Research Award for outstanding research activities in HW/SW Codesign, 1995.
- Coritt's Foundation Travelling Scholarship, 1995.
- Has since 2000 attracted national and EU funding for a total amount of **60+ mio DKK** (DTUs share).

Selected professional activities

- 2018 – Member of Advisory Board of Digital Hub Denmark.
- 2018 – Health Technology Expert for the Innovation Expert Panel of New North Zealand Hospital.
- 2017 – Member of ATV (The Academy of Technical Sciences).
- 2016 – Co-Founder of the IoT Centre at DTU.
- 2016 – Member of the DTU Proof-of-Concept Committee.
- 2015 – Co-Founder and Member of the Management for CACHET (Copenhagen Centre for Health Technology).
- 2015 – National ICT expert for EU H2020.
- 2013 – 2017 Member of the review panel of the Swiss Research Program Nano-Tera.
- 2010 – 2016 Editorial Board of the IEEE journal "*Design & Test*".
- 2009 – 2020 Member of the steering group of Infinit, Innovation Network for ICT.
- 2009 – 2016 Member of ARTEMIS-IA and its Strategic Research Agenda working group.
- 2008 – 2012 Member of the Governing Board of ARTEMIS JU (EU PPP Program).
- 2008 – 2011 Strategic Management Board for ArtistDesign (EU Network-of-Excellence, Embedded Systems).
- 2007 – 2009 Swedish Research Council (Vetenskapsrådet), Member of the Computer Science Panel.
- 2007 – Member of EDAA (Electronic Design Automation Association). Since 2017, Member of the Steering Board.

Patents

- Co-inventor of patent notification, Mar 12, 2018. Smart wound dressing for early diagnosis of surgical site infection.
- Inventor of patent EPC 16203030.8-1371, Dec 16, 2016. Complementary pneumatic digital logic for on-chip control of microfluidic lab-on-a-chip devices.
- Patent no.: US 20110307734, Dec 15, 2011. Biologically Inspired Hardware Cell Architecture.
- Patent no.: EP 2370937, Oct 5, 2011. Biologically Inspired Hardware Cell Architecture.
- Inventor of patent WO/2010/060923, Biologically Inspired Hardware Cell Architecture, published 03.06.2010.

Research
Methods and tools for systems engineering of computing systems. Present research covers embedded systems-on-a-chip, wireless sensor networks (Internet-of-Things), microfluidic biochips (Lab-on-Chip) and synthetic biology. Has supervised 39 PhD students (to completion) and is currently supervising 3 PhD students.

Publication and citation statistics
More than **200** peer-reviewed journal and conference papers, 13 book chapters, 3 books and 4 edited books. More than 150 invited talks and keynotes. Several best paper nominations, 2 **best paper awards** (MECO 2013, CASES 2009), 1 paper among the 30 most influential papers from 10 years of Design Automation and Test in Europe (DATE), 3 papers among the highly-cited papers in System Codesign and Synthesis (one in the top 5 cited papers). Citations: 2962, h-index: 32, i10-index: 75 (Google scholar, March 29, 2020).

Chairing scientific events:
General Chair of DATE 2018, NOCS 2012, NORCHIP 2012, CODES 2001; General Vice Chair of DATE 2017; Program Chair of CODES+ISSS 2011, DATE 2007, CODES 2000; Vice-Program Chair DATE 2006; Tutorial Chair for DATE 2006; Workshop Chair for CODES+ISSS 2005. Member of conference Steering Committees CODES+ISSS since 2000, NOCS 2012-2015, NORCHIP 2003-2016. Member of the Technical Program Committee for numerous conferences.

Professor, head of department Peter Sestoft, Date of birth: 25 June 1962

Education: 1991 PhD; 1987 MSc CS; 1984 BSc CS & Maths – all from University of Copenhagen, Denmark.

Current Positions:

2017-	Head of Department of Computer Science IT University of Copenhagen (ITU)
2008-	Professor, IT University of Copenhagen

Current honorary Positions:

2016 -	Member of national industrial PhD and postdoc grant committee

Past employment:

1999-2007	Associate Professor, IT University of Copenhagen
2002-2007	Professor MSO, Royal Veterinary and Agricultural University, Denmark
1995-2002	Associate Professor, Royal Veterinary and Agricultural University, Denmark
1992-1995	Assistant Professor, Technical University of Denmark

Selected grants:

2020	PMI-AD, EU Joint Programme – Neurodegenerative Disease Research (24 MDKK), PI Tormod Fladby, Co-PI
2019	MATRIX, Novo Nordisk Foundation (60MDKK), PI Lars Hestbjerg, Co-PI
2016	Dementia Modelling (DEMO), H2020-MSCA-ITN-2016, (€ 85.3451), PI

Teaching:
I have been teaching and supervising students at all levels since 1992. Major accomplishments have been in contribution to the start of the IT University in 1999, and especially defining and heading its first BSc program (in Software Development) 2006-2010. As head of the Computer Science Department, I have overseen the introduction of the BSc Data Science, MSc Computer Science and MSc Software Design programs.

Supervision:
Supervised 165 MSc students since 2001 and 57 BSc students since 2010, supervised 9 graduated PhD students, and co-supervised a further 4.

Research profile
My early career was focused on program analysis and program transformation, including partial evaluation where I contributed to major breakthroughs in self-applicable program transformers. Subsequently I have worked with formal specification languages and tools, the description and implementation of functional and object-oriented programming languages (including the simplest precise description of lazy evaluation in functional languages), parallel programming on modern hardware, and radically improved support for end-user (domain expert) programming tools.

Bibliometric Overview:
Approximately 50 peer-reviewed publications, including multiple substantial books; see my Google Scholar account https://scholar.google.com/citations?user=qz1BCu8AAAAJ&hl=en&oi=ao
Google Scholar indicates H-index as 24 and number of citations as 5091.

Furthermore, please indicate the total number of publications within the following categories: articles, monographs, book chapters, proceedings, other:

	Articles (Peer reviewed)	Internationl monographs	Book chapters	Proceedings (peer reviewed)	Other
Total:	12	6 + PhD thesis	2	26	6

5 Selected Publications:

1. N.D. Jones, C.K. Gomard, and P. Sestoft. Partial Evaluation and Automatic Program Generation. Englewood Cliffs, NJ: Prentice Hall, 1993. 415 pages.

2. P. Sestoft. Deriving a lazy abstract machine. Journal of Functional Programming, 7(3):231–264, May 1997.

3. P. Sestoft. Spreadsheet Implementation Technology. Basics and Extensions. MIT Press, 2014. 325 pages.

4. P. Sestoft. Java Precisely. MIT Press, third edition, March 2016. 199 pages.

5. A. Bock, T. Bøgholm, P. Sestoft, B. Thomsen, L. Leth Thomsen: On the semantics for spreadsheets with sheet-defined functions. Journal of Computer Languages, Volume 57, April 2020, 100960.

Zhoulai Fu, Assistant Professor at IT University of Copenhagen

Rued Langgaards Vej 7, Copenhagen, 2300, Denmark, zhfu@itu.dk

EDUCATION

2009 — 2013
Ph.D. in Computer Science, INRIA - Université de Rennes 1
France

2008 — 2009
M.Eng. in Software Engineering, Télécom ParisTech
France

2005 — 2008
B.S. and M.S., École Polytechnique
France

EMPLOYMENT HISTORY

Sep 2017 — Present
Assistant Professor, IT University of Copenhagen
Denmark

Developped automated testing and program analysis techniques for scientific software, in particular, in the Robotic Operating System (ROS).

2014 — 2017
Postdoc Scholar, University of California, Davis
United States

Formal methods for testing, verifying, and analyzing numerical programs, their stability and accuracy.

Mar 2013 — Aug 2013
Research Visitor, IMDEA Software Institute, Madrid
Spain

Static analysis of Java memory models

2009 — 2013
PhD Candidate, INRIA Rennes - Bretagne Atlantique
France

Studied the theory of abstract interpretation, applying to static numerical analysis for programs manipulating pointers.

Apr 2008 — Sep 2008
Research Intern , Laboratoire d'informatique de l'École polytechnique (LIX)
France

Static analysis for the Eiffel Language. Joint work with École Normale Supérieure, France and ETH Zurich, Switzerland.

RESEARCH PROJECTS

Apr 2020 — Oct 2020
Co-PI, mini-InfinIT project
Denmark

Verification of Digital Systems

Jan 2017 — Dec 2020
Project researcher, Horizon 2020 ROSIN Program
Denmark

ROS-Industrial quality-assured robot software components

2016 — 2019
Co-PI, National Science Foundation (NSF) Grant
United States

Testing and Analysis for Reliable Numerical Software.

Publications of Zhoulai Fu

Peer-reviewed (9)

1. Daming Zou, Muhan Zeng, Yingfei Xiong, **Zhoulai Fu**, Lu Zhang, and Zhendong Su, "Detecting Floating-Point Errors via Atomic Conditions." In 47th ACM SIGPLAN Symposium on Principles of Programming Languages (POPL) 2020.
2. Anders Fischer-Nielsen, **Zhoulai Fu**, Ting Su, and Andrzej Wasowski, "The forgotten case of the dependency bugs: on the example of the robot operating system," in 42nd international conference on software engineering: software engineering in practice (ICSE-SEIP), 2020.
3. **Zhoulai Fu** and Zhendong Su, "Effective Floating-Point Analysis via Weak-Distance Minimization." In 40th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI), 2019.
4. **Zhoulai Fu** and Zhendong Su, "Achieving High Coverage for Floating-Point Code via Unconstrained Programming." In 38th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI), 2017.
5. **Zhoulai Fu** and Zhendong Su, "XSat: A Fast Floating-Point Satisfiability Solver." In 28th International Conference on Computer Aided Verification (CAV), 2016.
6. **Zhoulai Fu**, Zhaojun Bai, and Zhendong Su. "Automated Backward Error Analysis for Numerical Code." In Proceedings of the 2015 ACM SIGPLAN International Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA), 2015.
7. Ting Su, **Zhoulai Fu**, Geguang Pu, Jifeng He, and Zhendong Su. "Combining Symbolic Execution and Model Checking for Data Flow Testing." In 37th International Conference on Software Engineering (ICSE), 2015.
8. **Zhoulai Fu**, "Targeted Update – Aggressive Memory Abstraction Beyond Common Sense and its Application on Static Numeric Analysis." In 23rd European Symposium on Programming (ESOP), 2014.
9. **Zhoulai Fu**, "Modularly Combining Numeric Abstract Domains with Points-to Analysis, and a Scalable Static Numeric Analyzer for Java." In 15th International Conference on Verification, Model Checking, and Abstract Interpretation (VMCAI), 2014.

Not peer-reviewed (1)

1. **Zhoulai Fu**, "A Novel Methodology for Automated Software Testing." U.S. Provisional Patent Application No. 2881381 filed in Aug, 2019, patent pending

MICROCHIP

To whom it may concern

September 28, 2020

Subject: Letter of interest for DTU Compute project: Software-Defined Hardware (SDH) w.Martin Schoeberl

Microchip is a semiconductor company and the Denmark design center excels in developing high speed Ethernet swich ASICs. For these ASIC designs all digital hardware is described using description languages such as VHDL, Verilog and System Verilog. These languages do in general not provide a larger abstraction, which in terms cause long development times and with the larger number of code lines also more errors we need to make sure are found and corrected prior ASIC production.

For future development we are very interested in the attempts to improve the design throughput and the project from prof. Martin Schoeberl on Software-Defined Hardware addresses exactly this.

If you should have any questions or require any additional information, please feel free to contact me at + 45-4485-5954 or via email at thomas.aakjer@microchip.com.

Sincerely

Thomas Aakjer.

Thomas Aakjer
Assoc. Director
Microchip Corporation A/S
Denmark

SYNOPSYS®

Synopsys, Inc.
690 East Middlefield Road
Mountain View, CA 94043-4033

T 650.584.5000
F 650.965.8637
www.synopsys.com

Technical University of Denmark
Att: Martin Schoeberl
Anker Engelundsvej 1
2800 Kgs. Lyngby
Denmark

26th September 2020

Letter of interest

We at Synopsys would like to show our interest in the Software-Defined Harware project. Our customers are intensively using UVM for digital verification of chips and FPGAs. Hence teaching UVM to students at DTU is the right path.
We intend to contribute with tool support for this project.

Best Regards

Dr. Patrick R. Haspel
University Program Director

Napatech A/S
Tobaksvejen 23A
2860 Søborg
Denmark
Tel. +45 4596 1500
www.napatech.com
CVR 10109124

napa:tech;
RECONFIGURABLE COMPUTING

Technical University of Denmark
Att: Martin Schoeberl
Anker Engelundsvej 1
Bygning 101A
2800 Kgs. Lyngby
Denmark

29 September 2020

Letter of Interest

Napatech is hereby showing our interest in following the Software-Defined Hardware project (SDH).

The mixture of tight time-to-market constraints and an increasing design complexity associated with the design and development of hybrid systems consisting of both digital hardware and software of today implies efficient design, verification and qualification methodologies are more important than ever.

Besides following the project, we will be able to contribute to the project by adding use-cases and challenges during the project.

Best regards,

Jesper Birch
FPGA Liaison Manager

TELEDYNE MARINE
Everywhereyoulook™

Technical University of Denmark
Att: Martin Schoeberl
Anker Engelunds Vej 1
Bygning 101A
2800 Kgs. Lyngby
Denmark

Teledyne RESON A/S
Fabriksvangen 13
3550 Slangerup
Denmark
Tel.: +45 4738 0022
E-mail: reson@teledyne.com
www.teledynemarine.com

25. september 2020

Letter of Interest

We at Teledyne Reson would like to show our interest in the Software-Defined Hardware project. As a company, we work with system integration and development of components and subsystems. Therefore we need well-defined tests at all levels (components/module, integration, subsystem and system). One of the component types are FPGA. It is especially for the FPGA type of component, we would be interested in new test- and verification capabilities as the current practices are to use either hardware in the loop testing or very low-level simulation verification which both have severe but different shortfalls. For the former, testing can only be done very late in the development and this poses both a technical risk but also a schedule risk. For the latter the validation at low simulation level is very time consuming. There is a need for higher abstraction level simulation and verification for firmware like VHDL and Verilog.

Therefore we recognize the potential for the use of Chisel for both development and testing, which would require compatibility with both VHDL and Verilog. Furthermore, an interest lies in open-source tools that can be used to assist external CAD tools such as ModelSim, when doing simulations of systems or components.

What we can contribute is thoughts and ideas around our need for test and development systems. We have large digital designs that run tests for extended periods. These designs could be used for testing the created verification systems for seeing the capabilities in real-life applications.

With kind regards

Morten Rytter
RD Director
Teledyne RESON A/S

DANMARKS FRIE
FORSKNINGSFOND
INDEPENDENT RESEARCH
FUND DENMARK

Confirmation of budget

This template must be used when applying the Independent Research Fund Denmark. The template must be printed, filled in, signed and stamped and subsequently uploaded as an appendix to your E-grant form / application in PDF-format.

Applicant, Martin Schoeberl
confirms with my signature:

____
Signature

that the information provided in the application regarding the budget is correct, and that the following total sum is applied for from Independent Research Fund Denmark:

6053617 DKK

Applied for amount (DKK) including overhead/administration expenses.
The amount must correspond to the amount provided in the E-grant form

The administrating institution

The administrating institution, defined as the institution or enterprise which pays for and defrays the project's/applicant's expenses during the project period, confirms with the management's signature and stamp, that the budget is approved and that the project can be carried out at the institution/ organisation / company:

Per B. Brockhoff
____
Signature and stamp of institution/organisation/company