Curriculum Vitae of Marly Roncken

marly.roncken@gmail.com, mroncken@pdx.edu
Cell phone: +1 (503) 866-5034

July 25, 2018

(work address)
Asynchronous Research Center (ARC)
Maseeh College of Engineering & Computer Science
Portland State University
1900 SW Fourth Ave, FAB 105
Portland, OR 97201
United States of America
Website: http://arc.cecs.pdx.edu/

(home address)
2623 NW Northrup Street
Portland, OR 97210
United States of America

EXECUTIVE SUMMARY

EDUCATION

- M.Sc. Mathematics, University of Utrecht, The Netherlands, 1985.
- Research in verification of parallel systems, with Willem-Paul de Roever, 1981–1982.
- Diploma Gymnasium β, Paulus Lyceum, Tilburg, The Netherlands, 1977.

EMPLOYMENT

(job details in supplement)

- 2009–present, Co-Founder and Director of the Asynchronous Research Center (ARC), and Research Professor in the Computer Science Department of the Maseeh College of Engineering and Computer Science at Portland State University, USA.
- 1997–2009, Intel Corporation, USA.
  - 2005–2008, Intel GSRC Researcher in Residence at UC Berkeley.
  - 2002–2005, Intel Strategic CAD Labs, Oregon — asynchronous on-chip communication.
  - 1999–2002, Intel Santa Clara, California — fault analysis, test and design-for-test.
  - 1997–1999, Intel Strategic CAD Labs, Oregon — asynchronous high-speed circuits/CAD.
- 1985–1997, Philips Research Laboratories, Eindhoven, The Netherlands — asynchronous VLSI.

PUBLICATIONS AND PATENTS

(bibliography references and abstracts included in supplement)

- Over 30 publications, 2 best paper awards, 1 best paper finalist, multiple invited presentations.
- US Patent 5,590,275 issued 31 December 1996, assigned to Philips.

PROFESSIONAL ACTIVITIES

- Ph.D. Committee Swetha Mettala Gilla, Portland State University. Advisors: Marly Roncken and Xiaoyu Song. Thesis title: Silicon Compilation and Test of Dataflow Implementations in GasP and Click. Swetha successfully defended her Ph.D thesis on 3 November 2017.
- Ph.D. Committee William Lee, University of Utah. Advisor: Ken Stevens. Thesis title: Achieving Backend Robustness for Timed Asynchronous Circuits. William successfully defended his Ph.D. thesis on 10 December 2015.
- Ph.D. Committee Hoon Park, Portland State University. Advisors: Marly Roncken and Xiaoyu Song. Thesis title: Formal Modeling and Verification of Delay-Insensitive Circuits. Hoon successfully defended his Ph.D. thesis on 2 November 2015.
- Ph.D. Committee Yang Xu, The University of Utah. Advisor: Ken Stevens. Thesis title: Algorithms for Automatic Generation of Relative Timing Constraints. Yang successfully defended his Ph.D. thesis on 23 February 2011.
- M.Sc. Committee Swetha Mettala Gilla, Portland State University, Advisors: Marly Roncken and Xiaoyu Song. Thesis title: Library Characterization and Static Timing Analysis of Single-Track Circuits in GasP. Swetha successfully defended her M.Sc. thesis on 29 October 2010.
- Program Chair for Memocode 2013, Portland, Oregon.
- General Chair ASYNC 2007, Program Chair ASYNC 2002, ASYNC 2014, and ASYNC 2019 to be held in Japan. Various positions over the years as ASYNC Best Paper Award Chair, Industrial Chair, and member of the Technical Program Committee and Steering Committee.

INTERESTS

- Theory and tools for application, design, verification, test, and debug of distributed systems, including asynchronous circuits and bio-computers.
- Train the next generation through internships, seminars, M.Sc. and Ph.D. research programs.

PERSONAL

- Date and Place of Birth: 7 January 1959, Posterholt, The Netherlands.
- Nationality: Dual Citizen of The Netherlands and The United States of America.
- Home address: 2623 NW Northrup Street, Portland, OR 97210, USA.
- Contact info: marly.roncken@gmail.com, mroncken@pdx.edu, +1 (503) 866-5034.

SUPPLEMENT

PERSONALIA

Name	: Maria Elisabeth (Marly) Roncken
Date and Place of Birth	: 7 January 1959, Posterholt, The Netherlands
Dual Citizenship	: The Netherlands, The United States of America
Residency	: The United States of America
Home address	: 2623 NW Northrup Street, Portland, OR 97210, USA
Work address	: Asynchronous Research Center Maseeh College of Engineering & Computer Science Portland State University 1900 SW Fourth Ave, FAB 105 Portland, OR 97201, USA
Work Website	: http://arc.cecs.pdx.edu/
Email/Phone	: marly.roncken@gmail.com, mroncken@pdx.edu, +1 (503) 866-5034

EDUCATION

M.Sc. February 1985 "Doctoraal examen" (M.Sc.) Mathematics/Computer Science, University of Utrecht, The Netherlands.

M.Sc. thesis

Part I: Specimen Database for the Department of Systematic Botany.

Part II: A Proof System for Brinch Hansen's Distributed Processes.

"Small" thesis

Order characterization within $G(\mathcal{H})^*$. (Lambda Calculus)
(written in Dutch: KARAKTERISERING VAN DE ORDE BINNEN $G(\mathcal{H})^*$.)

1981–1982 Two-year study supplement: full-time research in semantics and proof systems for parallel programming languages and distributed systems, with Prof. Willem-Paul de Roever.

September 1977 Start study of mathematics at the University of Utrecht, The Netherlands.

Diploma 1977 Diploma Gymnasium β, Paulus Lyceum, Tilburg, The Netherlands.

JOB HISTORY

Director of the Asynchronous Research Center (ARC) (May 2009 – present)

I co-founded the Asynchronous Research Center (ARC) at Portland State University in Portland, Oregon, jointly with Ivan Sutherland. Ivan is my research partner as well as my husband. Our goal is to study and teach asynchronous and globally asynchronous locally synchronous (GALS) design principles, and make them accessible to students, sponsors, and anyone interested in designing distributed systems and systems or sub-systems without a clock. We investigate low-power as well as high-speed, variation-tolerant, testable, and verifiable design principles. Around 2015 I developed a collection of design and test principles that unify and embrace the various existing asynchronous circuit families by exposing the common interfaces that they share and by keeping the specific handshaking differences strictly internal. The resulting model, which we dubbed the *link-joint model*, separates communication and storage, done in links, from computation and flow control, done in joints. Links and joints talk to each other using a generic *full-empty protocol* shared by the various asynchronous circuit families. Each joint action has an individual *go* signal that can be controlled externally to enable or disable each action individually. Link data and full or empty state storage and joint *go* signals can be read or written externally as needed for initialization, test, debug, and characterization. By separating *ab initio* communication and storage from computation and flow control as well as

actions from states we are able to simplify many aspects of the design process, including compilation, verification, and test. The link-joint model with its full-empty protocol and its local action-state control provides a clean and simple interface for hardware-software co-design and co-test. We want to use this interface to enable computer scientists and electrical engineers to collaborate and to design and test — jointly — the systems of the future whose computations, we believe, will be distributed over space and time and will be of a self-timed nature.

Intel Researcher in Residence for GSRC (http://www.src.org/program/fcrp/gsrc/) (May 2005–Oct 2008)

I was Intel's Researcher in Residence at UC Berkeley for the GigaScale Systems Research Center (GSRC) for approximately three years. My job was to encourage, establish, and strengthen collaboration between researchers in GSRC and Intel, to advise both sides, and to work with primary investigators and students on pre-competitive research projects in (1) Concurrency (2) Resilient System Design (3) Alternative Computation Models (4) Overall Design Framework and (5) Design Drivers.

Incidentally, UC Berkeley had started research in asynchronous circuits. I worked with two Berkeley students of Prof. Jan Rabaey at the Berkeley Wireless Research Center, The two students, Tsung-Te Liu and Louis Alarcon, were developing ultra low-power designs using a new style of self-timed control logic. I helped them get up to speed on asynchronous design and timing principles, and reviewed their publications and presentations.

Looking back, being a Researcher in Residence enabled me to get a better overview of academic research programs in the United States and it enabled me to meet the researchers. Before this appointment, my research focus and familiarity had been with European Research Institutes.

Intel Strategic CAD Labs (SCL) Oregon: Latency-Tolerant Design for Communication (February 2002–2009)

Note: three years overlap with my appointment as Researcher in Resident at UC Berkeley.

From 2002 to 2006, I led the SCL Research and Development project in design and CAD for on-chip communication. We used asynchronous design methods to increase the latency-tolerance and re-use of off-the-shelf (IP) blocks connected to the communication fabric. I set up joint projects between Intel SCL in Hillsboro and the Intel Ultra-mobile and Handheld Design and CAD divisions in Austin, Chandler, and Folsom. Together with Mark Schuelein of the Mobility Group, I set up an Intel Capital investment project with UK startup Silistix to develop an EDA flow for self-timed network-on-chip communication and interfacing. In 2004, we successfully completed an initial 9-month Phase-I project with Silistix, and started a follow-up Phase-II project to mature the EDA flow for low-to-medium speed bus fabrics and to tackle the design of high-speed fabrics. This collaboration ended with the sale of the Handheld Group to Marvell, November 2006. We published the key results of the first project at ASYNC 2007 [13].

In 2006, I adjusted our research focus from "design and CAD" to CAD, specifically CAD for timing closure in multi-clock and mixed synchronous-asynchronous designs and interfaces. With funding from two SRC projects, I started new collaborations with Ken Stevens, who had moved by then from Intel SCL to the University of Utah, and with Peter Beerel of the University of Southern California (USC). Both Ken and Peter were developing CAD tools for formal timing verification and static timing analysis of asynchronous and multi-clock domains, and both developments were closely related to the prior Silistix project with Intel's Handheld Group. In 2007, I extended these collaborations to include the open-source asynchronous research activities that Ivan Sutherland from Sun Microsystems had started at UC-Berkeley. This collaboration provided us with circuits to test and tune the CAD tools, and resulted in one joint paper [11] and an M.Sc. thesis by Prasad Joshi — Peter Beerel's student at USC.

After Ivan and I joined Portland State University, in 2009, we continued Prasad's M.Sc. work. The resulting 2010 M.Sc. thesis by Swetha Mettala Gilla can be downloaded from our web

site at http://arc.cecs.pdx.edu/. We also continued Ken Stevens' research on generating timing constraints for asynchronous circuits, which resulted in a 2015 Ph.D. thesis by Hoon Park.

Intel IA64 DfT Methodology, Santa Clara (2000–January 2002)
I worked on the development and deployment of Test and Design-for-Test (DfT) methods and automated these into in-house CAD flows. This was in the context of the two new HP/Intel IA64 microprocessor designs — "Mckinley" and "Madison" — Intel's second- and third-generation Itanium processors.

Intel Strategic CAD Labs (SCL) Santa Clara, California: Test R&D (1999–2000)
SCL decided to start a new division in Santa Clara. I moved there to develop an R&D program in test methodologies for novel (including asynchronous) high-speed circuit styles.

Intel Strategic CAD Labs (SCL), Oregon: Switch-Level Fault Grading (1998–1999)
After cancellation of asynchronous follow-up projects to Rappid, I volunteered to help out with the development of a novel switch-level fault simulator. The fault simulator was built around "Shark" — Intel's switch-level circuit simulator. I helped initiate, manage and co-develop the tool automation for validation and debug of the simulation software.

Intel Strategic CAD Labs (SCL) Oregon: High-speed Asynchronous Circuits (1997–1998)
It was the "Rappid" project that motivated me to leave Philips Research in Eindhoven, and join Intel's Strategic CAD Labs in Portland, Oregon. Rappid stands for "Revolving Asynchronous Pentium®Processor Instruction Decoder." As its name indicates, Rappid is an asynchronous implementation of the Intel Architecture instruction (length) decoder. My new research position at Intel was to design and test asynchronous building blocks for CPUs. Given that there was already an initial Rappid design I started with the test part. My testability analysis showed that the asynchronous pulse-domino circuits in Rappid had stuck-at testability comparable to the clocked pulse-domino circuits in the Pentium®Pro. Results from the Rappid project can be found in references [15, 23, 24].

The high visibility and success of the recent Intel and the past and present Philips asynchronous design explorations led to a number of invited presentations [17, 20, 21, 22].

In addition to evaluating Rappid, I mentored a study project with two Indian research institutes to develop non-invasive built-in test methods, and I worked with two students from Stanford University on fault modeling. Results of these studies can be found in references [16, 19, 18].

1985–1997 Philips Electronics, The Netherlands: Philips Research Laboratories.
After my M.Sc. from the University of Utrecht, I joined Philips Research Laboratories in Eindhoven. I worked initially in my M.Sc. area: specification and verification of distributed systems. But after in-house training in VLSI design, I moved more and more into the area of electronic circuit design.

From 1989 onwards, I did research, development and implementation of test and DfT strategies for asynchronous VLSI circuits in the Tangram[1] project [25, 29, 30, 34], including all IC test operations and characterization measurements on the HP82000 tester [28, 32, 42]. All test strategies went hand-in-hand with manufactured demonstrator chips [26, 27, 33, 36, 37, 38, 40].

Below follows a summary and timeline of my projects within the Tangram team.

Philips Research Low-Power Project (1996–1997)
Evaluation and improvement of the test method developed with sponsorship from EC JESSI AC-6 (see below), and its application to a low-power decoder for a DECT-ADPCM function.

Philips EC JESSI project AC-6 (1995–1997)
Development of an economic, defect-oriented, push-button test method for asynchronous ICs

[1] In 2004, the Tangram team became the Philips incubator *Handshake Solutions*.

under JESSI project AC-6 "Test Engineering for Integrated Circuits and Systems," Goals of the project — all of which were accomplished — were:

1. assure high-quality defect coverage (based on inductive fault analysis),
2. allow (semi-automated) sequential test generation,
3. exploit (single) scan latches, and
4. preserve "clock-gating" in datapaths.

Philips EC Esprit project EXACT (1992–1995)
Development of a test method easy to integrate in the Tangram design flow. This test method, dubbed "programmed scan," was subsequently incorporated in the Tangram Digital Compact Cassette (DCC) player ICs and deployed in the final single-rail DCC demonstrator chip for Esprit project 6143 "EXploitation of Asynchronous Circuit Technologies" (EXACT).

AWARDS

- Outstanding Researcher Award for 2018 in Engineering and Computer Sciences, awarded by the Columbia Willamette Chapter of Sigma Xi, The Scientific Research Society.
- Award for best paper finalist of the International Symposium on Asynchronous Circuits and Systems (ASYNC), Mountain View, California, USA, 2015 — see reference [8].
- Award for best paper of the International Symposium on Advanced Research in Asynchronous Circuits and Systems (ASYNC), Barcelona, Spain, 1999 — see reference [24].
- Award for best paper of the International Symposium on Advanced Research in Asynchronous Circuits and Systems (ASYNC), Salt Lake City, Utah, USA, 1994 — see reference [34].

PATENTS

- Kees van Berkel, Marly Roncken, and Ronald Saeijs, Philips Research.
  Method for testing an integrated circuitry and an integrated circuit having a plurality of functional components and having junction/switch test components in interconnecting channels between functional components.

  EP556894A1 Applicant NV Philips' Gloeilampen Fabrieken, Issued August 25, 1993

  US5590275 Applicant U.S. Philips Corporation, Issued December 31, 1996

EXTERNAL PROJECTS

- **DARPA: Flexible Specification, Analysis, and Implementation of Self-Timed Circuits: December 2016-present**
  We teamed up with the formal verification research group of Dr. Warren Hunt at the University of Texas at Austin (UT Austin), under DARPA Order FA8650-17-1-7704 for UT Austin with a subcontract to Portland State University. This contract supports in part the joint development of a formal semantics and verification framework for self-timed circuits and systems designed using the link-joint model developed by the Portland State team and to be verified by using and extending the ACL2 theorem proving framework developed by the UT Austin team. ACL2 has been used successfully in verifying industry-scale synchronous circuits and systems. We published the initial results for this DARPA contract at LNCS 2017 [4] and ASYNC 2018 [2].

- **Semiconductor Research Corporation (SRC — http://www.src.org): 2006-2009**
  - Relative Timing for Interconnect of Multi-Frequency SoC Designs,
    by Dr. Ken Stevens (University of Utah),
    SRC Logic & Physical Design Task ID 1424.001.

– Static Timing Analysis for Interconnect of Multi-Frequency SoC Designs,
by Dr. Peter Beerel (University of Southern California),
SRC Logic & Physical Design Task ID 1425.001.

These two external research projects were sponsored by SRC and supervised by me on behalf of Intel. Together, the projects took a first step in closing the gap in timing closure up to but excluding Place and Route of multi-clock and mixed synchronous-asynchronous systems-on-chip. Project 1424 at Utah was set up to generate a complete set of relative timing constraints for a given design, i.e. timing constraints that make the circuit behave as intended. Project 1425 at USC was set up to format and partition these constraints into timing paths that can be handled by standard (synchronous-oriented) EDA tools like PrimeTime.

The two projects resulted in a basic flow for pre- and post-layout timing validation that has been used as a starting point by many other asynchronous research and development groups, including our own group at the Asynchronous Research Center.

- **DARPA/CLASS: Developing a Mainstream Clockless Design Infrastructure: 2004-2007**
Boeing was prime contractor on this CLASS project, which was managed by Bob Reuss of DARPA/MTO. This was an industrial-academic collaboration project with four participating companies: Boeing for flow integration and application, Handshake Solutions (who joined during the second half) for the main EDA flow, and Theseus Logic and Codetronix for ultra-robust asynchronous design and the corresponding EDA flow. This collaboration was augmented with five participating universities, to fine-tune the EDA flows with (1) higher-speed circuit styles (Columbia University, University of North Carolina at Chapel Hill, University of Washington), (2) test and debug (Yale) and (3) noise-coupling and Electro Magnetic Interference analysis (Oregon State University). I served on the Industrial Advisory Board.

- **Intel External Project Sponsorship: 1998-2001**
I mentored a study on "The Problems of Testing Asynchronous Circuits and their Solutions" by Dr. Parimal Pal Chaudhuri (Bengal Engineering College, Sibpur, West Bengal, India) and Dr. Susmita Sur-Kolay (Indian Statistical Institute, Calcutta, India) and their students.

  This study fits in the context of my job history at Intel Strategic CAD Labs, and is mentioned there under the years 1997-1998.

CONFERENCE INVOLVEMENT

- Various positions over the years within the IEEE ASYNC International Conference series on Asynchronous Circuits and Systems: as General and Program Chair, Best Paper Award Chair, Industrial Chair, and member of the Technical Program Committee and Steering Committee. More details follow below.
- Technical Program Chair ASYNC 2019, Japan, co-chaired with Andrey Mokhov (Newcastle Upon Tyne, UK).
- Technical Program Chair ASYNC 2014, Potsdam, Germany, co-chaired with Andreas Steininger (Vienna, Austria).
- General Program Chair ASYNC 2007, Berkeley, California, USA, co-chaired with Peter Beerel (USC, USA).
- Technical Program Chair ASYNC 2002, Manchester, UK, co-chaired with Simon Moore (Cambridge, UK).
- Technical Program Chair Memocode 2013, the 11th ACM/IEEE International Conference on Formal Methods and Models for Codesign, Portland, Oregon, USA, co-chaired with Jean-Pierre Talpin (Inria, France).

- Member of the Program Committee of the annual IEEE Memory Technology, Design, and Testing Workshop (MTDT), 2000–2001.
- Member of the Program Committee of the annual IEEE IDDQ Workshop, a.k.a. International Workshop on Current & Defect Based Testing (DBT), 1999–2001.
- Member of the Program Committee of the annual IEEE International Test Synthesis Workshop (ITSW), 1999–2001.
- Organization of a paper and panel session on testing of asynchronous circuits at the annual IEEE International Test Conference (ITC), Washington, D.C., USA, 1996. I organized this together with Philips colleague Eric Bruls.

EXTERNAL INVITED TALKS AND SEMINARS

- *Asynchronous Computing*, given jointly with Ivan Sutherland, and presented as part of the distinguished ShanghaiTech Lecture series, 5 July 2018. Presentation slides with notes can be downloaded from our web site at http://arc.cecs.pdx.edu/ under Publications, Selected Talks.
- *Understanding Self-Timed Circuits*, 8-hour seminar, given jointly with Ivan Sutherland, and presented both in full and in condensed form as a Distinguished Lecture Series at Nanyang Technological Institute in Singapore, and presented in part at Shanghai Jiao Tong University and at Fudan University in Shanghai, January-February 2016. The presentation slides can be downloaded from our web site at http://arc.cecs.pdx.edu/ under Publications, Selected Talks.
- *Think bigger than Crocks*, Advanced Research in Asynchronous Circuits and Systems, (ASYNC), Eilat, Israel, 2000 (panel).
- *Testability of Asynchronous Circuits*, Advanced Research in Asynchronous Circuits and Systems, (ASYNC), Eilat, Israel, 2000 (tutorial).
- *Asynchronous Design Issues and IDDQ Testing*, 5th IEEE International Workshop on IDDQ and Defect based Testing (IDDQ), Dana Point, California, USA, 1999 (talk).
- *Testability is* no *excuse to prevent a transit from Clocked to Asynchronous Design*, International Test Synthesis Workshop (ITSW), Santa Barbara, California, USA, 1999 (talk).
- *Trends in Asynchronous Circuit Design and Test*, Weekly Reliability and Testability Seminar (RATS), Center for Reliability Computing, Stanford University, January 25, 1999 (talk).
- *Asynchronous Design: Working the Fast Lane*, International Test Conference (ITC), Washington, D.C., USA, 1996 (panel).
- *Testing Asynchronous Chips*, Israel Workshop on Asynchronous VLSI, Kibutz Genossar, sponsored by Ministry of Science and the Arts and VLSI Systems Research Center Technion, Israel, 1995 (talk).

Note: The following references include, in chronological order, journal and conference publications, publications in books, publicly available reports, and invited presentations. Abstracts are included where possible. Recent references can be downloaded from the Asynchronous Research Center (ARC) web site at Portland State University — see "Publications" at http://arc.cecs.pdx.edu/.

References

[1] Marly Roncken and Ivan Sutherland. Asynchronous Computing. *The ShanghaiTech Lecture*, 5 July 2018.

This is the first time that we presented our vision of using the link-joint model as "the new RTL" — the "Register Transfer Level" hardware-software interface for computations distributed over space and time. The link-joint model with its full-empty protocol and with its local action-state control provides a clean and simple interface for hardware-software co-design-and-test. We want to use this clean and simple interface to enable computer scientists and electrical engineers to collaborate and to design and test — jointly — the systems of the future whose computations, we believe, will be distributed over space and time and will be of a self-timed nature.

[2] Cuong Chau, Warren Hunt Jr., Matt Kaufmann, Marly Roncken, and Ivan Sutherland. Data-Loop-Free Self-Timed Circuit Verification. *Proceedings of the 24th IEEE International Symposium on Asynchronous Circuits and Systems* (ASYNC), Vienna, Austria, pages 51–58, 13-16 May, 2018.

This paper presents a methodology for formally verifying the functional correctness of self-timed circuits whose dataflows are free of feedback loops. In particular, we formalize the relationship between their input and output sequences. We use the DE system, a formal hardware description language built using the ACL2 theorem-proving system, to specify and verify finite-state-machine representations of self-timed circuit designs. We apply a link-joint paradigm to model self-timed circuits as networks of computations that communicate with each other with protocols. Our approach exploits hierarchical reasoning and induction to support scalability. We demonstrate our methodology by modeling and verifying several data-loop-free self-timed circuits.

[3] Marly Roncken, Ivan Sutherland, Chris Chen, Yong Hei, Warren Hunt Jr., and Cuong Chau, with Swetha Mettala Gilla, Hoon Park, Xiaoyu Song, Anping He, and Hong Chen. How to Think about Self-Timed Systems. *Proceedings of the IEEE Asilomar Conference on Signals, Systems, and Computers*, pages 1597–1604, October, 2017.

Self-timed systems divide nicely into two kinds of components: communication links that transport and store data, and computation joints that apply logic to data. We treat these two types of self-timed components as equally important. Putting communication on a par with computation acknowledges the increasing cost of data transport and storage in terms of energy, time, and area. Our clean separation of data transport and storage from logic simplifies the design and test of self-timed systems. The separation also helps one to grasp how self-timed systems work. We offer this paper in the hope that better understanding of self-timed systems will engage the minds of compiler, formal verification, and test experts.

[4] Cuong Chau, Warren Hunt Jr., Marly Roncken, and Ivan Sutherland. A Framework for Asynchronous Circuit Modeling and Verification in ACL2. *O. Strichman and R. Tzoref-Brill (Eds.): Haifa Verification Conference*, LNCS 10629, pages 3–18, Springer International Publishing, 2017.

Formal verification of asynchronous circuits is known to be challenging due to highly non-deterministic behavior exhibited in these systems. One of the main challenges is that it is very difficult to come up with a systematic approach to establishing invariance properties, which are crucial in proving the correctness of circuit behavior. Non-determinism also results in asynchronous circuits having a complex state space, and hence makes the verification task much more difficult than in synchronous circuits. To ease the verification task by reducing non-determinism, and consequently reducing the complexity of the set of execution paths, we impose design restrictions

to prevent communication between a module M and other modules while computations are still taking place that are internal to M . These restrictions enable our verification framework to verify loop invariants efficiently via induction and subsequently verify the functional correctness of asynchronous circuit designs. We apply a link-joint paradigm to model asynchronous circuits. Our framework applies a hierarchical verification approach to support scalability. We demonstrate our framework by modeling and verifying the functional correctness of a 32-bit asynchronous serial adder.

[5] Marly Roncken, Chris Cowan, Ben Massey, Swetha Mettala Gilla, Hoon Park, Robert Daasch, Anping He, Yong Hei, Warren Hunt Jr, Xiaoyu Song, and Ivan Sutherland. Beyond Carrying Coal To Newcastle: Dual Citizen Circuits. *A. Mokhov (Ed.): This Asynchronous World — Essays dedicated to Alex Yakovlev on the occasion of his 60th birthday*, pages 241–261, Newcastle University, UK, 2016.

This paper makes self-timed circuits dual citizens by providing a clocked mode of operation in addition to their self-timed mode. The clocked or synchronous mode of operation re-uses the self-timed fabric and protocols, and thereby — beneficially — inherits the elasticity of the self-timed or asynchronous mode of operation. In exchange, clocked circuit operations can build confidence in self-timed circuit operations or replace aging or erratic self-timed circuit operations that need more time to finish. Once confidence is gained, the self-timed mode of operation can serve as a turbo mode to obtain better latency, throughput, energy, robustness to delay variations, or electromagnetic compatibility. The dual citizen circuits in this paper have individual action control. As a result, the circuits can either run in a fixed mode — self-timed or clocked — and switch modes on the fly, or run in both modes concurrently.

[6] Marly Roncken and Ivan Sutherland. Understanding Self-Timed Circuits. *Joint talk series for Singapore (NTU) and Shanghai (SJTU, Fudan)*, January-February 2016.

Nearly all modern digital computers march to the beat of a "clock." The computer clock divides each second into a few billion "clock periods" just as a school bell divides each day into fixed-length class periods. A 55-minute class period is so useful for scheduling students and classrooms that educators rarely ask if it is best for learning. In reality, 55 minutes is either too short or too long.

We are one of a few research groups who study how to replace the rigid clock with more flexible "self-timed" regimes. Self-timed systems allow each small task to take its own natural time just as "self-paced" learning allows each student to learn at his or her own natural pace. Easy tasks finish quickly and take little energy. Difficult tasks require more time and energy. In operation a self-timed system is as orderly as a kindergarten playground at recess. Marly and Ivan will show how the parts of such a system interact, and how they can be designed, validated, and tested.

– **Outline**

1. What and where is the ARC? : by Marly (sub-part of people introductions)
2. Self-Timed Circuits : by Ivan
3. Building Blocks and Protocols : by Marly
4. Measuring Performance : by Ivan
5. Timing Validation : by Marly
6. Arbitration: Who wins? : by Ivan
7. Initialization, Test, and Debug : by Marly
8. The Weaver, an 8x8 Crossbar : by Ivan

Each talk is about an hour, including a short break between talks. Talk 3 has sound effects.

– **Biography**

Marly Roncken and Ivan Sutherland have each worked with self-timed systems for more than 25 years. They joined forces by marriage in 2006. Marly, a mathematician by training, focuses on logic, structure, and correctness. Ivan, an electrical engineer by training, designs circuits and systems. Together they run the Asynchronous Research Center (ARC) at Portland State University (PSU). Ivan is best known for his early work in computer graphics, a field he left in 1976. Since his Turing Award lecture in 1988 he has dedicated his work to escaping from the "Tyranny of the Clock" in digital logic.

[7] Hoon Park, Anping He, Marly Roncken, Xiaoyu Song, and Ivan Sutherland. Modular Timing Constraints for Delay-Insensitive Systems. *Journal of Computer Science and Technology (JCST)* Vol. 31, No. 1, pages 77–106, Springer, January 2016.

> This paper introduces ARCtimer, a framework for modeling, generating, verifying, and enforcing timing constraints for individual self-timed handshake components. The constraints guarantee that the component's gate-level circuit implementation obeys the component's handshake protocol specification. Because the handshake protocols are delay insensitive, self-timed systems built using ARCtimer-verified components are also delay insensitive. By carefully considering time locally, we can ignore time globally. ARCtimer comes early in the design process as part of building a library of verified components for later system use. The library also stores static timing analysis (STA) code to validate and enforce the component's constraints in any self-timed system built using the library. The library descriptions of a handshake component's circuit, protocol, timing constraints, and STA code are robust to circuit modifications applied later in the design process by technology mapping or layout tools. In addition to presenting new work and discussing related work, this paper identifies critical choices and explains what modular timing verification entails and how it works.

[8] Marly Roncken, Swetha Mettala Gilla, Hoon Park, Navaneeth Jamadagni, Chris Cowan, and Ivan Sutherland. Naturalized Communication and Testing. *Proceedings of the 21st IEEE International Symposium on Asynchronous Circuits and Systems* (ASYNC), Mountain View, California, USA, pages 77–84, 4-6 May, 2015 (award for best paper finalist).

> We "naturalize" the handshake communication links of a self-timed system by assigning the capabilities of filling and draining a link and of storing its full or empty status to the link itself. This contrasts with assigning these capabilities to the joints, the modules connected by the links, as was previously done. Under naturalized communication, the differences between Micropipeline, GasP, Mousetrap, and Click circuits are seen only in the links — the joints become identical; past, present, and future link and joint designs become interchangeable.
>
> We also "naturalize" the actions of a self-timed system, giving actions status equal to states — for the purpose of silicon test and debug. We partner traditional scan test techniques dedicated to *state* with new test capabilities dedicated to *action*. To each and every joint, we add a novel proper-start-stop circuit, called MrGO, that permits or forbids the action of that joint. MrGO, pronounced "Mister GO," makes it possible to (1) exit an initial state cleanly to start circuit operation in a delay-insensitive manner, (2) stop a running circuit in a clean and delay-insensitive manner, (3) single- or multi-step circuit operations for test and debug, and (4) test sub-systems at speed.

[9] Hoon Park, Anping He, Marly Roncken, and Xiaoyu Song. Semi-Modular Delay Model Revisited in Context of Relative Timing. *IEEE Electronics Letters*, Vol. 51, No. 4, pages 332–334, February 2015.

> We present a new definition of semi-modularity to accommodate relative timing constraints in self-timed circuits. While previous definitions ignore such constraints, the new definition takes them into account. We illustrate the difference on a design solution for a well-known speed-independent circuit implementation of the Muller C element and a set of relative timing constraints that renders the implementation hazard-free. The old definition produces a false semi-modularity conflict that cannot exist due to the set of imposed constraints. The new definition correctly accepts the solution.

[10] Swetha Mettala Gilla, Marly Roncken, and Ivan Sutherland. Long-Range GasP with Charge Relaxation. *Proceedings of the 16th IEEE International Symposium on Asynchronous Circuits and Systems* (ASYNC), Grenoble, France, May 3–6, pages 185–195, 2010.

> GasP circuit modules communicate handshake signals in two directions over a single state wire. The 2008 Infinity test chip demonstrated GasP in 90 nm CMOS operating at four giga data items per second, but revealed that state wires about 5000 lambda long retard operation by about 10%. Simulations reported in this paper show that GasP modules will tolerate surprisingly long state wires, albeit at reduced throughput. The modules appear to operate correctly with state wires

whose delay exceeds the drive time. With such long wires, the receiving module waits until passive distribution of charge brings the wire within range of the receiver's switching threshold. Having put enough charge into the wire, or vice-versa removed enough charge from it, the sending module may proceed with its next task. This result applies equally to other single-track signaling methods. This behavior calls for a new kind of relative timing constraint to address when the wire charging or discharging process may cease rather than when the signal reaches the far end of the wire.

[11] Prasad Joshi, Peter A. Beerel, Marly Roncken, and Ivan Sutherland. Timing Verification of GasP Asynchronous Circuits: Predicted Delay Variations Observed by Experiment. *D. Dams, U. Hannemann, M. Steffen (Eds.): Willem-Paul de Roever Festschrift*, LNCS 5930, pages 260–276, Springer-Verlag Berlin Heidelberg, 2010.

This paper uses the method of Logical Effort to reason about the timing of 6-4 GasP asynchronous Network on Chip (NoC) control circuits. Logical Effort provides an analytic relationship between the drive strength of a circuit, the load it drives, and its delay. The important environmental variable in this relationship is the physical distance from one GasP control module to adjacent modules because longer wire connections present greater capacitance that retards the operation of their drivers. Remarkably, this analysis predicts correct operation over a large range of distances provided the difference in the distances to predecessor and successor modules is limited, and predicts failure if the distances differ by too much. Experimental support for this analytical prediction comes from the measured behavior of a test chip called Infinity built by Sun Microsystems in 90 nanometer CMOS circuits fabricated at TSMC.

[12] Peter A. Beerel and Marly E. Roncken. Low Power and Energy Efficient Asynchronous Design. *Journal of Low Power Electronics* (JOLPE), Vol. 3, No. 3, pages 234–253, December 2007 (invited paper).

This paper surveys the most promising low-power and energy-efficient asynchronous design techniques that can lead to substantial advantages over synchronous counter parts. Our discussions cover macro-architectural, micro-architectural, and circuit-level differences between asynchronous and synchronous implementations in a wide range of designs, applications, and domains including microprocessors, application-specific designs, and networks on chip.

[13] Andrew M. Scott, Mark E. Schuelein, Marly Roncken, Jin-Jer Hwan, John Bainbridge, John R. Mawer, David L. Jackson, and Andrew Bardsley. Asynchronous on-Chip Communication: Explorations on the Intel®PXA27x Processor Peripheral Bus. *Proceedings of the 13th IEEE International Symposium on Asynchronous Circuits and Systems* (ASYNC), Berkeley, California, USA, pages 60–72, March 12–14, 2007.

For today's SoC designer, on-die variation, clock distribution, timing closure, and power concerns confront the desire to get products to market quicker. Each new process generation makes the challenge greater as process skews, complexity, and frequency become more onerous. This is particularly true for signals that have to travel across larger portions of a chip such as clocks and buses. In this paper, we examine the use of GALS (Globally Asynchronous, Locally Synchronous) techniques to address on-chip communication between different synchronous modules on a bus. We explore issues related to validation, module interfaces and tool flows, while looking at advantages in power savings, timing closure and Time-to-Market/Time-to-Money (TTM). Our exploration vehicle is the Intel®PXA27x Peripheral Bus (PB) — a common interface for connecting peripherals on PXA27x and related processor families in Intel's cellular and handheld application and communication domain.

[14] John O'Leary and Marly Roncken. (IN MEMORIAM) Rob Tristan Gerth: 1956–2003. *Proceedings of the 16th International Conference on Computer Aided Verification* (CAV), Boston, Massachusetts, USA, LNCS 3114, pages 1–14, July 13–17, 2004.

On Friday November 28, 2003, computer scientist and logician Rob Gerth died from sudden cardiac arrest. It is impossible to say all that the loss of Rob means to Intel, to the verification community, and to each of us personally. Rob's passion for knowledge reached far beyond computer science

and logic: it embraced science, history, literature, art and music. The 2004 Proceedings of the 16th International Conference on Computer Aided Verification are dedicated to him. This article in the Proceedings surveys highlights of Rob's scientific career, followed by excerpts taken from some of the many tributes sent for Rob's funeral by friends and colleagues from around the world. The list of contributors, though too long to be printed here, is itself a tribute to Rob's impact on us all.

[15] Ken Stevens, Shai Rotem, Ran Ginosar, Peter Beerel, Chris Meyers, Kenneth Y. Yun, Rakefet Kol, Charles Dike, and Marly Roncken. An Asynchronous Instruction Length Decoder. *IEEE Journal of Solid-State Circuits* Vol. 36, No. 2, pages 217–228, February 2001.

This paper describes an investigation of potential advantages and pitfalls of applying an asynchronous design methodology to an advanced microprocessor architecture. A prototype complex instruction set length decoding and steering unit was implemented using self-timed circuits. The Revolving Asynchronous Pentium®Processor Instruction Decoder (RAPPID) design implemented the complete Pentium II®32-bit MMX instruction set. The prototype chip was fabricated on a 0.25-μ CMOS process and tested successfully. Results show significant advantages — in particular, performance of 2.5–4.5 instructions per nanosecond — with manageable risks using this design technology. The prototype achieves three times the throughput and half the latency, dissipating only half the power and requiring about the same area as the fastest commercial 400-MHz clocked circuit fabricated on the same process.

[16] Susmita Sur-Kolay, Marly Roncken, Ken Stevens, Parimal Pal Chaudhuri, and Rob Roy. Fsimac: A Fault Simulator for Asynchronous Sequential Circuits. *Proceedings of the 9th Asian Test Symposium* (ATS), Taipei, Taiwan, pages 114–119, December 4–6, 2000.

At very high frequencies, the major potential of asynchronous circuits is absence of clock skew and, through that, better exploitation of relative timing relations. This paper presents *Fsimac*, a gate-level fault simulator for stuck-at and gate-delay faults in asynchronous sequential circuits. Fsimac not only evaluates combinational logic and typical asynchronous gates such as Muller C-elements, but also complex domino gates, which are widely used in high-speed designs. Our algorithm for detecting feedback loops is designed so as to minimize the iterations for simulating the unfolded circuit. We use min-max timing analysis to compute the bounds on the signal delays. Stuck-at faults are detected by comparing logic values at the primary outputs against the corresponding values in the fault-free design. For delay faults, we additionally compare min-max time stamps for primary output signals. Fault coverage reported by Fsimac for pseudo-random tests generated by Cellular Automata show some very good results, but also indicate test holes for which more specific patterns are needed. We intend to deploy Fsimac for designing more effective CA-BIST.

[17] Marly Roncken. Testability of Asynchronous Circuits. *Tutorial Advanced Research in Asynchronous Circuits and Systems* (ASYNC), Eilat, Israel, April 2, 2000 (invited tutorial).

Where successful, the strength of asynchronous circuit implementations is often attributed to the elimination of surplus (particularly global) activity, resulting in lower power or in higher speed than the clocked version. Examples are Myna, the asynchronous 80C51 pager version by Philips (low power, low EM radiation), and RAPPID, the asynchronous version of the Pentium Instruction Length Decoder by Intel (high speed).

Although both objectives, low power and high speed, are obtained by using asynchronous (self-timed) operations, they strive for opposite effects with different testability aspects:

- **REAL low power circuits are seldom Active**
  This means that per test only a small portion of the circuit (namely the fraction activated by the test) can be controlled and observed. This is a real limitation, and can put an end to a low-cost supplementary and high-quality test method like IDDQ for which massive controllability is essential.
- **REAL high speed circuits are seldom Quiescent**
  Most test methods (voltage and current based) need quiescent (stable) states to inject test stimuli and observe the circuit's responses. Fewer quiescent states means lower controllability and observability for testing the circuit. This makes testing and debugging a lot harder.

In this tutorial, we illustrate and examine the above testability aspects at the hand of RAPPID and, more substantially, for benchmarks from Philips. The goal is not so much to present a solution (although we will do that for the Philips benchmarks) but to show that testability is an objective criterion of asynchronous circuit design on a par with timing and power, for which practical and realistic tradeoffs are possible.

[18] Philip P. Shirvani, Subhasish Mitra, Jo C. Ebergen, and Marly Roncken. DUDES: A Fault Abstraction and Collapsing Framework for Asynchronous Circuits. *Proceedings of the International Symposium on Advanced Research in Asynchronous Circuits and Systems* (ASYNC), Eilat, Israel, April 2–6, pages 73–82, 2000.

This paper addresses the problem of fault collapsing in asynchronous circuits. We investigate different transistor-level implementations of some basic elements that are used in delay-insensitive asynchronous circuit designs, and analyze them in the presence of single stuck-at faults. From this analysis, we conclude that all internal stuck-at faults which are detectable by Boolean testing, can be represented as pin faults. This abstraction makes it possible to perform fault simulation at the logic level (network of basic elements) rather than at transistor level, which reduces the simulation time. We show how this fault model, called DUDES, can be used for fault collapsing to reduce the size of fault lists at the logic level, thereby reducing the simulation time even further. We set the basis for a formal technique for deriving equivalence relationships among the faults under consideration, using trace expressions, and illustrate that this formal technique also supports fault collapsing at the system level. This framework can be expanded to a theory of fault abstraction and collapsing for asynchronous circuits that can reduce the complexity of test pattern generation and fault simulation.

[19] Marly Roncken, Ken Stevens, Rajesh Pendurkar, Shai Rotem, and Parimal Pal Chaudhuri. CA-BIST for Asynchronous Circuits: A Case Study on the RAPPID Asynchronous Instruction Length Decoder. *Proceedings of the International Symposium on Advanced Research in Asynchronous Circuits and Systems* (ASYNC), Eilat, Israel, pages 62–72, April 2–6, 2000.

This paper presents a case study in low-cost non-invasive Built-In Self Test (BIST) for RAPPID, a large-scale 120,000-transistor asynchronous version of the Pentium Pro Instruction Length Decoder, which runs at 3.6 GHz. RAPPID uses a synchronous 0.25 micron CMOS library for static and domino logic, and has no Design-for-Test hooks other than some debug features. We explore the use of Cellular Automata (CA) for on-chip test pattern generation and response evaluation. More specifically, we look for fast ways to tune the CA-BIST to the RAPPID design, rather than using pseudo-random testing. The metric for tuning the CA-BIST pattern generation is based on an abstract hardware description model of the instruction length decoder, which is independent of implementation details, and hence also independent of the asynchronous circuit style. Our CA-BIST solution uses a novel bootstrap procedure for generating the test patterns, which give complete coverage for this metric, and cover 94% of the testable stuck-at faults for the actual design at switch level. Analysis of the undetected and untestable faults shows that the same fault effects can be expected for a similar clocked circuit. This is encouraging evidence that testability is no excuse to avoid asynchronous design techniques in addition to high-performance synchronous solutions.

[20] Marly Roncken. Asynchronous Design Issues and IDDQ Testing. *5th IEEE International Workshop on IDDQ and Defect based Testing* (IDDQ), Dana Point, California, USA, 1999 (invited presentation).

[21] Marly Roncken. Testability is *no* excuse to prevent a transit from Clocked to Asynchronous Design. *International Test Synthesis Workshop* (ITSW), Santa Barbara, California, USA, 1999 (invited presentation).

This proposition paper argues that taking advantage of asynchronous design methodologies on top of existing high-speed VLSI design technologies can be done while keeping the testability risks manageable. Asynchronous, or self-timed, approaches utilize handshake protocols to guarantee the correct relative ordering of events in the circuit. Testability is considered a major potential risk

for such an approach, because the self-timed operation limits the controllability and observability of the circuit under test. Limited test control and access may be compensated by building additional Design-for-Test (DfT) measures into the circuit, provided that these still allow the design requirements to be met. In [29] *(reference number in this CV)*, we presented a synthesizable DfT solution for the low-power asynchronous VLSI design methodology used by Philips and targeted for audio-range applications. Here, we investigate the extra DfT requirements needed to accommodate the high-speed asynchronous design methodologies that Intel is interested in.

[22] Marly Roncken. Trends in Asynchronous Circuit Design and Test. *Weekly Reliability and Testability Seminar* (RATS), Center for Reliability Computing, Stanford University, Edward J. McCluskey, January 25, 1999 (invited presentation).

Synchronous design methodologies are less synchronous than they used to be. To meet performance targets, designers of clocked circuits already use various self-timed and self-resetting signaling techniques to bring out advantages in speed or power. Future chips will likely be asynchronous - to a very large extent.

This presentation starts with showing the implications of the 1997 Semiconductor Industry Association (SIA) Technology Roadmap that motivate this asynchronous design trend. Then we focus on two extremes of asynchronous design. The first one is a silicon compilation approach developed by Philips and targeted for low-power product lines (the first pager product came on the market in 1998). The second one is a highly custom design approach, under development by Intel and used in a pilot project design of a 400 MHz Pentium®Processor Instruction Length Decoder. We discuss design, test, and CAD/CAT implications for both extremes.

[23] Ken Stevens, Shai Rotem, Steve Burns, Jordi Cortadella, Ran Ginosar, Mike Kishinevsky, and Marly Roncken. CAD Directions for High Performance Asynchronous Circuits. *Proceedings ACM/IEEE Design Automation Conference* (DAC), New Orleans, Louisiana, USA, pages 116–121, 1999 (invited paper).

This paper describes a novel methodology for high performance asynchronous design based on timed circuits and on CAD support for their synthesis using Relative Timing. This methodology was developed for a prototype iA32 instruction length decoding and steering unit called RAPPID (Revolving Asynchronous Pentium®Processor Instruction Decoder) that was fabricated and tested successfully. Silicon results show significant advantages—in particular performance of 2.5-4.5 instructions per nS—with manageable risks using this design technology. RAPPID achieves three times faster performance and half the latency, dissipating only half the power and requiring a minor area penalty relative to a comparable 400MHz clocked circuit.

Relative Timing is based on user-defined and automatically extracted *relative* timing assumptions between signal transitions in a circuit and its environment. It supports the specification, synthesis, and verification of high-performance asynchronous circuits, such as pulse-mode circuits, that can be derived from an initial speed-independent specification. Relative timing presents a "middle-ground" between clocked and asynchronous circuits, and is a fertile area for CAD development. We discuss possible directions for future CAD development.

[24] Shai Rotem, Ken Stevens, Ran Ginosar, Peter Beerel, Chris Myers, Kenneth Yun, Rakefet Kol, Charles Dike, Marly Roncken, and Boris Agapiev. RAPPID: An Asynchronous Instruction Length Decoder. *Proceedings International Symposium on Advanced Research in Asynchronous Circuits and Systems* (ASYNC), Barcelona, Spain, pages 60–70, 1999 (best paper award).

This paper describes an investigation of potential advantages and risks of applying an aggressive asynchronous design methodology to Intel Architectures. RAPPID (Revolving Asynchronous Pentium®Processor Instruction Decoder), a prototype IA32 instruction length decoding and steering unit, was implemented using self-timed techniques. The RAPPID chip was fabricated on a 0.25μ CMOS process and tested successfully. Results show significant advantages—in particular, performance of 2.5-4.5 instructions/nS—with manageable risks using this technology. RAPPID achieves three times the throughput and half the latency, dissipating only half the power and requiring about the same area as an existing 400MHz clocked circuit.

[25] Marly Roncken. Defect-Oriented Testability for Asynchronous ICs. *Proceedings of the IEEE*, Vol. 87, No. 2, February 1999, pages 363–375 (invited paper).

> For a CMOS manufacturing process, asynchronous ICs are similar to synchronous ICs. The defect density distributions are similar, and hence, so are the fault models and fault detection methods. So, what makes us think that asynchronous circuits are much harder to test than synchronous circuits? Because the effectiveness of best known test methods for synchronous circuits drops when applied to asynchronous circuits? That may very well be a temporal hurdle. Many test methods have already been re-evaluated and successfully adapted from the synchronous to the asynchronous test domain. The paper addresses one of the final hurdles: $\mathbf{I_{DDQ}}$ testing. This type of test method, based on measuring the quiescent power supply current, is very effective for detecting (resistive) bridging faults in CMOS circuits. Detection of bridging faults is crucial, because they model the majority of today's manufacturing defects. $\mathbf{I_{DDQ}}$ fault effects are sensitized in a particular state or set of states, and can only be detected if we stop the circuit operation right there. This is a problem for asynchronous circuits, because their operation is self-timed.
>
> In the paper, we quantify the impact of self-timing on the effectiveness of $\mathbf{I_{DDQ}}$ based test methods for bridging faults, and propose a Design-for-Test (DfT) approach to develop a low-cost DfT solution. For comparison, we do the same for logic voltage testing and stuck-at faults. The approach is illustrated on circuits from Tangram, the asynchronous design-style employed at Philips Research, but is applicable to asynchronous circuits in general.

[26] Kees van Berkel, Hans van Gageldonk, Joep Kessels, Cees Niessen, Ad Peeters, Marly Roncken, and Rik van de Wiel. Asynchronous Does not *Imply* Low Power, But ... *Anantha Chandrakasan and Robert Brodersen (Eds.): Low Power CMOS Design*, pages 227–232, Wiley-IEEE Press, February 1998.

> Asynchronous circuits are often praised for their low-power properties. Although they definitely have a low-power potential it is generally non-trivial to achieve this. We identify five low-power opportunities of asynchronous VLSI circuits and illustrate how these can be realized by a DCC error corrector and standby circuits for pagers.

[27] Carel Dijkmans, Tonny Duisters, Cees Niessen, Marly Roncken, Frits Schalij, and Eric van der Zwan. Low Power in Mobile Applications. *Philips Research Bulletin on IC Design*, No. 29, pages 1–3, April 1997.

> Our exercises in low-power solutions for DECT baseband functions have shown that ample opportunities are available to reduce energy use in VLSI. Unconventional approaches are necessary and approaches such as hard-wired logic can be far more energy-efficient than their software counterparts. By using appropriate design tools we have shown that one can retain design efficiency and allow flexibility to changes in design requirements.

[28] Peter Janssen and Marly Roncken. DECT-ADPCM: Test Plan and Results. *NAT.LAB. Technical Note 050/97*, Philips Research, 1997.

> In the Nat.Lab. Low-Power Project the very low power IC "ADPCM" was designed using the Tangram (DICY) design flow. Power simulations indicated that the consumption of this asynchronous design should be a factor 100 less than the existing synchronous design. Besides scan DfT, this asynchronous design contains dedicated HOLD facilities to stop the circuit in the relevant states for Voltage or IDDQ measurements so as to guarantee adequate test quality. The layout was generated automatically with the standard Cadence layout tools. The chips were processed in C100DM on a multi-project wafer and 26 packaged samples were delivered. In the following chapters the test conditions and the test results are described.

[29] Marly Roncken and Eric Bruls. Test Quality of Asynchronous Circuits: A Defect-oriented Evaluation. *Proceedings International Test Conference* (ITC), Washington, D.C., USA, pages 205–214, 1996.

This paper investigates the test quality of asynchronous circuits using fault models that are grounded in realistic defect probabilities. As for synchronous designs, I_{DDQ} testing plays a prominent role in detecting CMOS manufacturing defects for asynchronous designs, too. However, for asynchronous circuits, I_{DDQ} testing is usually less effective because fewer states are quiescent, and our analysis shows that the test quality can only be improved by creating more quiescent states. We present a new Design-for-Test (DfT) method that provides good test quality in that all defects are detected that are likely to occur given the IC layout and process technology and that pose quality or reliability problems. Our DfT method is evaluated on three in-house manufactured designs.

[30] Marly Roncken, Emile Aarts, and Wim Verhaegh. Optimal Scan for Pipelined Testing: An Asynchronous Foundation. *Proceedings International Test Conference* (ITC), Washington, D.C., USA, pages 215–224, 1996.

This paper addresses the problem of constructing a scan chain such that (1) the area overhead is minimal for latch-based designs, and (2) the number of pipeline scan shifts is minimal. We present an efficient heuristic algorithm to construct near-optimal scan chains. On the theoretical side, we show that part (1) of the problem can be solved in polynomial time, and that part (2) is NP-hard, thus precisely pinpointing the source of complexity and justifying our heuristic approach. Experimental results on three industrial asynchronous IC designs show (1) less than 0.1% extra scan latches for Level-Sensitive Scan Design, and (2) scan shift reductions up to 86% over traditional scan schemes.

[31] Marly Roncken. Asynchronous Design: Working the Fast Lane. Position statement for ITC Panel Session "Asynchronous Design: Nightmare or Opportunity," organized together with Eric Bruls. *Proceedings International Test Conference* (ITC), Washington, D.C., USA, page 939, 1996.

[32] Marly Roncken. Testing Asynchronous Chips. *Israel Workshop on Asynchronous VLSI*, Kibutz Genossar, sponsored and organized by the Ministry of Science and the Arts and the VLSI Systems Research Center at Technion, Haifa, Israel, Handout of Presentation Slides, pages 336–339, 19-22 March 1995.

Overview of challenges encountered during structural and application mode testing of asynchronous ICs on a synchronous HP82000 tester. All ICs have been designed with the Tangram silicon compiler, starting from the test chip in 1992 to the single-rail DCC error corrector in 1995. Short-term as well as long-term solutions are presented.

[33] Kees van Berkel, Ronan Burgess, Joep Kessels, Ad Peeters, Marly Roncken, Frits Schalij, and Rik van de Wiel. A Single-Rail Re-implementation of a DCC Error Detector Using a Generic Standard-Cell Library. *Proceedings Second Working Conference on Asynchronous Design Methodologies*, London, UK, pages 72–79, IEEE Computer Society Press, 1995.

We present a fully asynchronous implementation of a DCC Error Detector. The circuit uses 4-phase handshake signaling and single-rail data encoding, and has been realized using standard cells from a generic cell library. The circuit is obtained by fully automatic translation from a high-level (Tangram) description, using handshake circuits as intermediate architecture. In comparison with a previous double-rail implementation the fabricated IC is 40% smaller (core area), three times faster, and consumes only a quarter of the power. Switching between two power supplies is described as a technique to reduce power even further. The comparison evaluation also includes an improved double-rail implementation and two synchronous circuits.

[34] Marly Roncken. Partial Scan Test for Asynchronous Circuits Illustrated on a DCC Error Corrector, *Proceedings International Symposium on Advanced Research in Asynchronous Circuits and Systems* (ASYNC), Salt Lake City, Utah, USA, pages 247–256, 1994 (best paper award).

We present a design-for-testability method for asynchronous circuits based on partial scan. More specificly, we investigate how the partial scan principles from the synchronous test world can be adapted for asynchronous circuits, and we show that asynchronous partial scan design can be approached as a high-level design activity. The method is demonstrated on an asynchronous error

corrector for the DCC player. It has been used effectively in the production and application-mode tests of this 155k transistor chip-set. In particular, it has led to the high 99.9% stuck-at output fault coverage in short 64 msec test time at the expense of less than 3% additional area.

[35] Marly Roncken. Partial Scan Test for Asynchronous Circuits. *ACiD-WG Workshop on Testing and Design for Testability*, organized by INESC-Aveiro, Aveiro, Portugal, 1994.

[36] Kees van Berkel, Ronan Burgess, Joep Kessels, Ad Peeters, Marly Roncken, and Frits Schalij. A Fully Asynchronous Low-Power Error Corrector for the DCC Player. *IEEE Journal of Solid-State Circuits*, Vol.29, No.12, pages 1429–1439, 1994.

A fully asynchronous implementation of a complete DCC error corrector is presented that consumes 10 mW at 5 V, only a fifth of its synchronous counterpart. This is achieved by eliminating clocks, and exploiting the additional freedom in architecture provided by the absence of a clock. The corrector has been integrated in an experimental player and is both functionally and audibly correct. Handshake circuits are proposed as an architecture that enables structured design of asynchronous circuits through a consistent application of handshake signaling at all design levels. Handshake circuits are compiled fully automatically from high-level descriptions, and are implemented quasi delay insensitively using 4-phase handshake signaling and double-rail data encoding. The resulting circuits are self-initializable and testable.

[37] Kees van Berkel, Ronan Burgess, Joep Kessels, Ad Peeters, Marly Roncken, and Frits Schalij. A Fully Asynchronous Low-Power Error Corrector for the DCC Player. *IEEE International Solid-State Circuits Conference* (ISSCC), San Francisco, California, USA, Digest of Technical Papers, pages 88–89, 1994.

The promise of chip-wide asynchronous operation is its potential for very low power consumption. This potential is demonstrated by an error-corrector based on digital compact cassette (DCC) specifications, dissipating 80% less than its synchronous counterpart. A reduction in power consumption means longer battery lifetime, important in portable products such as cellular radio and personal audio.

[38] Kees van Berkel, Ronan Burgess, Joep Kessels, Ad Peeters, Marly Roncken, and Frits Schalij. Asynchronous Circuits for Low Power: a DCC Error Corrector. *IEEE Design & Test of Computers*, Vol. 11, No. 2, pages 22–32, Summer 1994.

The authors describe a complete low-power digital compact cassette error corrector. Using Tangram, a high-level programming language, they designed two asynchronous circuits that correct errors on DCC specifications.

[39] Kees van Berkel, Ronan Burgess, Joep Kessels, Ad Peeters, Marly Roncken, and Frits Schalij. A Low-Power Error Corrector for DCC Players. *Philips Research Bulletin on IC Design*, December 1993.

[40] Kees van Berkel, Ronan Burgess, Joep Kessels, Ad Peeters, Marly Roncken, and Frits Schalij. Characterization and Evaluation of a Compiled Asynchronous IC. *Proceedings IFIP WG10.5 Working Conference on Asynchronous Design Methodologies*, Manchester, UK, pages 209–221, Elsevier Science Publishers, North-Holland, 1993.

This paper presents the characterization and evaluation of an asynchronous IC produced fully automatically by the Tangram compiler. The circuit comprises a 4-place buffer, a 100-counter, an incrementer, an adder, a comparator, and a multiplier. It appears well testable and it operates functionally correctly with a supply voltage as low as 1.6 volt. This value reduces to 1.2 volt if the pass transistors of a (novel) acknowledge circuit are set aside.

[41] Marly Roncken and Ronald Saeijs. Linear Test Times for Delay-Insensitive Circuits: a Compilation Strategy. *Proceedings IFIP WG10.5 Working Conference on Asynchronous Design Methodologies*, Manchester, UK, pages 13–27, Elsevier Science Publishers, North-Holland, 1993.

This paper considers the problem of testing delay-insensitive circuits for production defects. The authors present a test strategy than can be integrated with the design of VLSI circuits through silicon compilation. The strategy is based on a very simple test procedure for which circuits are enhanced with a special mode of operation. The resulting test times are linear in the size of the circuit.

[42] Marly Roncken and Jos van Beers. Validation of the DICY Silicon Compiler through Tested and Characterized Silicon. *NAT.LAB. TN309/92*, Philips Research, 1992.

In 1987 the DICY silicon compilation method has been validated through a hand-compiled chip "ZaP." Since then, the method is implemented, supporting design tools were built and a strategy for testability enhancement and test generation was developed. Spring 1992 the validity of the compiler implementation, the CAD tools, and the test strategy were confirmed through a special test and measurement chip. This is an account of the latter test and measurement results.

[43] Kees van Berkel, Marly Roncken, and Ronald Saeijs. Method for testing an integrated circuitry and an integrated circuit having a plurality of functional components and having junction/switch test components in interconnecting channels between functional components. *Patented as:*

Patent EP556894A1, NV Philips' Gloeilampen Fabrieken, issued August 25, 1993.
Patent US5590275, U.S. Philips Corporation, issued December 31, 1996.

[44] Kees van Berkel, Ronan Burgess, Joep Kessels, Marly Roncken, and Frits Schalij. An Error Decoder for the Compact Disc Player as an Example of VLSI Programming. *Proceedings of the IEEE European Design Automation Conference* (EDAC), Brussels, Belgium, pages 69–74, 1992.

Using a programming language for VLSI design, called Tangram, we design a fast and simple VLSI circuit for error decoding in the Compact Disc player. The derivation of the design is straightforward and the result is succinctly expressed in less than one page of Tangram text. All design decisions are based merely on algorithmic and architectural considerations. No particular VLSI knowledge is needed and, therefore, the exercise demonstrates that Tangram allows system designers to design VLSI circuits. The exercise also shows that in a VLSI programming language special language constructs are essential to obtain efficient designs.

[45] Kees van Berkel, Joep Kessels, Marly Roncken, Ronald Saeijs, and Frits Schalij. The VLSI-programming Language Tangram and its Translation into Handshake Circuits. *Proceedings of the IEEE European Design Automation Conference* (EDAC), Amsterdam, The Netherlands, pages 384–389, 1991.

In this paper we consider VLSI design as a programming activity. VLSI designs are described in the algorithmic programming language Tangram. The paper gives an overview of Tangram, providing sufficient detail to invite the reader to try a small VLSI program himself. Tangram programs can be translated into handshake circuits, networks of elementary components that interact by handshake signaling. We have constructed a silicon compiler that automates this translation and converts these handshake circuits into asynchronous circuits and subsequently into VLSI layouts.

[46] Marly Roncken and Rob Gerth. A Denotational Semantics for Synchronous and Asynchronous Behaviour with Multiform Time. *Proceedings International BCS-FACS Workshop on Semantics for Concurrency*, University of Leicester, UK, pages 21–38, Springer Verlag, 1990.

We develop a denotational semantics for a concurrent language that allows both synchronous and various degrees of asynchronous behavior. Here, synchrony means that parallel components execute within the same timing regime — e.g., because they share the same clock — whereas asynchrony means that the timing regimes of the components are independent — e.g., because their clocks are unrelated. The semantics allows the timing of components. But time is multiform in the sense that different parallel components have different timing regimes. The paper serves as a study to appraise and solve some of the semantic problems that are posed by the task of describing the timed behavior of digital systems. In the course of this study we define a semantics for a language related to a VLSI description language developed at Philips Research Laboratories.

[47] Ton Kalker, Thijs Krol, Cees Niessen, Marly Roncken, Frits Schalij, Cees van Trigt, and Maarten Vliegenthart. Specification of the SCN2681. *NAT.LAB TN287/86*, Philips Research, 1986.

In technical note TN 286/86 we describe our experiences with several specification techniques. In that report the actual results of trying to describe the UART SCN2681 were not included for practical reasons: the final result would contains so many pages that reading would become cumbersome. This technical note contains the above mentioned descriptions. For completeness sake the original datasheets by Signetics and an informal description in English of SCN2681 are included.

[48] Ton Kalker, Thijs Krol, Cees Niessen, Marly Roncken, Frits Schalij, Cees van Trigt, and Maarten Vliegenthart. VLSI Specification Methods. *NAT.LAB TN286/86*, Philips Research, 1986.

This technical note contains the evaluation of the following VLSI specification techniques: Finite State Machines, Statemate, Extended Register Transfer Languages, Petri Nets, Linear Temporal Logic. This evaluation is based on our experiences in applying these techniques to the UART SCN2681 of Signetics.

[49] Rob Gerth, Willem-Paul de Roever, and Marly Roncken. A Study in Distributed Systems and Dutch Patriotism. *Proceedings 2nd Conference on the Foundations of Software Technology and Theoretical Computer Science* (FST&TCS), Bangalore, India, 1982.

Also available as:
Report RUU-CS-82-10, Department of Computer Science, University of Utrecht, 1982.

[50] Rob Gerth, Willem-Paul de Roever, and Marly Roncken. Procedures and Concurrency: A Study in Proof. *Proceedings Vth International Symposium on Programming* (ISOP), Turin, Italy, LNCS 137, pages 132–163, Springer Verlag, 1982.

[51] Marly Roncken, Rob Gerth and Willem-Paul de Roever. A Proof System for Brinch Hansen's Distributed Processes. *Proceedings 3rd Conference of the European Cooperation in Informatics* (ECI), Munich, Germany, LNCS 50, pages 88–96, Springer Verlag, 1981.

Also available as:
Report RUU-CS-81-12, Department of Computer Science, University of Utrecht, 1981.

[52] Marly Roncken, Niek van Diepen, Mark Kramer, and Willem-Paul de Roever. A Proof System for Brinch Hansen's Distributed Processes. *Report RUU-CS-81-5*, Department of Computer Science, University of Utrecht, 1981.