Towards Self-stabilizing Byzantine Fault-Tolerant Clock Generation in Systems-on-Chip

Danny Dolev*, Matthias Függer†, Christoph Lenzen‡, and Ulrich Schmid†
* Hebrew University of Jerusalem, Jerusalem, Israel
Email: dolev@cs.huji.ac.il
† Vienna University of Technology, Vienna, Austria
Email: {fuegger,s}@ecs.tuwien.ac.at
‡ Weizmann Institute of Science, Rehovot, Israel
Email: clenzen@cs.huji.ac.il
Phone: 00972 2 658 5770

I. Introduction

The most obvious challenges resulting from the tremendous advances of *Very Large Scale Integration* (VLSI) technology, which has now reached the nanometer scale, are design complexity and robustness issues. Modern *Systems-on-Chip* (SoC) and *Networks-on-Chip* (NoC) accommodate Billions of transistors on a single die nowadays, and have in fact much in common with loosely-coupled fault-tolerant distributed systems: *Globally Asynchronous Locally Synchronous* (GALS) [3] is already the dominating design paradigm, and failures of components due to process variations and operating conditions (temperature, voltage) are no longer rare events. The resulting challenges for manufacturing technology [14] and circuit architecture [19] are well-known, yet become more pronounced with every new technology generation.

Another severe threat for the dependability of modern integrated circuits are dramatically increasing *transient failure* rates [4], resulting from ionized particles [1], [6], [11], [20], cross-talk and ground bouncing [16], [17]. Formerly, ionized particles caused problems only in aerospace applications [11], where high-energy particles are abundant due to cosmic rays interacting with the atmosphere. Nowadays, nanometer feature sizes combined with reduced voltage swings needed for high clock speeds and low power consumption have also raised the error rate of chips operated at sea-level beyond acceptable limits [1], [6] — despite considerable improvements of VLSI process technology. The same is true for errors caused by crosstalk and ground bouncing.

Although classic fault-tolerance techniques like *Dual Modular Redundancy* (DMR) and *Triple Modular Redundancy* (TMR) also work perfectly well for transient failures, they obviously cannot withstand arbitrary failure rates. For example, TMR fails if more than one of the three input lanes suffer from a transient failure. In case of high failures rates, this is not an unlikely event.

II. Research Goals

Our goal is to utilize *self-stabilizing* distributed algorithms [10] for building extremely robust integrated circuits, in particular, for critical applications with substantial transient failure rates e.g. in the aerospace domain. A self-stabilizing algorithm is guaranteed to resume regular operation from *any* corrupted system state, i.e., even from a burst of transient failures hitting all components of a circuit at the same time. Although there is a substantial body of work that can be relied upon, substantial research is required to make self-stabilizing VLSI circuits a reality.

First of all, classic self-stabilizing algorithms require that no additional failures occur during the *stabilization period*, where the system recovers from past transient failures. In VLSI circuits, however, we cannot rule out the possibility of also having a certain fraction of permanently faulty components: Process variations and bad operating conditions easily cause components to fail in an arbitrary (Byzantine) [18] way. For example, an off-spec output signal may easily be interpreted inconsistently at the input of different receivers. The existing work on *Byzantine fault-tolerant self-stabilization* [2], [5], [13], [15], where a certain number of (unknown) components may permanently exhibit Byzantine failures (even during stabilization), proves that it is principally possible to cope with this problem.

Nevertheless, all existing results have been obtained in the distributed systems context. In order to achieve our goals, all aspects that are unique to VLSI circuits must be adquately addressed as well. First of all, it is important to observe that self-stabilization is a property that cannot be added at some intermediate level, that is, atop of non-self-stabilizing lower-level services: A complete state corruption could render the latter in a possibly irrecoverable state, which does not allow the intermediate level algorithm to perform any useful action. Consequently, *all* basic services (clock generation, communication scheduling, etc.) must be implemented in a Byzantine fault-tolerant self-stabilizing manner in order to guarantee this property at the application level.

In addition, implementing any Byzantine fault-tolerant self-stabilizing algorithm in a VLSI circuit is particularly challenging for a number of reasons [12]:

- Computing model: The (usually relatively few) processes in a classic distributed system are modeled as transition systems, which map basic operations to zero-time state

transitions. By contrast, in VLSI circuits, Millions of basic logic gates *continuously* compute their outputs based on past inputs. Among the consequences is the potential of metastability, which has never been considered in distributed systems research.

- Processing constraints: Classic distributed algorithms are based on quite powerful basic operations, including numeric processing of arbitrary integer values. By contrast, basic gates in VLSI circuits have very limited processing capabilities, typically restricted to Boolean functions.
- Communication constraints: Classic fault-tolerant distributed algorithms usually assume fully-connected networks for exchanging large messages. By contrast, fully-connected wiring topologies are the exception in VLSI circuits, even in case of serial communication, and large messages are out of question.
- Failure models: Classic Byzantine failure models allow an arbitrary subset of f among n processes in the distributed system to behave Byzantine faulty, but require $n \geq 3f+1$ [9], [18] and an essentially fully connected network [7]. Since the latter cannot be assumed in typical VLSI circuits, less demanding Byzantine failure models are required.

Of course, adequately addressing all these issues from scratch would be too ambitious. Fortunately, we can build on the powerful framework for modeling and analysis of fault-tolerant asynchronous circuits developed recently [12]. In particular, in a recent paper [8], we demonstrated that a variant of the FATAL framework is also applicable to a Byzantine fault-tolerant self-stabilizing algorithm for clock generation in SoCs.

Thanks to the existing results, our current research concentrates on the following major goals:

(1) Extending the present modeling and analysis framework to support self-stabilization: (a) Generalize specifications to also cover self-stabilizing components. This primarily requires to replace correct components with such that may behave arbitrarily initially and only "gradually" become correct. (b) Provide an all-digital model for metastability generation and propagation analysis.

(2) Devising and implementing practical Byzantine fault-tolerant self-stabilizing algorithms for basic services in SoCs: (a) Devise self-stabilizing algorithms supporting realistic Byzantine failure models for sparsely connected circuit topologies. (b) Devise correctness proofs and performance analyses. (c) Implement the algorithms in VHDL and build a research prototype, which allows experimental evaluation.

REFERENCES

[1] R. Baumann. Radiation-induced soft errors in advanced semiconductor technologies. *IEEE Transactions on Device and Materials Reliability*, 5(3):305–316, Sept. 2005.

[2] M. Ben-Or, D. Dolev, and E. N. Hoch. Fast self-stabilizing byzantine tolerant digital clock synchronization. In *Proc. 27th symposium on Principles of Distributed Computing (PODC)*, pages 385–394, 2008.

[3] D. M. Chapiro. *Globally-Asynchronous Locally-Synchronous Systems*. PhD thesis, Stanford University, Oct. 1984.

[4] C. Constantinescu. Trends and challenges in VLSI circuit reliability. *IEEE Micro*, 23(4):14–19, July 2003.

[5] A. Daliot, D. Dolev, and H. Parnas. Self-stabilizing pulse synchronization inspired by biological pacemaker networks. In *Proceedings of the 6th International Symposium on Self-Stabilizing Systems, SSS'03*, volume 2704 of *LNCS*, pages 32–48, San Francisco, CA, USA, June 2003. Springer Verlag.

[6] A. Dixit and A. Wood. The impact of new technology on soft error rates. In *Proc. IEEE International Reliability Physics Symposium (IRPS'11)*, pages 5B.4.1 –5B.4.7, april 2011.

[7] D. Dolev. The Byzantine generals strike again. *Journal of Algorithms*, 3(1):14–30, 1982.

[8] D. Dolev, M. Függer, C. Lenzen, and U. Schmid. Fault-tolerant algorithms for tick-generation in asynchronous logic: Robust pulse generation - [extended abstract]. In *Proceedings 13th International Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS'11), Springer LNCS 6976*, pages 163–177, 2011.

[9] D. Dolev, J. Y. Halpern, and H. R. Strong. On the possibility and impossibility of achieving clock synchronization. *Journal of Computer and System Sciences*, 32:230–250, 1986.

[10] S. Dolev. *Self-Stabilization*. MIT Press, 2000.

[11] C. Dyer and D. Rodgers. Effects on spacecraft & aircraft electronics. In *Proceedings ESA Workshop on Space Weather*, ESA WPP-155, pages 17–27, Nordwijk, The Netherlands, nov 1998. ESA.

[12] M. Függer and U. Schmid. Reconciling fault-tolerant distributed computing and systems-on-chip. *Distributed Computing*, 24(6):323–355, 2012.

[13] E. Hoch, D. Dolev, and A. Daliot. Self-stabilizing Byzantine digital clock synchronization. In *Proceedings of the Eighth International Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS 2006)*, volume 4280 of *LNCS*, pages 350–362, Dallas, TX, USA, Nov. 2006. Springer Verlag.

[14] I. Koren and Z. Koren. Defect tolerance in VLSI circuits: Techniques and yield analysis. *Proceedings of the IEEE*, 86(9):1819–1838, Sep 1998.

[15] M. Malekpour. A byzantine-fault tolerant self-stabilizing protocol for distributed clock synchronization systems. In A. Datta and M. Gradinariu, editors, *Stabilization, Safety, and Security of Distributed Systems*, volume 4280 of *Lecture Notes in Computer Science*, pages 411–427. Springer Berlin / Heidelberg, 2006. 10.1007/978-3-540-49823-0_29.

[16] M. S. Maza and M. L. Aranda. Analysis of clock distribution networks in the presence of crosstalk and groundbounce. In *Proceedings International IEEE Conference on Electronics, Circuits, and Systems (ICECS)*, pages 773–776, 2001.

[17] A. K. Palit, V. Meyer, W. Anheier, and J. Schloeffel. Modeling and analysis of crosstalk coupling effect on the victim interconnect using the ABCD network model. In *Proceedings of the 19th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT'04)*, pages 174–182, Oct. 2004.

[18] M. Pease, R. Shostak, and L. Lamport. Reaching agreement in the presence of faults. *Journal of the ACM*, 27(2):228–234, April 1980.

[19] M. Peercy and P. Banerjee. Fault tolerant VLSI systems. *Proceedings of the IEEE*, 81(5):745–758, May 1993.

[20] P. Shivakumar, M. Kistler, S. Keckler, D. Burger, and L. Alvisi. Modeling the effect of technology trends on the soft error rate of combinational logic. *Proceedings International Conference on Dependable Systems and Networks(DSN'02)*, pages 389–398, 2002.

SHORT BIOGRAPHY OF CHRISTOPH LENZEN

Christoph Lenzen received a diploma degree in Mathematics from the University of Bonn, Germany, and subsequently performed his graduate studies in Distributed Computing in the group of Professor Roger Wattenhofer at ETH Zurich. In 2011, he was a postdoctoral Fellow at the Hebrew University of Jerusalem, with Danny Dolev. Currently, he is a postdoctoral fellow at the Weizmann Institue of Science, with Professor David Peleg. In 2013 and 2014 he will be a postdoctoral fellow at MIT, in the group of Professor Nancy Lynch.

His research interests cover distributed computing in a wider sense, including topics such as randomized load balancing, graph algorithms, clock synchronization. Recently, fault-tolerant algorithms for hardware clock synchronization have been a focus of his interest. He published e.g. at PODC, SPAA, FOCS, and STOC, and in JACM. He has been invited to give talks at ETH Zurich (Switzerland), Tel-Aviv University (Israel), Fraunhofer AISEC (Munich, Germany), IHP Microelectronics (Frankfurt at the Oder, Germany), Ben-Gurion University of the Negev (Beer-Sheva, Israel), Weizmann Institute of Science (Rehovot, Israel), Institute of Science and Technology Austria (Vienna, Austria), Vienna University of Technology (Austria), University of Lugano (Switzerland), and University of Bonn (Germany). In 2009, he and his coauthors received the PODC best paper award for their work on gradient clock synchronization. His Ph.D. thesis was awarded the ETH medal, which is conferred to at most 8% of the theses at ETH.