Dagstuhl Seminar 08371 on

# Fault-Tolerant Distributed Algorithms on VLSI Chips

Bernadette Charron-Bost, Shlomi Dolev, Jo Ebergen, Ulrich Schmid

CW 37, 2007

**Abstract**

The Dagstuhl seminar 08371 on *Fault-Tolerant Distributed Algorithms on VLSI Chips* was devoted to exploring whether the wealth of existing fault-tolerant distributed algorithms research can be utilized for meeting the challenges of future-generation VLSI chips. Participants from both the distributed fault-tolerant algorithms community, interested in this emerging application domain, and from the VLSI systems-on-chip and digital design community, interested in well-founded system-level approaches to fault-tolerance, surveyed the current state-of-the-art and tried to identify possibilities to work together. The seminar clearly achieved its purpose: It became apparent that most existing research in Distributed Algorithms is too heavy-weight for being immediately applied in the "core" VLSI design context, where power, area etc. are scarce resources. At the same time, however, it was recognized that emerging trends like large multicore chips and increasingly critical applications create new and promising application domains for fault-tolerant distributed algorithms. We are convinced that the very fruitful cross-community interactions that took place during the Dagstuhl seminar will contribute to new research activities in those areas.

**General Topics:** *Data structures/algorithms/complexity, hardware, networks.*
**Keywords:** *Fault-tolerant distributed algorithms, fault tolerance, VLSI systems-on-chip, synchronous vs. asynchronous circuits, digital logic, specifications.*

## 1 Description

Shrinking feature sizes and increasing clock speeds are the most visible signs of the tremendous advances in VLSI design, which will accommodate billions of transistors on a single chip in the near future [11]. This comes at the price of increased system-level complexity and vulnerability, however: In today's deep submicron technology with GHz clock speeds, wiring delays dominate transistor switching delays, and signals cannot traverse the whole die within a single clock cycle any more. In fact, a modern VLSI chip can no longer be viewed as a monolithic block of synchronous hardware, where all state transitions occur simultaneously [17]. Rather, VLSI chips are nowadays considered as systems of interacting subsystems — the advent of Systems-on-Chip (SoC) and Networks-on-Chip (NoC).

In addition, ever-increasing manufacturing variabilities increase the defect ratio, and the reduced voltage swing needed for high clock speeds and low power consumption also increases the adverse effects of $\alpha$-particle and neutron hits during operation [1, 15], as well as cross-talk and ground-bouncing sensitivity [13, 16]. The resulting increase of the



Dagstuhl Seminar Proceedings 08371
Fault-Tolerant Distributed Algorithms on VLSI Chips
http://drops.dagstuhl.de/opus/volltexte/2009/1927

transient failure rate (soft-error rate) [12], which was negligible in most former-generation chips, has hence raised general concerns about the dependability of future generation VLSI chips [3]. Consequently, suitable fault-tolerance mechanisms with respect to timing errors or value errors are vital for such devices [2, 14]: Fine-grained fault-tolerance like radiation-hardening, fault masking at transistor or gate level, error-correcting codes or error detection and recovery are the primary methods of choice here.

Due to the above trends, however, modern VLSI chips have much in common with the loosely-coupled distributed systems that have been studied by the fault-tolerant distributed algorithms community for decades. System-level fault tolerance based on replication and distributed agreement is the dominant approach here, and a wealth of different computing and failure models, algorithms & protocols, and theoretical results regarding solvability of problems and achievable performance have been established in the past.

The purpose of our Dagstuhl seminar was to explore whether fault-tolerant distributed algorithms research can indeed be utilized for meeting the challenges of future-generation VLSI chips: Just as Temporal Logic, established in the distributed computing scope decades ago, found its way to the VLSI domain, other radically new solutions and methods may also find their way. And indeed, some recent research suggested a positive answer to this question: For example, [10] demonstrated that distributed fault-tolerant clock generation algorithms can be adapted to the very special requirements of VLSI chips, and [5,6] demonstrated that self-stabilization [4] is a very promising approach for designing robust VLSI chips.

Fifteen participants from the distributed fault-tolerant algorithms community (and related fields, like verification), interested in the new application domain of VLSI chips, and twelve participants from the VLSI community, interested in system-level approaches to fault-tolerance, joined at Dagstuhl in order to survey the current state-of-the-art and identify possibilities to work together.

The cornerstones of the seminar program were:

- An intro session, where every participant briefly introduced herself/himself and her/his expectations.
- A tutorial on *Introduction to VLSI* by Jo Ebergen, with the intention to introduce the Distributed Algorithms (DA) community to the VLSI field.
- A tutorial on *Fault-Tolerant Distributed Algorithms* by Bernadette Charron-Bost, to introduce the VLSI community to DAs.
- Some invited presentations on selected topics:
  - *Distributed Algorithms and VLSI - An Appetizer* (U. Schmid),
  - *Synchrony and Asynchrony in VLSI* (J. Ebergen),
  - *Arbiter-free Synchronization* (L. Lamport),
  - *Self-Stabilization Copes with Soft Errors* (S. Dolev)
  - *Error and Fault Tolerance in VLSI* (L. Navinier),
  - *Fault Tolerance in Reconfigurable Fabrics* (R. Manohar)
- Invididual presentations of the participants.

- A wrap-up session, where every particiant briefly summarized her/his lessons learned.

The presentations and the unique setting of Dagstuhl, with its relaxed and stimulating atmosphere, fully achieved their purpose: Long discussions during the official seminar, and many fruitful cross-community interactions during the free times were stimulated, which even exceeded the amount of available time.

**Summary of the seminar results:**

1. Most existing research in Distributed Algorithms is too, and unnecessarily, heavy-weight for being immediately applied in the current VLSI digital design context, where power, area etc. are scarce resources. There are exceptions, though, as the research on fault-tolerant clocking and, in particular, self-stabilization reveals: Self-stabilizing VLSI is very attractive with respect to its unrivaled robustness, and seems to be fully compatible with the substantial body of existing distributed algorithms research.

   In addition, hardships are also challenges and opportunities: Distributed computing, which is primarily based on replication, could benefit from exploring ways to mimic error correcting schemes [7–9] and other VLSI fault-tolerance approaches.

2. Emerging trends like large multicore chips and increasingly critical applications also create new and promising application domains for fault-tolerant distributed algorithms: Multicores with 100+ cores match very well the traditional distributed computing abstractions of fault-tolerant distributed algorithms, and using this knowledge may open up radically new architectural solutions for building multicore chips. Moreover, the "light-weight" fault-tolerance approaches used in current VLSI may not be sufficient to handle more severe failures, like security threats; more heavy-weight distributed algorithms solutions may hence become viable alternatives here.

3. A big challenge is the development of a comprehensive "Theory of Dependable VLSI", which does not exist. Developing such a theory would require, for example, identifying and incorporating adequate fault-tolerance properties in VLSI modeling approaches, explicit handling of metastability issues, dealing with circuit/specification composition & decomposition and hierarchical proofs, providing support for realiability analysis, etc.

   Fault-tolerant distributed algorithms research could serve as a starting point for such a theory, although there is still a long way to go: Firstly, in distributed computing, the interactions between processors and their *environment* are supposed to be very simple. Indeed, systems are basically closed in the sense that even if we consider systems with input and output events, there is no feedback loop involving the environment. Conversely, a circuit is only defined with respect to its environment, and there is some "mirror symmetry" between both. This major difference leads to a quite more subtle modeling approach than in classic distributed computing, which considers processors and environment seperately. Secondly, existing distributed computing models are based on the abstraction of discrete and atomic computing steps performed by some reasonably small number of processors. The massive concurrency of continuously "computing" logic gates cannot adequately be

modeled with such an abstraction, however. Similarly, operations like sending a message, which are considered "cheap" in ordinary distributed systems, are often prohibitively costly (in terms of resource requirements and/or time complexity) in VLSI chips.

We are convinced that our seminar will contribute to the further development in both fields.

# References


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

[2] S. Borkar. Designing reliable systems from unreliable components: the challenges of transistor variability and degradation. *IEEE Micro*, 25(6):10–16, Nov. 2005.

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

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

[5] S. Dolev and Y. Haviv. Self-stabilizing microprocessors, analyzing and overcoming soft-errors. *IEEE Transactions on Computers*, 55(4):385–399, Apr. 2006.

[6] S. Dolev and Y. Haviv. Stabilization enabling technology. In *Proceedings of the Eighth International Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS 2006)*, LNCS. Springer Verlag, 2006. (to appear).

[7] S. Dolev, L. Lahiani, and M. Yung. Secret swarm unitreactive k-secret sharing. In *Proceedings INDOCRYPT*, pages 123–137, 2007.

[8] S. Dolev and N. Tzachar. Brief announcment: Corruption resilient fountain codes. In *DISC*, pages 502–503, 2008.

[9] R. Friedman, A. Mostefaoui, S. Rajsbaum, and M. Raynal. Asynchronous agreement and its relation with error-correcting codes. *IEEE Trans. Comput.*, 56(7):865–875, 2007.

[10] M. Fuegger, U. Schmid, G. Fuchs, and G. Kempf. Fault-Tolerant Distributed Clock Generation in VLSI Systems-on-Chip. In *Proceedings of the Sixth European Dependable Computing Conference (EDCC-6)*, pages 87–96. IEEE Computer Society Press, Oct. 2006.

[11] International technology roadmap for semiconductors, 2005.

[12] T. Karnik, P. Hazucha, and J. Patel. Characterization of soft errors caused by single event upsets in CMOS processes. *Dependable and Secure Computing, IEEE Transactions on*, 1(2):128–143, April-June 2004.

[13] 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.

[14] M. Nicolaidis. Design for soft error mitigation. *Device and Materials Reliability, IEEE Transactions on*, 5(3):405–418, Sept. 2005.

[15] E. Normand. Single-event effects in avionics. *IEEE Transactions on Nuclear Science*, 43(2):461–474, Apr 1996.

[16] 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.

[17] I. E. Sutherland and J. Ebergen. Computers without Clocks. *Scientific American*, 287(2):62–69, Aug. 2002.