

# Programming Unreliable Networks of Computation

## *Extended Abstract*

Phillip Stanley-Marbell

This dissertation presents the idea of *macro-scale sensor electromechanical systems (MSEMS)* [6], and the necessary foundational tools — fault-tolerance techniques, system analysis and evaluation metrics, a programming model and language implementation, system modeling tools, and prototype hardware — to prove the feasibility of their realization.

MSEMS are a logical future direction in harnessing advances in the computation and sensing facilities of integrated circuits. By integrating such computation and sensing resources alongside novel mechanical actuators directly embedded into otherwise *inanimate* materials, it is possible to forge new directions at the boundaries of computing and materials science and engineering, enabling the creation of *smart* or *computationally-animate* materials, whose mechanical properties are partly under the control of software. Such systems, composed of large numbers of computation, sensing, and mechanical actuation elements, bear some similarities to established research areas such as sensor networks and robotics, but due to the need for unobtrusive integration within materials, raise several research challenges of their own. Using MSEMS as a motivating platform, this dissertation makes several fundamental contributions to the general problem of *building programmable systems out of unreliable networks of computational elements*. The contributions include:

- Tools:** A full-system microarchitectural evaluation environment for networked embedded systems in general, and MSEMS platforms in particular [5]. The environment enables detailed microarchitectural simulation of multiple instances of complete embedded systems, their peripherals, and medium access control / physical layer communication between systems. The simulation platform models microarchitecture, computation and communication failure events under a variety of stochastic distributions, compute and communication power consumption, electrochemical battery systems, and power regulation circuitry, as well as analog signals external to the processing elements.
- Analysis Metrics:** To develop appropriate architectures for failure-prone, resource-constrained systems, evaluation metrics are needed that permit different proposals to be evaluated with respect to criteria of computation performance, reliability, power consumption, the effect of power consumption variation on the lifetime of battery-powered systems, or combinations of these criteria. The dissertation introduces such metrics [1; 2].
- Programming Model and Language Implementation:** The dissertation introduces a model of computation for failure-prone, resource-constrained concurrent hardware, and its implementation in a small programming language, **M** [3]. The model enables partitioning without replication, of applications, across multiple devices with constrained memory resources. It enables programs to specify the value deviation tolerable in individual variables, as well as tolerable latencies and erasures on communications. These constraints, together with compile-time assump-

tions about the logic upset distributions existent in the deployment environment, are used as input to compile-time transformations to enable forward error correction. The language model and implementation include constructs to enable the re-direction of control-flow when the tolerance constraints are violated, due to, e.g., more severe logic upset phenomena than assumed at compile-time.

- Deviation-Tolerance, and its Analytic Basis:** Analyses for determining the probability distribution of deviations in numeric program values, due to the occurrence of logic upsets, is presented. The analytic relations derived are functions of the digital encoding of values ascribed different types in a programming language, the statistical probability distribution of logic upsets across the bits in hardware logic, and the statistical probability distribution of (error-free) values in real programs. These analyses are used to develop methods for encoding values in programs, such that, in the presence of a given worst-case distribution of incurred logic upsets, the magnitude of value deviation is kept within programmer-specified bounds.
- Empirical Analysis of Program Behavior:** A study of the empirical probability distribution of the error-free values taken on at runtime by source-level program variables of different data types, along with their bit-level probability distributions is presented. It is shown that, across data types, for the different application domains investigated, there is considerable correlation between the values taken on by individual bits, precluding the use of bit-level probabilities alone as a compact representation of the distributions. Based on properties observed across the suites of programs investigated, a compact representation for the distributions is presented.
- Concrete Hardware Implementation:** The dissertation presents a custom hardware implementation [4], to serve as a building block for computationally-animate materials. The hardware platform uses a novel combination of a PIN photodiode array, switching regulators, and a super-capacitor, to provide a small footprint renewable energy source. The design provides software-controlled power-adaptation facilities, for both the main processor and its peripherals. The system's power consumption is characterized, and its energy-scavenging efficiency is quantified with field measurements under a variety of weather conditions.

## REFERENCES

- [1] P. Stanley-Marbell and D. Marculescu. Dynamic Fault-Tolerance and Metrics for Battery Powered, Failure-Prone Systems. In *International Conference on Computer Aided Design, ICCAD 2003*, November 2003.
- [2] P. Stanley-Marbell and D. Marculescu. Local Decisions and Triggering Mechanisms for Adaptive Fault-Tolerance. In *Proceedings of the conference on Design, Automation and Test in Europe, DATE '04*, February 2004.
- [3] P. Stanley-Marbell and D. Marculescu. A Programming Model and Language Implementation for Concurrent Failure-Prone Hardware. In *Proceedings of the 2nd Workshop on Programming Models for Ubiquitous Parallelism, PMUP '06*, September 2006.
- [4] P. Stanley-Marbell and D. Marculescu. An  $0.9 \times 1.2''$ , low power, energy-harvesting system with custom multi-channel communication interface. In *DATE '07: Design Automation and Test in Europe*, pages 15 – 20, 2007.
- [5] P. Stanley-Marbell and D. Marculescu. Sunflower: Full-System, Embedded Microarchitecture Evaluation. *2nd European conference on High Performance Embedded Architectures and Computers (HiPEAC 2007) / Lecture Notes on Computer Science*, 4367:168–182, 2007.
- [6] P. Stanley-Marbell, D. Marculescu, R. Marculescu, and P. K. Khosla. *Low Power Electronics Design*, chapter Modeling Computational, Sensing and Actuation Surfaces, pages 33–1 – 33–14. CRC Press, Piscataway, NJ, USA, July 2004.