Introduction to Special Section on Probabilistic Embedded Computing

We are very pleased to guest edit this special section on Probabilistic Embedded Computing, which features articles on the latest developments in probabilistic hardware, software, and analysis. Recent silicon technology nodes have exhibited less predictable behavior mostly due to their extremely small transistor sizes and the associated impact in terms of variability of the important physical parameters. Similarly, software design and analysis technology has reached performance and scalability limits that can only be overcome by introducing probabilistic rather than idealized correctness conditions. A variety of papers sporadically distributed over a significant number of conferences raised the following question: Why not collect some of this interesting body of work on methods for computing where probability concepts play a central role? We are very pleased with the result as displayed in this issue of the *ACM Transactions on Embedded Computing* journal.

The first article by K. Palem and A. Lingamneni provides an overview of over a decade of research in probabilistic computing, including several key early results, such as the exponential relationship based on a thermodynamic physics model between the probability of error and energy. In other words, for small increases in error rate, exponentially larger savings in energy may be obtained. This invited article surveys much of the work by the authors and places their work on inexact and probabilistic computing in context. The article is easy to read and entertaining as well as technically quite novel at its core.

The second article by S. Misailovic, D. Kim, and M. Rinard presents a parallelizing compiler, which generates potentially nondeterministic parallel programs that produce acceptably accurate results acceptably often on multicore hardware. The freedom to generate parallel programs whose output may differ within statistical accuracy bounds from the output of the sequential program enables a dramatic simplification of the compiler, a dramatic increase in the range of applications that it can parallelize, and a significant expansion in the range of parallel programs that it can legally generate.

The third article by J. Sartori and R. Kumar explores trade-offs in the design of processor architectures when error is included in the evaluation process. The authors present interesting results showing that distinct architectures excel under the assumption of nonzero circuit error rates—together with architectural features to tolerate such errors, for example, through error detection and subsequent reexecution of the pipeline—than excel under the assumption of zero errors in the digital logic circuitry. A large variety of processor architectures are examined, from superscalar to DSP and to simple RISC designs, with the unique impact of error on the design space shown in detail.

The fourth article by Chippa et al. presents an approach to dynamically adjusting circuit parameters, such as voltage, in response to runtime estimates of performance quality. A variety of mechanisms at the architectural and circuit levels are explored and proposed for this dynamic effort scaling. The key idea is that, since the required effort is likely to vary over time as datasets change, so should the energy expended also change. The main result is that at a cost of some tiny amount of error, significant energy savings can be achieved while dynamically mapping to application requirements.

DOI: http://dx.doi.org/10.1145/2465787.2465788

The fifth article by Salajegheh et al. examines powering FLASH memory at a less than recommended voltage using software techniques to mitigate the number of errors. Assuming that usage in an application is able to tolerate some small level of errors but with very stringent energy limitations, a variety of software approaches to writing to FLASH memory are explored for energy-error trade-offs.

The sixth article by A. Alaghi and J. P. Hayes surveys an approach where unweighted streams of bits (i.e., all bits have equal weight, so there is no "most significant" or "least significant" bit to speak of) are used to represent numbers. In this proposed approach, dating back to the 1960s, the authors explain and survey various ways of computing as well as the relatively high tolerance of errors in several application areas which do not require a very high level of numeric precision.

The seventh article by Lingamneni et al. presents a new design approach to prune logic based on error impact, so that the resulting logic is nearly always correct but with a significantly reduced area and energy cost. The inexact logic leverages an application's inherent error tolerance. A variety of energy-error trade-offs are explored for pruned adders and multipliers.

The next article by Cazorla et al. presents a probabilistic timing analysis of next-generation real-time embedded systems. The analysis addresses the problem of acquiring accurate models of hardware execution latency and knowledge of program timing behavior in the presence of varying hardware conditions. Experimental evidence shows that the analysis produces probabilistically safe and tight WCET estimations with reduced knowledge of the execution platform.

The final article by Abbas et al. presents a Monte-Carlo optimization technique for finding inputs to a system that falsifies a given Metric Temporal Logic (MTL) property. The idea is to perform a random walk over the space of inputs guided by a robustness metric defined by the MTL property. The resulting framework can be applied to nonlinear hybrid systems with external inputs falsifying properties with more consistency as compared to other means such as uniform sampling.

In conclusion, we are very pleased with the large variety of novel work presented in this special section, which provides an important foundation for future research in probabilistic embedded computing!

Christoph Kirsch
University of Salzburg

Vincent Mooney
Georgia Institute of Technology
Guest Editors