It is with great pleasure that we introduce this special issue on fault-tolerant embedded systems to the readership of *IEEE Micro*.

In recent years, computer professionals have seen the emergence and widespread application of fault tolerance for embedded computing systems. Industrial control, avionics, Internet computing, and banking are among the most obvious application areas. Society heavily relies on these systems, making their operations subject to close scrutiny by those who depend on them.

Researchers in this field generally acknowledge that embedding fault tolerance in embedded systems requires a radical change in the overall design process. Practical solutions require a full understanding of the hardware and software domains, their relationships in an embedded environment, and system behavior under fault-free and faulty conditions. Designers usually employ a combination of hardware and software techniques to take advantage of complementary features, such as detection and correction. This special issue covers topics—such as architecture, verification, and testing—that make this approach explicit.

Reliability and availability are only two of the many attributes that fault-tolerant embedded systems must have. Users also demand dependability, safety, and security, and these elements are an integral part of embedded-system operation. Hardware and software must meet ever changing and stringent requirements to remain readily adaptable to a multitude of applications and environments.
**Conflicting objectives**

As systematic characterization reaches unsurpassed levels of complexity, designers must address many often-conflicting objectives. These conflicts start in the initial design stages and continue up until field delivery. The increasing focus on fault-tolerant design must also work within the usual performance metrics associated with today's computing. So although many techniques—such as redundancy and fault avoidance—worked well in the past, their application is now severely limited by their high performance penalties.

**Growing restrictions**

Designers must also contend with growing restrictions. For example, operational analyses of the most advanced fault-tolerant systems may only be possible under very restrictive conditions because most of these systems are very extensive and complex. Without full access to operational analyses, it's difficult to build effective fault-tolerant systems or share information about how they are built. This lack of access could eventually discourage large-scale commercialization and industrial adoption of fault-tolerant design. These high-security systems also have critical functions and long mission times, both of which often dictate tight user requirements. As such, these systems are subject to detailed investigations—often in dynamic environments—to ensure that they indeed achieve fault-tolerant operation. Providing the mechanisms for such detailed scrutiny is another restriction that designers of fault-tolerant systems must live with.

**This special issue**

This special issue consists of six articles that cover a wide spectrum of techniques encountered in the most advanced designs of fault-tolerant embedded systems. As with all special issues, these topics only represent activities that the technical community is currently pursuing. As the guest editors, we believe that this issue adequately represents both experimental and speculative topics with an appropriate balance between hardware and software.

In the first article, “A Self-Repairing Execution Unit for Microprogrammed Processors,” Benso, Chiusano, and Prinetto address the problems associated with microcode-level dynamic reconfiguration; their approach replaces faulty functional blocks within the operation of fault-free ones. This self-repair doesn't use redundant or spare computation blocks; instead, it employs a microprogrammed-based reconfiguration that supports graceful degradation. The processor implements this reconfiguration internally, so it is transparent to users.

The self-repair process evokes different testing procedures prior to repairing the device; as the repair targets the processor execution unit, the processor reallocates fault-free resources, thus causing a gradual decrease in performance. Two different approaches—which are supported by tools and experimental results—make the proposed technique well suited for highly dependable systems and applications.

Next, in “Online Check and Recovery Techniques for Dependable Embedded Processors,” Pflanz and Vierhaus introduce an efficient approach for online detection of permanent and transient faults. To avoid the high replication levels of modular redundancy, this approach uses a combination of techniques, such as extended Berger and state codes. These codes provide hardware functionality that supports predictions, which in turn are the basis for fault detection in data path and controller operations. The authors then specify a fast recovery process that requires using a duplicated circuit; however, this technique implements redundancy at a level below that of full-scale replacement of a processor. An accompanying micro-rollback scheme integrates online self-checking into processor operations.

Richardson, Sieh, and Elkateeb address the problems associated with transient faults in critical systems under adverse environmental conditions (such as electromagnetic interference). “Fault-Tolerant Adaptive Scheduling for Embedded Real-Time Systems” discusses how fault-tolerant scheduling algorithms are becoming increasingly attractive for integrating real-time performance in embedded systems, such as an unmanned ground vehicle for the military. The adaptive deadline-monotonic scheduling algorithm provides a cohesive framework for scheduling under nominal and faulty conditions. This algorithm delivers high processor utilization and satisfactory task execution. It also reorganizes scheduling objectives to meet criticality, even
in the presence of transient surges of varying severity. This value-time scheduling approach relies on a tunable sliding window mechanism to allow scheduling mode changes that adaptively meet workload characteristics and the requirements of the expected operational environment.

Integrating off-the-shelf components is becoming increasingly attractive as a means to build systems with a wide range of applications. That’s why designers are building complex system structures from prepackaged hardware and software components. Although this strategy provides the necessary customization and functionality at a reasonable cost, the dependability of these integrated systems is sometimes questionable. The questions arise because current tools cannot capture the new relationships among the now integrated components; validation remains a challenge. “Design Validation of Embedded Dependable Systems,” by Bondavalli et al., deals with the important topic of verification by composition. In this article, the authors propose different environments that designers can use to incorporate composition, reuse, and integration into an effective framework. The authors also identify possible techniques for the validation and verification of embedded computing systems.

In “Model-Based Fault-Tolerant Control Reconfiguration of Arbitrary Network Topologies,” Provan and Chen describe a novel procedure that is applicable to real-time embedded computers. This technique achieves fault tolerance through the reconfiguration of arbitrary network topologies, blending system-level requirements with diverse tasks, such as fault diagnosis and control reconfiguration. Assuming a discrete-event system, the authors use a single underlying representation. They show how such a representation can model dynamic behavior and provide appropriate control strategies for a wireless sensor network. This method does so with a modest overhead for reconfiguration. The authors base their analysis on an inference approach; their technique has substantial implications for embedded fault-tolerant computing because of its general applicability to any arbitrary network.

The last theme article, “Fault Detection in a Tristate System Environment” by Feng, Karimi, and Lombardi, presents a novel approach for detecting faults in tristate system environments—systems made of multiple boards. Embedded computers commonly use these environments, which consist of an interconnect, and drivers and receivers with tristate features. The authors present a comprehensive fault model, which includes faults in terminals (drivers and receivers) and nets. This fault model accounts for physical (stuck-at and short) as well as functional (dominance, permanently enabled, or permanently disabled driver modes) faults. Simulation results show that the proposed approach outperforms previous methods.

We sincerely hope that this special issue will serve as a reference publication for many readers involved in the fault-tolerant embedded systems area. We also hope it promotes further research. It is our strong belief that the topics covered by these articles are timely and important, and that the authors provided us with excellent presentations and outstanding technical content. We extend sincere thanks to all the authors and reviewers who contributed to this endeavor. It would not have been possible without their dedication and professional contributions. Finally, a special thanks to Editor-in-Chief Ken Sakamura and the IEEE Micro staff for editing and assembling this issue. Please feel free to contact us if you have questions or comments.

Dimiter R. Avresky is a professor at Northeastern University. His research interests include network computing, performance analysis, cluster computing, parallel and distributed computing, fault-tolerant computing and diagnostics, embedded fault-tolerant systems, and testing and verification of protocols. He served as one of the guest editors for the March 2001 IEEE Transactions on Parallel and Distributed Systems special issue on dependable network computing, the December 2001 IEEE Transactions on Computers special issue on embedded fault-tolerant systems, the May 2000 The Journal of Supercomputing special issue on embedded fault-tolerant systems, and the September-October 1998 IEEE Micro special issue on embedded fault-tolerant systems. Avresky edited and coauthored four books in the field of fault-tolerant parallel and distrib-
uated systems. He is a senior member of IEEE and the IEEE Computer Society.

**Fabrizio Lombardi** is chair of the Department of Electrical and Computer Engineering and holder of the International Test Conference Endowed Professorship at Northeastern University. He is also the associate editor-in-chief of *IEEE Transactions on Computers*. His research interests include fault-tolerant computing, testing and design of digital systems, configurable computing, defect tolerance, and computer-aided design for VLSI. He has a BSc in electronic engineering from the University of Essex, UK; an MS in microwaves and modern optics and a diploma in microwave engineering from the Microwave Research Unit at University College London; and a PhD from the University of London. He received an IEEE Engineering Foundation Research Initiation Award, a Motorola Silver Quill Award, and an IEEE Computer Society Distinguished Visitorship. He has served as guest editor in archival journals and magazines such as the *IEEE Transactions on Computers*, *IEEE Micro*, and *IEEE Design & Test*.

**Karl E. Grosspietsch** is a researcher at the Fraunhofer Institute of Autonomous Intelligent Systems in St. Augustin, Germany. His main research interests are computer architecture, dependable computing, and autonomous systems. The results of his work have been published in more than 130 publications. Since 1998, he has been the speaker of the Technical Committee on Dependability and Fault Tolerance of the German Computing Society, Gesellschaft fuer Informatik.

**Barry W. Johnson** is a professor in the Department of Electrical and Computer Engineering at the University of Virginia. He is cofounder and director of the Center for Semicustom Integrated Systems of Virginia's Center for Innovative Technology, and cofounder and codirector of the Center for Safety-Critical Systems. His research interests include fault-tolerant computing, safety-critical systems, system testing, and system modeling and analysis. Johnson received the BS, ME, and PhD in electrical engineering from the University of Virginia, Charlottesville. He is the author of two textbooks: *The Design and Analysis of Fault-Tolerant Digital Systems*, (Addison-Wesley) and *The Co-Design of Embedded Systems: A Unified Hardware/Software Representation* (Kluwer Academic Publishers). He is an IEEE fellow for his contributions to fault-tolerant computing. He has served on the IEEE's Board of Directors, Executive Committee, Computer Society Executive Committee, Computer Society Board of Governors, and the Technical Activities Board. He is an IEEE Computer Society member and has served as its President, Vice President for Publications, Vice President for Conferences and Tutorials, Vice President for Press Activities, Vice President for Membership Activities, and Treasurer. He is a member of Tau Beta Pi and Eta Kappa Nu.

Direct comments or questions about this special issue to Dimiter R. Avresky, avresky@ece.neu.edu.

For further information on this or any other computing topic, please visit our Digital Library at [http://computer.org/publications/dlib](http://computer.org/publications/dlib).

**you@computer.org**

**FREE!**

All IEEE Computer Society members can obtain a free, portable email alias@computer.org. Select your own user name and initiate your account. The address you choose is yours for as long as you are a member. If you change jobs or Internet service providers, just update your information with us, and the society automatically forwards all your mail.

**Sign up today at** [http://computer.org](http://computer.org)