Test Spring School @
EUROPEAN TEST SYMPOSIUM
Talloires, France
May 25-28, 2012

Test Spring School at ETS'12

In conjunction with the DFG priority program SPP1500: "Dependable Embedded Systems"

Reliability, Online-Test and Fault-Tolerance

1. TSS@ETS 2012

The European Test Symposium offers a 3 day test spring school (TSS@ETS 2012) for Ph. D. and M. Sc. students which will introduce into modern test technology. Renowned experts will give lectures and will cover the main challenges of test and reliability of today's nanoelectronic systems.

This year, emphasis is put on dependability, online test, and fault tolerance of circuits and systems. TSS@ETS offers the unique opportunity to learn about the leading edge of the state of the art in dependable systems design in a comprehensive and compact way. The school will give the opportunity to earn credits and a certificate by passing an exam online.

The school is organized in conjunction with ETS 2012 (www.ieee-ets.org), and the last two lectures are open for the general ETS attendees without additional fee. The school is also organized in conjunction with the DFG priority program SPP1500: "Dependable Embedded Systems". TSS@ETS is offered to registered students at very low cost rate. The school is also available for professionals at higher rates, however priority is given to students on first-come, first-served basis, as the number of attendees is strictly limited.

Hans-Joachim
Wunderlich

Chair of the Steering Committee of ETS

Paolo
Prinetto

Organization Chair

2. TSS@ETS Scientific Committee

- Lorena ANGHEL – TIMA Grenoble – France
- Bernd BECKER – Universität Freiburg – Germany
- Paolo PRINETTO – Politecnico di Torino – Italy
- Michel RENOVELL – LIRMM – Montpellier – France
- Hans-Joachim WUNDERLICH (Chair) – Universität Stuttgart – Germany

3. Venue

Hotel Beau Site

118, rue André Theuriet
BP 4 74290
Talloires, France

Tel: +33 47 73 52 50 00
Email: info@beausite-talloires.com
Web: http://www.beausite-talloires.com

4. Registration

Registrations should be done at:
http://www.iti.uni-stuttgart.de/tss2012

Registration Fees:

- Regular PhD Students: 370 €
- Regular PhD Students (after April 2nd): 410 €
- Industrial PhD Students / Professionals: 700 €
- Industrial PhD Students / Professionals (after April 2nd): 770 €

The ETS2012 package includes food, accommodation and social events of the program.

5. Social Events

Social Activities in TSS 2012 will be done on Saturday, May 26th and on Sunday May 27th, 2012.

On Saturday afternoon of May 26th, students and teachers may join a Forest Rope course in the heart of the forest. It is a discovery adventure course on 5 sites during 2 hours. Tarzan swings, crossings of all kinds, new features and long zip lines will guarantee an adrenaline rush. You have to subscribe to this activity, upon your physical condition. This activity is maintained if the weather permits.

On Sunday May 27th, a special dinner is offered for all participants.

6. Schedule

Friday, May 25	
16:00-18:00	Registration
19:00-	Welcome reception
Saturday, May 26	
7:30-8:00	Registration
8:00-8:15	Welcome Address (Hans-Joachim WUNDERLICH)
8:15-9:30	Each attendee introduces her/himself
9:30-10:30	**Robust System Design: Overcoming Reliability Challenges** Subhasish MITRA (part I)
10:30-10:45	Coffee break
10:45-12:45	Subhasish MITRA (part II)
12:45-13:45	Lunch
13:45-15:15	Subhasish MITRA (part III)
15:15-15:30	Coffee break
15:30-16:30	**Soft Errors: Sources and Mitigation,** Dan ALEXANDRESCU (part I)
16:30-	Social Event
Sunday, May 27	
8:00-10:00	Dan ALEXANDRESCU (part II)
10:00-10:15	Coffee break
10:15-11:15	Dan ALEXANDRESCU (part III)
11:15-12:45	**Designing systems that are Dependable and Secure: Measurement, Analysis and Design,** Ravishankar K. IYER (part I)
12:45-13:45	Lunch
13:45-15:15	Ravishankar K. IYER (part II)
15:15-15:30	Coffee break
15:30-17:00	Ravishankar K. IYER (part III)
17:00-17:15	Coffee break
17:15-18:45	**Validation and Fault Tolerance of Microprocessors** Jacob ABRAHAM (part I)
19:00-	Social Dinner
Monday, May 28	
8:00-9:45	Jacob ABRAHAM (part II)
9:45-10:00	Coffee break
10:00-11:30	Jacob ABRAHAM (part III)
11:30-12:00	Checkout
12:00-13:00	Lunch
Track A 14:00-18:30	**Dependable Processor Design,** Peter HARROD
Track B 14:00-18:30	**Hardware- and Software-Fault Tolerance, Design and Assessment of Dependable Computer Systems,** Jean ARLAT

7. Course details

7.1. Robust System Design: Overcoming Reliability Challenges

Saturday, May 26: 9:30-10-30, 10:45-12:45, 13:45-15:15
Hotel Beau Site

– Speaker

Subhasish MITRA

Departments of Electrical Engineering and Computer Science
Stanford University
Gates 333
353 Serra Mall
Stanford, CA, 94305

Tel: +1 650 724 1915
E-mail: subh@stanford.edu

Professor Subhasish Mitra directs the Robust Systems Group in the Department of Electrical Engineering and the Department of Computer Science of Stanford University, where he is the D.E. Chambers Faculty Scholar of Engineering. Before Stanford, he was a Principal Engineer at Intel Corporation.

Prof. Mitra's research interests include robust system design, VLSI design, CAD, validation and test, and emerging nanotechnologies. His X-Compact technique for test compression has been used in more than 50 Intel products, and has influenced major CAD tools. The IFRA technology for post-silicon validation, created jointly with his student, was characterized as "a breakthrough" in a Research Highlight in the Communications of the ACM. His work on the first demonstration of carbon nanotube imperfection-immune VLSI circuits, jointly with his students and collaborators, was selected by the National Science Foundation (NSF) as a Research Highlight to the United States Congress, and was highlighted "as significant breakthrough" by the Semiconductor Research Corporation, the MIT Technology Review, and several others.

Prof. Mitra's honors include the Presidential Early Career Award for Scientists and Engineers from the White House, the highest US honor for early-career outstanding scientists and engineers, Terman Fellowship, IEEE CAS/CEDA Pederson Award for the IEEE Transactions on CAD Best Paper, and the Intel Achievement Award, Intel's highest corporate honor. He and his students presented award-winning papers at several major conferences:
IEEE/ACM Design Automation Conference, IEEE VLSI Test Symposium, IEEE International Test Conference, and the Symposium on VLSI Technology. At Stanford,

he was honored multiple times by graduating seniors "for being important to them during their time at Stanford."

Prof. Mitra currently serves on the Defense Advanced Research Projects Agency (DARPA) Information Science and Technology Board as an invited member.

– Abstract

Today’s mainstream electronic systems typically assume that transistors and interconnects operate correctly over their useful lifetime. With enormous complexity and significantly increased vulnerability to failures compared to the past, future system designs cannot rely on such assumptions. At the same time, there is explosive growth in our dependency on such systems.

Robust system design is essential to ensure that future systems perform correctly despite rising complexity and increasing disturbances. For coming generations of silicon technologies, several causes of hardware failures, largely benign in the past, are becoming significant at the system-level. With extreme miniaturization of circuits, factors such as transient errors, device degradation, and variability induced by manufacturing and operating conditions are becoming important. While design margins are being squeezed to achieve high energy efficiency, expanded design margins are required to cope with variability and transistor aging. Even if error rates stay constant on a per-bit basis, total chip-level error rates grow with the scale of integration. Moreover, difficulties with traditional burn-in can leave early-life failures unscreened.

This talk will address the following major robust system design objective: cost-effective tolerance and prediction of failures in hardware during system operation. Significant recent progress in robust system design impacts almost every aspect of future systems, from ultra-large-scale networked systems, all the way to their nanoscale components.

– Prerequisites

Basic concepts in digital circuits, systems, computer architecture, and some knowledge of VLSI testing.

– Learning Outcomes

The audience will learn circuit and system-level modeling and design aspects of errors (rather than just technology and physics aspects). Supporting data on designs and technologies, together with technology trends, will be covered. New techniques for analyzing circuit and system-level impact of errors will be discussed. New error resilience techniques will be presented. Finally, an extensive bibliography will be provided.

- **Syllabus**

Basic concepts in reliability; Various reliability failure mechanisms; Basic ideas of reliability, data integrity, silent data corruption and availability; overview of circuit and system-level impact of errors, estimation strategies; derating factors, resilience techniques:

Built-In Soft Error Resilience, Soft Error Correcting Combinational Logic, ECC, Concurrent Error Detection, Parity Prediction and other coding theoretic techniques, Multi-threading, Software Implemented Hardware Fault Tolerance, Application Dependent techniques, On-line self-test and diagnostics, error resilient system architectures.

7.2. Soft Errors: Sources and Mitigation

Saturday, May 26: 15:30-16:30
Sunday, May 27: 8:00-10:00, 10:15-11:15
Hotel Beau Site

– Speaker

Dan ALEXANDRESCU

iRoC Technologies
World Trade Center, PO Box 1510
Grenoble, 38025, France

Tel: +33 438 120 763
E-mail: dan.alexandrescu@iroctech.com

Dan Alexandrescu is the Vice President Engineering of iRoC Technologies and holds a Master Degree in Electronic Engineering from the Politehnica University of Bucarest, Romania and a PhD in Microelectronics from INPG France.

Dan manages the development of commercial products and services in the field of electronic systems reliability and especially their behavior with regard to Single Event Effects. This includes both software and hardware offerings that allows the manufacturer of electronic devices and systems to evaluate and improve the Soft Error performance of their products.

Dan also oversees the research and development efforts of the company in the context of international & regional collaborative R&D projects.

Additionally, he has published a number of research papers in the proceedings of international conferences and dedicated journals and a chapter in a book dedicated to Soft Errors.

– Abstract

Perturbations induced by Single Event Effects (SEEs or more widely known as Soft Errors) may cause system downtime, data corruption and maintenance incidents. Thus, the SEEs are a threat to the overall system reliability performance causing engineers to be increasingly concerned about the analysis and the mitigation of radiation-induced failures, even for commercial systems performing in a natural working environment.

The SEEs are physical phenomena, strongly dependent on technological process and design implementation. On the other end of the scale, any SEEs-induced faults have a potential for causing system-wide consequences. Thus, any Soft Error Rate (SER) analysis

approach will require a multitude of competencies. The reliability engineer will have to interact with all the actors from the design flow from the technology/library provider to the system architect, while taking in account the reliability targets that are required by the final application.

This lecture presents an overview of the Single Event Effects in complex ASICs from process aspects to system-wide consequences. The study relies on a complete approach that integrates tightly with the design flow, enabling the reliability engineer to closely support the circuit designers in order to improve the overall Soft Error Rate of the system. Additionally, we present an introduction on the available methods and approaches that could be used to improve the Soft Error performance of the circuit through architectural and design choices with the firm goal of improving customer experience when using high availability products.

Dealing with all these subjects, this lecture hopes to improve the SER awareness in the electronic design field and to offer practical solutions when dealing with these problems, in helping both SER analysis and improvement efforts.

– Prerequisites

- Fundamental knowledge of physics in microelectronic devices
- Basic knowledge of microelectronic VLSI design flow, cell & circuit design
- Elements of electronic systems reliability

– Learning Outcomes

This course will familiarize you with the challenges of the managing Soft Error Rate issues in advanced electronics systems and practical approaches to overcome these difficulties.

Assimilated to a reliability or design engineer, the participants will acquire the necessary knowledge to understand, evaluate and improve the SER issues in complex electronic devices and thus able to offer a valuable contribution through the entire design flow.

– Syllabus

The course is organized as follows:

1. The core module presents an overview of the Single Event Effects, a justification of the academic and industrial interest in this issue and the impact of SEE on the reliability of electronic devices and systems. We will discuss the SEE production and propagation, their manifestation at each circuit abstraction level and an overall SER characterization flow aiming at providing accurate information about the circuit SEE resiliency. The first step in this characterization flow consists in evaluating the energetic particle interaction with the electronic circuit at process, transistor and cell level. The produced event (such as Single Event Transient/Upset/Bit Upset/Multiple

Cell Upset/Single Event Latch-up, etc) will affect the function of the device causing temporary or permanent effects. The system-wide consequence can be very diverse: reboots, crashes, data corruption, and hardware failures with an obvious impact on the performance of the overall equipment.

2. The module presents practical approaches on how to integrate SER efforts during the various design phases (architecture, RTL or High-Level Synthesis, gate-level, cell) with an overall goal of analyzing the system performances and resiliency wrt. SEEs. The outcomes of this process are twofold:
   a. Quantitative results, such as the SER data for various cells/signals or block from the circuit, allowing the computation of SER metrics for each feature of the design.
   b. Qualitative results about the behavior of the circuit in the presence of errors that will be used during the error mitigation stage, especially when devising error protection strategies.
3. The manufactured system can be also tested and validated using a variety of hardware testing techniques. Radiation tests are very useful in advanced manufacturing stages. These tests can help identify most frequent out-of-spec issues, validate designs and quantify field risk. The most powerful advantage is the fact that the products are tested integrally in conditions that are very close to the natural atmospheric environment. Testing helps also debugging/testing software problems, especially in the presence of hardware failures and helps isolate critical hardware/software modules. It is a very effective tool for comparing expectations versus observed results. In particular, it allows evaluating the contribution of SER to the downtime of the product. The correct and opportune estimation of the SER definitely helps avoiding later problems during deployment.
4. Since the overall SEE management is a shared responsibility that requires inter- and intra-company collaboration, this module deals with the impact of SEE on the supply chain from the technology provider (foundry) to the system integrator and final users.

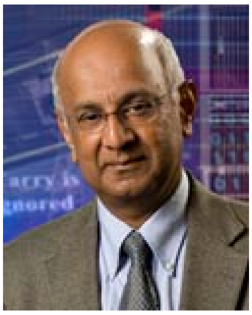
This module provides the required SER knowledge in order to allow the system architect and the designers to direct implementation choices, select a design hardening methodology, establish a failure recovery/mitigation strategy and help the support engineers to accompany the final users of the design in building reliable systems. Concerning the SER improvement task, multiple approaches are possible: process improvements, hardened cells, circuit and system error mitigation techniques, etc. Most of these solutions come at some added costs, thus some compromise must be found between error handling capability and cost overheads.

7.3. Designing systems that are Dependable and Secure: Measurement, Analysis and Design

Sunday, May 27: 11:15-12:45, 13:45-15:15, 15:30-17:00
Hotel Beau Site

– Speaker

Ravishankar K. IYER



Coordinated Science Laboratory
University of Illinois
1308 West Main Street
Room 255
Urbana, Illinois 61801
USA

Tel: +1 217 333 7774
E-mail: iyer@crhc.uiuc.edu

Professor Iyer is the George and Ann Fisher Distinguished Professor of Engineering at the University of Illinois at Urbana-Champaign. He holds joint appointments in the Department of Electrical and Computer Engineering, the Coordinated Science Laboratory (CSL) and the Department of Computer Science. He also serves as Director of the Center for Reliable and High-Performance Computing at Illinois and as Chief Scientist of the Information Trust Institute.

Prof Iyer's research contributions have led to major advances in the design and validation of dependable computing systems. He has authored or co-authored close to three hundred refereed publications, graduated over 35 PhD students and many Masters students. He currently leads the Trusted Illiac project at Illinois. Funded by both Industry and government, the project is developing adaptive, application-aware architectures for supporting a wide range of dependability and security requirements.

Iyer is also the founder of Armored Computing Inc. a profitable Illinois company with worldwide customers including Honeywell, Motorola, and Huawei.

Professor Iyer is a Fellow of the American Association for the Advancement of Science, the IEEE, and the ACM. He has received several awards, including the Humboldt Foundation Senior Distinguished Scientist Award for excellence in research and teaching, the AIAA Information Systems Award and Medal for "fundamental and pioneering contributions towards the design, evaluation, and validation of dependable aerospace computing systems," and the IEEE Emanuel R. Piore Award "for fundamental contributions to measurement, evaluation, and design of reliable computing systems." Professor Iyer is also the recipient of the degree of Doctor Honaris Causa from France's Sabatier University of Toulouse in recognition of his outstanding research contributions in dependable computing and for longstanding contributions to joint research LAAS/CNRS in Toulouse. He was recently awarded the 2011 Outstanding Contributions award by the Association of Computing Machinery (ACM)- Special Interest Group on Security and Audit (SIGSAC) for his fundamental and far-reaching contributions in secure and dependable computing systems.

– Abstract

Quick way of learning principles and techniques to build and validate reliable computing systems and networks.

This course introduces a system (both hardware and software) view of design issues in reliable computing. The material represents a broad spectrum of hardware and software error detection and recovery techniques. The lectures discuss how the hardware and software techniques interplay; e.g., what techniques can be provided in hardware, operating system and network communication layers, and what can be provided via a distributed software layer and in the application itself.

After introducing basic concepts and terms including reliability, availability, and hardware and software fault models, the course continues with discussions of hardware redundancy, coding techniques, signature-base error checking, processor-level error detection and recovery (e.g., duplicate execution and comparison), checkpoint and recovery (single process and distributed environment), software fault tolerance techniques (e.g., process pair, robust data structures, recovery blocks, and N-version programming), and finally, network specific issues (e.g., providing consistent data and reliable communications). The capabilities and applicability of discussed techniques are illustrated with examples of real applications and systems.

– Prerequisites

Basic understanding of computer systems, hardware and software. BS or MS in Computer Engineering or Computer Science.

– Learning Outcomes

You will learn the concepts, principles and practice that jointly underlie the development of systems that are reliable and secure. You will be exposed to new and challenging application domains and computing paradigms being implemented in practice and studied in research. Overall the course will allow you to be in a position to develop or research new systems and technologies in the context of their resiliency (dependability and security).

– Syllabus

1. Introduction
   - System view of high availability design

- Fault models
- Example of high-availability system

2. Hardware redundancy
   - Basic approaches to hardware redundancy
   - Static and dynamic redundancy
   - Voting

3. Error detection techniques
   - Timers, watchdogs, heartbeats
   - Audits, assertions, control flow and program invariants checks
   - Operating system exception handling
   - Example application

4. Coding techniques
   - Error detecting and error correcting codes
   - Hamming codes
   - codes for storage and communication
   - codes for arithmetic operations

5. Processor-level detection and recovery
   - Instruction retry, duplication, multithreading
   - Checker processor
   - RSE (reliability and security engine)

6. Disk arrays (RAID)
   - Organization of RAIDs
   - Example design and evaluation of cache-based RAID Controller

7. Checkpointing and recovery
   - Forward and backward error recovery
   - Checkpoint and recovery in networked systems
   - Synchronous checkpointing and recovery
   - Asynchronous checkpointing and recovery
   - Checkpointing in distributed databases
   - IRIX operating system checkpoint and restart

8. Software fault tolerance
   - Process pairs
   - robust data structures
   - N-version programming
   - recovery blocks
   - routines - example of BM-MVS

9. Network specific issues
    - Broadcast protocols
    - Agreement protocols (Byzantine agreement, Consensus, Interactive consistency)
    - Application of agreement algorithms

10. High Availability Middleware
    - Replication
    - Self-checking processes
    - Application example

11. Dependability Validation
    - Validation methods
    - Design phase- Hierarchical fault simulation
    - Prototype phase - HW or SW implemented fault injection
    - Operational phase - Measurement of field systems

7.4. Validation and Fault Tolerance of Microprocessors

Sunday, May 27: 17:15-18:45
Monday, May 28: 8:00-9:45, 10:00-11:30
Hotel Beau Site

– Speaker

Jacob A. ABRAHAM

The University of Texas at Austin
Electrical and Computer Engineering Department
1 University Station C8800
Austin, Texas 78712-0323

Tel: +1 512 471 8983, Fax: +1 512 4718967
E-Mail: jaa@cerc.utexas.edu
Url: http://www.cerc.utexas.edu/~jaa/

Jacob A. Abraham is Professor of Electrical and Computer Engineering at the University of Texas at Austin. He is also the director of the Computer Engineering Research Center and holds a Cockrell Family Regents Chair in Engineering. He received his Ph.D. in Electrical Engineering and Computer Science from Stanford University in 1974. From 1975 to 1988 he was on the faculty of the University of Illinois, Urbana, Illinois. His research interests include VLSI design and test, formal verification, and fault-tolerant computing. He has published extensively, has received many "best paper" awards, and is included in the ISI list of the most cited researchers in the world. He has supervised more than 80 Ph.D. dissertations, and is particularly proud of the accomplishments of his students, many of whom occupy senior positions in academia and industry. He has been elected Fellow of the IEEE as well as Fellow of the ACM, and is the recipient of the 2005 IEEE Emanuel R. Piore Award.

– Abstract

With the increasing complexities of systems, made possible through higher levels of integration and lower manufacturing costs, ensuring dependable operation is becoming more and more difficult. Factors affecting dependability include design bugs, manufacturing defects and process variations, environmental factors and external attacks. This lecture will focus on techniques for ensuring dependable operation of microprocessors.

These include approaches for validating that the designs are free of bugs, using engineering and formal techniques, as well as methods for system design so that the processors will continue to operate in spite of failures of some components. The lecture will also address ways of tolerating external attacks on processors.

– Prerequisites

Students should have a basic understanding of logic design, computer architecture and semiconductor electronics. Any introductory university textbook on these subjects would suffice for the preliminary reading.

– Learning Outcomes

Students will gain a basic understanding of the fundamentals of fault-tolerant microprocessors, including design and analysis of systems based on them. Students will also learn industry practice and university research directions in these topics. Students will have the background for fault-tolerant systems in industry, or for research in the field of fault tolerance.

– Syllabus

- Fundamentals of verification and validation
- Simulation techniques and use of assertions
- Formal equivalence checking and property checking
    - Industry applications and research directions
- Redundancy techniques for fault tolerance
    - Hardware and time redundancy
- System architectures for fault tolerance
    - Fault-tolerant system examples
    - Overview of fault-tolerant system evaluation
- Application-level techniques for fault tolerance
    - Hardware and software approaches
- Security issues
    - Dealing with attacks

7.5. Dependable Processor Design

Monday, May 28: 14:00-18:30
TRACK A of ETS 2012

– Speaker

Peter HARROD

Processor Division
ARM Ltd
110 Fulbourn Road
Cambridge
CB1 9NJ

Tel: +44 1223 400 473
E-mail: Pete.Harrod@arm.com

Peter Harrod (IEEE Member ’80-Senior Member ’99) graduated with a BSc(Eng) from the University of the Witwatersrand in 1976 and with MSc and PhD degrees from the University of Manchester Institute of Science and Technology in 1978 and 1982 respectively.

From 1982-1985, he was a Research Engineer in the Very High Performance IC Laboratory at the GEC Hirst Research Centre, where he was involved in pattern processing for E-beam lithography and in the implementation of CMOS-SOS VLSI ICs.

From 1985-1988, he was a Senior Staff Engineer in the High-End Microprocessor Group at Motorola Inc in Austin, Texas, where he did logic and circuit design for the MC68030 and MC68040 microprocessors.

In 1988, he joined Acorn Computers Ltd in Cambridge, UK, where he was a Senior Design Engineer in the Advanced R&D Department and was involved in the design of a floating point chip and carried out one of the first implementations of IEEE 1149.1 boundary scan.

ARM was spun out from Acorn Computers in 1990 and he was one of the founding team. Since then he has worked on a wide variety of CPU, SOC and debug and trace units.

He is now a Manager in the CPU design group at ARM, where he continues to work on the design and verification of embedded CPUs. He has a particular interest in the areas of design for test and debug and in the design of dependable processors.

He is a Fellow of the IET and has served on several IEEE standards and conference program committees.

– Abstract

In this session, I’ll try to bring together all the theory that you have learnt about fault tolerance and show how it can be applied in a real practical example. This example will be based around an existing dependable processor.

Embedded processors are used in many applications that require a defined level of reliability, safety and/or availability. There are many approaches to providing the required level of fault tolerance – at the circuit, logic, microarchitecture, chip and system level – each of which incurs a certain cost.

In many volume applications, for example in the automotive market, you need to achieve a balance between reaching the required level of reliability, safety and/or availability and the additional cost involved. So when designing a dependable processor, you need to consider not only the kinds of faults that might occur but also the effect that these faults might have on system operation – and think carefully about how to protect against these faults without adding too much to the system cost.

In this session, I'll start by saying what I mean by dependability in the context of an embedded processor and then take a look at safety standards and what these imply about processor design. I'll then discuss the kinds of faults that might affect a processor's operation – hard versus transient faults, latent faults & wearout mechanisms – and how these faults might be detected (and possibly corrected).

I'll then look at how the requirements for reliability, safety and availability can be translated into real systems and discuss some dependable architectures. Using the example of an embedded processor that was designed to satisfy this market, I'll look in detail at how features such as ECC on the memories, error caches, dual-core lock step and processor diversity address these requirements. I'll also look at how external monitoring hardware can be used in conjunction with an embedded processor to achieve the required dependability.

Finally, I'll discuss some future challenges in dependable processor design, including the effect of process scaling. I'll briefly describe some experimental test structures that could be used to detect and mitigate for failure mechanisms, such as wear-out.

– Prerequisites

No particular requirements. This session will build on the theory that has been presented in earlier sessions and the suggested preliminary reading for those earlier sessions will be equally applicable to this session.

– Learning Outcomes

- What is meant by dependability in the context of a processor
- Requirements of safety standards and how these can be incorporated into a processor
- How to analyze a processor for dependability
- An understanding of possible architectures for processor dependability
- An appreciation of some examples of existing dependable processor designs
- An understanding of how extra hardware can be added to a processor to build a dependable system
- An appreciation of some future challenges

- **Syllabus**

- Dependability as a combination of reliability, availability, maintainability and safety (RAMS)
- Dependability vs. security
- Principles of functional safety: functional safety standards like IEC 61508 and ISO 26262
- Requirements of IEC 61508 and ISO 26262 for processors
  - Requirements for HW random failures (for permanent and transient faults)
  - Requirements for systematic failures (verification of a processor, avoidance or validation of unpredictable instructions)
  - Requirements for common-cause failures (clock, reset, temperature, EMC)
  - Requirements for Worst Case Execution Time
- Safety and Dependability analysis of a processor
  - FMEA/FMEDA of a processor
  - Computing the failure rate of a processor
  - Vulnerability factors of a processor
  - Fault injection of a processor
  - The “safety manual” of a processor
- Architectures for processor dependability
  - Dependable processor (e.g. working at cell level, Razor, Logic BIST)
  - Homogenous redundancy (e.g. Dual-core lock-step, TMR)
  - Asymmetric redundancy (e.g. TMR, YOGITECH’s faultRobust CPU, Challenge-Response architecture)
  - Achieving dependability by software (e.g. designing a SW Test Library to test a processor at run-time)
  - Monitoring a processor with watchdogs and MPU/TPU units
- SW aspects of processor dependability
  - HW-SW interactions and configurations in a processor
  - A safe and dependable compiler
- Lock-step architecture using an ARM processor as the example (Cortex-R4,Cortex-R5,Cortex-R7)
  - Lock-step and compare outputs to avoid common-mode failures (delays, guard rings, macro rotation)
  - Split-lock functionality
- ECC and parity scheme to handle faults in the memory and bus interconnect of an ARM processor
  - Error caches
- Safety and dependability of L1, L2 and L3 caches – MBIST and repairable memories
- A safety eco-system for ARM Cortex-M3 processor
  - Highlights of the FMEDA of the ARM Cortex-M3 processor
  - FRCPU_armcm3, a tightly coupled optimized and diverse supervisor for ARM Cortex-M3 processor;
  - Failure identification and fail-operational strategies for an ARM Cortex-M3 with fRCPU

    - FRSTL_armcm3, a SW Test Library to reach the SIL2/ASILB level of safety integrity level for the ARM Cortex-M3 processor
    - FRMEM and fRBUS - IPs to detect faults in the memory and bus system of a Cortex-M3 processor
- Challenges for the future
    - Effect of process scaling on dependable processor design
    - On-chip monitors to detect and mitigate in-service degradation and/or failures – e.g. wear-out: NBTI, oxide degradation
    - Low-power systems - detecting and correcting state corruption after power-down

7.6. Hardware- and Software-Fault Tolerance, Design and Assessment of Dependable Computer Systems

Monday, May 28: 14:00-18:30
TRACK B of ETS 2012

– Speaker

Jean ARLAT

LAAS-CNRS
7, avenue du Colonel Roche
31077 Toulouse Cedex 4 - FRANCE

Tel: +33 5 61 33 62 70
E-mail: jean.arlat@laas.fr
Url: http://homepages.laas.fr/arlat

Jean Arlat was born in Toulouse (FR) in 1953. He received the Engineer diploma from the Toulouse National Institute of Applied Sciences (INSAT) in 1976 and the Docteur-Engineer and Docteur Dès-Sciences degrees from the Toulouse National Polytechnic Institute (INPT) in 1979 and 1990, respectively. He has been with LAAS-CNRS since 1976, where he currently holds a position of Directeur de Recherche at CNRS, the French National Organization for Scientific Research, within the Dependable Computing and Fault Tolerance Group that he led from 2003 to 2008. In January 2011, he was appointed as deputy director of the laboratory and he is currently the director. From 2007 to 2010, he coordinated the research area on Critical Information Systems, one of the 4 scientific domains characterizing LAAS research activities.

His research interests include the architecting of safe and secure embedded computerized systems and the dependability assessment of computer systems — using both analytical modeling and experimental approaches (especially, fault injection). He authored or co-authored more than 120 papers for international and national journals and conferences 3 books and 21 book sections.

He has contributed to several European projects and networks and also various contracts with industry. From that respect, from 1997 to 2000, he led the Laboratory for Dependability Engineering (LIS) set between LAAS and five leading companies: Airbus, Astrium, Électricité de France, AREVA TA and Thales. Subsequently, from 2001 to 2004, he coordinated the activities of the Network for Dependability Engineering (RIS) that extended the cooperation started within LIS.

From 1999 to 2005, he chaired the IFIP Working Group 10.4 on Dependable Computing & Fault Tolerance and received the IFIP Silver Core Award in 2007. In France, he is currently a member of the Board of Directors of the Association of

Carnot Institutes that gathers selected research labs featuring significant partnership with industry.

– Abstract

This lecture covers the main design and assessment issues that are to be considered when developing dependable computer systems. It is organized into four main parts.

After a short introduction aimed at motivating the relevance of the topic covered, the first part briefly introduces the general concepts and related terminology for dependable computing including the notions of fault, error and failure and the main approaches towards dependability: fault tolerance, fault removal and fault forecasting.

In the second part, it addresses the fault tolerance techniques (encompassing error detection, error recovery and fault masking) that can be used to cope with accidental faults (physical disturbances, software bugs, etc.) and to some extent, malicious faults (e.g., attacks, intrusions). In particular, several forms of redundancies (space, temporal, data, etc.), as well as the important notion of diversified design will be described and illustrated by means of examples.

The third part covers the methods and techniques — both analytical (stochastic processes) and empirical (controlled experiments) — that can be used to objectively assess the coverage of the fault tolerance mechanisms and then infer the level of dependability achieved. The actual impact of fault-tolerant architectures on dependability, leading to the essential notion of coverage (with respect to fault tolerance) is precisely identified and exemplified. A special focus is put on controlled experiments based on fault injection techniques (hardware-, simulation-, and software-based fault injection).

The fourth and last part describes most recent trends in controlled experiments aimed at developing benchmarks for robustness testing purpose and for fairly comparing the dependability features of several computer systems and components.

Finally, a few concluding remarks will depict some emerging challenges and future trends in the domain of dependable computing.

– Prerequisites

Awareness in:

- Digital Design,
- Computer Systems Architecture,
- Probabilities and Statistics,
- Reliability Modeling and Evaluation,

would be helpful to easily catch up with the topics being covered in the Lecture.

Suggested preliminary reading:

Highly recommended bibliographical references are identified by a "*" symbol.

Journal and Conference Papers

[Arlat *et al.* 1990] J. Arlat, M. Aguera, L. Amat, Y. Crouzet, J.-C. Fabre, J.-C. Laprie, E. Martins and D. Powell, "Fault Injection for Dependability Validation — A Methodology and Some Applications", *IEEE Trans. on Software Engineering*, 16 (2), pp.166-182, February 1990.*

[Laprie *et al.* 1990] J.-C. Laprie, J. Arlat, C. Béounes and K. Kanoun, "Definition and Analysis of Hardware-and-Software Fault-Tolerant Architectures", *Computer*, 23 (7), pp.39-51, July 1990.*

[Hsueh *et al.* 1997] M.-C. Hsueh, T. K. Tsai and R. K. Iyer, "Fault Injection Techniques and Tools", *Computer*, 30 (4), pp.75-82, April 1997.

[Carreira *et al.* 1999] J. V. Carreira, D. Costa and J. G. Silva, "Fault Injection Spot-checks Computer System Dependability", *IEEE Spectrum*, 36, pp.50-55, August 1999.*

[Koopman & DeVale 1999] P. Koopman and J. DeVale, "Comparing the Robustness of POSIX Operating Systems", in *Proc. 29th Int. Symp. on Fault-Tolerant Computing (FTCS-29),* (Madison, WI, USA), pp.30-37, IEEE CS Press, 1999.*

[Tsai *et al.* 1999] T. K. Tsai, M.-C. Hsueh, Z. Kalbarczyk and R. K. Iyer, "Stress-Based and Path-Based Fault Injection", *IEEE Trans. on Computers*, 48 (11), pp.1183-1201, November 1999.

[Cheynet *et al.* 2000] P. Cheynet, B. Nicolescu, R. Velazco, M. Rebaudengo, M. Sonza Reorda and M. Violante, "Experimentally Evaluating an Automatic Approach for Generating Safety-Critical Software with respect to Transient Errors", *IEEE Trans. on Nuclear Science*, 47 (6), pp.2231-2236, December 2000.

[Arlat *et al.* 2002] J. Arlat, J.-C. Fabre, M. Rodríguez and F. Salles, "Dependability of COTS Microkernel-Based Systems", *IEEE Trans. on Computers*, 51 (2), pp.138-163, February 2002.

[Arlat *et al.* 2003] J. Arlat, Y. Crouzet, J. Karlsson, P. Folkesson, E. Fuchs and G. H. Leber, "Comparison of Physical and Software-Implemented Fault Injection Techniques", *IEEE Trans. on Computers*, 52 (9), pp.1115-1133, September 2003.

[Avižienis *et al.* 2004] A. Avižienis, J.-C. Laprie, B. Randell and C. Landwehr, "Basic Concepts and Taxonomy of Dependable and Secure Computing", *IEEE Transactions on Dependable and Secure Computing*, 1 (1), pp.11-33, Jan.-March 2004.*

[Siewiorek *et al.* 2004] D. P. Siewiorek, R. Chillarege and Z. Kalbarczyk, "Reflection on Industry Trends and Experimental Research in Dependability", *IEEE Transactions on Dependable and Secure Computing*, 1 (2), pp.109-127, 2004.*

[Albinet *et al.* 2004] A. Albinet, J. Arlat and J.-C. Fabre, "Characterization of the Impact of Faulty Drivers on the Robustness of the *Linux* Kernel", in *Proc. IEEE/IFIP Int. Conf. on Dependable Systems and Networks (DSN-2004),* (Florence, Italy), pp.867-876, IEEE CS Press, 2004.

[de Andrés *et al.* 2008] D. de Andrés, J. C. Ruiz, D. Gil and P. Gil, "Fault Emulation for Dependability Evaluation of VLSI Systems", *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, 16 (4), pp.422-431, April 2008.

[Arlat & Moraes 2011] J. Arlat and R. Moraes, "Collecting, Analyzing and Archiving Results from Fault Injection Experiments", in *Proc. 5th Latin American Symposium on Dependable Computing (LADC-2011),* (São José dos Campos, Brazil), IEEE CS Press, 2011.

[Arlat 2011] J. Arlat, "Dependable Computing and Assessment of Dependability", in *GI/GMM/ITG Workshop on Reliability and Design,* (Hamburg, Germany), VDE, 2011.

Books and Chapters

[Siewiorek & Swarz 1992] D. P. Siewiorek and R. S. Swarz, *Reliable Computer Systems - Design and Evaluation,* 908p., Digital Press, Bedford, MA, USA, 1992.

[Arlat *et al.* 1999] J. Arlat, Y. Crouzet, P. David, J.-L. Dega, Y. Deswarte, J.-C. Laprie, D. Powell, C. Rabéjac, H. Schindler and J.-F. Soucailles, "Fault Tolerant Computing", in *Encyclopedia of Electrical and Electronics Engineering* (J. G. Webster, Ed.), 7, pp.285-313, J. Wiley & Sons, New York, USA, 1999.

[Benso & Prinetto 2003] A. Benso and P. Prinetto (Eds.), *Fault Injection Techniques and Tools for Embedded Systems Reliability Evaluation,* Frontiers in Electronic Testing, 23, 245p., Kluwer Academic Publishers, London, UK, 2003.

[Kanoun & Spainhower 2008] K. Kanoun and L. Spainhower (Eds.), *Dependability Benchmarking for Computer Systems* 362p., IEEE CS Press and Wiley, 2008.

[Powell *et al.* 2011] D. Powell, J. Arlat, Y. Deswarte and K. Kanoun, "Tolerance of Design Faults", in *Festschrift Randell* (C. B. Jones and J. L. Lloyd, Eds.), LNCS 6875, pp.428-452, Springer-Verlag, Berlin Heidelberg, 2011.

– Learning Outcomes

The students attending the Lecture will be acquainted with the main relevant concepts attached to dependable computing. They will be able to master the respective dependability attributes and identify the various categories of fault tolerance (aka, on-line testing) techniques and specific mechanisms, suitable to achieve high-levels of dependability. The various techniques described are meant to cope with hardware faults, as well as software faults and potentially malicious faults, thanks to the design diversity approach.

The Lecture will also allow the attendees to become knowledgeable about the overall approaches for dependability assessment encompassing and combining analytical modeling and controlled experimentation. This includes developing probabilistic behavioral models, testing the robustness and evaluating the efficiency (the notion fault tolerance coverage) of these fault tolerance mechanisms. Concerning the later, the material covered includes the various dimensions attached to a fault injection campaign: specification and the design (identification of the relevant measures and readouts, selection of the fault- and work- loads), the conduct of the series of experiments and the analysis of the outcomes (collection and processing of the readouts, inferences).

Thanks to didactic examples, the attendees will also learn up-to-date insights and hints about the specialization and adaptation of these notions when dealing with the emerging concept of "dependability benchmarking". Relevant features include: portability, reproducibility, acceptability, openness.

– Syllabus

Introduction: Motivation and Outline

Part 1: Basic Concepts and Terminology

1.1. The Notion of Dependability
1.2. The Dependability Attributes
1.3. Dependability Threats: Fault, Error, Failure Pathologies
1.4. Dependability Procurement
1.5. Dependability Assessment

Part 2: Fault-Tolerant Computer Architectures

2.1. Error Detection
2.1.1. Error Detecting Codes
2.1.2. Replication and Comparison
2.1.3. Temporal and Execution Checks
2.1.4. Likelihood Checks
2.1.5. Structured Data Checks
2.1.6. Wrapping
2.1.7. Self-Checking Component
2.2. System Recovery
2.2.1. Backward Error Recovery (Roll-back)
2.2.2. Forward Error Recovery (Roll-forward)
2.2.3. Error Compensation

- *Error Detection and Compensation*
  Error Masking
  Error Correcting Codes

2.3. Design Diversity

2.3.1. Redundancy and Common Mode Failures
2.3.2. Diversification Techniques
– *Recovery Blocks*
– *N-Version Programming*
– *N-Self-Checking Programming*
2.4. Examples of Fault-Tolerant Computer Systems
2.4.1. Airbus A3XX Series
2.4.2. Boeing 777
2.4.3. Ansaldo's Computer Based Interlocking
2.4.4. Safe and Secure Maintenance Laptop

Part 3: Experimental Assessment of Dependability

3.1. Dependability Evaluation
3.1.1. Fault Tolerance Coverage
3.1.2. Fault Injection-based Assessment
3.2. Fault Injection Techniques
3.2.1. (Physical) Hardware-implemented
3.2.2. Simulation-based Injection
3.2.3. Software-implemented
3.2.4. Combined and Emerging Techniques
3.3. Examples of Experimental Results
3.3.1. The MARS Fault-Tolerant Distributed System
3.3.2. The Delta-4 Dependable Distributed Architecture

Part 4: Dependability Benchmarking

4.1. Requirements and Characteristics
4.2. Robustness Benchmark
4.3. Fault Tolerance Benchmark
4.4. Integration of (C)OTS Components
4.5. Selected Examples
4.5.1. The Ballista Environment
4.5.2. The MAFALDA Prototype Tool
4.5.3. The DBench Project

Conclusion: Wrap up, Emerging Challenges and Future Trends