

FAULT INJECTION TECHNIQUES AND TOOLS FOR EMBEDDED SYSTEMS
RELIABILITY EVALUATION

FRONTIERS IN ELECTRONIC TESTING

Consulting Editor

Vishwani D. Agrawal

Books in the series:

Fault Injection Techniques and Tools for Embedded Systems Reliability Evaluation

A. Benso & P. Prinetto
ISBN: 1-4020-7589-8

High Performance Memory Memory Testing

R. Dean Adams
ISBN: 1-4020-7255-4

SOC (System-on-a-Chip) Testing for Plug and Play Test Automation

K. Chakrabarty
ISBN: 1-4020-7205-8

Test Resource Partitioning for System-on-a-Chip

K. Chakrabarty, Iyengar & Chandra
ISBN: 1-4020-7119-1

A Designers' Guide to Built-in Self-Test

C. Stroud
ISBN: 1-4020-7050-0

Boundary-Scan Interconnect Diagnosis

J. de Sousa, P. Cheung
ISBN: 0-7923-7314-6

Essentials of Electronic Testing for Digital, Memory, and Mixed Signal VLSI Circuits

M.L. Bushnell, V.D. Agrawal
ISBN: 0-7923-7991-8

Analog and Mixed-Signal Boundary-Scan: A Guide to the IEEE 1149.4 Test Standard

A. Osseiran
ISBN: 0-7923-8686-8

Design for At-Speed Test, Diagnosis and Measurement

B. Nadeau-Dosti
ISBN: 0-79-8669-8

Delay Fault Testing for VLSI Circuits

A. Krstic, K.-T. Cheng
ISBN: 0-7923-8295-1

Research Perspectives and Case Studies in System Test and Diagnosis

J.W. Sheppard, W.R. Simpson
ISBN: 0-7923-8263-3

Formal Equivalence Checking and Design Debugging

S.-Y. Huang, K.-T. Cheng
ISBN: 0-7923-8184-X

Defect Oriented Testing for CMOS Analog and Digital Circuits

M. Sachdev
ISBN: 0-7923-8083-5

Reasoning in Boolean Networks: Logic Synthesis and Verification Using Testing Techniques

W. Kunz, D. Stoffel
ISBN: 0-7923-9921-8

Introduction to JDD Testing

S. Chakravarty, P.J. Thadikaran
ISBN: 0-7923-9945-5

Multi-Chip Module Test Strategies

Y. Zorian
ISBN: 0-7923-9920-X

Testing and Testable Design of High-Density Random-Access Memories

P. Mazumder, K. Chakraborty
ISBN: 0-7923-9782-7

From Contamination to Defects, Faults and Yield Loss

J.B. Khare, W. Maly
ISBN: 0-7923-9714-2

FAULT INJECTION TECHNIQUES AND TOOLS FOR EMBEDDED SYSTEMS RELIABILITY EVALUATION

Edited by

ALFREDO BENSO

Politecnico di Torino, Italy

and

PAOLO PRINETTO

Politecnico di Torino, Italy

KLUWER ACADEMIC PUBLISHERS

NEW YORK, BOSTON, DORDRECHT, LONDON, MOSCOW

eBook ISBN: 0-306-48711-X
Print ISBN: 1-4020-7589-8

©2004 Springer Science + Business Media, Inc.

Print ©2003 Kluwer Academic Publishers
Dordrecht

All rights reserved

No part of this eBook may be reproduced or transmitted in any form or by any means, electronic, mechanical, recording, or otherwise, without written consent from the Publisher

Created in the United States of America

Visit Springer's eBookstore at:
and the Springer Global Website Online at:

<http://www.ebooks.kluweronline.com>
<http://www.springeronline.com>

Contents

Contributing Authors.....	xiii
Preface	1
Acknowledgments	3
PART 1: A FIRST LOOK AT FAULT INJECTION	5
Chapter 1.1: FAULT INJECTION TECHNIQUES	7
1. Introduction	7
1.1 The Metrics of Dependability.....	8
1.2 Dependability Factors.....	9
1.3 Fault Category	10
1.3.1 Fault Space	10
1.3.2 Hardware/Physical Fault.....	11
1.3.3 Software Fault	12
1.4 Statistical Fault Coverage Estimation.....	13
1.4.1 Forced Coverage.....	14
1.4.2 Fault Coverage Estimation with One-Sided Confidence Interval	16
1.4.3 Mean Time To Unsafe Failure (MTTUF) [SMIT_00].....	17
2. An Overview of Fault Injection.....	18
2.1 The History of Fault Injection	19
2.2 Sampling Process.....	20
2.3 Fault Injection Environment [HSUE_97].....	20

2.4	Quantitative Safety Assessment Model.....	21
2.5	The FARM Model	24
2.5.1	Levels of Abstraction of Fault Injection.....	25
2.5.2	The Fault Injection Attributes.....	25
3.	Hardware-based Fault Injection.....	28
3.1	Assumptions	29
3.2	Advantages	29
3.3	Disadvantages.....	30
3.4	Tools	30
4.	Software-based Fault Injection.....	31
4.1	Assumptions	32
4.2	Advantages	32
4.3	Disadvantages.....	32
4.4	Tools	33
5.	Simulation-based Fault Injection.....	33
5.1	Assumptions	33
5.2	Advantages	34
5.3	Disadvantages.....	34
5.4	Tools	34
6.	Hybrid Fault Injection	35
6.1	Tools.....	35
7.	Objectives of Fault Injection	35
7.1	Fault Removal [AVRE_92].....	36
7.2	Fault Forecasting [ARLA_90].....	37
8.	Further Researches	37
8.1	No-Response Faults.....	38
8.2	Large Number of Fault Injection Experiments Required	39
Chapter 1.2: DEPENDABILITY EVALUATION METHODS		41
1.	Types of Dependability Evaluation Methods	41
2.	Dependability Evaluation by Analysis	42
3.	Dependability Evaluation by Field Experience	45
4.	Dependability Evaluation by Fault Injection Testing.....	46
5.	Conclusion and outlook.....	47
Chapter 1.3: SOFT ERRORS ON DIGITAL COMPONENTS		49
1.	Introduction	49
2.	Soft Errors	51
2.1	Radiation Effects (SEU, SEE).....	51
2.2	SER measurement and testing	53
2.3	SEU and technology scaling.....	54

- 2.3.1 Trends in DRAMs, SRAMs and FLASHs.....54
- 2.3.2 Trends in Combinational Logic and
Microprocessor 55
- 2.3.3 Trends in FPGA..... 55
- 2.4 Other sources of Soft Errors 56
- 3. Protection Against Soft Errors..... 57
 - 3.1 Soft Error avoidance.....57
 - 3.2 Soft Error removal and forecasting57
 - 3.3 Soft Error tolerance and evasion 58
 - 3.4 SOC Soft Error tolerance.....58
- 4. Conclusions 59

PART 2: HARDWARE-IMPLEMENTED FAULT INJECTION.....61

Chapter 2.1: PIN-LEVEL HARDWARE FAULT INJECTION
TECHNIQUES 63

- 1. Introduction 63
- 2. State of the Art..... 64
 - 2.1 Fault injection methodology 64
 - 2.1.1 Fault injection 64
 - 2.1.2 Data acquisition 65
 - 2.1.3 Data processing.....65
 - 2.2 Pin-level fault injection techniques and tools.....65
- 3. The Pin Level FI FARM model.....66
 - 3.1 Fault model set67
 - 3.2 Activation set.....67
 - 3.3 Readouts Set67
 - 3.4 Measures set68
- 4. Description of the Fault Injection Tool 68
 - 4.1 AFIT – Advanced Fault Injection Tool 68
 - 4.2 The injection process: A case study 73
 - 4.2.1 System Description..... 73
 - 4.2.2 The injection campaign 74
 - 4.2.3 Execution time and overhead 77
- 5. Critical Analysis..... 78

Chapter 2.2: DEVELOPMENT OF A HYBRID FAULT INJECTION
ENVIRONMENT..... 81

- 1. Dependability Testing and Evaluation of Railway Control
Systems..... 81
- 2. Birth of a Validation Environment 82
- 3. The Evolution of “LIVE” 86

3.1 Two examples of automation	88
4. Example application	92
5. Conclusions	93
Chapter 2.3: HEAVY ION INDUCED SEE IN SRAM BASED FPGAS.....	95
1. Introduction	95
2. Experimental Set Up.....	96
3. SEEs in FPGAs.....	99
3.1 SEU and SEFI.....	99
3.2 Supply current increase: SEL?	103
3.3 SEU in the configuration memory.....	106
4. Conclusions	107
PART 3: SOFTWARE-IMPLEMENTED FAULT INJECTION	109
Chapter 3.1: “BOND”: AN AGENTS-BASED FAULT INJECTOR FOR WINDOWS NT	111
1. The target platform	111
2. Interposition Agents and Fault Injection	112
3. The BOND Tool.....	113
3.1 General Architecture: the Multithreaded Injection.....	114
3.2 The Logger Agent.....	115
3.2.1 Fault Injection Activation Event.....	115
3.2.2 Fault Effect Observation.....	117
4. The Fault Injection Agent.....	117
4.1 Fault location	117
4.2 Fault type	118
4.3 Fault duration.....	119
4.4 The Graphical User Interface	119
5. Experimental Evaluation of BOND.....	120
5.1 Winzip32	121
5.2 Floating Point Benchmark.....	122
6. Conclusions	123
Chapter 3.2: XCEPTION™ : A SOFTWARE IMPLEMENTED FAULT INJECTION TOOL	125
1. Introduction	125
2. The Xception Technique	126
2.1 The FARM model in Xception.....	127
2.1.1 Faults	127
2.1.2 Activations.....	128

2.1.3	Readouts	129
2.1.4	Measures	129
3.	The XCEPTION TOOLSET	129
3.1	Architecture and key features	130
3.1.1	The Experiment Manager Environment (EME)	131
3.1.2	On the target side.....	131
3.1.3	Monitoring capabilities.....	132
3.1.4	Designed for portability.....	133
3.2	Extended Xception	133
3.3	Fault definition made easy.....	134
3.4	Xtract – the analysis tool	134
3.5	Xception™ on the field – a selected case study	135
3.5.1	Experimental setup	136
3.5.2	Results	136
4.	Critical Analysis	138
4.1	Deployment and development time	138
4.2	Technical limitations of SWIFI and Xception.....	138

Chapter 3.3: MAFALDA: A SERIES OF PROTOTYPE TOOLS
FOR THE ASSESSMENT OF REAL TIME COTS
MICROKERNEL-BASED SYSTEMS..... 141

1.	Introduction	141
2.	Overall Structure of MAFALDA-RT	143
3.	Fault Injection.....	145
3.1	Fault models and SWIFI.....	146
3.2	Coping with the temporal intrusiveness of SWIFI	147
4.	Workload and Activation.....	149
4.1	Synthetic workload.....	149
4.2	Real time application.....	150
5.	Readouts and Measures	151
5.1	Assessment of the behavior in presence of faults	151
5.2	Targeting different microkernels	153
6.	Lessons Learnt and Perspectives	155

PART 4: SIMULATION-BASED FAULT INJECTION 157

Chapter 4.1: VHDL SIMULATION-BASED FAULT INJECTION
TECHNIQUES 159

1.	Introduction	159
2.	VHDL Simulation-Based Fault Injection	160
2.1	Simulator Commands Technique	161
2.2	Modifying the VHDL Model.....	162

2.2.1	<i>Saboteurs</i> Technique	162
2.2.2	Mutants Technique	164
2.3	Other Techniques.....	167
3.	Fault Models.....	167
4.	Description of VFIT	168
4.1	General Features.....	168
4.2	Injection Phases	169
4.3	Block diagram	170
5.	Experiments of Fault Injection: Validation of a Fault Tolerant Microcomputer System	173
6.	Conclusions	176
Chapter 4.2: MEFISTO: A SERIES OF PROTOTYPE TOOLS FOR FAULT INJECTION INTO VHDL MODELS		
		177
1.	Introduction	177
2.	MEFISTO-L	178
2.1	Structure of the Tool.....	179
2.2	The Fault Attribute	181
2.3	The Activation Attribute.....	182
2.4	The Readouts and Measures	183
2.5	Application of MEFISTO-L for Testing FTMs.....	184
3.	MEFISTO-C	185
3.1	Structure of the Tool.....	185
3.2	Reducing the Cost of Error Coverage Estimation by Combining Experimental and Analytical Techniques.....	187
3.3	Using MEFISTO-C for Assessing Scan-Chain Implemented Fault Injection.....	189
4.	Some Lessons Learnt and Perspectives	191
Chapter 4.3: SIMULATION-BASED FAULT INJECTION AND TESTING UNSING THE MUTATION TECHNIQUE.....		
		195
1.	Fault Injection Technique: Mutation Testing	195
1.1	Introduction	195
1.2	Mutation Testing	196
1.3	Different mutations.....	199
1.3.1	Weak mutation.....	199
1.3.2	Firm mutation	200
1.3.3	Selective mutation	200
1.4	Test generation based on mutation	201
1.5	Functional testing method	203
1.5.1	Motivations.....	203
1.5.2	Mutation testing for hardware	203

- 2. The Alien Tool207
 - 2.1 The implementation tool.....207
 - 2.1.1 General presentation of the tool.....207
 - 2.1.2 ALIEN detailed description.....208
 - 2.2 Experimental work210
 - 2.2.1 Before enhancement of test data.....211
 - 2.2.2 After enhancement of test data212
 - 2.2.3 Comparison with the classical ATPGs212
- 3. Conclusion213
 - 3.1 Approach robustness213
 - 3.1.1 Robustness with regard to the different hardware implementations.....213
 - 3.1.2 Robustness with regard to the different hardware fault models214
 - 3.2 Limitations and Reusability.....214

Chapter 4.4: NEW ACCELERATION TECHNIQUES FOR
SIMULATION-BASED FAULT-INJECTION217

- 1. Introduction217
- 2. RT-Level Fault-Injection Campaign219
- 3. Fault Injection.....221
 - 3.1 Checkpoints and Snapshot.....221
 - 3.2 Early stop.....222
 - 3.3 Hyperactivity223
 - 3.4 Smart resume223
 - 3.5 Dynamic Equivalencies224
- 4. Workload Independent Fault Collapsing.....224
- 5. Workload Dependent Fault Collapsing225
- 6. Dynamic Fault Collapsing.....226
- 7. Experimental Results.....227
- 8. Conclusions229

References231

Contributing Authors

Joakim Aidemark, Chalmers Univ. of Technology, Göteborg, Sweden
Jean Arlat, LAAS-CNRS, Toulouse, France
Andrea Baldini, Politecnico di Torino, Torino, Italy
Juan Carlos Baraza, Università Polytecnica de Valencia, Spain
Marco Bellato, INFN, Padova, Italy
Alfredo Benso, Politecnico di Torino, Torino, Italy
Sara Blanc, Università Polytecnica de Valencia, Spain
J  rome Bou  , LAAS-CNRS, Toulouse, France
Joao Carreira, Critical Software SA, Coimbra, Portugal
Marco Ceschia, Universit   di Padova, Padova, Italy
Fulvio Corno, Politecnico di Torino, Torino, Italy
Diamantino Costa, Critical Software SA, Coimbra, Portugal
Yves Crouzet, LAAS-CNRS, Toulouse, France
Jean-Charles Fabre, LAAS-CNRS, Toulouse, France
Luis Entrena, Universidad Carlos III, Madrid, Spain
Peter Folkesson, Chalmers Univ. of Technology, G  teborg, Sweden
Daniel Gil, Universit   Polytecnica de Valencia, Spain
Pedro Joaqu  n Gil, Universit   Polytecnica de Valencia, Spain
Joaqu  n Gracia, Universit   Polytecnica de Valencia, Spain
Leonardo Impagliazzo, Ansaldo Segnalamento Ferroviario, Napoli, Italy
Eric Jenn, LAAS-CNRS, Toulouse, France
Barry W. Johnson, University of Virginia, VA, USA
Johan Karlsson, Chalmers Univ. of Technology, G  teborg, Sweden
Celia Lopez, Universidad Carlos III, Madrid, Spain
Tomislav Lovric, T  V InterTraffic GmbH, K  ln, Germany
Henrique Madeira, University of Coimbra, Portugal

Riccardo Mariani, Yogitech SpA, Pisa, Italy
Joakim Ohlsson, Chalmers Univ. of Technology, Göteborg, Sweden
Alessandro Paccagnella, Università di Padova, Padova, Italy
Fabio Massimo Poli, Ansaldo Segnalamento Ferroviario, Napoli, Italy
Paolo Prinetto, Politecnico di Torino, Torino, Italy
Marcus Rimén, Chalmers Univ. of Technology, Göteborg, Sweden
Chantal Robach, LCIS-ESISAR, Valence, France
Manuel Rodríguez, LAAS-CNRS, Toulouse, France
Frédéric Salles, LAAS-CNRS, Toulouse, France
Mathieu Scholive, LCIS-ESISAR, Valence, France
Juan José Serrano, Università Polytecnica de Valencia, Spain
Joao Gabriel Silva, University of Coimbra, Portugal
Matteo Sonza Reorda, Politecnico di Torino, Torino, Italy
Giovanni Squillero, Politecnico di Torino, Torino, Italy
Yangyang Yu, Univ. of Virginia, VA, USA

Preface

The use of digital systems pervades all areas of our lives, from common house appliances such as microwave ovens and washing machines, to complex applications like automotive, transportations, and medical control systems. These digital systems provide higher productivity and greater flexibility, but it is also accepted that they cannot be fault-free. Some faults may be attributed to inaccuracy during the development, while others can stem from external causes such as production process defects or environmental stress. Moreover, as devices geometry decreases and clock frequencies increase, the incidence of transient errors increases, and consequently, the dependability of the systems decreases. High reliability is therefore a requirement for every digital system whose correct functionality is connected to human safety or economic investments.

In this context, the evaluation of the dependability of a system plays a critical role. Unlike performance, dependability cannot be evaluated using benchmark programs and standard test methodologies, but only observing the system behavior after the appearance of a fault. However, since the Mean-Time-Between-Failures (MTBF) in a dependable system can be of the order of years, the fault occurrence has to be artificially accelerated in order to analyze the system reaction to a fault, without waiting for its natural appearance.

Fault Injection emerged as a viable solution, and it has been deeply investigated and exploited by both academia and industry. Different techniques have been proposed and used to perform experiments. They can be grouped in *Hardware-implemented*, *Software-implemented*, and *Simulation-based* Fault Injection.

The process of setting up a Fault Injection environment requires different choices that can deeply influence the coherency and the meaningfulness of the final results. In this book we tried to collect some of the most significant contributions in the field of Fault Injection. The selection process has been very difficult, with the result that a lot of excellent works had to be left out. The criteria we used to select the contributing authors were based on the innovation of the proposed solution, on the historical significance of their work, and also on an effort to give the readers a global overview of the different problems and techniques that can be applied to setup a Fault Injection experiment.

The book is therefore organized in four different parts. The first part is more general, and motivates the use of Fault Injection techniques. The other three parts cover Hardware-based, Software-implemented, and Simulation-based Fault Injection techniques, respectively. In each of these parts three Fault Injection methodologies and related tools are presented and discussed. The last chapter of Part 4 discusses possible solutions to speed-up Simulation-based Fault Injection experiments, but the main guidelines highlighted in the chapter can be applicable to other Fault Injection techniques as well.

Alfredo Benso
alfredo.benso@polito.it

Paolo Prinetto
paolo.prinetto@polito.it

Acknowledgments

The editors would like to thank all the contributing authors for their patience in meeting our deadlines and requirements. We are also in debt with Giorgio Di Natale, Stefano Di Carlo and Chiara Bessone for their valuable help in the tricky task of preparing the camera ready of this book.