52. IWK

Internationales Wissenschaftliches Kolloquium

International Scientific Colloquium

PROCCEDINGS | **10 - 13 September 2007**

FACULTY OF COMPUTER SCIENCE AND AUTOMATION

COMPUTER SCIENCE MEETS AUTOMATION

VOLUME II

Session 6 - Environmental Systems: Management and Optimisation

Session 7 - New Methods and Technologies for Medicine and Biology

Session 8 - Embedded System Design and Application

Session 9 - Image Processing, Image Analysis and Computer Vision

Session 10 - Mobile Communications

Session 11 - Education in Computer Science and Automation

TECHNISCHE UNIVERSITÄT
ILMENAU

Bibliografische Information der Deutschen Bibliothek
Die Deutsche Bibliothek verzeichnet diese Publikation in der deutschen Nationalbiografie; detaillierte bibliografische Daten sind im Internet über **http://dnb.ddb.de** abrufbar.

ISBN 978-3-939473-17-6

Impressum

Herausgeber: Der Rektor der Technischen Universität Ilmenau
Univ.-Prof. Dr. rer. nat. habil. Peter Scharff

Redaktion: Referat Marketing und Studentische Angelegenheiten
Kongressorganisation
Andrea Schneider
Tel.: +49 3677 69-2520
Fax: +49 3677 69-1743
e-mail: kongressorganisation@tu-ilmenau.de

Redaktionsschluss: Juli 2007

Verlag:
Technische Universität Ilmenau/Universitätsbibliothek
Universitätsverlag Ilmenau
Postfach 10 05 65
98684 Ilmenau
www.tu-ilmenau.de/universitaetsverlag

Herstellung und Auslieferung: Verlagshaus Monsenstein und Vannerdat OHG
Am Hawerkamp 31
48155 Münster
www.mv-verlag.de

Layout Cover: www.cey-x.de

Bezugsmöglichkeiten: Universitätsbibliothek der TU Ilmenau
Tel.: +49 3677 69-4615
Fax: +49 3677 69-4602

Preface

Dear Participants,

Confronted with the ever-increasing complexity of technical processes and the growing demands on their efficiency, security and flexibility, the scientific world needs to establish new methods of engineering design and new methods of systems operation. The factors likely to affect the design of the smart systems of the future will doubtless include the following:

- As computational costs decrease, it will be possible to apply more complex algorithms, even in real time. These algorithms will take into account system nonlinearities or provide online optimisation of the system's performance.
- New fields of application will be addressed. Interest is now being expressed, beyond that in "classical" technical systems and processes, in environmental systems or medical and bioengineering applications.
- The boundaries between software and hardware design are being eroded. New design methods will include co-design of software and hardware and even of sensor and actuator components.
- Automation will not only replace human operators but will assist, support and supervise humans so that their work is safe and even more effective.
- Networked systems or swarms will be crucial, requiring improvement of the communication within them and study of how their behaviour can be made globally consistent.
- The issues of security and safety, not only during the operation of systems but also in the course of their design, will continue to increase in importance.

The title "Computer Science meets Automation", borne by the 52nd International Scientific Colloquium (IWK) at the Technische Universität Ilmenau, Germany, expresses the desire of scientists and engineers to rise to these challenges, cooperating closely on innovative methods in the two disciplines of computer science and automation.

The IWK has a long tradition going back as far as 1953. In the years before 1989, a major function of the colloquium was to bring together scientists from both sides of the Iron Curtain. Naturally, bonds were also deepened between the countries from the East. Today, the objective of the colloquium is still to bring researchers together. They come from the eastern and western member states of the European Union, and, indeed, from all over the world. All who wish to share their ideas on the points where "Computer Science meets Automation" are addressed by this colloquium at the Technische Universität Ilmenau.

All the University's Faculties have joined forces to ensure that nothing is left out. Control engineering, information science, cybernetics, communication technology and systems engineering – for all of these and their applications (ranging from biological systems to heavy engineering), the issues are being covered.

Together with all the organizers I should like to thank you for your contributions to the conference, ensuring, as they do, a most interesting colloquium programme of an interdisciplinary nature.
I am looking forward to an inspiring colloquium. It promises to be a fine platform for you to present your research, to address new concepts and to meet colleagues in Ilmenau.

Professor Peter Scharff
Rector, TU Ilmenau

Professor Christoph Ament
Head of Organisation

Table of Contents

CONTENTS

Page

6 Environmental Systems: Management and Optimisation

7 New Methods and Technologies for Medicine and Biology

8 Embedded System Design and Application

10 Mobile Communications

11 Education in Computer Science and Automation

52nd Internationales Wissenschaftliches Kolloquium
Technische Universität Ilmenau
10 – 13 September 2007

R.Ubar / G.Jervan / J.Raik / M.Jenihhin / P.Ellervee

Dependability Evaluation in Fault-Tolerant Systems with High-Level Decision Diagrams

INTRODUCTION

To achieve high dependability, systems today are often designed with fault tolerance features to first detect errors and then to mask or recover from the effects of those errors. Thus, testing of these features is extremely important in understanding how dependable the systems are with the incorporated fault tolerance mechanisms and in gaining insight into the success of error detection and recovery. Fault injection is a means to effectively test and stress the error handling and fault tolerance mechanisms, so that the system behavior can be studied prior to their actual deployment.

Fault injection techniques can be classified in three main categories [1]: physical (or hardware Implemented fault injection), software implemented and simulation-based. Simulation-based fault injection is a useful experimental way to evaluate the dependability of a system during the design phase.

Recently many simulation based fault injection techniques have been proposed for system dependability evaluation [2-6]. They are targeted for execution-based models of working systems. Simulating faults in a system model (e.g. based on hardware description language VHDL, Verilog, System C etc) assures high flexibility.

However, there exists a problem of selecting faults to be injected. Erroneous responses in a system, in many cases, do not necessarily lead to a failure at the application level, even when the discrepancy with the nominal behavior has a long duration. An accurate but high-level fault analysis in the complete system is therefore required to discriminate real failure conditions from non-critical errors. Such an analysis is very difficult to carry out on the execution-based models using languages like VHDL, Verilog, System C. In this paper a method is proposed based on high-level modelling of systems with Decision Diagrams (DD) to select faults for injection targeting dependability evaluation.

MALICIOUS FAULTS AND FAULT INJECTION

Traditionally in simulation-based fault injection techniques, the fault location, fault type and fault insertion time are typically selected at random. The drawback of randomly selected faults is the large probability that the injected fault will remain latent; that is, produce a no response fault injection experiment. This will reduce the quality of dependability evaluation procedure. Thus, there is a need to locate faults which do not belong to the no response category.

The goal of the fault injection experiment is to exercise the system's fault processing capabilities. Faults which fail a system in the absence of system fault detection capabilities are defined to be malicious [8,9]. Malicious faults systematically exercise the fault processing attributes of the system. A malicious fault if undetected will fail the system under test. Likewise, a malicious fault is guaranteed to produce an error which will produce a failure if it is not properly processed by the system. Thus, using malicious faults to estimate the dependability of the system eliminates the possibility of the no response fault injection experiment.

Malicious fault list can be generated by creating fault trees using fault dependency analysis by reverse implication and providing fault collapsing. To cope with the complexity of fault analysis when carried out at plain low (logic) levels, we propose a hierarchical approach where the fault tree creation and fault collapsing will be carried out first at a higher level, thereafter to refine the fault injection points at a lower level.

To carry out the fault dependency analysis and creation of malicious fault lists we propose to use the method of diagnostic modelling digital systems by decision diagrams (DD) [10-13].

MODELLING DIGITAL SYSTEMS WITH DECISION DIAGRAMS

DD-s allowe to investigate and solve the problems of fault dependency analysis at different abstraction levels of digital systems depending on the complexity of the system description. The well known Binary Decision Diagrams (BDD) [10] can be considered as a particular case of this model for using at logic level.

For lower (logic) level fault analysis we will use a special class of BDDs – structurally synthesized BDDs (SSBDD) [11] to represent the topology of digital circuits in terms of signal paths at the fanout free region (or macro) level. Macro level fault analysis can be carried out more efficiently than traditional gate level analysis [13].

In high-level architectural descriptions, we usually partition the system into high level components or subsystems where the descriptions of subsystems in general case can be modeled by control and data paths. These paths can be represented by a set of DDs where the nonterminal nodes in DDs correspond to the control variables (instruction codes, addresses, control words, logic or timing conditions etc), and the terminal nodes correspond to data variables, functional expressions or more complex behavioral or algorithmic descriptions of other subystems. The terminal nodes will be represented again by a set of DDs to allow entering a lower hierarchical level for disclosing the label of the terminal node [12,13].

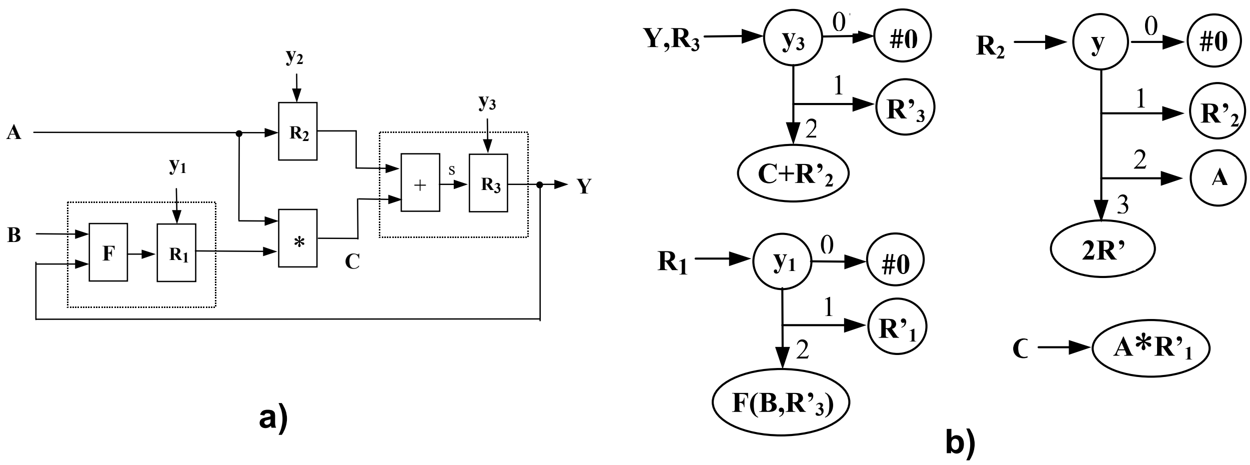

Fig.1. Simple data-path represented by DDs

In Fig.1 a simple register-transfer level data path consisting of 4 subcomponents (three registers with input logic and a multiplier) each represented by its DD is depicted. Each graph has a single nonterminal node labelled by a control variable whereas the terminal nodes are representing reset, hold, load and data manipulation (addition, shift, multiplication and a nondisclosed F(B,R'$_3$)) functions. The values on edges correspond to the values of control signals. The apostroph shows that the value of the register variable is taken from the previous clock cycle.

Each path in a DD describes the behavior of the subsystem represented by DD in a specific mode of operation (working mode). For example, the path from the node y_3 to the node C + R'$_2$ in the graph R'$_3$ represent the addition operation C + R'$_2$ activated by the control signal y_3 =2.

The faults having effect on the operation associated with activaqted path can be related to the nodes along the path. A fault may cause incorrect leaving of the activated path. From this point of view we can introduce a very general fault model for DDs related to faulty behavior of nodes. In the case of SSBDDs the faults of a binary node are equivalent to the traditional logic level stuck-at-0(1) fault model.

In more general case of DDs the faults of the nonterminal nodes can be interpreted as the addressing fault model of microprocessor cores [13]: output edge is either always activated (stuck) or always broken, or instead of the given edge, another edge or a set of edges is activated. This is a very general fault model that allows to represent most of control errors in the case of microprocessor cores.

It is easy to extend the described fault model of a DD node to model complex faults like shorts, delays, crosstalks etc. All of these faults can be represented by a checkpoint where the error manifests at certain conditions i.e. by a pair (fault site; fault condition). Since the nodes of DDs can be regarded as checkpoints the fault model on DDs can be represented as a pair (node; fault condition). In such a way the fault model on DDs can be seen as a very powerful model able to cover a very wide class of faults in digital systems.

The formalism of DDs allows to implement simple algorithms to analize the cause-effect and fault dependency relationships which would be impossible in case of descriptions written in languages like VHDL, Verilog, System C.

MALICIOUS FAULT LIST GENERATION WITH DECISION DIAGRAMS

Malicious faults will be generated for a given set of test sequences. A test activates a process in a system consisting of a sequence of iterations. Each iteration (clock cycle, microinstruction or instruction cycle, transaction etc.) activates paths in DDs of the corresponding DD-model of the system. The faults having effect on the behavior of the system at a given iteration can be related to the nodes of the activated path.

For each observable checkpoint C_i of a test sequence, a Fault Tree (FT) with a root N_i will be created. Assuming an erroneous signal is detected at C_i, a set of candidate faulty checkpoints C_k explaining the misbehavior in C_i is created by tracing the activated path on the correponding DD for C_i. This set of checkpoints C_k will be included into FT as successors of N_k. For each node N_k in FT, a list of malicious faults associated with C_k is determined. For all leaf nodes in FT to be constructed the same procedure is repeated until the input events for the test sequence are reached. If C_k represents stored data (register variables), we will continue fault dependency analysis in the previous test iteration. Each node in FT characterised with a list of malicious faults and the number of iteration can be used as a point for fault injection. To reduce the number of malicious faults, the FT can be compressed (collapsed) by using fault equivalence and fault dominance relationships. The fault model is not restricted by the approach. In general, arbitrary changes R + ε of the values of a variable R can be accepted. The set of accepted values of ε can be given by a fault list.

An example of FT created for the data path in Fig.1 and for the test sequence in Table 1 is shown in Fig. 2. The malicious fault list selected from FT is depicted in Table 2.

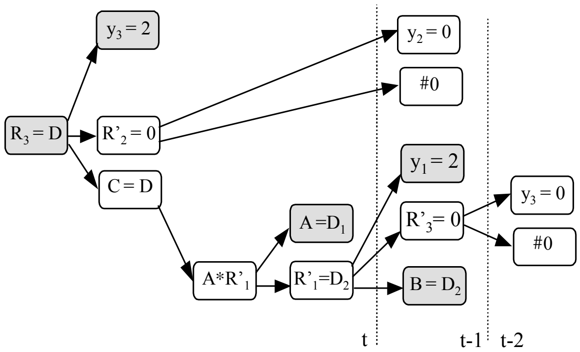

Fig.2. Fault tree

t	y_1	y_2	y_3	A	B	C	R_1	R_2	R_3	Y
1			**0**						0	
2	**2**	**0**			$\mathbf{D_2}$		D_2	0		
3			**2**	$\mathbf{D_1}$		D			D	**D**

Table 1. Test sequence

t	y_1	y_3	A	B	R_2	Y
2	0			$D_2 + \varepsilon_2$	$\varepsilon_3 \neq 0$	
3		0,1	$D_1 + \varepsilon_1$			$\mathbf{D = \varepsilon}$

Table 2. Malicious fault list

CONCLUSIONS

A new approach based on fault dependency analysis with Decision Diagrams is proposed for selecting faults to be injected targeting fault tolerance evaluation in digital systems. Such a fault analysis is impossible when the execution-based models written in VHDL, Verilog or System C are used. The proposed method allows to avoid no response fault injection experiments and as the result to increase the quality of dependability evaluation.

Acknowledgement: The work has been supported by Estonian Science Foundation grants 5910, 6717, 6829 7068, EC 6th framework IST project VERTIGO, Estonian Information Technology Foundation (EITSA) and Enterprise Estonia.

References:

[1] A.Benso and P.Prinetto, eds. Fault Injection Techniques and Tools for VLSI reliability evaluation. Kluwer Academic Publishers, 2003.

[2] E.Jenn, J.Arlat, M.Rimén, J.Ohlsson, J.Karlsson, “Fault injection into VHDL models: the MEFISTO tool”, in Proceedings 24th International Symposium on Fault-Tolerant Computing (FTCS-24), pp. 356-363, Austin, TX, USA, 1994.

[3] V.Sieh, O.Tschäche, F.Balbach, “VERIFY: Evaluation of Reliability Using VHDL-Models with Embedded Fault Descriptions”, in Proceedings 27th International Symposium on Fault Tolerant Computing (FTCS-27), pp. 32-36, Seattle, WA, USA, 1997

[4] D.Gil, J.Gracia, J.C.Baraza, P.J.Gil, “A Study of the effects of Transient Fault Injection into the VHDL Model of a Fault-Tolerant Microcomputer System”, in Proceedings 6th IEEE International On- Line Testing Workshop (IOLTW'2000), pp. 73-79, Palma de Mallorca, Spain, 2000.
[5] R. Leveugle and K. Hadjiat, “Multi-Level Fault Injections in VHDL Descriptions: Alternative Approaches and Experiments,” J. Electronic Testing: Theory and Applications (JETTA), vol. 19, no. 5, pp. 559-575, Oct. 2003.
[6] J.C. Baraza, J. Gracia, D. Gil and P.J. Gil. Improvement of Fault Injection Techniques Based on VHDL Code Modification. 10th IEEE International High-Level Design Validation and Test Workshop, 30 Nov.-2 Dec. 2005 Page(s):19 – 26.
[7] J.Sosnowski, P.Gawkowski, P.Zygulski, A.Tymoczko. Enhancing Fault Injection Testbench. Proceedings of the International Conference on Dependability of Computer Systems (DEPCOS-RELCOMEX’06), May 2006 Page(s):76 – 83.
[8] D.T.Smith, B.W.Johnson, J.A.Profeta. System Dependability Evaluation via a Fault List Generation Algorithm. IEEE TRANSACTIONS ON COMPUTERS, VOL. 45, NO. 8, AUGUST 1996, pp.974-979.
[9] A.Benso, P.Prinetto, M.Rebaudengo, M.Sonza, R.Ubar. A New Approach to Build a Low-Level Malicious Fault List Starting from High-Level Description and Alternative Graphs. Proc. IEEE European Design & Test Conference, Paris, March 17-20, 1997, pp.560-565.
[10] R.E.Bryant. Graph-based algorithms for Boolean function manipulation. IEEE Trans. on Computers, Vol.C-35, No8, 1986, pp.667-690.
[11] A.Jutman, A.Peder, J.Raik, M.Tombak, R.Ubar. Structurally Synthesized Binary Decision Diagrams. 6th International Workshop on Boolean Problems, Freiberg, Germany, Sept. 2004, pp.271-278.
[12] G.Jervan, R.Ubar, Z.Peng, P.Eles. Test Generation: A Hierarchical Approach. In “System-level Test and Validation of Hardware/Software Systems” by M.Sonza Reorda, Z.Peng, M.Violante. Springer Series in Advanced Microelectronics, Vol.17, 2005, pp. 63-77.
[13] O.Novak, E.Gramatova, R.Ubar. Handbook of Electronic Testing. CTU Printhouse, Prague, 2005, 400 p.

Authors:
Prof. Raimund Ubar
Dr. Gert Jervan
Dr. Jaan Raik
Mr. Maksim Jenihhin
Prof. Peeter Ellervee
Tallinn University of Technology, Raja 15
12617, Tallinn, ESTONIA
Phone: +372 6202252
Fax: +372 6202253
E-mail: {raiub, gerje, jaan, maksim}@pld.ttu.ee, lrv@cc.ttu.ee