POLITECNICO DI TORINO
Repository ISTITUZIONALE

Interactive Educational Tool for Memory Testing

Original
Interactive Educational Tool for Memory Testing / Bosio A.; Di Carlo S.; Di Natale G.; Fisherova M.; Pikula T.; Simlastik M.. - STAMPA. - (2006), pp. 100-103. ((Intervento presentato al convegno 6th International Workshop on Microelectronic Education tenutosi a Stockholm, SE nel 8-9 Jun. 2006.

Availability:
This version is available at: 11583/1499979 since:

Publisher:
KTH Royal Institute of Technology

Published
DOI:

Terms of use:
openAccess

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright
default_conf_draft

-

(Article begins on next page)

05 August 2020

Politecnico di Torino

Interactive Educational Tool for Memory Testing

Authors: Bosio A., Di Carlo S., Di Natale G., Fisherova M., Pikula T., Simlastik M.,

Published in the Proceedings of the 6th International Workshop on Microelectronic Education (EWME), 8-9 Jun. 2006, Stockholm, SE.

N.B. This is a copy of the ACCEPTED version of the manuscript. The final PUBLISHED manuscript is available at the KTH Library

URL: http://www.kth.se/en/kthb

© 2000 KTH. Personal use of this material is permitted. Permission from KTH must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

INTERACTIVE EDUCATIONAL TOOL FOR MEMORY TESTING

ALBERTO BOSIO[1], STEFANO DI CARLO[1], GIORGIO DI NATALE[1],
MÁRIA FISCHEROVÁ[2], TOMÁŠ PIKULA[2], MARTIN ŠIMLAŠTÍK[2]
[1]*Politecnico di Torino, Corso duca degli Abruzzi 24, I-10129 Torino, Italy*
[2]*Institute of Informatics, Dúbravská cesta 9, 845 07 Bratislava, Slovakia*

1. INTRODUCTION

Memories are one of the most important components in digital systems like SoCs. The high density of their cell array makes memories extremely vulnerable to physical defects. Hence, memory testing and Design-for-Test became one of the crucial tasks in the design of complex and heterogeneous SoCs.
Politecnico di Torino and the Institute of Informatics have a wide experience in the field of RAM testing (i.e., automatic march test generation, fault simulators, memory BIST generators etc.). This work is a tentative to put the joint experience of our research groups in developing an interactive educational tool for the students that should introduce standard and well-known methods of memory testing based on BIST.
The MemBIST Java Applet and the March Test Generator were two individual tools designed and implemented at the two mentioned institutions. They were merged into one tool in order to facilitate its usage also by the professionals.

2. MEMBIST JAVA APPLET

Memory testing needs special algorithms to generate the required memory test patterns. There are many algorithms (e.g., Zero-one, Walking, Galloping, or Checkerboard patterns) but at present, mostly many types of march like algorithms are realistic to use in testing of bit or word oriented memories [1], [2]. The memory embedded into SoC is usually difficult to test because of its poor controllability and observability. The proper test solution is the usage of the BIST method - test algorithm and test response analysis are implemented on the chip. The MemBIST applet is a software tool that demonstrates principles of RAM testing and BIST architectures and generates the memory BIST (MBIST) structure for a given memory [3]. It consists of learning and generation modules.

2.1 LEARNING MODULE

The learning module deals with the explanation of the BIST method for RAMs, presents its specific structure and functionality by an interactive animation. It is divided into two parts – Learning and Exercise.
In the Learning part, the principles and components of MBIST are explained. It starts with explanations of the memory model and the most common fault models in memories (Figure 1). The next step is setting the address, data and control multiplexer's parameters. The control unit is divided into the March C- algorithm

Manuscript ACCEPTED version copy

and an address generator. The March C- algorithm can be implemented as a finite state machine (FSM). As the address generator, the linear feedback shift register (LFSR) was chosen, due to its easy hardware implementation. The user can define the characteristic polynomial and the initial value of LFSR and observe the generated patterns. The last explained component of MBIST is a comparator of fault-free value with the output of the memory.
The Exercise part allows the user to define memory parameters – number of rows, number of columns and memory cell bit-width. After defining the memory, the user can inject faults into the memory. The user can then define characteristic polynomial and initial value of LFSR. The last step is the simulation of the configured memory with injected faults. It can run automatically, or the user can manually control the simulation steps. The simulation can be configured to stop after the first detected fault or can be configured to run until the end to observe all detected faults.

2.2 GENERATION MODULE

The generation module allows generating the VHDL description of a memory BIST architecture starting from a set of parameters defined by the student. The parameters are the size of the memory and its memory cells, the characteristic polynomial and seed for LFSR, and the type of faults the march test has to cover. The user selects from the list of classic fault models typical for memories or defines own fault models using the fault primitive's formalism [4]. The march test covering selected faults is generated by the March Test Generator [5] (Figure 2).

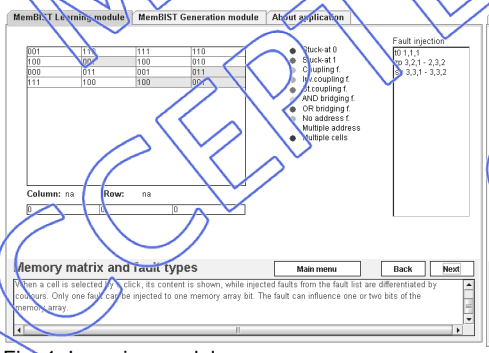

Fig. 1. Learning module.

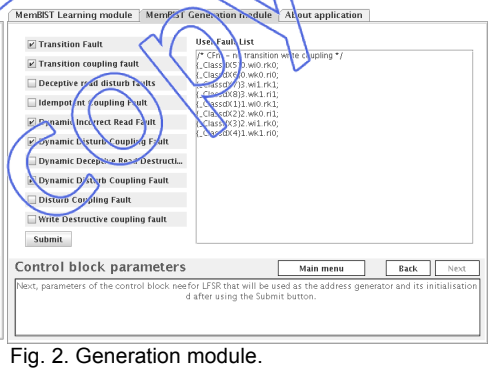

Fig. 2. Generation module.

The generator can build the BIST circuit for single port memories of arbitrary size (preferably the size of 2^N). As the BIST architecture is based on shifting, the tested memory can even be word-wide [1].
The MBIST architecture contains a type-1 (external-XOR) LFSR which is responsible for the address generation. This type of LFSR was chosen due to the possibility of generating the maximum-length address sequence including the all-zero pattern and its reverse ordered sequence. The maximum-length sequence depends on the selection of the characteristic polynomial which must be primitive.

Manuscript
ACCEPTED version

All set parameters are considered in the generation of the BIST circuit, especially its FSM. The results of the March Test Generator directly influence the VHDL description of FSM within the memory BIST (Figure 3). The design of MBIST was optimized for speed and area. The resulting VHDL is fully synthesizable in commercial or freely available design tools and can be used as an example implementation of memory BISTs in the educational process.

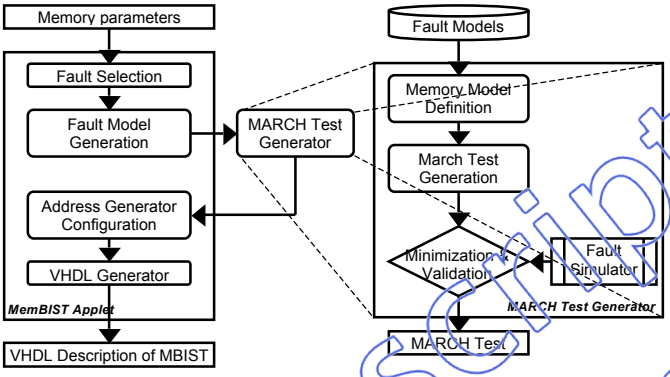

Fig. 3. Design flow of the tool.

3. MARCH TEST GENERATOR

The March Test Generator module is able to generate march tests starting from a user defined list of faults. The march tests are particularly simple memory test algorithms that use the regular structure of SRAMs to reduce the test complexity [6]. Several march tests targeting different set of memory faults have been proposed [7]. Most of them have been generated by hand, but with the occurrence of new and more complex fault models, the task of hand writing test algorithms is becoming harder and it may lead to non-optimal results.

The march test generation process starts from the definition of the list of faults to be tested. Besides classic models, user defined faults expressed in terms of fault primitives [4] are supported. The March Test Generator is able to deal with:

1. Static and Dynamic Faults
2. Linked and Unlinked faults
3. Single and multiple port memories

Given the list of faults to be tested, the March Test Generator is able to generate a non-redundant march test covering the selected faults. Each generated march test is then fault simulated to check both its coverage and to eventually optimize the final test. Using the March Test Generator students can become familiar with one of the most used memory test algorithm in the industry.

4. CONCLUSION

The MemBIST applet has been utilized in the educational process at the Faculty of Informatics and Information Technologies of the Slovak University of

Manuscript ACCEPTED version copy

Technology. It has been regularly used for practical exercises in testing area as new educational concepts at the lab works in the basic course Diagnostics and Reliability of Digital Systems for undergraduate students, in the advanced course Testing of Digital Systems for graduate students and in diploma works. The Web-based applet simulates the learning subject in a well illustrative graphical form that is self-explanatory, takes the advantage of learning by doing and involves interaction possibilities. Using such tools makes the course more attractive to the students. Students' opinions, remarks and suggestions have been gathered and analyzed in order to improve the MemBIST modules.

In a similar way the same tool has been used at Politecnico di Torino during lab sessions of the course Digital Systems Dependability for master students of Electronics and Computer Science Engineering. By comparing the interest of students with regard to the previous editions of the same course not using the MemBIST applet, more interest gained from the possibility of applying theoretical notion explained during lectures on real test cases.

Since the tool is freely accessible on Internet [8] (MemBIST is the part of a testability tool set [9]), students and teachers from other technical universities are also encouraged to exploit the modules in the teaching and learning process.

6. AKNOWLEDGEMENT

The presented work has been supported by the Slovak-Italian Science and Technology Co-operation project "Testable and Reconfigurable Digital Cores".

7. REFERENCES

[1] A.J. van de Goor, "Testing Semiconductor Memories, Theory and Practice", ComTex Publishing, Gouda, The Netherlands, 1998.

[2] R.D. Adams, "High Performance Memory Testing. Design Principles, Fault Modeling and Self-Test," Kluwer Academic Publishers, Bostom / Dordrecht / London, 2003.

[3] M. Fischerová, M. Šimlaštík, "MemBIST applet for learning principles of memory testing and generating memory BIST", EuroMicro Conference on Digital System Design, August 30 - September 3, 2005, Porto, Portugal, pp. 276–279.

[4] A. J. van de Goor, Z. Al-Ars, "Functional Memory Faults: A Formal Notation and a Taxonomy", 18th IEEE VLSI Test Symposium, 2000, pp. 281-289.

[5] A. Benso, A. Bosio, S. Di Carlo, G. Di Natale, P. Prinetto, "Automatic March tests generation for static and dynamic faults in SRAMs", 10th IEEE European Test Symposium, May 22-25, 2005, Tallinn, Estonia, pp. 122–127.

[6] A. J. van de Goor, "Using March Tests to Test SRAMs", IEEE Design & Test of Computers, pp. 8–14, 1993.

[7] A.J. van de Goor, B. Smit, "Automatic verification of March Tests", IEEE International Workshop on Memory Technology, Design and Testing, 1993, pp.131–136.

[8] www.ui.sav.sk/diag/tools

[9] M. Baláž, T. Pikula, R. Lauko, M. Fischerová, E. Gramatová, "eLearning and eTraining Tools for Testability Techniques of Digital Circuits and Systems", 5th International Conference Virtual University, Bratislava, Slovak Republic, December 16-17, 2004, pp. 95-100.

Manuscript ACCEPTED version copy