ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING
Vol. 2, Issue 1, January 2014

GUI based complex test pattern generation for high speed fault diagnosis in memory chips

Mohd Arshad Nazeer[1], Imthiazunissa Begum M-Tech[2], Korani Ravinder Phd[3], Md Abdul Khader[4]

M-Tech H.O.D, ECE Department, VIFCET JNTU Hyderabad[2]

Asst. Professor, ECE Department, VIFCET, JNTU Hyderabad[3]

PG Scholar in VLSI Design, Department of E.C.E VIFCET,JNTU Hyderabad[1&4]

Abstract: The memory blocks testing is a separate testing procedure followed in VLSI testing. The memory blocks testing involve writing a specific bit sequences in the memory locations and reading them again. This type of test is called March test.

A particular March test consists of a sequence of writes followed by reads with increasing or decreasing address. For example the March C- test has the following test pattern. $\{\uparrow(w0);\uparrow(r0,w1);\uparrow(r1,w0);\downarrow(r0,w1);\downarrow(r1,w0);\uparrow(r0)\}$

There are several test circuits available for testing the memory chips. However no test setup is developed so far for testing the memory blocks inside the FPGA. The BRAM blocks of FPGA are designed to work at much higher frequency than the FPGA core logic. Hence testing the BRAMs at higher speed is essential. The conventional memory test circuits cannot be used for this purpose. Hence the proposed work develops a memory testing tool based on March tests for FPGA based BRAM (block RAM testing).

The code modules for March test generator shall be developed in VHDL and shall be synthesized for Xilinx Spartan 3 Family device. A PC based GUI t.ool shall send command to FPGA using serial port for selecting the type of test. The FPGA core gets the command through UART and performs the appropriate and sends the test report back to PC.

Generally, in the studies of microelectronics engineer the approach of IC testing remains very theoretical. Only few concrete practices are commonly done and generally laboratory experiences are limited to the use of CAD tools. For this purpose, in our teaching department, we develop an experiment allowing a concrete learning of IC testing dedicated to the test of commercial memory chips. Through this environment, our students reached a better knowledge of the connection between the test sequences and the detected faults.

I. INTRODUCTION

The cost of verification and test for nowadays circuits represents an important part of the total IC final price [1]. Hence, the domain of test represents a cornerstone for the industry and consequently for the academic research and education. Thus, in order to properly complete the education of microelectronics students, it is crucial to develop some basic skills in such domain.

Within the microelectronics department of the graduate engineering school of Polytech'Montpellier, the learning of ASIC testing is introduced through the basic notion of 'stuck at fault'. For this purpose, a series of lectures presents the main fault models encountered in modern technologies like stuck @, bridging, open… and the algorithms classically employed for test vectors generation (D-Algorithm, PODEM, etc.). The tangible aspects of the test are then studied through practical class by the use of the CAD tool for ATPG (Automatic Test Pattern Generation) TetraMAX, and the industrial ATE (Automatic Test Equipment) Verigy.

While the test of the ASICs represents an important part of the efforts made in the domain, there is a family of components which requires a particular attention: memories.

Indeed, the silicon area dedicated to memory elements is constantly growing in recent designs [1].

Memory testing strongly differs from the test of conventional ASICs. Consequently, we underline the necessity of introducing this subject into a curriculum of our engineers in microelectronics. The University of Turin (Italy) [2] proposes a tool for learning memory testing. This very interesting working environment remains however very abstract and virtual because the test memory is not a real one, i.e. it does not actually exist, but its function is emulated. Furthermore, this tool implements only a single test algorithm.

In this article, we present an original memory test framework: an SRAM memory test bench, roaming and programmable. This test bench allows not only to employ different commercial SRAM memories but also to apply various algorithms for test. With this new test bench, students can concretize the memories testing's lectures and enlighten the inherent properties of the various applied algorithms as well as the differences between the memory architectures and technologies.

II. VERSATILE MARCH TEST GENERATOR

Memory testing may be considered as a full disciplinary subject. Commonly, test sequences or test algorithms for memories are known under the name of March tests. Every March test has specific capabilities that allow

ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING
Vol. 2, Issue 1, January 2014

revealing the typical defects of memories [3]. A typical didactic test bench has to allow not only the implementation of March tests existing in the literature but also the creating of new test algorithms. A March test consists of a sequence of March elements. A March element has a certain number of operations (or March primitive) that must be applied to all memory cells of an array. Thus, ↑(r0;w1) is a March element and r0 and w1 are March primitives. The addressing order of a Marchelement can be done in an up (↑), down (↓) way or(↕) if theorder is not significant. A March primitive can be a write 1 (w1), write 0 (w0), read 1(r1) and read 0 (r0) that can be performed in a memory cell.

Here, we introduce, for example, the March C- :

{↑(w0);↑(r0,w1);↑(r1,w0);↓(r0,w1);↓(r1,w0);↑(r0)}

This well-known March test allows to detect all the stuck @ and transition fault of a memory cell array, as well as all address decoder faults and coupling (interaction between two cells) faults.

A way to create a function allowing the implementation of any March test is to use the structure described on table 1.

TABLE 1 : Input data

0	1	1	0														
1	0	0	1	0	0	X	X	X	X	X	X	X	X	X	X	X	X
1	0	1	0	1	0	0	1	X	X	X	X	X	X	X	X	X	X
1	0	1	0	1	1	0	0	X	X	X	X	X	X	X	X	X	X
0	0	1	0	1	0	0	1	X	X	X	X	X	X	X	X	X	X
0	0	1	0	1	1	0	0	X	X	X	X	X	X	X	X	X	X
1	0	0	1	1	0	X	X	X	X	X	X	X	X	X	X	X	X
...																	

A 274 bits register is needed to memorize the March test data. Table 1 gives for example the data-base implantation of the March test C-.

III.TEST BENCH ARCHITECTURE

Our test bench architecture for memories is composed of one computer, a versatile March test generator, a serial interface (for the communication between the programmable generator and the computer) and a deck containing 4 SRAM memories under test, Fig. 1.

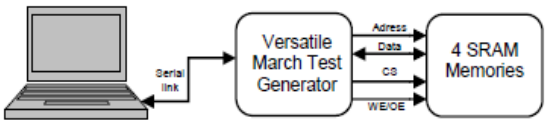

Fig. 1 : Test bench architecture

A user interface presented in Fig. 2, allows students to choose or set a specific March Test of the literature (March A, March C-, Matt, Matt +) or introduce a new one (Custom….) The chosen March test is uploaded through the serial connection to the programmable test generator and then applied to each memory on the deck.

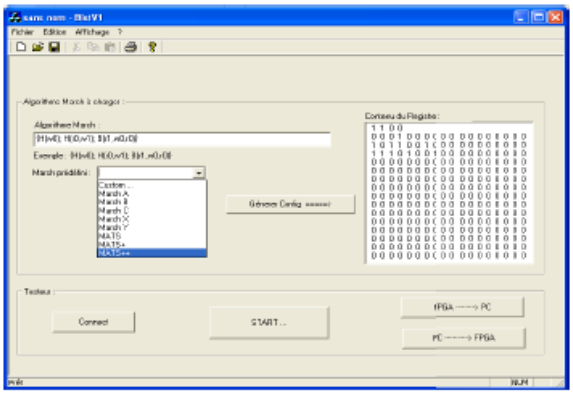

Fig. 2 : User interface

If no fault is detected, the programmable generator returns a positive acknowledgement on the four memories. Whether the opposite case occurs, i.e. when a reading operation (r0 or r1) does not return the expected data, the test bench returns the following data: the failing memory. Only the knowledge of this information allows the identification of physical defects beside the observed fault, or at least to make reasonable suppositions.

IV. LABORATORY

Fig. 3 depicts the test bench (with four memories under test) that is proposed to our students. A serial cable connects the test bench to a computer. The four memories (on the top of the picture) are tested in sequence (not simultaneously) following a scheduling that is fixed by the user.

Fig. 3 : Test bench (not connected to the computer)

One or more than one memory can be replaced by a memory emulator, in which we can introduce any king of fault model [4] such as stuck @, transition, address decoder, coupling faults, etc.

The use of the fault injection through the memory emulator is important to make the student anle to check the efficiency of March test algorithms to test specific fault models that may affect the memories. With the analysis of the test report the student obtains useful data to uncover the correlation between the detected fault and the sequence of read/write perations that allow the sensitization and observation of the fault itself. This analysis highly helps the student's knowledge of memory

ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING
Vol. 2, Issue 1, January 2014

failing processes as well as his skill to generate appropriate March test algorithms to target specific pull of faults.

V. CONCLUSION

The generation and refinement of this framework come from the observation that the teaching of the rest of the integrated circuits is too often approached in theoretical or virtual ways. We believe that, for our engineering students, especially those close to the conclusion of their studies, it would be important to have the opportunity to develop theoretical and practical skills to generate adequate test solutions for actual electronic devices. Although this Platforms has still to be improved, the early feedback of our students is very encouraging. In all cases, they have affirmed to have clearly understood the way March test sequences are applied to memories as well as the sensitization and observation processes of the various fault models. Moreover, they also showed to have highly perfected their knowledge of the memory architecture and function.

REFERENCES

[1] Semiconductor Industry Association (SIA), "International Technology Roadmap for Semiconductors (ITRS) ", 2005 edition.
[2] A. Borsio et al. "Interactive Educational Tool for Memory Testing " pp.100-103. 6th International Workshop on Microelectronics Education. 2006
[3] AJ. Van de Goor, "Testing Semiconductor Memories: Theory and Practice, " John Wiley and Sons, ISBN 0-471-9586-1, 1991
[4] P. Rech et al., " A Memory Fault Simulator for Radiation-Induced Effects in SRAMs" IEEE 19TH Asian Test Symposium. 2010.