OPEN ACCESS

International
Journal
Of Modern Engineering Research (IJMER)

Complex Test Pattern Generation for high speed fault diagnosis in FPGA based memory blocks

K.Jhansi[#1], V.Balaji[*2]

# M.Tech Student, ECE-Department, Vizb, Jntuk University Vishakapatnam, A.P, India
Asst.Proffessor & HOD, ECE-Department, Vizb, Jntuk University Vishakapatnam, A.P, India

***ABSTRACT**: The memory blocks testing is a separate testing procedure followed in VLSI testing. The memory block testing involves 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.

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 tool 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. The results shall be verified in simulation with Xilinx ISE simulator and also in hardware by using Chip scope. Xilinx Spartan 3 family FPGA board shall be used for hardware verification of the developed March test generator.

I. INTRODUCTION

The cost of verification and test for nowadays circuits represents an important part of the total IC final price. Hence, the domain of test represents a cornerstone for the industry. Recently, the advances in semiconductor memory technologies have become more complex resulting in a rapidly increasing transistor per memory design.

New design techniques enable a higher memory capacity implementation on a fixed die size. However, larger memory capacities require more extensive testing. Inevitably memory testing time increases the fault model to effectively model the variety of physical failures that could occur because of interference between closely packed cells. On the other hand, the more compact size will produces more defects during chip manufacturing, pushing yields down. Therefore,memory testing and repairing will be more important in the future memory chips. Testing and diagnosis techniques play a key role in the advancement of semiconductor memory technologies. The challenge in obtaining reliable failure detection has created intensive investigation on efficient testing and diagnosis algorithms for better fault coverage and diagnostic resolution.There are a number of test techniques that have been well studied. Many test algorithms were proposed based on functional-level fault models using manual technique analysis. Although March Test Algorithm (MTA) is one of the best solutions and widely use in testing memories, due to the technology advance there are more new fault models will be introduced. Therefore, to overcome this coming fault models a new technique is required to develop which can be automatically analyzed and added the new technique.

This paper aims to propose a new solution for researchers and engineers to find an efficient test and diagnosis algorithm in shorter time. A combination march-based test algorithm will be implemented for this purpose. Universities and industry involved in memory Built-in-Self test, Built-in- Self repair and Built-in-Self diagnose will benefit by saving a few years on research and development due to the fact that the manual and automatic test procedures developed in this work is compatible an expandable for BRAM memory testing.

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 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 March element can be done in an up (↑), down (↓) way or (↕) if the order 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.

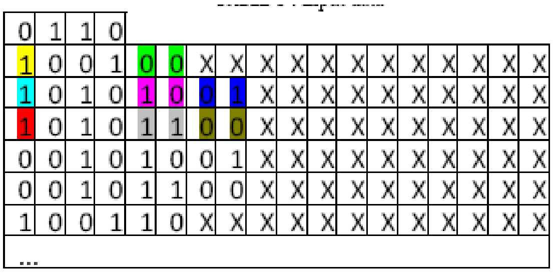
Here, we introduce, for example, the March C- :

{↑(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.



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 implementation of the March test C-.

III. THEORETICAL ANALYSIS

Field programmable gate arrays (FPGAs) are a popular choice among VLSI devices, any logical circuit can be implemented into the FPGA at low cost. It consists of an array of configurable logic blocks (CLBs), programmable interconnect and programmable Input/output blocks, Block Random Access Memories (BRAMs), a multiplier, a Digital Clock Manager. Many methods have been proposed to test FPGAs. In some works, the circuits under consideration are programmed FPGAs, in which logic circuits have been implemented. Since an FPGA can be programmed in many different ways, this method is not applicable to manufacturing time testing, as we do not know the final configuration. Testing faults in general FPGAs has been proposed by many researchers. In these methods, the FPGA under test is not mapped to a specific logic function. As a result, multiple test sessions are usually required, with each session dealing with one configuration.

For applications requiring large, on-chip memories, Spartan-3 FPGAs provides plentiful, efficient BRAMs. All Spartan-3 devices feature multiple block RAM memories, organized in columns. The total amount of block RAM memory depends on the size of the Spartan-3 device.

Spartan-3E devices incorporate 4 to 36 dedicated block RAMs, which are organized as dual-port configurable 18 Kbit blocks. Functionally, the block RAM is identical to the Spartan-3 architecture block RAM. Block RAM synchronously stores large amounts of data while distributed RAM, is better suited for buffering small amounts of data anywhere along signal paths. This section describes basic block RAM functions. Each block RAM is configurable by setting the content's initial values, default signal value of the output registers, port aspect ratios, and write modes. Block RAM can be used in single-port or dual-port modes.

Each block RAM contains 18,432 bits of fast static RAM, 16K bits of which is allocated to data storage and, in some memory configurations, an additional 2K bits allocated to parity or additional "plus" data bits. Physically, the block RAM memory has two completely independent access ports, labelled Port A and Port B. The structure is fully symmetrical, and both ports are interchangeable and both ports support data read and write operations. Each memory port is synchronous, with its own clock, clock enable, and write enable. Read operations are also synchronous and require a clock edge and clock enable.

IV. IMPLEMENTATION

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.

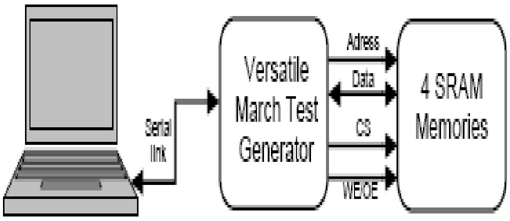


Fig. 1 : Test bench architecture

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.

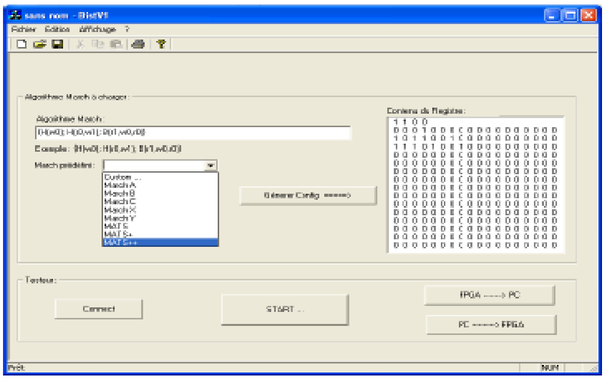


Fig. 2 : User interface

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, the failing address, the failing march element and operation. Only the knowledge of this information allows the identification of physical defects beside the observed fault, or at least to make reasonable suppositions.

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.

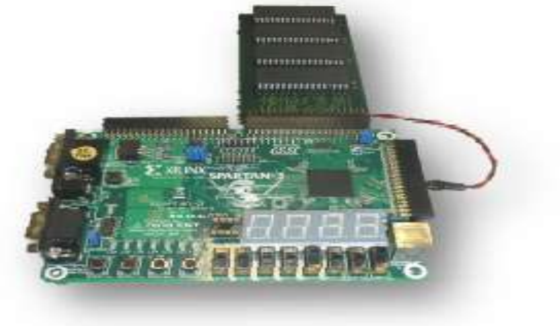


Fig. 3 : Test bench (not conntected to the computer)

One or more than one memory can be replaced by a memory emulator, in which we can introduce any kind 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 able 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 operations that allow the sensitization and observation of the fault itself. This analysis highly helps the student's knowledge of memory failing processes as well as his skill to generate appropriate March test algorithms to target specific pull of faults.

V. EXPERIMENTAL RESULTS AND CONCLUSION

The following chapter consists of all the software and hardware results observed in the project. The results include snapshots of each and every module individually with all the inputs, outputs and intermediate waveforms.

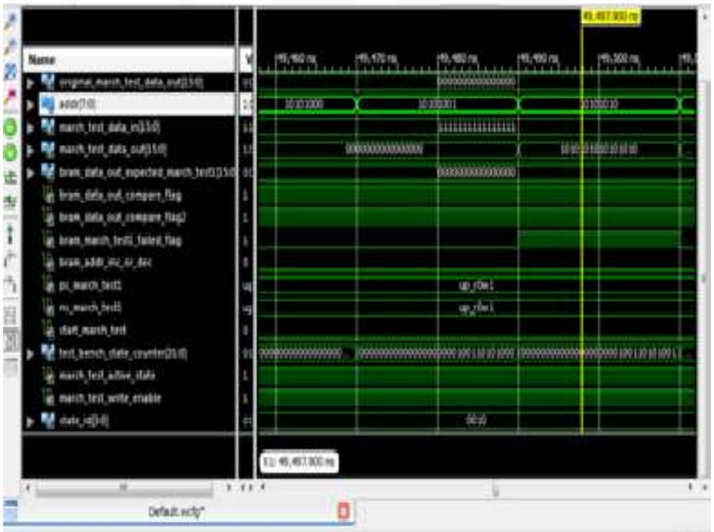

Name
original_march_test_data_out[15:0]
addr[7:0]
march_test_data_in[15:0]
march_test_data_out[15:0]
bram_data_out_expected_march_test1[15:0]
bram_data_out_compare_flag
bram_data_out_compare_flag2
bram_march_test1_failed_flag
bram_addr_inc_or_dec
ps_march_test1
ns_march_test1
start_march_test
test_bench_state_counter[31:0]
march_test_active_state
march_test_write_enable
state_id[3:0]
up_r0w1
up_r0w1
Default.wcfg*

Figure 4: Data output of BRAM using March C-

The written input values into BRAM are read with expected values. Here the contents of BRAM are read without any failure. Hence March test C- is Successful and BRAM is fault Free.

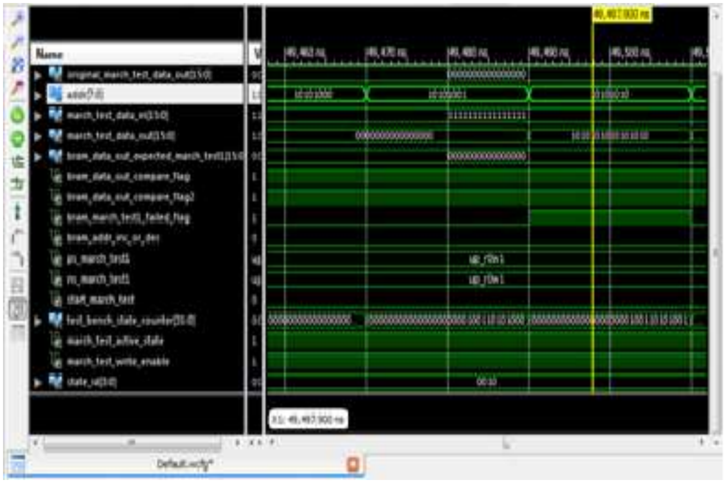
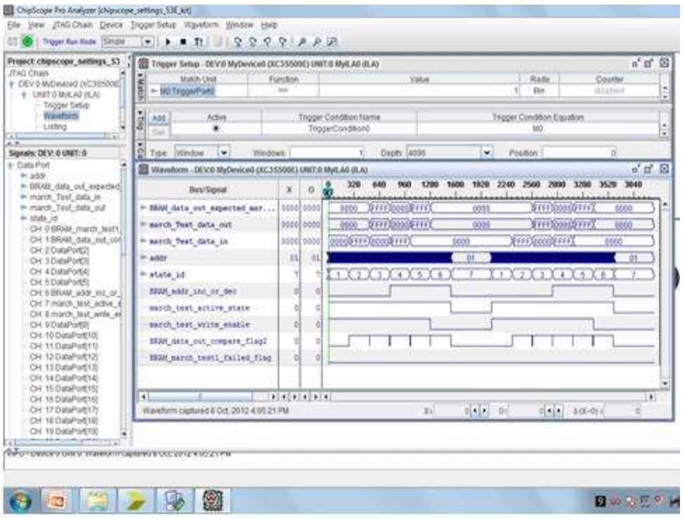

Name
original_march_test_data_out[15:0]
addr[7:0]
march_test_data_in[15:0]
march_test_data_out[15:0]
bram_data_out_expected_march_test1[15:0]
bram_data_out_compare_flag
bram_data_out_compare_flag2
bram_march_test1_failed_flag
bram_addr_inc_or_dec
ps_march_test1
ns_march_test1
start_march_test
test_bench_state_counter[31:0]
march_test_active_state
march_test_write_enable
state_id[3:0]
up_r0w1
up_r0w1
Default.wcfg*

Figure 5: Data output of BRAM using March C- with Fault insertion

The written input values into BRAM are read with expected values except at address “aa”. Here the contents of BRAM are read with fault. Hence March test C- is Successful and BRAM is faulty.

Chip scope results:



Waveform captured 8 Oct, 2012 4:06:21 PM

Figure 6: Data output of BRAM using March C-

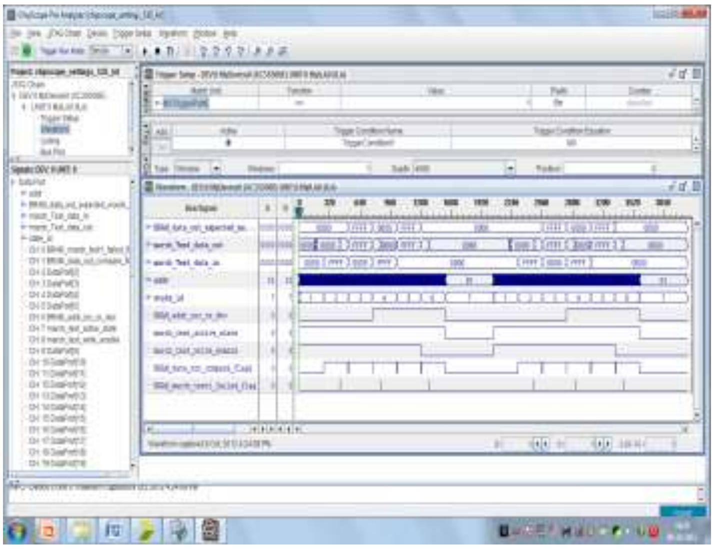
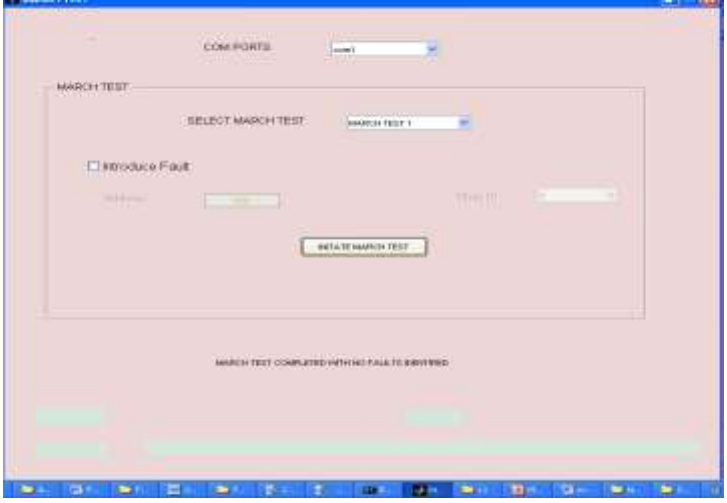


Figure 7: Data output of BRAM using March C- with Fault insertion

Results of Integrated top module with GUI: Fault diagnosis using March C- test:



COM PORTS com1

MARCH TEST

SELECT MARCH TEST MARCH TEST 1

☐ Introduce Fault

INITATE MARCH TEST

MARCH TEST COMPLETED WITH NO FAULTS IDENTIFIED

Figure 8: Without Fault insertion

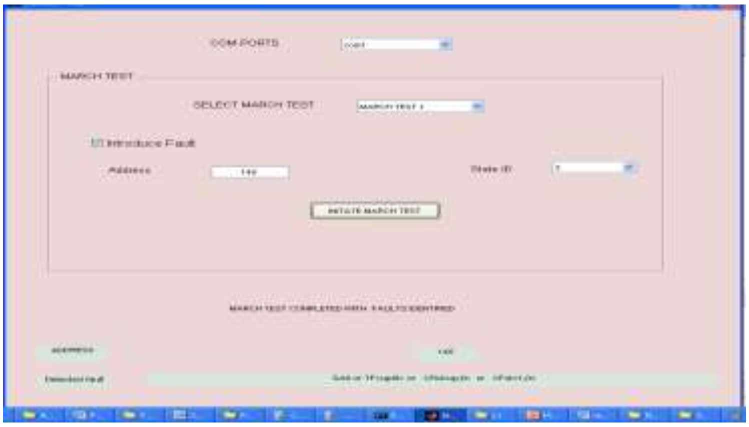


Figure 9: Without Fault insertion

Fault coverage and test lengths of March tests

Table 2: Fault coverage of March tests

Algorithm	SAF	TF	ADF	CFst
March C-	All	All	All	All
March SR	All	All	-	All
March B	All	All	All	-

Table 3: Test length of March tests

Algorithm	Complexity
March C-	10n
March SR	14n
March B	17n

VI. CONCLUSION

With technology scaling, process variations result in functional failures in memory systems. In this work, physical failure mechanisms in BRAM on FPGA boards are analyzed and classified into the established logic fault models. March test sequences are compared and optimized to target these emerging failure mechanisms. March C- test sequence, the memory test time is reduced.

Future scope

An open problem to be investigated in the future is diagnosis under a mixed-fault model, locating unlinked faults. Here the testing time depends upon the no. of memory locations (i.e; memory size) rather than the no. of faults. Hence we look forward for an alternative for this purpose. The complexity of the BRAM test configuration will increase in some worst case. All these problems need to be studied in the future research work in
this area.

REFERANCES

[1]. Versatile march test generator for hands-on memory testing laboratory, Galliere, J.; Dilillo, L.; Polytech. Montpellier, Univ. of Montpellier, Montpellier, France Microelectronic Systems Education (MSE), 2011 IEEE International.

[2]. Efficient Testing of SRAM With Optimized March Sequences and a Novel DFT Technique for Emerging Failures Due to Process Variations, Qikai Chen, Hamid Mahmoodi, Swarup Bhunia, and Kaushik Roy, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, VOL. 13, NO. 11, NOVEMBER 2005.

[3]. A.J. Van de Goor,"Testing Semiconductor Memories: theory and practice," John Wiley& sons, ISBN 0-471-92586-1,1991.

[4]. A.J. Van De Goor, 1993, "Using March Test to test SRAMs", IEEE Design of Computers, USA.