Global Journal of Computer Science and Technology: G Interdisciplinary
Volume 18 Issue 3 Version 1.0 Year 2018
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

FPGA Implementation of NPSF Testing using Block Code Technique

By K.L.V Ramana Kumari, M. Asha Rani, N. Balaji & SK. Salauddeen

Abstract- This paper presents a test structure for high speed memories. Built in self test (BIST) give the solution for testing memories and associate hardware for test pattern generation and application for a variety of test algorithms. Memory test algorithm for neighborhood pattern sensitive faults (NPSF) is developed by using block code technique to identify the base cell and deleted neighborhood cells. Test pattern generation can be done by using LFSR and Euler pattern generation. The testing process is verified using Xilinx ISE 14.2 and implemented on Nexys 4 DDR Artix 7 FPGA board.

Keywords: LFSR pattern generation, euler pattern generation, block code technique, Nexys 4 DDR Artix 7 FPGA.

GJCST-G Classification: D.2.5

FPGA Implementation of NPSF Testing using Block Code Technique

K.L.V Ramana Kumari [α], M. Asha Rani [σ], N. Balaji [ρ] & SK. Salauddeen [ω]

Abstract- This paper presents a test structure for high speed memories. Built in self test (BIST) give the solution for testing memories and associate hardware for test pattern generation and application for a variety of test algorithms. Memory test algorithm for neighborhood pattern sensitive faults (NPSF) is developed by using block code technique to identify the base cell and deleted neighborhood cells. Test pattern generation can be done by using LFSR and Euler pattern generation. The testing process is verified using Xilinx ISE 14.2 and implemented on Nexys 4 DDR Artix 7 FPGA board.

Keywords: LFSR pattern generation, euler pattern generation, block code technique, Nexys 4 DDR Artix 7 FPGA.

I. Introduction

Built in self test (BIST) design technique is a part of circuit which is used to test the circuit itself. Engineers design BISTs to satisfy demands such as lower repair cycle times or constraints such as limited technician accessibility and cost of testing during manufacture and high reliability. The main purpose of BIST is to reduce the test complexity, and thereby decrease the cost and reduce dependence upon external (pattern programmed) test equipment [1]. Built in self test techniques are very useful in testing of logic circuits, because they offer a cost effective way to test complex digital devices. First the BIST concept was implemented on combinational circuits, after that it found quick demands in the testing of regular structures like programmable logic arrays, read only memories, and random access memories.

Quick increment of multifaceted nature in the integrated circuits has prompt impact following on memory testing. On one view, the limit of random-access memories upgrades, in this manner expanding the test time and cost, on the other view, the complexity of memory circuits becomes more, and therefore more failure modes and faults should be considered in order to get a product with good quality [4]. Accordingly, different constraints need to take care while considering a test algorithm.

Minimizing the number of memory operations in order to allow large capacity memories to be tested in proper period of time and covering a larger collection of memory faults. In the last previous days of memory design test procedures were developed in an ad-hoc manner. The fault coverage of these ad-hoc test procedures was limited and often indeterminable [2]. This shortcoming, acknowledged by most researchers, motivated the introduction of such fault models as stuck-at faults, decoder faults, coupling faults. If density of memory circuits increases, it increases coupling effects then the pattern sensitive fault PSF is becoming very important fault model [3].

The rest of the paper describes in the following way. Part 2 illustrates neighborhood pattern sensitive fault model to be covered. Part 3 illustrates the design of testing the memory using block code technique. Using a flow chart the operation of Block code technique is explained in detail. Part 4 illustrates the results and FPGA implementation of memory testing on Nexys 4 DDR Artix 7 FPGA board. Synthesis report is generated using XILINX. Part 5 gives the conclusion and future scope of the NPSF testing.

II. Faults in Memory Design

a) Memory structure

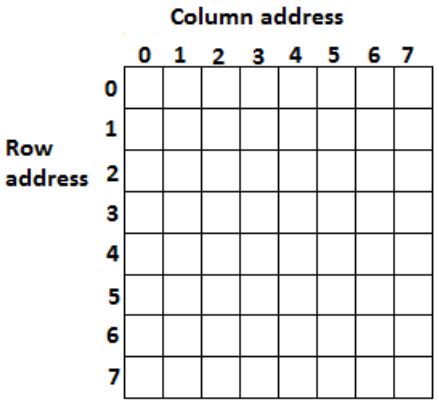

Fig. 1: 8*8 Memory structure

Fig 1 shows the memory structure of 8*8 RAM with 8 rows and 8 columns.

b) Neighborhoods Pattern Sensitive Faults

A Pattern Sensitive Fault is a restrictive coupling fault in which the content of a memory cell, or the capacity to change its content, is affected by a specific bit pattern in different cells in the memory. Here the

Author α: Asst. Professor, ECE, VNRVJIET.
e-mail: ramanakumari_jlv@vnrvjiet.in
Author σ: Professor, ECE, JNTUH. e-mail: ashajntu1@yahoo.com
Author ρ: Professor, ECE, JNTUK. e-mail: narayanamb@rediffmail.com
Author ω: M. Tech (VLSI System Design).
e-mail: shaiksalauddeen@gmail.com

information maintenance and change of the victim cell are influenced by an arrangement of aggressor cells. A neighborhood pattern sensitive Fault (NPSF) is a special instance of pattern sensitive faults [8], wherein the affecting (coupling) cells are in the neighborhood of the impacted (coupled) cell. The coupled cell is known as the base (or victim) cell and the coupling cells are known as the deleted neighborhood cells. The neighborhood incorporates 4 deleted neighborhoods and one base cell. It forms the coupling of these 4 cells with the base cell considering effect as the NPSF fault modeling.

c) Active NPSF

The base cell changes its contents because of changes in the deleted neighborhood pattern. To identify these affect, every cell must be read in state 0 and in state 1 for every single conceivable change in the deleted neighborhood pattern. There are two distinctive conceivable states for the base cell (0 and 1), N-1 methods for picking the deleted neighborhood cell which must experience one of two conceivable advances (↑ or ↓), and N-2 potential outcomes for the rest of the neighborhood cell contents.

2* (N-1)*2*2N-2 = (N-1)* 2N is the aggregate number [8] of active neighborhood patterns (ANPs). It require 128 patterns for identifying active NPSF faults. For type-1 NPSF, 2 is base cell and 0, 1, 3 and 4 are deleted neighborhood cells.

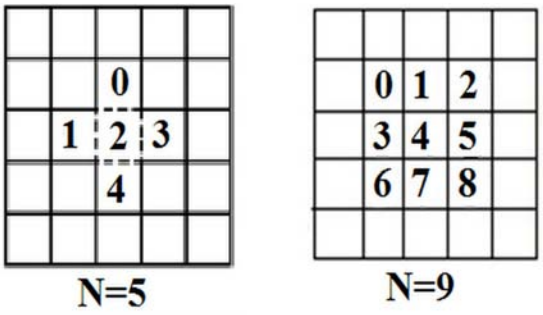

Fig. 2: Type-1 NPSF and TYPE-2 NPSF

Type-2 NPSF is used when diagonal couplings are significant. For type-2 NPSF, 4 is base cell and 0, 1, 2, 3, 5, 6, 7 and 8 are deleted neighborhood cells.

d) Passive NPSF

The content of the base cell can't be changed because of a specific neighborhood pattern. Every cell must be written and read in state 0 and in state 1 for all changes of the deleted neighborhood pattern. For each of the 2N-1 deleted neighborhood patterns, the two conceivable advances ↑ and ↓ must be confirmed. Subsequently, the aggregate number of PNPSFs is 2*2N- 1=2N. Here N is number of cells. It require 32 test patterns for recognizing active NPSF faults.

e) Static NPSF

The content of a base cell is compelled to a specific state because of a specific neighborhood pattern. To distinguish these faults, apply the 2N combinations of 0s and 1s to the N-cell neighborhood, and check by perusing every cell that each pattern can be stored. It varies from ANPSF that it is not necessary to have a change to transition to sensitize an SNPSF. We required 32 test patterns for recognizing static NPSF faults.

III. Memory Testing

a) Block diagram of testing process

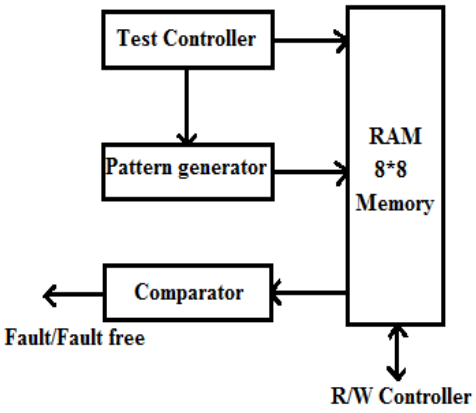

Fig. 3: Block diagram of memory testing

Fig 3 shows the testing process [12]. The pattern generator block generates the patterns for memory testing. These patterns are given as inputs to the deleted neighborhood cells of the selected base cell which are selected by the test controller. Then using the comparator we can find out the base cell content is changed or not. Deleted neighborhood cell and base cells are selected by using test controller, writing and reading of base cell can be done by using R/W controller.

b) Flow chart

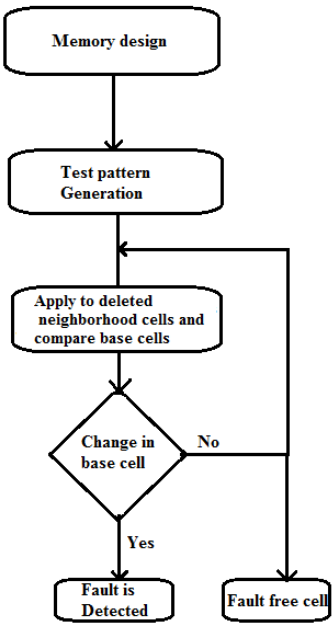

Fig. 4: Flow chart of Memory testing

The flow chart shown in Fig 4 represents the operation of memory testing. 8*8 memory design and testing can be done using Verilog HDL [7]. Generate the test patterns by using LFSR and Euler pattern techniques. Choose base cells according to the test cases and apply the test pattern to deleted neighborhood cells. Compare the content of base cells to test the fault.

Fig 5 shows that the memory is divided into nine different blocks. Nine test cases are used for testing the memory since maximum size of the block is 9. For test case 1 each block of first cell is selected as base cell then the base cells are selected as 00, 03, 06, 30, 33, 36, 60, 63, 66 and the pattern are applied to the deleted neighborhood cells of each base cell. For test case 2 each block of second cell is selected as base cell then the base cells are selected 01, 04, 07, 31, 34, 37, 61, 64, and 67. Likewise for test case 9 the base cells are selected as 22, 25, 52 and 55.

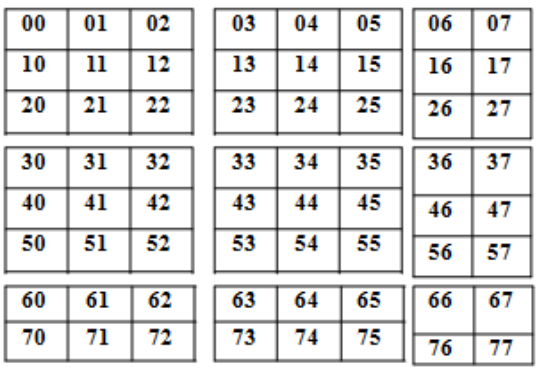

Fig. 5: 8*8 memory divides into blocks

If base cell positions are not available in blocks then the base cells are chosen based on their availability [10]. Table 1 shows the base cells according to their respective test case.

c) Memory testing by using different test cases

Table 1: Base cell positions for test cases

S.NO.	TEST NUMBER	BASE CELL POSITIONS
1	TEST1	00,03,06,30,33,36,60,63,66
2	TEST2	01,04,07,31,34,37,61,64,67
3	TEST3	02,05,32,35,62,65
4	TEST4	10,13,16,40,43,46,70,73,76
5	TEST5	11,14,17,41,44,47,71,74,77
6	TEST6	12,15,42,45,72,75
7	TEST7	20,23,26,50,53,56
8	TEST8	21,24,27,51,54,57
9	TEST9	22,25,52,55

Table1 shows positions of base cells for detection of NPSF faults in memory. This table describes how base cells are selected for respective test cases.

d) Algorithm

Step 1: Write all base cells to 1.

Step 2: Applying pattern to neighborhood cells

Step 3: Read base cells

Step 4: Write all base cells to 0

Step 5: Read base cells

Step 6: Compare base cells if any change in base cell fault is detected.

e) Operation

During the 1st operation, cell 1 of each block will be selected as a base cell and the corresponding deleted neighborhood cells will be chosen accordingly shown in Fig 6. For test case 1 the base cell in 4th block is 30, the deleted neighborhood cell positions are 20, 31 and 40. During the 2nd test all the cells are in position 2 will be selected. For test case 2 the base cell in 5th block is 34, the deleted neighborhood cell positions are 24, 33, 44 and 35. Similarly for test case 9 the base cell in 5th block is 55, the deleted neighborhood cells are 45, 54, 65 and 56. Thus the entire memory can be tested after 9 independent test operations. The test patterns are applied to the deleted neighborhood cell and read the contents of base cell. If the content of base cell is changed then the NPSF fault can be identified for the particular pattern.

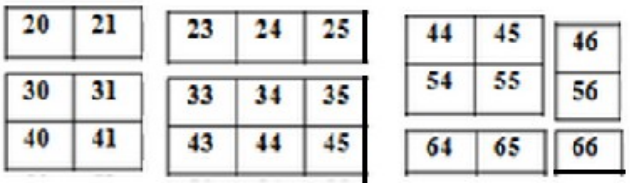

Fig. 6: Formation of deleted neighborhood cell

IV. Results

a) Modelsim Simulation

The simulation results are shown for different test cases and different modes of operation.

i. *Memory testing for test case 2 write mode*

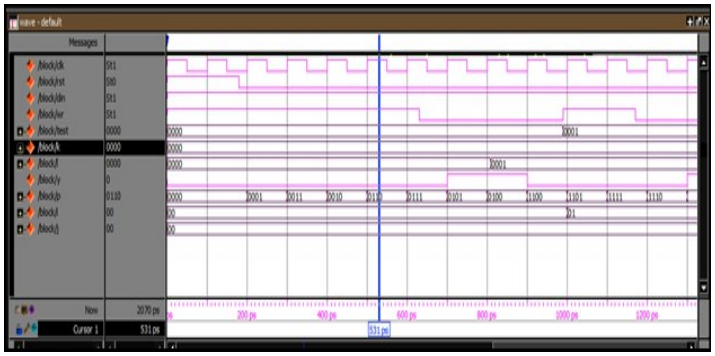

Fig. 7: Simulation output of test case 2 write mode

Fig 7 shows the simulation resuls of memory in test case 2. The wr signal controles the read, write operations. The binary representation of test case 2 for memory testing is 0010. Whenever the "wr" signal is high all base cells are written in state 1 and pattern is applied to deleted neighborhood cells.

ii. *Memory testing for test case 2 read mode*

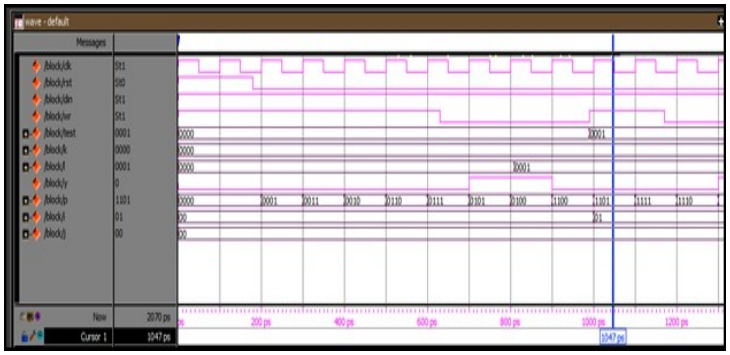

Fig. 8: Simulation output of test case 2 read mode

The fig 8 shows output result of read mode in the memory. Whenever "wr" signal is low it performs the read operation, all the base cells are read from the memory.

iii. *Fault detection in memory*

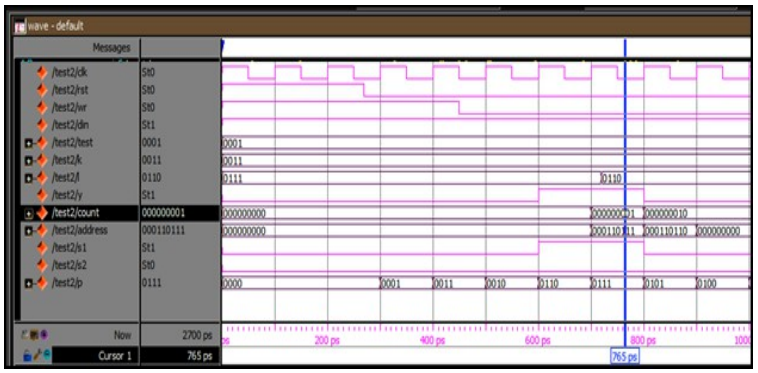

Fig. 9: Simulation output of fault detection in memory

Fig 9 shows the simulation results. Compare the contents of base cell with fault free memory in test case 0001. If base cell changes its contents then the error signal goes high, identified it as a fault and the address of the fault location also stored. Wherever the base cell changes the fault is detected and it shows the exact location of faulty base cell. All the NPSF faults are detected by using this technique and we applied all test cases for detecting NPSF faults in memory.

b) Xilinx Reports

Xilinx ISE is a software tool is used for synthesis and analysis of HDL designs [11]. Allows the developer to synthesize the design, gives timing analysis, RTL diagrams, and configure the target device with a programming kit.

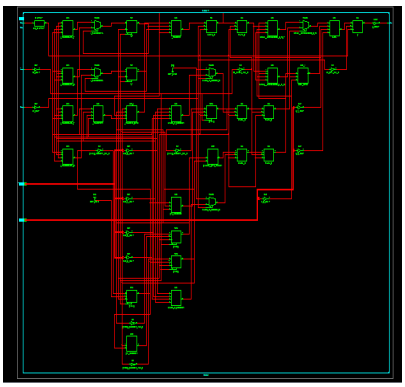

Fig. 10: Technological view

The RTL Schematic has been generated using XILINX 14.2 version. Fig 10 shows the RTL Schematic and technological view generated by using Xilinx. The Nexys4 DDR FPGA is optimized for high performance logic and offers 9,312 slices, each containing four input LUTs and eight flip-flops. Hence, the device facilitates the generated hardware to successfully fit into the available microcells.

c) FPGA Implementation

FPGA configuration files are transferred via JTAG port in bit file format. Xilinx's ISE Web Pack and EDK software can create bit file from Verilog. Nexys 4 DDR Artix 7 FPGA internal clock frequency is 100MHz; it has 16 LED outputs and 16 switches. The switches are used for giving the inputs and applying test cases. The output LEDs are used for showing the address of fault location in memory. The last LED is indicating detection of fault. If fault is detected then LED will ON otherwise it will OFF. The next 6 LEDs are showing the location of fault address.

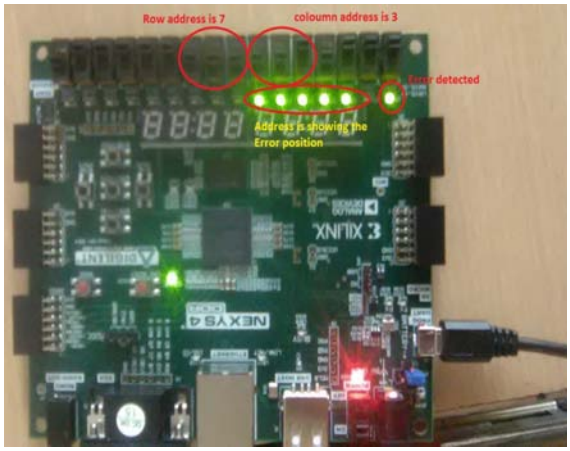

Fig. 11: Detecting fault in memory for test case 0001 with Nexysv4 DDR Artix 7 FPGA

The fig 11 shows the output of memory in reading base cells in test case 0001. The fault is detected by showing the LED glown and the faulty base cell coloumn address is 3and row address is 7. The six LEDs are indicating the location of faulty address.

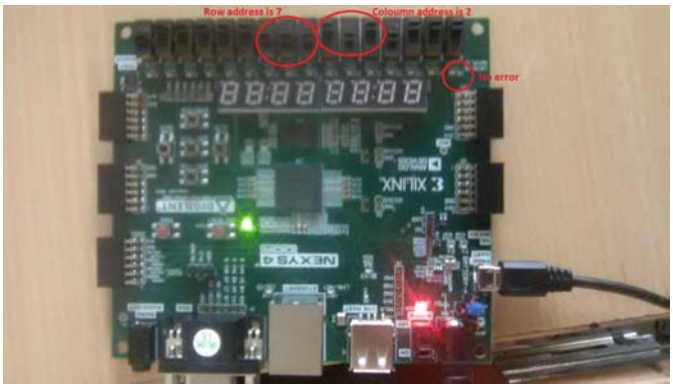

Fig. 12: Fault free condition of base cell in test case 0000

The fig 12 shows the output of fault detection in test case 0000. All base cells are compared and no faults are detected in test case 0000. So the fault signal remains low and not showing the address of the fault.

Table 2: Device utilization summary

SLICE LOGIC UTILIZATION	LFSR	Euler pattern
number of slices flip flops	44	44
number of 4-bit LUTs	32	31
Number used as logic	42	36
Number of occupied slices	42	36
Number of bonded IOBs	34	27
Number of slices contains only related logic	18	19
Number of full used LUT ff	50	42

The table 2 shows the summary of device utilization for Nexys 4 DDR Artx 7 FPGA. The utilization of device gives the information of number of LUTs, logic blocks and number of bonded IOBS used in FPGA, it can be obtained from synthesis report generated by Xlinx ISE.

V. Conclusion and Future Scope

For memory arrays, the excessive test algorithmic time associated with NPSF fault model. This paper presents a BIST implementation using block code technique to select base cells and deleted neighborhood cells of neighborhood pattern sensitive faults (NPSFs) in random access memories (RAMs). Testing process is synthesized and implemented on Nexys 4 DDR Artix 7 FPGA board. In order to improve the effectiveness of coupling faults, type-2 NPSFs can be modelled and tested.

References Références Referencias

1. D. A. Michael Lee Bushnell, Vishwani D. Agrawal, “Essentials of Electronic Testing for Digital, Memory, and Mixed-Signal VLSI Circuits”. New York: Kluwer Academic Publishers, 2nd ed., 2002
2. M. A. Miron Abramovici, Melvin A. Breuer, Arthur D. Friedmaan, “Digital Systems Testing and Testable Design ”Jaico Publishing House, 2002.
3. K.-L. Cheng, M.-F. Tsai, and C.-W. Wu, “Neighborhood pattern-sensitive fault testing and diagnostics for random access memories,” IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems, vol. 21, no. 11, pp. 1328– 1336, Nov. 2002.
4. A. J. vandeGoor, Testing Semiconductor Memories: Theory and Practice. Chichester, England: John Wiley & Sons, 1991.
5. Testing Active Neighborhood Pattern-Sensitive Faults of Ternary Content Addressable Memories Yu-Jen Huang and Jin-Fu Li Advanced Reliable Systems (ARES) Laboratory, Proceedings of the Eleventh IEEE European Test Symposium.
6. Implementation of Random Number Generator Using LFSR for High Secured Multi Purpose Applications. International Journal of Computer Science and Information Technologies, Vol. 3 (1), 2012, 3287-3290
7. AJ. van de Goor, Testing Semiconductor Memories: Theory and Practice, John Wiley & Son, Chichester, England, 1991.
8. K. L. Cheng, M. Fo Tsai and C. W. Wu, "Neighborhood Pattern Sensitive Fault Testing and Diagnostics for Random-Access Memories", IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems, Vol. no. Nov. 2002.
9. K. Cheng and C.-W. Wu, " Efficient neighborhood pattern sensitive fault test algorithms for semiconductor memories", in Proc. IEEE VLSI Test Symp(VTS), Marina Del Rey, California, Apr 2001, pp. 225-237.
10. S.V. Yarmolik and Mrozek, "Multi Background Memory Testing", In Proc. Th MIXDES 14 International Conference, Ciechocinek, Poland, June 2000.
11. Nexys4 DDR FPGA xilix user guide, in xilinx provided web page.
12. K.L.V. Ramana kumari,Asha rani and Balaji, ”FPGA implementation of memory design and testing”, 2017 IEEE 7th International Advance Computing Conference.