International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2017 IJSRCSEIT | Volume 2 | Issue 4 | ISSN : 2456-3307

Design of Alu-Based Symmetric Transparent Online Bist for Ram

V. Rajendra Chary
Department of Electronics and Communication Engineering, Sreenidhi Institute of Science and Technology,
Affiliated to Jawaharlal Nehru Technological University, Hyderabad, Telangana, India

ABSTRACT

A 3-bit RAM is tested using 5-bit ALU with the help of symmetric transparent online BIST scheme. This architecture skips signature prediction. It also reduces test time. If the RAM output consists if all 1's, then it indicates that there is no error in the RAM module, otherwise it indicates that there is an error. The implemented scheme utilizes an ALU in order to generate the test pattern and compress the responses of the memory module, the word width of memory can be smaller than the number of stages of ALU. Hence multiple non-identical memories can also be tested in a pipeline way. All synthesis and simulation results are performed using xilinx14.2 ISE

Keywords: RAM, Signature Prediction, Test Pattern

I. INTRODUCTION

Testing RAM is a DFT method. DFT is the acronym for Design for Testability. DFT is of 2 types viz.... ATE and BIST. ATE is the older method of DFT technique. ATE is the acronym for Automated test equipment. In this method we will compare expected result and obtained result. But there are some disadvantages with this technique. Some of them are as follows:

- Occupies large space.
- Costly.
- Atspeed.
- Can't detect the location of error.
- Any time testing is not possible.

To overcome these disadvantages, another technique called BIST was introduced. BIST is of 2 types, they are online BIST and offline BIST. Another classification of BIST consists of memory BIST and Logic BIST. Memory BIST is used when storage of result is required.

Any type of BIST consists of IC logic and test logic. IC logic includes DUT/MUT. Test logic includes TPG,ORA and BIST controller.

BIST has become a standard industrial practice[1]-[3].They are fabricated with minimal design rule tolerance which makes it more susceptible to defects. Testing of RAM modules can be performed after or before manufacturing. But testing before manufacturing is a preferred option of testing. While testing, some algorithms has to be used. In testing RAM module in this architecture, March-c algorithm is used. Its operation is listed in table1 for both symmetric and symmetric transparent version [4]-[5].A March test comprises a series of march elements that perform a predetermined of operations of read/write in every word[6].

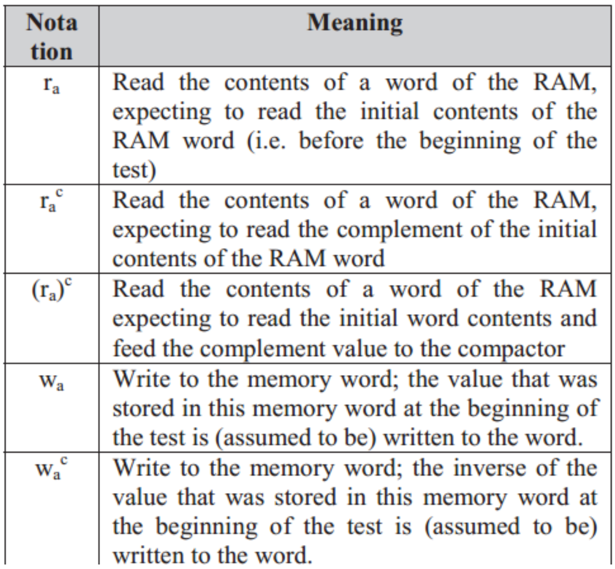
	TRANSPARENT VERSION	SYMMETRIC TRANSPARENT VERSION
M0	(ra),((ra)^c),(ra), ((ra)^c),(ra)	((ra)^c)
MI	(ra,wa^c)	(ra,wa^c)
M2	(ra^c,wa)	(ra^c,wa)
M3	(ra,wa^c)	(ra,wa^c)
M4	(ra^c,wa)	(ra^c,wa)
M5	(ra)	(ra)

Table 1.March-c algorithm

The advantages of BIST over ATE are as follows:

- Lower cost of test.
- Shorter test times.
- Easier customer support.
- Capability to perform tests outside the production electrical testing environment.

The tabular form 2 illustrates the notations used in march-c algorithm for symmetric transparent BIST.



Notation	Meaning
r_a	Read the contents of a word of the RAM, expecting to read the initial contents of the RAM word (i.e. before the beginning of the test)
r_a^c	Read the contents of a word of the RAM, expecting to read the complement of the initial contents of the RAM word
$(r_a)^c$	Read the contents of a word of the RAM expecting to read the initial word contents and feed the complement value to the compactor
w_a	Write to the memory word; the value that was stored in this memory word at the beginning of the test is (assumed to be) written to the word.
w_a^c	Write to the memory word; the inverse of the value that was stored in this memory word at the beginning of the test is (assumed to be) written to the word.

Table 2. Notations for symmetric transparent BIST

Transparent BIST was proposed by Nicolaidis[7]-[8]. In transparent BIST, the initial write-all zero phase is skipped and a signature prediction phase is issued during which a signature is captured and stored. One of the issues arising when transparent BIST is employed is that of the test data generator and response compactor. Further, the concept of transparent BIST was developed by Yarmolik et al[9]-[10]. He proposed symmetric transparent version of BIST. In this method, the signature prediction phase is skipped and March series is modified in such a way that the final signature is equal to all-zero state irrespective of RAM initial contents[11]-[15].

The rest of the paper is organized as follows: implemented architecture is discussed in section II. Simulation results are illustrated in section III. Conclusion is discussed in section IV.

II. IMPLEMENTED ARCHITECTURE

The implemented architecture is shown in figure 1. The architecture consists of different modules such as:

1. ALU
2. Register
3. OR gates
4. RAM module.

ALU Module : The ALU module in the architecture stands for Arithmetic and Logic Unit which performs arithmetic and logical operations on the data. The ALU implemented in this architecture is a 1's complement ALU.

In ALU addition or subtraction will be performed in this architecture based on these equations:

Addition:1 +[(A-1+B)%(2^n-1)]
Subtraction: if A > B then not(A-B)
Else B-A

If A is register content, B is RAM content.

RAM module : The RAM module implemented consists of two modes of operations such as Read & Write. In the RAM module, during Write operation address is given and the data that has to be stored is also given. The data will be stored in the specified address. During Read operation the address is specified. The data that is present in the specified address is given on the output data signal.

In the architecture three RAMs are implemented with different word widths. The RAMs can store the 3bits, 4bits and 5bits respectively.

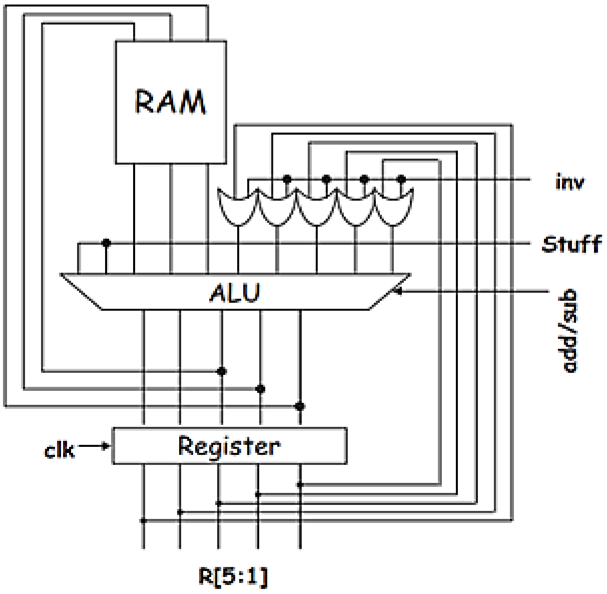


Figure 1. Block diagram of 3-bit RAM

Stuff: Stuff signals are connected to the output signals of the RAMs that are passing through the ALU. If RAM of 3bits is being accessed then the Stuff values must be given in order to make it 5 bits. If RAM of 5 bits is being accessed then the Stuff values must not be given because it is already 5bits.

The testing of RAM can be extended to muti-RAM with different widths also. The architecture of three RAM modules with different width is shown in figure2.

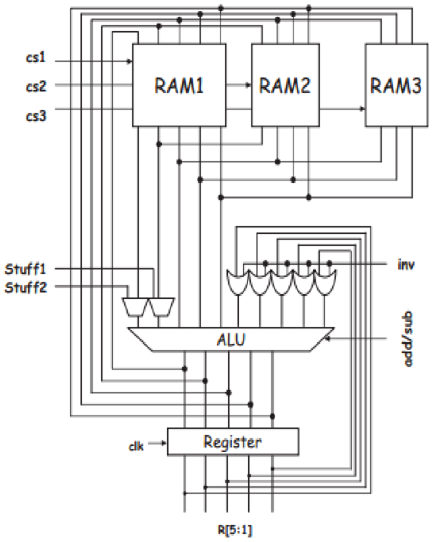
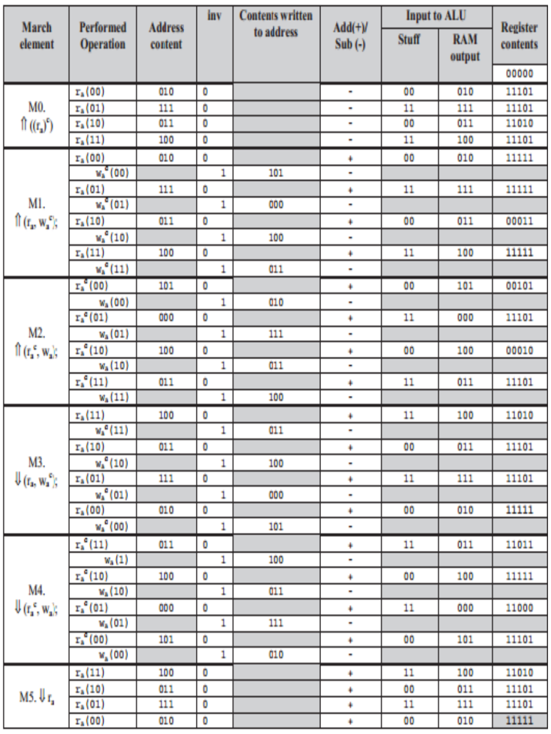


Figure 2. transparent testing of 3 RAM modules of different width

The operation of 3-bit RAM using 5-bit ALU is illustrated in table3. Initially, register contents are all-zeros. Using March-c algorithm, read and write operations will be performed. At the end of algorithm, the register contents are all-1's,it indicates that there is no error in RAM. If the register contents are not all-1's it indicated an error in RAM functionality.



March element	Performed Operation	Address content	inv	Contents written to address	Add(+)/ Sub (-)	Input to ALU		Register contents
						Stuff	RAM output	00000
M0. $\Uparrow((r_a)^c)$	$r_a(00)$	010	0		-	00	010	11101
	$r_a(01)$	111	0		-	11	111	11101
	$r_a(10)$	011	0		-	00	011	11010
	$r_a(11)$	100	0		-	11	100	11101
M1. $\Uparrow(r_a, w_a^c);$	$r_a(00)$	010	0		+	00	010	11111
	$w_a^c(00)$		1	101	-			
	$r_a(01)$	111	0		+	11	111	11111
	$w_a^c(01)$		1	000	-			
	$r_a(10)$	011	0		+	00	011	00011
	$w_a^c(10)$		1	100	-			
	$r_a(11)$	100	0		+	11	100	11111
	$w_a^c(11)$		1	011	-			
M2. $\Uparrow(r_a^c, w_a);$	$r_a^c(00)$	101	0		+	00	101	00101
	$w_a(00)$		1	010	-			
	$r_a^c(01)$	000	0		+	11	000	11101
	$w_a(01)$		1	111	-			
	$r_a^c(10)$	100	0		+	00	100	00010
	$w_a(10)$		1	011	-			
	$r_a^c(11)$	011	0		+	11	011	11101
	$w_a(11)$		1	100	-			
M3. $\Downarrow(r_a, w_a^c);$	$r_a(11)$	100	0		+	11	100	11010
	$w_a^c(11)$		1	011	-			
	$r_a(10)$	011	0		+	00	011	11101
	$w_a^c(10)$		1	100	-			
	$r_a(01)$	111	0		+	11	111	11101
	$w_a^c(01)$		1	000	-			
	$r_a(00)$	010	0		+	00	010	11111
	$w_a^c(00)$		1	101	-			
M4. $\Downarrow(r_a^c, w_a);$	$r_a^c(11)$	011	0		+	11	011	11011
	$w_a(1)$		1	100	-			
	$r_a^c(10)$	100	0		+	00	100	11111
	$w_a(10)$		1	011	-			
	$r_a^c(01)$	000	0		+	11	000	11000
	$w_a(01)$		1	111	-			
	$r_a^c(00)$	101	0		+	00	101	11101
	$w_a(00)$		1	010	-			
M5. $\Downarrow r_a$	$r_a(11)$	100	0		+	11	100	11010
	$r_a(10)$	011	0		+	00	011	11101
	$r_a(01)$	111	0		+	11	111	11101
	$r_a(00)$	010	0		+	00	010	11111

Table 3 Operation of 3-bit RAM with 5-bit ALU

III. RESULTS

The figure 3 illustrates the RTL schematic of the implemented architecture.

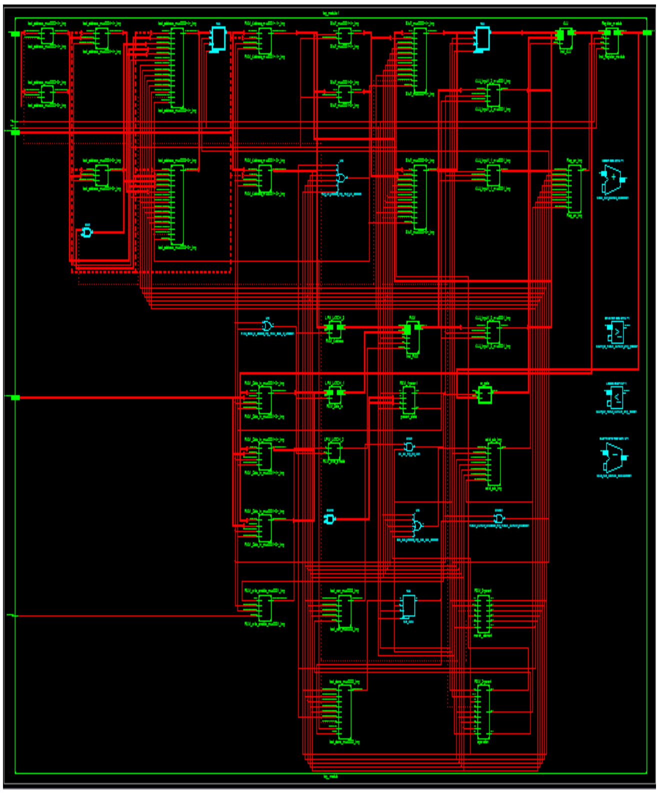


Figure 3. RTL schematic of 3-bit RAM

The figure 4 illustrates the final output of architecture in which the register contents will be all-1's which is an indication that there is no error in RAM functionality.

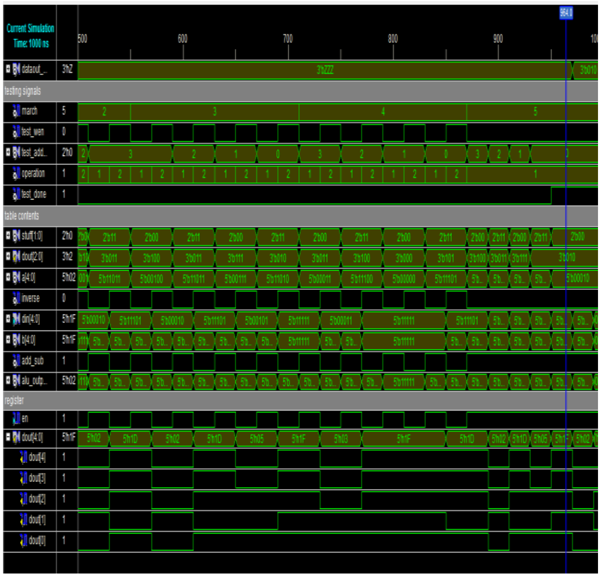


Figure 4. Final output of 3-bit RAM

IV.CONCLUSION

In this work we implemented a BIST scheme that can be utilized to test 3-bit RAM. The BIST scheme used is symmetric transparent version. With this scheme, we avoided the use of signaturepredictor with which architecture complexity reduced. This 3-bit RAM can be extended to higher order RAMs as well even with different widths.

V. REFERENCES

[1]. Voyiatzis, C. Efstathiou, C. "Symmetric Transparent Online BIST forArraysof Word-OrganizedRAMs", IEEE 2013

[2]. R. Aikten,et.al, "A Modular Wrapper enabling high speed BIST and repair for small wide memories"proc.of Int. Test conference,pp.997-1005,2004

[3]. X.Du,N.Mukherjee,W.T. Cheng and S.M.Reddy, " Full speed field programmable memory BIST architecture". proc.of Int. Test conference,pp.1173-1182,2005

[4]. A.J.Vande Goor, "Using March Tests to test SRAMs",IEEE design and test of computers,1993

[5]. K.Thaller and A.Steininger, " A transparent online memory test for simultaneous detection of functional faults and soft erros in memories," IEEE trans.pp 413-422,2003

[6]. Jin-Fu Li, "Transparent test methodologies for RAM with/without ECC" CAD of Integrated Circuits and Systems,IEEE,pp 1888-1893,vol 26,issue 10,2007

[7]. M.Nicolaidis, "Theory of transparent BIST for RAMs."IEEE trans on computers,vol.45,no-10,pp1141-1156,1996

[8]. M.Nicolaidis, "An efficient BIST for functional test of embedded RAMs"15th symposium on fault tolerant computing,1985.

[9]. V. N. Yarmolik, S. Hellebrand. H.-J. Wunderlich, "Symmetric Transparent BIST for RAMs",1999.

[10]. V.N. Yarmolik., I.V. Bykov, S. Hellebrand, 11.-i. Wunderlich, Transparent Word-Oriented Memory BIST based on Symmetric March Algorithms", in European Dependable Computing Conference, 1999.

[11]. I.Voyiatzis, "Accumulator based compression on Symmetric transparent RAM BIST" IEEE International conference on design and technology of Integrated Systems in Nanoscale technology,2006

[12]. M.Psarakis, D.Gizopoulos,M.Sonza Reorda, "Microprocessors software based self testing"IEEE design and test of computers magazine,vol 27,no.3,pp 4-19,2010

[13]. Ioannis Voyiatzis, "An Accumulator- Based Compaction Scheme for online BIST of RAMs". IEEE xplore 2009

[14]. Paul McEvoy and Ronan Farrel, "Built-in test engine for memory test", IEEE IC Test Workshop, Sep 2004.

[15]. Arthur D. Friedman, Melvin A. Breuer and Miron Abramovci, "Digital Systems Testing And Testable Design", A Jaico Book, 2006.

[16]. X. Du, N. Mukherjee, W.T Cheng and S.M Reddy, "Full- speed field-programmable memory BIST architecture", Proc. of mt. Test Conference, pp. 1173-1182, 2005

[17]. X. Du, N. Mukheijee, W-T Cheng, S. M. Reddy, "A Field-Programmable Memory BIST Architecture Supporting Algorithms and Multiple Nested Loops", Proc. of the Asian Test Symposium, paper 45.3, 2006.