International Journal of Emerging Engineering Research and Technology
Volume 2, Issue 8, November 2014, PP 76-83
ISSN 2349-4395 (Print) & ISSN 2349-4409 (Online)

FPGA Implementation of ALU Based Address Generation for Memory

Mr. Nookala SaiRam[1], Mrs. Athira G. Krishna[2]

[1]ECE Department, CVR College of Engineering, Hyderabad, India (Student)
[2] ECE Department, CVR College of Engineering, Hyderabad, India (Assistant Professor)

Abstract: *One of the key components of memory BIST is the Address Generator (AG). In order to detect speed-related faults, the address generator has to generate different address sequences to allow for appropriate address transitions. Its complexity is a major design issue since it requires large area and limits the BIST speed. For generating the address for Radom Accesses Memory (RAM) it requires extra hardware. If an Arithmetic and Logic Unit (ALU) uses for generating the address for RAM it reduces the hardware overhead.*

ALU based address generation are used in many applications like Smartcard Chips, Digital Signal Processors and Microcontrollers etc. This technique is designed by using Verilog HDL. Simulation is done to verify the functionality and synthesis is done to get the Netlist. The code is simulated and synthesized using Xilinx ISE 13.2 and the designed is implemented in SPARTAN 3E FPGA board.

Keywords: *Memory BIST, Address Generation, ALU-based implementation.*

1. INTRODUCTION

Built-In Self-Test (BIST) has become a standard industrial practice for testing memories since memory cores constitute a major part of the die area. One of the key components of memory BIST is the address generator (AG). In order to detect speed-related faults, the address generator has to generate different address sequences to allow for appropriate address transitions. Its complexity is a major design issue, since it requires large area and limits the BIST speed. For memory testing, the address generator has to generate different Counting Methods (CMs) since each counting method has its own detection. The main purpose of the address generator is to generate the test patterns. Such address generators are Linear Feedback Shift Registers (LFSRs), Pipeline LFSR, Binary Up down Counter and Gray Code up down counter.

For memory testing, the address generator has to generate different Counting Methods (CMs) since each counting method has its own detection capability. Hence, the detection of different memory fault classes requires different address generators [3, 6, 8 and 9]. Table 1 highlights two of the most well-established counting methods by giving an example for a memory with N=4 words. The Linear (LI) counting method specifies the address sequence: 0, 1, 2, 3... N-1 when going Up' '; and N-1... 3, 2, 1, 0 when going Down' '. The Linear counting method is used for detecting single-cell and coupling faults. The Address Complement (AC) counting method specifies an address sequence: 0000, **1111**, 0001, **1110**, 0010, **1101**, etc. [10]. Each bold address is the one's complement of the previous address, as shown in column denoted 'AC' in Table 1. The AC counting method stresses the address decoders, because all address bits switch upon address transitions. This causes lots of noise, a large power surge, and maximal delay and is used for detecting speed-related faults.

Table1. *Linear and Address Complement counting methods*

Step	LI	AC
0	00000000	00000000
1	00000001	**11111111**
2	00000010	00000001
3	00000011	**11111110**
4	00000100	00000010
5	00000101	**11111101**
6	00000110	00000011
7	00000111	**11111100**
8	00001000	00000100
9	00001001	**11111011**
10	00001010	00000101
11	00001011	**11111010**
12	00001100	00000110
13	00001101	**11111001**
14	00001110	00000111
15	00001111	**11111000**

In this work we contribute to the field of memory BIST by proposing a novel address generator based on an ALU. ALU modules commonly exist in current circuits; moreover, the outputs of ALU modules drive the address input of RAM modules, as shown in the elementary schematic diagram of the MIPS architecture of Figure1. Therefore, the utilization of the existing ALUs as address generators for the testing of RAMs may drive down the hardware overhead.

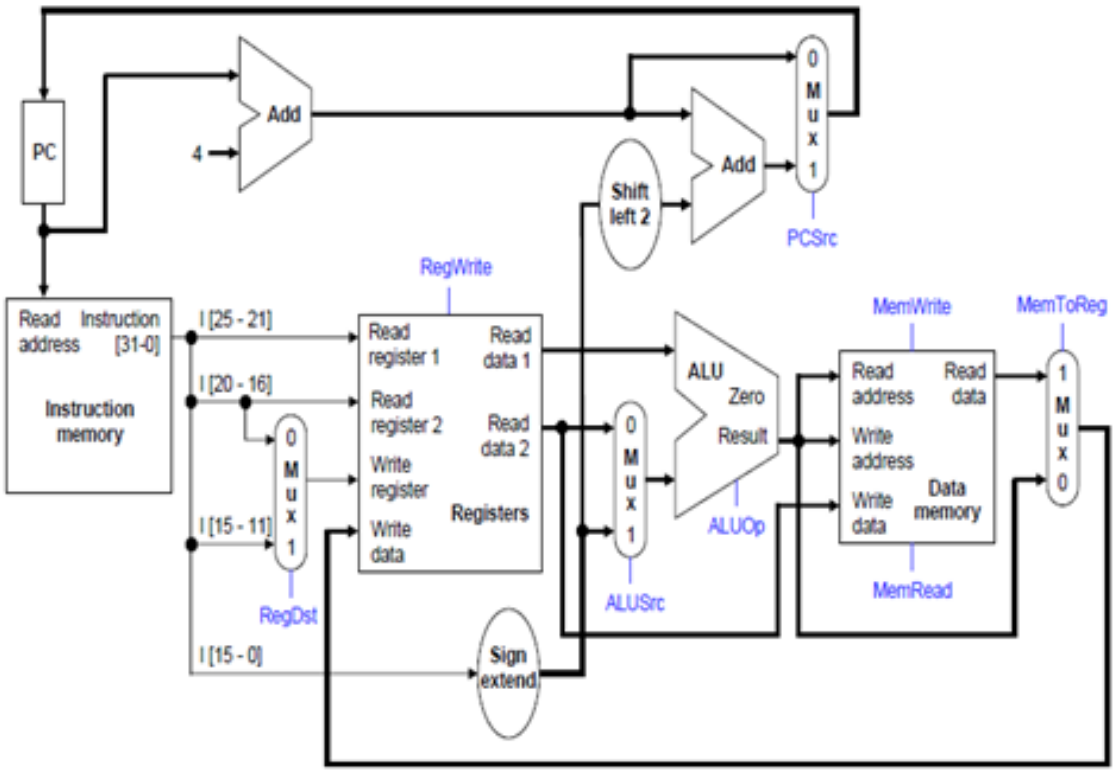
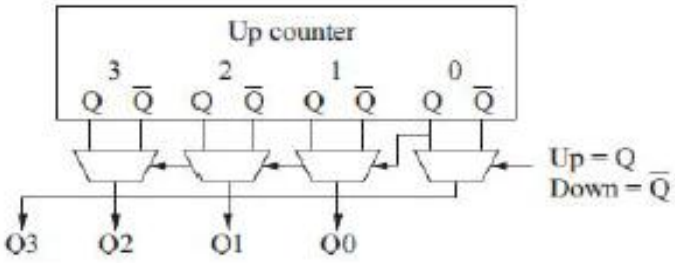


Figure1. *Basic diagram of the MIPS architecture*

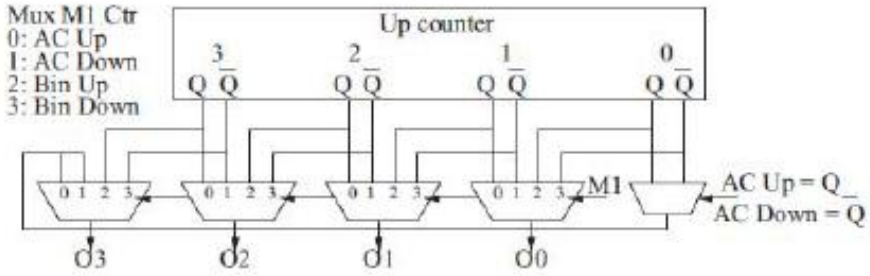
2. PREVIOUS WORK

Address generator implementations that can combine linear and address complement counting methods have been proposed in [4]. In the implementation proposed in [4] was integrated into a more generic scheme to generate more counting methods. In Figure 2(a) we present the address

complement address generator implementation using a counter proposed in [4]. The 'U/D' signal controls the most-significant address bit O3, which is the least-significant counter bit '0', because O3 of address complement changes with each clock period. The output of bit0 controls the muxes of all bits x, with x>0. The address generator module shown in Figure 1(a), generates the sequence shown in Table 1 (third column) or its inverse, depending on the value of the signal Up/Down.



(a)



(b)

Figure2. *4-bit address generator (a) Address Complement and (b) combined Address Complement and Linear address generator*

In order to generalize the generator in order to also generate the linear counting method, the authors of [4] also proposed the scheme shown in Figure 2(b) which uses the signal 'M1' to control the multiplexers. The scheme presented in Figure 2(b), apart from the n-stage counter, requires n 4-to-1 multiplexers and one 2-to-1 multiplexer, as well as some logic to control the value of the 2-bit signal M1.

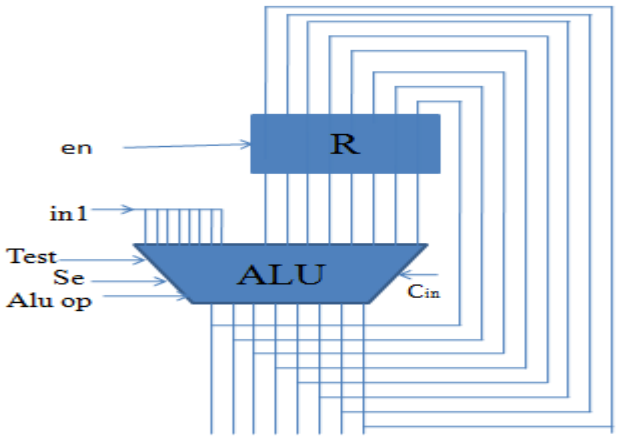
3. PROPOSED SCHEME

In this work we propose an ALU based address generator; the proposed scheme assumes an accumulator-like structure, like the one shown in Figure 3(a).

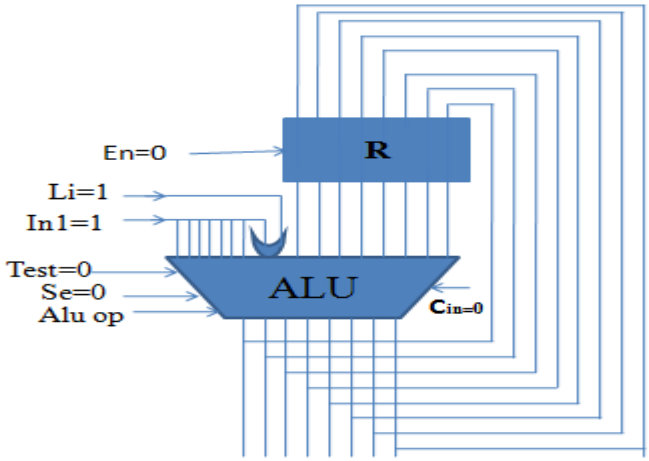
In computing, an Arithmetic and Logic Unit (ALU) is a digital circuit that performs arithmetic and logical operations. The ALU is divided into two different units that perform two different operations. The Arithmetic unit performs the arithmetic operations such as addition, subtraction, multiplication and division where as the logic unit performs the bit-wise AND, OR, XOR, NOT, NAND, NOR.

For memory testing, the address generator has to generate different Counting Methods since each counting method has its own detection. The Linear counting method specifies the address sequence 0, 1, 2, 3... N-1 when going up and N-1 ...3, 2, 1, 0 when going down. The Address Complement (AC) counting method specifies an address sequence: 0000000, **11111111**, 00000001, **11111110**, 00000010, **11111101**, etc. Each bold address is the one's complement of the previous address, as shown in

column denoted 'AC' in Table 1. The AC counting method stresses the address decoders, because all address bits switch upon address transitions. In the project while implementing the linear address, the arithmetic unit which can perform only the addition operation is required. There will be three data inputs for the ALU block that is given by Data in, Cin and from the Register (R) block.



(a)



(b)

Figure3. *The proposed scheme for the generation of the counting methods (a) Address Complement (b) combined Linear and Address Complement*

The select signal to ALU block is provided by the Add/Sub that performs the required operation; the block diagram for the ALU based address generation module is given in Figure 3 Here, the addition operation is required. The first bit of the data in is connected from the output of a multiplexer which has inputs '1' and Datain (0) with a select signal. When the linear address has to be generated then the select line selects the '1' input. The output of the ALU is given to a register that stores the output and gives the output as input to the ALU which is the second input data. The data has to be stored in these memory locations of the memory.

For the address complement counting method, the signal denoted Li is constantly 0. The operation of the proposed scheme for the cases utilize the following signals the 'in-signal' provides the value to be fed to the one input of the ALU, the selection decides on the addition or subtraction to be performed by the ALU the en signal enables the Register (R) to capture the ALU output; when en=0, the register

does not change value. The Cin signal provides the carry input to the ALU. For the case where a combined Linear or Address Complement address generator is required, an additional signal

Table 2. *Address Complement Counting Method.*

Cycle#	In	Se	R	En	Cin	ALU
0	0	1	00000000	1	0	00000000
1	1	0	00000000	0	0	11111111
2	0	1	00000000	1	1	00000001
3	1	0	00000001	0	0	11111110
4	0	1	00000001	1	1	00000010
5	1	0	00000010	0	0	11111101
6	0	1	00000010	1	1	00000011
7	1	0	00000011	0	0	11111100
8	0	1	00000011	1	1	00000100
9	1	0	00000100	0	0	11111011
10	0	1	00000100	1	1	00000101
11	1	0	00000101	0	0	11111010
12	0	1	00000101	1	1	00000110
13	1	0	00000110	0	0	11111001
14	0	1	00000110	1	1	00000111
15	1	0	00000111	0	0	11111000

Li or Ac enables the multiplexer to select the low-order input of the ALU between the in and the steady 1 signal For the case of the combined Linear or Address Complement address generator shown in, when linear count is required the Li or Ac signal enables the low order of the ALU to be constantly to 1,the in signal is constantly to 0, the en signal is enabled in every cycle and the C_{in} signal is constantly 0.The selection is 0 or 1 depending on whether up or down linear count is required.

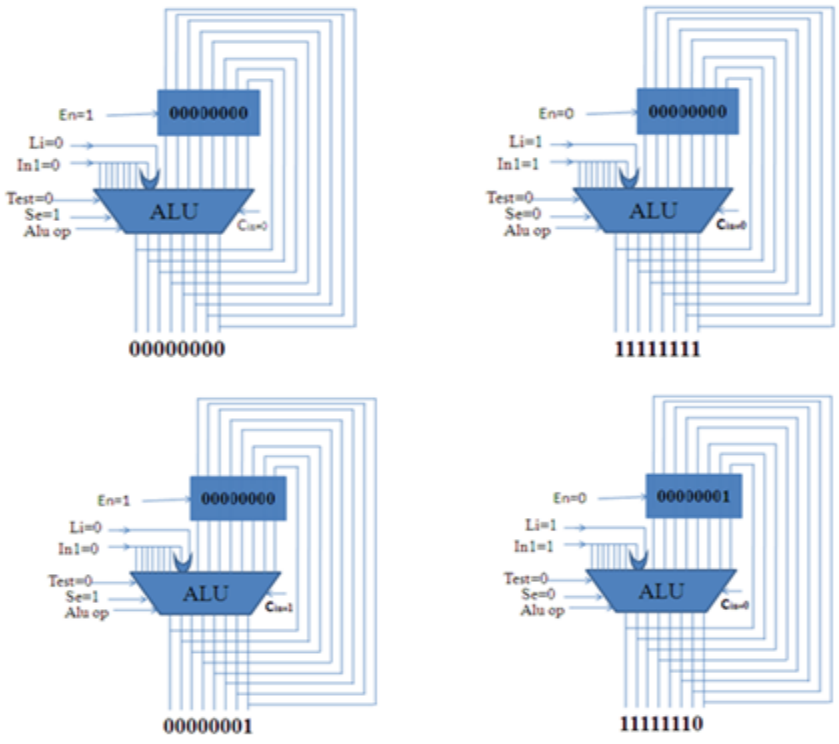


Figure4. *Operation of the Proposed Scheme for Address Complement Generation and Linear Address Generation*

4. COMPARISONS AND DISCUSSION OF RESULTS

Assuming the existence of the ALU-based accumulator-like structure of Figure 1, the hardware overhead of the proposed scheme consists of the multiplexers required to multiplex between the normal inputs of the ALU and the test input 'in'. This is also the case for the scheme proposed in [4]. For the case where both linear and AG addressing is targeted (as is the case in modern memory BIST schemes), the proposed scheme requires an additional multiplexer to multiplex between the 'in' and '1' signals in the low-order input of the RAM. The scheme proposed in [4] requires n 4-input multiplexers. Since a 4-to-1 multiplexer is typically 3 2-to-1 multiplexers, it is trivial to see that the hardware overhead of the scheme [4] requires 3 times more than the proposed here. An additional merit of the proposed scheme is that in typical configuration (like the MIPS architecture shown in Figure 1) an ALU driving the address inputs of the memory is either already available, or can be easily incorporated in the design while counter modules usually must be inserted in the design of the circuit.

The simulation results of MIPS architecture and ALU are shown in the figures 5 and 6 respectively.

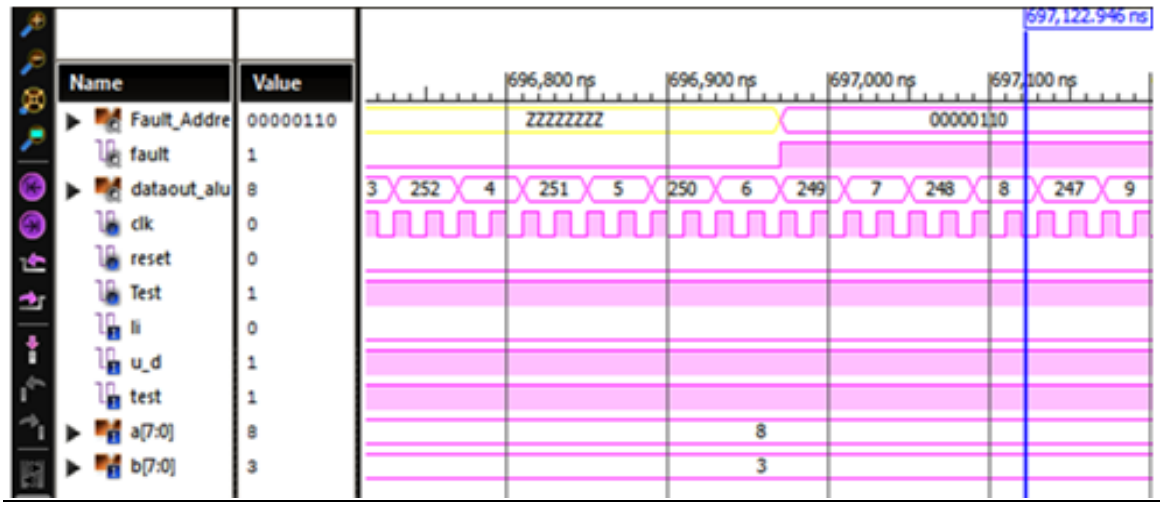


Figure5. *Simulation results of the MIPS architecture with detection of fault*

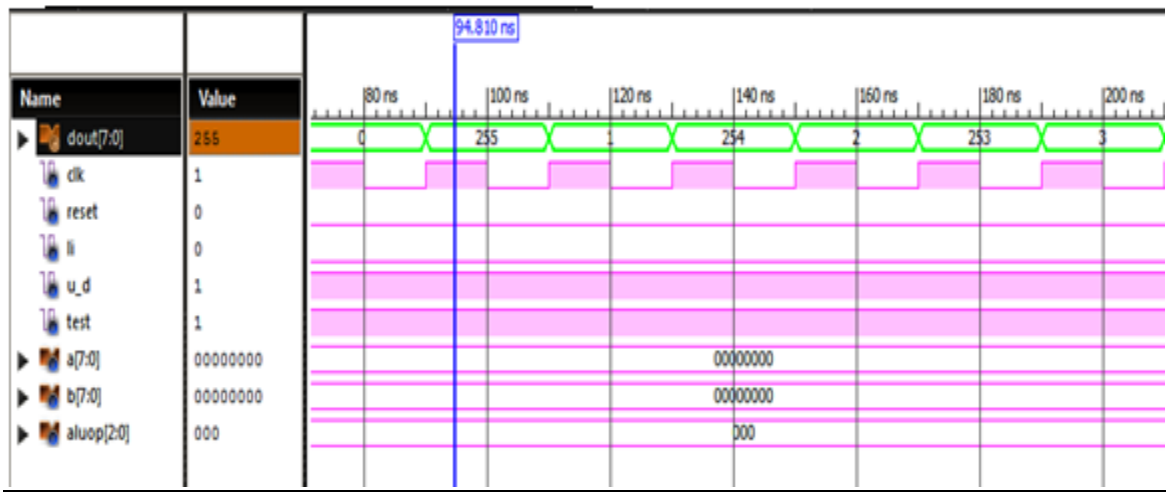


Figure6. *Simulation results of the ALU with Linear and Complement Address Generation*

The synthesis results and device utilization summary of MIPS architecture are shown in the figure7 and table3 respectively.

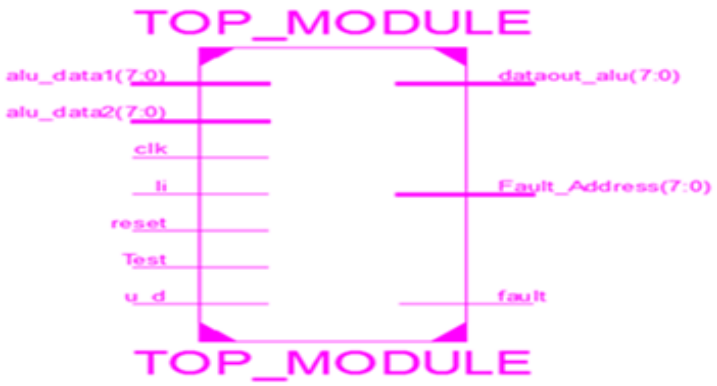


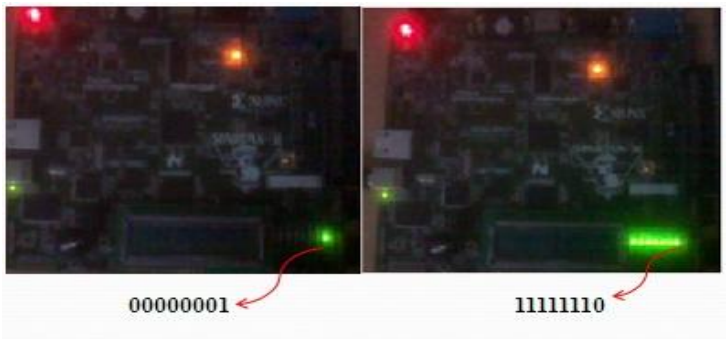
Figure7. *Synthesis results of the MIPS architecture*

Table 3. *Device Utilization Summary results of the MIPS architecture*

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Registers	2178	42000	5%
Number of Slice LUTs	6060	21000	28%
Number of fully used LUT-FF pairs	2102	6136	34%
Number of bonded IOBs	20	210	9%
Number of BUFG/BUFGCTRLs	2	32	6%

The FPGA implemented output of ALU on Spartan-3e is shown in the figures 8(a) and (b).

(a)



(b)

Figure8. *Output of ALU based Linear and Complement Address on Spartan 3E FPGA*

5. CONCLUSIONS

In this work we have proposed the utilization of ALU modules for the address generation for memory BIST. The proposed generator can generate the linear and address complement counting methods. The proposed method evolves naturally in systems where the address inputs of memories are driven by ALU modules. Comparatively to a previously proposed scheme based on counters, the proposed one presents lower hardware overhead when the combined power of linear and address complement counting methods are required.

REFERENCES

[1] R. Aitken, et. al, 'Modular Wrapper Enabling High Speed BIST and Repair for Small Wide Memories', Proc. of Int. Test Conference, pp. 997-1005, 2004.

[2] 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.

[3] X. Du, N. Mukherjee, 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.

[4] Ad J. van de Goor, H. Kukner, S. Hamdioui, "Optimizing Memory BIST Address Generator Implementations", Proceedings of the 2011 Conference on Design and Technology of Integrated Systems in nanoscale era, Athens, Greece, April 2011.

[5] Y. Park, J. Park, T. Han, and S. Kang, An Effective Programmable Memory BIST for Embedded Memory, IEICE Transactions on Information and Systems E92-D (2009), no. 12, 25082511.

[6] A.J. van de Goor, S. Hamdioui and G. N. Gaydadjiev, 'New Algorithms for Address Decoder Delay Faults and Bit Line Imbalance faults', Proc. of the 18th Asian Test Symposium, pp. 31-36, 2009.

[7] H. Kukner, 'Generic and Orthogonal March Element based Memory BIST Engine' Master Thesis, CE-MS-2010-01, Delft University of Technology, September 2010.

[8] A. van de Goor, C. Jung, S. Hamdioui and G.N. Gaydadjiev, Low-cost, Customized and Flexible SRAM MBIST Engine, In Proc. of the IEEE Int. Symp. on Design and Diagnostics of Electronic Circuits and Systems, Vienna, 2010, pp. 382-387.

[9] L. Dilillo, et.al, 'Dynamic read destructive fault in embedded-SRAMs:analysis and march testsolution', Proceedings Ninth IEEE European Test Symposium, pp. 140-145, 2004.

[10] A.J. van de Goor, Testing Semiconductor Memories: Theory and Practice, ComTex Publishing, The Netherlands, 1998,Ad.vd.Goor@kpnplanet.nl.

AUTHORS' BIOGRAPHY

Mr. Nookala SaiRam, currently pursuing M.Tech VLSI-System Design, ECE Department, CVR College of Engineering and received a graduation degree from Annamacharya Institute of Technology and Sciences in 2008.

Mrs. Athira G. Krishna, currently working as a Assistant professor in CVR College of Engineering. She completed her Post Graduation.