International Journal of VLSI System Design and Communication Systems

ISSN 2322-0929
Vol.02, Issue.07,
October-2014,
Pages:0512-0517

www.ijvdcs.org

Memory Testing using March C-Algorithm

M.MAMATHA[1], M.MURALIDHAR[2]

[1]PG Scholar, Dept of ECE, NEC, Nellore, Andhrapradesh, India, Email: mamatha418@gmail.com.
[2]Professor, Dept of ECE, NEC, Nellore, Andhrapradesh, India, Email: murali_dhar@yahoo.co.in.

Abstract: Built In Self Test (BIST) is the technique of designing additional hardware and software features into an Electronic System to allow them to perform self testing. The basic idea of BIST is to design a circuit that can test itself and determine whether the circuit is fault-free or faulty. BIST is mainly used for background checking of memory without stopping the actual functionality of a system and it is an online test scheme. BIST checks the errors and also correct the errors sometimes. Some of the applications of BIST are RFID integrity check and avionics systems. One of the applications of BIST is Radio Frequency Identification (RFID). RFID is the wireless non contact use of radiofrequency electromagnetic fields to transfer data, for the purpose of automatically identifying and tracking tags attached to objects/persons and contain electronically stored information. My goal is to simulate MARCH C- algorithm for memory testing(Xilinx) and imitatiting various types of memory faults in the FPGA itself for example Stuck at faults(SAF) and coupling faults will be studied.

Keywords: Built In Self Test (BIST), Simulate MARCH C- Algorithm ,Stuck at Faults(SAF) And RFID.

I. INTRODUCTION

Embedded Memories are growing rapidly to a large amount in terms of its size and density. As Embedded memories are using complex design structures the chances of occurring manufacturing defects is more compared to any other embedded core on SOC. Hence testing of embedded memory is a real challenge for design architect. For SOC the inability to have direct access to a core is one of the major problems in testing and diagnosis .Further the available bandwidth between the primary inputs of the system chip and the embedded core is usually limited. Hence the external access for test purpose is often infeasible. This has prompted a very strong interest in self test of embedded arrays. In particular, functional March tests have found wide acceptance, mostly because they provide defined detection properties for classical memory array faults such as stuck at faults and transition faults. Memory tests are used to confirm that each location in a memory device is working. This involves writing a set of data to each memory address and verifying this data by reading it back. If all the values read back are the same as those that were written, then the memory device is said to pass the test, otherwise device fails. Different test methodologies have been evolved from the years to identify the memory defects, one such test is memory built in self test which involves built in self test circuitry for each memory array.

The advantage of March tests lay in the fact that high fault coverage can be obtained and the test time were usually linear with the size of the memory which makes it acceptable from industrial point of view. March based algorithms were capable of locating and identifying the fault types which can help to catch design and manufacturing errors. Especially SAF dominate the majority of defects that occur in embedded RAMS. Due to the rapid progress in the very large scale integrated (VLSI) technology and large number of transistors can be fabricated onto a single silicon die. Although million gate the increased chip complexity requires robust and sophisticated test methods. Hence, manufacturing test is becoming an enabling technology that can improve the declining manufacturing yield, as well as control the production cost, which is on the rise due to the escalating volume of test data and testing times. Therefore reducing the cost of manufacturing test, while improving the test quality required to achieve higher product reliability and manufacturing yield, has already been established as a key task in VLSI design.

A. Role of Testing

While designing a product, fabricate and test it, and it fails the test, then there must be a cause for the failure. Either (1) the test was wrong, or (2) the fabrication process was faulty, or (3) the design was incorrect, or (4) the specification had a problem. Anything can go wrong. The role of testing is to detect whether something went wrong and the role of diagnosis is to determine exactly what went wrong, and where the process needs to be altered. Therefore, correctness and effectiveness of testing is most important for quality products (another name for perfect products.) If the test procedure is good and the product fails, then it can suspect the fabrication process, the design, or the specification. The benefits of testing are quality and economy. These two attributes are not independent and neither can be defined without the other. Quality means satisfying the user's needs

at *a* minimum cost. A good test process can weed out all bad products before they reach the user. However, if too many bad items are being produced then the cost of those bad items will have to be recovered from the price charged for the few good items that are produced. It will be impossible for an engineer to design a quality product without a profound understanding of the physical principles underlying the processes of manufacturing and test.

B. VLSI Testing Process

VLSI chip testing is done in several different places by several different types of people. When a new chip is designed and fabricated for the first time, testing should verify correctness of design and the test procedure. This often requires the involvement of the design engineer and the testing may even take place in the design laboratory rather than in a factory. Based on the result, both the design and the test procedure may be changed. This is called verification testing. Successful verification testing usually results in some good chips. These are the earliest chips and are normally used by the designers of systems that will use this design. A successful verification also signals the beginning of production. Production means large scale manufacturing. Fabricated chips are tested in the factory. This is called manufacturing testing. Finally, when the manufactured chips are received by a customer, they may be again tested to ensure quality. This testing, known as incoming inspection (or acceptance testing), is conducted either by the user or for the user by some independent testing house.

C. Digital Test Methodologies: ATE vs. BIST

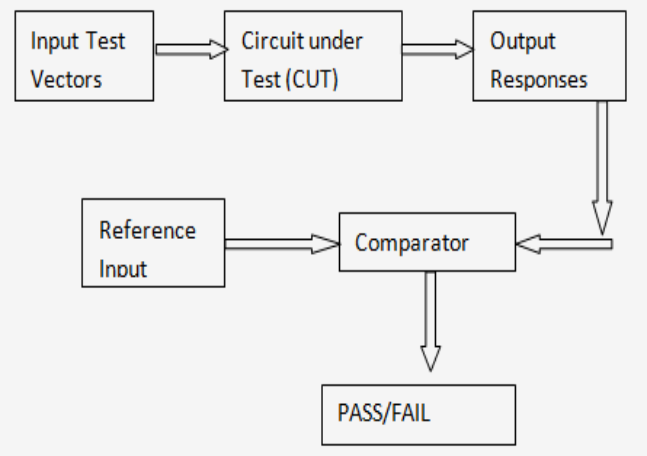


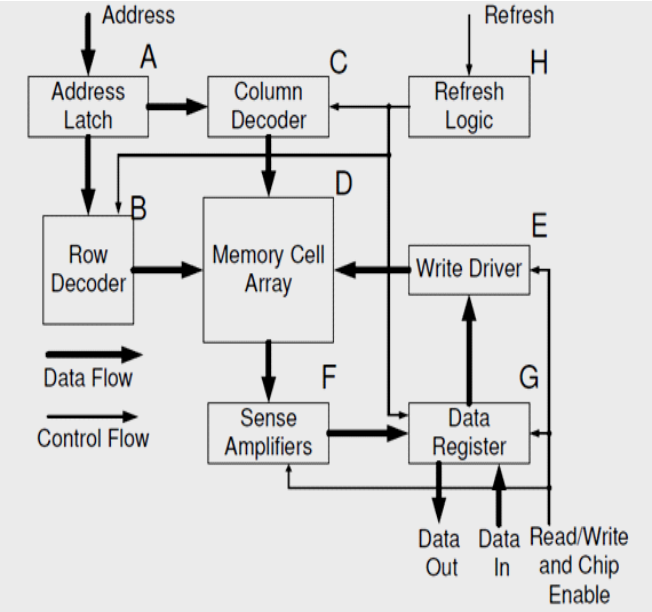
Figure1. Basic Principle of Digital Testing

The basic principle of manufacturing testing is illustrated in Figure1 Circuit under test (CUT) can be the entire chip or only a part of the chip (e.g., a memory core or a logic block). Input test vectors are binary patterns applied to the inputs of the CUT and the associated output responses are the values observed on the outputs of the CUT. Using a comparator output responses are checked against the expected correct response data, which is obtained through simulation prior to design tape-out. If all the output responses match the correct response data, the CUT has passed the test and it is labeled as fault-free. Based on the techniques how the test vectors are applied to the CUT and how the output responses are compared. There are two main directions to test electronic circuits: external testing using automatic test equipment (ATE) and internal testing using built-in self-test (BIST). When external testing is employed, the input test vectors and correct response data are stored in the ATE memory. Input test vectors are generated using ATPG tools, while correct response data is obtained through circuit simulation.

For external testing the comparison is carried out on the tester. Although the ATE-based test methodology has been dominant in the past, as transistor to pin ratio and circuit operating frequencies continue to increase, there is a growing gap between the ATE capabilities and circuit test requirements (especially in terms of speed and volume of test data).ATE limitations make BIST technology an attractive alternative to external test for complex chips. BIST is a design-for-test (DFT) method where part of the circuit is used to test the circuit itself (i.e., test vectors are generated and test responses are analyzed on-chip). BIST needs only an inexpensive tester to initialize BIST circuitry and inspect the final results (pass/fail and status bits). However, BIST introduces extra logic, which may induce excessive power in the test mode, in addition to potential performance penalty and area overhead. BIST circuitry can further be divided into logic BIST for random logic blocks (e.g., control circuitry or data path components) and memory BIST for on-chip memory cores. The cost and quality of logic BIST has been subject to extensive research over the last two decades and, since the focus of this thesis is on embedded memory BIST, the reader is referred to for more information.

II. MEMORY TESTING

There are two kinds of memory test methods: electrical (technology-dependent) and functional (technology-independent). Electrical memory testing consists of parametric testing, which includes testing DC and AC parameters, IDDQ and dynamic testing for recovery, retention and imbalance faults. DC and AC parametric tests are used to verify that the device meets its specifications with regard to its electrical characteristics, such as voltage, current, and setup and hold time requirements of chip's pins. Since embedded memories in SOCs usually do not have their I/O ports directly connected to chip's pins, parametric testing for embedded memories is not a necessity. IDDQ and dynamic testing need a detailed description of the specific process technology. This thesis focuses on technology-independent functional memory testing, whose purpose is to verify the logical behavior of a memory core. Because functional memory testing allows for the development of cost-effective short test algorithms (without requiring too much internal knowledge of the memory under test), it is widely accepted by industry as a low-cost/high-quality solution. This chapter provides a theoretical background and explains the memory functional test models and March algorithms.



Figur2: Functional Memory Module.

III.EXISTING SYSTEN

A. Built In Self Test

In a digital instrument designed for troubleshooting by signature analysis, this method can find the components responsible for well over 99% of all failures, even intermittent Built-in Self Test, or BIST, is the technique of designing additional hardware and software features into integrated circuits to allow them to perform self-testing, i.e., testing of their own operation (functionally, parametrically, or both) using their own circuits, thereby reducing dependence on an external automated test equipment (ATE). BIST is a Design-for-Testability (DFT) technique, because it makes the electrical testing of a chip easier, faster, more efficient, and less costly. The concept of BIST is applicable to just about any kind of circuit, so its implementation can vary as widely as the product diversity that it caters to.The basic idea of BIST, in its most simple form, is to design a circuit so that the circuit can test itself and determine whether it fault-free or faulty. This typically requires that additional circuitry and functionality be incorporated into the design of the circuit to facilitate the self-testing feature. This additional functionality must be capable of generating test patterns as well as providing a mechanism to determine if the output responses of the circuit under test (CUT) to the test patterns correspond to that of a fault-free circuit. Basic approach of testing is shown in the fig3.

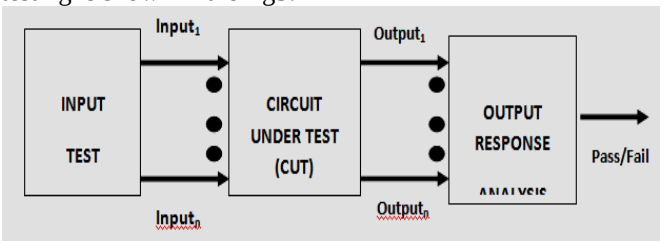


Figure3: Basic approach of Testing and boards

IV. VARIOUS MBIST ALGORITHMS

A. Classical Test Algorithms

Classical test algorithms are either (1) simple, fast but have poor fault coverage, such as Zero-one, Checkerboard; or (2) have good fault coverage but complex and slow, such as Walking, GALPAT, Sliding Diagonal, Butterfly, MOVI, and etc.. Due to these imbalanced conflicting traits, the popularity of these algorithms is decreasing.

B. March-based Test Algorithms

A March-based test algorithm is a finite sequence of March elements. A March Element is specified by an address order and a number of reads and writes. Examples of some March-based tests are MATS, MATS+, Marching 1/0, March C-, March Y, March A, March B, and etc.. Since March-based tests are all simple and possess good fault coverage, they are the dominant test algorithms implemented in most modern memory BIST. The basic BIST architecture requires the addition of three hardware blocks to a digital circuit: a test pattern generator, a response analyzer, and a test controller. The test pattern generator (TPG) automatically generates test patterns for application to the inputs of the circuit under test (CUT). The output response analyzer (ORA) automatically compacts the output responses of the CUT into a signature. Specific BIST timing control signals, including scan enable signals and clocks, are generated by the BIST controller for coordinating the BIST operation among the TPG, CUT, and ORA. The BIST controller provides a pass/fail indication once the BIST operation is complete. The ORA compacts the output responses of the CUT to the many test patterns produced by the TPG into a single Pass/Fail indication (usually a multiple-bit "signature").

The ORA is sometimes referred to as an output data compaction(ODC) circuit . The significance of the ORA is that there is no need to compare every output response from the CUT with the expected output response external to the device. Only the final Pass/Fail indication needs to be checked at the end of the BIST sequence in order to determine the fault-free/faulty status of the CUT. Comparison-based ORAs use a comparator to detect mismatches in the fault-free and faulty circuits. An example of a comparator-based approach is the simple BIST where the expected responses were stored in a ROM and compared on a vector-by-vector basis to the output responses of the CUT. Signature analysis is the most commonly used technique for ORAs in BIST implementations. Signature analysis uses an LFSR as the primary component of the ORA implementation. The basic idea behind signature analysis is to divide the polynomial representing the output response of the CUT by the characteristic polynomial of the LFSR used to implement the ORA. The resultant "signature" is the remainder of the polynomial division and is compared to the signature for the fault-free circuit at the end of the BIST sequence.

Figure4 shows the BIST system hierarchy for the 3 level of packaging which is the system level, board level and chip

level. The system has several PCBs, each of which, in turn, has multiple chips. The system Test Controller can activate self-test simultaneously on all PCBs. Each Test Controller on each PCB can activate self-test on all chips on the PCB. The Test Controller on a chip executes self-test for that chip, and then transmits the result to the PCB Test Controller, which accumulates test results from all chips on the board and sends the results to the system Test Controller. The system Test Controller uses all of these results to isolate faulty chips.

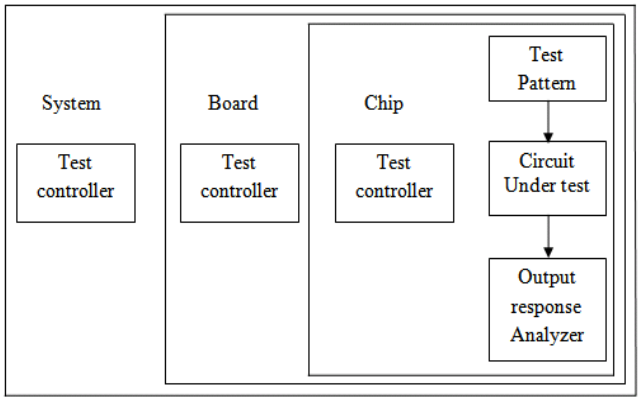


Figure4: BIST hierarchy

V.PROPOSED SYSTEM

A. March Test Algorithm

Many algorithms have been developed for testing semi-conductor memories which are shown in the Table1 & 2, from which the most popular and advantageous are the March tests. A March test contains a sequence of March elements which is composed by a read/write operation that have to be performed into every cell of the memory. March tests are able to detect several fault models such as Stuck-at Faults (SAF), Address Faults (AF) and some Coupling Faults (CF).The operations that can be executed in the cells may be: write zero (w0), write one (w1), read zero (r0) and read one (r1). The read operation checks if the value inside the cell is the expected one. The order in which cells are considered can be ascending or descending. A typical march test used to test RAMs is MATS++ which can be adapted to test also EEPROMs. It has the three march elements M0: ↕ (w0); Ml: ↑ (r0; w1); and M2: ↓ (r1; w0; r0); these are written with commas or semicolons separating them, and the entire march sequence is enclosed in braces.

Table1. Irredundant March Test Algorithms

Algorithm	Description
MATS	$\{ \Updownarrow (w0); \Updownarrow (r0, w1); \Updownarrow (r1) \ \}$
MATS+	$\{ \Updownarrow (w0); \Uparrow (r0, w1); \Downarrow (r1, w0) \}$
MATS++	$\{ \Updownarrow (w0); \Uparrow (r0, w1); \Downarrow (r1, w0, r0) \}$
MARCH X	$\{ \Updownarrow (w0); \Uparrow (r0, w1); \Downarrow (r1, w0); \Updownarrow (r0) \}$
MARCH C–	$\{ \Updownarrow (w0); \Uparrow (r0, w1); \Uparrow (r1, w0); \Downarrow (r0, w1); \Downarrow (r1, w0); \Updownarrow (r0) \}$
MARCH A	$\{ \Updownarrow (w0); \Uparrow (r0, w1, w0, w1); \Uparrow (r1, w0, w1); \Downarrow (r1, w0, w1, w0); \Downarrow (r0, w1, w0) \}$
MARCH Y	$\{ \Updownarrow (w0); \Uparrow (r0, w1, r1); \Downarrow (r1, w0, r0); \Updownarrow (r0) \}$
MARCH B	$\{ \Updownarrow (w0); \Uparrow (r0, w1, r1, w0, r0, w1); \Uparrow (r1, w0, w1); \Downarrow (r1, w0, w1, w0); \Downarrow (r0, w1, w0) \}$

Table2. Irredundant March Test Summary

	Fault Coverage						
Algorithm	**SAF**	**AF**	**TF**	**CF in**	**CD id**	**CF dyn**	**SCF**
MATS	All	Some					
MATS+	All	All					
MATS++	All	All	All				
MARCH X	All	All	All	All			
MARCH C-	All	All	All	All	All	All	All
MARCH A	All	All	All	All			
MARCH Y	All	All	All	All			
MARCH B	All	All	All	All			

All operations of a March element are done before proceeding to the next address. The MATS++ algorithm is described as follows:

↕ (w0); ↑ (r0; w1); ↓ (r1; w0; r0)

Figure4 presents the interpretation of this notation for MATS++ as a testing algorithm. Word-oriented memories, such the ones found in an RFID, need a slightly different approach. By extending the 0 or 1 to 16 bits, March algorithm can be easily applied to RFID's word-oriented memories with a reduction on the coverage of CF.

```
M0:   {March element ↕ (w0)}
      for cell:=0 to n-1(or any other order) do
      begin
      Write 0 to A [cell];
      end;

M1:   {March element ↑ (r0,w1)}
      for cell:=0 to n-1 do
      begin
      read A [cell];{ Expected Value=0}
      write 1 to A [cell];
      end;
M1:   {March element ↓ (r1,w0)}
      for cell:= n-1 down to 0 do
      begin
      read A [cell];{ Expected Value=1}
      write 0 to A [cell];
      end;
```

B. Characteristics of March Algorithms

March-based memory test algorithms have several important characteristics:

- Up (down) address sequence must be the exact reverse down (up) sequence; however its internal order is irrelevant. For example, if a 3 bits up address sequence is {0, 5, 2, 3, 7, 1, 4, 6}, then the down sequence must be {6, 4, 1, 7, 3, 2, 5, 0}.
- Most March algorithms are only a simple combination of several March elements (e.g., (r0, w1) is a March element). By analyzing the March algorithms shown in Table 5, it can be observed that the background pattern during execution can be inferred by the previous operation. For example, a read operation infers the same

background data used in the last operation. Similarly, a write operation infers the reversed background data used in the last operation. For example, the first operation of the March C- test shown in Table 4 is w0. The next operation is read (must be r0) and the following operation is write (must be w1). Based on this observation, one can reduce the number of March elements and the complexity of their implementation. For March C-, only three March elements are needed: (w), (r, w), (r). The total number of March elements for the most practical March algorithms is less than ten.

- One can generate novel March algorithms (based on a limited number of March elements implemented in hardware), to detect new technology-specific faults.• For word-oriented memories, one needs to run the March test several times using different background patterns to improve the fault coverage or to use modified March algorithms, such as March-CW, to reduce the testing time.

C. March Algorithms with Diagnosis and Repair Support

When a memory is fabricated using new technology, it is desirable to have a fast yield learning curve. Therefore, it is critical to perform very detailed failure analysis through fault diagnosis to identify the particular defects (for example, it is essential to distinguish faults between SAF and CF). The ultimate outcome of failure analysis is a redesigned set of masks for the next fabrication run, which will improve the manufacturing yield. A new set of March algorithms will be used for the best process-specific fault coverage. For large memory chips or SOCs with large embedded SRAMs or DRAMs, to increase the yield, it is crucial to also use redundant memory locations to repair the faulty rows (columns). This leads to new type of algorithms, called fault location algorithms. This type of algorithms can, for example, locate the aggressor cell of a coupling fault (CF).

IV.SREEN SHOTS

The behavioral simulation for the FSM without fault in the memory locations is shown in figure 5

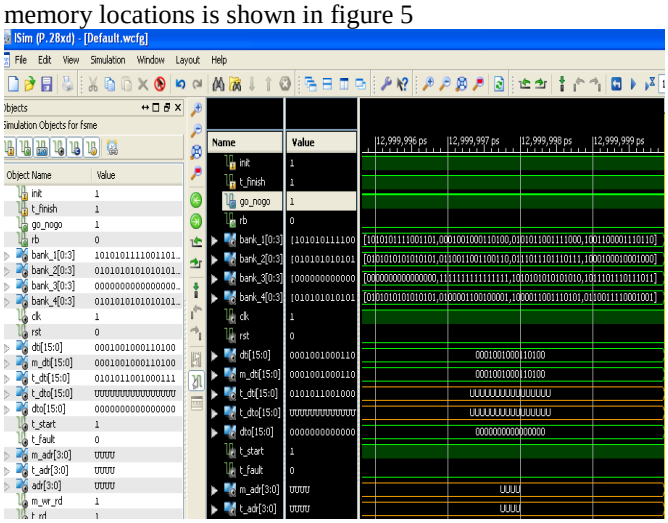


Figure5: Simulation waveform for the FSM without fault in the memory location.

The behavioral simulation for the FSM with fault in the memory location is shown in figure6.

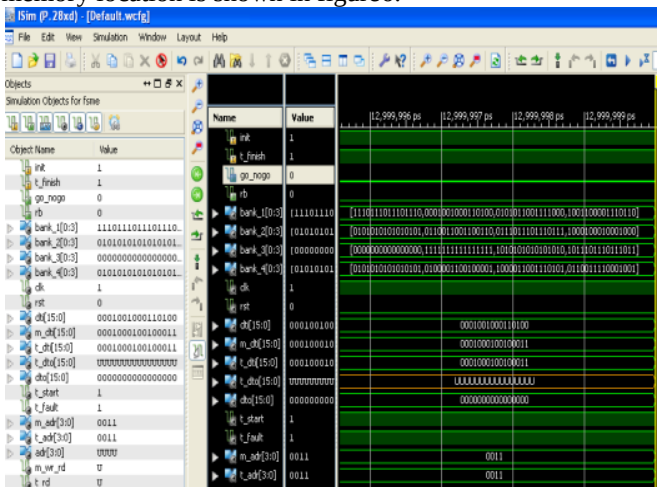


Figure6. Simulation waveform for the FSM with fault in the memory location.

V. CONCLUSION & FUTURE SCOPE

A. CONCLUSION

An implementation of March Test Algorithms for supporting online test in memory was presented. The new method takes advantage of the idle state of system/equipment while waiting to be accessed by the interrogator to perform the test of their internal memory. The BIST finite state machine describing the access scheme was presented and the architecture of the transparent BIST circuit was described. Various module level test of the algorithm is checked and found working. There were some integration problems that need to be addressed for the complete working of MARCH C- algorithm as BIST.

B. FUTURE SCOPE

Future work will include other testing approaches which provide a direct testing command to the interrogator and a larger list of supported march algorithms. In future it can also be used to test stand alone memories like processors and to detect DRDFs, WDFs etc by improving this MARCH C- algorithm.

VI. REFERENCES

[1] Erwing R.Sanchez and Maurizio Rebaudengo,"ANovel Access Scheme for online Test in RFID Memories",IEEE - 2011.

[2] J.McDonnell, J. Waters, H. Balinsky, R. Castle, F. Dickin, W. W. Loh, and K. Shepherd, "Memory spot: A labeling technology," Pervasive Computing, IEEE, vol. 9, no. 2, pp. 11–17, april-june 2010.

[3] E. Welbourne, L. Battle, G. Cole, K. Gould, K. Rector, S. Raymer,M. Balazinska, and G. Borriello, "Building the internet of things using rfid: The rfid ecosystem experience, Internet Computing, IEEE, vol. 13, no. 3, pp. 48–55, may-june 2009.

[4] EPCGlobal, EPC radio-frequency identity protocols Class-1 Generation-2 UHF RFID air interface Version 1.2.0, Oct. 2008.

[5] T. Cheng and L. Jin, "Analysis and simulation of rfid anti-collision algorithms," in Advanced Communication Technology, The 9th International Conference on, vol. 1, 2007, pp. 697–701.
[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].
[7] R. Dekker, F. Beenaker, L.Thijssen, "A realistic fault model and test algorithm for static random access memories", IEEE Trans. Computer-Aided Design, vol,9,99.567-572, june 1990.
[8] J. van de Goor. Testing Semiconductor Memories: Theory and Practice.A.J.vandeGoor,1998.
[9] Verilog Digital System Desigh, 2nd Edition, Zainalabedin Navabi, Tata McGraw Hill, 2008
[10] A.J.Van de Goor, Testing Semiconductor Memories, Theory and Practice, John Wiley & Sons, Chichester, England, 1991.
[11] Van De Goor, A.J. Sehanstra. I., "Address and Data Scrambling: Causes and Impact on Memory Tests", IEEE DELTA Workshop, Christchurch, January 2002.
[12] ER.Manoj Arora, Er.Shipra Tripathi, " Comparative Simulation of MBIST using March Test Algorithms", International Journal of Scientific & Engineering Research, Volume 2, Issue 12, December-2011. mu;w minimum rf input power," Solid-State Circuits, IEEE Journal of, vol. 38, no. 10. pp. 1602 – 1608, 2003. nd Exhibition, vol. 0, p. 702, 1999.