MODIFIED MARCH C- WITH CONCURRENCY IN TESTING FOR EMBEDDED MEMORY APPLICATIONS

Muddapu Parvathi [1] , N.Vasantha [2] and K.Satya Parasad [3]

[1]Department of Electronics and Communication , M.R.I.T.S, Hyderabad, India.
bendalamieee@gmail.com
[2]Department of Information Technology, V.C.E, Hyderabad, India.
[3]Department of Electronics and Communication, J.N.T.U.K, Kakinada, India.

ABSTRACT

March algorithms are known for memory testing because March-based tests are all simple and possess good fault coverage hence they are the dominant test algorithms implemented in most modern memory BIST. As March algorithms are well known algorithms for testing embedded RAMS, out of which March C- is known for finding all SAF, SOF, CF. This March C- is used frequently in the industry also. The proposed march algorithm is modified march c- algorithm which uses concurrent technique. Using this modified march c- algorithm the complexity is reduced to 8n as well as the test time is reduced greatly. Because of concurrency in testing the sequences the test results were observed in less time than the traditional March tests. This technique is applied for a memory of size 256x8 and can be extended to any memory size.

KEYWORDS

Embedded RAMS, March c-, Modified March c- algorithm, concurrent technique, complexity, traditional March tests.

1. 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 [1]. 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.

DOI : 10.5121/vlsic.2012.3504

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[6]. 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.

The method proposed in this paper is Modified March C- algorithm with concurrent technique. This algorithm retains the high fault coverage of March C but at reduced time the tests can be done. The paper further describes the functional fault models in the memory, classical and March based tests in section II. The proposed Modified March c- algorithm and corresponding hardware implementation was discussed in section III. Results and comparisons were discussed in section IV. Conclusions were given in section V.

2. HISTORY OF FUNCTIONAL FAULT MODELS

For testing purpose the functional fault models are modelled after faults in memories so that functional tests to detect these faults can be used. This modelling helps to clarify, simplify and generalize the testing approach of a memory. The quality of tests is strongly dependent on the fault model in terms of its fault coverage, its test length as well as the test time required.

There are various fault models to test the functional faults such as stuck at faults; coupling faults are considered when it deals with SRAM. Address decoder faults and bridging faults will be considered when it deals with DRAM. Hence the most possible faults which occur in general are stuck at faults.

Stuck at fault (SAF) : The stuck-at fault (SAF) considers that the logic value of a cell or line is always 0 (stuck-at 0 or SA0) or always 1 (stuck-at 1 or SA1). To detect and locate all stuck-at faults, a test must satisfy the following requirement: from each cell, a 0 and a 1 must be read.

Transition Faults(TF): The transition fault (TF) is a special case of the SAF. A cell or line that fails to undergo a 0 → 1 transition after a write operation is said to contain an up transition fault. Similarly, a down transition fault indicates the failure of making 1 → 0 transitions. According to van de Goor [2], a test to detect and locate all the transition faults should satisfy the following requirement: each cell must undergo an ↑ transition (cell goes from 0 to 1) and a ↓ transition (cell goes from 1 to 0) and be read after each transition before undergoing any further transitions.
The fault detection for both SAFs and TFs will be done by considering MATS++ algorithm and March C- algorithm. Although different in test length, these tests are capable of detecting both faults while being capable of detecting other faults as well. The detection process can be understood by examining the Mach C- algorithm as indicated in expression below.

March Test Notation:

A March test consists of a finite sequence of March elements [2][3]. A March element is a finite sequence of operations or primitives applied to every memory cell before proceeding to next cell. For example, ↓(r1, w0) is a March element and r0 is a March primitive. The address order in a March element can be increasing (↑), decreasing (↓), or either increasing or decreasing (↕). An operation can be either writing a 0 or 1 into a cell (w0 or w1), or reading a 0 or 1 from a cell (r0 or r1). Accordingly notation of March C- test is described as follows:

$$\{\updownarrow(w0);\uparrow(r0,w1);\uparrow(r1,w0);\downarrow(r0,w1);\downarrow(r1,w0);\downarrow(r0)\}$$

March C- algorithm has 6 elements as shown with a complexity of 10n. As it uses both directions of addressing it can find Address decoder Faults. From the first two March elements SA1 fault can be identified from a particular address. Similarly from second and third elements SA0 faults can be identified. From third and fourth elements one can find coupling faults also. It is a simple test algorithm approved and used frequently by the industry for testing embedded memories.

3. MODIFIED MARCH C- ALGORITHM AND PROPOSED HARDWARE IMPLEMENTATION

The proposed Modified March C- algorithm almost similar to March C- but it follows concurrency in testing the sequences. The steps for following the concurrency are as follows:

- Group entire memory into subgroups.
- For each subgroup, generate all test vectors for the first fault in the group.
- Simulate all faults in the subgroup to select one vector that detects most faults in that subgroup. If more vectors than one detect the same number of faults within the group, then select the vector that detects most faults outside the group as well.
- Apply the final test vectors to all subgroups concurrently

In the proposed method the memory is divided in to two subgroups such as M1 and M2. Then applied the algorithm for concurrency. The following are the elements in Modified March C- algorithm.

$$M1: \{\uparrow(w0);\uparrow(r0,w1);\uparrow(r1);\downarrow(w0);\downarrow(r0,w1);\downarrow(r1)\}$$
$$M2: \{\uparrow(w1);\uparrow(r1,w0);\uparrow(r0);\downarrow(w1);\downarrow(r1,w0);\downarrow(r0)\}$$

The number of March elements is same as traditional March c- and is 6 but because of concurrency the complexity is reduced to 8n. This algorithm applied in parallel on two memory blocks. In first memory block named as M1 using first two march elements SA1 faults can be found, at the same time in second memory block using first two march elements SA0 faults can be found. At the same time as addresses are given up and down directions hence address decoder faults also can be found. Using third and fourth march elements coupling faults can be found simultaneously from the two memory blocks.

Common to all memory BIST implementations is an address generator, a test pattern generator and BIST control logic[7],[9-12]. The BIST controller can be implemented by either microcode, hardwired logic, processor-based , FPGA based or FSM based[14]. Any embedded Ram cell is addressed using column decoder and row decoder. One way to increase the speed in BIST operation is by using efficient adder circuits which are used as address generators. One method of achieving faster adders was proposed in [13]. Because adders will generate the input sequence for the decoder through which address sequence will be generated. Fig 1 shows general embedded RAM structure and Fig 2 shows proposed RAM structure which uses modified March C- algorithm. In general one row decoder and one column decoder will be used to select a memory cell. In this also one row decoder will be used to identify one address from one block of memory say M1, at the same time it identifies another memory address from memory block M2. Similarly column decoder is used to identify memory cell from block M1, and another cell from other memory block M2. Now the job of LFSR is to supply data in to these cells at a time. Using one inverter in-between data will be written in to both the cells in true and complement form as shown. Writing or reading the data in/to the two different cells at a time, hence concurrency is proved.

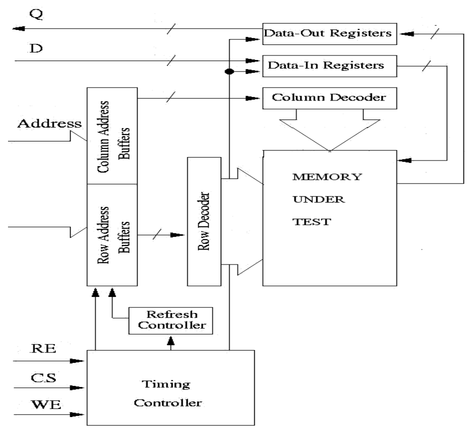

Fig 1. General Embedded Ram Structure.

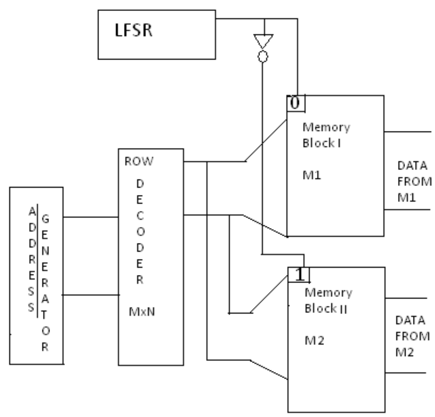

Fig 2. Proposed RAM structure to implement Modified March C-

The pseudo code for modified march c- is as follows:

```
//for writing 0s in block 1 and writing 1s in block 2, let n and m are rows and columns
  for(i=0;i<(n-1)/2;i=i+1)
     begin
          for(j=0;j<(m-1);j=j+1)
              mem[i][j]=0;//write 0 in m1
    end
 for(i=(n-1)/2;i<(n-1);i=i+1)
    begin
        for(j=0;j<(m-1);j=j+1)
           mem[i][j]=1;//write 1 in m2
   end
//for reading background and for writing      alternate
        for(i=0;i<(n-1)/2;i=i+1)
             begin
              for(j=0;j<(m-1);j=j+1)
             begin
                if(mem[i][j]==0)
               mem[i][j]=1;
```

```
            else return;
         end
      end
  for(i=(n-1)/2;i<(n-1);i=i+1)
      begin
         for(j=0;j<(m-1);j=j+1)
         begin
            if(mem[i][j]==1)
            mem[i][j]=0;
            else return;
         end
      end
```

According to Modified March C- elements, when 0s are written in one block of memory , 1s will be written in another block of memory concurrently. So the test sequence can be taken through an inverter hence true form will goes to M1 and complement form will goes to M2. Hence the test sequence generator requires additionally one inverter in order to perform test concurrently. The method directly reduces the time required to write and read the bit concurrently. This reduces the test time and test costs also. Finally, there may be additional design cost in terms of inverter only which need to generate complement test sequence to other part of the memory block.

4. RESULTS AND COMPARISIONS

Table I indicates delay performance for each element present in traditional March C- algorithm given for fault free condition and faulty condition. Under faulty condition using SA fault models the overall delay observed as 13.782ns.

Table II shows the delay performance using Modified March C- algorithm. In this also delay performance were calculated separately for fault free as well as faulty conditions. Under faulty condition the overall delay was observed as 11.784ns. Hence it is proved that using Modified March C- algorithm using concurrency the overall delay is greatly reducing. It is giving at speed test performance than any other traditional algorithm. The result tables also provide the information on minimum input arrival time before clock and maximum output time after the clock. Simulation was carried using Xilinx ise 10.1i tool for the device XC3S4004tq144 and tested on Spartan 3 kit. Fig 1 and 2 shows the simulation results respectively for modified march elements I and II when fault is imposed.

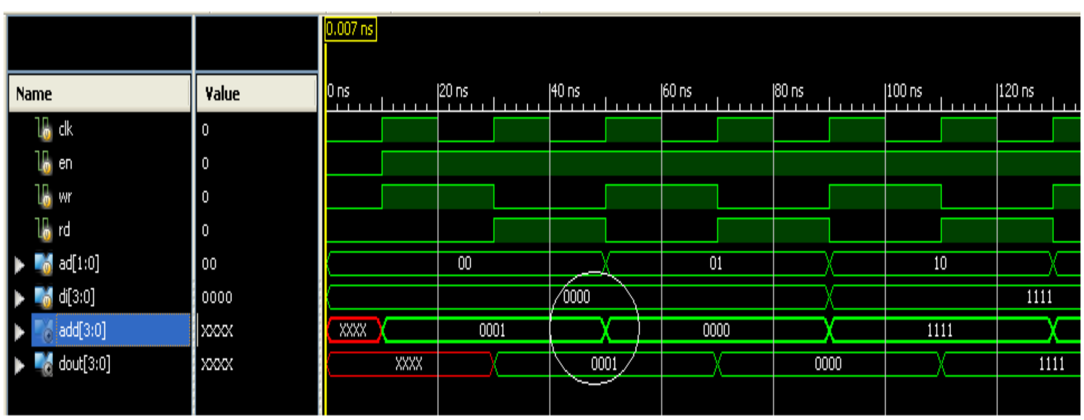

Figure 1 : simulation results for modified march C- element I {M1:↑(w0)}{M2:↑(w1)}

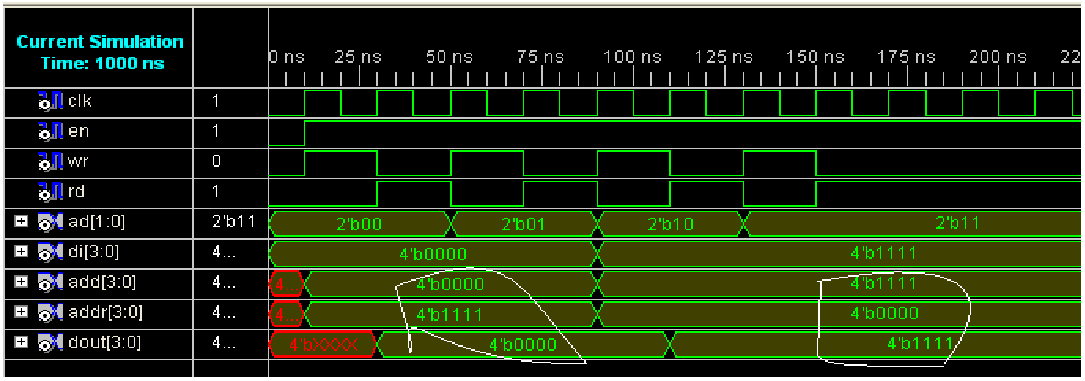

Figure 2 : simulation results for modified march C- element II {M1:↑(r0,w1)}{M2:↑(r1,w0)}

TABLE I. RESULTS FOR TRADITINAL MARCH C- ALGORITHM

MARCH ELEMENT	MINIMUM PERIOD IN NANO SEC		MINIMUM INPUT ARRIVAL TIME BEFORE CLOCK IN NANO SEC		MAXIMUM OUTPUT REQUIRED TIME AFTER CLOCK IN NANO SEC	
	WITH NO FAULT	WITH FAULT	WITH NO FAULT	WIYH FAULT	WITH NO FAULT	WIYH FAULT
M_I : ↕(W0)	1.483	2.075	3.439	4.033	6..314	6..28
M_{II}:↑(R0,W1)	1.585	2.085	3..504	3..529	6..318	6..314
M_{III}: ↑(R1,W0)	1.585	2.085	3.504	3.529	6.318	6.314
M_{IV}: ↓(R0,W1)	1.585	2.085	3.504	3.529	6.318	6.314
M_V: ↓(R1,W0)	1.585	2.085	3.504	3.529	6.318	6.314
M_{VI}: ↕(R0)	2.196	3.367	3.955	4.170	6..318	6..3

TABLE II. RESULTS FOR MODIFIED MARCH C- ALGORITHM

MARCH ELEMENT	MINIMUM PERIOD IN NANO SEC		MINIMUM INPUT ARRIVAL TIME BEFORE CLOCK IN NANO SEC		MAXIMUM OUTPUT REQUIRED TIME AFTER CLOCK IN NANO SEC	
	WITH NO FAULT	WITH FAULT	WITH NO FAULT	WIYH FAULT	WITH NO FAULT	WIYH FAULT
M1: ↑(W0) M2:↑(W1)	1.483	2.111	3.439	3.473	6..31	6..28
M1:↑(R0,W1) M2:↑(R1,W0)	2.132	2.196	4.755	3..979	6..28	6..441
M1: ↑(R1) M2:↑(R0)	2.132	1.585	3.96	3.50	6.28	6.318
M1: ↑(W1) M2:↑(W0)	1.483	2.111	3.439	3.473	6.31	6.28
M1:↓(R0,W1) M2:↓(R1,W0)	2.132	2.196	4.755	3..979	6.28	6.441
M1: ↓(R1) M2:↓(R0)	2.132	1.585	3.96	3.50	6.28	6.314

TABLE III. COMPARISIONS

TYPE OF ALGORITHM USED	COMPLEXITY	DELAY (NANO SEC)
TRADITIONAL MARCH C-	10N	13.782
MODIFIED MARCH C-	8N	11.783

5. CONCLUSIONS

This paper defines the functional fault model and compared the traditional march c- algorithm with modified march c- algorithm in terms of speed of the test sequence and complexity of the number of test sequences. The crucial part in testing is how well the test can be completed in minimum time with minimal test length. The modified march algorithm has proved that the test length is minimal as well as the time required to test SAF also minimum when compared with traditional march c-. Hence this modified march c- is much comparable and could be used for detection of various faults other than SAF as future work.

REFERENCES

[1] Shibaji Banerjee,Dipanwita Roy Chowdhury and Bhargab B.Bhattacharya, in proceedings of the 2005 IEEE International Workshop on Memory Technology, Desgin and Testing(MTDT'05).

[2] J. van de Goor. Testing Semiconductor Memories: Theory and Practice.A.J.vandeGoor,1998.

[3] J.F.Li, K.L.Cheng, C.T.Huang, and C.W.Wu, "March based RAM diagnostic algorithms for stuck-at and coupling faults",Proc,IEEE ITC pp,758-767,2001

[4] Verilog Digital System Desigh, 2nd Edition, Zainalabedin Navabi, Tata McGraw Hill, 2008

[5] 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].

[6] 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.

[7] Jing-Reng Huang ,Chi-Feng Wu, Cheng-Wen Wu, Tsin-Yuan Chang "A programmable BIST core for embedded DRAM" Design & Test of Computers, IEEE, Volume: 16 , Issue: 1 pp: 59 – 70. 1999.

[8] Tom Chen and Glen Sunada, "Design of a self Testing and self Repairing structure for highly hierachical ultra capacity memory chips", IEEE transactiopns on very large scale integration (VLSI) sustems, Vol 1, No 2, June 1993.

[9] Dr. R.K. Sharma Aditi Sood "Modeling and simulation of microcode based built in self test for multi operation memory test algorithms" . IJCSI, Internation Journal Computer Science Issues, Vol 7, Issue 3,, No 2, May 2010.

[10] Abilio parreira, J.Paulo Teixeira, Marcelino B.Santos, "Built in Self Test Quality assessmentusing hardware fault emulation in FPGAs, Computing and Informatics, Vol 23, 2004, 537-556

[11] Hiroshi Takahashi,K Wame O.Boateng, KEwal K. Sluja,Yuao Takamastu, " On diagnising Multiple Stuck at Faults Using Multiple and Single Fault simulation in Combinational Circuts", IEEE Transactions on Computer Aided Design of Integrated Circuts and Systems, April 2002.

[12] Balvinder Singh, Sukhleen Bindra Narang, and Arun Khosla "Address Conuters/Gnerators for low power Memory BIST", International Journal of Computer Science Issues Vol.8, Issue 4, No:1, July 2011.

[13] M.Parvathi, N.Vasantha, K. Satya Prasad "Design of High Speed -Low Power-High Accurate (HS-LP-HA) Adder ", ICECT, Internation conference on Electronics Computer Technology Proceedings, 2012, pp: 523-527, 978-1-4673-1850-1/12@2012, IEEE.

[14] Sonal Sharma, Vishal Moyal, " FSM based BIST Architecture" International Journal of Engineering Science and Management, 3(1), April-June 2012.

AUTHORS

Mrs.M.Parvathi has done her B.Tech in Electronics and Communication Engineering from N.I.T.Warangl,A.P, India. She ghas done her M.Tech in the specialization Digital Systems and Computer Electronics from JNTUH, A.P, India. Presently She is serving as Associate Professor in the Department of Electronics and Communication since 11 years. She is pursusing her Ph.D in the field of VLSI Design from JNTUK, Kakinada, A.P, India. Her areas of interests are Micro Electronics, VLSI Design, Digital Design, Micro Processors and Embedded Systems. She is Member of IEEE and Life Member of IETE and ISTE Technical Societies. She has published few Science and Technical papers in various International and National reputed Journals an Conferneces.

Dr..N.Vasantha Garu has done her B.E from College of Engineering, Guindy, Madras University and M.Tech in Advanced Electronics from JNTUH, Hyderabad, AP, India. She received her Doctorate in ECE in the field of VLSI Design from Osmania University. She is serving as Head of the Department of I.T and has more than twenty years of experience in teaching. She is senior member of IEEE and life member of IETE and ISTE Technical Societies. She has published more than twenty Science and Technical papers in various International and National reputed journals and conferences.

Dr.K.Sathya Prasad Garu was born in 1955 in Kakinada, Andhra Pradesh, India. Since 1980 he is in service of Teaching Profession and presently he is serving as Rector in J.N.T.U.K, Kakinada, Andhra Pradesh, India. He guided 13 PhD Scholars. He was author of text book Electronics Devices and Circuits. He is member of Technical Societies such as IEEE, and life member of IETE and ISTE and Fellow member of IE. He organized as convener over 8 conferences. He has 17 publications in various national and international journals. His areas of interest are Signals Systems, Communications, DSP.