Volume : 1 | Issue : 6 | Nov 2012 • ISSN No 2277 - 8179

Research Paper

Engineering

KEYWORDS : DEFECT DIAGNOSIS; MEMORY BIST (MBIST); MARCH; SOC

A Generic Design for Defect Diagnosis of Memories

SHAIK.SIRAJ — Guru nanak institute of technology

SK.SAIDULU — Guru nanak institute of technology

ABSTRACT

This paper mainly designed and implemented Memory BIST targeting the System-on-chip (SOC). Failure analysis and diagnosis of memory cores plays a key role in SOC product development and yield ramp-up. Diagnosis technique plays a key role for catching the design and manufacturing failures and improving the overall yield and quality. The increasing time-to-volume pressure on semiconductor products calls for new development flow that enables to reach a profitable yield level as soon as possible. MBIST is designed in such a way that it can be used for any number of memories depending upon the application. We are going for a modified MARCH algorithm, using which we can test the memory and not only locate fault cells but also identify their types.This algorithm proposes a systematic diagnosis approach based on failure patterns and functional fault models of semiconductor memories. Algorithm: Ý (w0); Ý (r0,w1,r1); Ý (w5,r5); ß (r5,wa,ra);

I. INTRODUCTION

Due to the advent of deep-submicron VLSI technology, core-based system-on-chip (SOC) design is attracting an increasing attention. On an SOC, popular reusable cores include memories (such as ROM, SRAM, DRAM and flash memory), processors (such as CPU, DSP and microcontroller), input/output circuits, etc. Defects in memory arrays are generally due to shorts and opens in memory cells, address decoder and read/write logic. These defects can be modeled as single and multi cell memory faults. The dominant use of embedded memory cores along with emerging new architectures and technologies make providing a low cost test solution for these on-chip memories a very challenging task. The addition of extra circuitry to facilitate testing of memory chips, called design for testability (DFT). "TO DFT OR NOT TO DFT ?"[1] is not a question for today's chip for the great density and complexity in today's chip, DFT is done as a must to ensure the testability for most of the chips.

Built-in Self Test (BIST) is effective in reducing test cost [2]. The amount of memory embedded in the ASICs is rapidly increasing. Modern ASICs typically contain a variety of embedded memory arrays like caches, branch prediction tables or priority queues for instruction execution [3][4].Therefore, Memory BIST is widely used in modern chip design for the following reasons: no external test equipment, Reduced development efforts, tests can run at circuit speed to yield a more realistic test time, on-chip test pattern generation to provide higher controllability and observability, On-chip response analysis; Test can be on-line or off-line, Adaptability to engineering changes, Easier burn-in support. The two types of BIST are On-line BIST and Off-line BIST. On-line BIST has tests implemented on-chip. It has shorter test time but an area overhead of one to three percent. Off-line BIST, on the other hand has tests implemented off-chip. It has longer test time but no area overhead. On-line BIST can further be classified into three subgroups: Concurrent BIST, Non-Concurrent BIST and Transparent BIST.

FAULT MODELS AND DEFINITIONS:

Faults modelled from the memory defects can be sum-marized as follows :

1. ***Stuck-at-Fault (SF***): Either a cell or a line is stuck to logical '0' or '1'.
2. ***Transition Fault (TF***): the 0→1 (or 1→0) transition is impossible on a cell or a line.
3. ***Coupling Fault (CF):*** When a cell is written to 0→1(or 1→0), the content of the other cell is changed.
4. ***Address Decoder Fault (ADF***): No cell will be accessed with a certain address or multiple cells are accessed simultaneously or a certain cell can be accessed with multiple addresses.
5. ***Retention Faults (RF***): A cell fails to retain its logic value after some time. This fault is caused by a broken pull-up resistor.

The most commonly used BIST are Logic BIST which are based on pseudo random vectors. But to achieve a 99.9% fault coverage need to access each memory cell as many times as possible. To save testing time and power, we have to choose a more efficient algorithm compared some of the most popular March algorithm on fault detect ability we proposed a modified march algorithm which will test with reduced number of tests there by reducing the test time.

II .ALGORITHM & ARCHITECTURE

This March algorithm consists of several March elements, separated by semicolon. The up-arrow stand for ascending order of address sequence, down-arrow stands for descending order. Inside the parenthesis is the specification of read writes operation and the corresponding data background.

The March based memory bist Architecture consists of 2 different blocks, i.e. BIST controller block and Memory block. This architecture uses n block of memories better understanding however a provision for further increment of memories is provided. Each of the blocks are designed separately using a HDL code and then verified individually. After the satisfactory implementation of all the blocks, these blocks are integrated to perform the memory diagnosis operation. Each individual block is discussed in depth below.

Memory Bist Operation:

MBIST starts performing Write zero, read zero, write one, and read one operations as described by the protocol Each of the set of operations in the parenthesis is performed in a state. There are 5 different kind of read and write operations required to be performed as part of the Memory BIST protocol.

State Machine consists of following states:

1. IDLE_st
2. W0_A_st
3. R0W1R1_A_st
4. W5R5_A_st
5. R5WaRa_D_st
6. ST_STATUS

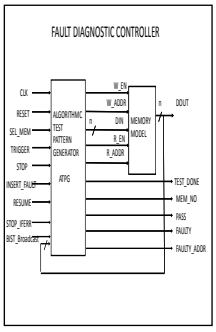

fig 1.BIST Architecture

Volume : 1 | Issue : 6 | Nov 2012 • ISSN No 2277 - 8179

After RESET Memory BIST State machine will be in IDLE_STATE State, waiting for TRIGGER to be programmed. As soon as TRIGGER is programmed, state machine jumps to W0_ A_st state and starts writing zeros from address 0 to max address in ascending order. It will be in this state till it performs the write operation to the overall memory, then it jumps to the next state R0W1R1_ A_st and starts reading zeros, writing ones, and reading ones at each address from address 0 to max. After performing read it compares against the expected values, if comparison mismatches, it jumps toST_ STATUS. While jumping to this state it also provides information like bist_comp_fail, faulty_addr, mem_no and the appropriate state information. If STOP is programmed anytime during the running of Memory BIST Operation, it jumps to ST_STATUS , asserts TST_DONE on the next clock cycle, and asserts the PASS or FAULTY along with error information . In ST_STATUS it latches the information it got from the earlier state . If STOP_IFERR was programmed the time when memory BIST was kicked off by TRIGGER, it waits in this state for RESUME or STOP. If it gets a STOP, it asserts DONE on the next clock and jumps back to IDLE_STATE and wait for TRIGGER; If it gets a RESUME in this state it clears all the error information and jumps back to the state (state_info) from where it arrived and starts performing Memory BIST operation from the next address into the memory

If STOP_IFERR is not programmed when memory BIST was kicked off, while doing memory BIST operation, whenever it gets any failure, it jumps to ST_STATUS, latches all the error information and jumps back to the state from where it arrived (state_info), irrespective of whether it gets RESUME or not. After coming back to the earlier state, state machine continues performing Memory BIST Operation to the rest of the memory and jumps to the next state, if it gets any error in the consecutive states; it jumps to ST_STATUS and performs the same operation described earlier. When the state machine is done with the last Memory BIST state (R5WaRa_D_st), it jumps to ST_STATUS and asserts DONE to indicate completion of Memory BIST Operation. Memory BIST Operation is performed onto the memory whose no is programmed in SEL_MEM. While programming TRIGGER, if bist_broadcast is programmed, state machine will trigger memory BIST for each individual memory one after the other in a sequential fashion. If INSERT_FAULT is programmed while kicking off Memory BIST, it forces the state machine to insert errors while writing data into the memory. The consecutive read operation will land into an error. This bit is used for negative testing of Memory BIST State machine.

III. IMPLEMENTATION:

The algorithm is implemented using Verilog-HDL coding in Modelsim10.1 First, the algorithm is tested in the modelsim to check the correct functionality of the proposed algorithm.

IV. SIMULATION RESULT

The simulation results for the proposed algorithm are as shown below

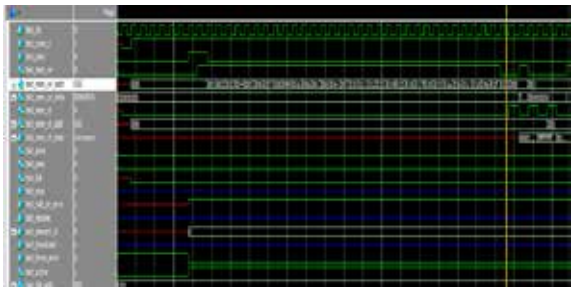

Fig2: w0 operation of algorithm.

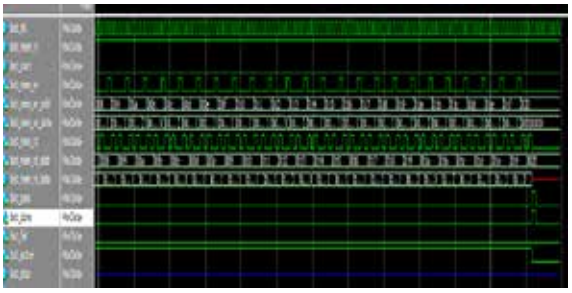

Fig3: verifying pass/fail condition

V.CONCLUSION:

I have proposed a fault-pattern oriented methodology for semiconductor

memory defect diagnostics, which greatly reduces the effort in memory testing and provides profitable yield. The proposed notion of fault pattern combines the strengths of the conventional failure-pattern approach and our previous fault-type approach for easier isolation of real defects. I have also developed a systematic procedure to explore the fault patterns, which includes a layout-based defect injection tool that provides very accurate results from realistic defect models. With the proposed approach, high-quality defect diagnostics can be automated. An industrial case has been shown to justify the work. The main contribution of the project is thus a methodology and procedure for accelerating FA and yield optimization for semiconductor memories.

REFERENCE

[1] TO DFT OR NOT TO DFT?* S. Wei, P. K. Nag, R. D. Blanton, A. Gattiker and W. MalyElectricall & Computer Engineering DeptCarnegie Mellion lJniversity, Pittsburgh, PA 152 13 | [2]DongkyuYoun,TaehyungKim,Sung juPark, "A microcode-based memory BIST implementing modified march algorithm," Asian Test Symposium, 2001.Proceedings. 10th, 2001, pp. 391-395 | [3] Memory Yield Improvement - SoC Design Perspective Jitendra B. Khare Ample Communications, Inc., Fremont, CA USA | [5] A March Test for Functional Faults in Semiconductor RandomAccess Memories D. S. SUK AND S. M. REDDY IEEE TRANSACTIONS ON COMPUTERS, VOL. C-30, NO. 12, DECEMBER 1981 | [6] Fault Pattern Oriented Defect Diagnosis for Memories Chih-Wea Wang, Kuo-Liang Cheng, Jih-Nung Lee, Yung-Fa Chou, Chih-Tsun Huang, and Cheng-Wen Wu Department of Electrical Engineering National Tsing Hua University Hsinchu, Taiwan | [7] Hans-Joachim Wunderlich, University of Stuttgart, BIST FOR SYSTEMS-ON-A-CHIP,1999. | [8] http//www.xilinx.com | [9] http//www.mentorgraphics.com