IJDACR
ISSN: 2319-4863

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary research
Website: www.ijdacr.com (Volume 2, Issue 1, July 2013)

A Comprehensive review of contemporary BIST Architecture techniques

Miss. Shradha Khemka
Shradha.khemka19@gmail.com

Mr. Lalit Bandil
lalitbandil@acropolis.in

Mr. Manish Sharma
manishsharma@acropolis.in

ABSTRACT

With increasing design complexity in modern SOC design, many memory instances with different sizes and types would be included. To test all of the memory with relatively low cost becomes an important issue. Providing user-defined pattern for screening out various manufacturing defects is also a major demand. To ease the trade-off between the hardware cost and test flexibility, Programmable Built-In Self-Test (P-MBIST) method is an opening approach to complete the memory testing under these circumstances. Many researchers have been focused on P-MBIST design. Processor-based architecture provides high test flexibility, but it increases the test development costs while applying to various processor families. To lower the design cost, a customized processor and instruction have been developed. It uses program memory to store the test program. To further reduce the hardware cost, the instruction can be serially input and saved in one internal register by adopting simple controller. With 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. Memory cores are obviously among the most universal ones-almost all system chips contain some type of embedded memory. However, to provide a low cost-cost test solution for the on-chip memory cores is not a trivial task. This paper shall serve as a knowledge base for future design in memory BIST.

Keywords

System on Chip (SOC) , BIST , Embedded BIST , Programmable BIST

1. Introduction

As current System-on-Chip (SOC) designs become memory intensive, the manufacturing yield of such devices greatly depends on the yield of embedded memories. The embedded memory takes 80%~90% of the total SoC transistors and testing embedded memory has become a significant issue in developing SOC. Currently, the most widely used method for testing embedded memory is the Built-In Self-Test(BIST) method[1][2]. The Memory BIST method performs the self-test by the built-in test circuiting inside the chip. Therefore it carries certain overheads such as larger internal area. However, the self-test are performed to each module, which simplifies the complexity of these tests. Also, due to the fact that fast tests can be carried out without the need for expensive external testing devices, the BIST method is very widely used for embedded memory test.

The memory BIST consideration is that many functional faults may exist in memories, including the address decoder fault(AF), stuck-at fault(SAF), transition fault(TF), stuck open fault(SOF), coupling fault(CF) and data retention fault(DRF) [3]. One single March algorithm can only test a subset of the memory faults. In the embedded memory, in order to obtain high fault coverage, multiple March algorithms are usually required [4]. The conventional BIST controller based on the finite state machines (FSM) [5] [6] is not flexible to integrate multiple test algorithms.

2. EMBIST(Embedded Memory BIST) Architecture

The proposed structure of the memory BIST for testing the embedded memory is consisted of the following components. There is the EMBIST controller that generates the addresses and the control signals. And there is a pattern generator for generating data patterns. Also, it is composed of a response analyzer that compares the test results. The EMBIST controller controls the memory BIST, and generates the control signals that are needed during testing memories. In addition, the EMBIST controller generates the memory addresses in the memory tests. The pattern generator generates the data patterns during memory tests. The response analyzer compares between the output data and the expected data for determining if the faults exist or not. The Figure 1 shows this embedded memory BIST (EMBIST) architecture.

EMBIST controller

EMBIST controller controls the operation of each memory BIST module during the test process. This

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary research
Website: www.ijdacr.com (Volume 2, Issue 1, July 2013)

module determines the starting and the ending positions of the test process. Also it ensures the running of test cycles by approving the signals to each module. EMBIST controller sends the BIST_control signal to the pattern generator that generates the data patterns, which correspond to each March element that forms the algorithm. Also, the EMBIST controller ensures the data pattern performs the read/write operations at the correct address of the memory

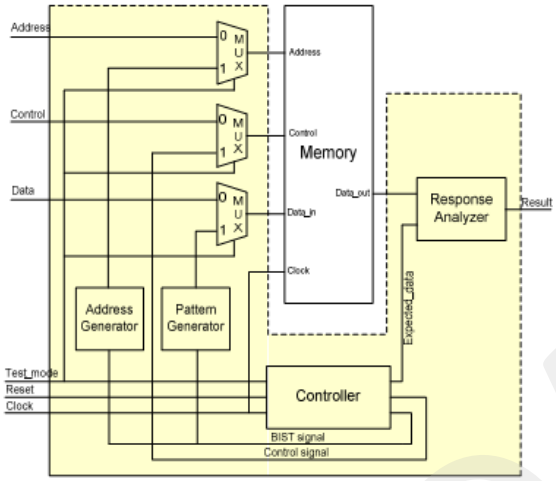

Figure 1: EMBIST Architecture

Furthermore, the Read enable control signal is approved or the response analyzer, through which faults are verified. It also generates the control signals of the memory BIST that are needed during the memory tests.

3. Microcode-based BIST

Predefined instructions or microcode is used to write the selected test algorithms. The written tests are loaded in the memory BIST controller. This microcode-based type of memory BIST allows changes in the selected test algorithm with no impact on the hardware of the controller. This flexibility, however, may come with the cost of higher logic overhead for the controller. A recent microcode-based memory BIST implementing modified march algorithm was proposed in [7]. This design requires only one third of microcode storages of others. The proposed memory BIST can be widely used for the embedded memory testing, especially under SOC design environment because of its superior flexibility and extendibility in applying different combination of memory test algorithms.

4. Processor-based BIST

An on-chip microprocessor is utilized to test the memory cores [8]. The processor-based memory BIST is done by executing an assembly-language program in the on-chip microprocessor to generate test patterns including the address sequence, data patterns, and control signals. The memory outputs are then compared with the expected correct data. The advantage of such a scheme is that it is highly flexible because various test algorithms can be realized by simply modifying the assembly programs run on the microprocessor and it is also easy to support testing of multiple memory cores. However, the drawback is a much longer test time. The BIST core is inserted between the CPU core and the on-chip bus, which also connects the memory cores. In normal operation mode, the CPU transparently accesses the system bus with slight time overhead introduced by the multiplexers. The overhead can be minimized by careful design of the multiplexers which can be integrated with the bus drivers. In memory BIST mode, the BIST circuitry takes over the control of the on-chip bus. It executes certain test algorithm programmed by the CPU and generates the addresses, input data, and control signals of the memory core. It also compares the memory output response with the expected correct data. In order to allow these two different modes, several multiplexers are used to multiplex the address bus, data input bus, data output bus, and control bus between the CPU core and the BIST circuitry. Other blocks in the BIST circuit include: (1) an address counter which generates the test address sequence; (2) a comparator which compares the memory output response with the expected correct data; and (3) a BIST controller which controls the BIST circuit.

5. Hardwired-based BIST

A hardwired-based controller is a hardware realization of a selected memory test algorithm, usually in the form of a Finite State Machine (FSM). This type of memory BIST architecture has optimum logic overhead, however, lacks the flexibility to accommodate any changes in the selected memory test algorithm. This results in re-design and re-implementation of the hardwired-based memory BIST for any minor changes in the selected memory test algorithm. Although it is the oldest memory BIST scheme amongst the three, hardwired-based BIST is still much in use and techniques have been kept developing.

Scheme	Test Time	Area OH	Routing OH	Flexibility
Hardwired	short	low	high	Zero
Microcode	average	high	low	Low
Processor	long	zero	zero	high

Table 1: Trade-offs between Different Memory BIST Schemes

6. BIST Controller

Typically the BIST controllers that are going to be generated will have test algorithms built into them. (E.g. Marching 0's, 1's, Checker - Board ...). To summarize BIST Logic is going to test your RAM. BIST Controller is internally built by Finite State Machine (FSM) which consists of March algorithm, counter, etc. It takes instructions from

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary research
Website: www.ijdacr.com (Volume 2, Issue 1, July 2013)

the IR, test the memory and detect the faults. Instructions are sent to the PMBIST using register write data, at some address using register address, by which we are selecting up count, pattern select as 01, march array as 0011, wherein, we are operating the march operation as read, write, read, which can be viewed at memory write and memory read. These pins are Memory read address and memory writer address are incremented to write the binary values and test the memory.

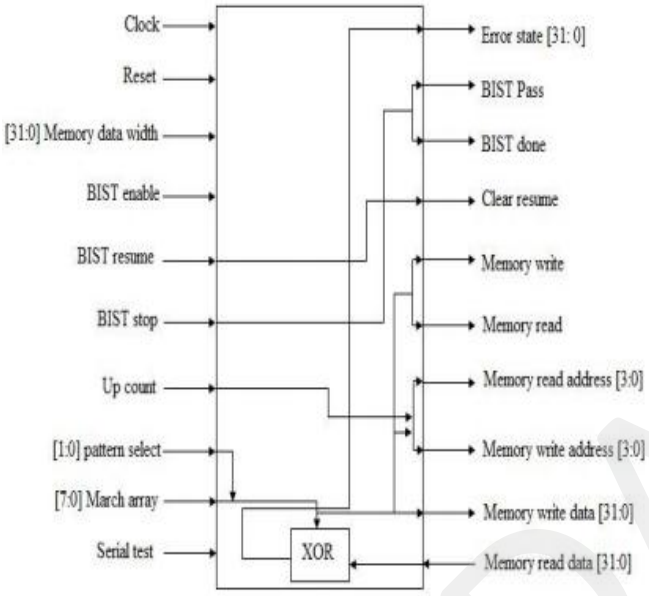

Figure. 2 Typical BIST Controller Design

7. Conclusion

With the rapid growth of SOC (System-on-Chip), the BIST is being widely used to increase the yield of the built-in memories, which is a very important component in SOC. By using the separate on-chip test logic, the memory test can be performed to reduce the test time. Also it has the advantage of being able to detect the faults at system operation speed. The proposed BIST architectures are a very efficient BIST structure that is capable of detecting all faults in the embedded memory. Different BIST architecture enables efficient tests of the high-performance embedded memories that are required essentially for the computer system, and it reduces the test cost. Then, it increases the fault coverage, and obtains high confidence. Besides, these architectures provide smallest hardware overhead by organizing the system to control plural memories with a single controller.

8. References

1. A. Benso, S. Di CarloDi, G. Natale, P. Prinetto, and M. Lobetti Bodoni, "A programmable BIST architecture for clusters of multiple-port SRAMs," Proceeding of IEEE International Test Conference, pp. 557-566, 2010

2. A. Bommireddy, J. Khare, S. Shaikh, and S. T. Su, "Test and debug of networking SoCs a case study," Proceeding of IEEE VLSI Test Symposium, pp. 121-126, 2010.

3. M. Abramovici, M. A. Breuer, and A. D. Friedman, "Digital Systems Testing and Testable D Design," Computer Science Press, New York, 2010

4. W. L. Wang, K. J. Lee, and J. F. Wang, “An On-Chip March Pattern Generator for Testing Embedded Memory Cores,” IEEE Transactions on VLSI Systems, Vol. 9, No. 5, pp. 730–735, 2011

5. S.-Y. Huang,D.-M. Kwai, and C. Huang, "A High-Speed Architecture For At-Speed DRAM Testing," Journal of Chinese Institute of Electrical Engineering, Vol. 8, No. 4, pp. 387-394, Nov. 2011.

6. C.-T. Huang, J.-R. Huang,C.-F. Wu,C.-W. Wu and T.-Y. Chang, “A programmable BIST core for embedded DRAM," IEEE Transactions on Design & Test of Computers, Vol. 16, No. 1, pp. 59-70, Jan.-Mar. 1999.

7. C. Hunter, “Integrated diagnostics for embedded memory built-in self-test on PowerPCTM,” Proceeding of IEEE International Conference on Computer Design, pp. 549-554, 2009.

8. V. K. Kim and T. Chen, “On comparing functional fault coverage and defect coverage for memory testing,” IEEE Transactons on Computer-Aided Design of Integrated Circuits, vol. 18, No. 11, pp. 1676-1683, 2009

9. S. Hamdioui, A. Al-Ars, and A. J. van de Goor, “Testing static and dynamic fault in random access memories,” Proceeding of IEEE VLSI Test Symposium, pp. 395-400, 2012