A 180-MHz 0.8-μm BiCMOS Modular Memory Family of DRAM and Multiport SRAM

Allan L. Silburt, *Member, IEEE*, Richard S. Phillips, *Member, IEEE*, G. F. Randall Gibson, *Member, IEEE*, Steven W. Wood, *Member, IEEE*, Armin G. Bluschke, *Member, IEEE*, James S. Fujimoto, *Member, IEEE*, Stephen P. Kornachuk, *Member, IEEE*, Benoit Nadeau-Dostie, *Member, IEEE*, Rajesh K. Verma, and Peter M. J. Diedrich, *Member, IEEE*,

***Abstract*—This paper describes a family of modular memories with a built-in self-test interface designed using a synchronous self-timed architecture. This approach is ideally suited to modular memories embedded within synchronous systems due to its simple boundary specification, excellent speed/power performance, and ease of modeling. The basic port design is self-contained and extensible to any number of ports sharing access to a common-core cell array. The same design has been used to implement modular one-, two-, and four-port SRAM's and a one-port DRAM based on a four-transistor (4-T) cell. The latter provides a 45% core cell density improvement over the one-port SRAM. Nominal access and cycle times of 5.5 ns for 64-kb blocks have been shown for a 0.8-μm BiCMOS process with no memory process enhancements. System operation at 100 MHz has been demonstrated on a broad-band time-switch chip containing 96 kb of two-port SRAM.**

I. Introduction

MODULAR memory is an essential component of any digital IC design library. Telecommunications circuits make extensive use of embedded RAM in a wide variety of applications including processor data and microcode stores, time-switch cores, register files, elastic stores, buffers, finite state machine registers, and caches. The ability to create any memory size and configuration provides system designers with the power and flexibility required to optimize designs for custom applications. The availability of a multiport capability adds an additional dimension of flexibility that can often simplify a system design. Systems are further simplified when embedded memories present a synchronous interface in which a minimum number of timing-critical signals are presented at its boundary.

Thus, in developing the modular RAM family for a mixed-signal 0.8-μm BiCMOS ASIC process technology [1], the following objectives were set out:

Manuscript received July 23, 1992; revised October 30, 1992.

A. L. Silburt, G. F. R. Gibson, S. W. Wood, A. G. Bluschke, J. S. Fujimoto, S. P. Kornachuk, and B. Nadeau-Dostie are with Bell-Northern Research, Ottawa, Ont., K1Y 4H7, Canada.

R. S. Phillips is with Bayview Technologies Inc., Constance Bay, Ont., K0A 2M0, Canada.

R. K. Verma was with Bell-Northern Research, Ottawa, Ont., K1Y 4H7, Canada. He is now with ATI Technologies, Scarborough, Ont., M1W 3S2, Canada.

P. M. J. Diedrich is with the Massachusetts Institute of Technology, Cambridge, MA 02139 on leave from Bell-Northern Research, Ottawa, Ont., K1Y 4H7, Canada.

IEEE Log Number 9206360.

TABLE I
RAM Family Modularity Range

Parameter	Min	Step	Max
Words	16	–	8k
I/O Bits per Word	1	1	64
Total Bits	16	–	64k
Rows	8	2	256
Columns	2	2	256
Columns per I/O Bit	2	2	32
Row Address Bits	3	1	8
Column Address Bits	1	1	5

1) Design an interface which would enable system designers to easily deploy embedded RAM's within a predominantly synchronous design environment in such a way that the system function and timing are easily predicted, implemented, and tested.
2) Achieve 100-MHz worst-case system operation (4.5 V, 105°C) for a broad-band time switch containing 96K of two-port SRAM with memory power dissipation less than 500 mW.
3) Create a reusable design with fine grain modularity up to a maximum 64K contiguous block size as specified in Table I.
4) Provide multiport functional capability (up to four ports).
5) Exploit the existing BiCMOS process technology, which has no memory enhancements in its flow, to yield a high-density memory beyond that achievable with a standard six-transistor (6-T) cell.

These objectives were achieved using a synchronous self-timed design style. The result is a four-member family of DRAM and multiport SRAM built around 6-, 8-, and 12-transistor SRAM cells (for one-, two-, four-port functions) and a 4-T one-port DRAM cell. Each member has virtually identical schematics and is capable of operating at 5.5 ns or better nominal cycle times over the entire modularity range listed in Table I.

The achievement of a single, reusable design covering this broad range of size and functionality has not been previously reported. Previous works on modular memory such as [2]–[5] have not employed internal self-timing and thus do not have the simple synchronous interface presented here. Although other fixed-block-size, self-timed RAM's have been reported

0018–9200/93$03.00 © 1993 IEEE

A 200 MHz 0.8 μm BiCMOS MODULAR MEMORY FAMILY OF DRAM AND MULTIPORT SRAM

Allan L. Silburt*, Richard S. Phillips†, G.F. Randall Gibson*, Armin G. Bluschke*, Rajesh K. Verma*, Peter M.J. Diedrich*, Stephen P. Kornachuk*, James S. Fujimoto* and Benoit Nadeau-Dostie*

* Bell-Northern Research Ltd.
P.O. Box 3511, Station C
Ottawa, ON, K1Y 4H7, Canada

† Bayview Technologies Inc.
565 Bayview Drive, Box 50
Constance Bay, ON, K0A 3M0, Canada

Abstract: A family of modular memories has been designed in a 0.8 μm BiCMOS process based on a synchronous self-timed architecture. Nominal access and cycle times are 5 ns for 64KBit blocks of 1, 2 and 4 port SRAM as well as a DRAM using a four transistor (4T) core cell.

INTRODUCTION

The goal in developing the modular RAM family for the Northern Telecom Electronics Ltd. 0.8 μm BiCMOS Telecom process (BATMOS) [1], summarized in Table 1, was to design circuitry which could be adapted with few changes to implement SRAM, multiport SRAM and 4T DRAM over the modularity ranges outlined in Table 2. The result is a four member family with virtually identical schematics, and layout optimized to achieve high density and speed. This paper will discuss the modular self-timed specification and design, its extension to multiport and dynamic RAM cells, and the measured performance.

Table 1: BATMOS Process Summary

Vertical NPN:	Emitter size (min)	0.8 × 4.0 μm
	F_t	11 GHz
NMOS and PMOS LDD:	Lmin	0.8 μm
	VTP / VTN	-0.85 / 0.80 V
	tox	17.5 nm
Interconnect:	Metal 1 line 0.8 μm, pitch 2.0 μm Metal 2 line 0.8 μm, pitch 2.0 μm Metal 3 line 1.0 μm, pitch 2.4 μm Stacked via and contact structures Local Interconnect	

Table 2: RAM Family Modularity Range

Parameter	Min	Step	Max
Words	8	1	8K
I/O Bits per Word	1	1	64
Total Bits	8	–	64K
Rows	4	2	256
Columns	2	2	256
Columns per I/O Bit	2	2	32
Row Address Bits	2	1	8
Column Address Bits	1	1	5

SELF-TIMED RAM SPECIFICATION

The specifications for the modular memory family are geared directly for ease of use by the system designer and integration within a synchronous design methodology. A read cycle timing diagram for a generic port is shown in Figure 1.

A self-timed memory architecture is used with a registered CMOS interface on all input pins. This provides the best possible integration of embedded memory into large synchronous ICs since the interface timing is similar to an edge triggered flip-flop. Worst case setup and hold times are 3 ns and 0 ns respectively. This natural interface to a synchronous system also allows the creation of simple bus cycle models for the memory which map easily into high level synthesis tools.

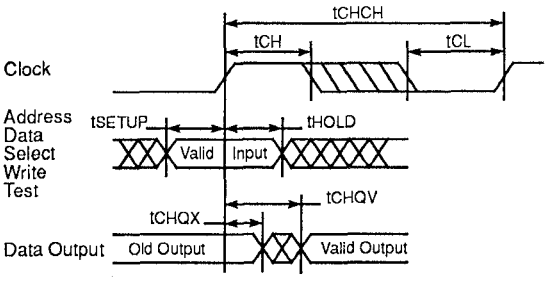


Figure 1: Read Cycle Timing Specification

The buffered outputs of all registers are available to the system designer and can be used, for example, to form counters and scan registers. They also constitute the majority of circuitry required for a Built in Self Test (BIST) circuit [2]. Characterizing the memory block and the input registers with an analog simulator as a single unit and modeling it as such at the logic level, improves the accuracy of the analysis of system timing paths through the RAM. The system designer need not rely on a logic simulator and pre-layout wire load estimates to margin the critical register-to-memory input timing.

The self-timed shut down yields a power dissipation characteristic that is inversely proportional to clock frequency and independent of duty cycle making it well suited to both high and low speed applications. In addition, the length of the active portion of the cycle decreases with faster process conditions. As a result, the power dissipation is less sensitive to process and temperature than other RAMs which depend on level sensitive clock or enable inputs that must be derived from a fixed duty cycle system clock. A synchronous memory select pin is also included to allow system level block subdecoding. Since unselected blocks draw no current, this is useful for trading off power and area at the chip level when constructing large memory functions.

Two test modes are included to provide compatibility with the SCAN and memory BIST test methodologies used at BNR. The first test mode reconfigures the memory data input / output path to a scan compatible mode in which the data inputs bypass the memory core and are transferred directly to the data output pins. This ensures controllability of the memory outputs during scan testing of the logic around the RAM. The

0-7803-0246-X/92 $3.00

A 5 Gb/s 9-Port Application Specific SRAM with Built-In Self Test

Steven W. Wood†, G.F. Randall Gibson†, Saman M. I. Adham†, Benoit Nadeau-Dostie‡

†Northern Telecom, P.O. Box 3511, Station C, Ottawa, Ontario, Canada K1Y 4H7
‡LV Software (Canada) Inc., 1525 Carling Avenue, Suite 404, Ottawa, Ontario, Canada K1Z 8R9

Abstract

This paper describes the architecture of a Time-Slot Interchange (TSI) SRAM for a SONET switching application and its associated BIST architecture. To reduce the number of data RAMs required for full switching, the memory throughput is boosted by providing multiplexed access to the core at twice the system clock rate. The nature of the memory requires a novel BIST architecture to ensure full test coverage and ensure easy access of the BIST function at different levels of system integration.

I. Introduction

This paper describes the architecture of a Time-Slot Interchange (TSI) SRAM for a SONET switching application and its associated BIST architecture. The memory port architecture of eight byte-wide, read-only ports and a single, 128-bit write-only port was designed to provide maximum data throughput with the minimum number of memories and to fit easily into the data flow at the ASIC level. To reduce the number of data RAMs required for full switching, the memory throughput is boosted by providing multiplexed access to the core at twice the system clock rate. Other mission-mode features such as self-contained, sequential write-address generation, application-specific address mapping, no-power mode, and page-synchronization controls provide system-level flexibility and simplified ASIC design. A synchronous memory interface with scan path collar and special test modes is provided to simplify scan application and design while enhancing multiport-BIST coverage. The nature of the memory requires a novel BIST architecture to ensure full test coverage.

II. 9-Port Memory Architecture

The 9-port SRAM, shown in Figure 1, is configured as 384 bytes of memory storage with a single 24×128-bit (16 byte) write port and eight 384×8-bit read ports. It is implemented as a 48 row by 64 column core cell array. The 128-bit write is row interleaved with 64-bits written to each of two selected rows per write cycle to give a more optimum array aspect ratio.

A 5-port core cell with four fully-differential read ports and a single-ended pseudo-differential write port has been designed. Differential read ports were selected for enhanced performance relative to a more compact single-ended architecture. An indirect read-access architecture is used to eliminate multiport cell stability problems associated with multiple simultaneous accesses to a cell. The single-ended write scheme combined with row interleaving saves two write bit lines per column. A local bit line inversion scheme is used to provide a pseudo-differential write capability at the core-cell for equivalent-to-differential write performance.

The 9-port operation is obtained by time multiplexing the four physical read ports in the core array. Two full read accesses to the core are performed on these ports in each clock cycle. The read data is then re-timed and latched into eight output ports for presentation to the user on the rising edge of the system clock. All read port inputs are provided to the 9-port interface at the same rising clock edge and are internally pipelined to perform the read

0-8186-7102-5/95 $04.00 © 1995 IEEE

A BIST Algorithm for Bit/Group Write Enable Faults in SRAMs

Saman Adham & Benoit Nadeau-Dostie, LogicVision, Inc.

Abstract

The use of group (or bit) write enable in memories is becoming very common in embedded memories. The circuitry used to achieve these functions need be thoroughly tested for different kind of defects using specific test sequence. However, most BIST algorithms assume that these write enables are forced active during the global write cycle in the BIST run. This paper presents a serial interface BIST algorithm that is used to test defect on bit/group write enables of these memories.

.

1. Introduction

Static Random Access Memories (SRAM) is being extensively used in system on chip (SOC) designs. It is common to see tens or even hundreds of SRAMs integrated into chips to perform different functions [1]. With the wide used of SRAMs new functionalities are being added to the memories to address specific design requirements. Among those is the ability to perform partial write operation where only specific portion of the memory word is written. This is achieved by disabling the memory internal write circuitry from writing the whole word by using write enable control ports. These write enable ports may control individual bits or a group of bits in a word. However, using the bit or group write enables does not necessarily replace the need for global write enable. Most memories maintain the global write enable port to initiate a write operation to the memory.

The existence of bit and group write enable functionality in SRAMs present new testing challenges. Treating these ports as global write enable surely reduces the fault coverage on the circuitry associated with them.

This paper presents a BIST algorithm and test circuit architecture for detecting defects in bit and group write enable that are internal to the memory. In section 2 we present the defects of interest and the fault models used. The algorithm for detecting bit write enable defects is presented next. Section 4 extends the algorithm for the detection of the group write enable faults. The memory interface circuit architecture is introduced in section 5. Conclusions are provided in section 6.

2. Bit and Group Write Enable Defects and Fault Models

Figure 1 shows a typical memory implementation to support bit write enable function. The active high write enable logic is implemented at the input data path (before the bit line decode circuit).

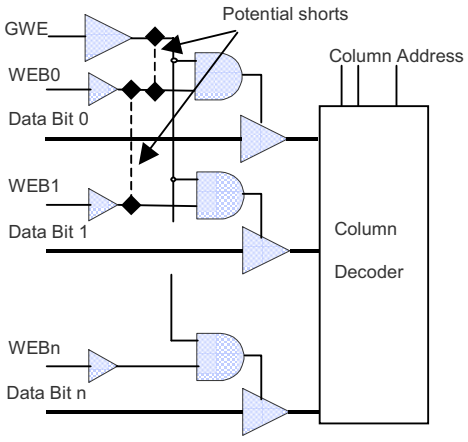


Fig. 1 A Bit write Enable implementation

Here the global write enable (GWE) is used to control all bit write enables (WEB). A data bit is written to the memory only when the global write enable and the specific bit write enable are active.

Controlling the WEBs and the GWE from one source in test mode will mask stuck active faults on the WEBs. Shorts between the WEB will also be masked. Furthermore, shorts between GWE and any WEB will not be detected, Wired-OR (for active low write enable signals); Wired-AND (for active high write enable signals) short between the Global Write Enable (GWE) and the bit write enable of the Most Significant Bit (MSB) will not be tested.

The coverage is further reduced if the write enable circuitry is implemented after the column decoder. The defects in this case will affect all bit lines in the memory array.

Traditionally test engineers use functionally generated patterns to test for these defects. The patterns perform read and write operations with different data and bit write enable values. However, generating these patterns is becoming too complicated for deeply embedded memories.

3. Detection Algorithm for Bit Write Enable

To detect the defects of interest, GWE and WEBs are controlled independently. The detection of these defects is achieved using a serial interfacing technique [2]. The memory data inputs and outputs are connected to form a shift path from one bit to the next. The first bit receives a single bit of data that will be used to apply the test data. Data is shifted from the LSB to the MSB.

Records of the 2004 International Workshop on Memory Technology, Design and Testing (MTDT'04)
1087-4852/04 $20.00 © 2004 **IEEE**

Adapting an Industrial Memory BIST solution for testing CAMs

Jais Abraham, Uttam Garg
Intel, Bangalore, India

Glenn Colon-Bonet,
Intel, Fort Collins, USA

Ramesh Sharma, Chennian Di
Intel, Santa Clara, USA

{jais.abraham, uttam.k.garg, glenn.t.colon-bonet, ramesh.sharma, chennain.di}@intel.com

Benoit Nadeau-Dostie, Etienne Racine
Mentor Graphics, Canada

Martin Keim
Mentor Graphics, USA

{benoit_nadeau-dostie, etienne_racine, martin_keim}@mentor.com

Abstract

Content Addressable Memories (CAMs) have found widespread use in applications that require high speed search capabilities. Each cell in the CAM array is associated with a storage unit and a comparator logic. Due to the various customized features in the CAM implementations, creation of an automated BIST solution for testing them has presented unique challenges. This paper shows that, with suitable modifications to the CAM test collar, an existing BIST solution and flow traditionally used to test embedded SRAMs can be used to test the CAMs.

Keywords: Binary CAM, Ternary CAM, BIST

1. Introduction

Content Addressable Memories (CAMs) have been employed widely for high speed search operations in applications such as networking and cache look up [1][2][3][4][5][6]. Compared to RAMs, which return the data based on the address provided to it, the CAMs compare the incoming data with the keys present in the array and return the address(es) of the matching entries in the array. A "hit" or "match" is said to occur when the incoming compare data can be found in the CAM array. Thus, the CAM cells have a comparator logic in addition to the storage element. CAMs are usually followed by a priority encoder and a payload RAM. The priority encoder resolves multiple matching entries in the array to the highest (or lowest) matching address and in many high speed implementations, the match lines serve as fully decoded address to the payload RAM.

CAMs are usually classified as Binary CAMs (BCAM) and Ternary CAMs (TCAM). Quaternary CAMs have recently been reported [7]. BCAMs generate a hit if the complete entry is matched. On the other hand, the Ternary CAMs (TCAM) provide the capability to match a part of the entry, while treating the remaining part as don't cares, which are specified using additional mask bits. TCAMs need an extra storage array to store the mask bits. In addition, different TCAM implementations provide the capability to mask specific bits of an entry or parts of the entries or the complete entry itself.

Several CAM BIST solutions have been presented in the past. Representative examples can be found in [8-11]. In our experience, CAM designs often include new features or features that are implemented differently in a way that would affect the BIST implementation and its automation. The contribution of this work is to demonstrate how an existing memory BIST solution for embedded SRAMs can be used to test CAMs in that context. Although the solution in this paper describes a particular CAM design, it will become clear that the solution can easily be extended to other implementations as well. The paper explains two exemplary CAM algorithms, providing insight into how these algorithms differ from those for SRAMs. In practice, variations of these algorithms should be considered, based on the presence or absence of specific CAM features. However, the treatment on the algorithms is not exhaustive and other works on CAM test [8-11] can be referred for this purpose. The only requirements for the existing BIST tool is to have a powerful algorithm programming capability used in combination with flexible operation definition of user-defined control signals.

The paper is organized as follows. Section 2 discusses the architecture of CAMs and provides a description of the generic ports that are available on the CAMs. Section 3 provides an overview of the fault models considered for testing the CAM comparator logic and the algorithms used to test them. Section 4 discusses the details of an example custom CAM collar. Section 5 discusses the differences in the design flow compared to the one used for RAM BIST. Section 6 concludes the paper.

2. CAM Architecture

The logical view of a Binary CAM array is shown in Fig. 1. The Binary CAM can be conceptualized as having two functions: storage and compare. The storage function behaves like a RAM and needs to be tested for the faults in the address decoder, storage cells and the read/ write circuitry. Associated with every bit is also a comparison logic. The result of the comparison is calculated across the complete entry to generate a match/hit.

A High Speed Embedded Cache Design with Non-Intrusive BIST

Steve Kornachuk†, Larry McNaughton††, Robert Gibbins†, Benoit Nadeau-Dostie††

†Northern Telecom, ††Bell-Northern Research
P.O. Box 3511, Station C, Ottawa, ON, Canada, K1Y 4H7

Abstract

This paper describes a 155 MHz wide-word cache design and its test integration features. Design techniques for high speed CAM with single ended match line sensing and highly integrated RAM are described. A new cache BIST algorithm based on the SMARCH [1] algorithm is presented. New techniques are described for the insertion of cache BIST access points into a high speed data path without compromising mission mode performance. Performance results of cache memory used for telecommunications microprocessor applications with 1Kb of CAM referencing a 5 Kb RAM are presented.

I. Introduction

With device gate counts and pin counts increasing and the cost of external functional testing rising, increases in hardware speed and functional complexity must be complemented by robust and complete self-test capability. The requirement to integrate special function macros into high speed data paths combined with the need to insert BIST for these components presents a potential conflict between performance and testability. One such macro, the cache design described here, provides high speed prefetched instruction storage and retrieval for a telecommunications processing application.

From its inception, the memory and BIST circuits were developed with the following objectives:

1) Accommodate the system cycle rate of 60 Mhz and various I/O overheads by completing the system address look-up compare cycle and the prefetched instruction RAM read within 12 ns, worst case (4.5V, 105°C).

2) Minimize active cycle power.

3) Incorporate into the cache's CAM and RAM blocks an interface that facilitates the connection of a cache BIST and is easily testable with functional vectors.

4) Design a BIST that has maximum memory fault coverage for both CAM and RAM blocks of the cache.

5) Insert the BIST circuitry into the high speed memory data path without incurring additional mission mode delay.

The memory performance and power objectives have been achieved by using a new self-timed CAM internal timing loop integrated tightly with the accompanying RAM read circuitry control signals. The self-timed nature of the cache give reasonable and predictable power consumption which increases linearly with frequency of operation. BiCMOS driver circuits used in the memories allow easy expansion from 4 to 256 rows and from 4 to 128 CAM compare data bits with no architectural change.

To achieve an easily testable cache interface, match line outputs are provided to the user to assist with on-chip diagnostics and cache BIST testing.

Cache BIST coverage is maximized by executing a complete SMARCH on the CAM read/write port while simultaneously running compare cycles on the compare port. Priority encoding of match line outputs ensures the correct hit/miss sequence. The CAM SMARCH sequence ends with a preload of the CAM with known states to

0-8186-6245-X/94 $04.00 © 1994 IEEE

An Embedded Technique For At-Speed Interconnect Testing

Benoit Nadeau-Dostie, Jean-Francois Cote, Harry Hulvershorn and Stephen Pateras

LogicVision Canada, Inc.
Ottawa, Ontario

LogicVision, Inc.
San Jose, CA

Abstract

A new embedded test technique which provides full at-speed testing of board level interconnect is described. The proposed technique is fully compatible with the IEEE 1149.1 boundary scan standard. The technique extends the standard's architecture to provide for synchronized at-speed timing control of the boundary scan cells so that test data can be applied and captured across the interconnect at system speeds.

1. Introduction

Rapid advances in silicon, packaging and board technologies have created new opportunities and challenges for equipment manufacturers. Indeed, the complexity and speed of today's boards are creating severe manufacturing test challenges. This is particularly true when it comes to testing for delay related defects in the board interconnect.

Many companies today perform interconnect testing using in-circuit test (ICT) techniques [10]. Testers that provide this capability typically contain hundreds to thousands of physical probes (often called a "bed-of-nails") that are used to contact chip pins and board test points to perform the interconnect tests. However, the continued practicality of ICT is severely challenged by advances in packaging technology. Dense, double-sided boards require thousands of probes, and modern surface-mount chip packages leave most chip pins physically inaccessible. In addition, testing for interconnect delay faults is not practical due to the inability of the tester hardware to apply at-speed signals through the probes.

ICT's physical access problem drove the creation and subsequent industry-wide acceptance of the IEEE 1149.1 Boundary Scan Standard [1-9] as a foundation for probeless interconnect testing. Unfortunately, the IEEE 1149.1 standard is not designed for applying and capturing data across interconnects at application speed. Because of the standard's serial control approach, a minimum of 2.5 test clock (TCK) cycles are required between the time test data can be launched from one chip and captured at another. So for example, with a TCK operating frequency of 25 MHz, the effective interconnect test frequency available from 1149.1 becomes 10 MHz, clearly much lower that typical board application speeds.

In light of the above limitations, the method most commonly used to achieve at-speed interconnect testing is through functional testing at the board or system level. Functional testing though has some strong limitations. The time and resource required to develop the functional tests can be very high. Perhaps even worse is the time needed to diagnose the physical location of a fault once it is detected.

In this paper, a new embedded test technique which provides full at-speed testing of board level interconnect is described. The proposed technique builds upon the standard IEEE 1149.1 test approach. The technique uses new at-speed boundary scan cells as well as embedded timing control for these cells. A key feature of the proposed technique is that it is compatible to IEEE 1149.1 based testing. A board design can incorporate a mixture of chips incorporating the technique described in this paper and chips simply compliant to the IEEE 1149.1 standard. Any existing boundary scan test hardware as well as test and diagnostic software tools can still be used without modification.

The remainder of this paper is organized as follows. Section 2 describes the high level architectures at both the chip and board levels required to implement the proposed technique. Section 3 describes the new at-speed boundary scan cells while the at-speed timing controller is described in section 4. Section 5 describes the use model of the proposed technique in the context of boundary scan based testing. Section 6 concludes the paper.

2. High Level Architectures

The diagram in Figure 1 illustrates the chip level architecture required to implement the proposed technique. Shown are the standard IEEE 1149.1 TAP and controller together with at-speed boundary scan cells and an at-speed interconnect (ASI) controller. The at-speed boundary scan cells consist of IEEE

0-7803-5753-1/99 $10.00

A new hardware fault insertion scheme for system diagnostics verification

Benoit Nadeau-Dostie[‡], Harry Hulvershorn[†], Saman M. I. Adham[†]
[†]Northern Telecom, P.O. Box 3511, Station C, Ottawa, Ontario, Canada K1Y 4H7
[‡]LV Software (Canada) Inc., 1525 Carling Avenue, Suite 404, Ottawa, Ontario, Canada K1Z 8R9

Abstract

A new fault insertion method to help debug diagnostic software of telecommunications systems is described. The method makes use of Boundary Scan to inject multiple and un-correlated faults in a telecom system carrying traffic. Both hardware and software implementation aspects are discussed. The new method allows the use of structural test as part of diagnostics software to locate faults.

I. Introduction

Fault insertion (also known as fault injection) has been used for many years to evaluate the effectiveness of diagnostic software. Even though it is possible to perform this evaluation using simulation, it is usually difficult because of the absence of proper tools and models. A common way of performing this verification is still to inject faults around components of a board using switches or other similar hardware. Usually only a few carefully chosen faults could be inserted using this primitive mechanism. For each revision of a board, a few boards are selected and modified to add the fault injection means. This method is becoming less practical because it is becoming necessary to inject more faults in more complex systems and, at the same time, the physical access to components and tracks of the board is reduced. The process has several drawbacks. It is expensive since the modifications on the selected boards are done after the board is designed. Faults are injected into limited number of tracks on the board which in turn results in low fault coverage.

This is why alternate methods are needed in modern telecommunications systems. Recently, several authors demonstrated how Boundary Scan (Bscan) [4] could be used to insert faults in complex systems containing Application Specific Integrated Circuits (ASICs) [1-3]. We first review those schemes. We then describe a significant improvement over the method introduced by Wilcox et al [3]. This improvement is essential to the verification of diagnostic software making use of structural tests (including all the ones involving Bscan) to locate faults. The new scheme retains all the other benefits of Wilcox's. Multiple un-correlated faults can be injected at the same time while the system is running. No extra delay is inserted in the Bscan cells to inject the faults. Finally, existing Boundary scan cell layouts can be reused to implement a complete fault insertion cell. Those benefits are traded off against area.

A few applications of this new fault insertion method are demonstrated. Several aspects of the system software required to generate and apply faults is also described. Limitations of the scheme are discussed before we conclude.

II. Review of previous schemes

Sedmak [2] suggests three ways to inject faults at the output of ASICs. For example, all outputs of a given ASIC can be faulted using standard Bscan instructions (HIGHZ, CLAMP or EXTEST). Another way is to load

INTERNATIONAL TEST CONFERENCE

0-7803-2991-0/95 $4.00 © 1995 IEEE

A New Procedure for Weighted Random Built–In Self–Test

Fidel Muradali, Vinod K. Agarwal and Benoît Nadeau-Dostie*

VLSI Design Laboratory, McGill University, Montréal, Québec, Canada
* Bell–Northern Reserach (Corkstown), Nepean, Ontario, C anada

Phone: (514) 398-7136
Fax: (514) 398-4470
e-mail: agarwal@pike.ee.mcgill.ca

Abstract

A procedure for determining weight distributions and implementing weighted random pattern generation in built–in self–test circuits is examined. Test lengths orders of magnitude shorter than those encountered in conventional pseudorandom testing are recorded.

1. Introduction

In an attempt to address the unreasonable test times encountered using conventional pseudorandom testing, this paper describes a procedure for incorporating weighted random pattern generation into self-testing circuits.

At the root of most common test pattern application mechanisms used today is stored pattern testing, pseudorandom pattern generation and the inevitable hybrid of both. Stored pattern testing involves the application of specific test vectors, each of which provides an incremental level of coverage. Externally applied, this approach may require large storage capability and expensive test units. The technique is straightforward and suitable for many *present* testing needs. However, considering that the specifications of a test unit are static, upper limits are physically imposed on such variables as test frequency, the number of I/O channels and the size of input (pin) buffers. On the other hand, circuit attributes are dynamic, that is various technology advancements can result in higher operating speeds, higher pin counts, and progressively larger test sets. This conflict ultimately implies costly test equipment upgrades or replacement. Therefore, it is questionable if long term economical testing is feasible with a standard stored pattern approach [Bas89].

Uniform random pattern or pseudorandom testing relaxes the functional and memory requirements of the external tester used. Often, relatively simple sequential circuits, such as linear feedback shift registers (LFSRs) [Gol67][Bar87] or cellular automata (CA) [Hor89], can be used to pseudorandomly generate test inputs such that there is an equal probability of assigning a 1 or 0 value to an input. The intention is that after a large number of these "pseudorandom"[1] vectors, a reliable level of fault coverage will be achieved. Unfortunately, experience has shown that with current circuit densities, an excessive number of test patterns may be needed to attain this goal. Moreover, only a relatively small number of the initial vectors cover a large portion of the detectable single stuck-at faults, leaving the majority of the sequence to be wasted in an attempt to detect random pattern resistant faults. In fact, most of the time penalty associated with random pattern testing is due to the application of these unnecessary patterns.

An obvious alternative is to use a joint test strategy – apply a reasonable number of pseudorandom test patterns and supplement this with stored pattern testing. However, it has been found that in many cases, the size of the stored test set needed nears 70% of the full deterministic set [Bas89]. Thus, this approach does not quite address the problem of limited storage in external testers.

The idea of incorporating test circuits on-chip, i.e. that of built–in self–test (BIST), offers numerous advantages, such as increased test portability, more efficient low level (probe) tests and easier diagnosis of failed chips at the board level. Of course, there are shortcomings which must be considered, for example an increase in area, and potential speed degradation if test circuitry is inserted in a critical path.

A refinement of conventional pseudorandom testing is the application of a non-uniform distribution of 1's to 0's as test inputs. This idea of *biased* or *weighted* random pattern (WRP) generation can result in a much higher rate of coverage than that of uniform random pattern testing alone. One drawback though is that proposed schemes to determine weights are quite complicated and the associated on-chip weighted pattern generators have, so far, been much more area "hungry" than their uniform random counterparts [Wun87][Sch75][Brg89].

This paper examines a built–in self–test (BIST) imple-

[1] In the course of this text, "pseudorandom" or "random" patterns refers to those which are pseudorandomly generated with a device such as an LFSR [Bar87].

1990 International Test Conference CH2910-6/0000/0660$01.00 © 1990 IEEE

A Scan-Based BIST Technique Using Pair-Wise Compare of Identical Components

Benoit Nadeau-Dostie *, Philip S. Wilcox *, Vinod K. Agarwal **

* Bell-Northern Research Ltd.,
P.O. Box 3511 Station C
Ottawa, Ontario, Canada K1Y 4H7

** Department of Electrical Engineering
McGill University, 3480 University Street
Montréal, Québec, Canada H3A 2A7

Abstract

This paper addresses the problem of efficiently testing scannable ASICs in a board-level and system-level environment. The method makes use of a serial testability bus (ETM or IEEE 1149.1) and takes advantage of the presence of identical components on the boards. The main benefits of the method are a significant reduction in test time and test data to be stored. Results obtained for an actual system show a reduction in test time of about 20 times for a module with 50 ASICs. The extra board area required was less than 2% for all boards of the module.

Introduction

This paper addresses the problem of efficiently testing scannable Application Specific Integrated Circuits (ASICs) in a board-level and system-level environment. A new Scan-based BIST technique is described and the results of its application to an actual system are presented. The method makes use of a serial testability bus (IEEE 1149.1 or ETM) and takes advantage of the presence of identical components on the boards.

The paper is organized as follows. The first section describes the system used as the test vehicle for the new testing method and its main test features. The second section presents a possible test procedure and highlights the issues associated with conventional scan testing at the board and system level. The third section describes the key components of the proposed test and diagnosis procedure (Pair-Wise Compare or PWC method) which is detailed in section 4. The fifth section discusses the costs and benefits of the procedure in general and for the particular system described in the first section. Conclusions are drawn from that experience in the last section.

1) System and test features

The system module to be discussed is composed of 4, or optionally 5, Printed Circuit Boards (PCBs), backplane and a processor card (see figure 1). A complete system consists of dozens of such system modules. The Port boards perform the Input/Output operations of the system. The module contains of order 50 ASICs but with less than 10 unique types. On figure 1, each fill pattern can be associated to an ASIC type (the number of instances is not necessarily exact). The main boards have 15 to 20 ASICs each.

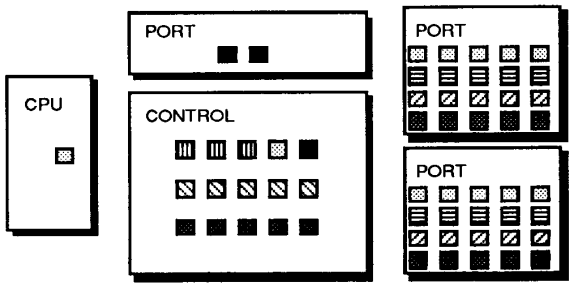


Figure 1 Schematic view of the module showing the ASIC distribution

The ASICs are scan-testable designs, with boundary scan as an integral component. Memories are tested using specialized self-test methods described in [1]. A standard serial slave interface on each chip is used for test access and control. The five-pin testability bus is similar to the Element Test and Maintenance

A Serial Interfacing Technique for Built-In and External Testing of Embedded Memories

Benoit Nadeau-Dostie*, Allan Silburt*, Vinod K. Agarwal**

*Bell-Northern Research Ltd., 185 Corkstown Road, Nepean, Ontario, Canada, K2H8V4
** McGill University, 3480 University Street, Montréal, Canada, H3A 2A7

ABSTRACT

This paper describes the design and implementation of a test access method and built-in self-test scheme for embedded static RAMs based on a new serial interfacing technique.

INTRODUCTION

Various BIST algorithms and techniques have been proposed [1-6,9,11] to test RAMs. All of the techniques cited evaluate all the bits of each memory word in parallel as it is read. A comparator or compressor module is then usually used for each bit of the output data path. This may be costly in terms of area for wide data paths. Some of the BIST schemes [1,4-6,9] require modifications of the RAM circuitry which may not be possible or practical in many ASIC design environments. When signature compression of the data is used [4,6,9], it adds the problem of aliasing uncertainties to the calculation of fault coverage [8]. There is also a significant problem in applying all schemes cited above when multiple embedded RAMs of varying sizes and port configurations are used on one chip. If each memory requires its own dedicated BIST circuit, the chip area devoted to testing can be unacceptably high. Therefore, it is advantageous to share the BIST circuit among several RAM blocks. It is also desirable to re-use as much of the test circuitry as possible in the normal operation of the chip. It can be quite difficult or inefficient to achieve this sharing of circuitry with the existing schemes.

This paper describes a serial interfacing technique for embedded RAMs which has been successfully applied to static single port and dual port memories in custom integrated circuits. A single bit of the input data path of a RAM (or a group of RAMs) is controlled by the BIST circuit and a single bit of the output data path is observed during the execution of the algorithms. The other bits are controlled and observed indirectly through the serial data path.

Automatically generated BIST circuits have been developed around this serial technique which embed an algorithm suited for the application with the RAM. The serial data path interface has also been used to provide external access to memories on cost sensitive chips which could not justify the full BIST overhead. This provides a simple external test access mode which uses a minimal number of pins yet exercises the memory at full speed.

SERIAL ACCESS TECHNIQUE

Figure 1 shows a block diagram of a static RAM memory (inside box with dotted outline) and the additional external connections required to implement the shifting capability. The address bus is not latched and applied directly to the X (row) and Y (column) decoders. The input and output data paths are separate. When the READ strobe is high, a word is read and transfered to the output of the memory. When the strobe goes back low, the data is maintained by the transparent latches located at the output of the sense amplifier until the next read operation. When the WRITE strobe is high, a word is written into memory at the location determined by the address.

The shifting capability is provided by the multiplexers connected along the I/O data path. The multiplexers select either the normal inputs or the test inputs depending on the value of the Testmode signal. The test input applied to input i of the memory is simply output i-1 or a signal controled directly by the BIST circuit in the case of the least significant bit. Suppose that M_i and M_{i+1} are two logically consecutive bits of the same word. They may or may not be physically adjacent in memory depending on the layout. In order to move the contents of M_i to M_{i+1}, a read operation of the word containing these two bits is first performed bringing the bit in M_i to the corresponding output latch. A write operation is then performed at the same location to store the bit in M_{i+1}. Actually, the full word gets shifted by one position. The memory cell storing

IEEE 1989 CUSTOM INTEGRATED CIRCUITS CONFERENCE CH2671-6/89/0000-0136 $1.00 © 1989 IEEE

BIST of PCB Interconnects Using Boundary-Scan Architecture

Abu S. M. Hassan, *Member, IEEE,* Vinod K. Agarwal, *Fellow, IEEE,* Benoit Nadeau-Dostie, *Member, IEEE,* and Janusz Rajski, *Member, IEEE*

***Abstract*—This paper addresses the issues of printed circuit board (PCB) interconnect testing in the context of boundary-scan architecture. Boundary-scan architecture is treated here as the framework for a PCB level built-in self-test (BIST). A novel BIST method is developed which utilizes various features of the architecture. Boundary-scan architecture is shown to have the capability to generate time efficient test vector sets. Response compaction within the boundary-scan chain is introduced to reduce shift out time as well as to simplify detection and diagnosis. However, the proposed BIST schemes require some extensions of the standard boundary-scan cells, and schemes can work only if every boundary-scan cell of every IC on the PCB has the proposed extensions.**

I. Introduction

IN RECENT years, structured design-for-testability at the printed circuit board (PCB) level has become an activity of major interest. This is a natural evolution, following a wide acceptance of the structured DFT (i.e., scan and built-in self-test) [21] at the IC level and the realization that the cost associated with implementing scan cannot be justified unless it can be used to simplify the testing efforts at the PCB and higher levels as well. This, combined with the emergence of very high density packaging technology at the PCB level, in particular, that of surface mount interconnects, and the growing interest in multichip modules, made it essential to develop the concept of boundary-scan, as detailed in [15].

The boundary-scan concept allows one to access and control the primary input and output pins on each component of the PCB from its edges. This is done by connecting all the primary inputs and outputs of each component into a shift register which has a boundary-scan input and a boundary-scan output. A simple boundary-scan cell is shown in Fig. 1 [15, fig. 1-1]. The shift registers on all the components of a PCB can be connected together to form a larger shift register with a single scan in edge and a single scan out edge, as shown in Fig. 2 [15, fig. 1-2]. A test clock line and a test mode select line, along with the boundary scan in and scan out lines, form a four-wire testability bus [15]. Thus, in effect, the boundary scan concept provides a type of electronic in-circuit testing facility.

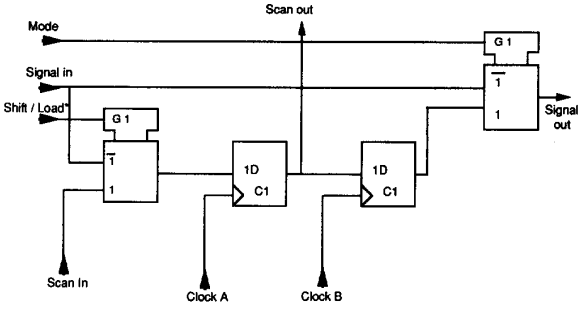


Fig. 1. A Simple boundary-scan cell.

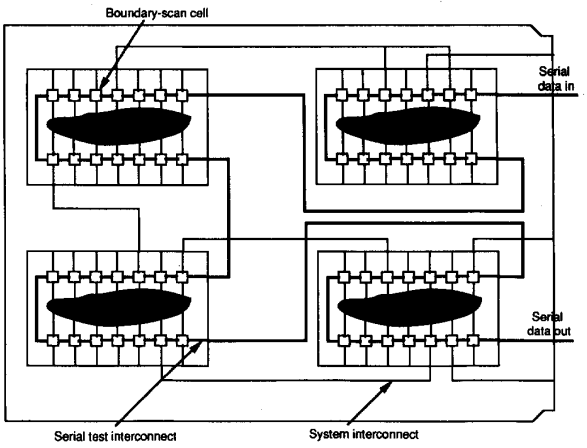


Fig. 2. Boundary-scan chain.

Using this concept at the PCB level, it should be possible to confirm that each component, such as an IC, performs its required function, and that the IC's are interconnected in the correct manner. The problem of interest in this work is that of using this concept to verify that the interconnects connecting these IC's on a PCB are free from structural faults.

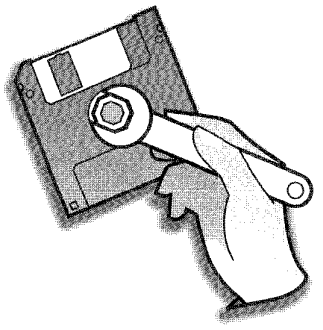
The interconnection of IC's and other discrete components on a PCB is a complex maze of multilayer electrical

Manuscript received November 27, 1990; revised November 6, 1991. This paper was recommended by Associate Editor S. Seth.

A. S. M. Hassan and B. Nadeau-Dostie are with Bell Northern Research Department 5E31, Mail Stop 049, Corkstown, Ottawa, Ont. K1Y 4H7, Canada.

V. K. Agarwal and J. Rajski are with the Microelectronics and Computer Systems Laboratory, Department of Electrical Engineering, McGill University, Montréal, P.Q. H3A 2A7, Canada.

IEEE Log Number 9200812.



Built-In Self-Test: Assuring System Integrity

Bernd Könemann
Ben Bennetts
Najmi Jarwala
Benoit Nadeau-Dostie
LogicVision

Today's complex electronic products are harder to test using traditional external methods. BIST can frequently be used without significantly increasing a product's size, cost, and production time.

Microelectronics products and their components are tested many times during their life cycles. Integrated circuits begin their life on a wafer and are tested even before the wafer is diced into chips. The chips subsequently are packaged and used in higher level electronics. Each manufacturing step can introduce new defects that require new testing.

Each stage of the assembly process poses a different challenge for accessing, testing, and verifying chips and other product elements.

Accessing deeply embedded complex functions in higher level products to perform system-level tests can be very difficult. System-level integration turns each chip into a hierarchical assembly containing an ever-growing number of embedded core functions, such as memories, processors, and multimedia.

Determining the effects of the manufacturing process and environmental conditions on a product's ac and dc characteristics, debugging, and verification present additional problems. After all, it is not easy to probe submicron signal lines on buried wiring layers in the middle of a chip with a logic analyzer or an oscilloscope.

Also, limitations on bandwidth and on the performance of both the test equipment and the interface between the equipment and the product being tested make access difficult. Low-cost interfaces, like the ones used for system-level test access, tend to be serial and thus too slow for large amounts of test data. On the other hand, it is very difficult for automatic test equipment to access chips on a wafer through complicated high-performance interfaces.

Moreover, testing for calibration, stability, and noise is becoming more difficult to tackle as chip technologies begin to operate in the gigahertz operating frequency range. It is difficult to maintain clean waveforms across the interface between the tester and chips operating at such high frequencies.

Finally, regenerating and managing huge amounts of chip test data for each new assembly level, hardware revision, and system configuration quickly become a major burden on engineering resources and product development time.

In many cases, built-in self-test (BIST) alleviates these problems. And the addition of BIST features to electronics hardware frequently does not significantly increase a product's size, cost, and production time, as was the case in the past. This makes BIST practical in many cases.

ADVANTAGES OF BIST

In the 1980s, a number of system houses developed BIST solutions for distributing and embedding critical test functions into product components, in part because of data volume and access issues. The system houses did this because BIST avoids the complexities of external testing and access mechanisms by moving critical test and measurement functions inside products.

Embedded test and measurement utilities match the chip's technology capabilities and open up new possibilities for high-speed access to internal functions. In addition, BIST reduces the need for complex and expen-

0018-9162/96/$5.00 © 1996 IEEE

2024 29th IEEE European Test Symposium (ETS)
- Industry Paper -

Combining Built-In Redundancy Analysis with ECC for Memory Testing

Luc Romain
Siemens Digital Industries Software
Ottawa, Canada
luc.romain@siemens.com

Paul-Patrick Nordmann
Siemens Digital Industries Software
Hamburg, Germany
paul-patrick.nordmann@siemens.com

Benoit Nadeau-Dostie
Siemens Digital Industries Software
Ottawa, Canada
benoit.nadeau-dostie@siemens.com

Lori Schramm
Siemens Digital Industries Software
Atlanta, GA, USA
lori.schramm@siemens.com

Martin Keim
Siemens Digital Industries Software
Orlando, FL, USA
martin.keim@siemens.com

2024 IEEE European Test Symposium (ETS) | 979-8-3503-4932-0/24/$31.00 ©2024 IEEE | DOI: 10.1109/ETS61313.2024.10567190

***Abstract*—Error Correction Codes (ECC) in current designs typically serve two purposes. For emerging non-volatile memory types (NVM), such as embedded magnetoresistive random access memory (eMRAM), ECC is necessary to counter the probabilistic behavior of the NVM, rendering the combined NVM/ECC a deterministic memory again. The second and much more prominent usage of ECC today is to protect the system against transient faults in the memory, here typically for SRAMs. On the other hand, new defect types for such emerging memories and new technology nodes may exceed reasonable costs of conventional row and column repair. To improve yield, users explore ECC as an option to augment the repair capability of the memory. This paper brings such an augmentation into a standard memory test and repair flow. It allows a user-defined, post silicon trade-off of using all or parts of the corrective power of the ECC for yield improvements and/or for system protection. Experimental results underline the very low area cost of this augmentation.**

***Index Terms*—Error Correction Codes (ECC), Built-In Redundancy Analysis (BIRA), Built-In Self Repair (BISR)**

I. Introduction

Since NVM technology is still under development, ECC logic and redundant elements are commonly used to improve device reliability [1], [2]. The test and repair solution should offer the flexibility to adjust the number of defects to be corrected using ECC and to collaborate with the Built-In Redundancy Analysis (BIRA) circuitry during memory manufacturing test. Using ECC correction and redundant elements during manufacturing is proposed in [3]. However, most of the techniques rely on failure data collection and off-chip post-processing to determine the final redundant element allocations. This paper describes an autonomous on-chip solution consisting of an augmented BIRA circuitry capable of calculating an optimal repair solution using both redundant elements and ECC, given user-defined requirements. An overview of the standard repair test flow is shown in Section II. The augmented BIRA circuit, referred to as ECC-aware BIRA, is presented in Section III. The built-in solution uses the memory's ECC correction capability and redundant elements to improve manufacturing yield, especially for NVMs. It can also be used with ECC-equipped SRAMs during in-system memory testing. Furthermore, we show a test sequence with fault injection where the ECC-aware BIRA hardware automatically allocates redundant elements in an optimal way. Simulation results are provided to demonstrate the functionality of the proposed solution. Finally, we analyze the area and performance of the solution using a logic synthesis tool in SectionIV.

II. Standard Memory Repair Flow

The memory test hardware involved in testing memories with redundant elements is shown in Figure 1. The memory interface is connected to the memory, on the right, and the memory test controller is on the left. A BIRA module is located inside the memory interface. The elements in red illustrate the new ECC-aware components, which will be introduced in Section III-B. These elements do not participate in the standard memory test flow. The comparator registers

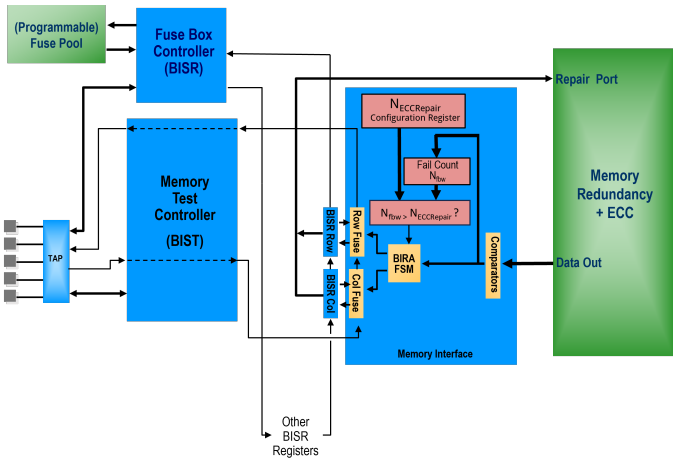


Figure 1: Built-in memory test architecture overview

capture the defect locations during the memory test execution. The BIRA logic uses the values of these registers to calculate a

Complete, Contactless I/O Testing – Reaching the Boundary in Minimizing Digital IC Testing Cost

Stephen K. Sunter & Benoit Nadeau-Dostie, LogicVision, Inc.

Abstract

Embedded test of memory and random logic can enable very low cost ATE to test large, high speed ICs because high quality at-speed tests can be generated on-chip. However, it is also necessary to test the DC and AC parameters of the input/output (I/O) circuitry. This paper describes how most I/O pin characteristics can be tested cost-effectively with a variety of novel techniques that exploit the 1149.1 and 1149.4 test standards. The techniques measure VOL/IOL, VOH/IOH, VIH, and VIL at DC, perform at-speed I/O wrap, and test on-chip power rail impedance, all via minimum pin-count (MPC) access. The 1149.4 bus is also suitable, of course, for testing mixed-signal functions. The paper then discusses costs and benefits of MPC testing of high pin-count ICs on a low cost tester to show that testing costs can be reduced to insignificance.

1. Introduction

Reduced pin-count (RPC) testing has been proposed [1], [2], [3] to facilitate low-cost probe testing of high pin-count ICs – it can also facilitate multi-site testing. These techniques contact between 10 and 64 digital pins, including at least the four or five IEEE 1149.1 Test Access Port (TAP) pins. Conventional probe cards limit robust digital test frequencies to less than 10 MHz, however, probe cards are available that can handle much higher frequencies for a smaller number of probes or a higher cost. Of course, the tester must be able to accurately deliver these higher frequencies (or more accurate signal edge timing).

Although the reported RPC probe techniques test the logic core and embedded memories via scan or BIST, and test the basic structure of wrap-around input/output (I/O) circuitry, the I/O pin DC and AC parameters are not tested, with the exception of pin leakage current as reported in [3][9]. The resultant yield dropout in the subsequent full pin-count (FPC) Final test, due to insufficient I/O parametric performance, will usually be less than 1% as suggested in [2], so not probe testing I/O parametric performance would appear to have minimal cost impact. However, I/O parametric tests are often regarded as essential at Wafer Sort, for several reasons: Wafer Sort and Final (package) tests are typically performed at different temperatures to improve fault coverage; parametric tests during Wafer Sort provide an early indication of processing quality; some ICs are only ever tested at the wafer level, i.e., those destined for chip scale packaging (CSP) or multi-chip modules (MCM).

BIST is widely used for testing embedded SRAM because the quality of simple scan-access testing is inadequate, the number of memories is too many for direct access, the fault coverage of some BIST algorithms has been demonstrated to be equivalent to direct access, the patterns can be applied at higher speed by BIST, and the number of test patterns is too many for single-site testing by conventional ATE. Programmable BIST for embedded DRAM offers the same benefits, plus new algorithms can be applied if new defect mechanisms are found after silicon fabrication. The test time for memory BIST which applies the patterns at-speed is therefore the minimum possible (or very nearly). For example, a megabit of SRAM, partitioned as four 8Kx32b SRAMs, can be tested sequentially by one BIST controller having a 22N algorithm, and operating at 25 MHz, in 70 ms.

Recent experiments by Prof. McCluskey and his students [22] provide an interesting conclusion regarding BIST for random logic: the only combinational logic test that achieved zero test escapes, relative to an exhaustive test, was pseudo-random patterns applied at-speed until every stuck-at fault had been detected at least fifteen times.

Therefore, one can conclude that random logic and embedded memories could be structurally tested by BIST, and that functional tests for these functions might no longer be necessary to achieve zero test escapes (if the BIST is applied consistent with the above experiment). The test time for BIST of logic can be reduced, almost arbitrarily, by reducing the length of the scan chains and increasing the number of scan chains. In practice, the number of scan chains is limited to several hundred per pseudo-random pattern generator (PRPG) to minimize interconnect. Circuit macro-blocks tend to contain less than 50,000 scan bits, so a hierarchical approach that uses one PRPG per macro block is sufficient to reduce all scan chains to less than 300 bits, and many blocks can be tested in parallel. If each pseudo-random scan pattern is shifted at half-speed (to consume power that is comparable to normal operation) but applied at-speed for a few clock cycles around the mission-mode cycle, then test times can be near minimum (for example: 500 bits × 50K vectors / 50MHz = 500 ms).

It has previously been reported [20] that BIST for PLLs can be implemented digitally, and that the test time

0-7803-7542-4/02 $17.00

Configurable BISR Chain For Fast Repair Data Loading

Wei Zou
Siemens Digital Industries Software
Wilsonville, Oregon, United States
zou@siemens.com

Benoit Nadeau-Dostie
Siemens Digital Industries Software
Ottawa, Ontario, Canada
benoit.nadeau-dostie@siemens.com

***Abstract*—Embedded memories take a significant portion of current system-on-chip (SOC) designs. Built-in self-repair (BISR) is widely used to improve the design yield by replacing the faulty elements of a memory with spare ones. BISR programs the repair data obtained by a memory built-in self-test (BIST) circuit to a fuse box and transfers the repair data to the memory when the chip is powered up. However, loading the repair data to the memories using a serial scan chain is very time consuming due to the increased number of memories. To reduce the long repair data loading time, we propose a configurable BISR chain repair system. In the proposed solution, the serial scan chain is partitioned into an optimal number of segments. Only the segments connected to defective memories are scanned. Experimental results show that the proposed method can significantly reduce the repair data loading time compared to two baseline methods.**

Keywords—BISR, BIRA, BIST, Embedded Memory, Yield

I. INTRODUCTION

Current high-density system-on-chip (SOC) designs include a large number of embedded memories. The number of memories can be over thousands and even tens of thousands in some applications. For example, in [1], a Graphcore MK2 IPU has 1472 powerful processor cores, and each core has an unprecedented 900MB memory. Designed tightly to the technology limits, memories are more prone to failures than other circuits, which can affect not only manufacture yield but also circuit reliability adversely. Built-in self-test (BIST) techniques [2] are typically employed to identify defects and problems in the memories. Moreover, a circuit having embedded memories usually includes built-in self-repair (BISR) circuitry [3] for performing a repair analysis (built-in repair analysis or BIRA) and for replacing faulty elements with spare ones. Repair information obtained by the built-in self-repair (BISR) circuitry can be stored in a non-volatile storage device such as a fuse box. When the circuit is powered up, the repair information can be retrieved and loaded. The transportation of the repair information between the memories and the fuse box involves a scan network, which is often referred to as a BISR chain. Data delivery via a single scan network is serial in nature. A circuit can have hundreds or thousands of memories. Using a conventional scan network can take too long time for manufacture and in-system test and repair.

To reduce the repair data loading time when a chip is powered up, two methods were proposed in [4] and [5]. In [4], the failure information coming from different memories or segments of memories tested is merged to calculate a common repair solution which can be broadcasted. Although the proposed method can reduce both the repair hardware overhead and the repair data loading time, it is difficult to achieve more than an order of magnitude due to routing issues. In [5], bypass logic is added to exclude a memory from the BISR chain if a memory is fault free and doesn't need a repair. There is one data bit which controls the bypass logic. If the data bit is '0', the memory will be included in the BISR chain, otherwise it will be excluded from the BISR chain. All the data bits controlling the bypass logic are concatenated into a scan chain dedicated for the chain configuration data shifting. The chain configuration data is programmed to the fuse box during the repair data programming, and it is loaded to the configuration chain before the repair data loading. A similar method is also proposed in [6]. For a mature process, only a very small number of memories need repair. Most of the memories are bypassed during the repair data loading. The repair data loading time can only be reduced by an order of magnitude with this method as the majority of memories only use one or two spares and the length of each repair register associated to a memory is about 10 bits on average. In this paper, we propose a new method which divides the BISR chain into an optimal number of segments and each segment contains a number of repair registers which may or may not be shared by several memories. A segment selection circuit is added to each segment to exclude the segment from the BISR chain if the memories of the segment are fault free. Different from the method of [5], the data bits controlling the segment selection circuits are part of the BISR chain, which reduces the routing overhead by not introducing a new scan chain. The proposed method can further reduce the repair data loading time compared to the method in [5].

The rest of this paper is organized as follows. Section II describes the general repair system widely used in the industry and the prior work on fast repair data loading. Section III describes the proposed configurable BISR chain repair system. Section IV summarizes experimental results. Section V concludes this paper.

II. THE GENERIC REPAIR SYSTEM AND PRIOR WORK OF FAST REPAIR DATA LOADING

The generic repair system widely used in the industry is shown in Fig. 1. For illustration purpose and easy understanding, we only show six memories which are tested by two memory BIST controllers. For each memory, there is a BISR register connected to it. The data stored in the BISR register is used for memory repair. The size of the BISR register is determined by the memory size and the number of spare elements that the memory has. Assume all the memories in Fig. 1 have 128 rows and only one spare row for repair. Any failing row detected during the test can be replaced with the spare row during the repair. For this memory, the BISR

2022 IEEE International Test Conference (ITC) | 978-1-6654-6270-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/ITC50671.2022.00012

2378-2250/22/$31.00 ©2022 IEEE
DOI 10.1109/ITC50671.2022.00012

Regular Paper

Implementing Design-for-Test within a Tile-Based Design Methodology - Challenges and Solutions

Venkat Yellapragada, Suresh Raman, Banadappa Shivaray
Xilinx
India

Ashok Anbalan
Mentor, A Siemens Business
India

Luc Romain, Benoit Nadeau-Dostie, JF Cote, Albert Au
Mentor, A Siemens Business
Canada

Giri Podichetty, Martin Keim
Mentor, A Siemens Business
USA

***Abstract*—A tile based design methodology consists of developing design blocks that are inserted in design layouts by placing blocks next to each other, making a tile-to-tile connection by abutting corresponding physical signal lines at the border of the tile. Very large systems can be easily and rapidly developed by seamlessly integrating tile elements in the layout. Further, the ease of top-level integration underlines the advantages over a bottom-up approach. However, this tile-based approach is incompatible with traditional DFT tools, which were created to work in accordance with the bottom-up design methodology. This paper outlines some of the obstacles to overcome, to support a truly tile-based DFT methodology. We describe here a working solution for a large production design, underlining a successful implementation of a tile-based Memory Test methodology.**

Keywords—tile-based design, bottom-up design, Memory BIST, Memory Repair, IJTAG, IEEE 1687

I. Introduction

A tile-based design methodology is not a new invention. It has been used in industry for many years, see e.g. [1][2]. However, there's no clear definition or description that is universally accepted. For example, in [1] a tile refers to an IEEE 1500 wrapped core. The authors of [1][3] show that such a core-based methodology has its advantages in particular with respect to logic test pattern generation. Similarly, we refer here to a tile as a design object, containing large quantities of logic, as well as embedded memories, both of which must be tested. However, the key difference that will become important here, is that the tiles we use have pass-through signals, allowing (jog-free) abutting of tiles in the layout, eliminating the need for any routing in the parent hierarchy level. Note that our tile-based notation here is driven from layout concepts; the physical location of a signal line (polygon) at the border of one tile matches up to the physical location of the very same signal line in the neighboring tile, and through this make a connection in the layout, without the need of a logical (and physical) port at the IOs of either tile. The latter usually implies some level of routing in the common parent instance, something that is not needed in our methodology.

Further differences are that each of our tiles do not have anchor logic near the ports, like an embedded TAP controller, IEEE 1500 WSP, or a pre-existing IEEE 1687 (IJTAG) host scan interface. Instead we expect the memory test insertion tool to add IEEE 1687 ports as needed, which we may also be used for other IJTAG compliant instruments in the tile. In addition, for the tile-based design methodology to work, all the used DFT tools must insert pass-through signal lines as needed.

The initial problem with the existing memory test tool is that it does not know about the tile-based design methodology. Instead, it follows a hierarchical, core-based design methodology. In this methodology, the DFT interface is inserted in each block, and expected to be connected only to a parent instance. An example is shown in Figure 1. This bottom-up methodology does not work in our design, as it leaves tiles unconnected, breaking the DFT signal connectivity. At the core of this paper, we describe how the used memory test tool can be taught to operate correctly for a tile-based design methodology, including the automation of insertion and connectivity of pass-through signals. The crux of the issue is automation, without which our DFT engineers would have to fix each and every tile connection manually, which is error-prone and time consuming.

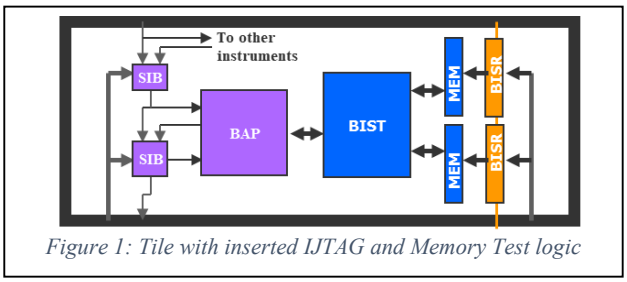


Figure 1: Tile with inserted IJTAG and Memory Test logic

In the next sections, we first review the typical hierarchical, core-based design methodology, and contrast it to the tile-based methodology we use for the production design, shortly outlined thereafter. However, let us first introduce the IJTAG components which form the backbone of the network connecting the memory test hardware components.

978-1-5386-5180-3/18/$31.00 ©2018 IEEE
DOI 10.1109/ITC-Asia.2018.00018

Improved Core Isolation and Access for Hierarchical Embedded Test

Benoit Nadeau-Dostie, Saman M.I. Adham, and Russell Abbott
LogicVision

Editor's note:
IEEE Std 1500 enables automation and hence allows for easier and faster integration of embedded cores into an SoC. This article describes an automated test development system based on the concept of embedded test.
—Erik Jan Marinissen, IMEC

IEEE Std 1149.1 is a widely accepted testability standard that has been in use since the early 1990s.[1] Although the standard was primarily focused on board-level assembly verification testing via the boundary scan register, its test access port (TAP) and many optional provisions make it usable for a broader range of applications. Since its inception, researchers have proposed numerous extensions and applications that allow designers to use the standard's TAP for general system-level test and maintenance tasks and for accessing chip-level testability features.

Until the late 1990s, chip-level applications used TAP for accessing a chip's scan design or on-chip BIST features. However, it became apparent that increasing chip complexity would force new design and test methods. On the design side, circuits are now partitioned into blocks to facilitate synthesis and layout. Iteration turnaround times can be maintained at a reasonable level. Concurrent engineering enables several design teams to develop the various blocks in parallel. Some preexisting blocks can be reused, especially for standard functions, such as the PCI bus interface. Finally, several circuits exploit the architecture's regularity by using several instances of the same block.

These new design methods have introduced completely new test constraints.[2,3] Unlike ICs, cores can't be manufactured and tested individually. Direct access to core ports is virtually impossible. The problem is similar to testing a fully assembled printed circuit board (PCB) without having previously tested the individual components. Test and diagnosis of such a board is possible only if each IC comes with a standard test interface such as IEEE Std 1149.1, which has internal and external test modes. IEEE P1500 was proposed to similarly address the testing of core-based circuits. The internal test information is encapsulated in the form of scan test vectors or BIST procedures, and no design knowledge is necessary. This is especially useful when the core is provided by a third party. Each core's external test information makes it possible to test the connections and logic between cores.

In this article, we describe a new hierarchical embedded-test methodology developed at LogicVision for core-based designs. The wrapper serial port (WSP) of IEEE Std 1500 is at the heart of the test architecture.[4] Each core is equipped with a WSP and controlled by the top-level TAP to provide access to all core test controllers. An improvement to the standard, called *shared isolation*,[5] makes it possible to significantly reduce core isolation costs while enabling at-speed testing of core interface signals. Our methodology provides a fixed core test interface regardless of the number of test resources under its control. The star architecture allows verification and testing of cores independently, enabling concurrent engineering and better diagnostic resolution.

Hierarchical test architecture

Figure 1 shows an example IC using LogicVision's hierarchical test methodology. This simple circuit

0740-7475/09/$25.00 © 2009 IEEE Copublished by the IEEE CS and the IEEE CASS

Low-Power Programmable PRPG With Test Compression Capabilities

Michał Filipek, Grzegorz Mrugalski, *Senior Member, IEEE*, Nilanjan Mukherjee, *Senior Member, IEEE*, Benoit Nadeau-Dostie, *Senior Member, IEEE*, Janusz Rajski, *Fellow, IEEE*, Jędrzej Solecki, and Jerzy Tyszer, *Fellow, IEEE*

***Abstract*— This paper describes a low-power (LP) programmable generator capable of producing pseudorandom test patterns with desired toggling levels and enhanced fault coverage gradient compared with the best-to-date built-in self-test (BIST)-based pseudorandom test pattern generators. It is comprised of a linear finite state machine (a linear feedback shift register or a ring generator) driving an appropriate phase shifter, and it comes with a number of features allowing this device to produce binary sequences with preselected toggling (PRESTO) activity. We introduce a method to automatically select several controls of the generator offering easy and precise tuning. The same technique is subsequently employed to deterministically guide the generator toward test sequences with improved fault-coverage-to-pattern-count ratios. Furthermore, this paper proposes an LP test compression method that allows shaping the test power envelope in a fully predictable, accurate, and flexible fashion by adapting the PRESTO-based logic BIST (LBIST) infrastructure. The proposed hybrid scheme efficiently combines test compression with LBIST, where both techniques can work synergistically to deliver high quality tests. Experimental results obtained for industrial designs illustrate the feasibility of the proposed test schemes and are reported herein.**

***Index Terms*— Built-in self-test (BIST), low-power (LP) test, pseudorandom test pattern generators (PRPGs), test data volume compression.**

I. Introduction

ALTHOUGH over the next years, the primary objective of manufacturing test will remain essentially the same—to ensure reliable and high quality semiconductor products—conditions and consequently also test solutions may undergo a significant evolution. The semiconductor technology, design characteristics, and the design process are among the key factors that will impact this evolution. With new types of defects that one will have to consider to provide the desired test quality for the next technology nodes such as 3-D, it is appropriate to pose the question of what matching design-for-test (DFT) methods will need to be deployed. Test compression, introduced a decade ago, has quickly become the main stream DFT methodology. However, it is unclear whether test compression will be capable of coping with the rapid rate of technological changes over the next decade. Interestingly, logic built-in self-test (LBIST), originally developed for board, system, and in-field test, is now gaining acceptance for production test as it provides very robust DFT and is used increasingly often with test compression. This hybrid approach seems to be the next logical evolutionary step in DFT. It has potential for improved test quality; it may augment the abilities to run at-speed power aware tests, and it can reduce the cost of manufacturing test while preserving all LBIST and scan compression advantages.

Attempts to overcome the bottleneck of test data bandwidth between the tester and the chip have made the concept of combining LBIST and test data compression a vital research and development area. In particular, several hybrid BIST schemes store deterministic top-up patterns (used to detect random pattern resistant faults) on the tester in a compressed form, and then use the existing BIST hardware to decompress these test patterns [6], [7], [20]–[22], [27], [30], [51]. Some solutions embed deterministic stimuli by using compressed weights or by perturbing pseudorandom vectors in various fashions [16], [17], [29], [31], [46], [54], [55]. If BIST logic is used to deliver compressed test data, then underlying encoding schemes typically take advantage of low fill rates, as originally proposed in LFSR coding [24], which subsequently evolved first into static LFSR reseeding [10], [15], [18], [26], [50], [52], and then into dynamic LFSR reseeding [2], [39]. Thorough surveys of relevant test compression techniques can be found, for example, in [23] and [44].

As with conventional scan-based test, hybrid schemes, due to the high data activity associated with scan-based test operations, may consume much more power than a circuit-under-test was designed to function under. With overstressing devices beyond the mission mode, reductions in the operating power of ICs in a test mode have been of concern for years. Full-toggle scan patterns may draw several times the typical functional mode power, and this trend continues to grow, particularly over the mission mode's peak power. This power-induced over-test may result in thermal issues, voltage noise, power droop, or excessive peak power over multiple cycles which, in turn, cause a yield loss due to instant device damage, severe decrease in chip reliability, shorter product lifetime,

Manuscript received November 14, 2013; revised May 5, 2014; accepted June 9, 2014. Date of publication July 14, 2014; date of current version May 20, 2015.

M. Filipek, J. Solecki, and J. Tyszer are with the Faculty of Electronics and Telecommunications, Poznan University of Technology, Poznan 60-965, Poland (e-mail: michal.piotr.filipek@gmail.com; jedrzej.solecki@gmail.com; jerzy.tyszer@put.poznan.pl).

G. Mrugalski, N. Mukherjee, B. Nadeau-Dostie, and J. Rajski are with Mentor Graphics Corporation, Wilsonville, OR 97070 USA (e-mail: grzegorz_mrugalski@mentor.com; nilanjan_mukherjee@mentor.com; benoit_nadeau-dostie@mentor.com; janusz_rajski@mentor.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVLSI.2014.2332465

MBIST Supported Multi Step Trim for Reliable eMRAM Sensing

Jongsin Yun
Mentor, A Siemens Business
Wilsonville, USA
jongsin_yun@mentor.com

Benoit Nadeau-Dostie
Mentor, A Siemens Business
Ottawa, Canada
benoit_nadeau-dostie@mentor.com

Martin Keim
Mentor, A Siemens Business
Wilsonville, USA
martin_keim@mentor.com

Lori Schramm
Mentor, A Siemens Business
Atlanta, USA
lori_schramm@mentor.com

Cyrille Dray
ARM
Sophia Antipolis, France
cyrille.dray@arm.com

Mehdi Boujamaa
ARM
Sophia Antipolis, France
mehdi.boujamaa@arm.com

Khushal Gelda
ARM
Bangalore, India
khushal.gelda@arm.com

2020 IEEE International Test Conference (ITC) | 978-1-7281-9113-3/20/$31.00 ©2020 IEEE | DOI: 10.1109/ITC44778.2020.9325218

***Abstract*—eMRAM (embedded Magnetoresistive Random Access Memory) has many attractive properties such as small size, fast operation speed, and good endurance. However, MRAM has a relatively small TMR (Tunneling Magnetoresistance) ratio, which means a small on-off state separation. It is a challenge to set an optimal reference resistance to reliably differentiate "1" and "0" states. Several trimming circuits were suggested in the literature to adjust a reference value and its search range. The trim setting can be controlled manually by user input; however, it consumes huge test time and requires off-chip engineering analysis to search and apply a trim setting for an individual memory array. In this paper, we will discuss the recent silicon results of fully automated trim process leveraging existing MBIST (Memory Built-in Self-Test) resources and new features to accommodate more complicated multi-step reference setting implementation through minor update of an existing MBIST circuit. The proposed MBIST solution uses a minimal number of tests to analyze massive array properties and automatically set complicated multi-step trim settings within a chip without the need for an external tester or manual adjustments.**

Keywords—MRAM, BIST, yield, reference trim, read operation.

I. Introduction

STT (Spin- Transfer Torque)-MRAM is a type of memory that encodes data as a spin polarity of magnets in its ferromagnetic metal layer. The electron spin polarity of the layer switches based on the write-current direction. As a result, the MRAM cell can be switched to either a high resistance state (anti-parallel, R_{AP}) or a low resistance state (parallel, R_P) [1, 2]. During the Read operation, a sense amplifier latches a differential between the developed voltage or current level of the selected BL (bit line) relative to that of a reference BL. Therefore, the reference level (either resistance, voltage, or current) is one of the most important settings for reliable data sensing.

Due to process variations, On state and Off state memory has a certain range of resistance distribution similar to a normal distribution. An example of On and Off state resistance distribution of an MRAM array is illustrated in Fig. 1 (a). To read all memory data correctly, the reference resistance should be set between the distribution tails of the highest R_P and lowest R_{AP} values. The optimum reference setting is especially challenging in MRAM because the On and Off state resistance separation is relatively narrow compared to other types of resistive memories by an order of magnitude. The small separation of the on/off resistance makes reference setting even more sensitive to variations of resistances, which inevitably happen due to process shift, array configuration, total array size, temperature use case, etc. For that reason, using a fixed hardware preset R_{REF} for the entire array is quite risky to make large array applications. An incorrect reference setting leads to incorrectly interpreting the data, and it also limits the speed of read operation until a suitable differential level is achieved. The silicon measurement results summarized in section IV, allow you to visualize different optimum trim settings required for each individual array segment in the tested chips. To mitigate the reference setting issue, several trimming circuits were suggested in recent MRAM applications which allow post-process reference value control [3, 4]. The reference trim process allows for the update of reference values relative to accessed array properties and improve read margin for reliable sensing; thus ultimately improving yield.

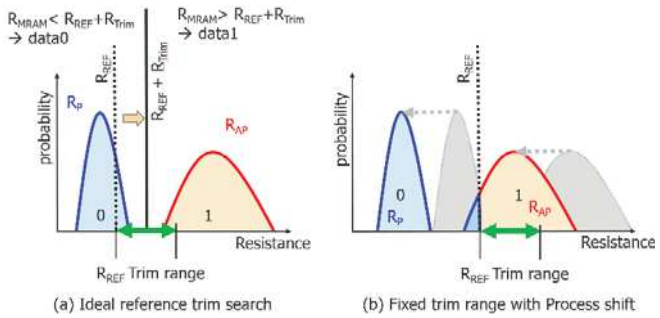


Fig. 1. Distribution of MRAM resistance for value "1" and "0" with sensing reference resistance setting

A trim search function is very efficient when it is executed as a part of MBIST. MBIST can test a chip without using expensive ATE (Automatic Test Equipment), and it is able to automatically evaluate the individual trim setting for each array based on a real-time measured test result. Although a trim search can be accomplished through external IO, collecting and analyzing the full set of memory data for each trim setting is a painful challenge. It requires an impractically large amount of

2020 25th IEEE European Test Symposium (ETS)

MBIST Support for Reliable eMRAM Sensing

Jongsin Yun
Mentor, A Siemens Business
Wilsonville, USA
jongsin_yun@mentor.com

Benoit Nadeau-Dostie
Mentor, A Siemens Business
Ottawa, Canada
benoit_nadeau-dostie@mentor.com

Martin Keim
Mentor, A Siemens Business
Wilsonville, USA
martin_keim@mentor.com

Cyrille Dray
ARM
Sophia Antipolis, France
Cyrille.Dray@arm.com

Mehdi Boujamaa
ARM
Sophia Antipolis, France
Mehdi.Boujamaa@arm.com

***Abstract*—eMRAM (embedded Magnetoresistive Random Access Memory) is an attractive solution in many non-volatile memory applications because of its small size, fast operation speed, and good endurance. However, due to a relatively small on/off resistance separation, it is a challenge to set an optimal reference resistance to reliably differentiate between a read memory data "1" and "0". Several trimming circuits are described in the literature to finely adjust a reference resistance value. These circuits are controlled from chip inputs causing time-consuming tests and off-chip engineering analysis. This paper presents a fully automated on-chip trimming process leveraging existing memory BIST (Built-In Self-Test) resources. It analyzes a massive amount of array property data with a minimal number of tests and optimizes the reference trim settings on-chip without the need for any external intervention.**

Keywords—MRAM, yield, trim, reference, read operation.

I. Introduction

STT (Spin-Transfer Torque)-MRAM is a type of memory that encodes data as a spin polarity of magnets in its ferromagnetic metal layer. Based on the polarity of the layer, the resistivity of the MRAM cell can be switched to either a high (anti-parallel, R_{AP}) or a low (parallel, R_P) value as shown in Fig. 1. The switching of a memory datum is achieved by spin-polarized tunneling current flowing through MTJ (Magnetic Tunnel Junction) which exerts torque on the local magnetization. Slonczewski [1] and Brinkman [2] well explained the switching mechanisms and tunneling conductance behaviors. Several studies show fast switching and high endurance of MRAM to propose MRAM as a potential replacement solution for last level cache memory [3-6].

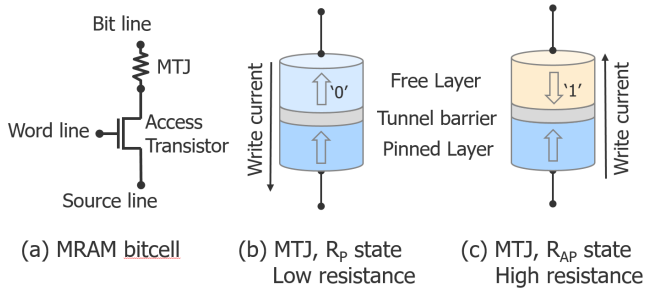


Fig. 1. The component of MRAM bitcell and its MTJ states

During Read operation, an MTJ resistance in a selected bitcell (R_{bit}) is compared with a reference resistance (R_{REF}). The read value is said to be "1" ("0") when its resistance is higher (lower) than that of R_{REF}. The resistance comparison is done by amplifying the differential of Bit Line (BL) discharging rates between the active BL and the reference BL.

An example of On and Off state resistances distribution of an MRAM is illustrated in Fig. 2 (a). To read all memory data correctly, the reference resistance value must be set lower than any R_{AP} state ("1") resistance and, at the same time, higher than any R_P state ("0") resistance. The reference resistance should be set between the distribution tails of R_P and R_{AP}. These tail bits are weakest bits in the array that is unstably changing by process update, array configuration, total array size, temperature use case, etc. Setting R_{REF} is especially challenging in MRAM because the On and Off state resistance separation is 1 or 2 orders of magnitude smaller in MRAM with respect to other types of resistive memories. Moreover, inherent variations from a large memory array and inhomogeneous temperature sensitivity between R_P and R_{AP} makes it even more challenging to set the correct reference resistance.

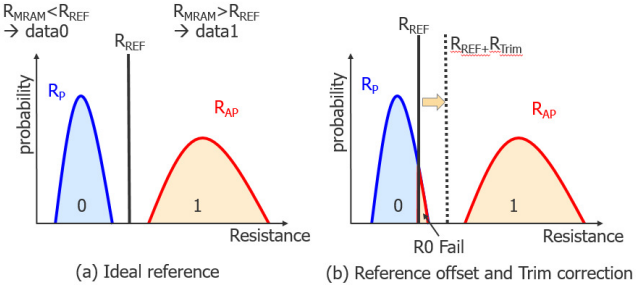


Fig. 2. distribution of MRAM Resistance for value "1" and "0" with sensing reference resistance setting.

A hardware preset R_{REF} may become inadequate for some of the bit-cells due to process update and other variances that happen after the design completion. This reference setting will incorrectly interpret some of the logical "0" state memory data as a logical "1" and hence cause a Read0 (R0) fail as we illustrate in Fig. 2 (b). Reference trimming circuits are suggested in recent MRAM application literature [3, 4] to accommodate this issue. The authors connect switchable series resistances to the reference BL to add additive resistance R_{Trim} into R_{REF}. This shifts the overall reference resistance value to a proper value. Such a trimming circuit allows post-manufacturing fine-tuning of the reference resistance from the device IOs. The reference trim adjustment process improves the read margin for both of the data types '1' and '0', therefore, increasing reliable sensing of the stored logical value and increase yield.

Collecting and analyzing the full set of memory properties is another challenge. It requires an impractically large amount of test resource to measure and analyze analog properties of a full array, such as resistance value of each cell at the state of "1" and "0". If only a small number of cells would be sampled for the test, it may be difficult to capture the tail behavior of

978-1-7281-4312-5/20/$31.00 ©2020 IEEE

Memory repair logic sharing techniques and their impact on yield

Benoit Nadeau-Dostie
Mentor, A Siemens Business
Ottawa, Canada
benoit_nadeau-dostie@mentor.com

Luc Romain
Mentor, A Siemens Business
Ottawa, Canada
luc_romain@mentor.com

2020 IEEE International Test Conference (ITC) | 978-1-7281-9113-3/20/$31.00 ©2020 IEEE | DOI: 10.1109/ITC44778.2020.9325280

***Abstract*— Techniques for sharing memory repair logic amongst memories are described. The techniques allows reducing silicon area and loading time of repair information upon power up. The impact on yield is predicted using two different methods based on defect density and clustering or past silicon experience.**

Keywords*— *Memory repair, silicon area, repair time, yield

I. INTRODUCTION

The amount of memory embedded in integrated circuits continuously increases. In [1], it is reported that as much as 400Mbits had been observed at 28nm and 500Mbits at 16nm. This memory is distributed over thousands and even tens of thousands of memory instances depending on the application. The number of bits per instance does not necessarily increase. It is the number of instances which tends to increase in order to achieve parallel processing. For example, increasing the number of cores in a processor chip or increasing the number of channels of a telecommunications chip.

Memory repair has become essential to achieve sufficient yield on circuits with such high memory content. However, only a small percentage of memories actually need to be repaired on any one chip. This fact allows to significantly compress the memory repair information stored in a Non-Volatile Memory (NVM) as shown in [2] and [3], for example.

In [4], sharing of the repair analysis logic calculating a repair solution for each memory is described to reduce area. However, test time is not reduced because memories sharing the repair analysis logic need to be tested one at a time. Also, there is no reduction of the power up time since the number of repair registers to be loaded is not reduced. Authors in [5] address the first limitation and explain in more detail how the repair analysis logic can be shared for memories of different dimensions and different physical address mappings. This is done by using an "align and merge" logic module allowing the repair analysis logic to calculate a solution in a unified repair format. Again here, the number repair registers is not reduced and the area of the repair analysis logic increases linearly with the number of memories tested in parallel.

The purpose of our work is to further reduce the amount of repair logic and the time required to load repair registers upon power-up. We propose to achieve both goals by merging failure information coming from different memories or segments of memories tested to calculate a common repair solution which can be broadcasted. A method to limit the potential yield loss resulting from using a common repair solution which might not be optimal is also proposed.

Section II describes the hardware modifications required to share the repair logic. Section III analyzes in detail the potential impact on yield followed by some conclusions.

II. REPAIR LOGIC OPTIMIZATION

Each Memory BIST controller is typically shared by several memories. Tens or even hundreds of memories can be tested by a single controller. These memories can be different in term of sizes, port configurations (single/multi-port), physical organization (with/without column multiplexing) and repair method (row-only, column-only, row and column). Access to these memories can be direct or indirect through a shared bus such as in [6]. Memories can be tested in parallel, in series or a combination of the two. All these aspects are considered in our proposed solution for sharing the repair logic.

The proposed method does not require modifying existing partitioning algorithms used to assign memories to controllers. These algorithms are mainly driven by parameters such as power domains, clock domains, memory placement, test time, power levels, and others. The proposed method creates repair logic sharing groups after this initial partitioning. These groups are local to each controller. The groups are automatically defined based on different criteria explained later and a user-defined limit defining the maximum size of each group. For example, a group might consist of at most 100K bits of memory. This could represent several memories, a single memory or a portion of a large memory. The built-in repair analysis (BIRA) module associated to a group must be able to compute a common repair solution even if the group is composed of memories with different characteristics (size, physical organization, pipeline depth, etc…) similar to what the memory BIST controller itself can do. This means that the column address range processed by the BIRA module can be defined by one memory and the row address range defined by a different memory.

A BIRA module allocates at most one unique spare row unit and one unique spare column unit within a group. This means that if a memory has multiple segments, each with their own spare resources, the same repair solution is applied for all segments. A typical example is a memory with two segments, left and right, and each segment has one spare IO which can be allocated independently if repair sharing is not enabled.

978-1-7281-9113-3/20/$31.00 ©2020 IEEE

OCI: Open Compression Interface

Bruce Cory, Rohit Kapur, Mick Tegethoff, Mark Kassab, Brion Keller,
Kee Sup Kim, Dwayne Burek, Steve Oakland, and Benoit Nadeau-Dostie

Abstract

Before on-chip scan compression, it was possible to use different EDA tool vendors to do scan insertion, pattern generation, and diagnosis. On-chip scan compression changed that use model since each tool vendor supplied a different type of scan compression logic and had tool-specific ways to pass necessary information from scan insertion to pattern generation and from pattern generation to diagnosis. OCI (Open Compression Interface) is a standardization of how the necessary data is passed from test logic insertion to pattern generation to diagnosis such that different vendors can be used for each step independent of the on-chip scan compression logic used. This document discusses the need for OCI and gives a conceptual overview of the OCI standard.

1. Introduction

As technology shrinks, more and more logic is able to fit into the same die area. While a few design companies are shrinking the size of their die, most are combining multiple functionalities or increasing the power of the functionality available and keeping the total die size the same or even larger. This, plus the need for more types of test patterns to maintain quality of test levels at lower technology nodes, is causing the amount of test data needed to verify that the parts are manufactured correctly to skyrocket. The rate is far greater than the growth of tester memory can economically accommodate and also causes issues with the time required to test each part. The main way used to handle these problems is to use more and more complex on-chip scan compression logic. The use of on-chip scan compression logic has been steadily growing for the last several years.

Due to growth in demand for on-chip scan compression, solutions are available from most EDA test vendors. Some example products include OPMISR, TestKompress, DFT compiler Max, VirtualScan, and ELT-Comp. Some design companies have published papers with details of their in-house compression schemes. One example is X-compact. Academic papers on test compression have been published for many years with many of the concepts being used by industry. One example of this is Illinois scan. Each of these compression schemes has similar concepts but different implementation details.

Flows for scan-based test tools are broken into three main stages. Test logic insertion, pattern generation, and diagnosis. Test logic insertion does insertion and verification of test logic. Pattern generation uses the test logic to make test patterns that can be used to verify if the design is fabricated correctly. Diagnosis is used to identify the failing location in a specific device. Diagnosis information can be used to increase future yield and to solve problems that keep a design from going to market.

Most on-chip scan compression schemes are only supported by a limited number of test vendors with most only supported by one vendor. This causes many issues in the test community. Figure 1 shows one example of the issues caused by vendor-specific on-chip scan compression logic and flows.

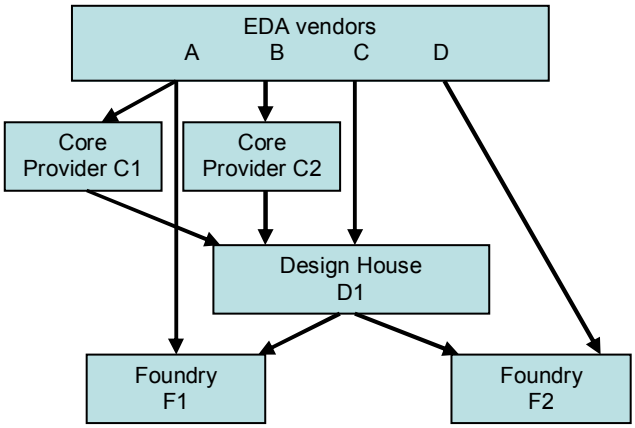


Figure 1: Example use model when on-chip scan compression is not interoperable

Companies A, B, C, and D each have different compression logic and flows only supported by their tool. Hard core provider C1 requires in-core scan compression logic using tool A while hard core provider C2 requires in-core scan compression logic using tool B. Both core developers need to own one tool to insert and verify the on-chip scan compression logic. The design house (D1) is making a design using core C1, C2, plus other in-house developed logic and uses tool C for its in-house on-chip scan compression insertion, pattern generation, and diagnosis. The design house must own three vendor tools to do pattern generation and diagnosis for the different compression structures of the design. Design house D1 then plans to produce the design using foundries F1 and F2. F1 uses tool A for diagnosis while F2 uses tool D for diagnosis. Foundries F1 and F2 need to support 3 or 4 different tools for diagnosis. The quality tradeoffs, support burden, design size overhead, and expense incurred by the design house and foundry because tool C and D can't do pattern generation and diagnosis using the

1-4244-0292-1/06/$20.00 © 2006 IEEE

Power-Aware At-Speed Scan Test Methodology for Circuits with Synchronous Clocks

Benoit Nadeau-Dostie*, Kiyoshi Takeshita, Jean-François Côté
LogicVision Inc
* : contact and presenter

Abstract

The BurstMode™ test clocking methodology, first presented in [8]*, is improved to handle circuits with synchronous clocks of different frequencies. An on-chip clock controller allows to select a large number of clock waveforms necessary to test synchronous cross-domain paths at-speed and control supply voltage variations. The methodology is applicable to both ATPG and BIST and only requires combinational analysis tools. The methodology is applied to a large circuit to adjust power supply margins of an at-speed BIST test.*

Topics: Power issues in test, high-speed digital test, case study

I. Introduction

Defects that affect circuit speed are more prevalent in sub-90 nm CMOS ICs making it more important to provide a thorough at-speed test. Another trend is the replacement of functional test by structural test to implement all tests. Functional tests are too difficult to implement and have limited diagnostic capabilities. Diagnosis is especially important in a context of debug and yield learning. Scan-based structural tests automatically achieve high coverage of DC defects and have excellent diagnostic capabilities.

However, implementing an at-speed structural test is more difficult than simply clocking a circuit at functional speed during the capture cycle. Circuits have several characteristics that complicate the task. Functional clocks can cover a wide frequency range, and can be asynchronous (i.e., non-integer frequency ratio and/or unknown relative phase) or synchronous (i.e., their frequency ratio is fixed as well as their relative phase) to each other. Asynchronous cross-domain paths must be correctly handled to prevent false timing violations caused by false paths while providing proper test of the interface logic. Within each domain, paths with multi-cycle delays are sometimes used, instead of pipelining, to implement functions at a lower cost. Gated clocks are often used for low power designs. PLLs are used to control groups of synchronous clocks and/or multiply external reference clock frequencies. Synchronous cross-domain paths must be tested for propagation and hold times.

For circuits with multi-cycle paths and multiple synchronous clocks, it is necessary to have the possibility of testing all paths simultaneously. This is needed to catch defects due to crosstalk or other defects related to local supply variations. Transitions in coupled net pairs must coincide as much as possible over the normal process variations spread [4] to detect these defects. The detection of crosstalk faults requires an excellent coverage of transition faults. Also, the complexity introduced by the simultaneous handling of multi-cycle paths and synchronous clocks makes the fault simulation and ATPG more complex.

An at-speed test method should impose a minimum number of test-specific timing constraints on the design. These constraints should be easy to meet so that timing closure does not become more difficult to achieve. Preferably, the clocking methodology should be applicable to BIST and ATPG, including test compression applications.

Other design aspects affect the implementation of an at-speed test. The circuit is designed to perform at a certain temperature and supply voltage. Circuit activity has a direct influence on the average current, local temperature, and reduction of supply voltage (di/dt, IR-drop) which in turn influence the timing. For example, in [5], the authors show that the clock period varies by more than 15% due to supply variations, as shown in Fig. 1 The supply variations are due to the sudden change in circuit activity (di/dt) occuring during the application of two consecutive clock cycles during the capture phase of a scan test. In [6], the delays of specific paths were measured for different values of don't care bits within the test pattern. The delay varied by 10% to 30%. The two effects are independent so the total imprecision can be even larger. Clearly, the test clocking methodology must

1-4244-4203-0/08/$20.00 ©2008 IEEE

Power Aware Embedded Test

Xijiang Lin, Elham Moghaddam, Nilanjan Mukherjee
Benoit Nadeau-Dostie, Janusz Rajski
Mentor Graphics Corporation
Wilsonville, OR 97070, USA

Jerzy Tyszer
Poznań University of Technology
60-965 Poznań, Poland

***Abstract*—In this paper we examine several embedded low power test schemes that we have proposed over the last few years. These solutions are aimed at reducing the switching activity during all scan-based test operations, particularly including those developed for BIST or deployed to perform on-chip test data compression.**

I. Introduction

Embedded test resources are being viewed increasingly as essential to reduce test cost. In particular, scan-based DFT schemes have gained broad acceptance as reliable test solutions. When it comes to energy consumption, however, scan-based test operations can dissipate much more power than low-power circuits were designed to function under. With overstressing devices beyond the mission mode, reductions in the operating power of ICs in a test mode have been recurring themes for years. A full-toggle scan pattern may draw several times the typical functional mode power, and this trend continues to rise, particularly over the mission mode's peak power. This so-called power-induced over-test may result in thermal issues, voltage noise, power droop, or excessive peak power over multiple cycles, which, in turn, cause a yield loss due to instant device damage, severe decrease in chip reliability, shorter product lifetime, or a device malfunction because of, for example, timing failures following a significant circuit delay increase.

Abnormal switching activity may also cause fully functional chips to fail during testing because of phenomena such as IR-drop, crosstalk, or di/dt problem. IR-drop refers to the power-rail voltage decrease due to higher current and the resistance of the devices between the rail and a circuit's node. Its large value increases path delays, and this degradation only worsens with elevated test frequencies or low supply voltages. Therefore, a device passing a slow speed scan test at nominal voltage may fail at-speed scan test where either the capture frequency is raised, the supply voltage is lowered, or both.

The underlying mechanism in crosstalk is related to capacitive coupling between IC's neighboring nodes. If the aggressor and victim nodes switch in the same direction, a delay on the victim node decreases, whereas it tends to increase, if they switch in opposite directions. Depending on the extent of coupling and a switching pattern during at-speed test, a failure can only be observed during testing, not in the functional mode.

The di/dt problem occurs in circuits using at-speed scan tests and is due to the sudden current changes within a few nanoseconds when applying the capture clock burst (typically a pair of launch and capture cycles) after a long pause where all clocks are inactive. Hence, the power distribution network may not be able to maintain the supply voltage in desired limits (the supply lines become unstable and reach levels below the ones encountered in a mission mode). The circuit might appear to be faster or slower than it actually is. It seems to be faster, if the clock stretching effect is dominant, and slower, if the combinational slowdown is forceful due to the supply variations.

The problem of excessive power dissipation during scan testing is typically split into two separate domains dealing with power reduction in shift and capture modes, respectively. Accordingly, numerous techniques for test power reduction and management have been proposed, focusing on test scheduling and reordering, partitioning and modifications of scan chains, scan cell reordering, as well as transition blocking and various forms of clock gating. Similar schemes have also been proposed for built-in self-test (BIST), including low-transition pseudorandom test pattern generators and gated scan cells.

Other solutions tailor patterns to the requirements of tests with the lowered switching activity, for instance, by reassigning certain non-random values to unspecified positions in test cubes causing power violations. It minimizes the number of transitions during scan loading, as done in power-aware ATPG. Deterministic test vectors can yield patterns leading to a reduced switching activity because of their usually low fill rates. Instead of random fill, don't care bits may assume (in the process of X-filling) values minimizing the amount of transitions during scan-in shifting. Other forms of X-filling decrease capture power by assigning particular values to unspecified bits so that the number of transitions at the outputs of scan cells in the capture mode is minimized.

On-chip test compression is facing similar test power problems. As on-chip decompressors expand and subsequently load test cubes into scan chains, don't care bits are typically filled with random values, and therefore the amount of flip-flop toggling during test may result in a power droop condition. The bulk of test power consumption can also be attributed to unloading test responses. Several low power test data encoding schemes were presented. Some of them rest on static LFSR reseeding techniques with certain extensions that allow reducing the scan-in transition probability. Other methods reduce test power in dynamic reseeding by using available encoding capacity to limit transitions in scan chains. These techniques freeze a decompressor in certain states by providing control data through external channels. It allows loading scan chains with decompressed patterns having low transition counts, and thus reduced scan-in power dissipation.

1081-7735/11 $26.00 © 2011 IEEE
DOI 10.1109/ATS.2011.49

2009 24th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems

Reducing Test Point Area for BIST through Greater Use of Functional Flip-Flops to Drive Control Points

Joon-Sung Yang[1], Benoit Nadeau-Dostie[2] and Nur A. Touba[1]

[1]Computer Engineering Research Center
Dept. of Electrical & Computer Eng.
University of Texas, Austin, TX 78712
{jsyang, touba}@ece.utexas.edu

[2]Logic Vision, Inc.
25 Metro Drive, Third Floor
San Jose, CA, 95110
benoit@logicvision.com

Abstract

Recently, a new test point insertion method for pseudo-random built-in self-test (BIST) was proposed in [Yang 09] which tries to use functional flip-flops to drive control test points instead of adding extra dedicated flip-flops for driving the control points. This paper investigates methods to further reduce the area overhead by replacing dedicated flip-flops which could not be replaced in [Yang 09]. A new algorithm (alternative selection algorithm) is proposed to find candidate flip-flops out of the fan-in cone of a test point. Experimental results indicate that most of the not-replaced flip-flops in [Yang 09] can be replaced and hence even more significant area reduction can be achieved with minimizing the loss of testability.

1. Introduction

Built-in self-test (BIST) embeds test pattern generation and output response analysis on-chip. It provides numerous advantages in terms of reducing test generation costs, reducing tester storage requirements, allowing the rapid application of many patterns to target non-modeled faults, test reuse, and in-field testing where there is no access to a tester. The most efficient BIST techniques are based on pseudo-random testing where the test patterns can be generated using a linear feedback shift register (LFSR) which has a very compact structure. However, the presence of *random-pattern-resistant (r.p.r.)* faults which have low detection probabilities may prevent pseudo-random BIST from achieving sufficiently high fault coverage. There are two general approaches for improving the fault coverage for pseudo-random BIST: (1) modifying the pattern generator so it generates patterns to detect r.p.r faults, and (2) modifying the circuit-under-test (CUT) to eliminate the r.p.r. faults by increasing their detection probability.

In pattern generator modification, a number of techniques have been proposed including weighted pattern testing [Pomeranz 92], [Bershteyn 93], [Kapur 94], [Jas 01], pattern mapping [Chatterjee 95], [Touba 95a, 95b], LSFR reseeding [Konemann 91, 01], [Hellebrand 92, 95], [Krishna 01], [Rajski 02] and others.

In CUT modification, test points are inserted [Eichelberger 83] to improve the fault coverage. Observability is enhanced by adding observation points and controllability on a particular node is enhanced by adding a control point. Control points insert AND or OR gates at a node and are activated by pseudo-random values from a dedicated flip-flop during BIST operation. Since test point insertion (TPI) adds extra gates in a design, area and performance overhead are issues. [Krishnamurthy 87] proved that finding optimal locations in circuits with reconvergent fanouts is NP-complete and hence there has been a lot of research on test point insertion techniques. Test point insertion

1550-5774/09 $26.00 © 2009 IEEE
DOI 10.1109/DFT.2009.33

ScanBist:
A Multifrequency Scan-Based BIST Method

BENOIT NADEAU-DOSTIE
DWAYNE BUREK
ABU S.M. HASSAN
Bell-Northern Research

The authors present ScanBist, a low-overhead, scan-based built-in self-test method along with its performance in several designs. A novel clock synchronization scheme allows at-speed testing of circuits.

BUILT-IN SELF-TEST is gaining popularity as a means to address test issues at the different packaging levels of digital systems. BIST does not require that patterns be stored in the test equipment, which simply provides a clock and a few control signals. This feature becomes especially important when testing high-performance systems. BIST makes the chip, board, and system more independent of the specific test resources available at each manufacturing stage. It also provides a convenient method of applying more test patterns to compensate for the weaknesses of the stuck-at fault model.[1]

Specialized self-test circuits for embedded blocks (such as memories and programmable logic arrays) have reached a good level of maturity, and existing methods provide the basis for making adequate quality-area-performance trade-offs. However, BIST has a great deal of room for innovation, particularly in unstructured or random logic that requires testing for performance when a variety of design styles apply. At the same time, the new techniques that provide this capability should not significantly increase either the implementation cost or the design interval of the system.

In telecommunications applications, designers often create circuits that operate at different frequencies derived from at least one reference frequency. For example, designers embed communication protocols in a hierarchical fashion in a single frame of data. Several finite-state machines, working at appropriate subfrequencies, then extract the embedded protocols by simple decimation of the input sequence. Using subfrequencies derived from a master clock instead of the master clock itself saves power, reduces electromagnetic interference, and reduces silicon area.

Traditionally used to generate very high quality tests for faults detectable at low speed, Scan has also demonstrated efficiency in testing for delay faults by accurately controlling all clock phases from a tester.[2] However, BIST does not lend itself to this approach. LeBlanc developed a scan-based BIST solution, LSSD on-chip self-test (LOCST), that is applicable at all levels of system integration.[3] However, he used a single test frequency and limited the speed of application of the scan patterns generated by the BIST circuitry to the lowest frequency used in the system. Therefore, part of the logic operates at an untested speed in systems with more than one frequency.

Overview of ScanBist

Figure 1 (next page) shows a block diagram of the ScanBist implementation. The IEEE Standard 1149.1 test access port (TAP)[4] connects to a macro circuit called the ScanBist core. In turn,

0740-7475/94/$04.00 © 1994 IEEE

ScanBist
A Multi-frequency Scan-Based BIST Method

Benoit Nadeau-Dostie, Dwayne Burek, Abu S. M. Hassan

Bell-Northern Research Ltd.
P.O. Box 3511 Station C
Ottawa, Ontario, Canada K1Y 4H7

Abstract

This paper presents a BIST technique that allows the synchronization of multiple scan chains clocked at different frequencies. The technique is used to improve performance testing of scannable circuits. A few new design rules and small modifications to the existing ATPG were necessary to implement the technique.

Introduction

Built-in Self-test (BIST) is gaining popularity as a means to address test issues at the different packaging levels of digital systems. One of the benefits of BIST is the fact that no patterns need to be stored in the test equipment, which is simply required to provide a clock and a few control signals. This is especially important when high-performance systems are being tested. BIST also makes the chip/board/system more independant of the specific test resources available at each manufacturing stage. BIST is also a convenient way of applying more test patterns to compensate for the weaknesses of the stuck-at fault model [Pan90].

Specialized self-test circuits for embedded blocks (memories, PLAs, etc...) have reached a good level of maturity and adequate quality-area-performance trade-offs can usually be made based on the existing methods. However, there is much room for innovation in the field of BIST for unstructured or random logic that needs to be tested for performance given the various design styles.

It is frequent, in telecommunications applications, to design circuits requiring to operate of different frequencies derived from one (or more) reference frequency. For example, communication protocols can be embedded in a hierarchical fashion in a single frame of data and several finite-state machines, working at appropriate sub-frequencies, can be used to extract embedded protocols by simple decimation of the input sequence. Using sub-frequencies derived from a master clock instead of the master clock itself can save power, reduce electromagnetic interference and reduce silicon area.

Scan techniques have been traditionally used to generate very high quality test for faults detectable at low speed. The speed of application of the scan patterns is typically limited to the lowest of the inverse of the longest propagation path delay found on the chip or the frequency of the test clock often not optimized for high speed operation. Delay faults are not adequately tested in such an environment.

In this paper we describe a BIST technique for scan-based designs, called ScanBist, that removes some of the barriers for performance testing in a scan environment. The first section provides an overview of the technique. The BIST hardware, the clocks synchronization, the fault coverage considerations and the software support associated with this technique are addressed in separate sections. Conclusions and results are also presented.

Overview of ScanBist

ScanBist is a BIST technique for scan-based designs that allows synchronization of multiple chains running at different frequencies. The frequencies can be generated by division of a reference (system) clock or driven from primary inputs or both. The technique is a significant improvement over the well-known STUMPS method [Bar82] developed by IBM. The synchronization of the various scan chains is such that the performance of the circuit can be tested in an environment that is as close as possible to the reality. An exhaustive stuck-at and transition fault coverage analysis can be performed using a fault simulator for combinational circuits.

INTERNATIONAL TEST CONFERENCE 1992
0-8186-3167-8/92 $3.00

SCAN TESTING OF LATCH ARRAYS

Michael M.Y. Hui Benoit Nadeau-Dostie

Bell-Northern Research Ltd.

Abstract

A novel scan-based test method that allows latch-based arrays to be fully tested using conventional scan software tools is described. Very little support circuitry and simple modeling are required. Several useful applications of the method are described. The method was used on several production chips.

Introduction

A digital integrated circuit designed to be tested using scan design techniques must be decomposable into a set of registers and combinational blocks [1,2]. A popular style of scan, known as the Stanford style, requires the use of D-type flip-flops as the only valid storage element for registers. The registers are connected such that they operate as shift registers in a special mode of operation of the circuit called scan mode. In scan mode, the state of the registers can be controlled (observed) from a serial test input (output) pin. Figure 1 shows an example of such a circuit. Only one shift register (or scan chain) is shown for simplicity. The main advantage of this test method is that only the combinational part of the circuit needs to be considered during the test generation step. It is well known that the complexity of test generation for combinational circuits is significantly less than for sequential circuits.

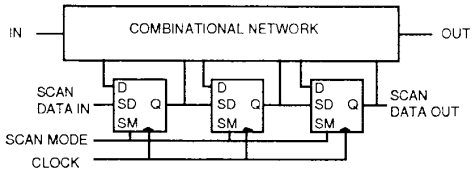


Figure 1

This test method can become expensive for register intensive designs. It would be nice to relax the requirement that imposes the use of full "scannable" D-type flip-flops for every storage element of the circuit under certain circumstances and yet be able to fully test these "illegal" storage elements using the same software tools. For example, small FIFOs, addressable control/status registers and other storage blocks are commonly found in digital circuits. Although these circuit structures can be built using arrays of "scannable" D-type flip-flops, it is more attractive to use transparent latches from a silicon area point of view (3 to 1 ratio on average). Embedded memories could also be used in some cases but they would require different test generation tools because of their specific failure modes [3-5]. Also, embedded memories might not be available for a particular technology, especially when gate-arrays are used.

We have devised a scan-based test method that allows latch-based arrays to be fully tested using conventional scan software tools. With the addition of support circuitry and some restrictions on how the latches may be connected, each latch may be modeled as a combinational gate to the pattern generation software.

This paper is divided as follows. The method is introduced in section II. Applications of the method are described in section III. Results are presented in section IV followed by the conclusion.

Method

The principle of the method is to make the transparent latches used to build the various registers and arrays look like combinational gates. This principle was used in [6]. In this paper, it was proposed to force all latches into their transparent mode for the whole duration of the scan test to verify the interface between the array and the rest of the circuit. However, this method can not test the ability for the latch to hold any data.

0-7803-0623-6/92 $3.00

SERIAL INTERFACING FOR EMBEDDED-MEMORY TESTING

BENOIT NADEAU-DOSTIE
ALLAN SILBURT
BNR

VINOD K. AGARWAL
McGill University

The authors present a serial interfacing scheme in which several embedded memories share the built-in, self-test circuit. For external testing, this approach requires only two serial pins for access to the data path. There is considerable savings in routing area and fewer external pins are needed to test RAMs with wide words, such as those in telecommunication ASICs. Even though the method uses serial access to the memory, a test pattern is applied every clock cycle because the memory itself shifts the test data. The authors have adapted this method to four common test algorithms. In their implementations of the BIST circuitry on several product chips, they found the area overhead acceptable.

Application-specific digital ICs have become increasingly memory intensive in recent years. The design flexibility and performance offered by module generators for single-port and multiport SRAMs and content-addressable memories are opening up new possibilities in system design as the constraint of standard memory dimensions and operating modes is relaxed. However, these new possibilities in design are posing several new challenges for test. For example, today's telecommunication ICs often have a variety of multiport memories on one chip, some with very wide words. External testing becomes complex because it is difficult to bring all the signals to the pins. Tester hardware and design-automation tools also impose limitations on external testing because usually neither can support the large algorithmic pattern sequences in simulation or high-level test generation. Moreover, even if an algorithmic memory test was expanded into individual vectors, they still must be stored and loaded. These requirements could easily exceed a tester's maximum pattern depth, particularly if the patterns had to be multiplexed.

It is these challenges to external testing that are making built-in self-test more attractive to chip designers and manufacturers. Additional incentives for BIST are the ability to provide full-speed tests with minimal test hardware and the reuse of these tests for diagnostics at the board and system levels.

Various BIST algorithms and techniques have been proposed,[1-8] most of which evaluate all the bits of a memory word in parallel as it is read. Some[1,4,6,7] require modification of the RAM circuitry, which may not be practical or even possible in many ASIC design environments. Others[4,6,7] perform signature analysis and then compress the data. These techniques must then consider aliasing uncertainties when calculating fault coverage.[9] We can encounter significant problems when applying these BIST schemes to chips that have multiple embedded RAMs of varying sizes and port configurations. If each memory required a dedicated BIST circuit, the chip area devoted to testing would be unacceptably high. A better approach is to share BIST circuitry among several RAM blocks. We would also want to reuse as much test circuitry as possible in the normal operation of the chip, called the mission mode. With existing BIST schemes, sharing the circuitry between BIST and mission mode can be quite difficult.

Our solution is a serial interfacing technique for embedded RAMs that allows the BIST circuit to control a single bit of a RAM's (or group of RAMs') input data path. Only one bit of the output data path is available to the BIST circuit for observation while the test algorithms are executed.

0740-7475/90/0004-52$1.00©1990 IEEE

Structural test with functional characteristics

Benoit Nadeau-Dostie[1]*, Muhiuzzaman Mahmud[2], Jean-François Côté[1], Fadi Maamari[1]

1: LogicVision Inc
2: Freescale Semiconductor Inc
* : contact and presenter

Abstract

A novel structural test clocking architecture is proposed to reproduce characteristics of the functional mode of operation that are essential to a high-quality test. Bursts of functional mode clocks are controlled to provide accurate timing tests taking into account functional timing constraints such as multi-cycle paths and false paths. Any mix of asynchronous and synchronous clocks is supported. Long term (Idd, temperature) and short term (IR-drop) power characteristics related to circuit activity are independently controlled.

I. Introduction

Defects that affect circuit speed are becoming more prevalent and it is important to provide a thorough at-speed test. Another trend is the replacement of functional test by structural test methods to implement all tests. However, it is not trivial to implement an at-speed structural test. Circuits have several characteristics that complicate the task. Functional clocks can cover a wide frequency range. These clocks can be asynchronous (ie non-integer frequency ratio and/or unknown relative phase) or synchronous (ie their frequency ratio is fixed as well as their relative phase) to each other. Cross-domain paths must be correctly handled to prevent false timing violations caused by false paths while providing proper test of the interface logic. Within each domain, multi-cycle paths are sometimes used, instead of pipelining, to implement functions at a lesser cost. Gated clocks are used for low power designs. PLLs are used to control groups of synchronous clocks and/or multiply external reference clocks. Other aspects of the design affect the implementation of an at-speed test. The circuit is designed to perform at a certain temperature and supply voltage. The circuit activity has a direct influence on the local temperature and reduction of supply voltage (IR-drop) which in turn influence the timing. An at-speed structural test should be implemented such that the supply current, temperature and supply voltage drops can be controlled in a way that physical limits are not exceeded and, preferably, approach the conditions of the functional operation. An at-speed test method must be such that a minimum number of test-specific timing constraints are added to the design. These constraints must be easy to meet so that timing closure is not impacted.

II. Clocking scheme

The clocking scheme proposed in this paper addresses these issues. Clock bursts are applied to the circuit to reproduce characteristics of the functional mode of operation. The burst length is programmable. Burst lengths longer than 2 are needed to test multi-cycle paths [1] and deal with issues related to IR-drop. In [2], it was reported that the results of at-speed tests were different when using a burst of length of 2 as opposed to a free-running clock (ie “infinite” burst length). It is our experience as well that modifying the burst length will cause variations of the supply voltage and change the circuit timing. Some timing defects can only be identified if the burst length is sufficiently long to cause a supply variation that is comparable to the one observed during functional operation. The burst is modifiable at run-time to adjust the number of at-speed clock pulses applied.

Testability Analysis and Test-Point Insertion in RTL VHDL Specifications for Scan-Based BIST

Samir Boubezari, *Member, IEEE,* Eduard Cerny, *Senior Member, IEEE,* Bozena Kaminska, *Member, IEEE,* and Benoit Nadeau-Dostie, *Senior Member, IEEE*

***Abstract*— This paper proposes a new testability analysis and test-point insertion method at the register transfer level (RTL), assuming a full scan and a pseudorandom built-in self-test design environment. The method is based on analyzing the RTL synchronous specification in synthesizable very high speed integrated circuit hardware descriptive language (VHDL). A VHDL intermediate form representation is first obtained from the VHDL specification and then converted to a directed acyclic graph (DAG) that represents all data dependencies and flow of control in the VHDL specification. Testability measures (TM's) are computed on this graph. The considered TM's are controllability and observability for each bit of each signal/variable that is declared or may be implied in the VHDL specification. Internal signals of functional modules (FM's) such as adders and comparators are also analyzed to compute their controllability and observability values. The internal signals are obtained by decomposing at the RTL large FM's into smaller ones. The calculation of TM's is carried out at a functional level rather than the gate level, to reduce or eliminate errors introduced by ignoring reconvergent fanouts in the gate network, and to reduce the complexity of the DAG construction. Based on the controllability/observability values, test-point insertion is performed to improve the testability for each bit of each signal/variable. This insertion is carried out in the original VHDL specification and thus becomes a part of it unlike in other existing methods. This allows full application of RTL synthesis optimization on both the functional and the test logic concurrently within the designer constraints such as area and delay. A number of benchmark circuits were used to show the applicability and the effectiveness of our method in terms of the resulting testability, area, and delay.**

***Index Terms*—Built-in self-test, register transfer level, testability analysis, testability measures, test point insertion.**

I. Introduction

NOWADAYS, register transfer level (RTL) synthesis or logic synthesis has become an integral part of a design process of digital circuits. Most industrial digital designs use automated RTL synthesis and we can thus achieve design-for-testability (DFT) by incorporating test and synthesis into a single methodology that is as automated as possible. Indeed, considering testability during design synthesis can reduce the overall design and manufacturing time. Even more important, the testability enhancement at the entry level to a synthesis tool makes it independent of the tool and the implementation technology. It becomes part of the design specification and may be optimized with the other synthesis tasks in terms of area and delay.

The main objective of our method is thus to raise the level of abstraction at which testability analysis and test-point insertion are performed. We propose a new testability analysis and test-point insertion method at the RTL, assuming full scan and pseudorandom built-in self-test (BIST) design environment. Full scan in combination with pseudorandom patterns is widely adopted in the industry due to its ease of implementation and fault diagnostic. Unfortunately, the presence of random pattern resistant faults in many practical circuits poses a serious limitation to its success. The solutions to tackle this limitation can be broadly classified as those that modify the input patterns or those that modify the circuit-under-test. In this paper, we are interested in the second class of solutions, circuit modifications, that introduce test points to improve the random pattern testability of a circuit. Our goal is to analyze and modify the very high speed integrated circuit hardware descriptive language (VHDL) RTL description of the circuit, in order to generate an easily testable gate-level circuit by a pseudorandom sequence under the BIST environment. This is the main advantage and motivation of this work. That is, whatever the complexity of the circuit, our objective is to apply synthesis compilation and optimization technology directly to a testable VHDL description, thus optimizing functional and inserted test logic concurrently, rather than introducing testability after the VHDL has been compiled to gate-level.

The proposed method uses as the starting point a VHDL specification given at the synthesizable synchronous RTL. It is analyzed to produce an intermediate representation, called the VHDL intermediate format (VIF), and transformed into a DAG on which testability analysis is performed by computing and propagating testability measures (TM's) forward and backward through the VHDL statements. The TM's are the controllability and the observability for each bit of each signal/variable. Internal signals of functional modules (FM's) such as adders, comparators, and multiplexers are also analyzed to determine their controllability and observability values. The internal signals are obtained by decomposing at the RTL large FM's into smaller FM blocks, each such

Manuscript received November 19, 1997; revised November 17, 1998. This work was supported by LogicVision (Canada) and Micronet Centre of Excellence (Canada). This paper was recommended by Associate Editor J. Rajski.

S. Boubezari was with the Electrical Engineering Department, École Polytechnique de Montréal, PQ, Canada H3C 3A7. He is now with Synopsys, Inc., Mountain View, CA 94043-4033 USA (e-mail: samirb@synopsys.com).

E. Cerny is with the Département d'informatque et de recherche opérationnelle, Université de Montréal, Montréal, PQ, Canada H3C 3J7 (e-mail: cerny@iro.umontreal.ca).

B. Kaminska is with the Electrical Engineering Department, École Polytechnique de Montréal, PQ, Canada H3C 3A7 (e-mail: bozena@opmaxx.com).

B. Nadeau-Dostie is with LogicVision, Ottawa, Ont., Canada K1Z 8R3 (e-mail: benoit@lvision.com).

Publisher Item Identifier S 0278-0070(99)06631-2.

2012 IEEE 30th VLSI Test Symposium (VTS)

Test Generator with Preselected Toggling for Low Power Built-In Self-Test

Janusz Rajski, Jerzy Tyszer*, Grzegorz Mrugalski, Benoit Nadeau-Dostie
Mentor Graphics Corporation
Wilsonville, OR 97070, USA
*Poznań University of Technology
60-965 Poznań, Poland

Abstract – **This paper presents a new pseudorandom test pattern generator with preselected toggling (PRESTO) activity. It is comprised of a linear finite state machine (a linear feedback shift register or a ring generator) driving an appropriate phase shifter and armed with a number of features that allows this device to produce binary sequences with low toggling (switching) rates while preserving test coverage achievable by the best-to-date conventional BIST-based PRPGs with negligible impact on test application time.**

I. Introduction

Various forms of embedded test are increasingly viewed as essential to reduce test cost. Among them, scan testing has gained broad acceptance as a reliable solution. However, due to the high data activity associated with scan-based test operations, a circuit under test can dissipate much more power than it was designed to function under. With overstressing devices beyond the mission mode, reductions in the operating power of ICs in a test mode have been of concern for years. A full-toggle scan pattern may draw several times the typical functional mode power, and this trend continues to rise, particularly over the mission mode's peak power. This power-induced over-test may result in thermal issues, voltage noise, power droop, or excessive peak power over multiple cycles which, in turn, cause a yield loss due to instant device damage, severe decrease in chip reliability, shorter product lifetime, or a device malfunction because of timing failures following a significant circuit delay increase, for example. Abnormal switching activity may also cause fully functional chips to fail during testing because of phenomena such as IR-drop, crosstalk, or di/dt problem.

Numerous schemes for power reduction during scan testing have been devised [8]. Among them there are solutions specifically proposed for built-in self-test (BIST) to keep the average and peak power below a given threshold. For example, the test power can be reduced by preventing transitions at memory elements from propagating to combinational logic during scan shift. This is achieved by inserting gating logic between scan cell outputs and logic they drive [4], [9]. During normal operations and capture, this logic remains transparent. Gated scan cells are also proposed in [2] and [20]. A synergistic test power reduction method of [21] uses available on-chip clock gating circuitry to selectively block scan chains while employing test scheduling and planning to further decrease BIST power in the Cell processor. A test vector inhibiting scheme of [5] masks test patterns generated by an LFSR as not all produced vectors, often very lengthy, detect faults. Elimination of such tests can reduce switching activity with no impact on fault coverage.

The advent of low-transition test pattern generators has added a new dimension to power aware BIST solutions [3], [11], [15]. A device presented in [19] is comprised of an LFSR feeding scan chains through biasing logic and T-type flip-flop. Since this flip-flop holds the previous value until its input is asserted, the same value is repeatedly scanned into scan chains until the value at the output of biasing logic (e.g., a k-input AND gate) becomes 1. Depending on k, one can significantly reduce the number of transitions occurring at the scan chain inputs. A dual-speed LFSR of [18] consists of two LFSRs driven by normal and slow clocks, respectively. The switching activity is reduced at the circuit inputs connected to the slow-speed LFSR, while the whole scheme still ensures satisfactory fault coverage. Mask patterns are used in [14] to mitigate the switching activity in LFSR-produced patterns, whereas a bit swapping of [1] achieves the same goal at the primary inputs of CUT. A gated LFSR clock of [6] allows one to activate only half of LFSR stages at a time, thus reducing power consumption, as only half of the circuit inputs change every cycle. A scheme that combines the low transition generator of [19] (handling easy-to-detect faults) with a 3-weight PRPG (deployed to detect random pattern resistant faults) can also be used to reduce switching activity during BIST, as demonstrated in [17]. The schemes of [10], [13], and [16] suppress transitions in LFSR-generated sequences by either statistical monitoring or injecting intermediate and highly correlated patterns. Finally, a random single-input change generator can produce low power patterns in a parallel BIST environment, as shown in [7].

As the BIST power consumption can easily exceed the maximum ratings when testing at speed, scan patterns must be shifted at a programmable low speed, and only the last few cycles and the capture cycle are applied at the maximum frequency. In the burst-mode approach presented in [12], typically five consecutive clock cycles are used. The first four cycles serve shifting purposes, whereas the last one is designated for capture. The objective is to stabilize the power supply before the last shift and capture pulses are applied, which are critical for at-speed tests. To reduce the voltage droop related to a higher circuit activity, a burst clock controller slows down some of the shift cycles. It allows a gradual increase of the circuit activity, thereby reducing the di/dt effect. The controller can gate the shift clocks, depending on the needs for gradual warming up of the circuit.

In this paper, we propose a new pseudorandom test pattern generator (PRPG) for low power BIST applications. The generator is aimed at reducing the switching activity during scan loading due to its preselected toggling (PRESTO) levels. It can assume a variety of configurations that allow a

978-1-4673-1074-1/12/$31.00 ©2012 IEEE

Test Point Insertion Using Functional Flip-Flops to Drive Control Points

Joon-Sung Yang[1], Benoit Nadeau-Dostie[2], and Nur A. Touba[1]

[1]Computer Engineering Research Center
Dept. of Electrical and Computer Engineering
University of Texas, Austin, TX 78712
{jsyang,touba}@ece.utexas.edu

[2]Logic Vision, Inc.
25 Metro Drive, Third Floor
San Jose, CA 95110
benoit@logicvision.com

Abstract

This paper presents a novel method for reducing the area overhead introduced by test point insertion. Test point locations are calculated as usual using a commercial tool. However, the proposed method uses functional flip-flops to drive control test points instead of test-dedicated flip-flops. Logic cone analysis that considers the distance and path inversion parity from candidate functional flip-flops to each control point is used to select an appropriate functional flip-flop to drive the control point which avoids adding additional timing constraints. Reconvergence is also checked to avoid degrading the testability. Experimental results indicate that the proposed method significantly reduces test point area overhead and achieves essentially the same fault coverage as the implementations using dedicated flip-flops driving the control points.

1. Introduction

Built-in self-test (BIST) involves the use of on-chip test pattern generation and output response analysis. BIST provides a number of important advantages including the ability to apply a large number of test patterns in a short period of time, high coverage of non-modeled faults, minimal tester storage requirements, can apply tests out in the field over the lifetime of the part, and a reusable test solution for embedded cores. The most efficient logic BIST techniques are based on pseudo-random pattern testing. A major challenge is the presence of *random-pattern-resistant (r.p.r.) faults* which have low detection probabilities and hence may limit the fault coverage that can be achieved with pseudo-random patterns. There are two approaches for detecting r.p.r. faults: either modify the pattern generator so that it generates patterns that detect them, or modify the circuit-under-test to eliminate the r.p.r. faults.

A number of techniques have been developed for modifying the pattern generator. These include weighted pattern testing [Schnurmann 75], [Wunderlich 87], [Pomeranz 92], [Bershteyn 93], [Kapur 94], [Jas 01], [Lai 05], pattern mapping [Chatterjee 95], [Touba 95a, 95b], bit-fixing [Touba 96], bit-flipping [Wunderlich 96], and LFSR reseeding [Konemann 91, 01], [Hellebrand 92, 95], [Krishna 01], [Rajski 02].

The other approach is to modify the circuit-under-test (CUT) by inserting test points [Eichelberger 83]. Test points are very efficient for eliminating r.p.r. faults and improving the fault coverage. Test point insertion (TPI) involves adding control and observation points to the CUT. Observation points involve making a node observable by making it a primary output or sampling it in a scan cell. Control points involve ANDing or ORing a node with an activation signal as illustrated in Fig. 1. When the activation signal is a '1', it controls the node to a 0 (1) for a control-0 (control-1) point. Typically the activation signal is driven by a dedicated flip-flop which receives pseudo-random values during BIST and is set to a non-controlling value during normal operation. Test points are added to the circuit before layout so that the performance impact can be minimized. Circuit re-structuring is routinely used during layout to take into account additional delay due to metal wires.

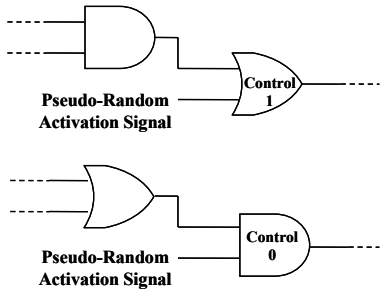


Figure 1. Example of Control Points

Since test points add area and performance overhead, an important issue for test point insertion is where to place the test points in the circuit to maximize the coverage and minimize the number of test points. Optimal placement of test points in circuits with reconvergent fanout has been shown to be NP-complete [Krishnamurthy 87]. A number of approximate techniques for placement of test points have been

978-1-4244-4867-8/09/$25.00 ©2009 IEEE

Test Point Insertion with Control Points Driven by Existing Functional Flip-Flops

Joon-Sung Yang, *Member*, *IEEE*, Nur A. Touba, *Fellow*, *IEEE*, and Benoit Nadeau-Dostie, *Senior Member*, *IEEE*

Abstract—This paper presents a novel test point insertion method for pseudorandom built-in self-test (BIST) to reduce the area overhead. The proposed method replaces dedicated flip-flops for driving control points by existing functional flip-flops. For each control point, candidate functional flip-flops are identified by using logic cone analysis that investigates the path inversion parity, logical distance, and reconvergence from each control point. Four types of new control point structures are introduced based on the logic cone analysis results to avoid degrading the testability. Experimental results indicate that the proposed method significantly reduces test point area overhead by replacing the dedicated flip-flops and achieves essentially the same fault coverage as conventional test point implementations using dedicated flip-flops driving the control points.

Index Terms—Dedicated flip-flop, functional flip-flop, logic cone analysis, test point insertion.

1 Introduction

Test cost for complex designs has increased significantly. The amount of test data volume required is growing rapidly. Testers have limited I/O channels and speed, and hence pose a major bottleneck for conventional external testing.

Built-in self-test (BIST) helps to reduce test data bandwidth requirements and test storage requirements by orders of magnitude [1], [13]. It involves the use of on-chip test pattern generation and output response analysis. BIST provides a number of important advantages including the ability to apply a large number of test patterns in a short period of time, minimal tester storage requirements, at-speed testing, application of tests out in the field over the lifetime of the part, and a reusable test solution for embedded cores. The most economical logic BIST techniques are based on pseudorandom pattern testing. One of the attractions to pseudorandom pattern testing is the simple logic structures as a part of circuit under test (CUT) which provide the input stimuli and the circuit response compression. This allows significant compaction of test data. Pseudorandom pattern testing also can achieve high coverage of nonmodeled faults which are not explicitly targeted during deterministic test generation. However, a major challenge is the presence of random-pattern-resistant (r.p.r.) faults which have low detection probabilities and hence may limit the fault coverage that can be achieved with pseudorandom patterns. Various methods have been proposed to overcome the problem. One approach is to modify the pattern generator using methods such as weighted pattern testing [2], [13], [14], [19], [23], [25], pattern mapping [4], [30], [31], bit-fixing [32], bit-flipping [34], and LFSR reseeding [8], [9], [15], [16], [17], [24].

The other approach to make the CUT random pattern testable is to modify it by inserting test points [6]. Test point insertion (TPI) involves adding control and observation points to the CUT. Observation points make a node observable by adding an extra primary output or sampling it in a scan cell. Control points involve ANDing or ORing a node with an activation signal where the activation signal is driven by a dedicated flip-flop which receives pseudorandom values during BIST and is set to a noncontrolling value during normal operation. Additional hardware is needed to form the test points which adds area and performance overhead to a design. Since optimal test point placement is NP-complete [18], a number of TPI methods have been proposed using fault simulation [3], [12] and testability measures [26]. Two general strategies for TPI have been widely studied to overcome the overhead issues: 1) TPI for minimizing performance overhead, and 2) TPI for minimizing area overhead.

To minimize the performance overhead for TPI, Cheng and Lin [5] and Tsai et al. [33] proposed timing driven test point insertion techniques which avoid TPI on critical timing paths. They showed that by avoiding control point insertion on critical timing paths, high fault coverage can be achieved without performance degradation. Reducing the number of test points to minimize the area overhead, TPI techniques like path tracing [32] and multiphase TPI [29] were introduced. Tamarapalli and Rajski [29] partition the entire test into multiple phases by divide and conquer method and control points are activated only during certain phases and deactivated during other phases. This provides greater control over the interaction of the control points with each other which can help reduce the total number of test points required. Nakao et al. [21] and Youssef et al. [37] propose methods for having

- *J.-S. Yang is with SungKyunKwan University, Suwon-Si, Gyeonggi-Do 440-746, Korea. E-mail:js.yang@skku.edu.*
- *N.A. Touba is with the University of Texas at Austin, 1 University Station, #C0803, Austin, TX 78712-1084. E-mail: touba@ece.utexas.edu.*
- *B. Nadeau-Dostie is with Mentor Graphics, 8005 Southwest Boeckman Road, Wilsonville, OR 97070. E-mail: Benoit_Nadeau-Dostie@mentor.com.*

Manuscript received 22 Oct. 2010; revised 8 May 2011; accepted 10 Aug. 2011; published online 30 Sept. 2011.
Recommended for acceptance by C. Metra.
For information on obtaining reprints of this article, please send e-mail to: tc@computer.org, and reference IEEECS Log Number TC-2010-10-0582.
Digital Object Identifier no. 10.1109/TC.2011.189.

Transitioning eMRAM from Pilot Project to Volume Production

Cyrille Dray[1], Khushal Gelda[2]
Benoit Nadeau-Dostie[3], Wei Zou[4], Luc Romain[3], Jongsin Yun[4], Harshitha Kodali[4], Lori Schramm[5], Martin Keim[6]

[1]Arm Sophia Antipolis, France; [2]Bengaluru, Karnataka, India
Siemens [3]Ottawa, Canada; [4]Wilsonville, USA; [5]Atlanta, USA; [6]Orlando, USA

***Abstract*—Embedded non-volatile RAM technology, and in particular Magneto-resistive RAM (MRAM), continues making great progress in read/write speed and cycling endurance, now rivaling traditional memory technologies. This advancement together with their low-energy consumption and non-volatility opens huge application markets. While stand-alone MRAM products are already being deployed, the first pilot applications for embedded MRAM are just starting to emerge. The availability of Design-for-Test (DFT) tools is key in the transition from low-volume technology exploration and pilot projects to high-volume production. In this paper we explore a new aspect of a Memory Built-In Self-Test (MBIST) tool for eMRAM, namely Error Correcting Code (ECC)-aware test and repair technology. This new MBIST capability provides the ability to make in-system, user-programmable trade-offs concerning the eMRAM repair resources. Having extra control over the repair resources in turn increases both manufacturing yield as well as the longevity of the product in its application.**

Keywords—eMRAM, trimming, ECC, ECC tuning, ECC-aware test, memory fault injection, memory test, memory repair, memory yield, memory aging, DFT, MBIST

I. Introduction

The rapidly growing IoT market creates a huge demand for embedded non-volatile memories. For example, IoT devices [1], with AI technology [2], require high density and high performance, low power memory operation. However, scaling traditional eFlash memories has limitations going below 28nm due to its high cost. The industry offers several alternative embedded non-volatile memories such as MRAM, CBRAM (Conductive Bridge Random-Access Memory), FeRAM (Ferroelectric Random-Access Memory), ReRAM (Resistive Random-Access Memory), etc. Recent MRAM technology announcements by Samsung [3] and IBM [4] show very promising results. MRAMs manufactured at 14nm FinFET technology sustained a sub 100ns write speed with 1E14 cycles of endurance. This is very impressive compared to eFlash technology which offers micro-second to milli-second write speed with 1E5 cycling endurance [5]. It also promises downscaling capability to fill the embedded NVM market needs beyond the 28nm node. This type of high speed, high endurance, and high-density MRAM is expected to replace some of the cache memories in the near future.

In this Industrial Short Paper, we describe a new MBIST solution for eMRAM. This MBIST solution provides the same fully integrated implementation environment to build MBIST DFT logic for both SRAMs as well as MRAMs. Furthermore, much of a standard solution for SRAM transfers directly, so the same MBIST methodology can be used for testing both SRAM and MRAM. Accordingly, Section II introduces first the standard MBIST test and repair flow for SRAM. In Section III we will develop this flow into a production-ready test and repair flow for eMRAMs highlighting only the differences. For completeness, this section also summarizes eMRAM specific fault models and the necessary trimming used in the experimental setup outlined in Section IV. The focus in Section IV is however on "ECC-Aware Test and Repair". This combines the power of the ECC, which every emerging NVM requires, with the power of repair for the purpose of improving yield and durability of the memory in the product application. The data presented in Section IV clearly demonstrates the applicability of the ECC-aware test and repair technology to eMRAMs. Lastly, Section V concludes the paper.

II. Standard Memory BIST Overview

Figure 1 illustrates a widely used standard MBIST test and repair manufacturing flow. The same manufacturing flow can be used for SRAMs implementing redundancy (repair) or ECC logic. ECC is not typically used for SRAMs during manufacturing test, but it can be used during in-system testing to ensure tolerance of soft errors.

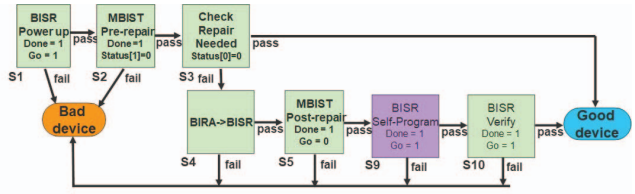


Fig. 1. MBIST test and repair manufacturing flow

The Built-in Self Repair (BISR) power-up step (S1) of the flow emulates a power-up event. The BISR chain is loaded with a repair solution previously calculated and stored in a fuse box. All spare resources of repairable memories are flagged as unused when executing the flow for the first time. The MBIST pre-repair step (S2) calculates a new repair solution based on the previous one. It tests all memories in the design and

2023 IEEE International Test Conference (ITC) | 979-8-3503-4325-0/23/$31.00 ©2023 IEEE | DOI: 10.1109/ITC51656.2023.00021

2378-2250/23/$31.00 ©2023 IEEE
DOI 10.1109/ITC51656.2023.00021