# Test of Future System-on-Chips

Yervant Zorian[1], Sujit Dey[2, +], Michael J. Rodgers[3]

[1] LogicVision Inc., San Jose, CA
[2] Dept. of ECE, University of California at San Diego, La Jolla, CA
[3] Test Technology, Microprocessor Products Group, Intel Corp., Santa Clara, CA

zorian@lvision.com, dey@ece.ucsd.edu, michael.j.rodgers@intel.com

## Abstract

*Spurred by technology leading to the availability of millions of gates per chip, system-level integration is evolving as a new paradigm, allowing entire systems to be built on a single chip. Being able to rapidly develop, manufacture, test, debug and verify complex SOCs is crucial for the continued success of the electronics industry. This growth is expected to continue full force at least for the next decade, while making possible the production of multimillion transistor chips. However, to make its production practical and cost effective, the industry road maps identify a number of major hurdles to be overcome. The key hurdle is related to test and diagnosis. This embedded tutorial analyzes these hurdles, relates them to the advancements in semiconductor technology and presents potential solutions to address them. These solutions are meant to ensure that test and diagnosis contribute to the overall growth of the SOC industry and do not slow it down. This embedded tutorial in addition presents the state-of-the-art in system-level integration and addresses the strategies and current industrial practices in the test of system-on-chip. It discusses the requirements for test reuse in hierarchical design, such as embedded test strategies for individual cores, test access mechanisms, optimizing test resource partitioning, and embedded test management and integration at the System-on-Chip level. Processor cores being one of the most common cores embedded in a SOC, issues related to self-testing embedded processor cores are addressed. Future research challenges and opportunities are discussed in enabling testing of future SOCs which use deep submicron technologies.*

## 1. Introduction: Industrial Trends and Challenges

Two key semiconductor supplier cost challenges are changing the way VLSI ICs are tested today. One is that the cost of manufacturing test has not been scaling. Secondly, the engineering effort to generate tests has been growing geometrically along with product complexity. A general rule of thumb is that capital costs run in the range of 50% of the overall IC test cost in the industry, so looking at capital costs is an essential analysis for manufacturing test.

Figure 1 shows a plot extrapolated from the 1997 SIA technology roadmap for semiconductors [1]. It shows the capital costs for chip fabrication versus the capital costs for manufacturing test, normalized per transistor. The top curve shows the consistent reduction in chip fabrication cost per transistor that is the basis for Moore's law, which in turn drives the continued expansion and evolution of the semiconductor business. The bottom curve, which can be traced back 20 years, indicates capital expenses for IC test have been essentially flat per transistor. Based on the 1997 SIA data, this trend of flat test capital cost per transistor was projected to continue for the foreseeable future.

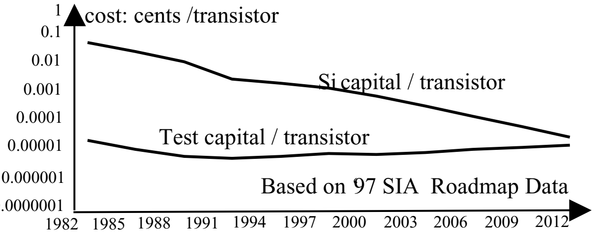

Figure 1. Moore's Law for Test: Fab vs. Test Capital

Has IC test really been standing still the last 20 years? Not at all. In fact, the capabilities, methodologies, and technologies of DFT and manufacturing test have continued to move steadily forward, from the curve tracers used on the original discreet commercial semiconductors to the multi-million dollar ATEs used at the high end of the VLSI testing today. Rather, the historical test capital per transistor trend indicates we have used a tremendous amount of effort and technology in test to keep up with the continued increases in IC device performance and complexity. If not addressed, the data in Figure 1 would project the industry could reach a point in several years where the general cost of testing ICs exceed the cost of fabricating them.

As an industry, the semiconductor suppliers have recognized the business models for IC manufacturing test must change to support the continued evolution of the semiconductor business. They have already moved to increase the use and effectiveness of more comprehensive embedded test methods to enable greater re-use of older ATEs and new simpler ATEs for the future for manufacturing test. The trend is to invest a larger share of the additional transistors enabled by Moore's law for improved DFT and test manufacturability rather than to continue to add hardware and features to ATE used for high volume manufacturing.

The test capital per transistor extrapolation from the more recent 1999 ITRS roadmap now shows a downward trend moving forward [2]. This shows a commitment and belief on the part of semiconductor manufacturers that new business models for IC testing will be successful based on DFT, reduced ATE capital expenditures, and increasing use of BIST.

Increasing integration is occurring across many IC product lines today. In addition to true system-on-a-chip ICs, everything from mixed signal to DSPs, CPUs and non-volatile memories are being integrated with each other and themselves (multi-core DSPs,

+ This work was partially supported by SRC under contract #98-TJ-648, and the DARPA/MARCO Gigascale Silicon Research Center.

CPUs) as the number of transistors per die moves from 10s to 100s of millions.

While the cost of silicon real estate in a high integration die or SOC is often nicely predicted by die size and defect density, the effort and tools needed to “cut and paste” different design blocks and types together is often underestimated. More often, the effort and cost for design validation and manufacturing test of re-used design blocks and high integration ICs is much greater than the sum of those for the original designs blocks.

## 1.1 SOC Industrial Test Challenges

### *EDA Tools and Design Methods*

In many cases, the integrated or SOC product extends beyond the available technology envelope used to design and manufacture ICs otherwise. For test, these envelopes could be the database size and raw performance limitations for the design, validation, and test EDA tools. Today, many EDA tools are unable to handle full chip databases for some levels of abstraction on some of the largest stand alone VLSI devices. The SOCs and high integration devices across the industry's roadmaps today will stretch that challenge even further.

Cores for re-use in SOCs and integrated ICs must have embedded test solutions or be connectable to SOC global test accesses and DFT schemes. Neglecting to plan and build the embedded test and DFT scheme for any given core re-used in a larger design could sink the new SOC device's ability to hit engineering and manufacturing cost expectations, as well as time to market expectations.

### *ATEs and Manufacturing Test Methods*

SOCs also challenge the capital basis for test on the historical model of bolting more combinations of hardware on the ATEs themselves as the increases in device complexity and in the variety cores used in new product integrations continues. Totally embedded RAM, mixed signal, or logic cores and buses can be much more expensive to test in terms of test generation effort, test database size, test time and other key parameters, if not tied into effective chip global schemes.

For example, a combination of fully scannable logic blocks or IP cores could be amenable to scan stitching and scan pattern compression techniques and still not fit on a given ATE in the integrated or SOC product.

This points to an evolving trend first seen in integrated SRAMs, which have been used for many years in high end logic devices. Across the industry, most of these are tested with self-test hardware built into chip design and as these RAMs move from Kilobit to Megabit sizes, these self test schemes are increasing in usage and in capabilities.

Moving forward, SOC and high integration devices will further drive chip design to increase the use of BIST. BIST has been around for many years, but SOCs will drive their use in the industry to a much greater extent than seen previously. BIST use will increase for a larger variety of cores, from I/Os, mixed signal, PLLs, logic, and RAM cores rather than adding more hardware or more performance or more memory to the ATEs at the same rates we have in the past.

A beneficial business re-use of additional BIST hardware will be to make it available for application and end user diagnostics, at the SOC level. It can then be re-used by the physical platform and system. This is useful both to the system OEM as well as to the end use customer, such as in high reliability market segments (e.g. industrial control, server networks, and internet backbone applications).

### *Power Management and Delivery*

Even if better integrated BIST enables SOCs and high integration ICs to stay within new test business models, device power of SOCs under test could drive technology extensions for power delivery at manufacturing test. Device power, power delivery in the end use platform and for applied test face major challenges and redefinition of the critical metrics, where di/dt is as important as Pavg and Pmax. 500mA/nS was viewed as a di/dt test power delivery wall only a couple of years ago, but we now project devices capable of tens of amps per nS of di/dt (Figure 2). New configurations of applied test power delivery could likely include, for example moving power supplies from the ATEs to DUT interface hardware and boards, closing the electrical and physical distance to the DUT and allowing more flexibility for the manufacturing test solutions and trade offs.

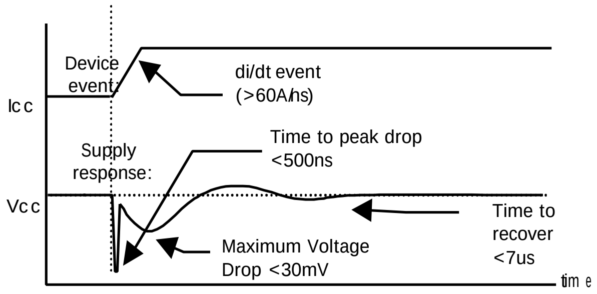

Figure 2. Test Power Delivery Challenge (Vcc at die)

Power delivery at the platform level is moving to materials and schemes for lowering intrinsic R and L values, increased decoupling C, and local power regulation. For applied component test, the unique constraints of these environments often drive more stringent power requirements than the device “spec” or end use. These boundaries will drive new requirements, methods, and tools into VLSI design, i.e., design-for-power-delivery, design-for-di/dt, and global power management schemes on chip for SOCs and highly integrated ICs.

We've used power supply scaling over the last few years to manage overall device power, as we broke the 5V barrier and have steadily reduced Vcc lower and lower. However, that trend will not continue much below 1.0V. As design-for-power schemes involve chip globals and intrinsic properties from clock distribution to synthesis to standard cell design, it will require us to develop, modify, and integrate our logic DFT solutions to the device design and power solutions at several levels of abstraction.

### *Summing Up SOC Industrial Test Issues*

IC testing is already undergoing a major change today to reduce engineering and capital costs moving forward through more systematic use of DFT to enable use of cheaper ATEs for high volume manufacturing. At the same time, high integration devices and SOCs are increasing across IC product lines today. These SOCs challenge the industry to accelerate the changes to IC testing, requiring more extensive use of BIST and more disciplined DFT inserted into all design cores re-used in SOCs and highly integrated ICs. The companies (IC suppliers, EDA, and ATE) that figure out how to most effectively develop processes to “cut and paste” new integrations and SOCs as quickly as possible with the least effort and still hit time-to market and manufacturing test expectations will be the ones that are most successful moving forward.

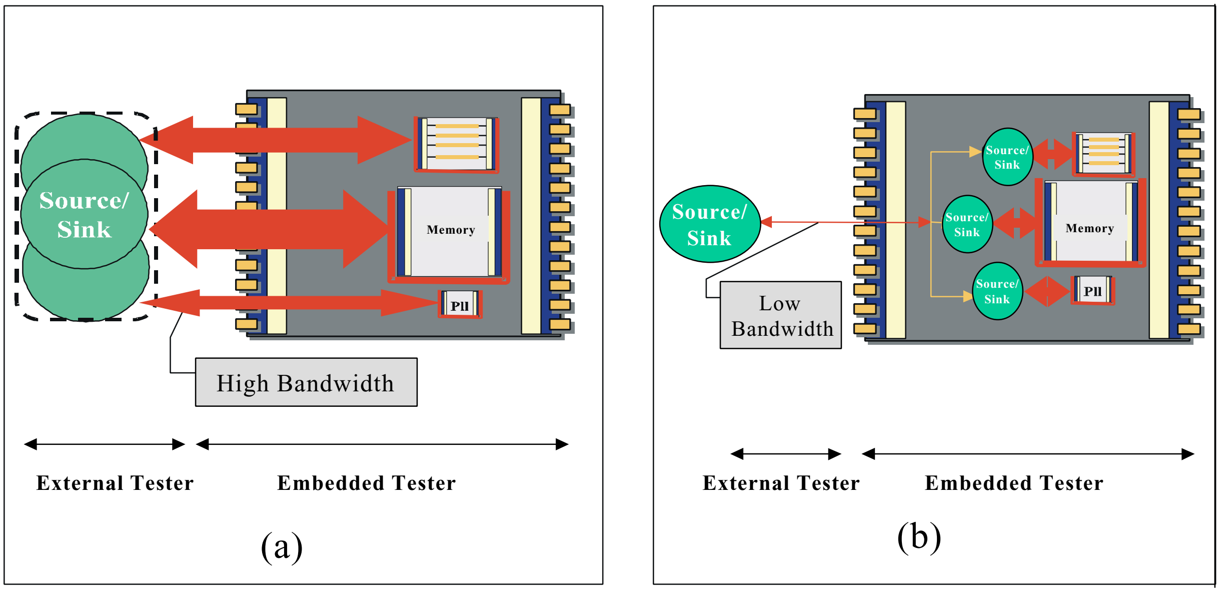

Figure 3. External and Embedded Tester Partitioning: (a) Conventional Partitioning, (b) Partitioning with Embedded Sources/Sinks

## 2. Embedded Test and Test Resource Partitioning

Embedded test is an advanced solution, which addresses the above listed trends and challenges for testing future SOCs [3, 4, 5, 7]. Embedded test is comprised of two distinct test approaches: External ATE and conventional DFT. Building on conventional DFT approaches such as scan and BIST, embedded test integrates high–speed and high-bandwidth portions of the external ATE directly into the ICs. This integration facilitates the chip, board and system level test, diagnosis, debug and repair. Embedded test is typically implemented in two components: user configurable test IP (intellectual property) in the form of design objects delivered as Register Transfer Level (RTL) soft cores; and a suite of test automation tools to automate generating, integrating, analyzing, and verifying test and diagnostic patterns.

Embedded test integrates multiple disciplines: DFT features; BIST pattern sources and sinks; precision and high speed timing for at-speed test; test support for many different core types (logic, memory, processors, and analog); and capabilities for diagnosis and debug.

The integration of embedded test circuitry into the SOC design results in a new on-chip and off-chip distribution of test resources compared to the conventional test resource partitioning, as shown in Figure 3. With embedded test, the on-chip test data generation reduces the volume of external patterns and can be customized per core type. Also, the on-chip test and diagnostic data compression reduces ATE data logging requirements. Moreover, the on-chip timing generation achieves true at-speed test that can scale to match process performance.

Embedded test provides an effective solution for partitioning and optimal allocation of test resources that are allocated externally or are embedded in hardware and/or software forms. This solution provides several key advantages, such as: cost-effective at-speed test of random logic, memory, and analog cores; diagnostics and debug for random logic, embedded memories, analog and legacy cores on-chip; test and debug for I/Os, interconnects and external memory modules on-chip; redundancy and repair capability to enhance the yield of the SOCs under production [3, 4, 5].

## 3. Hierarchical Embedded Test

As discussed earlier, today's chips have already started to mix diverse circuits, such as random logic, embedded memories and analog cores into a single SOC. As chip integration continues, more advanced circuits will be added to this list in future SOCs, such as embedded FPGA, Flash memories, RF/Microwave, and may even move beyond the electronics domain to contain micro-electromechanical (MEMS) and optical elements.

An easier and more cost effective way to handle these mixed circuit chips is by inserting embedded test IP (hardware Sources and Sinks corresponding to each circuit type), for example an embedded Source/Sink for random logic, another for memories and a third for the analog circuit. Such an SOC will not require more than a single, existing and lower-cost external ATE, as shown in Figure 3 (b). Although an embedded test IP requires more silicon area, the savings realized through automation of the testing process, higher quality of testing , and reuse at all levels in the design hierarchy (core to chip to board to system) makes this method very attractive. The appropriate IP types used with each type of core are typically accessed through the five-pin IEEE-standard 1149.1 TAP [6].

Embedded core-based system-on-chip (SOC) design implies the reuse of pre-designed complex functional cores, also called Virtual Components [7]. These embedded cores can come with different degrees of readiness for reuse in system level design, from different sources, and are designed for use in a multiplicity of different SOCs. Being pre-designed, an embedded core may not only originate in a different organization, but it is also developed at a different time than the SOC that will use it. The embedded core design must be able to anticipate the desired SOC-level test constraints for all target SOC designs. Further, it must be possible to package the results of any enabled core-test in a form that is compatible with the test methodology contexts, and with the test-development tools available to the SOC designers who wish to reuse the core.

Core designs need to be more test-friendly to simplify the SOC integration task, while giving SOC designers more flexibility in choosing the best overall test methodologies for their chips. To ensure the test-friendliness and interoperability of cores from diverse sources, a standard for embedded core test in under development, namely IEEE P1500 [8]. The standard does not standardize a core's internal test methods or chip-level test access configuration. It rather concentrates on [9]:

- ❑ a standardized core test language (CTL), capable of expressing all test-related information to be transferred from core provider to core user; and
- ❑ a standardized, but configurable and scalable, core test wrapper, which allows easy test access of the core in a SOC design. The standard core test wrapper interfaces with an on-chip test access mechanism and may operate under several test modes (internal, external, diagnosis, etc).

While it is possible to route the test access mechanism to the I/Os of the chips in order receive/transmit the test patterns from/to external test equipment, but it more practical and cost effective to use on chip test Sources and Sinks. They may be realized in two scenarios, either the embedded core would have a dedicated Source and Sink to perform its self-test; the test access mechanism connected to this core may obtain connect to a Source and Sink at the SOC or any other intermediate level. This is mainly meant for reusing the Source and Sink for more than one embedded core.

The most used cores today are the embedded memories. These cores have widely accepted the embedded Source and Sink approach. Most chip manufacturers have adopted memory BIST generation tools. As the monster chips incorporate more complex and larger numbers of embedded cores, such as microprocessors, analog cores, and DSPs, the embedded Source and Sink approach need to be extended as the test solution of the other cores in an SOC [10, 11, 12].

## 3.1 Random Logic Embedded Test

Typically, random logic embedded test is based on the extension of a scan circuit into a self-test version. This self-test is performed using a pseudo-random pattern source as stimuli generator and a multiple input signature register (MISR) for output results compression. The random logic BIST IP must be capable of operating at full application speed. In addition to the RTL design objects for the random logic BIST IP, this BIST capability also provides RTL design objects for the IEEE 1149.1 TAP and boundary-scan cells, a scan chain interface, a clock prescaler and testpoints to implement a more random pattern sensitive scan design.

An important need for logic BIST is its capacity to test a complete chip with clock domains running at different frequencies, testing at speed all clock domains and the interfaces between each of these domains. This also reduces test time because all portions of the chip are tested simultaneously. Often during logic BIST, the output drivers are disabled by default to minimize power consumption during BIST and to avoid bus contentions or other hazards at the board and system level.

In addition to at-speed logic BIST, several levels of diagnostic are often available, for instance, a multiple external scan chain mode or a single scan through TAP mode. Signatures, seeds and all other registers of the BIST circuit can be loaded and inspected through a serial interface for diagnostic purposes.

## 3.2 Memory Embedded Test

Future SOCs are expected to embed very dense memories of large sizes (256M bits). These dense memories may include: SRAMs, DRAMs and/or Flash memories. For more than a decade, the smaller scale memories have been embedded in mostly logic chips and became an integral part of the ASIC libraries. These memories were among the first to use BIST (on-chip Sources and Sinks). This is utilized during the manufacturing test to avoid using a dedicated external memory tester, in addition to the external logic tester used for the rest of ASIC. Beyond a certain size such as 256K bits, memories necessitate redundancy and repair during manufacturing test. This has been performed regularly for large stand-alone memory. This is typically a fuse blow process using external laser repair equipment.

Due to the large sizes of its embedded memories, an SOC needs to have redundant rows and columns to help reconfigure it, if there were faulty cells detected. For the same reasons as for the smaller memories, these will rely on embedded Sources and Sinks to generate and evaluate the test data. Moreover, since the memory response data is evaluated by the embedded Sink, the role of this Sink could be slightly expanded in order to perform diagnosis of the failed bits. Furthermore, to avoid sending a large failed bit map to the ATE via limited I/O bandwidth, the embedded Sink can be expanded further to perform built-in redundancy analysis in order to identify the actual rows and columns needed for reconfiguration. In this case, only the repair list can be communicated to the external tester and hence the laser repair equipment can perform a hard repair.

The final augmentation of the embedded memory Source and Sink is to make the memory self-repairable. This is motivated by the fact that laser repair is often very expensive and some times continuous periodic field repair is desired. This will be achieved by expanding the embedded test resources even further to include a storage repair data and a soft reconfiguration mechanism. In summary, embedded test for very large memories may by required to move beyond fault detection to include failed bit diagnosis, redundancy analysis and self-repair.

## 3.3 Analog Embedded Test

Embedded analog cores may be tested with a similar embedded test approach to random logic and memories. The analog embedded test automatically generates sythesizable RTL code, synthesis scripts and verification and test patterns. Analog BIST allows at-speed testing of analog cores using a standard digital ATE. For instance in PLL BIST, the RTL design objects connect to only the inputs and outputs of the PLL to test. No changes or connections to the internal nodes of the PLL are necessary. A digital multiplexer drives the input of the PLL. Testing the PLL is fully synchronous, making it suitable for very high speed tests. PLL BIST measures the loop gain, frequency lock range, lock time, and jitter, one parameter at a time. This is a characteristic of the advanced analog embedded test techniques.

## 3.4 Beyond Diagnosis

In addition to fault localization and failure analysis, producing the SOC requires integrated yield analysis capabilities that make use of the defect and failure analysis data. These capabilities need be in software tools to automatically access multiple databases and establish correlation between data of different types. Some data sources are time-based, others are chip-based or wafer-based. Automated data reduction algorithms to source defects from multiple data sources must be developed to reduce defect sourcing time. The ITRS roadmap identified this as one of the key requirements for yield learning and improvement [2].

# 4. Self-Testing of Embedded Processor Cores

Since processor cores are the most common cores embedded in SOCs, we next address the issue of testing embedded processor cores. For future GHz SOCs, it will be critical to test embedded processor cores at-speed. Hence, a growing need for self-testing of embedded processor cores. By generating the required test patterns on-chip and applying the tests at the speed of the circuit, a GHz processor core can test itself without relying on high-speed, prohibitively expensive external testers.

One option for self-testing a processor core is to use existing BIST [13] techniques. While embedded memory components in processors widely use memory BIST techniques, the non-memory (logic) parts of processors have not yet seen much use of logic BIST techniques. In this section, we first review some of the experiences reported in applying logic BIST techniques to experimental as well as commercial processor cores. We summarize the possible problems to be overcome in applying logic BIST to self-test processor cores. Next, we describe recent progress made in an alternative set of methods of testing processor cores – using the instruction set of the processor to compose random as well as deterministic self-test programs to test the processor core for manufacturing defects.

## 4.1 Applying Logic BIST to Processor Cores

Recently, several attempts have been reported at applying hardware-based logic BIST techniques to processor cores to make them self-testable [14, 15]. Through these experiences, we point out key problems encountered, and solutions used, in applying logic BIST tools to processor cores.

Most of the current hardware-based logic BIST techniques are based on the application of pseudo random test patterns generated by on-chip test pattern generators like LFSRs. Processors, due to their complex control structures, are highly random-pattern-resistant. Acceptable fault coverage cannot be achieved by simply applying random test patterns to the entire processor, as certain internal control signals need to be set properly to ensure the free flow of test data. Since logic BIST does not have an architectural view of the processor to understand its control structures and data flow, structural techniques, like test point insertion, typically needs to be performed to increase the fault coverage.

For example, [14] reports the application of a logic BIST tool to the picojava processor core from Sun Microsystems [16]. While application of logic BIST originally produced an unacceptably low fault coverage of 58.8%, applying logic BIST after inserting test points, with the same configuration of LFSRs and scan chains, yielded a significantly improved fault coverage of 82.5%.

However, it has been observed, that even with the help of other techniques like test point insertion, which inserts additional hardware to make the circuit more random pattern testable, conventional LFSR-based logic BIST techniques cannot always achieve very high fault coverage for processor cores. The fault coverage can be improved by increasing the size of the LFSR and the number of random patterns. However, the improvement does not seem to be significant. It is to be noted that the experiences reported use commercial logic BIST tools, which do not incorporate all the recent research advances made in hardware logic BIST. Directions that look promising include the use of deterministic BIST [17], and additional hardware to enable application of some deterministic patterns, besides the pseudo-random patterns generated by the LFSR, to increase the fault coverage [18]. While these techniques have been applied to benchmark circuits, application to real processor cores have not been reported yet.

Besides inserting additional hardware like test points to improve the fault coverage, application of logic BIST typically requires extensive design changes [14, 15]. Certain violations that do not happen in the functional mode, such as bus-contentions and the forming of combinational loops, could occur during the application of random test patterns, making testing difficult. For example, the signals in a combinational loop may toggle when the loop is activated, causing the generation of undefined values. To avoid these violations, additional design changes need to be made, like breaking the combinational loops with control points. Similarly, embedded memories have to be bypassed with scan flip-flops in the test mode [14], as otherwise they could become sources of undefined values (X-generators), leading to the corruption of MISR signatures. Other design changes include splitting all bi-directional pins into separate I/O pins, and replacing tri-state buffers with selectors.

Due to the diversity of the designs, the existing commercial logic BIST tools cannot support the automation of the required design changes. Thus, many of these design changes have to be carried out manually, significantly adding to the design time. In addition, the design changes may degrade the final performance of the circuit, making it necessary to re-design in some cases.

## 4.2 Self-Testing Using Processor Instructions

An alternative to hardware-based self-testing techniques like BIST is software-based self-testing. While computer systems are regularly equipped with software programs to perform in-field testing, the tests done are typically used for checking the functionality of the system, but not for detecting manufacturing defects. Functional validation suites have been regularly used to perform manufacturing testing of processors. However, its application relies on external testers and its results in terms of manufacturing fault coverage are low, as functional tests are not targeted at structural faults. Recently, researchers have started investigating self-test techniques for processors using processor instructions. Shen et al., and Batcher et al. have proposed techniques for functional self-testing of processors [19, 20]. Both techniques rely on generating and applying random instruction sequences to processor cores. In [21, 22, 23, 17], the processor functionality has been used for on-chip test pattern generation and test response compaction. In [21] and [22], random operations and operands are generated and applied to test the ALUs of DSP cores. In [23] and [17], the processor is used to generate random test patterns, and scan chains are used to apply the test patterns.

A new software-based self-testing methodology is proposed in [15, 24], which uses a software tester embedded in the processor memory as a vehicle for applying structural tests. The software tester consists of programs for test generation and test application. The software-based approach has the advantage of programmability and flexibility, which can be used to generate desirable random test sets on-chip without any hardware overhead. In addition, software instructions can enable on-chip test application by guiding test patterns through the complex control structure of the processor, rather than with the help of scan chains and boundary-scan chains as is done in the case of hardware-based logic BIST techniques.

To circumvent the low fault coverage associated with random pattern testing of processors, the approach first determines the structural test needs of processor components, which are usually much less complex than the full processor, and hence much more amenable to random pattern testing. At the processor level, the instructions of the processor are used to apply the tests to each component at-speed. Since the instructions satisfy the complex control flow of the processor, the flow of test data to/from the component under test will not be impeded, as in the case of hardware BIST applying random patterns to the entire processor. The effectiveness of the proposed method has been shown on a processor core, Parwan [15].

Recently, research has also started in addressing the problem of testing path delay faults in a processor core using instructions [25]. It has been observed that a structurally testable path in a processor, a path testable through at-speed scan, may not be testable by its instructions because no instruction sequence can produce the desired test sequence which can sensitize the paths and capture the fault effect into the destination output/flip-flop at-speed. It has been shown that such functionally untestable paths need not be tested. Hence, identification of such paths helps determine the achievable path delay fault coverage and reduce the subsequent test generation effort. Experimental results for two processor cores (Parwan [15] and DLX [26]) indicate that significant percentage of structurally testable paths are functionally untestable and thus need not be tested. The issue of synthesizing test programs to test the functionally testable paths in a processor core has been addressed in [27].

In general for wide applicability of the instruction-based self-test techniques, more systematic approaches need to evolve, which can be applied to different processor cores without having to manually synthesize the self-test programs, which may involve significant engineering cost.

# 5. Future Research in SoC Test

Having described proposed methodologies and current industrial practices in testing system-on-chips, we next investigate some research issues that need to be addressed for testing future SOCs.

## 5.1 Testing for Noise in DSM SOCs

The use of ultra deep submicron technologies will allow GHz chips with billions of transistors, enabling the integration of giga-scale systems on single chips [1, 2]. However, the use of nanometer technologies is also imposing severe challenges to testing such nano-chips, as new defect mechanisms and new fault effects evolve. In this section, we briefly discuss some of the imminent noise problems of nano-chips, and the test mechanisms that are being developed to address such DSM defects.

Manufacturing process variations and defects in DSM SOCs will lead to effects like cross-coupling between interconnects, as well as voltage drops and ground bounces, beyond the design margin of an aggressive design. Several design [28] and analysis techniques [29, 30, 31, 32] have been developed to help design for margin and minimize signal integrity problems. However, the amount of over design may be prohibitive. Moreover, it is impossible to anticipate in advance, all the process variations and manufacturing defects that may significantly aggravate the cross-coupling effects. Hence, the need to test for manufacturing defects leading to signal integrity problems.

Several crosstalk test generation techniques have been developed recently that can be applied to generate tests for local interconnects in gate-level circuits [33, 34, 35, 36, 37]. However, no DSM test techniques have yet been developed addressing SOCs, and the above gate-level DSM test techniques may not be able to scale to the complexity of SOCs. Also, since the DSM defects can severely affect the delay of the manufactured chip, it is critical to develop DSM test techniques that can be applied at the operational speed of the chip under test. Hence, to facilitate at-speed testing, as well as circumvent the prohibitively rising cost of external ATEs, the problem of self-testing for crosstalk and other DSM faults in SOCs need to be addressed.

DSM effects will most significantly affect the complex interconnects that will dominate the performance of future system-on-chips [38, 39]. Hence, fault models and self-test techniques need to be developed to test for crosstalk and other DSM defects in long interconnects in SOCs. A self-test technique, using on-chip test generators and error detectors, has been proposed for testing crosstalk defects in SOC interconnects [40]. However, the cost involved may be excessive for some chips, and new low-cost DSM self-test techniques need to be researched.

## 5.2 DSM-Aware Self-Testing

Besides the need to test for DSM defects, the test methods applied to DSM chips need to be aware of potentially adverse DSM effects, and need to consider the constraints imposed by nano-meter technologies. For example, our empirical study, partially reported below, shows significantly different characteristics of power consumption in DSM technology than considered and accounted for traditionally.

To ascertain the effect of DSM technologies on the power consumption of long interconnects, we conducted experiments using a 100 nm, 1.2V, 5-line bus system running at 100 MHz. The technology parameters used are from the SIA Roadmap [1]. Under the assumption that all input transitions are launched simultaneously and have the same slope (100 ps), we simulated a transmission-line model of the bus interconnects using HSPICE [41]. Figure 4 plots the power consumed in interconnects varying with the length of the interconnects. The plot labeled "without crosstalk" does not take into consideration the coupling capacitances and inductances due to the use of nanometer technology, while the plot labeled "with crosstalk" does.

Figure 4 shows that DSM effects play a significant role in energy dissipation. Energy dissipation on DSM interconnects is more significantly affected by the cross-coupling capacitances between interconnects than the load capacitances. This also means that power consumed will increase significantly in long interconnects, as shown in Figure 4. Our studies also show that the power consumed will significantly depend on the vectors applied. Hence, it is critical to avoid excessive costs associated with the significant power needs of testing DSM chips, including the need for more expensive packaging as well as more expensive power grids on chip. Self-test techniques need to be developed that minimize the peak and average power consumption during testing, taking into account DSM effects.

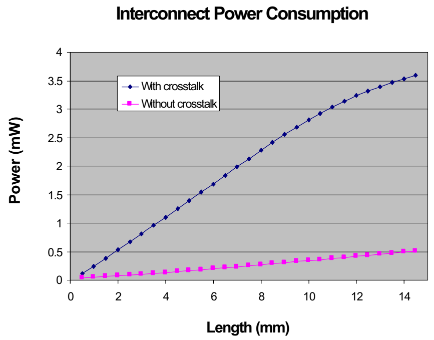

Figure 4. Interconnect Power Consumption of 1 mm wire.

Other potentially adverse effects during testing DSM chips, like EM effects due to high current density and temperature during test application, need to be investigated. Self-test methodologies need to be developed, which takes into account the potential costs involved in testing DSM system-on-chips, and any potential damage that may be caused by the significantly different effects of nanometer technologies.

## 5.3 Testing Configurable and Platform-based SOCs

Recently, there has been a rising academic as well as commercial interest in reconfigurable cores and system-on-chips. Several reconfigurable cores are already available and being used, for example processor cores from LSI Logic, Tenselica, and ARC, whose instruction set as well as peripherals can be configured before use in a system-on-chip to best fit the needs of the intended SOC application. Similarly, instead of creating a new SOC for every new application, there is a push towards a platform-based approach, where a platform-based SOC can be programmed for the desired application.

While reconfigurable cores, and platform-based SOCs, will greatly enhance productivity, test reuse and hence test productivity may be a problem. For example, since a core can be used in the SOC in any of its numerous configurable forms, it may be difficult for the core providers to have pre-determined test sets and test structures for use by the cores users, thereby increasing the SOC test development time and cost. Similarly, issues related to standardization of test and access mechanisms need to be addressed for programmable and platform-based SOCs.

## *Acknowledgement*

We would like to acknowledge Yi Zhao and Li Chen for their help with the paper.

# References

1. The National Technology Roadmap for Semiconductors (ITRS), 1997 Edition. Semiconductor Industry Association.
2. The International Technology Roadmap for Semiconductors, 1999 Edition, ITRS.
3. Rajesh K. Gupta and Yervant Zorian. Introducing Core-Based System Design. IEEE Design & Test of Computers, 14(4), pp.15–25, December 1997.
4. Yervant Zorian, Erik Jan Marinissen, and Sujit Dey. Testing Embedded Core Based System Chips. In Proceedings IEEE International Test Conference (ITC), pages 130–143, Washington, DC, IEEE Computer Society Press, Oct. 1998.
5. Yervant Zorian, Erik Jan Marinissen, and Sujit Dey. Testing Embedded-Core-Based System Chips, IEEE Computer, 32(6), pp. 52–60, June 1999.
6. IEEE Computer Society. IEEE Standard Test Access Port and Boundary-Scan Architecture IEEE Std. 1149.1-1990. IEEE, New York, June 1993.
7. Erik Jan Marinissen and Yervant Zorian. Challenges in Testing Core-Based System ICs. IEEE Communications Magazine, 37(6), pp. 104–109, June 1999.
8. IEEE P1500 Web Site. http://grouper.ieee.org/groups/1500.
9. Erik Jan Marinissen, Yervant Zorian, Rohit Kapur, Tony Taylor, and Lee Whetsel. Towards a Standard for Embedded Core Test: An Example. In Proceedings IEEE International Test Conference (ITC), pages 616–627, Atlantic City, NJ, September 1999. IEEE Computer Society Press.
10. Lee Whetsel. Addressable Test Ports: An Approach to Testing Embedded Cores. *Proceedings IEEE International Test Con-ference (ITC)*, pages 1055–1064, Atlantic City, NJ, September 1999. IEEE Computer Society Press. IEEE Computer Society Press.
11. Erik Jan Marinissen, Sandeep Kumar Goel, and Maurice Lousberg. Wrapper Design for Embedded Core Test, *Proceedings IEEE International Test Conference (ITC)*, Atlantic City, NJ, October 20000. IEEE Computer Society Press. April 2000.
12. Yervant Zorian. A Distributed BIST Control Scheme for Complex VLSI Devices, *Proceedings IEEE VLSI Test Symposium (VTS)*, pages 6–11, Princeton, NJ, April 1993. IEEE Computer Society Press.
13. V.D. Agrawal et al.. Built-in self-test for digital integrated circuits, *AT&T Technical Journal*, pp. 30, Mar. 1994.
14. T.G. Foote, D. E. Hoffman, W. V. Huott, T. J. Koprowski, B. J. Robbins, and M. P. Kusko, Testing the 400 MHz IBM Generation-4 CMOS Chip, *Proceedings of the International Test Conference 1997*, Washington DC, pp. 106 – 114, Nov. 1999.
15. L. Chen, S. Dey, P. Sanchez, K. Sekar, and Y. Chen. Embedded Hardware and Software Self-Testing Methodologies for Processor Cores, *Proceedings of the Design Automation Conference,* Los Angeles, CA, pp. 625 – 630, June 2000.
16. "PicoJava Microprocessor Core," Sun Microsystems, http:// www. sun. com /microelectronics/picoJava/.
17. R. Dorsch and H.-J. Wunderlich, "Accumulator based deterministic BIST," *Proceedings of the International Test Conference*, Washington DC, pp. 412 - 421, Oct. 1998.
18. N. Touba and E. J. McCluskey, Synthesis of Mapping Transformed Pseudo-random Patterns for BIST, Proceedings of International Test Conference, pp. 674-682, 1995.
19. J. Shen and J. A. Abraham, "Native mode functional test generation for processors with applications to self test and design validation," *Proceedings of the International Test Conference 1998*, Washington, DC, Oct. 1998, pp. 990-999.
20. K. Batcher and C. Papachristou. Instruction randomization self test for processor cores, *Proceedings of the 17th IEEE VLSI Test Symposium*, Dana Point, California, pp. 34 – 40, April 1999.
21. J. Rajski and J. Tyszer. *Arithmetic Built-in Self-Test for Embedded Systems*, Prentice Hall, 1998.
22. K. Radecka, J. Rajski, and J. Tyszer. Arithmetic built-in self-test for DSP cores, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol.16, no.11, pp. 1358 – 69, Nov. 1997.
23. S. Hellebrand and H.-J. Wunderlich. Mixed-mode BIST using embedded processors, *Proceedings of the International Test Conference 1996*, Washington DC, pp. 195 – 204, Oct. 1996.
24. L. Chen and S. Dey. DEFUSE: A Deterministic Functional Self-Test Methodology for Processors, *Proceedings of the 18th IEEE VLSI Test Symposium*, pp. 255 – 262 ,April 2000.
25. W.-C. Lai, A. Krstic, and K.-T. Cheng. On Testing the Path Delay Faults of a Microprocessor Using its Instruction Set. *Proceedings of IEEE VLSI Test Symposium* , pp. 15-20, May, 2000.
26. M. Gumm. VLSI Design Course: VHDL-Modelling and Synthesis of the DLXS RISC Processor. University of Stuttgart, Germany, December 1995.
27. W. C. Lai, A. Krstic, and K. T. Cheng. Test Program Synthesis for Path Delay Faults in Microprocessor Cores. *Proceedings of International Test Conference,* Oct., 2000.
28. K. L. Shepard. Design Methodologies for Noise in Digital Integrated Circuits, Proceedings Design Automation Conference, pp. 94-99, 1998.
29. D. S. Gao, A. T. Yang and S. M. Kang. Modeling and simulation of interconnection delays and crosstalk in high-speed integrated circuits, *IEEE Trans. on Circuits and Systems*, Vol. 37, pp. 1-9, January 1990.
30. H. You and M. Soma. Crosstalk and transient analysis of high-speed interconnects and packages, *IEEE Trans. on Solid State Circuits*, Vol. 26, pp. 319-30, March 1991.
31. C. Gordon and K. M. Roselle. Estimating crosstalk in multiconductor transmission lines, *IEEE Trans. On Components Packaging and Manufacturing Technology*, Vol.19, May 1996.
32. K. Rahmat J. Neves, and J. Lee. Methods for calculating coupling noise in early design: a comparative analysis. *Proceeding International Conference on Computer Design VLSI in Computers and Processors*, pp.76-81, 1998.
33. N. Itazaki, Y. Matsumoto, and K. Kinoshita. An Algorithmic Test Generation Method for crosstalk Faults in Synchronous Sequential Circuits, *Proceedings Sixth Asian Test Symposium*, pp. 22-7, Nov. 1997.
34. K. T. Lee, C. Nordquist, and J. Abraham. Automatic Test Pattern generation for Crosstalk Glitches in Digital Circuits, *Proceedings IEEE VLSI Test Symposium*, pp. 34-39, 1998.
35. W. Chen, S. K. Gupta, and M. A. Breuer. Test Generation in VLSI Circuits for Crosstalk Noise, *Proceedings IEEE International Test Conference*, pp. 641-650, 1998.
36. W. Chen, S. K. Gupta, and M. A. Breuer. Test Generation for Crosstalk-Induced Delay in Integrated Circuits, *Proceedings IEEE International Test Conference*, pp. 191-200, October 1999.
37. A. Sinha, S.K. Gupta, and M.A. Breuer. Validation and Test Generation for Oscillatory Noise in VLSI Interconnects, Proceedings of the International Conference on Computer-Aided Design, November 1999.
38. P. Nordholz, D. Treytnar, J. Otterstedt, H. Grabinski, D. Niggemeyer, and T.W. Williams, Signal Integrity Problems in Deep Submicron arising from Interconnects between Cores, *Proceedings IEEE VLSI Test Symposium,* pages 28-33, 1998.
39. M. Cuviello, S. Dey, X. Bai, and Y. Zhao. Fault Modeling and Simulation for Crosstalk in System-on-Chip Interconnects, *Proceedings of the International Conference on Computer-Aided Design*, pp. 297-303, November 1999.
40. X. Bai, S. Dey, and J. Rajski. Self-Test Methodology for At-Speed Test of Crosstalk in Chip Interconnects, *Proceedings of the Design Automation Conference*, June 2000.
41. HSPICE, v98.2, Avant! Corporation, Fremont, California.