Cover Feature

Testing Embedded-Core-Based System Chips

The attributes that make the design of system chips built with IP cores an attractive methodology—design reuse, heterogeneity, reconfigurability, and customizability—also make testing and debugging these system chips a complex challenge.

Yervant Zorian
LogicVision

Erik Jan Marinissen
Philips Research Laboratories

Sujit Dey
University of California, San Diego

Recently, reusable modules have captured the imagination of designers who understand the potential of embedding such modules to build on-chip systems. Doing so is similar to using integrated circuits on a printed circuit board (PCB), and designers are forming rich libraries of predesigned, preverified building blocks. These so-called *embedded cores* make it easier to import technology to a new system and differentiate the corresponding product by leveraging intellectual property advantages. Most importantly, the use of embedded cores shortens the time-to-market for new systems due to design reuse.[1]

Embedded cores incorporated into system chips cover a wide range of functions. Typically, cores are the hardware description of today's standard ICs: digital signal processor, RISC processor, or DRAM core. They also use an unprecedented range of technologies, from CMOS logic to DRAM to analog circuits. Cores sometimes come in hierarchical compositions; these so-called complex cores incorporate one or more simple cores.

Cores come in a range of hardware description levels, categorized as *soft* (register-transfer level), *firm* (netlist), and *hard* (technology-dependent layout). These three types offer trade-off opportunities. Soft cores leave much of the implementation to the designer, but are flexible and process-independent. Hard cores have been optimized for predictable area and performance, but lack flexibility. Firm cores offer a compromise between the two. Each type of core has different modeling and test requirements.[1]

However, the practical implementation of the core-based design scenario is fraught with unresolved issues: design methods for building single-chip systems; sign-off for these systems; and intellectual-property licensing, protection, and liability. The most critical challenges of this emerging discipline include manufacturing test and design debug.

SYSTEM CHIP TEST CHALLENGES

Even though the design process in core-based system chips is conceptually analogous to traditional board design, their manufacturing test processes are quite different. In the traditional system-on-board approach, as shown in Figure 1a, the component provider performs chip design, manufacturing, and testing. The system integrator is responsible for design, assembly, and test of the PCB. Under the assumption that components are fault-free, testing is limited to testing the interconnects between the chips.

A core is only a description of a module, and is not yet manufactured as it is transferred from core provider to core user. Hence the core provider cannot test his product for manufacturing defects. This can only be done once the core, embedded into the system chip, is manufactured. This makes the test of the embedded core a joint responsibility of core provider and system integrator, as shown in Figure 1b.

Core-level test

The system integrator has a very limited knowledge of the structure of the adopted core, and often deals with it as a black box. This is especially true if the core is hard or is an encrypted intellectual-property block. In such cases, the core provider develops the core test—including the design-for-test structures and the corresponding test patterns—and delivers it with the core.

On the other hand, the core provider has little or no knowledge about the system chip environment of the core and possibly even the target fabrication process. For instance, they may not know

0018-9162/99/$10.00 © 1999 IEEE

Terminology and Acronyms

ATE (automatic test equipment): Hardware engines to perform test, diagnosis, and measurement; and application software to control these engines. Typically a stand-alone (external) machine; recently, some of its hardware functions are being embedded on-chip.

ATPG (automatic test pattern generation): Typically, ATPG is based on one or more fault models—abstractions of defect behavior—and yields structural tests.

BIST (built-in self-test): The capability of a circuit to test (parts of) itself. Test stimuli are generated on-chip and test responses are either compared or compacted on-chip.

Functional versus structural test: A functional test lets the chip function as in normal operation. A structural test specifically focuses on detecting manufacturing defects and might let the chip behave different from normal operation to efficiently detect certain defects.

I_{DDQ} and I_{DDT} test: Test methods that measure supply current I_{DD} . For CMOS circuits, I_{DD} in a quiescent (stable) state (I_{DDQ}) should be almost zero, and hence an elevated current indicates the presence of a defect. I_{DDQ} testing is especially effective in detecting shorts between wires, a defect difficult to detect using voltage-based test methods. Transient current (I_{DDT}) testing, measures the current pulses following each clock edge. I_{DDT} might be used for circuits in which the I_{DDQ} is too high, such as is the case for large deep-submicron ICs.

PCB (printed circuit board): The functionality of a PCB, including its ICs, is referred to as a "system."

Scan design: Through additional dedicated hardware, a chip test mode is created in which state-holding elements such as flip-flops form one or more serial shift registers. By shifting these *scan chains*, the IC's internal state can be controlled and observed. In *full scan*, all flip-flops are part of a scan chain, whereas in *partial scan*, only a subset of flip-flops are scannable.

SOC (system on chip): The integration of a complete system, which until recently consisted of multiple ICs on a PCB, onto one chip. An SOC often includes multiple types of circuitry, such as digital logic, memories, and analog circuitry. Typically, SOCs are designed using embedded reusable cores.

TAM (test access mechanism): On-chip hardware infrastructure to transport test stimuli from the on- or off-chip hardware that generates them to the embedded core. TAMs also transport test responses from the embedded core to the on- or off-chip comparator.

UDL (user-defined logic): SOC logic that does not belong to acquired external cores, but is created by the system-chip designer.

- which test method—built-in self-test, scan, I_{DDQ}, or functional test—to adopt (for a brief explanation see the "Terminology and Acronyms" sidebar),
- what types of faults (static, dynamic, or parametric) to target, and
- what level of fault coverage to use.

Because they lack this information, core providers may develop tests that do not ensure a suitable quality level. If the fault coverage is too low, the quality of the system chip is at risk; if it is too high, the test cost may become prohibitive in terms of test time, performance, area, or power. Furthermore, different manufacturing processes have different defect densities and distributions.

The core-internal test developed by the core provider must be adequately described, such that the system integrator can use it to build a chip-level test. To do so, core tests need a standard format. Such a standard format—a core test description language—is under development by the IEEE P1500 working group (see the "Toward Standards for Interoperability" sidebar).

Test access

Another key difference between traditional approaches and those for system chips is the accessibility to component terminals—the primary inputs and outputs of chips and cores. With a system on board, the chips are tested as stand-alone unts, and during their

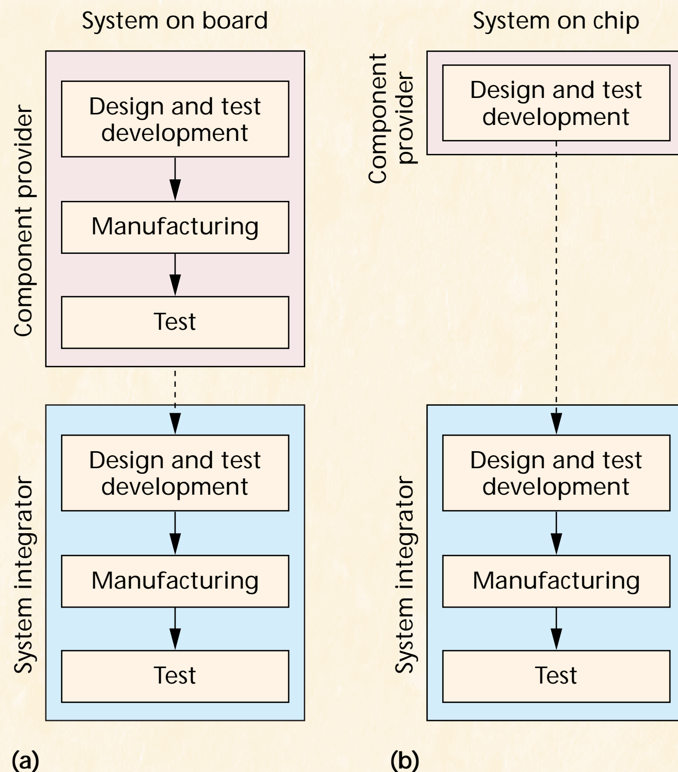


Figure 1. Design development varies between (a) system-on-board and (b) system-on-chip design.

Toward Standards for Interoperability

The IEEE P1500 working group is working toward a standard to facilitate interoperability with respect to testing. The standard does *not* standardize a core's internal test methods or chip-level test access configuration. The standardization effort focuses on *nonmerged* cores—cores that are tested as stand-alone units—and addresses two main issues:

- 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 system chip design.

Recently, P1500 has also started a study into the test interoperability issues concerning *mergeable* (soft) cores.

IEEE P1500 has active participation from system companies, electronic design automation vendors, core providers, IC manufacturers, and ATE vendors. It has close ties to the Virtual Socket Interface Alliance (VSIA, http://www.vsi.org). The P1500 working group holds public meetings, often in conjunction with major conferences (see http://grouper.ieee.org/groups/1500 for more information).

Much of the leading-edge work in system chip test is presented at the International Test Conference (http://itctestweek.org), at the International Workshop on Testing of Embedded Core-Based Systems (http://grouper.ieee.org/groups/1500/tecs), and in *IEEE Design & Test of Computers* magazine (http://computer.org/dt). All three receive strong support from the IEEE Computer Society's Test Technology Technical Council.

tests, we have direct physical access to chip pins. In contrast, cores are often deeply embedded in a system chip, so direct physical access is not available. The chip design must then provide an electronic test access infrastructure from the chip pins to the terminals of the embedded core. In a later section, we define a conceptual architecture for such an infrastructure. On top of being able to deliver the core tests to the embedded core, this infrastructure should also provide features to test the hardware in between the cores, and to isolate a core from its surroundings, if test modes require it.

System-chip-level test

One of the major challenges in developing a system chip is the integration and coordination of on-chip test and diagnosis capabilities. Compared to conventional PCBs, the requirements of system chip test are far more complex than PCB assembly test.

The system chip test is a composite test that consists of the individual tests for each core, tests for user-defined logic (UDL), and tests for interconnect logic and wiring. The system chip's composite test requires adequate test scheduling. Test scheduling must meet several chip-level requirements, such as total test time, power dissipation, area overhead, and so on.[2,3] Also, test scheduling is necessary to run intracore and intercore tests in a certain order to avoid affecting the initialization and final contents of individual cores. The composite system chip test must comply with these scheduling constraints.

In addition, tests for system chips share the testing challenges inherent in very deep-submicron chips—providing sufficient defect/fault coverage, containing overall test cost, and meeting time-to-market.

CONCEPTUAL ARCHITECTURE FOR CORE TEST

We distinguish three elements in the embedded core test infrastructure, as depicted in Figure 2.

- *Test pattern source and sink.* The source generates the test stimuli for the embedded core, and the sink compares the response(s) to the expected response(s).
- *Test access mechanism.* The test access mechanism transports test patterns. It provides on-chip transport of test stimuli from a test pattern source to the core under test. It also transports test responses from the core under test to a test pattern sink.
- *Core test wrapper.* The wrapper forms the interface between the embedded core and its environment. It connects the terminals of the embedded core to the rest of the IC and to the test access mechanism.

All three elements can be implemented in various ways, such that a whole palette of possible approaches to testing embedded cores emerges. We review the various alternatives and classify current approaches.

TEST PATTERN SOURCE AND SINK

Designers can implement test pattern sources and sinks either off-chip, using external automatic test equipment (ATE), or on-chip, using built-in self-test (BIST), or a combination of both. Source and sink need not be of the same type; that is, an embedded core's source can be off-chip, while its sink is on-chip. Three factors influence the choice of a certain type of source or sink:

- the type of circuitry in the core,
- the predefined tests that come with the core, and
- quality, test time, and cost considerations.

Core circuitry

Today, system chips use three main types of circuitry: digital logic, memory, and analog. Simple cores consist of only one type; complex cores combine multiple simple cores, possibly of different circuitry types.

These three types of circuitry exhibit different defect behavior and require different tests.[4] The various tests require different types of sources to generate the stim-

uli and sinks to compare the responses. Typically, distinct ATE systems as well as BIST schemes are used for logic, memory, and analog circuitry. System chips, which often incorporate all three types of circuitry into one IC, are encouraging ATE vendors and BIST providers to integrate their traditionally separate solutions for logic, memory, and analog into combined product offerings. Hence, instead of using a separate ATE for the logic part of the system chip, a second ATE for the embedded memory and a third for the analog circuitry, ATE vendors are offering "super" ATE systems to combine the test capabilities of all three types.

Core tests

The variety of core tests is much larger than the three circuitry types. Tests are classified by the type of circuit they test, but also by the measurements they require (voltage or current), by the way they are generated (based on the IC's function or structure), by the amount of core-internal adaptation they require (scan or test points), and so on.

Examples of current measurement tests are I_{DDQ} and I_{DDT},[5] which measure quiescent and transient currents. Current can be measured both by sinks on-chip (current monitors)[6] as well as off-chip (current monitors in an ATE system).

Not all test patterns can be generated on-chip in a cost-effective manner. The test patterns of cores that come with function tests and/or ATPG-generated tests are often irregular in structure. It is difficult to generate such irregular deterministic test patterns on-chip at acceptable area costs.

Quality and cost

Off-chip sources and sinks often require large capital investment and, because they are built using yesterday's technology, suffer from various problems. Faster ICs require increasing accuracy to detect timing signals at the IC pins. Although tester accuracy has improved by 12 percent annually, IC speeds have improved by 30 percent per year. This growing gap reduces off-chip testers' ability to properly identify bad chips, leading to yield losses and cost increases. Furthermore, it is becoming increasingly difficult for off-chip ATE to keep up with the very high frequencies needed to sufficiently test performance-related defects in today's ICs.

In addition to these test-quality-related issues, the increased pin count and the mixing of diverse technologies in system chips will cause ATE costs—for running the tests as well as the equipment itself—to rise toward $20 million, according to the *1997 SIA National Technology Roadmap for Semiconductors*.[7] These problems with the quality and cost of external ATE will only become worse for high-speed, high-density, and mixed-technology system chips, rendering external ATE unacceptably inaccurate and prohibitively expensive.

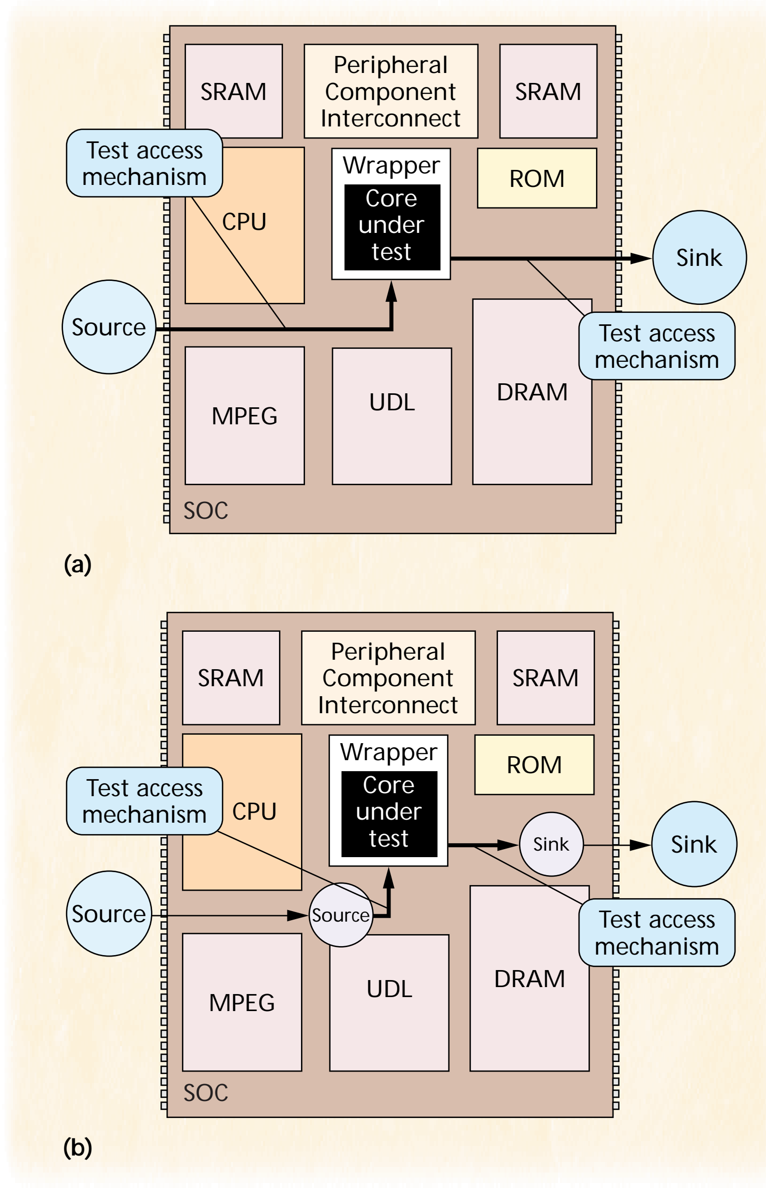


Figure 2. A test pattern source uses a test access mechanism to transport test patterns to a wrapper around the core. A test access mechanism also transports test responses to the test pattern sink for evaluation. This architecture applies to test patterns generated by automatic test equipment (ATE) from (a) outside the chip. It also applies to built-in self-test (BIST) configurations, which have source and sink (b) inside the chip.

The type of circuitry a core uses and the predefined tests that come with it determine the implementation options for test pattern source and sink. The option chosen is in general determined by quality and cost considerations. On-chip sources and sinks provide better accuracy and performance-related defect coverage. On the other hand, they increase the silicon area and hence can reduce manufacturing yield—the big-

The core test wrapper is the element that, if standardized, could preeminently contribute to the interoperability of multiple cores for various types and sources.

ger the chip area, the more chance of a particle falling on that chip and causing a defect.

In theory, all kinds of test patterns can be generated on-chip, but only a few are used in practice. Only algorithmic patterns, such as the regular patterns for memory testing,[4] functional patterns for analog cores,[8] or pseudorandom patterns for random logic[4] are generated on-chip without using excessive silicon area. In most cases, on-chip sources and sinks also need some form of off-chip ATE, for example, to perform initialization or comparison of a signature. Hence on-chip solutions do not completely avoid ATE costs.

TEST ACCESS MECHANISM

A test access mechanism takes care of on-chip test pattern support. It can be used to transport

- test stimuli from the test pattern source to the core under test, and
- test responses from the core under test to the test pattern sink.

By definition, the test access mechanism is implemented on-chip. Although one core often uses the same type of test access mechanism for transporting both stimulus and response, such consistency is not required and various combinations may coexist.

Designing a test access mechanism involves making trade-offs between the mechanism's transport capacity (bandwidth) and its test application cost. Bandwidth is limited by the bandwidth of source and sink and the silicon area you want to spend on the test access mechanism itself. A wider test access mechanism provides more bandwidth, but consumes more wiring area. For example, if the test pattern source is an external ATE, it does not make much sense to provide a mechanism wider than there are IC pins available to connect it to. In this case, the IC pins are the bandwidth bottleneck, and a wide mechanism costs more silicon area without adding to that bandwidth.

Test time is a result of the test data volume of the individual cores and the bandwidth of the test access mechanism. How expensive test time is per unit of time depends on the type of source and sink. There is a wide range of external ATE with similarly wide-ranging associated test costs, and these again differ from the cost of test application time for BIST.

There are several options for implementing a core test access mechanism, which can

- reuse existing functionality to transport test patterns or be formed by dedicated test access hardware;
- go through other modules on the IC—including other cores—or pass around those other modules;
- provide access for only one core or for multiple cores; or
- be a plain signal transport medium or may contain certain intelligent-test-control functions.

We describe and reference several proposed and currently used test access mechanisms in an earlier publication.[9]

CORE TEST WRAPPER

The core test wrapper is the interface between the embedded core and its system chip environment. It connects the core terminals both to the rest of the IC as well as to the test access mechanism. By definition, the core test wrapper is implemented on-chip and should have the following mandatory modes:

- *Normal operation* (nontest). In this mode, the core is connected to its system IC environment, and the wrapper is transparent.
- *Core-internal test*. The test access mechanism is connected to the core such that a source can apply stimuli at the core's inputs, and a sink can observe responses at the core's outputs.
- *Core-external test*. The test access mechanism is connected to the interconnect wiring and logic such that a source can apply stimuli at the core's outputs, and a sink can observe responses at the core's inputs.

Apart from these mandatory modes, a core test wrapper can also have several optional modes. For example, a wrapper can include a detach mode to disconnect the core from its system chip environment and the test access mechanism.

Depending on the test access mechanism's implementation, some of these modes can coincide. For example, if the test access mechanism uses existing functionality, normal and core test modes can coincide.

Predesigned cores have their own internal clock distribution system. Different cores have different clock propagation delays, which might result in clock skew for intercore communication. The system-IC designer should take care of this clock skew in the functional communication between cores. However, clock skew might also corrupt the data transfer over the test access mechanism, especially if multiple cores share this mechanism. The core test wrapper is the best place in the test access paths between cores to implement clock skew prevention.

The test collar[10] and the TestShell[11] are examples of core test wrappers. Both support the features we've just described.

Of the entire core test architecture, the core test wrapper is the element that, if standardized, could preeminently contribute to the interoperability of multi-

ple cores for various types and sources. Such a standard wrapper is under development by the IEEE P1500 working group (see the "Toward Standards for Interoperability" sidebar).

DESIGN VALIDATION AND DEBUG

Although using predesigned cores can reduce the product design cycle, and system-level integration improves system performance and costs, productivity gains could be significantly reduced by the challenges of validating and debugging a system chip. Though each core is individually preverified before system integration, validating the correct functionality and timing of the complete system chip and debugging and diagnosing any possible error can be time-consuming. The main challenges in validating and debugging a system chip arise from

- the heterogeneity of the system chip,
- the need to validate and debug software and hardware simultaneously, and
- the accessibility problems associated with deeply embedded cores.

A system chip needs to be validated against design and timing errors at various phases of its design cycle: specification and algorithmic validation, architectural validation, and finally prototype validation, the latter referring to validating the prototype or actual silicon of the system.

Prototype-level system validation is the most accurate and fast, but debugging, diagnosing, and correcting errors this late in the design cycle can be costly. For this reason, system validation techniques like cosimulation seek to help validate systems early in the design cycle. However, prototype-level validation and silicon debug remain indispensable, especially for systems that require at-speed validation. Such speed is unattainable by current high-level validation methods, including cosimulation. We describe techniques for prototype validation, particularly focusing on system debug.

Prototype validation

The most widely used prototype validation technique for conventional systems has been *emulation*—imitation of part or all of the target system by another system. Emulation typically employs field-programmable gate arrays (FPGAs) to implement hardware parts of the target system, and processors for the software. While emulation allows nearly at-speed validation of the system (about 1 million cycles per second), it can be very expensive. Moreover, though the availability of FPGA cores is on the rise, FPGA-based emulation may be inefficient and incompatible for hard cores, necessitating silicon debug.

Debug of system chips, consisting of programmable components like microprocessors and DSPs, requires the capability to monitor software executing on the programmable cores. This capability in turn requires access to the internal registers and buses of deeply embedded processor cores.

Emulation-based validation and debug, however, provides poor access to a processor's internals. This problem can be overcome by an *in-circuit emulator* (ICE), a box of hardware that can emulate the processor along with the rest of the target system. An ICE can execute code in the target system's memory and allows the user to set breakpoints and perform other debug tasks. However, an ICE physically replaces the actual processor and so does not replicate errors that stem from electrical characteristics. In addition, a deeply embedded processor increases cost because a design must include extra paths to bring the internal signals out to the ICE.

The wide variation in test strategies, test structures, and test requirements employed by heterogeneous components poses serious challenges to testing system chips.

Another silicon debug technique permits debug of the actual processor in the target system chip. A *debug monitor* is a program that resides in the system chip's memory. The debug monitor controls program execution on the processor, communicating with the debug host system through an RS-232 or UART serial port. It can provide debug features like setting breakpoints, uploading data from target memory, and downloading application programs. A clear advantage of this approach is the ability to debug the processor cores and application software in the presence of other hardware cores. However, placing the debug monitor in the system chip's memory can entail significant extra cost, unless it is used and then removed from the final product.

Embedding debug in cores

Many of the problems and costs associated with debugging system chips can be overcome by embedding appropriate debug capabilities in cores. Several processor and DSP cores incorporate in-circuit emulation, providing a "virtual ICE" that interfaces between a source-level debugger/emulator and the core embedded in a system chip. For example, Advanced RISC Machines (ARM) builds EmbeddedICE into its ARM 7TDMI processor,[12] and Motorola incorporates a debug module in its ColdFire cores.

FUTURE CHALLENGES

Evolving design methodologies and the aggressive shrinking of semiconductor technologies will have profound effects on testing system chips. These factors are also beginning to stretch the limits of test equipment.

Design methodologies

Although design reuse could significantly improve design productivity, it poses serious challenges to test-

ing the system chip because of the wide variation in test strategies, test structures, and test requirements employed by the heterogeneous components. The industry needs to develop new test methodologies to address several issues.

Analog/mixed-signal cores and system chips. Several applications, such as wireless telecommunications products, demand mixed-signal systems. Such systems consist of digital, radio frequency (RF), analog, and mixed-signal components. As system integration technologies advance, system chips consisting of RF, analog, and microelectromechanical (MEMS) components combined with cores that contain new high-performance and low-power devices will become common. The industry needs to develop techniques for testing such cores, investigating issues like test specification, access mechanisms, and isolation mechanisms.

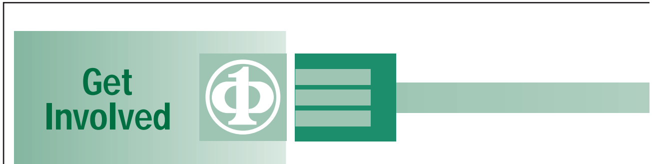
Customizable and reconfigurable cores. A major advantage of system design using IP cores and system-level integration is the configurability and customizability available to the system designer. In the system-on-chip design paradigm, the processor core, associated peripheral units, and underlying bus, memory, and communication architectures, can all be configured and customized to best match the embedded-system application. This customization yields tremendous cost and performance advantages over traditional system-on-board approaches. As tools and methodologies mature to enable configurable and customizable system chips, the industry will need to address the standardization of test and access mechanisms. For example, when core users can customize the cores themselves, how can test sets and test structures be predetermined for reuse during system chip integration and test?

Soft cores. As core-based design reuse matures and the reuse of nonprocessor functions and protocols in multimedia and telecommunications systems become more prevalent, the delivery and use of soft cores will increase significantly. Unlike hard cores, soft cores, which are typically high-level descriptions of the functionality, cannot be characterized for testability, so it's impossible to reuse the tests. To deliver testable soft cores and reuse tests, industry must develop effective high-level testability analysis and design-for-testability techniques.

Validation and debug. As the software content in system chips increases and complex buses and protocols are used to communicate and interface among on-chip components, debugging and diagnosing design errors will consume a significant portion of the design cycle time. Because cores are preverified, the primary source of design errors will be in the interfaces and protocols between cores. To make the system chip validation and debug problem more tractable, the industry needs to develop an interface-based system chip validation and debug methodology. We need to define validation wrappers and access mechanisms, as in the case of testing and manufacturing defects. Also, core providers can embed debug structures in other cores, enabling easy debug of system chips. The embedded debug structures should also reuse as much of the embedded test structures as possible.

Deep-submicron technology

The *1997 National Technology Roadmap for Semiconductors*[7] predicts reusable core-based system chips using 100-nm technology, gigahertz clock frequency, and less than 1-V power supply by 2003 to 2006. This will lead to on-chip noise, due to increased cross coupling of capacitances, inductances, and electromagnetic fields. Recent studies show significant increases in signal delay and hazards due to cross-coupling



About the Test Technology Technical Council

This article is based on a tutorial presented at the 1998 International Test Conference, an annual event that this year will be held 26-30 September. In addition to ITC, Test Week (http://www.itctest-week.org) includes 16 full-day test technology tutorials, three full days of advanced test workshops, test standardization working group meetings, a roundtable discussion to appear in *IEEE Design & Test of Computers*, and other events.

Creating such forums is a main goal of TTTC (Test Technology Technical Council), a professional organization sponsored by the IEEE Computer Society. TTTC's main objectives are to contribute to its members' professional advancement and to help advance the state of the art in the testing of electronics. While ITC and Test Week are its flagship events, the realization of these objectives requires year-round efforts.

Technical Diversification

Since its inception 21 years ago, TTTC has covered a wide range of technical domains to meet the challenges of the dynamic test industry. We expanded our offerings from manufacturing-related activities such as IEEE 1450 STIL (Standard Tester Interface Language) to the next stages of the electronic product life cycle. We began covering the premanufacturing design stage with workshops such as the International Test Synthesis Workshop and post-

capacitances between bus interconnects. The effects are most dramatic for wire lengths greater than 10 mm, but are significant for wire lengths as short as 2 mm in 100-nm technology.

Also, cores themselves may not be immune to chip-level noise; for instance, bus interconnects going over a core may affect the core's operation. These effects are why the *NTRS* recommends developing signal integrity tests.

Several cross-talk extraction and analysis tools have been developed recently. Although useful for design validation, these tools cannot be used for manufacturing testing, as they don't specifically target manufacturing defects. Also, which vectors will be effective for cross-talk simulation is unclear. Hence, we need to develop fault models and test and diagnosis methodologies for cross talk and other noise. Such techniques must cover buses and global interconnects that connect the cores of a system as well as the interference effects on the cores themselves. Also, since cross-talk effects will be most evident in high-frequency circuits, test and diagnosis methodologies must work at the speed of the cores.

Test equipment

External ATE faces accuracy and performance limitations and is expected to become prohibitively expensive.

A common solution to these problems is using self-testing methods for system chips. Self-testing methods allow at-speed testing and also reduce the need for expensive external testing by ATE. One strategy is to put the external ATE functionality (or the speed-critical parts of it) on the system chip itself, which side-

---

manufacturing with activities such as the International On-Line Test Workshop.

TTTC covers all the levels of integration. Although we initially concentrated on the chip level with events such as the VLSI Test Symposium, we moved into boards and systems, offering related workshops and standards such as IEEE 1149.1, then moved to sub-chip levels by providing embedded core test standards such as P1500 and workshops like Testing Embedded Core-Based System Chips. TTTC addresses analog technology with the Mixed-Signal Test Workshop, memory technology with the Memory Technology Design and Test Workshop, and processor technology with the Microprocessor Test and Verification Workshop.

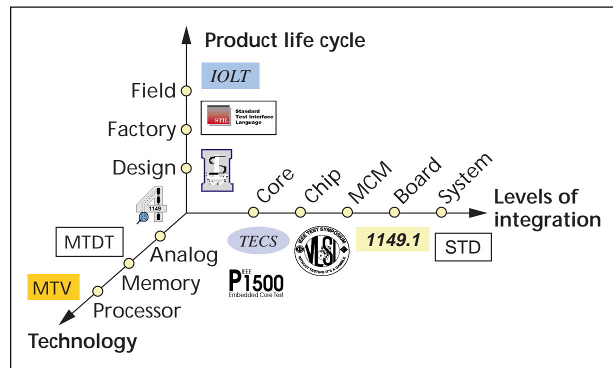


Figure A. TTTC activities advance to match changes in the test field.

Global Focus

In addition to this technical diversification, TTTC has always been keen on globalization. We have strengthened key events around the world such as the Asian Test Symposium, DATE, the European Test Workshop, and the Latin American Test Workshop. Also, TTTC's four regional groups, in Asia/Pacific, Europe, Latin America, and North America, sponsor technical meetings and educational programs and host standardization efforts.

Leadership in Education

In addition to sponsoring more than 30 technical meetings, TTTC provides educational opportunities through a tutorials program. This year, TTTC initiated a comprehensive Test Technology Educational Program (TTEP) that will allow test professionals to earn official TTTC certification if they complete a certain number of full-day tutorials, offered at five TTTC technical meeting sites.

TTTC sponsors 18 Technical Activity Committees (TACs) that address emerging test technology topics. Under TAC guidance, TTTC initiates and encourages new test standards. IEEE standardization efforts include six working groups sponsored by TTTC.

TTTC stays in touch with its members through newsletters, e-mail, and a Monthly Planner. Membership is free; benefits of membership include an online member home page directory at http://computer.org/tttc and reduced fees for test books and journals.

TTTC advances test technology and influences the careers of its members by being a source for the state of the art in test technology, providing world-class test education, initiating standards, and creating a strong network of professionals. If you are interested in joining this network, contact the TTTC Office (tttc@computer.org). For suggestions or further information, contact the TTTC Chair, Yervant Zorian (zorian@logicvision.com).

steps the speed-related problems of external ATE. Of course, system developers must evaluate the trade-off between external ATE cost and the extra silicon area for embedded ATE.

BIST is emerging as a viable embedded ATE methodology for certain types of cores like those for embedded memory,[4] random logic,[4] and analog components.[8] However, the industry still needs to develop self-testing methodologies for complex cores like general-purpose and digital signal processors, and for RF, flash memory, and electromechanical components.

While design reuse of embedded cores contributes to the efficiency of system chip designers, true interoperability can only be achieved if the tests for these cores can also be reused. The solutions and the proposed standards we describe are expected to play a key role toward developing a plug-and play methodology in the core-based design paradigm. ❖

Acknowledgments

We thank our colleagues at LogicVision, Philips, and UCSD, as well as the members of the IEEE P1500 working group for many stimulating discussions.

References

1. R.K. Gupta and Y. Zorian, "Introducing Core-Based System Design," *IEEE Design & Test of Computers*, Oct.-Dec. 1997, pp. 15-25.
2. Y. Zorian, "Test Requirements for Embedded Core-Based Systems and IEEE P1500," *Proc. IEEE Int'l Test Conf.,* IEEE CS Press, Los Alamitos, Calif., 1997, pp. 191-199.
3. Y. Zorian, "A Distributed BIST Control Scheme for Complex VLSI Devices," *Proc. IEEE VLSI Test Symp.*, IEEE CS Press, Los Alamitos, Calif., 1993, pp. 6-11.
4. V.D. Agrawal et al., "Built-In Self-Test for Digital Integrated Circuits," *AT&T Technical J.*, Mar. 1994, p. 30.
5. M. Sachdev, P. Janssen, and V. Zieren, "Defect Detection with Transient Current Testing and its Potential for Deep Sub-Micron CMOS ICs," *Proc. IEEE Int'l Test Conf.,* IEEE CS Press, Los Alamitos, Calif., 1998, pp. 204-213.
6. K. Wallquist, A. Righter, and C. Hawkins, "A General Purpose I_{DDQ} Measurement Circuit," *Proc. IEEE Int'l Test Conf.,* IEEE CS Press, Los Alamitos, Calif., 1993, pp. 642-651.
7. *The 1997 National Technology Roadmap for Semiconductors,* Semiconductor Industry Assoc., San Jose, Calif., 1997.
8. S. Sunter and N. Nagi, "A Simplified Polynomial-Fitting Algorithm for DAC and ADC BIST," *Proc. IEEE Int'l Test Conf.,* IEEE CS Press, Los Alamitos, Calif., 1997, pp. 389-395.
9. Y. Zorian, E.J. Marinissen, and S. Dey, "Testing Embedded-Core-Based System Chips," *Proc. IEEE Int'l Test Conf.,* IEEE CS Press, Los Alamitos, Calif., 1998, pp. 130-143.
10. P. Varma and S. Bhatia, "A Structured Test Re-Use Methodology for Core-Based System Chips," *Proc. IEEE Int'l Test Conf.,* IEEE CS Press, Los Alamitos, Calif., 1998, pp. 294-302.
11. E.J. Marinissen et al., "A Structured and Scalable Mechanism for Test Access to Embedded Reusable Cores," *Proc. IEEE Int'l Test Conf.,* IEEE CS Press, Los Alamitos, Calif., 1998, pp. 284-293.
12. *The ARM7TDM1 Debug Architecture,* ARM DAI 0028A, Advanced RISC Machines Ltd., Dec. 1995; http://www.arm.com/Documentation/AppNotes/Apps28vA.

Yervant Zorian *is the chief technology advisor for LogicVision Inc., San Jose, Calif. He was previously a distinguished member of technical staff at Bell Labs, Lucent Technologies. Zorian has an MS in computer engineering from the University of Southern California and a PhD in electrical engineering from McGill University, Canada. He is the editor-in-chief of* IEEE Design & Test of Computers *and chair of the IEEE Computer Society Test Technology Technical Council. He is the founder and chair of the International Workshop on Testing Embedded Core-based System-Chips (TECS) and the IEEE P1500. He is a Golden Core member of the IEEE Computer Society and a Fellow of the IEEE.*

Erik Jan Marinissen *is a member of the scientific staff at Philips Research Laboratories in Eindhoven, the Netherlands. His research interests include all topics in the test and debug of digital ICs. Marinissen has an MS in computing science and an MTD (master of technological design) in software engineering, both from Eindhoven University of Technology, the Netherlands. He is also on the organizing committees of several test workshops, an active participant in the IEEE P1500 working group, and a member of the IEEE Computer Society.*

Sujit Dey *is an associate professor in the ECE department at the University of California, San Diego. His recent research focuses on developing design, validation, and test methodologies for core-based system chips. Dey has a PhD in computer science from Duke University. He has coauthored more than 70 journal and conference papers, one book, and holds 10 US patents. He is in the organizing committees of several conferences and workshops about test and computer-aided design. Dey is a member of the IEEE.*

Contact Yervant Zorian at LogicVision Inc., 101 Metro Dr., Third Floor, San Jose, CA 95110; zorian@logicvision.com.