DesignCon 2008

On-Chip Instrumentation for In-System IP Validation of a 6 Gbps Serial ATA Platform Device

Martin Schrader, Infineon Technologies AG
martin dot schrader at infineon dot com

Miron Abramovici, DAFCA, Inc.
miron at dafca dot com

Daniel Hoggar, DAFCA, Inc.
Daniel dot hoggar at dafca dot com

Abstract

This paper describes an innovative solution to the challenges of in-system IP validation, in particular the implementation and silicon results for a new, high-speed, 6 Gigabit-per-second Serial-ATA platform chip from Infineon. Using a commercially available silicon validation platform, Infineon successfully inserted the on-chip instrumentation with the newly developed Infineon IP, and employed it in silicon, to both validate and optimize the 90nm platform device for use by Infineon's licensees. Additional results describe down-stream system-level development and analysis applications, enabled through the use of an API to access the on-chip instrumentation in the lab to monitor communications between the IP blocks.

Authors' Biographies

Martin Schrader is Staff Engineer Digital Design and Project Manager, ASIC Design and Security (ADS) Business Unit at Infineon Technologies. He is responsible for SATA IP core development and integration into customer SoC. He founded the Serial Interfaces Expert Group of Infineon ADS and is member of the digital workgroup of the SATA International Organization (SATA IO). Martin has over 9 years of experience in digital design and project management working with worldwide cross functional teams.

Miron Abramovici, Ph.D., is co-founder and CTO of DAFCA. Previously he was a Distinguished Member of Technical Staff at Bell Labs in Murray Hill, NJ. He is a Fellow of IEEE. He co-authored *"Digital Systems Testing & Testable Design",* adopted world-wide as the standard textbook in this field. Dr. Abramovici has 28 issued patents and over 80 publications.

Daniel Hoggar is Director of Applications Engineering at DAFCA. He has extensive experience in EDA and ASIC Design most notably in areas of physical synthesis and optimization. Prior to DAFCA, Daniel leveraged his technical expertise as the Product Specialist at TeraSystems. He has also held multiple technical positions during seven years with Mentor Graphics and before that, with Compass Design Automation, in Sophia Antipolis, France.

Introduction

The ever-increasing complexity of SoCs makes IP reuse a key design strategy, and the integration of IP cores in an SoC design has become an integral part of pre-silicon SoC verification. In-system silicon validation and debug has become the biggest bottleneck in the development cycle of a new SoC, and the validation of IP cores in silicon is a big headache in this task. Successful pre-silicon verification of an IP core is a necessary but insufficient condition for its successful validation prior to volume production. Even the most sophisticated SoC design methodology cannot fully account for all the parameters that impact silicon behavior, or for all logic "corner cases" that occur in the real life of a chip working at speed and in system. For example, the simultaneous occurrence of two unlikely events may not be anticipated pre-silicon, so it is never simulated or analyzed; however, it may cause unexpected behavior when it occurs in system. Pre-silicon verification methods – simulation, emulation, FPGA prototyping, static timing analysis, and formal verification – do not address many deep sub-micron problems that occur in the actual device, such as unexpected behaviors resulting from signal integrity, power, noise, cross-talk, thermal stress, or process-related issues.

Additional difficulties occur when silicon validation teams are not fully familiar with the details of the IP core operation, and when the IP providers do not supply observability hooks into the behavior of the IP core. Unfortunately, these are typical situations. Many times one can witness a finger-pointing game played between the silicon validation team claiming, “Your IP does not work in silicon!” and the IP provider team countering with “Show me that you are using it correctly!” There is no easy way of objectively refereeing such a dispute without means of observing and controlling the action in silicon.

In this paper we first review how different features of a new silicon validation solution can be used for IP validation, and then we provide details on its use in the validation of a high-speed, 6 Gigabit-per-second Serial-ATA platform chip that includes several IP cores. The objective of the IP provider was to validate in silicon all SATA protocols involving the IP core, so that any possible silicon failures would point to the IP environment and not to the IP core.

A New Silicon Validation Solution

This section is an overview covering a new and comprehensive approach to in-system silicon validation and debug [1, 2]. Pre-silicon, instrumentation tools enable and guide the user to insert reconfigurable instruments for validation; insertion is done at RTL and the instrumented SoC is processed by standard synthesis-based design flows. The

instruments do not require any additional pins or special libraries, and they are independent of technology and functional applications. The user can tailor the instruments and their characteristics for the target design architecture. The instrumentation creates an infrastructure platform that is dynamically configured and controlled by post-silicon tools to implement validation and debug functions. Instruments can be configured during normal operation of the device; they do not interfere with its function or impede performance. Dynamic configuration enables continuous reuse of the instrumentation for a variety of applications. The post-silicon tools communicate with the instruments via a JTAG cable connected to a standard IEEE 1149.1 Test Access Port.

The reconfigurable platform supports many validation paradigms such as on-chip signal capture and logic analysis, assertions, stimulate-and-capture, performance monitoring, fault and error injection, and "what-if" experiments. The tools bridge the pre-silicon verification with in-system silicon validation, by extracting behavior from silicon to verify within simulation and by reproducing simulation experiments in silicon. In-system scan-based validation augments the flexibility of reconfigurable at-speed instruments with complete register observability enabled by existing scan chains. Our techniques can efficiently deal with problems inherent in system-level validation, such as reduced internal observability, lack of expected values, and pseudo-intermittent behavior. The reconfigurable platform and its applications automate tasks that previously required an intensive engineering effort.

The platform is a distributed fabric composed of reconfigurable instruments. Figure 1 illustrates an instrumented SoC. The instruments include a *programmable trigger engine* (PTE), a *signal probe network* (SPN), a *reconfigurable logic engine* (RLE), and a *capture-and-stimulate* instrument (CapStim). An SPN is a pipelined multiplexer network with automatically inserted FIFOs to handle asynchronous clock domain crossings that collects a large group of signals and is configured to select a subset to be connected to analysis instruments. The PTE and the RLE process system signals connected to their inputs and are configured to implement functions such as detecting and counting events, identifying transactions, and checking assertions. A CapStim contains a circular buffer memory that can capture input signals as well as apply preloaded (post silicon) or captured signals to other blocks. A *Tracer* implements only the capture function of a

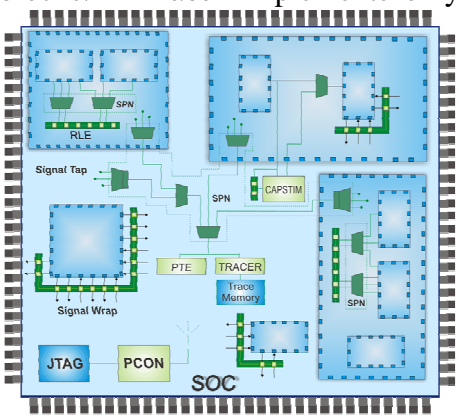

Figure 1. An instrumented SoC

CapStim. The primary controller (PCON) is the interface between the JTAG port and instruments. The instruments work at speed using system clocks; access and configuration are done with the JTAG clock.

The two analysis resources target different kinds of flexibility vs. efficiency trade-offs. A PTE can be created with built-in counters, timers, and comparators, and includes one built-in finite state machine (FSM) template, while an RLE can implement more general combinational and sequential logic circuits.

In the remainder of this section, we show how different platform features can be used for in-silicon IP validation.

Key Signals

Each of the following scenarios assumes that the IP designers have provided a list of key I/O and internal signals in their core. The key signals tell an external observer what is going on inside the core, so we can analyze them to validate correct operation or to detect misbehavior. Proper documentation should be also provided, so that the chip designer will know what behavior to expect to observe on the key signals. The key signals typically include:

- Signals that identify and are associated with important transactions the core is involved in.
- Signals that determine the mode of operation of the core
- Signals of FSMs that control interfaces and communication protocols between the core and the rest of the SoC.
- Signals that control clock gating, power gating, dynamic voltage scaling, and frequency scaling.

Identifying the key signals enables tapping and bringing them via an SPN to analysis instruments. The key signals should be grouped by clock domains (if applicable) so that all signals in a group are available concurrently for analysis. The RTL instrumentation tools feature automatic capabilities for selecting groups of probe points, such as all inputs and/or outputs of a specified block or bus.

Logic Analysis

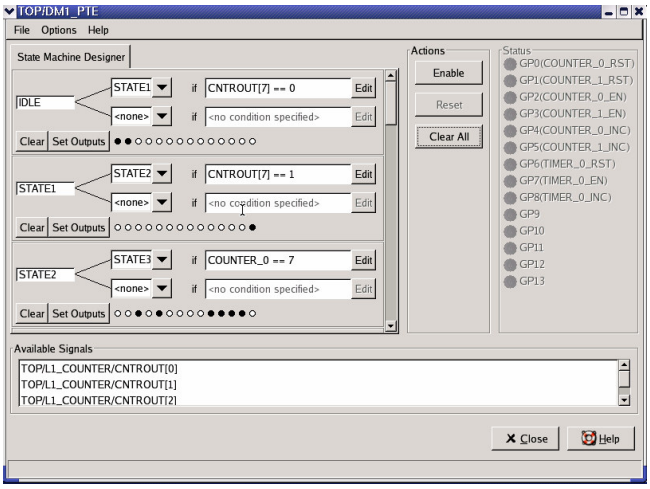

Figure 2. Implementing an FSM with a PTE

The instrumentation can be configured to operate as an on-chip logic analyzer. For example, assume we want to capture and analyze waveforms associated with a group of key signals in the IP core that are involved in a certain transaction. For this analysis, we configure the SPN to connect the target signals to a Tracer and a PTE. We configure a FSM in the PTE to recognize the completion of the transaction (Figure 2). We configure the Tracer to continuously record its inputs. We program the FSM to stop the trace when the transaction is detected; thus, the trace buffer will contain signal values captured immediately before the end of the transaction. After the recording stops, we extract the time-stamped values from the trace buffer via the JTAG port and display them with a VCD or FSDB waveform viewer (Figure 3). In this way the waveforms from silicon appear exactly as if they originated in simulation.

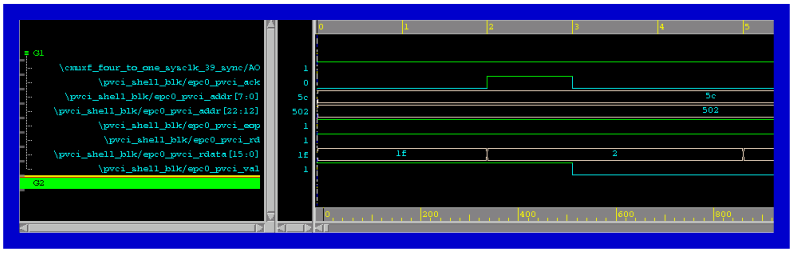

Figure 3. Actual silicon waveforms

Assertions in Silicon

An *assertion* represents a relation between signals that must always be satisfied in a correctly operating system. For example, an assertion may state that the IP core should not get a new *Request* for processing data while its current operation is not yet *Done*, while another assertion may specify that a *Request* should be answered by an *Acknowledge* in, at most, 3 cycles. An assertion fires when it detects misbehavior.

Assertion checkers are widely used in pre-silicon verification. The reconfigurable instrumentation enables, for the first time, a large number of assertions to be used for *post-silicon* validation and debug. In the example above, *Request*, *Done*, and *Acknowledge* should be in the same group of key signals available via an SPN. Assertions are implemented in PTEs and/or RLEs connected to output of SPNs. The system provides parameterized configurations for the well-known Open Verification Library (OVL) assertions. The user can create a new assertion by programming the FSM that implements the assertion checker in a PTE (Figure 2), or an assertion can be defined in Verilog and compiled into an RLE configuration.

The IP provider may specify assertions to check the groups of key signals tapped in the IP core. Usually these will be assertions that have been checked in pre-silicon verification, but new assertions may be defined, as well. Note that an assertion that never fired in simulation may fire in silicon when the IP core reaches a corner state not encountered pre-silicon. Even an assertion that has been formally proven as correct may fail in silicon because of timing, cross-talk, or other unforeseen problems.

Reconfigurability enables processing a large number of assertions by dividing the set of target assertions into groups and configuring one group at a time. Several assertions can run concurrently all the time or for specified intervals. Just several minutes of real-time system operation provides each assertion with more stimuli than it would get in days of simulation. The ability to repeatedly configure and run different groups of assertions in a loop allows for continuous reuse of the instrumentation and an *automated, at-speed, in-system validation procedure.*

Validating assertions of an IP core in silicon shows that the core behaves correctly with respect to the target assertions; on the other hand, if an assertion fires, we know exactly what the problem is. Assertions of an IP core can also check whether the assumptions of correct usage (e.g., no *Request* until *Done*) are respected and determine the responsibility (the core or its environment) in case of a problem. Using the set of assertions supplied by the IP provider frees the validation team from the need to understand the detailed internal behavior of the IP core and enables an automated validation procedure.

Assertions can be used in conjunction with the embedded logic analyzer to aid the debugging process. For example, an assertion firing can be employed as a trigger to stop recording in the trace buffer; the recorded signals provide a window into activity preceding the malfunction detected by the assertion.

On-Chip Functional IP Core Test

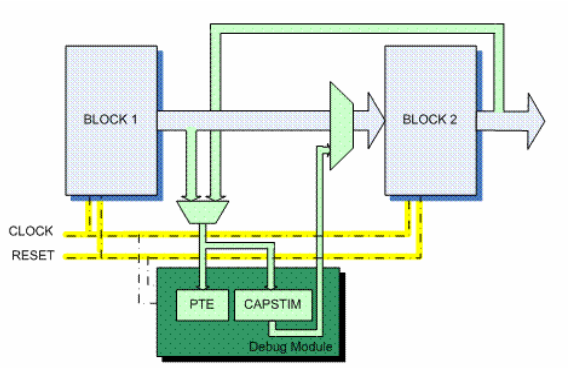

Figure 4. On-chip test with CapStim

Many times we would like to validate on-chip a certain aspect of the operation of an IP core by applying a transaction to the inputs of the target and observing its response. This on-chip functional test can be done using a PTE and CapStim as illustrated in Figure 4, where BLOCK2 is the target IP core. Each input to BLOCK2 feeds a MUX to enable the replacement of the functional inputs with a pattern provided by the CapStim. The stimuli to be applied are preloaded in the CapStim memory via JTAG access. We configure the instrumentation so that when the PTE detects a trigger event, it selects the CapStim outputs to drive the block inputs. Concurrent with applying its pre-stored vectors as at-speed stimuli to the block, the CapStim also captures its output signals, which are then extracted and analyzed off-chip. The PTE-CapStim pair can be shared among several blocks, so that the same CapStim can be repeatedly reused with different stimuli and with different target IP cores.

The CapStim instrument was developed specifically for on-chip IP core validation, based on Infineon requirements. It has now become a standard instrument in the in-system silicon validation solution, useful in any situation in which the user wants to control an embedded block.

Integrating At-Speed and Scan-Based Validation

In addition to the tapped key signals, other registers inside an IP core may carry useful functional information regarding the current state of the core. Clearly, it would be impractical to tap all flip-flops of a core. However, usually all registers have a scan mode of operation where they are configured as scan chains, easily controllable and observable by scan-in and scan-out operations during manufacturing testing. The pre-silicon instrumentation tools modify the serial access to existing scan chains so they can be accessed in-system via the JTAG interface just like the other instruments.

State information stored in registers of the IP core is valuable only if captured at specific times of interest. For example, we can analyze the tapped key signals at speed using

triggers and/or assertions, and selectively stop the functional clocking when an event of interest occurs. After the clocks stop, the post-silicon tool can scan out the contents of the internal registers via the JTAG port. This operation is called a *scan dump*. In this way, the strength of in-system, at-speed validation is combined with complete register observability provided by scan chains**.** This combination takes advantage of the complementary nature of the two methods; while the at-speed part accesses a limited group of signals for a large number of clock cycles, the scan dump accesses a large number of signals at a single clock cycle.

After the scan dump, the software can extract the values of registers of interest and automatically check their values. Scan dumps are most effective when used with at-speed assertions and the embedded logic analyzer. We can make the firing of an assertion stop the clocks when misbehavior is detected and use the register values obtained by scan dumps to complement information provided by signals recorded in the trace buffer.

Bridging Pre-Silicon Verification and Silicon Validation

Today pre-silicon verification and silicon validation are quite separate environments. For example, it is very difficult to take a problem encountered in system validation and reproduce it in simulation, where it could be easier to analyze. The new in-system silicon validation solution creates a bridge between pre-silicon verification and silicon validation, and enables different ways to cross it. We have already seen how assertions used in simulation to check signals in the IP core can be brought to silicon.

Now let us assume that an assertion checking the internal behavior of the IP core is firing in silicon without ever failing in simulation. A first question is likely to be "Can we reproduce this problem in simulation?" The answer is important since it determines the type of problem we are dealing with. The solution is to record the inputs to the core in the trace buffer, and use the assertion firing to stop recording. Then we extract the input stimuli from the trace buffer, create a testbench with these stimuli and the firing assertion, and simulate. If the assertion does also fire in simulation, the problem is indeed a bug in the core, and it is exposed by a corner case that escaped pre-silicon verification. If the assertion fires only in silicon, we have a mismatch between the simulation of the core and its silicon implementation. In this case, the root cause of the misbehavior is likely to be a silicon-only problem not visible in simulation, such as cross-talk, speed path, or a defect.

Let's look at another way of crossing the bridge between pre-silicon and post-silicon. Assume that, in simulation, we verified a particular function of the IP core, but in system it is difficult to reproduce the same input sequence to the core. Then we extract the inputs applied in simulation to the core during a specified interval and load them into the CapStim connected to the IP core. We configure the instrumentation so that when the CapStim detects a trigger event, its outputs drive the core inputs. Concurrent with applying its pre-stored vectors as at-speed stimuli to the core, the CapStim also captures the outputs of the core, which are then extracted and compared with the corresponding simulation results.

Validating the SATA Platform

The SATA Platform

Figure 5 describes the SATA platform device. It consists of two SATA subsystems each build of a PHY and an Link Transport Layer following the SATA specification. Both SATA subsystems are able to operate independed from each other and allow complete SATA communication with one chip only. The SATA platform device implements only the pure SATA IP cores without an own application layer like a complete host adapter or disk drive device. Instead of this the on-chip instrumentation replaces the required functionality to operate the SATA subsystem.

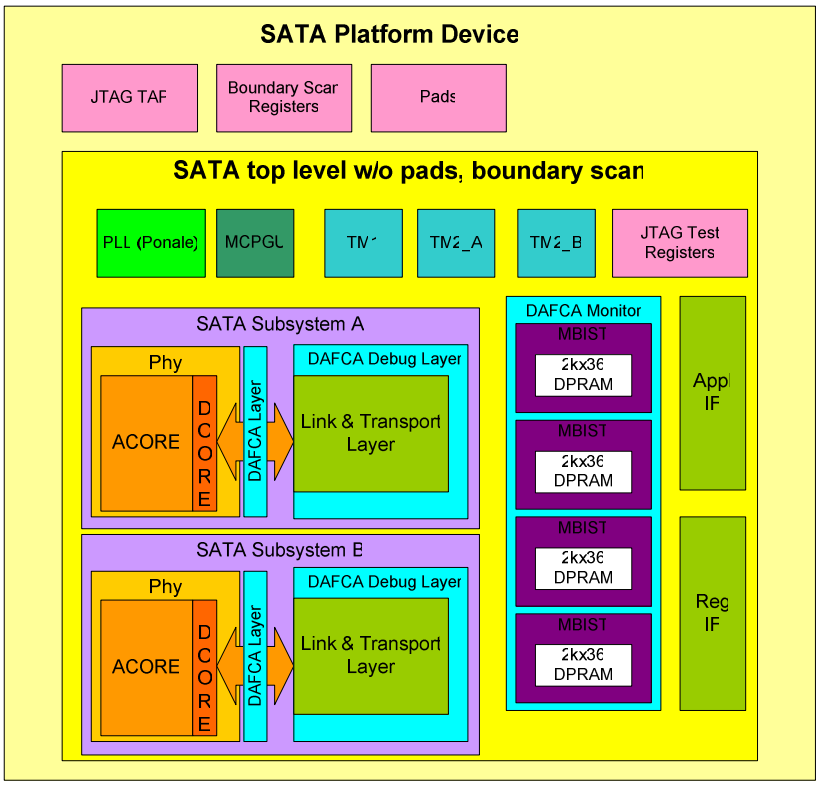

Figure 5. The SATA platform

Figure 6 describes the SATA subsystem and the instrumentation in more detail. It shows the main building blocks PHY Control, Link Layer, and Ttransport Layer with enbedded state machines and FIFOs.The blue circles highlight the instrumentation inserted at the interfaces between major modules. These instruments allow to operate the PHY in an stand-alone mode as well as the complete SATA subsystem like a real application.

The objective of the instrumentation was to enable PHY stand-alone silicon debug and overall SATA compliance testing in the lab. It was required to stimulate real situations in combination with external SATA equipment, such as data transfers and power-down scenarios.

The PHY Stand-alone Mode

The PHY stand-alone mode was motivated by the requirement of evaluating the base PHY functions in the lab. This mode enables direct access to internal PHY functions to stress the device under different operation conditions. It is possible to apply usual compliance patterns according the SATA specification or own patterns. With the real-time trace function it is possible to trace and stimulate complex patterns reqired for long-term bit error rate testing. The requirement was to apply any patterns without restrictions of upper layer encoding, scambling, or frame composition.

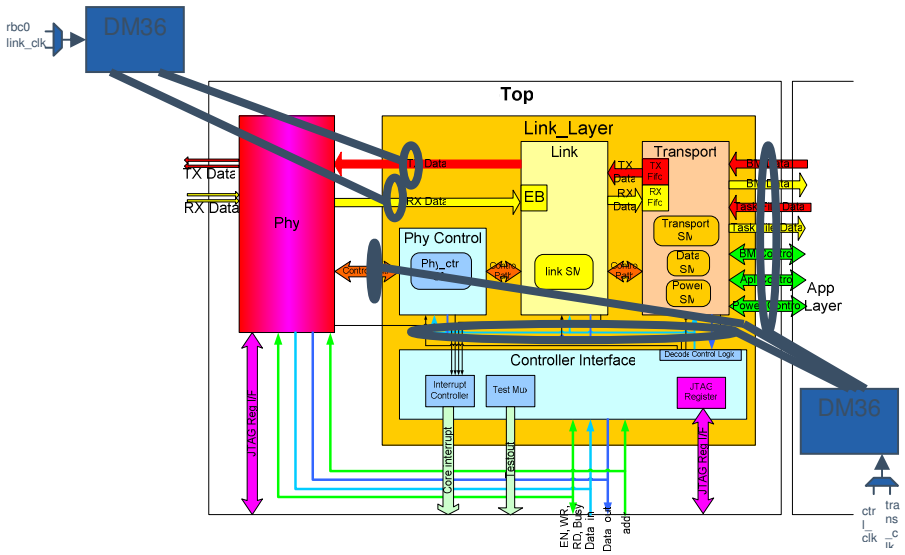

Figure 6. The SATA Subsystem

The Application Mode

The main objective of the project was the PHY 6 Gbps development in Infineon's 90nm technology. However, the system level behavior of the PHY together with a link and transport layer was necessary the be proven on real silicon. But the link and transport layer is not able to operate as is, it is normaly connected to an on-chip application layer and CPU subsystem. The effort of building up a full-blown system including application layer, CPU system, required firmware devoplment and complex application board was way out of the project's budget . Another possibilty would be to bring all required interface signals to external pads on the chip and combine it with an external FPGA to a demonstrator platform. This option was not achievable due to the large number of required interface signals.

The instrumentation platorm provided a simple and easy replacement of the full-blown application layer. The instrumented modules enabled the operation in an system-like behavior. It is possible to setup real FIS transfers including non-data FIS like Register FIS, PIO Setup FIS, DMA Setup FIS, as well as data FIS. Here we used again the real-

time trace and CapStim instruments which are connected to large trace memories to send and receive long data transmissions.

With such long data transmissions we were able to prove frame error rates as well as overall system stability and the cooperation of the link and transport layer with the analog blocks of the PHY.

A second area of the lab evaluation using on-chip instruments was the power management mode. Here we were able to execute the defined power management operations of the SATA specification "Partial" and "Slumber". The functionality itself was proven as well as the measurement of the real power saving and correlation with our pre-silicon simulations and estimations.

Instrumentation and Results

The creation of a new 90 nm high-speed IP sub-system for use in the emerging 6Gbps Serial ATA market posed special challenges for Infineon. We had to deal simultaneously with an evolving system specification and the need to rapidly integrate numerous in-house and third-party IP to deliver a fully-functioning silicon platform for end-users.

The platform-chip development team established several key requirements for on-chip instrumentation, including a) observability and control of critical block-to-block interfaces, b) generation of on-chip stimuli of system-level transactions, including both those detailed in the specification as well as corner cases that cover stress and error conditions, and c) means to directly reuse pre-silicon simulation-based validation tests on the silicon device in the lab. The post-silicon objectives included the rapid silicon validation of the Link, Transport, and PHY blocks of the SATA IP using a software-based solution running in the system environment at the lab bench.

The on-chip reconfigurable instrumentation was used in lieu of lab equipment for stimulus generation, system monitoring, and circuit analysis. In addition to providing on-chip capture and control capabilities, at-speed data could be brought off-chip for additional analysis. The IP development team defined and inserted an optimal on-chip instrumentation solution, which included four instrument subsystems each composed of an SPN, a PTE, and a CapStim; each of these targeted specific IP blocks in different clock domains. All post-silicon scenarios were developed in Tcl and verified in the simulation environment prior to the arrival of first silicon. This enabled immediate application of the validation, data acquisition, and test scripts upon first silicon arrival.

Another interesting and unanticipated application was created early in system validation and testing, highlighting the flexibility of the post-silicon validation platform. Infineon required certain on-chip coefficient registers to be set and behavioral results measured at speed; in fact, different coefficients had to be applied dynamically at different times. This revised requirement in the lab-based test environment well demonstrated how on-chip instrumentation can be used for a broad range of applications. A C-based optimization algorithm was implemented and inserted into the device using the software API. The

instruments were also configured to monitor the circuit behavior subsequent to each coefficient setting.

Conclusions

The results from the IP validation and performance optimization applications provided by the on-chip instrumentation were impressive, and were subsequently showcased at the SATA 6G PHY Workgroup meeting held in May 2007 in Milpitas, California. Overall, the post-silicon applications enabled the silicon validation of all the SATA protocols involving the IP core and significantly simplified and accelerated the work of the validation and debug team.

The techniques illustrated in this paper can provide the basis for a standard methodology of validating IP cores in silicon. The standard would define the additional information expected from IP providers to enable the silicon validation of their cores. This information can be provided at several levels. At the simplest level, the IP provider supplies the groups of key internal signals that should be observed and analyzed outside the core to validate its operation. The documentation associated with the core should also describe the expected behavior of the key signals. This specification does not impose any constraints on how the key signals are to be processed outside the core. The system integrator has complete freedom to use commercially-available instrumentation and tools or develop them in-house. At the next level, the IP provider supplies a set of assertions that formally describe the properties associated with the key signals. Again, the chip designer has complete freedom on whether or how these assertions will be checked in silicon. At the third level, the IP provider supplies a set of I/O behaviors of the IP core that specify applied stimuli and their expected responses with the understanding that these behaviors should be observed in silicon. Following such a standard would provide a very good solution to the problem of validating IP cores in silicon.

References

1. M. Abramovici, P. Bradley, K. Dwarakanath, P. Levin, G. Memmi, and D. Miller, "A Reconfigurable Design-for-Debug Infrastructure for SoCs," *Proc. 43rd Design Automation Conf.*, July 2006.
2. M. Abramovici, "A Silicon Validation and Debug Solution with Great Benefits and Low Costs," *Proc. International Test Conf.*, October 2007