WHITE PAPER

METRIC-DRIVEN VERIFICATION ENSURES SOFTWARE DEVELOPMENT QUALITY

BY STEVE BROWN, CADENCE DESIGN SYSTEMS AND MICHEL GENARD, VIRTUTECH

cadence®

INTRODUCTION

The complexity of electronic systems is rapidly increasing, placing new demands on developers to ensure system-level quality while meeting tough time-to-market constraints. The semiconductor and EDA industries have developed methodologies that allow the predictable creation of highly complex systems-on-chip (SoCs). These SoCs, however, must fit into larger systems that are driven by embedded software. More and more, software development is becoming the largest cost—and bottleneck—in the development of electronic systems, and system and software quality is becoming the biggest factor in the success or failure of products in the marketplace.

The need for faster and better systems and software development has led to a methodology called virtualized systems development (VSD). This approach lets design teams quickly build "virtual platforms," or software models of systems hardware, drawing upon libraries of processor and SoC models. Systems architects and software developers can then perform hardware/software partitioning, develop and debug hardware-dependent code, and start applications software development before any system hardware is actually built.

In the hardware world, meanwhile, great strides have been taken towards building a rigorous methodology for both design and verification. Functional verification, which begins as soon as a hardware specification is produced, employs a variety of powerful tools and techniques. For example, IC functional verification tools include software simulation, hardware emulation, assertion-based verification, equivalence checking, and formal (static) property checking.

In recent years, functional verification has moved from a reliance on directed tests to constrained-random stimulus, which can uncover unexpected bugs that directed tests would be unlikely to find. Further, engineers are increasingly using coverage metrics to help determine when the verification task is completed. This includes code coverage, which measures whether lines of code have been executed, as well as functional coverage, which defines scenarios that should be tested.

Recently, a methodology called metric-driven verification (MDV) has been developed to make the IC verification process more effective and predictable. MDV starts with a verification plan that has measurable goals based on coverage metrics. As verification proceeds, metrics are captured and used to guide the verification process.

A recent article[1] identified five steps of verification planning:

- Analyze the device specifications
- Scope the verification objectives
- Identify the feature set of the design
- Design detailed coverage metrics
- Select aggregate metrics to track progress

In the mainstream SoC market, software verification has traditionally been an ad-hoc process using debuggers and "printf" statements. Software developers have not typically used a rigorous verification methodology, or been exposed to the advantages of constrained-random simulation or functional coverage metrics. Recently, however, a collaboration between Virtutech and Cadence Design Systems has extended the benefits of MDV to systems architects and software developers working with virtual platforms.

The collaboration integrates the Cadence® Incisive® Software Extensions with the Virtutech Simics® virtual platform environment. Incisive Software Extensions brings techniques such as verification planning, testbench generation, constrained-random stimulus, and coverage metrics to hardware/software co-verification and embedded software development. By integrating Incisive Software Extensions with Simics, the systems architects and software developers who use Simics can take advantage of the same advanced functional verification features that are available with the Cadence Incisive Enterprise Simulator.

Moreover, the integration links the virtual platform environment into the EDA flow, and provides benefits for hardware development as well. It lets developers test the software/hardware interface and determine whether bugs are occuring in the hardware representation of the platform or the applications software. Testbenches developed in Incisive Software Extensions for use with Simics can give IC designers a good starting point for their own functional verification efforts.

A LOOK AT SIMICS

Virtutech Simics is a flexible and scalable software solution that models electronic systems with high performance and fidelity. By allowing early software development and debugging, it speeds time to market. Simics also enables higher product quality and lowers project risks.

With Simics, design teams can create a complete, transaction-level model (TLM) virtual platform, leveraging an extensive library of target devices, operating systems, CPU architectures, and SoC families. Once the platform is created, developers can run the same unmodified binaries that would run on the actual hardware, including full applications software, in a simulation environment that runs at hundreds of MIPS. Simics provides deterministic execution and offers features such as non-intrusive profiling and fault injection.

The Simics family consists of the following components:

- **Simics Model Builder** – the development platform used to create new device models.
- **Simics Virtual Platform** – a model of the physical target hardware that's being simulated.
- **Simics Accelerator** – allows acceleration on multiprocessor and multicore hosts.
- **Simics Ethernet Networking** – provides connectivity between virtual systems inside a simulation.
- **Simics Hindsight** – lets developers run and debug code, provides visibility into registers, and supports reverse execution that runs code backwards to find the source of a defect.

It's important to note that Simics models complete systems, not just SoCs or boards. Simics can also model multicore and multiprocessor systems, and greatly ease programming and debugging for parallel hardware. Although multicore systems are inherently non-deterministic, Simics can re-introduce control and repeatability in the debugging process. This makes debugging a multiprocessor as easy as debugging a single program on a single processor.

A virtual platform environment such as Simics allows many capabilities that would be impossible in actual hardware, including control over time, inspection of all hardware and software states in the system, ability to span multiple software domains, and visibility into hardware-software boundaries.

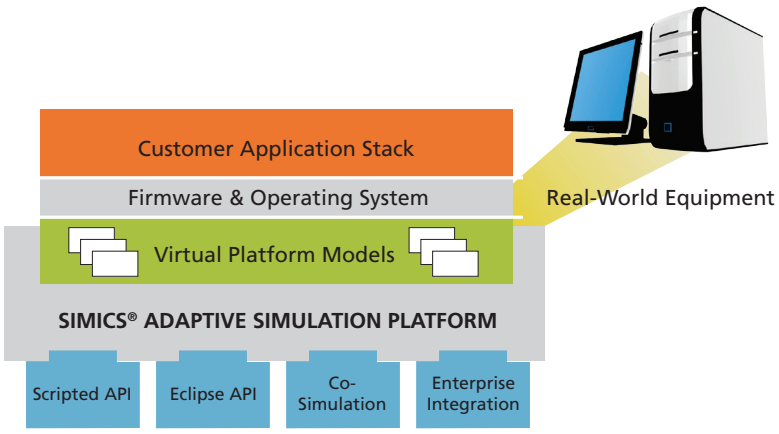

Figure 1: Simics provides a simulation environment based on virtual platform models.

A LOOK AT INCISIVE SOFTWARE EXTENSIONS

Incisive Software Extensions is part of the ESL Option to the Incisive Enterprise Simulator. The Incisive Enterprise Simulator is a multi-language, multi-level functional verification solution that extends RTL verification to the TLM level.

Initially aimed at hardware/software co-verification, Incisive Software Extensions gives the verification testbench access to software executing on processor models. Using the Incisive GUI, the verification team can control and verify system-level behavior including software processes, function calls, and variables, along with processor and register behavior. Incisive Software Extensions brings the constrained-random stimulus generation and MDV provided by the Incisive family to the embedded software portions of a design.

Incisive Software Extensions works with processor models in any form. It allows fast processor models to co-simulate with fast TLM 2.0 models (Figure 2). Incisive Software Extensions provides post-process software debug with no need to re-run software, and lets users trace backwards and forwards. If a system-level verification run fails, Incisive Software Extensions can show what was happening in the software when things went wrong.

When Incisive Software Extensions is used within a complete MDV solution, Incisive Enterprise Planner creates a verification plan containing the goals for functional verification closure. The plan can include software functions, input parameters, variables, and data structures. This plan is used to measure progress and guide verification teams in making the most effective decisions to reach closure with an optimal resource utilization. Metrics are collected and used to guide the verification effort.

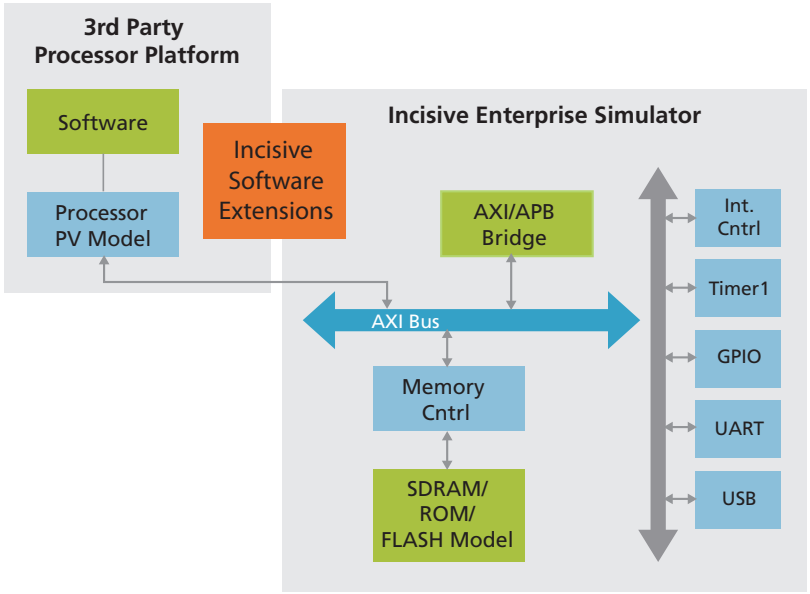

Figure 2: Incisive Software Extensions links the Incisive Enterprise Simulator with software running on processor models.

With Incisive Software Extensions, users can build software tests that automate stimulus generation. Incisive Software Extensions doesn't generate software randomly, but it permits constrained-random variability in parameter values. Software tests can randomly vary sequence and timing within desired constraints, and can react to hardware states. Incisive Software Extensions can monitor hardware and software in a combined manner to check across domains and provide combined cross-functional coverage.

Incisive Software Extensions has an embedded software configuration that gives SimVision, the Incisive GUI, software debug features such single-stepping, breakpoints, stepping forward and backward, signal tracing, and probing.

INTEGRATING SIMICS AND INCISIVE SOFTWARE EXTENSIONS

On a technical level, the integration of Simics and Incisive Software Extensions is straightforward. Incisive Software Extensions has been extended to read Simics function calls and to understand the specific needs and uses of virtual platform environments. As a result of the integration, Incisive Software Extensions non-intrusively accesses the virtual platform. Prior to the integration, it would have been necessary to create an Incisive Software Extensions agent that would have consumed memory and network bandwidth.

Simics has not been modified in any way, and no additional software is required to take advantage of the integration. All that's needed is Simics and the Incisive Enterprise Simulator with ESL Option.

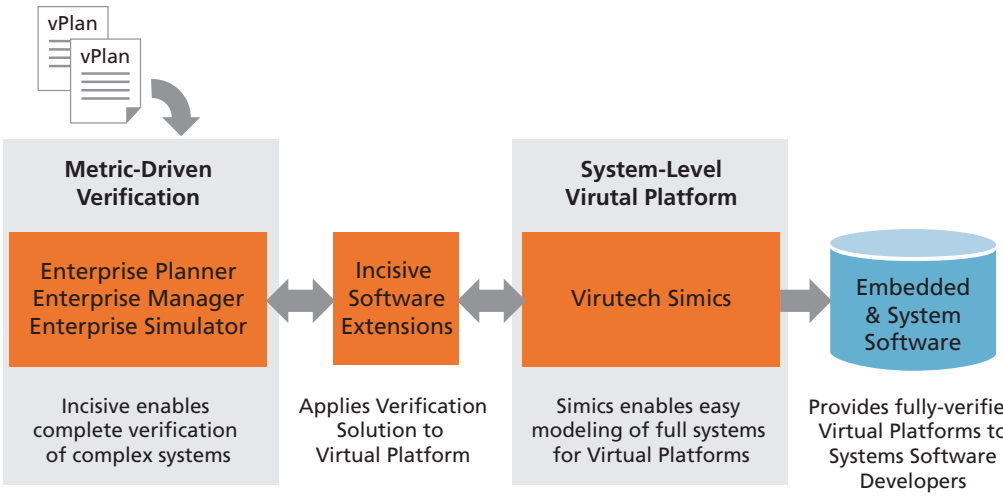

Figure 3 – The integration of Incisive Software Extensions with Simics brings verification planning and metric-driven verification into the virtual platform environment.

The integration brings verification planning and management, embedded software tracing, and MDV to Simics users. It provides stimulus generation and coverage measurement for software development and virtual platform verification.

Simics and Incisive Software Extensions are highly complementary. While Simics provides strong features for non-intrusively isolating and identifying bugs, Incisive Software Extensions' constrained-random testing makes it easy to provoke bugs. Users can work within the Incisive GUI to generate stimulus, track coverage, set breakpoints, and step through code. Meanwhile, Simics lets users snapshot, or checkpoint, any state of the virtual platform for storage and for later replay.

Who benefits from this integration? First of all, software developers using Simics can move beyond directed tests, which find only what one is looking for, and employ constrained-random testing to find unexpected corner cases. They can develop a verification plan that determines the items that need to be exercised and monitored. And they can set measurable goals and track those goals through coverage metrics.

Secondly, hardware developers and systems architects can test the hardware/software interface, and can quickly identify whether errors are occuring in hardware or software. This provides more confidence that the system they have architected is functioning in accordance with system specifications.

Third, any verification plan or testbench developed with Incisive Software Extensions can provide an excellent starting point for RTL verification engineers. With the first TLM design, the verification testbench developed through Incisive Software Extensions can be used with only minor enhancements, and then with further refinements, it can be used in the RTL design. In this sense, the virtual platform environment flows easily into the chip design flow, beginning at the TLM level and extending down to RTL.

CONCLUSION

The Cadence-Virtutech collaboration addresses two key issues – quality and time-to-market. By bringing a more rigorous, formalized methodology to software developers, it helps assure system and software quality. By bringing more capability into the virtual platform environment, and linking virtual platforms into the IC verification flow, it helps speed time-to-market. The end result is not only a verified virtual platform, but a higher quality end product.

Reference

1 *"Project management is all about planning and execution," Chip Design Magazine, http://www.chipdesignmag.com/display.php?articleId=264.*

For more information contact Cadence sales at:

+1.408.943.1234

or visit:

www.cadence.com/contact_us

cadence®

Cadence Design Systems, Inc.

CORPORATE HEADQUARTERS
2655 Seely Avenue
San Jose, CA 95134
P: +1.800.746.6223 *(within US)*
+1.408.943.1234 *(outside US)*
F: +1.408.943.5001
www.cadence.com

21017 05/09 RG/DM/SB/PDF