Panel

Your Core – My Problem? Integration and Verification of IP

Chair: Gabe Moretti, EDN Magazine
Organizers: Nanette Collins, Dave Kelf, Gabe Moretti

Abstract

As the popularity of reusing existing designs — or Intellectual Property (IP) — continues to grow, design challenges escalate. The most time-consuming and critical part of IP design and reuse is verifying that it will work as it was designed to and as the user intends. Designers are pushing the limits of IP for new, distinctive and innovative applications. With this innovation come problems that need creative solutions. Product verification, for example, will become more and more important in ensuring the correctness of the design. Over the years, various solutions have come on the market, all seemingly useful, but none reducing the time or manpower it takes to verify the design. With designs becoming increasingly more complex with each new project and verification consuming up to 70 percent of a design cycle, something must be done to alleviate the bottleneck.

Most of today's verification techniques rely on old simulation and emulation technologies, combined with add-on products designed to target specific functional items facilitated by the increased importance of the functionality they provide. These environments have led to an overall degrading in productivity, with a decrease in tool speed and a sharp rise in learning curve and installation issues. In addition, interaction between add-on products created in isolation lead to further complications, usually discovered as products are incorporated in design flows.

An improved verification flow is required to provide high-level productivity improvement over the entire design. The larger and more complex the design, the higher probability of errors slipping through the verification process, making System-On-Chip (SOC) devices the most vulnerable. With the integration of entire systems within single chips, the need to test hardware and software before the circuitry is produced, within as natural an environment as possible is critical to ensuring design success.

The most important aspect in the selection and verification of IP is the collaboration of the vendor and the foundry. Designers need to be able to evaluate the core before a final selection is made. The evaluation should not just use the testbench provided by the vendor, but should provide an indication of the behavior in the intended use environment. In parallel, the designer must look at fabrication option by obtaining its fabrication profile. How many foundries have certified the core? How many times has the core been used in previous designs from each foundry? And, more important, is the core certified by the foundry chosen for the ASIC under development? With the microprocessor or microcontroller IP, the designer might have projected the use of an off-the-shelf RTOS. In this case, it is imperative to make sure that either the IP or RTOS has been used successfully already, or that the software vendor and the hardware vendor are committed to insure proper integration in a timely manner.

Most of this work is tedious and costly because it must take place before final selection and business negotiations can take place. Once the IP has been chosen, user and vendor must work as a team during the verification process. The vendor has an interest in this process, since bugs can be found when a new set of tests and a new use methodology is available. Those who assume that IP commerce is similar to standard parts commerce are mistaken and are apt to encounter serious obstacles to IP integration.

An SOC must be verified at various levels of abstraction: functional to check the correctness of system design, RTL to insure that what is going to be synthesized is correct, and gate for timing accuracy and possibly to check signal integrity and power consumption. Many verification improvements have been proposed which target specific parts of the design flow. The panel will look at various alternatives for IP verification today, including universal simulation, hardware acceleration, formal verification and semi-formal verification. Functional verification, used to verify that the implementation meets the specification, requires a significant amount of time because of the large amount of test vectors required to meaningfully test all of the functionality of a complex system, including the interfaces among the various logical blocks that make up the system. Traditional HDL simulation falls short of the required capabilities when Verilog is used because the language lacks appropriate behavioral constructs. Although VHDL provides the required constructs, its popularity is more limited, so C based dialects or new languages are required. Being able to develop a testbench is only the first of many obstacles. The amount of time required to execute a full suite of functional simulations is significant, even if processors speeds continue to increase. Simulation farms are becoming a popular tool to decrease functional verification time.

Moving from behavioral to RTL code is the step in the methodology that has the least amount of tool support. Verifying the RTL implementation of a system is crucial. Given the size of today's designs, traditional simulation methods are running out of steam. Formal verification and hardware acceleration methods are being adopted and improved, respectively. EDA tools implementing formal methods have seen a resurgence in the last year, due to improved user interface as well as algorithms implemented to prove the equivalence between a RTL and a gate-level representation of a system. Emulation is also becoming a more popular method to verify gate-level implementations, although the initial cost of the equipment is still quite high.

Panelists, experienced designers and representatives of EDA tools providers, as well as IP providers, will explore ways to beat the verification bottleneck and to identify the methodology best suited for IP design. They will attempt to answer the question, "What methodology works best for IP Design?"

Panelists/Position Statements

Tom Anderson
Vice President of Applications Engineering
O-In Design Automation, Inc., USA

There are many aspects to designing for reuse, but increasingly verification is a key for virtual components (VCs) that are successful either as reusable cores in a company's design repository or as

commercial semiconductor IP products. There are at least four critical dimensions of this important topic:

- The thoroughness of the stand-alone VC verification methodology by the VC creator
- Communication of the verification methodology from the VC creator to the VC integrator
- Provision for the VC integrator to re-verify at least some aspects of the stand-alone VC
- Provision for the VC integrator to re-verify the VC in the context of the full-chip design

It is clearly critical for the VC creator to fully verify the VC in such areas as specification Verification, System Verification, Performance Verification, Timing Verification, Testability Verification and Silicon Verification. It is important for the VC creator to clearly document the verification process and communicate it so that the VC integrator can intelligently assess the suitability of the VC. The VC integrator may wish to re-run some of the verification steps on the delivered VC to do a sanity check or to ensure that any customizations have not introduced fatal bugs. This last dimension, re-verification of the VC in the full-chip context, is the essence of verification reuse since the VC integrator can leverage the stand-alone VC verification. Many experienced VC integrators argue that it is impossible to have effective design reuse without accompanying verification reuse.

Janick Bergeron

Chief Technology Officer
Qualis Design Corporation, USA

Verification is not a tool problem. It is a process problem. Whether you are a provider or integrator of IP, no magic solution is going to solve your verification problem. Superlog, SystemC, TestBuilder, Vera, Specman, HDLs, are all vehicle for implementing a verification process. They are tools, not solutions. Your success (or failure) depends on HOW you use them, not whether or not you use them. The best verification methodology is to have a plan. Simple as that. Know what you want to verify, and how you are going to determine whether it is functionality correct or not.

Ashish Dixit

Director of Hardware
Tensilica Inc., USA

As an IP core provider, I certainly do not want my customers to ask: Your core…my problem? We strive hard to ensure that when they use our core their integration problems are minimized and that they can use Xtensa with confidence. A singular methodology cannot be used to achieve this goal of robustness and customer satisfaction. We use a variety of methodologies encompassing simulations, formal- and semi-formal verification techniques, checkers and monitors, and hardware acceleration. Use of rigorous methodologies of documented test plans and reviews, directed diagnostics, random diagnostic and configuration generators and co-simulation techniques are essential and require faster simulation methods. But other verification techniques are also employed, for example using formal verification for checking the implementation of instruction extensions in our configurable processor. Providing protocol checkers and other functional checkers/monitors to our customers, who are integrating our IP into their testbench is very useful. It makes it easier for them to ensure that their design is meeting the interface and other constraints of our core. If there is one area that the EDA industry can help us, other than faster simulators, it is with functional coverage analysis. Gathering and analyzing functional coverage of diagnostics is too slow and cumbersome. We need more powerful languages to model complex interactions of logic and tools to do the analysis of gathered data.

Peter Flake

Chief Technical Officer
Co-Design Automation, Inc., USA

The promise of design reuse has lead many electronics companies to consider platform-based methodologies, where large portions of the design are provided from older models or through third-party engineering groups. An initial inspection of this technique suggests the possibility of dramatic reductions in engineering cycles for new products. However, “hidden costs” of IP-based design flows, in particular, verification, bring into question true benefits of this approach. How may IP be leveraged without the impact of extended verification and integration efforts? IP reuse brings an element of uncertainty to the design effort where IP consumers must deal with “black boxes” of functionality. With much of this IP being complex components, effective inter-block communication and verification of the unknown are to issues that must be addressed. Communication mechanisms that allow consumer and producer to leverage an understood interface, coupled with automated verification enabling the IP producer to drive test plans through to the consuming design are critical for these techniques to become mainstream. Mechanisms targeted at solving these issues tend to create bottlenecks in the design flow that slow the overall process. An efficient, long lasting solution will revolve around improvements to the modeling language utilized, enabling communication and verification to be intrinsically included in the modeling effort, while still retaining a sensitivity to legacy code. SUPERLOG provides enabling constructs to target these issues, allowing design to leverage a smooth process without pitfalls of hidden functionality.

Tim Hopes

Engineering Manager Debug and Modeling Solutions
ARM Ltd., UK

Ramesh Narayanaswamy

Vice President of Engineering
Tharas Systems Inc., USA

IP Verification calls for an order of magnitude or more verification than a comparable ASIC Verification. The increase is driven by two factors: the increase in configurations and usage scenarios of a piece of IP, and the need to do negative tests on the reference model to make sure that unexpected usages are flagged by the model. The IP Verification Plan has to start with a specification that covers functional behavior for valid inputs, and also precisely define illegal inputs. The Verification Plan has to cover verification of behavior under valid inputs and checks of illegal inputs. Usage scenario based testing will cover usability and performance. This is in marked contrast to ASIC verification which is largely driven by a single usage scenario and a limited set of valid behaviors. The IP Verification Team should be able to execute their simulation model and their test bench language orders of magnitude faster to cover the increased verification load. The simulation platform should support RTL, test bench code, and assertion checks. The reference model has to be delivered as a high-performance model that include assertion checks. A bonded out core or precompiled FPGA blocks do not solve assertion checking requirements.