Infineon innovates in firmware verification by combining OVM and SystemVerilog

The No. 1 chip supplier to the automotive industry, Infineon worked with Mentor Graphics to pilot a new verification methodology. The result: a better, more structured approach to hardware/software co-verification.

Consider embedded microcontrollers. They rarely receive the attention paid to the latest chips that power desktops, servers, game consoles and the like. However, when it comes to ubiquity, microcontrollers can't be beat. These blocks of processing capability, memory and programmable peripherals are found in a range of products, from power tools to toys. Their reach is in part due to the metronome of Moore's Law, which of course for decades has steadily pushed prices down across the semiconductor industry. Today 8-bit microcontrollers, which account for the majority of all CPUs sold in the world, sell for as little as $0.25 each. Consider that in the early 1970s Intel's 8008, the world's first 8-bit processor, sold for $120, an amount roughly equal to $520 today.

Microcontrollers are niche devices, usually built to execute a small handful of tasks. For example, an engine microcontroller might take input from various sensors and adjust fuel mix and spark plug timing. However, the specificity of these chips does not equate to design simplicity. High-end 32-bit Infineon microcontrollers that drive many automotive applications have as many as 70 distinct IP blocks that must be integrated and verified. And the hardware challenges are only the half of it.

Like all microcontrollers, those designed by Infineon rely heavily on firmware. The firmware is critical, and not just the higher-level code that is closest to the application itself and that usually resides in flash memory. The lower level boot read only memory (ROM) code executes an increasing number of background processing tasks, including bootstrap loading, memory checking and so on.

As is true of the hardware, the firmware itself is increasingly complex. Just a few years back the firmware for Infineon's automotive chips – the Munich, Germany-based company is the No. 1 chip supplier to the auto-

"We were the first guys to adopt this methodology at Infineon, and to show that it works at a relatively complex SoC level. I see a good future for this combined use of OVM and SystemVerilog in the company."

RANGA KADambi
SENIOR STAFF ENGINEER
INFINEON TECHNOLOGIES

www.mentor.com
tive industry – amounted to just a few hundred lines of code. Today the firmware file is 16 kilobytes, and growing larger with each release.

For those writing the firmware, the challenge is a bit like building a plane while flying it. Namely, they are writing software for early-stage hardware that is nowhere near stable. How do you verify something when everything from the individual IP blocks to the overall design is still a work-in-progress? That was the challenge in a recent pilot project to design and verify a power train microcontroller at Infineon in Singapore.

The solution was a layered methodology. The first layer is a standard Open Verification Methodology (OVM) testbench that is used to drive input interfaces using constrained-random pattern generation. The layer also observes outputs, measures functional coverage, and compares the results against expected values, a process known as scoreboarding. A second layer implements a well-defined structure for observing (using the SystemVerilog bind construct) and driving internal nodes in the VHDL design (using SignalSpy™, a technology within the Mentor Graphics Questa® simulator).

“We believe this combined approach will be more widely used in the future,” says Ranga Kadambi, Infineon senior staff engineer. Kadambi, who has been with the company for 12 years, is currently, leading a team of functional verification and design-for-test engineers in Infineon’s automotive, microcontroller division in Singapore.

**Starting from scratch: time intensive but ultimately worth it**

Kadambi and his colleagues work on Infineon’s TriCore architecture, the first single-core 32-bit microcontroller-DSP architecture optimized for real-time embedded systems. TriCore unifies the best of three worlds - real-time capabilities of microcontrollers, computational prowess of DSPs, and highest performance/price implementations of RISC loadstore architectures.

When building the new testbenches with OVM, the goal was to use the same firmware verification methodology the team used in e, a verification language developed by Cadence and approved in IEEE Standard 1647. Kadambi and his colleagues chose to start from scratch rather than migrate portions of the e testbench to SystemVerilog because it did not have an e Reuse Methodology (eRM) compliant testbench. Additionally, it would not be easy to migrate from e to OVM because of fundamental language differences. This also gave the team the opportunity to make all of the OVM verification components (OVC) more structured, a contrast to the former e environment.

Building an OVM testbench from scratch certainly takes a bit of effort. There's no getting around the work to understand the OVM technology and its guidelines, and then make its firmware verification methodology fit accordingly. However, as the project progressed, Kadambi and his collaborators became convinced that the OVM methodology and technology were worth the initial ramp up. The main benefit: OVM enables a structured approach that lends itself to reuse.

This effort to learn OVM took place against a backdrop of increasing time and resources required to verify firmware in general. Five years ago, verification of automotive Infineon microcontrollers took no more than four man-months. Today, Infineon spends twice as long, largely due to rising complexity.

Even seemingly simple tasks can be confounding. Take, as a hypothetical case, firmware written to toggle a particular port. It should be straightforward enough to verify the code and check the ports that are toggled. But what happen when there are additional conditions, as is inevitably the case?

“Perhaps the firmware reads the counter value from another address and is coded to toggle every set number of
cycles," says Eric Eu, Infineon senior verification engineer who focuses on firmware verification for the company’s 32-bit TriCore microcontrollers. "And maybe there’s input from another pin that tells the code whether the counter should be reset or just stopped with each toggle. Verifying all this functionality at the design stage is difficult, especially with unverified underlying hardware."

Layered approach

The design under test (DUT) is mainly coded in VHDL with some IP blocks coded in Verilog. The DUT is instantiated by a VHDL top-level testbench used for SoC verification. The SystemVerilog/OVM top-level is instantiated under this VHDL top-level.

The first layer of the OVM test environment consists of an interface layer for observing and driving signals into the DUT. Firmware verification differs from the conventional bus functional model (BFM) because Kadambi and Eu are mostly interested in whitebox testing. Instead of a BFM model, the team used a signal map, a collection of internal signals of interest during verification. The signal map implements methods for observing and driving internal signals. In this project, the used the SystemVerilog bind construct to observe the internal VHDL signals and the Mentor Graphics Questa SignalSpy technology for driving them.

The second layer consists of the Open Verification Components (OVCs), encapsulated and reusable components that follow consistent architecture and communication channels. The team uses TLM analysis ports and TLM analysis fifos for the OVC interconnections. TLM analysis ports provide simple and powerful transaction-based communication because of their ease of implementation, support of multiple connections, and execution in the delta cycle.

OVCs are critical in helping the team to deal with large numbers of IP blocks. Each such block more or less maps to a corresponding OVC, and together these OVCs interact and cross check at a high level in such a way as to hide the lion's share of the complexity. If a future Infineon product incorporates a new or replacement block, the team simply needs to add or swap out one OVC. Given the modular nature of OVC and of SystemVerilog in general, the rest of the stitched-together design can be left mostly as is, a boon to the design team. Modularity is also a bit unusual in an era in which complexity often hides interdependence and tugging on one loose thread too often causes an entire digital fabric to unravel.

Finding bugs while laying foundation for future

Verification methodologies must prove their mettle by finding bugs, and Infineon's from-scratch approach did just that. The team found 12 firmware bugs and five hardware bugs using the OVM for firmware verification. Common firmware bugs were the result of the implementation not meeting specification (these were detected by assertions) or implementations that did not cover all possible scenarios in the firmware (detected by random stimulus generation and coverage). Firmware verification quite often also detects hardware bugs (through assertions) caused by registers that are not writable or readable because either their protections are not set correctly in the RTL or their top-level connections are incorrect. Most significantly, the team hit verification targets related to functional coverage and code branch coverage. The latter is a methodology that executes both trunk and branch blocks of code, a technique that helps to deal with multiple revisions, a fact of life in all software development.
Of course, Infineon’s success came at a price, particularly in terms of time. In the past for a project of this scale, the company would generally spend six to nine man-months on the firmware. The OVM-SystemVerilog pilot took 10 man-months.

"One reason is that OOP does sometimes require more lines of code, compared to AOP (aspect oriented programming), though the extra lines of code required by OOP enable us to reach our primary goal of a more structured approach," says Eu. "Importantly, whereas our former environment was not well structured, OVM infrastructures are. Furthermore, the compile issues inherent to AOP initially required an effort greater than that required to write the extra lines of code. On subsequent projects, the amount of effort and workarounds associated with the OVM should also decline."

Eu and Kadambi worked extensively with Mentor Graphics throughout the pilot project. The catalyst was a one-day training on OVM and SystemVerilog delivered by Mentor’s Christoph Suehnel in China in fall 2008. And when the project began in earnest in spring 2009, the Munich-based Suehnel flew to Singapore and worked side-by-side with Eu and Kadambi for three weeks while the two engineers were ramping up.

"Overall Mentor has provided a very classy level of support and Christoph's work, in particular, was essential; indeed, he is a major reason we achieved first silicon success with this methodology," says Kadambi, who is already looking ahead to the future.

"We were the first guys to adopt this methodology at Infineon, and to show that it works at a relatively complex SoC level," he says. "I see a good future for this combined use of OVM and SystemVerilog in the company."

Ranga Kadambi, Infineon Senior Staff Engineer, on Mentor Graphics Verification Technologist Christoph Suehnel, who provided consultation and training at the start of the pilot project.