www.scdsource.com
Sunday, June, 1st 2008

In My Opinion

Why today's virtual platforms aren't the answer

By Larry Lapides, Imperas

04/02/08

We've been hearing for a while now that virtual platforms are the answer to system-on-chip (SoC) embedded software development problems. But, what can yesterday's virtual platforms — *hardware* virtual platforms that run at 20 MIPS top speed and cannot handle multicore architectures without slowing down another 10X — do for embedded software development?

If these hardware virtual platforms were the answer to software development problems, companies providing this technology would be doing quite well. But this technology has failed to address the real needs of end users. Why hasn't the technology lived up to its potential? Is it due to lack of speed and model interoperability, or insufficient infrastructure to tackle growing system-on-chip (SoC) complexity, or proprietary languages? Let's include SystemC in the list of proprietary languages, because language "flexibility" coupled with a lack of results from SystemC committees has resulted in different flavors of SystemC for each and every user.

SoC embedded software development teams want to know how they can make the software verification process more efficient, more complete, more predictable and more measurable. This is the challenge platform vendors should be addressing. The answer is that verification, debug and analysis combined with a verification methodology, and built specifically for embedded software development, can do that.

This focus on verification is especially important for multicore SoC architectures. With more concurrency coming to embedded software, the software state space is increasing exponentially. This means more bugs, more simulation cycles to identify bugs, and more time required to track down the source of each bug. This can't be done with hardware virtual platforms that were not built for multicore architectures. They cannot support the requirements for speed, design complexity and early verification because of the difficulty of constructing a hardware virtual platform.

Software virtual platforms are a means to an end, with the end being better embedded software development results. That's why there's a movement called Open Virtual Platforms (OVP), an effort to support a software virtual platform technology with a reference simulator available as open source. OVP addresses the shortcomings of the previous generation of hardware virtual platforms by providing:

- An instruction-accurate abstraction level that achieves simulation speeds of hundreds of MIPS — near enough to real time speeds for embedded systems development
- Standard application programming interfaces (APIs) for processor, component, peripheral and platform models that ensure model interoperability
- Open source models that the community can use and add to, but with licensing so that IP developers can protect and commercialize their own IP
- An infrastructure built to handle the complexities of multi-processor SoCs

Everything is provided free, so that what is commercialized are the high-value tools and IP, not the virtual platform. Collectively, the industry is rallying around companies building businesses that are solving embedded software verification problems ─ and it's easier to do that when there is a good platform/simulation infrastructure available to users.

This may look like a virtual platform business killer -- and it is. What it shouldn't do is kill the companies that have been developing *hardware* virtual platforms. With a performance advantage over last-generation virtual platform technology, OVP is an effort to create an ecosystem of users, tool developers, intellectual property (IP) providers and service providers where all can thrive.

That's because virtual platform providers can't do this alone. Working together, with an open source infrastructure that encourages and enables the exchange of ideas and data, as well as quick turnaround on these ideas, the industry can make this happen. The current virtual platform suppliers have service experience, IP and engineering know-how that could be turned to higher margin activities in the area of embedded software verification.

This call for software verification sounds eerily reminiscent of what happened in hardware verification. After the Verilog hardware description language (HDL) was placed into the public domain, the industry saw a 10X speedup in simulation, followed by the development of a verification tools industry. It was a good business model to build on, but wait … there's more. What happened if the chip wasn't well-verified? Millions of dollars in additional expenses were required for a respin, plus more millions of dollars lost by being late to market. These were great motivators to improve verification.

That isn't the case with embedded software development because there is no single, spectacular crash-and-burn event when embedded software doesn't work. Rather, it's the death by one thousand cuts scenario, and is similar to what has happened with FPGAs.

When FPGAs were small, they could be designed, programmed and debugged within hours in the lab. As FPGAs got bigger, debugging took a couple of days. Finally, FPGAs were more than one million ASIC gates, and debugging was taking weeks in the lab. Developers spent and lost millions. The industry realized that there was a perfectly fine simulation and verification technology that could be directly applied to FPGA development.

Embedded software used to take hours to debug. And now? Too much time is spent per bug and there are too many bugs. In the worst case, an embedded software team puts out a new release each week.

What about those software teams that don't spend money to solve problems? They are finding that they're not just on the critical path for SoC shipment — they *are*
the critical path. This path meanders unpredictably and without measure, making management of the embedded software development process impossible. Managers looking at that situation are reassessing their options. Throwing people at problems doesn't solve them. Instead, they are applying technical smarts and know-how to make their teams more efficient with the goal to get high-quality products delivered on time.

And that is where our odyssey ends. SoC developers, embedded software developers, tool companies, IP providers and service providers are pulling together toward a common goal. These companies include Azul Systems, Beyond Semiconductor, Calypto, Carbon Design Systems, CriticalBlue, Denali, ElementCXI, EVE, Forte, Imperas, MIPS Technologies, Novas Software, SiBridge Technologies, Sigmatix, and Tensilica. OVP is providing a vehicle for solving embedded software verification problems. Come on and join the party; there's room enough for everyone.

Larry Lapides is vice president of sales at Imperas. Prior to joining Imperas, Larry Lapides was vice president of sales at Averant and Calypto Design Systems, and served as vice president of worldwide sales during Verisity's IPO.

Related articles

Imperas preps multicore virtual platform tools
Imperas donates 'open' virtual platform infrastructure
Virtual platforms – a reality check

All materials on this site Copyright © 2007-2008 Tech Source Media, Inc. All Rights Reserved | Privacy Statement