www.scdsource.com
Tuesday, March, 4th 2008

**First Look**

# Imperas donates 'open' virtual platform infrastucture

**By Richard Goering**

**03/03/08**

Aiming to boost the virtual platform market for software developers, Imperas Ltd. this week (March 3) is launching a set of technology donations called Open Virtual Platforms (OVP). With APIs, models, and a simulator, OVP promises an open, standard infrastructure for virtual platform development.

According to Imperas, OVP-based platforms can meet the needs of embedded software developers with simulations that execute at hundreds of MIPS. OVP-based platforms also support multicore architectures, which was Imperas' original focus. Imperas will support and manage the OVP web site, which provides the following for free download:

- C language APIs for building platform verification infrastructure, and developing behavioral and processor models.
- Open-source library of models for processors, components, peripherals, and platform templates.
- OVPsim, a free reference simulator shipped as an executable.

According to Simon Davidmann, Imperas CEO, OVP is the most important system-on-chip technology donation since SystemVerilog. He also compared the launch of OVP to the opening of the original Verilog language. "Verilog enabled an RTL chip design ecosystem," Davidmann said. "OVP enables multicore software development, and creates an ecosystem around it."

The OVP announcement comes with a long list of supporters, including Azul Systems, which reportedly built an OVP-based platform for its Azul Vega multiprocessing architecture. Other supporters include Beyond Semiconductor, Brian Bailey Consulting, Calypto Design Systems, Carbon Design Systems, CriticalBlue, Denali Software, Element CXI, EVE, Forte Design Systems, Jennic Ltd., MIPS Technologies, Novas Software, Sigmatix, SiBridge Technologies, and Tensilica.

Notably absent from the list of supporters are commercial virtual platform providers including ARM, CoWare, Synopsys, Vast, and Virtutech. This could be because Imperas, which has yet to announce a product for purchase, has signaled its intent to enter the virtual platform market in order to support multicore architectures.

OVP was not Imperas' original intent. The company was launched with the goal of providing a new software programming paradigm for multicore platforms, along with tools that would allow users to write, optimize and map parallel software to heterogeneous processors. While this remains a longer-range goal, Imperas shifted its focus late last year to virtual platform tools for prototyping, simulation and debugging.

But first, Davidmann said, Imperas discovered that it needed to build an underlying modeling and simulation infrastructure. "We didn't want to build simulators when we started Imperas four years ago, but we couldn't find anything out there that met the requirements for hundreds of millions of instructions per second and was easy to use," Davidmann said. "Quite frankly I'm surprised that four years later we're giving it away, but it's precisely what's needed."

Imperas is giving away technology that other virtual platform providers sell. So how can Imperas make money? The answer, said Larry Lapides, Imperas vice president of sales, is by selling value-added tools that sit on top of the OVP infrastructure. These will include "productivity tools" along with verification, debug and analysis tools, he said.

"I think it is exciting, but I think they have a long way to go to actually show success," said verification consultant Brian Bailey. OVP, he said, "has to be considered a very good seed for the industry to define a whole new level of virtual prototype." Bailey said OVP and its APIs should help foster model interoperability, which is vitally needed now in electronic system level (ESL) design. Hopefully, he said, people outside Imperas will write models that comply with those APIs and make the models available.

Imperas isn't giving away the store, Bailey noted. Imperas is giving away the executable of the simulator rather than the source code, and the free OVP simulator doesn't include all the simulation technology Imperas has. "It doesn't include the debug capabilities and it doesn't contain many of the other things you'd like to see in a full product," Bailey said.

## Running too slowly

OVP was born of the conviction that current virtual platforms, especially those based on SystemC, are much too slow for embedded software development. "Mostly people are focused on hardware virtual prototypes that are timing and cycle-accurate, often in SystemC," said Lapides. "The models are complex, slow, and time consuming to build."

SystemC, said Lapides, is "RTL for free." It's a hardware-centric language that's not used for software development, he noted. Rather than SystemC, OVP focuses on instruction-accurate C models.

"If it's used for software development, it needs to be a couple of orders of magnitude faster than a hardware virtual platform," said Davidmann. "Today there is no standard way of building models that are fast enough for software." Davidmann noted that SystemC models run at a few MHz, which may be okay for developing small amounts of software at the detailed hardware level. But engineers building operating systems and applications software really need models that run in the hundreds of MHz, he said.

Another problem with today's virtual platforms, Lapides said, is that they're built for single processors. "Software developers do not just have to write code for multi-processor architectures, they have to debug and verify code for multi-processor architectures," he said. "With the single-processor oriented, hardware-oriented virtual platforms out there today, it is really impossible to do leading edge software development."

"Everybody who builds a multi-processor chip today has to build it themselves," said Davidmann. "It's absurd. Every time you build a new chip, you have to get a simulator built by a third party. But that's the situation today."

Yet another problem is the virtual platform business model. According to Lapides, virtual platforms are basically a "services" business today, where providers make most of their money building proprietary models as opposed to selling tools. Further, the models aren't interoperable across different tools.

"Each service provider has a different way of building processor models and platforms, and they're completely incompatible with other tools," Davidmann said. "It's like where hardware was when everybody had different simulators in the 1980s. Everyone was doing things slightly differently. You can't get the benefits unless you coalesce around a similar methodology."

Virtual platforms are the enablers for multi-processor systems-on-chip, "but we don't have a good solution in the market today," Davidmann said. "Nothing out there is targeted at the right level."

## Building an infrastructure

What's needed, said Davidmann, are "common representations and ways of doing things." Imperas, he said, will "make public and give away the technologies we believe are needed to establish an open standard platform for virtual prototypes. And we're not just giving it away and going off and doing other things – we'll commit to supporting and managing it."

The real money-making opportunity, Davidmann said, is at the "higher level" of virtual platforms, but first there must be some solid building blocks underneath. With no simulator, ecosystem, or common ways of doing things, the virtual platform industry will "never get off the ground," he said. What is really needed, he said, is an ecosystem with multiple tool providers "just like in the HDL world."

"We believe we can be successful selling tools, as people need software verification and sophisticated debug for multi-processing," Davidmann said. "That's the exciting area for us. We see simulation as something that just needs to happen."

But if virtual platforms are created, will software developers – who are notoriously cost-sensitive – purchase and use them? Davidmann thinks so. "The software world is so complex and the challenges are so difficult you've got to use simulation," he said. "When you have three or four processors, you really have to have simulation."

Part of the OVP initiative involves the publishing of C OVP APIs for processor, peripheral, and platform modeling, along with their documentation and header files. The APIs are called from C/C++ including, if desired, SystemC. A peripheral modeling API lets users model behavior for peripherals. A platform modeling API makes it possible to instantiate different components in a platform. And a processor modeling API lets users describe the behavior of processors in terms of their operations.

Open-source models provided include ARM, MIPS, and OpenRisc OR1K processors, with more to come later. Imperas also provides models of standard embedded peripheral devices. On average, models run at around 200 to 300 MIPS, with some peak speeds ranging up to 1.2B instructions/second, Davidmann said.

The models can run in SystemC environments, but you won't get the performance benefits, Davidmann said. "The problem is that the SystemC bus runs at about 1 MHz," he said. "You put the SystemC scheduler in there, and you get about a gazillionX slowdown."

Third parties can write models to the OVP APIs, but it's up to them whether or not to include the models in the open-source library. "Ours are fully open source models," Davidmann said, "but you don't have to put your source code back into the public domain." Imperas models are available under the GNU Lesser Public License (LGPL).

The free OVP reference simulator, according to Imperas, runs processor models up to 500 MIPS, interfaces to the GDB debugger, is multi-processor capable, is callable from a SystemC wrapper, and can encapsulate existing instruction accurate processor models. "Multi-processor capable" means that two processor models running concurrently can share resources, Davidmann said. The simulator is available with a click-through license.

## Third-party support

The OVP announcement comes with a quote sheet from some 16 companies. Representatives of several spoke with SCDsource.

"I think it's a great idea to have an open platform for system-level, embedded software verification," said Scott Sandler, Novas Software CEO. "This area is one that's quite fragmented and closed today. As this [the OVP] emerges, we expect to add value by supplying advanced debugging tools that work in this context."

"The OVP announcement gives our customers and potential customers a high-speed, debuggable method to quickly generate platforms into which they can plug their Carbon models," said Bill Neifert, CTO of Carbon Design. "We will be adding OVP to our list of supported platforms."

OVP's biggest potential challenge, Neifert said, will be the "orphan model problem." He questioned whether customers will want to develop models that have no connection to system implementation. SystemC carries a stigma of being too hardware-oriented for software engineers, but it's a valuable tool for architectural exploration and hardware creation, and is used increasingly for software development, he said. "The lack of an innate connection to implementation is both OVP's greatest strength and also its greatest weakness," he said.

"The virtual prototype market is fragmented now," said John Sanguinetti, CTO of Forte Design. "I believe Simon [Davidmann] is right when he says that the value is not in the platform infrastructure itself, but in whatever tools can be built on it, and the models that can run on it. By providing an open standard infrastructure, it should be a lot more efficient for people to do things that actually add value."

If the third-party quote sheet is any indication, OVP does appear to be sparking the development of a multi-vendor ecosystem. And that's what it's really all about. "To develop software efficiently, you need an ecosystem," Davidmann said. "We are all destined to fail if something like this doesn't happen."

Related articles

Imperas preps multicore virtual platform tools
Virtual platforms – a reality check
Virtual platforms – a reality check, part 2