

Posedge Software, Inc.

Web: www.posedgesoft.com
Email: info@posedgesoft.com
Phone: (612) 548-4784

Posedge Software has expertise in the listed areas, and provides design and consulting services for the development of virtual platforms, embedded software, and related tools.

Virtual platforms (or virtual prototypes) are valuable tools for hardware and software development. They provide an environment for running embedded software earlier in the development cycle, or when using real hardware is not an option. They also provide more precise control and visibility into the target hardware which allows for many other capabilities, including automated testing, verification, and advanced debugging.

A variety of skills and knowledge are needed to take full advantage of virtual platforms and implement cutting-edge solutions, including the following:

- **Modeling of hardware at various abstraction levels using hardware description languages**
- **Embedded software, including compilation tools, debuggers, and target operating systems**
- **Virtualized software execution and other simulated/emulated hardware techniques**
- **Electronic design automation, including modeling and verification languages, methodologies, and tools**

VIRTUALIZING HARDWARE

There are many benefits to using virtual instead of real hardware. A significant advantage is that it can be available earlier in the development cycle, before hardware is ready, or even while the design is still being finalized. Even when the actual hardware is available, a virtual platform can be made available more easily to a larger number of developers or customers.

Hardware can be modeled at various abstraction levels. However, embedded software often does not need a very precise timing model. This is particularly the case with modern hardware, where multiple cores, and many levels of cache, introduce significant variability to the timing.

MODEL BUILDING

While a typical Instruction Set Simulator (ISS) will just model the processor, virtual platforms usually model the entire set of peripherals. This enables running of the complete application without modification, and allows for testing of drivers and other low-level code.

A variety of methods can be used to construct processor or peripheral models. They can be written by hand, or tools can be used to generate them from the specifications. Existing IP or simulation models could also be integrated into the platform.

PORTING THE OPERATING SYSTEM

A real-time operating system (RTOS) or other system kernel is often used in embedded designs. This can simplify firmware design and abstract away many of the low-level details. However, getting an OS to boot on new hardware can be a difficult, time-consuming process. Virtualization eases this process by providing greatly improved visibility into both the hardware and software of the running system, even when fatal errors occur. Booting an operating system also requires execution of a significant amount of code. While simulation time is a major concern with RTL or other simulations, modern virtual platforms can often match speeds of the actual hardware.

TESTING THE TARGET APPLICATION

A primary use of the virtual platform is testing of the target applications. This may require running individual unit tests on each module, or running the entire system-wide application. Having increased visibility into, and control of, the platform greatly assists with this. Assertions or other checks can be added to the platform to monitor for significant events without modifying the source code. If a problem is encountered, additional debugging capabilities are available that could not easily be done on real hardware, even with a hardware debugger or other mechanisms. Runtime info such as coverage and profiling can be valuable during the software development cycle. While it usually requires recompiling or making other changes to the software, virtual platforms can be used to do this transparently, without modifications to the target software.

MULTI-CORE DEBUGGING

Multi-core systems provide additional challenges, especially when trying to test and debug problems. A virtual platform can assist with this by simultaneously providing visibility into all the cores, and allow for synchronously stepping the entire platform. In addition, shared memory, interrupts, and other synchronization mechanisms can be monitored.

AUTOMATED SOFTWARE TESTING

Regression testing is often used in the software world. However, for hardware (where the cost of fixing a bug can be significantly higher) numerous tools, languages, and methodologies have been developed specifically for testing and verification, such as constrained-random testing and coverage-driven verification. A virtual platform can be used to extend many of these techniques to the target software and provide a robust, flexible verification environment needed to handle the complexity of current and future designs.

CO-VERIFYING HARDWARE AND SOFTWARE

Besides writing tests independently for the hardware and for the software (as is usually done), a virtual platform can be used to test the interactions between them. Often, this testing does not occur until very late in the development cycle, once hardware has been finalized. Tests can instead be written to check how both hardware and software respond to an event. This co-verification is also useful in testing corner cases, and other cases that would be difficult to test using real hardware.

SYSTEM-LEVEL VERIFICATION

While simulations are often used as part of a

Supported Tools and Languages:

- * C/C++/SystemC and Verilog RTL (both commercial and open source tool flows)
- * Transaction Level Modeling with SystemC/TLM2
- * Imperas M*DEV, M*SDK, and related OVP tools
- * Cadence Virtual System Platform (VSP)
- * Xilinx Zynq EPP (using Cadence VSP)
- * Scripting and automations using TCL, Perl, Python, and other languages

system-level verification environment, traditional methods often result in limited performance or functionality, or require expensive specialized hardware. In contrast, a modern virtual platform should allow the complete software stack to be run unmodified, at very high performance levels on standard hardware. This allows for the complete algorithms and data interfaces to be tested together, and for more tests, and greater amounts of stimulus to be run against the platform.

Founded in 2003, Posedge Software provides tools and services for the EDA and related industries. With expertise in hardware modeling, software development, and system verification, Posedge Software can deliver custom, state-of-the-art virtual platforms and prototypes to meet the customer's needs. With connections to leading vendors in the industry, Posedge Software can provide guidance when selecting a simulation framework, or can make use of other available solutions.

Partnerships

Cadence Connections System Realization Alliance and Verification Alliance Member

Imperas Development Tools Service Provider

Open Virtual Platforms Contributor

Xilinx Alliance Program Member

