Improving the embedded Intel® architecture design process with Simics

White Paper

Tian Tian
Sr. PAE

Intel Corporation

November 2011

326341-001

Executive Summary

Embedded system design can be a challenging and lengthy process due to complexity introduced by system, hardware and software interactions and unique design constraints. The products often use non-standard components, adopt unique form factors, and have the requirements to sustain/validate customized OS/software/drivers. The products often have a very long life cycle compared to desktop/laptop counterparts. Some products are required to run non-stop for many years without any maintenance shutdowns. This paper introduces a full system simulation technology that can help embedded IA designers tackle these problems. The technology can improve all phases of product life cycle and help designers take products to market sooner with higher quality, and allow product teams provide better sustaining support many years after launch.

Wind River Simics (Ref [1]) is an industry-proven commercial solution for full system simulations ranging from one component to one board to many connected devices in a network. This versatile and powerful technology can contribute to multiple phases of the product life cycle from concept to system integration/test to customer support. The technology is professionally supported by Wind River and is ready to become an integral part of the embedded IA system design methodology.

Simics has a wide range of adaptability and proven a track record in many market segments including networking, telecom, high-performance server, industrial, and aerospace & defense. In this paper, we focus on the impact of Simics to the embedded IA space. We walk through several classic usage models, and highlight practical situations that engineers

may use on a daily basis. As proved by years of commercial success, this technology can help designers improve product quality and reduce time to market. It is also positioned to become one of the primary weapons in tackling an increasingly complicated design process associated with the next wave of the system computing paradigm.

The Intel® Embedded Design Center provides qualified developers with web-based access to technical resources. Access Intel Confidential design materials, step-by step guidance, application reference solutions, training, Intel's tool loaner program, and connect with an e-help desk and the embedded community. Design Fast. Design Smart. Get started today. http://www.intel.com/p/en_US/embedded.

Contents

Simics Overview

Simics (Ref [1]) is a full-system simulation technology that has been used commercially with proven success for more than a decade. It is a versatile technology that is capable of simulating digital systems as small as an individual device (e.g. CPU, interrupt controller, memory) or as large as rack-based large scale systems that include thousands of processors or even the network. It has been used by a wide range of market segments including telecommunications, networking, aerospace, defense, high performance computing and other emerging embedded segments.

The idea of simulating hardware is not a new concept. There have been many simulation solutions used in many companies internally in software and driver development, system validation, or prototyping. These kinds of solutions in general are closely coupled with internal hardware specifications and are available to internal development teams only. Simics is unique in its approach and takes this common concept to the next level. First, Simics targets software and system developers. This means that the simulated target system must run the complete software load from BIOS to RTOS to complete application and must run it fast enough that developers can use it for real work instead of the physical hardware.

Simics simulates the functional behavior of the target hardware. This enables the target software (same BSP, firmware, RTOS, middleware, and application) to run on the virtual platform the same way it does on the physical hardware. The simulation is fast enough that it can be used interactively by developers. With Simics, engineering, integration, and test teams can use approaches and techniques that are simply not possible on physical hardware. For example, developers can freeze, save, email, and restore the whole system; they can view and modify every device, register, or memory location, and they can run the whole system in reverse to find the source of a bug.

The Wind River Simics website (http://www.windriver.com/products/simics) is a great resource, which contains overviews, videos, white papers, and key contacts. An extensive set of technical documents is distributed as part of Simics package.

Simics Intel® Architecture models

Simics is widely used by Intel to develop and improve next generation solutions. The technology is becoming an integral part of the design process in various phases such as prototype, test, sustaining etc. Simics IA models are being tested and improved through engineering usage. As more and more Simics IA models are made available in Simics, customers and embedded system designers will have more IA building blocks to design their products. For the current list of released Simics IA models or early access Simics models for next generation solutions, please contact Wind River (Ref [1]).

IA customers can take advantages of this technology, use the virtual environments, and gain expertise in writing and developing their own device models and become owners for their own ingredients. The technology allows customers to use or create a virtual platform that they can use to interact with real OS, BIOS, drivers and other ingredients (Figure 1). As highlighted in the Simics architecture overview (Figure 1), customers can contribute their own models into Simics, run software applications on top and have this virtual system interact with other real equipment or other software components.

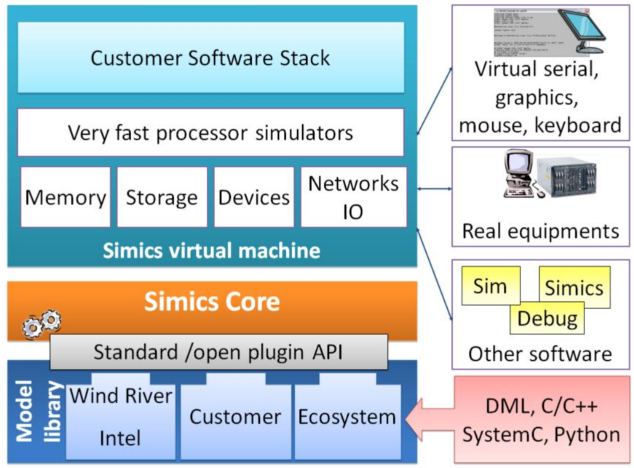

Figure 1 A simplified view of Simics Architecture

Classic Simics usage models

As a commercial solution, Simics has established many successful usage models over the years. Users continue to come up with new and innovative ways of using this technology on an ongoing basis. In a whitepaper for software development (Ref [3]), Jakob Engblom summarized multiple classic usage models throughout the project life cycle. These usage models apply to IA designs as well. Figure 2 highlights several of these usage models.

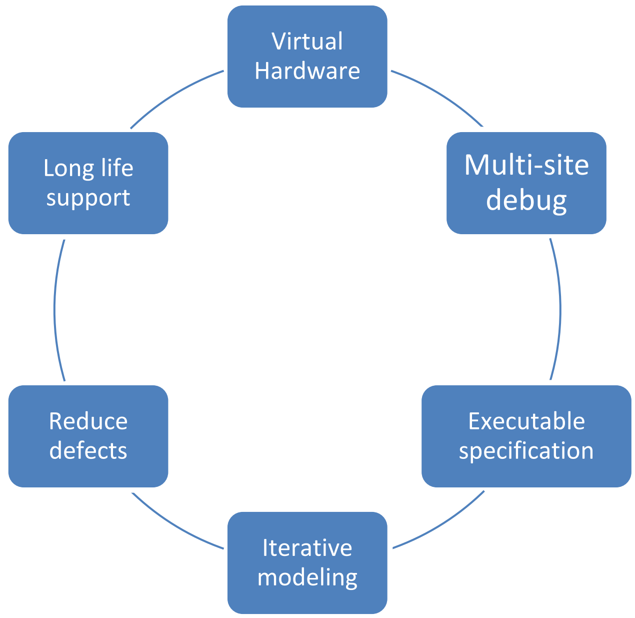

Figure 2 Examples of Simics IA usage models

- Virtual Hardware – allows developers access to hardware, when hardware is not available or very difficult to get access to (pre-silicon, supply issue, access issues (such as physical boards and expertise are located at different sites)
- Multisite debug – allows issues to be captured and transported easily (checkpoint etc.) and transferred to different sites
- Executable specification – allows a design spec to be presented in a simulation rather than on a piece of paper or power point
- Iterative modeling – In the same amount of time, this technology allows designers to tune up designs with full system simulation
- Reduce defects – extensive testing/usage in simulation environment helps detect issues that may be very hard to catch on real hardware
- Long term support – when physical boards are hard to find, this allows technology to positively impact post-launch and long term support phase and continue to bring values to users after product launch

Intel Architecture has a strong software solution base and a huge developer base, the potential and possible interactions among developers, ecosystem vendors, software solutions and Simics environment are virtually unlimited. One thing we know is that we are barely scratching the surface here on what Simics may bring to the IA community, even with so many known usage models.

Examples: from Power Button to OS

In this section we walk through several classic examples to demonstrate the versatility of this technology, starting from the very first moment when the "power button" is pushed in the simulated environment and the BIOS starts running.

Get into BIOS actions

It is possible for the Simics IA environments to interact with production-quality BIOS images in a functionally accurate fashion. The example below (Figure 3) shows the Simics EFI boot log (on the left) versus the boot log on a real matching hardware platform, using the same BIOS image. The boot logs show excellent correlation. From a functionality perspective, Simics is able to simulate the BIOS's interaction and can be a valuable weapon for debugging and testing BIOS.

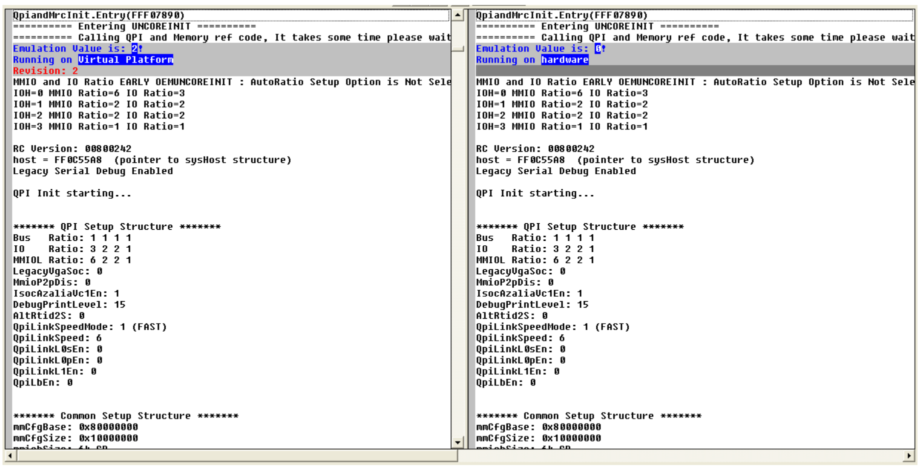

Figure 3 EFI boot log in simulated environment versus real hardware

Single-step through instructions

Figure 4 shows the capability of single-stepping through instructions in a simulated environment. This is easily accessible in Simics through the GUI or command line. This is a nice utility when engineers are trying to locate which instruction is causing issues. In Simics you can even do reverse execution.

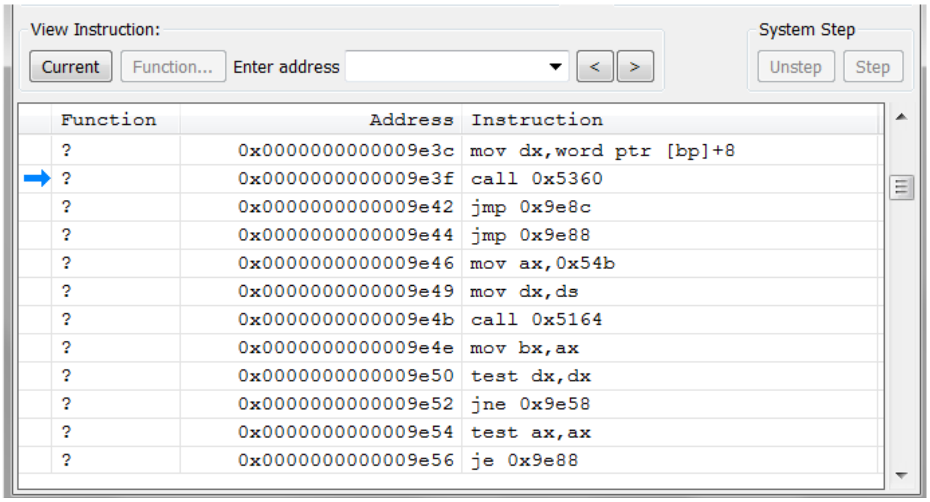

Figure 4 Step through instructions

Probing PCI Express Configuration Space

Figure 5 shows simulated hardware in the PCIe configuration space. This can be captured during BIOS initialization, after BIOS initialization or even after the OS is running. This example demonstrates that users can use this kind of interface to easily access the register configuration space for the devices of interest anytime during the simulation. This can be potentially used to debug BIOS, debug driver, and support customer issues.

Offset	...	Value	Register	Descrip...
pci_config				
0x0	2	32902	vendor_id	
0x2	2	15361	device_id	
0x4	2	0	command	
0x6	2	16	status	
0x8	1	0	revision_id	
0x9	3	394240	class_code	
0xc	1	0	cache_line_size	
0xd	1	0	latency_timer	
0xe	1	1	header_type	
0xf	1	0	bist	
0x10	4	0	no_bar_0	
0x14	4	0	no_bar_1	
0x18	1	0	primary_bus_number	
0x19	1	0	secondary_bus_number	
0x1a	1	0	subordinate_bus_number	
0x1b	1	0	secondary_latency_timer	
0x1c	1	240	io_base	
0x1d	1	0	io_limit	
0x1e	2	0	secondary_status	
0x20	2	65520	memory_base	
0x22	2	0	memory_limit	
0x24	2	65521	prefetchable_base	
0x26	2	1	prefetchable_limit	
0x28	4	4294967295	prefetchable_base_upper	
0x2c	4	0	prefetchable_limit_upper	

Figure 5 Example of PCI express configuration space

Boot into OS

For users that are more interested in getting into the OS and beginning their work there, good news here is that Simics can work with many operating systems. Figure 6 shows simulated hardware boots into the Windows* environment.

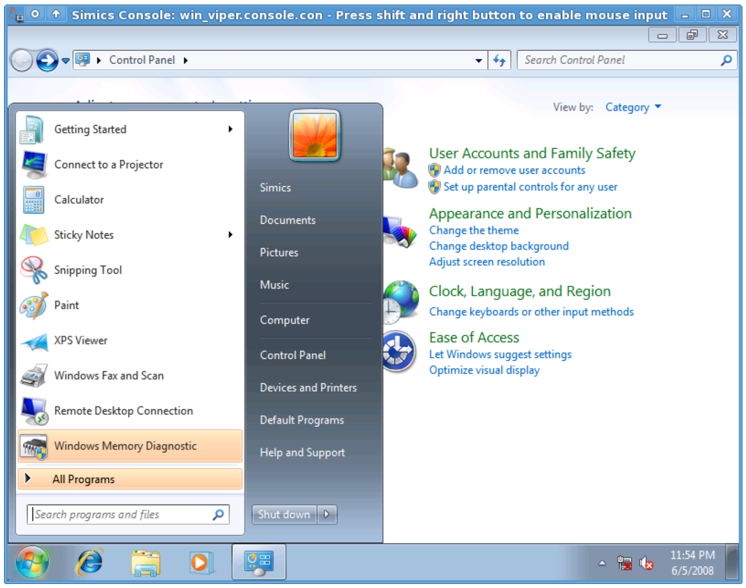

Figure 6 Boot into Windows

With just a simple mouse click, developers can quickly switch to a different simulation using another OS such as Linux*, shown in Figure 7. Now users are ready for some Linux activities!

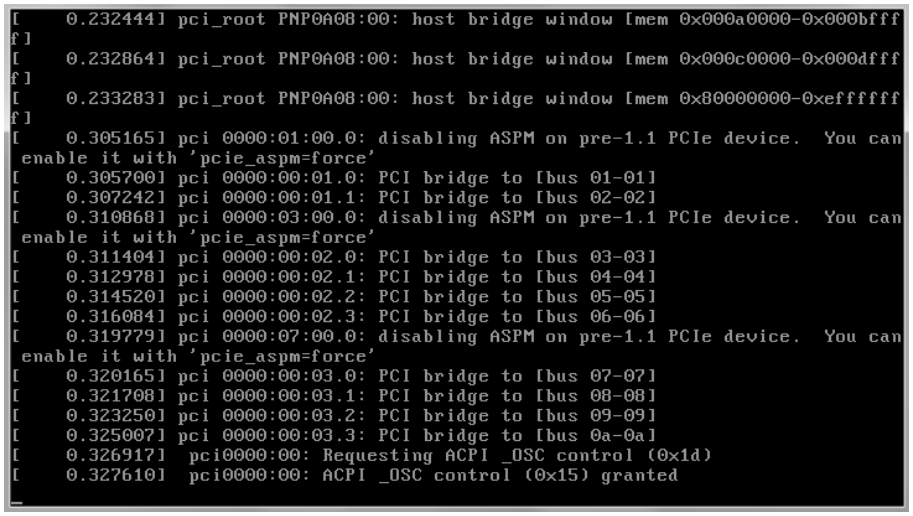

Figure 7 Boot into Linux

Now that we are in the OS, developers are ready to take over and start doing their development work. With features such Simics file system and networking, users will be able to easily transfer files and data between virtual and real systems and create different applications and data flows.

So far we have seen several interesting examples and in particular, the capability of Simics IA to debug BIOS and probe register level information. For more details, the Wind River Simics package contains many examples and sample scripts that allow you to quickly come up to speed. For example, the Simics package includes detailed application notes on getting the Ethernet network working in a simulated network and connecting to a real network. For access to this technology, refer to contact information listed on the Wind River Simics website (Ref [1]).

Unique Capabilities in Simics allow you reach further

Simics offers several unique capabilities that just cannot be duplicated with the physical hardware. Because of this, developers continue to use Simics long after physical hardware is available and reliable.

When the best approach to trace problems is to go back in time

It may be possible that in some system debug situations the best way is to start from the point of the failure and go backwards. In a simulation environment like Simics, the reverse execution technology allows you to go back in time and inspect the sequence and system states leading to that failure. This may potentially provide the break-through you are looking for in the debug process, where issues are evasive and hard to trap on real silicon.

When it is difficult to debug real-time on silicon

As technologies evolve, system debug on real silicon can become more and more challenging. Not only are there a lot of new factors introduced with technologies such as security, virtualization, the speed of packet processing and the scale of the computing are increasing at an amazing pace. This presents challenges for the traditional debug and development approaches. In Simics, since users are the owners of the environment, they can halt live traffic, go backwards, single step through, source level debug, and even get behind the scene. In these situations, Simics provides users with backstage access to the system and allows them to reach out further than on real hardware.

When there is an access issue with real silicon

There may be multiple reasons that real silicon may not be available, some of them are listed below:

- Pre-silicon, the physical boards are not yet available
- Show stopper, the physical boards get stuck in some critical issues that gate the progress of other activities
- Not enough physical boards for all engineers to take advantage of the access
- Many years after launch, the boards are nowhere to be found, or need X weeks to be brought back to life and get transferred to the relevant site

In these situations, simulated hardware provided by Simics can serve as an effective bridging solution and enable design teams make progress.

Conclusions

We provided an overview of Simics technology (Ref [1]) and discussed several classic usage models in an embedded IA design situation. We walked through multiple examples to demonstrate the versatility of the technology and touched on the fact that it can contribute in various areas. It is a proven technology with a solid track record and many success stories. The technology is ready to become an integral part of IA design process. With professional services provided by Wind River, the customers can tailor the solutions to meet their specific needs and truly become the owners for the technology. We expect this technology to bring value to IA customers and help them design products faster, go to market sooner and improve the quality of the solutions.

The Intel® Embedded Design Center provides qualified developers with web-based access to technical resources. Access Intel Confidential design materials, step-by step guidance, application reference solutions, training, Intel's tool loaner program, and connect with an e-help desk and the embedded community. Design Fast. Design Smart. Get started today. http://www.intel.com/p/en_US/embedded.

References

1. Wind River Simics Product website, http://www.windriver.com/products/simics/
2. Jakob Engblom, Daniel Aarno, Bengt Werner, “Full-System Simulation from Embedded to High-Performance Systems”, Processor and System-on-Chip Simulation; Leupers, Rainer; Teman Oliver (Eds.), 2010. ISBN: 978-1-4419-6174-7
3. Jakob Engblom, Wind River Simics for Software Development, http://www.windriver.com/whitepapers/whitepaper.php?f=WP_Simics_Software_Development_0411 21-46-12.pdf

Acknowledgements

Special thanks to contributions and feedback from Eddie Glenn, Senior Manager, Simics marketing, Wind River.

Author

Tian Tian is a Senior Platform Architecture Engineer in the Intelligent Systems Group at Intel Corporation.

§