



THE *FASTEST* VERIFICATION

**ZeBuTM : A Unified Verification Approach for
Hardware Designers and Embedded
Software Developers**

White Paper

April, 2010

www.eve-team.com

Introduction

Moore's law continues to drive both chip complexity and performance to new highs every year, and continues to stress and periodically "break" existing design flows. Fortunately for EDA users, the same shrinking geometries that make their design problems tougher are also helping to improve the performance for their EDA tools.

But when it comes to functional verification, traditionally the largest bottleneck in the design process, software-based approaches like simulation continue to lose ground. Why isn't simulation speed keeping pace with device complexity? Because many new devices like 3G cell phones, internet routers, image processors, etc. require massive verification sequences that would take many CPU-years to simulate on even the fastest PC. These sequences are often a result of the need to run long, contiguous, serial protocol streams or complex embedded software in order to fully verify a new SoC or system design.

Increasingly, embedded software is overtaking the hardware content of SoC devices. The net result is a kind of chicken-egg problem: which comes first - the "final" hardware or the "final" software? Embedded software developers need an accurate model of the hardware in order to validate their code, while the hardware designers need fairly complete software to fully validate their ASIC or SoC. Software developers can sometimes get started using a bare-bones, non-cycle accurate high-level C model of the processor or an instruction set simulator (ISS). Similarly, chip designers can simulate their design along with small code snippets or diagnostics to verify basic functionality. But eventually both these groups need to come together on a common model to verify the complete hardware and the embedded software together. Unfortunately, for most teams that first complete model is the actual silicon.

The problem with waiting for real silicon is that the embedded software can't be fully validated in the context of a complete, accurate system model until very late in the design cycle. That increases the probability that significant problems will be found in the silicon, the software or both, often necessitating additional silicon re-spins and code revisions. Both of these have significant cost and time-to-market implications. This is illustrated in figure 1 below.

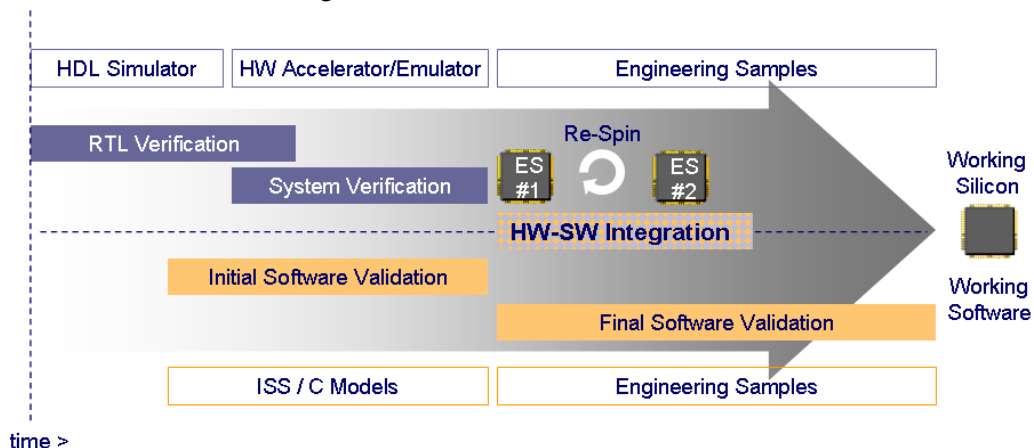


Figure 1. Most design teams wait for real silicon before performing HW-SW integration and final software validation. The net result is a significant time to market delay.

The costs can be substantial. At 65 nm, a new mask set costs approximately \$3M and may add weeks to months of additional delay. What's more, a 3 month delay in a fast-moving market can cause a loss of

25% of total product lifetime revenue, according to IBS, Inc. How can these costs be avoided? What's needed is a hardware-assisted verification approach that can enable hardware/software integration to begin well ahead of first silicon.

Traditional Approaches to Hardware Assisted Verification

Logic emulation and FPGA prototyping have been around for close to thirty years but neither of these approaches provides an optimum solution for hardware/software integration.

Traditional emulation systems offer the large capacity required to emulate complex systems, in addition to hardware debugging features that help designers isolate bugs in their chip design. However they are very expensive assets, while their performance of sub-megahertz is inadequate for embedded software validation. As a result, these systems tend to be relegated exclusively to hardware verification activities, and mostly to large companies with big EDA budgets.

Embedded software teams typically need better performance to run long code sequences, and need more cost-effective systems since they may need multiple copies to support a team of several software developers. FPGA-based rapid prototyping systems are an order of magnitude faster than emulators and are much less costly. However, for SoC designs in excess of 10 million ASIC gates they are not easy to handle since the design mapping process is virtually manual and therefore error-prone and time-consuming. Further, they are not well suited to hardware debugging tasks since probe points must be added selectively at compile time. As a result, FPGA prototypes appeal more to software development teams.

Introducing ZeBu - The Best of Both Worlds

What's needed is a new approach to hardware-assisted verification that combines the strengths of traditional emulation with those of rapid prototyping into a unified solution that is accessible to both hardware designers and software developers. That is what EVE has set out to do with the breakthrough architecture used in ZeBu (for Zero Bugs). As shown in the table below, ZeBu is much faster and more affordable than traditional emulation systems, and is much easier to use, offers much higher capacity and better hardware debugging than FPGA prototyping systems. In addition, ZeBu's integration with popular HDL simulators and, even more so, with high-level of abstraction test benches at transaction-level offers virtual interfaces to potential unavailable parts and the highest performance in the industry, reducing co-emulation overhead by at least one order of magnitude.

ZeBu has been architected to meet the needs of both hardware and embedded software engineers, and to provide them with the hardware and software debugging capabilities to help to debug complex embedded SoC designs. ZeBu incorporates the largest FPGAs available with the highest level of integration and smallest footprint - all key to the system's capacity, performance and cost-effectiveness. The ZeBu compiler automatically handles any design, regardless of size, coding style, clocking scheme, or memory structure, making it easy to map even large designs. These have historically been the biggest problems with other FPGA-based rapid prototyping systems, but not with ZeBu, thanks to its complete software infrastructure.

	HW Emulators	ZeBu	FPGA Prototyping
Primary Application	Hardware Verification	Hardware & Software Verification	Software Validation
Max Design Size	250M ASIC gates	1B ASIC gates	10M ASIC gates
Speed	0.1 – 1MHz	1 – 60MHz	1 – 100MHz
Cost	\$\$\$\$	\$-\$	\$-\$
Setup Time	Days-Weeks	Days-Weeks	Months
HDL Co-emulation	Good	Good	No
Transaction-level Co-emulation	Fair	Excellent	No
HW Debugging	Good	Good	Poor
HW/SW Integration	Fair	Excellent	Poor
SW Validation	Poor	Excellent	Excellent

Table 1. Comparison between ZeBu, traditional emulation systems and FPGA-based prototypes.

Reconfigurable Test Bench (RTB) Architecture

ZeBu’s patented Reconfigurable Test Bench (RTB) architecture makes the system especially well suited for hardware debugging applications. The RTB controls and optimizes how the emulated system receives stimulus and communicates with software running on the workstation. Based on a powerful C/C++ API, the RTB communicates directly with HDL test benches (VHDL/Verilog/SystemVerilog) and/or abstract system level models (C/C++/SystemC™ or SystemVerilog). The RTB supports communications both at the signal/bit-level and at the transaction-level with high-level data such as frames, packets, or blocks. The latter is especially efficient, with ZeBu systems able to sustain up to 5 million transactions per second. The RTB is used to implement complex synthesizable transactors, synthesizable test benches, or even synthesizable assertions.

The RTB also provides all the controllability and accessibility functions required by hardware designers in order to debug their design. For example, the RTB provides interactive read/write access to all internal registers and memories at run-time, without needing to compile internal probes.

In addition, there is a clean separation of the testbench from the design-under-test (DUT) resulting in significantly better utilization of DUT targeted FPGA’s. Not to mention that a change in the testbench, for example from Verilog to SystemVerilog or from cycle-level to transaction-level, only requires recompilation of the RTB and not of the DUT, saving a significant amount of time.

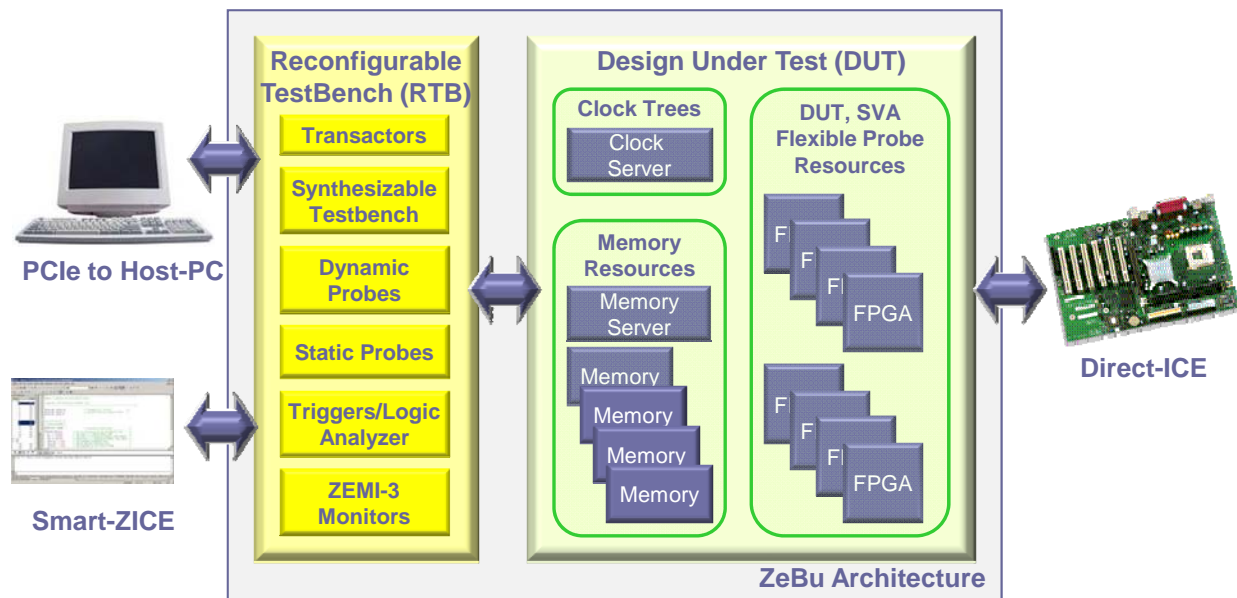


Figure 2. ZeBu's patented Reconfigurable Test Bench (RTB) architecture enables efficient co-emulation with PC-based simulators, or C/C++/SystemC models written at high-level of abstraction and provides extensive visibility to the emulation resources.

ZeBu Applications

ZeBu's flexibility enables its use throughout the hardware and embedded software development cycle to:

- Verify and debug ASICs/SoCs, IP cores
- Co-verify hardware and software
- Develop and debug embedded software

These applications are detailed below:

- **Verify and debug ASICs/SoCs, IP cores**

With its integrated hardware debugging resources and high execution speed, ZeBu is ideal for verifying and debugging complex hardware blocks, IP cores and full chip designs. Early in the design cycle ZeBu can be used as a simulation accelerator in co-emulation with leading HDL simulators and/or C/C++ code. ZeBu can be driven by existing HDL testbenches with no modification, and by HVL testbenches such as SystemVerilog, Vera and *e*. In addition, test benches can be applied externally by the workstation or synthesized and mapped into the emulator.

- **Co-verify hardware and software**

Later in the design cycle as the pieces of the design come together, ZeBu can be used to validate the entire system by running the embedded software with either a synthesizable test bench or in co-emulation with C++/System C code on the workstation. ZeBu has been architected for especially fast co-emulation at the transaction level, with operating speeds up to 30 MHz. For software debugging, the system's processor can be connected through a JTAG cable or a JTAG transactor interface to standard embedded software debuggers from ARM, TI, ARC, etc. The JTAG connection via transactor allows for stopping the clock for tracing HW bugs in the design hardware. In this way ZeBu can execute software drivers, operating systems or applications at MHz speed, while providing full hardware and software debugging capabilities to both hardware and software engineers on the same design representation. This enables hardware and software

development teams to collaborate on the resolution of integration issues in a way that was never possible before.

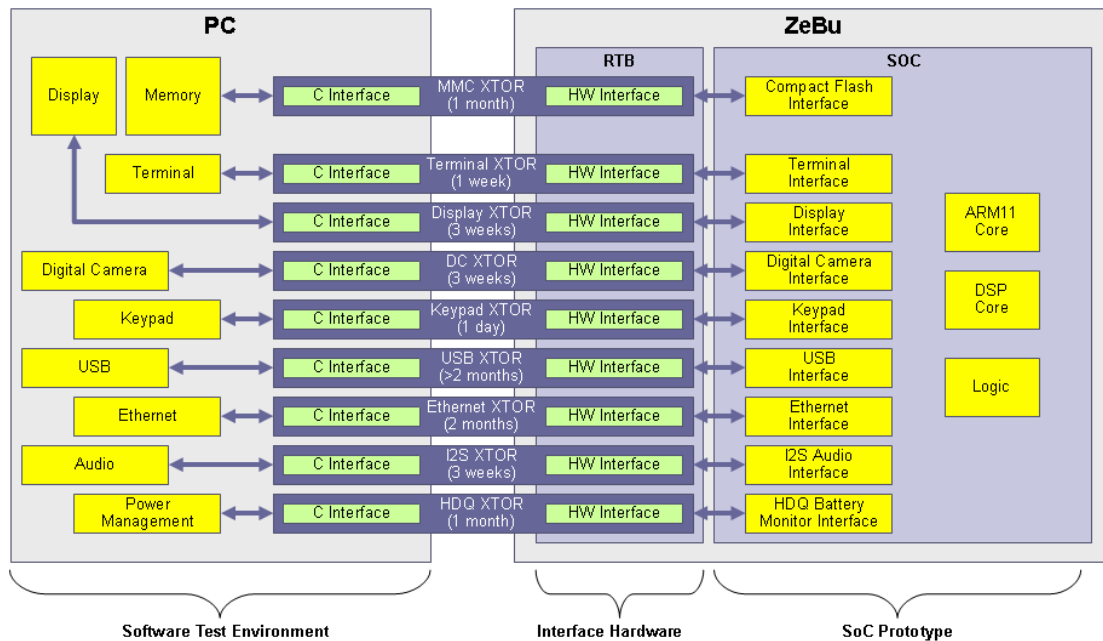


Figure 3: Example of virtual platform based on a set of transactors used to verify a popular PDA core

- **Develop and debug embedded software with reduced-cost “replicates”**

ZeBu can also be used as a dedicated software development platform for embedded software developers, with full access to software development and debugging tools. The embedded software can be downloaded into the design memory via the PCI-Express interface in a fraction of second, much faster than via the JTAG port. Also, ZeBu’s *save* and *restore* capability can be used to skip the OS boot phase and advance directly to the point in the code at which a problem was detected.

Since hardware debugging features are not required at this stage, EVE offers a “replicate” version of ZeBu that is optimized for the needs of embedded software developers, and available at a fraction of the cost of a full ZeBu system. Many teams utilize one fully configured ZeBu system for hardware debugging while providing multiple replicate systems to the software development teams. In this way, they can leverage the same system model used for hardware verification while controlling overall costs. The net result is that ZeBu users are better able to parallelize hardware and software validation tasks, and accelerate the time to market significantly. This is illustrated in figure 3 below.

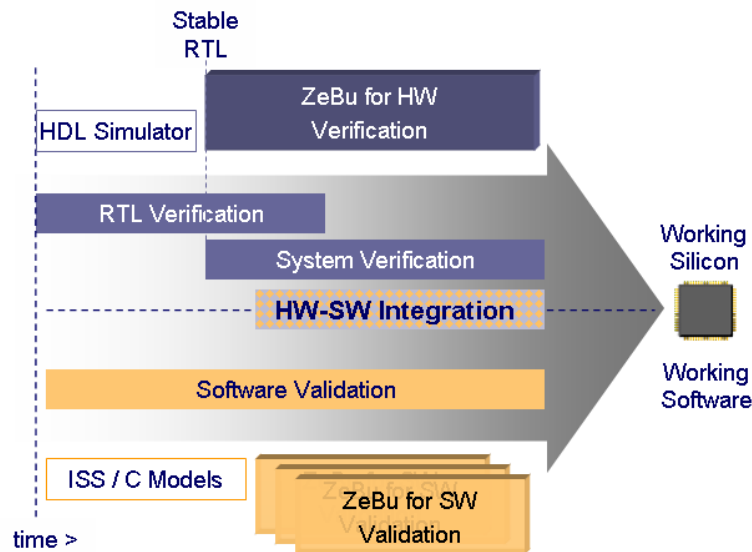


Figure 4. ZeBu users perform hardware verification and software validation in parallel- ahead of first silicon. This translates directly into a significant time to market advantage.

ZeBu Operating Modes

In order to provide the most value for each of these applications ZeBu supports several operating modes that help exploit ZeBu's performance throughout the design cycle. The 7 different operating modes are as follows:

Co-emulation with commercial HDL simulators: A co-emulation link connects ZeBu to any VHDL/Verilog/SystemVerilog simulator compliant with the IEEE Verilog PLI interface, such as NC-SIM™, VCS™ and ModelSim™. Specman Elite and Vera are also supported via the HDL simulator. The run-time performance is strictly dependent on the testbench and it can be quickly estimated by profiling the testbench versus the DUT via a simulator. A top speed of 100kHz is probably the maximum achievable.

Co-emulation with C/C++/SystemC at Signal-level or Cycle-level: ZeBu provides a C/C++ API with which one can create efficient C/C++/SystemC test benches to stimulate the design-under-test at the cycle level. The C++ API also supports the connection to a SystemC model at the cycle/event level or to an Instruction Set Simulator (ISS). An order of magnitude faster execution than in co-emulation with an event-driven simulator can easily be obtained.

Co-emulation with C++/SystemC/SystemVerilog at Transaction-Level: When the design-under-test includes bus interfaces (such as Ethernet, PCI, USB, Sonet...), it is possible to stimulate it with C++ or SystemC or SystemVerilog test benches at transaction level (i.e. with high level data such as frames, packets or blocks) instead of at signal level to minimize the amount of data exchanged between the host PC and the ZeBu emulator. The C++ API also supports the connection to a SystemC model at transaction level (through channels). Another order of magnitude faster execution than in co-emulation with a C/C++/SystemC at the cycle-level can typically be achieved.

Test Vectors: This pattern-based mode accelerates the regression testing of designs before tape-out. A pattern file, captured during HDL or C/C++ co-emulation, can be played back at higher speed without the

HDL simulator or the C/C++ test bench present. The pattern execution tool detects output signal differences between the expected responses and the current values provided by the emulated design. Similarly to the co-emulation at the cycle-level, a maximum speed of 500kHz is possible.

Emulation with Synthesizable Test Bench: A synthesizable test bench can be embedded in the FPGA of the RTB to enable emulation at several MHz. This mode provides the best level of performance since there is no interaction with any external software or hardware components. EVE provides a catalog of synthesizable memory models for many Micron and Samsung devices.

Emulation with Software Debugger Control. The emulated design can be connected via the Smart-ZICE interface to any software debugger via a JTAG cable or JTAG transactor. The debugger will then be in control of the software execution.

In-Circuit Emulation with Target System and/or Hard IP Cores: In this mode, the emulated design-under-test is cabled to the target system, which provides all stimulus and control. One or more hard IP cores can also be included in the system model and speed rate adapters can connect the design to real-time interfaces. A built-in logic analyzer can trace the design states over long emulation runs. Complex triggers can be added to control the tracing mechanism.

Multi-mode Operation: ZeBu also supports combinations of the modes outlined above. For example, a multi-mode test environment could include a synthesizable test bench mapped in the RTB, an application program embedded in a DRAM synthesizable model, a set of protocol transactors (USB, Ethernet, Audio, etc.) driven by C++ models executed on the host PC/Linux, a JTAG transactor connected to a S/W debugger running on a PC/Windows and an LCD transactor displaying the LCD image stream directly on the host PC.

Summary

Unprecedented device complexity and embedded software content are driving the need for a new hardware-assisted verification approach. ZeBu is a breakthrough architecture that combines the strengths of traditional emulation and rapid prototyping into a unified platform for both hardware and embedded software development. For the first time a single platform and design representation can satisfy the needs of both these applications, enabling hardware designers and software developers to communicate and collaborate in a way that was never possible before. The net benefit of ZeBu's unifying approach is to accelerate hardware/software integration well ahead of first silicon, reduce unnecessary re-spins and software revisions, and shorten time to market.

With a highly cost-effective architecture, ZeBu makes emulation more accessible than ever before-accessible to both SoC designers and embedded software developers, accessible throughout the design cycle, and accessible by groups with modest EDA budgets. Overall, ZeBu clearly provides the best return on investment of any hardware-assisted verification approach.