

EVE Taps Xilinx for Multiple Generations of Emulators

Engineers can debug complex chip designs on FPGA-powered ZeBu system.

by Ludovic Larzul
Vice President of Engineering
EVE
ludovic_larzul@eve-team.com

Creating a hardware emulation system is no easy task. At a minimum, each generation of emulation system has to accommodate a growing number of logic gates, memory and DSP blocks to allow ASIC and ASSP system-on-chip (SoC) designers to debug their extremely complex devices before sending them off to the foundry for production. Emulation systems must also be easy to program, reliable and, what's more, affordable. Here at EVE, we've developed several generations of emulation systems—leveraging the power of Xilinx® FPGAs—to emerge as a leader in the market.

Today's SoCs are exceedingly complex pieces of silicon. They contain one or more processors that will execute software. The software code they run is every bit as important a part of the final system as the silicon itself. The software and the silicon have to act as a seamless solution; if there's a problem, it might be the software, or it might be the silicon.

Designers can only do so much software testing on a development host. No reasonable host development system can reflect the true parallelism of the target SoC. You can really only test out such issues as synchronization, data integrity and resource contention in situ, and that's far too late to identify problems. Simulation isn't a viable solution; it's simply too slow to allow the execution of any realistic code.

As a result, engineers have been using emulation systems for well over two decades to verify the most advanced ICs the semiconductor industry can build. Most of these earlier-generation emulation systems were powered by custom ICs that the emulator vendors designed themselves. They would then pass the cost of the custom IC development on to their customers, making the power of emulation more cost-prohibitive for companies struggling with ever-tighter IC development budgets.

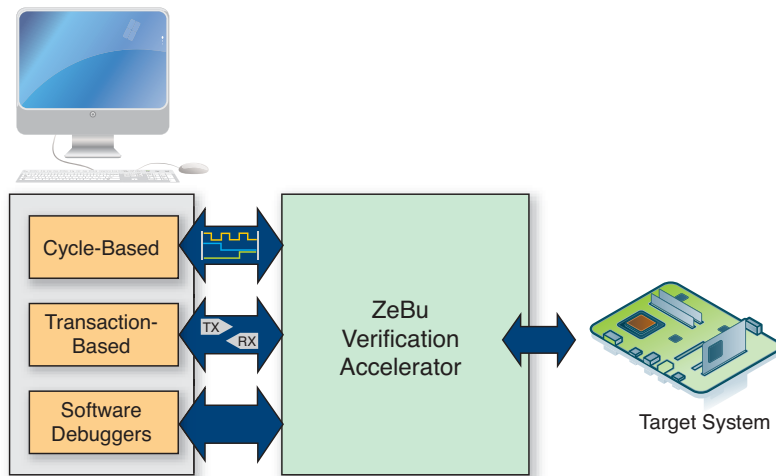


Figure 1 – EVE bases its ZeBu (for “zero bug”) emulation system around Xilinx FPGAs.

In 2001, EVE broke with tradition by basing our innovative emulation system on Xilinx FPGAs. The goal was to provide the lowest hardware-assisted verification cost of ownership in the industry, as achieved through a combination of high execution speed, high capacity (which today means up to a billion gates), quick design revision, flexible and powerful debugging capabilities, lowest cost per gate and most cycles per dollar. In addition, we wanted to make the

system easy to use for ASIC designers who might not be familiar with FPGA design.

The result was the ZeBu (for “zero bug”) emulation system (Figure 1). We’ve now developed six generations of emulators, the most recent of which is ZeBu Server (see sidebar), and we’re still going strong.

Separation of Powers

Our approach is to split the device-under-test (DUT) from an interface to

the test environment that we call Reconfigurable TestBench (RTB). The RTB allows for test configuration and control. The DUT will change with each rev of the design, but the RTB never changes unless the test environment does. Having a single mass of FPGAs containing a mix of the RTB and DUT designs would have been messy and required unnecessary recompilation of the RTB design, so we separated them out.

As a result, we have one set of FPGAs for the DUT and another set for the RTB (Figure 2). The number of DUT FPGAs varies by system size; bigger and smaller systems are available for bigger and smaller designs.

The RTB FPGAs provide communication and control. We can optimize them much more aggressively for performance and capacity, since the design will remain constant. While no one reconfigures the contents of this set of FPGAs, the design allows a rich set of configurations of the test setup under user control.

For the DUT FPGAs, the overwhelming priority in terms of required features was density. We needed to be able to put really big designs in here in order to provide value. Key among the necessary hardware features were multipliers, which later gave way to DSP blocks. In addition, big designs needed

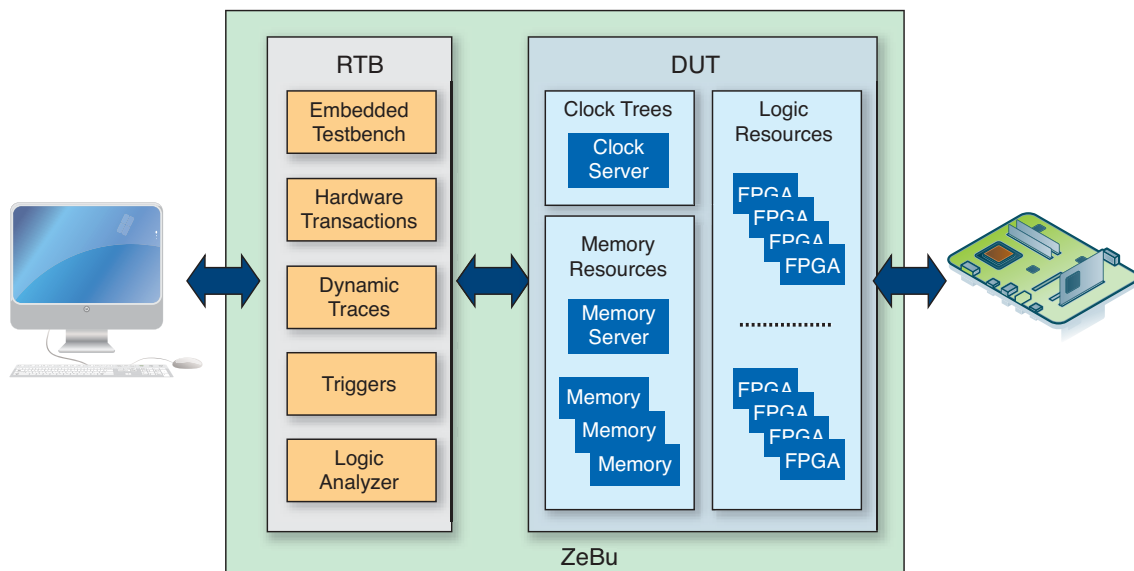


Figure 2 – ZeBu uses one set of FPGAs for the device under test and another for the Reconfigurable TestBench (RTB).

big memory, so the largest possible memory blocks were a requirement.

Close behind density on our “must have” list were bandwidth and latency. The data transmission between the FPGAs had to be fast enough to keep from becoming a serious bottleneck. We didn’t plan to use clock-data-recovery (CDR) circuits—in fact, they weren’t even available when we started out—but we did use high-speed LVDS I/Os, layering our own proprietary low-latency protocol over them.

To provide robust debugging capabilities, we needed readback of internal states. We wanted a way of interrogating what was going on inside so that designers could understand the inner workings of their technology—especially when it didn’t seem to be working. We might implement readback using JTAG or some other means, but it had to be there.

Finally, to keep the challenge of implementing a single design across multiple

FPGAs tractable, we wanted to use only a single technology on the board and across different members of the emulator family. So the FPGA family we chose had to have a broad range of densities and pin counts.

Our considerations for the RTB FPGAs were different. Again, one of the primary requirements was keeping to one FPGA technology per system. This would ensure that all of the FPGAs were in sync with respect to version and availability. Of course, we still needed to be able to meet our performance requirements, no easy task given that the RTB FPGAs run at twice the speed of the DUT FPGAs.

Given the sum of requirements, the Virtex®-II family was our clear choice for the first ZeBu generation. At the time, it led in gate and memory density, provided greater bandwidth with less latency and excelled in its readback capability. Our first system used two Virtex-II 8000s for the DUT and one Virtex-II 6000 for the RTB.

Over time we’ve continued to evaluate the Virtex family against others as the technology has advanced, but to date we have never found another choice compelling enough to move us away from the Virtex. As such, we have progressed with Virtex-4 (up to 64 Virtex-4LX220s for the DUT) and Virtex-5 (as many as 400 Virtex-5LX330s), and are eyeing the new Virtex-6 family with eager interest.

Going With a Flow

We design the RTB FPGAs with a standard FPGA flow, using XST for synthesis and the standard Xilinx ISE® tools for place and route. The effort we make here is typical of what anyone designing with FPGAs might do to create a complex design. The differentiation isn’t in the flow; it’s in the hard work and our system knowledge.

When it comes to the DUT flow, however, our challenge is far greater than the one the typical FPGA designer faces. By

Evolution of Hardware Emulation Culminates in ZeBu-Server

Hardware emulation has become a valued component of the hardware/software co-design flow, enabling hardware engineers and software developers to share system and design representations and work together to debug hardware/software interactions.

Use of this tool as a popular EDA solution has evolved over the past 20 years, from the early days of standard FPGA-based emulators to custom ASIC-based models, back again to emulators based on standard FPGAs.

CPU and graphics chip engineers were the first to use emulation, because the sheer complexities of their designs crippled traditional event-based hardware description language (HDL) simulators. The tool quickly gained acceptance by the wireless community due to the extensive use of embedded software in its hardware designs. Today, consumer electronics companies widely use hardware emulation for the design of digital TVs, set-top boxes, digital still cameras and camcorders, multifunction printers and other products.

The early version of hardware emulation delivered execution speeds four to five orders of magnitude faster than simulation, making it ideal for accelerating the time required to develop and validate the hardware of ASIC or system-on-chip (SoC) designs. However, it did not meet the minimum speed of execution required for efficient testing of embedded software—namely, 1 MHz. Moreover, cost of ownership hampered emulation’s popularity and restricted its adoption to large corporations in limited numbers.

The newer FPGA-based emulators are more reasonably priced and cost-effective, and have shortened the overall verification cycle

of complex chip and electronic-systems designs. These emulators have a smaller footprint than prior emulation tools, are fast, efficient and easy to use.

Setup is straightforward and newer emulators consist of fewer FPGAs than older machines. The latest generation of emulation systems can execute billions of verification cycles, as required in embedded designs, in a short period of time. They provide a full view of the design, necessary to debug the hardware. These machines also support transaction-level verification, which is needed for hardware debugging at a high level of abstraction, via monitors, checkers and assertions.

EVE is an FPGA-based emulation trendsetter. On July 14, it launched the latest incarnation of the ZeBu (for “zero bugs”) product line, called ZeBu-Server, a sixth-generation version of its emulator based on the Xilinx Virtex LX330.

Providing design capacity of up to 1 billion ASIC-equivalent gates, ZeBu-Server can be used across the entire development cycle for verifying and debugging a new ASIC or SoC design. Designers can use it to test the integration between hardware and software, and to validate embedded software before silicon availability.

This new emulator improves on previous-generation offerings in terms of capacity, speed, setup time, integration and debugging capabilities. And last but not the least, it is also cost-effective.

— Ludovic Larzul

definition, we deliver an unfinished system, in the same way that Xilinx delivers an unfinished chip. Our customers have to fill in the DUT design details. So we must provide them with a way to implement their designs—while assuming that they may not know or care about how to do FPGA design. To help facilitate this, we created the ZeBu Compilation User Interface (zCUI).

A key concern is the fact that the DUT design, by definition, will change numerous times throughout the design cycle. Minimizing the design turn time has always

For this reason we developed our own zFAST synthesis tool, which could take SystemVerilog, VHDL or Verilog designs and synthesize them as much as 10 times more quickly than standard synthesis products. We now make zFAST available alongside the more traditional synthesis tools so that our customers can invoke it for large designs as needed. Once the design is synthesized, our own partitioning tool partitions it at the gate level.

It should be noted that some of our users are accomplished FPGA designers,

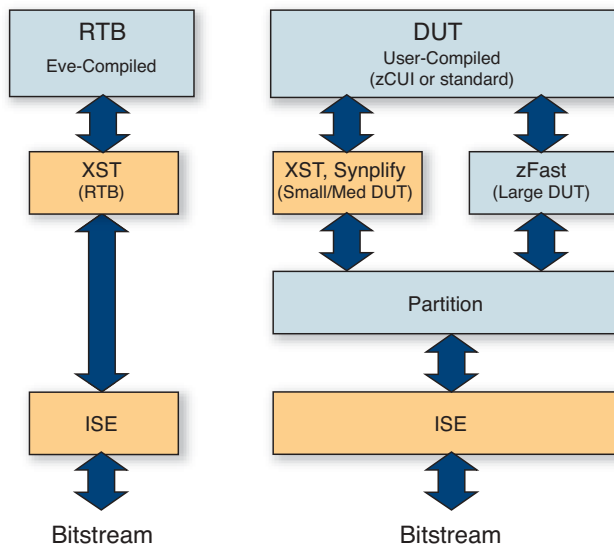


Figure 3 – The ZeBu design flow makes use of Xilinx's ISE tools.

been a high priority. Because the first systems were relatively small, we didn't focus on the compiler performance; we simply worked through the standard Xilinx flow. But as we moved up in density, compile time—and, in particular, synthesis time—became more critical. A design that will occupy 400 of Xilinx's largest FPGAs is not trivial to synthesize.

We were concerned that standard synthesis tools were spending too much time doing too good a job. Our priority was not optimal synthesis—optimal meaning highest performance, lowest gate count. Those are good things, but for us, compile time was a higher priority, and we could live with 20 percent worse performance or density—especially since we're executing at less than 30 MHz.

and we willingly hand over greater control if they want it. The customer is free to use the standard FPGA flow to try to get more performance out of the DUT, constraining and using any tricks they know as necessary. So the system makes it easy for a non-FPGA designer, but does not restrain an expert.

The continuing performance improvements of the ISE tools from Xilinx have also helped our overall compile times. EVE was one of the first users of ISE 11, allowing us to keep our FPGA compile times typically under two hours per FPGA. Parallel compilation means that we can turn large designs with multiple FPGAs in a reasonable amount of time. Our overall flow is summarized in Figure 3.

Trust, but Verify

Verification is a bit tough for us, specifically because we don't ship a finished design. We have to be confident that, once our customer compiles a design onto the ZeBu platform, everything will work as promised.

While the verification of our first set of RTB FPGAs was a challenge, since then we've been able to use our own existing ZeBu systems to verify the RTB for the next-generation ZeBu system. We run the new RTB for hours and days, executing billions of cycles, to confirm that it's solid.

For the DUT FPGAs, in conjunction with our own tools we have a large vault of designs accumulated over years of experience. Each night, we run thousands of designs and test the results using application-specific test bench suites associated with each individual design, as well as through cycle-by-cycle simulation comparisons that include internal state as well as inputs and outputs.

We also check to make sure that our results remain deterministic as our tools and those from Xilinx evolve. Whenever there's a revision change, we run through the suite of tests to ensure that the results we get with the new version are the same as those we got on the older version.

Looking back, then, it is clear that EVE's technology has been intricately bound up with Xilinx's technology. Starting with the original choice of Virtex-II, we have dovetailed the growing capacity of the Virtex chips and the evolution of the Xilinx tools with our own system design and tools to provide a solution that continues to gain traction in the emulation market. Given the Xilinx and EVE road maps, that symbiosis is likely to continue unabated. ●●●

Ludovic Larzul, a founder of EVE and its vice president of engineering, has 13 years of experience in CAD development. He has worked on numerous projects, including the integration between hardware emulators and software simulators, efficient interfaces based on transactors, communication among multiple emulation systems and system-level place and route. Larzul holds a Diplôme d'Ingenieur de l'Institut de Recherche et d'Enseignement Supérieur aux Techniques de l'Electronique (Ecole Polytechnique de Nantes).