A significant evolution is underway in SoC verification and validation.

The complexity of SoC designs has resulted in the need to perform both comprehensive verification as well as system-level validation very early in the design cycle, often before stable RTL code is available for the entire design. This same complexity has also created the need for extensive internal visibility into the design to understand subtle problems that can occur during silicon bring-up.

While the needed level of visibility can be provided with a model of the design, it requires sufficient execution speed in the modeling environment to run content that matches silicon tests to highlight issues. Hardware emulation has sufficient execution speed, full visibility capabilities and ease-of-use in model creation and model updates to span the entire range of needs throughout the life of the design development process.

While the use models for hardware emulation are numerous, basic requirements and concerns are common to all users:

- Efficient bring-up time in days, not weeks or months
- Fast design compile and test execution times
- Thorough and extensive debug
- Flexibility to accommodate a wide variety of stimulus

Moreover, while the core objective of emulation is functional verification, other aspects of verification and validation are becoming equally important: low power verification, power measurement, coverage closure, software bring-up, safety, security, and design for test (DFT), to name a few. Emulators now tackle issue-specific verification and validation in non-traditional areas.

Since these areas are becoming more important, the requirements for supporting use models, different types of users, and more teams with different verification and validation objectives is also growing. As well, teams are no longer in the same physical location, making remote location support an imperative. To satisfy these requirements, an emulation platform must continue to evolve.

This article identifies the three main components you should look for in an emulation platform and describes the capabilities and advantages each delivers to the verification of SoC and system-on-system designs. Whereas each component has its own strengths, they all should work together to enhance user benefits and deliver exceptional verification productivity.

CUSTOMIZED PROGRAMMABLE LOGIC ARCHITECTURE

Ideally the heart of the emulator should be a highly specialized SoC with a customized programmable logic architecture. This type of chip delivers many advantages to users, including fast design compile and bring-up times, highly-flexible memories, and the ability to scale from small to very large designs.

Advantage #1: Fast Design Compile and Design Bring-Up

Fast compile is extremely important, as this allows users to modify their design and build a model to run on the emulator. The faster the compile, the shorter the total turnaround time will be.

In addition, the chip should be customized for modeling purposes, as is done with the Veloce® Strato Crystal3 chip. This enables faster design bring-up. The models created to run on Veloce® Strato are far more “tuned” when compared to what is done with other hardware emulators, which are based upon other technologies like commercial FPGAs.
The Crystal3 chip’s VirtualWires® high-speed network implementation delivers faster and 100 percent-successful compiles of large designs in a fully automated way, which is key for faster design bring-up.

This approach to compile and modeling depends on a set of integrated systems at the hardware and software levels that are constructed so they work to create models that satisfy timing closure by construction while adhering to HDL semantics.

The Crystal3 chip has programmable fabric, but it is not an FPGA. The Crystal3 fabric’s distinctive architecture makes it the most efficient chip for emulation, specifically from a timing closure perspective. This is different from what is commonly done for implementation on an FPGA to achieve compile time and compile success.

The unique architecture of Crystal3 SoC delivers quick bring-up, quick turn times with faster compile time, and automated compile for large designs with 100 percent success.

From a memory perspective, the issue confronted in emulation is that integrated circuit designs come with a wide variety of memories: they can be large, small, have different widths, different numbers of ports, come from standard libraries, or be custom made. Users who are modeling systems may also have discrete memory devices that connect externally to their SoCs. Users may create types of DDR and FLASH external memory devices that need a wide variety of support. Veloce also provides the mechanisms for modeling this variety of memories using a highly flexible software approach that can vary across a large dynamic range: from a simple memory of four to eight words to a memory of billions of words, as in the case of FLASH.

The Veloce® Strato makes a broad range of memory modeling resources available. This improves ease-of-use and ease-of-design-bring-up for modeling memories and utilizing emulation resources efficiently. It provides total flexibility to accommodate a full range of memory types so users do not have to force fit what they want to model. For both chip and system memory, the compiler automatically handles the process without intervention from the user thanks to a common description interface. Veloce® software is smart about detecting memories in RTL and matching them to the characteristics of the memory needed.

**Advantage #3: Ability to Scale from Small to Very Large Designs**

Models built for emulation are designed to be mapped to multiple chips on the available boards in the emulator. The number of boards in an emulator ranges from one to hundreds depending on the size of the design under test (DUT). The VirtualWires® high-speed network allows the emulator to move data between the chips and boards to optimize and streamline the emulator for best use.

Emulation performance is impacted by both the bandwidth and the latency of the network connecting the chips and boards. For most designs, the most important aspect for performance is latency because many design paths are distributed across multiple chips and traverse the network multiple times in a design cycle. Bandwidth in the network effect on performance for large and highly connected designs,
where a large amount of data must typically move in and out of emulation boards or between connected emulators.

At the system level, Veloce® Strato uses a network that has a mix of direct connections that are close together as well as indirect programmable connections across board and system boundaries. This means that a large majority of the chip connections are handled by direct low latency connections, but the emulation can also handle large designs where there is a need to interconnect more remote resources through intermediate switching.

The network infrastructure inside the Crystal3 chip is both an end-point and a switching device. Some Crystal3 chips only serve as switching devices, and some serve as both the end-points to source and receive data as well as intermediate switching for traffic between other chips. Most connections are directly drawn from Crystal3-to-Crystal3 chips, end-point to end-point. Other connections go through one or more intermediate chips that allow data to arrive at its ultimate location. These attributes are what make the Crystal3 chip the foundation for an extremely flexible, multi-user model—allowing a design to be compiled to a wide array of physical resources and mapped to multiple boards. It delivers total flexibility for moving designs around to many different places in the emulation system to maximize the use of available resources, without requiring recompilation. The networking structure of the Crystal3 chip lets this happen (dynamic remapping), and it enables enterprise-level resource utilization. This underlying technology is unique to the Veloce® Strato emulation platform.

**Advantage #1: Ease of Design Bring-Up**

The emulator compiler software enhances the user’s design bring-up experience. It supports a broad set of modeling constructs commonly used for building verification environments, which minimizes the changes required by the user to build emulation models. This dramatically improves design bring-up time and the user experience. Veloce® Strato OS automatically exploits modeling resources built into the Crystal3 chip (e.g., different types of memories and CAMs) for mapping designs to the emulator without requiring remodeling. Faster design bring-up leads to more time for verification and fewer dedicated engineers for building emulation models. Together, these features make the emulator a verification platform for everyone, not just emulation experts.

**Advantage #2: XRTL Transactor Modeling**

With the advent of high-speed, scalable, co-model architectures and virtual emulation capabilities, emulators are becoming the verification hub for pre-silicon verification, post-silicon debug, and everything in between. Virtual solutions provide access to complete software/SoC co-verification via a diverse set of both general and segment specific protocols; such as PCIe, Ethernet, USB, MIPI, SATA, and CAN. Transactor modeling capabilities are provided at the xRTL compilation stage, which supports transactors compliant with the SCEMI standard and a large set of additional modeling capabilities. This facilitates rapid creation and customization of transactor solutions. It also allows for a broad portfolio of solution offerings as well as a means to implement user developed, protocol-specific transactors.

**SPECIALIZED OPERATING SYSTEM AND APPS**

The emulation platform operating system and the software-based applications that run on it should enable platform-independent, fully automated, fast compilation, modern debug methods and numerous verification use modes. This establishes the foundation for a multi-user, multi-project approach and modular resource access that leads to high verification productivity.
Transactor-based connections between emulation and software are used for much more than stimulus, as shown in Figure 3. For instance, they can be used for debug by employing a number of mechanisms that make signal access at unlimited depth possible via co-model data streaming. They can also be used for advanced applications like power analysis and DFT. For example, metadata can be exchanged between the emulator and the software for application-layer interlock, or data can be extracted from the design and used in a verification methodology based on Universal Verification Methodology (UVM) monitors. Coverage data (dynamic coverage closure) can also be extracted for things like system-level analysis, toggle analysis and cryptographic hardness.

Transactor-based connections can also be heavily involved in creative software debug methodologies. Offline debug of processors, hybrid systems and communication links between software and the DUT allow for the use of advanced debug methods and trace streaming iterations. Co-models can also be used in conjunction with virtual or hardware-based JTAG probe methodologies. Today, co-model channels permit specific software protocol analyzers for any number of protocols in storage, networking, Wi-Fi, cellular, automotive, multimedia, and mobile applications, to name just a few.

**Advantage #3: Fast, Complete Debug Visibility**
The Veloce® Crystal3 chip comes with a dedicated resource, the Trace, which is used to capture the set of information that is necessary to determine the value of all design signals for all time stamps. The chip-level data is further processed with specific software to get all the signal values. This is a complex task to solve because of the required integration between chip and software. Technology in the Crystal3 chip allows smart data capture for full visibility to enable fast and detailed debug. With this smart trace capture technology, the capture of signals happens faster and does not consume modeling resources, impact performance, or reduce capacity. The combination of smart trace capture and on-demand waveform generation features offer fast access to waveforms for debug and an infrastructure to do efficient and accurate power and performance computation for long emulation runs.

**SCALABLE, ENTERPRISE-CLASS HARDWARE**
The Veloce® Strato emulation platform is a true enterprise class verification and validation solution, scalable to accept very large designs (up to 15 billion gates) and a large number of simultaneous users that can connect from anywhere in the world, 24/7. The modular architecture was designed to support a very large number of small designs or support medium to very large designs by combining resources. This is done without affecting runtime performance, design loading, or memory/runtime operations. The emulator hardware must be equipped with technology to efficiently support and enable traditional In-Circuit Emulation (ICE), software testbench acceleration (UVM/SV/SC, etc.), virtual emulation, software/firmware development, and power and performance analysis. This provides a common solution that spans from very early design conceptualization, through RTL coding and verification, to complex post-silicon debug.

**Advantage #1: Scalability**
The emulator chassis architecture should offer on-demand scalability as the size of designs, number of users, and use modes increase. Modeling capability is provided via logic boards that offer about 40 million gates and represent the granularity for incremental
design capacity as well as aggregate system capacity. For example, the Veloce® Strato chassis comes in three configurations, offering peak capacities of 0.6, 1.25 and 2.5 billion gates (BG), each in a single monolithic chassis. The chassis configurations are extensible, allowing a 600 MG system to be expanded to 1.25 BG and a 1.25 BG system to be expanded to 2.5 BG, while preserving the original chassis investment. The Veloce® Strato logic boards plug into an active back plane that delivers a highly reliable system with no performance degradation even as designs grow from 40 MG to 2.5 BG. The system maps designs to any available logic board or combination of logic boards.

**Advantage #2: Multi-User and Global Resource**
The emulator must be a shared, multi-user resource. Logic boards are connected together through a switching system where a design need be compiled only once to target a set of virtual logic boards. At runtime, the compiled design database can be mapped on a nearly arbitrary set of physical logic boards. The switching system is configured to create the expected connectivity between the physical logic boards.

Multiple models used for many tasks by multiple groups often result in a heterogeneous workload consisting of jobs of mixed sizes and durations. The emulator architecture must allow for greater flexibility in finding optimal resources dynamically (disparate/orientations) without recompilation. Thus, the platform can support hundreds of simultaneous users running jobs in parallel. For example, the Veloce® Enterprise Server (ES) App makes access to emulation resources transparent to the user. In addition, the Veloce® ES App delivers suspend and resume functionality that allows the scheduling of high priority, time-critical jobs. This is achieved by suspending the low-priority jobs that are currently running. Once the high priority jobs are complete, the low-priority jobs are restarted from the same emulation time point where they were suspended. This makes the emulation platform a data center resource accessed by distributed teams worldwide, 24x7.

**Advantage #3: High Performance and Flexible Co-Model Infrastructure**
The emulator should have a high-bandwidth, low-latency communication channel between the emulation platform and a set of host workstations used for executing software stimuli. At a physical level, the channel is distributed so that multiple users each have access to a distinct communication channel. For large designs, many channels can be automatically used, aggregating both communication bandwidth and host processing power to provide scalable resources for the most demanding stimulus requirements – such as the ones we see with large Ethernet switching design testbenches. This flexible approach also provides mixed use of communication channels between stimulus and advanced debug capabilities.

**Advantage #4: Stimulus Options (Acceleration, ICE, and Virtual)**
The emulator must provide design model access to a variety of stimulus, whether designs consume a single logic board or hundreds of logic boards. ICE targets (which are physical devices) interact with logic boards through the multi-user switching system. This allows specific ICE targets to be flexibly connected to models on any set of logic boards. In addition, as indicated above, any collection of logic boards has access to a proportional quantity of host communication transactor channels. These can connect, in turn, to HVL simulation environments to perform acceleration, high-level virtual solutions, hybrid ISS CPU virtual models, and a multitude of other possible stimulus formats. Individual designs can mix stimulus from both ICE and virtual stimulus sources. Host-based stimulus can be changed in nearly zero time when re-assigning logic resources from one user design to another.

**Advantage #5: Full, Faster, and Smarter Debug**
The emulator should provide a variety of services – such as clock control, line break point, triggering, checkpoint save/restore, and record/replay – which, when combined with native, full (100%) visibility and streaming waveform data capture (as described above), support highly sophisticated and effective debug flows.
Both virtual stimulus and ICE are required to deterministically repeat test execution and restart from the middle or even the end of a long run while repeating prior behavior identically. This allows full visibility at unlimited depth for any use mode or form of stimulus. The emulator must be able to compile assertions natively as well as other protocol monitors and checkers modeled using xRTL. These debug aids can be selectively enabled on any desired run, and they can also be enabled during test re-execution from a middle timepoint even if they were not active during the initial run.

CONCLUSION
These three main components of an emulation platform – the chip, the OS software, and the hardware – form the foundation for a state-of-the-art hardware emulation platform. No other emulator on the market matches Veloce® Strato in terms of scalability, versatility, and proven technology advantages. No other emulator on the market embodies these three components more than the Veloce® Strato, which delivers the highest efficiency and value and lowest cost of ownership on the market.
VERIFICATION ACADEMY
The Most Comprehensive Resource for Verification Training

32 Video Courses Available Covering
- UVM Framework
- UVM Debug
- Portable Stimulus Basics
- SystemVerilog OOP
- Formal Verification
- Metrics in SoC Verification
- Verification Planning
- Introductory, Basic, and Advanced UVM
- Assertion-Based Verification
- FPGA Verification
- Testbench Acceleration
- PowerAware Verification
- Analog Mixed-Signal Verification

UVM and Coverage Online Methodology Cookbooks
Discussion Forum with more than 9625 topics
Verification Patterns Library

www.verificationacademy.com