

Prototyping Speeds Hardware Validation, Interface IP Certification

[Electronic Design](#)

[Hugo Faria Antonio Salazar](#)

Wed, 2015-08-26 15:31

The SoC design world is full of challenges and unforeseeable hurdles, especially for protocol developers and early adopters. Time-to-market (TTM) pressures, rapidly changing requirements due to competitive forces, specification updates, and tight budget restrictions put additional pressure on the development team.

Related

[Point/Counterpoint: Today's FPGA Prototyping—Breaking the Typecast Mold](#)

[Prototype Your Way To Success](#)

[Hardware Emulation: A Weapon of Mass Verification](#)

To accelerate TTM, software/hardware co-design becomes increasingly important, introducing a natural interdependence that extends from design to development, verification/prototyping, and all stages of production. Hardware elements that were traditionally considered static from a software perspective are now more flexible due to the use of FPGAs and emulation environments that permit IP core parameter exploration and allow for design modification at the hardware level. Consequently, the SoC development challenges are more complex and far-reaching, while placing more control and adaptability in the hands of developers.

The steady release of higher-performing and feature-rich IP interface protocols and standards enables designers to constantly improve their products, resulting in increasing pressure to release product updates more often. Now more than ever, the fast pace of updates comes via software. Most modern systems require software elements not only for interfaces, but also for low-level hardware resource management, including initialization, configuration, internal process definition, and hardware training.

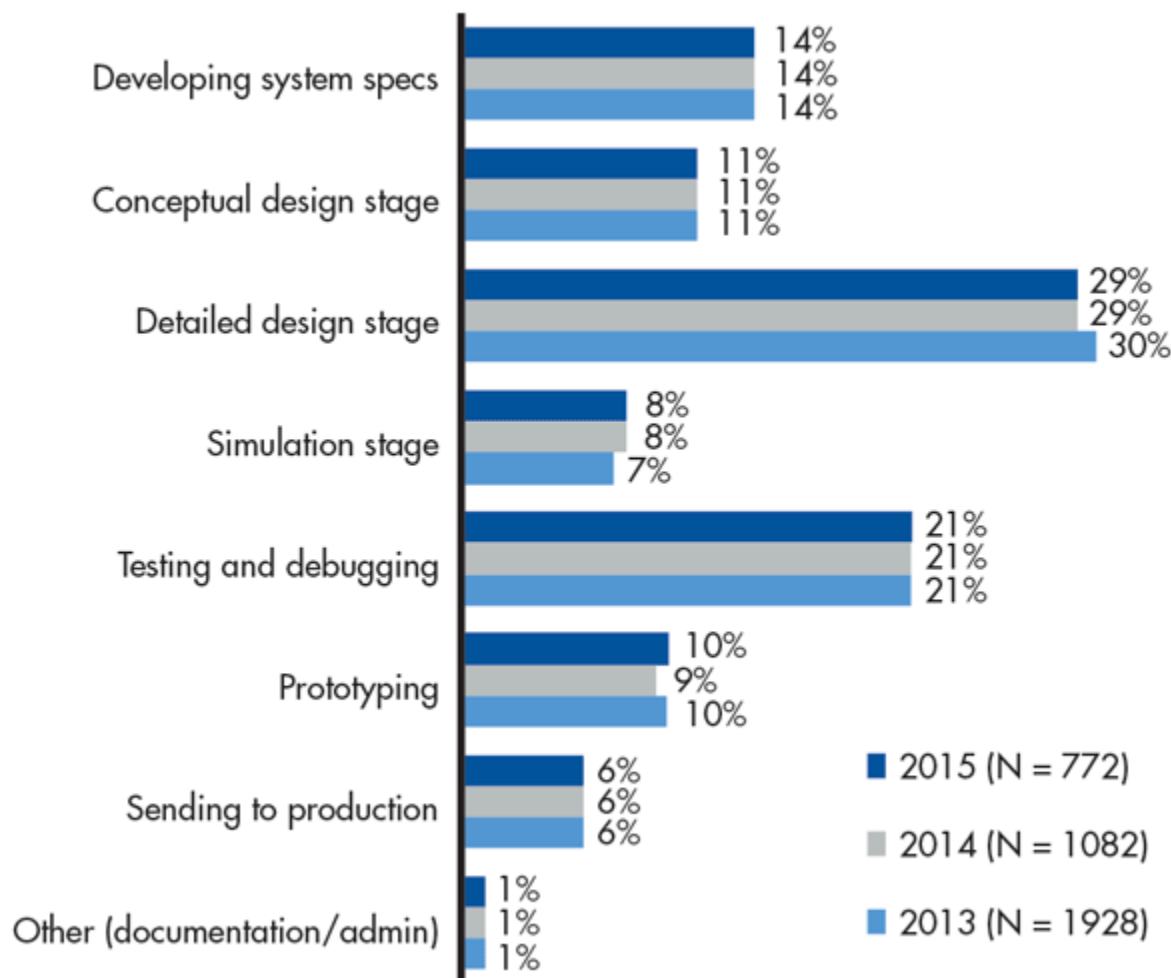
Software elements can appear as applications and/or drivers running in an OS environment (such as Linux) or RTOS, firmware, or the code for an internal core process. Every software element requires associated hardware updates, in particular when working at low levels of abstraction. Software development must take place at earlier stages of the SoC design process, which when referring to concept validation and interface IP certification, can drive software engineers deep into the specifications. The design bring-up process entails hardware parameter definitions with software-register configurations, which both empowers system designers with improved performance and features, and represents a challenge that requires new approaches and mindset.

Investing in Hardware Validation and Interface IP Certification

With all of these considerations, product developers must determine where to most effectively invest their time and effort on hardware validation and interface IP certification. Hardware validation aims to confirm that a target unit, which may include a complex combination of IP, satisfies its specified purpose and serves as an

ly window of functional interoperability.

Feature exploration, hardware/software co-design, early software development, and system validation are satisfied through hardware validation because it permits the functional corroboration of features and procedural flows that are otherwise confined to the simulation environment. Hardware validation is a powerful approach that can help streamline processes, catch issues early (avoiding exponential cost increases), confirm integration of specific features, and more.



<http://bd.eduweb.hhs.nl/es/2014-embedded-market-study-then-now-whats-nex...>

Despite its benefits, hardware validation alone doesn't guarantee functional validity and can be quite complicated due to the specifics of the approach/platform to be utilized. Most development flows include some type of prototyping stage. The UBM Tech 2014 Embedded Market Study reveals that prototyping maintains a consistent 10% of the amount of time spent at each stage of the design process (*Fig. 1*).

Interface IP certification takes hardware validation a step further by removing some of the subjective aspects and replacing them with agreed-upon specifications and defined testing procedures in the form of Compliant Testing Specifications (CTS). In other words, interface IP certification translates to industry-standard connectivity backed by regulation, arbitration, and standardization for interoperability with elements with corresponding compliance/certification.

Certification alone doesn't guarantee that the device actually works properly or can interoperate, though, so it requires an associated methodology and process flow. When a mainstream product utilizes standard protocols with known testing procedures and commercially available analyzers, end users can be assured that earlier

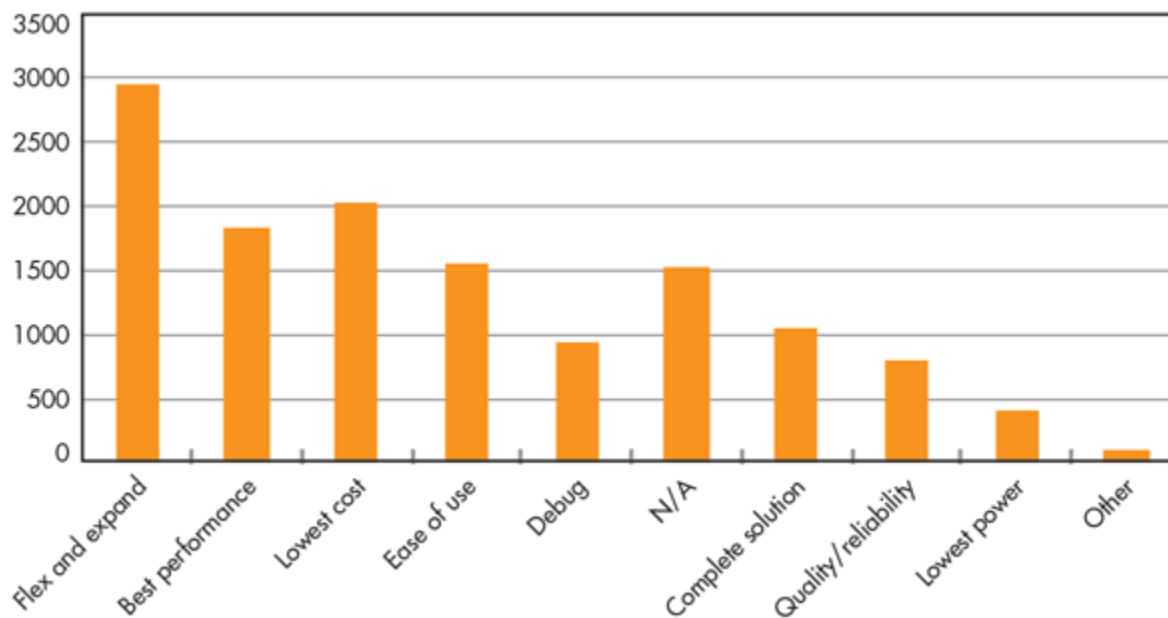
ducts have worked through ambiguous interpretation issues, debugging, and other inconsistencies.

However, as interface IP protocols are constantly being improved to address growing performance and/or feature needs, early adopters and designers must deal with the lag in the availability of testing specifications, instruments, and methodologies.

Hardware validation and interface IP certification can identify possible issues, functional shortcomings, or interoperability failures that seldom can be found via simulations or design analysis. The addition of testing that can address functional and parametric variables at multiple levels of the design introduces a robustness and resilience to most processes. In this regard, prototyping introduces a systematic approach that enables accelerated iterations and facilitates hardware validation.

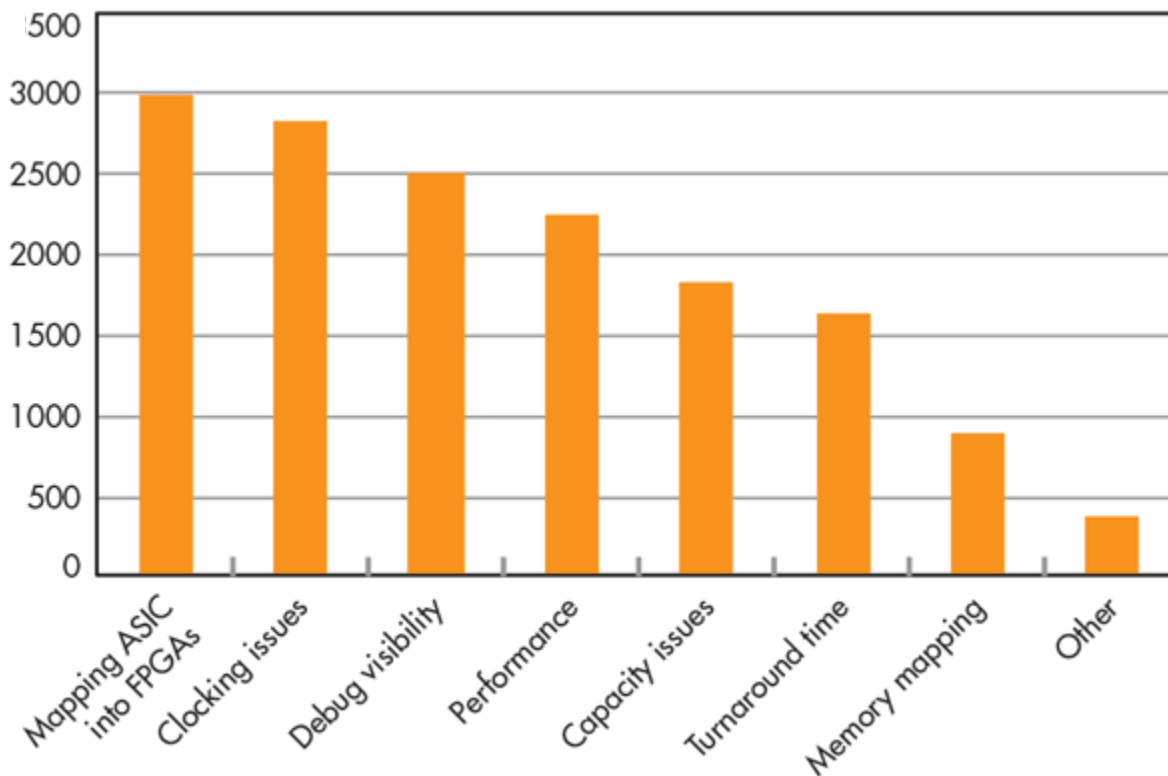
Benefits of FPGA-Based Prototyping

FPGA-based prototyping provides a flexible and adaptable environment that permits a wide array of testing and feature exploration. When combined with signal observability, FPGA-based prototyping represents a must-have advantage. In a survey conducted for [Synopsys' FPGA-Based Prototyping Methodology Manual \(FPMM\)](#), flexibility and expandability were identified as the leading criteria for prototyping platform selection, followed by low cost, performance and ease of use (*Fig. 2*).



Interestingly enough, while debug capabilities are situated in seventh place as a selection criteria, they are third when considering FPGA challenges, with ASIC mapping into FPGAs and clocking issues at the top of the list (*Fig. 3*). While flexibility and lowering cost are key selection criteria, the challenges aren't necessarily addressed by such criteria. Designers are aware that the ability to reutilize and/or expand the prototyping platform to cover evolving and unforeseen needs is critical.

When considering hardware validation and interface IP certification, FPGA-based IP prototyping provides clear advantages in a number of scenarios. For instance, IP prototyping permits the testing of situational variability and "real-life" conditions. Another advantage is the testing of scenarios that are complex to replicate in simulation/emulation, such as connectivity with end devices. FPGA-based IP prototyping also accelerates the validation of some features that would require hours of simulation.

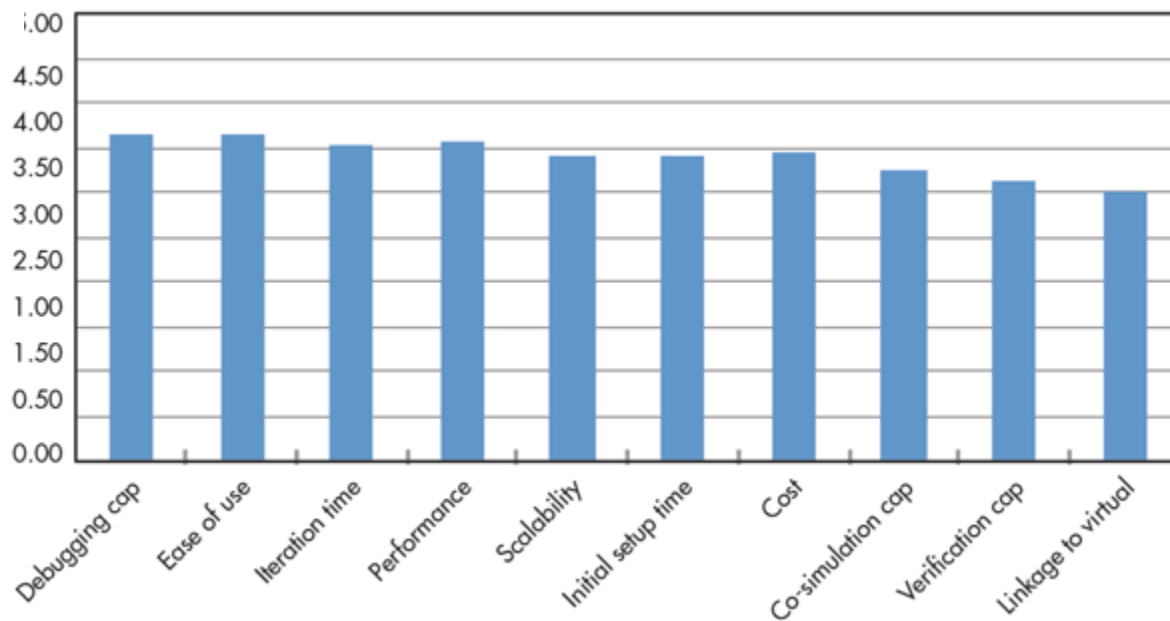


FPGA-based IP prototyping traditionally implies adapting the RTL and deviating from the target design. However, most designs are adapted for simulation regressions and design for testability. Moreover, the cost of not achieving first-time-right silicon outweighs the challenges of adopting additional methodologies that can add robustness to a design, such as FPGA-based IP prototyping.

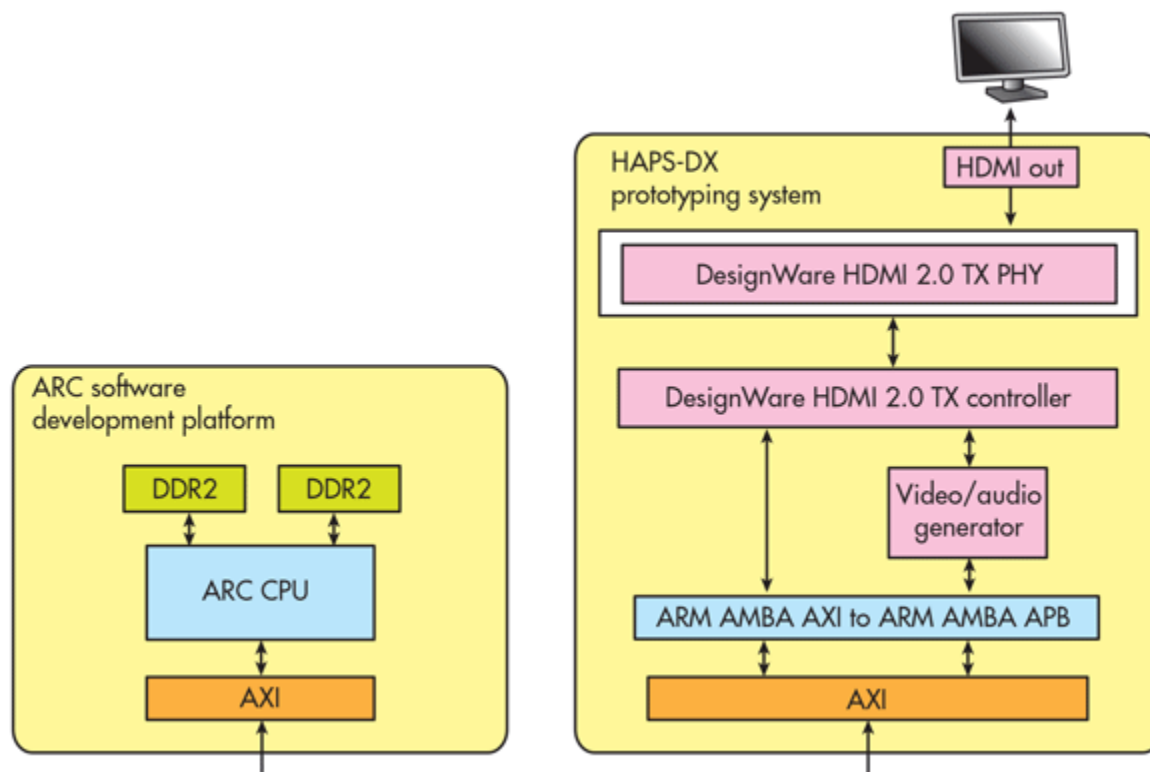
FPGA-based IP prototyping introduces configurability, controllability, and observability to designs while preserving a significant portion of the target ASIC/SoC RTL. Therefore, both the functional concept of the design is being tested along with a considerable part of the actual code. Added to this is the flexibility to connect to protocol/logic analyzers, commercial devices, and software development boards, and to confirm interaction with target PHYs, modules, and/or sensors.

Required Features for FPGA-Based IP Prototyping Platforms

Not all FPGA-based prototyping platforms are the same, and FPMM survey participants set debugging capabilities and ease of use at the top of their feature lists, followed by performance and iteration time (*Fig. 4*). Prototyping is a venue for design exploration, which implies that being able to locate what is wrong in a design or explore performance/functional issues remains a top priority. Ease of use—focusing on the design, not the tool—follows naturally.



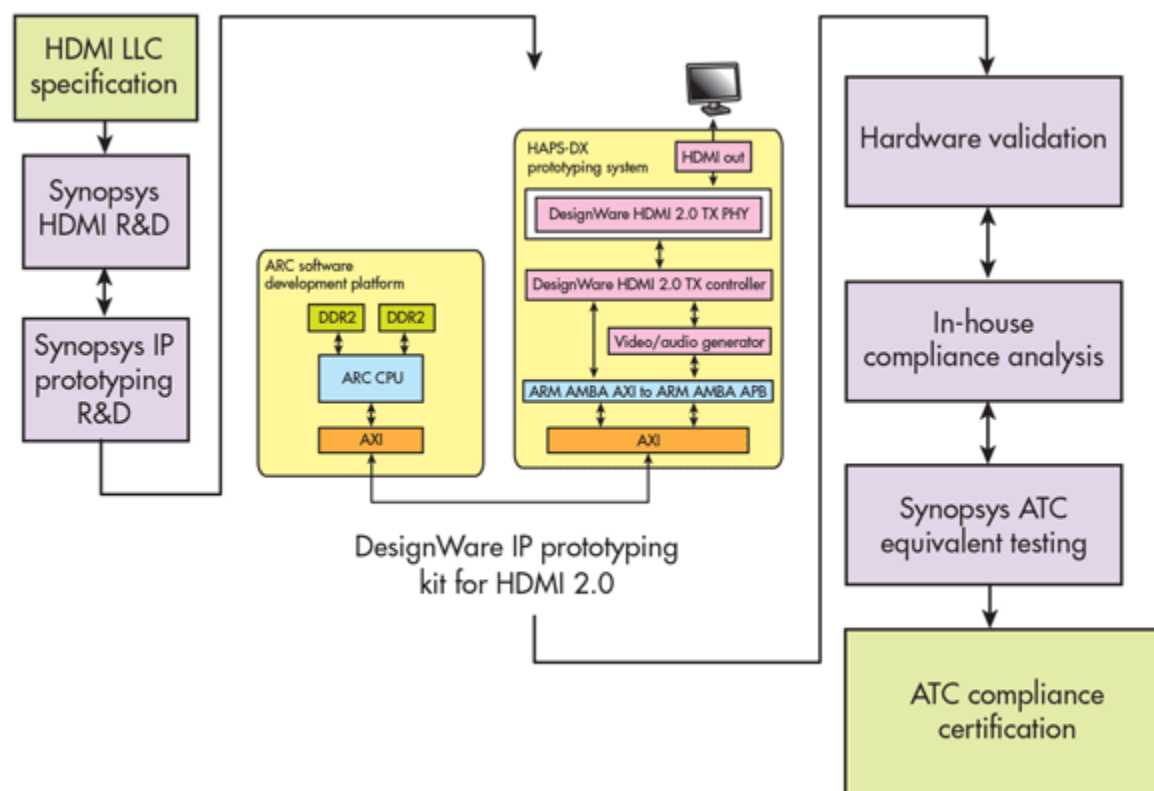
Synopsys' internal DesignWare HDMI 2.0 TX/RX hardware validation and certification process illustrates the advantages of FPGA-based IP prototyping. DesignWare IP Prototyping Kits for HDMI 2.0 TX and RX are at the heart of Synopsys' hardware-validation process (Fig. 5). The kits include instantiations of the DesignWare IP, including all of the logical and standardized flow processes for synthesis, simulation, and software development.



The kits provide a modular, mobile, and adaptable solution based on the HAPS-DX FPGA-based prototyping platform, which allows software/hardware exploration as well as a uniform environment supported by hardware-aware software that accelerates iterations. A clear advantage of such an approach is that each kit can quickly adapt and conform to testing instrumentation or changing environment and specifications.

Compliance Testing with Prototyping Kits

The HDMI hardware validation and certification flow followed at Synopsys (*Fig. 6*) can be viewed as a streamlined framework, with well-defined stages toward the final objective—to achieve IP certification as quickly and efficiently as possible, using in-house prototyping expertise and products as needed. The starting point is set by the R&D release of a new controller or PHY IP version, traditionally related to new specifications or customer-requested features.



In the meantime, the software and hardware teams update the firmware and software. They consider all new requirements and software flows needed to make the hardware work as expected and prepare the IP for certification.

The next step is the hardware prototyping. When the new controller or PHY is released, the hardware team starts the implementation of the FPGA build, according to the objectives and specifications for not only the IP Prototyping Kit itself, but also for the new controller and/or PHY versions.

At bring-up, the software team loads the bitfile and tests the new FPGA build. This is done first with basic tests, and work is done toward full functionality verification, in cooperation with the hardware team. However, new features sometimes are more cutting-edge than the test equipment's capabilities. Synopsys can bypass this challenge by testing interoperability of new features with another of the company's IP products—this is the advantage of developing both the TX and RX sides of a protocol. When the certification tests are developed and deployed, the IP can earn final certification from the standards body.

For all other features, when the software and hardware is considered stable, the IP passes to the in-house equivalent of an authorized testing center (ATC) for testing. The in-house ATC performs testing by a group that did not participate in the product development; therefore, the team is not naturally biased. If issues are found and the test does not pass, then the product goes back to the in-house compliance analysis step. When all the tests are passed, then the system is sent to an official ATC for certification.

Summary

Compliance testing certification of IP provides SoC designers with increased confidence of its interoperability with similarly compliant IP and devices. This translates to an increased marketability of the end product, as it conforms to the specifications and requirements established by the specification body, such as the HDMI LLC. Nevertheless, SoC and system architects must remember that passing compliance or certification only indicates achievement of the minimum testing requirements. Compliance doesn't guarantee interoperability or even functionality, as these standards are the responsibility of the integrator and adopter.

These apparently contradictory statements actually address the subtlety of product robustness through associated work flows, i.e., the testing process within itself isn't a guarantee of final product functionality. The robustness of the entire process is important. Moreover, end-to-end solutions verified by the same provider have the added value of implicit guaranteed interoperability, cooperative testing, additional supported features, and flow methodology equivalence.

Although certifications are an important aspect of interoperability confidence, imperative for product marketability, it's the quality and prototyping process itself that needs to guarantee the functional aspects of the final product. Prototyping frameworks that accelerate validation, exploration, and interconnectivity analysis, as well as serve as a staging area to a number of possible development paths, has a truly positive impact in most development stages and end products.

For more information, go to <https://www.synopsys.com/IP/ip-accelerated/ip-prototyping-kits/Pages/default.aspx>.

Source URL: <http://electronicdesign.com/test-measurement/prototyping-speeds-hardware-validation-interface-ip-certification>