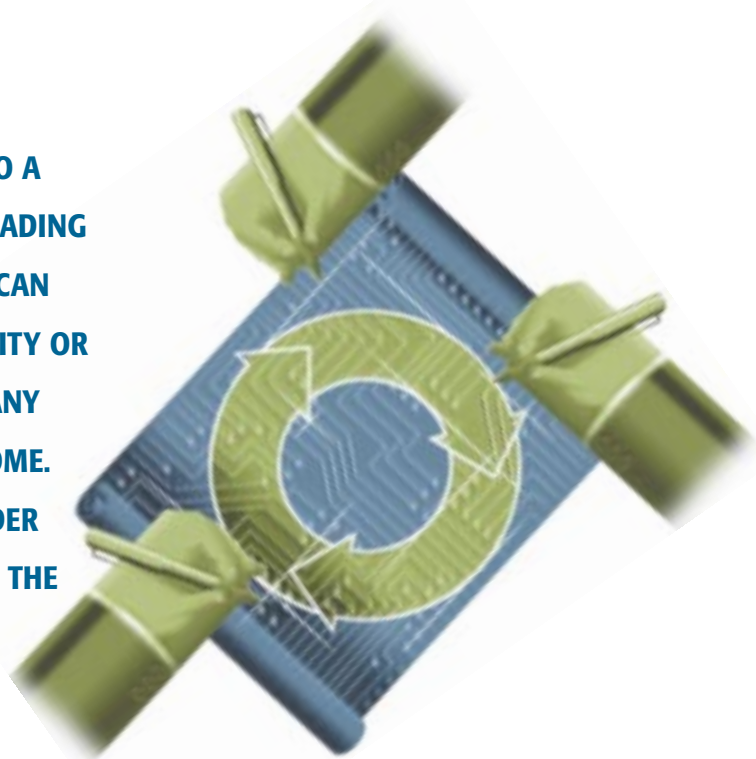


INTEGRATING THIRD-PARTY CORES INTO A DESIGN REQUIRES MORE THAN JUST READING A DATA SHEET. VIRTUAL COMPONENTS CAN GREATLY ENHANCE DESIGN PRODUCTIVITY OR DOOM A PROJECT TO FAILURE, AND MANY FACTORS DETERMINE THE FINAL OUTCOME. TO ENSURE SUCCESS, THE CORE PROVIDER MUST BECOME A TRUSTED MEMBER OF THE DESIGN TEAM.



Your core, my design, our problem

<i>At a glance</i>	58
<i>Verification IP helps</i>	
<i>SOC design</i>	60
<i>For more information</i>	62

FOLLOWING THE PRECEPT that it is better to reuse than to reinvent, the electronics industry has created a new business segment: commerce in IP (intellectual-property) cores, also known as virtual components. This article refers to reusable logic as a virtual

component because the term “IP” refers to more than just electronics design and has other connotations in our industry. The premise seems to be a matter of common sense: Instead of investing money designing circuitry to implement a function that someone else has already implemented, obtain that implementation and integrate it into your design.

Venture capitalists have funded a number of companies specializing in the development and sale of reusable cores, and those companies had expected a steep ramp-up in business. Systems companies also had believed that they could exploit the new market by reselling parts of designs they had already developed.

Reality turned out to be a bit different. Designing and marketing reusable logic is not as simple as just designing some familiar circuitry, and a number of promising companies did not survive their first year or two of operation. The use of a third-party virtual component in a design also turned out to be more difficult than expected. The industry originally assumed that a virtual component was just another off-the-shelf part. Therefore, given an adequate data sheet, an engineer would be able to choose the correct part and use it, just as a pc-board designer uses a discrete component.

When designing an SOC (system on chip), engineers must consider a design

Illustration by Dan Guidera

as an artist considers a mosaic. You use each block not only for its color or for how well it looks with its neighboring pieces, but also for its contribution to the entire design. You, therefore, choose a virtual component keeping in mind the contributions it will make to the entire design and the requirements it will put on the entire design. Recently, virtual-component vendors were offering a large variety of components ranging from ones that handled simple functions to microprocessors. Today, independent virtual-component providers concentrate on products that implement complex functions, such as DSPs, controllers, processors, and protocol handlers.

Semiconductor vendors and foundries still offer an inventory of simple reusable blocks, such as adders, multipliers, decoders, FIFOs, and so on. FPGA vendors have found that offering their customers an inventory of predesigned and tested basic logic and arithmetic functions is a competitive requirement, but designers of application-specific chips are more interested in the availability of complex functions. Both foundries and FPGA vendors have established OEM agreements with the providers of the most popular virtual components, giving system designers better quality as well as foundry and technology choices.

A NEED FOR STANDARDS

In 1998, Pierre Bricaud of Mentor Graphics and Michael Keating of Synopsys authored the *Reuse Methodology Manual*, which provides a guideline for the development and integration of virtual components in a system (Reference 1). Synopsys and Mentor Graphics used the book to develop the Open Measure of Reuse Excellence (OpenMore). OpenMore is an assessment program and spreadsheet that helps users determine the reusability of cores in designs. The two companies donated OpenMore to the VSIA (Virtual Silicon Initiative Alliance) in June 2001. The consortium plans to use the program to create an online metric that will assist users and vendors in developing and integrating quality virtual components into their designs. Despite these efforts, the design and integration of virtual components does not follow a standard methodology but depends almost entirely on the provider and integrator for a successful project-

AT A GLANCE

- ▶ SOC complexity fosters design reuse and third-party virtual-component integration.
- ▶ A standard methodology would help to diminish costs and shorten schedules.
- ▶ Deep-submicron processes require both behavioral and physical verification of designs.
- ▶ By using third-party vendors as if they were an integral part of the design team, managers can improve product quality.

management strategy. A panel at the 2001 Design Automation Conference exploring the use of third-party cores concluded that, because the development process depends on EDA tools and because of the lack of standards, vendor support is required to ensure successful core integration (Reference 2).

The first issue facing a virtual-component provider is deciding what to develop and market. The safe choice is to fully or partially implement an industry standard. The fact that a standard has been developed means that the industry needs it and that a market understands it. It also means that a defined interface exists between the portion of the design that the standard describes and the rest of the system. Therefore, a product that implements a standard requires less documentation and support effort than a fully proprietary one. Noteworthy exceptions to this rule are processors developed by companies such as ARM and Tensilica, which have found acceptance by filling a void in the market in a timely fashion. These processors have established their own de facto standards, such as the ARM AMBA (Advanced Microprocessor Bus Architecture) bus. Yet, ARM, which licenses its virtual components through its silicon partners, has found that it must support a number of design flows to accommodate the different tools that each of its licensees use.

Processors also require that designers be able to test and debug the hardware/software integration, and vendors must also ensure that software-development platforms are available. Tensilica provides a prototype evaluation board

for software development using FPGAs. Of course, because the product has an extendable instruction set, the prototyping board must offer a modifiable bus to interface between the processor and the rest of the board. The interface to the rest of the system is fixed, and the interface with the processor is reconfigurable. When you use only one core or when all of the virtual components in a design come from the same vendor, you can use a platform-based development methodology. Many companies are offering products in this market, and more, such as Mentor Graphics with Platform Express, are expected to enter the fray (Reference 3).

PHYSICAL CONSIDERATIONS

Once a vendor identifies a market, it must implement the virtual components. To increase the quality of the product, a vendor must consider not just how the core functions in stand-alone mode, but also how it behaves as an integral part of a system. One aspect to consider, for example, is the I/O requirements that the core imposes on the rest of the design. Designers want to minimize both the frequency and the size of traffic between the virtual component and the rest of the system. Also, if the vendor plans to allow the use of the virtual component in a less-than-0.2-micron process, its designers must also consider signal integrity, power consumption, electromagnetic radiation, and manufacturability during the design. NurLogic, a provider of virtual components for high-speed networks, found that a lack of standards has increased the costs of behavior modeling, timing analysis, and place-and-route support for its products. Most Nurlogic customers are using 0.13- or 0.18-micron processes, and the company must support foundry-dependent run sets for design-rule checks and layout-versus-schematics verification. In other words, the company must run a separate verification for each foundry it supports. Timing closure also presents a problem, especially if the system uses virtual components from more than one vendor. Each vendor is likely to have used different tools for the stand-alone verification, thus forcing the integrator to develop a new script and to correlate the results with data obtained from unfamiliar tools.

Analog virtual components require

physical modifications to obtain the proper performance characteristics, so vendors must offer their customers an option that eases this task. Barcelona Design addresses the requirement by offering a Web-based virtual-component-production environment. Customers pick the type of analog core, choose the foundry, input the values for the appropriate physical parameters, and receive back the virtual component that meets their specifications. Antrim Design Systems has adopted a different approach. Its customers receive not only the virtual component, but also a set of tools called Analysis Characterization Verification. The package supports the behavioral simulation of the core in Verilog-A, synthesis using Antrim's tool, and characterization of the resulting circuit.

VERIFICATION BOTTLENECK

One of the most challenging aspects of SOC development is design verification.

Developing a testbench or a collection of testbenches with the associated test vectors is time-consuming. Executing the required simulation and analyzing the results are tasks that require not just a significant amount of computing resources, but also a significant amount of insight into the behavior of the design (Reference 4). Statistics show that verification is the most time-consuming aspect of product development, and it demands the most resources (Reference 3). Both vendors and users of virtual components face significant verification problems. Vendors must not only ensure the quality of their products by exhaustively testing the behavior and physical characteristics of the core, but also anticipate system-level functions. Designers must develop tests at the functional level that simulate both correct and erroneous use of the device. It is always more difficult to develop meaningful tests that simulate incorrect use, because

predicting misunderstanding and accidental errors requires a careful study of the written specification and insight into human frailty.

If the core implements functions that are part of an industry or de facto standard, documenting the product becomes easier, because the vendor can reference the specification and assume that users have a certain level of familiarity with its intended operation. Yet, there will be areas peculiar to the product; after all, differentiation is a significant marketing weapon. The verification process must span the entire design flow—from system-level design to layout, because deep-submicron issues force designers to consider silicon-level issues even during system design and partitioning. Qualis Design Systems has found that by developing a good verification strategy, a company can produce a foundation of reusable verification products that addresses specific application areas.

VERIFICATION IP HELPS SYSTEM-ON-CHIP DESIGN

Recently, many communications systems have incorporated new optical connections to fully use the data bandwidth available in OTNs (optical-transport networks). New standards ensure interoperability of OTN systems from different vendors. Zaiq Technologies has developed optical-channel-verification IP, which generates data frames used in OTN. The optical channel can carry IP, ATM, Ethernet, or SONET/SDH packets mapped into optical-channel frames (Figure A). The ITU has standardized optical-channel as recommendation G.709.

Optical-channel frames comprise three levels of encapsulation, each with corresponding headers carrying signaling and maintenance information. To correct optical-transmission errors, such as spectral dispersion, the optical channel uses FEC (forward-error-correction) encoding—the most common being Reed Solomon encoding. The FEC-encoded optical-channel frames are then scrambled

and passed to a serializer for optical transmission. The reverse operation consists of optical-channel frame delineation, descrambling FEC decoding (and error correction), and extraction (demapping) of data.

Accelerating SOC verification is critical. For example, Zaiq Technologies Inc has developed optical-channel-verification IP, which comprises a comprehensive test suite that exercises all the features defined in recommendation G.709, in addition to a robust test environment that you can use for stand-alone verification of optical-channel RTL as well as system-level verification that you can migrate to the lab. The optical-channel test environment not only can generate data frames in the required OTN formats, but can receive frames and send results back to a common database for functional and performance analysis, according to Alex Genusov of Zaiq.

The verification methodology, Sylver, is based on C and C++. It

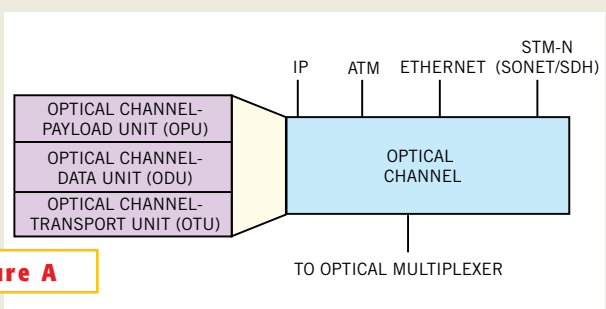


Figure A

The ITU has standardized data frames that optical channels carry.

provides powerful data generation and comparison, flexible simulation control, and an easy-to-use test style. The Sylver methodology uses transaction-based verification, significantly reducing the overhead of the verification environment in simulation.

Zaiq's optical-channel-verification IP has been used to verify SOC (system-on-chip) designs implementing the optical-channel function. This SOC incorporated many customer-specific features, including a micro-processor interface, overhead processing, and single-channel

and multichannel modes, among others.

The starting point was the optical-channel environment enhanced to support additional design features. Engineers used the optical-channel tests as a basis for the SOC verification with minor modifications. Because optical-channel-verification IP development overlapped SOC verification, the gain in optical-channel-verification functions was approximately 50%; complete optical-channel-verification IP, however, provided a nearly 70% gain.

Some virtual-component vendors have recognized the value of reusable verification elements and are either bundling it with cores or selling it separately (see **sidebar** “Verification IP helps SOC design”). Tensilica offers TIE (Tensilica instruction extension), a kit that helps customers verify that the new instructions used to tailor the processor to the design work with the fixed portion of the processor architecture. The most effective way to improve quality, shorten project schedules, and decrease costs is to improve the development methodology. In many SOC designs, a team of designers, often geographically distributed, implements the product. Managers need to consider a third-party virtual component not as an off-the-shelf component but as a design block developed by a group that is part of the team. Conversely, a virtual-component vendor must be prepared to act as a team member. Services are a viable part of the business, and, to be successful, virtual-component companies must be able to provide their customers with design and verification support.

DOCUMENTATION

The main problem with precisely describing the behavior of a product or its expected use at both the physical and the behavioral level is the language used for the description. Most documentation is

in English, which has reasonably complex syntax and very complex semantics. Although documentation generally describes salient features well, it rarely precisely conveys the nuances of a product’s behavior. By its nature, English (or any other human language for that matter) gives users an incomplete understanding of a product’s capabilities; thus, it becomes difficult for users to develop complete test suites.

Development work is ongoing to find a suitable language to use in writing a specification that a computer can parse, so that verification engineers can automatically and reliably derive a complete and accurate test suite. Developers have used UML (Unified Modeling Language) to describe software systems, and some companies are experimenting with it for describing hardware specifications (**Reference 3**). Formal verification techniques also offer a way to more rigorously specify hardware structures and behavior. 0-In Design Automation offers CheckerWare and CheckerWare Monitors that allow core developers to make assertions about a design so that users can develop a more formal set of testbenches. Tensilica uses some 0-In products to provide guidance to its customers in developing a test suite. A number of EDA companies are now offering formal verification products. Once you can combine formal

FOR MORE INFORMATION...

For more information on products such as those discussed in this article, go to www.ednmag.com and click on the Reader Service link under the Tools & Services section. When you contact any of the following manufacturers directly, please let them know you read about their products in *EDN*.

Antrim Design Systems
1-831-430-1900
www.antrim.com
Enter No. 321

Mentor Graphics Corp
1-503-685-7000
www.mentor.com
Enter No. 324

Synopsys Corp
1-650-584-5000
www.synopsys.com
Enter No. 327

Zaiq Technologies Inc
1-781-932-2442
www.zaiqtechnologies.com
Enter No. 330

ARM Holdings plc
www.arm.com
Enter No. 322

Nurlogic Design Inc
1-858-455-7570
www.nurlogic.com
Enter No. 325

Tensilica Inc
1-408-879-1990
www.tensilica.com
Enter No. 328

0-In Design Automation
1-408-487-3640
www.0-in.com
Enter No. 331

Barcelona Design Inc
1-408-245-2442
www.barcelonadesign.com
Enter No. 323

Qualis Design Corp
1-503-670-7200
www.qualis.com
Enter No. 326

Virtual Silicon Initiative Alliance
www.vsi.org
Enter No. 329

SUPER INFO NUMBER

For more information on the products available from all of the vendors listed in this box, go to www.ednmag.com, click on the Reader Service link, and enter no. 332

techniques with a well-defined specification language, such as UML or Rosetta, product documentation and verification will become respectively more precise and more accurate. Given that requirements or interface errors cause around 25% of bugs, such methods will significantly impact the cost of electronics products.

In the final analysis, the best way to increase the probability of success is to intelligently use the accumulated knowledge of all the project participants. Therefore, when using a third-party virtual component, it is best to build a team that includes the vendor, the EDA-tool provider, and the silicon foundry, as well as the product developers. Don't be afraid to ask for help and be prepared to pay for it. Larger profits from high-quality and timely products will more than compensate for the original investment. □

REFERENCES

1. Bricaud, Pierre, and Michael Keating, *Reuse Methodology Manual*, Kluwer Academic Publishers, 1998.
2. Collins, Nanette, David Kelf, and Gabe Moretti, "Your core—my problem? Integration and verification of IP," proceedings of the 2001 Design Automation Conference, ACM Order 477010, pg 170.
3. Moretti, Gabe, "Hardware tools aid engineers in design verification," *EDN*, Aug 30, 2001, pg 77.
4. Moretti, Gabe, "Compute farms: the new data centers," *EDN*, July 19, 2001, pg 54.
5. Douglass, Bruce Powel, "UML for executable specification," *EDN*, Aug 16, 2001, pg 83.



Gabe Moretti successfully developed and sold virtual components in 1991, before it became fashionable. His experience convinced him that third-party virtual-component integration is like trying to make friends: Sometimes, it works; sometimes, it doesn't. When it doesn't, it is helpful if you discover the cause of the failure, so that you can avoid or employ certain methods in the future, as circumstances dictate. You can reach Technical Editor Gabe Moretti at 1-303-517-2328, e-mail gabe@eda.org