

BASICS of Design

SoC DESIGN

System-on-a-Chip 101: Adding Intellectual Property

Dave Bursky, Digital ICs/DSP Editor

System-on-a-chip (SoC) design is a complex effort that encompasses many disciplines, from circuit design to thermal management to testing. When working with a design library from a particular ASIC supplier, creating an SoC solution can be relatively straightforward because the library's circuit building blocks are designed to work together. However, the challenge gets more complex when designs must incorporate blocks of intellectual property (IP) acquired from third-party suppliers into the design flow. Such blocks may follow different design rules, have incompatible interfaces, and require unique test support. However, there are ways to mitigate the potential incompatibilities. Sticking to a set of guidelines will smooth the IP's path into the design.

Today's multi-megagate SoC designs often include many different blocks of IP to reduce design time. In the example shown in Figure 1, the left half of a highly integrated SoC combines a RISC processor with associated instruction and data caches, interrupt and DMA controllers, a double-data-rate SDRAM controller, a local

Following several simple guidelines makes it easy to incorporate blocks of intellectual property when designing systems-on-a-chip.

The Guidelines To Ease Integration

1 Select the IP by considering more than just the functionality. Be sure to examine the accompanying documentation, test data, models, and other support data.

2 Determine the on-chip bus and the interface that will be needed to tie the IP block into the rest of the system. Encapsulate the IP block with the appro-

appropriate signal translation logic (a wrapper) if the available signals don't match those of the on-chip bus.

3 Ensure that you have all "views" of the IP block required to do system design, modeling, and simulation or emulation. These include block diagrams, C-based software models, an open-core protocol model,

and a Verilog or other HDL model.

4 Maintain the ability to use vendor-supplied testbench information by adding an inverse wrapper to recreate the original signal interface to the core. That will allow the original vendor-supplied test procedures to be run once the core is integrated into the SoC design.

The Custom SoC that lets you call all the shots,

with a six month wrap!



No compromise, no kidding.

The Toshiba SoCMosaic™ custom chip utilizes a flexible, Soft IP Platform that allows you to mix and match high-speed I/O, mixed-signal, and other IP to meet your SoC performance and size specs. Plus, starting with your block diagram, we can usually deliver the chip you want in less than six months. The Toshiba Soft IP platform approach can also save you money—up to 50% or more of development costs—with no production unit cost penalty.

Methodologies	Any Digital IP Including MPU and DSP	Analog/ Mixed-Signal IP	High-Speed I/O	Optimum Die Area, Performance, Power	Over 20M Gates + Flexible Embedded Memory	Efficient Platform IP Reusability	System Debug Platform
FPGA	✓	✓	✓				✓
Structured Array	✓	✓	✓				
SoCMosaic	✓	✓	✓	✓	✓	✓	✓

The light colored check marks represent limited features.

Wrap it up.

Get your Custom SoC to market faster, with no compromises, and keep full design control with the Toshiba SoCMosaic custom chip. The world's third-largest semiconductor manufacturer, Toshiba is now producing Custom SoCs using the latest 130 and 90 nm process technology, with a wide variety of packaging and complete software support. Find out more, and enter to win your own, personalized director's chair at www.SoCMosaic.Toshiba.com. No purchase necessary.



TOSHIBA

scratchpad memory (8 kbytes), and JTAG test support. The other half of the chip includes complex I/O functions such as a 10/100/1000 Ethernet media access controller (MAC), a PCI interface block, a USB interface, and possibly a Bluetooth baseband controller. These blocks can either be selected from the SoC vendor's design library or imported into the design from an independent IP supplier.

Also shown is a block of custom logic (which may include imported IP) that can implement special functions required by the designer. Such logic will uniquely define the chip's functionality. Between the left and right sections is a bus structure or other interconnect structure that provides the control, datapaths, and other signals needed for the blocks to all function together.

Each SoC design will be different. Some may incorporate dozens of CPUs or multiple Ethernet MACs, a crossbar switch rather than a bus for block-to-block connections, or other unique functions. Numerous small-scale blocks of IP, ranging from gates to DMA controllers, come with most design libraries. Larger, more complex blocks, such as PCI interface controllers, interrupt controllers, MACs for local-area networks (LANs), universal serial bus interfaces, and memory controllers, are other fairly common offerings from ASIC suppliers, foundries, and third-party design companies. The largest blocks, such as CPUs, cache memories, embedded blocks of SRAM and DRAM, are also rather readily available. The design challenge is to find ways to get all

Picking the Correct Fork in the Road

When starting an SoC design that requires externally supplied IP, many decisions must be made early in the process to smooth the path. One approach might be to set up an IP qualification checklist to make sure all necessary issues are addressed. The checklist can be set up in four different categories, or levels, to categorize the degree to which the IP block meets all of the criteria defined for the project (see the table).

Until a year or two ago, level D in the checklist represented the smallest set of information requirements typically provided by IP vendors. Still, today it's more common to get the information summed up in level C, which not only includes the IP source files, but also an RTL simulation environment, the simulation vectors, synthesis scripts and constraints, and even the open-core protocol (OCP) model of the bus interface and its test requirements. As system designs grow in complexity, IP vendors must supply more data about the IP blocks. Consequently, the industry has started to focus on the level B requirements, which, in addition to proving silicon functionality, better show designers how to incorporate the IP block into the overall ASIC design flow. The information enumerates special cell design requirements, provides external interface models, and even supplies clocking and layout tips.

For some of the most complex blocks, such as CPU or DSP cores, the level A requirements provide the highest level of design support. On top of all the level B requirements, level A adds more application support in the form of application libraries, software-development tools, and software such as real-time operating-system software for CPU and DSP cores. Additional test support, including assertion-based verification and formal verification techniques, make it easier for designers to verify correct operation of the block. But overall, using such a checklist is just one of the first steps you must take.

of these blocks to work cohesively in an SoC implementation.

The Ties That Bind

As you start planning the SoC design, the architecture, which includes the on-chip bus or intercon-

INTELLECTUAL PROPERTY: QUALIFICATION CHECKLIST FOR A SOFT IP PLATFORM

	Requirement	Deliverable format
LEVEL D	User guide	
	User guide, instruction set in case of CPU	Documents
	Core design	
	Core RTL source files	Data
	RTL testbench environment	Data
LEVEL C	User guide	
	Specification, user guide, instruction set in case of CPU	Documents
	I/O pin descriptions and timing diagrams	Documents
	Core design	
	Core RTL source files	Data
	RTL simulation environments	Data
	RTL simulation vectors, directed functional test, and constraint random test	Data
	Logic design	
	Synthesis scripts and constraints	Data
	OCP interface	
OCP specification of core	Documents	
OCP co-recreator package	Data/documents	
OCP testbenches	Data/documents	
LEVEL B	User guide	
	Silicon proven	Documents
	ASIC implementation and flow	Documents
	Special cell requirement	Documents
	System design	
	Simulation stimuli and test results for system verification	Data
	External interface models	Data/documents
Logic design		
Clock design	Data/documents	
Layout guideline	Data	
Test requirements		
Test methodology	Documents	
LEVEL A	All level B requirements	Data/documents
	User guide	
	Software tool manuals	Documents
	DSP applications and suppliers	Documents
	CPU RTOS	Documents
	System Design	
	Performance evaluation environments	Data/documents
	Assertion-based verification and formal verification technique	Data/documents
C code and device driver	Data/documents	

Notes: Level D - deliverables for most IPs today. Level C - minimal requirements of IPs for efficient implementation in SoC systems. Level B - deliverables for silicon proven IPs (i.e., have passed through synthesis and back-end process). Level A - full deliverables for IPs that have the additional supporting elements needed or full hardware/software-level integration. Table courtesy of Toshiba America Electronic Components Inc.

passed through synthesis and back-end process). Level A - full deliverables for IPs that have the additional supporting elements needed or full hardware/software-level integration. Table courtesy of Toshiba America Electronic Components Inc.

Co-Integration Connection

An SoC custom chip design contains a typical mixture of IP blocks that designers might co-integrate: a CPU subsystem, complex I/O support blocks, and some custom logic. This illustration includes blocks from the chip manufacturer's library, as well as blocks licensed from third-party IP suppliers.

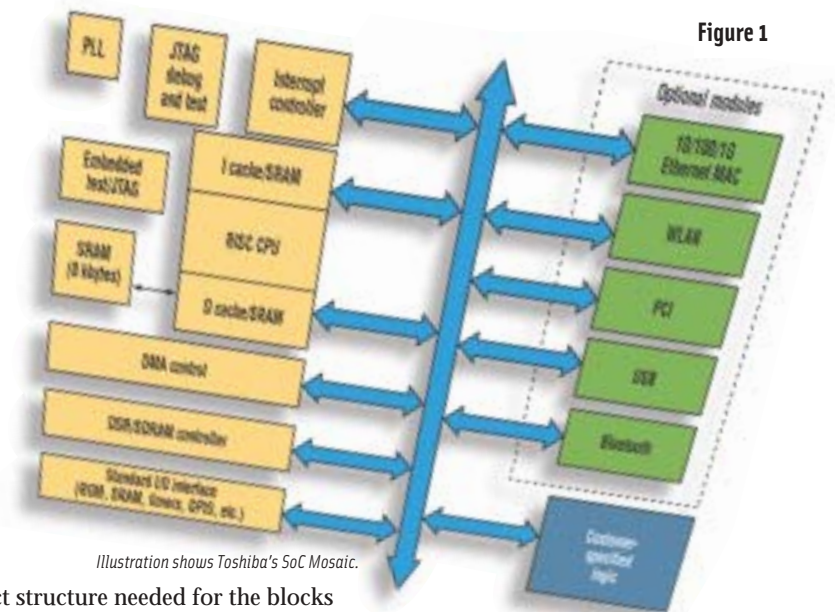


Illustration shows Toshiba's SoC Mosaic.

nect structure needed for the blocks to intercommunicate, must be defined early in the overall process. Several well-defined buses that were "standardized" by the Virtual Socket Interface Alliance (VSIA) are now listed on the VSIA's Web site (www.vsia.org). Some of the criteria that should be considered for bus selection include easy turnaround, timing closure, configurability, robustness, and bus efficiency. Of course, you can also create your own bus if no available option meets your performance criteria. The VSIA also started work to define a quality metric for evaluating IP blocks and the data supplied by the IP vendor. This will ensure that the blocks come with all data needed to integrate them into the SoC.

Once you select a candidate block of IP and run it through the checklist to prove that it meets all criteria to start the integration process, the next challenge is to integrate it into the rest of the design (see the table). In many cases, that's a straightforward effort. However, many blocks may not have the specific signal outputs or signal inputs needed for a direct interface to the SoC you're creating.

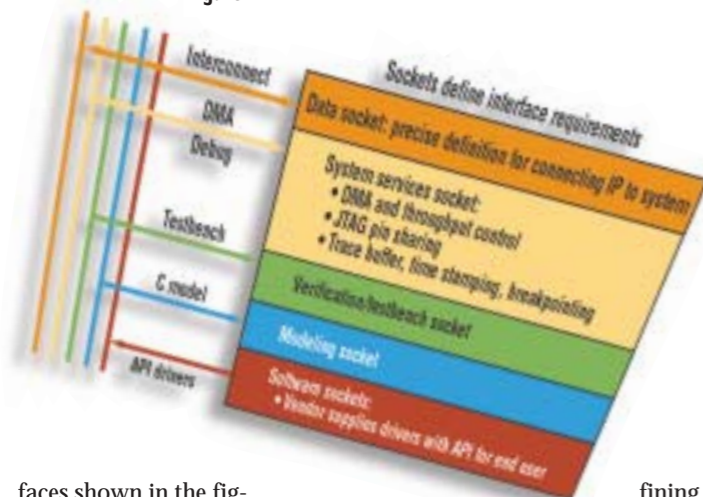
One approach that makes it easier to merge in blocks that may not be directly compatible is to surround the block in a wrapper (sometimes called encapsulation). The wrapper contains circuitry that translates the IP block's signals into a standardized set of signals that ties into the on-chip bus (Fig. 2). This approach tries to level the playing field by standardizing the interface for blocks of IP imported into the design framework.

Many aspects of the standardized and well defined interface, as outlined in Figure 2, help designers integrate the block into the rest of the chip design. For instance, the most basic aspect of the interface is the data socket—a definition for connecting the IP to the on-chip system bus. Following that is what some call the system services socket—a definition of the underlying support and testability aspects of the IP block. Included in this socket are aspects such as DMA and throughput control, JTAG pin sharing, trace buffer control, time stamping, and breakpoint control. Additional socket inter-

Sockets That Fit

A well-defined interface between the IP block and the on-chip bus eases the task of integrating the IP into the design. By encapsulating IP blocks with some additional logic, the bus interface can be standardized to what has become known as the socket interface.

Figure 2



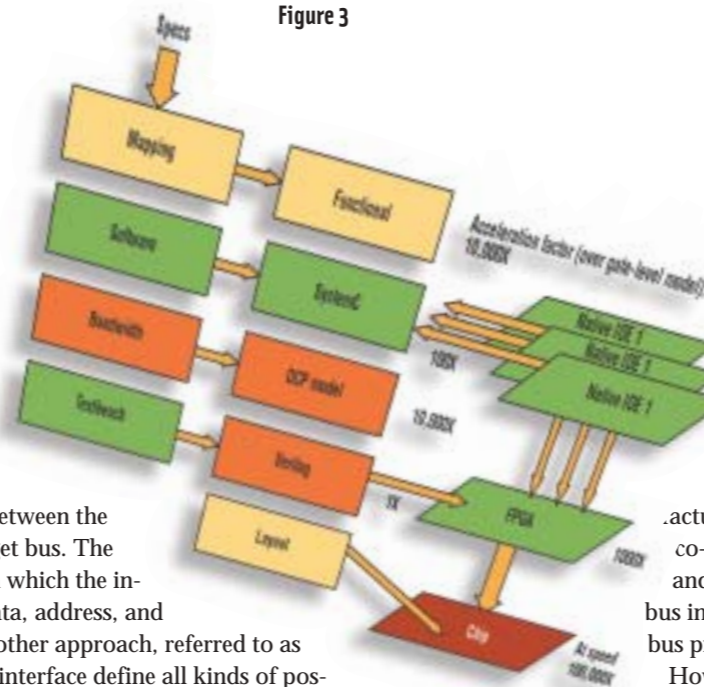
faces shown in the figure are intended to ease verification. These include testbench definitions, modeling support, and software sockets for vendor-supplied drivers, as well as an application programming interface for the designer(s) who must craft the application software.

Two basic approaches are taken when de-

IP Views

When a block of IP is obtained from a vendor, it should come with a collection of different views (different levels of abstraction) so that various

Figure 3

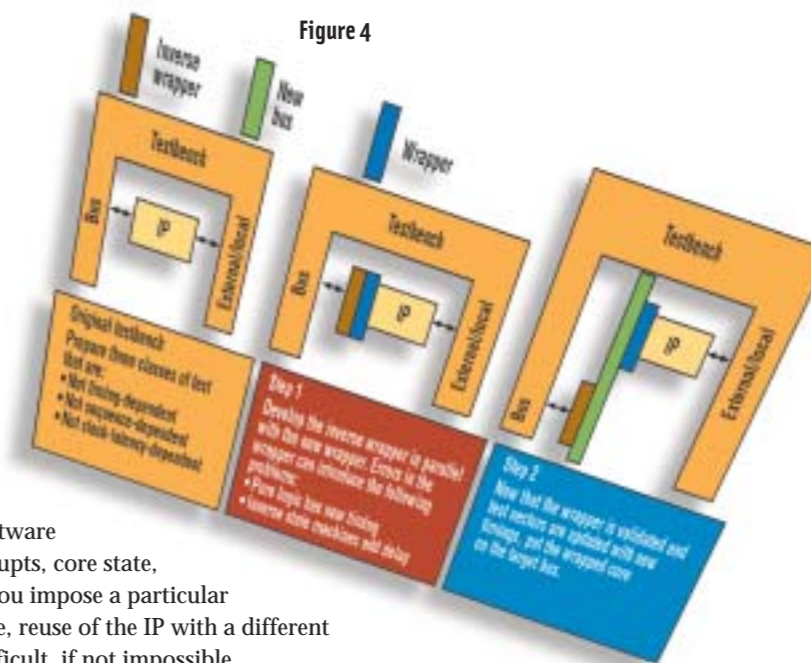


fining the interface between the IP block and the target bus. The first is bus-centric, in which the interfaces define the data, address, and control signals. The other approach, referred to as core-centric, has the interface define all kinds of possible signals coming from a core and how those signals must be propagated through the bus, including manu-

Testbench Tips

After an IP block is encapsulated, the testbench that came with the block is rendered useless because the block's native interface is no longer available. However, by employing an inverse wrapper, the native interface can be added back. Thus, the testbench can be used and there's no need to create new test sequences.

Figure 4



facturing test signals, software co-debug signals, interrupts, core state, and activity signals. If you impose a particular bus interface on the IP core, reuse of the IP with a different bus protocol would be difficult, if not impossible.

However, in a well-specified core-centric protocol like the OCP, the IP core must only be designed to comply with the interface boundary defined by OCP. The OCP-compliant bus will fully address communications between the IP cores using the OCP interface. The OCP is the result of efforts by members of the Open Core Protocol International Partnership (OCP-IP, www.ocpip.org). This organization is dedicated to developing a common standard for IP core interfaces, or sockets, that facilitate the "plug-and-play" aspects of SoC design. This effort complements the work done by the VSIA in helping to set various standards for most aspects of SoC design.

IP Blocks Can Wear Many Faces

When you purchase or license a block of IP, it often can be delivered as either a soft or hard core. Further, the vendor should be able to deliver the block in several views so that multiple hardware and software co-verification and debugging approaches can be applied to ensure functionality and performance. For example, during system architectural design, functional block diagrams are the best representation, while for data-transport and throughput studies, C-based models using SystemC are, perhaps, a more optimum representation. Additionally, when it's necessary to go into the details of high-performance bus structures, the OCP model is needed. During realization and detailed debugging of the system RTL, Verilog simulations would be helpful. To check functionality and co-debug your software and hardware in tandem, an FPGA emulation would be the optimum approach (Fig. 3).

A soft core provides more flexibility because it's just a software description of the logic function. The software description then can be fed into synthesis tools that are tied into the appropriate logic library tuned to a specific manufacturing process. The soft-core description can be delivered in a high-level description language or as a register-transfer-logic (RTL) file, either of which can be integrated into the rest of the circuit description. The resulting implementation delivers the desired functionality, but a question may linger on whether or not it can deliver the desired performance.

Speed can be an issue with soft cores, especially if you're trying to push the block to deliver maximum performance. That's the

result of leaving the physical design up to the automated synthesis and physical layout tools. Often, the tools have settings that allow you to set several parameters, which can be optimized during synthesis or layout. But there's no guarantee that the final physical implementation will meet all performance criteria due to variations in the paths taken. Therefore, specific optimization tools can be used in addition to the synthesis software. Those tools can optimize the soft cores such that they can often achieve better performance than possible with many hand-hardened macros.

A predictive analysis tool for RTL descriptions, such as SpyGlass from Atrenta, will help detect IP problems early in the design process. Such a tool enforces downstream performance requirements early in the implementation, thus minimizing design iterations. It can help detect issues such as clock-domain crossings, synchronization, timing, testability, and other potential problems at the RTL level. Once the analysis and modification are completed, the design can be compiled, then run through a logic-synthesis program to create the logic equivalent of the soft core.

A handcrafted physical design (a hard core) provides a more predictable performance result because its operation and performance can be fully characterized, from logic elements all the way through physical layout. However, a hard core tends to be more difficult to reuse. It might take several man-months of effort to transfer the core to a different process node if the next application is expected to use a higher-performance fabrication process.



Go Backward To Test Forward

Today, most blocks of IP are accompanied by testbench information that provides designers with tests to check that the block is functioning properly. However, if the block will be encapsulated so that it ties into an on-chip bus, the testbench probably won't work because it needs the native interface of the IP block. Thus, you can create an inverse wrapper to recreate the native interface from the socket interface and use that recovered native interface for the testbench (Fig. 4). Using exact inverse wrappers and bridges is an absolute must to help identify errors and enable efficient validation by existing testbenches.

You can follow a simple step-by-step approach to first isolate the testbench timing sensitivities and coarse wrapping errors by creating an inverse wrapper. Then, fine-grain testing can be done on the new bus by reusing the IP vendor-supplied testbench. The first testbench will include three classes

of tests that are not timing-dependent, sequence-dependent, or clock-latency-dependent. By developing the inverse wrapper in parallel with the wrapper that provides the

socket interface, you can leverage the information in creating the wrapper to ensure that the inverse wrapper recreates the original interface. However, when developing the inverse wrapper, you might cause timing problems. Pure logic has new timing, and the inverse state machine adds some delay that must be taken into account.

As part of the design, strong consideration should be given to incorporating on-chip instrumentation as well as a basic boundary-scan Joint Test Action Group 1149 standard test port. Included as part of the instrumentation can be logic-analyzer-like monitoring and bus snooping capabilities to better capture an image of block activity as each steps through the application.

ASICs as big as your ideas and as small as 90nm.

Unique expertise from the smallest to the biggest

A lot of companies are making big promises, but only a few can truly deliver.

- Accepting 90nm high-performance designs today
- Offering high-volume, production-proven 130nm manufacturing
- Producing 10 million plus gates

Complete process integration makes the difference

Toshiba really delivers with a seamless process from beginning to end. And because we operate our own fabs, you don't have to worry about manufacturing, test, or packaging. Toshiba delivers:

- High-performance ARM and MIPS cores and unique capabilities in eDRAM, MPEG-4, 802.11, and Bluetooth.
- Support for design planning, IP integration and applications engineering
- Proven design flow with predictable timing closure and latest EDA tools

- Flexible engagements from specification to GDS handoff
- Expertise through U.S.-based design centers

Get more than the basics

Toshiba has shipped thousands of successful ASICs and can help you turn your bright ideas into reality. Get everything you want in an SOC/ASIC, including worldwide capacity now. Contact us at bright.toshiba.com



TOSHIBA