

OCP-IP News

Membership Announcements

OCP-IP is proud to announce the following new members:

Beneware – develops functional verification software, Based in Helsinki Finland

Global Unichip – full service SoC Design Foundry based in Taiwan, providing total solutions from silicon-proven IPs to complex time-to-market SoC turnkey services

IC Design, Research & Education Centre – is a leading IC center in Viet Nam for training, research and transferring technology in the IC industry

Innotech Corporation – provide the latest technology in wireless and voice control technology. Introduce American and European companies to advanced technology available through Japanese semiconductor manufacturers to help create new business

ITT Electronic Systems - provide Electronic Countermeasures, Interference Mitigation technology, secure voice, data link and command and control products

Pragam Technologies, Pvt. Ltd., - provide embedded development services for safety critical systems

Productivity Design Tools – simplify the process of collaboratively converting requirements and specifications into logic design, verification, embedded software, validation and documentation deliverables

Swan on Chips - consultant on Soc Design and Methodology

EDAC Investigates Software Piracy

A key role of the EDA Consortium is to identify and address issues that are common to its members and the design community that they serve. EDAC operating committees, chaired by representatives from member companies, spearhead the effort to address those issues for the benefit of both EDA vendors and customers.

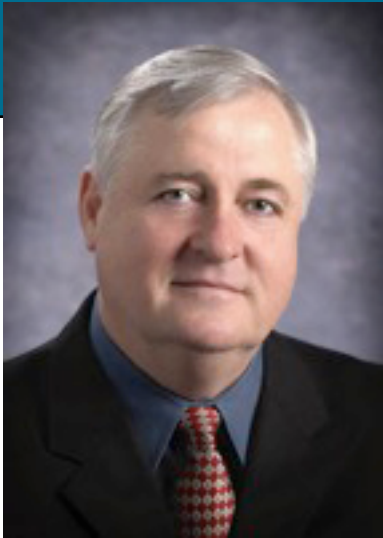
The EDAC Anti-Piracy Committee, chaired by Scott Baeder of Cadence

Design Systems, is currently working with three suppliers of anti-piracy tools and services to assess the nature and impact of



EDA and IP software piracy. Using a carefully chosen group of EDA tools covering a range of applications from IC physical layout and analysis to FPGA and PCB tools, EDAC is working with their suppliers to assess the types of tools that are being pirated, where the piracy is occurring, and the business impact of the piracy. While it is up to individual EDA tool vendors to decide what steps to take, assessing the business impact is a common issue. EDA vendors normally compete on functionality, performance, price, value, etc. However, if a company is using pirated software, they likely won't buy tools from any of the EDA vendors. Thus, pirating one company's tools hurts the industry in general.

For more information about these efforts please see our recent press release at: http://www.edac.org/downloads/pressreleases2009/antiPiracyRelease20090429_final.pdf or visit our website at www.edac.org



*Ian Mackintosh
OCP-IP President and
Chairman*

President's Overview

The first six months of 2009 have been a very active and extremely exciting time for OCP-IP. [OCP 3.0](#) has nearly completed member review and the specification now includes the additions of [cache coherence extensions](#), [power management signaling](#), a write response extension as well as a [new consensus profile](#).

Our System Level Design Working Group's project to evolve and enhance the entire Transaction Level Modeling kit is complete and now supports modeling proposed in [OSCI 2.0](#). The new kit is [available](#) to members through the OCP-IP website and has already been downloaded hungrily in just the first few months of availability. The kit represents the first, and most advanced TLM-2.0 based, industry-ready kit in existence today. The enthusiasm with which members have embraced our kits has established them as the first port of call for the entire ESL world. Proactive work has already begun on the next release to add features supporting OCP 3.0. Please see the centerpiece article on page six for more technical details about these kits.

Our Meta Data Working Group has completed its proposal for extensions to [IP-xact](#) that will enable us to better capture the richly detailed OCP interface in this meta data format. This information will allow our members to leverage both IP-xact and OCP ensuring lower development costs and quicker time to market. A detailed technical article

describing the OCP-IP extensions can be found [here](#).

In other [SPIRIT](#) news, SPIRIT and [Accellera](#) have announced the two standards organizations will merge. The merger should save costs significantly for members already common to both organizations. Merger details are expected to be complete by the end of 2009 and be fully executed by May 2010. For more information on the merger please see the [Acclera press release](#) and [FAQ's](#).

OCP-IP also recently [announced](#) the availability of a new [NoC white paper](#) discussing an approach to Performance Analysis of Network-on-Chip (NoC) Architectures for Video Systems on Chip (SoCs). The paper describes a typical video SoC system, and the traffic profiles for each of the processing engines providing performance analysis measures of interest. On page four you will find a detailed technical introduction to this new white paper.

The Marketing working group continues to publish press releases and in-depth technical articles detailing our members use of OCP. If your company would like help in publishing such an article, please send an email to admin@ocpip.org indicating your interest and we would be more than happy to assist.

Best Regards,
Ian R. Mackintosh
President and Chairman OCP-IP

Working Group Reports and Updates

Debug Working Group

The DWG continues work specifying a supporting analysis infrastructure, including API, tools, and reference designs for multicore and ESL integration. The group continues to enjoy its status as a useful networking forum. Work on the next revision of the debug specification to align with the new features of OCP 3.0 can begin in the second half of 2009.

Functional Verification Working Group

A new Chair for the Functional Verification Working Group was announced in April. The group has a very strong team and will become extremely active during the second half of 2009 as OCP 3.0 completes member review and requires their contributions.

Marketing Working Group

The Marketing Working Group remains very active helping member companies compose and place their OCP-related articles awhile publishing the OCP-IP newsletter and press releases. If your company would like assistance placing an article, prominently targeted to the industry or directly to our very focused community, please contact us at admin@ocpip.org.

Metadata Working Group

The Metadata Working Group (MDWG), continues work on enhancements to fully capture OCP interfaces using the IP-XACT format defined by the [SPIRIT Consortium](#). The MDWG is proposing extensions that can best describe OCP within the existing IP-XACT XML. This groupfa coordinates with SPIRIT Consortium and its member companies with the goal to incorporate this work into future releases of IP-XACT. The MDWG includes representatives from Magillem, Nokia, Sonics, STMicroelectronics, Synopsys, Texas Instruments and Toshiba. If you would like to participate in the work of the MDWG, please let us know by sending an email to admin@ocpip.org

Network-on-Chip Benchmarking Working Group

The Network-on-Chip Benchmarking Working Group (NoC BWG) has published a two-part NoC Benchmarking Specification, which is available on the OCP-IP web site. The NoC BWG continues to assess methods for creating benchmarks. If your company or university is interested in donating benchmarks to the group or working with the OCP-IP NoC BWG, email admin@ocpip.org.

Specification Working Group

The Group released OCP v.3.0, to Member Review in May. Member review will close in July so that the Spec WG can collectively review all comments received. OCP 3.0 includes the additions of cache coherence extensions, power management signaling, a write response extension and a third consensus profile. OCP 3.0 will become the specification of record in 2H09.

System Level Design Working Group

The working group's project to evolve and enhance the entire Transaction Level Modeling kit and also support modeling now proposed by OSCI 2.0 is now complete and is available from our website. The next phase for the group will be to further advance the TLM Kits to include support for the new features newly proposed in OCP 3.0.

A Methodology for Performance Analysis of Network-on-Chip Architectures for Video SoC

By Krishnan Srinivasan, Sonics

Introduction

System-on-Chip (SoC) architectures integrate several processing units on a single die. With every passing generation, the need for enhanced processing power vis-à-vis higher system complexity has resulted in an increase in the number of processing engines that are integrated on the chip. While designs with tens of processing engines are already in production, next generation SoCs will easily integrate hundreds of processing cores. SoC architectures can be broadly classified into symmetric multi-processor architectures (SMP) such as those used in the MIT RAW project [1], and asymmetric application specific architectures that are widely used in targeted application domains such as high definition video decoders. The complexity of on-chip communication between the processing engines of a SoC has led to several innovations in the communication architecture design and synthesis [2][3][4].

The design of the interconnection network for the application specific domain presents a unique challenge. The processing engines in these architectures have varied physical attributes and performance requirements. Moreover, care must be taken to balance the system performance and the associated costs. Each processing engine has its own unique set of traffic characteristics, such as burst length of transactions, spatial and temporal distribution of addresses, etc. Under these conditions, design of an optimal interconnection network can be extremely difficult.

The complexity of SoC systems encourages intellectual property (IP) based design methodology. The SoC designer gets pre-verified IPs from different vendors, and integrates them onto the SoC architecture. For example, a high definition television may consist of a CPU from a CPU vendor, a few in-built cores, an interconnect from a NoC vendor, and an external DRAM memory from a memory vendor. The advantage of such an approach is that it “distributes” the effort among the

different vendors, and the SoC designer can focus on his core competency. Overall, such an approach can minimize time-to-market, and thus increase productivity. However, from a vendor perspective, it presents an important problem. The IP vendor does not know the chip into which its IP is going to be integrated! So, how does one design the IP? What are the unique characteristics of the SoC to which the IP should be optimized? This problem is critical for the interconnect vendor, as the interconnect has a global view of the system.

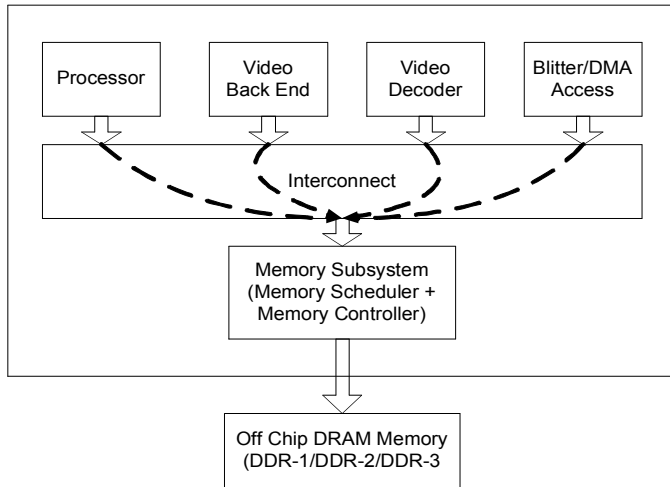
Among the application specific SoC architectures, the video decoder is among the most challenging. Depending on the quality of the video output, the bandwidth requirement of the application can span from 2GBytes/sec all the way to 10GBytes/sec [5], with varied traffic profiles at each processing engine. The challenging communication requirement of the video decoder makes it an interesting and in some ways ideal candidate in performance analysis of the interconnection network. In a new [white paper](#) from OCP-IP, we study the traffic generation problem of a video decoder SoC. We have provided essential characteristics of a video decoder SoC that can be used to generate traffic patterns. These traffic patterns can then be used to design or tune the interconnection network. This paper is useful to interconnect architects, who can use the traffic patterns for performance analysis of new features. It is also useful to application architects who can use these patterns to demonstrate the superiority of one design over other.

Analysis Framework

The primary focus of our paper is to provide an analysis framework for interconnection networks which consists of a methodology to describe the performance requirements of the video SoC system, and guidelines for reporting results that are interesting from the video SoC’s viewpoint.

Continued next page

Figure 1: A video SoC architecture



System Performance Requirements

Figure 1 describes a typical video SoC system. It consists of several types of processing engines, of which four are of primary importance. They are, i) the video decoder, ii) the graphic accelerator/blitter/DMA, iii) the video back end engine, and iv) the CPU. Most of the traffic in the interconnection network consists of communication between these processing engines, which proceeds through a shared DRAM memory.

We describe the system shown in Figure 1, in two levels of hierarchy. At the top level of the hierarchy, the overall system architecture is described by specifying the numbers of each processing engine type, the total bandwidth (in Gbytes/sec) required at the DRAM memories, the number of individual DRAM memory channels, the configuration of each DRAM memory channel, and other information such as data width at the processing engines. The second level of hierarchy takes the first level parameters as input, and generates processing engine specific traffic profiles, such as bandwidth requirement, average burst-length of transactions, spatial and temporal distribution of addresses, and the distribution of read-versus-write transactions.

Result Reporting

Under the results section, the paper presents details of what constitutes a good interconnection

network. Previous work limited performance evaluation just to the measurement of bandwidth and latency. However, in video SoC systems, the performance requirements differ with the processing engine type. In the case of CPU, it is important to minimize the average and worst case latency. On the other hand, the video decoder and back-end engines aim to minimize the jitter in the traffic. Finally, the paper also describes a method to report the area-cost of the design so that a pareto point representing the best trade-off between performance and area can be obtained.

Conclusion

Current day NoC performance analysis schemes treat the interconnect in isolation, and abstract away the rest of the system as black boxes that generate traffic with a certain bandwidth and latency. However, with the increase in the use of IP-based designs, it is becoming increasingly important for interconnect vendors to design the NoC with the end application in mind. Without system knowledge, the entire interconnect may have to be re-architected late in the SoC design cycle, which may not be acceptable. In our paper, we target one of the most challenging SoC applications, the video decoder, and propose realistic traffic generation and performance analysis schemes. Our framework can help identify potential communication bottlenecks early in the design cycle, by providing realistic traffic scenarios during the design of the NoC. It can also be used to generate a set of video benchmarks that can be employed as a standard for evaluating different NoC architectures.

[1] The MIT Raw Project, <http://groups.csil.mit.edu/cag/raw/documents>

[2] Srinivasan et al, "Linear Programming based Techniques for Synthesis of Network-on-Chip Architectures", IEEE Transactions on VLSI, April 2006

[3] Marculescu et al, "Outstanding Research Problems in NoC Design: System, MicroArchitecture, and Circuits Perspectives", IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems (IEEE TCAD), Vol 28, No 1, January 2009

[5] Sonics Inc, www.sonicsinc.com

[4] Drew Wingard, "Achieving High Memory Bandwidth for Cortex Family based Video SoCs Using Multiple DRAM Channels". ARM Developer's Conference. 2008

Modelling OCP Interfaces in SystemC: Standards built on top of OSCI's TLM-2

James Aldis, (Texas Instruments France) Mark Burton, (Greensocs Ltd.,)
Robert Günzel, (Greensocs Ltd and TU Braunschweig,) Herve Alexanian, (Sonics Inc)

OSCI has recently published their TLM-2.0 technology which provides for interoperability between SystemC models of electronic components whose primary interfaces are memory-mapped busses. This is a very important development in the ESL industry, but it does have some limitations. In particular, TLM-2.0 needs to be extended for some real memory-mapped busses, and TLM-2.0 is only applicable at high levels of timing abstraction: untimed, loosely-timed and very approximately-timed models. These limitations are inevitable given that OSCI can only create technology that is generic in nature and not associated with any specific real component.

OCP-IP, as the publisher of a memory-mapped IP core interface, is able to address these limitations and has done so. This paper describes the approach adopted by OCP-IP to providing SystemC modelling interfaces for a real memory-mapped bus family. TLM-2.0 has shown itself to be an effective and efficient base technology for all variants of OCP at all levels of abstraction, from cycle-accurate to untimed. Furthermore, the nature of TLM-2.0 and its use as a base technology for everything means that bridges between different protocols or between different levels of abstraction are very easy and efficient. We even see that in a very large number of cases an OCP component is directly interoperable with the OSCI Base Protocol.

The Open Core Protocol (OCP) is a public standard memory-mapped core interface, defined by collaboration between systems houses, EDA vendors and IP suppliers within the OCP International Partnership (www.ocpip.org). The motivations for defining the OCP are to enable better IP reuse and a wide spectrum of common tooling and shared experience.

Transaction-level modelling (TLM) is an essential part of all modern system-on-chip development methodologies and where IP is being reused and purchased from third-party suppliers, standard interfaces for TLM models of IPs are as important as standards for the RTL interfaces. Because of this, the OCP-IP provides standard SystemC APIs for modelling OCP interfaces at multiple levels of abstraction. These APIs, along with infrastructure code in the form of SystemC channels, ports, monitors and so on, have been available to the

OCP community for more than 5 years. They are mature and widely-used. OCP-IP has shipped many hundreds of thousands of copies each representing many hundreds of thousands of dollars in collective development expertise, and delivering a recognized world-class technical solution for users.

This code uses SystemC, the dominant TLM modelling language, as defined by the Open SystemC Initiative (OSCI). In the summer of 2008, following a lengthy process, OSCI released version 2 of its TLM modelling library. OCP-IP members contributed to TLM-2 and have worked hard to ensure its applicability to OCP.

Importance of the OSCI TLM-2.0 Standard

OSCI's release of TLM-2.0 marks a seminal point in the life of electronic system-level design. For the first time a non-proprietary modelling technology has been created which offers

- High performance simulation, using shared memory (pointer) communications rather than data copying, and allowing for bypass of network and bridge models
- Fully-defined data payloads, meaning complete interoperability interfaces
- Detailed API rules and conditions, ensuring the ability to create API checkers and verify standards compliance
- Extensibility in both functionality and timing-accuracy

Within TLM-2 OSCI has provided various bits of technology of general utility and also an interoperability standard for basic memory mapped busses. This *Base Protocol* can be used to model the bus interfaces of many IP components, but it is not a superset of all bus interfaces for all components. Therefore, it needs extension in many cases. OSCI provides an extension mechanism but does not provide any "standard extensions".

Until the release of TLM-2.0 there had been a plethora of different mechanisms used to model bus interconnect. TLM-2.0 will provide commonality of approach for all of these, and

Modelling OCP Interfaces in SystemC

enable levels of direct interoperability that were not achievable before.

The standard provides for three levels of commonality:

1. **Commonality of approach:**
Interoperability between TLM models is not just about technology, but also about methodology. Understanding whether and how different features of a protocol should be modelled, what sort of structures should be used, who should be expected to provide those structures, and what services should be provided.
In this case, simply using the same methodology and approach to modelling greatly assists users in understanding each other and enables the market for providers of both training and IP.
2. **Commonality of basic technology:**
OSCI TLM-2.0 provides a wealth of basic technology, including the Base Protocol for memory mapped busses, but also a host of utility functions and structures which are of use to model interface development across-the-board. Reusing this basic technology not only reduces the burden for organisations like OCP-IP (and others who are providing specific interfaces built on TLM-2.0,) but also greatly assists in ensuring a basic degree of interoperability. Furthermore, OSCI specifies exactly how the basic technology is to be extended, and in doing so, how compatibility can be maintained. Even in cases where direct plug-in interoperability is not possible, use of a common approach and common technology will frequently ensure that simple and efficient bridges can be constructed.
3. **Use of the *Base Protocol*:**
For IP components whose interfaces are relatively unsophisticated, the OSCI *Base Protocol* (BP) may provide a complete off-the-shelf solution on how to model them, at an approximately-timed or loosely-timed/untimed level of abstraction. Every BP bus master (initiator) is 100% compatible with every BP bus slave (target).
The BP does not include common bus features such as out-of-order processing of transactions, bus locking or semaphores, or sophisticated addressing modes, but even without these features a huge number of components will find it a superset of their bus functionality. For example, all components following the OCP-IP consensus profiles should be able to use the BP unmodified, even though the BP is not in any way OCP-specific. At this level of abstraction the fact that a

bus interface is OCP *in the hardware* becomes irrelevant and TLM models enjoy a much higher degree of interoperability than the hardware they represent. OCP components might plug directly to CoreConnect™ components, for example.

Clearly, memory-mapped busses are not the only interfaces on TLM models that need to be standardised in this way before the problem of TLM model interoperability goes away completely.. However, OSCI's TLM-2.0, by focussing on memory-mapped busses, has managed to solve one part of the problem at the same time as providing a technology framework that can be used for other parts, such as clocks, interrupts, network interfaces, and so on. And, because OCP is just a memory-mapped interface protocol, TLM-2.0 already contains all that is needed for OCP-IP to specify modelling APIs for OCP users.

Levels of Abstraction

OCP-IP recommends 5 levels of abstraction for memory-mapped busses, as described in Table 1.

Layered TLM Technology for Multiple Levels of Abstraction

In order to make it possible to bridge simply and efficiently between level of abstraction, OCP-IP is developing a layered technology for SystemC modelling. For the 5 levels of abstraction listed above, 3 SystemC APIs are provided by OCP-IP: (Table 2)

The three SystemC APIs are mutually incompatible, but adapters between them are provided. The goal is to allow fully automatic instantiation of adapters within the components, so that a system model can be assembled from whatever component models are available, while the level of abstraction for each memory-mapped bus connection can be selected independently.

The SystemC APIs are quite different from an RTL model of an OCP interface. Whereas in RTL an interface consists exclusively of signals on wires, the SystemC interfaces are all based on the OSCI TLM-2.0 concepts of...

1. a single payload object carrying the necessary information about the transaction, to which references are exchanged by function call between components.
2. data being exchanged directly between the transaction target (the endpoint memory or register bank) and a data buffer internal to the transaction initiator (the CPU or DMA that

Modelling OCP Interfaces in SystemC

Table 1

OCP Term	Timing Accuracy	Abstractions	OSCI-TLM-2.0 equivalent
TL0	Cycle accurate	None, this is the RTL level	SystemC synthesise-able subset
TL1	Can be fully cycle-accurate, requiring clock synchronisation between bus master and bus slave, and respecting the OCP protocol. All beats of a burst are modelled.	Wires and signals are not modelled	none so far
TL2	User selectable number of timing points per bus burst	No clock synchronisation therefore some non-determinism. Optional averaging of bus occupancy over bursts or parts of bursts. Flow control not modelled explicitly.	none so far
TL3	4 timing points per bus burst, bus occupancy determined only by 'data receiver'	No modelling of independent write data phases, no ability to model intra-burst timing effects, no distinction between address order within a burst (eg wrapping and incrementing bursts are equivalent)	Approx-timed (AT) nb_transport()
TL4	Minimum necessary to run software on a virtual platform	"Pure functional" representation of memory-mapped bus. No flow control or ordering effects are modelled.	Loosely-timed (LT) b_transport() and nb_transport()

Table 2

OCP Term	OCP-IP SystemC Interface	OSCI TLM compatibility
TL0	Not specified by OCP-IP separately for SystemC from other HDLs	None, this is the RTL level
TL1	OCP-IP TL1	Uses TLM-2 generic payload, sometimes with extensions. Uses different protocol phases and rules from OSCI TLM-2.0 BP. Uses nb_transport()
TL2	OCP-IP TL2	Uses TLM-2 generic payload, sometimes with extensions. Extensions are a subset of the extensions used at OCP-IP-TL1. Uses different protocol phases and rules from OSCI TLM-2.0 BP and from OCP-IP-TL1. Uses nb_transport()
TL3	OCP-IP TL3	Uses TLM-2 generic payload, sometimes with extensions. Extensions are a subset of the extensions used at OCP-IP-TL2. Uses the same protocol phases and rules as OSCI-TLM-2.0 BP. Extensions may be ignorable in which case OCP-IP-TL3 is directly interoperable with OSCI-TLM-2.0-BP.
TL4		Uses nb_transport() and b_transport()

Modelling OCP Interfaces in SystemC

launched the transaction) without being touched in any way by intermediate components such as bus routers, arbiters, protocol conversion bridges, and so on.

Because of this, bridging between the SystemC APIs and a TL0 (RTL) model of a component is less efficient than bridging between two SystemC APIs. This is not considered an “issue” because models with TL0 bus interfaces are expected to run slowly, anyway. Adapters will of course be available.

The layering is illustrated in Figure 1:

In general all technologies used at one level of abstraction are also used at all lower levels. The only exception to this is that the mechanisms for modelling bandwidths and splitting of transactions into chunks are specific to TL2. They are not needed at TL1 where explicit modelling of all beats of bursts, with flow control, is compulsory.

OCP-IP Extensions

The OCP protocol requires extensions to the OSCI generic protocol in the following areas:

- Bus locking and semaphores (all levels of abstraction).
 - Exotic addressing modes (all levels of abstraction).
- The OSCI TLM-2 GP supports only transactions addressing a contiguous set of addresses in the memory map, possibly with a repeat for accessing FIFOs. A sophisticated protocol will allow forms of transaction outside this scope.
- Transaction re-ordering enabling (required for TL3, TL2 and TL1).

- Posted write support (required for TL2 and TL1).
The OSCI TLM-2 GP only supports one form of “write” command, which to be useful, must be assumed to be a “non-posted-write”, meaning that it always has a response and this response indicates effective completion of the write process at the end target.
- Non-blocking flow control (required for TL2 and TL1)
Most conventional memory-mapped busses have mechanisms for flow control that can be described as “blocking”, in that there is a risk that the wires of the bus will remain occupied by one transaction for a long period, thus preventing other transactions from progressing. OSCI-TLM-2 BP only considers this sort of flow control.
- Address ordering within a transaction (required for TL2 and TL1).
Certain addressing modes do not change the functional interpretation of a entire transaction but do have an effect on performance, if effects internal to the bus burst are considered. For example a “wrapping” burst contains the same sequence of addresses as an “incrementing” burst, but starts with the most critical rather than the lowest address. Such effects cannot be modelled at TL3 or TL4 and so it is unnecessary at those levels of abstraction to distinguish wrapping from incrementing addressing modes.
- User extensions (required for TL1).
Since OCP is extensible in itself, it is necessary to support user extensions to OCP in the

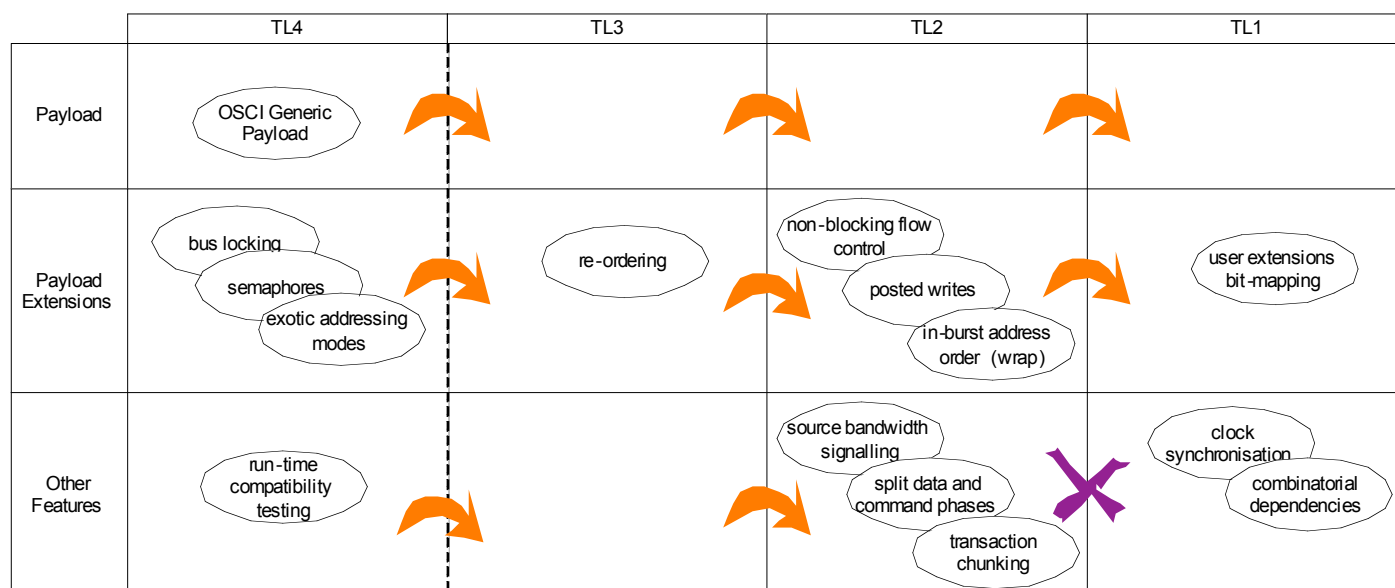


Figure 1

Modelling OCP Interfaces in SystemC

SystemC APIs. This is trivial at TL3 and TL2, but at TL1 some consideration needs to be given to how these are mapped to binary wires in hardware, so that it is possible to model components that modify such extensions without being aware of their functional purpose. An example would be a network-on-chip, configured to map initiator extensions onto target extensions, whose SystemC model is capable of doing this mapping only as a modification of anonymous bits.

Every OCP interface has a configuration, which specifies which memory-mapped bus features it includes. The intersection between the bus Master and bus Slave configurations will determine which of the above extensions is present, optional, or absent from each of the SystemC APIs. As the majority of bus interfaces encountered in real life tend to be simple, extensions should be regarded as uncommon. The OSCI TLM-2 generic protocol unextended is able to cover a wide range of SOC components' bus interfaces.

Configuration and Run-Time Compatibility Testing (Figure 2)

OCP is not one memory-mapped bus protocol; it is a family of protocols spanning from very simple to highly complex. A component with an OCP interface on it publishes values for a set of around 90 parameters, the combination of which is commonly called the component's *OCP configuration*. The

parameters describe which commands and addressing modes are used, address and data widths, and so on. In hardware, a component's OCP configuration is determined by its internal synchronous digital circuitry and the "configuration file" is a documentation of the hardware. The configuration file is used when an SOC (or testbench) is integrated, to confirm the compatibility between an OCP Master and the

OCP Slave it is connected. This process is in general, automated. In some cases the OCP configuration of one side of the interface may be used to create the RTL code for the other component. In RTL we refer to OCP "sockets," but in this case a socket is nothing more than a set of pins on the boundary of a component.

In SystemC TLM, on the other hand, the components' source code may be much more generic. A component may be able to adapt its behaviour to support an externally-supplied OCP configuration, or may even be able to adapt to the OCP configuration of the other OCP component to which it is bound (this possibility exists, but is not shown in the diagram). In SystemC we also refer to OCP "sockets," but in SystemC the socket has more of a real existence than in RTL. It is a SystemC object containing the SystemC port and SystemC export needed for binding, and it can provide a number of other services to the user-code for the component, such as removing timing annotations, providing memory management for transaction payload objects, instantiating adaptors between abstraction levels and testing configuration during simulation elaboration.

In the OSCI TLM-2.0 technology, the sockets are C++ template classes, for which an *interface traits class* needs to be specified. This traits class is also known as a *protocol*, hence the OSCI Base Protocol is a such a traits class, used to determine the type of a TLM-2.0 socket. Only a Base Protocol socket can bind to another Base Protocol socket.

The principle behind this is that when extending TLM-2.0 to model some functionality beyond the scope of the Base Protocol, a new traits class should be declared, but the same underlying C++ technology used. The extensions should use the extension mechanism of the OSCI Generic Protocol. This ensures that sockets using incompatible protocols can not accidentally be bound to one another, but that creating a bridge between them is simple and efficient.

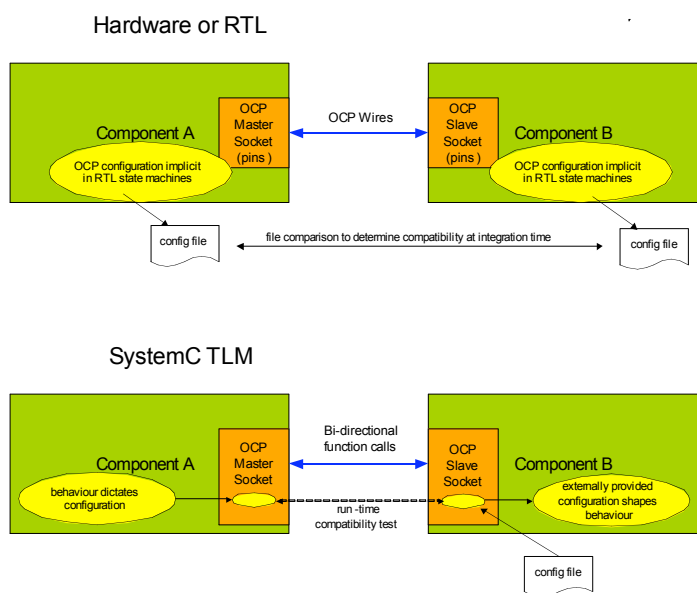


Figure 2

This mechanism cannot meet all the requirements for modelling OCP, and therefore it has been complemented by an elaboration-time compatibility check between the bound sockets. The reasons for this are:

- The almost unlimited number of different OCP configurations makes it impractical to create a traits class for every one
- OCP compatibility rules are more sophisticated than the mechanism permits. There are many cases where a pair of components with non-identical configurations are compatible, and the sets of compatible configurations are not disjoint.
- It is highly desirable that where possible, direct binding to the OSCI TLM-2.0 Base Protocol be possible, which dictates the use of the Base Protocol traits class in some cases
- It is desirable to be able to create SystemC sockets that are able to adapt to the level of abstraction of the thing they are bound to, if necessary.

Summary

The OCP family of memory-mapped bus protocols can be effectively modelled using the OSCI-TLM-2.0 technology recently published. OCP-IP is proposing a set of extensions and extra technologies to provide interoperability standards for the OCP at all levels of TLM abstraction.

The following technologies from OSCI TLM-2.0 have been used unchanged:

- generic payload
- base protocol phases
- transport functions with timing annotation

The following technologies have been added to OSCI TLM-2.0:

- generic payload extensions
- transaction 'chunking' to increase timing accuracy
- sender bandwidth information
- phases for independent data and command, and for chunked transactions
- clock synchronisation for cycle accuracy
- support for combinatorial dependencies, for cycle accuracy
- run-time bindability testing

And, no part of the OSCI TLM-2.0-BP has been discarded, except that compile-time bindability testing is more limited than anticipated by OSCI.



James Aldis is a senior member of the Group Technical Staff at Texas Instruments, where he works on the architecture of OMAP SoCs. His degree is in pure mathematics from the University of Liverpool and his PhD is from the University of York.



Dr. Mark Burton is the founder of GreenSocs. He graduated from Warwick University with a degree in computer systems engineering and completed a PhD in artificial intelligence within education. He is the chair of the OCP-IP SLD working group.



Herve Alexanian is a staff software engineer at Sonics, Inc. His main focus is on reference models for SoC interconnects. He has been the Sonics representative to the OCP-IP System Level Design workgroup since 2004.



Robert Guenzel received his diploma in Computer and Communications Engineering from the technical university of Braunschweig in 2005. Since 2005 he has been a researcher at the department of integrated circuit design (E.I.S.) at the same university.

Network-on-Chip Benchmarking Workgroup and Tampere University of Technology announce an open source simulation tool called Transaction Generator

Erno Salminen, TUT, Chairman of NoC Benchmarking Workgroup.



OCP-IP's Network-on-Chip (NoC) Benchmarking workgroup develops standardized methods and test set for evaluating on-chip networks. The NoC paradigm brings the techniques developed for macro-scale, multi-hop

networks into a chip. A multitude of NoCs has been proposed in literature, but detailed comparison have been difficult due to lack of common test sets.

A specific simulation tool, called Transaction Generator (TG), has been developed at Tampere University of Technology (TUT), Finland. The tool is written in SystemC and it enables the modeling of complex

multiprocessor system-on-chip (MPSoC) with relative ease and is therefore also suitable for evaluating NoCs. The benchmark-set defines application workload, available computation resources, and mapping between these elements. Descriptions are in XML format which is automatically parsed, and based on this SystemC models are generated. The evaluator provides a model of the NoC under study and connects it to terminals of the TG. Subsequently, SystemC codes can be compiled with any C++ compiler and executed simply on a workstation.

The experiences with Transaction Generator have affected the workgroup's benchmarking methodology. Now, TG will be modified to fully support our workgroup's modeling methodology and then made available under LGPL open source license. The actual public release is expected within the second half of 2009.

Recent Publications

Available in the OCP-IP Press Room at www.ocpip.org/pressroom.

Press Releases

June 03,, 2009

[OCP-IP Announces Availability of New Network on Chip Benchmarking White Paper](#)

May 06,, 2009

[OCP-IP Releases OCP 3.0 Specification](#)

April 21, 2009: [OCP-IP Delivers New Advanced SystemC TLM Kit](#)

Articles

May 01, 2009 - [Using OCP and Extensions to Support System-Level Cache Coherence](#)
Tech Online

Announcements: Now Available

New NOC Benchmarking White Paper

A new white paper presenting an approach to Performance Analysis of Network-on-Chip (NoC) Architectures for Video Systems on Chip (SoCs). The paper describes a typical video SoC system, and the traffic profiles for each of the processing engines providing performance analysis measures of interest. Using the performance analysis measurements provided, companies can easily and quickly determine the performance of the analyzed system.

New OCP Checker Now Part of CoreCreator II

The OCP checker a fourth-generation solution for validating protocol compliance of master and slave devices using OCP. It is based on SystemVerilog assertions (SVA) and can be used with all major logic simulators. It supports the complete set of protocol compliance checks defined in the OCP specification and spans the full range of OCP socket configuration options. The OCP checker can also generate trace files in the standard “.ocp” format for post-processing. The checker can be obtained, as part of CoreCreator II at www.ocpip.org/socket/corecreator/.

OCP 2.2 Specification Rev 1.1

The Specification Working Group officially released the OCP 2.2 Specification Rev 1.1 in May 2008. This version incorporates published errata and two consensus profiles from key OCP-IP members, and adds a trace file format (.ocp) description. OCP 2.2. Revision 1.1 is available on the OCP-IP Web site. For more detailed information, visit www.ocpip.org.

Debug Specification Version 1.0

The Debug Specification provides guidelines and recommended signal interfaces for on-chip debug of OCP-based systems and related multicore architectures. It provides a framework for IP and tools providers to develop comprehensive and re-usable debug and instrumentation environments that provide on-chip analysis and control features. These include trace, triggering, multicore synchronization, etc., along with recommendations for integration within ESL environments. The specification can be downloaded at www.ocpip.org/socket/ocpspec/.

NoC Benchmarking Specification, Part 2

Part 2 of the two-part NoC Benchmarking Specification presents a generic NoC architecture, a comprehensive set of synthetic workloads as micro-benchmarks, workload scenarios and evaluation criteria. These micro-benchmarks enable you to measure and pinpoint particular properties of NoC architectures, complementing application benchmarks. The specification can be downloaded at www.ocpip.org/socket/ocpspec/.

NoC Benchmarking Specification, Part 1

The specification presents a modeling methodology for applications running on multicore systems and it defines an XML format for documenting and distributing NoC benchmarks. It defines a black-box view of the processing elements that discloses only the relevant computational aspects for interacting with the on-chip data transport mechanism. The specification can be downloaded at www.ocpip.org/ocpspec/.