

On-chip Interconnection Design and SoC Integration with OCP

Chih-Wea Wang, Chi-Shao Lai, Chi-Feng Wu, Shih-Arn Hwang, and Ying-Hsi Lin
Realtek Semiconductor Corp.
Hsinchu, Taiwan

Abstract

With the increasing complexity of modern System-on-Chip (SoC) designs, more and more intellectual property (IP) blocks will be integrated into a chip. An open and flexible standard for IP core interface is quickly becoming necessary for efficient on-chip interconnection design and SoC integration. In this paper, we address the issues and share experiences on using Open Core Protocol (OCP) as the standard interface protocol, defining reusable profiles to fit different IPs, on-chip interconnection design, verification, and SoC integration with them.

1 Introduction

The continuous innovation of semiconductor technology enables more complex System-on-Chip (SoC) designs. Tens, even hundreds of intellectual properties (IPs) are integrated into an SoC to provide various functions, including communications, networking, multimedia, storage, etc. An increasing number of electronic devices such as mobile phones, digital media players, digital TVs, are designed and manufactured using SoCs. The usually short market life cycle of these electronic devices, however, does not allow a long schedule for chip designers to integrate a such complex SoC.

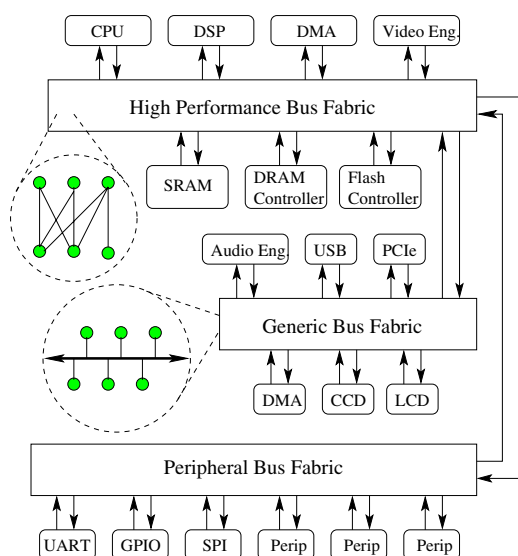


Figure 1: An SoC Architecture.

Figure 1 shows an example of on-chip interconnection design and SoC integration. When using IPs came from different providers with different interface standards, chip designers usually have to design adapters or bridges to connect them to a common in-house standard, or another popular standard, such as AHB [1].

Designing such adapters or bridges is not a difficult task, however, verification is an issue due to standard translation and compliance check. Subtle errors or incompleteness usually exist and lower their re-usability. Tedious reworking and verification are nearly inevitable when integrating a new SoC, which usually contains more IPs with more interface standards, and different system requirements.

Open Core Protocol [2] represents a collaborate work done by the OCP-IP members with the above issues in mind. The standard itself is evolved and public tools have been contributed by members to address these issues [3].

In this paper, we demonstrate the usage of OCP as an interface standard. The establishment of *profiles* is proposed for easy adoption and adaptation. Bus fabric design schemes are demonstrated to show the simplicity of interconnection IP design using these profiles. Experimental results show that the use of profiles simplifies the on-chip interconnection design and leads to low area cost.

2 Open Core Protocol

Open Core Protocol (OCP) is a protocol to adopt a non-profit, open standard based, and a complete socket that facilitates core reuse and SoC integration.

As a complete socket standard, OCP can be configured for high performance processors or DMA blocks with the ability of outstanding and out-of-order transactions. Moreover, it can also adapt to simple peripherals only with single transfers. OCP further provides all test/debug and sideband signals for various purposes such as interrupts or protections. Most protocol features and signals in OCP are optional, that allows users to choose the best socket configuration for their IP cores.

OCP further introduces another advantage that makes the cores independent of the physical interconnection or integrated subsystem. That allows the chip designers to choose an interconnection topology with better cost-effective factor. This advantage dramatically improved IP core re-usability, which leads directly to a more predictable and more productive platform-based SoC design methodology. With this abil-

ity, chip designers can move IP cores around in the SoC if required, which makes system analysis and architecture exploration much more feasible. It also increases the system configurability and flexibility that enables chip designers to re-allocate these building blocks for different market needs.

OCP has been adopted by the industry with good results [4]. There are more than 60 IPs with about 90 OCP interfaces at top-level. These OCP interfaces are different in protocol features or signals to optimize the needs of IP cores. However, all of them follow the same OCP timing and validation rules, which simplifies the cost in verification and implementation. Since OCP makes the IP core independent of interconnection technology and integration environment, this allows easier reuse of these OCP-compliant cores across multiple SoC products and targeted market.

The large number of existing OCP-compliant building blocks on a very flexible architecture leads to products that quickly address different market needs and better time to market.

3 OCP Profile

Since OCP is a complete and flexible socket standard, it has the ability to configure an interface to match a core’s communication requirements. Most extensions and signals are optional depending on user requirement. This approach is highly flexible, however, it is also quite challenging for users to determine their own OCP configurations.

As well as SoC integration, flexible interfaces also increase the complexity in bridge or interconnection development. With the high flexibility, chip designers must consider all possible OCP extensions and signals for various OCP-compliant IP cores, which also leads to higher hardware cost and lower performance.

To solve these issues, OCP provides predefined profiles to help map the requirements of common and standard communication to OCP configurations. The OCP profiles only focus on the OCP interface, with each profile consisting of OCP specific protocol features, related interface signals, and application guidelines.

In Realtek, we also faced the same challenges when we define the OCP configuration for IP cores and develop the bridge/interconnection designs. A proper set of OCP profiles effectively help the adoption of OCP.

HP	Simple Extension	Data Handshake	Burst Extension	32/64-bit Data	Tag Extension
GP	Simple Extension	Data Handshake	Burst Extension	32/64-bit Data	
PP	Simple Extension	32-bit Data			

Figure 2: OCP profiles and features.

In the development of in-house OCP profiles, we have to consider following requirements:

- The profiles must match most communication requirements in our applications.
- Other legacy IP cores or outsourcing IPs with other bus protocols can be easily translated into these in-house profiles with minimal hardware cost and performance downgrade.
- Since these profiles will coexist in an SoC, they must have the ability to translate between each other with minimal cost.

Considering the above requirements, we first classify all IP cores used in our system into three types: high-end IP cores with the capability of out-of-order response, generic IP cores with block data transfer, and low-end peripheral devices only with single transfers.

Based on these reasons, we define three types of in-house OCP profiles: high-performance profile (HP), generic profile (GP), and peripheral profile (PP) as shown in Figure 2.

The PP only implements the simple read/write transfer without other OCP extensions. It can match the requirement of access to most simple peripheral devices or control registers.

The second level, GP, extends the PP with additional data handshake phase and burst extension for generic device with block data transfer. Combining the data handshake phase, GP can provide single-request-multiple-data burst transfer with efficient block transfer. It targets on most IP cores with the capability of burst transfer.

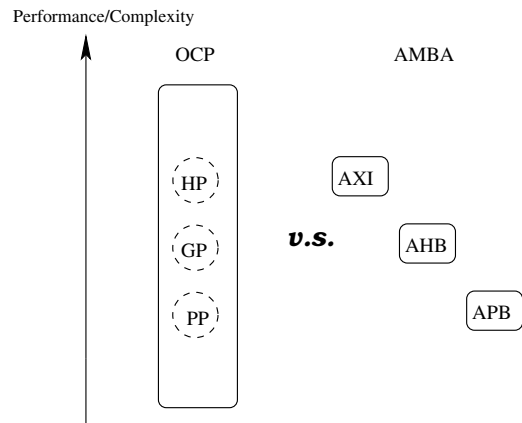


Figure 3: OCP and AMBA.

The highest-level, HP, further extends the GP with OCP tag extensions to support out-of-order response. For high-end processor, multi-channel DMA, and DRAM system, the capability of out-of-order response will increase the overall throughput.

Both of HP and GP utilize all 3 phases in transaction: request phase, data handshake phase, and response phase.

Therefore, HP and GP also can issue outstanding (pipelined) transactions and support 32-bit or 64-bit data width.

The three types of profiles correspond largely to the AXI, AHB, and APB in AMBA [1], as depicted in Figure 3. Note that, these three types of profiles can be easily translated between corresponding AMBA protocols. For example, an IP core with AHB interface can be adapted to GP with a simple bridge.

Nevertheless, all these profiles are OCP-compliant within the same OCP specification. The timing and verification rules are also consistent. On the contrary, these 3 AMBA protocols are different with respective validation rules that increase the cost in SoC integration and verification. Based on the mapping between our profiles and AMBA protocols, we only need to develop a few bridge designs: AXI to HP, AHB to GP, APB to PP, in order to integrate AMBA-based IP cores. This methodology greatly simplifies the integration especially in interconnection design.

4 Bus Fabric

By using OCP, IP cores can be independent of the interconnection topology and implementation. The bus fabric is also an IP core that provides communication function in an SoC. Moreover, the internal protocol inside the interconnection can be any other protocol to optimize for specific communication characteristic. Its architecture and topology are also application dependent.

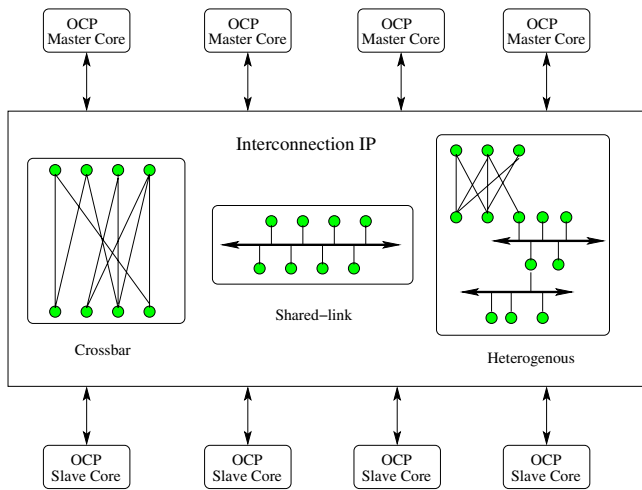


Figure 4: A hierarchical and heterogeneous interconnect system.

There are several interconnection approaches used in on-chip communication [5]. For instance, a shared-link approach puts all traffic shared the common interconnection resource. This topology results in the least hardware cost, but causes lower performance when there are collisions by multiple master devices.

On the other hand, a crossbar topology provides higher performance with parallel communication for different master to slave pairs simultaneously, but comes with higher hardware cost. The trade-off between hardware cost and performance will have to be made based one the application specific constraints.

A hierarchical and heterogeneous interconnect system provides flexible topologies and optimal trade-off between cost and performance. The present system constraints will determine the best architecture for a given application. Since the bus fabric IP is an independent IP for communication, chip designers can determine bus architecture as well as other functional IP cores into their system as shown in Figure 4.

Implementation of the interconnection IP is quite challenging due to the highly configurable OCP protocol. Outsourcing third-party interconnection IPs with versatile functionalities is one of available solutions as well as outsourcing other functional IPs. In-house solution that supports full features of OCP protocol may not be necessary. However, developing a bus fabric which only supports a subset of OCP protocol or profiles as mentioned previously is a more cost-effective solution.

As described in the previous section, an IP core with HP interface is classified as a high-end IP core with higher performance. A crossbar topology, in many cases, can match the requirement of such IP cores. Other IP cores with GP or PP interface are usually attached on shared-link type of fabric with lower performance requirement. Under this classification, we can reduce the development in bus fabric and bridge design.

5 Verification

OCP is a highly configurable core-specific interface, which can be customized and used in many IP cores. To efficiently verify relevant protocol features, it is an important task for users to perform verification on OCP-compliant cores. Typically it requires both compliance and functional tests [6].

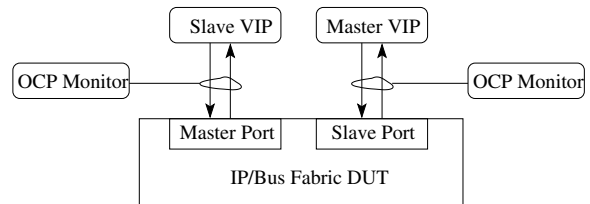


Figure 5: OCP Verification IP.

OCP-IP and many EDA vendors provide OCP verification IP which enables the verification of master and slave devices in OCP systems and cores, as shown in Figure 5.

Since OCP is a peer-to-peer protocol, IP core verification can be achieved independently with a proper verification IP and OCP monitors. The test environment can apply to both the functional IP cores and bus fabric before the integration.

Several verification approaches such as directed test or random test can be applied upon the test environment.

Using directed test method, engineers write many test cases or test patterns to verify the multitudinous specifications straightforwardly. Nowadays random test technology can cover hidden or corner test cases and bugs especially in future reuse. Both above approaches can be implemented by controlling the master VIP to issue deterministic or random test patterns and check the protocol/response and then measure the coverage by monitors.

OCP-IP is also taking important steps to alleviate the verification challenges by providing the compliance check. The compliance checks can be used in several different ways. Formal verification can be used to prove OCP compliance of an IP.

They can also use the same checks in simulation environment with users specified directed test or random test. Compliance checks facilitate the assurance of verification quality and prove IP blocks to be compatible at the system level for further reuse.

6 Experimental Results

This section demonstrates the bus fabric development results based on our in-house profiles. As mentioned previously, the crossbar topology only targets the IP cores with HP interface and the shared-link focus on the GP and PP interfaces. Bridges are only necessary between different profiles, and between our profiles and legacy IPs.

Table 1: Implementation Results.

Design	Configuration	Gate Count (K)
Crossbar	3 masters to 3 slaves with 64-bit data	10
	3 masters to 3 slaves with 64-bit data and 2-entry FIFO	33
Shared-link	1 master to 8 slaves with 32-bit data	5.5
	1 master to 8 slaves with 32-bit data and 2-entry FIFO	42
HP to GP bridge	64-bit to 32-bit data	2.5
PP to APB bridge	32-bit data	0.2
GP to AHB bridge	32-bit data	1

Table 1 lists the implementation results for a certain interconnect configuration. The above cases are targeted at 333MHz on a 0.13 μ m process technology.

With the open and well-defined interface standard, easily configurable bus fabric, and the complete verification environment, we are seeing more and more design teams moving from proprietary interfaces to OCP. The increasing number of OCP-

compliant IP blocks quickly make a design platform for better IP-reuse.

In addition to easy integration and verification, some products, such as ADSL routers and Wireless LAN gateway controllers, also report better routing performance over legacy designs based-on proprietary interfaces.

7 Conclusion

The design of a large scale SoC is becoming challenging not only due to its complexity, but also the use of a large amount of IPs. A consensus interface standard for IP cores is becoming important and even inevitable for a successful SoC design. OCP, with its openness, collaborative, not-for-profit nature, and inherent large industry member base, is quickly becoming a feasible and preferable solution over a close or an in-house standard.

In this paper, we demonstrate the use of OCP and SoC integration with a hierarchical and heterogeneous interconnect. The proposed OCP profiles effectively simplify the design and verification of bus fabrics and bridges. Experimental results show the advantage of this methodology with small area cost. The comparison with a full-set OCP implementation is not shown due to its diversity and implementation cost. The area cost of a full-set OCP implementation is obviously much higher, but it also brings high flexibility. Commercial IP and tools are available for that purpose [7] [8] [9].

By sharing the experience of using OCP as an IP standard, we hope there will be more industry people aware of these SoC integration issues, and seeing this as an opportunity to enhance current IP and EDA environment, and finally make more large scale SoC designs possible.

References

- [1] ARM, "AMBA overview", <http://www.arm.com/>, 2007.
- [2] OCP-IP, "Open core protocol international partnership", <http://www.ocpip.org/>, 2007.
- [3] Vesa Lahtinen, "System level design experiences and the need for standardization", in *Proc. IEEE Int'l Conf. on System-on-Chip*, Tampere, Finland, Nov. 2006, pp. 1-4.
- [4] James Aldis, "Use of OCP in OMAP 2420", <http://www.ocpip.org/>, 2005.
- [5] Partha Pratim Pande, Cristian Grecu, Michael Jones, André Ivanov, and Resve Saleh, "Performance evaluation and design trade-offs for network-on-chip interconnect architectures", *IEEE Trans. Computers*, vol. 54, no. 8, pp. 1025-1040, Aug. 2005.
- [6] Natale Barsotti, Riccardo Mariani, Matteo Martinelli, and Mario Pasquariello, "Dynamic verification of OCP-based SoC", in *Proc. IEEE Int'l Conf. on System-on-Chip*, Tampere, Finland, Nov. 2005, p. 22.
- [7] Sonics Inc., "SMART interconnectTM", <http://www.sonicsinc.com/>, 2007.
- [8] ArterisTM, "Automated NoC IP design", <http://www.arteris.com/>, 2007.
- [9] Silistix, "CHAINworksTM", <http://www.silistix.com/>, 2007.