

# On-Chip Interconnect Protocol Stack Exploration for FPGA Board-to-Board Bridging

Ashkan Beyranvand Nejad<sup>1</sup>, Matias Escudero Martinez<sup>1</sup> and Kees Goossens<sup>2</sup>

A.BeyranvandNejad@tudelft.nl, M.EscuderoMartinez@student.tudelft.nl

K.G.W.Goossens@tue.nl

<sup>1</sup>Technical University of Delft, <sup>2</sup>Techniccal University of Eindhoven, the Netherlands

**Keywords:** *Protocol Stack, Bridge, Preserving QoS, On-Chip Interconnect, FPGA*

FPGA prototyping of Systems on Chip (SoC) has become very common for early software development and hardware design verification. Moreover, electronic consumer application demands and recent advances in chip design technologies are increasing the size of new SoCs dramatically. It causes the FPGAs to be very small, in terms of capacity, for prototyping the large designs in which recently there has been a large efforts of utilizing scalable on chip interconnection networks instead of the traditional bus paradigm. Such an on chip interconnection network known as Network on Chip (NoC) occupies large area on FPGAs, whereas its signals routing and floorplanning are very complex. Hence, to prototype a large design, a system is required to be partitioned into number of sub-systems, each of them is implemented on a single FPGA. An off-chip scheme is required then to connect these FPGAs in order to emulate the design. On the other hand, an FPGA emulation board typically includes one FPGA chip (e.g. Xilinx ML510 and ML605). There also exists multi-FPGA emulation boards that are very expensive to be afforded for research projects (e.g. BEE2). This pops up a need of having a bridging scheme between FPGA boards (off-board bridge). Significant research efforts have been done in different aspects of interconnecting separate chips. These aspects regard either the bridging over chips that are located on the same board (on-board bridging) or bridging over different boards (off-board bridging). The concept of sub-NoCs and their interconnection are discussed in [2].

The unified system resulting from connecting the sub-systems located on different boards, should provide the same behavior of the original system on a single board, i.e. the applications show the same functional behavior when running on a cluster of sub-systems and on the original system. In other words, the off-chip bridge maintains the application Quality of Service (QoS) requirements in terms of throughput and latency. In this work, we propose a bridging scheme between FPGA emulation boards on which a NoC-based SoC is implemented. There are a lot of possibilities to divide such a system into smaller sub-systems. We explore the various possible bridging schemes by investigating the possible bridge implementation at different layers of the protocol stack on each link.

The interconnect protocol stack which is based on the OSI reference model, is proposed for NoC-based system models in [1]. This model introduces 5 layers of Session, Transport, Network, Data link, and Physical for an on chip interconnect. Figure 1 gives an overview of on-chip interconnect which makes the data communication between a Master and a Slave Intellectual Properties (IPs). The interconnect takes advantage of both the traditional bus and the new NoC paradigms. As it is illustrated in the figure, there are various physical communication links in the interconnect to be cut and insert a bridge in between. This would divide the interconnect into

two sub-interconnects. Moreover, there is more than one layer of the interconnect protocol stack which is involved in a communication link. This would also imply that the bridge can be implemented at different layers of the protocol stack.

Here we explore the possible bridging schemes at different protocol stack layers of the interconnect links. For this purpose, first the requirements that a bridge should fulfill are prioritized as following: 1) *Transparency*: Ideally from an application point of view the bridge should be invisible. The lower protocol stack layer the bridge implemented at, the more transparent to the applications. 2) *Decoupling*: The bridge should support the full decoupling among two systems implemented on separate FPGA boards. Decoupling includes both physical and temporal dependencies, e.g. voltage levels, frequency, location distance, etc. 3) *Quality of Service (QoS)*: the bridge should preserve the Quality of Service (QoS) of the applications running on the bridged systems. 4) *Cost*: the cost is both the implementation cost in terms of area, and the time cost in terms of latency.

The possible bridging schemes are proposed starting from the lowest layer of the protocol stack at point 5 of the interconnect shown in Figure 1. The Schemes are illustrated in Figure 2. At point 5 and 4, Scheme I is at Physical layer and Scheme II is at Link Layer. One layer upper, Scheme III is at Network layer. Scheme IV and V are at Transport layer of point 3. Finally, Scheme VI is at the Session layer of point 1 and 2. In order to be able to compare different bridging schemes, selected properties of the interconnect links which have direct or indirect impact on the above mentioned requirements, are distinguished as design criteria. These criteria are: Parallel/Serial link, Flow Control, Buffering, Path Manipulation, Frequency Dependency, and Separate Connections. Based on these bridging design criteria and their affect on the requirements, the bridge schemes are compared in Table 1.

The comparison concludes that the Scheme V which is realized at Transport layer of the interconnect protocol stack, is the most suitable bridging scheme. In this Scheme the bridge is implemented at the serial streaming data link of the Network Interface (NI). The end-to-end flow control inside the sub-NoC is taken care by the NI and the link level flow control between two FPGA boards is provided by the bridge. There is a dedicated buffer per connection inside the bridge. The frequency dependency of two ends of the link is not an issue here since the interface of NI to the bridge is hand-shaked based. The bridge therefore implements a Time Division Multiplexing scheduler to preserve the connections quality of services. The architecture of the proposed bridge is depicted in Figure 3.

## References

- [1] A. Hansson and K. Goossens, "An on-chip interconnect and protocol stack for multiple communication paradigms and programming models," in *Int'l Conf. on Hardware/Software Codesign and System Synthesis (CODES+ISSS)*, Oct. 2009.
- [2] S. Evain, J.-P. Diguët, and D. Houzet, "NoC design flow for TDMA and QoS management in a GALs context," *EURASIP J. Embedded Syst.*, vol. 2006, no. 1, pp. 4-4, 2006.

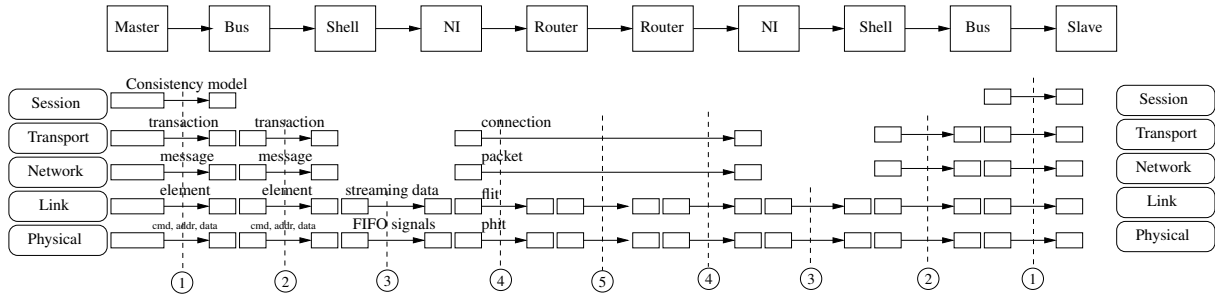


Figure 1. Overview of on-chip interconnect connection with the involved protocol stack layers at the links

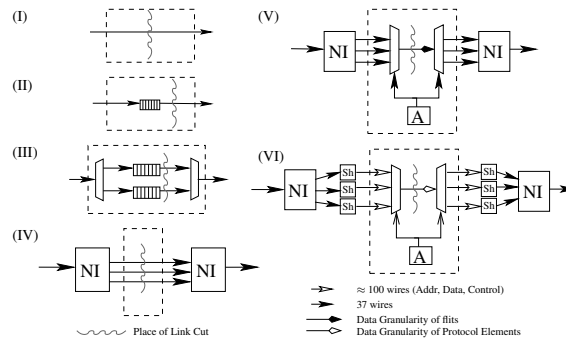


Figure 2. Bridging schemes based on the interconnect protocol stack

**Table 1. Pros. and Cons. of the bridging schemes**

Scheme	Decoupling	QoS	Area Cost	Latency
I (Physical Level)	-	-	+	-
II (Link Level)	-	-	+	-
III (Network Level)	-	+	-	-
IV (Transport Level)	+	-	-	+
V (Transport Level)	+	+	+	+
VI (Session Level)	+	+	-	-

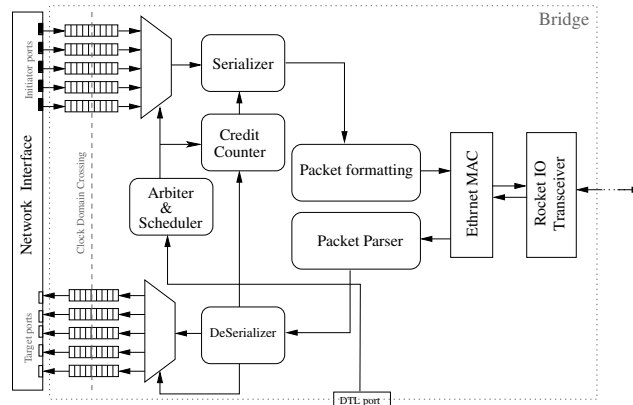


Figure 3. The bridge architecture