

# Evaluation of NoC on Multi-FPGA Interconnection Using GTX Transceiver

Atef Dorai

University of Rennes 1, IRISA/INRIA  
Lannion, France

Olivier Sentieys

University of Rennes 1, IRISA/INRIA  
Lannion, France

Helene Dubois

University of Rennes 1, IRISA/INRIA  
Lannion, France

**Abstract**—Multi-FPGA platforms are very popular today for pre-silicon verification of complex designs due to their low cost and high speed. The idea is to divide these systems into smaller sub-systems and implement each one on a separate chip. The challenge is that the number of I/Os available on FPGA remains constant despite the technological evolution. This problem is resolved by multiplexing several cut-signals using the time division multiplexing scheduling mechanism. This structure has a strong effect on the speed of transmission between FPGAs. However, an inter-FPGA bottleneck appears. In this paper, we focus on evaluating the Network-on-Chip on multi-FPGA using the high speed serial transceiver GTX block. In order to speed up the transmission between FPGAs, GTX Transceiver is used to provide a high bandwidth while using fewer pins compared to existing approaches based on ordinary FPGA I/O pins. Depending on the available multi-gigabit transceiver, the bandwidth per connection is between 3.125 and 28 Gb/s which allows for large amounts of data to be moved quickly between multiple FPGAs. In our evaluation, a VC707 platform based on the Virtex-7 device is used. The simulation results show that the proposed architecture provides low area consumption and latencies under different traffic patterns.

**Index Terms**—Inter-FPGA link, NoC, GTX Transceiver, multi-FPGA.

## I. INTRODUCTION

As the number of IPs (Intellectual Property) increases rapidly inside a chip, interconnecting them becomes increasingly challenging [1]. Network-on-Chip (NoC) is the most efficient architecture to build a many-core interconnect system. The debug and test of such complex architectures can be performed by prototyping them on programmable devices. Therefore, mapping a large-size NoC on a multi-FPGA requires several challenges in the prototyping flow in order to get the best performance due to the limited resources on FPGA (Field Programmable Gate Arrays). Therefore, a complete system is required to be partitioned into several FPGAs. As a result, a multi-FPGA platform is formed.

Multi-FPGA platforms offer the potential to supply high-performance solutions for computationally intensive tasks, logic emulation and rapid prototyping [2]. However, the large delays in the inter-FPGA communication and the limited I/O pins per FPGA restrict the bandwidth system performance and create a bottleneck [3]. The logic capacity per FPGA is increasing at a much faster scale as compared to the number of I/O pins. Indeed, emulating a large NoC on multi-FPGA is mainly divided into two steps. In the first step, the design is

divided and partitioned into several sub-NoCs. A successful partitioning approach ensures that every sub-NoC does not exceed the logic capacity of the FPGA [4]. The second step routes the signals of sub-NoC from inter-FPGA to the off-chip according to the available physical I/O resources. Over the past year, Time Division Multiplexing Access (TDMA) is used to route the signals of NoC to the limited inter-FPGA I/O [5]. However, an efficient inter-FPGA architecture is required for multiplexing the signal of sub-NoC at the source FPGA and demultiplexing them at the destination FPGA through a high-speed adapter protocol [2]. All the multiplexing, demultiplexing, sub-NoC and adapter protocol can be performed in different clock domains.

Last generation boards based on FPGAs provide a wide range of multi-Gigabit transceivers (GBT), clock conditioning modules and large amount of logic gates. The serial transceiver is much useful for applications where high-speed data transmission is required. It offers a higher bandwidth and a superior auto-adaptive equalization. Higher-order modulation schemes can also be used to increase performance [6]. The GTX which is the basic block for common interface protocols is becoming an increasingly popular solution for communication between FPGAs. The GTX supports line speeds from 500Mb/s to 12.5Gb/s [7]. Moreover, configuring the GTX for a single channel that moves the data to the other side securely is easy. The architecture of the 7<sup>th</sup> series FPGA [8] contains four different types of receivers: GTX, GTP, GTH and GTZ. These options range from high-performance options to low-power options. Therefore, in this work we propose an approach to interconnect a NoC on multi-FPGA based on the GTX transceiver available in almost all modern FPGAs.

The rest of the paper is organized as follows. In section II, we present an overview of our contributions. In section III we introduce our design flow. In section IV, we give a general introduction and provide detailed information on the GTX transceiver used in our evaluation. The experimental results are presented and analyzed in section V. Finally, section VI presents the conclusion of this work.

## II. RELATED WORK

[9] presents a hierarchical NoC architecture to support multi-chip platforms, which incorporates the required quality of service for multi-FPGA systems. The interconnection between chips is done by a generic bridge scheme at different

hierarchical levels of the NoC protocol stacks. The generic bridge is based on the Ethernet protocol to accelerate the transmission between chips. [2] proposes a new architecture for inter-FPGA traffic management dedicated to NoC on multi-FPGA. The proposed architecture is easily placed between the external protocol and the sub-NoC. The comparisons show that the random access mechanism can be an efficient solution for inter-FPGA compared to the planned schedule. [4] proposes an exploration flow in order to optimize the inter-FPGA for multi-FPGA prototyping. To check the exploration and optimization of the partitioning tools, five different FPGA boards are used when the number of FPGAs on board varies from two to six.

To ensure a fast communication with FPGA, [10] proposes a serial transceiver architecture based on dynamic clock phase shifting technology. This solution can handle all possible phase offsets between the transmitter and the receiver. [11] presents an auto-adaptive serial link able to support the reconfiguration of GTX parameters to take full advantage of the available bandwidth link by setting the highest rate.

### III. DESIGN FLOW

To evaluate the performance of GTX transceiver on multi-FPGA system, we propose a design flow based on existing NoC architecture. The flow generation of a NoC on multi-FPGA requires passing through several stages as synthesis and partitioning of the NoC on multi-FPGA. These steps require an adaptation between the sub-NoC, the TDMA inter-FPGA architecture and the GTX transceiver. A detailed discussion of these steps is provided in the following.

#### A. Synthesis and Partitioning of the NoC

The synthesizable architecture depends on the configuration of NoC as the number of routers, flit size and the routing algorithm used. Once the NoC design is defined, it can be generated as input for the Vivado tools from Xilinx. Synthesis is the first process to adapt the NoC architecture with the target FPGA device. Firstly, it transforms the hardware description into a grid level representation and mapped it onto the target FPGA. Since the NoC architecture considered is larger than the number of resources available in the target FPGA, it must be partitioned into several parts and each part should be implemented on multiple chips. Partitioning applied on the design is a very critical step as it has a direct relationship to architecture performance. Efficient partitioning takes the size of each partition into account and is directly related to the capacity of the target FPGA and the number of connections between the different partitions. Once the design is partitioned, an inter-FPGA architecture is integrated to adapt the NoC signals with the multi-Gigabit transceiver.

#### B. Inter-FPGA Architecture

Although partitioning a NoC on multi-FPGA attempts to tackle the resource limitation problem on FPGA, but due to a large gap between the FPGA logic capacity that can support a significant number of routers, the number of cut signals is larger than the number of IO pins available on FPGA. Indeed,

an inter-FPGA structure which adapts the on-chip with the off-chip is interesting. It can be seen from Figure 1 that inter-FPGA architecture based on TDMA is used to route the cut signals of sub-NoC to the transceiver. The objective of the inter-FPGA architecture is to ensure a communication between the sub-NoC and the GTX with low latency and resource. The inter-FPGA architecture is generic in the sense that it supports all sub-NoC based handshake flow control signals. Once the inter-FPGA architecture is designed, we can assume that each sub-NoC can be successfully placed and routed into an FPGA.

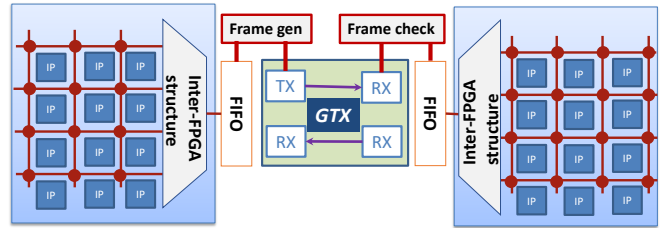


Fig. 1: Structure for NoC partitioning on multi-FPGA

### IV. GTX TRANSCEIVER

The multi-Gigabit Transceiver used in this work is based on the GTX Transceiver deployed in the Xilinx Virtex7 series. Inside the FPGAs, GTXs are available as robust configurable cells. Each cell contains a pair of transceivers that share basic components such as the reset logic and the phase-locked loop (PLL). Figure 2 illustrates a simplification of the components included in each transceiver such as the transmitter (TX) and the receiver (RX). Before explaining the configuration of the GTX, it is interesting to detail the basic elements constituting this architecture.

#### A. Structure of the GTX Transceiver

The GTX transceiver consists of two main blocks TX and RX. First, at the transmitter and before serialization, a parallel data stream at a frequency well below 1 GHz is converted if necessary into a transmit signal using an appropriate Line Encoder (TX) as 8b/10 encoder. Indeed, most multi-Gigabit transceivers allow to bypass the line encoder to leave coding to the FPGA fabric. After, at the serialization step, the TX PLL generates the reference clock for the high-speed transmitter clock and for the PISO (Parallel Input Serial Output). Finally, an optional equalizer is applied in the analog front end of the transceiver to correct the disturbance of signals.

At the receiver side, the transmitted signals pass firstly through another equalizer. This latter is used to correct any frequency distortion at the transmission lane. Then, the incoming signals are detected and a recovered clock is generated using a receive clock that is provided by the reference clock RX PLL. This step is very interesting because in most cases the clock of the receiver is not exactly equal to the transmitted clock located on another FPGA. After, the latest signals are deserialized via a SIPO (Serial Input Parallel Output) block. This deserialization is done at a frequency below 1 Ghz. The

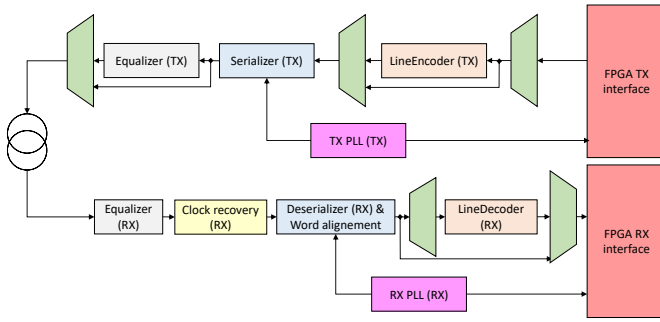


Fig. 2: Simplified scheme of the GTX Transceiver

correct alignment of the received signals is also performed in this step. Finally, data is decoded if necessary using an appropriate decoder as 10b/8b and is transferred to the elastic buffer in order to be adapted to the difference clock domains. Also, as illustrated in Figure 1, Frames *gen* and *check* can be used to adapt the packet with FPGA TX and RX interfaces.

### B. GTX Lane Configuration

GTX transceiver requires a specific packet format which is illustrated in Figure 3. The chosen packet length is fixed to 32 bits. Any transmission of packet is started by a header and followed by a payload data. The header contains five fields that are TAG, destination, size, source and the number of packets. The ID field allows to identify and route the packet between the inter-FPGA structures. The GTX receiver searches the incoming serial comma field, so that all the packets that follow are aligned and RX receiver reorganizes parallel data. The GTX can be configured by supporting a wide variety of

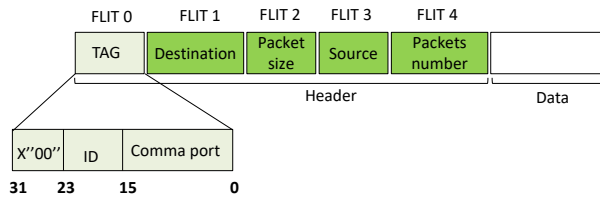


Fig. 3: Packet format

protocols like PCI-Express, SATA or Ethernet. These protocols are used to operate data links in the range of several Giga-bits per second. All links operate at a speed of 3.125 Gb/s requiring a reference clock operating at a frequency of 156.25 MHz.

The *Frame gen* module adapts the control signals and the packet format. It depends on the protocol and the Encoder/Decoder used in the GTX. The packets received from the multiplexer side are of size 32-bit. The incoming packets are stored in the FIFO. At the output of the *Frame gen* block, the packet size is 80 bits. The frame checker monitors the received packet, ensures the adaptation of packets format and control signal between RX transceiver and demultiplexer. The *Frame check* works by first scanning the TAG field (comma port) to detect the incoming packets sent from sub-NoC.

## V. EXPERIMENTS

In this section, we evaluate the number of resources and the transmission time of our proposed flow under different traffic patterns. We interconnect two sub-NoC based SoC to the GTX transceiver. The complete system is implemented on the same VC707 board based on Xilinx FPGA Virtex-7 [8]. Several configurations are applied to the GTX to prove the performance of NoC on multi-FPGA systems.

### A. GTX configuration

Due to hardware constraints, there are different sources of clock like SGMII, Si570 and FMC HPC connector. The SGMII mode supports the Ethernet interface and is deployed in our evaluation. The clock frequency is fixed to 156.25 MHz.

### B. Emulation platform

In order to evaluate our architecture, an emulation platform is used to generate different traffic patterns. This emulation platform is constituted by traffic generators and receivers that are used to generate and process the packets, respectively. The choice of the synthetic traffic patterns is based on the objective evaluation of NoC on multi-FPGA. The source-destination pairs must be in different sub-NoC structures. The synthetic traffic patterns are bit-complement, bit-reverse and transpose [12]. The simulated NoC has a size of  $4 \times 4$ , with a total of 16 routers. We performed our simulation by comparing the static XY routing algorithm with the dynamic XY (Dxy) routing algorithm [13]. The Dxy considers the local traffic state in decision making in which each router compares the congestion condition in the instance input buffers of neighboring routers. The average latency is measured with the packet injection rate from 10% to 100%. The FIFO integrated between the inter-FPGA structure and GTX transceiver is large and can support up to 1024 packets.

### C. Resource analysis

The on-chip architecture that includes the GTX transceiver and the inter-FPGA structure is synthesized on Virtex-7 using Vivado tools and also is implemented on VC707 board. As observed in Table I, the number of resources used by the on-chip structure is less than 1% of the total FPGA. The number of resources does not exceed 5% for the complete system when adding the  $4 \times 4$  NoC architecture. We observe that the evaluation platform requires more LUTs than registers for a given NoC size.

TABLE I: The on-chip synthesis results

Cell	On-chip structure	Complete system
Slice as LUTs	1189 (0.39%)	16 277 (5.36%)
Slice as Flip Flop	609 (0.1%)	6186 (1%)

### D. Timing analysis

The first test is based on the bit-complement synthetic traffic (Figure 4). Each router sends 100 packets and the packet size is 10 Flits. With this synthetic traffic, there are eight transmitter and receiver routers in each sub-NoC. In the first

test, the XY and Dxy routing algorithms demonstrate a very similar behavior. The XY have slightly lower latency than the Dxy routing algorithm during the packet injection rate. The saturation point is obtained around 20%.

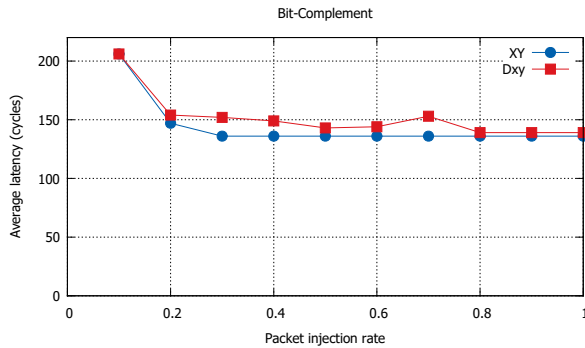


Fig. 4: Average latency under bit-complement traffic pattern

The second test scenario is based on the transpose synthetic traffic (Figure 5). Each router sends 100 packets and the packet size is 10 Flits. Using this synthetic traffic, there are four transmitter and receiver routers in each sub-NoC. Also in this test, the XY and Dxy routing algorithms demonstrate a very similar behavior. However, the Dxy have slightly lower latency than the XY routing algorithm during the packet injection rate. The saturation point is obtained around 30%.

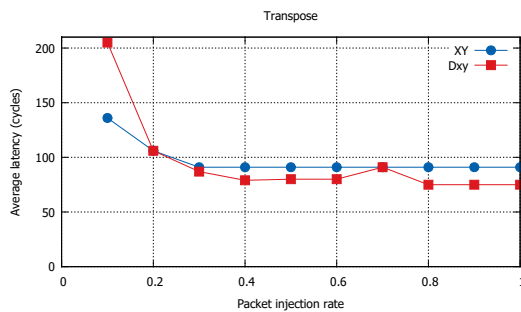


Fig. 5: Average latency under transpose traffic pattern

The bit-reverse traffic model is applied on the third test scenario (Figure 6). Each router sends 100 packets and the packet size is 10 Flits. With this synthetic traffic, there are three transmitter and receiver routers in each sub-NoC. Unlike the others test scenarios, it is difficult to extract the appropriate routing algorithm as the results are very similar. The saturation point is now obtained around 50%.

## VI. CONCLUSION

Multi-FPGA platforms are now widely used for prototyping a SoC with large scale. However, these platforms suffer from a bottleneck on inter-FPGA structure due to the delay between on-chip and off-chip as well as the limited bandwidth between FPGAs due to the limited number of IOs per FPGA. In order to solve this bottleneck problem, we multiplexed the cut-signals of NoC to a Multi-Gigabit transceiver (MGT). This MGT used

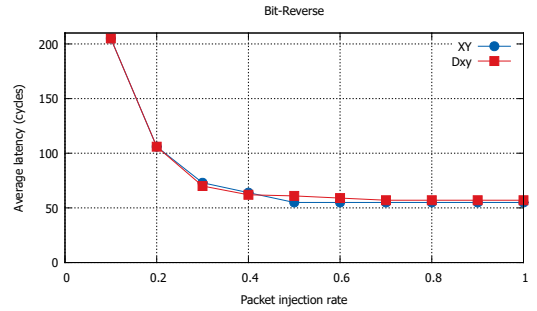


Fig. 6: Average latency under bit-reverse traffic pattern

is the GTX transceiver for achieving higher performance. The included inter-FPGA structure and GTX transceiver represent an efficient on-chip architecture for NoC on multi-FPGA. The complete system is implemented on VC707 board and the resources used are less than 1%. The timing results showed that the on-chip structure provides low latency guarantees to traffic for different considered traffic patterns.

## REFERENCES

- [1] F. Lemonnier et al., "Towards future adaptive multiprocessor systems-on-chip: an innovative approach for flexible architectures," in *Proc. IEEE Int. Conf. on Embedded Computer Systems: Architectures, Modeling and Simulation (IC-SAMOS)*, Jul. 2012, pp. 1–6.
- [2] A. Dorai, V. Fresse, C. Combes, E.-B. Bourennane, and A. Mtibaa, "A collision management structure for noc deployment on multi-fpga," *Microprocessors and Microsystems*, vol. 49, pp. 28–43, 2017.
- [3] Q. Tang, H. Mehrez, and M. Tuna, "Multi-fpga prototyping board issue: the fpga i/o bottleneck," in *2014 International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS XIV)*, July 2014, pp. 207–214.
- [4] U. Farooq, R. Chotin-Avot, M. Azeem, M. Ravoson, M. Turki, and H. Mehrez, "Inter-fpga routing environment for performance exploration of multi-fpga systems," in *Rapid System Prototyping (RSP), 2016 International Symposium on*. IEEE, 2016, pp. 1–7.
- [5] F. Mao, W. Zhang, B. Feng, B. He, and Y. Ma, "Modular placement for interposer based multi-fpga systems," in *Great Lakes Symposium on VLSI, 2016 International*. IEEE, 2016, pp. 93–98.
- [6] J.-M. Philippe, S. Pillement, and O. Sentieys, "A low-power and high-speed quaternary interconnection link using efficient converters," in *Proc. of the IEEE International Symposium on Circuits and Systems, ISCAS'05*, 2005, pp. 4689–4692.
- [7] M. Cannon, M. Wirthlin, A. Camplani, M. Citterio, and C. Meroni, "Evaluating xilinx 7 series gtx transceivers for use in high energy physics experiments through proton irradiation," *IEEE Transactions on Nuclear Science*, vol. 62, no. 6, pp. 2695–2702, Dec 2015.
- [8] Xilinx, "Vc707 evaluation board for the virtex-7 fpga user guide," Tech. Rep., UG885 (v1.7.1) August 12, 2016.
- [9] A. B. Nejad, A. Molnos, M. E. Martinez, and K. Goossens, "A hardware/software platform for qos bridging over multi-chip noc-based systems," *Parallel Computing*, vol. 39, no. 9, pp. 424–441, 2013.
- [10] X. Liu, Q.-X. Deng, and Z.-K. Wang, "Design and fpga implementation of high-speed, fixed-latency serial transceivers," *IEEE Transactions on Nuclear Science*, vol. 61, no. 1, pp. 561–567, 2014.
- [11] A. Aloisio, R. Giordano, V. Izzo, and S. Perrella, "A frequency agile, self-adaptive serial link on xilinx fpgas," *IEEE Transactions on Nuclear Science*, vol. 62, no. 3, pp. 955–962, 2015.
- [12] N. Nedjah, L. S. Junior, and L. de Macedo Mourelle, "Congestion-aware ant colony based routing algorithms for efficient application execution on network-on-chip platform," *Expert Systems with Applications*, vol. 40, no. 16, pp. 6661–6673, 2013.
- [13] M. Ebrahimi, "Fully adaptive routing algorithms and region-based approaches for two-dimensional and three-dimensional networks-on-chip," *IET Computers & Digital Techniques*, vol. 7, no. 6, pp. 264–273, 2013.