

Multi-FPGA Communication Interface for Electric Circuit Co-Simulation

Michel Lemaire, Daniel Massicotte and Jean Bélanger*

Université du Québec à Trois-Rivières, Department of Electrical and Computer Engineering,
Laboratoire des signaux et systèmes intégrés
3351, Boul. des Forges, Trois-Rivières, Québec, Canada

*OPAL-RT Technologies, 1751 Rue Richardson suite 2525, Montréal, QC

{michel.lemaire1, daniel.massicotte}@uqtr.ca, jean.belanger@opal-rt.com

Abstract – Real-time simulation of electric circuit is most often used to test real components connected to a real-time simulator. The increasing size and complexity of the simulation as well as the demand for better accuracy, lower time step, have pushed these simulations onto new hardware. For already more than a decade FPGA simulation is used by real-time simulation companies around the world to effectively simulate circuits under the μ s. With the computation requirement growth, multi-FPGA simulation needs to be considered as a valuable asset but attention must be given to the latency between the simulations for accuracy and stability. In order to minimize the communication latency, a custom interface and communication architecture for co-FPGA simulation is proposed. This paper presents detailed work on this architecture and shows promising results.

Keywords – Communication inter-FPGA, real-time simulator, low time-step, low latency.

I. INTRODUCTION

Real-time simulation of electric circuits has gained a lot of popularity over the year, allowing research facilities to test real equipment with the help of a simulated circuit. This approach not only fasten time to market and reduce development costs, but it also allows researchers to easily test equipment in hard to reproduce and hazardous condition without risk of damaging real equipment. The increasing complexity of electric circuits, as well as the higher switching frequency of power electronic devices and the demand for better accuracy, as driven real-time simulation to simulate circuit with time-step under the microseconds. For this purpose, a lot of simulations are now computed on FPGA to ensure the latency required by the targeted application.

To achieve the lowest latency, electric circuits implemented on FPGA are modeled using mainly one approach, the Modified Nodal Admittance (MNA). The inverse of the matrix defining the electric circuit is often pre-calculated prior to the computation. The FPGA is used only to solve the system of equation directly through multiplication of a matrix and a vector. One of the main differences between electric circuit model is how switching components, like transistors and diodes, are modeled. For example, Pejovic [1] uses only one inverse calculation for all possible switches combinations in a circuit while other methods need to store all possibilities. This method greatly limits the memory size required, which is greatly limited on FPGA compared to CPU and is a key advantage to achieve very low time-steps. When high accuracy is required, modeling switches as resistive component is limited by the available local memory since every possibility of the

switch's states are pre-calculated. Such method has been used successfully for several years on CPU-based solvers [2] since modern processors are equipped with large caches of memory. However, this technique is difficult to implement on FPGA for large circuit due to the limit of local memory [3][4].

In order to simulate large and complex electrical circuits, two-step solvers like ARTEMIS-SSN [5] and MATE [6] has been successfully implemented on CPU processors. These techniques first calculate several sub-circuits in parallel and then solve the circuit equation with interface voltages using Thevenin or Norton equivalents to computes the results at each time step. The accuracy of the results obtains by these solvers is very high since no artificial delays are introduced to solve a large circuit in parallel. CPU-based simulation is, however, typically limited to time steps between 5 μ s and 100 μ s, which is sometimes too large for some applications requiring time step below 1 μ s as it is the case for precise simulation of power electronic circuits. One solution to reach a very small-time step for large circuit simulation is to use more than one interconnected FPGA [7].

Large circuits are becoming more and more present with the electrification of transport and the rising complexity of micro-grid implemented in-land (PV, wind turbines, battery storage, etc...), in trains, ships, and more electrical aircraft. For example, the more electric aircraft has not only a complex electric circuit but also a lot of redundancy. At first glance, the problem seems simple since it is only a matter of establishing communication between two or more FPGA, but latency is crucial for accuracy and stability in real-time simulation. The problem is caused by the simulated circuit that is split on multiple FPGA and computed in parallel. If the latency between the two co-simulation is too high, phase shift is introduced in the circuit, the accuracy is then compromised, and depending on the circuit, the delay can even cause instability. When simulating circuits under the microsecond the communication latency is crucial and every tenth of nanoseconds allows for better accuracy. A customized approach is therefore needed to reduce latency to the lowest possible.

In this paper a customize interface was created using the Multi-Gigabit Transceiver (MGT). This interface is specially made for real-time simulation of electric circuits on multiple FPGA. This article will focus on data transmission of 32 bits, representing single-precision floating-point data.

The paper is organized as follows; Section 2 provides a description of the developed communication protocol dedicated

to the simulation of electrical circuits, Section 3 provides a description of our comparison study methods and performance. Finally, Section 4 reports the conclusions.

II. FPGA TRANSMISSION

Communication between FPGA platforms is a topical issue for multiplatform systems. In fact, delays caused by communications can be slower than the running logic implemented on a single FPGA. Therefore, the latency of communication is crucial, especially in real-time simulation.

High-level communication architectures mainly use Logic Multiplexing (LM), Serializer/Deserializer (SerDes), Low Voltage Differential Signal (LVDS), and Multi-Gigabit Transceiver (MGT) [2]. Each of them can be used independently or in conjunction. The issues surrounding these inter-FPGA communication architectures are presented and compared in [8]. Some approaches to communication architecture use multiple LVDS with SerDes [9]. The structure of the transmitters and receivers at the base of this architecture makes it possible to obtain promising results. In fact, according to the results obtained by the author, communication times are three times lower, 160 ns, than those used by standardized architectures using MGTs, such as the AURORA™ protocol developed by Xilinx®. The work in [10], where over twenty architectures are compared, presents a custom communication architecture using MGT. This work shows that standard communication protocol is not necessarily optimized for inter-FPGA communication. He specifically criticizes the AURORA™ protocol for having no reliability layer which has the effect of reducing the speed of information transfer. AURORA™ is also criticized in [11]. Following an analysis, the authors reject its use and define a custom architecture based on an MGT architecture to reduce latency. The results show communication delays lower than standard architectures. It achieves its results by disabling some blocks used in conventional MGT communications. For example, the phase adjustment register is replaced by an alignment circuit to reduce delays. It should be noted that this article is based on communication with an external controller and not another FPGA. The proposed architecture is still suitable for inter-FPGA communications.

Built-in MGT are becoming faster with each new product being released by FPGA company and presents ease of use that cannot be circumvented. In this article, MGT are selected because they show better latency than other wired communication architecture as seen in literature [10] [11] and are better suited for future implementation in a real-time simulator.

MGT are built within the FPGA and are essentially SERDES that converts parallel signal to serial and serial to parallel capable of operating at a high bit rate. They are composed of a transmitter and a receiver operating with an external clock with a frequency define by the desired bit rate. They are composed of different blocks allowing 8-bit/10-bit encoding and decoding, clock recovery, buffers, gearbox to allow 64-bit transmission, pseudo-random binary sequence (PRBS), and integrated comma detection. Xilinx® MGT needs to be programmed using the transceiver wizard IP core that generates the desired structure of the transmitter and the receiver.

A. FPGA TX interface protocol

The transmitter of the MGT transceiver is composed of two blocks, the Physical Coding Sublayer (PCS) and Physical Medium Attachment (PMA) as shown in Figure 1 [12]. Following the user guide [12], certain blocks of the PMA can be bypassed to minimize latency, like the 8B/10B encoder and Phase Adjust First-In, First-Out (FIFO) when using a 32-bit data path or less. In the proposed architecture, in order to minimize the latency of the transmitter, the Phase Adjust FIFO was bypass but to ensure proper transmission 8b10b encoder is kept within the MGT architecture. The delay associated with the encoder is only one clock cycle and the reliability of the communication is crucial for our application, so encoding cannot be bypass.

In order to achieve minimum latency, the interface between the simulation and the data sent through the transmitter is carefully designed. To ensure the integrity of the data sent onto the receiver, Cyclic Redundancy Check (CRC) is used within the interface. The CRC used is implemented in parallel as stated in [13], allowing calculation in one clock cycle. The communication between the interface and the simulated circuit is also asynchronous as the interface uses the MGT clock which is different from the system clock used by the simulation. The interface uses two different physical clocks for proper

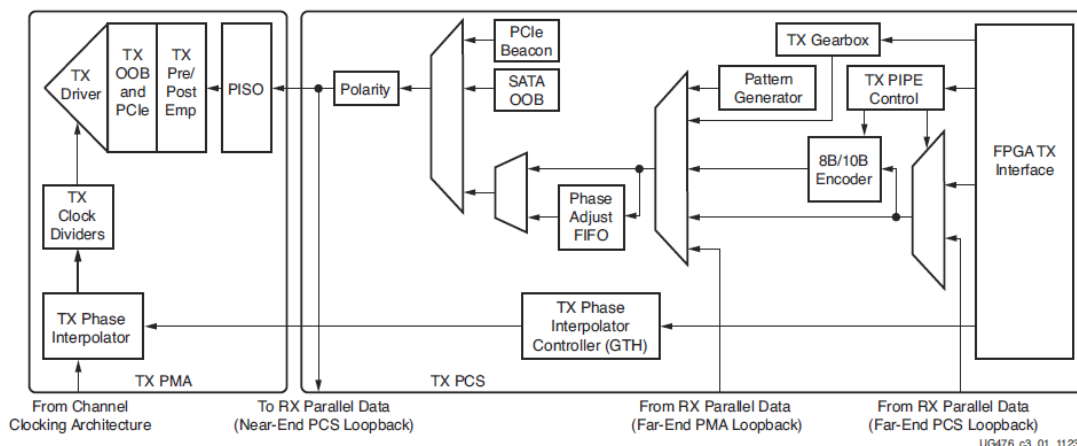


Figure 1. MGT transmitter structure schematics

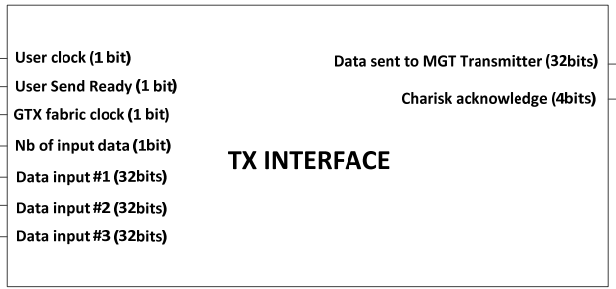


Figure 5. Proposed transmitter FPGA interface blocks schematics

acknowledgment of data sent by the users and allows up to 3 parallel inputs to represents decoupling of the circuit composed of 3 phases. Figure 2 shows the input and output of the MGT transmitter interface used for FPGA TX Interface as in Figure 1.

The transmitter interface is conceived as follows. When the interface is waiting for user data, only a comma for proper alignment is sent to the transmitter. When the user acknowledges that his data is ready (handshake), considering that the simulation sends a 3-phase signal, a free-running counter is activated. The 3 data are directly sent to 3 parallel CRC blocks while the first data is already being sent to the transmitter and all data are stored in registers. The CRC value for the 3 data input is then concatenated in one 32-bit word and sent to the transmitter. The second data is then sent to the transmitter followed by the third data. In the fifth cycle, a comma is sent to ensure the proper alignment of the data. Redundancy is applied, if selected as an option, to ensure the integrity of the data. It introduces 4 more cycles in the transmission protocol and if no error is detected on the data

received, the total latency remains untouched. This is selected only if the total latency of the simulation time step allows the extra cycles. The FPGA interface is built so that it takes only 5 clock cycles to send 3 32-bit words, plus the CRC, needed on the receiver end. Figure 3 shows the algorithm describes.

B. RX interface protocol

The receiver of the MGT transceiver is composed of two blocks, PCS and PMA [11] as shown in Figure 4. In order to minimize latency certain block of the PMA can be bypass like the 8-Byte/10-Byte decoder, the comma detect and align, and the RX elastic buffer used to resolve differences between the clock domains. In order to minimize the latency associated with the MGT receiver, RX elastic buffer was bypass as it is the one associated with the highest latency. Comma detects and align and 8-bit/10-bit encoder is kept ensuring data integrity and alignment. The interface uses 2 clocks and output up to 3 data, a data error code and read acknowledgment as shown in Figure 5.

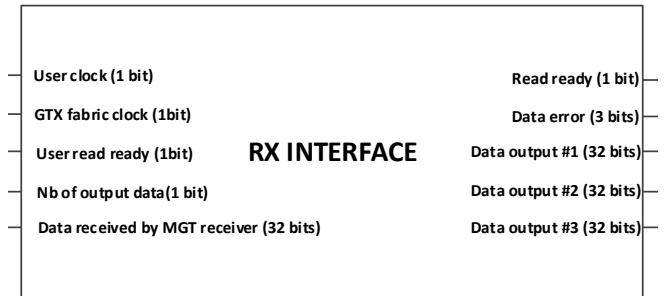


Figure 2. Proposed receiver FPGA interface blocks schematics

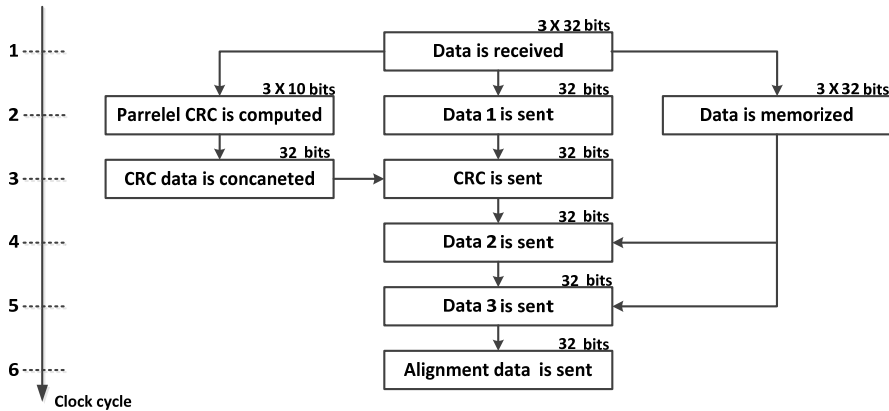


Figure 4. Transmitter FPGA interface algorithm

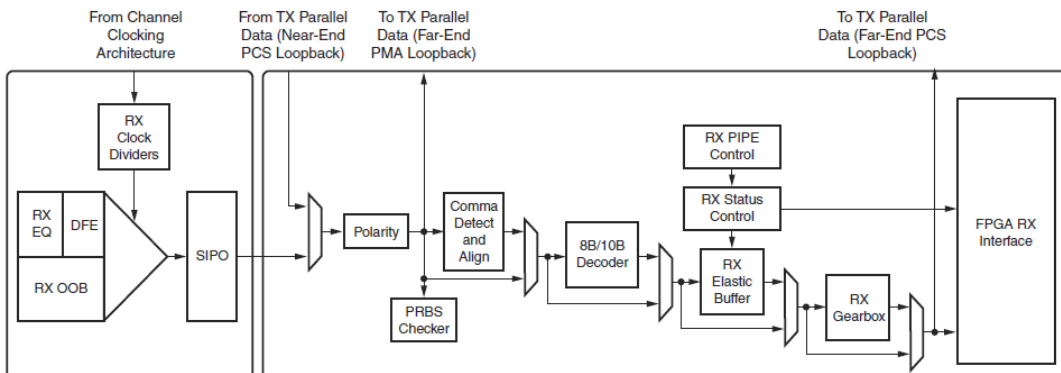


Figure 3. MGT receiver structure schematics

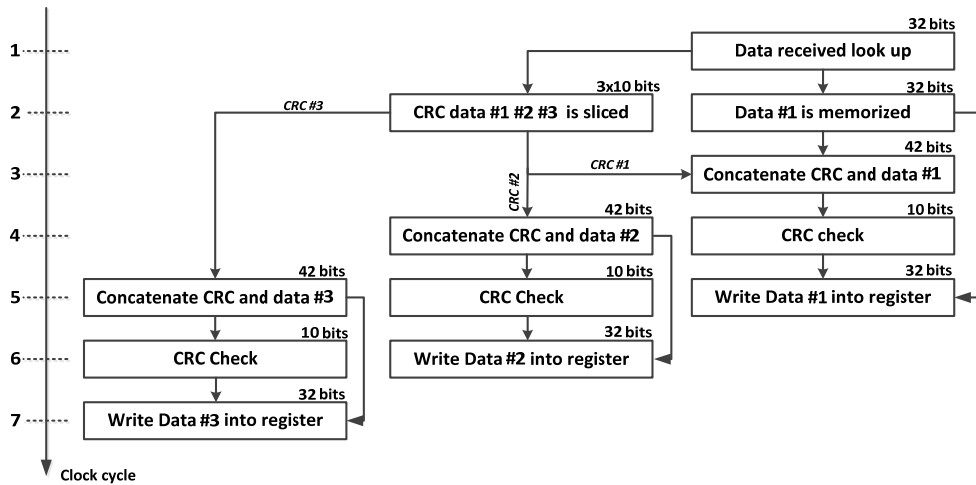


Figure 6. Receiver FPGA interface algorithm

The receiver interface is pipelined with proper parallelism to minimize the clock cycle needed to interface the simulation with the incoming data. The receiver interface is conceived as follows. The interface looks if the data received is the alignment data or not. If not a free-running counter is activated. The first data received is saved in a register and then the second data received is sliced in order to obtain the 3 data CRC code. On the third, fourth, and fifth cycle the 3-phase data received are concatenated with their respective CRC and sent to the CRC receiver. The process is pipeline so that the first data is sent from the CRC receiver to the proper register while the second data is computed in being process and so on. Within seven cycles, considering incoming data lookups, all data are written in the register and the user is acknowledged that the data is ready to be read. Therefore 7 clock cycles are needed in order to process the incoming data. Figure 6 shows the receiver interface architecture.

III. EXPERIMENTAL RESULTS AND DISCUSSIONS

The results shown in this article are obtained using Xilinx® FPGA VC707 Development Board using the fiber-optic link. The MGT are programmed using Vivado® transceiver wizard IP core and the interface is programmed in VHDL. The data sent to the transmitter interface is refreshed every 200 ns time step running with a 100 MHz clock to represent the electric circuit simulation and the MGT uses a line rate of 5 Gb/s and a clock of 125 MHz using 32-bit transmission data path. All the results are obtained using Integrated Logic Analyzer (ILA) IP core as a logic analyzer. The data transmitted are 4 bytes words, the handshake on the transmitter is sent every 200 ns while the receiver handshake from the user side is kept to logical 1 state.

First, as we can see from Table 1, the resources needed for both interfaces are negligible using less than 1% of the resources on a VC707 FPGA.

Figure 7 shows the latency between the data sent from the simulation up to the user receiving end. In other words, from the input to the transmitter interface input to the receiver interface output. We can see that 27 clock cycles are necessary for total data transmission. Figure 8 shows the delay between the data sent out of the transceiver interface to the MGT transmitter up to the receiver interface input. We can see that

17 clock cycles are necessary for total transmission over fiber-optic cable and PCS. The delay between the data input of the transmitter interface until all data has been transmitted to the MGT transmitter is shown in Figure 9. We can see that 6 clock cycles are necessary for the total transmission of the 4 data needed. In Figure 10 the delay between the data received from the MGT receiver until it is read by the user is shown. Note that the user acknowledges rad bit is always one for this result. We can see that 8 clock cycles are necessary for total transmission.

IV. RESULTS COMPARISON WITH AURORA PROTOCOL

As we can see from Table 1, by bypassing the buffer in the MGT transceiver we are able to gain 7 clock cycles. The results from our proposed MGT communication are compared to AURORA which uses RX buffer. The results are shown in Table 2.

When comparing the overall latency with AURORA protocol, we can see that the transmission of one 4 bytes word is 2 times lower using the custom interface built in this article for the same line rate and clock frequency. The results are shown in Table 3.

The result shown in this paper shows that proper customization of the transceiver interface of the MGT with buffer bypass can reduce by two the latency compared to standard communication protocol like AURORA. Since the application is specific, it is possible to reduce the transmission delay with a custom

Table 1: Resources used by the proposed communication interface

Communication Interface	LUTs	Slice registers
Transmitter interface	203	195
Receiver interface	142	437
MGT with interface	598	912

Table 2. Comparison between custom protocol and AURORA™

Method	M1	M2
Aurora protocol	24	48
Proposed	17	24

M1 MGT Transmitter to MGT receiver latency (clock cycle), transmission delay without interface.

M2 Transmitter interface input to receiver interface output for one data (clock cycle).

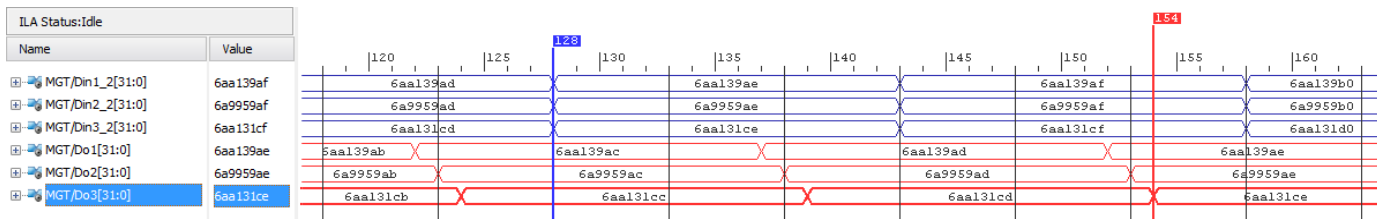


Figure 7. ILA result for transmission from MGT transmitter FPGA interface to MGT receiver FPGA interface

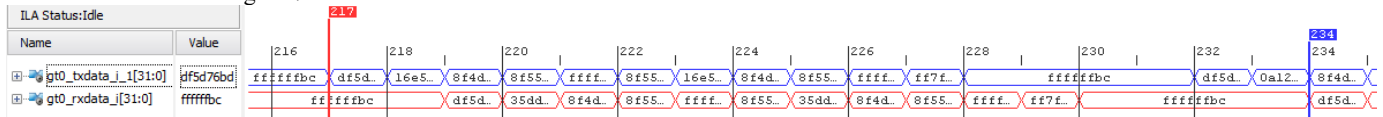


Figure 8. ILA result for transmission from MGT transmitter to MGT receiver

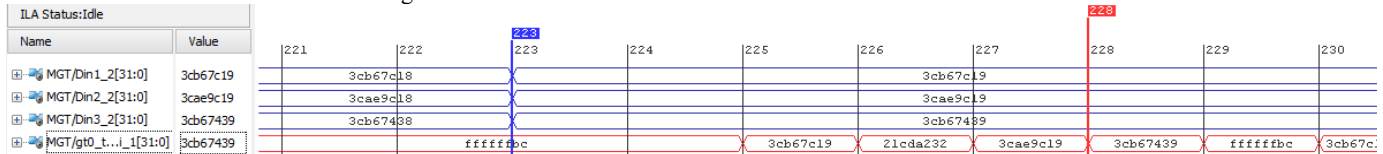


Figure 8. ILA result for transmission from MGT transmitter FPGA interface to MGT transceiver input

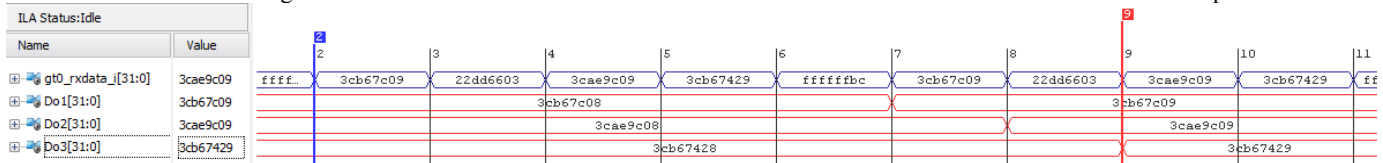


Figure 7. A result for transmission from MGT receiver to MGT receiver FPGA interface

protocol. It is important to understand that Aurora is a commercial product that was not intended for multi-FPGA simulation but for high throughput and ease of use.

These results are promising as they reduce the total latency of the communication path allowing, in theory, better accuracy of future simulations on a multi-platform system.

V. CONCLUSION

In conclusion, this work shows that custom communication protocol is better suited for grid and power electronic circuit simulation on multiple FPGA since the communication latency is easily sliced in two compared with commercial products. On the other hand, the delay associated with inter-FPGA communication still introduced a delay and the accuracy of the simulation might still be a problem even if it is two times faster than the AURORA protocol. In this article, a line rate of 5 Gb/s was used but the VC707 allows for up to 12 Gb/s. This means that the delay associated with the communication can be easily sliced in two, leading to a total latency of around 108 ns compared with the latency of 216 ns obtain in this paper. Since the delays are known, it is always possible to use a proper decoupling method to compensate for the delay within the simulation to augment accuracy and assure stability. For grid-type simulation, by decoupling simulations on long-distance lines, it is possible to convert the delay in line distance to minimize the error but for circuit simulation where line distance is negligible other methods need to be developed in order to maximize the accuracy. Future works will integrate co-simulation on multiple FPGA of electric circuit simulation in order to show the impact of the delay on the accuracy of the overall simulation and will be based on decoupling technique to ensure stability and assure accuracy.

ACKNOWLEDGMENTS

This work has been funded by the Natural Sciences and Engineering Research Council of Canada grant, Mitacs, CMC Microsystems, Opal-RT, and the Chaire de recherche sur les signaux et l'intelligence des systèmes haute performance.

REFERENCE

- [1] Pejović, P.; Maksimovic, D., "A method for fast time-domain simulation of networks with switches," *IEEE Transactions on Power Electronics*, vol.9, no.4, pp.449-456, Jul 1994
- [2] C. Dufour, S. Abourida and J. Belanger, "InfiniBand-Based Real-Time Simulation of HVDC, STATCOM and SVC Devices with Custom-Of-The-Shelf PCs and FPGAs," *2006 IEEE International Symposium on Industrial Electronics*, Montreal, Que., 2006, pp. 2025-2029.
- [3] X. Zhou, G. He and X. Zhou, "FPGA Design and Implementation for Real-Time Electromagnetic Transient Simulation System," *IEEE International Conference on High Performance Computing and Communications*, New York, NY, pp. 848-851, 2015.
- [4] M. Matar and R. Iravani, "FPGA Implementation of the Power Electronic Converter Model for Real-Time Simulation of Electromagnetic Transients," *IEEE Transactions on Power Delivery*, vol. 25, no. 2, pp. 852-860, April 2010.
- [5] C. Dufour, J. Mahseredjian and J. Belanger, "A combined state-space nodal method for the simulation of power system transients," *2011 IEEE Power and Energy Society General Meeting*, Detroit, MI, USA, 2011, pp. 1-1.
- [6] M. Matar and R. Iravani, "FPGA Implementation of the Power Electronic Converter Model for Real-Time Simulation of Electromagnetic Transients," *IEEE Transactions on Power Delivery*, vol. 25, no. 2, pp. 852-860, April 2010.
- [7] M. Rivard, C. Fallaha, A. Yamane, J. Paquin, M. Hicar and C. J. P. Lavoie, "Real-Time Simulation of a More Electric Aircraft Using a Multi-FPGA Architecture," *IECON 2018 - 44th Annual*

- Conference of the IEEE Industrial Electronics Society*, Washington, DC, 2018, pp. 5760-5765.
- [8] Q. Tang, H. Mehrez and M. Tuna, "Multi-FPGA prototyping board issue: the FPGA I/O bottleneck," *Int. Conf. on Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS XIV)*, Agios Konstantinos, pp. 207-214, 2014.
- [9] P. Godbole, A. Batth and N. Ramaswamy, "High speed multi-lane LVDS inter-FPGA communication link," *IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, Coimbatore, pp. 1-4, 2010.
- [10] A. Theodore Marketos, P. J. Fox, S. W. Moore and A. W. Moore, "Interconnect for commodity FPGA clusters: Standardized or customized?," *International Conference on Field Programmable Logic and Applications (FPL)*, Munich, 2014, pp. 1-8
- [11] D. Makowski, G. Jablonski, P. Predki, and A. Napieralski, "Low latency data transmission in LLRF systems," Particle Acceleration Conference, New York, NY, USA, Apr. 2011, pp. 877-879.
- [12] Xilinx (2018). 7 series FPGAs GTX/GTH Transceivers [Online]. https://www.xilinx.com/support/documentation/user_guides/ug476_7Series_Transceivers.pdf, Accessed on: Aug 14, 2018.
- [13] G. Albertengo and R. Sisto, "Parallel CRC generation," *IEEE Microwave*, vol. 10, no. 5, pp. 63-71, Oct. 1990.