Implementation of Multi-FPGA Communication using Pulse-Coupled Phase Oscillators

Dinda Pramanta, Takashi Morie, Hakaru Tamukoh
Graduate School of Life science and systems engineering, Kyushu Institute of Technology
2-4 Hibikino, Wakamatsu-ku, Kitakyushu 808-0196, Japan
E-mail: dinda-pramanta@edu.brain.kyutech.ac.jp, morie@brain.kyutech.ac.jp, tamukoh@brain.kyutech.ac.jp

Abstract

This paper proposes an implementation of multi-Field Programmable Gate Array (FPGA) communication using pulse-coupled phase oscillators. At first, we construct a digital phase oscillator circuit with adjustable in-phase parameters. While performing the synchronization task, the oscillators are connected into the first input and first output (FIFO) interface. The communication in-between of FPGAs will occur by enabling the following inputs and outputs on the FIFO through the Gigabit Transceiver (GTX) clock domain. Pulse-coupled phase oscillators of Winfree's model are utilized as a spike generator and we expect the ideal of model circuit will synchronize. We employ two FPGA boards of Virtex6 ML605 and implement four oscillators on a hardware level. Experimental results show that first spike synchronizing over two FPGAs takes 12.47 μsec with data bit speed stream 3.2Gbps.

Keywords: pulse-coupled oscillators, synchronization, Winfree's model, FIFO, GTX, multi-FPGA,

1. Introduction

Pulse-coupled phase oscillator is one of the spiking neural networks. It consists of individual oscillators and a network of mutual interactions, which represented by the phase sensitivity function. [1, 2] Coupling between oscillators is determined by the timing of the spike pulse output from each oscillator. The oscillators are often assumed to be identical or nearly identical. In this case, the network reaches the *synchronization*-state where all the oscillators have the same phase. Based on the updating principles, one-bit signal line is required between oscillators for spiking the pulse. This feature is efficient for hardware implementation.[3, 4]

A field programmable gate array (FPGA) is a reconfigurable integrated circuit (IC) to perform complex interface and logic processing in a small footprint.[5] In addition, a multi-FPGA based design[5] overcomes the limitation of single FPGA resources and enables to implement large scale pulse coupled phase oscillator networks. However, communication delay between FPGAs may affect *synchronizing*-state. Therefore, high-speed communication is required to implement oscillators over the multi-FPGA and *synchronizing*-state should be verified.

In this paper, we propose an implementation of multi-FPGA communication using pulse-coupled phase oscillators. First, we design a digital circuit of pulse-coupled oscillators[3] based on the Winfree model.[2] Second, we propse a communication method between two FPGAs using the designed digital oscillator, the first input and first output (FIFO) interface and the serial connection of Gigabit Transceiver (GTX). As the multi-FPGA platform, we employ two FPGA boards and connect them by the serial connection of GTX. Experimental results show that the four oscillators network reaches to *synchronization*-state and first spike synchronizing over two FPGAs takes 12.47 μsec with data bit speed stream 3.2Gbps.

2. Pulse Coupled Oscillator model

In order to design the hardware implementation of pulse-coupled oscillator, Winfree model[2] provides an efficient ways, the fundamental of coupled phase oscillators is expressed as follows:

$$\frac{d\phi_i}{dt} = \omega_i + Z(\phi_i)Spk(t), \quad (1)$$

© The 2017 International Conference on Artificial Life and Robotics (ICAROB 2017), Jan. 19-22, Seagaia Convention Center, Miyazaki, Japan

where ϕ_i is the *i*-th phase variable with 2π periodicity, ω_i is the *i*-th natural angular frequency, $Z(\phi_i)$ is a phase sensitivity function, which gives the response of the *i*-th oscillator. Inputs from other oscillators, *Spk*(t), are assumed here pulse input as follows:

$$Spk(t) = \frac{K_0}{N}\sum_{j=1}^{N}\sum_{n=1}^{\infty}\delta(t - t_{jn}), \qquad (2)$$

where K_0 is the coupling strength, *N* is the number oscillators, and t_{jn} is the firing time. Mathematically, δ is a Dirac's delta function that represents the input spike pulse has a defined width Δ*t* during which ϕ_i is updated according to the value of $Z(\phi_i)$.

2.1. *Discretized model*

In order to implement Eqs. (1) and (2) into digital hardware, oscillator is regularized into following expressions:

$$\phi_i(t+1) = \phi_i(t) + \omega_i \frac{K_0}{N}\sum_{j=1}^{N} Z(\phi_i)\mathrm{Spk(t)} \qquad (3)$$

$$Spk_j(t) = \begin{cases} 1, & if\ \phi_j(t) = \phi_{th} \\ 0, & otherwise \end{cases} \qquad (4)$$

The phase value of the next time step is calculated by summing the current phase value, natural angular frequency, and a product of the phase sensitivity function and input pulses. Natural angular frequency ω_i is set at a constant value. The oscillator outputs will spike the pulse when the phase variable reaches a threshold value ϕ_{th}. By the above discretization, the proposed model can be implemented with simple logic circuit.[3]

2.2. *How the oscillator works*

Basically the concept of oscillator works is based on neighbor connection between each oscillator. The model explains the local interaction only, but can generate various phenomena in the large oscillator network (in-array), based on a coupling function formula as shown in Eqs. (1) and (2).

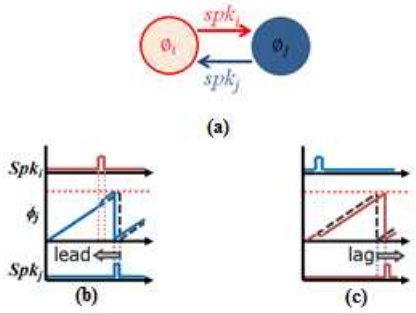


Fig. 1. Pulse-coupled phase oscillator: (a) Coupled network schematic, (b) positive update, and (c) negative update

.

Figure 1 shows a two pulse-coupled phase oscillators and their updating manner. Synchronization is occured by updating the pulse timing. There are two phases of updating the from each oscillator, which are positive updating and negative updating. Positive update will conducts whenever spike input (Spk_i) to another oscillator ϕ_j , the leading condition triggers the function $Z(\phi_j)$, which gives effect to the pulse timing for reaching the maximum value become earlier. Negative update conducts whenever Spike input (Spk_j) to another oscillator ϕ_i , the lagging condition triggers the function $Z(\phi_i)$, which gives effect to the pulse timing for reaching the maximum value become later.

3. Hardware architecture design

Hardware architecture design of a pulse-coupled phase oscillator is shown in Fig.2.(a). It consists of an oscillator circuit, a function generator circuit and an update circuit.[3] The oscillator circuit consists an *n*-bits counter (CNT), a spike generator (SPKGEN) and combinational circuits. The *n*-bit counter represents a phase variable ϕ_i and counts the clock inputs for realizing ω_i in Eq.(4). In this design, time step *t* in the discretized model corresponds to a clock cycle.

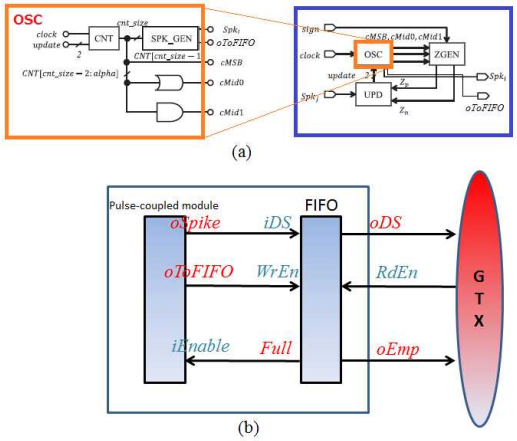


Fig. 2. Design of pulse-coupled phase oscillator circuit: (a) Properties of pulse-coupled oscillator circuit, and (b) FPGA's for each board overview.

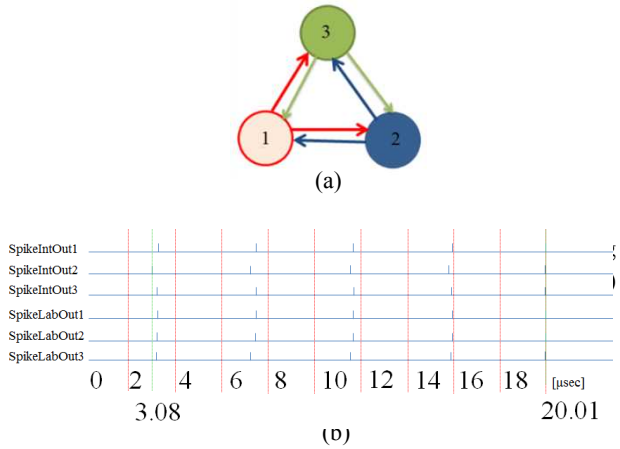
The oscillator circuit also outputs intermediate signals *cMSB*, *cMid0* and *cMid1* which determine the shape of the function $Z(\phi_i)$ and are used in the function generator circuit. The function generator circuit combines signals *cMSB*, *cMid0* and *cMid1* into *Zp* and *Zn*. The update circuit receives *Zp*, *Zn* and spike pulses Spk_j received from other oscillators and outputs signal update.

In order to write data into the FIFO, spike output (oSpike) goes into the DataIn (iDS) and then strobe the write enable signal input WrEn high for one clock cycle as shown in Fig.2.(b). WrEn is output from oToFIFO into the FIFO's internal memory. If writing in bulk the WrEn signal can be left high while changing the data on the DataIn (iDS) for each clock cycle. When the Full flag goes high, this means that the FIFO's memory is full and will not accept any more writes (iEnable = 0) until data is read using the RdEn.

4. Experimental results

Evaluation of the proposed model of hardware design was realized. The pulse-coupled phase oscillator networks over two FPGAs through serial communication of Gigabit Transceiver (GTX) explains through subsection parts. In all experiments, respectively basic synchronizing phenomena could be observed (in-phase mode).

4.1. *In-phase three oscillators of pulse coupled spiking oscillator inside one FPGA*



Real-time implementation results of simple *rotary* three oscillator connection are shown on Fig.3.(a). Each of oscillator has two outputs represents as label and unit. Figure 3(b) shows the condition of FPGA when running from the beginning time zero (0µsec), Δclock between sychronzing was achieved 132 clock cycles. Testing architecture was using 16-bit of data and internal max clock (200MHz) from FPGA Virtex-6 ML605. Inside the simulation part, for one second could trigger and count the counter 200 Mega cycles, in this case first spike was occurred on 3.08 µsec and reached the *synchronization*-state on 20.01 µsec. The total of data rate is equal to 16bit

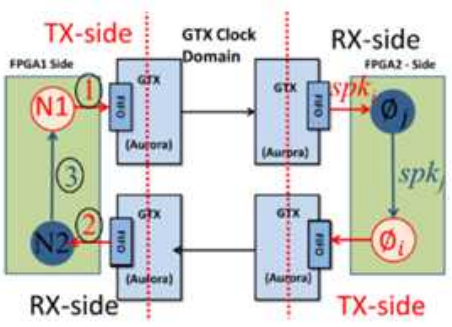


Fig. 4. Four oscillators over tow FPGAs using serial communication.

* 200 Mega counter/sec which reaches the maximum value of 3200bps or 3.2Gbps.

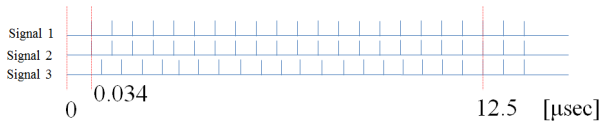


Fig. 5. Realtime result of four pulse-coupled phase oscillators over two FPGAs using *chipscope*. Signals 1, 2, 3 are corresponds to the numbers in Fig. 4.

4.2. *Four oscillators over two FPGAs using serial communication*

After we achieved with simple oscillator inside one FPGA, we employ two FPGA boards as the multi-FPGA platform. By using Virtex-6 XC6VLX240T ML605 Xilinx board design, it provides a free and open high-speed protocol for multi-FPGA communication called Aurora.[7] It is intended for serial communication between FPGAs with speeds up to and above 10 Gbps.[8]

Figure 4 shows the experimental setup using two FPGA boards of Virtex6 ML605 and implementation of four oscillators on a hardware level. Figure 5 shows the observation result of synchronization in FPGA. Here, we employed *chipscope* which is in-circuit debugger provided by Xilinx to observe spike pulses. From the result, the spike synchronization in-phase was achieved and a total time consumed around 12.47 µsec.

4.3. *FPGA design summary*

Table 1 shows the results of total usage resources of the multi-FPGA, communication using pulse-coupled phase oscillators. Four oscillators were implemented and carried out onto the hardware level with the FPGA synthesizer of Xilinx Tools. By using ISE Design Suite software information, we obtained the report of the proposed model circuit implemented with total maximum frequency 434.972MHz.

5. Conclusion

In this paper, we proposed an implementation of multi-FPGA communication using pulse-coupled phase oscillators. In the experiment, two FPGA boards were used to implement four oscillators network, and first spike synchronization over two FPGAs took 12.47 µsec with data bit speed stream 3.2 Gbps. From experimental results, we verified that the pulse-coupled phase oscillator synchronized over two FPGAs via the high-speed serial communication and the FIFO interface.

Table 1. Device utlization

	Used	Available
Slice Register	416	301440
Look-Up Tables	348	150720
IOBs	101	600

In future work, we will increase the number of FPGAs and implement large scale pulse coupled phase oscillator networks, then apply it to engineering application such as image processing.

Acknowledgements

This research was supported by JSPS KAKENHI Grand Numbers 26330279 and 15H01706.

References

1. Y. Kuramoto, *Chemical Oscillation, Waves, and Turbulence,* (Springer, Berlin, 1984).
2. A. T. Winfree, *The Geometry of Biological Time,* (Springer, NewYork, 1980).
3. Y. Suedomi, H. Tamukoh, K. Matsuzaka, M. Tanaka, and T. Morie, *Parameterized Digital Hardware Design of Pulse-coupled Phase Oscillator Networks*, Neurocomputing, **165** (2015), 54–62.
4. Kenji Matsuzaka et al, *VLSI Implementation of Coupled MRF model using pulse-coupled phase oscillators,* Electronics Letters. (2015), 46-48.
5. J. Li, Y. Atsumari, H. Kubo, Y. Ogishima, S. Yokota, H. Tamukoh, and M. Sekine, *A Multidimensional Configurable Processor Array - Vocalise*, IEICE Trans. on Information and Systems, **E98-D**(2) (2015), 313-324.
6. X. Y. Wang, A. Apsel, Pulse coupled oscillator synchronization for low power UWB wireless transceivers, in: *Proc. of 50th Midwest Symposium on Circuits and Systems* (2007) 1524–1527.
7. Xilinx Inc, *Aurora 8B/10B Protocol Specification*, (2010),http://www.xilinx.com/support/documentation/ipdocumentation/au-rora8b10bprotocolspecsp002.pdf.
8. Abhijit Athavale and Carl Christensen, *High-Speed Serial I/O Made Simple*. Xilinx,Inc., 1.0 edition, (2005), http://www.xilinx.com/publications/archives/books/serialio.pdf.