Synchronization of Pulse-Coupled Phase Oscillators over Multi-FPGA Communication Links

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 study proposes an implementation of pulse-coupled phase oscillators over multiple field-programmable-gate-array (FPGA) communication links. Two FPGAs are connected by a gigabit transceiver and a First-In First-Out interface. To verify the effect of communication delay between FPGAs on oscillator synchronization, we implement four oscillators on the multi-FPGA platform. We have successfully observed synchronization over two FPGAs correctly, despite of a 0.1 μs communication delay. The measurement results show that first spike synchronization requires 12.47 μs with a 3.2 Gbps communication throughput.

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

1. Introduction

Pulse-coupled phase oscillator networks are a type of spiking neural network and a promising model for realizing brain-like very large-scale integration (VLSI). Mathematical models of coupled oscillators have been proposed and intensively studied.[1-6] In these models, mutual interactions between oscillators are represented by a simple phase sensitivity function, and the couplings between oscillators are determined by the timings of spike pulses output by each oscillator. Based on this principle, nonlinear processing is performed by a single transition in spike timing. In addition, only one-bit signal lines are required between oscillators for spike pulse communication. These features allow us to achieve efficient hardware implementation, and high-performance and low-power intelligent information processing VLSI systems can be constructed based on spike-based computation.

To implement spike-based computation and demonstrate its effectiveness, we have proposed analog VLSI circuits[7-9] and digital circuits using field programmable gate arrays (FPGAs).[10,11] For the analog VLSI, we implemented a region-based coupled Markov random field model using pulse-coupled phase oscillators and applied it to image region segmentation. Measurement results showed that the proposed device achieved high efficiency in terms of performance per power consumption.[9] In the FPGA Implementation, we introduced a parameterized design that enabled us to emulate various sizes and interconnections of pulse-coupled oscillator networks.[11] From this previous work, we have confirmed that spike-based computation is suitable for realizing brain-like VLSI. However, a large-scale network cannot be implemented on a single chip owing to limited circuit resources.

A multi-FPGA-based design would overcome the resource limitations of a single FPGA[12] and enable us to implement large-scale pulse-coupled phase oscillator networks. However, communication delays between FPGAs might affect synchronization between oscillators, which is the most important function of pulse-coupled phase oscillator systems. Therefore, high-speed communication is required to implement oscillator systems over multiple FPGAs, and the synchronization state over a multi-FPGA system should be verified.

In this study, we propose an implementation of multi-FPGA communication links for pulse-coupled phase oscillators. First, we design a digital circuit of pulse-coupled oscillators based on the Winfree model.[2] Second, we propose a method for communication between two FPGAs using these digital oscillators, First-In First-Out (FIFO) interface, and a serial connection using a gigabit transceiver (GTX).[13] We employed two FPGA boards for the multi-FPGA platform and connected them by a GTX serial connection. Experimental results show that a four-oscillator network reached a synchronization state, and first spike synchronization over two FPGAs took 12.47 μs, with a datastream bitrate of 3.2 Gbps. This study is an extension of our previous work.[15]

2. Pulse-coupled Oscillator Model

The Winfree model[2] provides an efficient way of designing pulse-coupled oscillator circuits. The fundamental relation for pulse-coupled oscillators can be expressed as follows:

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

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

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

Where K_0 is the coupling strength, *N* is the number of oscillators, and t_{jn} is the firing time. Mathematically, δ is a Dirac delta function that represents the timing of input spikes without a pulse width. However, in physical hardware, spike pulses have a definite width Δt, during which ϕ_i is updated according to the value of $Z(\phi_i)$.

2.1. *Discretized model*

To implement Eqs. (1) and (2) in digital hardware, the model of pulse-coupled phase oscillators is discretized as follows.

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

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

The phase value at the next time step is calculated by adding the current phase value, natural angular frequency, and the sum of products of the phase sensitivity function and input pulses. The natural angular frequency ω_i is assumed to be constant. The oscillator outputs a spike pulse when the phase variable reaches threshold value ϕ_{th} and then resets; $\phi_i = 0$. Using the above discretization, the pulse-coupled phase oscillator model can be implemented with simple digital logic circuits.[11]

2.2. *Pulse-coupled phase oscillator dynamics*

Conceptually, the oscillator operation is based on the connections between each oscillator and its neighbors. This model is determined by only local interactions but can generate various phenomena in a large oscillator network based on the coupling formulas shown in Eqs. (1) and (2).

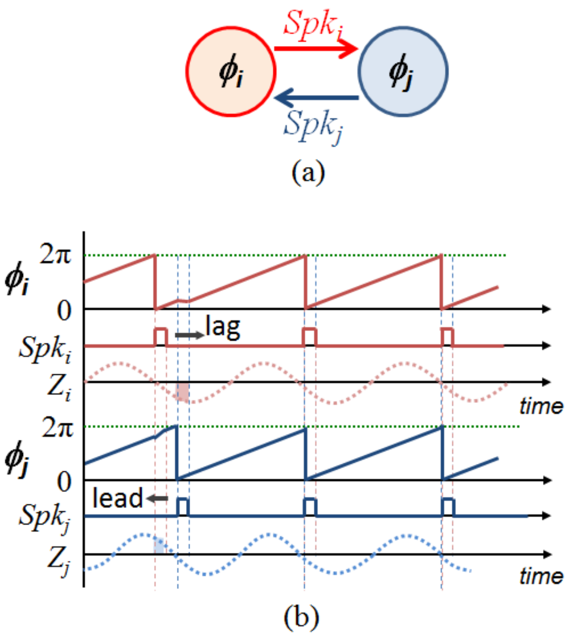

Fig. 1 Pulse-coupled phase oscillators. (a) Coupled network schematic. (b) Positive and negative updates.

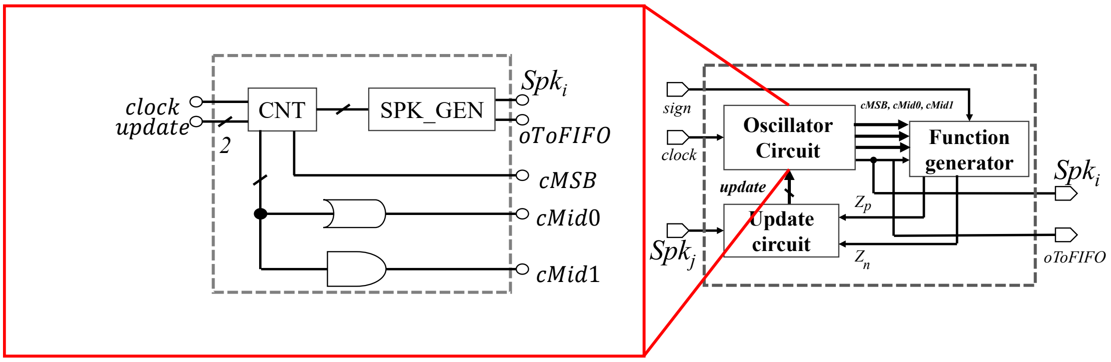

Fig. 2 Pulse-coupled phase oscillator circuit design.

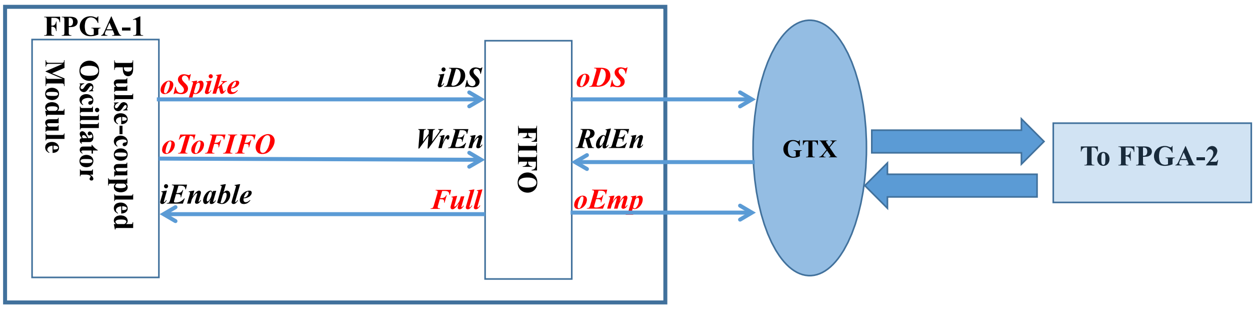

Fig. 3 Multi-FPGA communication system.

Figure 1 shows a schematic of two pulse-coupled phase oscillators and a timing diagram that explains their updates. The two oscillators are coupled by the spikes Spk_i and Spk_j. We assume that the phase sensitivity function is $Z(\phi_i) = -\sin(\phi_i)$. Synchronization occurs by updating the pulse timing. There are two update phases for each oscillator: positive updating and negative updating. Positive updating occurs whenever there is spike input (Spk_i) to another oscillator ϕ_j and the leading condition triggers the function $Z(\phi_j)$, making the timing of the maximum value of ϕ_j earlier. Negative updating occurs when there is spike input (Spk_j) to another oscillator ϕ_i and the lagging condition triggers the function $Z(\phi_i)$, making the timing of the maximum value of ϕ_i later.

3. Hardware Architecture Design

3.1. *Digital design of pulse-coupled oscillators*

The hardware architecture of the pulse-coupled phase oscillator is shown in Fig. 2. It comprises an oscillator circuit, a function generator circuit, and an update circuit.[11] The oscillator circuit contains an *n*-bit counter (CNT), a spike generator (SPKGEN), and combinational circuits. The CNT represents a phase variable ϕ_{i} and counts clock inputs to implement ω_i in Eq. (3).

In this design, each time step *t* in the discretized model corresponds to a clock cycle. The signals *cMSB*, *cMid0*, and *cMid1* determine the shape of the function $Z(\phi_i)$ and are used in the function generator circuit. The function generator circuit combines these signals to output *Zp* and *Zn*. The update circuit receives *Zp* and *Zn*, and Spk_{j} that is the output received from the other oscillator, and outputs an update signal.

3.2. *Multi-FPGA communication system*

Figure 3 shows a block diagram of the multi-FPGA communication system. A FIFO interface is inserted between the pulse-coupled oscillator module and the GTX.

The pulse-coupled oscillator circuit module has three main channels, as shown in Fig. 3. For every spike-data output (*oSpike* = 1) from the oscillator module, a data valid signal (*oToFIFO* = 1) is sent at the same time to the FIFO interface.

The FIFO module is used as an interface between two clock domains, i.e., the oscillator and GTX clocks. When the FIFO receives a spike (*iDS* = 1) with a write enable signal (*WrEn* = 1), the FIFO writes the spike into its internal memory. When the *Full* flag is high (*Full* =

1), the FIFO's internal memory is full and it will not accept any further data for writing (*iEnable* = 0).

The GTX is a configurable transceiver (Tx) and is tightly integrated with the programmable logic resources of the FPGA. The FIFO interface is connected to the GTX and transfers spikes from the FPGA-1 side to the FPGA-2 side. To configure the GTX, we employed a free and open high-speed communication protocol called Aurora.[13] It controls FIFO's interfaces (*RdEn*, *oDS*, and *oEMP*) to read data from the FIFO and provides serial communication between the FPGAs at up to 10 Gbps.[14]

3.3. *Multi-FPGA platform using two FPGAs*

We employed two Virtex-6 XC6VLX240T ML605 Xilinx FPGA boards as the multi-FPGA platform. The hardware connections between the two FPGAs are shown in Fig. 4. A SubMiniature version A (SMA) cable connection was used for the physical environment hardware between the two FPGAs' GTX channels. The ML605 boards provide a differential signal connection for each Tx and receiver (Rx). Therefore, four SMA cables are required to connect to the two FPGA boards.

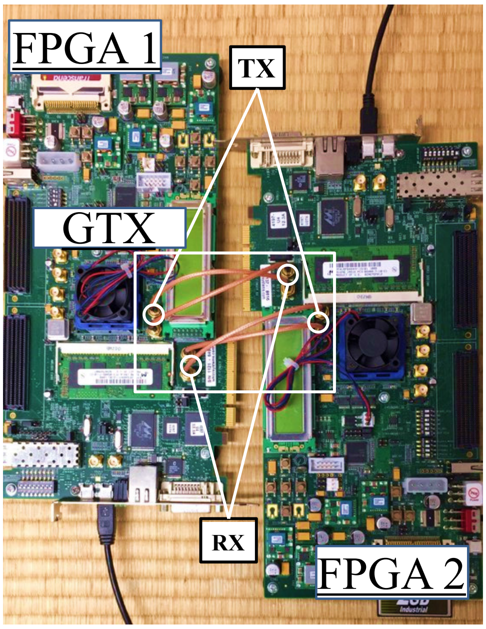

Fig. 4 Multi-FPGA platform using two Virtex6-XCVL240T ML605 boards.

4. Experimental Results

In general, physical connections between FPGAs incur communication delays. If the communication delay affects the synchronization states of the pulse-coupled phase oscillator networks, we cannot implement a large-scale network using the multi-FPGA platform. Therefore, to show the feasibility of the multi-FPGA platform, we examined basic synchronization phenomena in pulse-coupled phase oscillator networks.

4.1. *Performance of multi-FPGA communication*

To evaluate the performance of the serial connection, we measured the throughput and latency of the multi-FPGA communication. We implemented a 16-bit incremental data generator (counter) in FPGA-1. FPGA-2 received the incremental data and looped them back to FPGA-1. The clock frequency of the FPGAs was set to 200 MHz. Figure 5 shows the measured throughput results for the multi-FPGA platform. "Counter" represents the number of clock cycles for FPGA-1. It counted up to 200 million every second because the clock frequency is 200 MHz. "Throughput" represents the number of valid 16-bit data values that were received in a given time.

From these results, we can calculate the data rate as 16 bit * 199719549/s = 3195 Mbps ≈ 3.2 Gbps. In addition, we measured the difference in the incremental data between the Tx and Rx modules to find the latency of the multi-FPGA communication. The results show that the total latency from Tx to Rx was 20 counters or 0.1 µs.

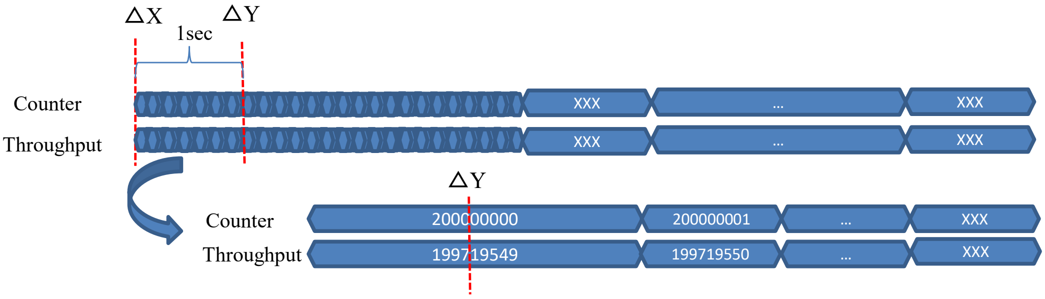

Fig. 5 Throughput measurement results using a 200 MHz clock of the Virtex6-XCVL240T ML605.

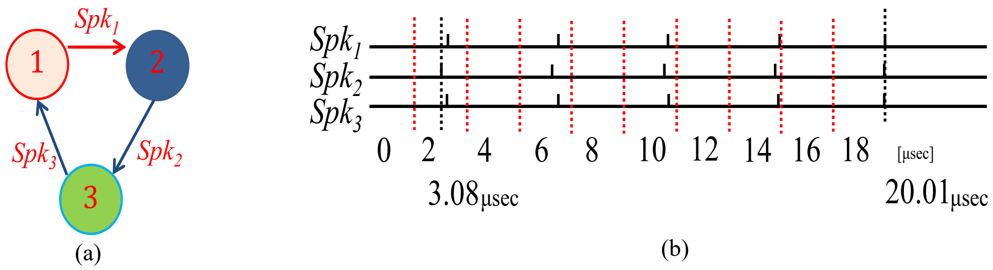

Fig. 6. Pulse-coupled spiking oscillator with three in-phase oscillators. (a) Three oscillators with rotary connections, and (b) synchronization results from ChipScope.

4.2. *Synchronization inside one FPGA*

To demonstrate the basic operation of pulse-coupled oscillator networks in one FPGA, we implemented a simple rotary network, as shown in Fig. 6(a). Here, we employed ChipScope, which is an in-circuit debugger provided by Xilinx[11] to observe spike pulses. Figure 6(b) shows the state of the FPGA, starting from the beginning at time zero (0 µs). In this case, the first spike occurred at 3.08 µs and it reached a synchronization state at 20.01 µs.

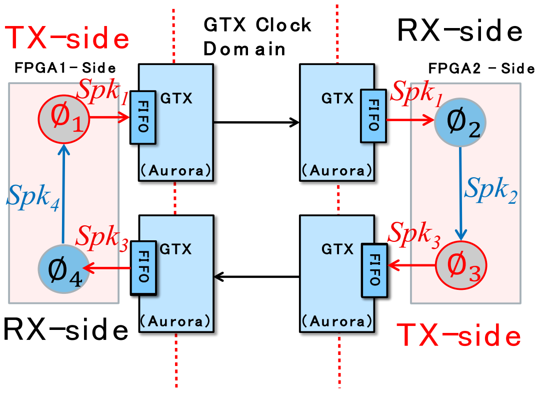

Fig. 7. Implementation of four oscillators over two FPGAs.

Table 1. Device utilization.

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

4.3. *Synchronization over two FPGAs*

Figure 7 shows an implementation of four oscillators on the multi-FPGA platform. Figure 8 shows the measurement results of synchronization in FPGA-1 using ChipScope. These results show that in-phase synchronization was correctly observed at 12.47 µs. We have therefore confirmed that our pulse-coupled phase oscillator networks were able to reach a synchronization state over the multi-FPGA platform, even though communication between the FPGAs included a 0.1 µs latency.

4.4. *FPGA design summary*

Table 1 shows synthesized results for the total resources used on one FPGA. Two oscillators were implemented at the hardware level with the FPGA synthesizer in Xilinx Tools. Using ISE Design Suite software information, we can report that the proposed model circuit was implemented with a maximum frequency of 434.972 MHz.

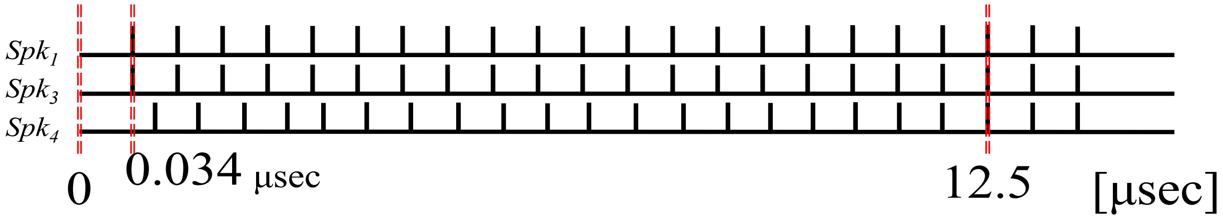

Fig. 8. Measurement results for four pulse-coupled phase oscillators over two FPGAs using ChipScope. The signals Spk_1, Spk_3, and Spk_4 correspond to those in Fig. 7.

5. Conclusion

In this study, we proposed an implementation of multi-FPGA communication links for pulse-coupled phase oscillators. In our experiments, two FPGA boards were used to implement a four-oscillator network, and first spike synchronization over two FPGAs required 12.47 µs with a datastream bitrate of 3.2 Gbps. From our experimental results, we verified that the pulse-coupled phase oscillators synchronized over two FPGAs via high-speed serial communication with a 0.1 µs delay.

In future work, we will increase the number of FPGAs and implement large-scale pulse-coupled phase oscillator networks, and then apply them to engineering applications such as image processing for service robots.

Acknowledgments

This research was supported by JSPS KAKENHI Grant Numbers 17H01798, 26330279, and15H01706.

References

1. Y. Kuramoto, *Chemical Oscillation, Waves, and Turbulence* (Springer, Berlin, 1984).
2. A. T. Winfree, *The Geometry of Biological Time* (Springer, New York, 1980).
3. R. Mirollo and S. Strogatz, Synchronization of pulse-coupled biological oscillators, *SIAM J. Appl. Math.* **50**(6) (1990) 1645–1662.
4. Y. Kuramoto, Collective synchronization of pulse-coupled oscillators and excitable units, *Physica D* **50**(1) (1991) 15–30.
5. F. C. Hoppensteadt and E. M. Izhikevich, *Weakly Connected Neural Networks* (Springer, New York, 1997).
6. E. M. Izhikevich, *Dynamical Systems in Neuroscience: The Geometry of Excitability and Bursting* (MIT Press, Cambridge, MA, 2007).
7. D. Atuti, N. Kato, K. Nakada and T. Morie, CMOS circuit implementation of a coupled phase oscillator system using pulse modulation approach, in *European Conference on Circuit Theory and Design (ECCTD)*, (Seville, Spain, 2007), pp. 827–830.
8. K. Matsuzaka, T. Tohara, K. Nakada and T. Morie, Analog CMOS circuit implementation of a pulse-coupled phase oscillator system and observation of synchronization phenomena, *Nonlinear Theory and Its Applications, IEICE* **3**(2) (2012) 180–190.
9. K. Matsuzaka, H. Tanaka, S. Ohkubo and T. Morie, VLSI implementation of coupled MRF model using pulse-coupled phase oscillators, *Electron. Lett.* **51**(1) (2015) 46–48.
10. Y. Suedomi, H. Tamukoh, M. Tanaka, K. Matsuzaka and T. Morie, Parameterized digital hardware design of pulse-coupled phase oscillator model toward spike-based computing, in *20th International Conference on Neural Information Processing (ICONIP2013)*, vol. III, (2013), pp. 17–24.
11. 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.
12. 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.
13. Xilinx Inc, *Aurora 8B/10B Protocol Specification*, (2010), http://www.xilinx.com/support/documentation/ipdocumentation/au-rora8b10bprotocolspecsp002.pdf.
14. A. Athavale and C. Christensen, *High-Speed Serial I/O Made Simple*. Xilinx, Inc., 1.0 edition, (2005), http://www.xilinx.com/publications/archives/book-s/serialio.pdf.
15. D. Pramanta, T. Morie and H. Tamukoh, Implementation of multi-FPGA communication using pulse-coupled phase oscillators, *Proc. 2017 Int. Conf. on Artificial Life And Robotics (ICAROB2017)*, (2017), pp. 128–131.