A SIMPLE DIGITAL SPIKING NEURAL NETWORK:
SYNCHRONIZATION AND SPIKE-TRAIN APPROXIMATION

HIROAKI UCHIDA, YUYA OISHI AND TOSHIMICHI SAITO*

Department of Electrical and Electronic Engineering, Faculty of Science and Engineering
Hosei University
Koganei, Tokyo, 184-8584, Japan

Abstract. This paper studies synchronization phenomena of spike-trains and approximation of target spike-trains in a simple network of digital spiking neurons. Repeating integrate-and-fire behavior between a periodic base signal and constant firing threshold, the neurons can generate various spike-trains. Connecting multiple neurons by cross-firing with delay, the network is constructed. The network can exhibit multi-phase synchronization of various spike-trains. Stability of the synchronization phenomena can be guaranteed theoretically. Applying a simple winner-take-all switching, the network can approximate target spike-trains automatically. In order to evaluate the approximation performance, we present two metrics: spike-position error and spike missing rate. Using the metrics, approximation capability of the network is investigated in typical target signals. Presenting an FPGA based hardware prototype, typical synchronization phenomenon and spike-train approximation are confirmed experimentally.

1. Introduction. This paper studies dynamics and engineering applications of a simple network of digital spiking neurons (DSNs [18] [19] [20]). The DSN is a digital dynamical system inspired by analog spiking neuron models [11] [6] [15] [17]. Repeating integrate-and-fire behavior between a periodic base signal and a constant firing threshold, the DSN can generate various spike-trains. The dynamics of the DSN is integrated into a digital spike map (Dmap). The Dmaps can be regarded as a digital version of analog one dimensional maps represented by the logistic map [9]. As is well known, the logistic map exhibits period doubling bifurcation to chaos. Nonlinear dynamics of the analog maps has been studied extensively [10] [5]. In sharp contrast to the analog maps, the Dmap cannot generate chaos but various periodic orbits because the Dmap is defined on a finite number of points. We have studied classification and stability of periodic orbits in various Dmaps [16] [8] [24]. The Dmap is related not only to the DSN but also to cellular automata [23] [3] [21] [13] and dynamic binary neural networks with signum activation function [12].

The digital dynamical systems such as the DSNs are suitable for precise analysis of various nonlinear phenomena and FPGA based hardware implementation for engineering applications [18] [20]. The practical/potential applications of the DSNs include time series approximation [19], encoders for digital communication.

2020 Mathematics Subject Classification. Primary: 37B15; Secondary: 93C62, 92B20.
Key words and phrases. Neural networks, Digital spiking neurons, Synchronization, Stability.
This work is supported in part by JSPS KAKENHI 18K11480.
* Corresponding author: Toshimichi Saito.
Connecting multiple DSNs in a ladder form, a network of digital spiking neurons is constructed. The connection is realized by cross-firing with delay and the network is referred to as the digital spiking neural network (DSNN). In this paper, we consider two problems.

The first problem is analysis of synchronization phenomena. Before the connection, parameters of each DSN are set to generate a stable periodic spike-train (PST). Depending on the parameters and initial condition, the DSNN can exhibit multi-phase synchronization of various PSTs. We clarify possible class of PSTs that the DSNN can output. After theoretical consideration, we give a condition that guarantees stability of the synchronization. The stability means that some initial states fall directly into the PST.

The second problem is application of the synchronization phenomena to time series approximation by means of spike-trains. In order to generate an appropriate output of the DSNN, we present a simple switching method based on the winner-take-all (WTA) function. The WTA switching activates either neuron temporally and can approximate target spike-trains automatically. In order to evaluate the approximation performance, we present two metrics: spike-position error and spike missing rate. Using the metrics and artificial target signals from a typical probability distribution, we investigate relation between approximation performance and the number of activated DSNs. The experimental results suggest existence of an efficient combination of activated DSNs for spike-train approximation. Presenting an FPGA based hardware prototype, typical synchronization phenomenon and spike-train approximation are confirmed experimentally. The hardware is fundamental in engineering applications and will be developed into digital time series approximation systems as is considered in analog reservoir computing [1] [14].

As main novel contributions, it should be noted that this is the first paper of the DSNN in the following points: the stability condition for multi-phase synchronization phenomena, the WTA switching for automatic spike-train approximation, the two metrics to evaluate spike-train approximation, and the FPGA based hardware of the DSNN.

The rest of this paper is organized as follows. Section 2 introduces DSNs and Dmaps. Section 3 presents the DSNN and gives a condition for stability of synchronization. Section 4 presents the two metrics and investigates spike-train approximation capability. Section 5 presents FPGA based hardware and demonstrates typical waveforms. Section 6 concludes the results and considers outlooks.

2. Digital spiking neuron. We introduce a digital spiking neuron (DSN) and digital spike map (Dmap) [18] [20]. Let \( x_1(\tau) \) denote a discrete state variable at discrete time \( \tau \). Repeating integrate-and-fire behavior between a periodic base signal \( b(\tau) \) and a constant firing threshold \( N_T \), the DSN outputs a spike-train \( y_1(\tau) \). The dynamics is described by

\[
\begin{align*}
\text{Integrating:} & \quad x_1(\tau + 1) = x_1(\tau) + 1, & y_1(\tau) = 0 & \text{if } x_1(\tau) < N_T \\
\text{Self-firing:} & \quad x_1(\tau + 1) = b(\tau), & y_1(\tau) = 1 & \text{if } x_1(\tau) = N_T
\end{align*}
\]

where \( x_1(\tau) \in \{0, 1, \cdots, N_T\} \) and the base signal is period \( T_p \): \( b(\tau + T_p) = b(\tau) \).

In order to simplify theoretical analysis, we assume the following condition:

\[
\tau - 2T_p + 1 \leq b(\tau) - N_T \leq \tau - T_p \text{ for } \tau \in \{0, \cdots, T_p - 1\}, \ N_T \leq 2T_p - 1.
\]
In this case, the DSN outputs one spike per one base period $T_p$:

$$y_1(\tau) = \begin{cases} 1 & \text{for } \tau = \tau_n \\ 0 & \text{for } \tau \neq \tau_n \end{cases} \quad \tau_n \in I_n = [(n-1)T_p, nT_p)$$

where $\tau_n$ denotes the $n$-th spike-position and the initial spike is given in the first period $\tau_1 \in [0, T_p)$.

Detailed discussion of the condition in Eq. (2) can be found in Refs. [19] [4]. Stability analysis in Section 3 is extremely hard without this condition. Let $\theta_n = \tau_n \mod T_p$ ($\theta_n \in \{1, \cdots, T_p\}$) be the $n$-th spike-phase. A spike-position is given by $\tau_n = \theta_n + T_p(n-1)$ and a spike-train $y_1(\tau)$ is governed by a digital spike map (Dmap):

$$\theta_{n+1} = F(\theta_n) = f(\theta_n) \mod T_p, \quad f(\theta_n) = \theta_n - b(\theta_n) + N_T + 1$$

The Dmap is represented by a characteristic vector of integers:

$$\delta \equiv (\delta_1, \cdots, \delta_{T_p}), \quad F(i) = \delta_i, \quad \delta_i \in \{1, \cdots, T_p\}, \quad i \in \{1, \cdots, T_p\}$$

Fig. 1 shows two examples of Dmaps. Since the Dmap is defined on a finite number of points ($T_p$ is finite), steady state must be a periodic orbit. Using the Dmap $F$, we give definitions of periodic orbits and stability.

A point $\theta_p \in \{1, \cdots, T_p\}$ is said to be a periodic point with period $p$ if $F^p(\theta_p) = \theta_p$ and $F(\theta_p)$ to $F^p(\theta_p)$ are all different where $F^p$ denotes the $p$-fold composition of $F$. A sequence of periodic points $\{F(\theta_p), \cdots, F^p(\theta_p)\}$ is said to be a periodic orbit (PEO). A PEO with period $p$ is equivalent to a PST with period $pT_p$. 

---

**Figure 1.** DSNs and Dmaps for $T_p = 18$. (a) A PST with period $6T_p$ and corresponding PEO with period 6. $\delta = (16, 17, 18, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15)$. (b) A direct stable PST with period $6T_p$ and corresponding PEO with period 6. $\delta = (17, 17, 17, 2, 2, 2, 5, 5, 5, 8, 8, 8, 11, 11, 11, 14, 14, 14)$. 


A PEO and corresponding PST are said to be stable if some initial states fall directly into the PEO: \( F(\theta) = \theta_p \) for some \( \theta_p \), some \( \theta \neq \theta_p \), and some \( l \).

A PEO and corresponding PST are said to be direct stable if all the initial states fall directly into the PEO: \( F(\theta) = \theta_p \) for some \( \theta_p \) and all \( \theta \). Fig. 1 (b) shows a direct stable PEO and corresponding direct stable PST.

As mentioned in Section 1, the Dmap can be regarded as a digital version of analog one-dimensional map. In the study of the analog maps, important study subjects include stability of periodic orbits, basin of attraction, and invariant measure of chaos [10] [5]. In the Dmaps, a periodic orbit is equivalent to a sequence of binary vectors and stability of periodic orbits corresponds to bit error correction. More detailed discussions of Dmaps can be found in [16] [8] [24].

3. Digital spiking neural networks. We present a digital spiking neural network (DSNN) and consider multi-phase synchronization of PSTs. Connecting \( M \) pieces of DSNs in a ladder form, a DSNN is constructed as shown in Fig. 2. The dynamics is described by

\[
\begin{align*}
\text{Integrating:} & \quad x_i(\tau + 1) = x_i(\tau) + 1, \quad y_i(\tau) = 0 \quad \text{if} \quad x_i(\tau) < N_T \\
\text{Self-firing:} & \quad x_i(\tau + 1) = b(\tau), \quad y_i(\tau) = 1 \quad \text{if} \quad x_i(\tau) = N_T \\
\text{Cross-firing:} & \quad x_{j+1}(\tau + 1) = N_C + 1, \quad \text{if} \quad z_j(\tau) = 1
\end{align*}
\]

Figure 2. Digital spiking neural network (DSNN) in a ladder form.

Figure 3. Cross-firing.
Figure 4. 6-phase synchronization of PSTs with period 6 in DSNN.

Connection signal: \( z_i(\tau) = \begin{cases} 1 & \text{if } x_i(\tau) = N_T \text{ and } x_{i+1}(\tau) \leq N_C \\ 0 & \text{otherwise} \end{cases} \) (7)

where \( N_C \equiv N_T - T_p \) is a connection threshold, \( i \in \{1, \cdots, M\} \), and \( j \in \{1, \cdots, M-1\} \). The periodic base signal \( b(\tau) \) is common for all the DSNs. The integrating and self-firing are the same as the single DSN in Eq. (1). The cross-firing is characterized by the connection signal \( z_i \). The 1st DSN is independent and can output a connection signal \( z_1 \) to the 2nd DSN if \( x_1 \) reaches the firing thresholds \( N_T \) and the 2nd state variable \( x_2 \) does not exceeds the connection thresholds \( N_C \) as shown in Fig. 3. The 2nd DSN can output a connection signal to the 3rd DSN, and so on. This master-slave connection propagates from the 1st DSN to the last \( M \)-th DSN. Let \( T = MT_p \) be basic period of the DSNN. Let one basic period interval \( I \equiv [0, T) \) be divided into \( M \) subintervals

\( I_1 = [0, T_p), I_2 = [T_p, 2T_p), \cdots, I_M = [(M-1)T_p, MT_p) \).

The output spike-train \( y \) of the DSNN is given by the binary valued selection matrix \( W = (w_{ij}) \):

\[
y(\tau) = \sum_{i=1}^{M} w_{ij} y_i(\tau) \text{ for } \tau \in I_j, \ j \in \{1, \cdots, M\}, \ y(\tau + T) = y(\tau) \quad (8)
\]

The connection matrix \( W \) determines operation of selection switches \( S_i \) such that \( w_{ij} = 1 \) and \( w_{ij} = 0 \) mean \( S_i \) = on and \( S_i \) = off for \( \tau \in I_j \), respectively. For simplicity, we assume the following two conditions in the DSNN:

1. Each DSN generates a stable PST with period \( MT_p \) before the connection.
2. The DSNN exhibits \( M \)-phase synchronization of the PST defined by Eq. (9).

\[
x_i(\tau) = x_i(\tau + MT_p), \ y_i(\tau) = y_i(\tau + MT_p), \ i \in \{1, \cdots, M\} \\
x_i(\tau) = x_{i+1}(\tau + T_p), \ y_i(\tau) = y_{i+1}(\tau + T_p) \quad (9)
\]

Fig. 4 shows spike-trains of the 6-phase synchronization of PSTs with period 6. The two conditions guarantee that, adjusting \( W \), the DSNN can output a PST \( y \) consisting of any combination of \( M \) spike-phases in \( S_p \) where each subinterval includes at most \( M \) spikes. Fig. 4 shows an example of output \( y \) for the following
Figure 5. Stability of master-slave synchronization.

6-dimensional selection matrix.

\[ W = \begin{pmatrix}
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 0 \\
0 & 1 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 
\end{pmatrix} \]  

(10)

Here we consider stability of the \( M \)-phase synchronization. Referring to the definition of stability for PST, an \( M \)-phase synchronization is said to be stable if there exist some initial states falling into the synchronization. First, note again that the DSNN is based on master-slave connection of cross-firing that propagates from the 1st DSN to the \( M \)-th DSN. Let each DSN output a PST characterized by \( M \) spike phases \( \{\theta'_1, \ldots, \theta'_M\} \equiv S_p \) before the connection. In this case, the Dmap exhibits periodic orbit with period \( M \) as shown in Fig. 1 where the periodic orbit consists of 6 spike-phases \( S_p = \{\theta'_1, \ldots, \theta'_6\} \equiv \{\theta_a, \ldots, \theta_f\} \). The order of the spike-phases depends on an initial condition. If the master-slave synchronization from the 1st DSN to the 2nd DSN is stable then the stable synchronization propagates successively and reaches the last DSN. Hence it is sufficient to consider the synchronization from the 1st DSN to the 2nd DSN.

Second, we assume existence of two spike phases such that \( \theta_b \equiv F(\theta_a) > \theta_a + 1 \), \( \theta_a \in S_p \), and \( \theta_b \in S_p \) as shown in Fig. 5. In the case where connection signal does not exist, we assume that the 2nd DSN (slave) outputs the first spike at time \( \tau_1 = \theta_a \). The state variable \( x_2 \) is reset to the base at time \( \tau_1 + 1 \). After that \( x_2 \) reaches the connection threshold \( N_C \) at time \( \tau_1 + (\theta_b - \theta_a) = \theta_b \) and reaches the firing threshold \( N_T \) at time \( \tau_2 = \theta_b + T_p \) when the 2nd DSN outputs the second spike. Since \( \theta_b > \theta_a + 1 \), we notice

\[ x_2 \leq N_C = N_T - T_p \quad \text{for} \quad \tau \in \{\theta_a + 1, \ldots, \theta_b\} \equiv I_S. \]  

(11)

where \( I_S \) is referred to as a synchronization interval. That is, \( x_2 \) does not exceed the connection threshold \( N_C \) for \( \tau \in I_S \).

Third, we assume that the 1st DSN outputs the first spike at time \( \theta_c \) as shown in Fig. 5. If \( \theta_c \in I_S \) then the 1st DSN outputs the connection signal to the 2nd
DSN at time $\theta_c$ when and $x_2$ resets to the connection threshold $N_C$

$$x_2(\theta_c + 1) = N_C + 1, \ z_1(\theta_c) = 1, \ y_1(\theta_c) = 1. \quad (12)$$

After that $x_2$ rises, reaches the firing threshold, and outputs a spike

$$x_2(\theta_c + T_p) = N_T, \ y_2(\theta_c + T_p) = 1. \quad (13)$$

After that the 2nd DSN synchronizes the 1st DSN with time delay $T_p$: the master-slave synchronization is achieved. Note that, for $T_p < \tau \leq \theta_c + T_p$, $x_2 > N_C$ is satisfied and no connection signal come from the 1st DSN. In the next period of the PST, $\theta_c + T = \theta_b + T$ is fulfilled and the master-slave synchronization from the 1st to the 2nd DSN is held by the connection signal $z_1(\theta_b + T) = 1$. If analogous situation is satisfied for the $i$-th and the $(i+1)$-th DSNs, $i \in \{2, \cdots, M-1\}$, then the $M$-phase synchronization is achieved. In summary, the $M$-phase synchronization is achieved and is stable if each master-slave pair outputs the following spikes if the connection signal does not exist:

The $(i+1)$-st DSN (slave) : $y_{i+1}(\tau_m) = 1, \ z_{i+1}(\tau_m+1) = 1, \ \tau_{m+1} - \tau_m > T_p + 1$

$$(\tau_m \mod T_p \equiv \theta_a, \ \tau_{m+1} \mod T_p \equiv \theta_b, \ \theta_b > \theta_a + 1)$$

The $i$-th DSN (master) : $y_i(\tau_m') = 1, \ \tau_m < \tau_m' < \tau_{m+1} - T_p$

$$(\tau_m' \mod T_p \equiv \theta_a, \ \theta_a < \theta_c < \theta_b)$$

where $m$ is some integer, $\tau_m$ is the $m$-th spike position of the slave DSN, and $\tau_m'$ is the $m$-th spike position of the master DSN. $\theta_a, \theta_b$, and $\theta_c$ are elements in $S_p$. For the PEO in the Dmap of Fig. 1, the interval $I_S = [\theta_a, \theta_b]$ is wide and all the other spike-phases are included in $I_S$. Such a PST is suitable to realize stable $M$-phase synchronization.

4. **Spike-train approximation.** Here we consider application to time-series approximation by means of spike-trains. In this application, a target time series is periodic and is represented by a PST

$$y_t(\tau) = \begin{cases} 1 & \text{for } \tau = \tau'_k \\ 0 & \text{otherwise} \end{cases} \quad k \in \{1, \cdots, Q\}, \ 0 < \tau'_1 < \cdots < \tau'_Q < T \quad (15)$$

where $T$ is a period of the PST ($y_t(\tau + T) = y_t(\tau)$) and $\tau'_n$ denotes the $n$-th spike-position. The target PST consists of $Q$ spikes. For convenience, the target PST is represented by the spike position sequence.

$$P = (\tau'_1, \tau'_2, \cdots, \tau'_Q). \quad (16)$$

In order to approximate the target PST, we prepare a DSNN discussed in Section 3: the DSNN exhibits stable $M$-phase synchronization of a PST with period $T = MT_p$ as shown in Eq. (9). The PST consists of $M$ spike phases as shown in Fig. 4. In this case, the DSNN can output a PST with period $T = MT_p$ consisting of any combination of the $M$ spike-phases. Such an output is impossible without the $M$-phase synchronization. The problem is automatic tuning of the selection matrix $W$ for approximation of a target PST. Since full search space of $W$ is $2^{M^2}$, the full search becomes impossible as $M$ increases. In order to tune the $W$ automatically, we present a simple switching method based on the winner-take-all function (WTA):

$$w_{ij} = \begin{cases} 1 & \text{if } y_i(\tau) = 1 \text{ and } x_i(\tau) \text{ is the maximum in } \{x_1, \cdots, x_M\} \text{ at } \tau \in I_j \\ 0 & \text{otherwise} \end{cases} \quad (17)$$
$w_{ij} = 0$ at time $\tau = 0$, $w_{ij}$ is updated for $0 \leq \tau < T = MT_p$, and $w_{ij}$ is fixed after time $T$. If $x_i$ is the maximum when $y_t(\tau'_k) = 1$ ($\tau'_k \in I_j$) then the $i$-th DSN becomes a winner and is activated ($w_{ij} = 1$) on subinterval $I_j$. Fig. 6 illustrates the WTA switching for $M = 6$. In the circuit, the $i$-th selection switch $S_i$ turns on for $\tau \in I_j$. Note that multiple DSNs can be activated on one subinterval. Using the WTA switching, the DSNN can approximate a target spike-train automatically.

In order to evaluate spike-train approximation, we present two basic metrics. The first one is the spike-position error

$$\varepsilon_p(y_t) = \frac{1}{Q} \sum_{k=1}^{Q} |\tau'_{k} - (\tau_{k} - \Delta)|, \quad \Delta = \tau_1 - \tau'_1$$

(18)

where $\tau'_{k}$ and $\tau_{k}$ are the $k$-th spike position of target PST $y_t$ and approximated PST $y_a$, respectively. If the WTA switching misses some target spike(s) then the spike position error $\varepsilon_p$ is calculated after removing the missing spike(s). The second one is the spike missing rate

$$\text{SMR} = \frac{\text{the number of missing spikes}}{Q} \times 100[\%]$$

(19)
If no spike is missed (SMR=0) then the $\varepsilon_p$ measures the average of approximation error. If several spikes are missed (SMR > 0) then the approximation performance is evaluated by combination of $\varepsilon_p$ and SMR $^1$. The evaluation becomes extremely hard without the SMR: we must present a translation method of the missing spikes into the approximation error.

$^1$The two metrics cannot measure deviation of the error distribution and time-dependency of the error. Improvement is necessary for more detailed evaluation.

**Figure 7.** Dmap for each DSN in the DSNN for $T_p = 36$ and $M = 12$. $\delta = (35, 35, 35, 2, 2, 2, 5, 5, 8, 8, 11, 11, 11, 14, 14, 17, 17, 17, 20, 20, 20, 20, 23, 23, 23, 26, 26, 26, 26, 29, 29, 29, 32, 32, 32)$. The PEO with period 12 corresponds to the PST with period $12T_p$.

**Figure 8.** Target PST $y_t$, approximated PST $y_a$, and 12-phase synchronization of PSTs $y_1 \sim y_{12}$ with period 12. Verilog simulation of DSNN for $M = 12$. 
We try to investigate spike-train approximation capability in numerical experiments. Before the connection, each DSN is set to generate a direct stable PST with period \(12T_p\) \((M = 12)\) from the Dmap in Fig. 7. This Dmap generates a direct stable PEO period 12 and the PST has long synchronization interval \(I_S = \{2, \cdots, 35\}\). The DSNN exhibits stable 12-phase synchronization of the PSTs with period 12 as shown in Fig. 8. Applying the cross-firing to \(M = 12\) DSNs and applying the WTA switching, the DSNN is constructed and can approximate various target PSTs automatically.

In order to generate a target PST with period \(T = 12T_p\), we define target inter-spike-interval (ISI) sequence

\[
D = (d_1, d_2, \cdots, d_Q), \quad d_l = \tau_{l+1} - \tau_l, \ l \in \{1, \cdots, Q-1\}
\]  

(20)

where \(d_l\) is the \(l\)-th ISI. The target ISI sequence is given from discrete exponential distribution \(f(d)\) as shown in Fig. 9 where \(d\) is a random variable corresponding to ISI. If the firing frequency of spikes per each subinterval \(I_n\) follows the Poisson distribution then the corresponding ISI follows the exponential distribution, as is well known. In the exponential distribution, we have fixed parameters as \(d_{min} = 7\) and \(\lambda = 0.25\). In this case, at most 5 spikes can appear in each subinterval. An example of target PST is shown in Fig. 8. Applying the WTA switching to the PST, we have obtained the following selection matrix.

\[
W = \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
\end{pmatrix}
\]  

(21)

Performing the Verilog based numerical calculation, we have confirmed that the DSNN outputs an approximated PST as shown in Fig. 8. The Verilog is a utility software to realize an FPGA based hardware and the algorithm outline is shown with an FPGA circuit in Section 5.

![Discrete exponential distribution for target PSTs.](image)
Using the spike position error $\varepsilon_p(y_t)$ and the spike missing rate SMR, this approximation is evaluated by $\varepsilon_p(y_t) = 1.97$ and SMR = 0. Next, we investigate relation between approximation performance and the number of activated DSNs in the following two cases:

**Case 1:** The number of activated DSN is reduced in the order of activation frequency by the WTA switching. For example, if the $j$-th DSN has the lowest activation frequency then the $j$-th DSN is inactivated ($S_j$=off all the time) and $y_j$ is removed from the DSNN output.

**Case 2:** The number of activated DSN is reduced randomly.

We have executed 10 trials and the results are evaluated by three feature quantities: average of the spike position error (AVG $\varepsilon_p$), standard deviation of the spike position error (SD $\varepsilon_p$), and the spike missing rate (SMR). The results are summarized in Tables 1 and 2. In the tables, we can see the following. As the number of
HIROAKI UCHIDA, YUYA OISHI AND TOSHIMICHI SAITO

activated DSNs (#DSN) decreases, AVG and SD of spike-position error $\varepsilon_p$ increase both in Case 1 and Case 2 as expected. The Case 1 realizes smaller value of the metrics than the Case 2: the activation frequency is basic factor to determine the number of activated DSNs. In the spike missing rate, the Case 1 realizes smaller values in almost all cases. Here we consider the spike missing rate. Since the number of DSN is $M$, at most $M$ DSNs can be activated and the DSNN can outputs up to $M$ spikes in each subinterval. The DSNN has potential to approximate a target PST including at most $M$ spikes in each subinterval. However, spike missing occurs for $M = 7$ in Case 1 and for $M = 8$ in Case 2 even if the target PST includes at most 5 spikes in each subinterval. The spike missing occurs depending not only on the target PST but also on combination of activated DSNs. Fig. 10 illustrates the spike missing in a DSNN of 6 DSNs ($M = 6$). In Fig. 10 (a) the 1st, the 2nd, and the 3rd DSNs are activated and 6 missing intervals exist. The missing interval means an interval between some target spike position and the closest backward spike position of DSNs. If a target spike arrives in some missing interval (the 4th missing interval in the figure), the target spike is missed. On the other hand, in Fig. 10 (b), the 1st, the 3rd, and the 5th DSNs are activated and no spike arrives in either of 6 missing intervals: spike missing does not occur. That is, spike missing depends on combination of activated DSNs. Although the WTA switching is simple and can activate appropriate DSNs automatically, we should select suitable combination of activated DSNs to avoid spike missing.

5. FPGA based hardware implementation. As stated earlier, the DSNN is a digital dynamical system and is suitable for FPGA-based hardware implementation. For simplicity, we try to design the DSNN for $M = 12$, $T_p = 36$, and $N_T = 71$ as is Fig. 8. Algorithm 1 and Algorithm 2 show outline of the Verilog source code for the DSNN body and WTA switching for spike-train approximation, respectively. The Verilog has ability to design digital circuits from mathematical models and the algorithms are obtained from Eqs. (6) and (7). Fig. 11 shows the outline of basic circuit design of the DSNN body and WTA switching. In the implementation, we have used the following tools:

- Verilog version: Vivado Design Suite 2019.1
- FPGA board: DIGILENT BASYS3, Clock frequency: 10.0[MHz].
- Measuring instrument: ANALOG DISCOVERY2, Multi-instrument software: WaveForms 2015

| #DSN | AVG $\varepsilon_p$ | SD $\varepsilon_p$ | SMR |
|------|---------------------|---------------------|-----|
| 3    | 6.60                | 1.35                | 30.3|
| 4    | 5.17                | 0.74                | 17.5|
| 5    | 4.19                | 0.43                | 10.1|
| 6    | 3.56                | 0.31                | 5.10 |
| 7    | 3.02                | 0.30                | 0.82 |
| 8    | 2.63                | 0.24                | 0    |
| 9    | 2.43                | 0.19                | 0    |
| 10   | 2.25                | 0.18                | 0    |
| 11   | 2.11                | 0.14                | 0    |
| 12   | 2.02                | 0.10                | 0    |

| #DSN | AVG $\varepsilon_p$ | SD $\varepsilon_p$ | SMR |
|------|---------------------|---------------------|-----|
| 3    | 8.91                | 1.84                | 44.5|
| 4    | 6.22                | 0.90                | 21.7|
| 5    | 5.12                | 0.74                | 10.8|
| 6    | 4.18                | 0.46                | 2.78 |
| 7    | 3.62                | 0.44                | 1.09 |
| 8    | 3.22                | 0.30                | 0.27 |
| 9    | 2.91                | 0.29                | 0    |
| 10   | 2.59                | 0.22                | 0    |
| 11   | 2.28                | 0.14                | 0    |
| 12   | 2.03                | 0.11                | 0    |
Algorithm 1 DSNN

module dsnn(outputy, inputCLK)
wire y_1 - y_M, z_1 - z_{M-1}
reg X_1 - X_M
Base = initial condition and realize base signal
//judge each x and output connection signal
connection signal(y_1(y_1), ..., y_M(y_M), x_1(x_1), ..., X_M(X_M),
z_1(z_1), ..., z_{M-1}(z_{M-1}))
assign z_i = X_{i+1}[N_c : 0] \times y_i
//Instantiate the output layer
output layer(y_1(y_1), ..., y_M(y_M), y(y), W(W))
for (i = 1; i <= M; i++) do
assign y'_i = y_i \times W
end for
reg y = |[y'_1 : y'_M]|
always @(posedgeCLK)
X_i <= (X_i << 1)
if (X_i(N_x) = 1) then
X_i is reset to the base signal
end if
if (z_i == 1) then
X_{i+1} jumps to \(N_x - T_p + 1\)
end if
end always
end module

Algorithm 2 WTA

module WTA(outputy, inputy, y_1, ..., y_M)
//input target spike-train y_t
//input target spike-train(y_t)
reg W
//wta-switching
always @(posedgey_t)
max = X_1
for (t = 2; t <= M; t++) do
if (max < X_t) then
max == X_t
end if
end for
for (t = 1; t <= M; t++) do
if (max == X_t) then
w_t <= 1
end if
end for
end always
determine W
end module
Using the Verilog, we have implemented the circuit on the FPGA board and have confirmed the DSNN operation experimentally. Fig. 12 shows typical observations corresponding to numerical simulation in Fig. 8: target spike-train, approximated spike-train, and multi-phase synchronization of PSTs for $M = 12$. This hardware implementation is fundamental to consider engineering applications.

**Figure 11.** Circuit design of DSNN with the WTA.

**Figure 12.** Target spike-train, approximated spike-train and multi-phase synchronization in an FPGA board.
6. **Conclusions.** The DSNN with the WTA switching has been presented and multi-phase synchronization of PSTs has been considered in this paper. The DSNN is based on master-slave connection of cross-firing with delay. Stability of the multi-phase synchronization is guaranteed theoretically if a condition is satisfied. The multi-phase synchronization is applicable to spike-train approximation and the WTA switching can realize the approximation automatically. Presenting two metrics (spike-position error and spike missing rate), relation between application capability and the number of activated DSNs has been investigated. Presenting an FPGA based hardware, typical multi-phase synchronization and spike-train approximation have been confirmed experimentally.

In our future works, we should consider many problems including synchronization of PSTs in DSNNs of various topology (e.g., ring, bi-directional ring, star), stability of the synchronization phenomena in the various DSNNs, and engineering application to spike-based time series approximation/prediction.

**REFERENCES**

[1] L. Appeltant, M. C. Soriano, G. Van der Sande, J. Danckaert, S. Massar, J. Dambre, B. Schrauwen, C. R. Mirasso and I. Fischer, *Information processing using a single dynamical node as complex system*, Nat. Commun., 2 (2011), Article number: 468.

[2] S. R. Campbell, D. L. Wang and C. Jayaprakash, *Synchrony and desynchrony in integrate-and-fire oscillators*, Neural Comput., 11 (1999), 1595–1619.

[3] L. O. Chua, *A Nonlinear Dynamics Perspective of Wolfram’s New Kind of Science I*, World Scientific, 2006.

[4] T. Iguchi, A. Hirata and H. Torikai, *Theoretical and heuristic synthesis of digital spiking neurons for spike-pattern-division multiplexing*, IEICE Trans. Fundam., E93-A (2010), 1486–1496.

[5] A. Lasota and M. C. Mackey, *Chaos, Fractals, and Noise*, second edition, Springer-Verlag, 1994.

[6] G. Lee and N. H. Farhat, *The bifurcating neuron network 1*, Neural networks, 14 (2001), 115–131.

[7] A. Lozano, M. Rodriguez and R. Barrio, *Control strategies of 3-cell central pattern generator via global stimuli*, Sci. Rep., 6 (2016), 23622.

[8] A. Matoba, N. Horimoto and T. Saito, *Basic dynamics of the digital logistic map*, IEICE Trans. Fundam., E96-A (2013), 1808–1811.

[9] R. May, *Simple mathematical models with very complicated dynamics*, Nature, 261 (1976), 459–467.

[10] E. Ott, *Chaos in Dynamical Systems*, Second edition. Cambridge University Press, Cambridge, 2002.

[11] R. Perez and L. Glass, *Bistability, period doubling bifurcations and chaos in a periodically forced oscillator*, Phys. Lett. A, 90 (1982), 441–443.

[12] R. Sato and T. Saito, *Stabilization of desired periodic orbits in dynamic binary neural networks*, Neurocomputing, 248 (2017), 19–27.

[13] M. Schüle and R. Stoop, *A full computation-relevant topological dynamics classification of elementary cellular automata*, Chaos, 22 (2012), 043143, 10pp.

[14] G. Tanaka, T. Yamane, J. B. Héroux, R. Nakane, N. Kanazawa, S. Takeda, H. Numata, D. Nakano and A. Hirose, *Recent advances in physical reservoir computing: A review*, Neural Networks, 115 (2019), 100–123.

[15] H. Torikai and T. Saito, *Resonance phenomenon of interspike intervals from a spiking oscillator with two periodic inputs*, IEEE Trans. Circuits Syst. I, 48 (2001), 1198-1204.

[16] H. Torikai and T. Saito, *Analysis of a quantized chaotic system*, Int’l J. of Bifurcation and Chaos, 12 (2002), 1207–1218.

[17] H. Torikai and T. Saito, *Synchronization phenomena in pulse-coupled networks driven by spike-train inputs*, IEEE Trans. Neural Networks, 15 (2004), 337–347.

[18] H. Torikai, H. Hamanaka and T. Saito, *Reconfigurable spiking neuron and its pulse-coupled networks: basic characteristics and potential applications*, IEEE Trans. Circuits Syst. II, 53 (2006), 734–738.
[19] H. Torikai, A. Funew and T. Saito, Digital spiking neuron and its learning for approximation of various spike-trains, Neural Networks, 21 (2008), 140–149.

[20] H. Uchida and T. Saito, Multi-phase synchronization phenomena in a ring-coupled system of digital spiking neurons, IEICE Trans. Fundam., E102-A (2019), 235–241.

[21] W. Wada, J. Kuroiwa and S. Nara, Completely reproducible description of digital sound data with cellular automata, Phys. Lett. A, 306 (2002), 110–115.

[22] D. L. Wang and D. Terman, Locally excitatory globally inhibitory oscillator networks, IEEE Trans. Neural Networks, 6 (1995), 283–286.

[23] S. Wolfram, A New Kind of Science, Wolfram Media, 2002.

[24] H. Yamaoka and T. Saito, Steady-versus-transient plot for analysis of digital maps, IEICE Trans. Fundam., E99-A (2016), 1806–1812.

Received October 2019; revised March 2020.

E-mail address: atsuken5926@gmail.com
E-mail address: yuya.oishi.0411@gmail.com
E-mail address: tsaito@hosei.ac.jp