PAPER

Analog MOS Circuits Implementing a Temporal Coding Neural Model

Gessyca Maria Tovar, Tetsuya Asai, Daichi Fujita and Yoshihito Amemiya

Graduate School of Information Science and Technology, Hokkaido University
Kita 14, Nishi 9, Kita-ku, Sapporo 060-0814, Japan
E-mail: asai@ist.hokudai.ac.jp

Abstract **One of the most important processes of the brain is learning and recalling information; the memory function. Because the world in which we live is continuously changing, it is essential for intelligent systems to encode, recognize and generate temporal patterns. Therefore, temporal information processing has great significance in bio-inspired memory systems. A possible model for the storage of temporal sequences was proposed in [1]. On the basis of this model, we propose a neural model capable of learning and recalling temporal sequences. The model is designed to be suitable for implementation in analog metal-oxide-semiconductor (MOS) circuits. In this paper, we numerically confirmed basic operations of the model. Moreover, we demonstrated fundamental circuit operations and confirmed operations of the circuit network consisting of 20 neurons using a simulation program with integrated circuit emphasis (SPICE).**

Keywords: neural networks, analog MOS circuits, temporal coding, oscillators

1. Introduction

The brain has the ability to process information that changes over time. Therefore, it is necessary that systems, whether natural or artificial, have the ability to process information that depends on the temporal order of events. Studies on neuroimaging have provided evidence that the prefrontal cortex of the brain is involved in temporal sequencing [2]. Furthermore, studies on the olfactory bulb have shown that information in biological networks takes the form of space-time neural activity patterns [3], [4].

Patterns whose content depends on time are commonly called *temporal sequence*. The processing of temporal sequences has been a long-standing problem in artificial neural networks. To process such kind of sequences, a short-term memory is needed to extract and store the temporally ordered sequences, and another mechanism is needed to retrieve them. Neural networks for processing temporal sequences are usually based on the multilayer perceptron or on the Hopfield models [5]. In [6], a network for processing temporal sequences has been proposed and applied to robotics. Making use of the Hebbian rule, the model is able to learn and recall multiple trajectories with the help of time-varying information. In addition, spatio-temporal sequence processing have been employed in neuromorphic VLSIs to mimic early visual processing [7] and associative memory functions [8].

In this paper, we focus on the implementation of such kind of temporal-coding neural networks in analog metal-oxide-semiconductor (MOS) devices. In [1], Fukai proposed a model for the storage of temporal sequences. In the model, the Walsh series expansion [9] was used to represent the input signal by linear superposition of rectangular periodic functions with different fundamental frequencies generated by an oscillatory subsystem. Often, the development of mathematical models for simulating large-scale neural networks suffers from problems of computer load (simulation time). This is avoided by using analog MOS circuits, which permit real-time emulation of large-scale networks because, in contrast to discrete step processing (carried out in computer simulations), one can design analog neural circuits in a parallel manner if a parallel computing structure, based on the construction of the brain, is known. Therefore, based on Fukai's model we propose a modified neural model that is suitable for implementation with analog MOS circuits and is capable of learning and recalling temporal sequences. The model consists of neural oscillators coupled to a common output cell through positive or negative synaptic connections. The weights of the synaptic connections are strengthened (or weakened) when the output of the oscillatory cells overlap (or do not overlap) with the input sequence.

This paper is organized as follows: Section 2 explains the structure and operations of our temporal coding model including the numerical simulation results. Section 3 presents MOS circuit implementation of the model. In section 4 we demonstrate the operation of the circuit network using a simulation program with integrated circuit emphasis (SPICE), and we conclude our work in section 5.

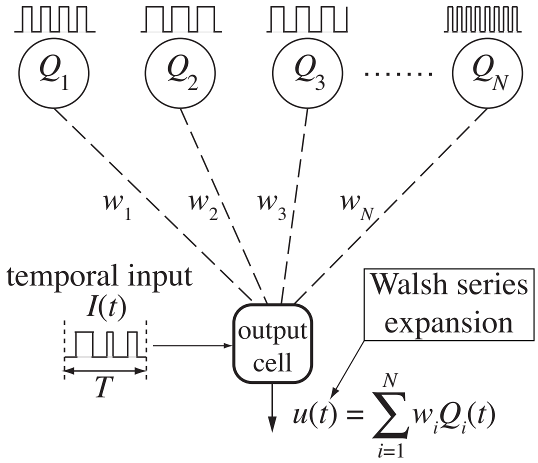

Fig. 1 Proposed temporal coding model

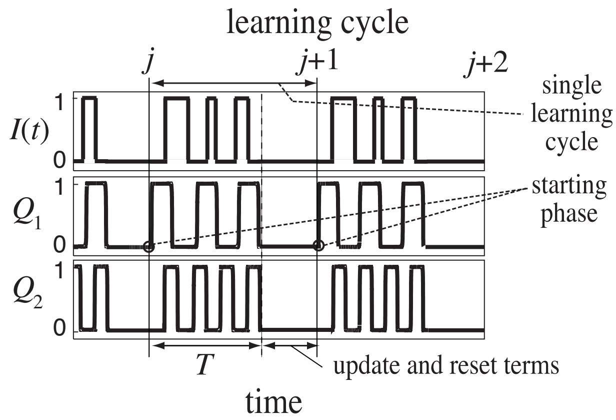

Fig. 2 Definition of single learning cycle

2. Temporal Coding Model for Analog MOS Circuits

Fukai proposed a model for the storage of temporal sequences in [1]. The main purpose of this model is the learning and the recalling of the temporal input stimuli. The model consists of an input unit that triggers the oscillatory subsystem. The oscillatory subsystem has N oscillatory subunits and an array of modifier cells. Each subunit consists of a pair of excitatory and inhibitory neural cells based on the Wilson-Cowan system [10] and generates oscillatory activity with various rhythms and phases. These oscillatory cells are connected through synaptic connections to an array of modifier cells which transforms the oscillatory activity into rectangular patterns, and controls their rhythms and phases. The outputs of the modifier cells are connected to an output cell, which is trained independently of the activity of modifier cells, through synaptic connections. The output cell sums up all the outputs of the modifier cells to recall the input signal in accordance with the Walsh function series [9].

Based on the Fukai's model, we propose a modified model for learning and recalling temporal sequences that is suitable for implementation with analog MOS circuits. The modified model is shown in Fig. 1. One of the characteristics of Fukai's model is the use of modifier cells. The modifier cells change the activities of the oscillatory cells into rectangular patterns, *i.e.*, the cells generate square-wave oscillations. In addition, threshold values of the modifier cells are modified to improve the accuracy of the input-output approximation after each learning cycle [1]. In our modified model, we eliminated these modifier cells. Instead we used neural oscillators that exhibit periodic square-wave oscillations. Therefore, the modification of thresholds in modifier cells is not carried out, which results in reducing accuracy of the learning in our model.

The function of the model is to learn (record) temporal input sequence $I(t)$ $(\in 0, 1)$ of length T and to recall it as recorded sequence $u(t)$. The model consists of N neural oscillators whose outputs $Q_i(t)$ $(\in 0, 1;\ i = 1, ..., N)$ are time-varying periodic square waves with different fundamental frequencies. Each of the oscillators is connected to an output cell through synaptic connections whose weights are denoted by w_i $(i = 1, ..., N)$. The output cell calculates the weighted sum of the oscillator outputs as

$$u(t) = \sum_{i=1}^{N} w_i Q_i(t) \tag{1}$$

Through cyclic learning processes, w_is in Eq. (1) are updated at every cycle to achieve $u(t) \to I(t)$. Note that this expression, *i.e.*, a weighted sum of square-wave functions with various fundamental frequencies, corresponds to a form of the Walsh series expansion [9], which is a mathematical method to approximate a certain class of functions, like the Fourier series expansion.

Now, given a periodic input signal ($I(t)$) with period T and the output ($u(t)$), we define the mean square error (E) between them as

$$E = \frac{1}{2T} \int_{jT}^{(j+1)T} [I(t) - u(t)]^2 \, dt \ \ (j = 0, 1, 2, \cdots) \tag{2}$$

where j represents the learning cycle. To learn the input signal ($I(t)$) correctly, we need to minimize this error. This is achieved by modifying the weights (w_i) between the oscillators and the output cell according to the gradient descent rule:

$$\delta w_i = -\eta \partial E / \partial w_i \tag{3}$$

where η represents a small positive constant indicating the

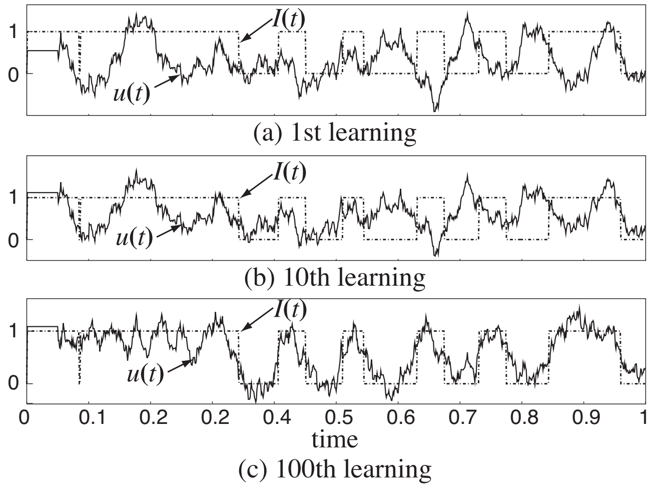

Fig. 3 Input ($I(t)$) and output sequences ($u(t)$) of proposed network with 200 oscillatory units after first (a) 10th (b) and 100th learning (c)

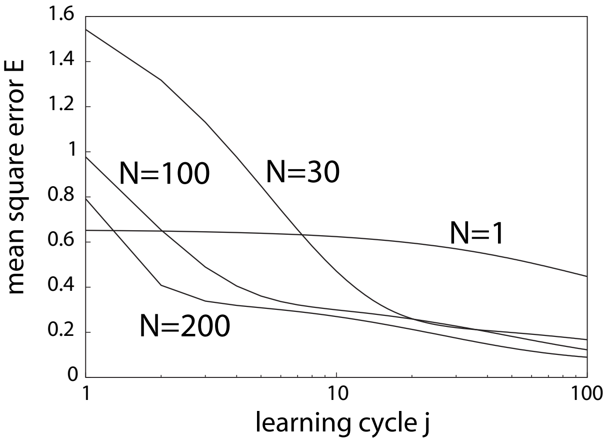

Fig. 4 Time evolution of mean square errors

learning rate. Substituting E in Eq. (2) into Eq. (3), we obtain

$$\delta w_i = \frac{\eta}{T} \int_{jT}^{(j+1)T} [I(t) - u(t)] Q_i(t)\, dt \tag{4}$$

The weights are updated at the end of each learning cycle ($t = (j+1)T$) as

$$w_i^{\text{new}} = w_i^{\text{old}} + \delta w_i \tag{5}$$

The procedures above, *i.e.*, numerical calculations of Eqs. (1), (4) and (5), are repeated ($j = 0, 1, \cdots$) until the error between the input and the output becomes small enough.

Because our model is meant for hardware implementation, it is necessary to take physical time for updating the weights (Eq. (5)) and resetting the integrated value in Eq. (4) before starting another learning cycle, even though the updating and resetting terms are assumed to be zero in Eqs. (4) and (5). In practical hardware, a single learning cycle consists of the input sequence's length (T), and the updating and resetting terms, as shown in Fig. 2. Note that each oscillator's starting phase must be the same at the beginning of each learning cycle. For example, oscillators Q_1 and Q_2 in Fig. 2 have the same starting phase at the beginning of each learning cycle. If the starting phases of Q_is at the j-th learning cycle are different from that of Q_is at the $(j+1)$-th cycle, the update value at the end of the j-th cycle (δw_i) has no meaning. Because the δw_i is calculated by phase activities of Q_is in the j-th cycle, and is effective only for decreasing errors with Q_is in the $(j+1)$-th cycle that has the same starting phases as in the j-th cycle.

Numerical simulations were conducted to confirm the operation of the model. In the simulation, output of the oscillatory units $Q_i(t)$ was defined by

$$Q_i(t) = H[\sin(2\pi f_i t)] \tag{6}$$

where f_i represents a random frequency distributed between 1 and 10 using white noise sources, and $H(x)$ is the step function defined as:

$$H(x) = \begin{cases} 1 & (x > 0) \\ 0 & (x < 0) \end{cases} \tag{7}$$

The results are shown in Fig. 3 ($N = 200$, $T = 1$ and $\eta = 0.01$). After the first learning (Fig. 3(a)), the input ($I(t)$) and the output sequences ($u(t)$) were completely different; however, $u(t)$ approached $I(t)$ after repeated learning cycles (Figs. 3(b) for 10th and (c) for 100th learning).

Figure 4 shows the time evolution of the mean square error (E) of the proposed network with N = 1, 30, 100 and 200. The error decreased as the learning cycle (j) increased, as expected. Since the error values for N = 30, 100 and 200 approached the same value (≈ 0.2), we can avoid implementing a large number of oscillators and synaptic connections in hardware. The error in our modified model, (≈ 0.2 with N=100 and 100 learning cycles) was about twice that of the original model (≈ 0.1 with N=100 and 100 learning cycles; [1]). Despite this difference our modified model is applicable in areas that do not require errorless learning, $e.g.$, low-quality voice recording (learning) for mobile products, etc.

Furthermore, we evaluated the storage capacity of the proposed network by defining pattern overlaps between the input and output sequences as a function of N and the complexity of the input sequences. To define the complexity ($\equiv \lambda$), we used Poisson spikes whose mean firing rate is represented by

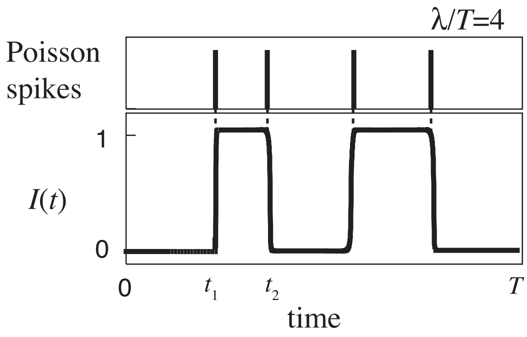

Fig. 5 Input sequence ($I(t)$) generated by Poisson spikes with $\lambda/T = 4$

λ. Let us assume binary input sequence $I(t)$ with period T and $I(0)$ = "0". The expected number of spikes within period T is thus λ/T. The value of the input sequence is flipped and kept when a spike is generated; *i.e.*, $I(t)$ $(t > 0)$ remains "0" if no spikes are generated, whereas $I(t)$ $(t > t_1)$ is flipped to "1" when a spike is generated at $t = t_1$. When a subsequent spike is generated at $t = t_2$, $I(t)$ $(t > t_2)$ is flipped to "0". Figure 5 shows examples with $\lambda/T = 4$. This process is repeated while $t \leq T$

The pattern overlap between input $I(t)$ and output sequences $u(t)$ is defined by

$$m \equiv \frac{1}{T}\int_0^T 2\left(I(t) - \frac{1}{2}\right) \times 2\left[H\left(u(t) - \frac{1}{2}\right) - \frac{1}{2}\right] dt \tag{8}$$

where $I(t)$ is expanded to ± 1, and the Boolean values for threshold evaluation $(u(t) > 0.5)$ are also expanded to ± 1.

Figure 6 shows the average of the pattern overlaps between 10 different sets of input sequences and their respective outputs for different values of λ when $T = 1$. Outputs $u(t)$ were obtained after the 100th learning cycle. We observed that the pattern overlap decreased as λ increased. As expected, sequences with small iterations are easier to learn than complex sequences.

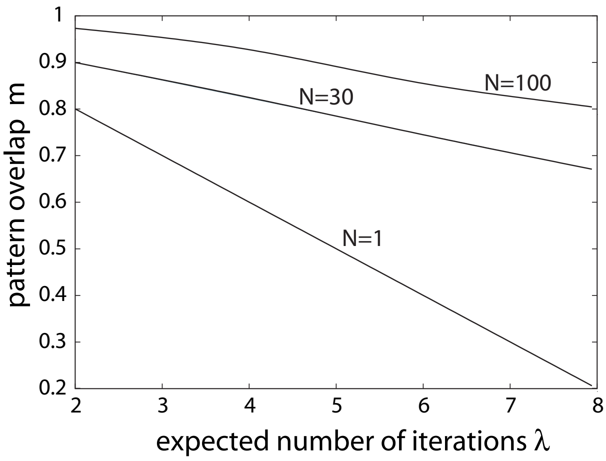

Fig. 6 Pattern overlap between input and output sequences

3. Analog MOS Circuits for Temporal Coding Model

First, we used Wilson-Cowan oscillators [10] to implement the oscillator circuits. The dynamics are given by

$$\frac{du_i}{dt} = -u_i + f_\beta(u_i - v_i) \tag{9}$$

$$\frac{dv_i}{dt} = -v_i + f_\beta(u_i - \theta) \tag{10}$$

where u_i and v_i represent the system variables of the i-th oscillator, θ is the threshold and $f_\beta(\cdot)$ is the sigmoid function with slope β. Figure 7 shows a MOS circuit that implements the Wilson-Cowan oscillator. The circuit consists of an operational transconductance amplifier (OTA) and a buffer circuit composed of two standard inverters. When the time constants of the Wilson-Cowan system are very small, we can rewrite Eqs. (9) and (10) as

$$u_i \approx f_\beta(u_i - v_i) \tag{11}$$

$$v_i \approx f_\beta(u_i - \theta) \tag{12}$$

The OTA's output voltage ($V_{\rm o}$) is expressed as $V_{\rm d} \cdot f(V_1 - V_2)$, while the output voltage of the buffer circuit ($V_{\rm o2}$) is given by $V_{\rm d} \cdot f(V_{\rm in} - V_{\rm th})$, where $f(\cdot)$ represents a nominal Sigmoid-like function, and $V_{\rm th}$ is the threshold voltage of the buffer circuit. Thus we obtain

$$u_i = V_{\rm d} \cdot f(u_i - v_i) \tag{13}$$

$$v_i = V_{\rm d} \cdot f(u_i - V_{\rm th}) \tag{14}$$

by connecting the inputs and outputs to u_i and v_i as shown in Fig. 7 ($V_1 = V_{\rm o} = u_i, V_2 = v_i, V_{\rm in} = u_i, V_{\rm o2} = v_i$), which corresponds to Eqs. (11) and (12). Here we use v_i to represent Q_i as V_i^Q. The oscillatory state (oscillating or resting) can be controlled by changing the power supply voltage (V_d), which is necessary for setting the same starting phases at the beginning of each learning cycle, as explained in section 2.

Second, let us implement synaptic connections and an output cell in the proposed model. Because the weights between the oscillatory units and the output cell (w_is) in our model take both positive and negative values, it is important to consider how to represent positive and negative synaptic weights in analog MOS circuits. Traditional circuits implement such bipolar weights as resistors with voltage mode neurons having positive- and negative-gain unity amplifiers. According to

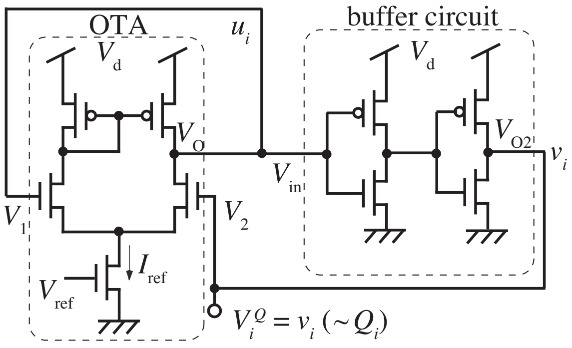

Fig. 7 Neural oscillator circuit

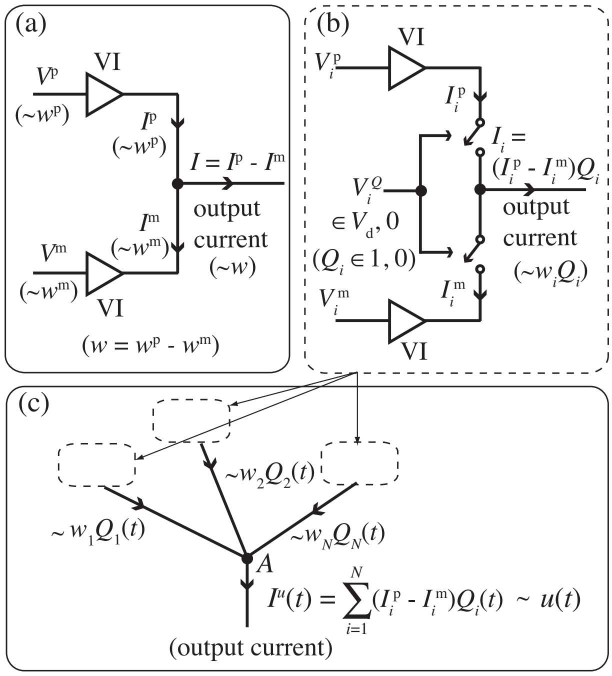

Fig. 8 Schematic showing the main idea for implementation of bipolar synapses and output cell

the sign of the weights, one of the amplifiers must be selected. Implementing negative unity-gain amplifiers and the selection circuit may occupy a large area in analog LSIs. Therefore we designed "current-mode circuits" where positive and negative synaptic weights are represented by "currents".

Let us define a differential weight $w \equiv w^{\rm p} - w^{\rm m}$, where both $w^{\rm p}$ and $w^{\rm m}$ take positive values, and introduce weight voltages $V^{\rm p}$ and $V^{\rm m}$ which are proportional to $w^{\rm p}$ and $w^{\rm m}$, respectively. Through voltage-to-current converters (VIs), $V^{\rm p}$ and $V^{\rm m}$ are also converted into currents $I^{\rm p}$ and $I^{\rm m}$ and then wired. This setup is illustrated in Fig. 8(a). Now, the output current I is given by $I^{\rm p} - I^{\rm m}$, which is proportional to I and w can take both positive ($I^{\rm p} > I^{\rm m}$) and negative currents ($I^{\rm p} < I^{\rm m}$). Based on this idea, we designed a synapse circuit that connects the oscillator circuits and an output cell circuit. Figure 8(b) shows the concept of the i-th synapse circuit which calculates Eq. (1). Two ideal switches are inserted into the output lines of the VIs. Since both switches are turned on (or off) when control voltage V_i^Q (output of the i-th oscillator) is "1" (or "0"), the output current is represented by $(I_i^{\rm p} - I_i^{\rm m})Q_i$ which is proportional to $w_i Q_i$. Figure 8(c) illustrates the concept of the output cell, which sums up the output currents of the synapse circuits. Since $(I_i^{\rm p} - I_i^{\rm m})Q_i$ is represented by current, the output current $I^u(t)$ flowing from node A is

$$I^u(t) = \sum_{i=1}^{N} (I_i^{\rm p} - I_i^{\rm m}) Q_i(t) \tag{15}$$

which is thus proportional to $u(t)$ (output of the proposed model).

Figure 9 illustrates the MOS circuit for the i-th synaptic circuit model shown in Fig. 8(b). The circuit consists of two pass transistors (m_5 and m_6) and a transconductance amplifier (m_1-m_4 and m_7-m_{12}) that acts as a voltage-to-current converter (VI in Fig. 8(b)) with limited linear range. The amplifier consists of a differential pair (m_1, m_2 and m_3) and current mirrors (m_7-m_8, m_9-m_{10}, m_{11}-m_{12} and m_3-m_4). When V_i^Q is logical "1", the current of transistor m_1 produced by differential voltage $V_i^{\rm p} - V_i^{\rm m}$ is copied to $I_i^{\rm p}$ by current mirror m_9-m_{10}. At the same time, the current of transistor m_2 is copied to $I_i^{\rm m}$ by current mirrors m_7-m_8 and m_{11}-m_{12}. The output current I_i is thus given by $(I_i^{\rm p} - I_i^{\rm m})Q_i(t)$.

As explained in section 2, to learn the input sequences correctly, it is necessary to minimize the error between the input and output sequences by updating the weights according to Eqs. (4) and (5). So our next step is to implement Eq. (4). Since δw_i takes positive and negative values, we use the same 'differential' strategy as in our synapse circuit. Assuming that $I(t)$ and $u(t)$ are represented by currents $I_{\rm in}(t)$ and $I^u(t)$, respectively, and that the currents are integrated by capacitors, we can rewrite Eq. (4) as

$$\delta w_i \sim V_i^I - V_i^u \tag{16}$$

$$V_i^I \equiv \frac{1}{C}\int_{jT}^{(j+1)T} I_{\rm in}(t) Q_i(t) dt \tag{17}$$

$$V_i^u \equiv \frac{1}{C}\int_{jT}^{(j+1)T} I^u(t) Q_i(t) dt \tag{18}$$

where C represents the capacitance. Currents $I_{\rm in}(t)$ and $I^u(t)$ are separately integrated by capacitors, and the integrated values are represented by voltages V_i^I and V_i^u.

A MOS circuit that implements Eqs. (17) and (18), which we call "integrator circuit", is shown in Fig. 10. The circuit

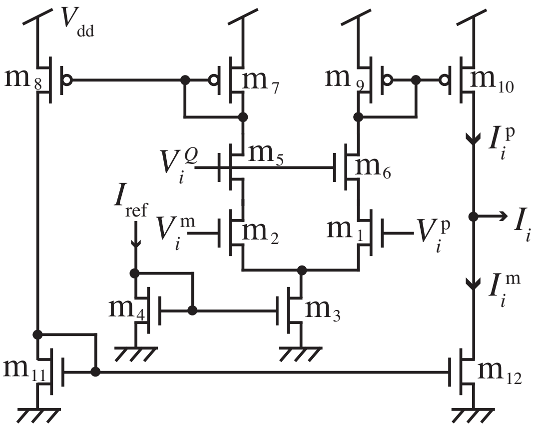

Fig. 9 Synapse circuit calculating weighted sum (I_i) of output of oscillator (V_i^Q) and stored weight voltages ($V_i^{\rm p}$ and $V_i^{\rm m}$)

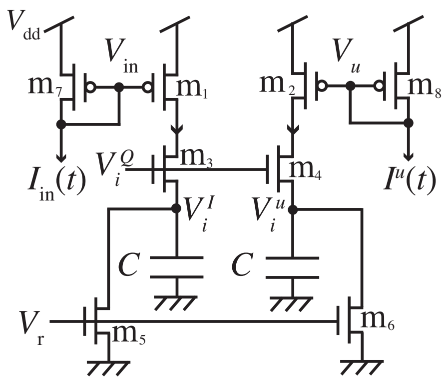

Fig. 10 Integrator circuit

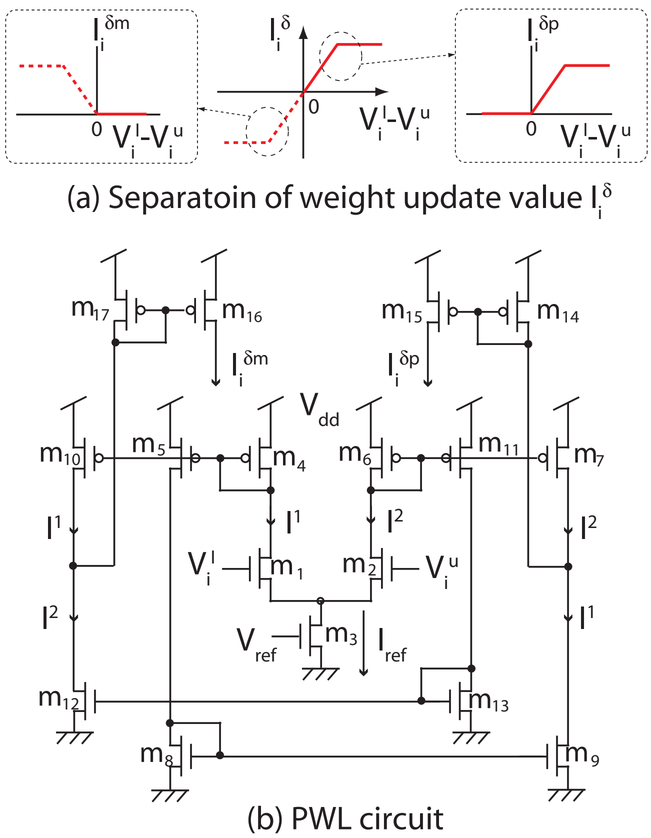

Fig. 11 MOS circuits for calculating weight update values

consists of two current mirrors (m_1-m_7 and m_2-m_8), two pass transistors (m_3 and m_4), two capacitors (C's), and two transistors for reset operations (m_5 and m_6). When V_i^Q is a logical "1", $I_{\rm in}(t)$ and $I^u(t)$ are copied to pass transistors m_3 and m_4, respectively, by the current mirrors, and are integrated by the capacitors. As explained in section 2, before starting each learning cycle, V_i^I and V_i^u, must be reset to 0 by setting $V_{\rm r}$ to "1". Remember that voltages $V_{\rm in}$ and V_u in Fig. 10 reflect the temporal input ($I(t)$) and output sequences ($u(t)$) that will be used to represent the simulation results in section 4.

Next, let us evaluate the difference between the integrated voltages V_i^I and V_i^u to calculate Eq. (16). Assume that the differential voltage is nonlinearly converted into current I_i^{δ} by the transconductance amplifier. The characteristic is illustrated in Fig. 11(a) (center). The transferred current is separated into positive and negative parts. The positive (or negative) I_i^{δ} is copied to $I_i^{\delta{\rm p}}$ (or $I_i^{\delta{\rm m}}$), whereas $I_i^{\delta{\rm p}} = 0$ (or $I_i^{\delta{\rm m}} = 0$) when $I_i^{\delta} < 0$ (or $I_i^{\delta} > 0$), as shown in Fig. 11(a) right (or left).

A MOS circuit that produces $I_i^{\delta{\rm p}}$ and $I_i^{\delta{\rm m}}$, which we call "piecewise linear (PWL) circuit", is shown in Fig. 11(b). The circuit consists of a differential pair (m_1 to m_3) and current mirrors (m_4 to m_{17}). When the differential pair is operating in the subthreshold region, currents I^1 and I^2 are given by

$$I^1 = I_{\rm ref}\frac{\exp(\kappa V_i^I)}{\exp(\kappa V_i^I)+\exp(\kappa V_i^u)} \tag{19}$$

$$I^2 = I_{\rm ref}\frac{\exp(\kappa V_i^u)}{\exp(\kappa V_i^I)+\exp(\kappa V_i^u)} \tag{20}$$

The resulting differential current ($I_i^{\delta} = I_1 - I_2$) is proportional to the hyperbolic tangent of $V_i^I - V_i^u$. Currents I^1 and I^2 are copied to m_7 and m_9, respectively. When $I_1 > I_2$ (or $I_1 < I_2$), current mirror m_{14}-m_{15} copies (or does not copy) $I_1 - I_2$ to $I_i^{\delta{\rm p}}$. This operation corresponds to Fig. 11(a) right. Simultaneously, currents I^1 and I^2 are copied to m_{10} and

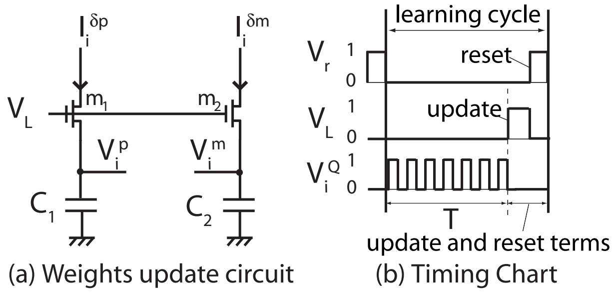

Fig. 12 Weights update circuit and timing chart

m_{12}. When $I_2 > I_1$ (or $I_2 < I_1$), current mirror m_{16}-m_{17} copies (or does not copy) $I_2 - I_1$ to $I_i^{\delta\mathrm{m}}$, which corresponds to characteristics in Fig. 11(a) left.

As explained in section 2, at the end of each oscillatory cycle T, the weights have to be updated according to Eq. (5). We have already separated δw_i into positive and negative parts, as shown in Fig. 11(a), and obtained two positive currents $I_i^{\delta\mathrm{p}}$ and $I_i^{\delta\mathrm{m}}$. Assuming that the bipolar weights are separately stored in capacitors and are updated with the amounts of $I_i^{\delta\mathrm{p}}$ and $I_i^{\delta\mathrm{m}}$, then Eq. (5) can be rewritten as

$$V_i^{\mathrm{p}}(t+\Delta t) = V_i^{\mathrm{p}}(t) + \frac{\Delta t}{C} I_i^{\delta\mathrm{p}} L \tag{21}$$

$$V_i^{\mathrm{m}}(t+\Delta t) = V_i^{\mathrm{m}}(t) + \frac{\Delta t}{C} I_i^{\delta\mathrm{m}} L \tag{22}$$

where C represents the capacitance, Δt the time step of learning, L the normalized binary value ($\equiv V_{\mathrm{L}}/V_{\mathrm{dd}}$) for controlling the weight update, V_i^{p} and V_i^{m} the integrated (updated) weight values. When $\Delta t \to 0$, we obtain the differential forms

$$C\frac{dV_i^{\mathrm{p}}}{dt} = I_i^{\delta\mathrm{p}} L \tag{23}$$

$$C\frac{dV_i^{\mathrm{m}}}{dt} = I_i^{\delta\mathrm{m}} L \tag{24}$$

Figure 12(a) illustrates a MOS circuit that calculates Eqs. (23) and (24). During the update cycle (V_{L} is logical "1"), $I_i^{\delta\mathrm{p}}$ and $I_i^{\delta\mathrm{m}}$ are separately integrated by capacitors C_1 and C_2, respectively, via pass transistors m_1 and m_2. Remember that the integrated values V_i^{p} and V_i^{m} represent the weight w_i ($\sim V_i^{\mathrm{p}} - V_i^{\mathrm{m}}$), and they are fed back to the i-th synapse circuit shown in Fig. 9.

Figure 12(b) summarizes the circuit's control voltages per single learning cycle (timing chart). Before each learning cycle is started, V_{r} is set to logical "1" to reset the weight update values δw_i ($V_i^I = V_i^u = 0$). At the beginning of each learning cycle, the V_{d} of the oscillator circuit shown in Fig. 7 is set to V_{dd} and V_i^Q starts to exhibit square-wave oscillations. At the end of the oscillatory cycle, V_{d} is set to 0 (thus the oscillation stops) and in turn the weight update begins (V_{L} = "1"). When the update is finished, V_{r} is set to "1". This process is repeated until the difference between the input and output sequences becomes small enough.

4. Simulation Results

We conducted SPICE simulations for each circuit component in section 3. In the simulations, we used TSMC 0.35-μm CMOS parameters. Figure 13(a) shows the results of a single oscillator circuit, integrator circuit and PWL circuit. In the oscillator circuit, all the dimensions (W/L) of the transistors were set to 2 μm / 0.24 μm, and V_{ref} was set to 450 mV. The supply voltage V_{d} was 2.5 V (or 0). We confirmed that i) the circuit oscillated when the supply voltage was given, and ii) the starting phases at the beginning of the learning cycles (at $V_{\mathrm{d}} = 0 \to 2.5$ V; *i.e.*, $t = 0.4$ μs and 0.8 μs) were the same, as shown in Fig. 13(a).

Simulation results for the integrator circuit are shown in Fig. 13(b). All the dimensions of the transistors in the circuit were set to 0.36 μm / 0.24 μm. Input currents I_{in} and I^u were set to 1 μA and 2 μA, respectively. Capacitance C was set to 1 pF, and the supply voltage V_{dd} was set to 2.5 V. Figure 13(b) shows that independently of the control voltage V_i^Q, integrated voltages V_i^I and V_i^u were reset to 0 when the reset control voltage (V_{r}) was set to logical "1" ($t = 0 \sim 0.25$ μs). The integration started when V_{r} was set to "0" and V_i^Q was "1", which resulted in an increase in V_i^I and V_i^u ($t = 0.25 \sim 0.5$ μs). Then the integration stopped and V_i^I and V_i^u were preserved when V_i^Q was "0" ($t = 0.5 \sim 0.75$ μs). Again, when V_{r} was set to "1", the integrated voltages were reset to zero ($t = 0.75 \sim 1$ μs).

Figure 13(c) shows the simulation results for a single PWL circuit. The transistor dimensions were 7.2 μm / 0.24 μm for m_7 and m_{10}, 1.6 μm / 0.24 μm for m_9 and m_{12}, 0.72 μm / 0.24 μm for m_{14} and m_{17}, and 0.36 μm / 0.24 μm for the remaining transistors. The supply voltage (V_{dd}), V_i^u and V_{ref} were set to 2.5 V, 1.25 V and 1 V, respectively. As shown in Fig. 13(c) we could obtain similar characteristics as Figs. 11(a) left and right; *i.e.*, when $V_i^I > V_i^u$, $I_i^{\delta\mathrm{p}}$ was monotonically increased as V_i^I increased, whereas $I_i^{\delta\mathrm{p}}$ remained zero when $V_i^I < V_i^u$. On the other hand, when $V_i^I > V_i^u$, $I_i^{\delta\mathrm{m}}$ was zero while $I_i^{\delta\mathrm{m}}$ was monotonically decreased as V_i^I increased when $V_i^I < V_i^u$.

We confirmed the learning operation of the entire circuit with $N = 20$. The fundamental frequencies (f_i's) of the oscillators were set by

$$f_i \approx 0.3i + 1.1(\mathrm{MHz}) \tag{25}$$

where i represents the neuron index, which results in a distribution between 1.4 MHz and 7.1 MHz. The learning cycle was set to 1 μs with T, the updating and the resetting terms were set to 0.7 μs, 0.1 μs and 0.2 μs, respectively. The input sequences ($I(t)$) were generated with current pulses of 0.1

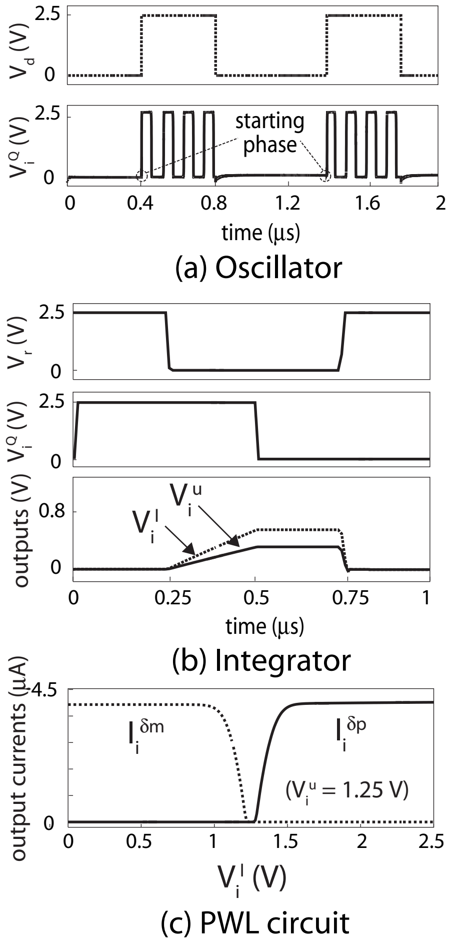

Fig. 13 Simulation results of circuit components

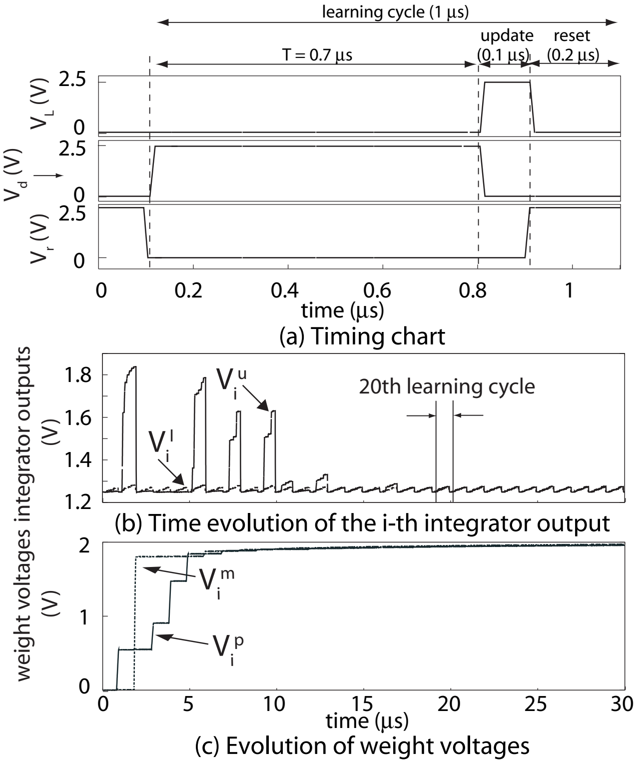

Fig. 14 Simulation results of circuit network with N=20

μA in amplitude, and λ/T was set to 4. Capacitances C_1 and C_2 in Fig. 12(a) were set to 1 pF, and the supply voltage $V_{\rm dd}$ was set to 2.5 V.

Figure 14(a) shows the timing chart for a single learning cycle. The time evolution of the i-th integrator outputs (V_i^I and V_i^u) and those of the weight voltages ($V_i^{\rm p}$ and $V_i^{\rm m}$) are shown in Figs. 14(b) and (c), respectively. We could observe that V_i^I and V_i^u took almost the same values; *i.e.*, errors between the input and output sequences became zero after approximately 20 learning cycles. The weight voltages were successfully updated at the end of each learning cycle; when $V_i^I > V_i^u$, the positive weight ($V_i^{\rm p}$) was increased, whereas, when $V_i^I < V_i^u$ the negative weight ($V_i^{\rm m}$) was increased, until the two attained stable values.

The time courses of temporal input voltage $V_{\rm in}$ ($\sim I(t)$; see Fig. 10) and learned output voltage V_u ($\sim u(t)$) are shown in Figs. 15 and 16. We could observe that $V_{\rm in}$ and V_u were different at the beginning (Fig. 15) but became similar after about 29 learning cycles (Fig. 16).

It is important to note that all the MOSFETs in the proposed circuit operate in their sub-threshold region. To ensure sub-threshold operation of the MOSFETs, we set bias voltage $V_{\rm ref}$ at the lower values of the MOSFETs threshold voltage, which results in fundamental frequencies of oscillators in the MHz range, (about 1 MHz to 10 MHz for the upper bound frequency). Note that, it is possible to learn temporal sequences in the audio frequency (kHz range) by changing the bias voltage value ($V_{\rm ref}$) of the oscillator circuit from 0.09 V to 0.1 V.

Finally, we calculated the pattern overlaps in Eq. (8) between the input and output sequences produced by our circuits for different sets of input sequences (λ). The input sequences were generated with current pulses of 0.5 μA in amplitude. The oscillatory cycle (T), updating and resetting terms were set to the same values as in the simulations of Figs. 14 to 16. The calculations were carried out for 1 and 30 neuron networks. The fundamental frequencies, set by Eq. (25), were distributed between 1.4 MHz and 10.1 MHz for N=30. Figure 17 shows the averaged pattern overlap between 10 different sets of input sequences and their outputs. For comparison, numerical results of the network model in section 2 with the same number of neurons are also shown

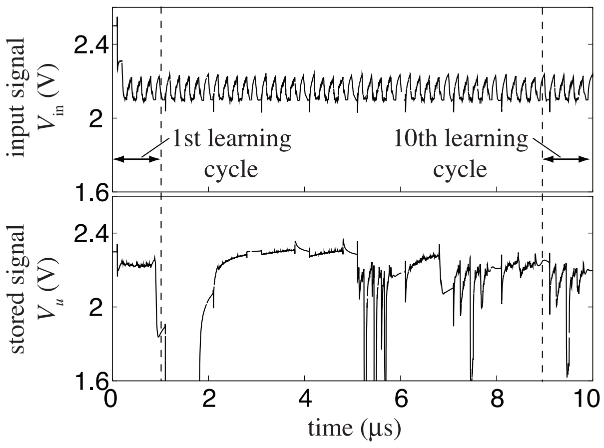

Fig. 15 Evolution of temporal input sequence $V_{\rm in}$ and learned output sequence V_u (first to 10th learning cycles)

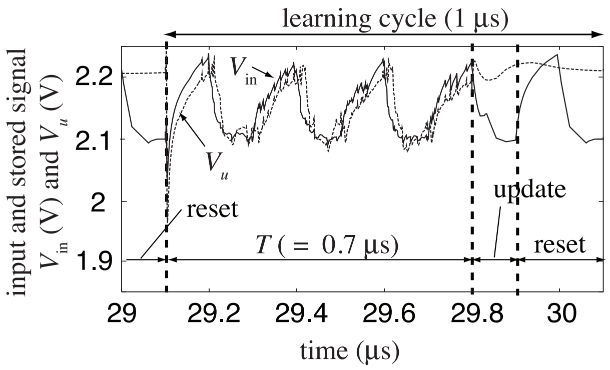

Fig. 16 Temporal input sequence $V_{\rm in}$ and learned output sequence V_u after 29th learning cycle

in the figure. The difference between the SPICE and numerical results are caused by the limited linear ranges of synapse circuit's VIs and PWL circuits. These results show that the circuit network of $N = 30$ can retrieve input sequence of $\lambda/T = 6/(0.7\ \mu\mathrm{s}) \approx 8.6 \times 10^6$ (s^{-1}) with an accuracy of 72% ($m \approx 0.72$), which indicates that the circuit can learn and recall temporal sequence of 4.3 MHz under our device setups.

5. Conclusion

In this paper, we designed a neural circuit for temporal coding. The network circuit was designed by analog metal-oxide-semiconductor (MOS) devices. The model consists of N oscillatory units connected to an output cell through synaptic connections. To facilitate the implementation of the

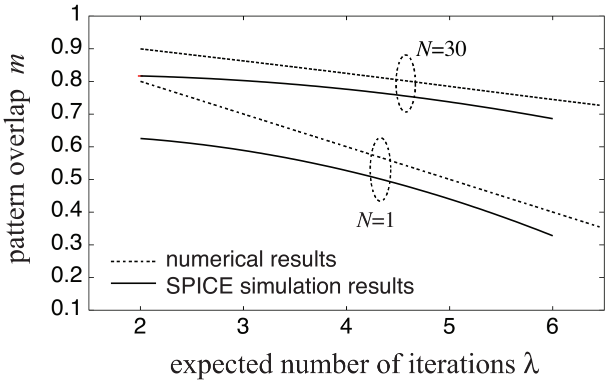

Fig. 17 Numerical and SPICE results showing pattern overlaps between input and output sequences for different Ns and complexity of input sequence λ

model, we designed current-mode circuits where the input, output, and the weight values were represented by currents. We demonstrated the operation of each component of the network using a simulation program with integrated circuit emphasis (SPICE). Moreover, we confirmed operation of the entire circuit with 20 neurons, and confirmed that after several leaning cycles, the input and output sequence had the same phase. The storage ability was also evaluated. When $N = 30$, the circuit could learn and recall binary temporal sequences with 6 iterations in the learning cycle with the accuracy of 72% under physically plausible device configurations.

Acknowledgments

This study was partly supported by the Industrial Technology Research Grant Program in '04 from New Energy and Industrial Technology Development Organization (NEDO) of Japan, and a Grant-in-Aid for Young Scientists [(B)17760269] from the Ministry of Education, Culture Sports, Science and Technology (MEXT) of Japan.

References

[1] T. Fukai: A model cortical circuit for the storage of temporal sequences, Biol. Cybern., Vol. 72, No. 4, pp. 321-328, 1995.

[2] K.M. Knutson et al.: Brain activation in processing temporal sequence: an fMRI study, NeuroImage, Vol. 23, No. 4, pp. 1299-1307, 2004.

[3] B. Baird: Nonlinear dynamics of pattern formation and pattern recognition in the rabbit olfactory bulb, Physica D, Vol. 2, No. 1-3, pp. 150-175, 1986.

[4] W.J. Freeman: Why neural networks don't get fly: inquiry into the neurodynamics of biological intelligence, Proc. IEEE Int. Conf. Neural Networks, Vol. 2, pp. 1-7, 1988.

[5] D.L. Wang: Temporal pattern processing, Handbook of Brain Theory and Neural Networks, MIT Press, pp. 967-971, 1995.

[6] A.R. Aluizo and G.A. Barreto: Context in temporal sequence processing: A self-organizing approach and its application to robotics, IEEE Trans. Neural Networks, Vol. 13, No. 1, pp. 45-57, 2002.

[7] S.C. Liu and R. Douglas: Temporal coding in a silicon network of integrated and fire neurons, IEEE Trans. Neural Networks, Vol. 15, No. 5, pp. 1305-1314, 2004.

[8] H.H. Ali and M.E. Zaghloul: VLSI implementation of an associative memory using temporal relations, Proc. IEEE Int. Symp. Circuit and Syst., pp. 1877-1880, 1993.

[9] J.L. Walsh, et. al.: A close set of orthogonal functions, Amer. J. Math., Vol. 45, No. 1, pp. 5-24, 1923.

[10] H.R. Wilson and J.D. Cowan: Excitatory and inhibitory interactions in localized populations of model neurons, Biophys. J., Vol. 12, pp. 1-24, 1972.

Gessyca Maria Tovar was born in Venezuela in 1981. She received the B.S. degree in electronic engineering from Jose Antonio Paez University, Venezuela, in 2004, and the M.S. degree from Hokkaido University, Hokkaido, Japan, in 2008. Currently she is working towards her Doctor degree in the Department of Electrical Engineering at Hokkaido University, Japan. Her research interests include neuromorphic computation and hardware implementation. She is a student member of the INNS.

Tetsuya Asai is an Associate Professor in the Graduate School of Information Science and Technology, Hokkaido University, Sapporo, Japan. His research interests concentrate around developing nature-inspired integrated circuits and their computational applications. Current topics that he is involved with include; intelligent image sensors that incorporate biological visual systems or cellular automata in the chip, neuro chips that implement neural elements (neurons, synapses, etc.) and neuromorphic networks, and reaction-diffusion chips that imitate vital chemical systems.

Daichi Fujita was born in Yamaguchi, Japan, in 1985. He received the B.S. degree in electrical engineering from Hokkaido University, Hokkaido, Japan, in 2008. Currently he is working towards his Master degree in the Department of Electronic for Informatics at Hokkaido University, Japan. His research interests include neuromorphic computation and hardware implementation. He is a student member of the IEICE.

Yoshihito Amemiya received the B.E., M.E., and Ph.D. degrees from the Tokyo Institute of Technology, Tokyo, Japan, in 1970, 1972, and 1975. He joined NTT Musashino Laboratories in 1975, where he worked on the development of silicon process technologies for high-speed logic LSIs. From 1983 to 1993, he was with NTT Atsugi Laboratories and developed bipolar and CMOS circuits for Boolean logic LSIs, neural network LSIs, and cellular automaton LSIs. Since 1993, he has been a Professor of the Department of Electrical Engineering, Hokkaido University, Sapporo. His research interests are in the fields of silicon LSI circuits, signal processing devices based on nonlinear analog computation, logic systems consisting of single-electron circuits, and information-processing devices making use of quantum nanostructures.

(Received June 9, 2008; revised September 16, 2008)