Neuromorphic MOS Circuits Implementing a Temporal Coding Neural Model

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

Graduate School of Information Science and Technology, Hokkaido University.
Kita 14, Nishi 9, Kita-ku, Sapporo, 060-0814 Japan.
Phone:+81-11-706-7147, Fax:+81-11-706-7890
† email: gessyca@sapiens-ei.eng.hokudai.ac.jp

Abstract

A model for the storage of temporal sequences was proposed. On the basis of this model, we propose a neural model suitable for implementation on CMOS circuits that is capable of learning and recalling temporal sequences. In this paper, we numerically confirmed basic operations of the model and demonstrate fundamental circuit operations using a simulation program with integrated circuit emphasis (SPICE).

1. Introduction

Many real-world tasks demand the ability of natural, or artificial neural systems to process patterns in which the information content depends on the temporal order of the input patterns. In consequence, temporal information processing is of fundamental importance in various brain functions. The brain routinely learns and recalls information as the environment changes over time. This set of temporally ordered patterns is commonly referred to as "spatio-temporal sequence learning". The processing of such sequences is a topic arising in several fields, such as pattern recognition.

In [1] Fukai proposed a model for the storage of temporal sequences. Based on this model, we propose a neural model being suitable for implementation on CMOS circuits that is capable of learning and recalling temporal sequences. The model consists of neural oscillators connected to an output cell through synaptic connections. The basic idea is to learn input sequences, by superposition of rectangular periodic activity (oscillators) with different frequencies.

In the following sections, we explain the operation of the temporal coding model. Then we present the CMOS circuit for implementing the model. Finally we demonstrate the operation of the network using a simulation program with integrated circuit emphasis (SPICE).

2. The model

The main purpose of the model [1] is learning and memorizing the input stimuli \( I(t) \). The temporal coding model is shown in Fig. 1. The model consists of \( N \) neural oscillators \( (Q_i) \) with different oscillation frequencies and an output terminal cell \( O \). One oscillator is composed of a pair of excitatory and inhibitory cells \((u_i, v_i)\) based on the Wilson-Cowan model [2], [3]. All the oscillators are connected to the output cell \( O \) through synaptic connections. The input sequence \( I(t) \) is given to the output cell as a supervisory signal. The synaptic weights, \( w_i \), are strengthened (weakened) when the \( Q_i \)'s oscillatory cells overlap (or not) with the input sequence \( I(t) \), and then, the output cell \( O \) to reproduce the temporal sequences. The weights \( w_i \) are modified according to the gradient descent rule \( \delta w_i \sim -\partial E/\partial w_i \), where \( E \) is the mean square error of the learning given by:

\[
E = \frac{1}{2T} \int_0^T [I(t) - u(t)]^2 dt
\]  (1)

where \( u \) is the membrane potential of \( O \) given by \( u = \sum_{i=1}^{N} w_i Q_i \), and then \( \delta w_i \) becomes:

\[
\delta w_i \sim -\frac{\partial E}{\partial w_i} = \frac{1}{T} \int_0^T [I(t) - u(t)]Q_i dt
\]  (2)

Numerical simulations were conducted to confirm the operation of the model. The simulation results are shown in Fig. 2. The number of neurons was set to \( N = 200 \), and the results were obtained after completing 100 learning cycles. As can

![Figure 1: Temporal coding model.](image-url)
be seen from the simulations, the time courses of membrane potential $V_m$ are similar to those of input $I$.

3. CMOS circuits and operations

The basic structure of a single neuron of the temporal coding circuit is shown in Fig. 3. There is one oscillator with input $V_d$ which acts as trigger signal for the oscillator. The output of the oscillator ($V_Q$) is given to the integrator, which calculates the weight difference $\delta w$ (Eq. 2) by dividing the integral into two parts. Then, at the end of each learning cycle $V_r$ is "1", and the integrator is reseted. The outputs of the integrator ($V_1$ and $V_2$) are given to the OTA which compares the two signals ($V_1 - V_2$) and outputs two currents, $I_p$ for positive weight differences ($V_1 - V_2 > 0$) and $I_n$ for negative weight differences ($V_1 - V_2 < 0$). Then, when $V_1$ is "1" at the end of each oscillation cycle, $T$, the outputs ($I_p$ and $I_n$) are integrated by capacitors $C_1$ and $C_2$ and thus converted to voltages ($V_p$ and $V_n$). Finally these voltages are given to the weight circuit, which compares the two weights (positive and negative) and gives the output of the model $I_o(t)$.

3.1. Oscillators

The construction of a single neural oscillator [3] is illustrated in Fig. 4. The oscillator consists of a differential pair $(m_7-m_8)$, one current mirror $(m_1-m_2)$, a bias transistor $(m_3)$, and a buffer circuit composed of two standard inverters $(m_6-m_7$ and $m_8-m_9)$. The oscillations are controlled by turning on/off the power supply ($V_d$). When $V_d$ is 0 there is no oscillation, and when $V_d$ is "1", the circuit starts the oscillations.

3.2. Integrator

The integrator calculates the weight difference $\delta w$ (Eq. 2) by dividing the integral into two parts:

\[
V_1 = \int_0^T I(t)V_Q \, dt \quad \text{(3)}
\]

\[
V_2 = \int_0^T I_o(t)V_Q \, dt \quad \text{(4)}
\]

The integrator circuit is shown in Fig. 5. The input current ($I$) is copied to node $V_1$ through current mirror $(m_7-m_1)$, when transistor $m_3$ is turned on (or off) by applying "1" (or 0) to $V_Q$ current $I$ is integrated (or is not) by capacitor $C_1$. The result ($V_1$) equivalent to the operation performed by Eq. (3). The same process is carried out for the input $I_o$ and the result ($V_2$) equivalent to that of Eq. (4). Results of the integration ($V_1$ and $V_2$) are reset at the end of each learning cycle by applying "1" to $V_r$.

3.3. OTA

In practical hardware, neuron devices have to be connected by special devices with both positive and negative resistive properties. However, implementing negative resistance is difficult, so we convert the signals into currents and divide the output into a current for positive weights and a current for negative weights, as shown in Fig. 6 (a).

The OTA subtracts one output given by the integrator from the other $(V_1 - V_2)$ and separates the results into two currents, $I_p$ (when $V_1 - V_2 > 0$) and $I_n$ (when $V_1 - V_2 < 0$). The OTA circuit, which consists of a differential pair and current
a small modification. Transistors m3 and m6 were added between the differential amplifier (transistors m3 and m1) and the current mirrors (transistors m2 and m5). In this way, the comparison of the weights (Vp - Vn) will be carried out only when VQ is high.

4. Simulation Results

We conducted SPICE simulations of the complete model. We used TSMC 0.35 µm CMOS parameters. The number of neurons was set to 1. Figure 8 (a) shows the simulations of the oscillator. For the simulations we set W/L = 2 µm/0.24 µm for all transistors. Vref was set to 450 mV. We observed from the simulations that at t = 0.4 µs Vd is turned on and the circuit starts the oscillation. Then, at t = 0.8 µs Vd is off. When Vd is turned on again and the circuit oscillates, "noticing that the oscillations start with the same phase as that of the previous oscillatory period is important". Simulation results of the integrator are shown in Fig. 8 (b). All transistor sizes were set to W/L = 0.36 µm/0.24 µm. Input currents for I and I0 were set to 1 µA and 2 µA, respectively. Capacitances C1 and C2 were set to 1 pF, and the supply voltage Vdd was set to 2.5 V. When Vp is equal to "1" (2.5 V) even if VQ is "0", V1 and V2 are grounded and remain on 0. At t = 0.25 µs, Vp is 0 and VQ remains "1". At this point currents I and I0 are integrated by the capacitors, so voltages at V1 and V2 increase. At t = 0.5 µs, VQ is 0. There is no integration, and voltages V1 and V2 remain at the same voltages as those of the previous state. Then, Vp is "1" at t = 0.75 µs, and voltages V1 and V2 are reset to 0 (the capacitors are discharged). Simulation results for the OTA circuit are shown in Fig. 8 (c). Transistor sizes of m7 and m10 were set to W/L = 7.2 µm/0.24 µm, m9 and m12 to W/L = 1.6 µm/0.24 µm, and m14 and m17 to W/L = 0.72 µm/0.24 µm. The remainder of the transistors was set to W/L = 0.36 µm/0.24 µm. Voltage V1 varies from 0 to 2.5 V. V2 was set to 1.25 V, Vref was set to 1 V, and the supply voltage was set to 2.5 V. When V1 was less than V2, current In is flowing and decreases as V1 increases while Ip.

3.4. Weight Circuit

At the end of the oscillation cycle, transistors m1 and m2 in Fig. 3 are turned on when Vp is "1". This updates weights Vp and Vn for positive and negative weights, respectively. These voltages are given to the weight circuit, which gives the output of the model \( V_p = \sum wV_{Q} \) by comparing the positive and negative weights, Vp and Vn. Our weight circuit is shown in Fig. 7. The circuit consists of a wide-range amplifier with mirrors, is shown in Fig. 6 (b). The current I1 generated by V1 is copied to I3 by current mirror m3 - m10. At the same time current I2 generated by V2 is mirrored to I4 by current mirrors m6 - m11 and m13 - m12. When V1 > V2, current Ip (Ip = I3 - I4) flows and is copied to the output through current mirror m17 - m16. The same process applies to output current In when V1 < V2.

Figure 5: a) Integrator circuit.

Figure 6: a) output characteristic for positive and negative weights, b) OTA circuit

Figure 7: Weight circuit.
5. Conclusion

In this paper, we designed a neural circuit for temporal coding. The network circuit was designed using 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 model, instead of using negative connections required for the implementation of negative weights, we used current signals and divide the weights into two currents: one for positive weights and one for negative weights. We demonstrate the operation of each component of the network separately using a simulation program with integrated circuit emphasis (SPICE). Finally, by making the input current equal to the oscillator current, we confirmed operations of the complete model with one neuron, and we confirmed that after ten learning cycles the output and the input have the same phase.

References