博 士 論 文

Neuromorphic systems performing early-sensory and cognitive processing with CMOS devices

生体の初期感覚および知覚情報処理を模擬するCMOS集積回路に関する研究

Graduate School of Information Science and Technology
Hokkaido University

北海道大学 大学院情報科学研究科

GESSYCA MARIA TOVAR NUNEZ

ジェシカ　マリア　トヴァー　ヌニエス

Feb. 2011

Acknowledgments

First, I would like to thank my supervisor Prof. Tetsuya Asai. I could not have imagined having better advisor, he is an excellent teacher, providing me encouragement, and giving me good advice, and good ideas.

My appreciation also goes to Prof. Yoshihito Amemiya for his support, help and understanding throughout my studies. My sincere thanks are due to Prof. Kanji Yoh and Prof. Yasuo Takahashi, the official referees of this thesis.

I would also like to thank my colleagues. I ' m especially grateful to Akira Utagawa for assisting me in many different ways through all the graduate studies.

During my graduate studies, I worked with many students and former faculty staff. My thanks to Andrew Kilinga Kikombo and Ken-ichi Ueno, for fruitful discussion and constructive advices through my studies. My thanks also go to Dr. Hirose and Dr. Oya for their motivation.

Finally but definitely not the least, I wish to thank my parents Gessy Nuñez de Tovar and Jesus Tovar. They raised me, taught me, supported me and love me. Without them I would not be the person I am today. The same goes to the rest of my family. I would like to extend my sincere thanks to Fumi Kobayashi for her extended support through my graduate studies.

To them I want to dedicate this thesis.

Neuromorphic systems performing early sensory and cognitive processing with CMOS devices

Abstract

This research aims at implementing "Neuromorphic Systems", i.e., circuits inspired by the organizing principles of animal neural systems, implemented using standard Complementary Metal-Oxide Silicon (CMOS) LSI technology. These kinds of circuits are usually parallel, and they respond in real time. They operate mainly in the sub-threshold region, where the transistors have physical properties that are useful for emulating neurons and neural systems, such as thresholding and exponentiation. Based on current knowledge of biological systems, this work aims at developing neural circuits and systems that emulate basic functions of the sensory system. The sensory system is the part of the nervous system responsible for processing sensory information, it consists of sensory receptors, neural pathways, and other parts of the brain involved in sensory perception. Sense perception depends on sensory receptors that respond to various stimuli. When a stimulus triggers an impulse in a receptor, the stimulus is transformed into pulses or action potentials. The action potential travels through a pathway to the cerebral cortex, where they are processed and interpreted. To this end, this research starts with the implementation of some functions of the early-sensory processing like, detection and transformation of input stimuli, role synaptic connections in sensory information processing. This is done by implementing a number of models such as, a) a temperature sensor, (somatosensory system), inspired by the operation of neurons in sea slugs and snails, in order to mimic sensory receptors whose function is to transform physical stimuli into a train of nerve impulses, b) this neuron model was extended for implementing a network for weak signal detection that exhibit tolerance to noises, to explore the ability of sensory systems to exploit noises inherit in their own elements (neurons) as well as noises from the environment (i.e. the input

stimuli), and c) the circuit implementation of a depressing synapse model, whose dynamic effects possibly have a functional role in encoding information brought by sensory stimuli. In auditory pathway, depressing synapses may provide an effective way of detecting emergent synchrony in afferent activities. Then, the attention is shifted to the cognitive processing area with the introduction of two models. a) a neural network for sensory segmentation. To analyze and understand natural scenes, i.e., images, sounds, etc. it is necessary to decompose the scene into coherent "segments", where each segment corresponds to a different component of the scene. This ability is known as sensory segmentation. The model consists of mutually coupled neural oscillators that exhibit synchronous (or asynchronous) activity. The basic idea is to strengthen (or weaken) the synaptic weights between synchronous (or asynchronous) neurons, which may result in phase-domain segmentation. Finally, this work concludes with b) the implementation of a neural model for the storage of temporal sequences. In order to study the brain ability to learn and recall information as the environment changes over time (i.e. information we perceive is time varying) which is of fundamental importance in various sensory functions. The model consists of neural oscillators coupled to a common output cell. The basic idea is to learn input sequences, by superposition of rectangular periodic activity (oscillators) with different frequencies. To mimic the operation of these neurons and networks of neurons, we employed biological nonlinear oscillators. The mathematical model of these oscillators consist of two nonlinear differential equations whose main term is a sigmoid function. The stability of the model depends on the magnitude of its variables. In other words, the model can be excitatory or oscillatory depending on the value of its variables. The models were implemented with basic circuits such as differential pairs (which emulate a sigmoid-like operation) and current mirrors. The operations of the systems were investigated through theoretical analysis, numerical simulations and circuit simulations. The implication of device fabrication mismatches and environmental noise were also studied.

Contents

List of Figures

Chapter 1

Introduction

This research is centered on the study, design and implementation of Neural elements, like neurons and networks of neuron. More specifically, the implementation of neural elements involved in the processing of sensory information. The implementation of such kind of systems is usually referred to as "Neuromorphic systems". It is consist of "simple circuits" inspired by the organizing principles of animal neural systems and implemented using standard Complementary Metal-Oxide Silicon (CMOS) technology. This chapter give an introduction to basic terminology and concepts necessary to have an idea of how the biological sensory system processed information.

1.1 Background

Research into neuromorphic systems is part of the large field of computational neuroscience. The era of neural networks is believed to begin in 1943 with the work of McCulloch and Pitts [1], where they proposed that brain cells (neurons) could be modeled by a "*simple electronic circuit*". During the next fifteen years there was considerable work on detailed logic of threshold networks. In 1958 Frank Rosenblatt introduced his architecture for classification. In 1982, John Hopfield published a paper describing the Hopfield network [2], a simple artificial network which is able to "store" certain patterns.

However, the term "neuromorphic" appears to have started off meaning neuron-like in the late 1980's particularly by those interested in optical implementation of neural networks. The meaning of the word is $mimic(ing)$ specific neurobiological functions, and the meaning seems to have come from the silicon implementation side from the work of mainly two research groups; Alspector's group [3] and Mead's group [4] [5].

The earliest neuromorphic systems were concerned with providing an engi-

neering approximation to some aspects of **sensory systems** such as, auditory system [6] and visual system [7]. More recently, there has also been work on robot control systems, on modeling various types of neurons, and on including adaptations in hardware systems.

Although the detailed information of the brain operation still a puzzle to be solve by neuroscientists, the knowledge that has been accumulated through the biological neural networks research, does give good clues toward the construction of artificial systems that **emulate** some of the characteristic of the nervous system. Biological neural networks provide us with resourceful guidance on building the intelligent machine and to pursue the "brain building". In addition, the progress in hardware implementation will contribute to a better understanding of paradigms and biological systems as well as many useful applications.

Therefore, based on current knowledge of biological systems, this work aims to develop basic neural circuits and networks that emulate characteristics of the processing of information carried-out by the biological sensory system. As known, the sensory system is the part of the nervous system responsible for the processing of sensory information, it consists of sensory receptors, neural pathways, and other parts of the brain involved in sensory perception. This thesis focus mainly in two areas of the sensory processing; the early sensory processing, including receptors and synapses; and the cognitive sensory processing.

1.2 Objective

This research is focus on the design and hardware implementation of biological systems, particularly in the human brain. The human brain is a complex, nonlinear and highly parallel system. Moreover, the brain can easily adjust to a new environment by "learning" and it can deal with information that is fuzzy, noisy or inconsistent. Owing to these characteristics, understanding how the brain works, in particular, how it extracts useful information from "noisy" neural signals, has been one of the most challenging tasks in neuroscience.

With the success of digital systems nowadays, one may ask, what alternative ways of exploring microelectronics are in there?. The answer is simple, the human brain outperforms any computer or supercomputer, not only in size, but also in "efficiency" and robustness. Moreover, the brain can adjust to a new environment by "learning" and it can deal with information that is probabilistic and noisy.

Digital computers of today solve a problem by imposing a computational recipe, or algorithm, on general purpose hardware. So unless the specific steps that a computer needs to follow are known the computer can not solve the

problem. Neuromorphic systems by contrast, embody in the physical behavior of their circuits "analogues" of the processes performs by neural systems. They exhibit fundamental neural functions because the structure of the nervous systems are reproduce on silicon chip. In other words, they transfer our knowledge of neuroscience into practical devices that can interact directly with the real world in the same way that biological neural systems do.

Although the detailed information of the brain operation still a puzzle to be solve by neuroscientists, the knowledge that has been accumulated through the biological neural networks research, does give good clues toward the construction of artificial systems that emulates some of the characteristic of the nervous system.

To this end, based on current knowledge of biological sensory systems, this thesis aims to implementing basic circuits that emulate some basic characteristics of sensory systems; detection of weak and noisy input stimuli, synchronization, properties of synaptic connections, separation or decomposition of natural scenes, and storage of temporal sequences.

The sensory system is a part of nervous system responsible processing sensory information. The sensory system informs areas of the cerebral cortex of changes that are taking place within the body or in the external environment. It consists of sensory receptors that receive stimuli from internal and external environment, neural pathways that conduct this information to the brain, and parts of the brain that process this information. Figure 1.1 shows a schematic of the different senses (from visual to olfaction) and the route they follow to transfer the input stimulus to their respective area in the cortex [8].

Receptors

Receptors are specialized endings of afferent neurons (sensory neurons) or separate cells that affect ends of afferent neurons. They collect information about external and internal environment in various energy forms (stimulus). Stimulus energy is first transformed into nerve impulses (electrical pulses) or receptor potentials by a process called stimulus transduction.

Sensory receptors respond to specific stimulus modalities. Some of them are:

- Thermoreceptors, respond to change in the temperature.
- Photoreceptors, respond to light.
- Mechanoreceptors, detect changes in pressure, position, or acceleration.
- Chemoreceptors, detect certain chemical stimuli.

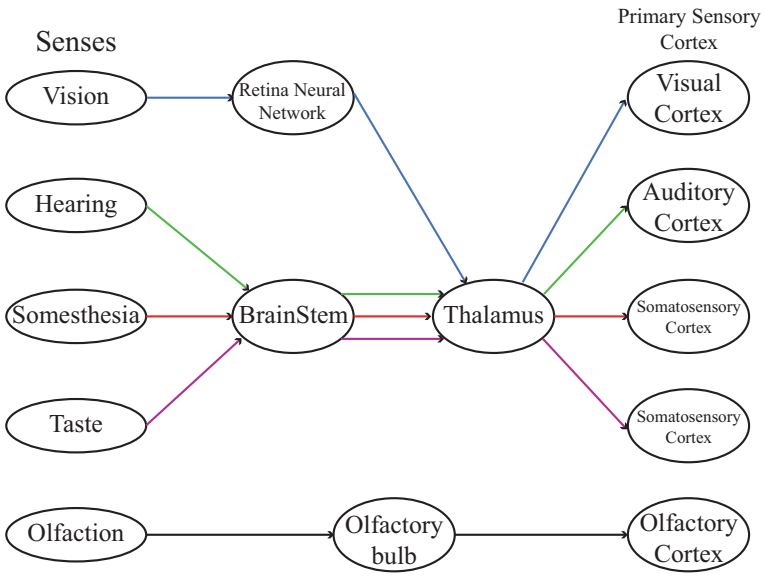


Figure 1.1: Route followed by the different inputs of the sensory system.

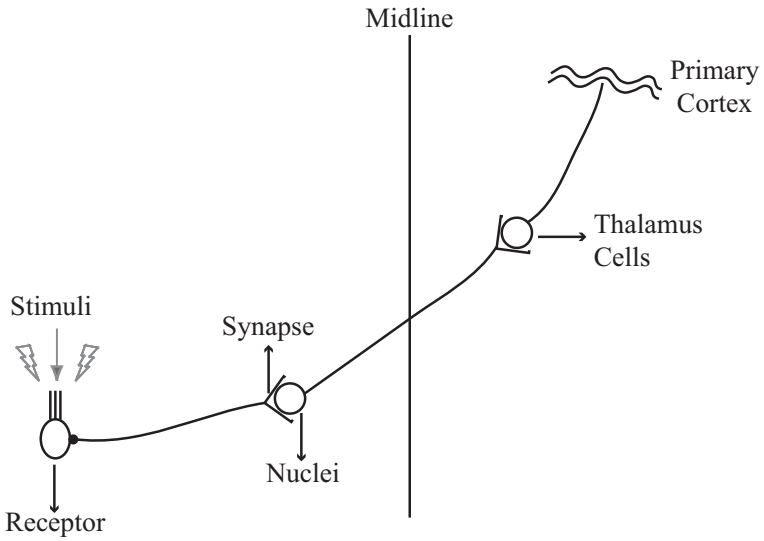


Figure 1.2: General outline of the ascending pathway.

Neural pathway

The sensory or ascending pathway is the route followed by a sensory nerve impulse from a receptor to the brain. Figure 1.2 is a schematic showing the general outline of the ascending pathway. The sensory neurons (that form the ascending pathway) are activated by input stimulus detected and transformed by the sensory receptors (specialized end of the sensory neurons). The active neurons then convey the information to their respective nuclei in the central nervous systems, where the information is further processed as it progress via sensory systems to the cerebral cortex.

The sensory pathway for, the visual, hearing and somesthesia senses are interrupted by synaptic transmission in the ventral thalamus, and the axon of these neurons project to regions of the cerebral cortex that are specific to each sense, and are known as the primary sensory cortices (Fig. 1.1). From there information progresses to secondary and association cortices. The fibers of the taste pathway, after making synaptic connections with cells in the brainstem projects not only to the ventral thalamus, but it also projects to other areas such as, Limbic system, motor pathways and pancreas. The cells in the thalamus project to the insular cortex and somatosensory areas.

The olfactory pathway reaches parts of the central nervous systems that are different from those of the four other senses. After reaching the olfactory bulb, where the first synapse is located, it projects to: anterior olfactory nucleus, Piriform cortex, medial amygdala and entorhinal cortex.

Following the basic characteristic of the biological sensory processing from stimuli reception to processing (sensory pathway) this thesis is outlined as follows:

- Chapter 1 explains the introduction, background and purpose of this research

- Chapter 2 gives an introduction to the basics of neural networks and neuromorphic systems and a brief explanation of CMOS circuits used in this research.

- Chapter 3 starts with the implementation of a circuit for the first step of sensory perception "receptors"; *A temperature receptor*. The model is inspired by the operation of **excitable sensory neurons**. The circuit consists of sub-threshold CMOS circuits whose dynamical behavior changes at a given threshold temperature, *i.e.*, switches to and from oscillatory and stationary. The threshold temperature is set to a desired value by adjusting an external bias voltage. The operation of the model was

studied in detail through theoretical analysis, extensive simulations, and experimentally through discrete MOS devices.

- Chapter 4 introduces a network model exhibiting array-enhanced stochastic resonance, for detection of weak input signal (stimuli). Sensory systems are expose to the noise in the environment and to the noise inherit in their own elements. Therefore, neural systems may employ different strategies that can exploit the properties of noise to improve the efficiency of neural operations. This chapter focuses on the implementation of such kind of noise driven networks in hardware. The model consists of a 2D grid network in which all elements (neuron) accept a common sub-threshold input. In addition, no external noise source is required for the operation of the network as each neuron interact with other neurons through the coupling to generate spatio-temporal noises.

- Chapter 5 introduces a depressing synapse model. Moreover, the dynamical effects of depressing synapses on synchronization are studied using a simple network of neurons. The model was studied through circuits simulations using a simulation program with integrated circuit emphasis (SPICE). Consequently, timing jitter among neurons was significantly reduced when using depressing synapse as compared to non-depressing synapses.

  In chapters 6 and 7, the focus is shifted to the cognitive processing area. Two model are introduced; a neural segmentation model, and a model for the storage of temporal sequences. Therefore, the following chapters are distributed as follow:

- In chapter 6 it is proposed a neural network model for sensory segmentation. Segmentation is refer to the ability to decompose natural scenes into coherent "*segments*" (each segment corresponds to a different component of the scene). The model consists of neural oscillators mutually coupled through synaptic connections. The model performs segmentation in temporal domain, which is equivalent to segmentation according to the spike timing difference of each neuron. Thus, the learning is governed by symmetric spike-timing dependent plasticity (STDP). The basic operations of the proposed model was studied numerically and with circuit simulations using a simulation program with integrated circuit emphasis (SPICE).

- Chapter 7 presents a model for learning and recalling the temporal input stimuli. The model consists of neural oscillators which are coupled to a common output cell through positive or negative synaptic connections. The basic idea is to learn input sequences, by superposition of rectangular

periodic activity (oscillators) with different frequencies, by strengthened (or weakened) the weights of synaptic connections when the output of oscillatory cells overlap (or do not overlap) with the input sequence. The operation of the model was numerically confirmed. Moreover, fundamental circuit operations were studied and the operations of the circuit network was confirmed through SPICE simulations.

- Finally, chapter 8 concludes this research.

Chapter 2

Basic concepts CMOS circuits and neural networks

This chapter will give a brief explanation of basic circuits and terminology necessary for the understanding of this work. Its start with the explanation of CMOS circuit structure, follow by the explanation of basic circuitry used for implementing the different sensory systems described in this thesis. In addition, the terminology used in the study of artificial and biological neural networks is explained.

2.1 CMOS circuits

In today's integrated circuits (IC) industry a good understanding of semiconductors devices is essential. In special the MOS transistor (MOSFET) that has become the most used semiconductor device today. Since late 1970's the MOSFET has been extremely popular,this is because, compared to other transistors they can be quite small and their manufacturing process is relatively simple. Furthermore, digital logic and analog designs can be implemented with circuits using only MOSFETs devices. They can be used as the building blocks of logic gates, fundamental in the design of digital circuits like microprocessors, in which transistors act as on-off switches. For analog circuits transistors respond to a continuous range of inputs with a continuous range of outputs. For these reasons, most very-large-scale integration (VLSI) circuits are made using MOS technology.

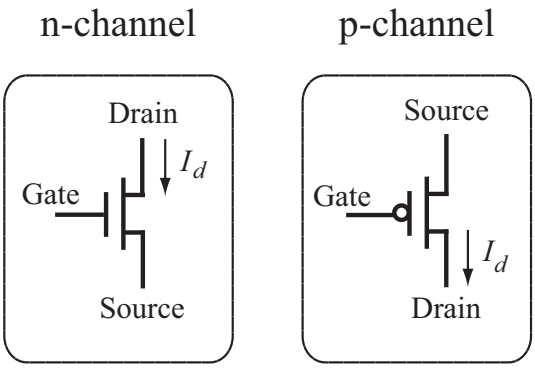


Figure 2.1: Symbols used to denote the MOSFET devices.

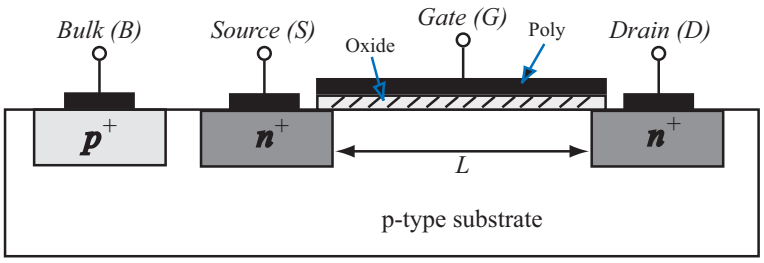


Figure 2.2: Simplified structure of an n-channel MOSFET.

2.1.1 The MOSFET

Symbology

Before explaining the operation of MOSFET, lets consider the symbology used to denote the devices. Figure. 2.1 shows the symbols used for a n-type and p-type MOSFET. It is important to note that a MOSFET is a four-terminal device, the symbols shown in the figure are the simplified model in which the bulk is connected to the source.

Structure

Figure 2.2, shows a simplified structure of an n-channel MOSFET (nMOS). The device is fabricated on a p-type substrate (also called "body"), it consists of two heavily doped n regions that form the source and the drain terminals, a thin layer of silicon dioxide (SiO_2) is insulating the gate from the substrate. Polysilicon (poly) operating as the gate terminal is deposited on top of the oxide.

It is important to note that the substrate forms pn junctions with the source and the drain regions. Since the drain will be at a positive voltage relative to the source (reverse-biased), the two pn junctions can be cut off by connecting the bulk terminal to the source.

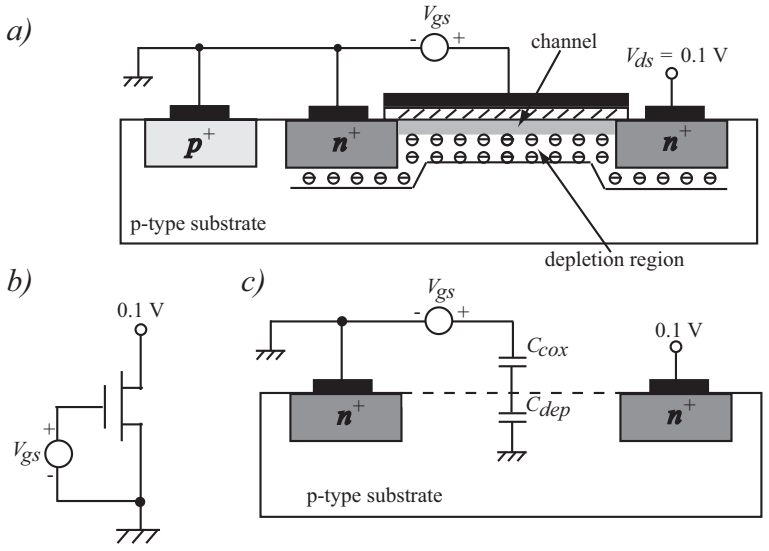


Figure 2.3: a) Structure of an nMOS transistor with gate voltage V_{gs}, b) corresponding symbology, and c) nMOS showing the depletion capacitance.

The structure of pMOS devices can be obtained by inverting all of the doping types. In practice nMOS and pMOS devices are fabricated on the same wafer, i.e., the same substrate. For this, one device type can be placed in a local substrate called "well".

Operation and I/V characteristic

Lets consider the Fig. 2.3 (a) the nMOS with the gate connected to an external voltage V_{gs}. The corresponding symbology is shown in Fig. 2.3(b). Since the gate and the substrate form a capacitance, when V_{gs} becomes more positive, holes in the p-substrate are repelled from the gate area however, for small V_{gs}, the voltage is not positive enough to attract a large number of electrons creating a depletion region. Under this condition, no current flows. When V_{gs} increases, the width of the depletion region also increase. At this point, the structure resembles two capacitors in series, C_{cox} and C_{dep} as shown in Fig. 2.3 (c). The increase on V_{gs} also attracts electrons from the n^+ regions (source and drain) where they are abundant. Thus, a "*channel*" of charge carriers is formed under the gate oxide and the transistor is "turned on" as shown in Fig. 2.3 (a). The value of V_{gs} for which this occurs is called "threshold voltage" (V_{th}).

Now lets consider the voltage V_{ds}, shown in Fig. 2.3 (a). This voltage causes a current I_d to flow from drain to source. The magnitude of I_d depends on the density of electrons in the channel, and the density of electrons depends on the magnitude of V_{gs}. When $V_{gs} = V_{th}$ the channel is just formed, so the current is

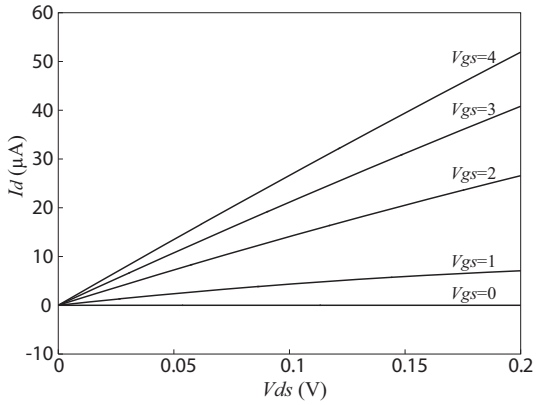


Figure 2.4: I_d - V_{ds} characteristic of an nMOS. (V_{ds} is small)

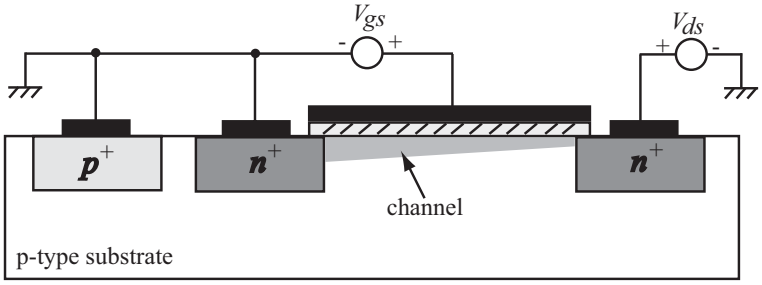


Figure 2.5: nMOS transistor channel after applying V_{ds}

still very small. When Vgs exceeds V_{th}, the channel charge density increases, (the channel's width increases). As a result, the conductance of the channel increases. The conductance of the channel is proportional to the $effective\ voltage$ ($V_{gs} - V_{th}$). The current I_d is proportional to this voltage ($V_{gs} - V_{th}$) and to the voltage V_{ds} that causes I_d to flow. Figure 2.4 shows the I_d - V_{ds} characteristic of an nMOS transistor, with small V_{ds} for various values of V_{gs}. It can be observed that the MOSFET operates as a linear resistance whose value is controlled by V_{gs}. When V_{gs} is small, the resistance is big, and as V_{gs} increases the resistance decreases.

The channel's width depends on voltage V_{ds}, therefore, when V_{ds} increases the potential of the channel at the drain decreases ($V_{gs} - V_{ds}$). It can be observed in Fig. 2.5, that the channel is not longer uniform. If V_{ds} keep increasing, the channel reduces more and more and its resistance increases correspondingly. Eventually, when the channel potential at the drain is reduce to V_{th} ($V_{gs} - V_{ds} = V_{th}$) the channel width is almost zero, and the channel is said to be "pinched off".

Figure 2.6 shows the I_d - V_{ds} characteristic of an nMOS transistor. From

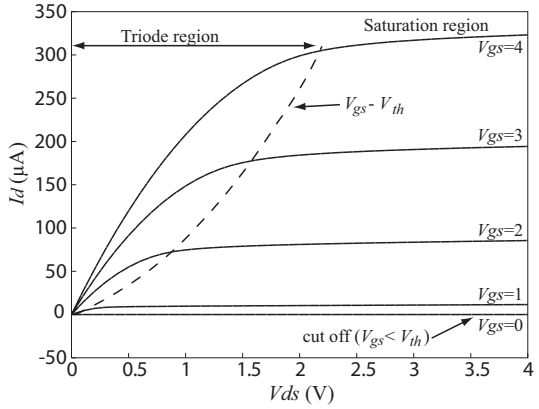


Figure 2.6: I_d - V_{ds} characteristic of an nMOS transistor.

there it can be indicated three regions of operation; the "$cut\ off$" region (also called *sub-threshold* or *weak inversion* region), the "*triode*" or *linear* region and the "*saturation*" region. The saturation region is used for the MOSFET to operates as an amplifier, for the operation as a switch, the cut off and the triode regions are used.

The MOSFET is cut off when $V_{gs} < V_{th}$. To operate in the triode region V_{gs} should be higher than V_{th} ($V_{gs} \geq V_{th}$) and V_{ds} should be kept small enough ($V_{gs} - V_{ds} > V_{th}$) so that the channel remains continuous.

In the triode region the I_d - V_{ds} characteristic can be describe by:

$$I_d = \beta[(V_{gs} - V_{th})V_{ds} - \frac{V_{ds}^2}{2}] \tag{2.1}$$

where β is the transconductance parameter given by:

$$\beta = KP_n \frac{W}{L}. \tag{2.2}$$

If V_ds is very small ($V_{ds} << 2(V_{gs} - V_{th})$), Eq. (2.1) can be expressed as:

$$I_d = \beta(V_{gs} - V_{th})V_{ds} \tag{2.3}$$

this linear relationship represents the operation of the MOS transistor as a linear resistor as shown in Fig. 2.4, with resistance R_d:

$$R_d = \frac{V_{ds}}{I_d} = [\beta(V_{gs} - V_{th})]^{-1} \tag{2.4}$$

whose value is control by V_{gs}.

The MOSFET operates in the saturation region when V_{gs} is grater than V_{th}

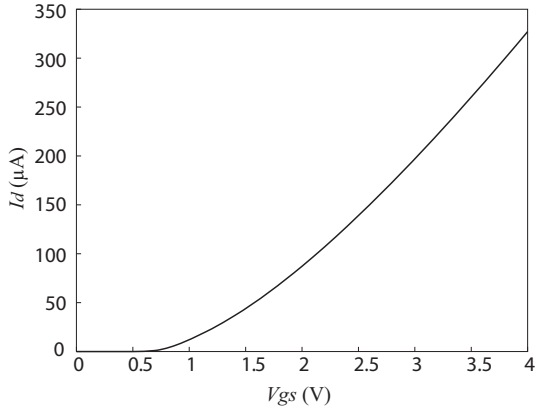


Figure 2.7: I_d - V_{gs} characteristic of an nMOS transistor.

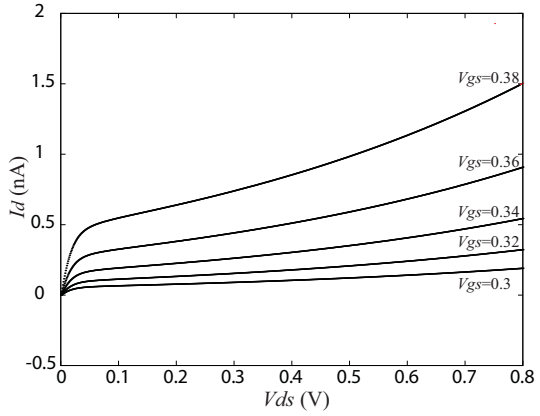


Figure 2.8: I_d - V_{ds} characteristic of an nMOS transistor operating in the sub-threshold region.

and the drain voltage V_{ds} does not fall below the gate voltage V_{gs} by more than V_{th}, ($V_{ds} \geq V_{gs} - V_{th}$). The saturation current can be expressed as:

$$I_d = \frac{\beta}{2}(V_{gs} - V_{th})^2. \tag{2.5}$$

In saturation the drain current I_d is independent of the drain voltage V_{ds}, instead is determined by the gate voltage V_{gs}. Figure 2.7 show the I_d - V_{gs} characteristic of an nMOS transistor. Thus, the MOSFET in saturation behaves as a current source whose value is controlled by V_{gs}.

2.1.2 Sub-threshold current

The current that flows when $V_{gs} < V_{th}$ is called sub-threshold, current. The MOSFET is said to be operating in the *weak inversion*, *cut off* or *sub-*

threshold region. This current is due to diffusion current between the drain and the source and is given by:

$$I_d = I_0 e^{V_g/\eta V_T}(e^{-V_s/V_T} - e^{-V_d/V_T}) \tag{2.6}$$

where η is the slope factor, V_T is the thermal voltage ($V_T = kT/q$), k is the Boltzmann's constant, T is the temperature, and q is the elementary charge. Current I_0 is given by:

$$I_0 = 2\eta\beta V_T^2 e^{-V_{th}/\eta V_T} \tag{2.7}$$

For small V_{ds} the transistor operates in the *triode region* (also called *linear region*) described by Eq. (2.8). In terms of V_{ds} this equation can be rewritten as:

$$I_d = I_0 e^{V_{gs}/\eta V_T}(1 - e^{-V_{ds}/V_T}) \tag{2.8}$$

As V_{ds} increases ($V_d > 4V_T$) the transistor operates in saturated region. The I-V relation in this region is described by:

$$I_d = I_0 e^{(V_g - V_s/\eta V_T)} \tag{2.9}$$

Figure 2.8 shows the Id-V_{ds} characteristic of a transistor operating in the saturation region.

2.1.3 Sub-threshold analog circuits

Since early 1980s, digital signal processing were becoming more powerful. The advance in IC technology provided compact, efficient implementation of circuits, so, many functions that were realized in the analog form were easily performed in the digital domain. However, in the past two decades, CMOS Technology has rapidly embraced the field of analog integrated circuits, providing low-cost, high-performance, rising this way the use of analog circuits. In addition, by careful use o the analog characteristics of transistors, arithmetic functions such as, addition, multiplication, exponential, logarithmic and tanh functions may be implemented using relatively few transistors as compare with digital circuits. Consequently, analog circuits have been proved fundamentally necessary for solving complex tasks, including the processing of natural stimuli. This chapter gives a brief explanation of commonly used analog circuits, including the current mirror, the differential pair and the transconductance amplifier.

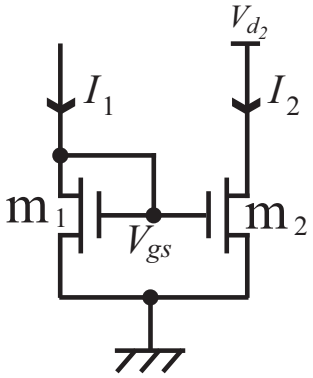


Figure 2.9: Current mirror circuit.

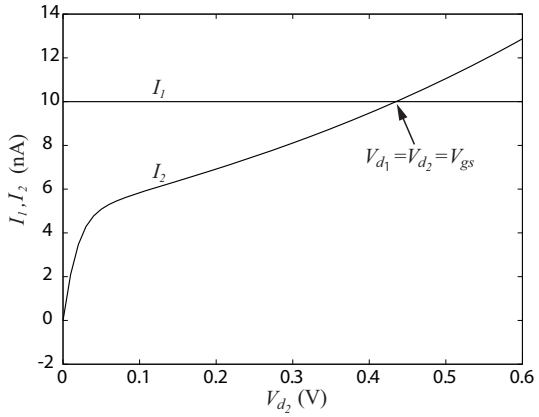


Figure 2.10: Current mirror simulation results.

Current mirror

The basic circuit of a current mirror is shown in Fig. 2.9. First lets us suppose that the two nMOS are identical. A current I_1 flows through transistor m_1, corresponding to the gate voltage, V_{gs_1}. Since the gates of m_1 and m_2 are connected, the gate voltage of m_2 is the same as the gate voltage of m_1, ($V_{gs_1} = V_{gs_2} = V_{gs}$). Ideally the same current that flows through m_1, flows through m_2. In other words, two identical MOS devices with equal gate voltages and operating in saturation carry equal currents.

The current I_1 is given by:

$$I_1 = I_{01} e^{(V_{gs_1})/\eta V_T}. \tag{2.10}$$

while the output current I_2 is:

$$I_2 = I_{02} e^{(V_{gs_2})/\eta V_T}. \tag{2.11}$$

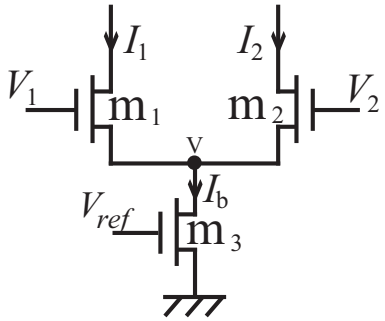


Figure 2.11: Schematic of the differential pair.

Since $V_{gs_1} = V_{gs_2}$ and $I_0 \propto \beta$ (Eq. (2.7)) the ratio of the currents can be written as:

$$\frac{I_2}{I_1} = \frac{\beta_2}{\beta_1} = \frac{W_2 L_1}{W_1 L_2} \tag{2.12}$$

This equation shows how to adjust the W/L ratio of the two devices to achieve the desired output current. By making $L_1 = L_2$, Eq. (2.12) is simplified as:

$$\frac{I_2}{I_1} = \frac{W_2}{W_1} \tag{2.13}$$

SPICE simulation results are shown in Fig. 2.10.

Differential pair

The differential pair is one of the basic amplifying stages in amplifier design. The differential pair's output is represented as the difference between the voltage of its inputs. The differential pair circuit is shown in Fig. 2.11. It consists of tree nMOS transistors. Transistor m_3 is used as a current source. Normally the drain voltage V is large enough so that the drain current I_b is saturated and its value is controlled by the bias voltage V_{ref}. The current I_b is divided between transistor m_1 and m_2 depending on their gate voltages V_1 and V_2.

$$I_b = I_1 + I_2 \tag{2.14}$$

As explained in the previous sections, the saturated drain current is given by Eq. (2.8). Applying this expression to the current of transistors m_1 and m_2 (Fig. 2.11)

$$I_1 = I_0 e^{\kappa(V_1 - V)/V_T} \tag{2.15}$$

$$I_2 = I_0 e^{\kappa(V_2 - V)/V_T} \tag{2.16}$$

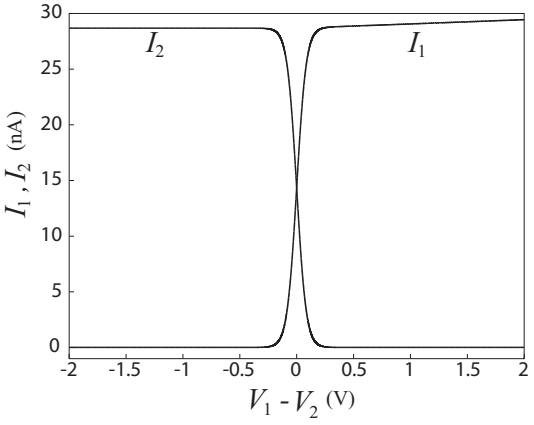


Figure 2.12: Differential pair output currents as function of $V_1 - V_2$.

So I_b can be expressed as:

$$I_b = I_0 e^{-V/V_T}(e^{\kappa(V_1)/V_T} + e^{\kappa(V_2)/V_T}) \tag{2.17}$$

Solving the equation:

$$e^{-V/V_T} = \frac{I_b}{I_0}\frac{1}{(e^{\kappa(V_1)/v_T} + e^{\kappa(V_2)/v_T})} \tag{2.18}$$

Substituting this expression in Eqs. (2.15) and (2.16) it is obtained:

$$I_1 = I_b \frac{e^{\kappa(V_1)/V_T}}{(e^{\kappa(V_1)/V_T} + e^{\kappa(V_2)/V_T})} \tag{2.19}$$

$$I_2 = I_b \frac{e^{\kappa(V_2)/V_T}}{(e^{\kappa(V_1)/V_T} + e^{\kappa(V_2)/V_T})} \tag{2.20}$$

If V_1 is higher than V_2, transistor m_2 is off, and all the current goes through m_1, ($I_1 \approx I_b$). The contrary is also true. Currents I_1 and I_2 as function of $V_1 - V_2$ are shown in Fig. 2.12.

Equations (2.19) and (2.20) can be expressed in terms of voltage difference $(V_1 - V_2)$ by subtracting them:

$$I_1 - I_2 = I_b(\frac{e^{\kappa(V_1)/V_T} - e^{\kappa(V_2)/V_T}}{e^{\kappa(V_1)/V_T} + e^{\kappa(V_2)/V_T}} \tag{2.21}$$

Then by multiplying and dividing by $e^{-(V_1+V_2)/2}$, it is obtained

$$I_1 - I_2 = I_b(\frac{e^{\kappa(V_1-V_2)/2V_T} - e^{-\kappa(V_1-V_2)/2V_T}}{e^{\kappa(V_1-V_2)/2V_T} + e^{-\kappa(V_1-V_2)/V_T}} \tag{2.22}$$

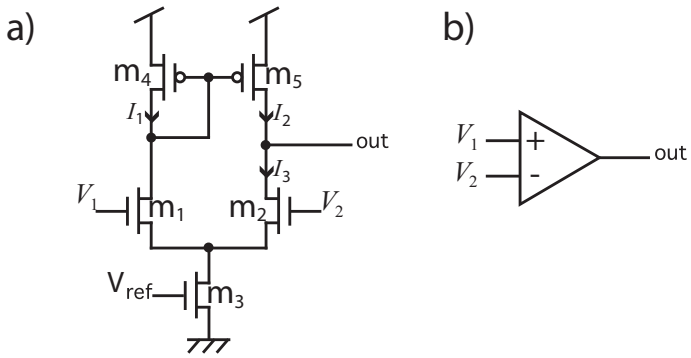


Figure 2.13: Transconductance amplifier, a)schematic, b) symbol.

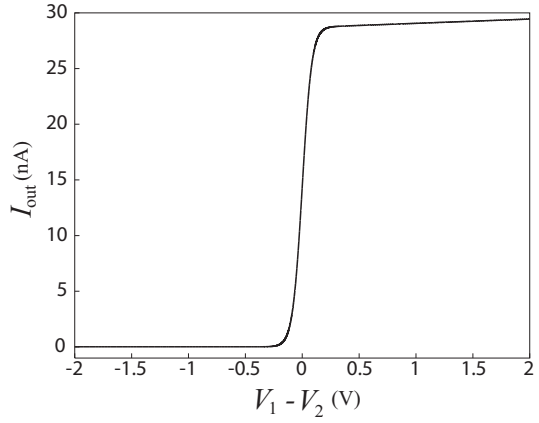


Figure 2.14: Transconductance amplifier output current

The r.h.s. of this equation can be express as tanh:

$$I_1 - I_2 = I_b \tanh \frac{\kappa(V_1 - V_2)}{2V_T} \tag{2.23}$$

Transconductance amplifier

The schematic of the transconductance amplifier is shown in Fig. 2.13 (a). The amplifier's symbol is shown in Fig. 2.13 (b). The circuit consists of a differential pair ($m_1 - m_2 - m_3$) and a current mirror ($m_4 - m_5$). Current I_1 is copied to I_2 by the current mirror. Thus, the output current will be the subtraction of the currents ($I_2 - I_3$).

The output current of a simple amplifier is shown in Fig. 2.14. The curve is very close to a (tanh) as expected, from the explanations of differential pairs circuits in the previous subsection.

2.2 Introduction to neural networks

Building intelligent systems to mimic biological systems, in particular neural systems has capture the attention of the world for years. So, it is not surprising that a technology such as neural networks has generated great interest. The human brain is a complex, non-linear and highly parallel system (it can process incoming stimuli simultaneously) that can easily outperform any existing computer. The brain has many features desirable in artificial systems:

- The brain is flexible. It can adjust to new environment by "learning".
- It is robust and fault tolerant. Nerve cell die every day without affecting its performance significantly.
- it is non-linear and highly parallel.
- It can deal with information that is fuzzy, probabilistic, noisy or inconsistent.

The advantage of parallel processing is that it allows the brain to simultaneously identify different stimuli which in consequence allows for quick and decisive actions. The brain can solve complex problems that are hardly approachable with traditional computers. A good example is the processing of visual information, even a baby is much better and faster at recognizing objects, faces, etc., than the most advance computer. While the computer's speed is a million times faster than a human's neural network, the brain have a large number of processors compared to computers.

Biological neural networks provide the best source of knowledge for developing powerful engineering neural networks.

2.2.1 Neurons

The brain contains many billions (about 10^{11}) of nerve cells or "neurons". Neurons are organized into a very complicated intercommunicating network. Typically each neuron is physically connected to tens of thousands of others neurons. Using these connections neurons can pass electrical signals between each other.

Individual neurons are complicated. They have a myriad of parts, subsystems, and control mechanisms. There are over one hundred different classes of neurons, depending on the classification method used. The artificial neural networks try to replicated the most basics elements of this complicated and powerful organism.

Figure 2.15 shows the schematic of a typical and simple neuron. The cell body or soma is the central part of the neuron, connected to the cell body are

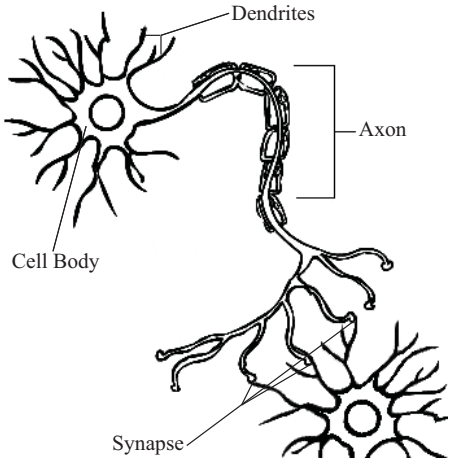


Figure 2.15: Typical neuron.

the dendrites, cellular extensions with many branches, and metaphorically this overall shape and structure is referred to as a dendritic tree. This is where the majority of input to the neuron occurs. Extending from the cell body is the axon, a finer, cable-like projection, which eventually arborizes into strands and substrands. The axon carries nerve signals away from the cell body. The axon terminal contains synapses or synaptic junctions transmitting the signal to other neuron's dendrites or cell bodies.

The transmission of a signal from one cell to another at a synapse is a complex chemical process, in which ion channels allow ions (sodium Na^+, calcium Ca^+, and chloride Cl^-) to move into and out of the cell. Ion channels control the flow of ions across the cell membrane by opening and closing in response to voltage changes and both internal and external signals. The membrane potential is the difference in electrical potential between the interior of a neuron and the surrounding extracellular medium. Current flowing into the cell changes the membrane potential to less negative or more positive values. If the membrane potential rises above a threshold level, a positive feedback process is initiated, and the neuron generates an action potential of fixed strength and duration. It is said then that the cell has "fired". After firing, the cell has to wait for a time called "refractory period" before it can fire again. For more information refer to [9]

2.2.2 Artificial neural networks

As mention in the previous section, the brain contains billions of neurons. Each neuron is connected to thousands of others neurons. Through these connections

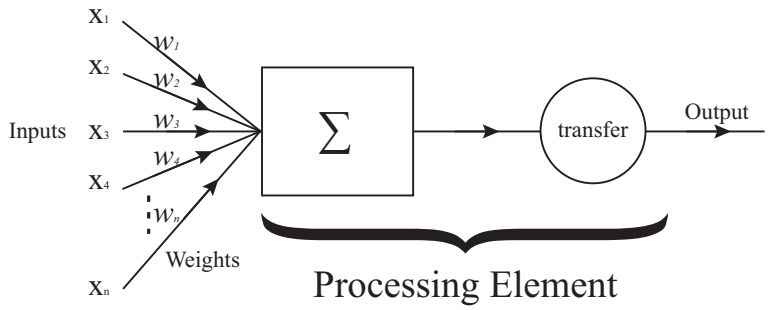


Figure 2.16: Basic artificial neuron.

(synapses) neurons can pass electrical signal between each other. These synaptic connections have varying strength which allows the influence of a given neuron on one of its neighbors to be strong, weak or just do nothing. Many aspects of brain function, particularly the learning process, are closely associated with the adjustment of these connections strengths. Brain activity is then represented by particular patterns of firing activity among this network of neurons. So, it is this simultaneous cooperative behavior of many simple processing units the source of the enormous computational power of the brain.

Artificial neural networks are electronic models based on the neural structure of the brain. Neural networks consists of many simple processing elements (PE) and weighted connections. These processing elements operate in parallel to solve specific problems. Their functions are determined by the networks structure, connections strengths and the processing performed by each element.

Neural networks can be though as devices that accept inputs and produce outputs. Basically, a biological neuron receives inputs from other sources through synapses of other neurons, then the soma combines them in some way, performs a generally non-linear operation on the result and finally outputs the final result through the axon and the synapse. Even when there are many variations of neuron, all natural neurons have the same four basic components. An artificial neuron simulate the four basic functions of a natural neuron. Figure 2.16 shows a fundamental representation of an artificial neuron.

In the model shown in Fig. 2.16, various inputs to the neuron are represented by X_i $(i = 0, 1, ..., n)$. Each of these inputs are multiplied by a connection weight w_i. In the simplest case, these products are summed, and pass through a transfer function to generate the result, and then the output. This electronic implementation is possible with other networks structures which utilize different summing functions as well as different transfer functions.

Connections

The weights w_i in Fig. 2.16 for $(i = 0, 1, ..., n)$ represent the strength of the synaptic connections from neuron ith. The connections define the flow through the network and modulates the amount of information passing between to the processing element.

Connections weights are adjusted during the learning process that capture the information. Connections weights that have positive values are "*excitatory*" connections. Connections weights with negative values are "*inhibitory*" connections. And those connections with a zero value are the same as not having connections present. By allowing a subset of all the possible connections to have nonzero values, sparse connectivity between processing elements (PEs) can be stimulated, because it is often desirable for a PE to have a internal bias value (threshold value).

Processing elements

The PE is the portion of the neural network where all the computing is performed. There are two important qualities that a PE must possess:

- PEs require only local information. The information necessary for a processing element to produce an output value must be present at the inputs and resides within the PE.
- PEs produce only one output value.

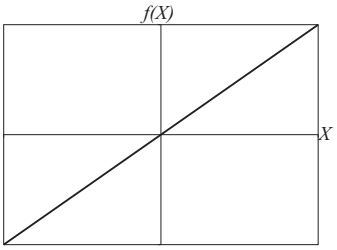
These two qualities allow neural networks to operate in parallel. Mathematically, the output of a PE is a function of its inputs and its weights.

$$Y = F(X, w_i). \tag{2.24}$$

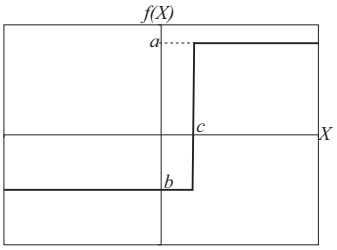
2.2.3 Processing elements transfer function

PEs transfer functions, also referred as activation functions can change the behavior of the network. Although the number of PE transfer functions possible is infinite, five are regularly employed by the majority of neural networks:

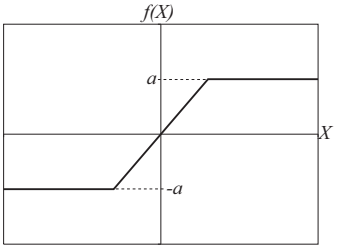
- Linear function
- Step function
- Ramp function
- Sigmoid function
- Gaussian function



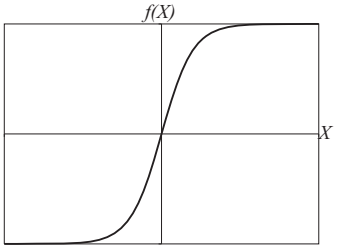
(a) Linear Function



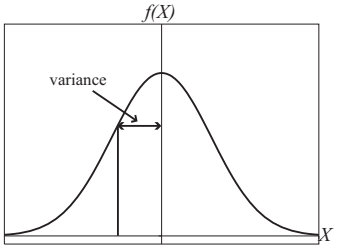
(b) Step Function



(c) Ramp Function



(b) Sigmoid Function



(d) Gaussian Function

Figure 2.17: Processing elements (PE) transfer functions.

With the exception of the linear function, all of these functions introduce a nonlinearity in the network dynamics by bounding the output value within a fixed range. Each function is shown in Fig. 2.17 (a-e).

Linear function

The linear function Fig. 2.17 (a), produces a linear output from the input X according to:

$$f(X) = \alpha X \tag{2.25}$$

where the X ranges over the real numbers and α is a positive scalar. If $\alpha = 1$ is equivalent as removing the transfer function.

Step function

The step function Fig. 2.17 (b) it produces two values a and b. If the input X is higher than a predefined value c (threshold value) the function produce the value a; otherwise will produce the value b, where a and b are scalars.

$$f(X) = \begin{cases} a, & \text{if } x \geq c \\ b, & \text{if } x \leq c \end{cases} \tag{2.26}$$

A particular step function, the "unit step function" or "Heaviside step function" is a discontinuous function whose value is zero for negative argument and one for positive argument

$$f(X) = \begin{cases} 1, & \text{if } x \geq 0 \\ 0, & \text{otherwise} \end{cases} \tag{2.27}$$

this kind of functions is common in neural networks, and have been implemented in models like the McCulloch and Pitts [1], and the Hopfield neural network [2].

Ramp function

The ramp function Fig. 2.17 (c). It can be though as a combination of the step function and the linear function. The functions has an upper and a lower bound and posses a linear response between the bounds.

$$f(X) = \begin{cases} a, & \text{if} X \geq a \\ X, & \text{if} |X| < a \\ -a, & \text{if} X \leq a \end{cases} \tag{2.28}$$

Sigmoid function

The sigmoid function Fig. 2.17 (d). Is a continuous version of the ramp function. Is a mathematical function that produces a sigmoid curve (S-shape curve). Sigmoid functions are often used in neural networks to introduce nonlinearity in the model and/or to bound signals to within a specified range. A popular neural net element computes a linear combination of its input signals, and applies a bounded sigmoid function to the result; this model can be seen as a "smoothed" variant of the classical threshold neuron.

$$f(X) = \frac{1}{1 + e^{-\alpha X}} \tag{2.29}$$

where $\alpha > 0$ provides an output from 0 to 1. It is important to note that there is a relationship between Eq. 2.27 and Eq. 2.29. When $\alpha = \infty$ in Eq. 2.29, the slope of the sigmoid function between 0 and 1 become step, and in effect become the Heaviside function.

Two alternatives to the sigmoid functions are the hyperbolic tangent

$$f(X) = tanh(X) \tag{2.30}$$

which ranges from -1 to 1, and the augmented ratio of squares

$$f(X) = \begin{cases} \frac{X^2}{1+X^2}, & \text{if} x > 0 \\ 0, & \text{otherwise} \end{cases} \tag{2.31}$$

which ranges from 0 to 1.

Sigmoid functions are very suitable for implementation in analog VLSIs because they can be implemented by using differential-pair circuits.

Gaussian function

The gaussian function Fig. 2.17 (e). Is a radial function that requires a variance value $v > 0$.

$$f(X) = \alpha e^{-\frac{(X-b)^2}{v}} \tag{2.32}$$

where α is the height of the Gaussian peak, b is the position of the center of the peak, and v is the variance which controls the width of the "bump".

2.2.4 Learning

Learning is one of the most important features on neural networks. Since all knowledge is encode in weights, "learning" is define as a change in connection

weight values.

On the network level, a weight represents how frequent the receiving unit has been activated simultaneously with the sending unit. Hence, weight change between two units depends on the frequency of both neurons firing simultaneously. In other words, the weight between two neurons will increase if the two neurons activate simultaneously; it is reduced if they activate separately. This form of weight change is called "*Hebbian learning*" [10], which provides a simple mathematical model for synaptic modification in biological networks. Its most general form is expressed as:

$$\Delta w_{i,j} = x_i x_j \tag{2.33}$$

or the change in the ith synaptic weight $w_{i,j}$ is the product of the output of unit i and unit j. Several modifications but the basic principle still accepted.

There are several ways of learning techniques. The most important are:

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

Supervised learning

Supervised learning is a process that incorporates an external teacher, it requires sample input-output pairs from the function to be learned, the data are available and are used to calculate weight change. In other words, supervised learning requires a set of questions with the right answers.

Supervised learning is further classified into two subcategories: structural learning and temporal learning. Structural learning is concerning with finding the best possible input/output relationship for each individual pattern pair. Temporal learning is concerned with capturing a sequence of patterns necessary to achieve some final outcome. In temporal learning, the current response of the network is dependent on previous inputs and responses.

Unsupervised learning

Unsupervised learning, also called self-organization, is a process that does not require external teacher, it relies upon local information during the entire learning process. Unsupervised learning organizes presented data and discovers its emergent collective properties. Examples of unsupervised learning include, Hebbian learning and competitive learning.

Reinforcement learning

Reinforcement learning is the problem faced by an agent that must learn behavior through trial-and-error interactions with a dynamic environment. Reinforcement learning differs from the supervised learning problem in that correct input/output pairs are never presented, nor sub-optimal actions explicitly corrected. In other words the learner is not told which actions to take, but instead must discover which actions yield the most reward by trying them.

2.3 Summary

This section gave a brief explanation of basic concepts and terminology regarding CMOS circuits as well as artificial neural networks. For a comprehensive explanation, I would refer the reader to "CMOS *circuit design*, *layout and simulations*" [11], *Theoretical Neuroscience* [9], *Introduction to the theory of Neural Computation* [12] and Artificial Neural Networks [13].

Chapter 3

Temperature receptor circuit

Sensory system is a part of the nervous system responsible for processing sensory information. The sensory system detects, transforms, transfers and processes stimuli from the environment. It consists of **sensory receptors** (that receive and transform stimuli from the external environment), neural pathway (that transfer the information to the brain), and parts of the brain (that processes the information).

The sensory receptors are specialized endings of afferent neurons (sensory neurons), or separate cells that affect ends of afferent neurons. They function as the first component in the sensory system. When activated by stimuli, sensory receptors collect information about external and internal environment. In response to the stimuli, the sensory receptor initiates sensory transduction, a process by which the physical energy of the stimuli is converted into electrical impulses that are later transferred to the brain.

Each sensory receptor responds primarily to a single kind of stimulus, and they are often classified into four categories. 1) *Mechanoreceptors* detect changes in pressure, position, or acceleration; include receptors for touch, hearing and joint position. 2) *Thermoreceptors* detect changes in the temperature. 3) *Chemoreceptors* detect ions or molecules; include receptors for olfaction and taste. 4) *Photoreceptors* that respond to light (vision).

This chapter focuses on the implementation of *Thermoreceptors* (specialized neurons which are designed to be sensitive to changes in temperature). Thermoreceptors are found all over the body, in the skin to provide the brain with information about environmental temperature, and inside the body they are part of the body's complex and interconnected series of systems which are

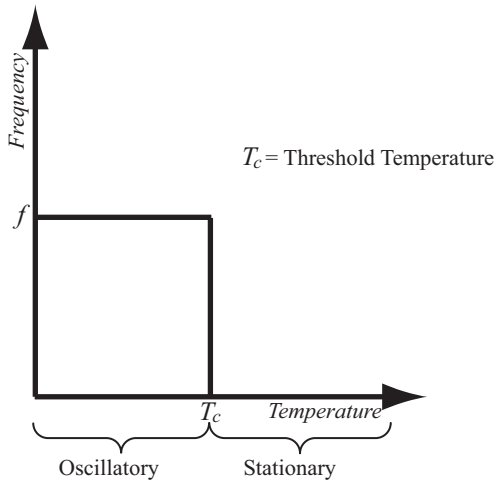


Figure 3.1: Temperature Receptor operation model.

designed to keep the body in balance.

It is important to note that every sensory system has a *threshold*. In other words, there is a minimum amount of the physical stimulus needed for the sensory system to elicit a response. If the stimulus is too small (under the sensory neuron's threshold; sub-threshold) no pulse is generated and the stimulus is not perceived. Therefore, the key idea is to use excitable circuits for implementing this kind of systems. There are some studies based on the respond of excitable neurons to temperature changes. A temperature increase causes a regular and reproducible increase in the frequency of the generation of pacemaker potential in most *Aplysia* and *Helix* excitable neurons [14]. The Br neuron shows its characteristic bursting activity only between 12 and 30°C. Outside this range, the burst pattern disappears and the action potentials become regular.

In this chapter, a sub-threshold CMOS circuit that changes its dynamical behavior (i.e., oscillatory or stationary behaviors) around a given threshold temperature is proposed. The threshold temperature can be set to a desired value by adjusting an external bias voltage. The circuit consists of two pMOS differential pairs, two capacitances, and two resistors with low temperature dependence.

The circuit operation was fully investigated through theoretical analysis, extensive numerical simulations and circuit simulations using the Simulation Program of Integrated Circuit Emphasis (SPICE). Moreover, the operation of the proposed circuit was demonstrated experimentally using discrete MOS devices.

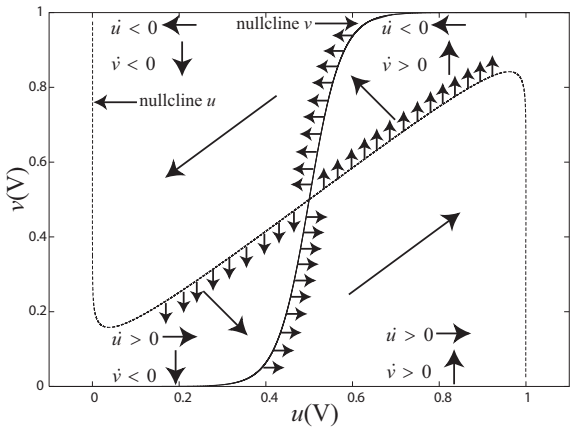


Figure 3.2: u and v nullclines with vector field direction.

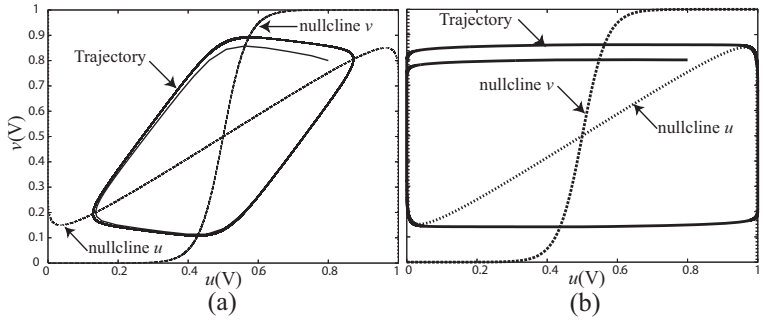


Figure 3.3: Trajectory when a) $\tau = 1$ and b) $\tau << 1$.

3.1 The model

The temperature receptor's operation principle is shown in Fig. 3.1. The model consists of a nonlinear neural oscillator that changes its operation frequency when it receives an external perturbation (temperature). There are many models of excitable neurons, but only a few of them have been implemented on CMOS LSIs, e.g., silicon neurons that emulate cortical pyramidal neurons [15], FitzHugh-Nagumo neurons with negative resistive circuits [16], artificial neuron circuits based on by-products of conventional digital circuits [17] - [19], and ultralow-power sub-threshold neuron circuits [20]. Our model is based on the Wilson-Cowan system [21] because it is easy to both, analyze theoretically and implement in sub-threshold CMOS circuits.

The dynamics of the temperature receptor can be expressed as:

$$\tau \dot{u} = -u + \frac{\exp(u/A)}{\exp(u/A) + \exp(v/A)}, \tag{3.1}$$

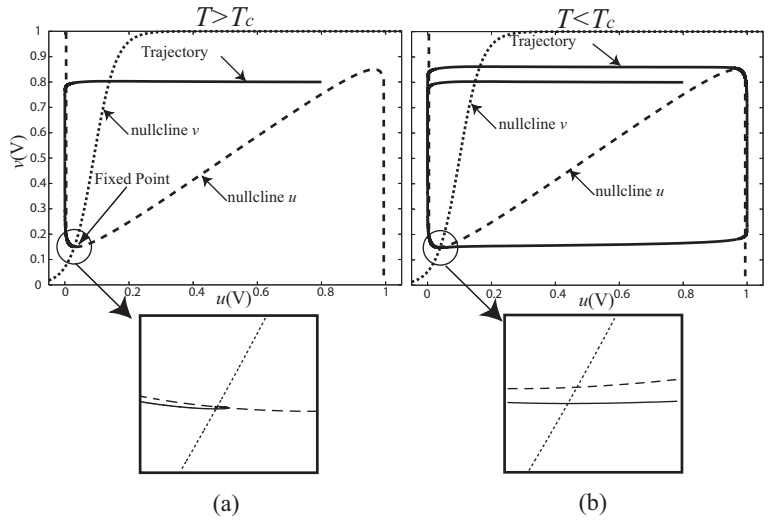


Figure 3.4: Nullclines showing the fixed point and the trajectory when a) system is stable b) system is oscillatory.

$$\dot{v} = -v + \frac{\exp(u/A)}{\exp(u/A) + \exp(\theta/A)}, \tag{3.2}$$

where τ represents the time constant, θ is an external input, and A is a constant proportional to temperature. The second term of the r.h.s. of Eq.(3.1) represents the sigmoid function, a mathematical function that produces an S-shaped (sigmoid) curve. The sigmoid function can be implemented in VLSIs by using differential-pair circuits, making this model suitable for circuit implementation.

To analyze the system operation, it is necessary to calculate its nullclines. Nullclines are curves in the phase space where the differentials $\dot{u}$ and $\dot{v}$ are equal to zero. The nullclines divide the phase space into four regions. In each region the vector field follows a specific direction. Along the curves the vector field is either completely horizontal or vertical; on the u nullcline ($\dot{u}$=0) the direction of the vector is vertical; and on the v nullcline ($\dot{v}$=0), it is horizontal. The u and v nullclines indicating the direction of vector field in each region are shown in Fig. 3.2.

When the vector field is plotted on the phase plane it is called *trajectory*. The trajectory of the system depends on the time constant τ, which modifies the velocity field of u. In Eq. (3.1), if τ is large, the value of u decreases, and for small τ, u increases. Figures 3.3(a) and (b) show trajectories when $\tau = 1$ and $\tau << 1$. In the case where $\tau << 1$, the trajectory on the u direction is much faster than that in the v, so only close to the u nullcline movements of vectors in vertical direction are possible.

Let us suppose that θ is set at a certain value where the threshold tem-

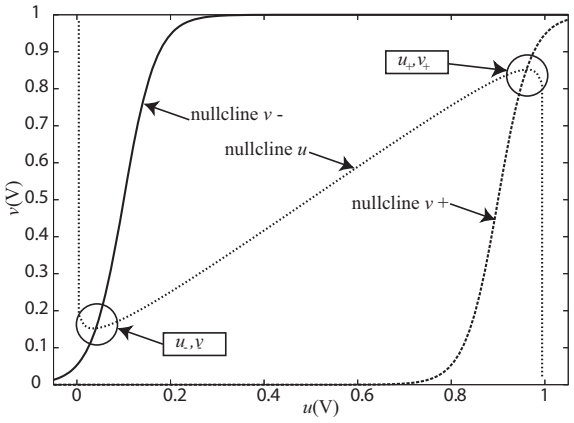


Figure 3.5: u and v local maximum and local minimum.

perature (T_c), which is proportional to A is 27°C. The threshold temperature represents the threshold temperature we desire to measure. When θ changes, the v nullcline changes to a point where the system will be stable as long as the external temperature is higher than T_c. This is true because the system is unstable only when the fixed point exists in a negative resistive region of the u nullcline. The fixed point, defined by $\dot{u} = \dot{v} = 0$ is represented in the phase space by the intersection of the u nullcline with the v nullcline. At this point the trajectory stops because the vector field is zero, and the system is thus stable. On the other hand, when the external temperature is below T_c, the nullclines move, and this will correspond to a periodic solution to the system. In the phase space we can observe that the trajectory does not pass through the fixed point but describes a closed orbit or limit cycle, indicating that the system is oscillatory. Figure 3.4 shows examples when the system is stable (a) and oscillatory (b). In (a) the external temperature is greater than the threshold temperature, hence, the trajectory stops when it reaches the fixed point, and the system is stable. In (b), where the temperature changes below the threshold temperature, the trajectory avoids the fixed point, and the system becomes oscillatory.

Deriving the nullclines equation ($\dot{u} = 0$) and equaling to zero, the local minimum (u_-, v_-) and local maximum (u_+, v_+) representing the intersection point of the nullclines are given by:

$$u_\pm = \frac{1 \pm \sqrt{1 - 4A}}{2}, \tag{3.3}$$

$$v_\pm = u_\pm + A \ln\left(\frac{1}{u_\pm} - 1\right), \tag{3.4}$$

The nullclines giving the local minimum and local maximum $(u_\pm, v_\pm)$ are shown

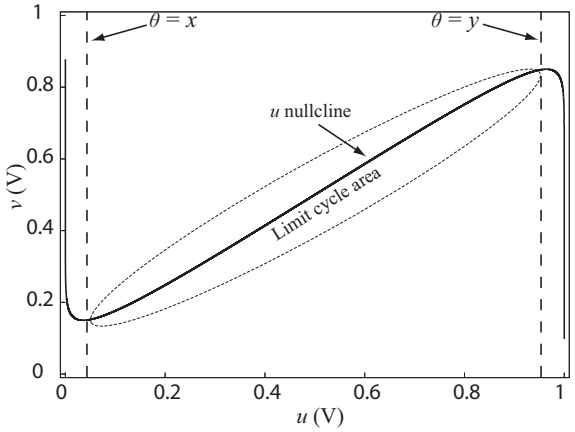


Figure 3.6: Threshold values x and y showing the area where the system is oscillatory.

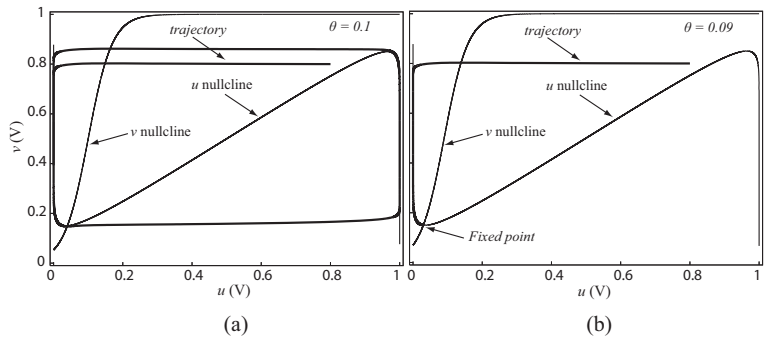


Figure 3.7: Nulclines and trajectories when a) $\theta = 0.1$ and b) $\theta = 0.09$.

in Fig. 3.5.

From the local minimum and maximum equations (Eq. (3.3) and Eq. (3.4)), the nullcline equation ($\dot{v} = 0$) and remembering that A is proportional to temperature, the relationship between θ and the temperature, can be written as:

$$\theta_{\pm} = u_{\pm} + A \ln\left(\frac{1}{v_{\pm}} - 1\right). \tag{3.5}$$

When $\tau << 1$ the trajectory jumps from one side to the other side of the u nullcline, so only along the u nullcline movement in the v direction are possible as shown in Fig. 3.3(b). It is necessary to emphasis this fact because this characteristic is necessary for the system operation; thus, $\tau << 1$ is assumed.

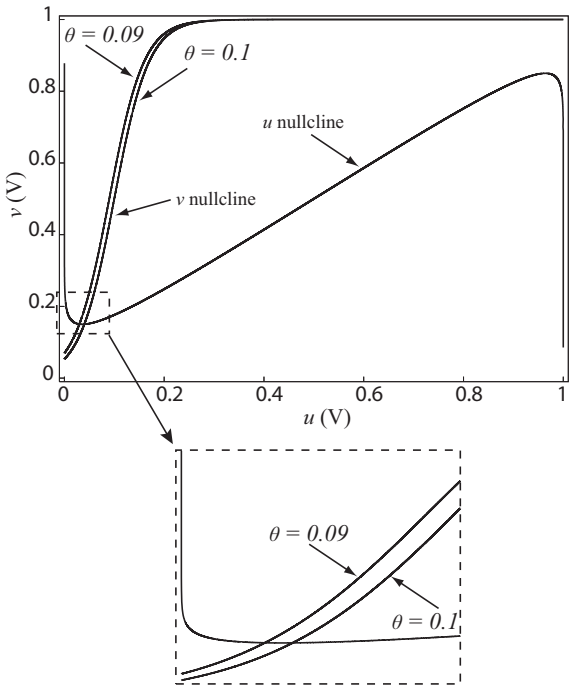


Figure 3.8: v nullcline when $\theta = 0.1$ and $\theta = 0.09$.

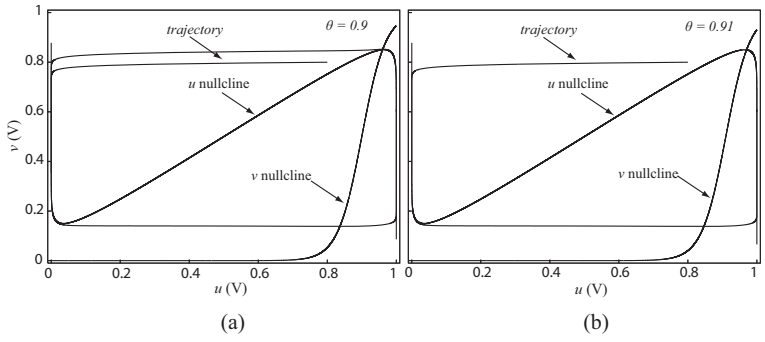


Figure 3.9: Nulclines and trajectories when a) $\theta = 0.9$ and b) $\theta = 0.91$.

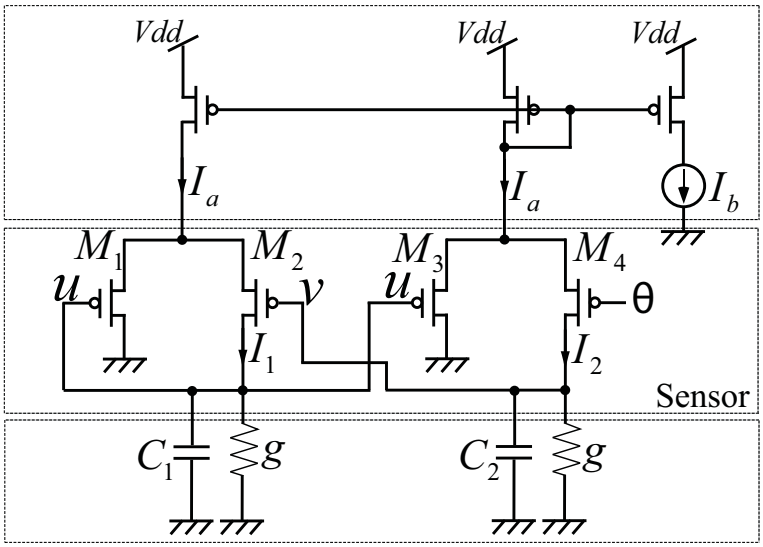


Figure 3.10: Temperature receptor circuit.

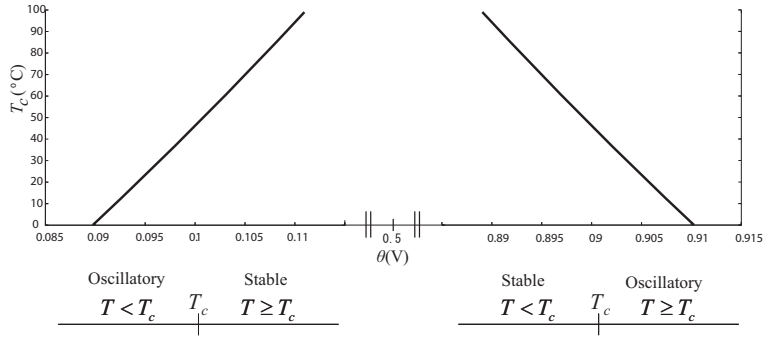


Figure 3.11: Relation between $\theta_{\pm}$ and T_c.

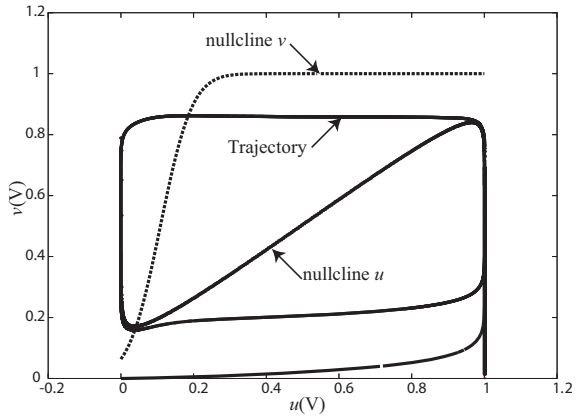


Figure 3.12: Trajectory and nullclines obtained through simulation results when the system is oscillatory.

3.1.1 Stability of the Wilson-Cowan system

Wilson and Cowan [21] studied the properties of a nervous tissue modeled by populations of oscillating cells composed of two types of interacting neurons: excitatory and inhibitory ones. The Wilson-Cowan system has two types of temporal behaviors, i.e. steady state and limit cycle. According with the stability analysis in [21], the stability of the system can be controlled by the magnitude of the all the parameters.

A simplified set of equations with and excitatory node u and an inhibitory node v, representing the Wilson-Cowan system are given by Eqs. (3.1) and (3.2). The nullclines of this system, which are pictured in Fig. 3.2, are given by:

$$v = u + A\ln(\frac{1}{u} - 1) \tag{3.6}$$

for the u nullcline (Eq. (3.1) $= 0$), and

$$v = \frac{e^{u/A}}{e^{u/A} + e^{\theta/A}} \tag{3.7}$$

for the v nullcline (Eq. (3.2) $= 0$).

For easy analysis, let us suppose that A is a constant. In such a case, there are some important observations for the stability of the system.

- There is a threshold value of θ bellow which the limit cycle activity can not occurs ($\theta < x$; see Fig. 3.6).
- There is a higher value of θ above which the system saturates and the limit cycle activity is extinguished ($\theta > y$).
- Between these two values the system exhibit limit cycle oscillation (area between θ's lower threshold x and θ's upper threshold y).

Let us suppose that the value of A is fixed to 0.03, in this cases, depending on the magnitude of the parameter θ (external input) the Wilson-Cowan oscillator will show different behaviors. Figure 3.6 shows the areas in which the system exhibits (or not) limit cycle activity. The threshold values x and y are shown in the figure.

The nullclines and trajectories for different values of θ are shown in Figs. 3.7 and 3.9. In Figure 3.7 (a), θ was set to 0.1, it can observed from the figure that the system is exhibiting limit cycle oscillations. Thus the system is unstable. When the value of θ is reduced to 0.09, show in Fig. 3.7 (b). It can be observed that the trajectory stops at the fixed point. The fixed point at this area is an attractor, i.e. a stable fixed point. Thus the system is stable. Figure 3.8 show

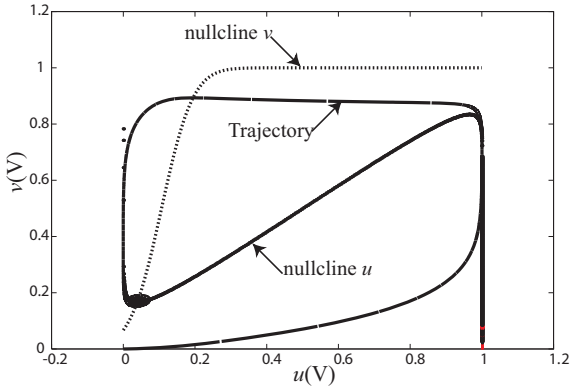


Figure 3.13: Simulation results when the system is stationary.

the position of the v nullclines when $\theta = 0.09$ and $\theta = 0.1$. The other case is shown is Fig. 3.9. In figure 3.9 (a) θ is set to 0.9, at this point the system is oscillatory. When θ is increased, ($\theta = 0.91$) the system is stable.

It could be observed that depending on the parameter θ (the external input) the stability of the system can be controlled. It is important to notice that the stability also depends on the magnitude of A, and that A is proportional to the temperature. These observations are the basis of the operation of the temperature receptor system. As explained before, the value of the input θ is set to a fix value; so, as temperature changes the system behavior also changes i.e. stable and oscillatory.

3.2 Circuit implementation

The temperature receptor circuit is shown in Fig. 3.10. The sensor section consists of two pMOS differential pairs ($M_1 - M_2$ and $M_3 - M_4$) operating in their sub-threshold region. External components are required for the operation of the circuit. These components consist of two capacitors (C_1 and C_2) and two temperature-insensitive off-chip metal-film resistors (g). In addition, for the experimental purpose, two current mirrors were used as the bias current of differential pairs. Note that for the final implementation of the temperature receptor a current reference circuit with low-temperature dependence [22] should be used.

Differential-pairs sub-threshold currents, I_1 and I_2, are given by [23]:

$$I_1 = I_a \frac{\exp(\kappa u/v_T)}{\exp(\kappa u/v_T) + \exp(\kappa v/v_T)}, \tag{3.8}$$

$$I_2 = I_a \frac{\exp(\kappa u/v_T)}{\exp(\kappa u/v_T) + \exp(\kappa\theta/v_T)}, \tag{3.9}$$

where I_a represents the differential pairs bias current, v_T is the thermal voltage ($v_T = kT/q$), k is the Boltzmann's constant, T is the temperature, and q is the elementary charge.

The circuit dynamics can be determined by applying Kirchhoff's current law to both differential pairs, which is represented as follows:

$$C_1\dot{u} = -gu + \frac{I_a \exp(\kappa u/v_T)}{\exp(\kappa u/v_T) + \exp(\kappa v/v_T)}, \tag{3.10}$$

$$C_2\dot{v} = -gv + \frac{I_a \exp(\kappa u/v_T)}{\exp(\kappa u/v_T) + \exp(\kappa\theta/v_T)}, \tag{3.11}$$

where κ is the sub-threshold slope, C_1 and C_2 are the capacitances representing the time constants, and θ is bias voltage.

Note that Eqs. (3.10) and (3.11) correspond to the system dynamics (Eqs. (3.1) and (3.2)) previously explained. Therefore, applying the same analysis, the local minimum (u_-, v_-) and local maximum (u_+, v_+) for the circuit equations can be calculated, expressed by:

$$u_\pm = \frac{I_a/g \pm \sqrt{(I_a/g)^2 - 4v_T I_a/(\kappa g)}}{2}, \tag{3.12}$$

$$v_\pm = u_\pm + \frac{v_T}{\kappa}\ln\left(\frac{I_a}{gu_\pm} - 1\right), \tag{3.13}$$

and the relationship between the external bias voltage (θ) and the external temperature (T):

$$\theta_\pm = u_\pm + \frac{v_T}{\kappa}\ln\left(\frac{I_a}{gv_\pm} - 1\right). \tag{3.14}$$

where the relation with the temperature is given by the thermal voltage defined by $v_T = kT/q$. At this point the system temperature is equal to the threshold temperature which can be obtained from:

$$T_c = \frac{q\kappa(\theta_\pm - u_\pm)}{k\ln\left(\frac{I_a}{gv_\pm} - 1\right)}. \tag{3.15}$$

The threshold temperature T_c can be set to a desired value by adjusting the external bias voltage (θ). The circuit changes its dynamic behavior, i.e., oscillatory or stationary behaviors, depending on its operation temperature and bias voltage conditions. At temperatures lower than T_c the circuit oscillates, but the circuit is stable (does not oscillate) at temperatures higher than T_c.

Figure 3.11 shows the relation between the bias voltage $\theta_{\pm}$ and the threshold temperature T_c with $\kappa = 0.75$; θ_- for u and v local minimums and θ_+ for u and v local maximums. When θ_- is used to set T_c, the system is stable at external temperatures higher than T_c; while when θ_+ is used, the system is stable when the external temperature is lower than T_c and oscillatory when it is higher than T_c.

3.3 Simulations and experimental results

Circuit simulations were conducted by setting C_1 and C_2 to 0.1 pF and 10 pF, respectively, g to 1 nS, and reference current (I_b) to 1 nA. Note that for the numerical and circuit simulations, two current sources were used instead of the current mirrors. The parameter sets used for the transistors were obtained from MOSIS AMIS 1.5-μm CMOS process. Transistor sizes were fixed at $L = 40$ μm and $W = 16$ μm. The supply voltage was set at 5 V. Figure 3.12 shows the nullclines and trajectory of the circuit with the bias voltage (θ) set at 200 mV and the external temperature (T) set at 27°C; the system was in oscillatory state. Figure. 3.13 shows the nullclines when the system is stationary with the bias voltage (θ) set at 90 mV.

The output waveform of u for different temperatures is shown in Fig. 3.14. The bias voltage θ was set to 120 mV, when the external temperature was 20°C the circuit was oscillating, but when the temperature increases up to 40°C the circuit becomes stable. Figure 3.15 shows the simulated oscillation frequencies of the circuit as a function of the temperature with the bias voltage set to 120 mV. The frequency was zero when the temperature was above the threshold temperature $T_c = 36$°C, and the frequency increases at temperatures lower than T_c.

Through circuit simulations, by setting the values for the threshold temperature (T_c) and changing the bias voltage (θ) until the system changed its state, it was established a numerical relation between T_c and θ. When comparing this relationship between θ and T_c obtained through different methods, it was found that there is a mismatch between the numerical simulations and the circuit simulations. This difference might be due to the parameters that are included in the SPICE simulation but omitted in the numerical simulation and theoretical analysis. Many of these parameters might be temperature dependent; thus, their value changes with temperature, and as a result of this change, the T_c characteristic changes. The difference between the two simulations is shown in Fig. 3.16

The temperature receptor's operation was successfully demonstrated using discrete MOS circuits. Parasitic capacitances and a capacitance of 0.033 μF

were used for C_1 and C_2 respectively, and the resistances (g) were set to 10 MΩ. The input current (I_b) for the current mirrors was set to 100 nA and an output current (I_a) was measured to be of 78 nA.

Measurements were performed at room temperature ($T = 23°$C). With the bias voltage (θ) set to 500 mV the voltages of u and v were measured. Under these conditions, the circuit was oscillating. The voltages of u and v for different values of θ were also measured. The results showed that for values of θ lower than 170 mV, the circuit did not oscillate (was stable), but that for values higher than 170 mV, the circuit became oscillatory. Figures 3.17 and 3.18 shows the oscillatory and stable states of u and v with θ set to 170 and 150 mV, respectively.

In addition, the nullclines (steady state voltage of the differential pairs) were measured. The v nullcline (steady state voltage v of differential pair $M_3 - M_4$) was measured by applying a variable DC voltage (from 0 to 1 V) on u and measuring the voltage on v. For the measurement of the u nullcline (steady state voltage u of differential pair $M_1 - M_2$), a special configuration of the first differential pair of the circuit was used. Figure 3.19 shows the circuit used for the u nullcline measurement. A variable DC voltage was applied (from 0 to 1 V) on v. For each value of v the voltage on u_1 was changed (from 0 to 1) and then measured the voltage on u_o and u_1. This enabled us to obtain the u nullcline by plotting the points where u_o and u_1 had almost the same value. In this way, it was obtained a series of points showing the shape of the u nullcline. The series of points was divided into three sections, and the average was calculated to show the u nullcline. Figure 3.20 shows the u nullcline divided into the three sections used for the average calculation. The trajectory and nullclines of the circuit with θ set to 500 mV are shown in Fig. 3.21.

Notice that in the experimental results there is a difference in the amplitude of the potentials u and v with respect to results obtained from the numerical and circuit simulations. This is due to the difference in the bias current of the differential pairs. From Eqs. (3.12) and (3.13), it can observed that by making g and I_b (used in numerical and circuit simulations) the same value, they cancel each other out; however, the output currents of the current mirrors were in the order of 78 nA, and g was set to 100 nS. This difference caused the decrease in the potentials amplitudes, as shown in Figs.3.12 and 3.21.

Measurements performed at different temperatures were made. The bias voltage (θ) was set to a fixed value and the external temperature was changed to find the value of the threshold temperature (T_c) where the circuit changes from one state to the other. With the bias voltage θ set to 170 mV at room temperature ($T = 23°$C), the circuit oscillated. When the external temperature was increased to ($T = 26°$C), the circuit changed its state to stationary (did not

oscillate). Once again, when the external temperature was decreased one degree ($T = 25°$C), the circuit started to oscillate; therefore, the threshold temperature was $T_c = 26°$C. Measures of the threshold temperature (T_c) for different values of the bias voltage (θ) were made.

In order to compare experimental results with, SPICE results and theoretical ones, the actual κ (sub-threshold slope) of the HSPICE model was measured and found to be in the order of 0.61. The threshold temperature for each value of θ obtained experimentally compared with the threshold temperature obtained with theoretical analysis using Eq. (3.14) (with $\kappa = 0.61$) is shown in Fig. 3.22. The curves have positive slopes in both cases. This is because the temperature difference between one value of bias voltage and the other decreases as the bias voltage increases. For θ= 140 and 150 mV the experimentally obtained threshold temperatures (T_c) are 0°C and 13°C, respectively, a difference of 13°C. For θ= 240 and 250 mV the threshold temperatures (T_c) are 54°C and 56°C, respectively: a difference of only 2°C.

The difference between the experimental, HSPICE, theoretical results is due to the leak current caused by parasitic diodes between the source (drain) and the well or substrate of the discrete MOS devices, and the mismatch between the MOS devices. In addition, because of the leak current, when temperature increases, the stable voltages of u and v also increase. Figures 3.23 and 3.24 shows the stationary state with θ set to 140 mV and temperature set to 23 and 75°C, respectively.

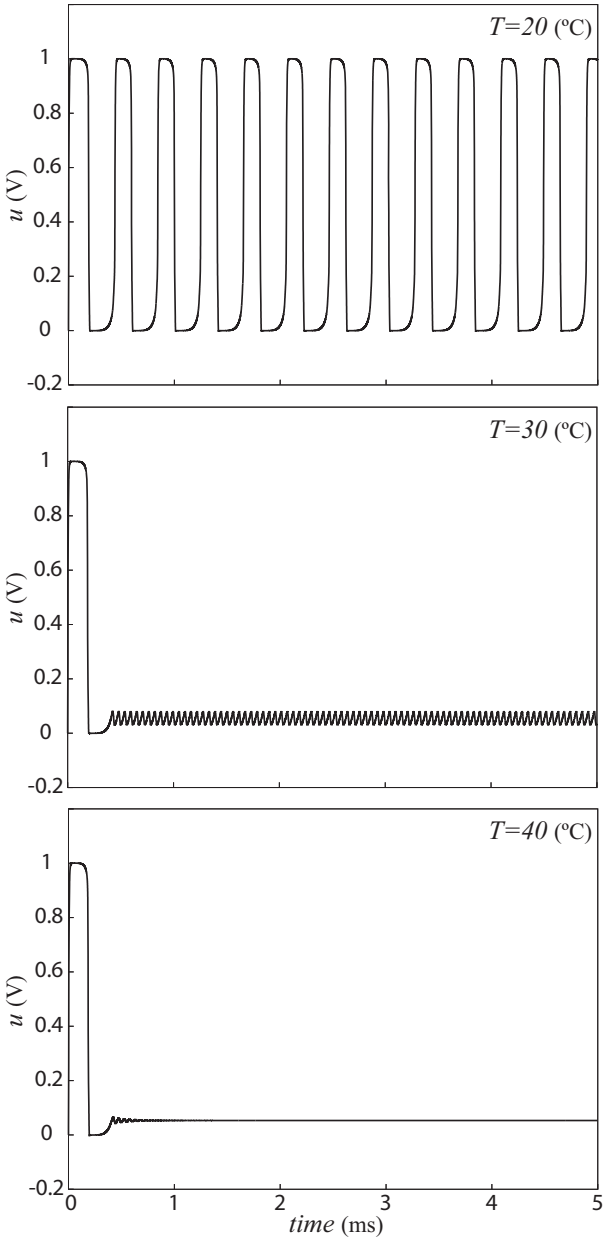


Figure 3.14: Waveform of u at different temperatures (from $T = 20°\text{C}$ to $T = 40°\text{C}$).

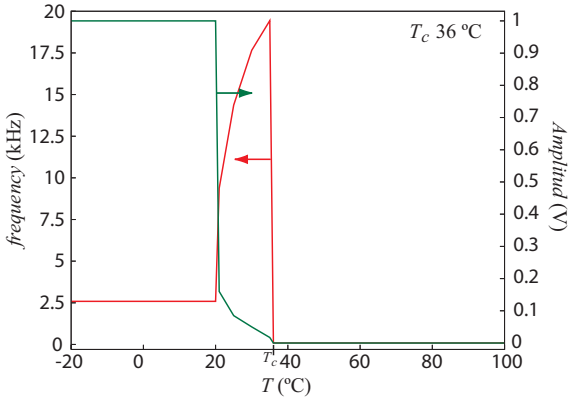


Figure 3.15: Oscillation frequencies of the circuit. ($T_c = 36°$C).

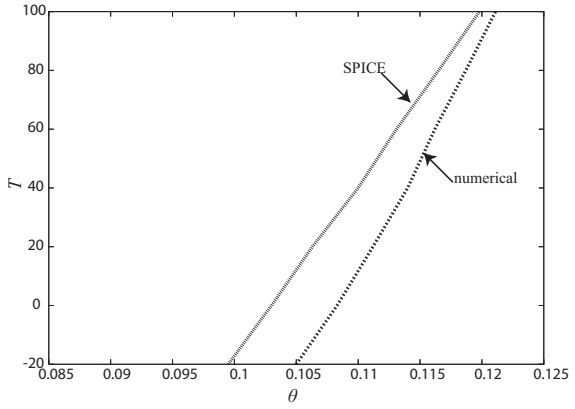


Figure 3.16: Relation between $\theta_{\pm}$ and T_c obtained through numerical and circuit simulations.

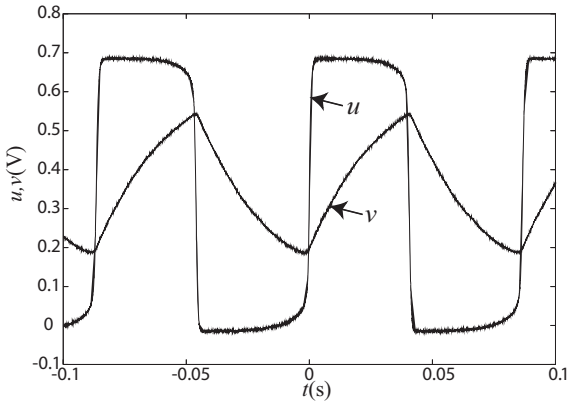


Figure 3.17: Experimental results: θ =170 mV at T= 23°C (oscillatory state).

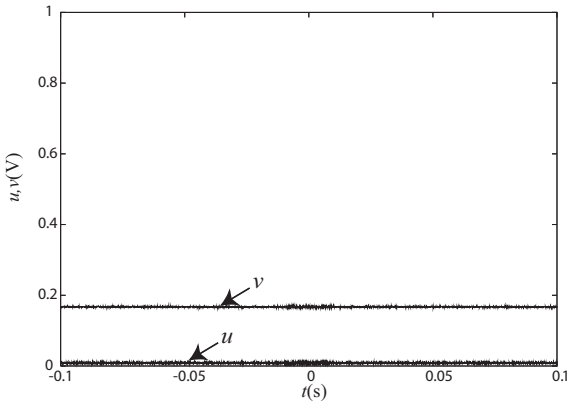


Figure 3.18: Experimental results: θ =150 mV at T= 23°C (stationary state).

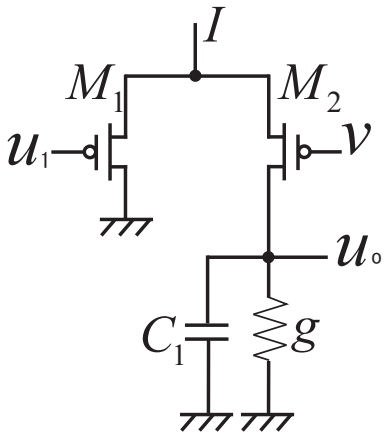


Figure 3.19: Circuit used for calculation of the u nullcline.

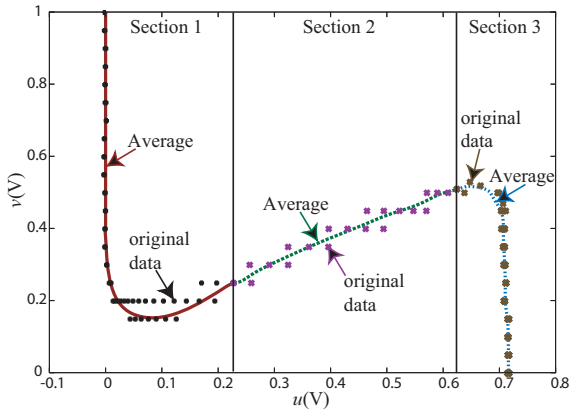


Figure 3.20: Sections used for the calculation of the u nullcline.

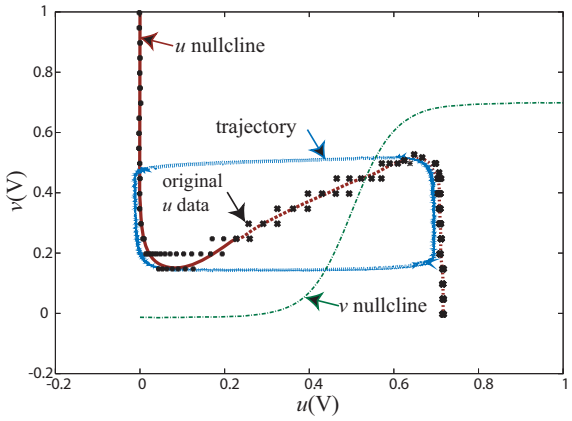


Figure 3.21: Experimental nullclines and trajectory.

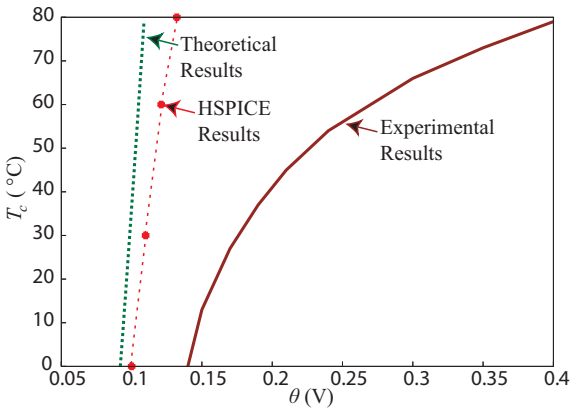


Figure 3.22: Bias voltage vs temperature, experimental results.

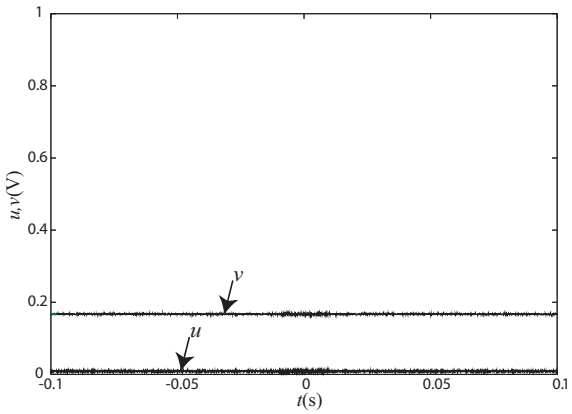


Figure 3.23: Stationary state with θ= 140 mV and T= 23°C.

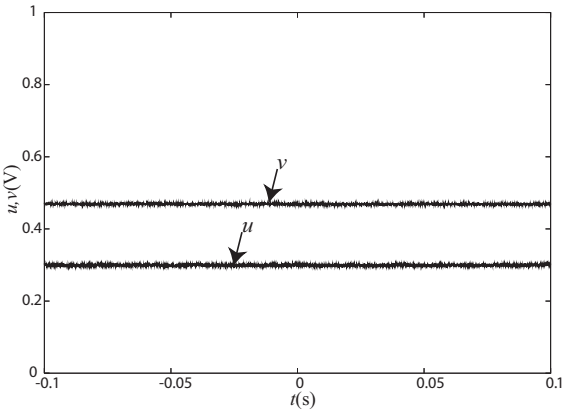


Figure 3.24: Stationary state with θ= 140 mV and T= 75°C.

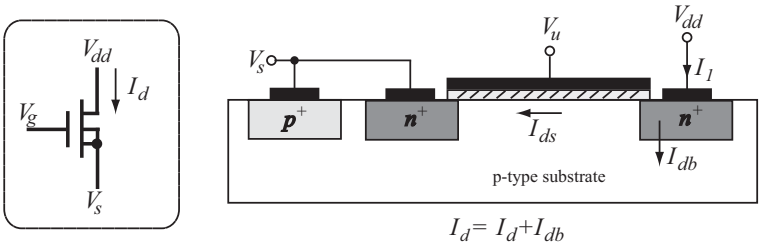


Figure 3.25: nMOS transistor structure showing leak current

3.4 nMOS transistor with temperature dependence

The structure of a nMOS transistor showing the temperature-sensitive drain to bulk leakage current (I_{db}) is shown in Fig. 3.25. The drain current of the transistor is thus given by the sum of the drain-bulk current (I_{db}) and the channel current (I_{ds}).

$$I_d = I_{ds} + I_{db} \tag{3.16}$$

and remembering that the saturated drain to source current when the transistor is operating in the sub-threshold region is given by

$$I_{ds} = I_0 e^{\kappa(V_g - V_s)/V_T} \tag{3.17}$$

the drain current becomes

$$I_d = I_0 e^{\kappa(V_g - V_s)/V_T} + I_{db} \tag{3.18}$$

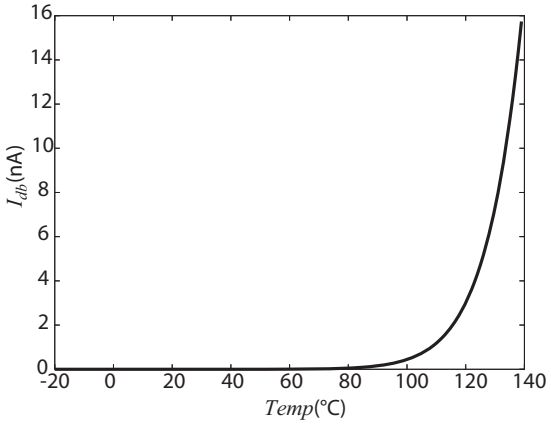


Figure 3.26: Drain-bulk current I_{db} vs $Temperature$.

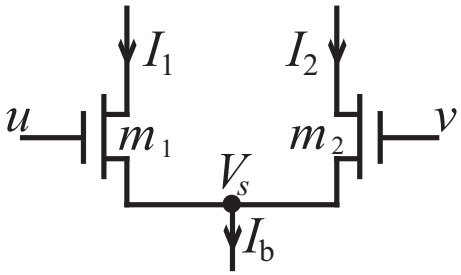


Figure 3.27: Differential pair.

where I_0 represents the zero bias current (fabrication parameter), and V_s the common source and bulk voltage.

The drain-bulk current (I_{db}) is given by:

$$I_{db} = G_{db}(V_{dd} - V_b) \tag{3.19}$$

where V_{dd} is the supply voltage, V_b the bulk potential, and G_{db} the temperature-dependent drain-bulk conductance expressed as:

$$G_{db} = G_S e^{\frac{E_g(T_{nom})}{V_{T_{nom}}} - \frac{E_g(T)}{V_T}} \tag{3.20}$$

where G_S represents the bulk junction saturation conductance (1×10^{-14}), $E_g(X)$ is the energy gap, and T_{nom} the nominal temperature (300.15 K). The temperature dependence of the energy gap is modeled by

$$E_g(T) = E_g(0) - \frac{\alpha T^2}{\beta + T} \tag{3.21}$$

Si experimental results give $E_g(0) = 1.16$ eV, $\alpha = 7.02 \times 10^{-4}$, and $\beta = 1108$.

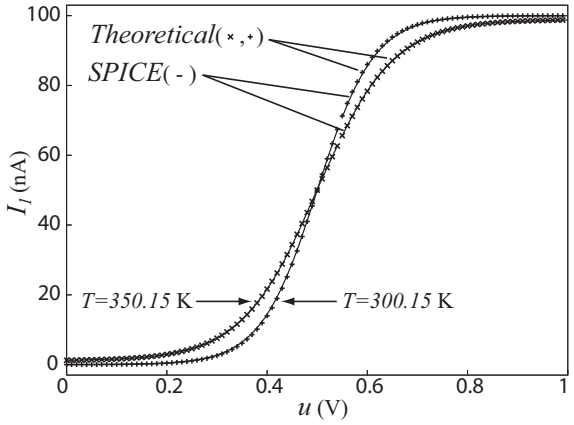


Figure 3.28: Theoretical and SPICE results of differential pair's current I_1 when temperature is 300.15 K and 400.15 K.

Numerical simulations where carried out. Figure 3.26 shows the drain-bulk current of a single transistor as the temperature changes. It can be observe that when the temperature is less than 80 $^\circ C$ the drain-bulk (I_{db}) current is in the order of pF ($\approx$ 30 pF), but as temperature increases, I_{db} also increases in an exponential manner reaching values in the order of nA ($\approx$ 16 nA for $T = 140$ $^\circ C$).

The same analysis can be applied to pMOS transistors, but in addition the leak current from the p-substrate to the n-Well is added to the drain current.

3.5 Differential pair with temperature dependence

Figure 3.27 shows a differential pair circuit consisting of two nMOS transistors (m_1 and m_2), and an ideal current source (I_b). According with the analysis done in the previous section, the drain currents (I_1 and I_2) are

$$I_1 = I_0 e^{\kappa(u-V_s)/V_T} + I_{db} \tag{3.22}$$

$$I_2 = I_0 e^{\kappa(v-V_s)/V_T} + I_{db} \tag{3.23}$$

Since $I_b = I_1 + I_2$, we obtain

$$e^{-\kappa V_s/V_T} = \frac{I_b - 2I_{db}}{I_0(e^{\kappa u/V_T} + e^{\kappa v/V_T})} \tag{3.24}$$

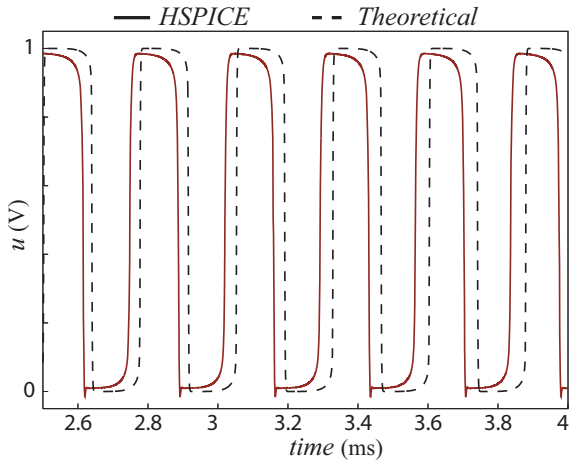


Figure 3.29: Comparison of temperature receptor oscillations, between HSPICE results and theoretical results without leak currents. $T = 127\ ^{\circ}C$

From Eqs. (3.22) and (3.23), the drain currents become

$$I_1 = \frac{(I_b - 2I_{db})e^{\kappa u/V_T}}{e^{\kappa u/V_T} + e^{\kappa v/V_T}} + I_{db} \tag{3.25}$$

$$I_2 = \frac{(I_b - 2I_{db})e^{\kappa v/V_T}}{e^{\kappa u/V_T} + e^{\kappa v/V_T}} + I_{db} \tag{3.26}$$

From Eq. (3.24) the common source voltage V_s is

$$V_s = \frac{V_T}{\kappa}\left\{\ln I_0 + \ln\left(e^{\kappa u/V_T} + e^{\kappa v/V_T}\right) - \ln\left(I_b - 2I_{db}\right)\right\} \tag{3.27}$$

Equations (3.25) and (3.26) were plotted and compared with the SPICE simulations results. The MOSIS AMIS 1.5-μm CMOS parameters (LEVEL 3) were used. Transistor sizes were set to $W/L = 4\ \mu$m/1.6 μm. I_b was set to 100 nA, and v was set to 0.5 V. From the SPICE simulations, the measured κ was found to be 0.47, I_0 was 18.8 pA when $T = 300.15\ ^{\circ}K$, and 62.6 pA when $T = 350.15\ ^{\circ}K$. We can observe that the theoretical results agreed with the SPICE results.

Then, the dynamics of the temperature receptor circuit (Eqs. (3.10) and (3.11)) with the temperature dependence analysis become

$$C_1\dot{u} = -gu + \frac{(I_a - 2I_{db} - 2I_{ws})\exp\left(\kappa u/v_T\right)}{\exp\left(\kappa u/v_T\right) + \exp\left(\kappa v/v_T\right)} + I_{db} + I_{ws}, \tag{3.28}$$

$$C_2\dot{v} = -gv + \frac{(I_a - 2I_{db} - 2I_{ws})\exp\left(\kappa u/v_T\right)}{\exp\left(\kappa u/v_T\right) + \exp\left(\kappa \theta/v_T\right)} + I_{db} + I_{ws}, \tag{3.29}$$

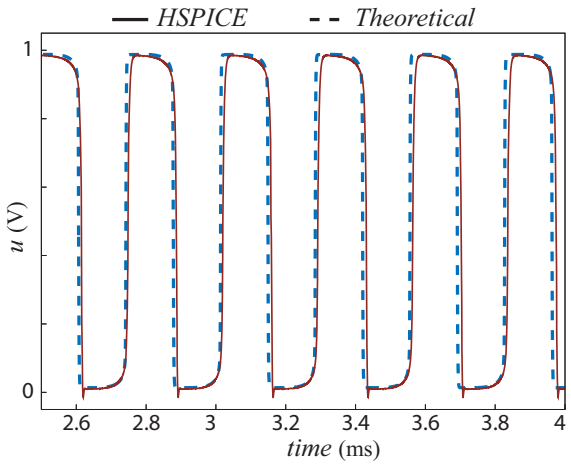


Figure 3.30: Comparison of temperature receptor oscillations, between HSPICE results and theoretical results including leak currents.

To confirm the effect of the leak currents in the temperature receptor system, a comparative analysis between HSPICE and the theoretical results was conducted without and with leak current. The comparison between HSPICE results and theoretical results without leak currents effect with the bias voltage θ set to 0.5 V and the external temperature set to $T = 127\ ^{\circ}C$, is shown in Fig. 3.29. It can be seen that in this case the results between the theory and the SPICE are very different, but in the same conditions when the effect of the leak current is include in the theory the results are very similar, Fig. 3.30.

3.6 Summary

A temperature receptor circuit was developed. The receptor consists of a subthreshold CMOS circuit that changes its dynamic behavior, i.e., oscillatory or stationary behavior, at a given threshold temperature. The circuit's operation was analyzed theoretically and through numerical and circuit simulations. Furthermore, the operation of the circuit was demonstrated using discrete MOS devices through experimental results. The threshold temperature (T_c) was set to a desired value by adjusting the external bias voltage (θ). The circuit changed its state between oscillatory and stationary when the external temperature was lower or higher than the threshold temperature (T_c). Moreover, the circuit nullclines were experimentally calculated, indicating the trajectory of the circuit when it is in oscillatory state.

Chapter 4

Noise in neural network

Noise permeates every level of the nervous system, from the perception of sensory signals to the generation of motor responses. In general, noise cannot be removed from a signal once it has been added. Therefore, it is though that neurons and neural networks may employ different strategies that can exploit the properties of noise to improve the efficiency of neural operations.

In recent years the extent to which noise is present and how noise shapes the structure and function of nervous systems have been studied. To this extend, it is well known that there are numerous noise (fluctuation) sources in nervous systems. External sensory stimuli are intrinsically noisy, at the first stage of perception energy in sensory stimulus is converted into a chemical signal (*e.g.* photon absorption arriving photoreceptors) or mechanical signal (*e.g.* movement of hair cells in hearing); the subsequent transduction process amplifies the sensory signal (with the noise) and converts it into an electrical one [24]-[26]. In each neuron, noise accumulates awing to randomness in the cellular machinery that processes information [27]. At the biochemical and biophysical level there are many stochastic processes at work in neurons. Electrical noise in neurons caused by the opening and closing of ion channels causes membrane potential fluctuations even in the absence of synaptic inputs [28] and also affects the propagation of action potential in axons [29]. Synaptic noise is caused by random events in the synaptic transmission machinery, such as protein production and degradation, fusing of synaptic vesicles, diffusion and binding of signaling molecules to receptors [30]-[33]. These, suggest that neural systems manage and may use noises to improve information processing [32].

Neural systems exploit noises in different ways, and one approach would be the stochastic resonance (SR) phenomenon. Stochastic resonance [26], [34] refers to a situation where the response of a system can be optimized by the addition of optimal noise levels. Since it was first discovered in cat visual neurons [35],

SR-like effects have been demonstrated in a range of sensory systems. These include crayfish mechanoreceptors [36], shark multimodal sensory cells [37], cricket cercal sensory neurons [38], and human muscle spindles [39].

Moreover, it has been observed that SR is further enhanced when a coupled array of similar nonlinear elements responds to the same signal. This phenomenon, known as array-enhanced stochastic resonance (AESR), was first observed in chains of nonlinear oscillators [40] and latter in ion channels [41], in arrays of FitzHugh-Nagumo neuron models [42], [43] and in a globally coupled network of Hodgkin-Huxley neuron models [44], [45].

Recently, Schweighofer *et.al.* [46] reported that the spiking behavior of a network of coupled inferior olive cells became chaotic for moderate electrical coupling, under these circumstances the input-output information transmission increased. In addition, in [47], Stacy *et.al.* demonstrated that an array of simulated hippocampal CA1 neurons exhibited SR-like behavior where an optimal correlation value between the sub-threshold input and output was obtained by tuning both the noise intensity and the coupling strength between the CA1 neurons; and, the correlation was further increased as the number of neurons increased.

Motivated by these findings; this chapter proposes a neural network model that exhibit AESR. The model is composed of Wilson-Cowan neural oscillators [21]. In the network, each neuron device is electrically coupled to its four neighbors to form a 2D grid network. All neurons accept a common sub-threshold input, and no external noise source is required as each neuron acts as a noise source to other neurons. The output of the network is defined as the sum of all the neurons. Numerical and circuit simulations were performed using standard (typical) device parameters. It was confirmed that without the electrical coupling, the circuit network exhibited standard SR behavior; and the network's behavior improved with the coupling strength.

4.1 Model and numerical simulations

The network model is illustrated in Fig. 4.1. The network has $N \times N$ neural oscillators consisting of the Wilson-Cowan neural oscillator model [21]. In the network, each neural oscillator is electrically coupled to its four neighbors to form a 2D grid network. The network dynamics are defined by

$$\tau \frac{du_{i,j}}{dt} = -u_{i,j} + f_{\beta_1}(u_{i,j} - v_{i,j}) + I_{in} + \sum_{j \neq i} g_{i,j} U_{j,j}, \tag{4.1}$$

$$\frac{dv_{i,j}}{dt} = -v_{i,j} + f_{\beta_2}(u_{i,j} - \theta), \tag{4.2}$$

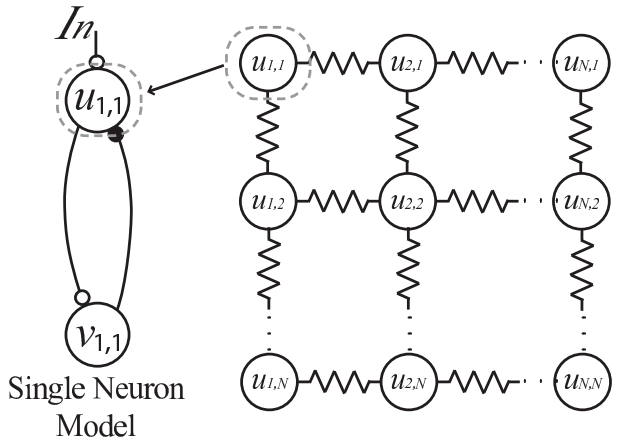


Figure 4.1: Network model.

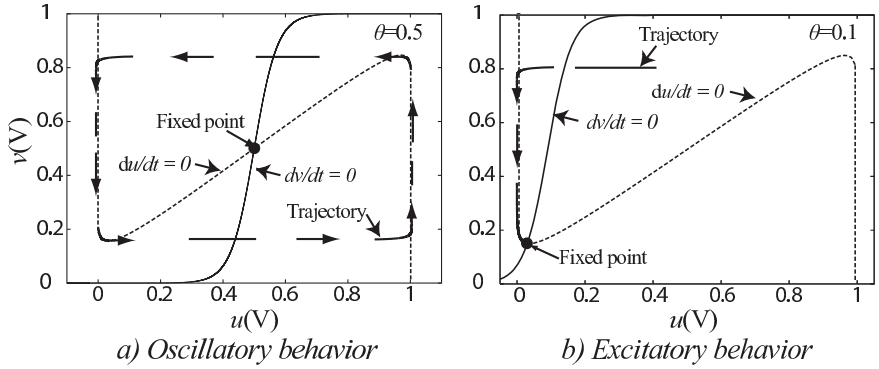


Figure 4.2: Nullclines of a single oscillator for different θs.

where $U_{i,j}$ is the potential observed at neuron $u_{i,j}$, given by

$$U_{i,j} = u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1} - 4u_{i,j}, \tag{4.3}$$

In Eqs. (4.1) and (4.2), τ represents the time constant, N represents the size of the matrix $(N \times N)$, $f_{\beta_i}(x-y)$ represents the sigmoid function defined by

$$f_{\beta_{i=1,2}}(x-y) = \frac{\exp(\beta_i x)}{\exp(\beta_i x) + \exp(\beta_i y)}, \tag{4.4}$$

From Eq. (4.1), I_{in} is the common input to the neurons, and $g_{i,j}$ is the coupling strength between oscillators i and j. The constant θ determines the state (behavior) of the neuron. Figure 4.2 shows the nullclines and trajectory of a single oscillator for different θs (0.1 and 0.5). The remaining parameters were set at $\tau = 0.1$, $\beta_1 = 5$ and $\beta_2 = 10$. As shown in the figure, depending on the position of the fixed point, the neuron exhibits oscillatory or excitatory behaviors. When

θ is 0.5, the fixed point is located on nullcline u at which $du/dv > 0$. In this case, the neuron exhibits limit-cycle oscillations (see Fig. 4.2(a)). On the other hand, when θ is 0.1, the fixed point is located on nullcline u at which $du/dv < 0$. In this case, the neuron exhibits excitatory behavior (see Fig. 4.2(b)) and it is stable at the fixed point as long as an external stimulus is not applied.

As explained in the previous chapter, models whose dynamics are described by Eqs. (4.1) and (4.2); are suitable for implementation in analog very large scale integrations (VLSIs) because the sigmoid function can be implemented using differential-pair circuits [23].

Excitability is observed in a wide range of natural systems. A list of examples includes lasers, chemical reactions, ion channels, neural systems, cardiovascular tissues and climate dynamics, to mention only the most important fields of research. Figure 4.3 show the nullclines, trajectory (dashed line) when the system is perturbed and activity (small square) of a typical excitable system. Common to all excitable systems is the existence of an "inactive" (or "rest") state (I), an "active" (or "firing") state (A), and a "refractory" (or "recovery") state (R). If unperturbed, the system resides in the rest state; small perturbations (sub-threshold input) result only in a small-amplitude linear response of the system (see fig. 4.3; small square). For a sufficiently strong perturbation (above-threshold input), however, the system can leave the rest state, going through the firing and refractory states before it comes back to rest again (see the nullclines in the figure). This response is strongly nonlinear and accompanied by a large excursion of the system's variables through phase space, which corresponds to a spike. The system is refractory after such a spike, which means that it takes a certain recovery time before another excitation can evoke a second spike.

Figure 4.4 shows the numerical solution of Eqs. (4.1) and (4.2) with 1000×1000 neurons, where the values of $u_{i,j}$ are represented in a black/white scale ($u_{i,j} < 0.5 \rightarrow$ black and $u_{i,j} \geq 0.5 \rightarrow$ white). The values of the remaining parameters were set at $\tau = 0.01$, $\theta = 0.1$ (excitatory behavior), $\beta_1 = 5$, $\beta_2 = 10$, $I_{in} = 0$, and the coupling strength $g_{i,j} = 0.035$. The solution was numerically obtained by solving the ordinary differential equations (ODE) with the fourth-order Runge-Kutta method. At each corner of the network, the values represented by $[i, j] \rightarrow [0, j]$ and $[i, j] \rightarrow [N + 1, j]$ were treated as $[i, j] \rightarrow [1, j]$ and $[i, j] \rightarrow [N, j]$, respectively. The initial conditions of the neurons were set as follows: the neurons represented by white lines in the figure ($t = 0$) were set to $u_{i,j} = 0.9$ and $v_{i,j} = 0.6$ (active mode), the neurons adjacent to each white line were set to $u_{i,j} = 0.0001$ and $v_{i,j} = 0.68$ (refractory mode), the remaining neurons were initially set to $u_{i,j} = 0.1$ and $v_{i,j} = 0.3$ (inactive or excitatory mode). The inactive neurons next to the active neurons (white line)

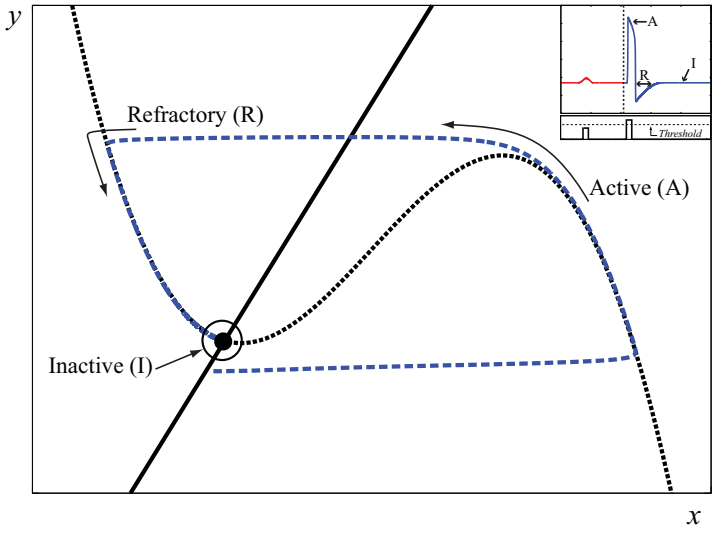


Figure 4.3: Nullclines and activity of a typical excitable system (Fitz-Hugh Nagumo) showing the different operation states.

were excited (activated) through their connection with the active neurons, they returned to the inactive mode following the pattern shown in the figure ($t = 1.5$ to $t = 7.5$) [48]. We used this continuous pattern as an internal noise source for the network.

Numerical simulations were conducted with I_{in} set as periodic sub-threshold pulses. The remaining parameters were set as previously described. The firing of each neuron was recorded and converted into a series of pulses of amplitudes 1 and 0 corresponding to the firing and non-firing states, respectively. The output (out) of the network was then defined by the sum of all the pulses divided by the number of neurons. To evaluate the performance of the network, correlation values C between converted sub-threshold input pulses (in) ($in = 0$ for $I_{in} = 0$, $in = 1$ for $I_{in} > 0$) and the output (out) were calculated, by:

$$C = \frac{\langle in \cdot out \rangle - \langle in \rangle \langle out \rangle}{\sqrt{\langle in^2 \rangle - \langle in \rangle^2} \sqrt{\langle out^2 \rangle - \langle out \rangle^2}}. \tag{4.5}$$

Figure 4.5 shows the simulation results. As shown, the correlation value between input and output increased with the coupling strength and reached a maximum peak when the coupling was around 0.12 and then decreased again. In addition, the noise levels was varied according to the number of spirals (number of white lines set in the initial conditions; see Fig. 4.4). When the number of spirals consisted of just a lines, the noise pattern became almost periodic; and it cannot be considered noise. When there are many spirals, the neurons activities

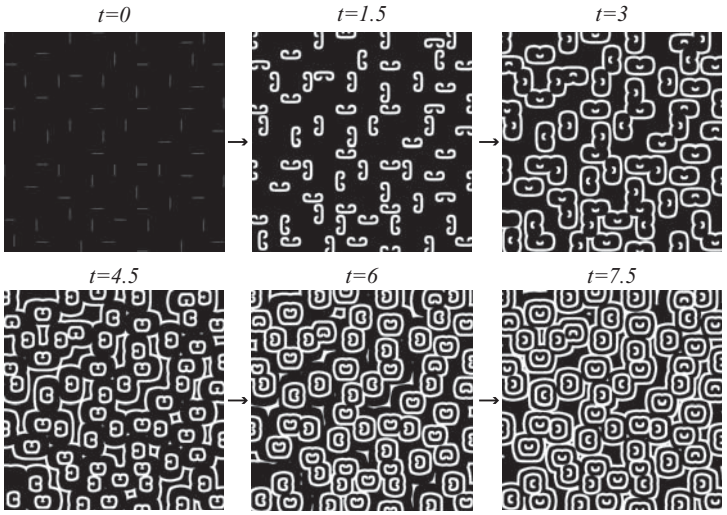


Figure 4.4: Numerical results of 1000×1000 network with $\theta = 0.1$ (excitatory behavior).

cancel each other and the pattern disappears. However, for a moderate number of spirals, (≈ 30) the noise pattern was similar to that shown in Fig. 4.4 at $t = 7.5$, which resulted in a random pattern as the time increases. As shown in Fig. 4.5, for moderate noise levels (about 30 spirals) the correlation values reached a maximum. These results suggest that, assuming signal transmission via an array of neuron devices under a noisy environment where the noise strength is fixed, the transmission error rate could be tuned by the coupling strength.

4.2 Circuit implementation

The Wilson-Cowan based neural oscillators have been implemented in the previous chapter. The oscillator consists of two pMOS differential pairs ($m_1 - m_2$ and $m_3 - m_4$), as shown in Fig. 4.6(a), two capacitors (C_1 and C_2) and two resistors (R). The differential-pairs sub-threshold currents, I_1 and I_2, are given by [23]:

$$I_1 = I_b \frac{\exp(\kappa u/V_T)}{\exp(\kappa u/V_T) + \exp(\kappa v/V_T)}, \tag{4.6}$$

$$I_2 = I_b \frac{\exp(\kappa u/V_T)}{\exp(\kappa u/V_T) + \exp(\kappa \theta/V_T)}, \tag{4.7}$$

where I_b represents the differential pairs bias current, V_T is the thermal voltage, and κ is the sub-threshold slope. The circuit dynamics can be determined by

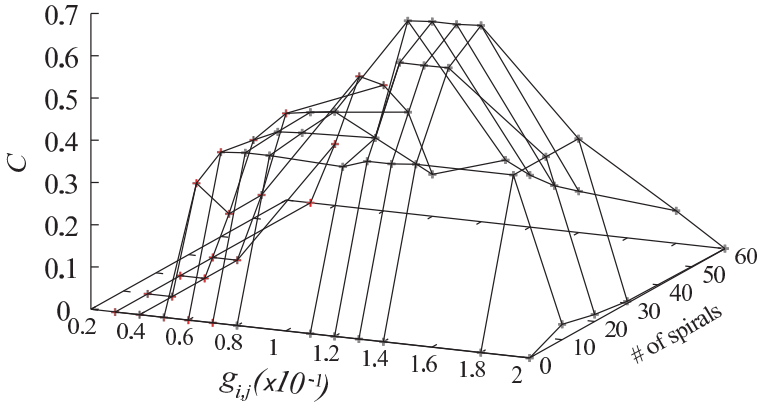


Figure 4.5: Numerical results showing the Correlation value vs coupling strength and noise levels.

applying Kirchhoff's current law to both differential pairs as follows:

$$C_1\dot{u} = -gu + \frac{I_b \exp(\kappa u/V_T)}{\exp(\kappa u/V_T) + \exp(\kappa v/V_T)}, \tag{4.8}$$

$$C_2\dot{v} = -gv + \frac{I_b \exp(\kappa u/V_T)}{\exp(\kappa u/V_T) + \exp(\kappa \theta/V_T)}, \tag{4.9}$$

C_1 and C_2 are capacitances representing the time constants, and θ is the bias voltage. Note that Eqs. (4.8) and (4.9) correspond to the dynamics of the network explained above (Eqs. (4.1) and (4.2), respectively) for $I_{in} = 0$ and $g_{i,j} = 0$. The simulated nullclines and the trajectory of a single neuron circuit for $\theta = 0.13$ are shown in Fig. 4.6(b).

Transient simulation results of the neuron circuit are shown in Fig. 4.7(a). In the figure θ was initially set at 0 V (in a relaxing state), and the neuron did not oscillate. Subsequently, θ was increased to 0.5 V at t = 2.5 ms, and the neuron (u) exhibited oscillations. Again, at t = 5 ms, θ was set to 0.09 V for the neuron to exhibit excitatory behavior. Since u had been excited before this time, the neuron emitted one spike and then relaxed, as expected. Then at t = 6 ms a sub-threshold input pulse (excitation) was applied, since the pulse was under the neuron's threshold it did not excite the neuron (no pulse was generated). Finally, at t = 8 ms, the amplitude of the input pulse was increased over the neuron's threshold, it can be observed that the neuron generated a pulse in response to the input. To further test the excitatory behavior of the neuron, circuit the nullclines and trajectory for θ= 0.09 V were plotted (from t=5 ms to 10 ms) in Fig. 4.7(b). When a sub-threshold pulse was applied (I_1 in Fig. 4.7(a)), the trajectory of the neuron could not overcome the attraction force of the fixed point and rapidly returned to its resting state. However, when

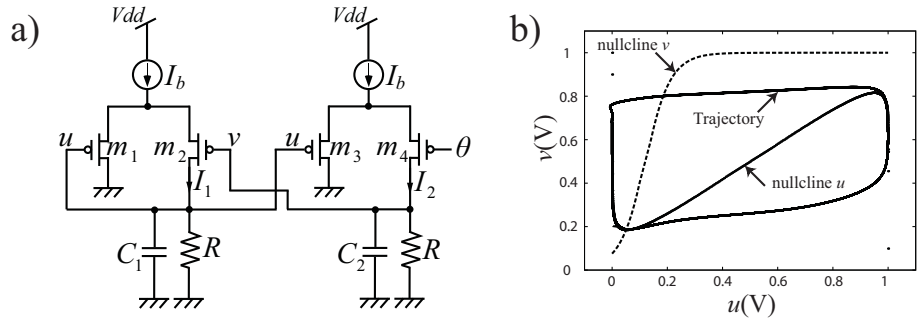


Figure 4.6: Single neural oscillator circuit and circuit's nullclines

the applied input pulse was over the neuron's threshold (I_2 in Fig. 4.7(b)), the neuron could trace a full trajectory and then return to its resting state.

4.3 Simulations results

Circuit simulations of a 10×10 circuit network were carried out . The parameter sets for the transistors were obtained from MOSIS 1.5-μm CMOS process. Transistor sizes of the differential pairs (Fig. 4.6(a)) were fixed at $L = 1.6$ μm and $W = 4$ μm. The supply voltage was set at 3 V. The neurons were locally connected by pass transistors instead of linear resistors. The connection strength was controlled by the common gate voltage (V_g) of these transistors. The values of the bias current I_b and the resistors R were set so that $R \times I_b = 1$. The capacitances C_1 and C_2 were set at 0.1 pF and 10 pF, respectively. Figure 4.8 shows the wave propagation of the circuit network with $V_g = 0.4$ V. For the simulations the initial conditions of the first neuron ($u_{i,j}$ and $v_{i,j}$ for $i, j = 1, 1$) were set to be in the active mode ($u_{1,1} = 0.99$ V and $v_{1,1} = 0.18$ V; white dot in the figure), the rest of the neurons were set to be in the inactive mode ($u_{i,j} = 0.01$ V and $v_{i,j} = 0.18$ V). In the numerical model, the inactive neurons located next to the active neuron (white dot) were activated through their connections with the active neuron, then returned to the inactive mode as the neuron's pulse completed its course following the pattern showed in the figure.

Circuit simulations of a network of 100×100 neurons were conducted. For the simulations, each neuron was excited with periodic sub-threshold current pulses at node u (Fig. 4.6(a)). The simulation results are shown in Fig. 4.9. The figure shows the correlation value as a function of the coupling strength, note that with a low coupling strength the correlation values were almost 0; however, as the coupling strength increased, the correlation value also increased. Since the network size was smaller compare to that of the numerical simulations,

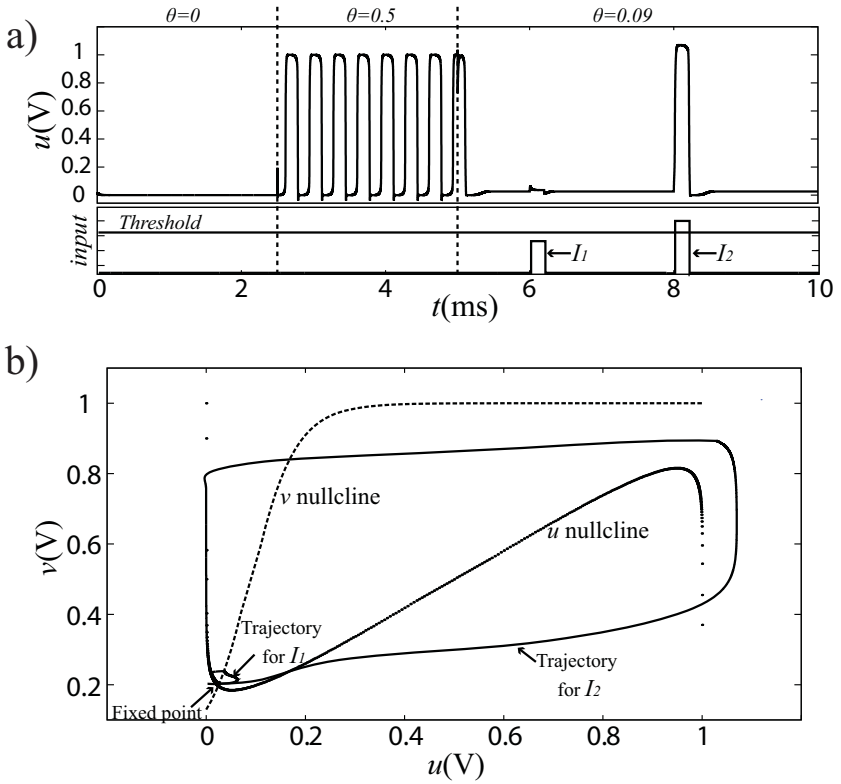


Figure 4.7: Simulation results of the neural oscillator circuit.

the number of spiral set in the initial conditions was also smaller (about 7 spirals). Hence, the maximum correlation value between the input and the output was around $C = 0.32$, which is less than half the value obtained by numerical simulations. However, these results showed that an increase in the correlation value could be realized by tuning the coupling strength, therefore it can be assumed that as the size of the network and number of spirals increase, the performance of the network can improve.

To further test the possibilities for implementing this kind of network, the effects that device mismatches might have in the circuit network was studied. For a single oscillator circuit, mismatches in the differential pairs ($m_1 - m_2$ and $m_3 - m_4$) can be negligible since they can be fabricated close to each other. However, if the bias currents I_bs (see Fig. 4.6(a)) are implemented with transistors (*i.e.* m_{b1} and m_{b2}); mismatches in these transistors might drastically change the behavior of the network. Therefore, numerical Monte-Carlo simulations by including threshold variations to the bias current I_b in the circuit dynamics were conducted. To do this, the bias currents I_bs (Eqs. (4.8) and (4.9)) was substituted by the sub-threshold current dynamics I_d given by:

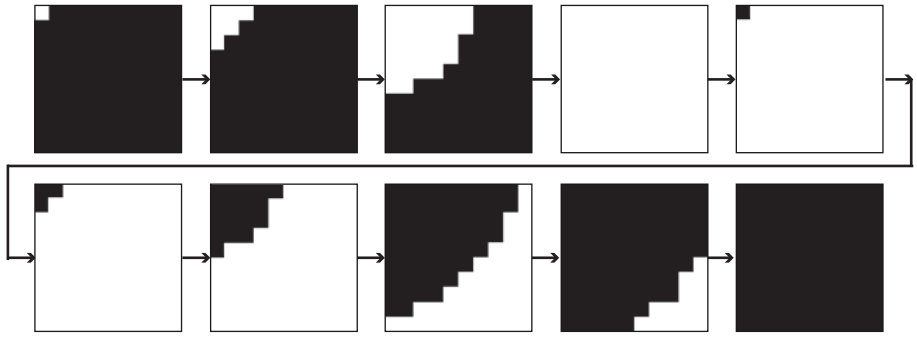


Figure 4.8: Circuit simulations results showing the wave propagation of the circuit with 10×10 neurons.

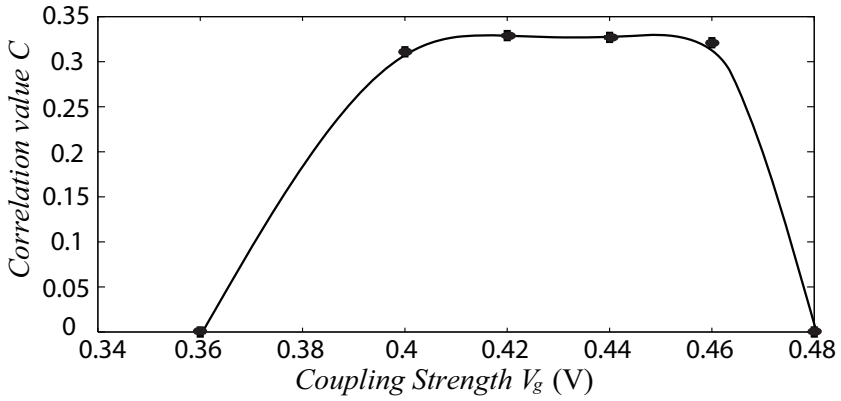


Figure 4.9: Circuit simulations results showing the correlation vs the coupling strength with 100×100 neurons.

$$I_d = I_0 K \exp \frac{V_{gs} - V_{th}}{\eta V_T}, \tag{4.10}$$

where I_0 is the zero bias current, $K = W/L$, and η is the sub-threshold slope factor. The threshold voltage V_{th} is proportional to the deviation $V_{th} \propto V_{tho} + \sigma_{Vtho}$. Numerical simulations of a large-scale network of 1000×1000 neurons were conducted. Parameter V_T was set at 26 mV, κ was set at 1.2, η and I_0 were calculated from SPICE simulations and found to be 1.5 and 4.79 pA, respectively. Initial conditions for the neurons in the inactive mode were randomly set between 1.12 and 1.2, while neurons in the active and refractory mode were set as previously described. For optimal operation of the network, moderate coupling strength was used, the noise was set with around 40 spirals. The mean value of the current I_d was set to be approximately 100 nA and R was set at $100 \times 10^9 \Omega$. Simulation results are shown in Fig. 4.10. As shown in the

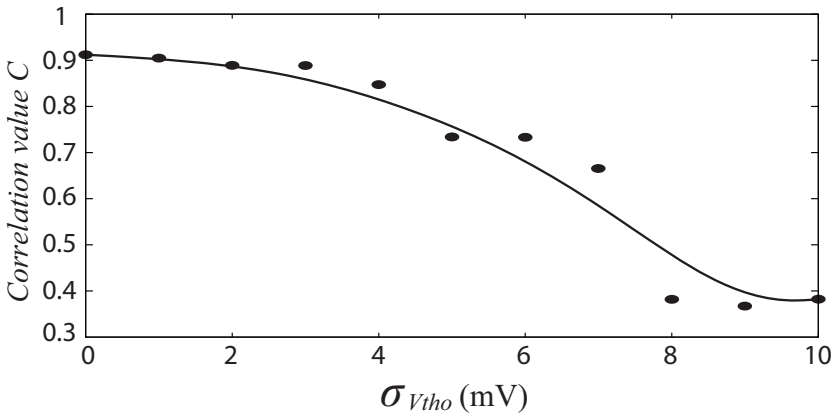


Figure 4.10: Numerical simulations of the circuit dynamic showing the correlation vs the threshold variation of bias current transistor in a network of 1000×1000 neurons.

figure, by introducing these new set of parameters the correlation value between input and output increased when no variation was applied (σ_{Vtho}=0) compared to previous numerical simulations. Moreover, the network showed tolerance to mismatches for variations up to σ_{Vtho}=6 mV with a minimum correlation value of C=0.7 for σ_{Vtho}=6 mV.

From the differences observed on numerical simulations (Figs. 4.5 and 4.10), it is understood that the performance of the network (Correlation value between input and output; C) depends on parameters set and initial conditions (**ic**). Therefore, numerical simulations for different sets of **ic** were conducted. Simulations results are shown in fig. 4.11). The error bars show represents the variation of C for different sets of **ic**, squares are the mean value of Cs obtained for every simulation set. It can be observed that with no threshold variation ($\sigma_{V_{tho}} = 0$), C greatly depends on **ic** (C varied between 0.6 to 0.9). However, when threshold variation are applied, C dependency on **ic** decreased (for $\sigma_{V_{tho}}$=4 mV C increased to 0.9). When threshold variations are applied, the bias current (I_b) of each neuron varied causing a shift on the behavior of some neurons (*i.e.* neurons changed from excitatory to oscillatory behavior; see fig. 4.12). Hence, oscillatory neurons became a constant noise source to the network. A further increase of $\sigma_{V_{tho}}$, resulted on more neurons becoming oscillatory, in that case the output of the network was governed by the oscillatory behavior of neurons and not by the input (I_{in}); therefore C decreased.

These results may be an important step toward the construction of robust brain-inspired computer systems.

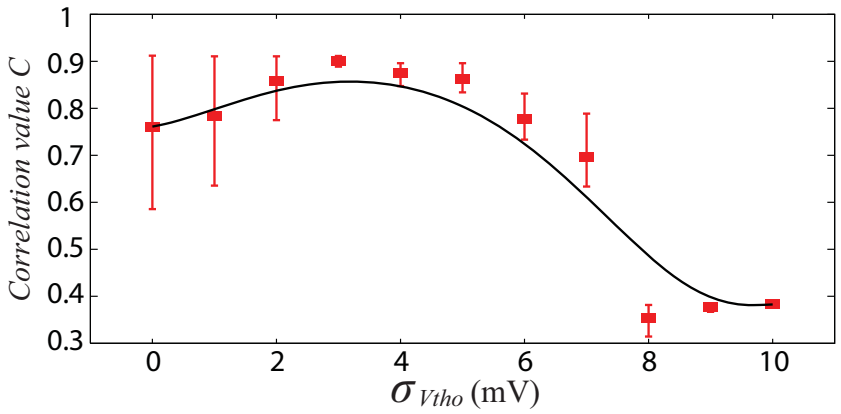


Figure 4.11: Numerical simulations of the circuit dynamic showing the correlation C vs the threshold variation σ_{Vtho} for different initial conditions in a network of 1000×1000 neurons.

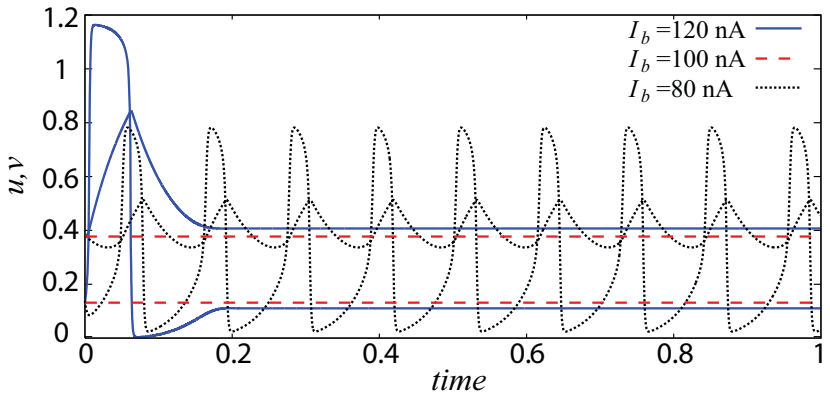


Figure 4.12: Neuron behavior for different values of bias current I_b

4.4 Summary

A neuromorphic network exhibiting array-enhanced stochastic resonance was proposed. The model consisted of $N \times N$ Wilson-Cowan neural oscillators. Each oscillator was connected to its four neighbors to form a 2D grid. Wave propagation characteristic of the network were used as internal noise sources. Numerical simulations of a 1000×1000 network were performed and it was shown that the correlation value between input and output can be increased by tuning the coupling strength. A circuit network of 10×10 neurons was simulated to demonstrate the wave propagation of the circuit network. To further test the circuit network, numerical Monte-Carlo simulations were conducted; the results showed that the network was tolerant to mismatches.

Chapter 5

Depressing synapses and synchronization

Changes in our external environment are detected by sensory receptors cells, which transduce the sensory stimulus into an electrical signal. This electrical signal is then graded depending on the stimuli intensity. Synapses in vision, balance and hearing transmit this graded information with high fidelity [49]. The computational potential of synapses is large because their basic signal transmission properties can be affected by the history of pre-synaptic and post-synaptic firing in many different ways. This potential has important implications for the diversity of signaling within neural circuits, suggesting that synapses have a more active role in information processing [50].

Depressing synapses are a type synapses with the characteristic of reducing synaptic strength. They have been shown to contribute in a wide range of sensory tasks such as; contrast adaptation in vision [51], adaptation have been observed also in the somatosensory cortex [52], suppression by masking stimuli in primary visual cortex [53], habituation [54], sound localization [55] [56] and sensory input selection [51], [57].

In addition, because depressing synapses produce transmission sequences that are more regular as compare to excitatory synapses, they have been proposed as a mechanism that removes redundant correlation so that transmission sequences conveys information in a more efficient manner [58]. To encode information brought by sensory stimuli, dynamic response of a single neuron and **synchronization** within an ensemble of active neurons play an essential role. Some studies suggest that depressing synapses may have an effect on neurons synchronization; in the auditory pathway, depressing synapses can provide an effective way of detecting emergent synchrony activities [59], they can generate

sustained oscillations of neural activity [60], and promote stability of cortical activty [61].

To this end, a neural network model with depressing synapses that exhibits synchronization even in a noisy environment was proposed by Fukai and Kanemura [62]. In this chapter a MOS circuits that 'qualitatively' imitate this network model is designed. The circuit is constructed with silicon neurons and depressing synapse circuits. Using a simulation program with integrated circuit emphasis (SPICE), it was demonstrates that depressing synapses facilitate synchronization among neuron circuits. Since a higher tolerance to external noises could be achieved by introducing spike-timing dependent plasticity (STDP) learning in the network model [62], an analog circuit for the STDP learning is also proposed.

5.1 Network model

Figure 5.1 illustrates the network model. Four pyramidal neurons (triangles) are shown. All of the outputs of the pyramidal neurons are sent to an interneuron (circle in the figure) through excitatory synapses, whereas the interneuron inhibits all of the pyramidal neurons through inhibitory synapses. Outputs of the pyramidal neurons are randomly connected to pyramidal neurons through depressing synapses (connection ratio). Since these synapses provide positive feedback connections to pyramidal neurons [62], firing one pyramidal neuron induces firing of other pyramidal neurons, which results in synchronous firing of pyramidal neurons. The divergence due to the positive feedback is attenuated by the interneuron that inhibits all of the pyramidal neurons.

The dynamics of a neural network model for precisely-timed synchronization [62] are given by

$$\begin{aligned}
\tau_m \frac{dV_i}{dt} &= -(V_i - V_{\rm rest}) - \frac{1}{NR}\sum_{j\neq i} c_{ij} g_{ij}^{\rm ee}(V_i - V_{\rm syn}) \\
&\quad -g^{\rm ei}(V_i - V_{\rm cl}) + E_i \\
\tau_e \frac{dE_i}{dt} &= -E_i + E_0\ \delta(t - t_i^{\rm inp}),\ \ (i = 1, \cdots, N) \\
\tau_i \frac{dV}{dt} &= -(V_i - V_{\rm rest}) - g^{\rm ie}(V - V_{\rm syn})
\end{aligned}$$

where V_i and V represent the membrane potentials of the i-th pyramidal (integrate-and-fire) neuron and an interneuron; E_i the postsynaptic potential of the i-th pyramidal neuron; $\tau_{m,e,i}$ the time constants of pyramidal neurons, excitatory synapses, and interneurons; N the number of pyramidal neurons; R the positive-feedback connectivity between the pyramidal neurons as described above (the

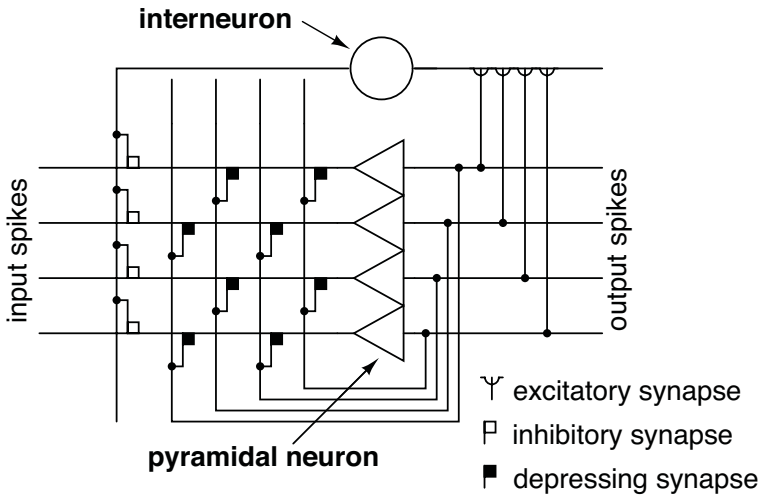


Figure 5.1: Neural network model for precisely-timed pulse synchronization

connection ratio); $V_{\mathrm{rest,syn,cl}}$ the resting potential of pyramidal neurons, depressing synapse, and interneurons; t_i the time at which the i-th input spike is given; c_{ij} the binary representing the existence of feedback connections between the i-th and j-th pyramidal neurons; and $g^{\mathrm{ee,ei,ie}}$ the synaptic conductance between excitatory-to-excitatory, excitatory-to-inhibitory, and inhibitory-to-excitatory neurons.

5.2 Circuit implementation

Using silicon neurons and depressing synapse circuits [63] [64] the network model described in the preceding section was constructed and the circuit's operational principles is explained.

Figure 5.2 shows a diagram of a neuron circuit that imitates the basic operations of an integrate-and-fire neuron (An integrate-and-fire neuron is one of the earliest models of a neuron and is represented by the time derivative of the law of capacitance). In addition, excitatory and inhibitory synapses are constructed by pMOS and nMOS current mirrors that receive input pulses as current. Delayed synaptic potentials (V_{inh} and V_{exc}) are generated by capacitors C_1 and C_2. The excitatory postsynaptic current generated by V_{exc} charges C_3 and consequently increases the membrane potential U_i, whereas the inhibitory postsynaptic current generated by V_{inh} decreases it. An increase in the membrane potential (U_i) in the soma circuit induces an increase in potential V_i by charging C_4. Thus, when the membrane potential exceeds a certain threshold, the membrane node (U_i) is suddenly shunted by transistor M1. Although the

shunted current increases exponentially with increasing membrane potential, the current is then decreased when C_4 is discharged by M4 with control voltage V_{B}. This sudden increase and decrease of shunting currents generate a pulse. The output pulse is obtained by the current of transistor M2 and converted to voltage by the diode-connected transistor M3. For the detailed dynamics and mathematical explanations, see ref. [63].

Figure 5.3 shows a MOS circuit for a depressing synapse constructed with a pMOS current mirror (M3, M4 and M5) and pMOS common-source amplifier (M2 and M4). It should be noticed that M4 of the common-source amplifier is shared by the current mirror with M5. When there is no input (current), voltage V_{e} at node A is zero because of a leak current from transistor M2. Therefore, transistor M1 is on. When there is an input current that increases V_{e}, M1 is turned off. The current is therefore mirrored to output I_{out} through transistor M_1. Because there is a parasitic capacitance (C_{dep}) at node A, the increase in V_{e} has a short time delay. Therefore, M1 is turned on for a short time, and the output current is generated. When the input current becomes zero again, M2 discharges the capacitance C_{dep}, and V_{e} returns to zero. Remarkably, the mirror effect of the pMOS common-source amplifier, which amplifies the value of additional parasitic capacitance between the drain and gate terminal of M4, increases this discharging time. When the current pulse is given at a short interval and subsequent pulses enter before V_{e} returns to zero, the amplitude of the output pulses decreases when V_{e} increases. Because the current of transistor M2 increases monotonically when V_{B} increases, the time until V_{e} returns to zero decreases. Thus by adjusting voltage V_{B}, the duration of the depression can be changed. Notice that, when V_{B} is set t0 V_{dd}, the circuit behaves as a nondepressing synapse because V_{e} is zero and M1 is always on.

Neural network hardware that is qualitatively equivalent to the network model shown in Fig. 5.1 is illustrated in Fig. 5.4. To evaluate basic operations of the network hardware, two neuron circuits for pyramidal neurons and one neuron circuit for an interneuron were used. Outputs of the pyramidal neuron circuits are sent to the interneuron circuit through nondepressing excitatory synapses constructed with pMOS current mirrors, whereas the output of the interneuron circuit is connected to nondepressing inhibitory synapses (nMOS current mirrors) of the pyramidal neuron circuits. Outputs of the pyramidal neuron circuits are also fed back to themselves through nondepressing or depressing synapses, each of which is an excitatory connection. The network accepts external input pulses at terminals IN1 and IN2 and produces the output pulses at terminals OUT1 and OUT2.

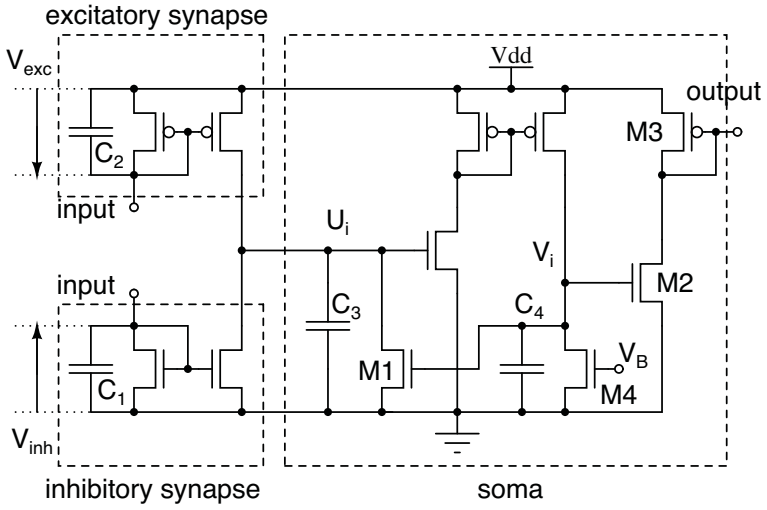


Figure 5.2: Neuron circuit with conventional excitatory and inhibitory synapses

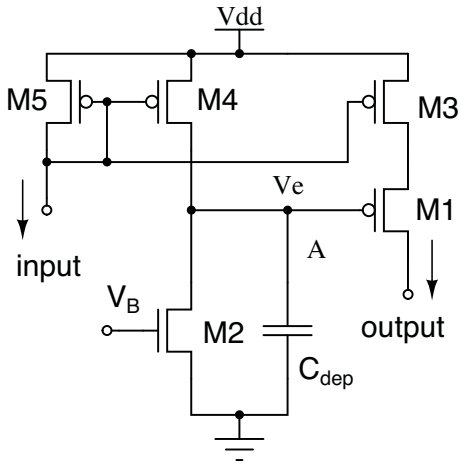


Figure 5.3: Depressing synapse circuit

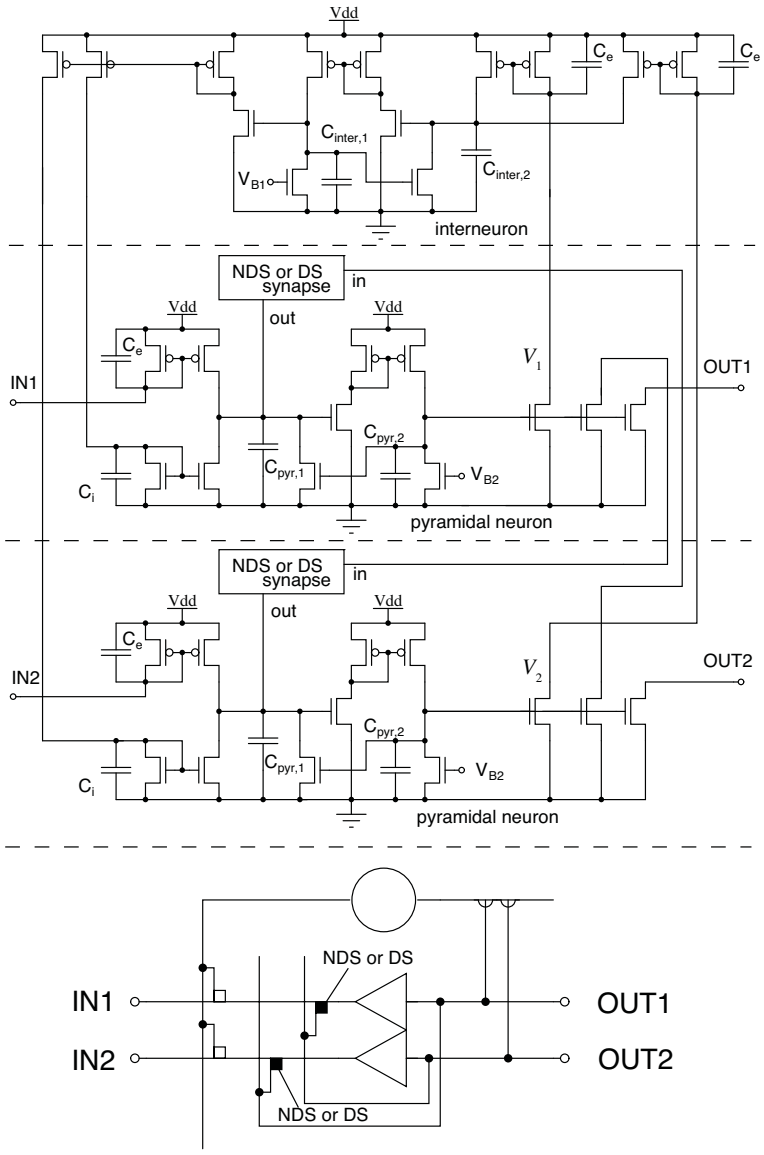


Figure 5.4: Circuit diagram of network model with two pyramidal neurons and one interneuron: Each pyramidal neuron circuit has positive feedback connection through nondepressing (NDS) or depressing synapses (DS).

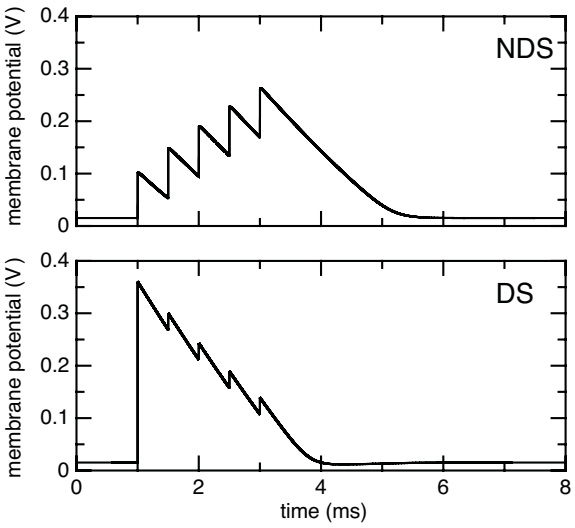


Figure 5.5: Membrane potentials of pyramidal neuron circuits for short time input spike trains through nondepressing (NDS) or depressing synapses (DS)

5.3 Simulation results

A simulation program with integrated circuit emphasis (SPICE) was used to evaluate the proposed circuit with MOSIS parameters (Vendor AMIS, feature size: 1.5 μm). All the transistors dimensions (channel width and length) were fixed at 2.3 and 1.5 μm, except for the channel length of M5 in depressing synapse circuits. To compare the effects of depressing synapses on timing precision of synchronization among pyramidal neurons, the network with nondepressing and depressing synapses for feedback connections between pyramidal neurons was evaluated.

Figure 5.5 shows membrane potentials of a pyramidal neuron circuit in response to short time burst input pulses (five pulses with intervals of 500 μs) through nondepressing and depressing synapse circuits. Amplitudes of input pulses, $V_{\rm dd}$, $C_{\rm dep}$, and $V_{\rm B}$ were set at 10 nA, 5 V, 100 fF, and 350 mV. The channel length of M5 was set at 3 μm for depressing synapses and 6.5 μm for nondepressing synapses, which evoked on average the same excitatory postsynaptic potential (EPSP), i.e., charges in membrane capacitances during the burst input pulses were fixed to constant values regardless of the type of synapse (nondepressing or depressing). This result ensures that the EPSP generated by the depressing synapse circuit has a larger response at the burst onset than that of the nondepressing synapse circuit. Figure 5.6 shows the change in amplitude of the output pulse against the input firing rate where $V_{\rm B}$ was set at 0.1, 0.2, and 0.3 V. As the pulses frequency increases, the amplitude of the output

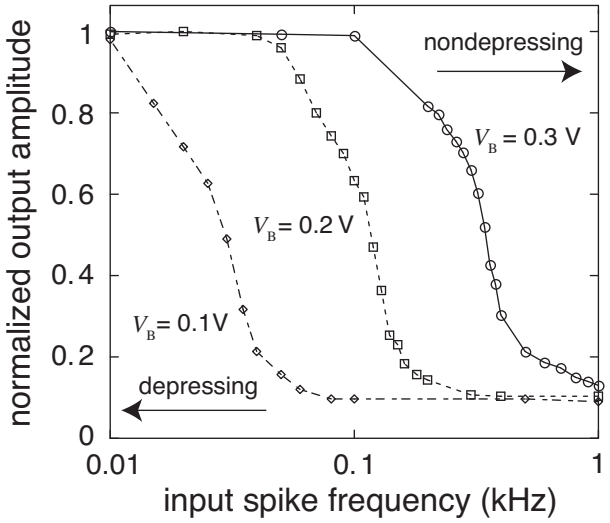


Figure 5.6: Changes in amplitude of output of depressing synapse circuit against firing rate of presynaptic neuron

	NDS	DS
average jitter (μs)	0.92	0.82
σ^2 (μs)	0.36	0.21

Table 5.1: Comparison of results averaged timing jitters and their standard deviations (σ^2) between nondepressing (NDS) and depressing synapses (DS)

pulse decreased. By increasing V_{B}, the cutoff frequency was successfully shifted toward the higher frequency (toward the nondepressing operation).

Based on the extracted parameter results of nondepressing and depressing synapse circuits, we evaluated the timing precision of synchronization in the network. In the following simulations, the input pulses frequency was fixed at 2 kHz. Capacitance values of $C_{\mathrm{inter},1}$, $C_{\mathrm{inter},2}$, $C_{\mathrm{pyr},1}$, and $C_{\mathrm{pyr},2}$ were set at 1 pF, 100 fF, 300 fF, and 1 pF, whereas capacitors of nondepressing inhibitory and excitatory synapses were removed in this simulation ($C_{\mathrm{i}} = C_{\mathrm{e}} = 0$). The neuron's bias voltages V_{B1} and V_{B2} were set at 650 and 560 mV. Figures 5.7 and 5.8 show output pulses train of pyramidal neuron circuits (V_1 and V_2 in Fig. 5.4) when nondepressing and depressing synapses were used to connect pyramidal neurons to each other. In both figures, each pyramidal neuron circuit tends to be synchronized in the phase space. For a simple evaluation of the synchronization, the following was calculated

$$S(t) = H(V_1(t) - \theta) \times H(V_2(t) - \theta) \tag{5.1}$$

where $H(\cdot)$ represents the step function and $\theta = 4.2$ V. When V_1 and V_2 are fired

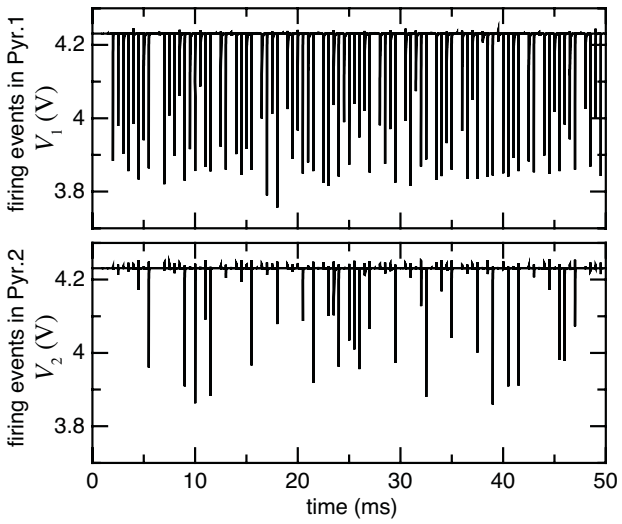


Figure 5.7: Output pulses train of pyramidal neuron circuits with nondepressing synapses

simultaneously at time t, $S(t)$ becomes 1. Normalizing neuron circuit's intrinsic firing frequency at the ratio of 3 μm (depressing) to 6.5 μm (nondepressing), $\sum_{t=0}^{40\ \mathrm{ms}} S(t)$ was 6 for nondepressing whereas it was 17 for depressing synapses, which quantitatively showed an improved synchronization between neuron circuits when depressing synapses were used. We also calculated the timing jitters of output pulses of pyramidal neuron circuits. Table 5.1 shows a comparison of the results of averaged timing jitters and their standard deviations (σ^2) between nondepressing and depressing synapses. We found that when depressing synapse circuits were used, the average jitter was 0.1 μs better than that of nondepressing synapse circuits. In addition, values of the standard deviation were 60 % better than those of nondepressing synapse circuits. Therefore, we concluded that depressing synapse circuits improve the timing precision of synchronization. Remember that an EPSP generated by a depressing synapse circuit has a larger response at a pulse onset than that of a nondepressing synapse circuit (Fig. 5.5). When nondepressing synapses are used, several pulses are required to evoke enough EPSPs to fire, whereas EPSPs evoked by depressing synapses easily make a pyramidal neuron fire with a few pulses, e.g., even a single pulse is sufficient if the threshold potential is set at a very low value. The resultant firing gives rise to the subsequent firing of other pyramidal neurons, which results in fast synchronization among all of the pyramidal neurons.

Synaptic depression is indeed able to detect partial synchrony in the burst times [59]. With nondepressing synapses, the postsynaptic membrane potential follows the presynaptic mean firing rate and is able to be set continuously

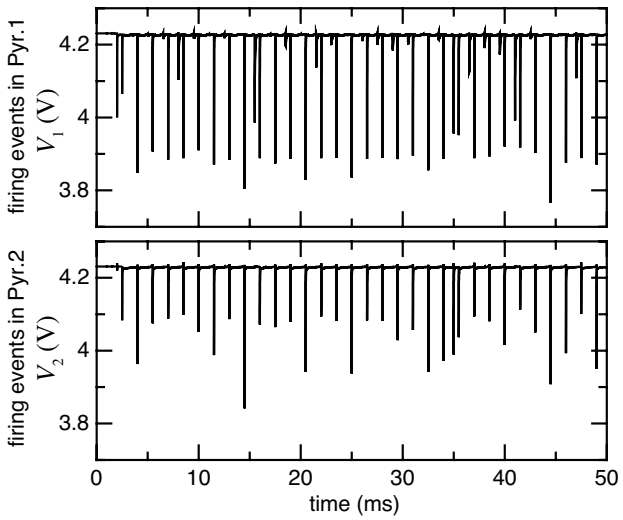


Figure 5.8: Output pulses train of pyramidal neuron circuits with depressing synapses

below the threshold of a neuron. With depressing synapses, however, the partially synchronized bursts push the postsynaptic membrane potential across the threshold repeatedly during stimulus.

5.4 STDP learning circuit

According to the Hebb principle, synapses increase their efficacy if two connected neurons are simultaneously fired. *Simultaneous* is to be defined by some time window of coincidence. This window of coincidence has being a function of the exact timing of the activity of the presynaptic and postsynaptic neuron, and this phenomenon is called spike-timing-dependent plasticity (STDP). By introducing STDP learning in the original network, Fukai and Kanemura demonstrated that the network exhibited robust synchronization in a noisy environment [62]. In this section, a novel analog circuit emulating the STDP learning is proposed. The circuit consists of two basic circuits: a spike-timing detector and an analog memory circuit.

To construct a spike-timing detector, a simple correlation neural network was used [65]-[67]. Figure 5.9 shows a local correlation scheme used to account for timing-sensitive responses of output neurons to input pulses (spike) train. A primitive correlation neural network consists of two input neurons (P_1 and P_2), a delay neuron (D), and a correlator (C), as shown in Fig. 5.9(a). The arrival of pulses from P_1 at the correlator is delayed by the delay neuron. The output is a correlation value representing the product of delayed and undelayed signals

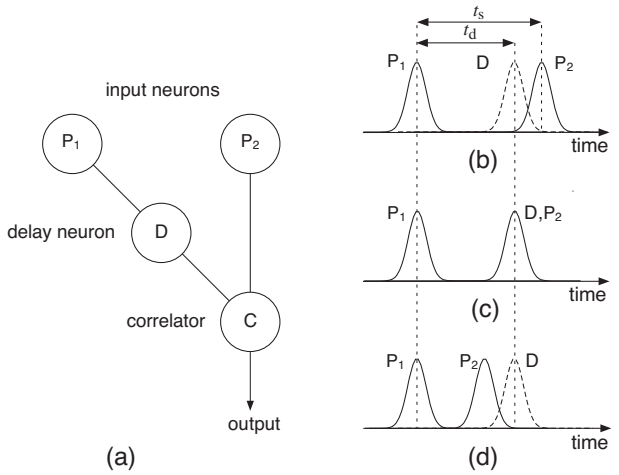


Figure 5.9: (Primitive correlation neural network consisting of two input neurons (P_1 and P_2), delay neuron (D), and correlator (C)

from D and P_2.

When an input pulse is given to P_1 and then to P_2 within the time t_s, which is longer than the delay time t_d, the delayed and undelayed signals from D and P_2 do not coincide at the correlator, as shown in Fig. 5.9(b). If an input pulse is given to P_1 and then to P_2 in a time equal to the delay time, the delayed and undelayed signals coincide at the correlator [Fig. 5.9(c)]. Namely, the output signal of the correlator reaches its maximum at the point of coincidence. On the other hand, if an input is given to P_1 and then to P_2 in a time shorter than the delay time, the output signal monotonically decreases as the time decreases [Fig. 5.9(d)]. Thus, the network can measure the degree of temporal difference by monotonically increasing output signals as the pulses (spike) intervals between P_1 and P_2 decrease.

Figure 5.10 show a circuit diagram of spike-timing detectors implementing the correlation neural networks. The circuit consists of a delay circuit (a source-common amplifier and a capacitor), which we denote CMA in Fig. 5.10(b); a *p*MOS unity-gain amplifier (UGA); and a current converter (diode-connected MOS transistor DCM). A circuit shown in Figs. 5.10(a) and (b) detects sequential inputs of pre-to-post spikes. If one input pulse is given to terminal **pre** and then the subsequent pulse input is given to terminal **post** ($t_{\rm post} - t_{\rm pre} \equiv \Delta t > 0$), $V_{\rm pot}$ increases because input of terminal **pre** is delayed by the source-common amplifier, while the unity-gain amplifier that accepts the delayed voltage is driven by the **post** input. Note that the source-common circuit amplifies not only the pre voltage input but also the decay time due to the Miller effect. Since the output of the unity-gain amplifier ($V_{\rm pot}$) is sent to a diode-connected *n*MOS

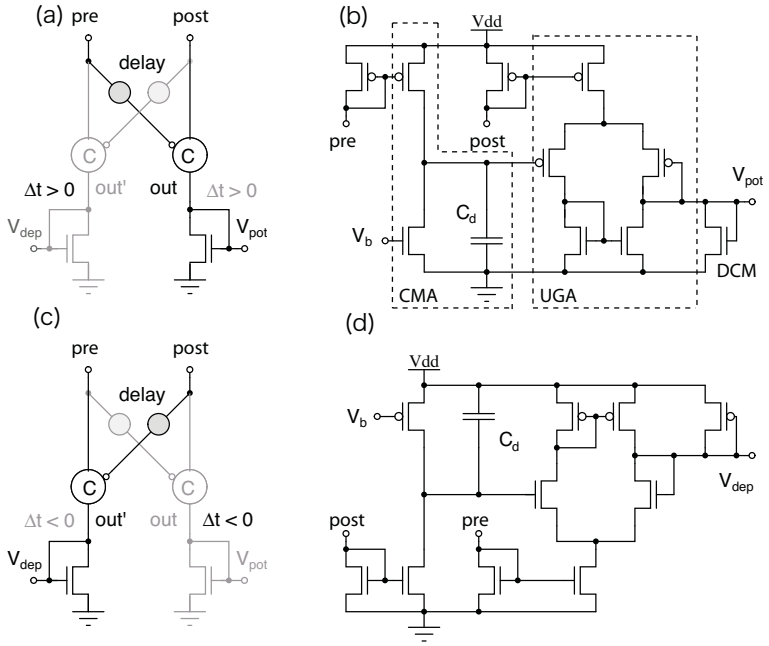


Figure 5.10: Spike-timing detectors

transistor, we can obtain a current output as a result of the current inputs (terminals **pre** and **post**). Similarly, Figs. 5.10(c) and (d) show the inverted circuit of Figs. 5.10(a) and (b) that detects sequential inputs of post-to-pre spikes ($\Delta t < 0$).

An analog memory circuit for STDP learning is illustrated in Fig. 5.11. The circuit consists of a pMOS differential pair, a storage capacitor ($C_{\rm memory}$), and pMOS and nMOS current sources that receive the output of spike-timing detectors ($V_{\rm pot}$ and $V_{\rm dep}$) constructing pMOS and nMOS current mirrors. The storage capacitor is discharged (or charged) by pre-to-post (or post-to-pre) input pulses through $V_{\rm pot}$ (or $V_{\rm dep}$), e.g., when $\Delta t > 0$, $V_{\rm pot}$ is increased and thus the storage capacitor is discharged. Synaptic weight strength w between pre and post neurons is defined by the ratio of input current $I_{\rm in}$ to output current $I_{\rm out}$ and controlled by the difference between capacitor voltage $V_{\rm mem}$ and reference voltage $V_{\rm ref}$. Initially $V_{\rm mem}$ is set to $V_{\rm ref}$ by manual reset switching, i.e., $I_{\rm out} = I_{\rm in}/2$ and thus $w = 2$. When all the transistors operate in their subthreshold regions, weight strength w is given by

$$\begin{aligned} w &= \frac{I_{\rm in}}{I_{\rm out}} = \frac{1}{f(V_{\rm mem} - V_{\rm ref})} \\ \left[f(x) \right. &= \left. \frac{1}{1 + \exp(-\kappa x / V_T)} \right] \end{aligned}$$

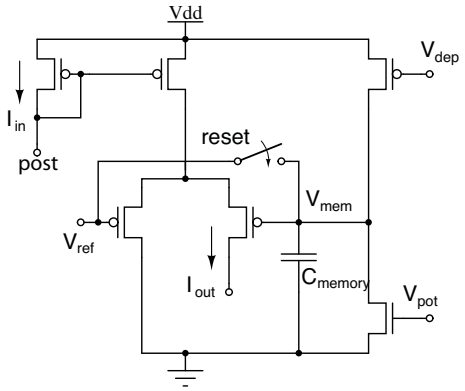


Figure 5.11: Analog memory circuit for weight storage

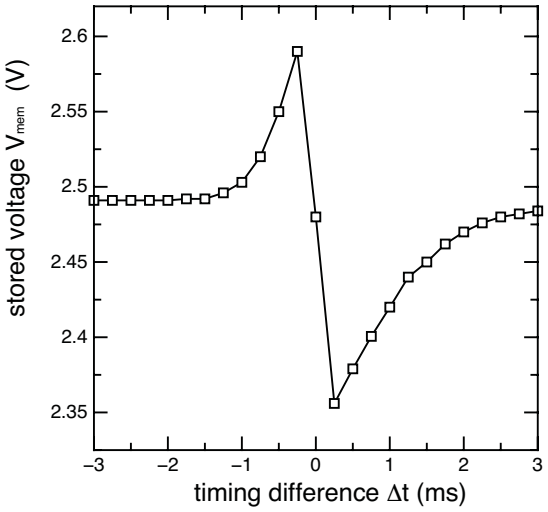


Figure 5.12: Simulation results of spike-timing dependent plasticity circuit

where κ represents the effectiveness of the gate potential, and $V_T \equiv kT/q \approx 26$ mV at room temperature (k is Boltzmann's constant, T the temperature, and q the electron charge) [68], [69].

Figure 5.12 shows simulation results of the proposed STDP circuit. The horizontal and vertical axes represent $t_{\rm post} - t_{\rm pre}$ (Δt) and capacitor voltage $V_{\rm mem}$. The power-supply and reference voltages were set at 5 and 2.5 V. The memory capacitance value was set at 500 fF. As expected, the circuit mimicked basic characteristics of STDP learning; however, the asymmetry characteristic was observed. This is simply due to the unbalanced saturating properties of pMOS and nMOS current sources in Fig. 5.11, which could be improved by using relatively long channels for the current sources.

5.5 Summary

A neural network circuit to demonstrate synchronization among neuron with depressing synapse circuits was designed. The key to synchronizing neurons precisely was introducing positive feedback to the neurons and using depressing synapses instead of nondepressing (conventional) synapses for the feedback connections. Consequently, precision was improved by 60% when depressing synapse circuits were used instead of nondepressing synapses. Furthermore, a novel synapse circuit that qualitatively mimics spike-timing dependent plasticity (STDP) learning characteristics was designed. By circuit simulations, the learning characteristics were demonstrated.

Chapter 6

Sensory segmentation

One of the most challenging problems in sensory information processing is the analysis and understanding of natural scenes, i.e., images, sounds, etc. These scenes can be decomposed into coherent "*segments*". The segments correspond to different components of the scene. Although this ability, generally known as sensory segmentation, is performed by the brain with apparent ease, the problem remains unsolved. Several models that perform segmentation have been proposed [70]-[73], but they are often difficult to implement in practical integrated circuits. A neural segmentation model called LEGION (Locally Excitatory Globally Inhibitory Oscillator Networks) [73], can be implemented on LSI circuits [74]. However, the LEGION model fails to work in the presence of noise. Our model solves this problem by including spike-timing dependent plasticity (STDP) learning with all-to-all connections of neurons.

In this chapter it is presented a simple neural segmentation model that is suitable for analog CMOS circuits. The segmentation model is suitable for applications such as figure-ground segmentation and the cocktail-party effect, etc.

The model consists of mutually coupled (all-to-all) neural oscillators that exhibit synchronous (or asynchronous) oscillations. All the neurons are coupled with each other through positive or negative synaptic connections. Each neuron accepts external inputs, e.g., sound inputs in the frequency domain, and oscillates (or does not oscillate) when the input amplitude is higher (or lower) than a given threshold value. The basic idea is to strengthen (or weaken) the synaptic weights between synchronous (or asynchronous) neurons, which may result in phase-domain segmentation. The synaptic weights are updated based on symmetric STDP using Reichardt's correlation neural network [65] which is suitable for analog CMOS implementation.

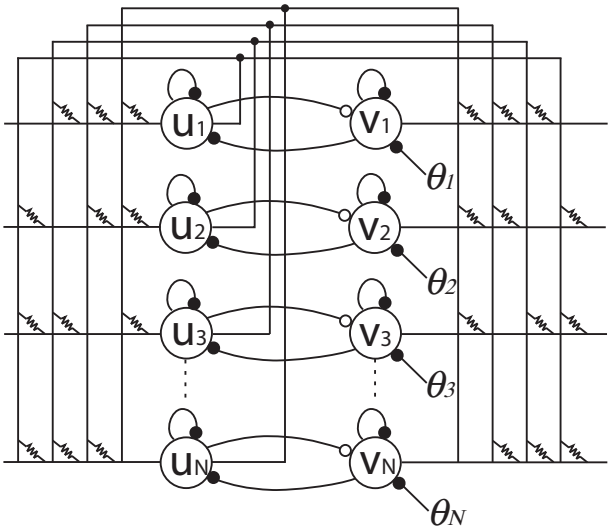


Figure 6.1: Network construction of segmentation model.

6.1 Model and basic operation

The segmentation model is illustrated in Fig. 6.1. The network has N neural oscillators consisting of the Wilson-Cowan type activator and inhibitor pairs (u_i and v_i) [20]. All the oscillators are coupled with each other through resistive synaptic connections, as illustrated in the figure. The dynamics are defined by

$$\tau \frac{du_i}{dt} = -u_i + f_{\beta_1}(u_i - v_i) + \sum_{j \neq i}^{N} W_{ij}^{\mathrm{uu}} u_j, \tag{6.1}$$

$$\frac{dv_i}{dt} = -v_i + f_{\beta_2}(u_i - \theta_i) + \sum_{j \neq i}^{N} W_{ij}^{\mathrm{uv}} u_j, \tag{6.2}$$

where τ represents the time constant, N the number of oscillators, θ_i the external input to the i-th oscillator. $f_{\beta_i}(x)$ represents the sigmoid function defined by $f_{\beta_i}(x) = [1 + \tanh(\beta_i x)]/2$, W_{ij}^{uu} the connection strength between the i-th and j-th activators and W_{ij}^{uv} the strength between the i-th activator, and the j-th inhibitor.

According to the stability analysis in [48], the i-th oscillator exhibits excitable behaviors when $\theta_i < \Theta$ where $\tau \ll 1$ and $\beta_1 = \beta_2$ ($\equiv \beta$), where Θ is

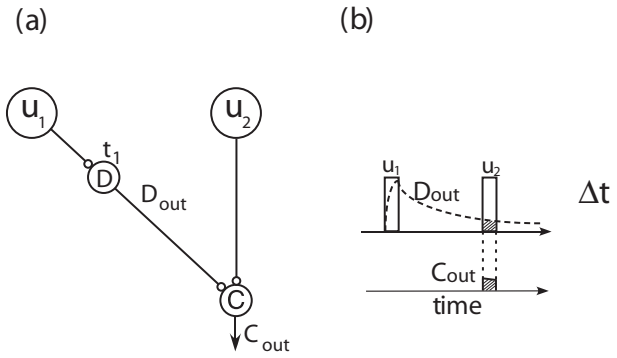


Figure 6.2: Reichardt's correlation network.

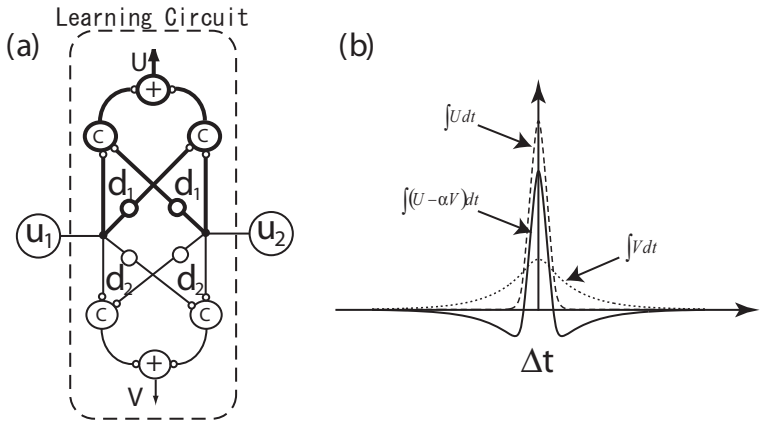


Figure 6.3: Learning characteristic: Reichardt's correlation.

given by

$$\begin{aligned} \Theta &= u_0 - \frac{2}{\beta}\tanh^{-1}(2v_0 - 1), \\ u_0 &\equiv \frac{1-\sqrt{1-4/\beta}}{2}, \\ v_0 &\equiv u_0 - \frac{2}{\beta}\tanh^{-1}(2u_0 - 1), \end{aligned} \tag{6.3}$$

and exhibits oscillatory behaviors when $\theta_i \geq \Theta$, if W_{ij}^{uu} and W_{ij}^{uv} for all i and j are zero.

Suppose that neurons are oscillating ($\theta_i \geq \Theta$ for all i) with different initial phases. The easiest way to segment these neurons is to connect the activators belonging to the same (or different) group with positive (or negative) synaptic weights. In practical hardware, however, the corresponding neuron devices have

to be connected by special devices having both positive and negative resistive properties, which prevents us from designing practical circuits. Therefore, it is better to simply use positive synaptic weights between activators and inhibitors, and do not use negative weights. When the weight between the i-th and j-th activators (W_{ij}^{uu}) is positive and W_{ij}^{uv} is zero, the i-th and j-th activators will be synchronized. Contrarily, when the weight between the i-th activator and the j-th inhibitor (W_{ij}^{uv}) is positive and W_{ij}^{uu} is zero, the i-th and j-th activators will exhibit asynchronous oscillation because the j-th inhibitor (synchronous to the i-th activator) inhibits the j-th activator.

The synaptic weights (W_{ij}^{uu} and W_{ij}^{uv}) are updated based on our assumption; one neural segment is represented by synchronous neurons, and is asynchronous with respect to neurons in the other segment. In other words, neurons should be correlated (or anti-correlated) if they received synchronous (or asynchronous) inputs. These correlation values can easily be calculated by using Reichardt's correlation neural network [65] which is suitable for analog circuit implementation [75]. The basic unit is illustrated in Fig. 6.2(a). It consists of a delay neuron (D) and a correlator (C). A delay neuron produces blurred (delayed) output D_{out} from spikes produced by activator u_1. The dynamics are given by

$$d_1 \frac{dD_{\mathrm{out}}}{dt} = -D_{\mathrm{out}} + u_1, \tag{6.4}$$

where d_1 represents the time constant. The correlator accepts D_{out} and spikes produced by activator u_2 and outputs $C_{\mathrm{out}} = D_{\mathrm{out}} \times u_2$. The conceptual operation is illustrated in Fig. 6.2(b). Note that C_{out} qualitatively represents correlation values between activators u_1 and u_2 because C_{out} is decreased (or increased) when Δt, inter-spike intervals of the activators, is increased (or decreased). Since this basic unit can calculate correlation values only for positive Δt, two basic units are used, called an *unitpair*, as shown by thick lines in Fig. 6.3(a). The output (U) is thus obtained for both positive and negative Δt by summing the two C_{out}s. Through temporal integration of U, impulse responses of this unit pair can be obtained. The sharpness is increases as $d_1 \to 0$. Introducing two unit pairs with different time constants, i.e., d_1 and d_2 ($\gg d_1$), one can obtain those two impulse responses (U and V) simultaneously. The impulse responses (U and V) are plotted in Fig. 6.3(b) by a dashed and a dotted line, respectively. The weighted subtraction ($U - \alpha V$) produces well-known Mexican hat characteristics, as shown in Fig. 6.3(b) by a solid line. This symmetric characteristic is used for the weight updating as a spike-timing dependent plasticity (STDP) in the oscillator network.

The learning model is shown in Fig. 6.4(a). The learning circuit is located between two activators u_1 and u_2. The two outputs (U and V) of the learning

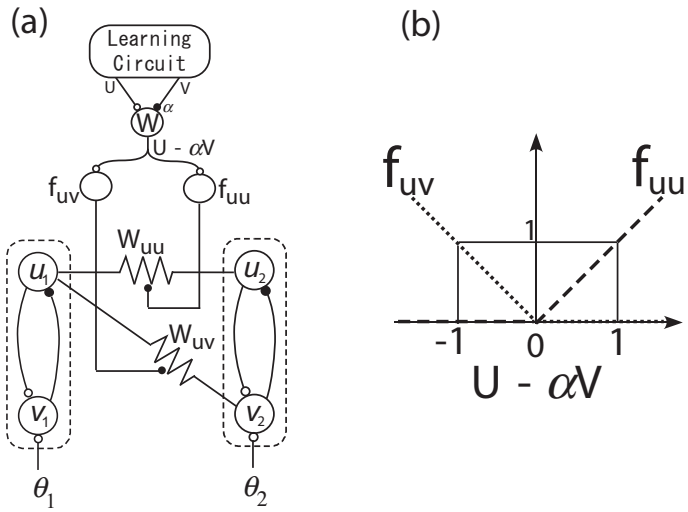


Figure 6.4: spike-timing dependent plasticity (STDP) learning Model.

circuit are given to interneuron W which performs subtraction $U-\alpha V$. According to the above assumptions for neural segmentation, when $U-\alpha V$ is positive, the weight between activators u_1 and u_2 (illustrated by a horizontal resistor symbol in Fig. 6.4(a)) is increased because the activators should be correlated. On the other hand, when $U-\alpha V$ is negative, the weight between activator u_1 and inhibitor v_2 (illustrated by a slant resistor symbol in Fig. 6.4(a)) is increased because activators u_1 and u_2 should be anti-correlated. To this end, the output of interneuron W is given to two additional interneurons ($f_{\rm uu}$ and $f_{\rm uv}$). The input-output characteristics of these interneurons are shown in Figs. 6.4(b). Namely, $f_{\rm uu}$ (or $f_{\rm uv}$) increases linearly when positive (or negative) $U-\alpha V$ increases, but is zero when $U-\alpha V$ is negative (or positive). Those positive outputs ($f_{\rm uu}$ and $f_{\rm uv}$) are given to the weight circuit to modify the positive resistances. The dynamics of the "positive" weight between activators u_i and u_j is given by

$$\frac{dW_{ij}^{\rm uu}}{dt} = -W_{ij}^{\rm uu} + f_{\rm uu}, \tag{6.5}$$

and the "positive" weight between activator u_i and inhibitor v_j is

$$\frac{dW_{ij}^{\rm uv}}{dt} = -W_{ij}^{\rm uv} + f_{\rm uv}. \tag{6.6}$$

Numerical simulations with $N=6$, $\tau=0.1$, $\beta_1=5$, $\beta_2=10$, $d_1=2$, $d_2=0.1$ and $\alpha=1.2$ were carried out. Time courses of activators u_i ($i=1\sim 6$)

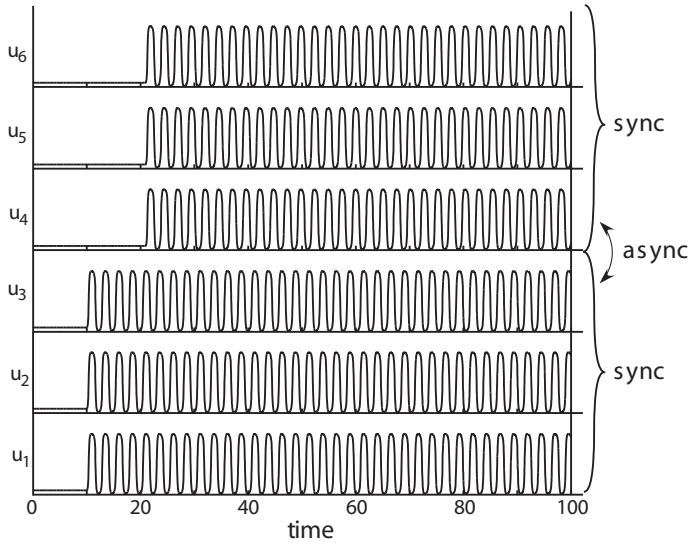


Figure 6.5: Numerical simulation results.

are shown in Fig. 6.5. Initially, the external inputs θ_i ($i = 1 \sim 6$) were zero ($< \Theta$), but θ_i for $i = 1 \sim 3$ and $i = 4 \sim 6$ were increased to 0.5 ($> \Theta$) at $t = 10$ s and 20.9 s, respectively. It can be observed that $u_{1\sim3}$ and $u_{4\sim6}$ were gradually desynchronized without breaking synchronization amongst neurons in the same group, which indicated that segmentation of neurons based on the input timing was successfully achieved.

In addition, numerical simulations to evaluate the "segmentation ability," which represents the number of survived segments after the learning were carried out. The number of segments as a result of the network's learning strongly depends on the STDP characteristic as well as the input timing of neurons (Δt). Let us remember that neurons that fire "simultaneously" should be correlated. "Simultaneously" is to be defined by some "time windows of coincidence" that here is called σ_{STDP}. Thus, neurons that receive inputs within the time windows should be correlated. Simulation results are shown in Fig. 6.6. The number of neurons (N) was set to 50. The neurons received random inputs within time $t_{\mathrm{in}}^{\max}$ (maximum input timing). It can be observed that when σ_{STDP} was 1 and neurons received their inputs within time 2, the number of segments was about 2. The contrary was observed when σ_{STDP} was 0.1 and $t_{\mathrm{in}}^{\max}$ was 10, where the number of segments was about 35.

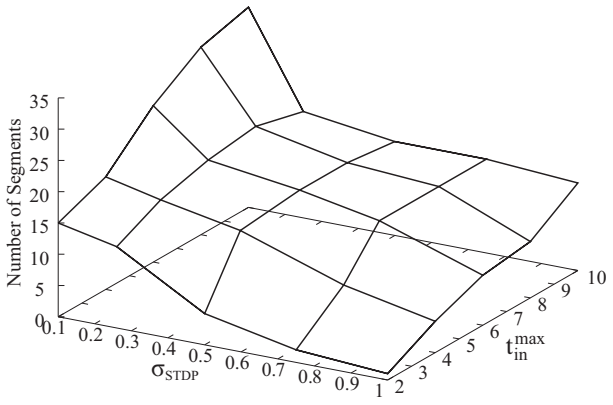


Figure 6.6: simulation results showing segmentation ability of the network

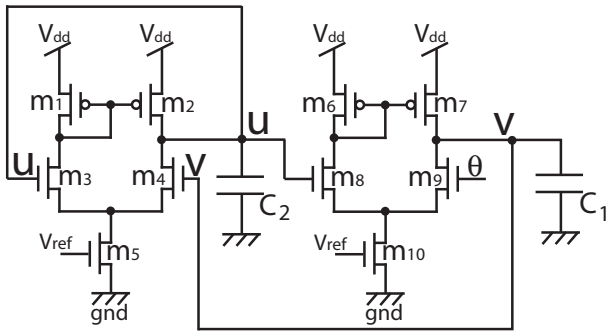


Figure 6.7: Unit circuits for neural segmentation

6.2 Circuit implementation

The construction of a single neural oscillator is illustrated in Fig. 6.7. The oscillator consists of two differential pairs (m_3-m_4 and m_8-m_9), two current mirrors (m_1-m_2 and m_6-m_7), bias transistors (m_5 and m_10); and two additional capacitors (C_1 and C_2). To explain the basic operation of the neural oscillator, let us suppose that W_{uu} and W_{uv} in Eqs. (6.1) and (6.2) are zero. Now in Eq. (6.1), when u is larger than v ($u > v$) u tends to increase and approach to 1 (vdd), on the contrary, when u is lower than v ($u < v$) u tends to decrease and approach to 0 (gnd). The same analysis can be apply to Eq. (6.2). When u is larger than θ ($u > \theta$) v tends to increase approaching to (vdd), and, when u is lower than θ ($u < \theta$) v tends to decrease and approaching to (gnd).

The simulated nullclines of a single neuron circuit for different θs (0.5 V and 2.5 V) and trajectories for $\theta = 2.5$ V with $C_1 = 10$ pF and $V_{\text{ref}} = 2$ V are shown in Fig. 6.8. Transient simulation results of the neuron circuit are

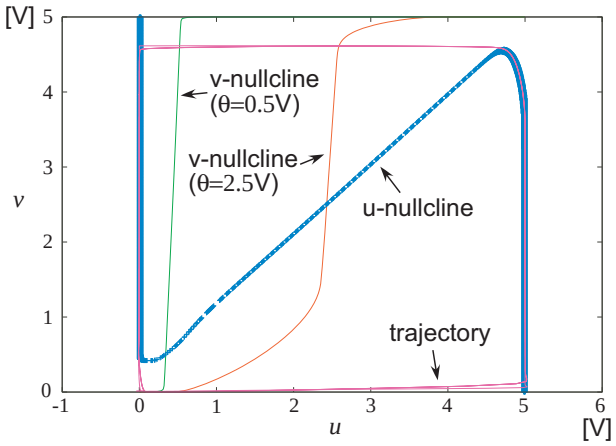


Figure 6.8: Nullclines and trajectory for $\theta = 2.5$ V obtained from circuit simulations.

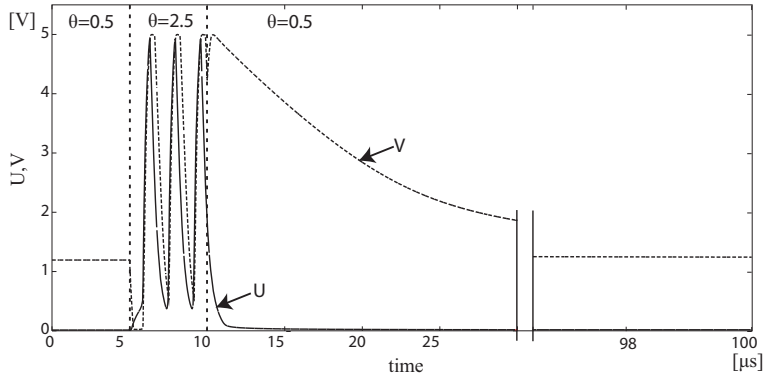


Figure 6.9: Simulation results of neural oscillator.

shown in Fig. 6.9. The parameter used for the transistors were obtained from MOSIS AMIS 1.5-μm CMOS process. All transistor sizes were fixed at $L = 1.6$ μm and $W = 4$ μm, the capacitors (C_1 and C_2) were set at 0.1 pF, and the differential amplifier's V_{ref} was set at 0.7 V, and the supply voltage was set at 5 V. Time courses of the activator unit (u) and (v) are shown. Initially, θ was set at 0.5 V (in relaxing state), and neither u nor v oscillated, instead u they are in equilibrium. Then θ was increased to 2.5 V at $t = 5$ μs, and both u and v exhibited oscillations with small phase difference between them. Again, θ was set at 0.5 V at $t = 10$ μs and u relaxed, and v to a high value (around V_{dd}) and decreases with time until it reach equilibrium, as expected.

A circuit implementing Reichardt's basic unit shown in Fig. 6.2(a) is shown in Fig. 6.10. Bias current I_1 drives m_6. Transistor m_5 is thus biased to gener-

ate I_1 because m_5 and m_6 share the gates. When m_3 is turned on (or off) by applying $V_{\rm dd}$ (or 0) to u_1, I_1 is (or is not) copied to m_1. Transistors m_1 and m_2 form a current mirror, whereas m_2 and m_4 form a pMOS source-common amplifier whose gain is increased as $V_{\rm b1} \rightarrow 0$. Since the parasitic capacitance between the source and drain of m_2 is significantly amplified by this amplifier, temporal changes of u_1 are blurred on the amplifier's output ($D_{\rm out}$). Therefore this "delayer" acts as a delay neuron in Fig. 6.2(a). A correlator circuit consists of three differential amplifiers (m_{12}-m_{13}, m_{14}-m_{15} and m_{16}-m_{17}), a pMOS current mirror (m_{19}-m_{20}), a bias transistor (m_{18}) and a bias current source (I_2). In this circuit, m_{12}, m_{14} and m_{17} are floating gate transistors. They reduce voltages of D_{out} and u_2 to $D_{out}/10$ and $u_2/10$ because the input gate sizes were designed to 'capacitively' split the input voltages with the ratio of 1:10. The output current of differential pair m_{14}-m_{15} is:

$$I_{out} = I_2 f(D_{out}/10) f(u_2/10), \tag{6.7}$$

where $f(x)$ is the sigmoid function given by $f(x) = 1/(1 + e^{-x})$. Current I_{out} is regulated by the bias transistor m_{18}. The result is copied to m_{20} through current mirror m_{19}-m_{20}. This operation corresponds to that of a correlator in Fig. 6.2(a).

Circuit simulations of the above circuits were conducted. The parameter sets used for the transistors were obtained from MOSIS AMIS 1.5-μm CMOS process. Transistor sizes of all nMOS and pMOS m_9, m_{10} and m_{18} were fixed at $L = 1.6$ μm and $W = 4$ μm pMOS transistors m_1, m_2, m_{19} and m_{20} were fixed at $L = 16$ μm and $W = 4$ μm. The supply voltage was set at 5 V.

Simulation results of the STDP circuits are shown in Fig. 6.11. Parameters V_{b1}, V_{b2} and V_{b3} were set at 0.41 V, 0.7 V and 4.1 V, respectively. The value of V_{b1} was chosen so that the delayer makes a reasonable delay. Horizontal axes (Δt) in Fig. 6.11 represent time intervals of input current pulses (spikes). Voltage pulses (amplitude: 5 V, pulse width: 10 ms) were applied as u_1 and u_2 in Fig. 6.10. Capacitance $C_{\rm out}$ was integrated during the simulation and the normalized values were plotted [(a) in Fig. 6.11]. Then the value of V_{b_1} was changed to 0.37 V. The lowered V_{b1} reduced the drain current of m_4 and made the delay larger. Again, $C_{\rm out}$ was integrated and normalized. The result is plotted [(b) in Fig. 6.11]. By subtracting (b) from tripled (a), it was obtained the STDP learning characteristic (c) in Fig. 6.11.

Simulations for testing the synaptic weights of two coupled neural oscillators were made. Figure 6.12(a) shows the two oscillators with all the synaptic connections. The oscillation of neurons u_1 and u_2 without applying any connection between them (V_{gs}=0 V for W_{uu} and W_{uv}) are shown in Fig. 6.12(b) where

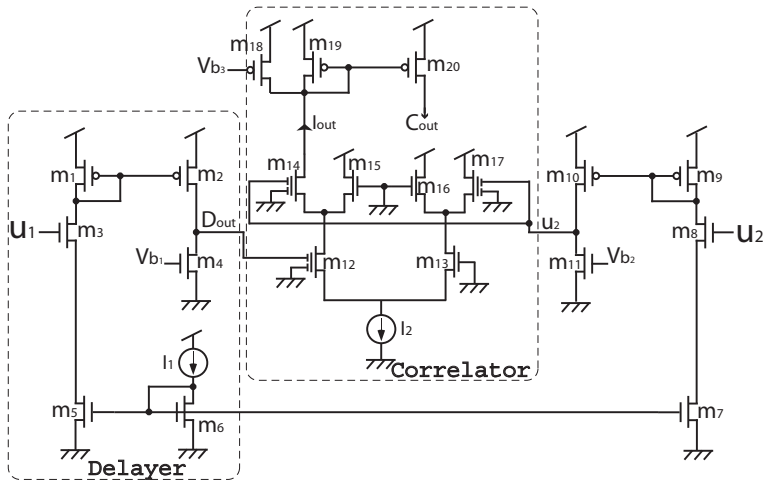


Figure 6.10: spike-timing dependent plasticity circuit.

the neurons oscillated independently. nMOS transistors with $L = 1.6$ μm and $W = 4$ μm were used as synaptic weight W_{uu} and W_{uv}, Fig. 6.12(a) shows the excitatory connection W_{uu} between neurons u_1 and u_2, and inhibitory connections W_{uv} between neurons $u_{1,2}$ and $v_{2,1}$. The oscillations of neurons u_1 and u_2 when applying an excitation through W_{uu} (the gate voltage of W_{uu} was set at 1 V and 0 V for W_{uv}) are shown in Fig. 6.13(a), in this case both neurons synchronized. On the contrary, when applying an inhibition through W_{uv} (the gate voltage of W_{uv} was set at 0.6 V and 0 V for W_{uu}) the neurons oscillated asynchronously as shown in Fig. 6.13(b).

A basic circuit implementing the interneurons (W, f_{uu} and f_{uv}) is shown in Fig. 6.14. The circuit consists only of current mirrors. Input current U (from Reichardt's circuit; correlation circuit) is copied to m_3 by current mirror m_1-m_3, and is copied to m_8 by current mirrors m_1-m_2 and m_7-m_8. At the same time, input current V is copied to m_6 by current mirror m_4-m_6, and is copied to m_{12} by current mirrors m_4-m_5 and m_{11}-m_{12}. Recall that we need the subtraction of $U - \alpha V$ to produce the Mexican-hat characteristic. Therefore, the weight (α) were set as $\alpha \equiv W_5/L_5 \cdot L_4/W_4 = W_6/L_6 \cdot L_4/W_4$, where W_i and L_i represent the channel width and length of transistor m_i, respectively. So, when current U is higher than current αV, current f_{uu} is outputted by current mirror m_{13}-m_{14}. Otherwise, current f_{uv} is outputted by current mirror m_{11}-m_{12}.

Circuit simulations for the interneuron circuit were carried out. Transistors sizes (W/L) were 4 μm/1.6 μm for m_1-m_4, 10 μm/1.6 μm for m_5 and m_6, 4.5 μm/16 μm for m_8 and m_{12}, 3.5 μm/16 μm for m_{13}, and 4 μm/16 μm for the rest transistors. The supply voltage was set to 5 V. Input current V was set to

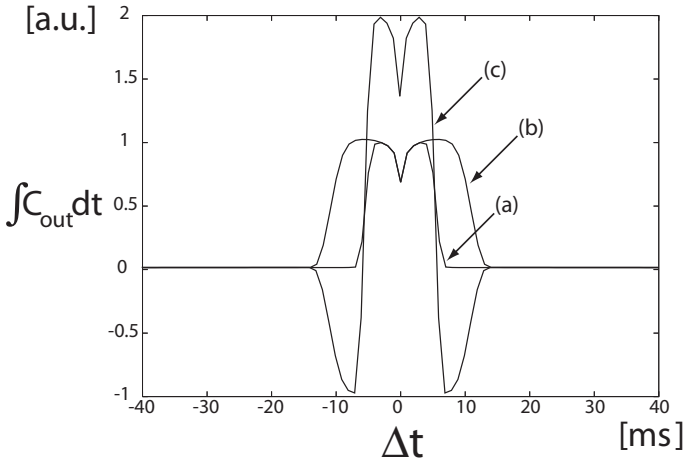


Figure 6.11: spike-timing dependent plasticity characteristics

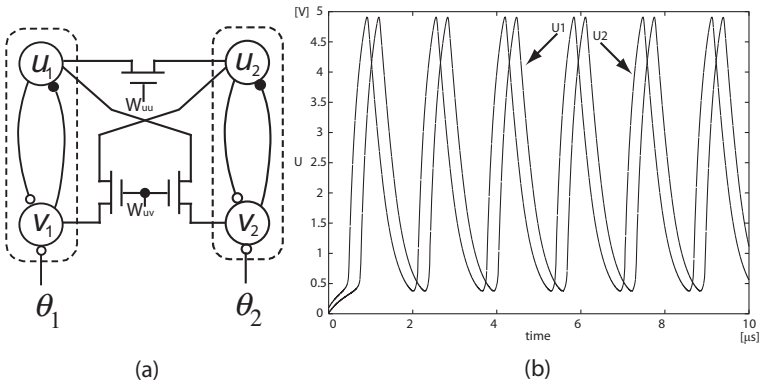


Figure 6.12: (a) Coupled neural oscillators (b) u_1 and u_2 oscillations.

100 nA, and input current U varied from 0 to 200 nA. The simulation results are shown in Fig. 6.15. When $U + \Delta I < V$ where $\Delta I \approx 20$ nA, output current f_{uv} flowed and f_{uu} was 0. When $U - V < \Delta I$, both f_{uu} and f_{uv} were 0. When hen $U - \Delta I > V$, f_{uu} flowed while f_{uv} remained at 0.

Next, circuit simulations of the circuit network with $N = 6$ were conducted. Transistor sizes (W/L) for the Recichardt's basic circuit (see Fig. 6.10) were 4 μm/1.6 μm for nMOS transistors and m_{20}, and 4 μm/16 μm for the rest of the transistors. Voltages V_{b2} and V_{b3} were set to 550 mV and 4.08 V respectively, while V_{b1} was set to 510 mV for delay τ_{d1}, and was set to 430 mV for delay τ_{d2}. With these settings, it was obtained positive W $(U - \alpha V)$ for $|\Delta t| \leq 1$ μs, and obtained negative W for $|\Delta t| > 1$ μs. In other words, when $|\Delta t| \leq 1$ μs, neurons should be correlated, otherwise, they should be anti-correlated, as explained before.

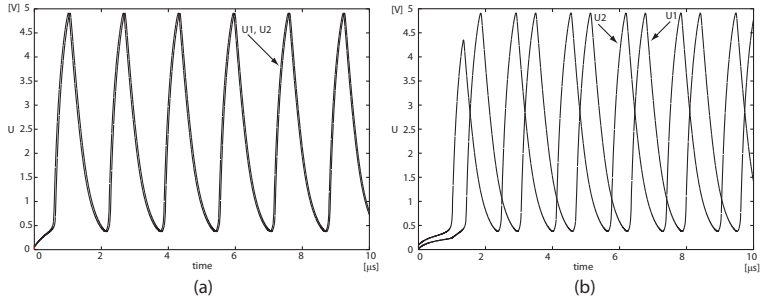


Figure 6.13: oscillation of neurons u_1 and u_2 when (a)excitation is applied and (b) inhibition is applied.

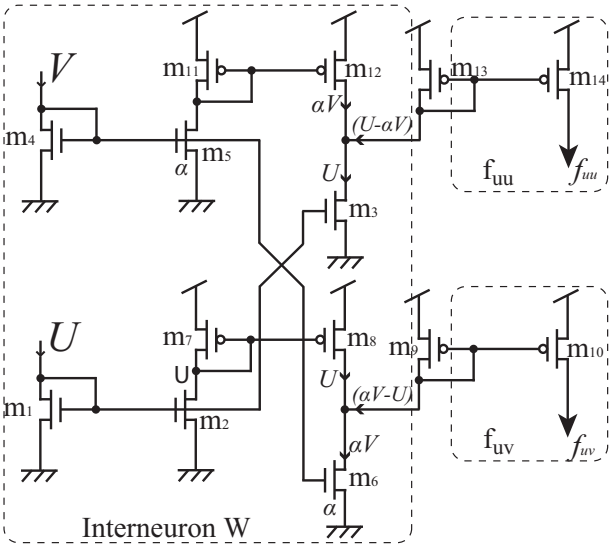


Figure 6.14: interneuron circuit

The normalized time courses of u_is ($i = 1 \sim 6$) are shown in Figs. 6.16(a) and (b). As shown in Fig. 6.16(a), at $t = 0$, external inputs θ_i ($i = 1 \sim 6$) were 2.5 V, which is equivalent to Δt=0. It can be observed that all neurons were gradually synchronized. On the contrary, Fig. 6.16(b) shows that at $t = 0$ external inputs $\theta_{1,2,3}$ were set to 2.5 V, and inputs $\theta_{4,5,6}$ were set to 0. Then, at $t = 3$ μs $\theta_{4,5,6}$ were set to 2.5 V, which is equivalent to $\Delta t = 3$ μs. Observed that $u_{1,2,3}$ and $u_{4,5,6}$ were desynchronized without breaking synchronization among neurons in the same group that were gradually synchronized. This indicated that segmentation of neurons based on the input timing was successfully achieved.

To consider the noise tolerance of the network, Monte-Carlo simulations were conducted in a circuit network with $N = 3$. The parameter V_{th} (threshold volt-

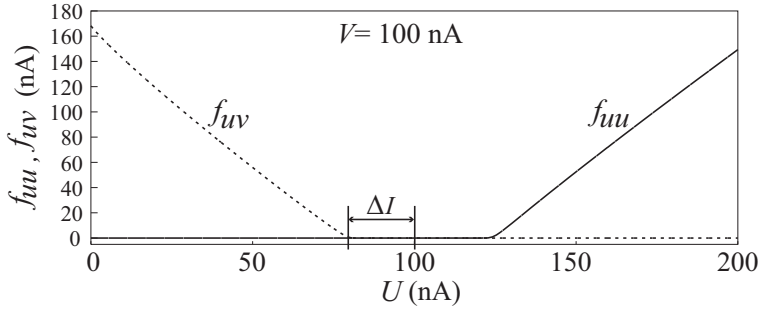


Figure 6.15: circuit simulation results of interneuron circuit

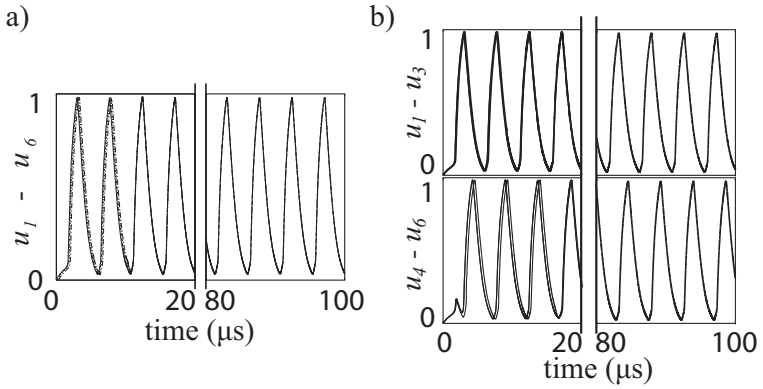


Figure 6.16: circuit simulation results for a) inter-spike interval $\Delta t = 0$, and b) $\Delta t = 3$ μs.

age) of all transistors was varied using Gaussian noises with standard deviation $\sigma_{\rm VT}$. When $t = 0$, external inputs to neurons (θ_1, θ_2, θ_3) were set to (2.5,0,0)V. Then, at $t = 1$ μs, (θ_1, θ_2, θ_3) were set to (2.5,2.5,0)V, whereas they were set to (2.5,2.5,2.5)V at $t = 2.4$ μs. In other words, neurons u_1 and u_2 should be synchronous with each other, and they should be asynchronous with u_3 because of Δt=1.4 μs. To evaluate the performance of the network, the correlation values C_{ij} between neurons u_i and u_j were calculated, given by

$$C_{ij} = \frac{\langle u_i u_j \rangle - \langle u_i \rangle \langle u_j \rangle}{\sqrt{\langle u_i^2 \rangle - \langle u_i \rangle^2} \sqrt{\langle u_j^2 \rangle - \langle u_j \rangle^2}}. \tag{6.8}$$

Correlation values C_{12} and C_{13} were calculated to evaluate the synchronicity between segments. Figures 6.17 and 6.18 show the simulation results. As observed in the figures, when $\sigma_{\rm VT}$ <10 mV neurons u_1 and u_2 were correlated, while the correlation value (C_{13}) between neurons u_1 and u_3 was low, *i.e.*, they

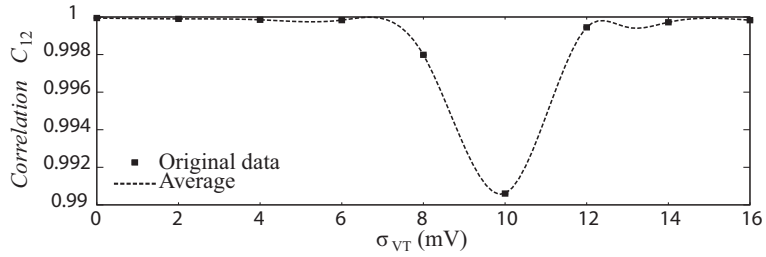


Figure 6.17: correlation values between neurons u_1 and u_2 for different σ_{VT}.

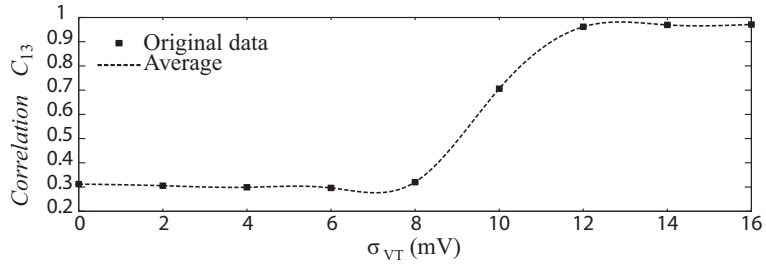


Figure 6.18: correlation values between neurons u_1 and u_3 for different σ_{VT}.

were anti-correlated. Due to imperfections of the CMOS fabrication process, device parameters, *e.g.*, threshold voltage, etc., suffer large variations [76]. These variations among transistors cause a significant change in general analog circuits. Nevertheless, the results obtained in Figs. 6.17 and 6.18 showed that our network successfully segmented neurons for σ_{VT}s lower than 10 mV, which indicated that the network is tolerant to threshold mismatch among transistors.

6.3 Summary

A neural segmentation model that is suitable for circuit implementation was proposed. In order to facilitate the implementation of the model, instead of employing negative connections required for anti-correlated oscillation among different segments, positive connections between activators and inhibitors among different neuron units were used. The segmentation ability of the network through numerical simulations was evaluated. The operation of the circuit network using six neurons was demonstrated. Finally, the effect of threshold mismatches among transistors in the network with three oscillators was explored, the results showed that the network was tolerant to device mismatches.

Chapter 7

Storage of temporal sequences

The brain has an ability to process information whose content changes over time. Therefore, it is necessary that systems, whether natural or artificial, have the ability to process information whose content depends on temporal order of events. Studies on neuroimaging have provided evidence that the prefrontal cortex of the brain is involved in temporal sequencing [77]. Furthermore, studies on the olfactory bulb have shown that information in biological networks takes the form of space-time neural activity patterns [78] [79].

Patterns whose content depends on time are commonly called *temporal sequence learning*. 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 to retrieve them is also needed. Neural networks for processing temporal sequences are usually based on multilayer perceptron or on the Hopfield models [80]. In [81], 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 sequences processing have been employed in neuromorphic VLSIs to mimic the early visual processing [82] and an associative memory functions [83].

This chapter, focuses on the implementation of such kind of temporal-coding neural networks with analog metal-oxide-semiconductor (MOS) devices. In [84], Fukai proposed a model for the storage of temporal sequences. In the model, the Walsh series expansion [85] was utilized to represent the input signal by linear superposition of rectangular periodic functions with different fundamen-

tal frequencies generated by an oscillatory subsystem. Often, the developments of mathematical models for simulating large-scale neural networks are suffering from problems of computer load (simulation time). This is avoided with the employment of analog MOS circuits permitting real-time emulation of large-scale networks, because they are designed so that the circuit dynamics correspond to the equation in the mathematical model. 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 which are 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 oscillatory cells overlap (or do not overlap) with the input sequence.

7.1 Model

Fukai proposed a model for the storage of temporal sequences in [84]. The main purpose of this model is learning and recalling the temporal input stimuli. The model consists of an input unit which gives a trigger signal to the oscillatory subsystem. The oscillatory subsystem has N oscillatory subunits and an array of modifier cells. Each of the oscillatory subunits consists of a pair of excitatory and inhibitory neural cells based on the Wilson-Cowan system [21], 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 between the output cell and the modifier cells. The output cell sums up all the outputs of the modifier cells to recall the input signal according to the Walsh function series [85].

Based on the Fukai's model, a modified model for learning and recalling temporal sequences that is suitable for implementation with MOS circuits is proposed. The modified model is shown in Fig. 7.1. 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 for the purpose of improving the accuracy of the input-output approximation after each learning cycle [84]. In the modified model, these modifier cells were eliminated. Instead neural oscillators which exhibit periodic square-wave oscillations are used. Therefore, the modification of thresholds in modifier cells is not carried out, which results in reducing accuracy of the learning in

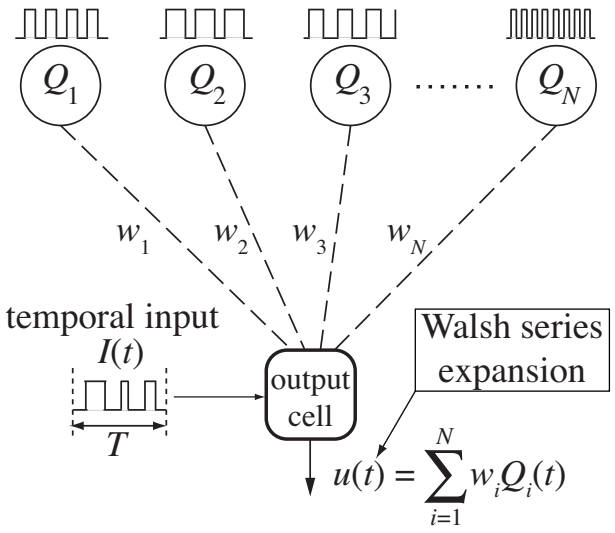


Figure 7.1: Proposed temporal coding model.

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's outputs as

$$u(t) = \sum_{i=1}^{N} w_i Q_i(t). \tag{7.1}$$

Through cyclic learning processes, w_is in Eq. (7.1) are updated at every cycle to achieve $u(t) \rightarrow I(t)$. Notice 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 [85] 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 $(u(t))$, the mean square error (E) between them is defined as:

$$E = \frac{1}{2T} \int_{jT}^{(j+1)T} [I(t) - u(t)]^2 \, dt \quad (j = 0, 1, 2, \cdots), \tag{7.2}$$

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)

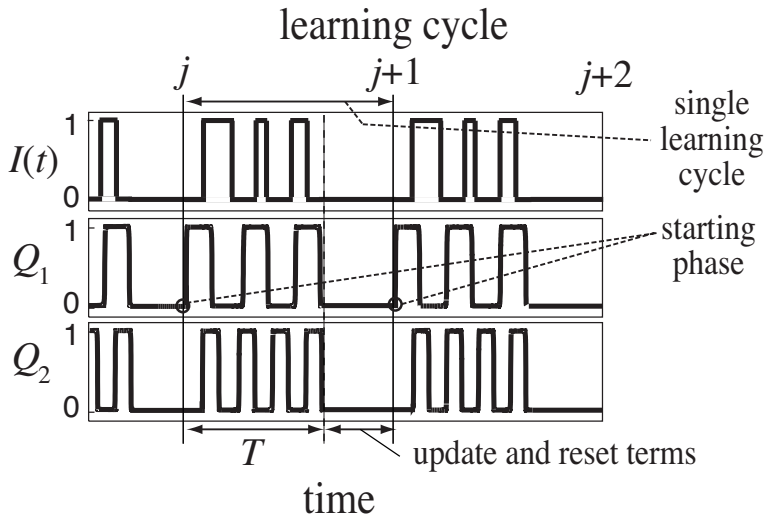


Figure 7.2: Definition of single learning cycle.

between the oscillators and the output cell according to the gradient descent rule:

$$\delta w_i = -\eta \partial E / \partial w_i, \tag{7.3}$$

where η represents a small positive constant indicating the learning rate. Substituting E in Eq. (7.2) into Eq. (7.3), we obtain

$$\delta w_i = \frac{\eta}{T} \int_{jT}^{(j+1)T} [I(t) - u(t)] Q_i(t)\, dt. \tag{7.4}$$

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

$$w_i^{\text{new}} = w_i^{\text{old}} + \delta w_i. \tag{7.5}$$

The procedures above, *i.e.*, numerical calculations of Eqs. (7.1), (7.4) and (7.5), are repeated ($j = 0, 1, \cdots$) until the error between the input and the output becomes small enough.

Because the model is meant for hardware implementation, it is necessary to take physical time for updating the weights (Eq. (7.5)) and resetting the integrated value in Eq. (7.4) before starting another learning cycle, although the updating and resetting terms are assumed to be zero in Eqs. (7.4) and (7.5). In practical hardware, a single learning cycle consists of the input sequence's length (T), the updating and resetting terms, as shown in Fig. 7.2. Note that each

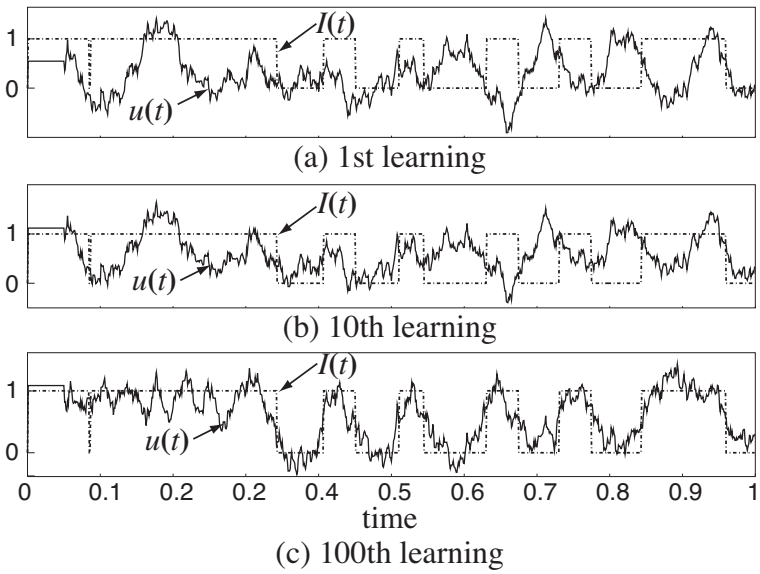


Figure 7.3: Input ($I(t)$) and output sequences ($u(t)$) of proposed network with 200 oscillatory units after first (a), 10th (b) and 100th learning (c).

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. 7.2 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 thus 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:

$$Q_i(t) = H[\sin(2\pi f_i t)] \tag{7.6}$$

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

$$H(x) = \begin{cases} 1 & (x > 0) \\ \\ 0 & (x < 0) \end{cases} . \tag{7.7}$$

The results are shown in Fig. 7.3 ($N = 200$, $T = 1$ and $\eta = 0.01$). After the first learning (Fig. 7.3(a)), the input ($I(t)$) and the output sequences ($u(t)$) were

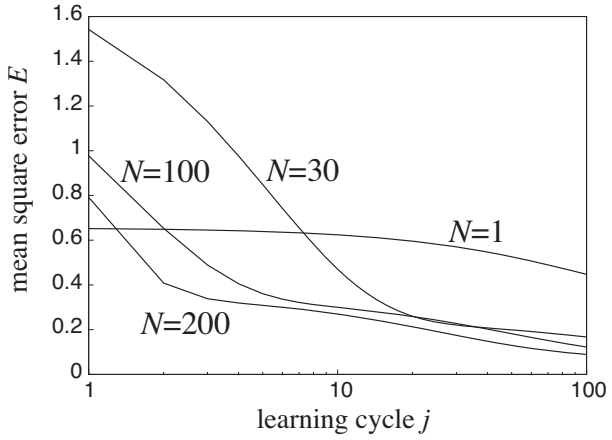


Figure 7.4: Time evolution of mean square errors.

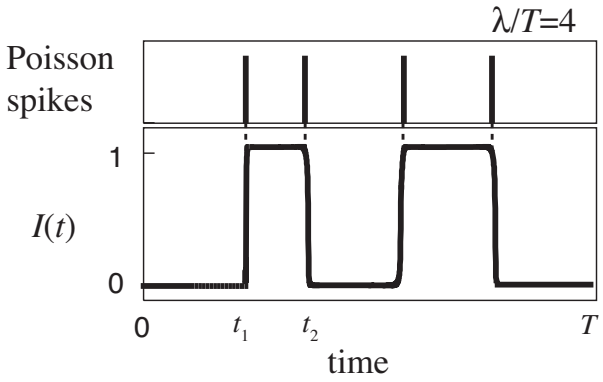


Figure 7.5: Input sequence ($I(t)$) generated by Poisson spikes with $\lambda = 4$.

completely different, however, $u(t)$ approached to $I(t)$ as repeating the learning (Figs. 7.3(b) for 10th and (c) for 100th learning).

Figure 7.4 shows time evolution of the mean square errors (E) of the proposed network with $N = 1$, 30, 100 and 200. The errors were decreased as the learning cycle (j) increased, as expected. Since the error values for $N = 30$, 100 and 200 approached to the same values (≈ 0.2), we may avoid implementing a large number of oscillators and synaptic connections on hardware. The error in our modified model being of ≈ 0.2 (N=100 with 100 learning cycles) was about two times higher than that of the original model (≈ 0.1 with N=100 with 100 learning cycles; [84]). Despite this difference the modified model is applicable in areas that do not require errorless learning, e.g., low-quality voice recording (learning) for mobile products, etc.

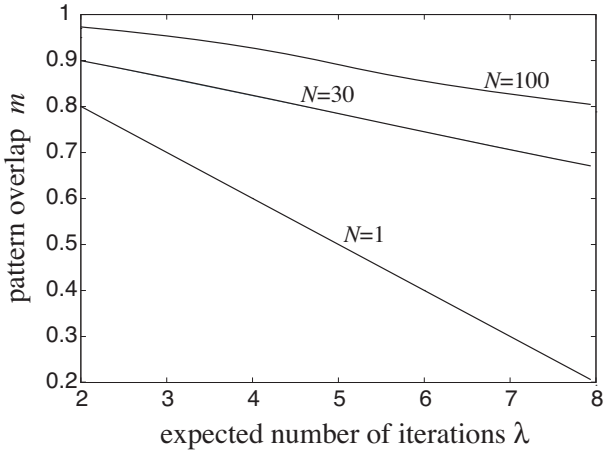


Figure 7.6: Pattern overlap between input and the output sequences.

Furthermore, the storage capacity of the proposed network was evaluated by defining pattern overlaps between the input and output sequences, as a function of N and complexity of input sequences. To define the complexity ($\equiv \lambda$), Poisson spikes whose mean firing rate is represented by λ were used. 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 were generated, whereas $I(t)$ $(t > t_1)$ is flipped to "1'' when a spike is generated at $t = t_1$. When the subsequent spike is generated at $t = t_2$, $I(t)$ $(t > t_2)$ is flipped to "0". Figure 7.5 shows the examples with $\lambda/T = 4$. This process is repeated while $t \leq T$

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

$$m \equiv \frac{1}{T}\int_0^T 2\left(I(t) - \frac{1}{2}\right) \times 2\left[H\left(u(t) - \frac{1}{2}\right) - \frac{1}{2}\right] dt, \tag{7.8}$$

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

Figure 7.6 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.

7.2 Circuit implementation

First, Wilson-Cowan oscillators [21] were use to implement the oscillator circuits. The dynamics are given by

$$\frac{du_i}{dt} = -u_i + f_\beta(u_i - v_i), \tag{7.9}$$

$$\frac{dv_i}{dt} = -v_i + f_\beta(u_i - \theta), \tag{7.10}$$

where u_i and v_i represent the system variables of the i-th oscillator, θ the threshold and $f_\beta(\cdot)$ the sigmoid function with slope β. Figure 7.7 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 time constants of the Wilson-Cowan system are very small, one can rewrite Eqs. (7.9) and (7.10) as

$$u_i \approx f_\beta(u_i - v_i), \tag{7.11}$$

$$v_i \approx f_\beta(u_i - \theta). \tag{7.12}$$

The OTA's output voltage (V_o) is expressed by $V_\mathrm{d} \cdot f(V_1 - V_2)$, while output voltage of the buffer circuit (V_o2) is given by $V_\mathrm{d} \cdot f(V_\mathrm{in} - V_\mathrm{th})$, where $f(\cdot)$ represents a nominal Sigmoid-like function and V_th the threshold voltage of the buffer circuit. Thus it was obtained

$$u_i = V_\mathrm{d} \cdot f(u_i - v_i), \tag{7.13}$$

$$v_i = V_\mathrm{d} \cdot f(u_i - V_\mathrm{th}), \tag{7.14}$$

by connecting the inputs and outputs to u_i and v_i as shown in Fig. 7.7 ($V_1 = V_\mathrm{o} = u_i, V_2 = v_i, V_\mathrm{in} = u_i, V_\mathrm{o2} = v_i$), which corresponds to Eqs. (7.11) and (7.12). Here v_i was used 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 on analog MOS circuits. Traditional circuits implement such bipolar weights by resistors with voltage mode neurons having positive- and negative-gain unity amplifiers. According to the sign of the weights, one of the amplifier must be selected. Implementing negative-gain unity amplifiers and the selection circuit

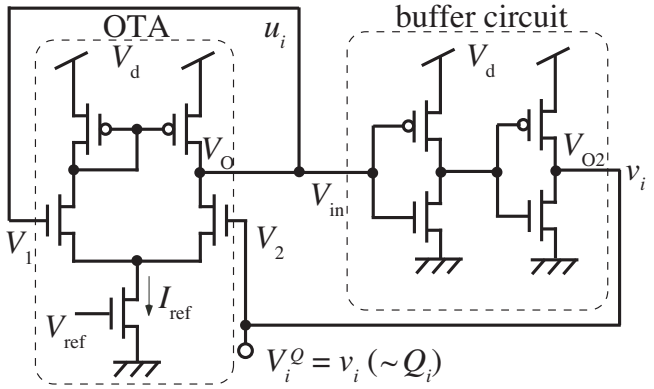


Figure 7.7: Neural oscillator circuit.

may occupy a large area on analog LSIs. Therefore a "current-mode circuits" was designed, 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 $V^{\rm m}$ that 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 to currents $I^{\rm p}$ and $I^{\rm m}$, and then wired. This setup is illustrated in Fig. 7.8(a). Now, the output current I is given by $I^{\rm p} - I^{\rm m}$ which is proportional to w, and I can take both positive ($I^{\rm p} > I^{\rm m}$) and negative currents ($I^{\rm p} < I^{\rm m}$). Based on this idea, we design a synapse circuit that connects the oscillator circuits and an output cell circuit. Figure 7.8(b) shows the concept of the i-th synapse circuit which calculates Eq. (7.1). Two ideal switches are inserted on the output lines of 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 7.8(c) illustrates a concept of the output cell that sums up all the output currents of the synapse circuits. Since $(I_i^{\rm p} - I_i^{\rm m})Q_i$ is represented by current, output current $I^u(t)$ flowing out from node A is

$$I^u(t) = \sum_{i=1}^{N} (I_i^{\rm p} - I_i^{\rm m}) Q_i(t), \tag{7.15}$$

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

Figure 7.9 illustrates a MOS circuit for the i-th synaptic circuit model shown in Fig. 7.8(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. 7.8(b)) with limited linear range. The amplifier

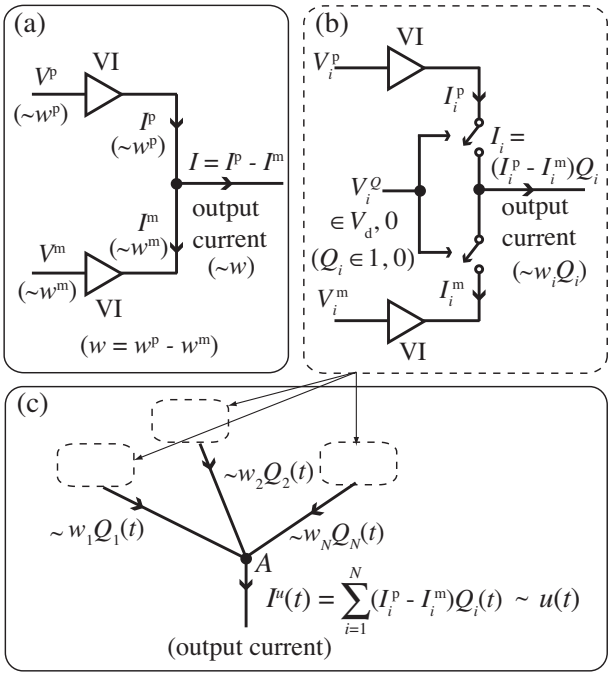


Figure 7.8: Schematic showing main idea for implementation of bipolar synapses and output cell.

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", 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, 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, in order to learn the input sequences correctly, it is necessary to minimize the error between the input and the output sequences by updating the weights according to Eqs. (7.4) and (7.5). So the next step is to implement Eq. (7.4). Since δw_i takes positive and negative values, the same 'differential' strategy employed for the synapse circuit was used. Assume that $I(t)$ and $u(t)$ are represented by currents $I_{\rm in}(t)$ and $I^u(t)$, respectively, and the currents are integrated by capacitors. Then it can rewritten Eq. (7.4) as

$$\delta w_i \sim V_i^I - V_i^u, \tag{7.16}$$

$$V_i^I \equiv \frac{1}{C}\int_{jT}^{(j+1)T} I_{\rm in}(t)Q_i(t)dt, \tag{7.17}$$

$$V_i^u \equiv \frac{1}{C}\int_{jT}^{(j+1)T} I^u(t)Q_i(t)dt, \tag{7.18}$$

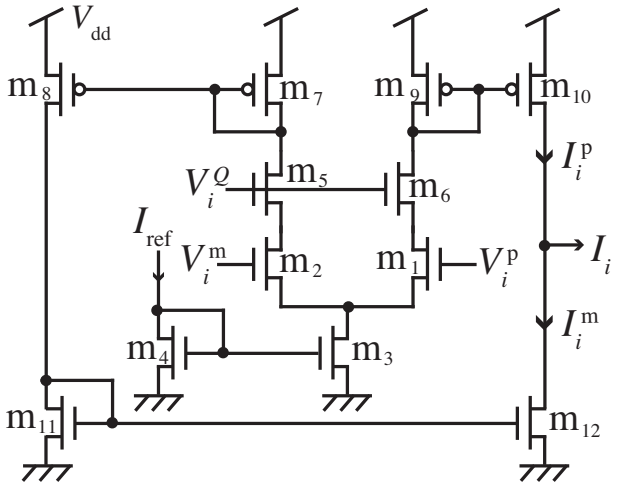


Figure 7.9: Synapse circuit calculating weighted sum (I_i) of output of oscillator (V_i^Q) and stored weight voltages ($V_i^{\rm p}$ and $V_i^{\rm m}$).

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. (7.17) and (7.18), which here called integrator circuit, is shown in Fig. 7.10. The circuit consists of two current mirrors (m_1-m_7 and m_2-m_8), two pass transistors (m_3 and m_4), two capacitors (Cs), and two transistors for reset operations (m_5 and m_6). When V_i^Q is 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. 7.10 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. (7.16). Assume that the differential voltage is nonlinearly converted to current I_i^δ by transconductance amplifier. The characteristic is illustrated in Fig. 7.11(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. 7.11(a) right (or left).

A MOS circuit that produces $I_i^{\delta{\rm p}}$ and $I_i^{\delta{\rm m}}$, which here called piecewise linear (PWL) circuit, is shown in Fig. 7.11(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

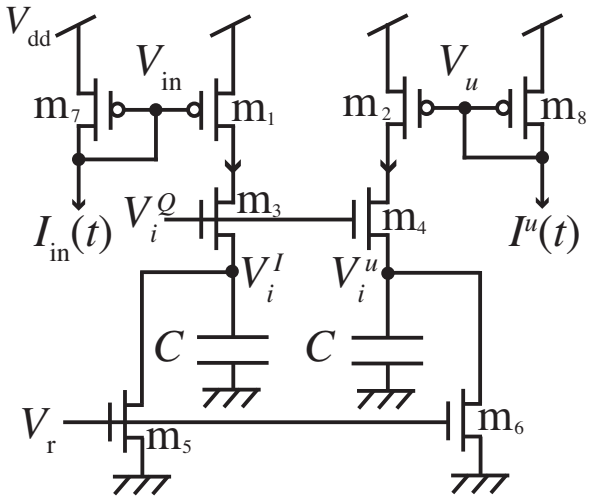


Figure 7.10: Integrator circuit.

operating in the subthreshold region, currents I^1 and I^2 are given by:

$$I^1 = I_{\text{ref}} \frac{\exp(\kappa V_i^I)}{\exp(\kappa V_i^I) + \exp(\kappa V_i^u)}, \tag{7.19}$$

$$I^2 = I_{\text{ref}} \frac{\exp(\kappa V_i^u)}{\exp(\kappa V_i^I) + \exp(\kappa V_i^u)}. \tag{7.20}$$

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 $I_i^{\delta\text{p}}$. This operation corresponds to Fig. 7.11(a) right. Simultaneously, currents I^1 and I^2 are copied to m_{10} and 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 $I_i^{\delta\text{m}}$, which corresponds to characteristics in Fig. 7.11(a) left.

As explained in section 2, at the end of each oscillatory cycle (T), the weights have to be updated according to Eq. (7.5). We have already separated δw_i into positive and negative parts, as shown in Fig. 7.11(a), and obtained two positive currents $I_i^{\delta\text{p}}$ and $I_i^{\delta\text{m}}$. Assume that the bipolar weights are separately stored in capacitors, and are updated with the amount of $I_i^{\delta\text{p}}$ and $I_i^{\delta\text{m}}$. Then Eq. (7.5) can be rewritten as

$$V_i^{\text{p}}(t + \Delta t) = V_i^{\text{p}}(t) + \frac{\Delta t}{C} I_i^{\delta\text{p}} L, \tag{7.21}$$

$$V_i^{\text{m}}(t + \Delta t) = V_i^{\text{m}}(t) + \frac{\Delta t}{C} I_i^{\delta\text{m}} L, \tag{7.22}$$

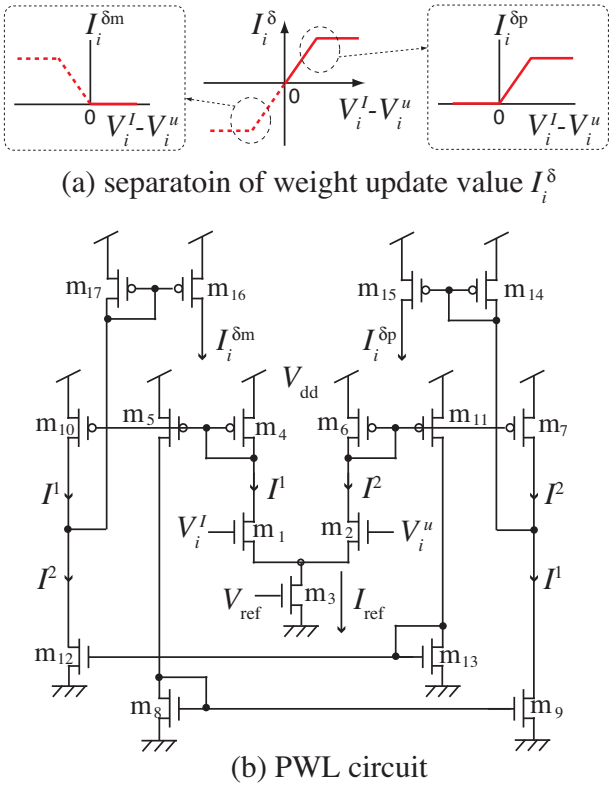


Figure 7.11: MOS circuits for calculating weight update values; (a) conceptual characteristics and (b) piecewise linear (PWL) circuit.

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

$$C\frac{dV_i^{\rm p}}{dt} = I_i^{\delta {\rm p}} L, \tag{7.23}$$

$$C\frac{dV_i^{\rm m}}{dt} = I_i^{\delta {\rm m}} L. \tag{7.24}$$

Figure 7.12(a) illustrates a MOS circuit that calculates Eqs. (7.23) and (7.24). During the update cycle ($V_{\rm L}$ is logical "1"), $I_i^{\delta {\rm p}}$ and $I_i^{\delta {\rm m}}$ are separately integrated by capacitors C_1 and C_2, respectively, via pass transistors m_1 and m_2. Remember that the integrated values $V_i^{\rm p}$ and $V_i^{\rm m}$ represent the weight w_i ($\sim V_i^{\rm p} - V_i^{\rm m}$), and they are fed back to the i-th synapse circuit shown in Fig. 7.9.

Figure 7.12(b) summarizes the circuit's control voltages per single learning cycle. Before starting each learning cycle, $V_{\rm r}$ is set to logical "1" to reset the

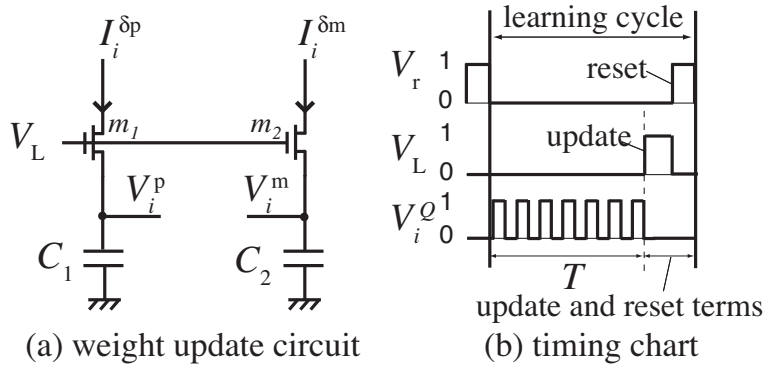


Figure 7.12: a) Weights update circuit, b) learning structure

weight update values δw_i ($V_i^I = V_i^u = 0$). At the beginning of each learning cycle, $V_{\rm d}$ of oscillator circuit shown in Fig. 7.7 is set to $V_{\rm dd}$ and V_i^Q starts to exhibit square-wave oscillations. At the end of oscillatory cycle, $V_{\rm d}$ is set to 0 (thus the oscillation stops) and in turn the weight update starts ($V_{\rm L}$ = "1"). When the update is finished, $V_{\rm r}$ is set to "1" again. This process is repeated until the difference between the input and the output sequences becomes small enough.

7.3 Simulation results

SPICE simulations were conducted for each circuit component in section 3. In the simulations, we used TSMC 0.35-μm CMOS parameters. Figure 7.13 shows the results of single oscillator circuit, integrator circuit and PWL circuit. In the oscillator circuit, all the dimensions (W/L) of transistors were set to 2 μm / 0.24 μm, and $V_{\rm ref}$ was set to 450 mV. The supply voltage $V_{\rm d}$ 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 learning cycle (at $V_{\rm d} = 0 \rightarrow 2.5$ V; *i.e.*, $t = 0.4$ μs and 0.8 μs) were the same, as shown in Fig. 7.13(a).

Simulation results of the integrator circuit are shown in Fig. 7.13(b). All the dimensions of transistors in the circuit were set to 0.36 μm / 0.24 μm. Input currents $I_{\rm in}$ and I^u were set to 1 μA and 2 μA, respectively. Capacitance C was set to 1 pF and the supply voltage $V_{\rm dd}$ was set to 2.5 V. Figure 7.13(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 ($V_{\rm r}$) was set to logical "1" ($t = 0 \sim 0.25$ μs). The integration started when $V_{\rm r}$ was set to "0" and V_i^Q was "1", which resulted in the increase of 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 $V_{\rm r}$ was set to "1", the integrated voltages were reset to zero ($t = 0.75 \sim 1$ μs).

Figure 7.13(c) shows simulation results for a single PWL circuit. 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 rest transistors. The supply voltage ($V_{\rm dd}$), V_i^u and $V_{\rm ref}$ were set to 2.5 V, 1.25 V and 1 V, respectively. As shown in Fig. 7.13(c) we could obtain similar characteristics as Figs. 7.11(a) left and right; *i.e.*, when $V_i^I > V_i^u$, $I_i^{\delta {\rm p}}$ was monotonically increased as V_i^I increased, whereas $I_i^{\delta {\rm p}}$ was always zero when $V_i^I < V_i^u$. On the other hand, when $V_i^I > V_i^u$, $I_i^{\delta {\rm m}}$ was always zero, while $I_i^{\delta {\rm m}}$ was monotonically decreased as V_i^I increased when $V_i^I < V_i^u$.

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

$$f_i \approx 0.3i + 1.1({\rm MHz}), \tag{7.25}$$

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 where 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 μA in amplitude, and λ/T was set to 4. Capacitances C_1 and C_2 in Fig. 7.12 were set to 1 pF and the supply voltage $V_{\rm dd}$ was set to 2.5 V.

Figure 7.14(a) shows a timing chart for the single learning cycle. Time evolution of i-th integrator outputs (V_i^I and V_i^u) and that of the weight voltages ($V_i^{\rm p}$ and $V_i^{\rm m}$) are shown in Figs. 7.14(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 cycles; 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 a stable value.

Time courses of temporal input voltage $V_{\rm in}$ ($\sim I(t)$; see Fig. 7.10) and learned output voltage V_u ($\sim u(t)$) are shown in Figs. 7.15 and 7.16. We could observe that $V_{\rm in}$ and V_u were different at the beginning (Fig. 7.15), but became similar after about 29 learning cycles (Fig. 7.16).

It is important to note that circuits in the model operate in sub-threshold region. In order to ensure the sub-threshold operation of circuits, fundamental frequencies of oscillators in the MHz range were used, (about 1 MHz to 10 MHz for the upper bound frequency). However, it is possible to learn sequences with lower frequency (kHz range) by changing the source current value (I_{ref} = 100 pA to 10 nA; Fig. 7.7) of the oscillator circuit.

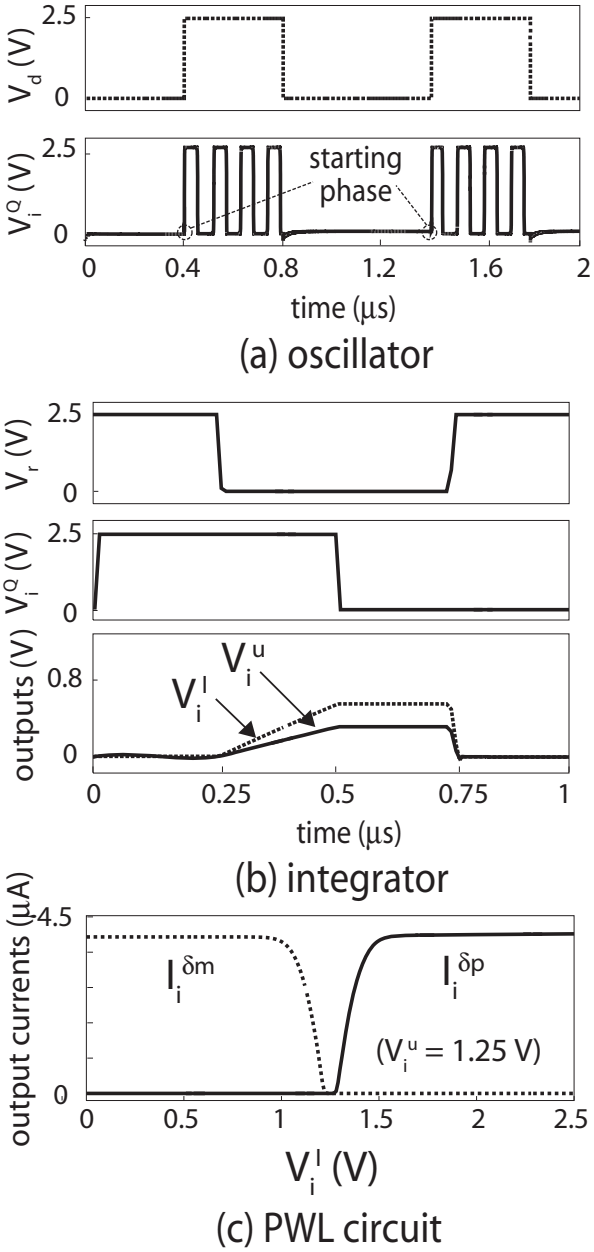


Figure 7.13: Simulation results of circuit components; (a) oscillator, (b) integrator and (c) piecewise linear PWL circuits.

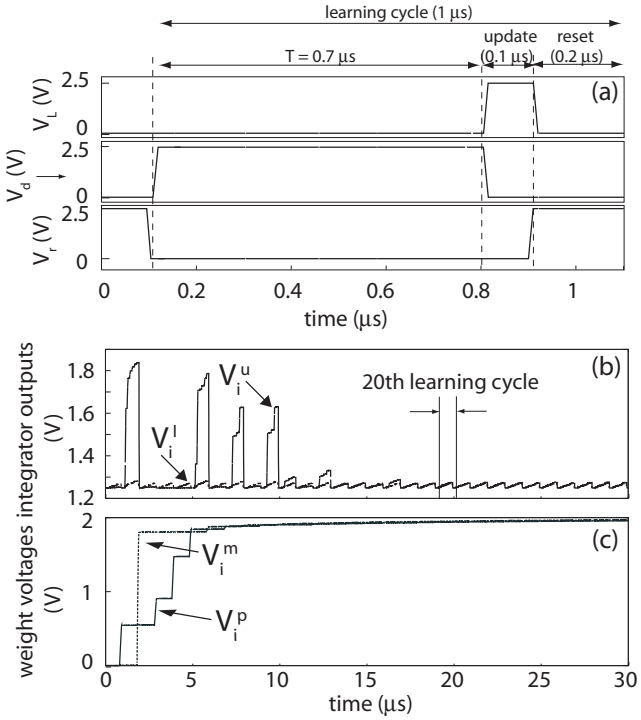


Figure 7.14: Simulation results of circuit network with $N = 20$; (a) timing chart, (b) time evolution of i-th integrator outputs and (c) evolution of weight voltages.

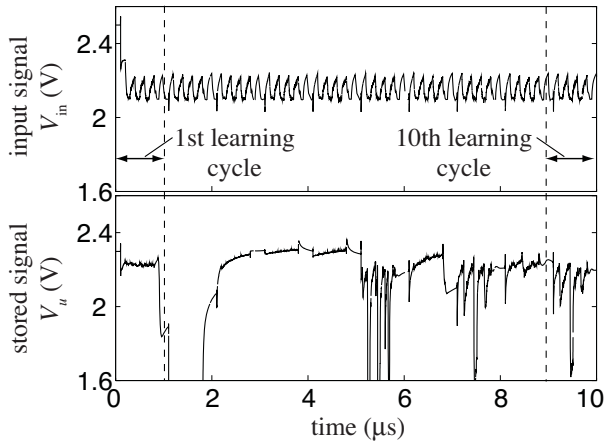


Figure 7.15: Evolution of temporal input sequence V_{in} and learned output sequence V_u (first to 10th learning cycles).

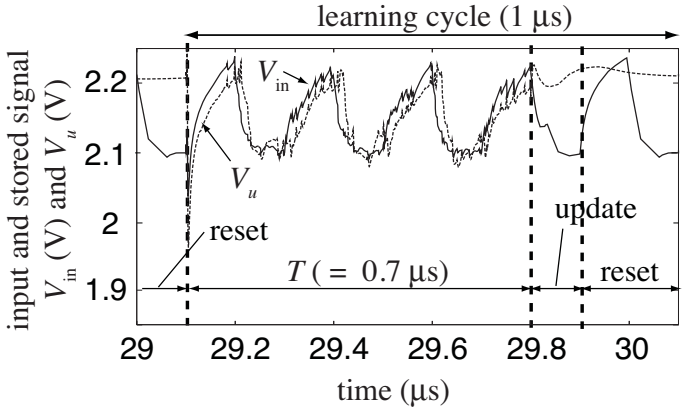


Figure 7.16: Temporal input sequence V_{in} and learned output sequence V_u after 29th learning cycle.

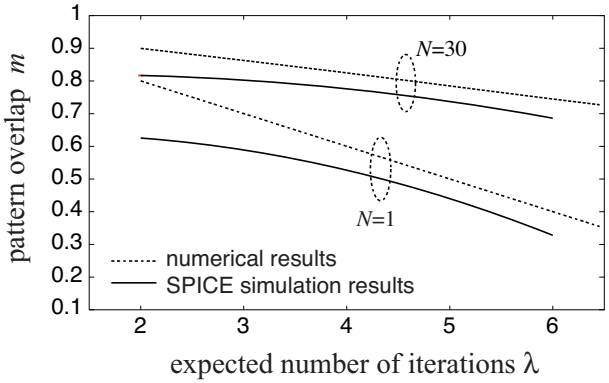


Figure 7.17: Numerical and SPICE results showing pattern overlaps between input and output sequences for different Ns and complexity of input sequence λ.

Finally, the pattern overlaps in Eq. (7.8) between the input and output sequences produced by the circuits was calculated 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 in the simulations of Figs. 7.14 to 7.16. The calculations were carried out for 1 and 30 neuron networks. The fundamental frequencies were set by Eq. (7.25), thus were distributed between 1.4 MHz and 10.1 MHz for N=30. Figure 7.17 shows the averaged pattern overlap between 10 different sets of the input sequences and their outputs. For comparison reasons, numerical results of the network model in section 2 with the same number of neurons were superimposed 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. This result shows that the circuit network of $N = 30$ can retrieve input sequence of $\lambda/T = 6/(0.7\ \mu\text{s}) \approx 8.6 \times 10^6\ (\text{s}^{-1})$ with the 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.

7.4 Summary

A neural circuit for temporal coding was designed. The model consists of N oscillatory units connected to an output cell through synaptic connections. To facilitate the implementation of the model, current-mode circuits where the input, output and the weight values were represented by currents were designed. The operation of each component of the network was demonstrated through circuit simulations. Moreover, the operations of the entire circuit with 20 neu-

rons was confirmed. The storage ability was also evaluated. When $N = 30$, the circuit can learn and recall binary temporal sequences with 6 iterations in the learning cycle with the accuracy of 72% under physically plausible device configurations.

Chapter 8

Conclusion

Biological organisms perform complex operations continuously and effortlessly. These operations allow them to quickly determine the motor actions to take in response to combinations of external stimuli and internal states. This thesis focused on the studied and hardware implementation of such kind of biological operations. In other words, it aims to implementing systems that mimic the sensory information processing performed by biological organisms. This is a small contribution, to reach the goal researchers in this area have in common, *the building of an artificial brain*. To accomplish this, in this research a series of circuits were proposed.

For implementation of circuits at the first stage of perception, a temperature receptor circuits was proposed. The receptor consists of a sub-threshold CMOS circuit that changes its dynamic behavior, i.e., oscillatory or stationary behaviors, at a given threshold temperature; where the threshold temperature, can be set to a desired value by adjusting the external bias voltage (θ)

In addition, as it is well know noise is present at every level of the nervous system, from the perception of sensory signals to the generation of motor responses. It is though that neurons and neural networks may employ different strategies that can exploit the properties of noise to improve the efficiency of neural operations. Therefore, a neural network that use noise and coupling (array-enhanced stochastic resonance) to improve signal detection was proposed. In the network, each neuron is electrically coupled to its four neighbors to form a 2D grid network. All neurons accept a common sub-threshold input, and no external noise source is required as each neuron acts as a noise source to other neurons.

Transmission of signals (stimuli) between is done by synapses. The computational potential of synapses has important implications for the diversity of signaling within neural circuits, suggesting that synapses have a more active

role in information processing. One special type of synapse (Depressing synapse) have been shown to contribute in a wide range of sensory tasks. Therefore, in chapter 5 a depressing synapses circuit was implemented and employed in a simple neural network model to demonstrate the effect of synaptic depression on synchronization (which is believe to have a important role in the coding of sensory information).

The remaining chapters shifted the focus to the cognitive processing area. With the introduction of two models. A neural network model for sensory segmentation. The model performs segmentation in temporal domain where the learning is governed by symmetric spike-timing dependent plasticity (STDP). This work concluded with the implementation of a model for learning and recalling the temporal stimuli. As, in neural systems to process information that changes over time a short-term memory is needed. The model consisted of neural oscillators which are coupled to a common output cell through positive or negative synaptic connections. The basic idea is to learn input sequences, by superposition of rectangular periodic activity (oscillators) with different frequencies.

All circuits operation were analyzed theoretically through mathematical models of its operation. Also, extensive numerical and circuit simulations were conducted. And the operation of the circuits and networks were demonstrated.

The combination of such kind of simple circuit will allow the design of hardware system that are capable of detecting, transforming, transferring, processing and interpreting sensory stimuli. The possibility to built complex neuromorphic systems which sense and interact with the environment will hopefully contribute to advancements in both, basic research and commercial applications. This technology is likely to become instrumental for research on computational neuroscience, and for practical applications that involve **sensory signal processing**, in uncontrolled environments.

Bibliography

[1] W. S. McCulloch, and W. H. Pitts, "A logical calculus of the ideas imminent in nervous activity," *Bull. Math. Biophy.*, vol. 5 pp. 115-133, 1943.

[2] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities", *Proc. Nat. Acad. Sci. U.S.A.*, vol. 79 no. 8 pp. 2554-2558, April 1982.

[3] J. Alspector and R. B. Allen, "A neuromorphic VLSI learning-system", *Adv. res. VLSI: Proc Stanford conf.*, pp. 313-349, 1987.

[4] M. A. Sivilotti, M. A. Mahowald, and C. A. Mead, "Real-time visual computation using analog CMOS processing arrays", *Adv. res. VLSI: Proc Stanford conf.*, pp. 295-311, 1987.

[5] C. A. Mead, "Neuromorphic electronic systems", *Proc. IEEE.*, vol. 78 no. 10 pp. 1629-1636, 1990.

[6] R. F. Lyon and C. A. Mead, "An analog electronic cochlea", *IEEE Trans. on Acoustic, Speech and Signal Proc.*, vol. 36 no. 7 pp. 1119-1134, 1988.

[7] J. Tanner and C. Mead, "Optical motion sensor", *Owen Kung and Nash, editors, VLSI Signal Proc.*, pp. 59-76, 1987.

[8] A. R. Moller, "Sensory Systems: Anatomy and Physiology", Academic press, Elsevier Science USA. 2003.

[9] Peter Dayan and L. F. Abbott, "Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems", The MIT press. 2001.

[10] Hebb, D.O. "The organization of behavior", *New York: Wiley*, 1949.

[11] R. Jacob Baker, Harry W. Li, and David E. Boyce, "CMOS circuit design, layout and simulations", IEEE press on Microelectronic Systems. 1998.

[12] John Hertz, Anders Krogh and Richard G. Palmer, "Introduction to the theory of Neural Computation", The MIT press. 2001.

[13] Edgar Sanchez-Sinencio and Clifford Lau, "Artificial Neural Networks: Paradigms, applications and hardware implementations", IEEE press. 1992.

[14] D. S. Fletcher and L. J. Ram, "High temperature induces reversible silence in Aplysia R15 bursting pacemaker neuron," *Comp. Biochem. Physiol.*, Vol. 98A, pp. 399-405, 1990.

[15] R. Douglas, M. Mahowald, and C. Mead, "Neuromorphic analogue VLSI," *Ann. Rev. Neurosci.*, 18, 1995, 255-281.

[16] B. L. Barranco, E. S. Sinencio, A. R. Vazquez, and J. L. Huertas, "A CMOS implementation of FitzHugh-Nagumo neuron model," *IEEE J. Solid-State Circuits.*, 26, 1991, 956-965.

[17] S. Ryckebusch, J. M. Bower, and C. Mead, "Modelling small oscillating biological networks in analog VLSI," *Advances in Neural Information Processing Systems 1.*, (D. S. Touretzky, Ed., Los Altos, CA: Morgan Kaufmann, 1989, 384-393).

[18] A. F. Murray, A. Hamilton, and L. Tarassenko, "Programmable analog pulse-firing neural networks," *Advances in Neural Information Processing Systems 1.*, (D. S. Touretzky, Ed., Los Altos, CA: Morgan Kaufmann, 1989, 671-677).

[19] J. L. Meador and C. S. Cole, "A low-power CMOS circuit which emulates temporal electrical properties of neurons," *Advances in Neural Information Processing Systems 1.*, (D. S. Touretzky, Ed., Los Altos, CA: Morgan Kaufmann, 1989, 678-685).

[20] T. Asai, Y. Kanazawa, and Y. Amemiya, "A subthreshold MOS neuron circuit based on the Volterra system," *IEEE Trans. Neural Networks*, 14(5), 2003, 1308-1312.

[21] 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.

[22] T. Hirose, T. Matsuoka, K. Taniguchi, T. Asai, and Y. Amemiya, "Ultralow-power current reference circuit with low-temperature dependence," *IEICE Transactions on Electronics*, Vol. E88-C, No. 6, pp. 1142-1147 (2005).

[23] S. Liu, J. Kramer, G. Indiveri, T. Delbruck and R. Douglas, "Analog VLSI: circuit and principles," The MIT press, Massachusetts Institute of Technology Cambridge, Massachusetts. London, England.

[24] P. G. Lillywhite, and S. B. Laughlin, "Transducer noise in a photoreceptor", *Nature*, Vol. 277, pp. 569-572 (1979).

[25] C. U. M. Smith., *Biology of Sensory Systems*, John Wiley & Sons Ltd, West Sussex, England, (2000).

[26] P. Hanggi, "Stochastic Resonance in Biology– How noise can enhance detection of weak signals and help improve biological information processing," *CHEMPHYSCHEM* vol. 3, pp. 285–290, (2002).

[27] W. H. Calvin, and C. F. Stevens, "Synaptic noise and other sources of randomness in motoneurons interspike intervals", *J. Neurophysiology*, Vol. 31, pp. 574-587 (1968).

[28] J. A. White, J. T. Rubinstein, and A. R. Kay, "Channel noise in neurons", *Trends in Neuroscience*, Vol. 23, pp. 131-137 (2000).

[29] A. A. Faisal, J. A. White, and S. B. Laughlin, "Ion-channel noise places limits on the miniaturization of the brain's wiring", *Curr. Biology*, Vol. 15, pp. 1143-1149 (2005).

[30] X. Lou, V. Scheuss and R. Schneggenburger, "Allosteric modulation of a pre-synaptic Ca^{2+} sensor for vesicle fusion", *Nature*, Vol. 435, pp. 497-501 (2005).

[31] N. A. Hessler, A. M. Shirke, and R. Malinow., "The probability of transmitter release at a mammalian central synapse," *Nature (London)*, Vol. 366, pp. 569–572, (1993).

[32] A. Aldo, P. J. Luc, and M. Daniel, "Noise in the nervous system," *Nature Reviews*, Vol. 9, pp. 292–303, (2008).

[33] M. N. Shadlen, and W. T. Newsome., "Noise, neural codes and cortical organization," *Curr. Opin. in Neurobiol.*, vol. 4, pp. 569–579, 1994.

[34] L. Gammaitoni, P. Hanggi, P. Jung, and F. Marchesoni, "Stochastic resonance," *Reviews of Modern Physics*, vol. 70, no. 1, pp. 223–287, (1998).

[35] A. Longtin, A. Bulsara, and F. Moss, "Time-interval sequences in bistable systems and the noise-induced transmission of information by sensory neurons", *Physics Review Letter*, Vol. 67, pp. 656-659 (1991)

[36] J. K. Douglass, L. Wilkens, E. Pantazelou, and F. Moss, "Noise enhancement of information transfer in crayfish mechanoreceptors by stochastic resonance", *Nature*, Vol. 365, pp. 337-340 (1993).

[37] H. A. Braun, H. Wissing, K. Schafer, and M. C. Hirsch, "Oscillation and noise determine signal transduction in shark multimodal sensory cells", *Nature*, Vol. 367, pp. 270-273 (1994).

[38] J. E. Levin, and J. P. Miller, "Broadband neural encoding in the cricket cereal sensory system enhanced by stochastic resonance", *Nature*, Vol. 380, pp. 165-168 (1996).

[39] P. Cordo, *et al*, "Noise in human muscle spindles", *Nature*, Vol. 383, pp. 769-770 (1996).

[40] J. F. Lindner, B. K. Meadows, W. L. Ditto, M. E. Inchiosa, and A. R. Bulsara., "Array Enhanced Stochastic Resonance and Sapatiotemporal Synchronization," *Phys. Rev. Lett.*, Vol. 75, pp. 3–6, (1995).

[41] S. M. Bezrukov, and I. Vodyanoy, "Noise-induced enhancement of signal transduction across voltage-dependent ion channels," *Nature (London).*, Vol. 378, pp. 362–364, (1995.

[42] J. J. Collins, C. C. Chow and T. T. Imhoff., "Aperiodic stochastic resonance in excitable systems," *Phys. Rev. E.*, Vol. 52 no. 4, pp. 3321–3324, (1995).

[43] T. Kanamura, T. Horita and Y. Okabe., "Theoretical analysis of array-enhanced stochastic resonance in the diffusively coupled FitzHugh-Nagumo equation," *Phys. Rev. E.*, Vol 64, pp. 031908, 2001).

[44] F. Liu, B. Hu and W. Wang., "Effects of correlated and independent noise on signal processing in neuronal systems," *Phys. Rev. E.*, Vol. 63, pp. 031907, (2001).

[45] M. E. Inchiosa ,and A. R. Bulsara, "Coupling anhances stochastic resonance in nonlinear dynamic elements driven by a sinusoid plus noise," *Physics Letters A.*, Vol. 200, pp. 283–288, April (1995).

[46] N. Schweighofer, K. Doya, H. Fukai, J. V. Chiron, T. Furukawa, and M. Kawato, "Chaos may enhance information transmission in the inferior olive," *PNAS*, Vol. 101, no. 13, pp. 4655–4660, March (2004).

[47] W. C. Stacy and D. M. Durand., "Noise and coupling affect signal detection and bursting in a simulated physiological neural network," *J. Neurophysiol.*, Vol. 88, pp. 2598–2611, (2002).

[48] T. Asai, Y. Kanazawa, T. Hirose, and Y. Amemiya., "Analog reaction-diffusion chip imitating the Belousov-Zhabotinsky reaction with Hardware Oregonator Model," *International Journal of Unconventional Computing*, Vol. 1, no. 2, pp. 123–147, (2005).

[49] G. Matthews and P. Fuchs, "The diverse roles of ribbon synapses in sensory neurotransmission," *Nature reviews*, Vol. 11, pp. 812–822, (2010).

[50] L. F. Abbott, and W. G. Regehr, "Synaptic computation," *Nature*, Vol. 431, pp. 796–803, (2004).

[51] F. S. Chance, S. B. Nelson, "Synaptic depression and the temporal response characteristic of V1 cells," *J. Neuroscience*, Vol. 18, pp. 4785–4799, (1998).

[52] S. Chung, X. Li, and S. B. Nelson, "Short-term depression at thalamocortical synapses contributes to rapid adaptation of cortical sensory responses in vivo," *Neuron*, Vol. 34, pp. 437–446, (2002).

[53] M. Carandini, D. J. Heeger, and W. A. Senn, "A synaptic explanation of suppression in visual cortex," *J. Neuroscience*, Vol. 22, pp. 10053–10065, (2002).

[54] V. F. Castellucci, *et al*, "Neuronal mechanisms of habituation and deshabituation of the gill-withdrawal reflex in *Aplysia*," *Science*, Vol. 167, pp. 1745–1748, (1970).

[55] D. L. Cook, *et al*, "Synaptic depression in the localization of sound," *Nature*, Vol. 421, pp. 66–70, (2003).

[56] H. Kuba, K. Koyano, and H. Ohmori, "Synaptic depression improves coincidence detection in the nucleus laminaris in brainstem slices of the chick embryo," *J. Neuroscience*, Vol. 15, pp. 984–990, (2002).

[57] M. A. Castro-Alamancos, "Role of thalamocortical sensory supression during arousal: focusing sensory inputs in neocortex," *J. Neuroscience*, Vol. 22, pp. 9651–9655, (2002).

[58] M. S. Goldman, P. Maldonado, and L. F. Abbott, "Redundancy reduction and sustained firing with stochastic depressing synapses," *J. Neuroscience*, Vol. 22, pp. 584–591, (2002).

[59] W. Senn, I. Segev, and M. Tsodyks, "Reading neuronal synchrony with depressive synapses," *Neural Computation*, Vol. 10, pp. 815–819, (1998).

[60] W. Senn, *et al*, "Dynamics of a random neural network with synaptic depression," *Neural Networks*, Vol. 9, pp. 575–588, (1996).

[61] M. Galarreta, and S. Hestrin, "Frequency-dependent synaptic depression and the balance of excitation and inhibition in neocortex," *Nature Neuroscience*, Vol. 1, pp. 587–594, (1998).

[62] T. Fukai and S. Kanemura, "Noise-tolerant stimulus discrimination by synchronization with depressing synapses," *Biol. Cybern.* Vol. 85, No. 2, pp.107–116, (2001).

[63] T. Asai, Y. Kanazawa and Y. Amemiya, "A subthreshold MOS neuron circuit based on the Volterra system," *IEEE Trans. on Neural Network,* Vol. 14, No. 5, pp. 1308–1312, (2003).

[64] Y. Kanazawa, T. Asai, M. Ikebe and Y. Amemiya "A novel CMOS circuit for depressing synapse and its application to contrast-invariant pattern classification and synchrony detection," *Int. J. Robotics and Automation,* Vol. 19, No. 4, pp. 206–212, (2004).

[65] W. Reichardt, "Principles of Sensory Communication," *Wiley*, New York, (1961).

[66] D.C. Carroll, N.J. Bidwell, S.B. Laughlin, and E.J. Warrant, "Insect motion detectors matched to visual ecology," *Nature*, Vol. 382, pp. 63–66, (1996).

[67] R. Kern, M. Egelhaaf and M.V. Srinivasan, "Edge detection by landing honeybees: behavioural analysis and model simulations of the underlying mechanism," *Vision Res.*, Vol. 37, pp. 2103–2117, (1997).

[68] A.G. Andreou, K.A. Boahen, P.O. Pouliquen, A. Pavasović, R.E. Jenkins and K. Strohbehn, "Current-mode subthreshold MOS circuits for analog VLSI neural systems," *IEEE Trans. Neural Networks*, Vol. 2, No. 2, pp. 205–213, (1991).

[69] E.A. Vittoz, "Micropower techniques, in Design of MOS VLSI Circuits for Telecommunications," *Y. Tsividis and P. Antognetti, Eds.*, Prentice-Hall, Englewood Cliffs, NJ, pp. 104–144, (1985).

[70] S. K. Han, W. S. Kim, and H. Kook, "Temporal segmentation of the stochastic oscillator neural network," *Physical Review*, E 58, 2325-2334, (1998).

[71] Ch. von der Malsburg and J. Buhmann, "Sensory segmentation with coupled neural oscillators," *Biological Cybernetics*, Vol. 67, 233-242, (1992).

[72] Ch. von der Malsburg and W. Schneider, "A neural cocktail-party processor," *Biological Cybernetics*, 54, 29-40, (1986).

[73] D.L.Wang and D. Terman, "Locally excitatory globally inhibitory oscillator networks", *IEEE Trans. on Neural Networks*, Vol. 6 no. 1, pp 283-286, (1995).

[74] H. Ando, T. Morie, M. Nagata, and A. Iwata, "An Image Region Extraction LSI Based on a Merged/Mixed-Signal Nonlinear Oscillator Network Circuit," *European Solid-State Circuits Conference (ESSCIRC 2002)*, pp. 703-706, Italy, Sept. (2002).

[75] T. Asai, M. Ohtani, and H. Yonezu, "Analog MOS circuits for motion detection based on correlation neural networks," *Japanese Journal of Applied Physics*, Vol. 38, no. 4B, pp. 2256-2261, (1999).

[76] M. J. M. Pelgrom, A. C. J. Duinmaijer, and A. P. G. Welbers, "Matching Properties of MOS Transistors." *J. Solid-State Circuits*, Vol. 24 no. 5, pp. 1433–1440 (1989)

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

[78] 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).

[79] 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).

[80] D.L. Wang, "Temporal pattern processing", *Handbook of Brain Theory and Neural Networks,* MIT Press, pp. 967-971, (1995).

[81] 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).

[82] 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).

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

[84] T. Fukai, "A model cortical circuit for the storage of temporal sequences," *Biol. Cybern.*, Vol. 72, No. 4, pp. 321-328, (1995).

[85] J.L. Walsh, *et. al.* "A close set of orthogonal functions," *Amer. J. Math.*, Vol. 45, No. 1. pp. 5-24, (1923).

List of Publications

Peer-reviewed Journal Articles

1. Tovar G.M., Asai T., Fujita D., and Amemiya Y., "Analog MOS circuits implementing a temporal coding neural model," Journal of Signal Processing, vol. 12, no. 6, pp. 423-432 (2008).

2. Tovar G.M., Asai T., Hirose T., and Amemiya Y., "Critical temperature sensor based on oscillatory neuron models," Journal of Signal Processing, vol. 12, no. 1 (2008), pp. 17-24 (2008).

3. Fukuda E.S., Tovar G.M., Asai T., Hirose T., and Amemiya Y., "Neuromorphic CMOS circuits implementing a novel neural segmentation model based on symmetric STDP learning," Journal of Signal Processing, vol. 11, no. 6, pp. 439-444 (2007).

4. Tovar G.M., Hirose T., Asai T., and Amemiya Y., "Neuromorphic MOS circuits exhibiting precisely-timed synchronization with silicon spiking neurons and depressing synapses," Journal of Signal Processing, vol. 10, no. 6, pp. 391-397 (2006).

Books & Chapters

1. Tovar G.M., "Analog Circuits Implementing a Critical Temperature Sensor based on Excitable Neuron Models," Advances in Analog Circuits, INTECH (in press, 2011)

2. Tovar G.M., Asai T., and Amemiya Y., "Noise-tolerant analog circuits for sensory segmentation based on symmetric STDP learning," Advances in Neuro-Information Processing, Koppen M., Kasabov N., and Coghill G, Eds., Lecture Notes in Computer Science, vol. 5507, pp. 851-858, Springer, Berlin / Heidelberg (2009).

3. Tovar G.M., "Fukuda S.E., Asai T., Hirose T., and Amemiya Y., "Analog CMOS circuits implementing neural segmentation model based on symmetric STDP learning," Neural Information Processing, Ishikawa M., Doya K., Miyamoto H., and Yamakawa T., Eds., Lecture Notes in Computer Science, vol. 4985, pp. 117-126, Springer, Berlin / Heidelberg (2008).

Invited Talks & Seminars

1. Tovar G.M., "Brain - inspired electrical circuits: life and research in Japan," Special Lecture in School of Electronics Engineering, Jose Antonio Paez University, Valencia, Venezuela (Sep. 21, 2009).

Int. Conference Proceedings

1. Tovar G., Asai T., and Amemiya Y., "Array-enhanced stochastic resonance in a network of noisy neuromorhic circuits," The 17th International Conference on Neural Information Processing, Sydney, Australia (Nov. 22-25, 2010).

2. Tovar G., Asai T., and Amemiya Y., "Coupling-enhanced stochastic resonance in noisy neuromorphic devices," The 14th International Conference on Cognitive and Neural Systems, Boston, USA (May. 19-22, 2010).

3. Tovar G.M., Asai T., and Amemiya Y., "Noise-tolerant analog circuits for sensory segmentation based on symmetric STDP learning," Proceedings of the 15th International Conference on Neural Information Processing of the Asia-Pacific Neural Network Assembly, pp. 199-200, Auckland, New Zealand (Nov. 25-28, 2008).

4. Tovar G.M., Fujita D., Asai T., Hirose T., and Amemiya Y., "Neuromorphic MOS circuits implementing a temporal coding neural model," 2008 RISP International Workshop on Nonlinear Circuits amd Signal Processing, Gold Coast, Australia (Mar. 6-8, 2008)

5. Tovar G.M., Fukuda E.S., Asai T., Hirose T.., and Amemiya Y., "Analog CMOS circuits implementing neural segmentation model based on symmetric STDP learning," The 14th International Conference on Neural Information Processing, Kitakyushu, Japan (Nov. 13-16, 2007).

6. Tovar G.M., Fukuda S.E., Asai T., Hirose T., and Amemiya Y., "Neuromorphic CMOS circuits implementing a novel neural segmentation model

based on symmetric STDP learning," 2007 International Joint Conference on Neural Networks, Florida, USA (Aug. 12-17, 2007).

7. Tovar G.M., Asai T., Hirose T., and Amemiya Y., "Critical temperature sensor based on spiking neuron models: experimental results with discrete MOS circuits," 2007 RISP International Workshop on Nonlinear Circuits amd Signal Processing, Shanghai, China (Mar. 3-6, 2007).

8. Tovar G.M., Asai T., and Amemiya Y., "Critical temperature sensor based on spiking neuron models," Proceedings of the 2006 International Symposium on Nonlinear Theory and its Applications (WIP session), pp. 84-88, Bologna, Italy (Sep. 11-14, 2006).

9. Tovar G.M., Asai T., and Amemiya Y., "Precisely-timed synchronization among spiking neural circuits on analog VLSIs," Proceedings of the 2006 RISP International Workshop on Nonlinear Circuits amd Signal Processing, pp. 62-65, Honolulu, USA (Mar. 3-5, 2006).

Awards

1. Tovar G.M., Asai T., Hirose T., and Amemiya Y., "Critical temperature sensor based on spiking neuron models: experimental results with discrete MOS circuits," The Research Institute of Signal Processing - NSCP'07 Student Paper Award, Mar. 2007.

Domestic conferences

1. Tovar G.M., Asai T., Amemiya Y. "Neuromorphic CMOS analog circuit exhibiting array-enhanced stochastic resonance behavior with population heterogeneity," Neuro 2010, Kobe. (Sep 2010)

2. Daichi F., Tovar G.M., Asai T., Hirose T., Amemiya Y. "Neural hardware for learning temporal signal." 18th Conference of the Japanese Neural Network Society, Ibaraki, (Sep. 2008).

   藤田 大地, Tovar Gessyca Maria, 浅井 哲也, 廣瀬 哲也, 雨宮 好仁, "時系列信号の学習を行うニューラルハードウェアの記憶容量評価," 日本神経回路学会 第 18 回全国大会, P1-26, (茨城), 2008 年 9 月.

3. Daichi F., Tovar G.M., Asai T., Hirose T., Amemiya Y. "Analog CMOS circuits for temporal coding." VDEC Designers Forum 2008, Tokyo, (June 2008).

藤田 大地, Tovar Gessyca Maria, 浅井 哲也, 雨宮 好仁, ”時系列コーディングを行う生体様 CMOS アナログ回路,” VDEC デザイナーフォーラム 2008, P-04, (東京), 2008 年 6 月.

4. Daichi F., Tovar G.M., Asai T., Hirose T., Amemiya Y. “Analog CMOS circuits implementing a temporal coding neural model.” The Isntitute of Electronics, Information and Communication Engineers, Kitakyushu, (March 2008).

   藤田 大地, Tovar Gessyca Maria, 浅井 哲也, 廣瀬 哲也, 雨宮 好仁, “時系列コーディングを行う神経モデルのアナログ CMOS 回路化,” 電子情報通信学会総合大会, (北九州), 2008 年 3 月.

5. Tovar G.M., Asai T., Hirose T., Amemiya Y.,“Neuromorphic LSI circuits for critical temperature detection, ”VDEC Designers Forum 2007, Sapporo, (Sep. 2007)

   Tovar Gessyca Maria, 浅井 哲也, 廣瀬 哲也, 雨宮 好仁, “Neuromorphic LSI circuits for critical temperature detection,” VDEC デザイナーフォーラム 2007（若手の会）, (札幌), 2007 年 9 月.

6. Asai T., Hirose T., Tovar G.M., Amemiya Y., “Integrated circuits implementing a Critical Temperature Sensor based on Excitatory Systems,” The 62th annual conference of The physical society of Japan, Chiba, (Sep. 2006)

   浅井 哲也, 廣瀬 哲也, Tovar Gessyca Maria, 雨宮 好仁, “興奮系を用いた臨界温度センサ集積回路,” 日本物理学会第 62 回年次大会, (千葉), 2006 年 9 月.