

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/325710976>

Design of Neuromorphic Circuit Based on Memristors

Thesis · June 2018

DOI: 10.13140/RG.2.2.21897.13924

CITATIONS

0

READS

883

1 author:



Weijie Jiang

KU Leuven

1 PUBLICATION 0 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



Design of Neuromorphic Circuit Based on Memristors [View project](#)



学士学位论文

THESIS OF BACHELOR



论文题目： Design of Neuromorphic Circuit Based on
Memristors

学生姓名： 蒋玮捷

学生学号： 5140219055

专业： 微电子专业

指导教师： 王国兴教授

学院(系)： 微纳电子学系

上海交通大学 学位论文原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：_____

日期：_____年_____月_____日

上海交通大学 学位论文版权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权上海交通大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

本学位论文属于

保 密口，在 _____ 年解密后适用本授权书。
不保密口。

(请在以上方框内打√)

学位论文作者签名: _____ 指导教师签名: _____

日 期: _____ 年 _____ 月 _____ 日

日 期: _____ 年 _____ 月 _____ 日

Design of Neuromorphic Circuit Based on Memristors

摘要

仿神经工程是上世纪八十年代由 Carver Mead 提出的一种概念。其内容涵盖了一类可以模仿生物特性的电路，或者从生物结构中受到启发的电路架构。在这门新学科中，模仿突触和神经元的特性以及神经系统的学习机制是非常重要的一环。一部分学者借助数字处理器和商用的储存技术，用复杂的数学模型来模拟搭建仿神经架构。这种方式的准确度非常高，但是开销也非常大，因此在目前的硬件科技下难以推广。一部分学者直接从特殊的电学器件入手，模拟神经系统的行为。这种方式显著降低了功耗开销，但是不够完善，且非常依赖非易失性的存储技术。还有一批学者摒弃了生物结构复杂的数学模型，将突触和神经元的行为描述成简单的逻辑操作。这部分学者在深度学习的领域取得了巨大成就，但由于他们所使用的模型与实际模型有较大差别，他们所研究的领域被逐渐划分到了数学学科。近年来，以阻变式存储器为代表的一类非易失性存储技术的提出给仿神经领域带来了新的活力。这类存储技术具有工艺简单，集成度高，读取方便的特点，并且与当今流行的 CMOS 工艺相兼容。它们被认为是最有利于分布式仿神经架构的存储技术。除此之外，有些学者提出以忆阻器为基本单元的 RRAM 与生物突触具有极其相似的动态特性。这些发现使得仿神经工程领域在近十年内迅速成为电路工程界的焦点。在本毕业设计中，我们将从一些传统的 CMOS 仿神经架构的基本单元入手，将忆阻器及其学习控制模块逐步纳入架构当中。我们不仅成功搭建了能模拟生物行为的 CMOS 突触和神经元电路，还设计出了一个能够控制突触传输比重的忆阻器模块，以及与现在流行的 STDP 学习机制相兼容的学习模块。利用这些电路子单元，我们设计并流片了一块集成了三个突触和一个神经元的芯片，其功能在后仿中得到了验证。除此之外，利用仿神经电路架构，我们成功实现了感知器学习算法，并对该毕业设计的进一步工作进行了探讨。

关键词：仿神经电路，忆阻器，感知器学习算法，STDP，突触，神经元

DESIGN OF NEUROMORPHIC CIRCUIT BASED ON MEMRISTORS

ABSTRACT

Neuromorphic engineering is a concept developed by Carver Mead in the late 80s. It describes electronic systems that mimic the behavior of neuro-biological architectures. One important task of neuromorphic engineering is to model biological synapse, neuron, and learning mechanism in electronic circuit. To do this, scholars over the world have made many attempts. Some of them try to use commercial processors and memories to build neuromorphic architecture based on mathematical models, but the cost of this choice is unaffordable to most of researchers. Some attempt to use specialized electronic device to mimic the biological structure, but they are impeded by the lack of localized non-volatile memory. Some abandon the dynamic and use highly abstract model of nervous system. They have made much progress especially in the field of deep learning. However, since models they use deviate too much from reality, they gradually become a mathematical discipline. The emergence of memristor sheds new light on neuromorphic engineering. Memristor is a novel nano-scale structure that shows retentive characteristic. It can be tuned to a desired value given certain stimuli. Shortly after the discovery of physical memristor, it has been shown that memristor updating mechanism is close to biological learning mechanisms. This provides a possible solution to the problem of state-of-the-art neuromorphic architecture. In this thesis, we will first review the basic concepts of memristor and neuromorphic engineering then build our memristor based neuromorphic architectures in Cadence. We build a new memristor based silicon synapse circuit that is able to show weighted transmission regardless of the resistance range of memristor. Furthermore, we use sub-threshold MOS transistor to mimic the dynamic of ion transmission. We successfully demonstrate that our neuromorphic circuit can imitate some biological behavior of nervous system, and perform perceptron learning task.

KEY WORDS: Neuromorphic engineering, memristor, perceptron learning, STDP, synapse, neuron

Contents

Chapter I Introduction	1
Chapter II Introduction to Memristor's Research Status	5
II.1 Memristor's Physical Model	5
II.2 Memristor's Mathematic Model	6
Chapter III Introduction to Neuromorphic Engineering	9
III.1 Artificial Neural Network	10
III.2 Neuromorphic Architecture	10
Chapter IV Memristor's Relationship to Neuromorphic Engineering	13
IV.1 Memristor as Synapse	13
Chapter V Memristor Based Neuromorphic Chip Design	17
V.1 Ideal Memristor Based Neuromorphic Circuit	17
V.2 Memristor based neuromorphic circuit on silicon	19
V.2.1 Synapse Circuit	20
V.2.2 Neuron Circuit	23
V.2.3 Function Generator Circuit	27
V.3 Three-Synapse One-Neuron Memristor Based Neuromorphic Chip	27
V.4 Conclusion	31
Chapter VI Implementing Perceptron Learning in Neuromorphic System	33
VI.1 Perceptron Learning	33
VI.2 Spike Encoding	34
VI.3 Perceptron Learning Neuromorphic Architecture	35
VI.3.1 Synapse Circuit	35
VI.3.2 Neuron Circuit	36
VI.3.3 Learning Circuit	38
VI.4 Perceptron Learning Test	39
VI.5 Conclusion	43

Chapter VII Reflections and Future Work	45
VII.1 About Neuromorphic Architecture	45
VII.2 About Perceptron Learning	45
VII.3 About Memristor Relationship to Neuromorphic Architecture	46
VII.4 Values of Neuromorphic Engineering	47
VII.5 Future Work	48
Summary	49
Bibliography	51
Acknowledgements	55

Chapter I Introduction

Memristor is a new group of basic circuit components. It was first hypothesized when observing the missing link between four fundamental circuit variables, namely, the voltage v , the current i , the flux ϕ , and the charge q . By calculating the possible elements that directly map one variable to another, scientists get five well-known relationships: Definition of current links current to charge; Definition of flux links flux to voltage; Resistor maps a voltage across it to a corresponding current; Inductor accumulates the voltage, derives the flux, and maps it to a current; Capacitor accumulates the current, derives the charge, and maps it to a voltage. These five relationships constitute the colorful electrical world and benefit our lives. However, in 1971, Leon. O. Chua pointed out an undefined relationship: the relationship between flux and charge. In [1], he postulated a circuit element that gives this definition and argued that this device acts like resistor at any time, whose resistance follows a function to the cumulative effect of current (charge) or voltage (flux). He thus named this device memristor, a combination of memory and resistor, because it acts like a resistor, and memorizes all the historical stimuli. To demonstrate the property of a memristor, which was not available through physical fabrication, he devised active circuits that mimiced the behavior of a memristor, of which the most important one is its pinched shape curve in the plane of voltage and current. This property is now recognized as a criterion to distinguish between an ordinary resistor and a memristor.

Although the notion of a memristor was widely accepted, it did not immediately become a hot research topic. In the subsequent days, scholars mainly discuss circuitry topologies that behave to the definition of a memristor. In these circuits, the relationships between the flux and charge are mostly nonlinear because of the use of diodes and other nonlinear circuit components. As a result, the first implementation of memristor was to build circuits with complex dynamics. These circuits are often referred to as chaotic circuits. The first chaotic circuit, known as Chua's Circuit, was proposed by Leon O. Chua in the late 90s^[2]. In this circuit, there are three linear energy storing units (two capacitors and one inductor), and a memristor with a piece-wise linear (PWL) $\phi - q$ relationship. This memristor is an active circuit element and it is able to observe a gallery of dynamic behaviors including strange attractors in this circuit^[3]. In [4], memristor-based oscillators with cubic nonlinearities is also considered. Since some memristors are also defined with a third variable, some researchers argue that complex behaviors can be expected in circuits with only two energy storing units^[5]. In all of the above works, ideal characteristics

of memristors are assumed, or modeled with elaborate electronic components.

The breakthrough appears in the year of 2008. In [6], a group of researchers in HP lab claims to find the physical existence of memristors. They argued that in nanoscale systems, the memory behavior is easy to observe due to particle movement, especially in metal oxide. Although the microscopic nature of conductance switching is not uniformly agreed by physicians, the switching behavior in metal oxide is believed to be owing to ionic diffusion and joule heating^[7]. The first mechanism involves migration of ions. The ionic migration can either strengthen the conductance of material when applied with positive bias, or depress the conductance of material when applied with negative bias. The second mechanism involves thermal effect. When applied with bias voltage, the formed electric field also generates joule heat. The heat, together with the applied voltage bias, will trigger the collisions of electron, which alters the structure of the device. However, the release of the heat is nonuniform, which depends on the charge being injected to the material. This leads to uncertainties in conductance switching. To extend this notion, researchers in HP lab built a two-terminal electrical device using two different materials. This device was made of one layer of insulated titanium dioxide and one layer of oxygen-poor titanium oxide. In this structure, the two materials are assumed to be divided by a barrier ' w '. When a bias voltage is imposed on the device, the electrical field will move this barrier thus altering the resistance of the device. Researchers observed the hysteresis loop when testing the physical device of this structure, which indicates the memory behavior of a device. Following this criterion, different memristors are found by scholars using different kinds of materials. For a complete summary of memristor fabrication methods, one can refer to [8]. Today, though memristor fabrication is still not mature in most foundries, it is foreseeable that memristor will play an important role in the market of integrated circuit (IC).

Because memristor is now available for research purpose, it has been a popular topic since 2008. In [9], Y. Hong propose a timing storage circuit based on memristor model. By recording information in a pair of matched memristor, his circuit can save and retrieve an analog timing information without quantization. In [10], Jahromi *et al.* implement memristor in traditional Miller operational amplifier design. In traditional operational amplifier design, compensation is a significant issue. Accurate adjustment of zeros and poles in a multi-stage operational amplifier is critical for improving gain, bandwidth, and frequency response. Instead of using additional circuit blocks for adjusting compensation, Jahromi *et al.* exploit the tunability of memristor to adjust compensation. Nevertheless, memristor receives most attentions in the realm of memory. Memristor memory can be viewed as resistive switching random access memory

(RRAM). The history of RRAM can be traced back to 1960s, when the observation that oxide can undergo abrupt resistive switching event was known^[11]. When used in the realm of RRAM, a memristor is regarded as a binary component that exhibits switching between two extreme states, namely, High-Resistance State (HRS) and Low-Resistance State (LRS). Scholars expect that RRAM will become a memory technology that can be integrated in traditional CMOS technology, which can be scaled up to large crossbar array. Based on this premise, a gallery of memristor implementation can be found. In [12], Lehtonen *et al.* build a circuit to carry out logic computation, which is now referred to as IMPLY operator in memristor-based logic. The circuit is simple and compact, which contains only two memristor and one resistor. Moreover, in [13], Haghiri *et al.* extend this notion and build common logic gates using IMPLY operators. They then construct a binary multiplier using only memristor logic. By combining this logic operational criteria with RRAM, we are able to expect significant achievement in computer technology.

Besides aforementioned implementations, memristor has more applications in the field of neuroscience. In late 1980s, the concept 'neuromorphic engineering' was introduced by Carved Mead, which describes very-large-scale integration (VLSI) systems containing nano electronic devices to mimic neuro-biological architectures in animals' nervous systems^[14]. In nervous systems, synapse is a crucial structure. In biological world, a synapse permits information to pass between different neurons. In other words, synapses are responsible for information flows in the joints of neurons. In traditional neuromorphic engineering, synapses are modeled using silicon integrated circuit. They transform a pre-synaptic voltage event to a post-synaptic corresponding current, which will be injected to a neuron. The efficiency of synaptic transmission is called 'synaptic weight'. Following the discovery of nano-scale memristive device, researchers immediately present the idea of modeling a synapse using a memristor. The analogy of memristors and chemical synapses is made on the basis that synaptic dynamics depend upon the discharge of neurotransmitters within a synaptic cleft^[15]. The first attempt of using memristor as synapse was proposed in [16]. Though the work done in [16] was only theoretically correct and neglected many practical issues, it demonstrated the idea that it is natural to derive tuning signals from neuron's behavior when connecting memristor as synapse in silicon spiking neural networks.

The rest of this thesis is organized as follows. In chapter two, a survey on the fabrication, modelling, and features of memristor will be covered. In chapter three, an introduce basic concepts about neuromorphic engineering will be introduced. In chapter four, memristor's

relationship with neuromorphic engineering is introduced. In chapter five, an introduce my memristor based neuromorphic chip design will be covered. In chapter six, I will use a neuromorphic architecture to perform perceptron learning. In chapter seven, out some future work and make some reflections will be pointed out.

Chapter II Introduction to Memristor's Research Status

II.1 Memristor's Physical Model

Since 2008, scholars over the world have conducted ample research in the field of memristor. The first physical model of memristor was proposed in the year of 2008. R. Stanley Williams of HP lab made a physical device with titanium dioxide sandwiched between two electrodes. Instead of using pure titanium dioxide, he doped part of it with oxygen vacancies. The pure titanium dioxide is highly resistive, but when doped with oxygen vacancies, it becomes conductive. Unlike semi-conductor devices, which uses magnetic field to store information as flux, or keeps charge the same as a capacitor, this memristor's memory feature is owing to chemical mechanism. When one side of titanium dioxide is depleted of oxygen, these vacancies will act like charge carriers and significantly lowering the resistance. The vacancies do not remain static, but will move according to outside stimuli. When the stimuli cease to exist, oxygen vacancies become idle. Thus the information is recorded. The idealized model is shown in Figure 2-1. Although the internal movement of particles is not a uniform process, Williams intentionally divides the device into two separate components. In this way, resistive switching can be explained by the movement of this imaginary boundary.

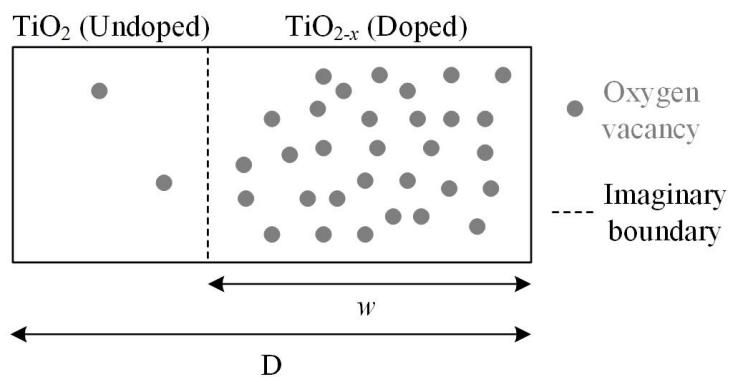


Figure 2-1 Idealized titanium dioxide memristor model

Besides titanium dioxide memristor, tantalum oxide based memristor is also frequently cited by scholars^[17]. It is made of a highly resistive tantalum pentoxide layer and a mildly resistive tantalum oxide layer. Like titanium dioxide memristor, these two layers are sandwiched between two palladium electrodes. When applying an electric field to the device, the oxygen particles

will move accordingly. If substantial amount of oxygen is removed from tantalum pentoxide, this particular region will become metallic. Interestingly, a conductive filament will appear^[18]. The device can be set to either a high-resistance state or a low-resistance state by controlling the property of the filament. Three environmental factors contribute to controlling of the filament: local electric field, concentration of oxygen vacancy, and temperature gradient. This model is shown in Figure 2-2. In fact, resistive switching in oxide material is common. In [8], Hadiyawarman *et al.* summarize recent works in metal oxide memristor. Most memristor structures resemble the previously discussed memristors.

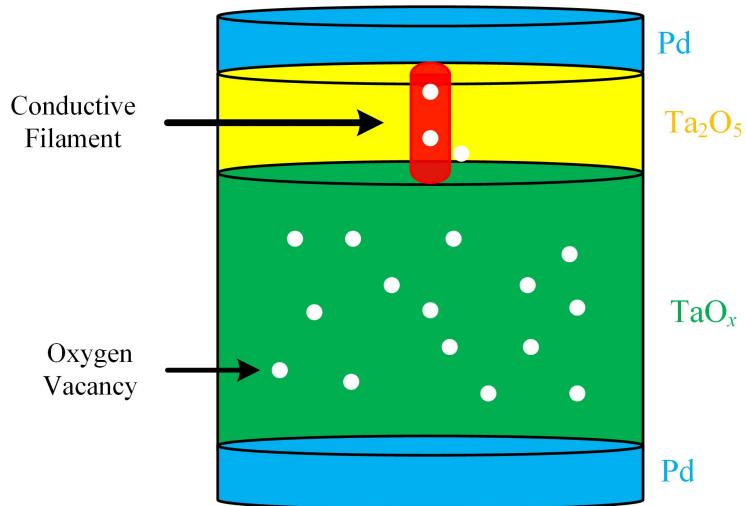


Figure 2-2 Tanalum oxide memristor model

II.2 Memristor's Mathematic Model

From circuit designer's perspective, mathematic model of memristor is more important. Not only does it allow us to simulate these models efficiently using circuit simulators, but also it gives us a direct sense from the level of circuit. In R. Stanley Williams's work, he initiated the modelling of a physical memristor. Figure 2-3 shows the model. The memristor is intentionally divided into a resistive region and a conductive region. The resistance of memristor, also known as memristance, is computed by summing up the resistance of these two regions. Assume that doping is even, and the resistance of each part is proportional to its length. The resistance equal to

$$R_{mem} = R_{ON} \frac{w}{D} + R_{OFF} \frac{D - w}{D} \quad (2-1)$$

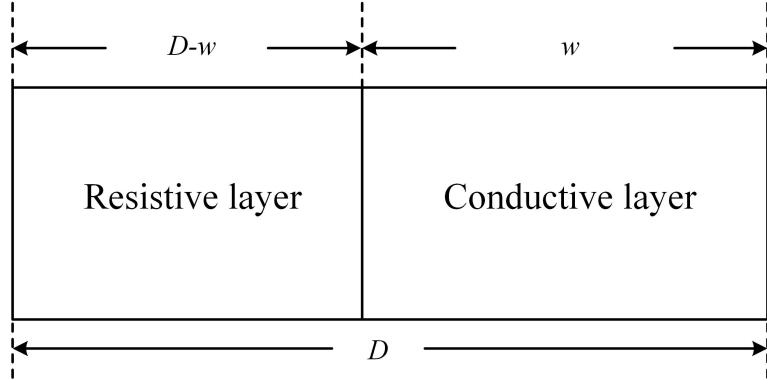


Figure 2–3 Mathematical model of a physical memristor

where D is total length of memristor, R_{ON} is the resistance unit for conductive region, and R_{OFF} is the resistance unit for resistive region. The movement of conductive region's length reflects the dynamic of a memristor. In [6], Williams proposed a linear ion drift model. In linear ion drift model, time derivative of w has a linear relationship to current, that is,

$$\frac{dw}{dt} = \frac{\mu_v R_{ON}}{D} i(t) \quad (2-2)$$

where μ_v is average ion mobility. Equation (2-1) and (2-2) determine all the dynamic of a memristor. However, aside from the inaccuracy of the linear ion drift model, it has very obvious defects. First, w is bounded by the length of memristor, which varies from 0 to D . Second, even if we intentionally restrict w to a certain range, $\frac{dw}{dt}$ violates this condition. In other words, additional constraints should be added to equation (2-2). Yogesh N. Joglekar and Stephen J. Wolf first observed these deficiencies and proposed a window function to equation (2-2), which was done to ensure that state variable w can always fall in the range $0 < w < 1$ ^[19]. The Joglekar window function is

$$F(w) = 1 - (2w - 1)^{2p} \quad (2-3)$$

where p is a positive integer. When applying Joglekar window function, motion of w saturates whenever it is travelling towards its boundary. Besides Joglekar window function, Zdenek Biolek *et al.* also proposed their window function^[20]. The Biolek window function only reduces velocity around the boundary whenever w approaches its boundary. The Biolek window function is described in equation (2-4) and (2-5).

$$F(w) = 1 - (w - stp(-I(t)))^{2p} \quad (2-4)$$

$$stp(I(t)) = \begin{cases} 1, & \text{if } I(t) > 0 \\ 0, & \text{if } I(t) < 0 \end{cases} \quad (2-5)$$

To conclude, window function cannot increase the accuracy of the model, but adjust singularities around the boundaries. To better model memristors, Dr. Mika Laiho *et al.* proposed a hyperbolic sinusoid model^[21]. The current and voltage relationship is not simply resistive, but a hyperbolic sinusoid modulated relationship. Equation (2-6) and (2-7) define the relationship.

$$I(t) = \begin{cases} a_1 w \sinh(b_1 V(t)), & V(t) \geq 0 \\ a_2 w \sinh(b_2 V(t)), & V(t) < 0 \end{cases} \quad (2-6)$$

$$\frac{dw}{dt} = \begin{cases} c_1 \sinh(d_1 V(t)), & V(t) \geq 0 \\ c_2 \sinh(d_2 V(t)), & V(t) < 0 \end{cases} \quad (2-7)$$

where $a_1, a_2, b_1, b_2, c_1, c_2, d_1, d_2$, are parameters to adjust $I - V$ response of this model. Also, this model was originally proposed without boundary conditions, so that the motion of state variable w cannot be guaranteed falling in the range $0 < w < 1$. To solve this problem, we can apply another window function to equation (2-7). The choice of window function is the same as linear drift model.

Another important memristor model is threshold adaptive memristor model (TEAM model). TEAM model is also based on a derivative of the state variable w , that can be accommodated to any memristive device. Unlike aforementioned models, the $I - V$ relationship in TEAM model remains undefined. The user can choose freely any appropriate $I - V$ relationship for TEAM model^[22]. Because experimental data show that memristive device current have highly nonlinear properties, TEAM model uses a piece-wise nonlinear function to define w function, as shown in equation (2-8).

$$\frac{dw}{dt} = \begin{cases} k_{off} \left(\frac{I(t)}{i_{off}} - 1 \right)^{\alpha_{off}} f_{off}(t), & 0 < i_{off} < I(t) \\ 0, & i_{on} < I(t) < i_{off} \\ k_{on} \left(\frac{I(t)}{i_{on}} - 1 \right)^{\alpha_{on}} f_{on}(t), & I(t) < i_{on} < 0 \end{cases} \quad (2-8)$$

where $k_{off}, k_{on}, i_{off}, i_{on}, \alpha_{off}, \alpha_{on}$ are model parameters. f_{off} and f_{on} are window functions mentioned before.

In my thesis, I choose to use hyperbolic sinusoidal model.

Chapter III Introduction to Neuromorphic Engineering

Neuromorphic engineering is a concept developed by Carver Mead in late 1980s. It describes a group of customized electronic circuit that either mimics organic behavior, or is inspired from biological architectures. The first neuromorphic chip is *Mazer*. It was an nMOS integrated circuit designed in 1977. *Mazer* was able to find the shortest path from one point to another on a sophisticated map. The whole algorithm was implemented directly from hardware level. Viewing from today's perspective, it displays many characteristics that we call neural systems nowadays. First, it is an example of parallel processing systems, with very fine-grain processing units. This is a typical characteristic of neural systems, which count on massive amount of simple processors to perform computation, in the same manner that biological systems rely on synapses and neurons. In *Mazer*, every processing node only consists of a few transistors and a capacitor, which is primitive compared with a general purpose computer. Second, nodes connectivity in *Mazer* is super complex, which is another significant characteristic of neural systems. Those individual processing nodes can be connected forward, recurrently, or to another *Mazer* chip. A different connective pattern refers to a different functionality. This means that *Mazer* is a highly customized computing architecture. In subsequent years, neuromorphic engineering focused on realizing biological organs in electronic circuits. These works enriched the content of neuromorphic engineering. However, it was the popularity of digital computer that tied this interdisciplinary subject with neuroscience. The computing power of digital computer sees a steady growth starting from the beginning of twenty-first century. Following the Moore's law, the number of MOS transistors in a dense digital chip doubles about every one year and a half, making the computing ability boom in the same speed. This enables scholars to do experiment using software in a manageable time length. Benefitted from this, neuromorphic engineers start to make braver move than implementing isolated biological function in a electronic system. They are divided into two groups. The first group of researchers handles with what is called artificial neural network today, while the other group sticks to the original definition of neuromorphic system, that is, to realize biological architecture directly in electronic systems.

III.1 Artificial Neural Network

In general, all neural networks have a similar structure. Figure 3-1 shows this structure. I_1 to I_n are n inputs to a neuron. These n inputs are accordingly adjusted by a structure called synapse. When they are weighted by their synapses, the output of those synapses are injected to a particular neuron, where it generates an output according to these inputs. For different generation of neural network, they have different dynamic range for inputs, different allowable values for the synaptic weight, and different strategy for deciding the output generation. Usually, the computing complexity at each neuron is low. This is to ensure that when the scale of neural network goes up, it is still manageable by computers. The artificial neural networks largely simplifies complex biological behaviors being observed by biologists. They only implement conceptual synapses and neurons, and connect them in an intricate way to find information. To find valuable information, one should first connect all inputs to a group of outputs in a certain way, then train the synaptic weight according to the labels of these inputs. Artificial neural network has been used to deal with a variety of tasks, such as computer vision, machine translation, image recognition, medical diagnosis and so on. More importantly, because artificial neural network is compatible with a general purpose computer, it is thought to be one of the solution to realize artificial intelligence.

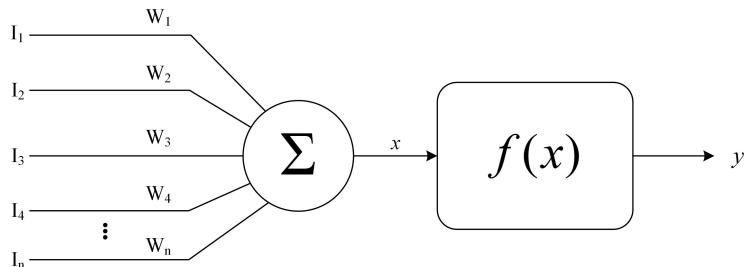


Figure 3-1 Structure of neural network

III.2 Neuromorphic Architecture

Although implementations of artificial neural network are very attractive, they are faced with difficulties. In traditional Von Neumann architecture, data must be retrieved from memory and processed by CPU. The bandwidth of this bus limits the maximum processing speed of such systems^[23]. Because of this, tasks like real-time image recognition are still difficult for our computers. The observation that our brain can perform these tasks in real time and consumes less power encourages us to establish a new computational system. In the past two decades,

neuromorphic computing has seen tremendous progresses. From software's level, machine learning is used to perform a lot of tasks like handwriting recognition^[24] and blood pressure prediction^[25]. From hardware's level, circuit blocks or electronic elements that mimic the biological cell's behavior have been explored for nearly thirty years^{[26]-[29]}. From system's level, massive parallel computational systems have been built for either biological research or practical applications. Examples include Blue Brain project, Neurogrid, and SpiNNaker project. These systems integrate large amount of cores and memories, and are able to perform formidable tasks. For example, SpiNNaker project, proceeded by University of Manchester, is able to simulate 1% of human brain^[30]. From this perspective, we can conclude that the foundation of neuromorphic is well established.

When referring to neuromorphic architecture, it usually means that dynamic behaviors of synapses and neurons are taken into account, and computation are not done instantaneously. Here we introduce two neuromorphic projects done by scholars. The first one is SpiNNaker project. SpiNNaker project does not refer to a chip, but an enormous nervous system simulated by multiple digital cores arranged in an elaborate way. The architecture scales from a single chip, in its smallest configuration, to a system of 65 536 chips, although the full-size SpiNNaker system will contain 57 600 chips with 1 036 800 processors, delivering peak processing power exceeding 228 Dhystone TIPS^[30]. With this, the system can simulate up to 10^{11} neurons, each with 10^4 synaptic connections. The system uses digital cores to calculate all the behaviors happening in nervous systems, thus it is quite different from neuromorphic chips that we would like to discuss in my thesis. However, the SpiNNaker project is a very brave attempt to simulate the brain of an animal, by using merely general purpose commercial digital chips, memories, and other products. It serves as a good platform for researchers from neuroscience field or computer science field, to do experiments when they are unable to perform them on living entities. More importantly, from algorithm level, the SpiNNaker project has proven that it is able to use these large-scale biological-inspired networks to solve actual problems.

The SpiNNaker project, though impressive, has very apparent shortcomings: it consumes too much power. When simulating only one percent of our brain, the SpiNNaker system costs the power of up to 90kW, while our brain burns only the energy of a meal to complete the same task (about 20W). The reason that causes this difference is profound and multi-fold. However, there are two critical factors contributing to this. First, unlike digital processing cores, our brain is customized to solve specific problems. The word, 'customized', has two meanings: that different parts of our brain are responsible for their very specific tasks; that the connection of these parts

cannot be configured to any random patterns. These two observations are deviated from our initial goals of building a general purpose computer, which benefits from the classical Von-Neumann architecture. Second, for digital processors, dynamics are simulated using software, while in our brain, dynamics happens naturally. In SpiNNaker project, this effort is one of the major source of energy consumer.

Besides SpiNNaker project, ROLLS is also a result of another neuromorphic project. ROLLS neuromorphic processor stands for Reconfigurable On-line Learning Spiking Neuromorphic Processor. It is a chip designed by Institute of Neuroinformatics, University of Zurich and ETH Zurich, Zurich, Switzerland^[6]. It models synapses and neurons directly in silicon integrated circuit. To be specific, the behavior of neurons can be modeled by charging and discharging a membrane capacitor, and the behavior of a synapse can be mimicked by a transistor working in sub-threshold region. ROLLS is able to perform tasks like perceptron learning and is low power compared with digital neuromorphic systems in the same scale. ROLLS is another impressive attempt to simulate nervous system, but it has some drawbacks. First, ROLLS stores synaptic information on a capacitor, which makes it volatile whenever it is powered off. To use it, a certain period of booting time or learning period has to be implemented. Second, since the information is stored in capacitor, it cannot drive any load, which makes it unfavorable for memristor synaptic connection.

Chapter IV Memristor's Relationship to Neuromorphic Engineering

In traditional neuromorphic engineering, synapses and neurons are modeled using silicon integrated circuit. Synapse is an important structure in neuron networks. They transform a pre-synaptic voltage event to a post-synaptic corresponding current, which will be injected to a neuron. The efficiency of synaptic transmission is called 'synaptic weight'. In computational models of neural network, scholars usually regard synapses operation as instantaneous computation, and disregard the dynamic behavior. On the other hand, researchers are more interested in the adjusting algorithms for synaptic weight. By observing the behavior of biological synapses, scholars proposed a learning scheme called Spike-Timing-Dependent-Plasticity (STDP). In STDP, the precise firing order between pre-synaptic neuron and post-synaptic neuron determines the sign of synaptic change, while the relative timing decides the magnitude of this change. Realizing STDP in an artificial neuron network is an important topic of today's neuromorphic engineering.

Following the discovery of nano-scale memristive device, researchers immediately present the idea of modeling a synapse using a memristor. The analogy of memristors and chemical synapses is made on the basis that synaptic dynamics depend upon the discharge of neurotransmitters within a synaptic cleft^[31]. The first attempt of using memristor as synapse was proposed in [16]. Though the work done in [16] was only theoretically correct and neglected many practical issues, it demonstrated the idea that it is natural to derive tuning signals from neuron's behavior when connecting memristor as synapse in silicon spiking neural networks.

IV.1 Memristor as Synapse

A synapse is a two-terminal structure that exists in the cleft between two neurons. It is essential to the function of nervous system. The reason is two-fold. First, a synapse allows information flows across neurons. Nervous system of an animal is not a connected whole, but consists many individual cells. As a result, information can flow as electric signal inside a neuron. However, it is unable to cross the boundary of neurons. When an electric signal arrives at the boundary of a pre-synaptic neuron, the membrane of this pre-synaptic neuron will release some chemical substances to the synapse, which is also known as neurotransmitters. These neurotransmitters can diffuse in the synapse, and after some time, reach the boundary of

some post-synaptic neurons. The post-synaptic neuron can filter out interference and capture these neurotransmitters, then excite the post synaptic neuron according to the amount of neurotransmitter in the synapse. Figure 4-1 shows the anatomy of a synapse. Second, a synapse is responsible for the decision making in nervous system. It has been shown that, a synaptic transmission is not merely a process of information movement, but possesses intricate dynamics and idiosyncrasies, which have significant effects on the response to the input stimuli^[32]. Even in the simplest manner, the synaptic transmission is modeled as an immediate weighted transmission. This means that computation in nervous systems happens in the synapse.

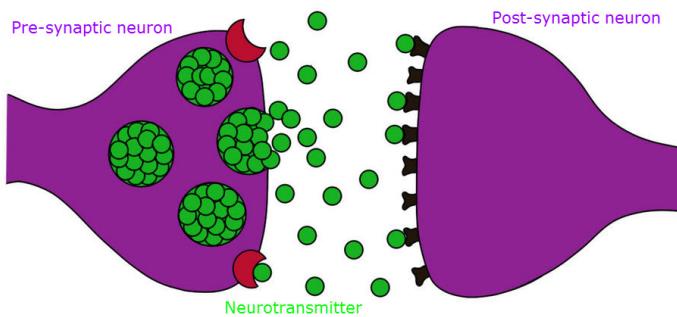


Figure 4-1 Anatomy of a biological synapse

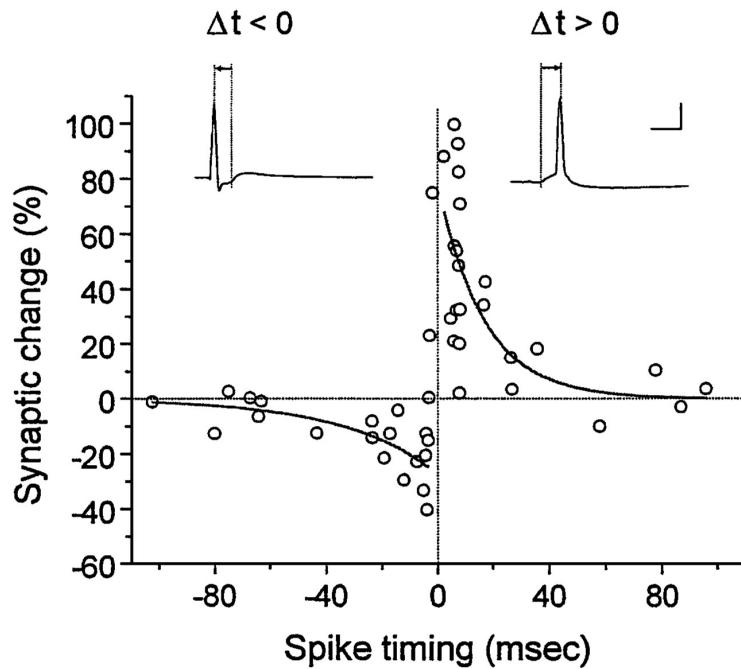


Figure 4-2 Experimental data for synaptic weight fluctuation^[34]

The question remains, that how does our body modify the behavior of a particular synapse.

Unfortunately, this is beyond state-of-the-art biological knowledge. In fact, if we manage to figure out this question some day, not only will we figure out how we humans distinguish different objects without previous knowledge, but also how we learn under the supervision of a third-party power. However, through experiments, scientists find some clues regarding synapse. It has been shown that, in some circumstances, the synapse becomes inactive to the input stimuli, and remains idle for a long time under strong environmental condition. This activity-dependent reduction is called Long-term depression (LTD). On the other hand, in some circumstances, the synapse becomes easy to stimulate and produce output event to the post-synaptic neuron with comparable weak input stimuli. This phenomenon is called Long-term potentiation (LTP). Both LTD and LTP are witnessed to occur in many regions of central nervous system^[33] (Long-term depression multiple forms). Based on these two observations, Bi and Boo made some hypothesis of the synaptic behavior^[34]. They observed that the relative timing between the post-synaptic spiking and the pre-synaptic spiking determines the direction and the magnitude of synaptic modification. If post-synaptic spiking happens within the window of 20 millisecond after pre-synaptic excitation, this synapse experiences LTP. To achieve strong LTP, post-synaptic spiking must be very close to the pre-synaptic spiking. On the other hand, LTD happens if post-synaptic spiking occurs before pre-synaptic spiking. The magnitude of LTD is also the largest when post-synaptic spiking is very close to the pre-synaptic spiking. Besides, significant LTP happens only when initial state of a synapse is relatively low, while LTD is uniformly observed given any initial state of a synapse. Figure 4-2 shows the experimental data of synaptic change to relative timing of pre- and post-synaptic spiking. This model of synaptic modification is known as spiking-timing-dependent-plasticity (STDP). In fact, STDP is consistent to our common sense. If post-synaptic spiking happens shortly after pre-synaptic spiking, then it is very likely, that this pre-synaptic event triggers the post-synaptic event. The synapse remembers this information by increasing the synaptic weight. If post-synaptic spiking happens before pre-synaptic spiking, then this post-synaptic event is not caused by this pre-synaptic event. The synapse records this information by decreasing the synaptic weight.

Interestingly, a memristor is very compatible with STDP. Recall that a memristor is also a two-terminal device, whose resistance can be altered according to the direction and magnitude of the applied voltage. It is reasonable to suppose that there is a certain function, whose magnitude is the largest around y axis. Moreover, the polarity of this function in region $x > 0$ is opposite to $x < 0$. If both conditions are satisfied, then the differential of two functions with different latencies can yield a model of STDP. B. Linares-Barranco and T. Serrano-Gotarredona

point out that membrane voltage has the shape satisfying these two conditions^[16]. They prove that if post-synaptic spiking is later than pre-synaptic spiking, their differential result will give a positive above-threshold voltage across the memristor, which results in a decrease to memristance. If pre-synaptic spiking is later than post-synaptic spiking, their differential result will give a negative above-threshold voltage across the memristor, which results in an increase to memristance. If pre-synaptic spiking and post-synaptic spiking are far away from each other, their differential result will remain under the threshold voltage of a memristor. In this scenario, the memristance will experience no change or subtle change. Shortly after this discovery, Sung Hyun Jo *et al.* experimentally proved that the aforementioned voltage difference can indeed yield corresponding memristance alteration^[35]. Recently, more synaptic behavior of memristor synapse has been reported. Ran Jiang *et al.* demonstrated that, for certain types of memristor, both short-term memory and long-term memory can be observed^[36]. The memristor exhibits fatigue response to certain direction of input. The emergence of habituation is almost simultaneous to the switching from short-term memory to long-term memory. From this perspective, the resemblance of a memristor and a biological synapse is even closer.

Chapter V Memristor Based Neuromorphic Chip Design

Designing memristor based neuromorphic chip has been a research interest since 2008. As discussed in the previous chapters, memristor crossbar can store the synaptic information. Since RRAM structure is the mainstream of fabricated memristor, memristor based neuromorphic circuit should be very compatible with crossbar structure. A few years ago, memristor bridge synapse circuit was proposed by Kim *et al.*^[37]. To retrieve synaptic information from the bridge synapse, one must obtain a pair of differential voltage signal from the memristor bridge. The bridge synapse is very compact, but it is hard to use it in dynamical neuromorphic circuit, where synaptic dynamic plays an important role. As a result, the bridge synapse is discussed in the realm of analog artificial neural network. In this chapter, I will focus on neuromorphic circuits. These circuits are more sophisticated and simulate the real biological behavior of nervous systems.

V.1 Ideal Memristor Based Neuromorphic Circuit

Though different neuromorphic circuit has different shapes and different usage, some blocks are necessary. In this section, I will first introduce these blocks in an ideal circuit.

An ideal memristor based neuromorphic circuit has four fundamental components: a synapse block, which is modelled by a memristor; an integrator, which is modelled by an operational amplifier with capacitive negative feedback; a comparator, whose positive input is connected to the output of the integrator, and negative input is connected to a fixed threshold voltage source; a spike generator, which gives a specific spike forward to next synapse and backward to the previous synapse^[38]. Figure 5-1 shows the blocked circuit.

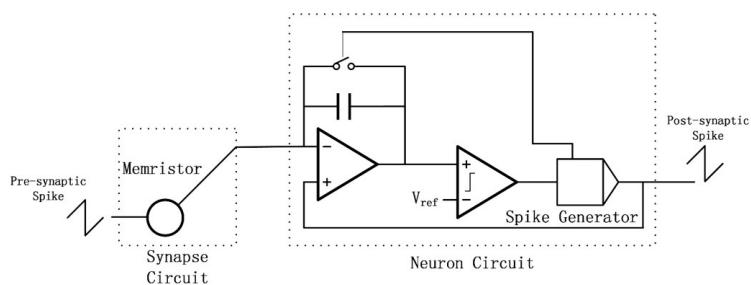


Figure 5–1 Ideal memristor based neuromorphic blocked circuit

When pre-synaptic event comes, the spike voltage is transformed to an input current by the synapse. The amplitude of this input current is determined by the memristance of the synapse. In the meantime, the input current is integrated by an ideal integrator. As charge accumulates on the integrator capacitor, the output of this integrator gradually raise. When it crosses the threshold voltage of the comparator, the comparator will fire a signal to the spike generator. The spike generator then does two jobs: first, it turns off the integrator, by connecting its negative input to its output; second, it generates a spike. This spike will travel to the next neuron, and back to the synapses that contribute to this event. Thus for all the input synapses, their top electrodes are connected to the input spike, while their bottom electrodes are connected to the output spike. From chapter 4, we know that for specific spike shapes, the time difference between the input spike and output spike gives rise to STDP. This means that, ideally speaking, neuromorphic circuit can constitute an STDP self-learning neural network.

Using neuromorphic circuit discussed above, we can construct a five-times-sixteen network capable of pattern learning. The network is shown in Figure 5-2. Pattern 'SJTU' is organized to a five-times-sixteen black-and-white image. All pixels go to synapse according to its spatial occurrence. For a black pixel, we instantly give a spike to the synapse, while for a white pixel, we also give a spike but with a large delay. All synapses are connected to a single neuron, because we assume all pixels in this image are related. This network is built and simulated in Spectre.

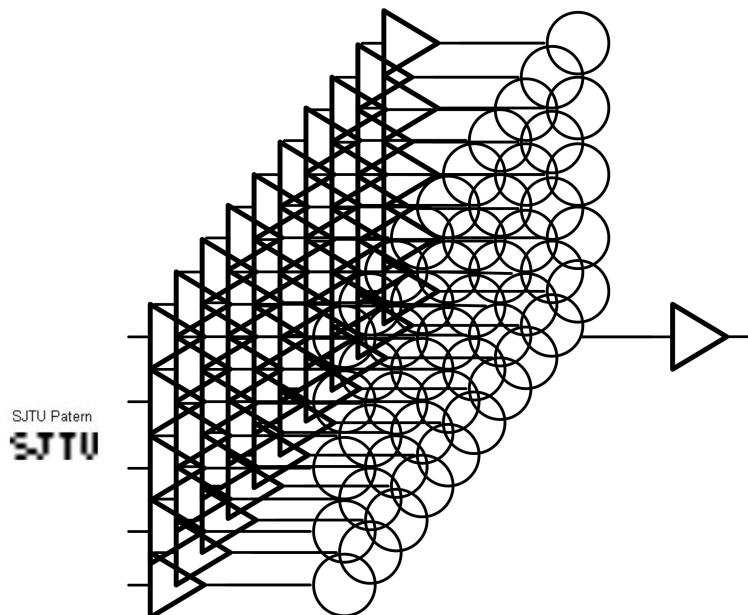


Figure 5-2 5 × 16 synaptic network

The result is shown in Figure 5-3. As can be seen, the memristance in all eighty synapses are randomly distributed, and the number of epochs represents the learning cycles. As can be seen, although initially, the network shows no sign of 'SJTU' pattern, after twenty five epochs, the network successfully stores 'SJTU' in it.



Figure 5-3 Simulation result for learning in ideal memristor based neuromorphic circuit

Although the simulation result shows that memristor based neuromorphic circuit can successfully record spiking information, there are still many things to consider. First, input spikes are generated without noise. For all white pixels, they have synchronous input, so do all black pixels. This is however not the situation in real world. In reality, jitter always exists. For some inputs that are supposed to be white pixels, they might have exceptional sluggish input. When recording information like this, the system can have false cognition of inputs. Second, in all circuit blocks, ideal circuit properties are assumed. All the circuits are not in gate level. Issues like non-linearity, mismatch will have influence on the system.

In the next section, memristor based neuromorphic circuit on silicon will be introduced.

V.2 Memristor based neuromorphic circuit on silicon

In this section, I will introduce my memristor based neuromorphic circuit on silicon block by block. Figure 5-4 shows the memristor based neuromorphic circuit architecture. All the circuits are designed using XFAB 0.18 μ m technology.

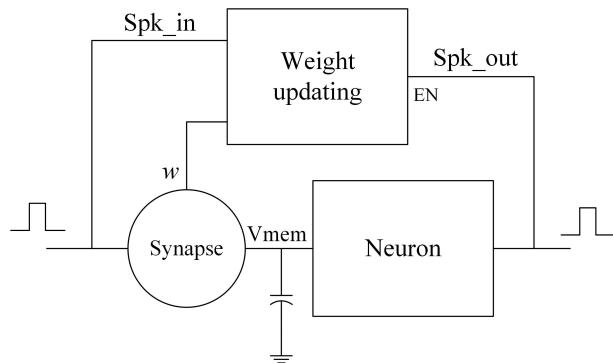


Figure 5-4 Memristor based silicon neuromorphic circuit architecture

V.2.1 Synapse Circuit

In previous chapters, a memristor is regarded as a synapse structure. Indeed, it is likely that these tiny two-terminal can solely imitate biological synapse. Unfortunately, different process of memristor have totally different electrical characteristics. For researchers, it is not wise to assume a categorical feature on memristor. Besides, as we shall see in the upcoming sections, even when we have chosen a mathematical memristor model for synapse circuit, it may not benefit much from the ideal memristor synapse topology. On the other hand, silicon integrated circuit has very robust performance. Analog integrated circuit is a mature topic. Here, I will propose a combined memristor-CMOS synapse circuit.

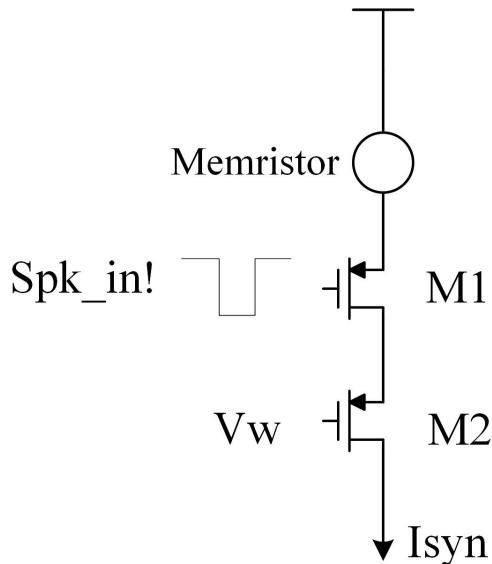


Figure 5-5 Memristor based pulsed current source synapse

The circuit is shown in Figure 5-5. This synapse circuit only has two pmos transistors (M1 and M2) and one memristor, thus it is very compact. Input spikes are digital events. Length of the event is determined by the output of sensors or the output of other neurons. When the spike is digital '0', the synapse is turned on. When the spike is digital '1', the synapse is turned off. The spike signal is connected to the gate of M1 as shown in Figure 5-5. The source of M1 is connected to the bottom electrode of memrsitor. When the synapse is turned on, equivalent resistance of M1 from drain to source is very small compared with memristance. Thus the equivalent resistance of M1 and memristor is

$$R_{equ} = R_{mem} \quad (5-1)$$

where R_{mem} is the memristance. M2 is another pmos transistor, biased by local current source. The function of M2 is to limit the current upper bound. In fact, if looking into M2 from its drain to its source, structure of M2 and resting circuit equals to a common source amplifier with source degeneration. Its output resistance is multiplied by a factor of $g_m R_{mem}$. In later blocks, I will demonstrate that input current is about 30 nano ampere, which means that M2 is in subthreshold region. Value of g_m equals to $\frac{I_d}{nV_t}$, which yields a small value of $g_m R_{mem}$. Output of this synapse is a pulsed current I_{syn} , which lasts as long as the input spike. This synapse is also called pulsed current-source synapse, which is probably the first silicon synapse circuit working in the subthreshold region. It is very simple and does no signal processing to the input voltage spikes. However, this circuit is widely used in neuromorphic architecture for its compactness and robustness. Figure 5-6 shows some simulation results of this circuit. In Figure 5-6, red curve is synaptic current when the memristor is in low resistance state, and green curve is synaptic current when the memristor is in high resistance state. It is easy to observe, that though maximum value of green curve is smaller than the red curve, it is not small enough to turn off the synapse circuit. This is because for our memristor model, its resistance does not dominate the behavior of the synapse. Since the circuit works in subthreshold region, the current is already very small, we need another way to retrieve information from memristors.

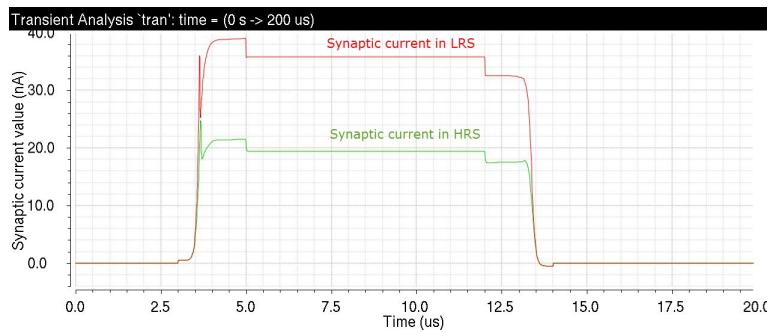


Figure 5-6 Simulation result for memristor based pulsed current source synapse

Figure 5-7 shows the new memristor based synapse circuit. Instead of controlling the state synaptic current, we can use memristor to control the state of the bias generator. The synapse structure is shown in the red dotted circle. The only difference between it and the aforementioned synapse, is that it does not have memristor connected to the synapse. Instead, in the bias generating block, another pmos transistor is used to control the bias voltage of the synapse. This transistor is further controlled by a resistance detection block, which detects the memristance and gives a digital '0' if the memristance is low, and a digital '1' if the memristance

is high.

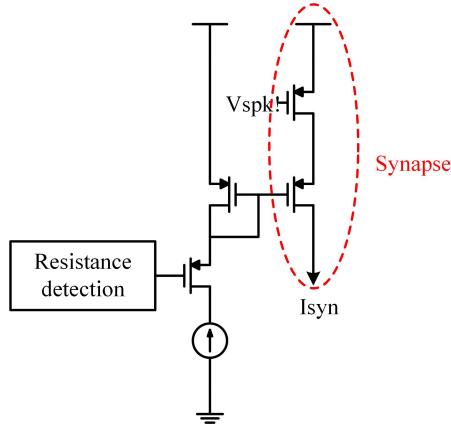


Figure 5–7 Modified memristor based pulsed current source synapse

Design of resistance detection block is based on two criteria: First, the power consumption should not dominate other circuit blocks; Second, the difference between the output voltage should be the extreme case. The simplest thought is to directly connect a fixed value resistor to the memristor. However, because the memristance is floating in a certain range, it is hard to make sure that the resistance of this particular resistor is in the suitable value. Figure 5–8 gives the current response of a low-resistance state memristor and a high-resistance state memristor to a pulsed 1.5V voltage input. We can conclude that equivalent resistance of our memristor is in the range of $M\Omega$. Thus the unit of current flowing through the memristor should be at least $100nA$.

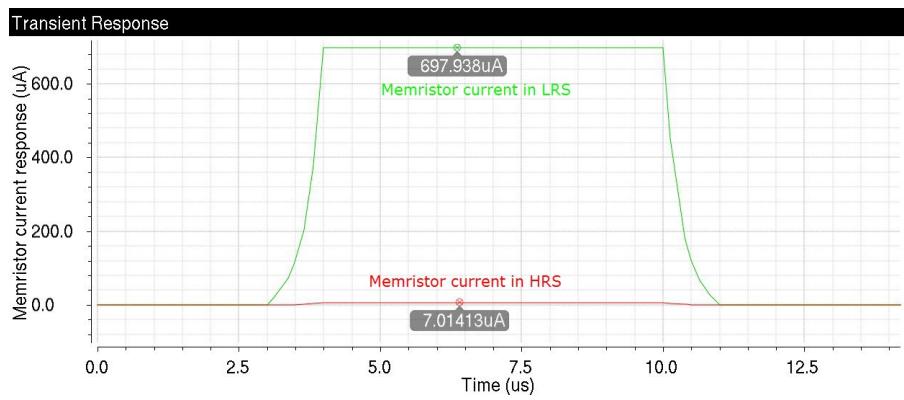


Figure 5–8 Memristance's response to pulsed 1.5V input

Figure 5–9 shows the design of resistance detection block. V_{ref1} and V_{ref2} are two reference voltages that yield $300nA$ current. V_{ref3} is a reference voltage that yields $1\mu A$. V_{thr} is the

threshold voltage that determines whether the memristor is in high resistance state or low resistance state. In this way, synaptic information is retrieved from memristor. The layout of the synapse and the resistance detection block is shown in Figure 5-10 and Figure 5-11. Note that local current source is designed both for synapse block. The current source is simple charge pump bias with a diode connected pmos to lower the current value, this is because, generating a current in the realm of nA is not easy. The size of the synapse is $24.66\mu m \times 26.22\mu m$, and the size of the resistance detection block is $9.2\mu m \times 18.5\mu m$. To make the circuit work, one needs to connect the output of resistance detection block to its corresponding Vres input of the synapse, and connects the memristor to the corresponding resistance detection block.

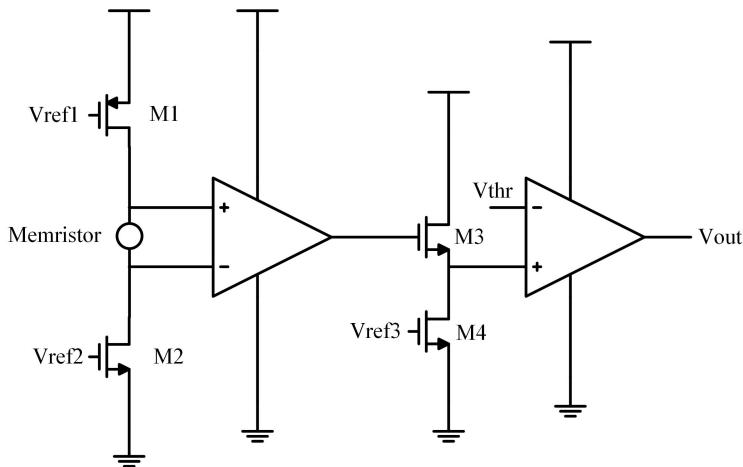


Figure 5-9 Resistance detection circuit

V.2.2 Neuron Circuit

Neuron circuit is another important circuit block in the memristor based neuromorphic circuit. Operation of neuron circuit is divided into following stages: First, it accumulates the synaptic contribution from all the synapses that connect to it. As discussed in section 5.1.1, synaptic contribution is modelled as pmos current. From the definition of capacitor, we know that a capacitor can linearly accumulates the injected current. This capacitor is also called membrane capacitor, because its working mode is similar to the membrane of a biological neuron. Second, cumulative effect of synaptic contribution, which is quantified as the membrane voltage on membrane capacitor, is compared to a threshold voltage, which determines weather the contribution of synapse is enough to trigger an output spike event. Third, a spike generation circuit block generates a fixed length spike whenever membrane voltage crosses threshold voltage. Fourth, a refractory block raises the membrane voltage above threshold during the

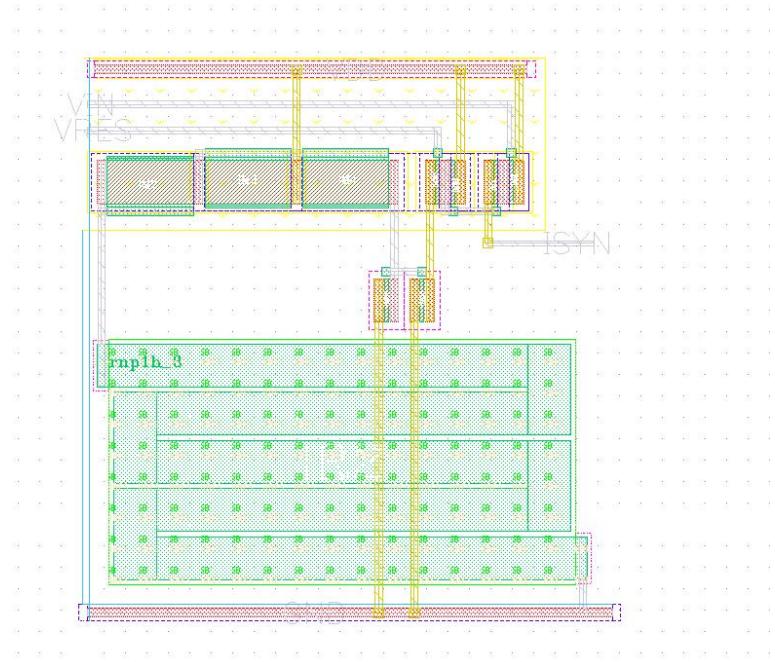


Figure 5–10 Layout for synapse circuit

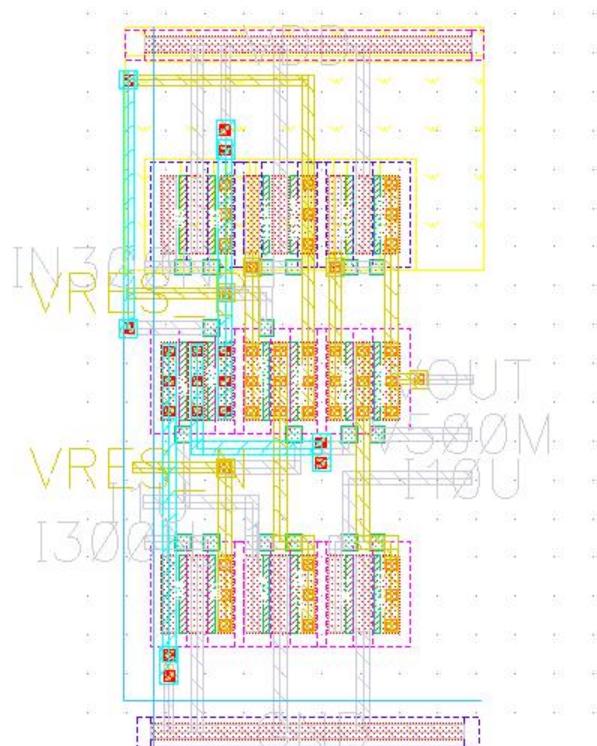


Figure 5–11 Layout for resistance detection circuit

duration of output spike event, and keeps it low shortly after the ending of output spike event, to protect the neuron from firing in a particular length of time. To model this operation, a wide range of circuit topology can be found in^[28]. For memristor based neuromorphic circuit, it is better to choose a simple neuron to demonstrate the idea..

The neuron circuit is shown in Figure 5-12. In this circuit, I_{in} is the synaptic input current. Because in biological neuron, if synaptic contribution is not able to trigger an output event, it gradually vanishes. This is modelled as a leakage current. In this circuit, V_{lk} controls the amplitude of leakage current. These two current sources are connected to charge a membrane capacitor C_{mem} . The operation amplifier is used to detect whether the membrane voltage crosses the threshold voltage. If the answer is yes, it outputs a high voltage, which is amplified by an inverter and gives a output spike event. In the mean time, the output spike turns on I_{Na} switch to pull up the membrane voltage, and turns on I_{Kup} switch to charge the refractory capacitor C_K . The spike time length t_{spk} is determined by the capacitance of C_K and the magnitude of I_{Kup} . When voltage on C_K increases to a certain value that turns I_K on, the membrane voltage will rapidly drop to zero and ends this output spike event. Moreover, I_K prevents synapse from contributing for a length of time determined by the magnitude of I_{Kdn} . The length of this time t_{refra} and t_{spk} satisfy the following equations.

$$t_{spk} = \frac{1}{2} \frac{Vdd * C_K}{I_{Kup}} \quad (5-2)$$

$$t_{refra} = \frac{1}{2} \frac{Vdd * C_K}{I_{Kdn}} \quad (5-3)$$

Figure 5-13 shows the simulation result of the neuron circuit. The red curve is the output spike, and green curve is the membrane voltage. From the figure we can observe that membrane voltage piles up gradually as synaptic contribution accumulates. When it crosses a threshold voltage, it ascends to VDD rapidly and the output spike event is generated. The length of spike is around $13\mu s$, which satisfies $\frac{VddC_M}{I_{Kup}} \approx 27\mu s$. This neuron, because of the usage of an operational amplifier, is also called an opamp-neuron.

The layout for opamp-neuron is shown in Figure 5-14. As can be seen, the area of capacitor occupies majority of the layout size. The total layout size is $24.8\mu m \times 70.1\mu m$. The circuit is biased by a local current source (not shown), which will be covered later.

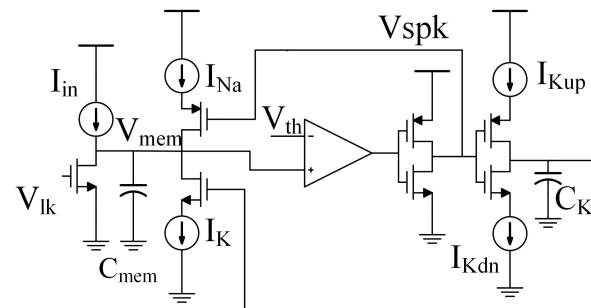


Figure 5–12 Opamp-neuron circuit

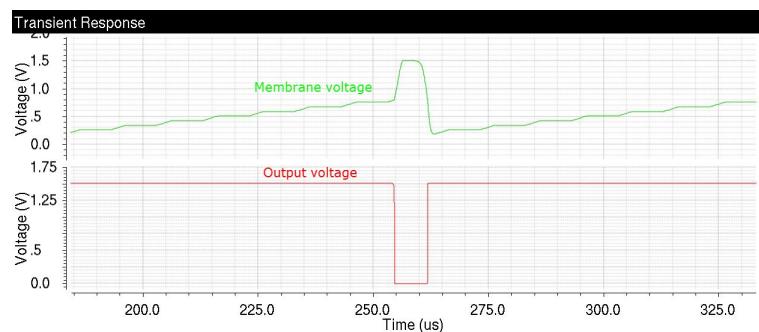


Figure 5–13 Simulation result for opamp-neuron

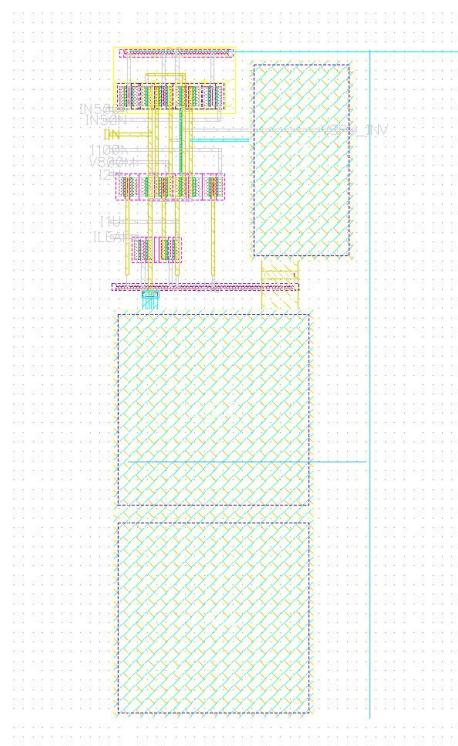


Figure 5–14 Layout for opamp-neuron

V.2.3 Function Generator Circuit

As discussed in chapter 4, memristor is related to STDP through specific shape of learning waveforms. Every synapse is connected to a function generator circuit block, which sends signals to the memristor crossbar and alters the memristance. The learning waveform of the function generator circuit should have following features: First, it rests at a middle level between Vdd and ground voltage. This is to ensure that memristance stays unchanged when no learning waveform is imposed. Second, it must switch rapidly from high voltage to low voltage. This is to meet the requirement of STDP waveform shape.

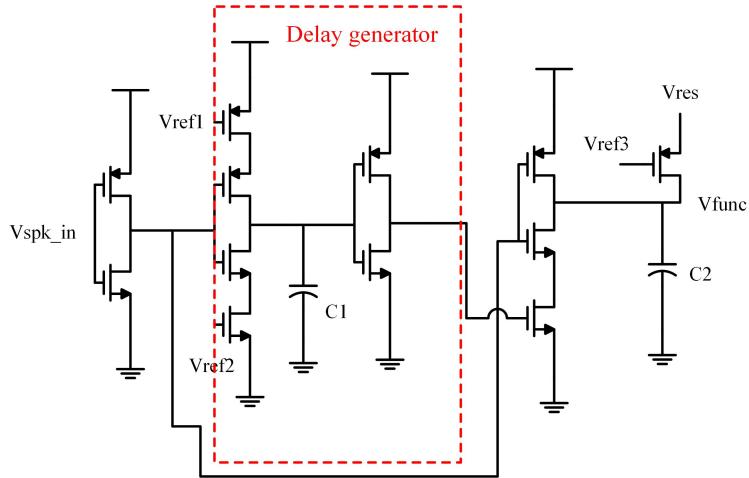


Figure 5–15 Function generator circuit

To meet these two features, I designed a function generator circuit, shown in Figure 5-15. V_{ref1} and V_{ref2} are two reference voltage to control the charging and discharging speed of $C1$. V_{ref3} is used to slowly drive the V_{func} to V_{res} when given no input spike. The delay generator postpones the input spike for a short period of time. This delayed signal, together with input spike, will generate the learning function for memristor. Figure 5-16 shows the simulation result. The waveform has both of two features mentioned earlier. The layout of function generator circuit is shown in Figure 5-16. Size of this layout is $22.7\mu m \times 32.3\mu m$. The circuit is biased by a local current source (not shown), which will also be covered later.

V.3 Three-Synapse One-Neuron Memristor Based Neuromorphic Chip

Using the circuit blocks designed in Section 5.2, we can further design a neuromorphic chip. To simplify the routing and better demonstrate the idea of memristor based neuromorphic chip, I choose to implement three synapses and one neuron, together with function generator in

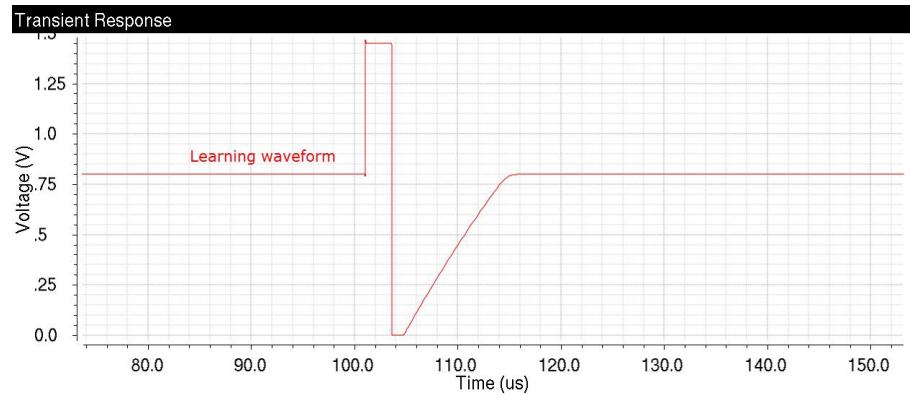


Figure 5–16 Simulation result for function generator circuit

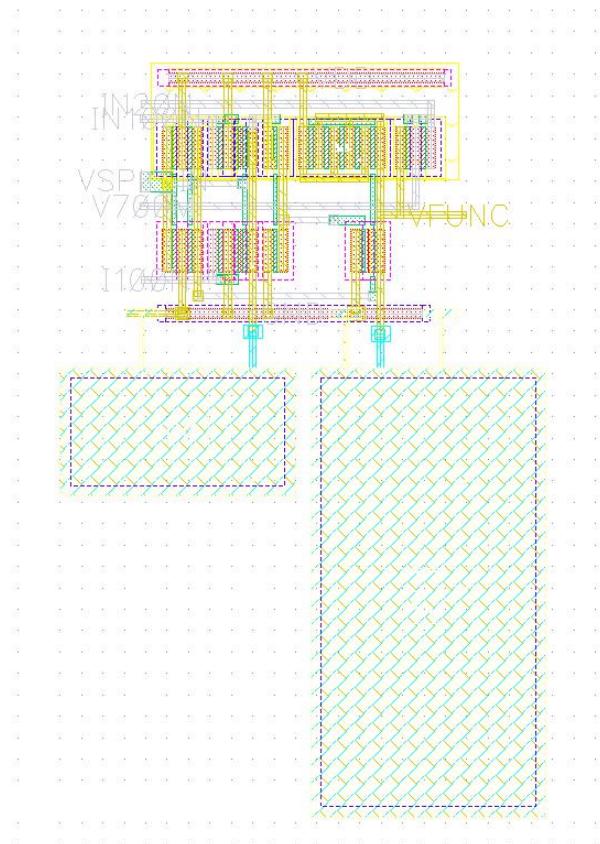


Figure 5–17 Layout for function generator

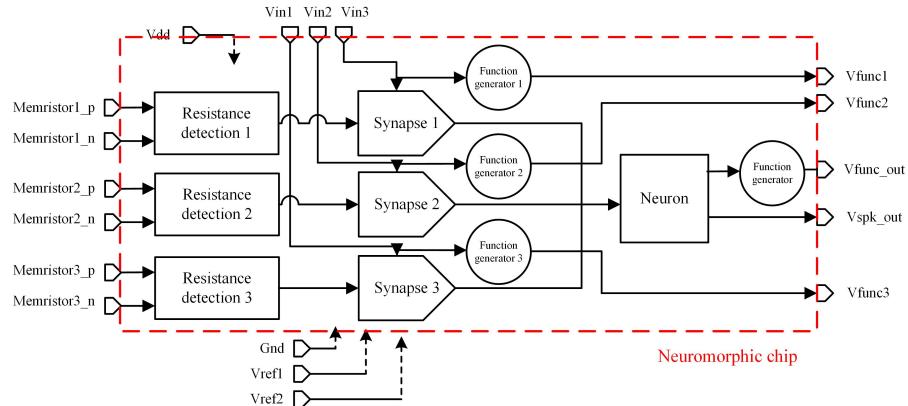


Figure 5–18 Schematic view for neuromorphic chip

Table 5–1 Pin definition for neuromorphic chip

Pin name	Pin definition
Memristor1 p	Memristor 1 top electrode input
Memristor1 n	Memristor 1 bottom electrode input
Memristor2 p	Memristor 2 top electrode input
Memristor2 n	Memristor 2 bottom electrode input
Memristor3 p	Memristor 3 top electrode input
Memristor3 n	Memristor 3 bottom electrode input
Vin1	Input spike to synapse 1
Vin2	Input spike to synapse 2
Vin3	Input spike to synapse 3
Vfunc1	Output function waveform for Vin1
Vfunc2	Output function waveform for Vin2
Vfunc3	Output function waveform for Vin3
Vspk out	Output spike waveform for neuron
Vfunc out	Output function waveform for Vspk out
Vdd	1.5V power input
Gnd	0V power input
Vref1	0.5V reference input
Vref2	0.7V reference input

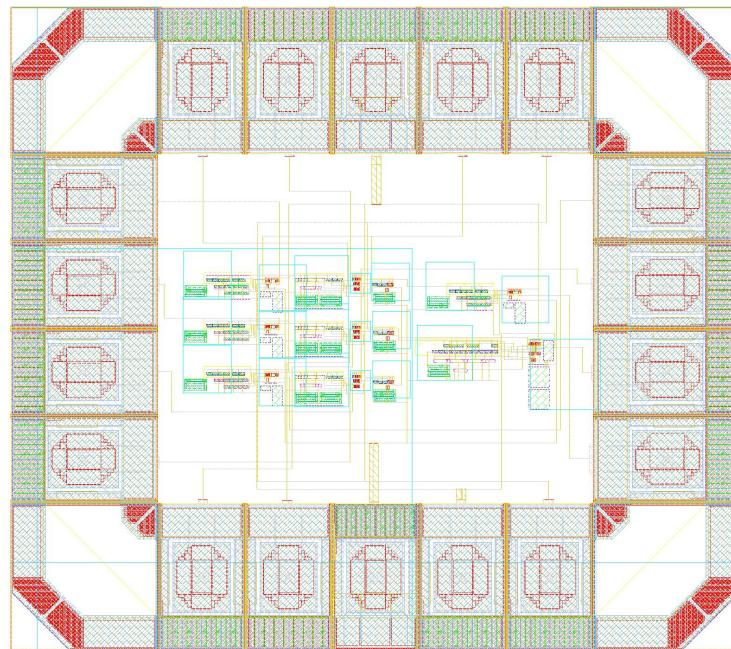


Figure 5–19 Layout for neuromorphic chip

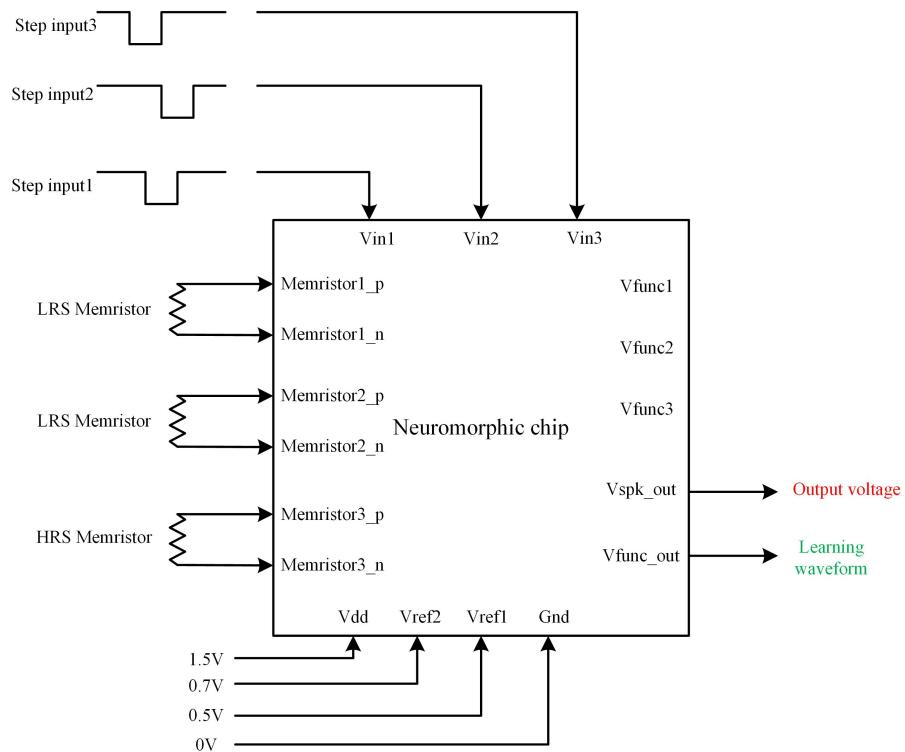


Figure 5–20 Testing circuit for neuromorphic chip

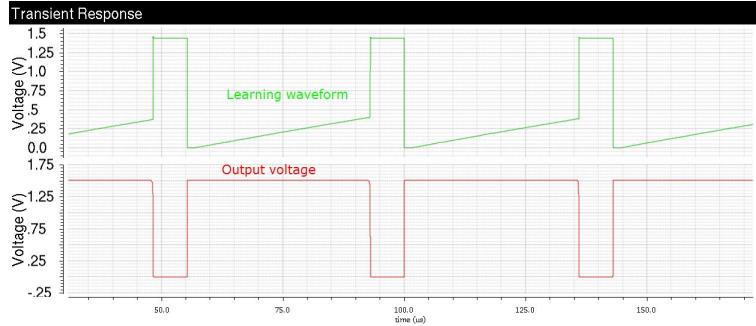


Figure 5-21 Post-layout simulation for neuron's behavior

one chip. Schematic for this chip is shown in Figure 5-18, and layout of this chip is shown in Figure 5-19. The chip has 18 pads. Among them, 6 pads are for memristor connection. 2 pads provide power for the chip. 3 pads provide input spike event for the synapse. Another 3 pads are used to observe the input learning function. The rest of pads are connected to the output of neuron. A specific pin definition is shown in Table 5-1. The whole chip size is $725\mu m \times 638\mu m$.

To demonstrate the correctness of chip's functionality, a testing circuit is designed. The schematic circuit for the testing circuit is shown in Figure 5-20. For Vin1, Vin2, and Vin3, three step voltages with different latencies are given. The memristor is initially set to low resistance state for synapse 1 and synapse 2, and to high resistance state for synapse 3. We observe the waveforms of Vspk out port and Vfunc out port. Post layout simulation result is shown in Figure 5-21. The green curve is the output of learning function, and red curve is the output spike train. Although the spike width is around $13\mu s$, which matches the pre-layout simulation, learning function has some distortion. However, as we will discuss in the next chapter, the learning function cannot be directly applied to memristor. As a result, buffer is needed, which can also modify the shape of learning spike.

V.4 Conclusion

In this chapter, memristor based neuromorphic circuit, is explicitly introduced. From ideal circuit level, I demonstrate the basic idea of building memristor based neuromorphic circuit. From schematic level, I give an example of circuit design using XFAB $0.18\mu m$ technology. From chip's level, I successfully build a simple neuromorphic chip using these circuit blocks.

Chapter VI Implementing Perceptron Learning in Neuromorphic System

In chapter 5, a memristor based neuromorphic circuit is designed. The circuit is able to mimic the behavior of synapse, neuron, and implement STDP. Since existence of spike is one major characteristic of neuromorphic system, it is also called spiking neural network (SNN). Now, the question is, what can neuromorphic circuit do? This question is not easy to answer. Unlike artificial neural network, where data flow as analog values, coding scheme of neuromorphic system is not completed figured out. However, there are some clues about how spiking neural network can be implemented in some applications. In this chapter, I will focus on a typical application of spiking neural network: implementation of perceptron learning.

VI.1 Perceptron Learning

The perceptron is a machine learning algorithm. It is a supervised learning rule for binary classifiers. To be more specific, under the supervision of a teacher, this learning machine can divide a group of points to a few clusters. This is extremely helpful, because sometimes the amount of data is enormous, and it is not economical to distinguish between them all. Though points may be different from each other, some of them can have the same effect or the same meaning to the user. In this scenario, we can use perceptron to lower the cost of recognizing data.

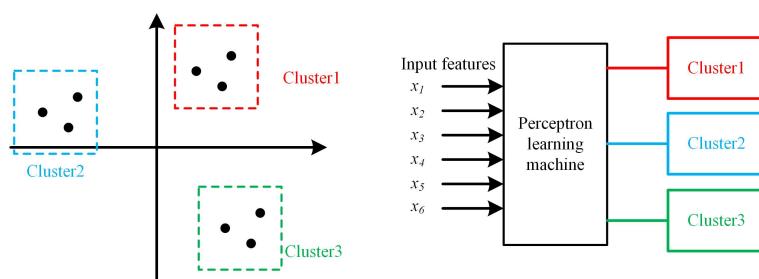


Figure 6-1 Perceptron learning

Figure 6-1 shows the idea of perceptron learning. Inputs are black dots that fall in a large set. These inputs can be classified following a certain rule, *e.g.* dividing points on plane according to their coordinates. When the perceptron learning machine receives these inputs, it first classifies

them into a certain number of categories with the help of an external teacher. After this learning period, it remembers the common features of each cluster. Next time somebody gives it an input from the input set, it immediately gives the answer to which this input belongs to. As we shall see in upcoming sections, the spiking neural network is able to complete this task.

VI.2 Spike Encoding

Technically speaking, using electronic devices, it is able to mimic synapses and neurons to the greatest extent. Thus neuromorphic systems can resemble biological nervous system according to the understanding of neuroscientists. The next step is to give the exact input spike train, to derive output that makes biological sense. For example, we can use these systems to build silicon cochlea and retina. The only thing we need to know, is to figure out how to encode these spikes. Unfortunately, this is also the hardest part of neuroscience. In this section, I introduced some hypothesized coding schemes.

- Rate coding

The rate coding model is a model for neuronal firing communication. It states that as the intensity of a stimulus, either from sensors or output from other neurons, increases, the frequency or rate of output spike event (action potential of this particular neuron) increases. It is also known as frequency coding. Rate coding assumes that most information, in the output of a neuron, is contained in the rate of firing. Since sequence of spike varies from trial to trial for a neuron, response of neurons is usually treated statistically. Thus firing rate is a good way to characterize a specific pattern of spike.

- Temporal coding

If precise timing of spike train is found to carry information, the coding scheme is called temporal coding. For example, in auditory cortex, tone-evoked responses have very clear temporal characteristic^[39]. When sequence of spike is found to have temporal coding scheme, it is natural to employ STDP in such systems. However, temporal coding spike train is highly susceptible to noise and fluctuations.

In reality, all aspects of spike train are potential to carry information. It is too ambitious to fully understand information flows in our brain, which is definitely a wonder by the nature. In my thesis, I will try to use frequency coding scheme to demonstrate perceptron in neuromorphic system.

VI.3 Perceptron Learning Neuromorphic Architecture

To realize perceptron, a new neuromorphic circuit structure is needed. This is because neuromorphic structure in chapter 5 is designed for temporal encoded spike sequence. However, perceptron learning is hard to implement to temporal encoded spike sequence, so I choose to design a new neuromorphic structure compatible for rate coded spike sequence. The circuit structure of the new neuromorphic architecture is shown in Figure 6-2. In this chapter, all circuits are designed using XFAB 0.35 μm technology.

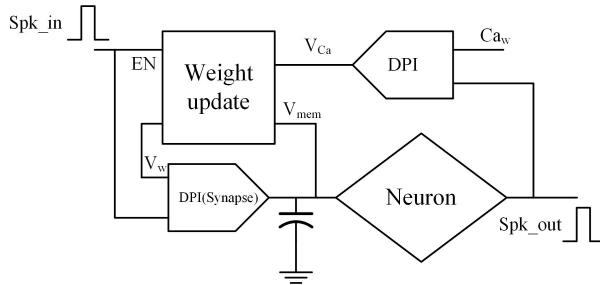


Figure 6-2 Neuromorphic blocked circuit capable of performing perceptron learning

VI.3.1 Synapse Circuit

To implement rate coding scheme, the synapse should have some biological features, so the synapse designed in chapter 5 is not suitable. In short, instead of directly transform a voltage spike to a synaptic current, it first does low pass filtering to the voltage spike, then injects the filtered current to the neuron. This can be done using weakly inverted transistors. Figure 6-3 shows the topology of synapse circuit. In this circuit, V_{in} is the terminal for input spike train. V_{thr} , V_w , and V_{tau} determine the filtering parameters. To compute the synthesized current, we first compute I_{in} .

$$I_{in} = I_w \frac{e^{\frac{\kappa V_{syn}}{V_t}}}{e^{\frac{\kappa V_{syn}}{V_t}} + e^{\frac{\kappa V_{thr}}{V_t}}} \quad (6-1)$$

where κ and V_t are process and temperature dependent parameters. By multiplying numerator and denominator of equation 6-1 by $e^{-\frac{\kappa V_{dd}}{V_t}}$, we derive

$$I_{in} = \frac{I_w}{1 + \frac{I_{syn}}{I_{thr}}} \quad (6-2)$$

where $I_{thr} = I_0 e^{-\frac{\kappa(V_{thr}-V_{dd})}{V_t}}$ represents a circuit parameter. For synthesized capacitor, it satisfies

$$C_{syn} \frac{d}{dt} V_{syn} = -(I_{in} - I_{tau}) \quad (6-3)$$

Combine 6-3 and 6-2 using relationship $I_{syn} = I_0 e^{-\frac{\kappa(V_{syn}-V_{dd})}{V_t}}$, we derive

$$\tau \frac{d}{dt} I_{syn} + I_{syn} = \frac{I_w}{I_{tau}} \frac{I_{syn}}{1 + \frac{I_{syn}}{I_{thr}}} \quad (6-4)$$

Equation 6-4 is a first-order non-linear equation. However, I_{syn} rises as time, eventually resulting in condition $I_{syn} \gg I_{thr}$. This gives to

$$\tau \frac{d}{dt} I_{syn} + I_{syn} = \frac{I_w}{I_{tau}} I_{thr} \quad (6-5)$$

Equation 6-5 becomes a first-order linear differential equation, thus the response of this circuit can mimic the behavior of synapse. Figure 6-4 shows the simulation result of this synapse. The green curve is the synthesized current I_{syn} . Though input voltage spike only lasts about $6\mu s$, its synaptic current response extends to a much longer time. This low pass filtering property will be beneficial for implementing perceptron. Since this synapse has a differential pair structure and does low-pass filtering to the input voltage signal, it is called Differential-Pair-Integrator (DPI).

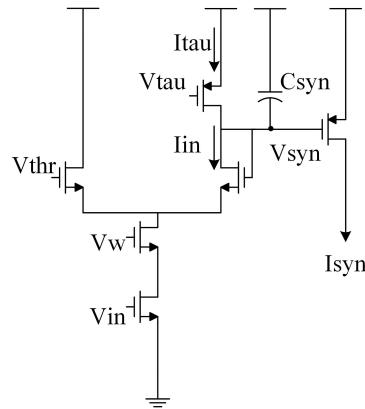


Figure 6-3 Differential pair integrator circuit

VI.3.2 Neuron Circuit

Like synapse circuit, neuron circuit also needs some modifications to incorporate more biological characteristic. By observing biological experimental data, it has been shown that given a constant synaptic input, frequency of action potential decreases if the neuron keeps on

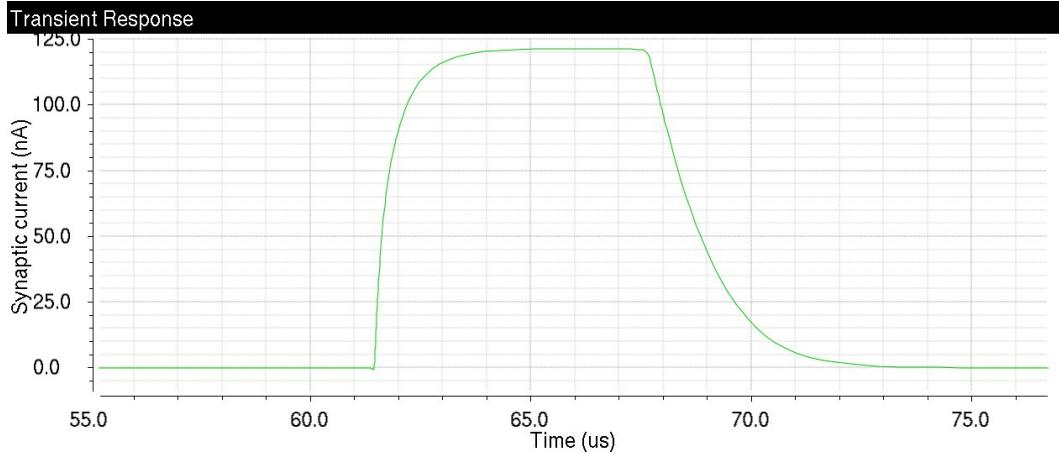


Figure 6-4 Simulation result for differential pair integrator circuit

firing. To model this behavior in circuit level, new low-power neuron is built. Figure 6-5 shows the circuit structure of this neuron. Compared with Figure 5-12, this neuron has an additional ahp leakage current at membrane. The ahp current block is a differential-pair-integrator (as introduced in section 6.3.1), whose input is the inversion of spike event. By doing so, when the neuron fires with short interval, the filtered signal keeps high. This will turn on the ahp leakage current switch and negatively contribute to the membrane voltage.

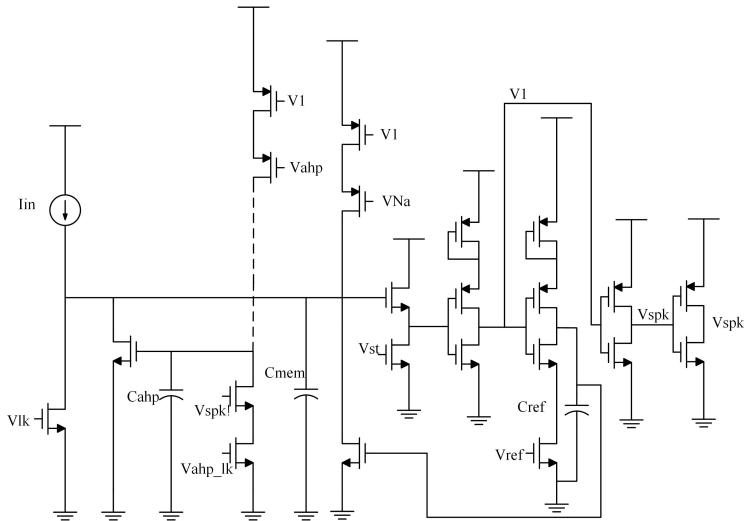


Figure 6-5 Low power neuron circuit

Figure 6-6 shows the neuron's response to a constant synaptic input. The green curve is the action potential of neuron, and the red curve is the membrane voltage of neuron. It can be seen that, at first, two times of synaptic contribution trigger an output spike (action potential). The

number two is indicated by zigzag number on membrane voltage. As the neuron continues to fire, the number of zigzags increases, which means that the neuron becomes harder to fire.

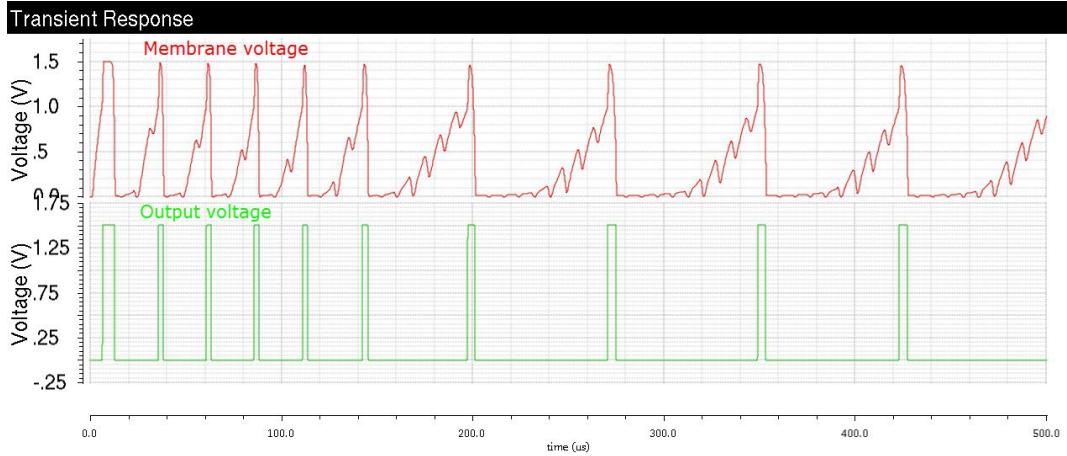


Figure 6-6 Simulation result for low power neuron with ahp leakage current

VI.3.3 Learning Circuit

To implement perceptron learning in neuromorphic architecture, modification is also needed to learning circuit (weigh updating circuit). First, function generator circuit designed in chapter 5 is an input spike driven learning module, which is highly sensitive to noise, yet in perceptron, noise in spike train is a very important ingredient. Second, function generator circuit is not sensitive to frequency information, which needs to be decoded when updating synaptic weight. To solve these two problems, a new learning circuit, which is output spike driven and capable of detecting output frequency is designed.

The learning process is divided into several procedures. First, to detect output frequency, another differential-pair-integrator is connected to the spike output, thus when the neuron fires, synthesized current of the differential-pair-integrator can reflect the firing rate. Second, membrane voltage of the neuron is retrieved to determine the learning direction. Third, a weight updating module aggregates all information and output a learning signal to w at synapse, as shown in Figure 6-7. The learning algorithm is

$$I_{UP} = \begin{cases} I_B, & \text{If } I_{k1} < I_{Ca} < I_{k3} \text{ and } V_{mem} > V_{mth} \\ 0, & \text{others} \end{cases} \quad (6-6)$$

$$I_{DN} = \begin{cases} I_B, & \text{If } I_{k1} < I_{Ca} < I_{k2} \text{ and } V_{mem} < V_{mth} \\ 0, & \text{others} \end{cases} \quad (6-7)$$

where I_b , I_{k1} , I_{k2} , I_{k3} , and V_{mth} are parameters. I_{Ca} is the output of differential pair integrator. Non-zero I_{UP} increases the synaptic weight and non-zero I_{DN} decreases the synaptic weight. The mechanism of this algorithm is as follows. When frequency of action potential is neither too high nor too low, the weight updating module modifies the synaptic weight according to the value of V_{mem} . If V_{mem} is greater than a threshold voltage V_{mth} , the action potential is following input synaptic contribution. The output spike event has a positive relationship to input spike event, so the weight updating module increases the synaptic weight. If V_{mem} is lower than V_{mth} , the action potential is followed by an input spike event, thus the output spike event has a negative relationship to input spike event. As a result, the weight updating module decreases the synaptic weight.

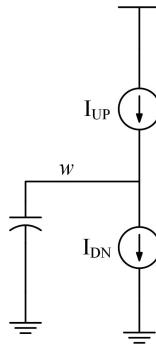


Figure 6-7 Weight updating circuit

VI.4 Perceptron Learning Test

To perform perceptron learning using the neuromorphic architecture, I build a 5×3 synapse network with an additional teacher synapse. These sixteen synapses are connected to a mutual neuron, as Figure 6-8 shown. Among them, fifteen synapses represent fifteen pixels, and the only teacher neuron represents the label of input pattern. The input pattern is divided into two groups. The first group is number 1, and the other group is number 2. Shape of these two patterns are shown in Figure 6-9. For each white bit, a low-frequency input Poisson spike train is given. For each black bit, a high-frequency input Poisson spike train is given. The term Poisson spike train means that the distribution of spike in a round obeying Poisson distribution. This spike train is generated in MATLAB using random number generator. The teacher signal

is either a high frequency Poisson spike train or a low frequency Poisson spike train.

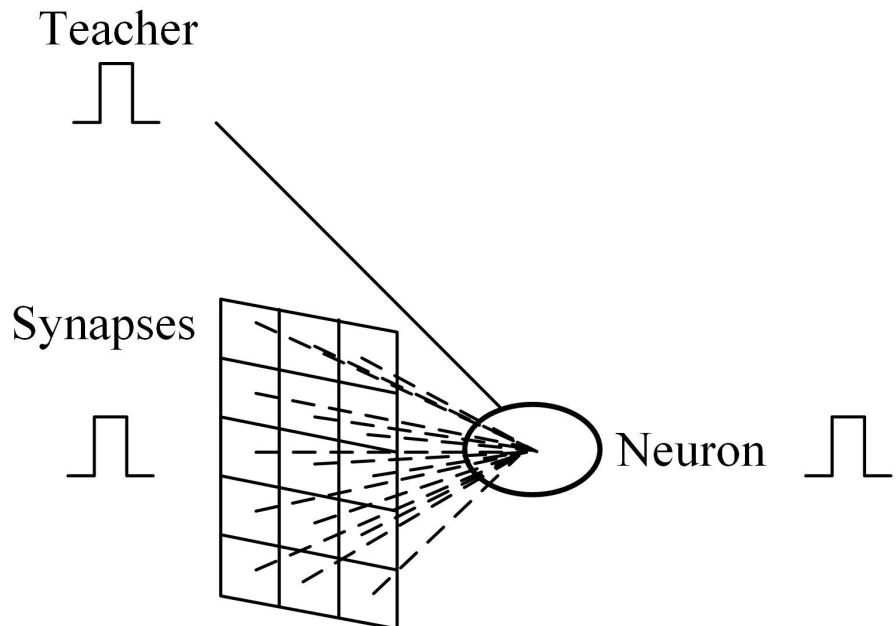


Figure 6–8 3×5 synaptic network

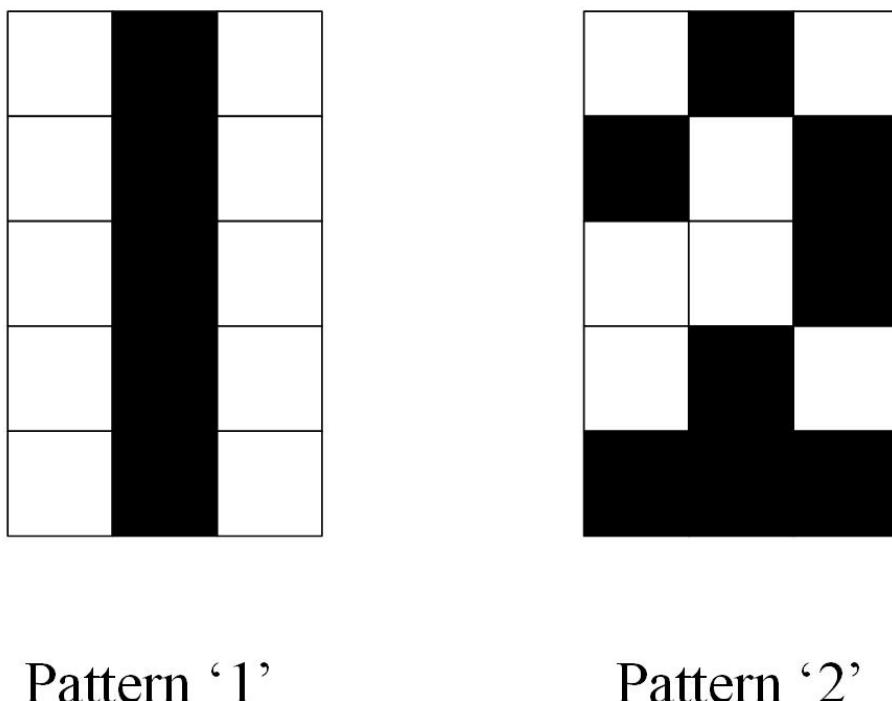


Figure 6–9 Shape of pattern '1' and pattern '2'

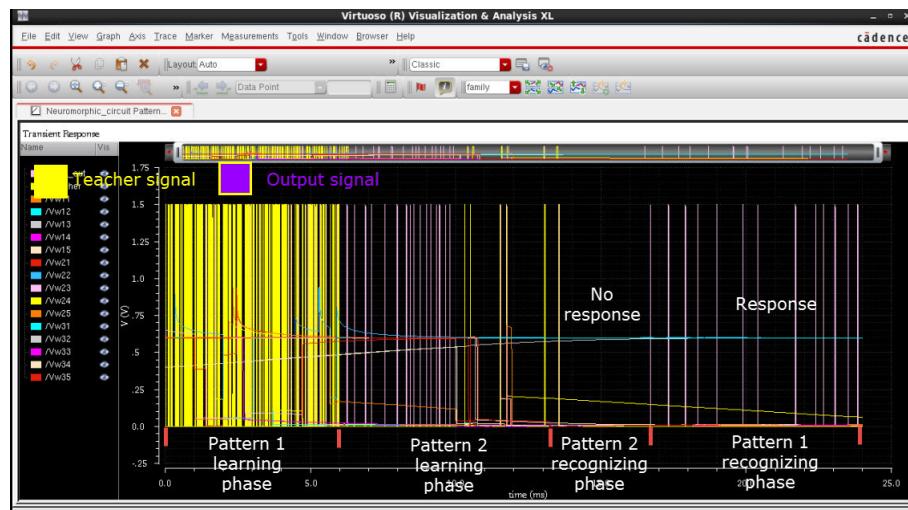


Figure 6–10 Successful case of classification (1)

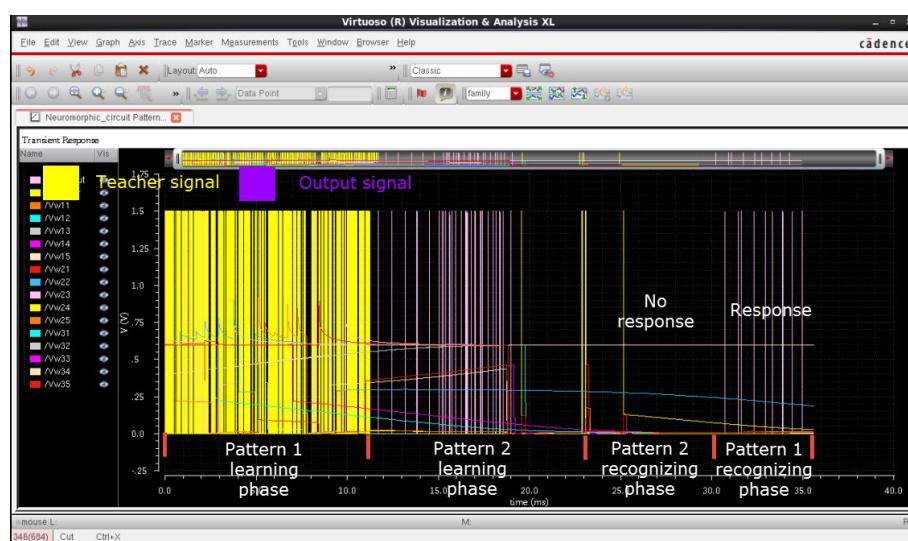


Figure 6–11 Successful case of classification (2)

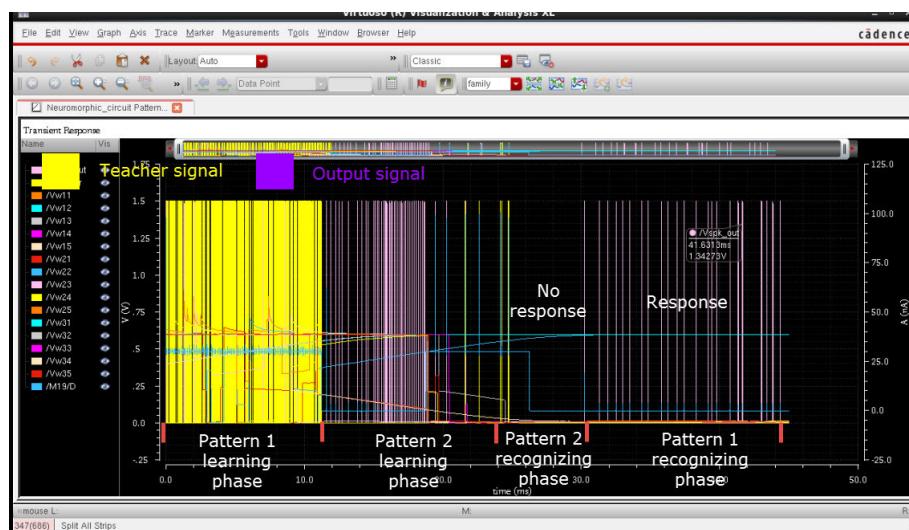


Figure 6–12 Successful case of classification (3)

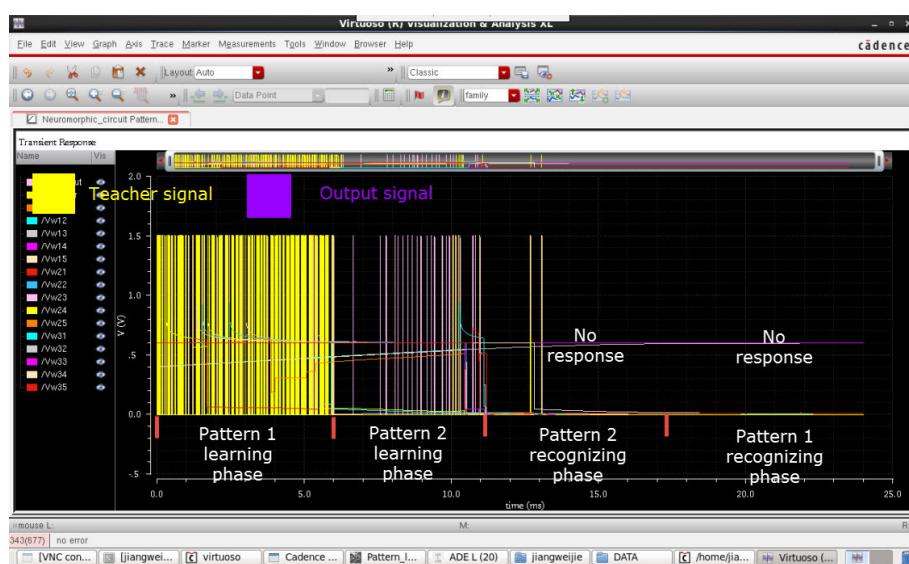


Figure 6–13 Failed case of classification

To start training the network, I first use pattern '1' as input, which is supervised by a high-frequency teacher. After the network recognizes pattern '1', I use pattern '2' as input, which is supervised by a low-frequency teacher. After the training period, a testing period is introduced. At this time, no teacher signal is imposed, and learning mechanism is closed (by tuning I_B to zero). The network is able to distinguish between these two different patterns. Figure 6-10 to Figure 6-12 show some successful results of recognition. The yellow curve is the teacher signal, and pink curve is output of neuron. Other curves represent w for synapses. Learning phase is indicated by the start and the end of the yellow curve, while the beginning of recognition phase is indicated by the end of the yellow curve. The order of input pattern is '1'(learning), '2'(learning), '2'(recognizing), '1'(recognizing). It can be seen that, in all three trials, the neuron still has response to pattern '1', but no response to pattern '2'. However, due to the stochastic nature of the input signal, some failed learning case can happen. One of these trials is shown in Figure 6-13.

VI.5 Conclusion

In this chapter, a neuromorphic architecture capable of perceptron learning is built. The circuit is able to be extended to any scale. In this architecture, circuit blocks have more biological features and are more close to their corresponding blocks in animal's nervous system. All the circuits are built using XFAB 0.35 μm technology.

Chapter VII Reflections and Future Work

In this chapter, I will give some explanations about the neuromorphic architecture. These reflections are based on the contents throughout the whole thesis.

VII.1 About Neuromorphic Architecture

In my thesis, neuromorphic architecture refers to synapse and neuron circuits that resemble the biological nervous system. However, the definition of this term is not limited. It also includes electronic devices that resemble other part of our body, not just nervous system. For example, one can build a silicon cochlea, a silicon retina, etc. They can also be classified as part of neuromorphic architecture. Under some circumstances, highly simplified mathematical models like artificial neural network are also classified as neuromorphic architecture, but I am not inclined to this. The reason is, in our body, information is not analog value, but series of complex events. In aforementioned neuromorphic architectures, information flows as action potentials (spikes), which is somewhat more close to biology. It should be always remembered that the ultimate goal of neuromorphic engineering, is to build an electronic system that mimics the behavior of an animal. Figure 7-1 shows the hypothesized whole neuromorphic architecture. In this architecture, central nervous system is modelled using electronic synapse and neurons. They are connected in a pre-defined topology and provide some configurable units after fabrication. All the synapses and neurons have self learning modules and local non-volatile memories. Peripherals are sensors or prosthesis that interact with outside world. These modules are specialized electronic devices that are able to transform information to spikes, encoded in a way that our sensors and prosthesis do. All information flows as spikes. From this perspective, combination of memristor, or non-volatile memory, with traditional neuromorphic circuit is critical, since it will be impossible to store all information in one large memory. Nevertheless, it is too ambitious to realize the whole structure in a short period of time. After all, progress of science is made step by step.

VII.2 About Perceptron Learning

In chapter 6, perceptron learning is implemented in neuromorphic architecture. It shows that neuromorphic structure can distinguish between different patterns. The mechanism of this

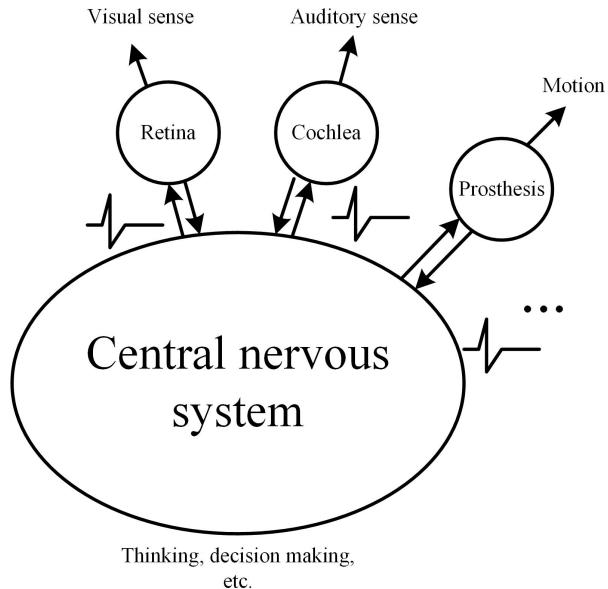


Figure 7–1 Complete neuromorphic architecture

phenomenon is as follows. When no teacher signal is supervising, or the frequency of teacher is low, synaptic input alone decides the firing of neuron. The inputs are modified with random noise, so the emerging of spike for these inputs is asynchronous. In STDP, when signals are asynchronous, even if they carry the same information (white or black pixel), they have negative effect to each other. In this scenario, all synaptic weights tend to be depressed. For black pixels, they have a high depressing speed, while for white pixels, they have a low depressing speed. When there is a teacher supervising the learning process, the output of neuron is dominated by the contribution of teacher. Since the frequency of teacher signal is so high that learning event happens continually, all synapses tend to potentiate. For black pixels, they have greater chance to potentiate than white pixels. As a result, when classifying two patterns, the synaptic differences eventually emerge at those pixels, whose values are different for the two patterns. In short, when implementing perceptron learning, valuable information is only presented in those changing bits. To increase the accuracy of classification, one still need to increase the resolution of an image.

VII.3 About Memristor Relationship to Neuromorphic Architecture

The emergence of memristor and non-volatile memory has significant meaning to the neuromorphic architecture. These localized memory can reduce the cost of booting, which drastically lower the power consumption. More importantly, since data can be fetched locally,

neuromorphic architecture can become distributed architectures. These architectures are highly customized for certain use. To conclude, memristor does not offer an opportunity to make a better general purpose computer, but a work oriented bio-inspired electronic architecture. In chapter 5, a memristor based neuromorphic architecture is built. Nowadays, since fabrication of memristor is expensive, and no free reliable data can be retrieved, exploring analog and dynamic characteristic of memristor will have limited meanings. On the other hand, when using memristors as digital memory cells that decide the on and off state of synaptic weight (V_w), the result will be more compatible with fluctuations. More importantly, since V_w presents in almost every synapse structure, the result is also compatible with different neuromorphic architecture. Figure 7-2 shows the relationship among synapse-neuron network, memristor crossbar array, and learning control module.

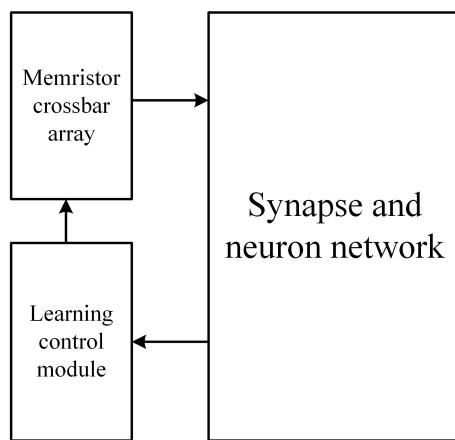


Figure 7-2 Memristor connections to weight updating module and synapse-neuron network

VII.4 Values of Neuromorphic Engineering

Although the ultimate goal of neuromorphic engineering is hard to realize, it can produce abundant by-products that can both make a profit and benefit our lives. Some possible applications are as follows:

- Process data in bio-electronics.

For bio-electronics, they interact with biological signals. Neuromorphic architecture can directly process these signals, without the use of a general propose processor.

- Pre-process data for sensors in Internet of Thing (IoT).

Sensors in Internet of Thing require low power consumption to maintain their operation. Neuromorphic architecture can extract useful information from raw data and only send

the valuable information, which can significantly reduce the power consumption.

VII.5 Future Work

In the future, researchers should conduct research in the following aspects:

- Explore accurate spike encoding schemes.
- Make standard synapse and neuron cells, and update them according to progress in neuroscience.
- Get access to a specific memristor fabrication process, and design customized circuit blocks for it.

Summary

Nowadays, digital processor becomes cheap and powerful. On a middle rank smart phone, we can easily calculate our science homework, listen to music, log in our email, and receiving messages from friends simultaneously. One can conclude that, the processing power of today's processor is way ahead of our daily need. However, when it comes to scientific research, power consumption of digital processor becomes expensive. In the field of neuromorphic engineering, researchers have tried to build neuromorphic architecture using commercial processor. Even though they managed to optimize their circuit structure and use low power processing cores, it is still impossible to realize the performance of a biological nervous system.

Emergence of memristor evokes a new generation of non-volatile memory. If memristor can be realized in silicon integrated circuit, not only does digital computer save the power for booting, but also it can lead to distributed processing systems. More importantly, through years of experiment, researchers have managed to mimic ourselves using sophisticated circuitry. The combination of these two findings will impel the building of powerful neuromorphic architecture.

In this thesis, a memristor based neuromorphic architecture is built. From circuit level, the transistor level circuit is built for every circuit block, and their layout has been made in XFAB $0.18\mu m$ technology. From chip level, a neuromorphic chip capable of learning is designed and taped-out. From architecture's level, I have discussed necessary building blocks for neuromorphic structure. From algorithm's level, I have implemented perceptron learning rule in a built neuromorphic architecture. Future work includes finding more accurate models for biological behavior, and designing standard cells for neuromorphic architecture.

Bibliography

- [1] Chua L O. Memristor-The missing circuit element[J]. IEEE Trans Circuit Theory, 1971, 18(5):507 - 519.
- [2] Chua L O, Komuro M, Matsumoto T. The double scroll family[J]. IEEE Transactions on Circuits & Systems, 1986, 33(11):1072-1118.
- [3] Makoto I, Chua L O. Memristor Oscillators[J]. International Journal of Bifurcation & Chaos, 2008, 18(11):3183-3206.
- [4] Bharathwaj M, Implementing Memristor Based Chaotic Circuits[J]. International Journal of Bifurcation & Chaos, 2010, 20(05):1002651-.
- [5] Bharathwaj M, Chua L O. Simplest Chaotic Circuit[J]. International Journal of Bifurcation & Chaos, 2010, 20(05):1567-1580.
- [6] Strukov D B, Snider G S, Stewart D R, *et al*. The missing memristor found.[J]. Nature, 2008, 453(7191):80-83.
- [7] Russo U, Ielmini D, Cagli C, *et al*. Self-Accelerated Thermal Dissolution Model for Reset Programming in Unipolar Resistive-Switching Memory (RRAM) Devices[J]. IEEE Transactions on Electron Devices, 2009, 56(2):193-200.
- [8] Hadiyawarman, Budiman F, Hernowo D G O, *et al*. Recent progress on fabrication of memristor and transistor-based neuromorphic devices for high signal processing speed with low power consumption[J]. Japanese Journal of Applied Physics, 2018, 57(3S2):03EA06.
- [9] Hong Y, Lian Y. A Memristor-Based Continuous-Time Digital FIR Filter for Biomedical Signal Processing[J]. IEEE Transactions on Circuits & Systems I Regular Papers, 2015, 62(5):1392-1401.
- [10] Jahromi M R, Shamsi J, Amirsoleimani A, *et al*. Ultra-low power Op-Amp design with memristor-based compensation[C]// Electrical and Computer Engineering. IEEE, 2017:1-4.
- [11] Hickmott T W. LowFrequency Negative Resistance in Thin Anodic Oxide Films[J]. Journal of Applied Physics, 1962, 33(9):2669-2682.

- [12] Lehtonen E, Poikonen J, Laiho M. Implication logic synthesis methods for memristors[J]. 2012, 57(1):2441-2444.
- [13] Haghiri S, Nemati A, Feizi S, *et al.* A memristor based binary multiplier[C]// Electrical and Computer Engineering. IEEE, 2017:1-4.
- [14] Mead C. Neuromorphic electronic systems[J]. Proceedings of the IEEE, 1990, 78(10):1629-1636.
- [15] Indiveri G, Linares-Barranco B, Legenstein R, *et al.* Integration of nanoscale memristor synapses in neuromorphic computing architectures[J]. Nanotechnology, 2013, 24(38):384010.
- [16] Linaresbarranco B, Serranogotarredona T. Memristance can explain Spike-Time-Dependent-Plasticity in Neural Synapses[J]. Nature Precedings, 2009.
- [17] Kim S, Choi S H, Lu W. Comprehensive Physical Model of Dynamic Resistive Switching in an Oxide Memristor[J]. Acs Nano, 2014, 8(3):2369-2376.
- [18] Waser R, Dittmann R, Staikov G, *et al.* RedoxBased Resistive Switching Memories – Nanoionic Mechanisms, Prospects, and Challenges[J]. Advanced Materials, 2009, 21(25-26):2632-2663.
- [19] Joglekar Y N, Wolf S J. The Elusive Memristor: Properties of Basic Electrical Circuits.[J]. European Journal of Physics, 2008, 30(4):661-675.
- [20] Bielek Z, Bielek D, Biolkova V. SPICE Model of Memristor with Nonlinear Dopant Drift[J]. Radioengineering, 2009, 18(2):210-214.
- [21] Laiho M, Lehtonen E, Russell A, *et al.* Memristive Synapses Are Becoming Reality[J]. 2010.
- [22] Kvatinsky S, Friedman E G, Kolodny A, *et al.* TEAM: ThrEshold Adaptive Memristor Model[J]. IEEE Transactions on Circuits & Systems I Regular Papers, 2013, 60(1):211-221.
- [23] Burr G W, Narayanan P, Shelby R M, *et al.* Large-scale neural networks implemented with non-volatile memory as the synaptic weight element: Comparative performance analysis (accuracy, speed, and power)[C]// IEEE International Electron Devices Meeting. IEEE, 2015:4.4.1-4.4.4.

- [24] Poznanski A, Wolf L. CNN-N-Gram for Handwriting Word Recognition[C]// Computer Vision and Pattern Recognition. IEEE, 2016:2305-2314.
- [25] Khan S M U, Manzoor J S, Lee S U J. Predicting Student Blood Pressure by Support Vector Machine Using Facebook[C]// Services. IEEE, 2014:486-492.
- [26] Mead C. Analog VLSI and neural systems[J]. 1989:257-278.
- [27] Indiveri G. A low-power adaptive integrate-and-fire neuron circuit[C]// International Symposium on Circuits and Systems. IEEE, 2003:IV-820-IV-823 vol.4.
- [28] Indiveri G, Linaresbarranco B, Hamilton T J, *et al.* Neuromorphic Silicon Neuron Circuits[J]. Frontiers in Neuroscience, 2011, 5(5):73.
- [29] Dutta S, Kumar V, Shukla A, *et al.* Leaky Integrate and Fire Neuron by Charge-Discharge Dynamics in Floating-Body MOSFET.[J]. Scientific Reports, 2017, 7(1).
- [30] Painkras E, Plana L A, Garside J, *et al.* SpiNNaker: A 1-W 18-Core System-on-Chip for Massively-Parallel Neural Network Simulation[J]. IEEE Journal of Solid-State Circuits, 2013, 48(8):1943-1953.
- [31] Indiveri G, Linares-Barranco B, Legenstein R, *et al.* Integration of nanoscale memristor synapses in neuromorphic computing architectures[J]. Nanotechnology, 2013, 24(38):384010.
- [32] Bear M F. Neuroscience: exploring the brain, 4th ed.[M]// Neuroscience : exploring the brain. Lippincott Williams & Wilkins, 2010:110.
- [33] Massey P V, Bashir Z I. Long-term depression: multiple forms and implications for brain function.[J]. Trends in Neurosciences, 2007, 30(4):176-184.
- [34] Bi G Q, Poo M M. Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type.[J]. Journal of Neuroscience, 1998, 18(24):10464-10472.
- [35] Jo S H, Chang T, Ebong I, *et al.* Nanoscale Memristor Device as Synapse in Neuromorphic Systems[J]. Nano Letters, 2010, 10(4):1297-1301.
- [36] Jiang R, Ma P, Han Z, *et al.* Habituation/Fatigue behavior of a synapse memristor based on IGZO-HfO₂ thin film[J]. Sci Rep, 2017, 7(1):9354.

- [37] Kim H, Sah M P, Yang C, *et al.* Memristor Bridge Synapses[J]. Proceedings of the IEEE, 2012, 100(6):2061-2070.
- [38] Zamarreñoramos C, Camuñasmesa L A, Pérezcarrasco J A, *et al.* On Spike-Timing-Dependent-Plasticity, Memristive Devices, and Building a Self-Learning Visual Cortex[J]. Frontiers in Neuroscience, 2011, 5(26):26.
- [39] Luczak A, Mcnaughton B L, Harris K D. Packet-based communication in the cortex[J]. Nature Reviews Neuroscience, 2015, 16(12):745-55.

Acknowledgements

I express my sincere gratitude to the following people.

Thanks to Prof. Wang who conferred me this topic.

Thanks to Dr. Khalil who helped me with technical problems in circuit design.

Thanks to my parents and friends, who helped me through when I was depressed.

Thanks to Prof. Indiveri, who guided me through emails, and encouraged me when I was confused.

Thanks to all the American colleges that rejected me. It is your denials that made me concentrate on my undergraduate thesis.

Thanks to Erroll Garner, whose music accompanied me through countless days and nights.