Spike Processing Circuit Design for Neuromorphic Computing

Chenyuan Zhao

Dissertation submitted to the faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

In

Electrical Engineering

Yang Yi, Chair
Dong. S. Ha
A. Lynn Abbott
Haibo Zeng
Xianming Bai

August, 8th 2019

Blacksburg, VA

Keywords: Neuromorphic computing, neuron, LIF, temporal encoding, spiking neural network,
Inter-spike interval, encoder, decoder, STDP, latency

Spike Processing Circuit Design for Neuromorphic Computing

Chenyuan Zhao

Abstract

Von Neumann Bottleneck, which refers to the limited throughput between the CPU and memory, has already become the major factor hindering the technical advances of computing systems. In recent years, neuromorphic systems started to gain the increasing attentions as compact and energy-efficient computing platforms. Spike based-neuromorphic computing systems require high performance and low power neural encoder and decoder to emulate the spiking behavior of neurons. These two spike-analog signals converting interface determine the whole spiking neuromorphic computing system's performance, especially the highest performance. Many state-of-the-art neuromorphic systems typically operates in the frequency range between 10^0KHz and 10^2KHz due to the limitation of encoding/decoding speed. In this dissertation, all these popular encoding and decoding schemes, i.e. rate encoding, latency encoding, ISI encoding, together with related hardware implementations have been discussed and analyzed. The contributions included in this dissertation can be classified into three main parts: neuron improvement, three kinds of ISI encoder design, two types of ISI decoder design. Two-path leakage LIF neuron has been fabricated and modular design methodology is invented. Three kinds of ISI encoding schemes including parallel signal encoding, full signal iteration encoding, and partial signal encoding are discussed. The first two types ISI encoders have been fabricated successfully and the last ISI encoder will be taped out by the end of 2019. Two types of ISI decoders adopted different techniques which are sample-and-hold based mixed signal design and spike-timing-dependent-plasticity (STDP) based analog design respectively. Both these two ISI encoders have been evaluated through post-layout simulations successfully. The STDP based ISI encoder will be taped out by the end of 2019. A test bench based on correlation inspection has been built to evaluate the information recovery capability of the proposed spiking processing link.

Spike Processing Circuit Design for Neuromorphic Computing

Chenyuan Zhao

General Audience Abstract

Neuromorphic computing is a kind of specific electronic system that could mimic biological bodies' behavior. In most cases, neuromorphic computing system is built with analog circuits which have benefits in power efficient and low thermal radiation. Among neuromorphic computing system, one of the most important components is the signal processing interface, i.e. encoder/decoder. To increase the whole system's performance, novel encoders and decoders have been proposed in this dissertation.

In this dissertation, three kinds of temporal encoders, one rate encoder, one latency encoder, one temporal decoder, and one general spike decoder have been proposed. These designs could be combined together to build high efficient spike-based data link which guarantee the processing performance of whole neuromorphic computing system.

Dedication

I would like to thank my advisor, Dr. Yang Yi, who has guided me and give me suggestions with her knowledge and professionalism. I would like to thank my lab mates and all Multifunctional Integrated Circuits and Systems (MICS) groups members. They helped me a lot during my Ph.D. studying period.

I would like to thank my parents and my girlfriend. They always support me during the graduate life. I would also like to thank all of my friends, they are my families at Blacksburg.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ISI	Inter-Spike Interval
STDP	Spike Timing Dependent Plasticity
I&F	Integrate and Fire
LIF	Leaky Integrate and Fire
HH	Hodgkin Huxley
CMOS	Complementary Metal Oxide Semiconductor
GF	Global Foundry
CLK	Clock
NMOS	N-Type Metal Oxide Semiconductor
PMOS	N-Type Metal Oxide Semiconductor
ADC	Analog to Digital Converter
BCD	Binary Coded Decimal
PSI	Partial Signal Iteration
SH	Sample & Hold
SOP	Sum of Product
De.	Discrete Extractor
LTD	Long Term Depression

Chapter 1. Introduction

1.1 Motivation

Traditional Von Neumann architecture-based computer [1-4] has been developed for more than 70 years [5, 6]. Now, such kind of architecture has already reached the performance ceiling [7, 8], e.g. operation frequency, power consumption, and etc. [9-12]. Furthermore, the traditional computer is not good at handle problems including but not limited to image recognition, pattern classification, sound recognition, etc. [13-16]. Neuromorphic computing system [17] is a brand-new computing system which adopting both traditional electronic devices and emerging materials to mimic biological's behavior, especially biological brain's behavior [18]. Comparing with traditional computer, neuromorphic computing system could achieve aforementioned recognition or classification tasks quickly and efficiently[19]. TrueNorth, which is one of the famous neuromorphic computing systems, can implement pattern tracking with only $20mW/cm^2$ power density, while traditional central processing unit chip need $501001mW/cm^2$ power density [8]. This value could be even lower if emerging devices[20], such as memristor[21-27], spintronic device [23, 28, 29], etc., are adopted.

Inspired by biological brain's operation principles[30, 31], the high-performance neuromorphic computing system[32], which has made great success, is gaining increased attention. More and more state-of-arts for neuromorphic computing designs are presented with different topologies [33-37]. In a biological neural system, the input and output signals of a neuron are all in spike format [38-41]. Therefore, how to transforming analog/digital signal into spike signal is the first problem need to be solved [42, 43]. In past decades, electronic circuit models of the biological neuron have been investigated [44-46]. Many electronic circuit models of the biological neuron [10, 19], which could be used as the basic units of the neuromorphic computing, make spike-based data processing link become possible. Among these hardware implementations, rate spike code [47, 48] is the simplest code format that has been adopted by most implementations [49, 50]. The rest implementations are based on latency spike code [51-53], which has been found in biological vision apparatus. However, scientists have found that in real biological body, there is another code format, which called temporal spike code, existing [54, 55]. Such kind of temporal

spike code is more complex than rate spike or latency spike code, which making it hard to be implemented in hardware [56-59]. Comparing with rate spike code and latency spike code, temporal spike code has higher information density [60]. Furthermore, it has potential to represent more than one dimensional information [61]. Therefore, to design and implement temporal encoder/decoder with electronic technology has become very important in neuromorphic computing field[62].

1.2 Contributions

During my Ph.D. study period, I have made deep researches on neural signal processing field which specifying on spiking format signal encoding and decoding. My research works include:

1. LIF neuron design and implementation
   I have designed an analog LIF neuron with double paths leakage current that could achieve frequency adaptation. This neuron has been fabricated successfully under GF180nm CMOS technology.
2. Parallel ISI encoder design and implementation
   This is the first-generation ISI encoder that could transform analog signal into ISI spike train. To the best of my knowledge, the proposed parallel ISI encoder is the first kind of ISI spike encoder. This ISI encoder has been fabricated successfully under GF180nm CMOS technology.
3. Full signal iteration ISI encoder design and implementation
   The second-generation ISI encoder is based on signal iteration scheme. This is a kind of asynchronous ISI encoder that consumes less power than parallel ISI encoder. It could generate 2^{N-1} spikes with just N neurons which could reduce design area greatly. This ISI encoder has been fabricated successfully under GF180nm CMOS technology.
4. Sample & Hold-based ISI decoder
   To the best of my knowledge, this ISI decoder is the first kind of ISI decoder that could transform ISI spike train back to voltage level signal. This ISI decoder is based on mixed-signal design methodology.
5. STDP based ISI decoder

This is the second-generation ISI decoder that could achieve multi-scale decoding output. This ISI decoder is based on STDP theory which has higher scaling up ability and smaller design area. This ISI decoder has been designed under GF180nm CMOS technology, which will be fabricated by the end of 2019.

6. Partial signal iteration ISI encoder

   This is the third-generation ISI encoder. The key innovation of the ISI encoder is reducing iteration signals, i.e. only current signal is used to make iteration, to achieve ISI encoding without loss accuracy. Furthermore, the power consumption of this encoder is much less than second-generation ISI encoder. I have finished this ISI encoder's chip design and it would be fabricated under GF180nm CMOS technology by the end of 2019.

Chapter 2. Neuron

2.1 General Spike Neuron

The fundamental unit of biological bodies is called neuron which plays one of the most important role in signal processing and transmission[63]. Though there are many different kind of neurons, the common components of neurons are including nucleus (cell body), synapse, and axon [64]. In Fig. 2.1.1, a typical biological neuron is illustrated.

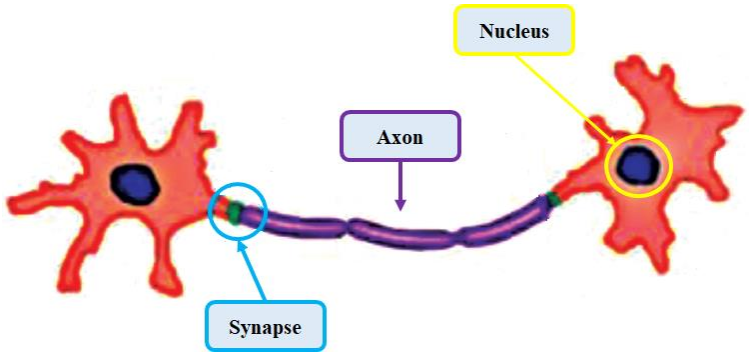


Fig. 2.1. 1 Typical biological neuron structure

In many state of arts, some minor components, such as Golgi apparatus, Endoplasmic reticulum, dendrite, and etc., are also included [65-68]. However, in neuromorphic computing design, only the proposed three components have been taken into account. Comparing with traditional electronic system, these three components could be analogous to processor (nucleus), memory (synapse), and transmission line (axon). In artificial neural network system, the neuron is defined as spike generator [69-72]. In this case, we can say neuron is equal to nucleus (in this dissertation, neuron only has the ability to generate spike[37], synapse and axon will be independent components). The signal flow chart under this assumption could be illustrated in Fig. 2.1.2.

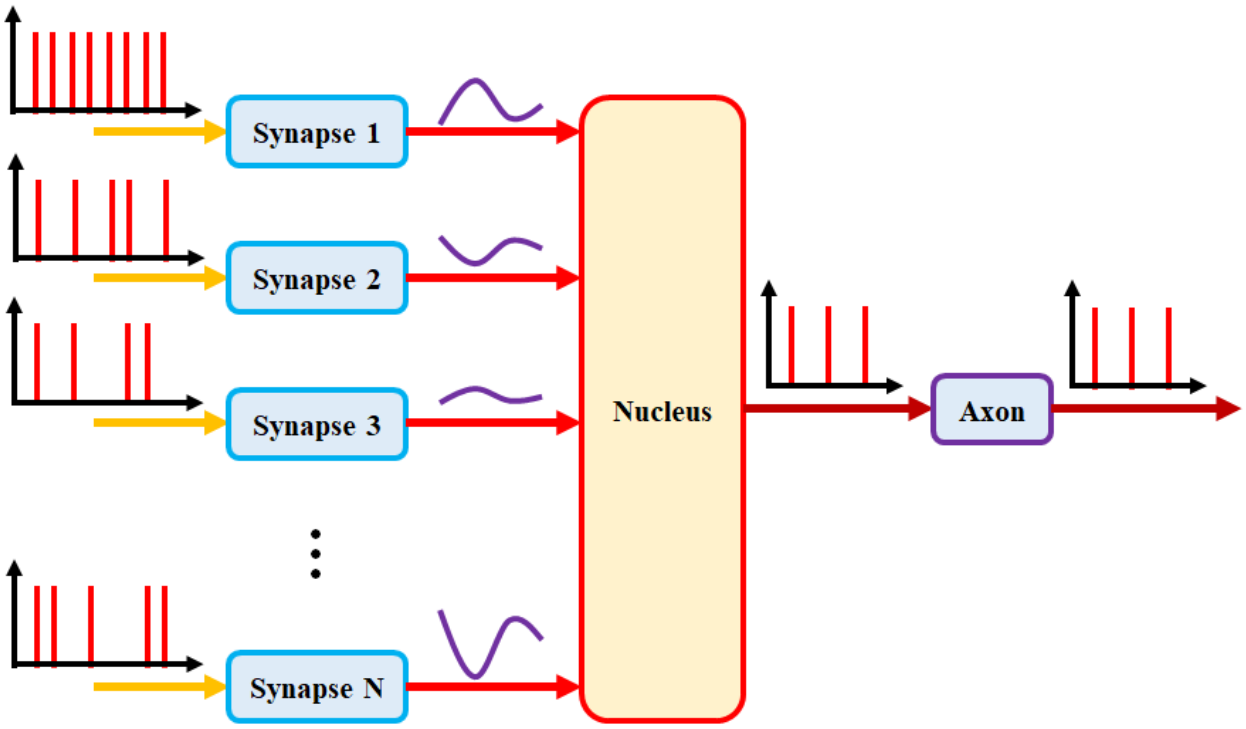


Fig. 2.1. 2 Signal flow chart with proposed neuron structure

As shown in Fig. 2.1.2, the input and output of a neuron module are all spike signals. Several spike signals apply on synapses and then combine together on one nucleus. The nucleus will process all these spikes and generate one path spike train out. This output spike train would transmit to other neuron through axon without degeneration [73]. Within these three modules, synapse and nucleus would change the information's format, while axon does not make any change of these spike trains. In fact, there existing delays during spike transmission, which may play essential affection on spike generation of next stage neurons. Therefore, in neuromorphic computing designs, axon would also have delay function to make all spike trains transmitting synchronously.

2.2 Neuron Model

2.2.1 Integrated and Fire Neuron

The most famous neuron model is called integrated and fire (I&F) neuron model [50, 74, 75]. The core part of this model is an integrator which could accumulate stimuluses, i.e. input signals, and generate spike when the accumulation level exceeds threshold value. After a spike generated,

a refractory signal would be applied on the integrator and latch it down. The mathematic expression of I&F model could be expressed as

$$\begin{cases} Firing\ Spike, & V_{mem} \geq V_{th} \\ 0, & V_{mem} < V_{th} \\ \dfrac{dV_{mem}}{dt} = I_{in} \end{cases}, \tag{2.2.1}$$

where C_{mem}, V_{th}, I_{in} represent membrane capacitance, threshold voltage, and input signals, respectively. In this model, the integrator is made with membrane. By applying this model, a CMOS based analog circuit is illustrated in Fig. 2.2.1.

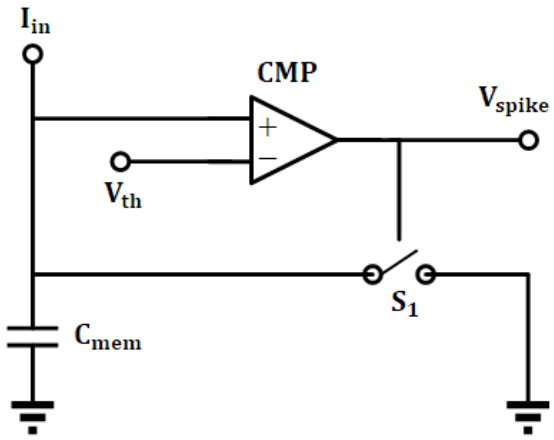


Fig. 2.2. 1 IF model-based circuit

As shown in Fig. 2.2.1, there are three main components in this circuit including membrane capacitor (C_{mem}), comparator (CMP), and trigger (S_1). In this circuit, when the input signal I_{in} applying on membrane capacitor, the voltage on C_{mem}, i.e. V_{th}, would increase linearly. Both the threshold voltage V_{th} and the membrane voltage V_{mem} would apply on comparator CMP, and a positive level signal would generate when $V_{mem} \geq V_{th}$. This positive signal would turn on the trigger S_1 which would pull V_{mem} to ground. Since capacitor's discharging speed is much fast, the output signal V_{spike} would appear as an extreme narrow pulse which would be used as spike signal in neuromorphic computing design.

2.2.2 Leaky Integrated and Fire (LIF) Neuron

As discussed in section 2.2.1, I&F model could generate spike successfully. However, there are several drawbacks that prevent I&F model to be a general purpose neuron model [76-78]. First, if the input signal is continuously (in practice, input signal is always continuously), the membrane capacitor may not be able to discharge. In other words, the membrane voltage may be locked to a specific value ("dead zone"), which will hold output voltage V_{spike} in a constant value [79]. Furthermore, even if I&F neuron avoids aforementioned "dead zone", the output spike train could not be used by following modules, e.g. neural network, decoder, etc., directly. Additional processing circuits are required to split this spike train into several spike clusters.

In order to overcome these two drawbacks, LIF neuron is introduced [80]. Comparing with I&F neuron, additional leak current and external control clock are adopted. In LIF, leak current could prevent "dead zone" effect, and external clock can let LIF generating ordered spike trains. The mathematic expression of LIF neuron is presented in equation. 2.2.2.

$$\begin{cases} Firing\ Spike, & V_{mem} \geq V_{th} \\ V_{rest}, & V_{mem} < V_{th} \\ \dfrac{dV_{mem}}{dt} = I_{in} + I_{leakD} + I_{leakU} \end{cases}, \tag{2.2.2}$$

where V_{rest} is the rest voltage when LIF neuron does not fire spike, I_{leakD} and I_{leakU} are upper path and down path leak currents respectively. In the LIF neuron, if the membrane capacitor reaches "dead zone", the down path / upper path leak current will pull/push the membrane voltage down/up to break the "dead zone" balance. In practice, it is possible to apply more than two paths leak currents to implement different spike train patterns. A typical LIF neuron circuit with two path leak currents is illustrated in Fig. 2.2.2.

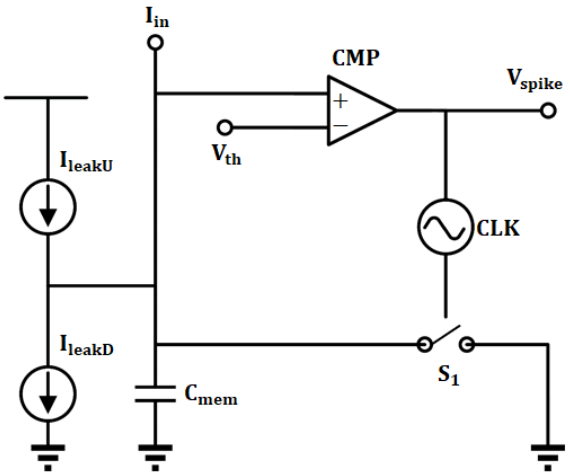


Fig. 2.2. 2 Two-path leak currents LIF neuron circuit

As shown in Fig. 2.2.2, there are two paths leak currents to prevent LIF neuron from being into "dead zone". The trigger S_1 is controlled not only by output spike signal but also by external clock signal. A special situation is turn off CLK, which will make LIF neuron generate out-of-order spike train as I&F neuron did. Furthermore, leak currents would have resistance effect that making the membrane capacitor charging behavior as an exponential function, which plays an important role in building different spike encoders. Details would be discussed in chapter 3.

2.2.3 High Order Neuron

Both I&F and LIF neuron have been widely used in neuromorphic computing design field due to the simplicity and reliability. However, in neuron model research field, high order neuron models, e.g. Hodgkin-Huxley (HH) model [81], are more important models that should be taken into account. Furthermore, such kind of high order neuron model could represent more properties that biological neuron should have, which is helpful to be used to find inner relationship of biological neural systems.

Typically, HH is the most successful high order neuron model which has multi-model spiking function, spike frequency adaptation function, threshold variability function, bi-stability function, and etc. [50]. In general, HH model is a kind of neuron model that could mimic biological neuron's behavior in high completion.

In HH model, the currents are divided into sodium-ion current, potassium-ion current and leakage current [50]. There three currents can be represented as I_{Na}, I_P, and I_{leak} respectively. There are more than 9 parameters which may change HH model's behavior. Without generality, the mathematic expression of HH model is shown in equation 2.2.3.

$$I = C_m \frac{dV_m}{dt} + I_{Na} + I_P + I_{leak}, \qquad 2.2.3$$

where I is the summation current and V_m represents membrane voltage. Within HH model, the leakage current is determined by several parameters. A typical HH circuit is illustrated in Fig. 2.2.3.

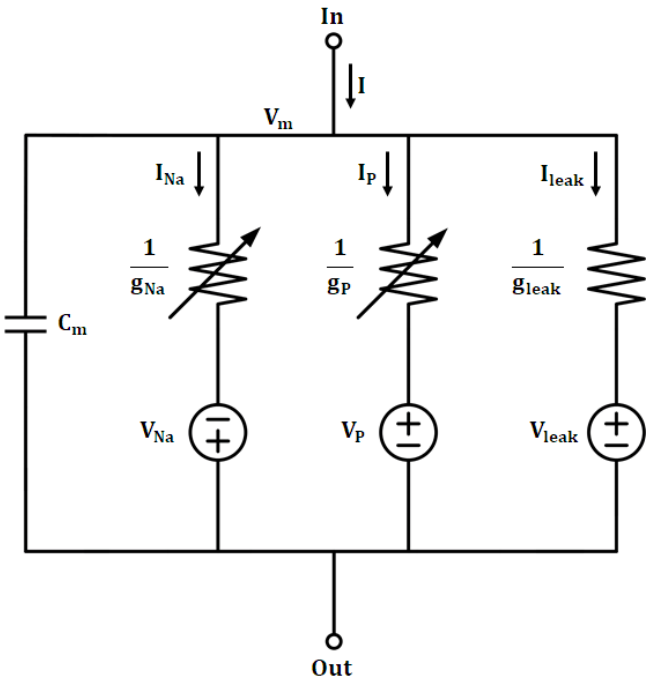


Fig. 2.2. 3 HH neuron circuit

As shown in Fig. 2.2.3, I_{Na} is determined by conductance g_{Na}, voltage potential V_{Na}; I_P is determined by conductance g_P and voltage potential V_P; I_{leak} is determined by conductance g_{leak} and voltage potential V_{leak}, respectively. These relationships can be expressed in equation 2.2.4.

$$\begin{cases} I_{Na} = (V_m + V_{Na})g_{Na} \\ I_{leak} = (V_m - V_{leak})g_{leak} \\ I_P = (V_m - V_P)g_P \\ g_P = \overline{g_P}n^4 \\ \dfrac{dn}{dt} = \alpha(1-n) - \beta n \end{cases}, \quad 2.2.4$$

where $\overline{g_P}$ is the average value of g_P, n is a dimensionless state variable, α and β are voltage dependent rate constants. Though the HH neuron model has been simplified, it is still too complex to be a fundamental unit to construct neuromorphic computing systems [82].

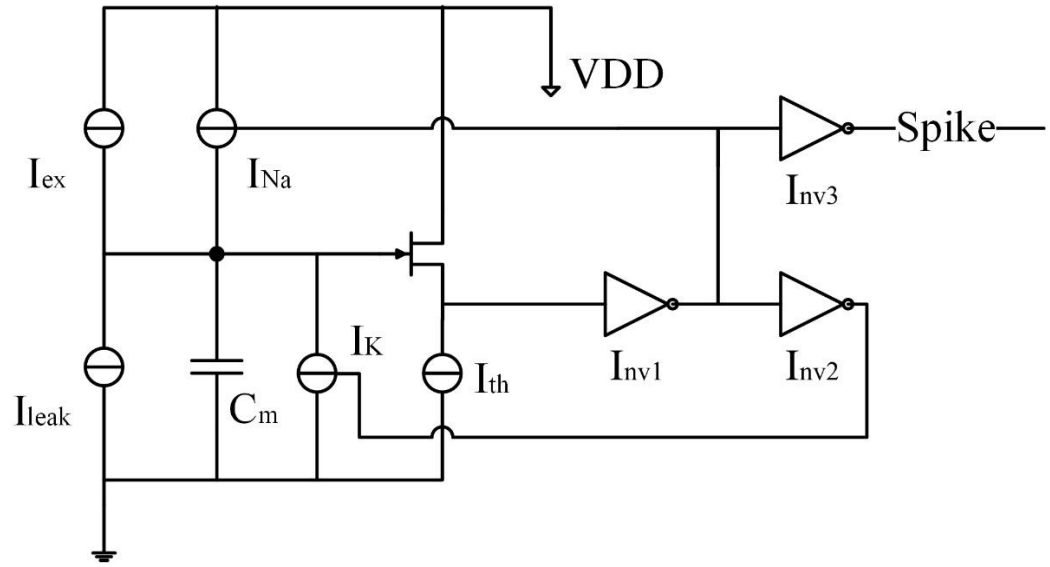
2.3 LIF Neuron Circuits

Many state-of-arts simply use lite LIF neuron circuit to serve as their design unit [83, 84]. One resistor is adopted to substitute leak currents. In this case, it is not possible to reshape the membrane potential's charging behavior. Furthermore, such kind of design does not have any anti-noise ability, or frequency adaptation ability, which is extra important when the neural network is scaling up.

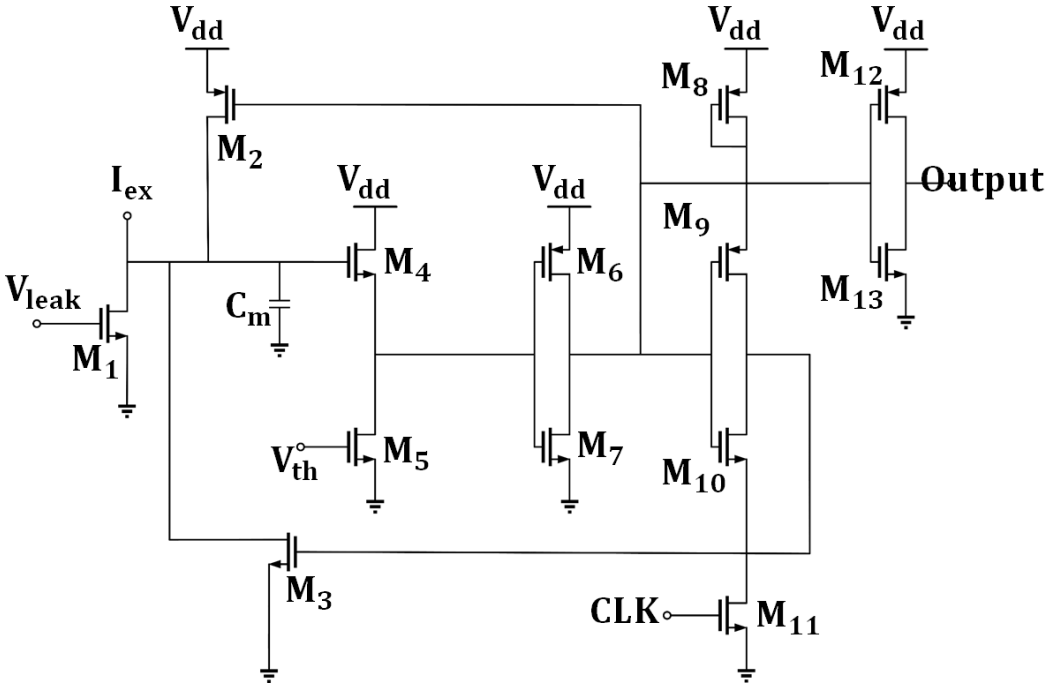
To overcome these issues, I have designed two versions of LIF neurons (subthreshold [85] version and feedback version) which has frequency adaptation function. Both current mode and voltage mode circuit have been designed for different applications. The voltage mode can be extended into multi-stage LIF neuron (MLN) that is the essential unit to build-up latency encoder and ISI encoder.

2.3.1 Subthreshold LIF Neuron

The key innovation of LIF neuron, comparing with IF neuron, is two paths leakage currents. Though both voltage mode and current mode can work correct, current mode would make the LIF neuron more compact. Furthermore, current mode would consume less power than voltage mode due to less transistor size. The simplified diagram of the proposed LIF neuron and related circuit schematic are illustrated in Fig. 2.3.1.



(a)



(b)

Fig. 2.3. 1 LIF neuron structure: (a) signal flow chart; (b) circuit schematic

As shown in Fig. 2.3.1(a), two paths leakage current have been marked with I_{Na} and I_K respectively. In Fig. 2.3.1(b), input signal is applied on I_{ex} terminal, V_{leak} is down-path leakage current control signal, CLK is sampling clock signal, and V_{th} is the threshold voltage of NMOS

transistor. The upper-path leakage current is generated by transistor M_2 which is controlled by feedback signal. Without generality, the down-path leakage current is required to be considered. In this case, the leakage current could be expressed as [86]

$$I_{leak} = I_0 e^{V_{gs}\alpha}, \tag{2.3.1}$$

where α is determined by physical process, i.e. [87], electron charge [88], etc., and I_o is the constant current value that transistor working under saturation region. The V_{gs} is the voltage potential between gate terminal and source terminal. In equation 3.4.6, the V_{gs} is less than threshold voltage, which is the pre-requisition for subthreshold operation point. Without generality, the equation 3.4.6 could be expanded with Taylor's series when it works in deep subthreshold region as

$$\begin{aligned} I_{leak} &\cong I_0\left(V_{gs}\alpha e^{V_{gs}\alpha} + \alpha e^{V_{gs}\alpha} + \alpha^2 {V_{gs}}^2 e^{V_{gs}\alpha}\right)|_{V_{gs}=0} \\ &= I_0\alpha. \end{aligned} \tag{2.3.2}$$

The total input signal I_{in} could be the sum of input current and excitation current. The membrane voltage would increase when I_{in} applied on C_m. After the membrane voltage reaches a specific voltage, i.e. membrane threshold voltage, transistor M4 would come into saturation region, which will drag current from V_{dd} to make M5 work. Since V_{th} is constant, the voltage on drain terminal of M5 would increase companion with the increasing of I_5. The proposed processing could be expressed by equation 2.3.3

$$I_5 = K_5(V_{th} - V_{tn5})^2(1 + \lambda V_{ds5}), \tag{2.3.3}$$

where K_5 is determined by CMOS process and transistor physic size, V_{tn5} is NMOS transistor intrinsic threshold voltage, λ is channel length modulation coefficient, and V_{ds5} is the voltage on transistor M5 (in this case drain-source potential is equal to drain terminal voltage). It is clear that V_{ds5} is related to I_5, and these following two inverters, i.e. M6-M7, M12-M13, would generate spike if V_{ds5} exceeds the membrane threshold voltage.

Transistor M3 would generate discharging current that would pull the membrane voltage back to rest potential, or refractory period. The signal generate by M6-M7 would also feedback to

transistor M2 which would generate upper-path leakage current that make the LIF neuron has frequency adaptation ability.

2.3.2 Multi-Stage LIF Neuron

In section 2.3.1, single stage LIF neuron with subthreshold-region based leakage current is designed. However, extra control signal, i.e. V_{leak} is required to control the leakage current. Furthermore, one-stage LIF's scaling up capability (increase the latency period) is restricted by the membrane capacitance's value. To improve these performances, I have proposed an adaptive leaky current technique to design the neuron circuit with abilities of the frequency adaptation and the latency generation. It is widely accepted that the I&F model could be expressed as [82]

$$\gamma \dot{v} = I + \alpha - \beta v, (v > V_{threshold} \implies v \leftarrow \delta), \tag{2.3.4}$$

where v is the membrane potential, I is the excitation current, α, β, and γ are constant parameters, $V_{threshold}$ is the threshold voltage, and δ is the peak value of an output spike. In this paper, I have proposed an MLN neuron, which could be expressed as

$$\begin{cases} \beta_1 \dot{v}_m = I_{ex} - \alpha_1 v_m \\ \beta_2 \dot{v}_a = I_a - G \\ I_a = \mathrm{H}(v_m, v_a) \end{cases}, \tag{2.3.5}$$

where v_m is the membrane potential, I_{ex} is the excitation current, $\alpha_1, \alpha_2, \beta_1, \beta_2$ are design parameters, v_a is the assistant voltage, I_a is the assistant excitation current, G is a nonlinear feedback function that is controlled by the output, and $\mathrm{H}(*)$ is a high-order nonlinear function. In the proposed MLN neuron, two differential equations presented in equation (2) ensure that the designed neuron has the ability of the latency generation, and the nonlinear feedback function G provides the ability of the frequency adaptation.

The multi-stage neuron circuit is designed through the proposed MLN model. In equation (2.3.5), two differential equations could be implemented with a two-stage I&F circuit. The adaptation function is based on the feedback loop, which could be achieved by the feedback current mirror structure, as illustrated in Fig. 2.3.2.

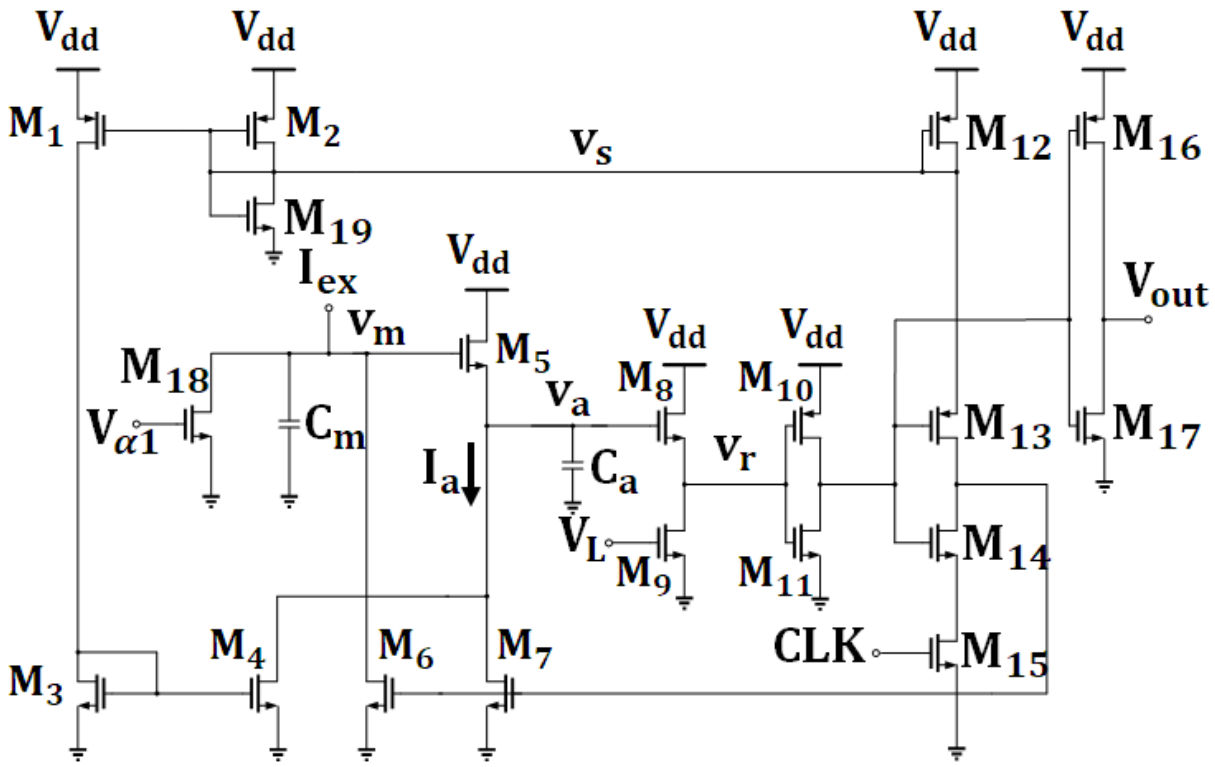


Fig. 2.3. 2 MLN neuron circuit schematic

As shown in Fig. 2.3.2, two capacitors, C_m and C_a, serve as design parameters of β_1 and β_2, respectively, as presented in equation (2.3.5). In the first stage, constructed based on C_m, the *bias* voltage V_{α_1} applied on the transistor M_{18} generates a leakage current that is represented as $\alpha_1 v_m$ in equation (2.3.5). By applying a $V_{\alpha_1} < V_{thn} + nkT/q$ constrained condition, the drain current on M_{18} could be considered as a constant value, which simplified the design equation. The adaptation ability is achieved in the second stage, which introduces a feedback signal that is generated by M_{12}. Assuming that the gate voltage on M_{12} is v_s, the current on M_1 can be determined by

$$I_{M1} = (W_1/L_1) I_o \exp(q(V_{dd} - v_s)/(nkT)), \tag{2.3.6}$$

where W_1 and L_1 are the channel width and length respectively of M_1, I_o is determined by the physical process, k is the Boltzmann constant, q is the electron-volt, T is the temperature, n is the subthreshold slop factor, and V_{dd} is the supply voltage. Since the current on M_1 and M_3 has the same value, the voltage on the gate terminal of M_4 could be expressed as

$$V_{g4} = (nkT/q) \ln\left(\frac{W_1 L_3}{W_3 L_1}\right) + V_{dd} - v_s. \tag{2.3.7}$$

Thereby, the current on M_4 could be written as

$$I_{M4} = W_4/L_4\, I_o e^{\ln\left(\frac{W_1 L_3}{W_3 L_1}\right)+\left(\frac{nkT}{q}\right)(V_{dd}-v_s)}. \quad (2.3.8)$$

The leaky current I_{M4} is only controlled by the feedback voltage v_s. From equation (2.3.5), the G could be implemented with equation (2.3.8). In Fig. 2.3.2, when a positive vibration $\Delta i \ll I_a$, the v_a reaches the threshold voltage more rapidly; thereby, the firing rate of the spike increases. However, such positive vibration reduces v_s, which will lead I_{M4} increases. In an ideal case, the value of $I_a - I_{M4}$ will not be interfered by Δi. The same firing mechanism could be achieved when negative vibration occurs.

The ability of the latency spike generation is one of the most important characteristics in the proposed MLN neuron design. In the initial status, the V_{gs5} of M_5 is smaller than V_{thn}, which will keep M_5 in the subthreshold region. Therefore, the current on M_5 is equivalent to I_{M4}, which will not charge up C_a. After v_m is increased to lead M_5 into the saturation region, v_a increases whereby C_a is charged up. This charging/discharging process can be achieved on M_8. Since the maximum potential of v_m, v_a and v_r have the relation of $v_m \approx 2v_a \approx 4v_r$, by adopting the charging behavior on a capacitor, it consumes 36.1% of the total integration time to store up 90% of the charge density across the capacitor, while the rest integration time is used to further store up the charge density. In most standard CMOS technologies [89], the threshold voltage is around $0.25V_{dd}$. In other words, it requires 63% of an integration time to generate a spike, and this period could be used as the latency in the construction of a neural network. Moreover, by tuning V_L, various latency values could be achieved.

2.4 Results Analysis

In this section, both single-stage LIF neuron and multi-stage neuron simulations and experiments results would be discussed.

2.4.1 Single LIF Neuron Results

For the single-stage LIF neuron, firing frequency, refractory period, and integrating period are these three important parameters that need to be considered. In Fig. 2.3.3, input signals with

different amplitudes have been applied on LIF neuron. The membrane potential and output spike signals are presented.

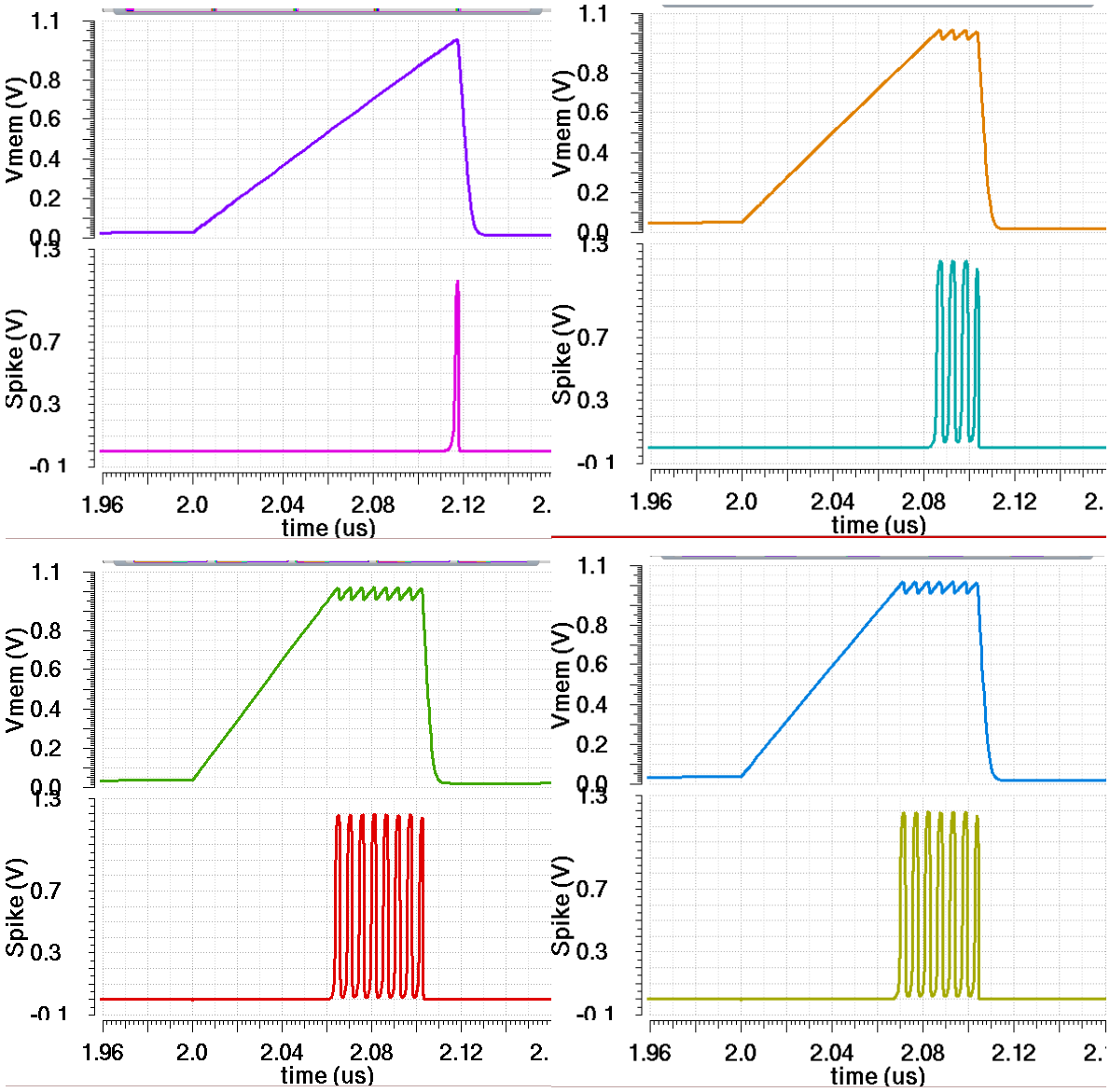


Fig. 2.4. 1 Different firing frequencies

As shown in Fig. 2.4.1, different amplitude signal would lead to different firing spike patterns, i.e. different spike number. In this design, the larger the amplitude would lead to higher firing rate after the external clock is determined.

Refractory period is also a significant parameter of LIF neuron. In the proposed design, the refractory period is determined by the threshold voltage. The higher the threshold voltage has, the shorter the refractory period would be.

The final parameter is integrating period. In general, membrane capacitor is fixed after finish LIF neuron design. So, the only thing that could change the integrating period is the amplitude of the input signal. The simulation result is shown in Fig. 2.4.2.

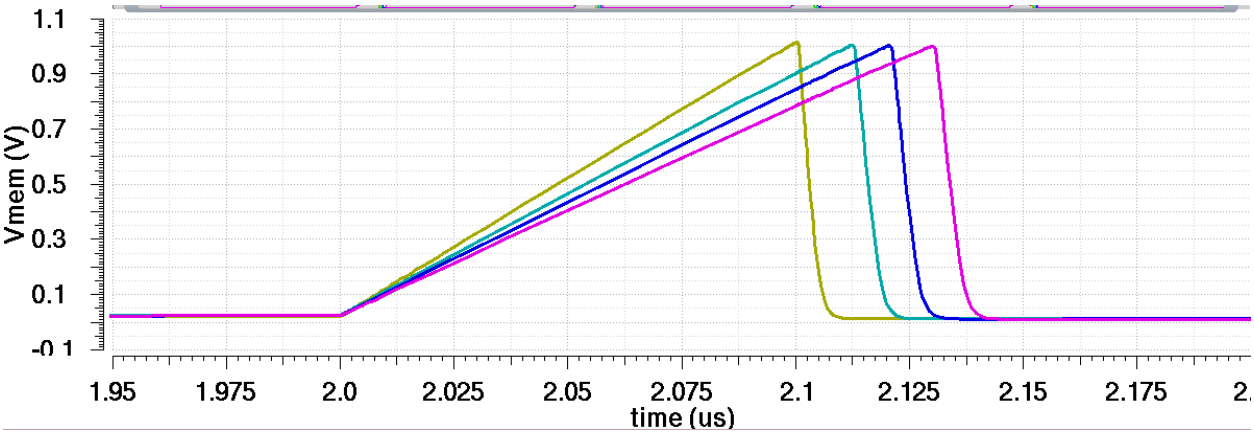


Fig. 2.4. 2 Different integrating periods

2.4.1 Multi-Stage LIF Neuron Results

Without generality, the MLN neuron could work in both current mode and voltage. The general construct of MLN neuron is illustrated in Fig. 2.4.3.

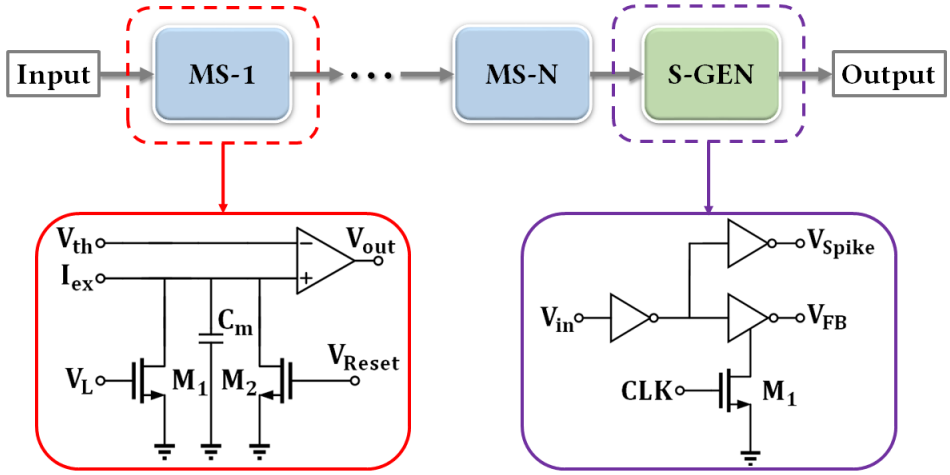


Fig. 2.4. 3 MLN neuron structure

As shown in Fig. 2.4.3, N-stage RC nodes serial connected to each other before reaching the S-GEN module. The S-GEN stage has the function to generate spike signal and reset signal that would be used as output signal and reset signal, respectively. Each RC node could be set back to

rest status by receiving reset signal. For this MLN neuron, the most important parameter is its latency code generating ability, i.e. generating latency spike smoothly.

The simulation result for each stage's latency period and final spike is illustrated in Fig. 2.4.4.

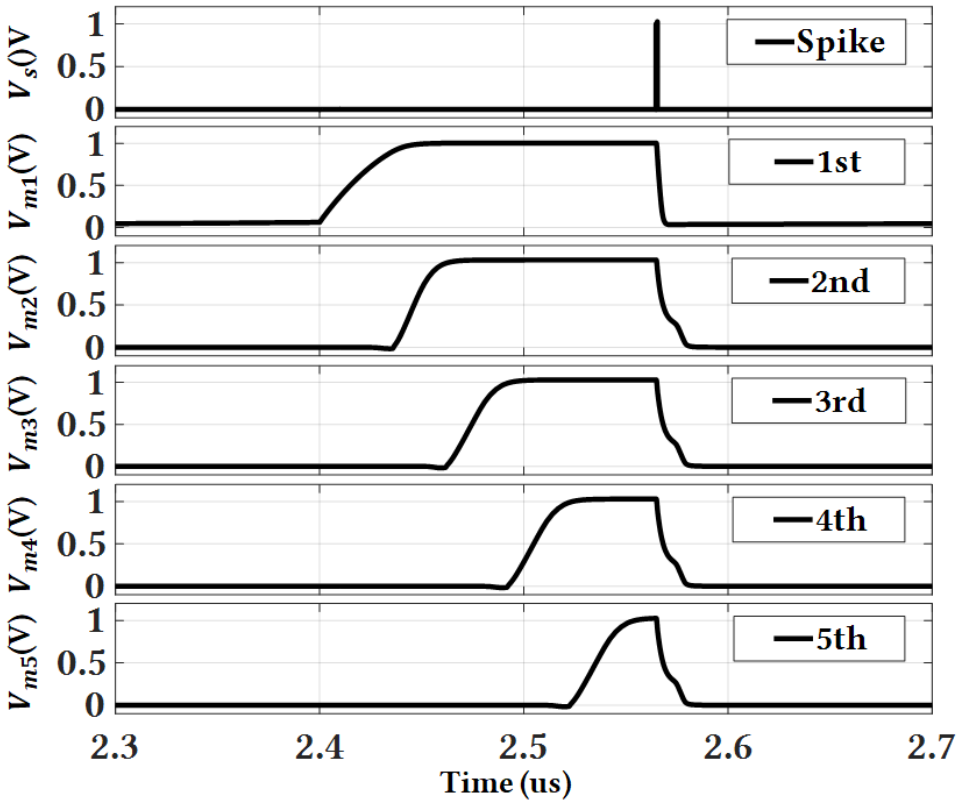


Fig. 2.4. 4 Each stage's latency period and final spike signal chart

It is clear that the proposed MLN could generate larger rang latency period than single-stage LIF neuron. In fact, its scaling up ability is unlimited if there is no neuron amount restriction.

Chapter 3. Encoder

3.1 Motivation

No matter in traditional electronic systems or in neuromorphic computing systems, it is not possible to process sensory information directly without any signal pre-processing. Analog to Digital Converter (ADC) [90] is such kind of pre-process module (in practice, the most frontend module is sensor and amplifier) that could transform analog signal to digital signal which can be accepted by digital processor [91-93]. In neuromorphic computing system[94], the signal format is spike pattern. Therefore, it is required to design a similar signal transforming module to convert sensory information into spike signal[16, 95].

In neuromorphic computing system, there are several spike signal formats including rate spike code, latency spike code, and ISI spike code [96]. Among these codes, rate spike code and latency spike code are one dimensional code, ISI spike code could be either one dimensional or high dimensional code. If timing information is considered, rate code is time independent code, latency code and ISI code are temporal code. To implement these codes, different encoding schemes are invented.

For rate spike code, input signal is encoded into the spike firing frequency or spike amount within a sampling window [97]. This type of encoding scheme is the most popular encoding scheme when considering hardware implementing since spike characteristics other than the spike numbers are not brought into consideration. Although it has simpler structure, many useful information has also been ignored. By adopting this code, the temporal information is only trivial information. This property makes it impossible to be applied in complex environment.

Recently, researchers have found that temporal order, or phase information, plays important role in information representing [79]. Latency code and ISI code are these two codes that has such kind of properties[98]. Latency code is a kind of code that only caring about the period between reference point and the first spike firing point [99]. The other temporal code is ISI code which depends on the time correlation of spikes [100]. In this case, the inter-spike intervals will carry the useful information, which is the main information carrier. Comparing with rate spike code, in ISI

spike code, the spike number and the inter-spike intervals are all used to carry the information, which could have higher information density.

3.2 Rate Encoder

3.2.1 Analog Rate Encoder

Rate encoding is a kind of frequency modulation technique. The core idea is transforming signal's amplitude, e.g. sinusoid wave's amplitude, into spike frequency. Such kind of frequency could be represented by different format. In neuromorphic computing field, the carrier signal's format is spike train, and the firing frequency is the encoded information that we need. In neuromorphic computing system, it is not convenient to use spectrum analysis technique to process such kind of spike train [10]. Therefore, it is widely accepted that counting spike number is the best method to post-process these spike trains when the spike total amount is in small level (< 100). When the spike amount is too large, synaptic method is adopted (in fact, small amount spike could also use synaptic method) which use synaptic current to represent the spike train's information.

Firstly, it is required to check the rate code and its encoding scheme. As aforementioned, rate encoding is transforming analog signal into spike train. Such kind of relationship is illustrated in Fig. 3.2.1.

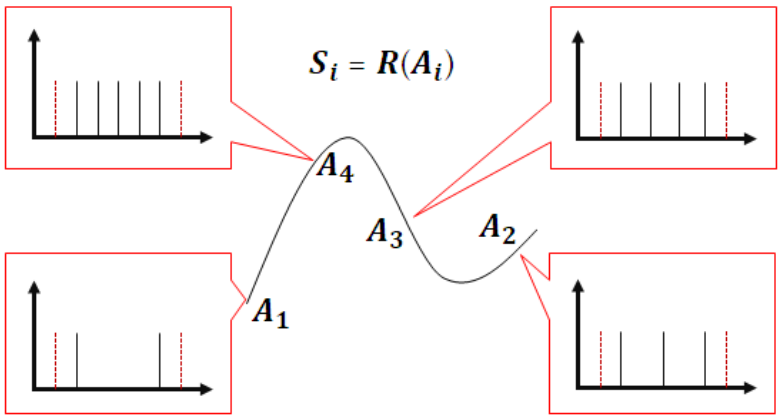


Fig. 3.2. 1 Rate spike code and encoding demonstration

As shown in Fig. 3.2.1, an analog signal has been transformed into spike trains. Four points with different amplitudes have been mapped to four spike trains. In this case, the relationship of

these values is $A_4 > A_3 > A_2 > A_1$. The outputs related to each value are 5-spike, 4-spike, 3-spike, and 2-spike respectively. This encoding scheme could be expressed in mathematic expression as

$$S_i = R(A_i), \tag{3.2.1}$$

where S_i and A_i represent output spike number, or firing frequency, and input signal's amplitude. The $R(\cdot)$ is the rate encoding function. The simplest way to implement a rate encoder is adopting one LIF neuron directly, which is illustrated in Fig. 2.3.1. In this case, the encoding equation could be expressed as

$$S_i = \frac{C_{mem}V_{mem}}{I_{leak}} \ln\left(\frac{\alpha V_{th} I_{leak} A_i}{V_{mem}}\right), \tag{3.2.2}$$

where C_{mem} is membrane capacitance, V_{mem} is membrane potential, I_{leak} is leakage current, V_{th} is threshold voltage, and α is design parameter. It is clear, the encoding process is nonlinear which may need post-processing if these spike train want to be used by following modules. A simulation result of rate encoding is illustrated in Fig. 3.2.2.

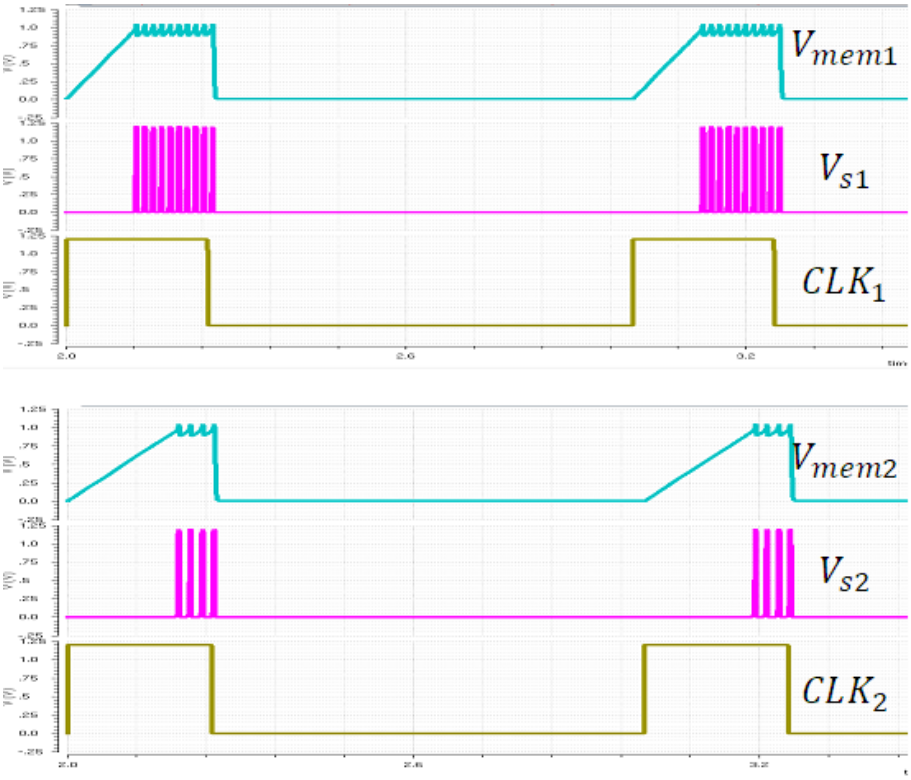


Fig. 3.2. 2 Rate encoder simulation results

AS shown in Fig.3.2.2, different spike amount spike trains are generated due to different input amplitudes.

In my Ph.D. research period, the proposed subthreshold LIF neuron has been fabricated with standard CMOS process. The die photo of single neuron is illustrated in Fig. 3.2.3.

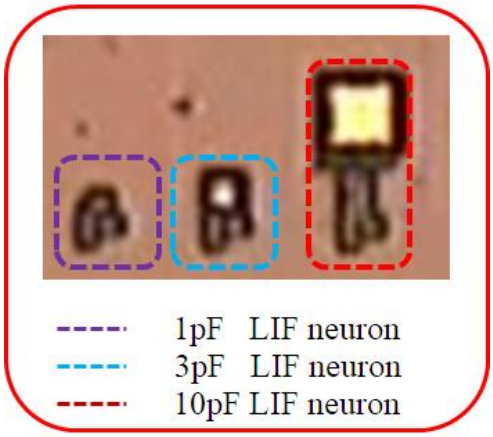
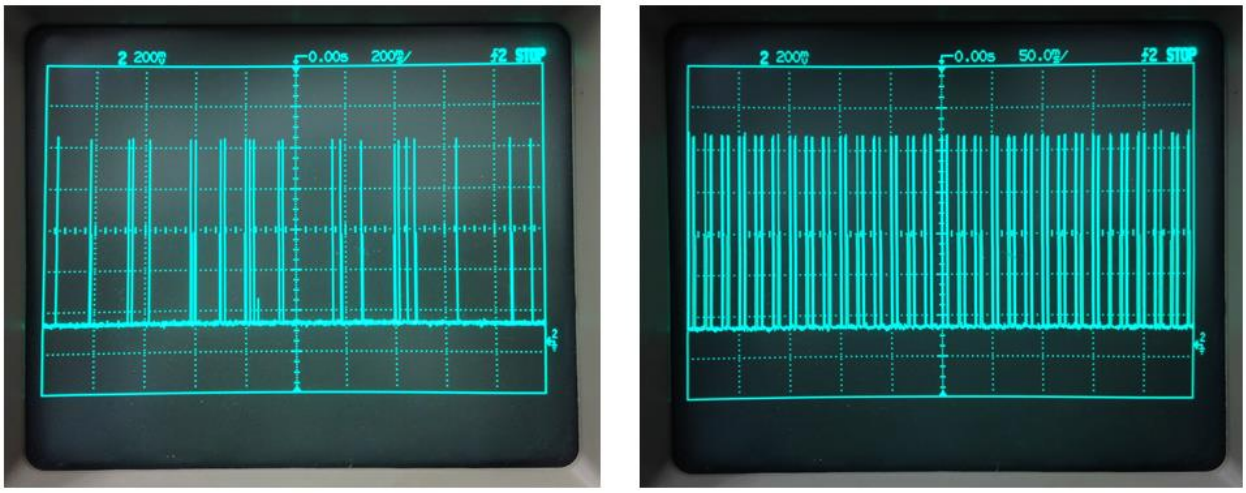


Fig. 3.2. 3 Three single LIF neuron die photo

It is clear that the membrane capacitor has occupied large die area. The measurement results of rate spike firing are illustrated in Fig. 3.2.4.



(a) (b)

Fig. 3.2. 4 Rate encoder measurement results (a) varying analog input signal; (b) DC input signal

As shown in Fig. 3.2.4, the proposed LIF neuron could achieve different firing frequency when the input signal's amplitude changing from time to time. For each rate encoder, the power consumption is related to the firing frequency. In typical frequency range, the power consumption's distribution is illustrated in Fig. 3.2.5.

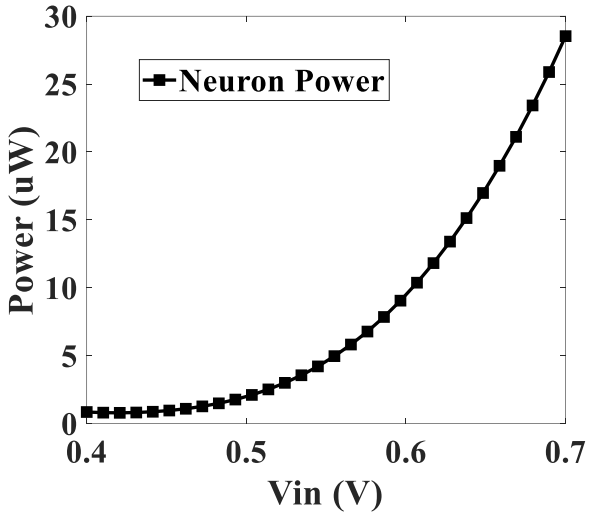


Fig. 3.2. 5 Power consumption of single neuron based rate encoder

As discussed in chapter 2, the neuron's firing frequency is related to the input excitation current. In most cases, such excitation current is proportional to input voltage V_{in}. Therefore, the firing frequency of neuron is also proportional to input voltage. As shown in Fig. 3.2.5, as the input voltage increases, the power consumption also increases rapidly. In most cases, due to the restriction of power capacity, the firing frequency has been restricted to a small value.

3.2.2 Digital Rate Encoder

Except analog rate encoder, some state-of-arts also introduced digital rate encoder which could be integrated with digital circuits directly. The fundamental idea of digital rate encoding scheme is transforming input signal's amplitude into different square wave (not digital code). This process is similar to ADC. The main difference between such kind of digital rate encoder and ADC is the output square wave's format, i.e. ADC will produce digital code (e.g. BCD code), but digital rate encoder will produce square wave train (each cycle represents one spike). The signal flow of the difference is illustrated in Fig. 3.2.6.

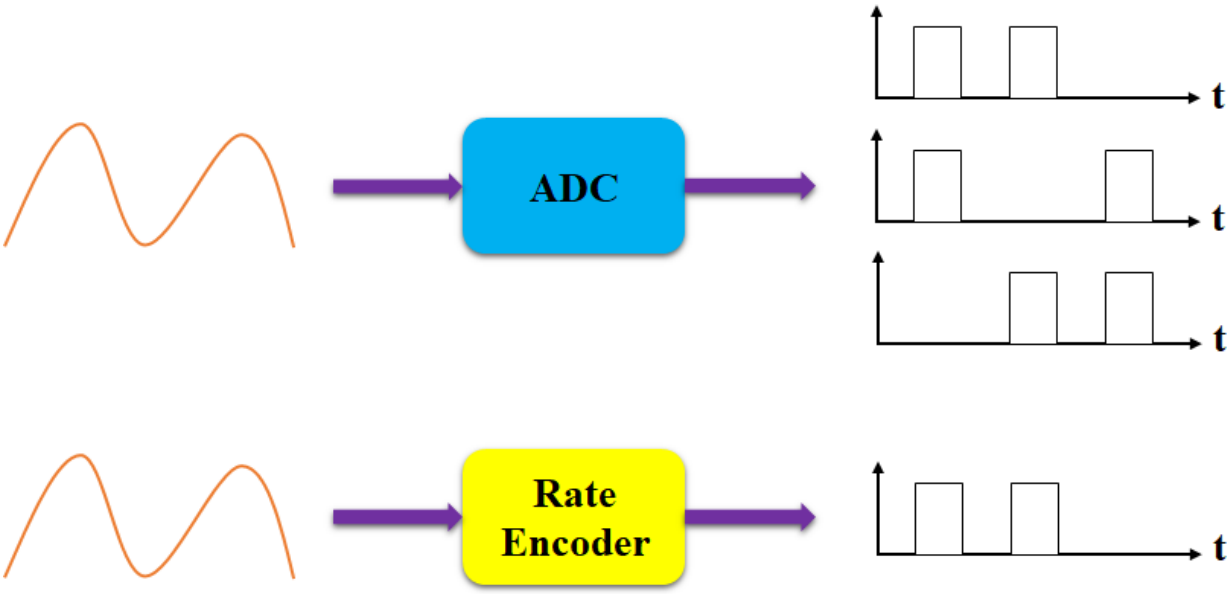


Fig. 3.2. 6 The signal flow of ADC and digital rate encoder

As shown in Fig. 3.2.6, the output of ADC is several paths square waves and the code in Fig. 3.2.6 is “110”, “101”, and “011” respectively. The output of digital rate encoder is a just one path square ware. If we split the output square wave train with sampling clock signal, in Fig. 3.2.6 the square ware in one sampling period is illustrated and it represent two spikes.

3.3 Latency Encoder

3.3.1 Analog Latency Encoder

Latency encoding scheme has been widely used to build optic sensor’s encoder [51]. The basic idea of latency code is transforming signal’s amplitude into the time delay period from starting time point or reference point to the first spike appearing point. It is a kind of level value converting to time distance value process. Not like rate code, latency code’s spike amount is not important. We only care about the first spike and the reference point. In practice, the reference point is the same as the starting edge of sampling window. In general, latency encoding process is illustrated in Fig. 3.3.1.

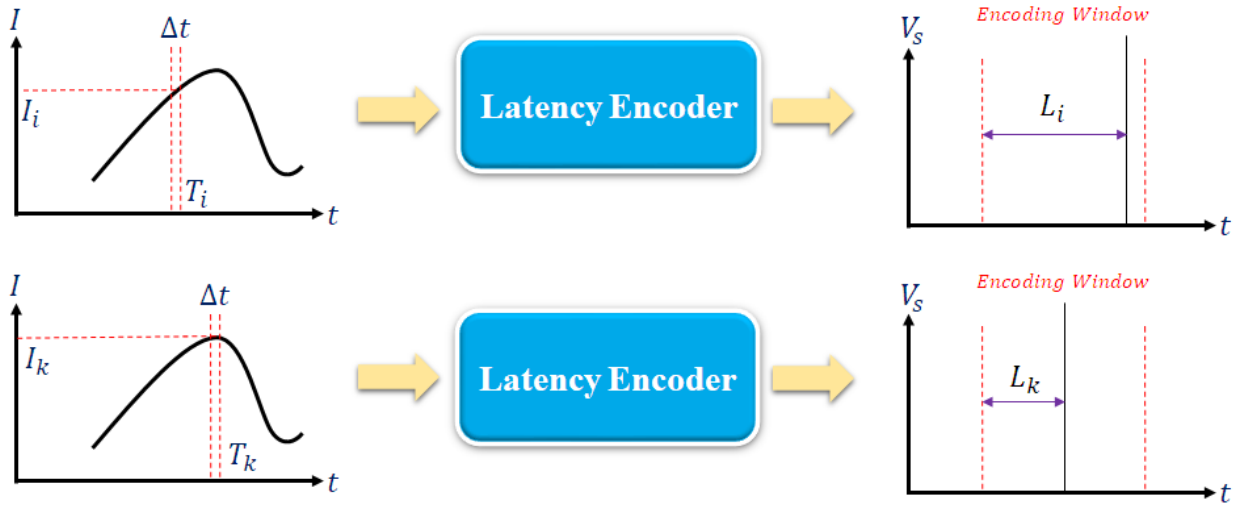
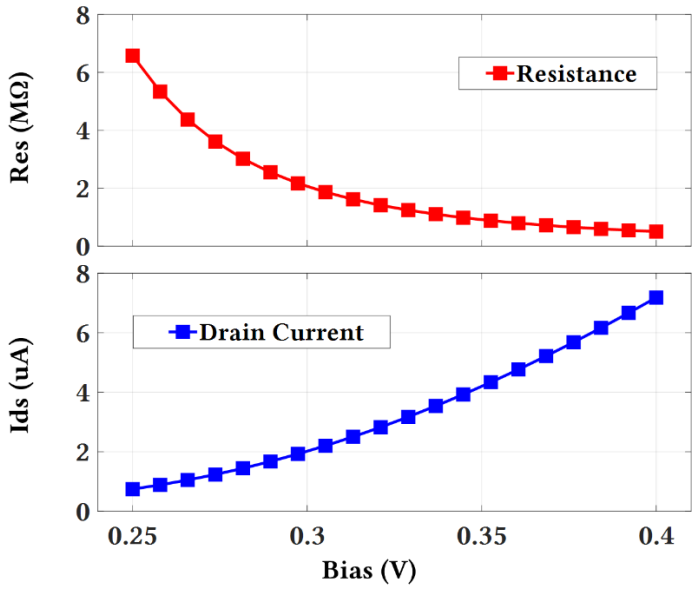


Fig. 3.3. 1 Latency encoding signal flow

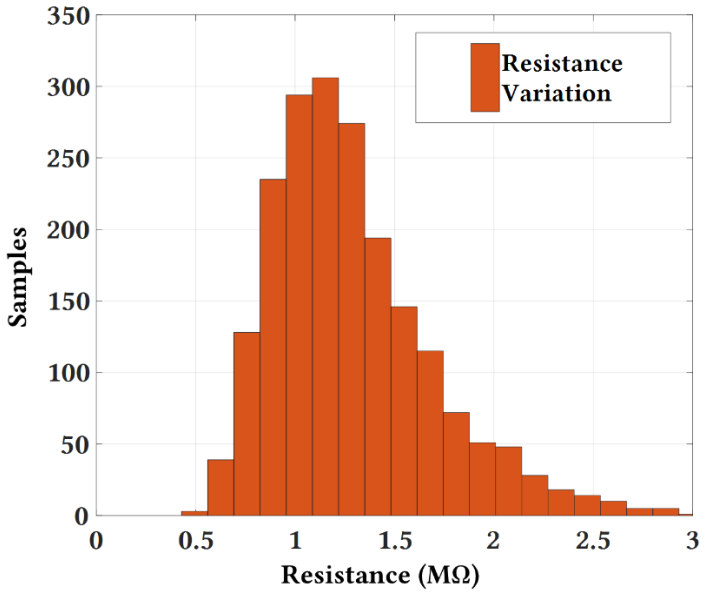
As shown in Fig. 3.3.1, the input signal's amplitudes (I_i and I_k) have been transformed into different latency periods (L_i and L_k). The simplest latency encoder could be built with a single neuron. However, one-stage neuron's integrating time would be interfered by many aspects such as resistor's variation. Since the integration time within I&F neuron is determined by a resistor-capacitor (RC) node, the integration time could be used as the latency distance, which is the key parameter for latency code. Therefore, the values of these capacitor and resistor determine the latency value, and the variations between these two parameters impact the accuracy. Practically, the product of RC is larger than 10^{-7} level, since smaller latency distance would lead to lower signal-noise-ratio (SNR). In analog spike neuromorphic computing, it is better to guarantee that the latency should larger than 100ns to guarantee enough high SNR. In this case, the resistance is in $10^{6}\Omega$ level and capacitance are in 10^{-13}f level. Under complementary metal–oxide–semiconductor (CMOS) technology, both RC are area consuming components, e.g. $664um^2$ for $1M\Omega$ high temperature coefficient polysilicon resistor (>4000ppm) and $1.26mm^2$ for $1M\Omega$ low temperature polysilicon resistor (<100ppm) in typical 180nm CMOS process, which are unacceptable in designs.

A good substitute for resistor is one MOSFET transistor. It is widely accepted that one MOS transistor could behavior like a linear resistor in linear region. In order to acquire 1MΩ or higher-level resistance, transistor working in subthreshold region would be the best choice. Under CMOS

process, typical threshold NMOS transistor's intrinsic resistance characteristics are illustrated in Fig. 3.3.2.



(a)



(b)

Fig. 3.3. 2 (a) NMOS transistor's characteristics under different biasing voltage; (b) output resistance variations with V_{th}-0.02 V biasing voltage

As shown in Fig. 3.3.2 (a), the output resistance has less variation within 0.35V~0.4V region which is around the intrinsic threshold voltage, i.e. 0.384V in this process. In Fig. 3.3.2(b), Monte Carlo analysis shows that the output resistance mainly appears within $0.9M\Omega$~$1.4M\Omega$. In most state of arts, RC node based electronic neuron implementation did not make any compensation for such huge variation which may lead to unpredictable system unreliable. Latency encoder built with aforementioned neuron could only generate trivial information. In order to make these outputs acceptable for standard training modules, it is better to adopt MLN neuron, which has been discussed in section 2.4.1, to build the latency encoder.

3.3.2 Digital Latency Encoder

Latency encoder could also be built by digital circuits. The encoding scheme is similar, except the spike format is standard square wave. The simplest way to implement a digital latency encoder is combining a digital neuron with a counter together. The signal flow of such kind of latency encoder is illustrated in Fig. 3.3.3.

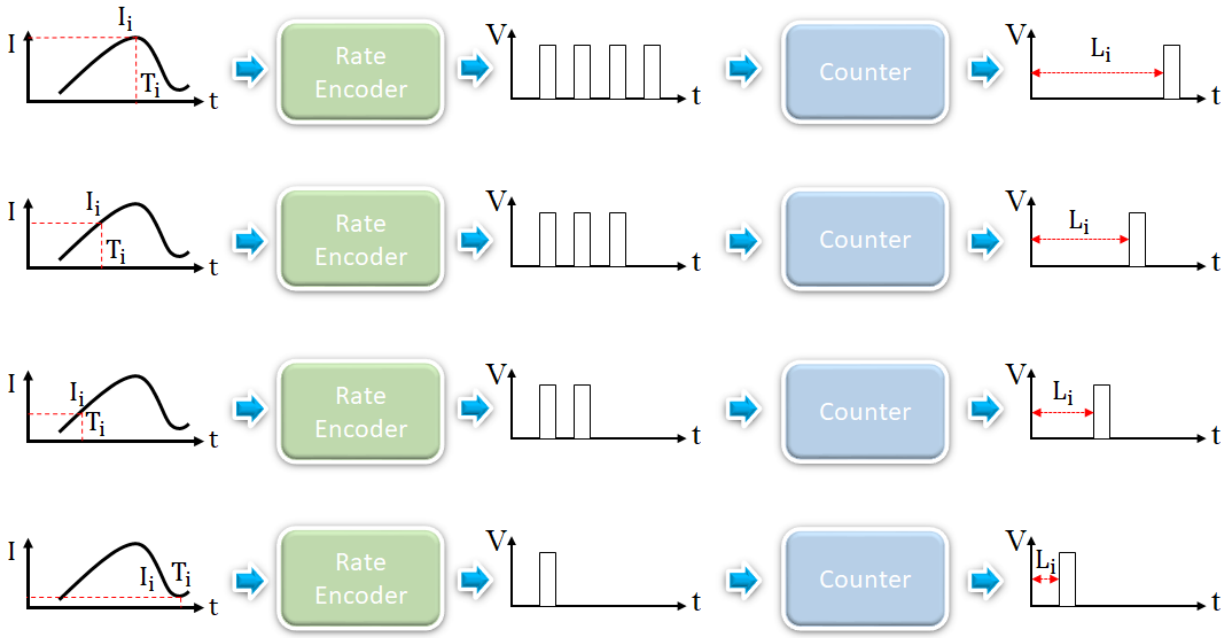


Fig. 3.3. 3 Digital latency encoder signal flow

As shown in Fig. 3.3.3, the digital rate encoder first transforms analog signal's amplitude to digital spike train. Then, the counter will count the spike number. After finishing counting, a digital spike will be generated.

Comparing with analog latency encoder, digital encoder has better performance in noise tolerance [101]. The latency period could be divided by square wave cycle. Therefore, the minimum resolution is only determined by the square wave's frequency. In analog latency encoder, though the resolution is unlimited theoretically. It is easy to be interfered by noise and other exceptions, e.g. transistor mismatch, voltage vibration, etc. Furthermore, it is also not easy to detect a single spike during sampling period. High accuracy amplifier is required to get the correct output value even if STDP technique [102] is adopted. Therefore, analog latency encoder would generate dummy spike train to make it easier to be detected. One example is illustrated in Fig. 3.3.4.

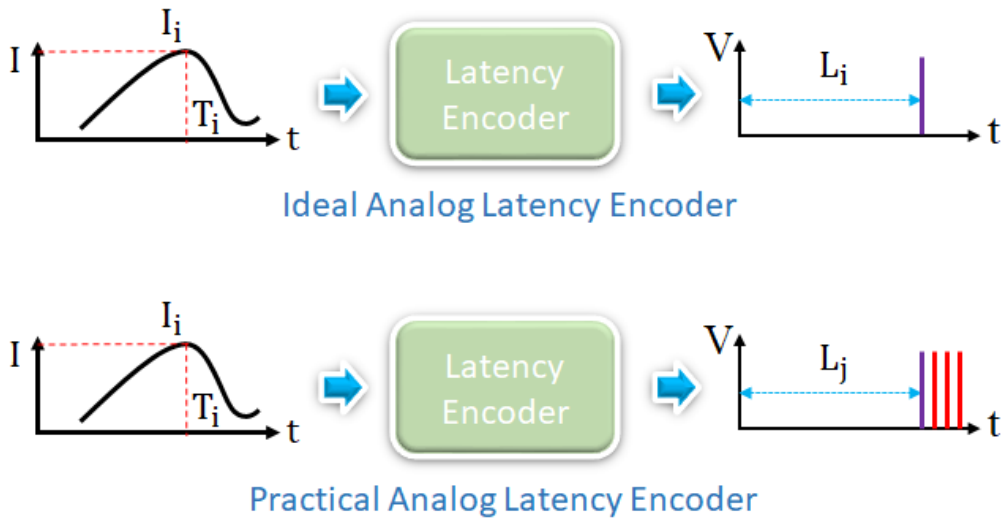


Fig. 3.3. 4 Analog latency encoder in ideal case and practical case

As shown in Fig. 3.3.4, the practical analog latency encoder would generate a spike train rather than a single spike. Therefore, the real latency period is not simply the time period from reference point to the first spike. The ideal latency distance and the practical distance has the relationship expressing as

$$L_j = L_i + \alpha T_f. \qquad 3.3.1$$

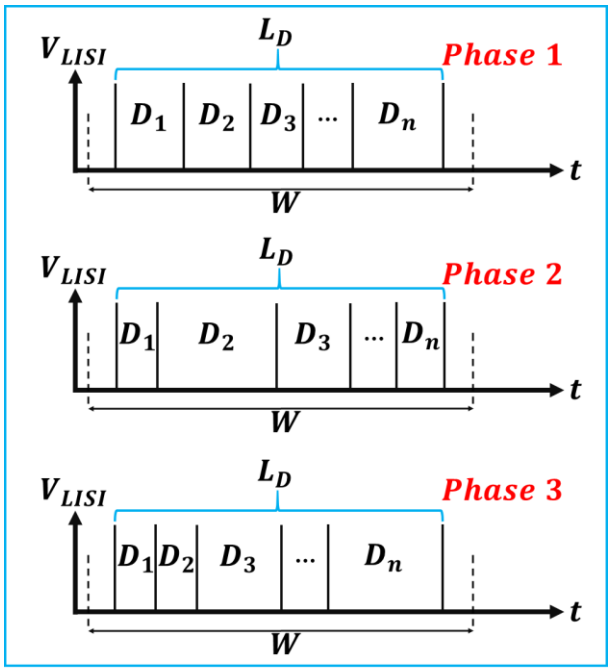
In equation 3.3.1, the L_j and L_i represent practical latency distance and ideal latency distance respectively. The parameter T_f is the spike firing period and α is the tuning coefficient (the typical

value of α could be 0.1 ~ 0.9). If the negative input value is considered, the αT_f part could be considered as leakage part. In other words, the practical latency encoder is a kind of leaky latency encoder. In digital latency encoder design, some delay node, e.g. converter-based buffer, is required to achieving such similar leakage function.

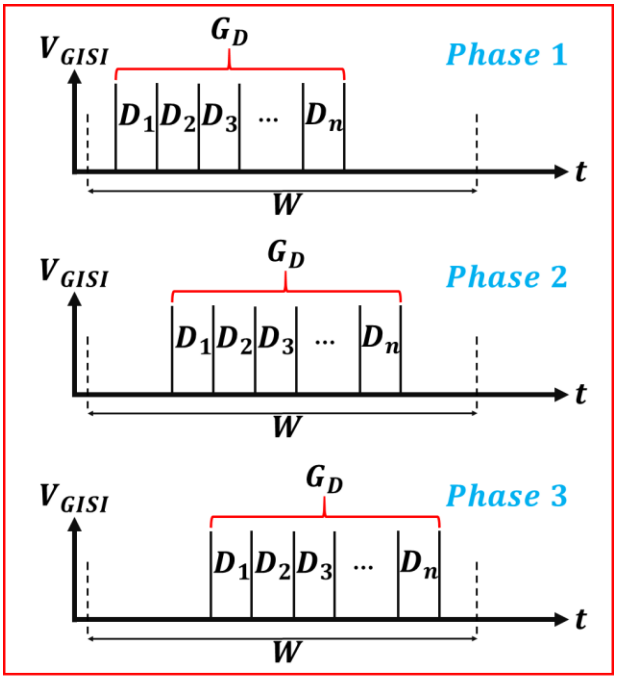
3.4 ISI Encoder

3.4.1 ISI Spike Codes Classification

ISI code is a kind of temporal code that including both spatial and temporal information [63]. In most cases, temporal code has been treated as phase code, since phase information could be converted into timing information in most situation. Under this assumption, the temporal code can be classified into two kinds of type: local phase code and group phase code. These two codes are illustrated in Fig. 3.4.1.



(a)



Group Phase

(b)

Fig. 3.4. 1 Local phase ISI code; (b) group phase ISI code

As shown in Fig. 3.4.1(a), the spike train's total width is constant and these inner spikes are in different position, or they have different relevant inner phase. In this case, the input signal's amplitude has been transformed into these inter-spike intervals. By looking at the total shape of these proposed local phase ISI spike code, it is clear that each ISI spike code could also represent one pattern. Furthermore, such kind of ISI spike code could also carry multi-dimensional information. One example is illustrated in Fig. 3.4.2.

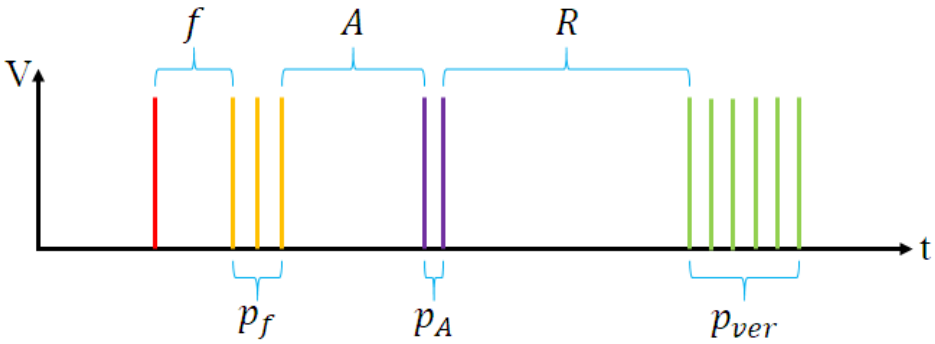


Fig. 3.4. 2 Multi-dimensional ISI spike code

As shown in Fig. 3.4.2, there are 6 main parts to construct the proposed ISI spike code. If this ISI spike code is carrying a picture's information, these 6 parts could be frequency (f), frequency unit (p_f), amplitude (A), amplitude unit (p_A), color gamut (R), and verification code (p_{ver}). If rate encoding is adopted, the same spike train could only represent the amplitude.

The other kind of ISI spike code is group phase ISI code which is illustrated in Fig. 3.4.1 (b). In this case, the whole spike train's group phase is the most important parameter the need to be considered. There is a simplest situation is that all inter-spike intervals have the same value. In this case, group phase ISI spike code is similar to latency code. However, this ISI spike code is much more robust than latency code due to its encoding scheme.

In practice, local phase ISI spike code and group phase ISI spike code are combined together to represent information. In this dissertation, three kinds of ISI encoders are designed including parallel ISI encoder, full signal iteration ISI encoder, and partial signal ISI encoder.

3.4.2 Parallel ISI Encoder

The fundamental unit to build neuromorphic computing system is neuron. The parallel ISI encoder is also built with neuron. In parallel ISI encoder, there is an external clock to control all neurons firing spikes. All these neurons are placed parallelism, and they will generate spike simultaneously. In other words, the priority level of these neurons is the same.

In design methodologies' view, each neuron should be in uniform structure and design parameters so that it is possible to make the encoder has scaling up ability. However, in order to generate ISI spike code, it is required to build an input layer for the proposed parallel encoder. Furthermore, it is also important to organize each neuron's output spike together to make it a spike cluster, i.e. ISI spike train. A good designed combiner would transmit each spike accurately without any distortion.

By adopting standard CMOS analog design techniques, a typical parallel ISI encoder's structure is illustrated in Fig. 3.4.3.

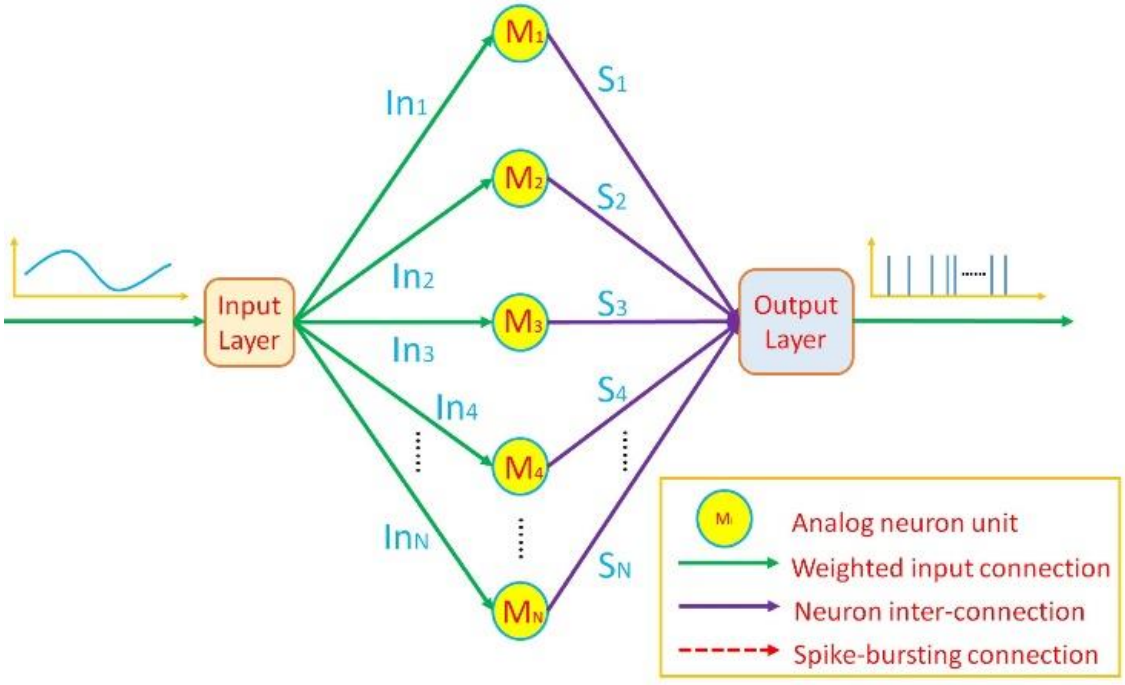


Fig. 3.4. 3 Parallel ISI encoder structure

In Fig. 3.4.3, analog signal has been converted into ISI spike train by the parallel ISI encoder. Within this ISI encoder, the main part is the neuron pool containing neuron arrays, i.e. $M_1, M_2, \dots, M_N$. The input module has been marked as "Input Layer" and the combiner has been marked as "Output Layer". A typical input module is shown in Fig. 3.4.4.

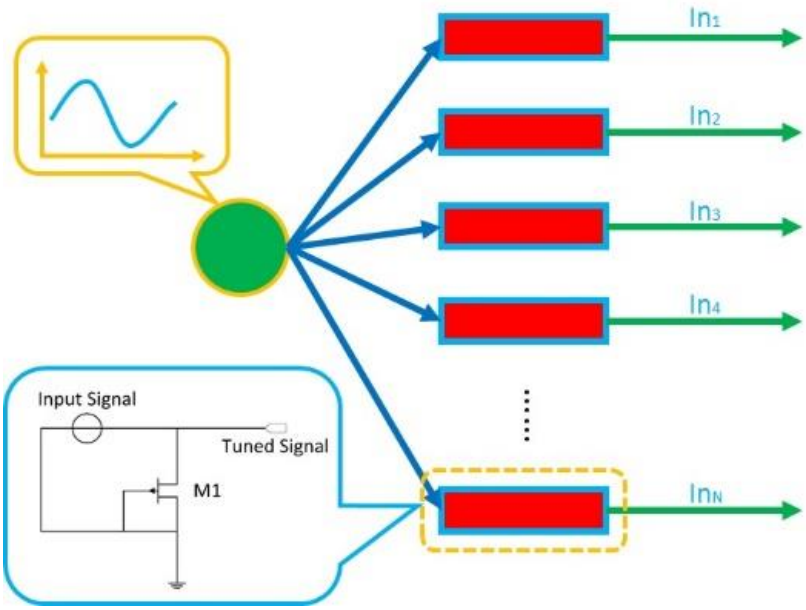


Fig. 3.4. 4 Input module

The basic part is the input module is a NMOS transistor which works as a diode. This transistor is served as a current tuner. For a fully functional input module, the input signal is first converted to current format and then split this current signal into N paths. Since each transistor has different size, the tuned current signal would be in different amplitude. Therefore, final output of input module is a current cluster with different levels. The signal flow is illustrated in Fig. 3.4.5.

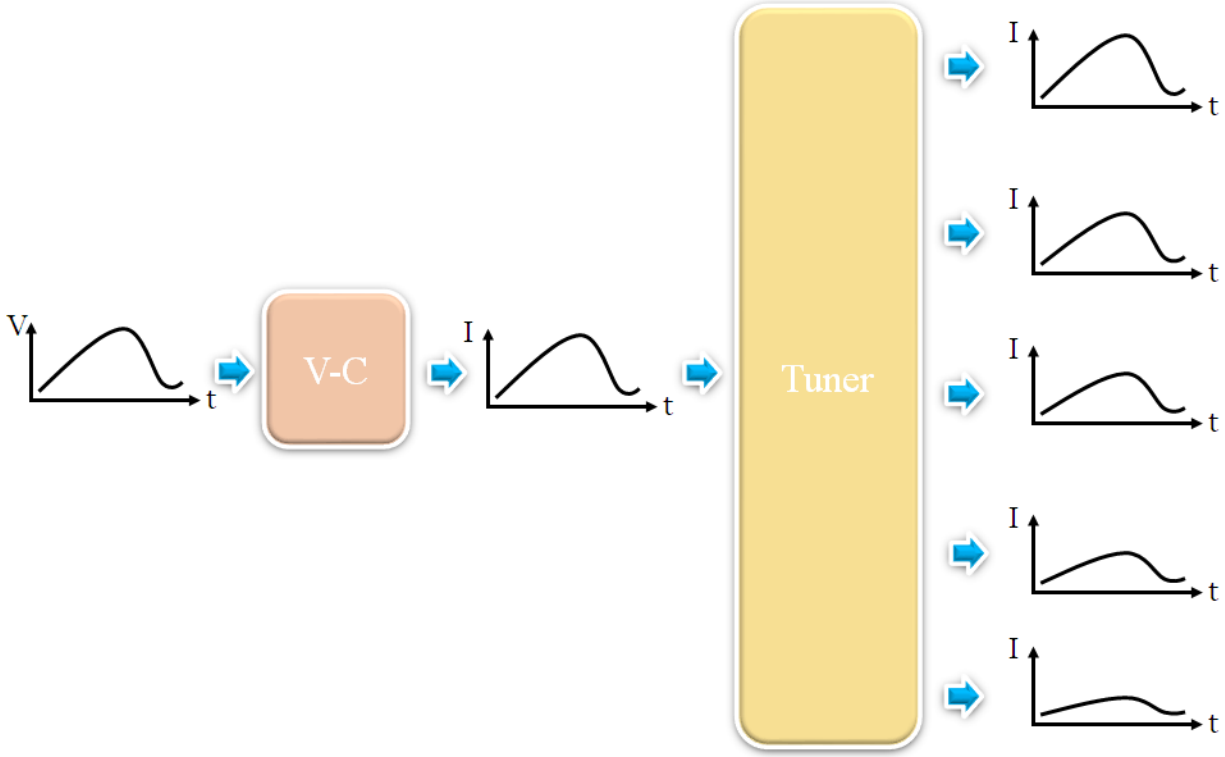


Fig. 3.4. 5 The signal flow of the proposed input module

It is clear that the input signal is first transformed into current format and then spliced into different levels current signals. After this processing, these currents signal could be sent to the neuron pool directly.

The simplified input layer circuit is illustrated in Fig. 3.4.6.

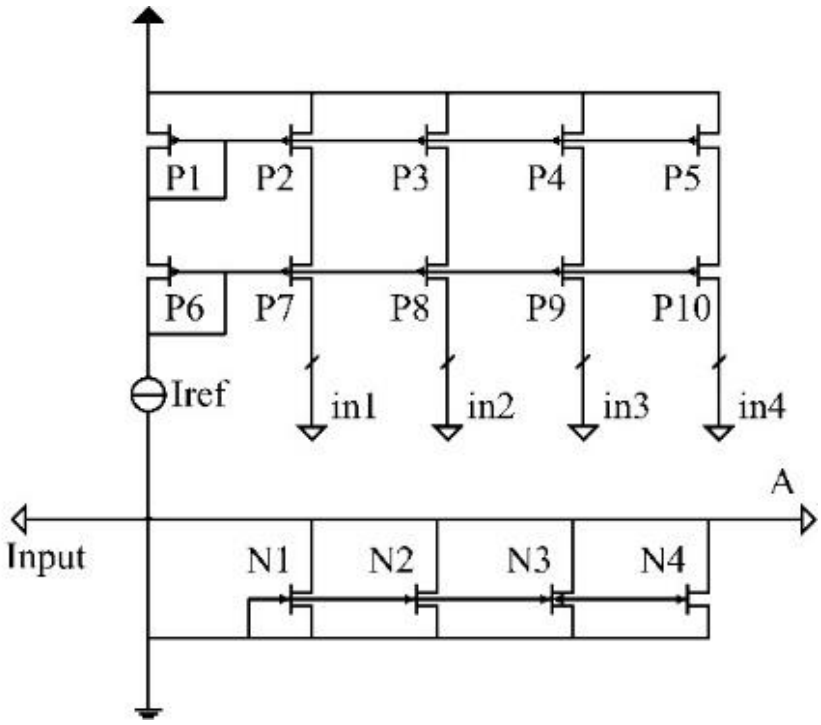


Fig. 3.4. 6 Simplified input layer circuit

As shown in Fig. 3.4.6, the input module has been divided into upper part and bottom part. The upper part is a current mirror cluster and the output excitation currents, in1 to in4, would be sent to neuron pool directly. The bottom part is serving as an input buffer.

The output module is another important part for this parallel ISI encoder. If the frequency is not too high (< 1GHz), a regular OR gate logic circuit could satisfy the requirement. If the single spike's frequency is too high, i.e. the spike width is too small, further processing unit is required to make sure that the output ISI spike train is correct.

In typical case, the output pike train of parallel ISI encoder is illustrated in Fig. 3.4.7.

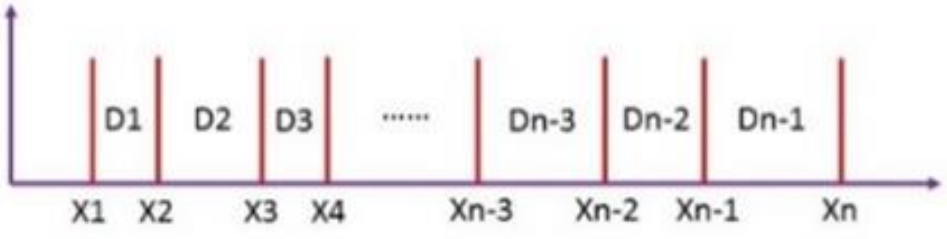


Fig. 3.4. 7 Parallel ISI encoder output spike train

As illustrated in Fig. 3.4.7, These D1, D2, … , Dn-1 representing inter-spike-intervals. Without generality, it is possible to describe the interval as

$$D_i = f(C_i, V_i) - f(C_{i-1}, V_{i-1}), \quad 3.4.1$$

where the function *f( )* has the shape as

$$f(C_i, V_i) = (C_i + 1)[\beta(V_i - \gamma) + \theta], \quad 3.4.2$$

where β, γ, and θ are design parameters. C_i and V_i are membrane capacitance and firing threshold voltage, respectively. In my Ph.D. studying period, the proposed chip has been fabricated with standard 180nm CMOS process. The whole chip die photo is illustrated in Fig. 3.4.8.

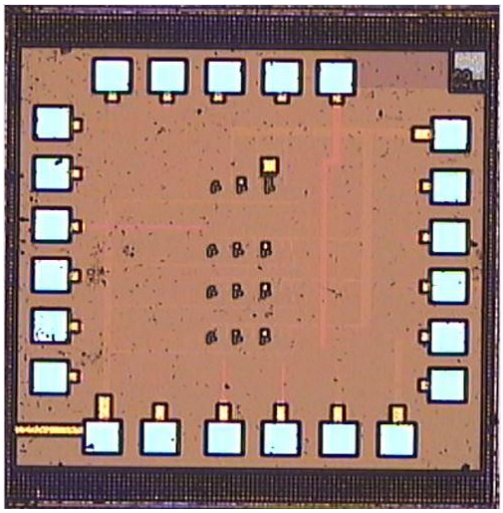
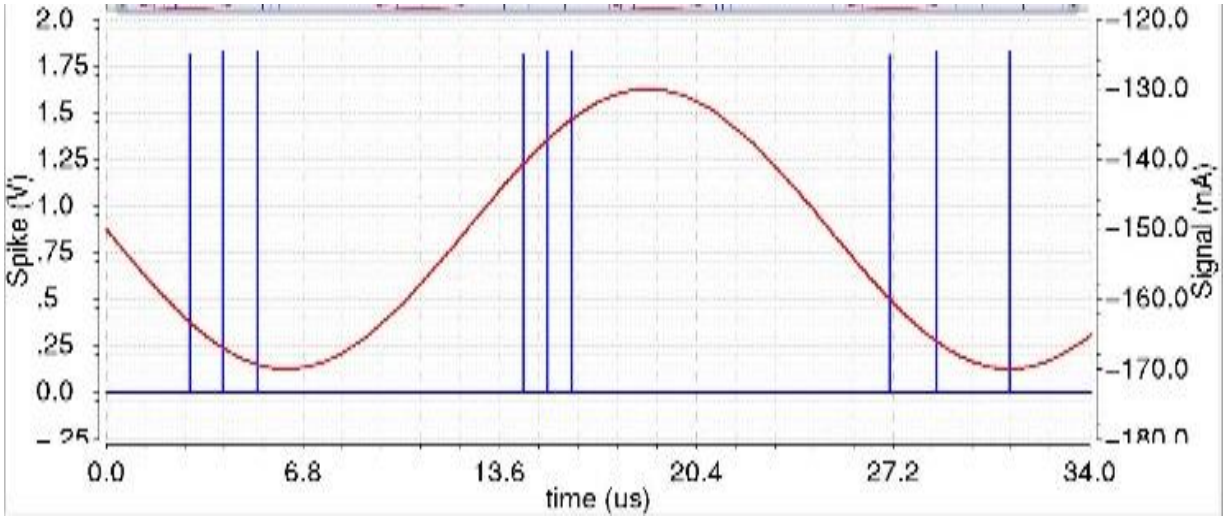
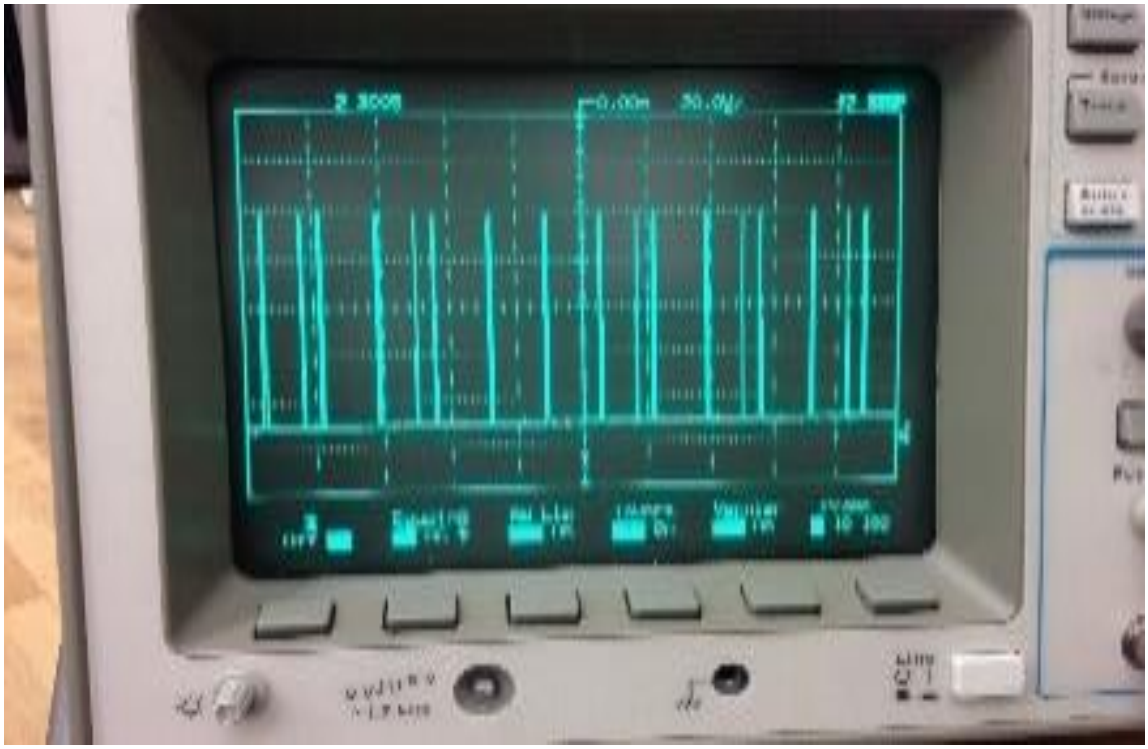


Fig. 3.4. 8 Full chip die photo of the proposed IS encoder

As shown in Fig. 3.4.8, The bottom part has three sets ISI encoder which has three neurons for each encoder. The post-layout simulation and measurement results are illustrated in Fig. 3.4.9. The full circuit schematic is illustrated in Fig. 3.4.10.



(a)



(b)

Fig. 3.4. 9 (a) Post-layout simulation; (b) measurement result

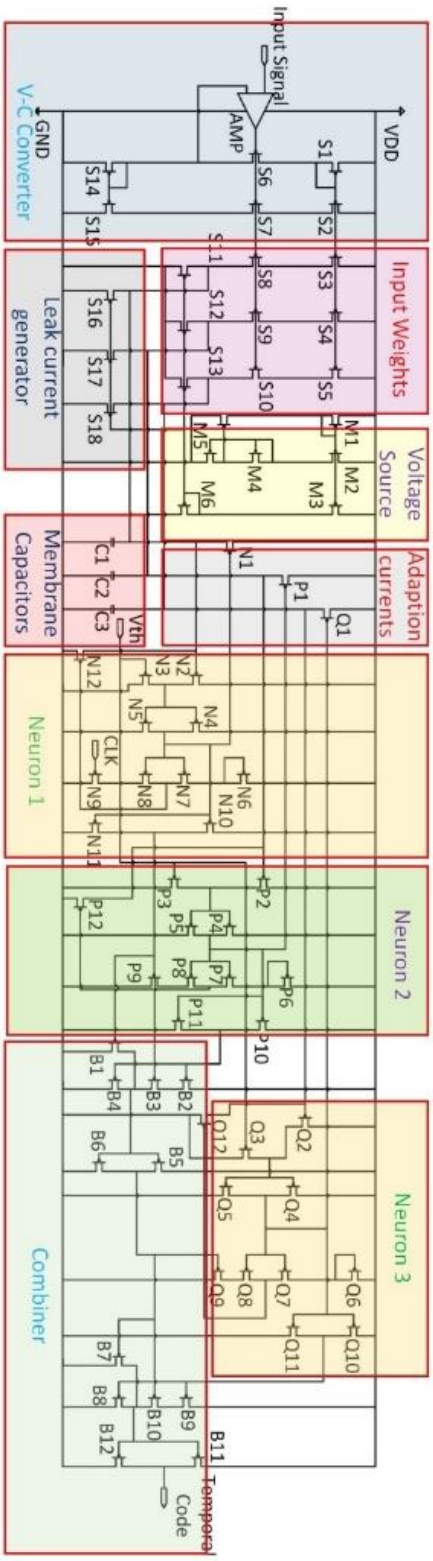


Fig. 3.4. 10 Full circuit schematic of the proposed ISI encoder

As shown in Fig. 3.4.9, the post-layout simulation and measurement result matches each other.

3.4.3 Iteration ISI Encoder

The second type of ISI encoder is a kind of temporal encoder that generating ISI spike train with iteration scheme. The structure of the proposed iteration ISI encoder also has three layers including input layer, neuron pool, and output layer. Not like parallel ISI encoder, these neurons inside iteration ISI encoder work asynchronously. In other words, all these neurons do not fire spike together. The system structure of the proposed iteration ISI encoder is illustrated in Fig. 3.4.11.

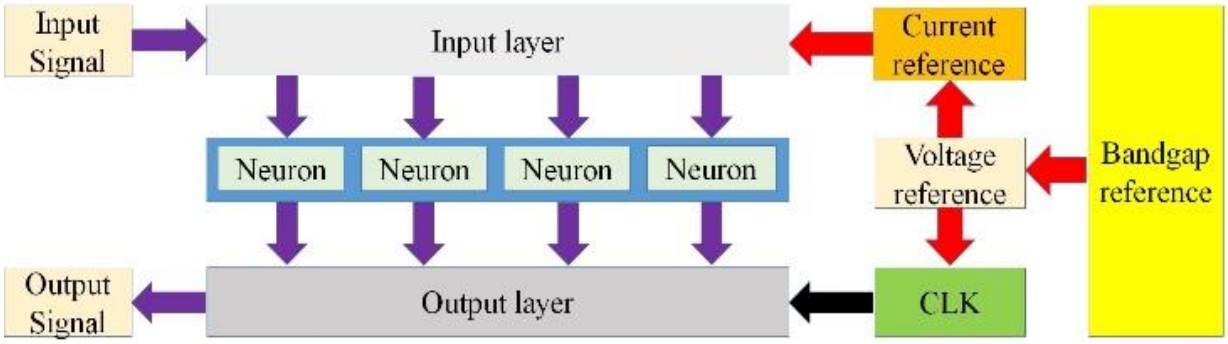


Fig. 3.4. 11 The whole structure of the iteration ISI encoder

As shown in Fig. 3.4.11, there are four main parts inside this ISI encoder, which are input layer, neuron pool, output layer, and peripheral function circuits, i.e. current reference, voltage reference, clock generator (CLK), and bandgap reference. The input layer has similar function and structure that parallel ISI encoder has. The output layer is designed based on OR logic but with iteration modification. The neuron pool is much more complex than parallel ISI encoder's neuron pool which could achieving spike signal iteration. Since the proposed iteration ISI encoder chip is a full functional chip, clock generator and source references (both current and voltage) are also built on chip, which make the proposed iteration ISI encoder working without any other external supporting circuits.

The iteration ISI encoder is the second-generation encoder which needs to consider about design trade-offs including power consumption, circuit complexity, design area, robustness, accuracy, verification ability, etc. Therefore, it is preferring to adopt purely transistor-based design methodology which using transistor to represent every component including resistor, capacitor,

diode, etc. In this ISI encoder, the core modules, i.e. input layer, neuron pool, and output layer, are all following the aforementioned design methodology.

To understand how the iteration scheme works, it is significant to have to deep view on the neuron pool. The detailed neuron pool signal flow diagram is illustrated in Fig. 3.4.12.

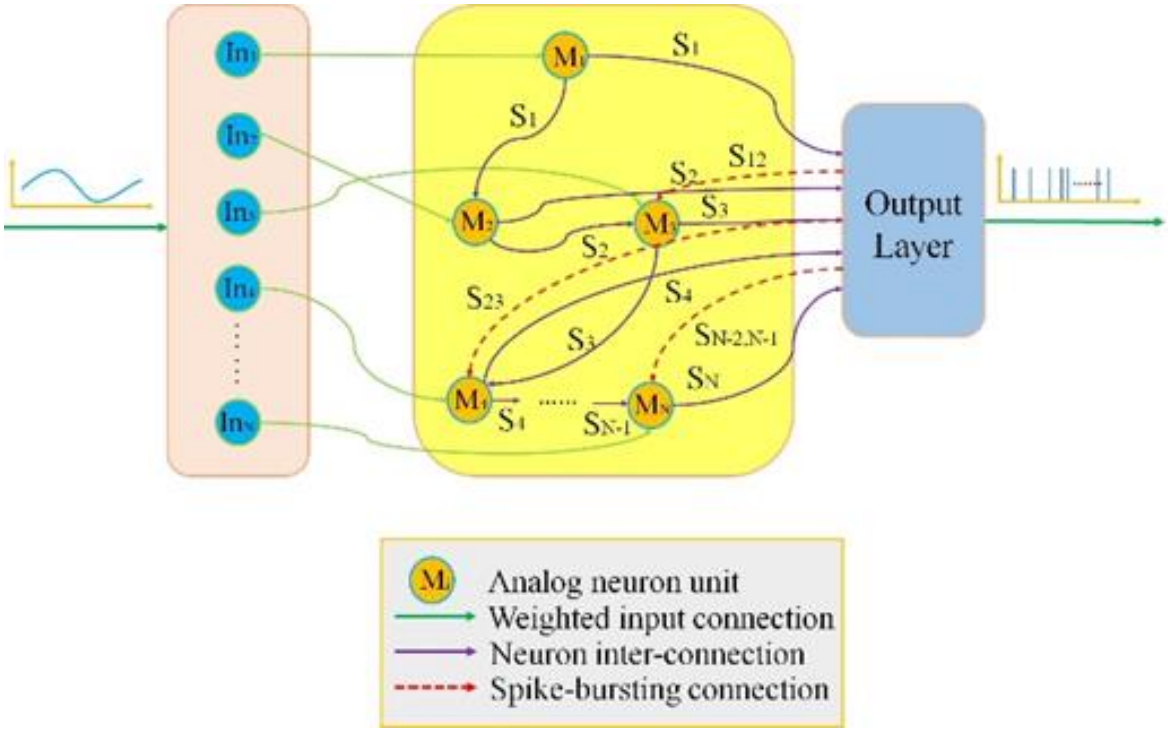


Fig. 3.4. 12 Iteration based signal flow

It is clear to show in Fig. 3.4.12, there are three kinds of signals that existing in this module including input signal (or weighted input connection signal marked with green), inner signal (or neuron connection signal marked with S_i), and feedback signal (or spike-bursting connection signal marked with red dashed line S_{ij}). In parallel ISI encoder design and latency encoder design, each neuron would only generate one spike, which will lead to the output spike number is equal to neuron number. In a sense, the information density is a little bit lower in these two encoders. In this iteration ISI encoder, it is desired to make the output spike number and the neuron number into an exponential relationship. Under such kind of relationship, it is possible to increase the information density and this is also the essential mechanism to make the output spike train having verification capability.

Typically, there are many exponential equations that could be used to serve as the design function. In the proposed iteration ISI encoding scheme, the 2 is chose to be the base number and the relationship between the neuron number and spike number could be expressed as

$$S_{spike} = 2^{N-1}, \tag{3.4.3}$$

where the N and S_{spike} represent the neuron number and spike number respectively. When designing a neuromorphic computing circuit, power efficient design has the highest priority ranking. Therefore, less neuron with more valid spike generating would be one of the good choice. Following the design equation 3.4.3, the circuit of neuron pool is illustrated in Fig. 3.4.13.

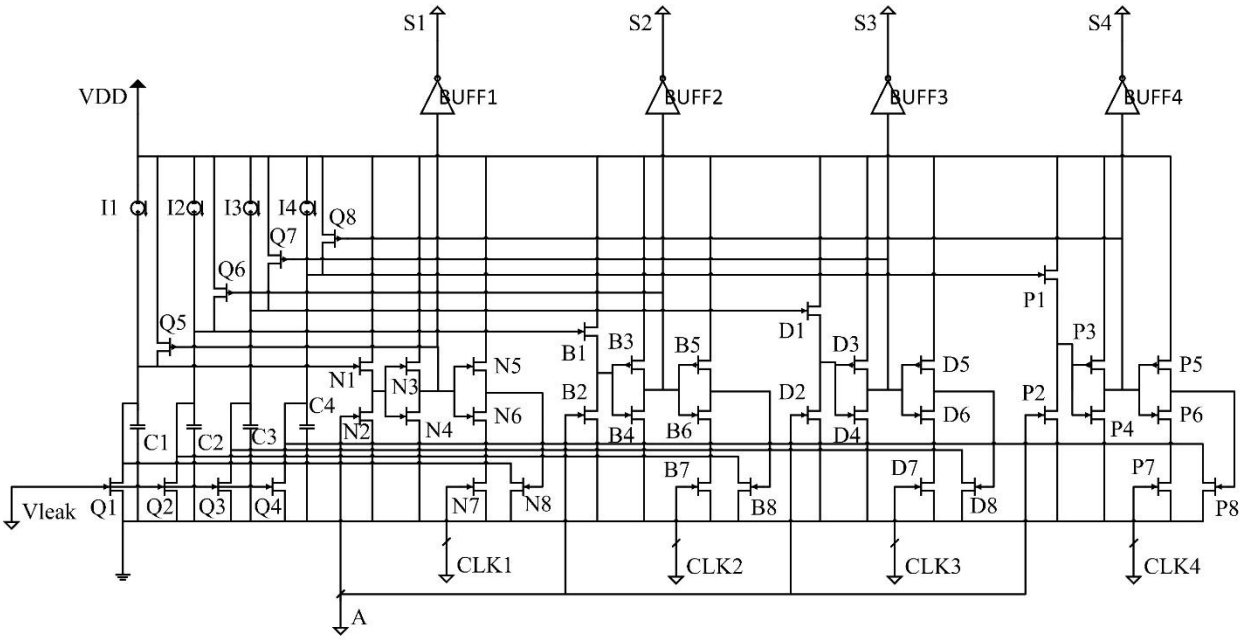


Fig. 3.4. 13 Neuron pool circuit of iteration ISI encoder

As shown in Fig. 3.4.13, there are four kinds of external signals that make this neuron pool work, which including leakage current control voltage V_{leak}, excitation currents I_1 to I_4, global clock signal $CLK1$, and input analog signal A. Among them, V_{leak} and I_i are used to make each neuron work correctly, $CLK1$ has determined the sampling window size, and A is the information carrier. In this circuit, $CLK2$ to $CLK4$ are the same signals as $S12$ to $S34$ which showing in 3.4.12. Four neurons' components transistors are marked with symbols N_i, B_i, D_i, P_i respectively. Q_i and C_i are each neuron's feedback signal transistor and membrane capacitor. The output signals are marked with S_1 to S_4.

Though these inner signals are processing with iteration scheme, each neuron's operation behavior is similar. For each neuron, excitation current I_i would first charge membrane capacitor C_i. Meanwhile, V_{leak} would make transistor $Q_{i,1-4}$ working under subthreshold region which could providing a constant leaking current. In this case, the total amount of current signal can be expressed as I_{in}. When considering the channel length modulation effect, the input current can be expressed as

$$I_{ds} = K\left(V_{gs2} - V_{thn}\right)^2(1 + \lambda V_{ds2}), \tag{3.4.4}$$

where K is related to CMOS process and design size, V_{gs} is determined by input signal, λ is channel length modulation coefficient, V_{th} is NMOS transistor threshold, and V_{ds} the potential between drain and source. Equation 3.4.4 could be simplified by combining constant value together into U, and the final expression is

$$I_{ds} = U(1 + \lambda V_{ds2}). \tag{3.4.5}$$

In equation 3.4.5, I_{ds} would increase if V_{ds2} increasing which is controlled by the membrane potential. Transistor $N3$ and $N4$ would map voltage V_{ds2} from high value to low value which could be adopted directly. The bottom path feedback control transistor $N8$ has been designed with wider finger width which can discharge membrane capacitor quickly. Transistor $Q5$ is serving as up path feedback leakage current controller that can holding membrane capacitor on a rest voltage level (> 0). The proposed charging and discharging process of membrane capacitor is illustrated in Fig. 3.4.14.

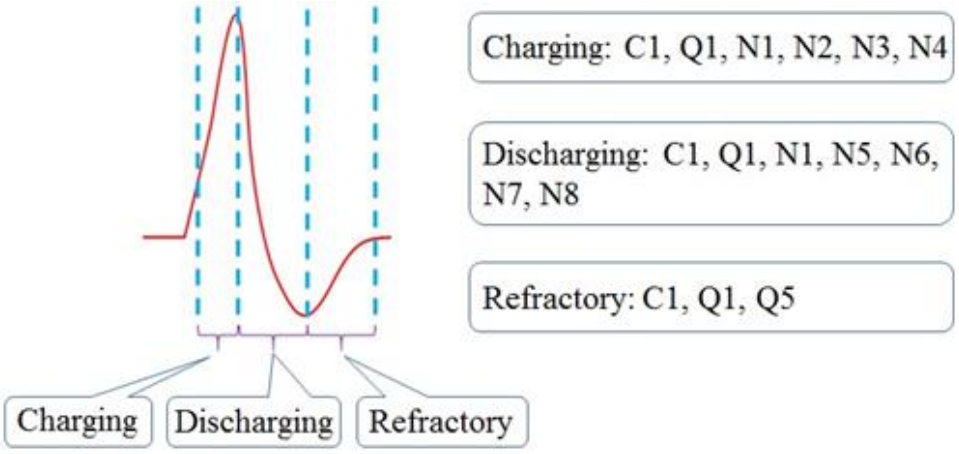


Fig. 3.4. 14 Membrane capacitor charging and discharging signal flow

It is clear that no spike would be generated during rest area, or refractory period which is shown in Fig. 3.4.14.

In iteration ISI encoder, the final part is output layer. This part is the key part to achieve signal iteration. The signal flow diagram is illustrated in Fig. 3.4.15.

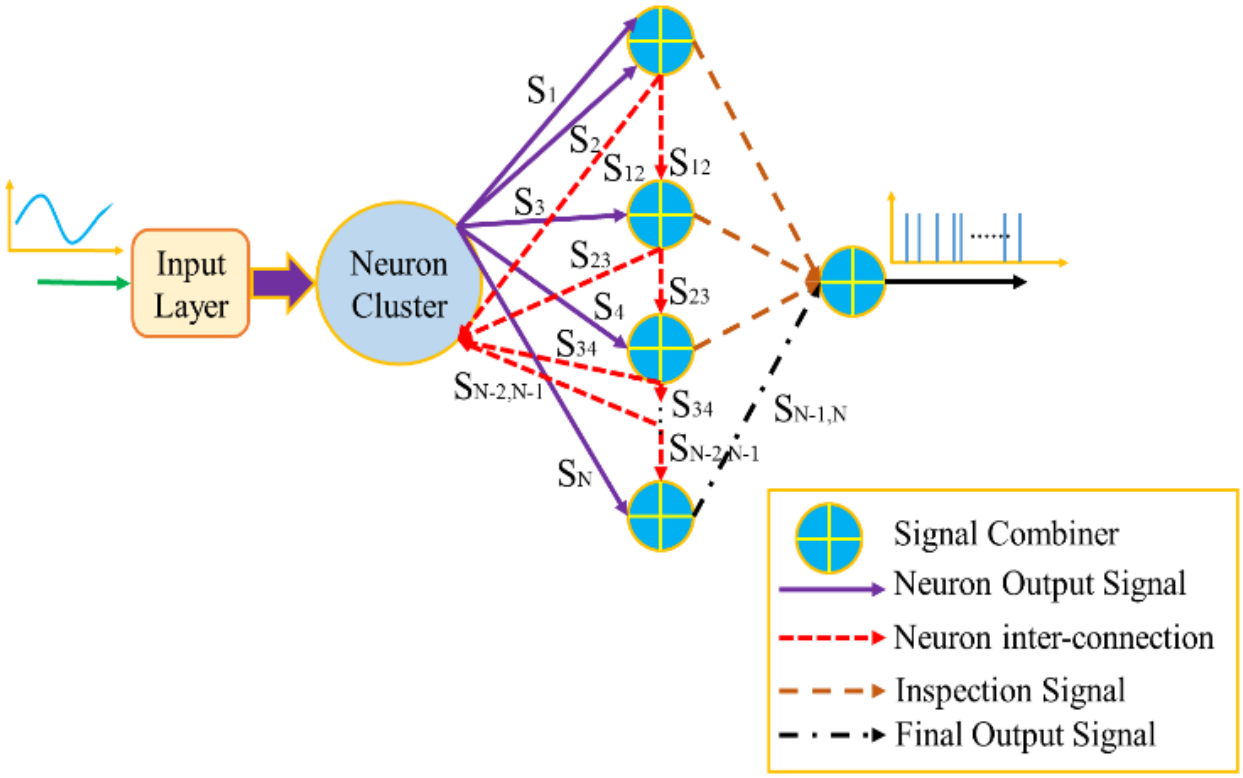


Fig. 3.4. 15 Output layer signal flow structure

There are four kinds of signals that appearing in the output module which including neuron output signal, neuron inter-connection signal, inspection signal, and final output signal. The proposed inspection signal would be used as verification information which would be discussed later.

To make these signals flow easy to follow, the simplified output layer circuit is also provided in Fig. 3.4.16.

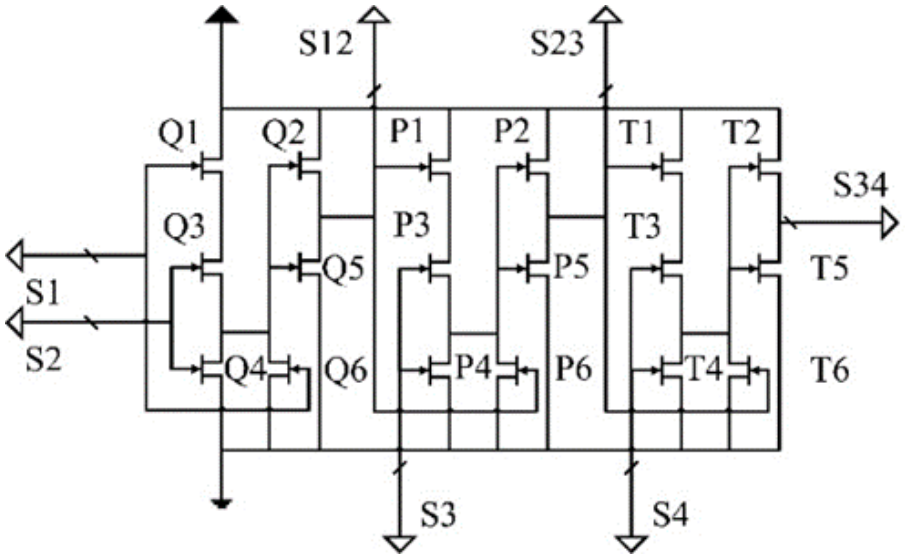


Fig. 3.4. 16 Simplified output layer circuit

As shown in Fig. 3.4.16, there are three similar modules which marked with Q_i, P_i, and T_i respectively. Among all the signals, $S1$ to $S4$ are input signals. Under one-time slot, only one spike would be sent to the combiner. Take $S1$ for example, when spike is applied on Q_i module, $Q1$ would work in cutoff region and $Q6$ will work in saturation region. In this case, the voltage on $Q2$ and $Q5$ would have the same value of the voltage on the drain terminal of $Q6$. Then $Q5$ will turn off and $Q2$ will stay high. The whole design is under 180nm CMOS process which could make the delay of the whole combiner in a small value (< 10ps). Therefore, no large error would appear in this proposed combiner. These rest two combiners, i.e. P_i sets and T_i sets, have similar operation process as aforementioned.

After analysis the circuit operation, it is required to make deep analysis on the encoding scheme. Since neuromorphic computing system is based on neuron unit, to make the encoding scheme easy following, it is desired to analysis from neuron model first. Simple LIF neuron model would be a good start point. For a LIF neuron, the key feature is the leakage current which can be expressed as

$$I_{leak} = I_0 e^{V_{gs}\alpha}, \tag{3.4.6}$$

where α is determined by physical process, i.e. Boltzmann constant, electron charge, etc., and I_o is the constant current value that transistor working under saturation region. The V_{gs} is the voltage potential between gate terminal and source terminal. In equation 3.4.6, the V_{gs} is less than

threshold voltage, which is the pre-requisition for subthreshold operation point. Without generality, the equation 3.4.6 could be expanded with Taylor's series when it works in deep subthreshold region as

$$I_{leak} \cong I_0\left(V_{gs}\alpha e^{V_{gs}\alpha} + \alpha e^{V_{gs}\alpha} + \alpha^2 {V_{gs}}^2 e^{V_{gs}\alpha}\right)|_{V_{gs}=0} = I_o\alpha. \quad 3.4.7$$

By applying equation 3.4.7, the time-based equation of LIF neuron circuit can simply re-write as

$$X_i = \frac{CV_{thi}}{I_{exi} - I_{leak}}, \quad 3.4.8$$

Where V_{thi}, C, I_{exi}, and X_i are transistor threshold voltage, membrane capacitance, and excitation current, and related timing point respectively.

For an iteration ISI encoder, both current input signal and previous signals work together which can be described as

$$S_n = f\left(S_1, S_2, \dots, S_{n-1};\ S_{n-2,n-1}; In_n\right), \quad 3.4.9$$

where S_n is current output signal of one node, $f(\cdot)$ is the active function, S_i is previous signal, and In_n is current input signal. In the proposed iteration ISI encoder, S_n is the output spike of one neuron, S_i sets are previous iterated spike train, and In_n is the sampled analog signal which pre-processing by input layer. To make the encoding scheme function looks clean and tidy, it is deserved to make abbreviation with $a = CV_{thi}$, $A_i = I_{exi} - I_{leak}$. Therefore, the equation 3.4.8 can be re-written as

$$X_i = \frac{a}{A_i}. \quad 3.4.10$$

Inter-spike interval's resolution is also an important performance that needs to be considered before finalizing the design. Since the interval is determined by the excitation current, it is required to determine the relationship between each interval. One possible definition is shown in equation 3.4.11.

$$A_N = \beta A_{N-1} = \beta^2 A_{N-2} = \cdots = \beta^{N-1} A_1, \quad 3.4.11$$

where β is rational coefficient.

As aforementioned above, the encoding scheme could be described by inter-spike intervals' relationship. Therefore, it is meaningful to begin with the basic case, i.e. N = 3 case. The 3-neuron based ISI iteration spike train is illustrated in Fig. 3.4.17.

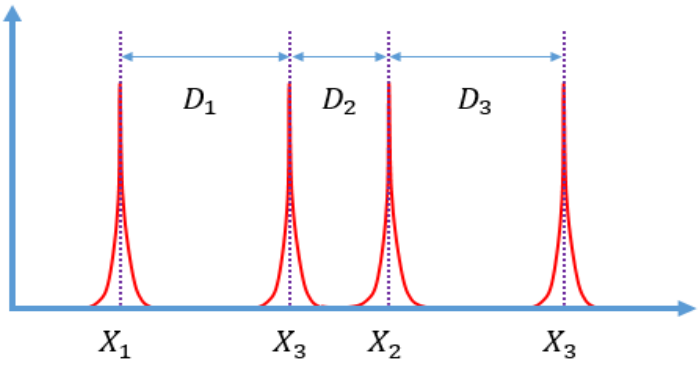


Fig. 3.4. 17 Three-neuron-based ISI encoder output

As shown in Fig. 3.4.17, each spike's time position is marked with X_i, and the intervals are marked with D_i. In this case, the intervals can be expressed as

$$D_1 = D_3 = \frac{a}{A_3} = \frac{a}{A_1} \cdot \frac{1}{\beta^2}, \tag{3.4.12}$$

$$D_2 = X_2 - X_3 = \frac{a}{A_1}\left(\frac{1}{\beta^1} - \frac{1}{\beta^2}\right). \tag{3.4.13}$$

The 4-neuron case would be more complex, and the related output spike train is illustrated in Fig. 3.4.18.

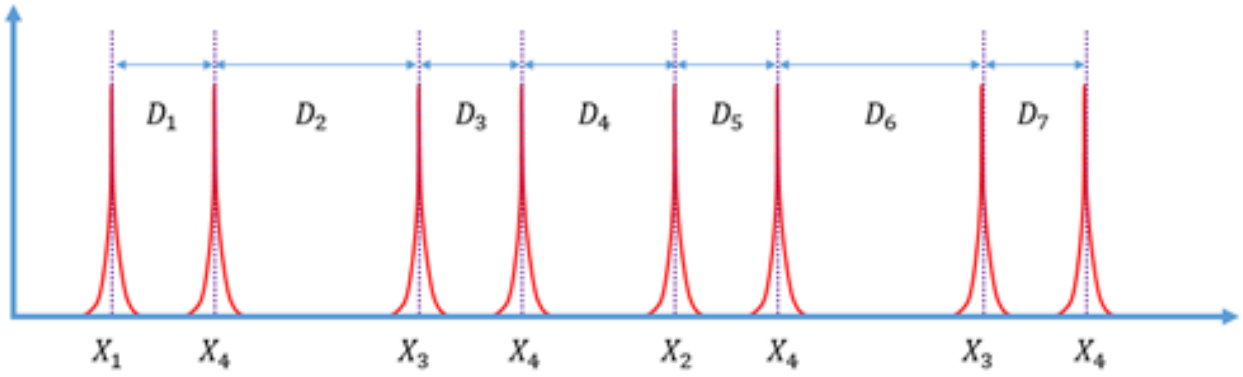


Fig. 3.4. 18 Four-neuron based ISI encoder output

The inter-spike intervals' relationship in Fig. 3.4.18 could be expressed as

$$D_1 = D_3 = D_5 = D_7 = \frac{a}{A_1} \cdot \frac{1}{\beta^3}, \tag{3.4.14}$$

$$D_2 = D_6 = X_3 - X_4 = \frac{a}{A_3} - \frac{a}{A_4} = \frac{a}{A_1}\left(\frac{1}{\beta^2} - \frac{1}{\beta^3}\right), \tag{3.4.15}$$

$$D_4 = X_2 - X_3 - X_4 = \frac{a}{A_2} - \frac{a}{A_3} - \frac{a}{A_4} = \frac{a}{A_1}\left(\frac{1}{\beta^1} - \frac{1}{\beta^2} - \frac{1}{\beta^3}\right). \tag{3.4.16}$$

Until now it is ready to get the final expressions for the general iteration ISI encoding scheme. The general ISI spike train is illustrated in Fig. 3.4.19.

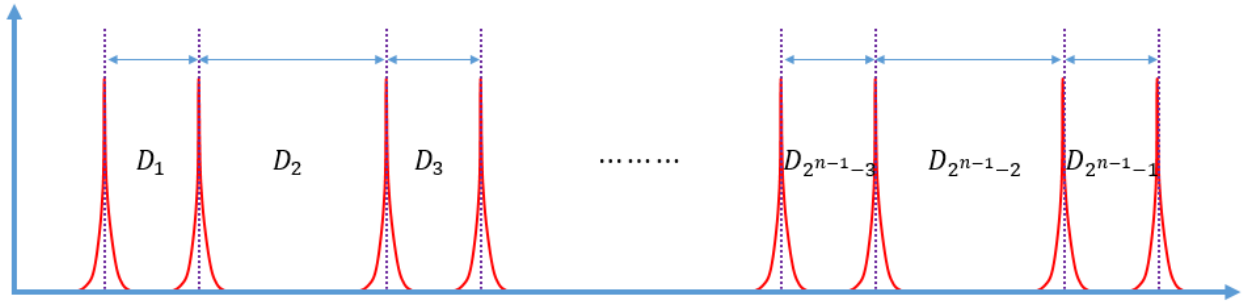


Fig. 3.4. 19 General iteration ISI encoder output

Following the trends that presented from equation 3.4.12 to 3.4.16, the general inter-spike interval relationships could be expressed as

$$D_{2^{n-1}-1} = \frac{1}{A_1} \cdot \frac{V_{n-1}}{\beta^{n-1}}, \tag{3.4.17}$$

$$D_{2^{n-1}-2} = \frac{1}{A_1}\left(\frac{V_{n-2}}{\beta^{n-2}} - \frac{V_{n-1}}{\beta^{n-1}}\right), \tag{3.4.18}$$

$$D_{2^{n-1}-4} = \frac{1}{A_1}\left(\frac{V_{n-3}}{\beta^{n-3}} - \frac{V_{n-2}}{\beta^{n-2}} - \frac{V_{n-1}}{\beta^{n-1}}\right), \tag{3.4.19}$$

$$D_{2^{n-1}-8} = \frac{1}{A_1}\left(\frac{V_{n-4}}{\beta^{n-4}} - \frac{V_{n-3}}{\beta^{n-3}} - \frac{V_{n-2}}{\beta^{n-2}} - \frac{V_{n-1}}{\beta^{n-1}}\right), \tag{3.4.20}$$

$$\vdots$$

$$D_{2^{n-2}} = \frac{1}{A_1}\left(\frac{V_1}{\beta^1} - \frac{V_2}{\beta^2} - \frac{V_3}{\beta^3} - \cdots - \frac{V_{n-2}}{\beta^{n-2}} - \frac{V_{n-1}}{\beta^{n-1}}\right). \tag{3.4.21}$$

By applying equations 3.4.17 to 3.4.21, it is possible to design required ISI encoder for specific applications.

3.4.4 Partial Signal Iteration (PSI) ISI Encoder

In order to reduce power consumption without loss accuracy, the PSI ISI encoder is designed. Considering the capability of the latency generation in the FAL neuron, the simplest way to implement spike-based ISI encoding is to map the latency distance into the inter-spike interval. The mapping process could be described as

$$D_i = T_{int} + L_i, \exists L_{ini} \longmapsto \mathfrak{S}_{ini}, \qquad 3.4.22$$

where D_i is the inter-spike interval, T_{int} represents the integration time, L_i represents the latency, L_{ini} represents the initial latency, and $\mathfrak{S}_{ini}$ represents the first spike. Equation (6) shows the mapping strategy, i.e. the generation of a latency spike and its following spikes based on the previous mapping strategy. This process is illustrated in Fig. 3.4.20.

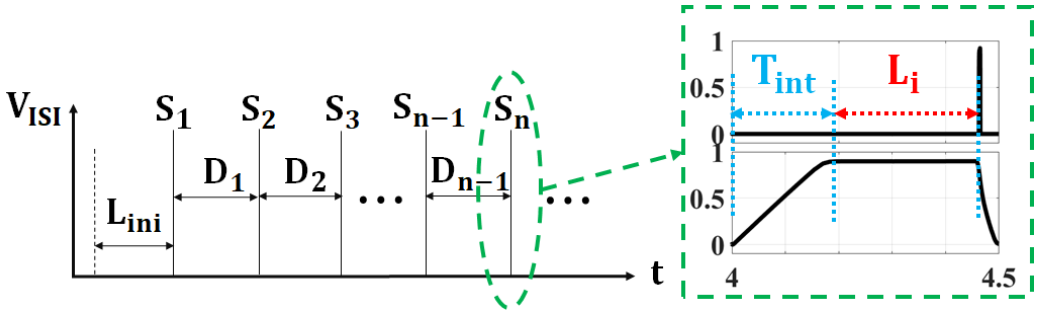


Fig. 3.4. 20 Mapping process from latency to ISI

The time phase for generating one single spike from the proposed FAL neuron is controlled by the CLK signal, as illustrated in Fig. 3.4.20. To implement the proposed ISI encoding scheme, an iterated structured implementation strategy is proposed, as shown in Fig. 3.4.21.

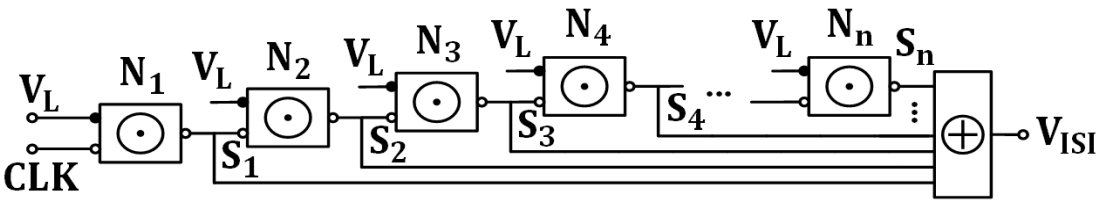
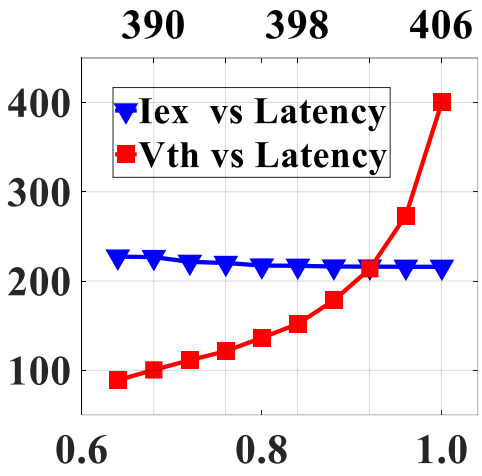


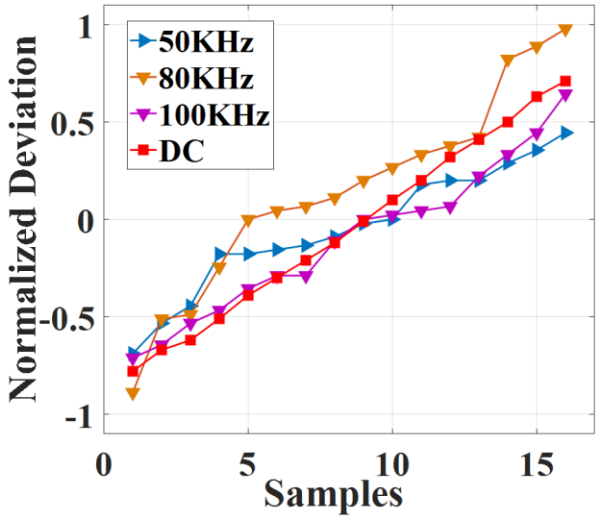
Fig. 3.4. 21 ISI encoder structure

As shown in Fig. 3.4.21, the external clock signal is only applied to the first FAL neuron N_1. Each following FAL neuron, N_i, adopts the generated output spike from its previous FAL neuron, S_{i-1}, to serve as its CLK trigging signal. All FAL neurons have the same excitation current, input signal V_L and leaky bias signal. By adopting such an iteration design, the robustness of the multi-spike ISI code could be enhanced by integrating multiple single-spike latency codes. By introducing the limitation of an encoding window (window size), it is possible to measure the proposed spike-based ISI code with both rate decoding scheme [103] and STDP-based decoding scheme. Furthermore, in the proposed ISI encoder, only one neuron works in active condition, i.e. generating spike, while the rest neurons are in quiescent condition. Compare with the spike rate encoder, such strategy would consume much less power.

In order to evaluate the proposed ISI encoder's performance, both S1 mode and S2 mode have been tested. The analysis results are illustrated in Fig. 3.4.22.



(a)



(b)

Fig. 3.4. 22 (a) S1 mode single latency range varying with excitation current and control voltage (V_L); (b) S2 mode spike train variation

In Fig. 3.4.22(a), it is clear that the latency is only sensitive to V_L, which makes it convenient to apply different voltages on V_L terminal. The piecewise linearity property of the output latency also provides the ISI encoder with the anti-distortion capability. S2 mode's test analysis is presented in Fig. 8(b). We randomly picked several ISI spike trains to calculate the average interval of each spike train and normalize it with simulation results, i.e. the ideal reference, to get deviations. Sixteen samples are picked under different frequencies including DC (0 Hz), 50 KHz, 80 KHz, and 100 KHz, respectively. It is clear that all these results are within ±1 range. Furthermore, these variations also follow the same trend which is easy to make compensation in the receiver terminal. The layout of the proposed ISI encoder is illustrated in Fig. 3.4.23.

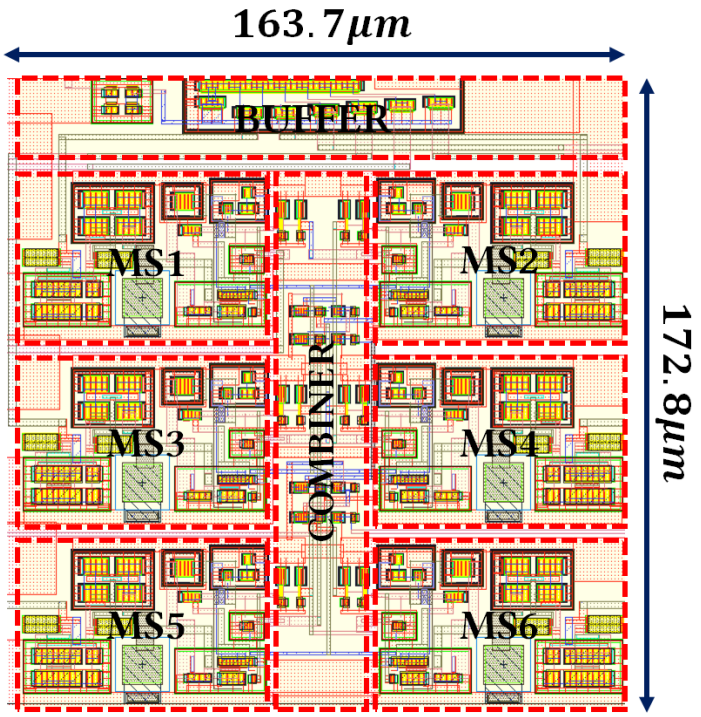


Fig. 3.4. 23 Layout of the proposed PSI ISI encoder

Since the neuron is a fundamental unit in neuromorphic systems, the most intuitive way to compare different neuromorphic systems' performances is by comparing the neuron's performance. In this paper, a brief comparison is presented in Table 3.4.1.

Table. 3.4. 1 Single Neuron Performance Comparisons

	PSI ISI Decoder	[104]	[105]	[106]
Pro.	$0.18\mu m$	$65nm$	$28nm$	$0.35\mu m$
Area	$0.101mm^2$	$0.024mm^2$	$70\mu m^2$	$0.0028mm^2$
V_{dd}	1.8V	2.5/1.2V	1V	3.3V
Freq.	1MHz	N/A	100Hz	N/A
Pow.	$1.63\mu W$	$14.4\mu W$	$0.36\mu W$	$40\mu W$

As shown in Table 3.4.1, the proposed FAL neuron has a good balance between operating frequency and power consumption.

Chapter 4. Decoder

4.1 Sample & Hold (SH) Based ISI Decoder

4.1.1 ISI Sum of Product Unit

There are two main parts to build the SH ISI decoder, which including sum of product (SOP) unit and ISI extractor unit. In this section, SOP would be discussed first.

The key idea about SOP is building a block that could sum up all the inter-spike intervals together and presenting the result with voltage level signal. The most robust way is using capacitor as the accumulator. It is widely accepted that the ideal capacitor charging and discharging, which companion with resistor, processes can be described by equations 4.1.1 and 4.1.2 [107].

$$V_c = V_{0c} + (V_s - V_{0c})\left(1 - e^{-\frac{t}{\tau}}\right), \quad 4.1.1$$

$$V_{dc} = V_{0d}e^{-\frac{t}{\tau}}, \quad 4.1.2$$

Where V_{0c} and V_{dc} are the initial voltage for charging and discharging case respectively, V_s is the supply voltage, and τ is time constant which related to capacitance and resistance. By looking at these two equations, it is clear that these two equations have built connections between voltage and timing period.

Following this idea, it is possible to build an SH based SOP circuit. In Fig. 4.1.1, the simplified SOP circuit is presented.

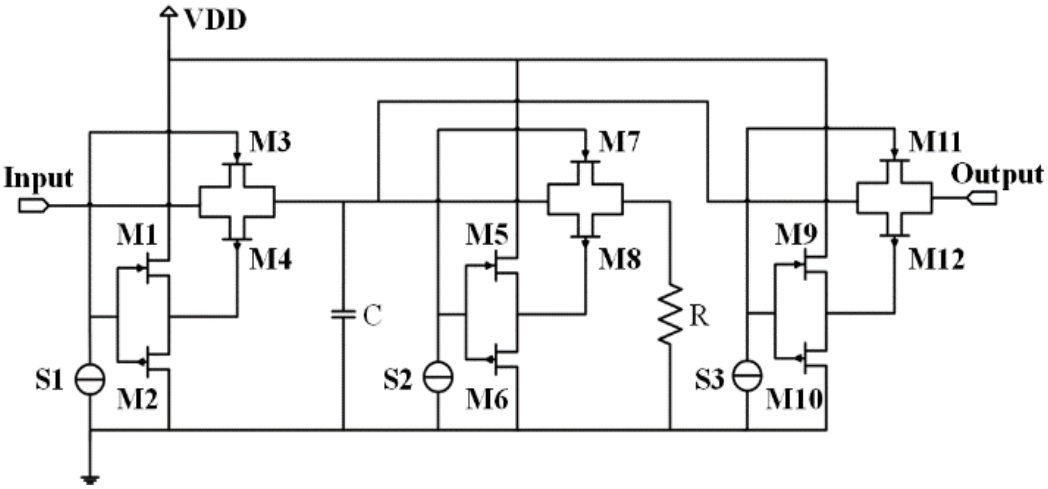


Fig. 4.1. 1 Simplified SOP circuit

The proposed SOP circuit could achieve ISI spike train based signal charging and discharging on capacitor easily. There are three external clock signals, i.e. S1, S2, and S3. S1 and S2 are used to control the charging and discharging process, and S3 will make the SOP generate proper output level signal. The phase difference between S1 and S2 is larger than 180^o which can guarantee the final output voltage level could be sent out through M11 and M12 before next ISI spike train come. S3 could hold the output voltage level until next ISI spike train finishing processing. M7, M8 and R would work together to reset the voltage on capacitor to low level. In practice, a reshape circuit can be adopted after this SOP circuit to make the output signal acceptable for the following circuits. A 3-spike ISI code example is illustrated in Fig. 4.1.2.

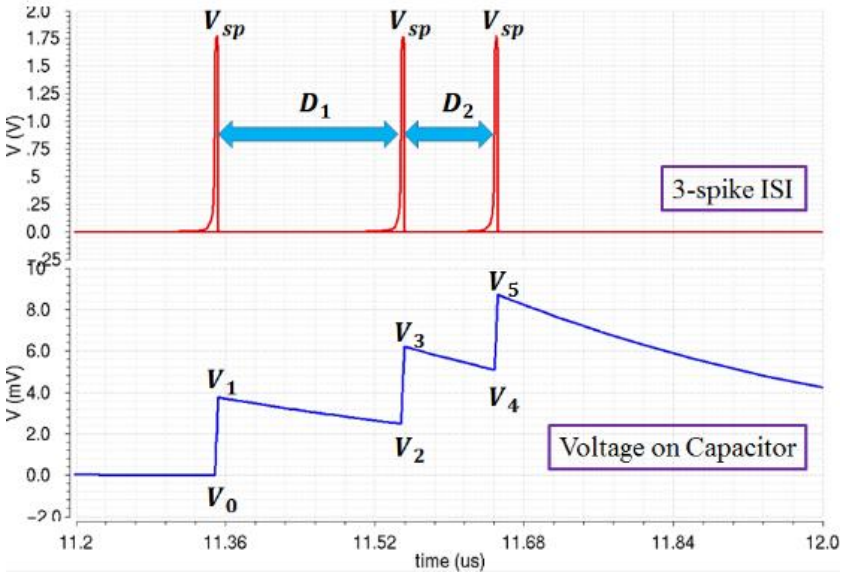


Fig. 4.1. 2 Three-spike ISI code SOP signal flows

As shown in Fig. 4.1.2, these intervals, i.e. D_1 and D_2, have been mapped into voltage level which is expressed with V_5. The relationship between ISI and output voltage is shown in equation 4.1.3.

$$\begin{cases} V_1 = V_{sp}\left(1 - e^{-\frac{\Delta t}{\tau}}\right), \\ V_2 = V_1 e^{-\frac{D_1}{\tau}}, \\ V_3 = V_2 + \left(V_{sp} - V_2\right)\left(1 - e^{-\frac{\Delta t}{\tau}}\right), \\ V_4 = V_3 e^{-\frac{D_2}{\tau}}, \\ V_5 = V_4 + \left(V_{sp} - V_4\right)\left(1 - e^{-\frac{\Delta t}{\tau}}\right). \end{cases} \quad 4.1.3$$

The V_5 can be further derived as

$$V_5 = (1-\alpha)e^{-\frac{D_2}{\tau}}\left[(1-\alpha)\alpha V_{sp} e^{-\frac{D_1}{\tau}} + V_{sp}\alpha\right], \quad 4.1.4$$

Where the α is $1 - e^{-\frac{\Delta t}{\tau}}$, Δt is typically around 8e-9, and τ is about 5.85e-10 level. By applying Taylor's expanding theory [108], equation 4.1.2 is able to re-write as

$$V_5 = \alpha(1-\alpha)V_{sp} e^{\frac{1}{\tau}}.(1-D_2)\left[(1-\alpha)e^{\frac{1}{\tau}}(1-D_1) + 1\right]. \quad 4.1.5$$

To show the relationship between voltage and ISI spike code clearer, equation 4.1.5 can be re-organized to

$$V_5 = \alpha V_{sp}[r_1(1 + D_1 D_2 - D_1 - D_2) + r_2(1 - D_2)], \quad 4.1.6$$

Where r_1 and r_2 represent r_2^2 and $(1-\alpha)e^{\frac{1}{\tau}}$ respectively. Equation 4.1.6 shows the design equation for 3-spike ISI code case. Following this trend, it is possible to derive the general design equation for SOP module, i.e. several ISI spike train's combing situations. Without generality, two paths ISI spike train combing case is taken into account. The signal charts are illustrated in Fig. 4.1.3.

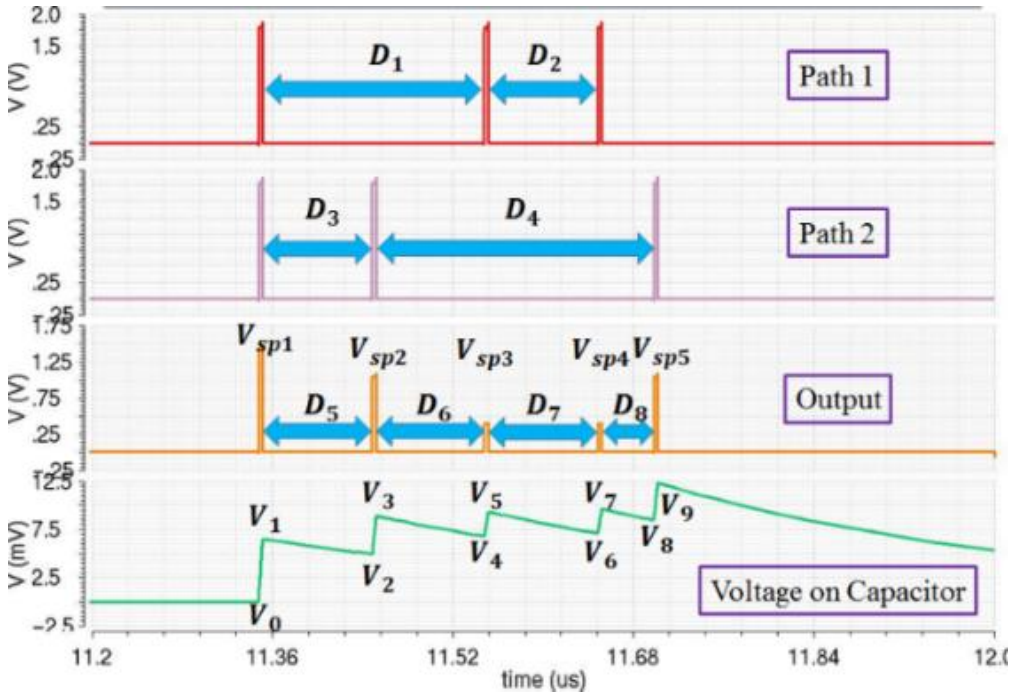


Fig. 4.1. 3 SOP signal charts of two paths ISI spike trains

In this case, it is desired to get the relationship between the sum of two paths ISI spikes and the output result V_9. By applying similar derivation process aforementioned for single three-spike case, the expression of V_9 is

$$V_9 = (1-\alpha)^4\alpha V_{sp1}e^{-\frac{D_5}{\tau}}e^{-\frac{D_6}{\tau}}e^{-\frac{D_7}{\tau}}e^{-\frac{D_8}{\tau}} + (1-\alpha)^3\alpha V_{sp2}e^{-\frac{D_6}{\tau}}e^{-\frac{D_7}{\tau}}e^{-\frac{D_8}{\tau}}$$
$$+ (1-\alpha)^2\alpha V_{sp3}e^{-\frac{D_7}{\tau}}e^{-\frac{D_8}{\tau}} + (1-\alpha)\alpha V_{sp4}e^{-\frac{D_8}{\tau}} + \alpha V_{sp5}. \quad 4.1.7$$

Equation 4.1.7 can be simplified as

$$V_9 = \sigma - k_4 D_8 - k_3 D_7 - k_2 D_6 - k_1 D_5. \quad 4.1.8$$

Within equation 4.1.8, each parameters, i.e. k_i and σ, are

$$\sigma = \beta^4\alpha V_{sp1}\theta^4 + \beta^3\alpha V_{sp2}\theta^3 + \beta^2\alpha V_{sp3}\theta^2 + \beta\alpha V_{sp4}\theta + \alpha V_{sp5},$$

$$k_4 = \beta^4\alpha V_{sp1}\theta^4 + \beta^3\alpha V_{sp2}\theta^3 + \beta^2\alpha V_{sp3}\theta^2 + \beta\alpha V_{sp4}\theta,$$

$$k_3 = \beta^4\alpha V_{sp1}\theta^4 + \beta^3\alpha V_{sp2}\theta^3 + \beta^2\alpha V_{sp3}\theta^2,$$

$$k_2 = \beta^4 \alpha V_{sp1} \theta^4 + \beta^3 \alpha V_{sp2} \theta^3,$$

$$k_1 = \beta^4 \alpha V_{sp1} \theta^4,$$

$$\beta = 1 - \alpha, and\ \theta = e^{\frac{1}{\tau}}.$$

It is clear that equation 4.1.8 has demonstrated that the proposed SOP could achieve sum of product of ISI spike trains successfully.

4.1.2 ISI Extractor

The second part of SH ISI encoder is ISI extractor. This module would extract each interval out directly and map them to voltage level signal. The whole ISI extractor is constructed with an array of ISI extracting units. Two ISI extracting units would work together to extract one interval out. For example, two intervals need three ISI extracting units to finish interval extracting work. The simplified single ISI extracting unit circuit is shown in Fig. 4.1.4.

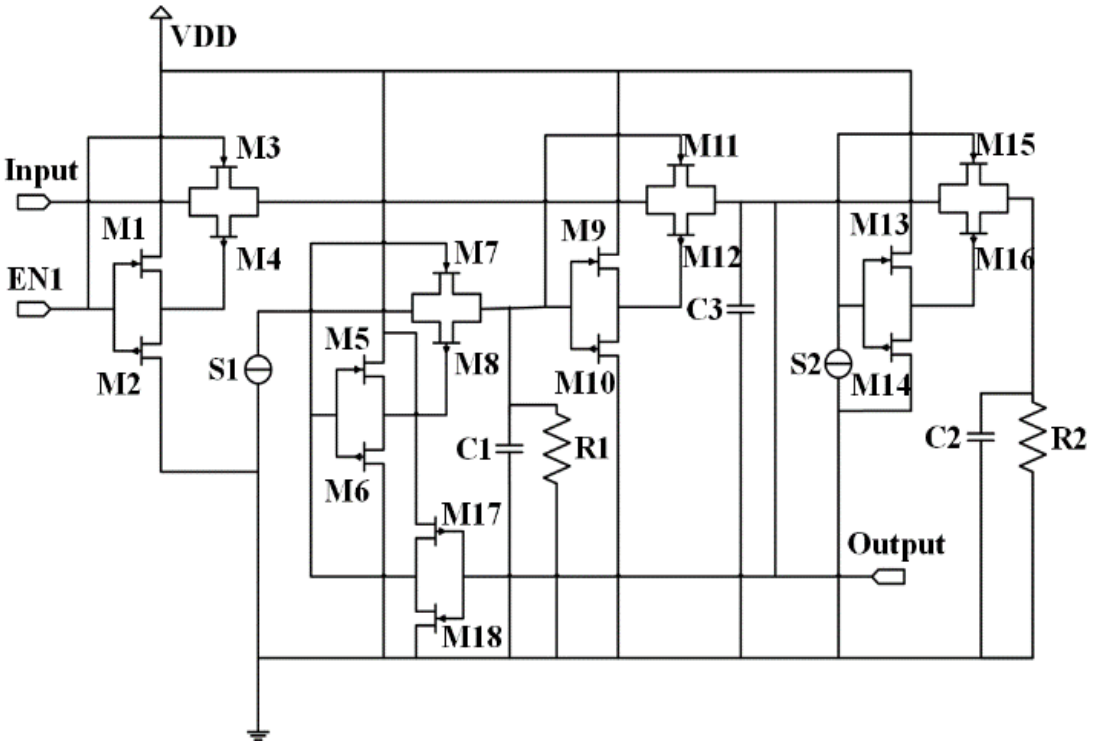


Fig. 4.1. 4 Simplified ISI extracting unit circuit

In Fig. 4.1.4, except charging and discharging control signals S1 and S2, there is an enabling signal EN1 that used to serve as trigger signal to make the whole circuit work or not. As mentioned above, the minimum working set requirement is two. Therefore, it is better to demonstrate the whole signal flow of the proposed ISI extractor, which is illustrated in Fig. 4.1.5.

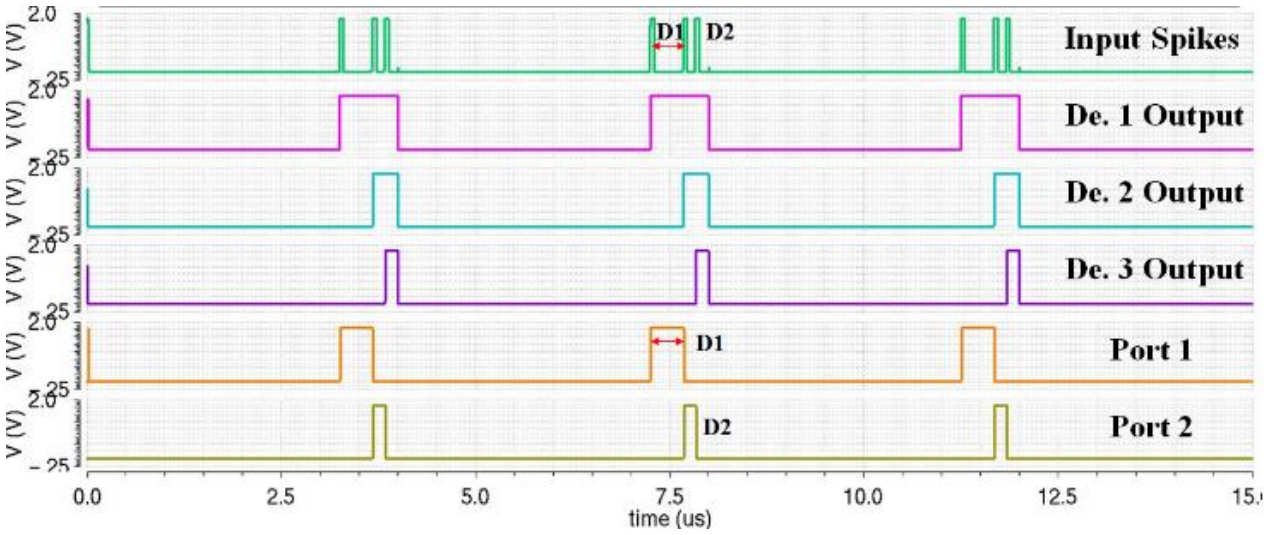


Fig. 4.1. 5 Signal flow of the proposed ISI extractor

As shown in Fig. 4.1.5, intervals D1 and D2 has been extracted out. The output results are shown in port 1 and port 2. The relationship between output of each ISI extracting unit and final output can be described as

$$Port_1 = De_i \; XOR \; De_{i+1}. \tag{4.1.9}$$

4.1.3 Signal Integrating Scheme

After finish design SOP part and ISI extractor part, it is required to integrate all these outputs together to get the final result. For SOP part, the output voltage level is in 10mV level which need a buffer, e.g. amplifier, to shift it up to a higher voltage, i.e. V_{sop}. For the ISI extractor, some SH circuits are required to convert square waves to different voltage levels, i.e. D_i. Then a combiner would combine these D_i into one voltage value, e.g. V_{ext}. Finally, V_{sop} and V_{ext} would multiply with each other to get the final result which can be expressed as

$$V_{out} = V_{sop} \cdot V_{ext} = V_{sop} \cdot \sum_1^{n-1} W_i V_{ext,i}\,, \tag{4.1.10}$$

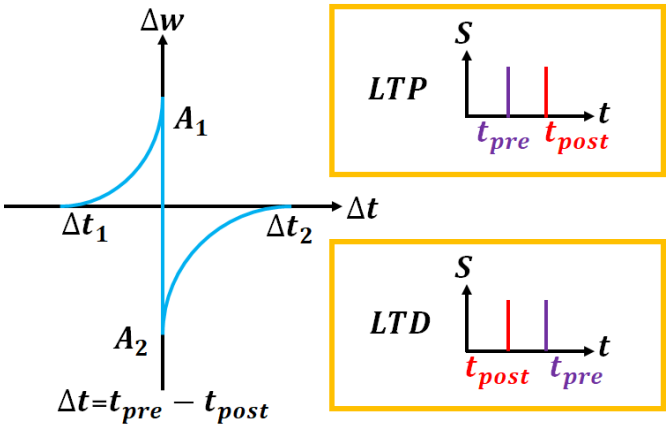
Where W_i is the weight, which implemented with resistor, that tuning each path's extracted ISI signal.

4.2 STDP Based ISI Decoder

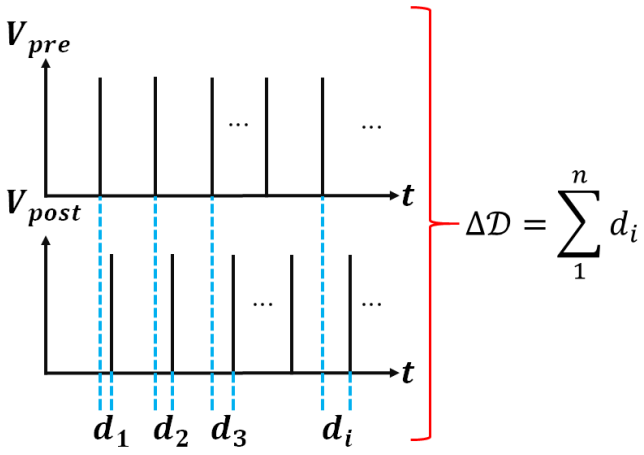
4.2.1 Decoder Circuit Design

How to extract inter-spike intervals out and represent these intervals with proper signal format is one of the most challenge part in building an efficient neuromorphic computing system.

In [109], a mixed-signal decoder is presented which constructed with a lot of sample & hold modules. The key idea of this decoder is adopting high speed sampling circuits to treat each spike as narrow pulse. It may work incorrectly if high frequency spike train is applied. Furthermore, this decoder does not have any spike adaptive capability for different length spike trains. It would also occupy too many die areas for practice use. In spiking neural network research field, STDP has been widely adopted to serve as the weight update strategy. In this dissertation, STDP is used to construct dynamic ISI decoder. In Fig. 4.2.1(a), the fundamental STDP scheme is presented.



(a)



(b)

Fig. 4.2. 1 (a) STDP function; (b) ISI decoding scheme

As shown in Fig. 4.2.1(a), the synaptic weight's variation trend is determined by the correlation of these temporal positions between pre-spike and post-spike. This correlation relationship is suitable for building ISI decoder if one path spike train, e.g. post-spike train, keeps constant. In Fig. 4.2.1(b), a general decoding example is provided.

As shown in Fig. 4.2.1(b), the difference between each spike pair, i.e. including one pre-spike and one post-spike, could be expressed as

$$d_i = D_{ini} + i\Delta D, \tag{4.2.1}$$

where D_{ini} is the initial time difference ($D_{ini} = d_1$), and ΔD is the $i^{th}(i \neq 1)$ difference among each spike pair. By applying nonlinear function as the mask shield, the final output could be expressed as

$$V_{out} = \sum_1^i W(D_{ini} + i\Delta D), \tag{4.2.2}$$

where the $W()$ is the nonlinear function such as the exponential function which has been widely adopted [110, 111].

Due to the dual-path synchronous principle, the best candidate design topology is symmetric structure. In this dissertation, the simplified ISI decoder circuit is illustrated in Fig. 4.2.2.

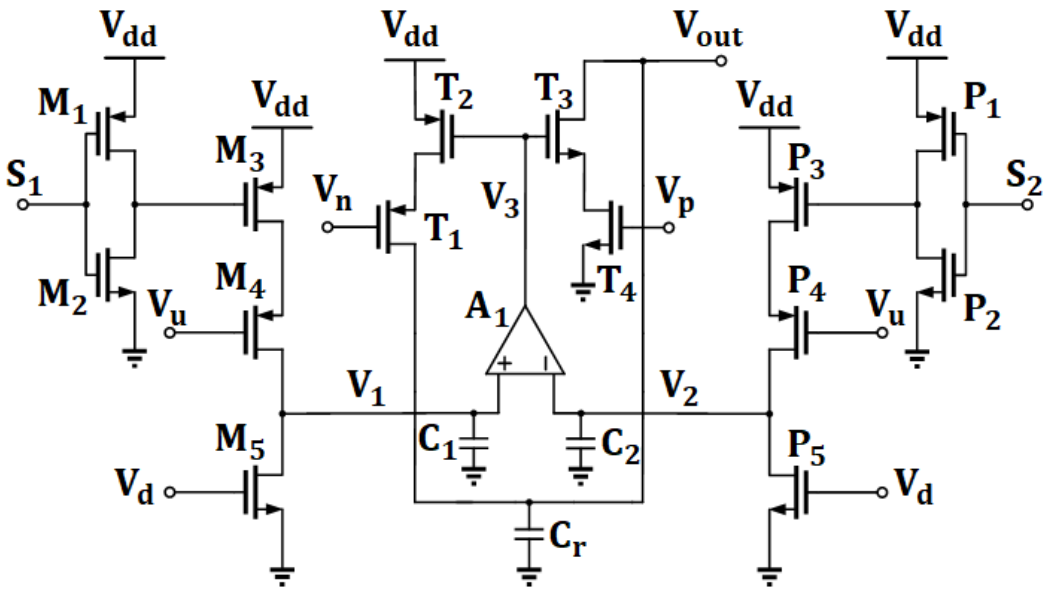


Fig. 4.2. 2 STDP based ISI decoder circuit

As shown in Fig. 4.2.2, the whole decoder can be classified into three parts including ISI spike train input part (marked with M_i), reference spike train input part (marked with P_i), and the combining part (marked with T_i).

Since these two input terminals are designed with P-type, the input spike must be pre-processed by inverter which is built with M_1 and M_2. Then these inverted spikes would be converted into current value by transistor M_3. The external control voltages V_u and V_d would be applied on M_4 and M_5 to tune the charging speed for capacitor C_1. The reference spike processing part has similar operation process. Comparing the voltages on C_1 and C_2, comparator A_1 would either drive T_2 or T_3 to charge or discharge capacitor C_r. The output voltage of A_1, hence, could be expressed as

$$V_3 = \begin{cases} V_{dd}, & V_1 - V_2 > 0 \\ 0.5V_{dd}, & V_1 - V_2 = 0 \\ 0, & V_1 - V_2 < 0 \end{cases}. \qquad 4.2.3$$

Once the potential difference of V_1 and V_2 is compared by A_1, T_1 and T_2 are used to charge C_r, while T_3 and T_4 are used to discharge C_r. Therefore, the output voltage of the decoder could be expressed as

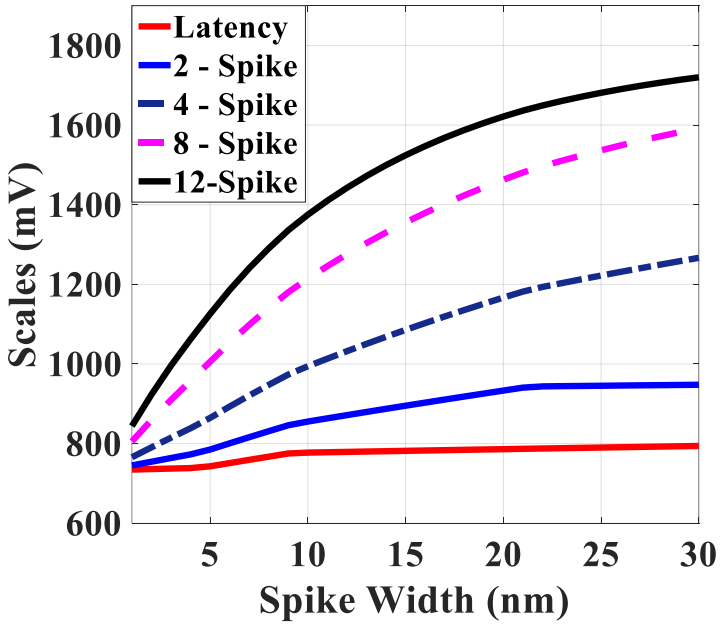
$$V_{out} = \begin{cases} V_{ex}(1 - \exp(-t_n/\tau_n)) \\ V_{ex}\exp(D_{ref} - t_n/\tau_p) \end{cases}, \quad 4.2.4$$

where V_{ex} is the voltage when capacitor changes its operation behavior, i.e. charging to discharging or discharging to charging, D_{ref} is the ISI whole period of the reference ISI spike train, t_n is controlled by V_3, τ_n and τ_p are timing constants that could be expressed as

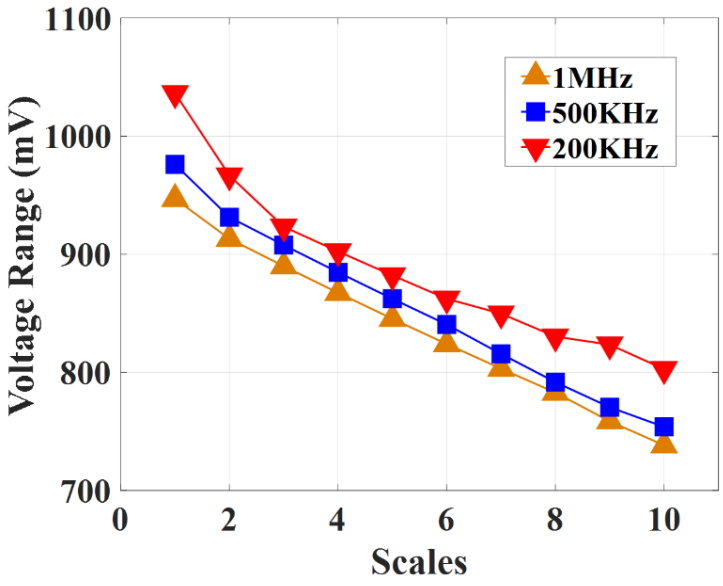
$$\begin{cases} \tau_n = C_r(W_{T4}/L_{T4})I_{oN}exp(V_p q/(nkT)) \\ \tau_n = C_r(W_{T1}/L_{T1})I_{oP}exp(V_n q/(nkT)) \end{cases}. \quad 4.2.5$$

4.2.2 Performance Evaluations

The linearity and the signal range are two important parameters which determine the whole performance of the decoder. Spike width adaptation is another capability which used to evaluate the dynamic performance of the decoder. In this section, several simulation results are provided to evaluate the performance of the decoder, which are illustrated in Fig. 4.2.3.



(a)



(b)

Fig. 4.2. 3 Simulation results of (a) the relationship between spike width and output signal's scales; (b) the linearity of the output signal

As shown in Fig. 4.2.3(a), the larger ISI spike train leading to higher signal output range; and the wider width would achieve larger range.

In this dissertation, 10-level case is used. As shown in Fig. 4.2.3 (b), the outputs are in linearity distribution under difference frequency and different scales cases. The aforementioned output signal could be applied on post-processing system without extra interface circuits. Furthermore, comparing with the design in [5], the proposed ISI encoder has the ability to generate multi-level output voltage, which could provide more accurate information.

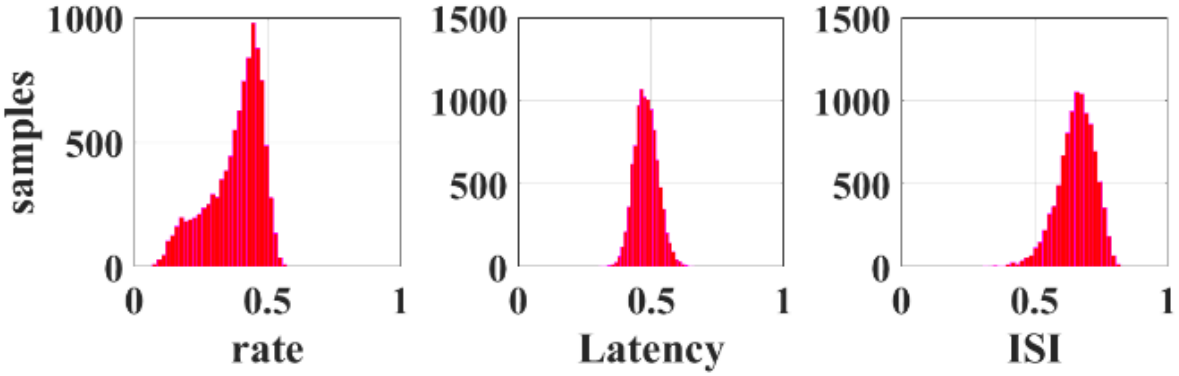


Fig. 4.2. 4 Distributions of three encoding/decoding schemes

Finally, it is required to make comparison among three main encoding schemes, i.e. the rate encoding, the latency encoding and the ISI encoding. Two-dimensional correlation algorithm is adopted to serve as the comparison strategy to evaluate these encodings' performance. Detailed expression of the 2D-correlation is expressed as [112].

$$corr = \frac{\sum_m \sum_n (V_{in,mn} - \overline{V_{in}})(V_{out,mn} - \overline{V_{out}})}{\sqrt{(\sum_m \sum_n (V_{in,mn} - \overline{V_{in}})^2)(\sum_m \sum_n (V_{out,mn} - \overline{V_{out}})^2)}}, \quad 4.2.6$$

where $V_{in,mn}$ is the normalized value of each pixel, V_{out} represents the output voltage, $\overline{V_{in}}$ and $\overline{V_{out}}$ represent the voltage values of the input and output. The simulation result, which based on CIFAR-10 dataset is illustrated in Fig. 4.2.4. As shown in Fig. 4.2.4, ISI encoding/decoding processing link has better performance than the rate encoding/decoding and the latency encoding/decoding.

Conclusion

In this dissertation, spike-based data processing links, including analog-to-spike encoder and spike-to-analog decoder, have been discussed. Among these date links, analog-rate date link is the simplest architecture that has been widely adopted. In my work, a two-path leakage LIF neuron is designed and fabricated. I also designed a multi-stage neuron to verify the merit of the proposed module design methodology for neuromorphic computing systems. The aforementioned multi-stage neuron has been used to build latency encoder which could achieve accurate latency encoding results.

My second stage of Ph.D. work was focusing on temporal encoder and decoder design. During this period, two versions of temporal encoder, or ISI encoder, have been developed and fabricated under standard CMOS process successfully. The first-generation ISI encoder is based on neuron parallel firing mechanism, while the second-generation ISI encoder works in signal-iteration condition. Comparing with these two kinds of ISI encoders, parallel ISI encoder has higher speed and iteration ISI encoder has lower power consumption and higher information density. Furthermore, iteration ISI encoder's output spike train has integrated with self-verification ability. Both these two ISI encoders have been fabricated with standard CMOS technology. An SH ISI decoder was also invented which could transform ISI spike code back to voltage level signal. Two decoding strategies, i.e. ISI sum of product and ISI extracting, were combined together to make the proposed SH ISI encoder achieving both accurate decoding ability and robustness.

There were two innovations during my Ph.D.'s third stage, which including partial-signal ISI encoder design and STDP-based ISI decoder design. The proposed partial-signal ISI encoder adopted the same iteration scheme that second-generation ISI encoder used. However, during encoding process, only partial-signal, i.e. current spike, participating iteration process. It has been proved that the proposed partial-signal ISI encoder could achieve the same encoding accuracy by consuming less power. Furthermore, this partial-signal ISI encoder has higher scaling-up ability than full-signal iteration ISI encoder. My second work during this period is designing STDP-based ISI decoder that could transform ISI spike code back to multi-scale voltage level signal. Comparing with traditional STDP-based applications, e.g. bi-stable decoding, the proposed ISI decoder could achieve multi-scale stable output. Both the partial-signal ISI encoder and STDP-based ISI decoder

have been designed with standard CMOS technology. My advisor has scheduled chip tape out for these two designs.

In my future research, there would be two aspects including bottom-level circuits, i.e. more robust encoding/decoding circuits design, and system level implementation. In my research, the spike train's working range is from 0Hz to 1MHz. However, such kind of speed is not able to make information transferring more efficient. It is required to further modify the encoding schemes so that higher information density could be achieved. For the system level's design, it is significant to build a full functional system that using ISI code to sever as the whole system's signal format.

Reference

[1] G. J. Myers, *Advances in computer architecture*. John Wiley & Sons, Inc., 1982.

[2] J. L. Hennessy and D. A. Patterson, *Computer architecture: a quantitative approach*. Elsevier, 2011.

[3] J. P. Hayes, *Computer architecture and organization*. McGraw-Hill, Inc., 2002.

[4] K. Hwang and N. Jotwani, *Advanced Computer Architecture, 3e*. McGraw-Hill Education, 2016.

[5] M. Campbell-Kelly, *Computer, Student Economy Edition: A History of the Information Machine*. Routledge, 2018.

[6] C. M. Kelty, *Two bits: The cultural significance of free software*. Duke University Press, 2008.

[7] C. S. Lent *et al.*, "Molecular cellular networks: A non von Neumann architecture for molecular electronics," in *2016 IEEE International Conference on Rebooting Computing (ICRC)*, 2016, pp. 1-7: IEEE.

[8] F. Akopyan *et al.*, "Truenorth: Design and tool flow of a 65 mw 1 million neuron programmable neurosynaptic chip," vol. 34, no. 10, pp. 1537-1557, 2015.

[9] A. Cassidy *et al.*, "TrueNorth: A high-performance, low-power neurosynaptic processor for multi-sensory perception, action, and cognition," in *Proceedings of the Government Microcircuits Applications & Critical Technology Conference, Orlando, FL, USA*, 2016, pp. 14-17.

[10] C. D. Schuman *et al.*, "A survey of neuromorphic computing and neural networks in hardware," 2017.

[11] C. Zhao and J. Huang, "A new high performance bandgap reference," in *2011 International Conference on Electronics, Communications and Control (ICECC)*, 2011, pp. 64-66: IEEE.

[12] C. Zhao, J. Liu, F. Shen, and Y. Yi, "Low power CMOS power amplifier design for RFID and the Internet of Things," *Computers & Electrical Engineering*, vol. 52, pp. 157-170, 2016.

[13] S. Samarasinghe, *Neural networks for applied sciences and engineering: from fundamentals to complex pattern recognition*. Auerbach publications, 2016.

[14] C.-h. Chen, *Handbook of pattern recognition and computer vision*. World Scientific, 2015.

[15] C. Zhao *et al.*, "SoC FPAA Hardware Implementation of a VMM+ WTA Embedded Learning Classifier..... S. Shah and J. Hasler 28 Analog Spike-Timing-Dependent Resistive Crossbar Design for Brain Inspired Computing."

[16] W. Danesh, C. Zhao, B. T. Wysocki, M. J. Medley, N. N. Thawdar, and Y. Yi, "Channel estimation in wireless OFDM systems using reservoir computing," in *2015 IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA)*, 2015, pp. 1-5: IEEE.

[17] J. Li, L. Liu, C. Zhao, K. Hamedani, R. Atat, and Y. Yi, "Enabling sustainable cyber physical security systems through neuromorphic computing," *IEEE Transactions on Sustainable Computing*, vol. 3, no. 2, pp. 112-125, 2017.

[18] C. J. P. o. t. I. Mead, "Neuromorphic electronic systems," vol. 78, no. 10, pp. 1629-1636, 1990.

[19] G. Indiveri *et al.*, "Neuromorphic silicon neuron circuits," vol. 5, p. 73, 2011.

[20] C. Xie, J. Tan, M. Chen, Y. Yi, L. Peng, and X. Fu, "Emerging technology enabled energy-efficient GPGPUs register file," *Microprocessors and Microsystems*, vol. 50, pp. 175-188, 2017.
[21] G. Indiveri, B. Linares-Barranco, R. Legenstein, G. Deligeorgis, and T. J. N. Prodromakis, "Integration of nanoscale memristor synapses in neuromorphic computing architectures," vol. 24, no. 38, p. 384010, 2013.
[22] M. Hu *et al.*, "Memristor crossbar-based neuromorphic computing system: A case study," vol. 25, no. 10, pp. 1864-1878, 2014.
[23] S. H. Jo, T. Chang, I. Ebong, B. B. Bhadviya, P. Mazumder, and W. J. N. l. Lu, "Nanoscale memristor device as synapse in neuromorphic systems," vol. 10, no. 4, pp. 1297-1301, 2010.
[24] H. An, J. Li, Y. Li, X. Fu, and Y. Yi, "Three dimensional memristor-based neuromorphic computing system and its application to cloud robotics," *Computers & Electrical Engineering*, vol. 63, pp. 99-113, 2017.
[25] M. A. Ehsan, Z. Zhou, and Y. Yi, "Modeling and analysis of neuronal membrane electrical activities in 3d neuromorphic computing system," in *2017 IEEE International Symposium on Electromagnetic Compatibility & Signal/Power Integrity (EMCSI)*, 2017, pp. 745-750: IEEE.
[26] M. A. Ehsan, H. An, Z. Zhou, and Y. Yi, "Adaptation of enhanced TSV capacitance as membrane property in 3D brain-inspired computing system," in *2017 54th ACM/EDAC/IEEE Design Automation Conference (DAC)*, 2017, pp. 1-6: IEEE.
[27] M. A. Ehsan, Z. Zhou, and Y. Yi, "Neuromorphic 3D integrated circuit: A hybrid, reliable and energy efficient approach for next generation computing," in *Proceedings of the on Great Lakes Symposium on VLSI 2017*, 2017, pp. 221-226: ACM.
[28] Y. Zhang *et al.*, "Spintronics for low-power computing," in *Proceedings of the conference on Design, Automation & Test in Europe*, 2014, p. 303: European Design and Automation Association.
[29] S. G. Ramasubramanian, R. Venkatesan, M. Sharad, K. Roy, and A. Raghunathan, "SPINDLE: SPINtronic deep learning engine for large-scale neuromorphic computing," in *Proceedings of the 2014 international symposium on Low power electronics and design*, 2014, pp. 15-20: ACM.
[30] D. Kuzum, R. G. Jeyasingh, B. Lee, and H.-S. P. J. N. l. Wong, "Nanoelectronic programmable synapses based on phase change materials for brain-inspired computing," vol. 12, no. 5, pp. 2179-2186, 2011.
[31] G. Rozenberg, T. Bäck, and J. N. Kok, *Handbook of natural computing*. Springer, 2012.
[32] M. A. Ehsan, Z. Zhou, and Y. Yi, "Hybrid three-dimensional integrated circuits: A viable solution for high efficiency neuromorphic computing," in *2017 International Symposium on VLSI Design, Automation and Test (VLSI-DAT)*, 2017, pp. 1-2: IEEE.
[33] W. Zhao, G. Agnus, V. Derycke, A. Filoramo, J. Bourgoin, and C. J. N. Gamrat, "Nanotube devices based crossbar architecture: toward neuromorphic computing," vol. 21, no. 17, p. 175202, 2010.
[34] S. Park *et al.*, "Nanoscale RRAM-based synaptic electronics: toward a neuromorphic computing device," vol. 24, no. 38, p. 384009, 2013.
[35] T. Potok, C. Schuman, R. Patton, T. Hylton, H. Li, and R. Pino, "Neuromorphic Computing, Architectures, Models, and Applications. A Beyond-CMOS Approach to

Future Computing, June 29-July 1, 2016, Oak Ridge, TN," USDOE Office of Science (SC)(United States). Advanced Scientific Computing …2016.
[36] C. Zhao, K. Hamedani, J. Li, and Y. Yi, "Analog Spike-Timing-Dependent Resistive Crossbar Design for Brain Inspired Computing," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 8, no. 1, pp. 38-50, 2017.
[37] J. Li, C. Zhao, K. Hamedani, and Y. Yi, "Analog hardware implementation of spike-based delayed feedback reservoir computing system," in *2017 International Joint Conference on Neural Networks (IJCNN)*, 2017, pp. 3439-3446: IEEE.
[38] A. R. Marathe, V. J. Lawhern, D. Wu, D. Slayback, B. J. J. I. T. o. N. S. Lance, and R. Engineering, "Improved neural signal classification in a rapid serial visual presentation task using active learning," vol. 24, no. 3, pp. 333-343, 2015.
[39] M. S. J. N. C. i. N. S. Lewicki, "A review of methods for spike sorting: the detection and classification of neural action potentials," vol. 9, no. 4, pp. R53-R78, 1998.
[40] C. Zhao, B. T. Wysocki, Y. Liu, C. D. Thiem, N. R. McDonald, and Y. Yi, "Spike-time-dependent encoding for neuromorphic processors," *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 12, no. 3, p. 23, 2015.
[41] C. Zhao, J. Li, and Y. Yi, "Making neural encoding robust and energy efficient: an advanced analog temporal encoder for brain-inspired computing systems," in *Proceedings of the 35th International Conference on Computer-Aided Design*, 2016, p. 115: ACM.
[42] A. Borst and F. E. J. N. n. Theunissen, "Information theory and neural coding," vol. 2, no. 11, p. 947, 1999.
[43] C. Zhao *et al.*, "Energy efficient spiking temporal encoder design for neuromorphic computing systems," *IEEE Transactions on Multi-Scale Computing Systems*, vol. 2, no. 4, pp. 265-276, 2016.
[44] A. Joubert, B. Belhadj, O. Temam, and R. Héliot, "Hardware spiking neurons design: Analog or digital?," in *The 2012 International Joint Conference on Neural Networks (IJCNN)*, 2012, pp. 1-5: IEEE.
[45] E. J. J. o. m. b. Oja, "Simplified neuron model as a principal component analyzer," vol. 15, no. 3, pp. 267-273, 1982.
[46] C. Zhao, W. Danesh, B. T. Wysocki, and Y. Yi, "Neuromorphic encoding system design with chaos based CMOS analog neuron," in *2015 IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA)*, 2015, pp. 1-6: IEEE.
[47] S. R. Deiss, R. J. Douglas, and A. M. J. P. n. n. Whatley, "A pulse-coded communications infrastructure for neuromorphic systems," pp. 157-178, 1999.
[48] C. D. Brody, A. Hernández, A. Zainos, and R. J. C. c. Romo, "Timing and neural encoding of somatosensory parametric working memory in macaque prefrontal cortex," vol. 13, no. 11, pp. 1196-1207, 2003.
[49] P. A. Merolla *et al.*, "A million spiking-neuron integrated circuit with a scalable communication network and interface," vol. 345, no. 6197, pp. 668-673, 2014.
[50] E. M. J. I. T. o. n. n. Izhikevich, "Simple model of spiking neurons," vol. 14, no. 6, pp. 1569-1572, 2003.
[51] X. Qi, X. Guo, and J. G. Harris, "A time-to-first spike CMOS imager," in *2004 IEEE International Symposium on Circuits and Systems (IEEE Cat. No. 04CH37512)*, 2004, vol. 4, pp. IV-824: IEEE.
[52] X. Guo, X. Qi, and J. G. J. I. S. J. Harris, "A time-to-first-spike CMOS image sensor," vol. 7, no. 8, pp. 1165-1175, 2007.

[53] C. Shoushun and A. J. I. T. o. V. L. S. I. S. Bermak, "Arbitrated time-to-first spike CMOS image sensor with on-chip histogram equalization," vol. 15, no. 3, pp. 346-357, 2007.
[54] N. Ding and J. Z. J. J. o. n. Simon, "Neural coding of continuous speech in auditory cortex during monaural and dichotic listening," vol. 107, no. 1, pp. 78-89, 2011.
[55] N. Ding, L. Melloni, H. Zhang, X. Tian, and D. J. N. n. Poeppel, "Cortical tracking of hierarchical linguistic structures in connected speech," vol. 19, no. 1, p. 158, 2016.
[56] F. Theunissen and J. P. J. J. o. c. n. Miller, "Temporal encoding in nervous systems: a rigorous definition," vol. 2, no. 2, pp. 149-162, 1995.
[57] Y. Zheng, S. Li, R. Yan, H. Tang, K. C. J. I. t. o. n. n. Tan, and l. systems, "Sparse temporal encoding of visual features for robust object recognition by spiking neurons," vol. 29, no. 12, pp. 5823-5833, 2018.
[58] Y. Yi, "Neuron Design in Neuromorphic Computing Systems and Its Application in Wireless Communications," The University of Kansas Center for Research, Inc. Lawrence2017.
[59] K. Bai and Y. Yi, "Opening the "Black Box" of Silicon Chip Design in Neuromorphic Computing," in *Bio-Inspired Technology*: IntechOpen, 2019.
[60] C. Zhao, Y. Yi, J. Li, X. Fu, and L. J. I. T. o. V. L. S. I. S. Liu, "Interspike-interval-based analog spike-time-dependent encoder for neuromorphic processors," vol. 25, no. 8, pp. 2193-2205, 2017.
[61] T. Zhong *et al.*, "Photon-efficient quantum key distribution using time–energy entanglement with high-dimensional encoding," vol. 17, no. 2, p. 022002, 2015.
[62] Y. Yi *et al.*, "FPGA based spike-time dependent encoder and reservoir design in neuromorphic computing processors," *Microprocessors and Microsystems*, vol. 46, pp. 175-183, 2016.
[63] C. Zhao, B. T. Wysocki, Y. Liu, C. D. Thiem, N. R. McDonald, and Y. J. A. J. o. E. T. i. C. S. Yi, "Spike-time-dependent encoding for neuromorphic processors," vol. 12, no. 3, p. 23, 2015.
[64] G. M. Shepherd, *Foundations of the neuron doctrine*. Oxford University Press, 2015.
[65] I. B. Levitan and L. K. Kaczmarek, *The neuron: cell and molecular biology*. Oxford University Press, USA, 2015.
[66] W. Gerstner and W. M. Kistler, *Spiking neuron models: Single neurons, populations, plasticity*. Cambridge university press, 2002.
[67] J. G. Nicholls, A. R. Martin, B. G. Wallace, and P. A. Fuchs, *From neuron to brain*. Sinauer Associates Sunderland, MA, 2001.
[68] H.-H. Chang, H. Song, Y. Yi, J. Zhang, H. He, and L. Liu, "Distributive Dynamic Spectrum Access Through Deep Reinforcement Learning: A Reservoir Computing-Based Approach," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1938-1948, 2018.
[69] C. R. Laing and C. C. J. J. o. c. n. Chow, "A spiking neuron model for binocular rivalry," vol. 12, no. 1, pp. 39-53, 2002.
[70] C. Zhao, J. Li, L. Liu, L. S. Koutha, J. Liu, and Y. Yi, "Novel spike based reservoir node design with high performance spike delay loop," in *Proceedings of the 3rd ACM International Conference on Nanoscale Computing and Communication*, 2016, p. 14: ACM.
[71] K. Bai, J. Li, K. Hamedani, and Y. Yi, "Enabling An New Era of Brain-inspired Computing: Energy-efficient Spiking Neural Network with Ring Topology," in *2018 55th ACM/ESDA/IEEE Design Automation Conference (DAC)*, 2018, pp. 1-6: IEEE.

[72] J. Li, K. Bai, L. Liu, and Y. Yi, "A deep learning based approach for analog hardware implementation of delayed feedback reservoir computing system," in *2018 19th International Symposium on Quality Electronic Design (ISQED)*, 2018, pp. 308-313: IEEE.

[73] C. Zhao, Y. Yi, J. Li, X. Fu, and L. Liu, "Interspike-interval-based analog spike-time-dependent encoder for neuromorphic processors," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 8, pp. 2193-2205, 2017.

[74] A. N. J. B. c. Burkitt, "A review of the integrate-and-fire neuron model: I. Homogeneous synaptic input," vol. 95, no. 1, pp. 1-19, 2006.

[75] Y. Yi, "Analog Integrated Circuit Design for Spike Time Dependent Encoder and Reservoir in Reservoir Computing Processors," University of Kansas Center for Research, Inc. Lawrence United States2018.

[76] Y.-H. Liu and X.-J. J. J. o. c. n. Wang, "Spike-frequency adaptation of a generalized leaky integrate-and-fire model neuron," vol. 10, no. 1, pp. 25-45, 2001.

[77] S. Mosleh, L. Liu, C. Sahin, Y. R. Zheng, and Y. Yi, "Brain-inspired wireless communications: Where reservoir computing meets MIMO-OFDM," *IEEE transactions on neural networks and learning systems*, no. 99, pp. 1-15, 2017.

[78] M. A. Ehsan, H. An, Z. Zhou, and Y. Yi, "A Novel Approach for Using TSVs As Membrane Capacitance in Neuromorphic 3-D IC," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 8, pp. 1640-1653, 2017.

[79] P. Lansky and S. J. B. c. Ditlevsen, "A review of the methods for signal estimation in stochastic diffusion leaky integrate-and-fire neuronal models," vol. 99, no. 4-5, p. 253, 2008.

[80] A. D. Rast, F. Galluppi, X. Jin, and S. B. Furber, "The leaky integrate-and-fire neuron: A platform for synaptic model exploration on the spinnaker chip," in *The 2010 International Joint Conference on Neural Networks (IJCNN)*, 2010, pp. 1-8: IEEE.

[81] L. Abbott and T. B. Kepler, "Model neurons: from hodgkin-huxley to hopfield," in *Statistical mechanics of neural networks*: Springer, 1990, pp. 5-18.

[82] E. M. J. I. t. o. n. n. Izhikevich, "Which model to use for cortical spiking neurons?," vol. 15, no. 5, pp. 1063-1070, 2004.

[83] M. Chu *et al.*, "Neuromorphic hardware system for visual pattern recognition with memristor array and CMOS neuron," vol. 62, no. 4, pp. 2410-2419, 2014.

[84] J.-s. Seo *et al.*, "A 45nm CMOS neuromorphic chip with a scalable architecture for learning in networks of spiking neurons," in *2011 IEEE Custom Integrated Circuits Conference (CICC)*, 2011, pp. 1-4: IEEE.

[85] B. Razavi, *Design of analog CMOS integrated circuits*. 清华大学出版社有限公司, 2005.

[86] P. R. Gray, P. Hurst, R. G. Meyer, and S. Lewis, *Analysis and design of analog integrated circuits*. Wiley, 2001.

[87] C. Daussy *et al.*, "Direct determination of the Boltzmann constant by an optical method," vol. 98, no. 25, p. 250801, 2007.

[88] R. C. Weast, M. J. Astle, and W. H. Beyer, *CRC handbook of chemistry and physics*. CRC press Boca Raton, FL, 1988.

[89] R. J. Baker, *CMOS: circuit design, layout, and simulation*. Wiley-IEEE press, 2019.

[90] R. H. J. I. J. o. s. a. i. c. Walden, "Analog-to-digital converter survey and analysis," vol. 17, no. 4, pp. 539-550, 1999.

[91] D. F. Hoeschele, *Analog-to-digital and digital-to-analog conversion techniques*. Wiley New York, 1994.
[92] M. A. Ehsan, Z. Zhou, and Y. Yi, "Modeling and optimization of TSV for crosstalk mitigation in 3D neuromorphic system," in *2016 IEEE International Symposium on Electromagnetic Compatibility (EMC)*, 2016, pp. 621-626: IEEE.
[93] K. Bai, Q. An, and Y. Yi, "Deep-DFR: A Memristive Deep Delayed Feedback Reservoir Computing System with Hybrid Neural Network Topology," in *Proceedings of the 56th Annual Design Automation Conference 2019*, 2019, p. 54: ACM.
[94] R. Atat, L. Liu, J. Wu, J. Ashdown, and Y. Yi, "Green massive traffic offloading for cyber-physical systems over heterogeneous cellular networks," *Mobile Networks and Applications*, pp. 1-9, 2018.
[95] S. Mosleh, C. Sahin, L. Liu, R. Zheng, and Y. Yi, "An energy efficient decoding scheme for nonlinear MIMO-OFDM network using reservoir computing," in *2016 International Joint Conference on Neural Networks (IJCNN)*, 2016, pp. 1166-1173: IEEE.
[96] C. Zhao *et al.*, "Energy efficient spiking temporal encoder design for neuromorphic computing systems," vol. 2, no. 4, pp. 265-276, 2016.
[97] M. J. Tovee and E. T. J. V. c. Rolls, "Information encoding in short firing rate epochs by single neurons in the primate temporal visual cortex," vol. 2, no. 1, pp. 35-58, 1995.
[98] C. Zhao, J. Li, H. An, and Y. Yi, "Energy efficient analog spiking temporal encoder with verification and recovery scheme for neuromorphic computing systems," in *2017 18th International Symposium on Quality Electronic Design (ISQED)*, 2017, pp. 138-143: IEEE.
[99] T. J. Gawne, T. W. Kjaer, and B. J. J. J. o. n. Richmond, "Latency: another potential code for feature binding in striate cortex," vol. 76, no. 2, pp. 1356-1360, 1996.
[100] A. Linares-Barranco, M. Oster, D. Cascado, G. Jiménez, A. Civit, and B. J. N. Linares-Barranco, "Inter-spike-intervals analysis of AER Poisson-like generator hardware," vol. 70, no. 16-18, pp. 2692-2700, 2007.
[101] R. J. Van de Plassche, *CMOS integrated analog-to-digital and digital-to-analog converters*. Springer Science & Business Media, 2013.
[102] S. Cassenaer and G. J. N. Laurent, "Hebbian STDP in mushroom bodies facilitates the synchronous flow of olfactory information in locusts," vol. 448, no. 7154, p. 709, 2007.
[103] M. Hu, H. Li, Y. Chen, Q. Wu, G. S. Rose, and R. W. Linderman, "Memristor crossbar-based neuromorphic computing system: A case study," *IEEE transactions on neural networks and learning systems*, vol. 25, no. 10, pp. 1864-1878, 2014.
[104] S. A. Aamir *et al.*, "An Accelerated LIF Neuronal Network Array for a Large-Scale Mixed-Signal Neuromorphic Architecture," *IEEE Transactions on Circuits and Systems I: Regular Papers*, no. 99, pp. 1-14, 2018.
[105] S. Moradi, S. A. Bhave, and R. Manohar, "Energy-efficient hybrid CMOS-NEMS LIF neuron circuit in 28 nm CMOS process," in *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2017, pp. 1-5: IEEE.
[106] J. H. Wijekoon and P. Dudek, "Compact silicon neuron circuit with spiking and bursting behaviour," *Neural Networks*, vol. 21, no. 2-3, pp. 524-534, 2008.
[107] C. A. Desoer, *Basic circuit theory*. Tata McGraw-Hill Education, 2010.
[108] P. Hummel and C. J. T. A. M. M. Seebeck Jr, "A generalization of Taylor's expansion," vol. 56, no. 4, pp. 243-247, 1949.

[109] C. Zhao, K. Hamedani, J. Li, and Y. Yi, "Analog Spike-Timing-Dependent Resistive Crossbar Design for Brain Inspired Computing," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems,* vol. 8, no. 1, pp. 38-50, 2018.
[110] Y. Dan and M.-m. Poo, "Spike timing-dependent plasticity of neural circuits," *Neuron,* vol. 44, no. 1, pp. 23-30, 2004.
[111] K. Cameron, V. Boonsobhak, A. Murray, and D. Renshaw, "Spike timing dependent plasticity (STDP) can ameliorate process variations in neuromorphic VLSI," *IEEE Transactions on Neural Networks,* vol. 16, no. 6, pp. 1626-1637, 2005.
[112] T. Mathworks, "Corr2-2d correlation coefficient," *MATLAB Image Processing Toolbox,* 2014.