2021 22nd International Symposium on Quality Electronic Design (ISQED) | 978-1-7281-7641-3/20/$31.00 ©2021 IEEE | DOI: 10.1109/ISQED51717.2021.9424267

Low-power Analog and Mixed-signal IC Design of Multiplexing Neural Encoder in Neuromorphic Computing

Honghao Zheng, Nima Mohammadi, Kangjun Bai and Yang Yi
Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA USA
E-mail: {zhenghh, nimamo, kangjun and yangyi8}@vt.edu

Abstract

The research on computing clusters comprising neuromorphic systems has drawn the interest of many researchers in the field. Neural encoding is a crucial component that determines how the information is conveyed through a train of spikes, greatly impacting the mode of operations' and systems' performance to a large extent. Numerous encoding schemes have been proposed in the literature, including latency encoding, ISI encoding, and phase encoding. Each of these schemes has its own benefits and shortcomings which brings up the idea to see if they can complement each other. Multiplexing encoding combines two different schemes with the aim of enhancing the performance via conveying more information, making the encoded spikes more robust against noise. In this paper, we introduce a mixed-signal IC design of multiplexing latency-phase encoder. A key principle of the multiplexing encoding, the gamma alignment, is employed to achieve enhanced functionality of spiking neurons supported by biological research. In the proposed encoding scheme, a set of predetermined spiking neurons, which can be perceived as dimensionality reduction over the grouped higher-dimensional stimuli, maps the input currents to latency spike trains. Consequently, these spike trains are aligned and then superimposed on each other to form the resulting spike train. The simulation result is carefully inspected for verification of the encoder. The introduced power-efficient circuit is designed with 180nm CMOS technology and, to the best of our knowledge, is the first IC design of the multiplexing latency-phase that is built upon two different encoding schemes. The power consumption of the encoder is generally proportional to the number of neurons, and for a 4-neuron structure, the layout-level simulation result shows the circuit consumes 10mW of power.

Keywords

Neuromorphic computing, multiplexing encoding, analog and mixed-signal IC design, gamma alignment

1. Introduction

Inspired by the mechanism that our human being process information, neuromorphic computing systems are developed to mimic the operations and characteristics of biological neural networks [1]. Neuromorphic computing has drawn tremendous interest in recent years due to its ability to outperform traditional computing systems and overcome the limitations, and yet, a pressing issue for data-intensive applications such as pattern recognition and machine learning. More importantly, neuromorphic computing consistently obtains more power-efficient realizations, a trait shared with biological systems. As an extreme example, the human brain, containing 10^{11} neurons, only consumes 10W of power [2].

In past decades, researchers have been working feverishly to integrate different analog, digital and mixed-signal devices to mimic the operation of biological neural networks. To this end, using the state-of-the-art CMOS technology to build neuromorphic computing systems has been a common pursuit.

Among all the processing elements constituted in a neuromorphic system, encoders specifically play a vital and indispensable role. Spike encoding refers to the process of converting the information (of input stimuli) into a set of spike trains that can be processed by downstream units. Initially, hardware implementations of rate encoders became a more prevalently used technique compared to other encoding schemes. This popularity mainly stems from the fact that the rate encoding is comparatively easier to realize than other schemes [3]. However, such simplicity comes with significant inefficiency of the encoder in conveying information. Due to neglection of timing elements in the encoding window, rate coding fails to account the temporal aspect of stimuli, drawing the researchers' attention in devising more efficient temporal encoding schemes. In contrast, temporal encoding employs the timing response for mapping information, embedding the temporal aspect into the encoded spike train [4].

Besides the temporal encoding schemes that have currently been proposed, researchers are still looking for novel schemes that can further enhance the performance of neuromorphic designs. All of these pose a doubt on whether two different types of encoders can be combined for a more efficient coding performance, a methodology known as multiplexing encoding [5]. Up till now, only few integrated circuit (IC) designs and software simulations of this scheme have been investigated.

In this paper, we introduce a latency-phase multiplexing encoder, which is designed using the GlobalFoundries 180nm CMOS technology. This encoder not only can make use of the time interval between the sampling onset and the first spike, but also is able to employ the phase characteristic of intrinsic oscillations to convey information.

2. Background of neural encoder

2.1. Encoding Scheme

The design of a proper neural encoding scheme mandates the format of the conveying signal to be carefully selected [4]. A natural encoding approach is to relate the number of spikes during the encoding window to the carried information, whereas other types of encoding transfer the information by exploiting other aspects of spike trains. Accordingly, the encoding schemes can be broadly classified into two main categories, namely rate encoding and temporal encoding [3].

As mentioned above, in rate encoding, the encoded information is carried only via the firing rate of spikes during the encoding window, neglecting other properties of a sequence of spikes that can be used to this end. Figure 1 depicts the rate encoding scheme within one sampling window. As evident from this one-dimensional mechanism, rate encoding is comparatively simplistic, a fact that has led to its wide-spread use. Nevertheless, this simplicity consequently equates to a lower amount of information to be carried, making it highly susceptible to noise.

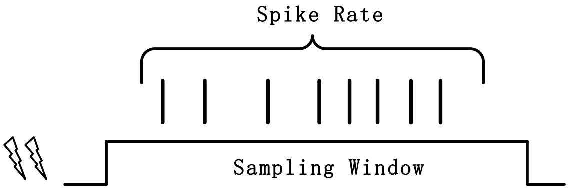

Figure 1: Representation of rate encoding.

On the other hand, as the name implies, the temporal encoding schemes employ the temporal patterns embedded in the exact timing and order of spikes to convey information. There have been multiple temporal encoding schemes introduced in the literature. The *time-to-first-spike (TTFS),* also known as *latency encoding*, is regarded as the simplest mechanism that falls into this category. Latency-encoded information is carried by the time difference between the onset of the encoding window and a single emitted spike, as illustrated in Figure 2.

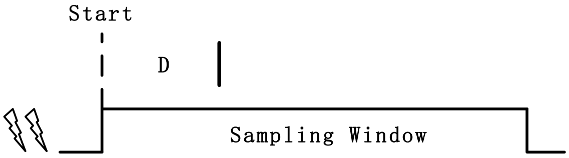

Figure 2: Representation of TTFS encoding.

Due to the dependence on the onset time of the sampling window, the performance of the latency encoding system highly depends on the precision of the starting point of the sampling window, which is often an external reference.

Avoiding this external reference brings us to another temporal encoding scheme, referred to as the *inter-spike interval (ISI)* encoding [6]. Different from the latency encoding, here, the information is encoded into time intervals between consecutive spikes. There are two kinds of circuit design for ISI encoding, the simpler version, namely the *parallel encoder* that maintains the following linear relationship,

$$N_S = N, \tag{1}$$

where N and N_s are the number of neurons and number of spikes, respectively. On the other hand, the other design, named the *iteration encoder*, holds an exponential relationship of the form

$$N_S = 2^{N-1}. \tag{2}$$

The ISI encoding scheme is shown in Figure 3.

It is evident that compared to the TTFS encoding, the ISI encoding scheme evokes more spikes during the sampling window; hence more information is carried with this scheme.

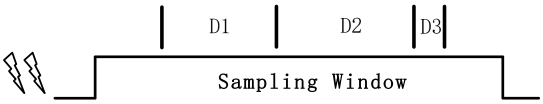

Figure 3: Representation of ISI encoding.

Another way of resolving the issue with precise time-dependence on the input onset is to rely on an intrinsic internal clock of neuron. This mechanism, referred to as phase encoding, relies on subthreshold membrane oscillations (SMOs) that provide such an intrinsic clock. Upon the SMOs crossing a certain threshold voltage, spikes will be fired, which may be operated as a means of conveying the information. The general expression of SMOs can be written as

$$SMO_i = A\cos(\omega t + \phi_i), \tag{3}$$

where A denotes the magnitude of the SMOs, ω is the phase angular velocity and ϕ_i is the phase of the i-th input, for $i \in \{1,2,3,\dots,\mathrm{N}\}$ with N being the input dimension. More specifically, ϕ_i can be defined as

$$\phi_i = \phi_0 + (i-1)\Delta\phi, \tag{4}$$

where ϕ_0 is the initial phase and $\Delta\phi$ is the phase shift between each SMO.

The fact that there exist numerous encoding schemes bring about more desirable encoding by combining these mechanisms in a complementary fashion to increase the performance of a neuromorphic computing system, a process known as *multiplexing* [5].

There are two main multiplexing encoding schemes, *latency-phase encoding*, and *ISI-phase encoding*. The latency-phase encoding represents the scheme that multiplexes the latency and the phase encoding mechanisms, whereas the ISI-phase scheme does the same except with the ISI encoding instead of the latency encoding. Both of these multiplexing schemes include one step called as gamma alignment, whose goal is to move the spikes to the next closest incoming SMO. Figure 4 and Figure 5 illustrate the latency-phase encoding and the ISI-phase encoding, respectively.

2.2. Advantages and Challenges of Multiplexing

Since different encoding schemes lead information to be expressed on different timescales, for example, while ISI encoding scheme operates in higher frequency, the phase encoding has a much coarser precision, the idea of multiplexing grows very naturally based on such facts. The encoding schemes with different timescales might be integrated together to code complementary information features. With such property, the whole system's encoding ability will be improved vastly.

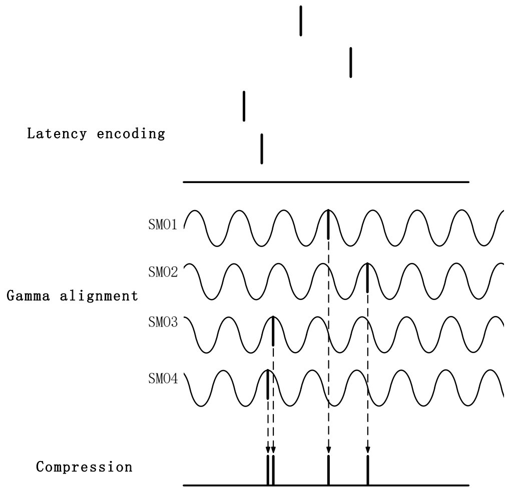

Figure 4: Representation of Latency-phase encoding.

While all the encoding schemes will be affected by input noises, the multiplexing ones are the least interfered. With the phase of firing encoding integrated with other schemes, the multiplexing encoding will contain at least one SMO. Such internal temporal reference frame has the property of stabilize the system, especially when receiving noisy signal.

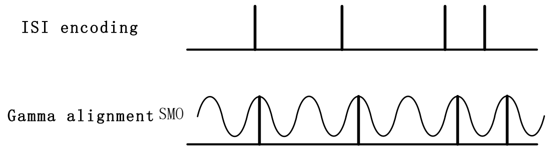

Figure 5: Representation of ISI-phase encoding.

Though there are more and more evidence proving multiplexing encoding has the advantage of robustness toward noise and less ambiguity caused than other schemes, challenge still exists in realizing a practical and efficient encoder. For example, integrating different processed signals when the number of neurons increases is a very challenging task, especially for ISI-Phase encoders. Such an encoding scheme not only need to integrate them together, but also need to investigate how to decode such signals.

2.3. Analog Neuron

Before exploring the detailed structure of the latency-phase encoder, it is crucial to discover an appropriate design for neurons as they will be used to realize the latency encoding functionality in our introduced multiplexing encoder.

From the first days of neuroscience, numerous researches have been carried out about the biological neuron, proposing various neuron models ranging from sophisticated biophysical models to more mathematically simplified ones. Due to the complexity and other limitations, only a few models are applicable for realizing in the IC area, in which two neuron models are commonly used in neuromorphic application due to their simplicity, namely the integrated and fire (IF) and the leaky integrate and fire (LIF) models [2]. With the consideration of simplicity and conciseness, the proposed encoder adopts the IF model as the neuron model.

In an IF circuit, there exists a critical design parameter called threshold voltage, V_{th}. The voltage across the capacitor will be charged linearly as the input current, I_{in}, is active. When the voltage across the membrane capacitor, V_{mem}, reaches the V_{th}, the circuit will fire a spike signal to the output. After that, V_{mem} will be reset to 0 through a switch transistor controlled by the output spike. This charging and firing process is the basic idea of IF neuron, who acts as the latency encoder in our design. The equation governing the relationship of I_{in}, V_{mem} and membrane capacitance can be written as

$$I_{in} = C\frac{dV_{mem}}{dt}. \tag{5}$$

3. Encoder Blocks

The introduced multiplexing encoder has three critical computing modules. The first module is called the latency encoding neuron, which is utilized to accomplish the latency encoding. The second module is implemented to fulfill the need of spike width required for the later computation, named as spike expander. The third module is the gamma alignment, moving the spike to the next maximum of SMOs. To enable the simultaneously operation with multiple input signals at once, multiple signal processing routes are built in parallel. Lastly, an OR gate is employed to integrate the outcomes from gamma alignment modules. The overview of our multiplexing encoder in shown in Figure 6.

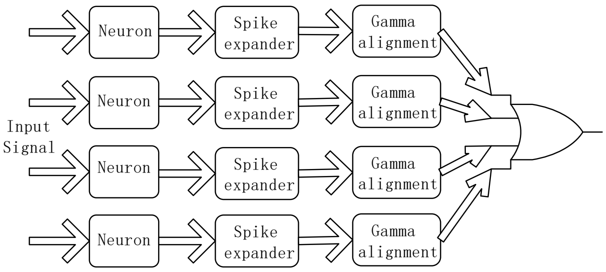

Figure 6: Overview of Multiplexing Encoder.

In the latency encoding module, the neurons corresponding to multiple routes integrate voltages across the capacitors at different speeds. With a larger input current, the voltage across the membrane capacitor rises to threshold voltage more quickly. Since they have the same threshold voltage, the firing spikes of the neurons will appear at different times. Thus, input of higher intensity leads to emission a spike closer to the onset of the encoding window.

In the gamma alignment module, a peak detector is implemented to detect the firing activity of spikes and hold the firing magnitude. Once a spike is detected, it will be injected into an AND gate with a SMO whose magnitude is carefully tuned so that its maximum will be exactly at the threshold of the AND gate. With the maximum of SMO detected by the AND gate, it will fire another spike. This output will go through a buffer to ensure its stability. Such a signal will also be used as a switching signal of the leaking switch at the spike input end of the AND gate. Therefore, once the output spike is fired, the voltage level at the spike input end will be reset to a certain level lower than the gate's threshold voltage. Hence, an input spike leads to a single output spike being fired.

Notice that there is an issue with the peak detector where the input spikes are required to last at least 10ns for detection, or there will not be enough time for the voltage level of the detector to rise above the threshold voltage before the spike disappears. The output of the latency encoding module, however, only has 1ns width. To overcome this issue, an additional module is introduced, which refers to as the spike expander, enforcing the spike width to 10ns.

Since the four routes of the signal require four SMOs, 45 degrees out of phase with each other, maintaining the same amplitude, a SMO generator is designed to provide such functionality with finely tuned magnitude.

4. Hardware Implementation

The multiplexing encoder introduced in this paper is designed and simulated in the GlobalFoundries 180nm CMOS technology.

The structure of the neuron utilized as the latency encoding is depicted in Figure 7. Upon the charging effect of the input current on the membrane capacitor, the voltage across the capacitor rises. Since the gate voltage of M1 increases, the voltage at the drain of M2 rises up accordingly. When the voltage exceeds a certain threshold, a voltage controlled by V_{ref}, a spike will be generated via the buffer consisting two NOT gates. At the meantime, the feedback mechanism starts to reset the voltage across the capacitor. The sampling rate of the neuron is controlled by the CLK signal. When a spike is fired, the gate of M11 is set to be a high voltage so that the charges at the top plate of the capacitor will leak through M11. Thus, the reset mechanism is achieved. In conclusion, this circuit can imitate the basic function of a biological neuron and is able to accomplish the latency encoding.

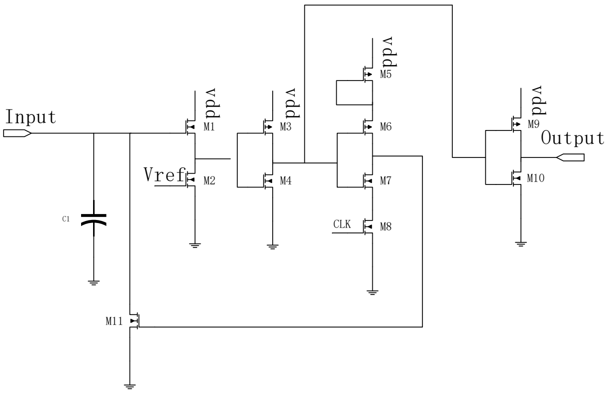

Figure 7: Circuit implementation of latency neuron.

The gamma alignment is the key module in our introduced multiplexing encoder, shifting the incoming spikes to the next maximum of SMOs. The structure is shown in Figure 8. In the circuit implementation, the diode-connected transistor and the capacitor are used as a spike detector. When a spike is received, M1 delivers the energy potential from the incoming spike to the capacitor, preventing the charge from leaking when the spike is reset. Thus, the signal at the spike input end of the AND gate will be held as digital 1 until the next maximum of SMO. The amplitude of the SMO needs to be carefully tuned so that it can be recognized as digital 1 only at the peaks. Therefore, when the two inputs of the AND gate both reach digital 1, a spike will be fired through the buffer to the output. Meanwhile, to reset the voltage at the spike input end, a feedback mechanism is utilized. After the firing process, the gate of M2 is set to be a high voltage, which allows the voltage at C1 to be reset to a certain value lower than digital 1 until the arrival of next incoming spike.

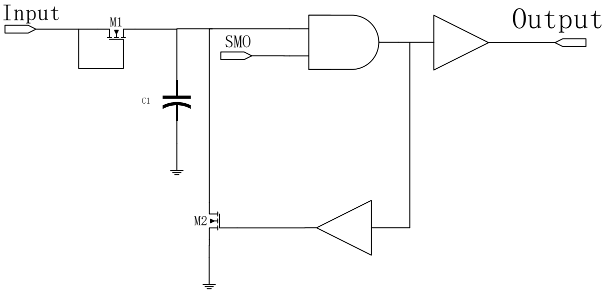

Figure 8: Circuit implementation of gamma alignment.

As mentioned in the previous section, the peak detector requires a certain pulse width of the spike to function properly. If the existing time of the spike is too short, the voltage across the capacitor will not be able to rise to digital 1 before the spike is reset. Due to the instant reset operation, the output spikes from the latency neuron are too narrow to be distinguished by the spike detector. To overcome this issue, an additional module is implemented to extend the existing pulse width of spikes, named as spike expander, as illustrated in Figure 9.

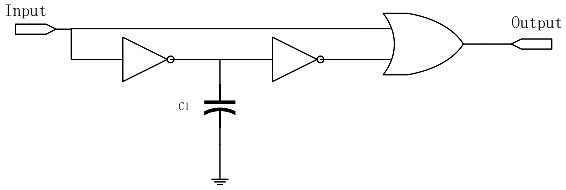

Figure 9: Circuit implementation of spike expander.

Two inverters and a capacitor are utilized to create certain delays to the initial spike, *e.g.*, 0.8 × of the initial spike width. After that, the initial spike and the delayed spike are integrated by an OR gate, forming a new set of spikes with nearly 1.8 × of former width. To ensure that the spikes are wide enough to trigger the gamma alignment module, four cascaded spike expanders are implemented, increasing the width of a spike to 10ns from less than 1ns. What needs to be noticed is that the capacitor on each spike expander has different values, since the spikes need to be delayed by a different amount of time.

Another critical module is the one used to provide SMOs. Since SMOs in the latency-phase encoder need to have the same magnitude with a specific phase shift, the circuit will have the functionality to shift the phase while keeping the magnitude steady. The structure of the SMO module is shown in Figure 10. Each pair of capacitors and resistors is utilized to shift the phase of the signal. Since the magnitude of the signal will be decreased along with the phase shifting, an amplifier structure with op-amp is used to elevate the amplitude of the phase-shifted signal back to the original. A total of four SMOs is implemented, where each of which is utilized to control the corresponding gamma alignment module.

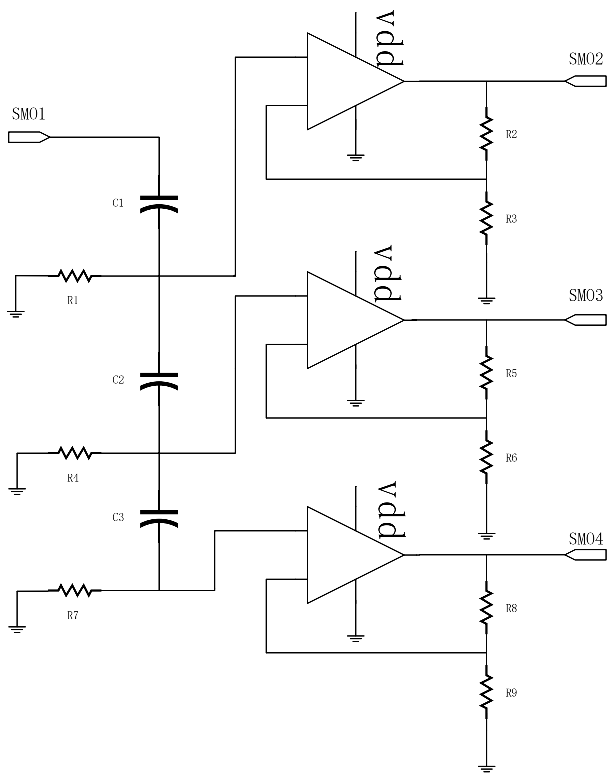

Figure 10: Circuit implementation of SMO.

5. Results Discussion

From the previous discussions, it can be observed that the output of each module is critical in this design, greatly impacting the accuracy of the final outcomes. Thus, a careful inspection needs to be carried out to the input and output signals.

Figure 11 demonstrates the mapping of input current to a TTFS spike. In this experiment, an input current, ranging from 0.3uA to 1.1uA was applied as the input, while the latency neuron sampled the input information at a rate of 0.67MHz. It can be observed that with an input of high magnitude, the output spike would be closer to the *CLK* signal, whereas input of lower intensity would lead to the spike further away from the *CLK* signal. Such property fulfills the requirement of latency encoding.

Figure 12 depicted the property of pulse width extension achieved by the spike expander module, such that the signal can be detectable by later modules. It can be seen that the outcome from the spike expander spans around 15ns width while the input spike only has a narrow 3ns width. Through our initial experiment, the peak detector in the gamma alignment module requires a width of at least 10ns in order to operate properly. It is reasonable to conclude that the spike expander module has met the design requirements for later computation.

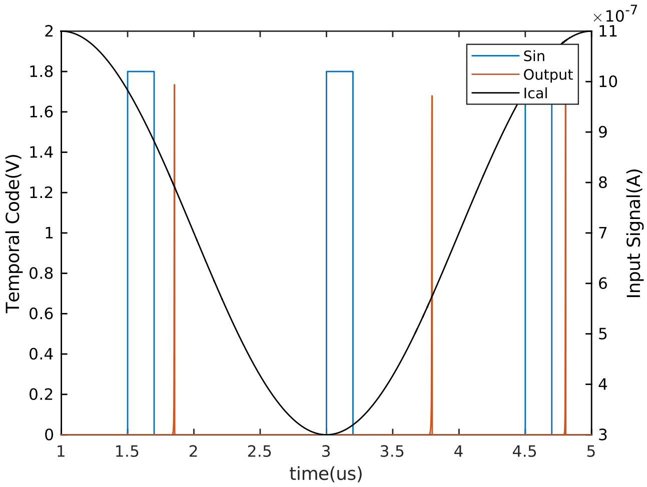

Figure 11: TTFS spikes of latency encoding module.

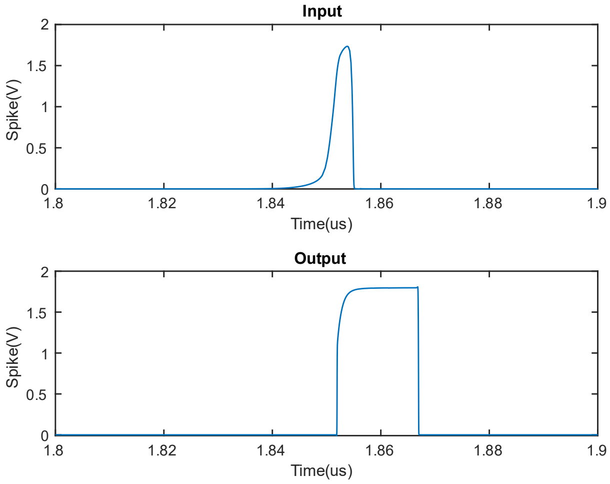

Figure 12: Illustration of pulse width extension.

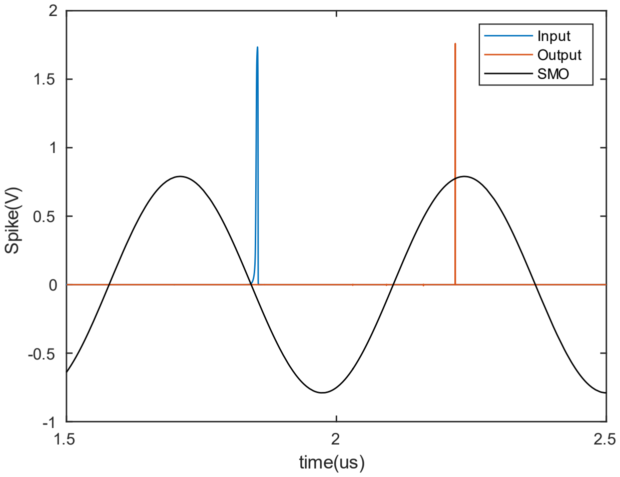

Figure 13: Illustration of gamma alignment.

The gamma alignment is the most critical module in the multiplexing encoder design. Figure 13 shows the relationship between the initial input spike, the SMO, and the shifted output spike. With the influx of the expanded spike, the peak detector in the gamma alignment module captures the incoming signal and maintains the ‘high’ state.

The fine-tuned SMO has the capability to maintain its maximum voltage at the threshold of the AND gate; thus, when it arrives at the max point, it will be in a ‘high’ state as well. An output spike will be fired at this moment with the help of a reset structure. From Figure 13, it can be observed that an input spike is indeed mapped to the next maximum peak of SMO.

6. Layout and Power Analysis

As discussed in Section 4, this specific design adopts four neurons to map four different signals synchronously. However, a latency-phase encoder may superimpose any number of neurons. Thus, the layout size and power consumption vary drastically with respect to the number of neurons. Since the SMOs module requires large capacitors to achieve the desired phase shift, the design area is also proportional to the number of neurons. Figure 14 shows the layout of a 4-neuron latency-phase multiplexing encoder. The overall design area of this encoder is around $760 \times 490um^2$.

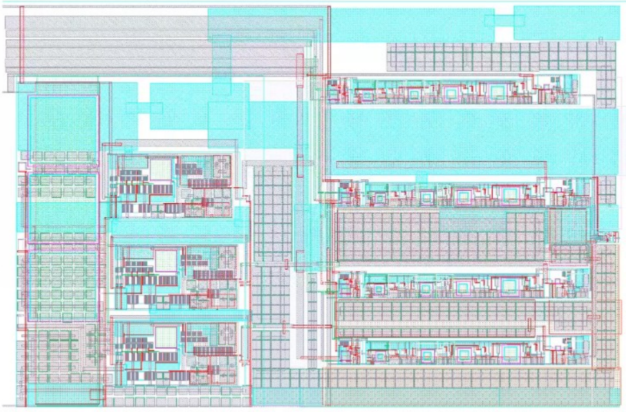

Figure 14: Layout of latency-phase multiplexing Encoder.

Table I: Power Consumption and Area Comparison

Encoding Scheme	Power (mW)	Area (um^2)
Latency	0.425	28 x 33
Latency-Phase	0.43	40 x 303
ISI	0.846	64 x 69
ISI-Phase	0.847	64 x 340

As shown in Table I, different encoding schemes are summarized on the scale of power and area. Due to the matter of fairness, every encoder has only one neuron. It can be observed that the power consumption of latency and latency-phase, ISI and ISI-phase are identical, respectively, meaning that the spike expander and gamma alignment circuit are indeed power efficient.

Acknowledgements

This work was supported in part by the U.S. National Science Foundation (NSF) under Grant CCF-1750450, Grant ECCS-1731928, and Grant CCF-1937487.

Reference

[1] K. Bai and Y. Yi, "Opening the “Black Box” of Silicon Chip Design in Neuromorphic Computing," in Bio-Inspired Technology: IntechOpen, 2019.

[2] 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, pp. 1-6.

[3] K. Hamedani, "Energy Efficient Deep Spiking Recurrent Neural Networks: A Reservoir Computing-Based Approach," Virginia Tech, 2020.

[4] 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, pp. 1-21, 2015.

[5] S. Panzeri, N. Brunel, N. K. Logothetis, and C. Kayser, "Sensory neural codes using multiplexed temporal scales," Trends in neurosciences, vol. 33, no. 3, pp. 111-120, 2010.

[6] C. Zhao, "Spike Processing Circuit Design for Neuromorphic Computing," Virginia Tech, 2019.

[7] Z. Nadasdy, "Information encoding and reconstruction from the phase of action potentials," Frontiers in systems neuroscience, vol. 3, p. 6, 2009.

[8] K. Hamedani, L. Liu, S. Liu, H. He, and Y. Yi, "Deep Spiking Delayed Feedback Reservoirs and Its Application in Spectrum Sensing of MIMO-OFDM Dynamic Spectrum Sharing," in Proceedings of the AAAI Conference on Artificial Intelligence, 2020, vol. 34, no. 02, pp. 1292-1299.

[9] H. E. Michel, D. Rancour, and S. Iringentavida, "CMOS Implementation of Phase-Encoded Complex-Valued Artificial Neural Networks," in ESA/VLSI, 2004, pp. 551-557.

[10] K. Bai, Q. An, L. Liu, and Y. Yi, "A training-efficient hybrid-structured deep neural network with reconfigurable memristive synapses," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 28, no. 1, pp. 62-75, 2019.

[11] A. Cattani, G. T. Einevoll, and S. Panzeri, "Phase-of-firing code," arXiv preprint arXiv:1504.03954, 2015.

[12] K. Hynna and K. Boahen, "Space-rate coding in an adaptive silicon neuron," Neural Networks, vol. 14, no. 6-7, pp. 645-656, 2001.

[13] M. Lukoševicius, "Reservoir computing and self-organized neural hierarchies," Jacobs University, Bremen, 2012.

[14] L. Appeltant, "Reservoir computing based on delay-dynamical systems," These de Doctorat, Vrije Universiteit Brussel/Universitat de les Illes Balears, 2012.

[15] B. Schrauwen, D. Verstraeten, and J. Van Campenhout, "An overview of reservoir computing: theory, applications and implementations," in Proceedings of the 15th european symposium on artificial neural networks. p. 471-482 2007, 2007, pp. 471-482.

[16] L. Wen-peng, C. Xu, and L. Hua-xiang, "A new hardware-oriented spiking neuron model based on set and its properties," Physics Procedia, vol. 22, pp. 170-176, 2011.

[17] C. Zhao, L. Liu, and Y. Yi, "Design and Analysis of Real Time Spiking Neural Network Decoder for Neuromorphic Chips," in Proceedings of the International Conference on Neuromorphic Systems, 2019, pp. 1-4.