Enabling a New Methodology of Neural Coding: Multiplexing Temporal Encoding in Neuromorphic Computing

Honghao Zheng, *Student Member, IEEE*, Kang Jun Bai, *Member, IEEE*, and Yang Yi, *Senior Member, IEEE*

***Abstract*— From rate to temporal encoding, spiking information processing has demonstrated advantages across diverse neuromorphic applications. In the aspects of data capacity and robustness, multiplexing encoding outperforms alternative encoding schemes. In this work, we aim to implement a new class of multiplexing temporal encoders, patterning stimuli in multiple timescales to improve the information processing capability, and robustness of systems deployed in noisy environments. Benefitted by the internal reference frame using subthreshold membrane oscillation (SMO), the encoded spike patterns are less sensitive to the input noise, increasing the encoder's robustness. Our design results in a tremendous saving on power consumption and silicon area compared with the power-hungry analog-to-digital converters. Furthermore, a working prototype of the multiplexing temporal encoder built based on an interspike interval (ISI) encoding scheme is implemented on a silicon chip using the standard 180-nm CMOS process. To the best of our knowledge, our introduced encoder demonstrates the first integrated circuit (IC) implementation of neural encoding with multiplexing topology. Finally, the accuracy and efficiency of our design are evaluated through standard machine learning benchmarks, including Modified National Institute of Standards and Technology (MNIST), Canadian Institute For Advanced Research (CIFAR)-10, Street View House Number (SVHN), and spectrum sensing in high-speed communication networks. While our multiplexing temporal encoder demonstrates a higher classification accuracy across all the benchmarks, the power consumption and dissipated energy per spike reach merely 2.6 μW and 95 fJ/spike, respectively, with an effective frame rate of 300 MHz. Compared with alternative encoding schemes, our multiplexing temporal encoder achieves at most 100% higher data capacity, 11.4% more accurate in classification, and 25% more robust against noise. Compared with the state-of-the-art designs, our work achieves up to 105× power efficiency without significantly increasing the silicon area.**

***Index Terms*— Analog integrated circuit (IC) design, gamma alignment, interspike interval (ISI), multiplexing encoding, neuromorphic computing.**

Manuscript received 16 June 2022; revised 18 October 2022 and 7 December 2022; accepted 23 December 2022. Date of publication 19 January 2023; date of current version 24 February 2023. 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. *(Corresponding author: Yang Yi.)*

Honghao Zheng and Yang Yi are with the Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA 24061 USA (e-mail: zhenghh@vt.edu; yangyi8@vt.edu).

Kang Jun Bai is with the Air Force Research Laboratory Information Directorate, Rome, NY 13441 USA (e-mail: kang.jun.bai@us.af.mil).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TVLSI.2023.3234514.

Digital Object Identifier 10.1109/TVLSI.2023.3234514

I. Introduction

NEUROMORPHIC computing was first introduced by Carver Mead in the late 1980s [1]. With the inherent self-learning capability, neuromorphic computing systems have drawn tremendous interest in recent years with the potential to accelerate computational efficiency and overcome the limitations of the traditional von Neumann architecture [1]. To be specific, neuromorphic computing systems can efficiently process and learn from data, making data-intensive applications, such as image classification and speech recognition, to be effectively carried out [2]. More importantly, neuromorphic computing systems consistently obtain higher power efficiency, in a way that is sufficient by modeling our human brain. In the past decades, efforts to imitate the operation in biological neural systems with very VLSI circuits have been made [3], [4]. For instance, the IBM TrueNorth has demonstrated excellent power efficiency (in less than 3 W) compared with the standard central processing unit (CPU) and graphics processing unit (GPU) over object recognition tasks [4]. Moreover, the Intel Loihi has also been evaluated through diverse applications—such as adaptive robot arm control and drone motor control with the state-of-the-art latency in response to the visual input—by only consuming far less than 1 W of power [3]. As for the Tianjic, the chip achieves high accuracy with high throughput and power efficiency for pattern recognition and has been used in a multimodal bicycle system [5]. Finally, the BiCoSS, which achieves high accuracy for the Modified National Institute of Standards and Technology (MNIST) dataset, can reproduce the network-level dynamic activities of different brain regions to further reveal the mechanisms responsible for human cognition and analyze the mechanisms from the neuron level [6]. In short, neuromorphic computing systems outperform the traditional structures in both the learning ability and power efficiency perspectives.

Similar to the biological neural systems, signals can be represented in terms of spikes in the neuromorphic computing systems. Thus, a spike encoder is essential for a neuromorphic computing system. To better understand the functionality of the spiking information processing, the neural encoding schemes need to be carefully investigated. Such an encoding scheme refers to converting the information of raw sensory inputs into a set of spike trains, which downstream units can process. In general, there are two major types of encoding schemes, rate encoding and temporal encoding [7]. Rate encoding is an encoding scheme that maps input information

TABLE I
PROS AND CONS OF DIFFERENT ENCODING SCHEMES

Encoding Scheme	Rate	Temporal	Multiplexing
Pros	Straightforward and Easy to implement	Higher data capacity	Highest data capacity and High robustness
Cons	Low data capacity	Low robustness against noise	High complexity and power & area cost

into the number of spikes within a sampling window. Due to its simplicity in realization, rate encoding is more widely used in both software and hardware [8]. However, such an encoding scheme has the disadvantage of low data density, in which only the firing rate is used to convey information and the temporal patterns of spikes are ignored. On the other hand, the temporal codes tend to represent information with the temporal patterns of spikes, thus using both the firing rate and time of spikes. Two different types of temporal encoding have been commonly investigated, the *time-to-first-spike (TTFS)* encoding and the *interspike interval (ISI)* encoding.

With the increasing data density for mission-critical applications [9], the demand for higher data capacity increases. It has been found that an encoder that combines multiple encoding topologies would increase the data capacity, known as multiplexing encoding. In addition to the improved encoding capability, multiplexing encoding is more robust against noise using an internal reference frame [10], [11].

Table I demonstrates the pros and cons of the aforementioned encoding schemes. It shows that multiplexing encoding can provide the highest data capacity and better robustness compared with alternative encoding schemes, and yet, the integrated circuit (IC) implementation of the multiplexing temporal encoder has not been discovered. In this work, we design and analyze a multiplexing temporal encoder, which can transfer raw sensory inputs into neural spike trains with multiple timescales. The major contributions of our work are summarized as follows.

1) Our work develops the first IC implementation of multiplexing temporal neural encoder using the GlobalFoundries standard 180-nm CMOS process.
2) High robustness is realized by the ISI encoding topology and the internal reference frame with subthreshold membrane oscillation (SMO). The SMO together with gamma alignment enables precise phase modulation within the encoder, thus enhancing its robustness and classification accuracy. Better yet, an improved data processing capacity with up to 100% over alternative encoding schemes is also achieved.
3) The multiplexing temporal encoder was evaluated with spiking neural networks (SNNs) in PyTorch, and the performances were compared with alternative encoding schemes (rate, TTFS, ISI). A classification accuracy of 93.78% on the MNIST dataset was reported, yielding up to 10.78% improvement over alternative encoding schemes. Beyond that, improvements in classification accuracy with up to 6.4% and 11.4% on the Canadian Institute For Advanced Research (CIFAR)-10 and Street View House Number (SVHN) data, respectively, were also achieved.
4) Under the performance evaluation of spectrum sensing in *multiple-input and multiple-output orthogonal frequency-division multiplexing (MIMO-OFDM)* systems, the detection accuracy with our multiplexing temporal encoder demonstrates 11% improvement over alternative encoding schemes, revealing the application potential of the proposed encoder in 5G or Internet of Things (IoT).
5) We eliminate the power-hungry analog-to-digital converters (ADCs) and op-amps, and thus, our multiplexing temporal encoder offers an ultralow power consumption at 2.6 μW with a reasonable silicon area of 0.024 mm^2. The dissipated energy per spike is 95 fJ/spike with an effective frame rate of 300 MHz.

In this article, the literal research of the neural encoding schemes and the general structure of our multiplexing temporal encoder are illustrated in Sections II and III, respectively, followed by the circuit implementation and experimental results in Section IV. The benchmark and application evaluations are discussed in Section V, and the article is then concluded in Section VI.

II. NEURAL ENCODING

Neural encoding defines the process that transfers the input signal (stimuli) to neural spike trains. Research efforts have been made on investigating different encoding schemes [12]. First, it has been proven that the input stimuli could be represented in terms of the firing rate within the sampling window. Afterward, an encoding scheme that uses the spike patterns to convey information has been found. The former is commonly known as the rate code and the latter is known as the temporal code. The rate encoding scheme, depicted in Fig. 1(a), converts the information into the number of spikes within the sampling window. Because of its simplicity in the working mechanism and hardware realization, rate encoding has been widely applied in designing SNNs. However, even the same input would lead to different patterns of spikes as long as the numbers are the same, and thus, the robustness of rate encoding is limited due to its low data density [10]. Such a low data density also leads to the fact that any mistake in the spike train would affect the output of the whole encoding process. On the contrary, the temporal encoding schemes can use internal/external reference frames to overcome minor mistakes. In [13], a rate encoder is implemented in a feedforward network, and yet, the power consumption of such design reaches 577 μW using the 130-nm technology.

On the other hand, temporal encoding uses time of firing spikes to represent stimuli, improving the accuracy and robustness of the encoded spike trains. In practice, several temporal encoding methodologies have been investigated, among which

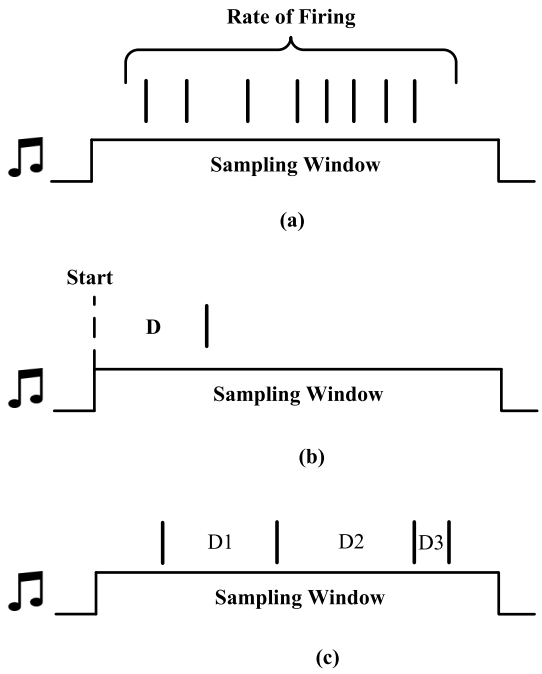


Fig. 1. Examples of the encoding schemes. Presentation of (a) rate encoding, (b) TTFS encoding, and (c) ISI encoding.

the TTFS encoding and the ISI encoding schemes are the two most common [12].

Latency encoding, also known as TTFS encoding, is a well-known temporal encoding scheme [14]. In TTFS, the stimuli are converted into a single spike, where the information is represented by the time difference between the rising edge of the sampling window and the fired spike, as illustrated in Fig. 1(b). In [15], a TTFS encoder was designed with merely 3.5 μW of power consumption. However, external sources are needed to initiate the sampling windows, in which the performance of TTFS code could be affected while its robustness against noise is limited. To be specific, when a neuron inaccurately fires a spike, all the later computing units could not achieve the desired output, leading to tremendous errors in the system.

To sidestep the aforementioned drawbacks, the ISI encoding scheme was introduced. Unlike latency encoding, ISI encoding maps the stimuli to time intervals between spikes, as shown in Fig. 1(c). The ISI spike train is generated in one sampling window without additional reference, as spikes can be relative references to each other. The ISI encoding generates more spikes in one sampling window than the TTFS encoding, thus conveying more information. It has been proven that ISI encoding has a better ability to mimic how biological neural networks encode stimuli. Two different circuit implementations of the ISI encoder have been proposed in [14]. The parallel encoder demonstrates lower latency when encoding information, while the iteration encoder would generate more spikes even with the same number of neurons to improve the area efficiency. In practice, the *parallel encoder* [8] maintains a linear relationship, such that

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

where N and N_S are the number of neurons used in the encoder and the number of spikes within a sampling window, respectively. In contrast, the *iteration encoder* [14] holds an exponential relationship

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

Instead of increasing the number of spikes within sampling windows, relying on an internal reference would also improve the performance. Specifically, the SMO, originated from the rhythmic activity pattern of biological neurons with a dominant frequency [10], is often modeled by a sine wave in neuromorphic applications, where its period needs to be smaller than the sampling time window [11]. Such SMO helps eliminate the onset precision issue, thus supporting phase encoding [16]. In this manner, the input stimuli are converted into a voltage value, where a spike will be fired when SMO reaches it, which can be expressed as

$$\mathrm{SMO}_i = A \cdot \cos(\omega T + \phi_i) \tag{3}$$

where A is the amplitude of SMO, ω is the phase angular velocity, and ϕ_i is the phase of the ith input for $i \in \{1, 2, 3, \ldots, N\}$ with N denoting the number of parallel inputs. 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 SMOs [11].

III. Multiplexing Encoding Design

Intending to improve the data processing capability of systems, the concept of applying the multiplexing encoding mechanism in neuromorphic computing has been introduced [17]. Multiplexing is a process that combines two different encoding schemes to increase the data encoding capacity. It has been proven that multiplexing occurs in biological neural systems, thus improving learning and processing capabilities. For instance, researchers have carried out an experiment using rats to verify the functionality of multiplexing encoding in which a multiplexing code consisting of rate and temporal codes is applied to rats to sense object texture [10]. It turns out that the multiplexing code can transfer more stimuli than whichever of rate encoding and temporal encoding alone [11]. Beyond that, neuroscientists have quantified the information in the analysis of single-unit data to compare the information density of different encoding schemes [18]. As illustrated in Fig. 2, the temporal encoding scheme provides higher information density than rate encoding, and multiplexing codes have higher data capacity than temporal or rate code alone. As such, the multiplexed code requires a smaller encoding time window for the same amount of information and thus increasing the data processing rate.

In addition, the multiplexing encoding schemes can convey more information [11]. With experiments to quantify the impact of sensory noise on different codes, the robustness against noise across various encoding schemes is demonstrated in Fig. 3 [18]. While the information density in various encoding schemes decreases when the noise level is rising,

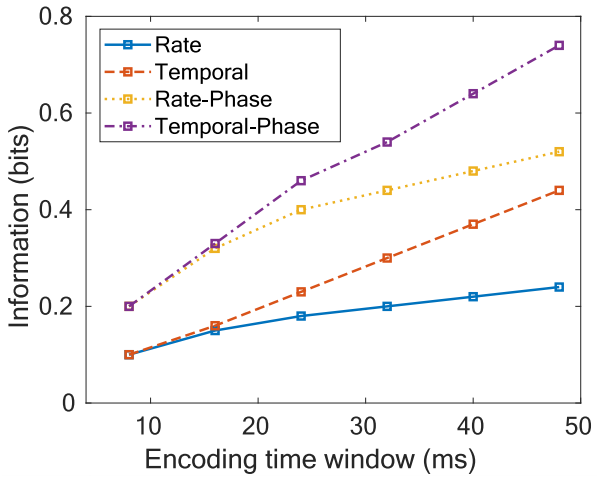


Fig. 2. Stimulus information for different encoding schemes.

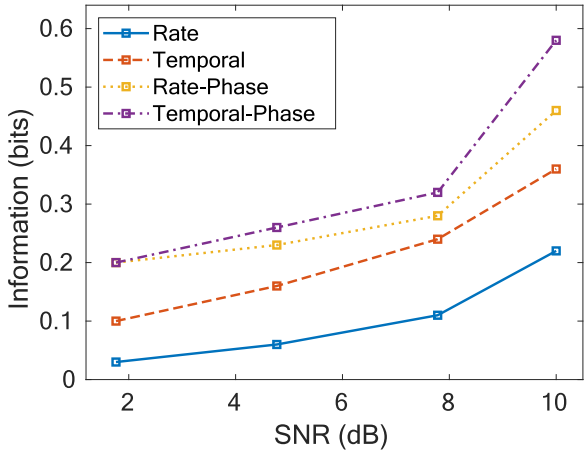


Fig. 3. Information in codes for different noise levels.

the multiplexing code still retains the most information across all noise levels.

The multiplexing operation can be divided into two main steps [11]. The first step is encoding, where the input analog stimuli are transferred into spike signals. Here, we define the excitation signal as S and the output spike (train) as P. When the latency encoding is applied, P will be a single spike. On the other hand, P will be in the form of spike trains if ISI encoding occurs.

The second step of multiplexing encoding is the transformation phase. To discuss this step, a terminology, *gamma alignment*, needs to be introduced. The gamma alignment refers to the process that shifts the generated latency (or ISI) spikes to the next local maximum of SMO, which can be written as

$$P'_\tau = P_t \tag{5}$$

where t is the exact time of P, and τ is the next local maximum of its corresponding SMO.

For instance, in TTFS-phase encoding, the gamma alignment shifts the output spikes of TTFS encoding to each channel's corresponding SMO. These SMOs have a particular

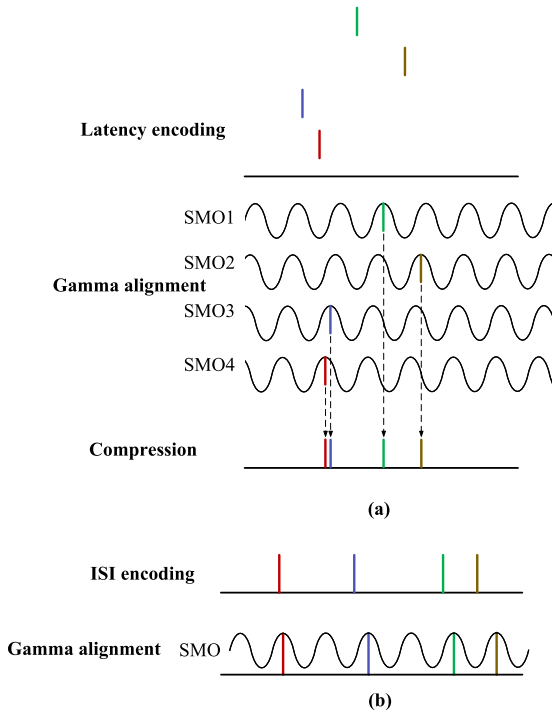


Fig. 4. Examples of multiplexing encoding. (a) TTFS-phase encoding and (b) ISI-phase encoding.

phase shift from each other, as illustrated in (4). Such a concept is introduced in [11] with a four-channel TTFS-phase multiplexing encoding scheme, as demonstrated in Fig. 4(a). With a similar operation manner, the gamma alignment procedure shifts the output spikes from the ISI encoder to the next local maximum of SMO in the ISI-phase encoding, as shown in Fig. 4(b).

The concept of multiplexing is proposed based on a mechanism in which information on different timescales can be integrated for higher information density. Multiplexing encoding schemes will combine complementary temporal patterns to transmit more information with the same sampling frequency, thus enhancing the performance of systems [19]. Under noisy environments, the robustness of multiplexing codes has significant advantages over others. The SMO introduced by the multiplexing encoding scheme helps with the system's stability and robustness, especially when operating in noisy environments [12].

In biological systems, the neurons tend to adjust themselves for dynamic input information; thus, the relationship between the stimuli and the output spikes varies from time to time. Such a variation in input–output relationships would lead to ambiguity of response. The spikes would be fired at different times, even with the same stimulus. However, multiplexing encoding would sidestep this shortcoming. For instance, it has been found that the motion-sensitive cell in a fly's visual system tends to encode the history of stimuli and the present stimuli with both rate encoding and ISI encoding. The ambiguity can be eliminated using the ISI scheme to encode the recent history of input information along with the current stimulus, and thus, the necessary information provided to ensure the ambiguity is not a problem [10], [20].

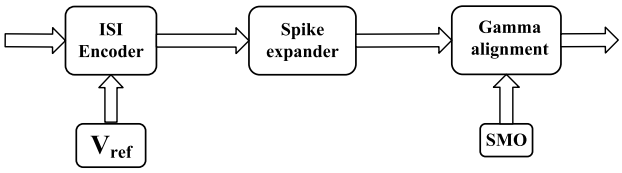


Fig. 5. Overview of the proposed multiplexing encoder.

Some literature has proven the advantages of multiplexing encoding toward noise and ambiguity over the other encoding schemes [10], [11], [12], and yet, to implement a practical circuit of the multiplexing encoder, there are still many limitations needed to be addressed. For instance, with the increasing number of neurons, the implementation complexity of the encoder increases exponentially, especially for the ISI-phase encoders.

To use these prominent properties of multiplexing codes, such as high data capacity and robustness, for large-scale data processing applications, the IC implementation of the multiplexing ISI-phase encoder is urgently needed. Not only because IC implementations are power-efficient but also due to their area efficiency. In this work, we introduce a new class of multiplexing encoders consisting of three essential blocks, namely, the ISI encoder, the spike expander for extending the spike width for later computation, and the gamma alignment. The block diagram of our ISI-phase encoder is demonstrated in Fig. 5.

In the ISI encoding module, neurons with different reference voltages fire spikes at different times. The larger the V_{ref}, the further the spike is away from the clock (CLK) signal. Therefore, with an OR gate in the ISI encoder to integrate two spikes, the ISI encoder can transfer the input current to a spike train.

As for the gamma alignment block, the firing amplitude of spikes will be sampled and held by a peak detector composed of a diode-connected transistor and a capacitor. This captured spike will be transmitted to one input of an AND gate, where the other input is from the SMO. This SMO is precisely tuned such that its peak point is right at the AND gate's threshold. One spike will be fired when the next local maximum of the SMO comes after a sampled input. Afterward, the reset mechanism is triggered, and the captured voltage decreases to its initial level.

It is noticeable that the peak detector can only sample spikes that are wider than 10 ns; else, the peak detector will not be triggered. However, the ISI encoder only generates spikes with 1 ns width, and thus, the spike expander is introduced to overcome this issue.

IV. IC Design, Optimization, and Result of Multiplexing Encoder

The introduced ISI-phase multiplexing encoder is designed and optimized using the GlobalFoundries standard 180-nm CMOS technology, and the simulation is performed in the Cadence Virtuoso platform.

A. Analog Neuron

The structure of the neuron used in this design is built based on the latency encoder [8], as shown in Fig. 6. When the input

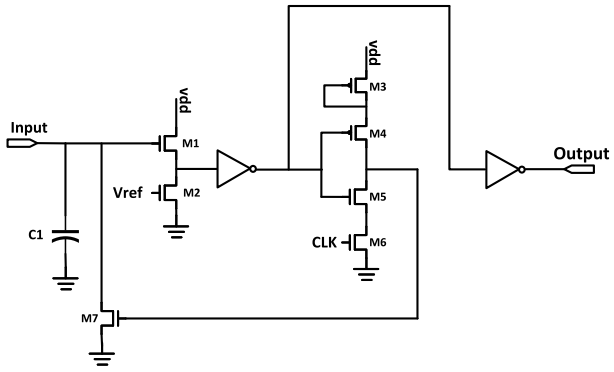


Fig. 6. Circuit schematic of the neuron.

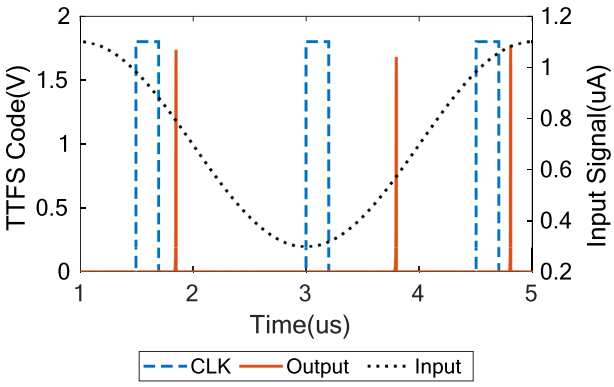


Fig. 7. Spikes of the latency encoding module.

current is being integrated, the voltage across the membrane capacitor C1 increases. Due to the voltage dividing effects, the voltage at the gate of M1 raises up the voltage at the drain of M2. The buffer then fires a spike when the voltage exceeds a certain threshold. Thus, the integration time, T, can be expressed as

$$T = \frac{C_m \cdot V_{\text{ref}}}{I_{\text{in}}} \tag{6}$$

where I_{in} is the excitation current, C_m is the membrane capacitance, and V_{ref} is the threshold voltage of the neuron. The CLK's frequency controls the sampling rate of the neuron. The charge integration on the membrane capacitor starts after the CLK reaches digitally high. Meanwhile, the reset mechanism forces the membrane voltage back to its initial level after the spike fires. To be specific, a fired spike triggers the reset switch M7, leading the membrane voltage to decrease to zero and wait for the next charge integration. Fig. 7 illustrates the conversion of an analog current signal into a latency spike. In this example, a sine-wave current is converted into latency spikes. It is noticeable that the larger input leads to a closer spike to the CLK, while the smaller input makes the spike further apart from the CLK. Such property fulfills the inversely proportional input and output signal relationship in latency encoding.

B. ISI Encoder

The number of neurons in the ISI encoder can be varied. As highlighted in Section II, the number of neurons and

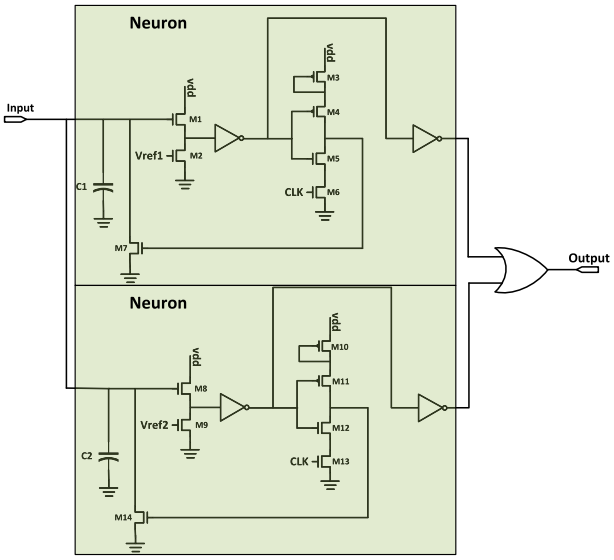


Fig. 8. Circuit schematic of the ISI encoder.

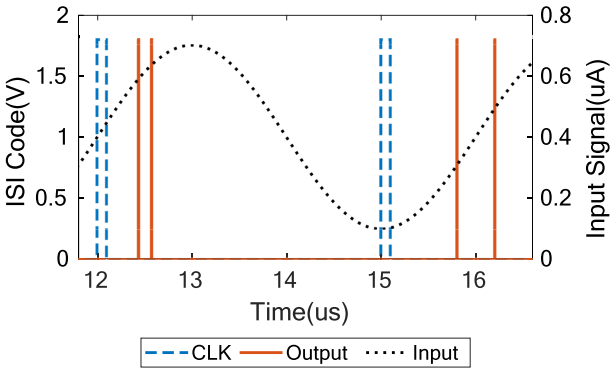


Fig. 9. Spikes of the ISI encoding module.

outputted spikes in the parallel structure have a linear relationship. The more neurons in the encoder, the more spikes in one sampling window. Increasing the number of neurons will lead to higher data capacity and yet increase the power consumption and silicon area. Since the neurons are the main source of power consumption and area used in the ISI encoder and the whole multiplexing encoder, increasing the number of neurons will drastically affect the power and area efficiency of the whole design. That is why there are only two neurons in the ISI encoder, as demonstrated in Fig. 8, which take different V_{ref} voltages. With the same input (excitation) current, these neurons fire spikes at different times, T_1 and T_2, according to (6). After the spikes are conveyed from the neurons, they will be integrated by an OR gate such that the ISI encoder generates a spike train. Therefore, the interval between the spikes can be written as

$$D = T_2 - T_1 = \frac{C_m(V_{\text{ref2}} - V_{\text{ref1}})}{I_{\text{in}}}. \tag{7}$$

This encoder transfers the input current signal to time intervals between spikes consisting of two neurons and an OR gate. As shown in Fig. 9, after being triggered by the CLK signal,

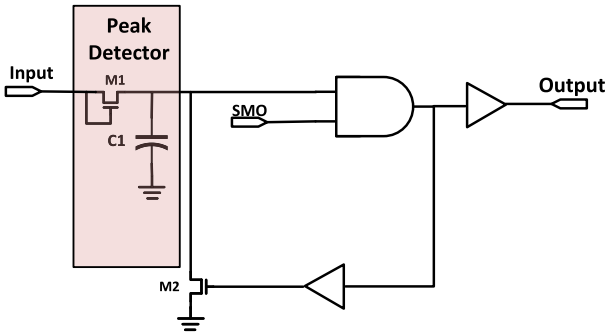


Fig. 10. Circuit schematic of the gamma alignment block.

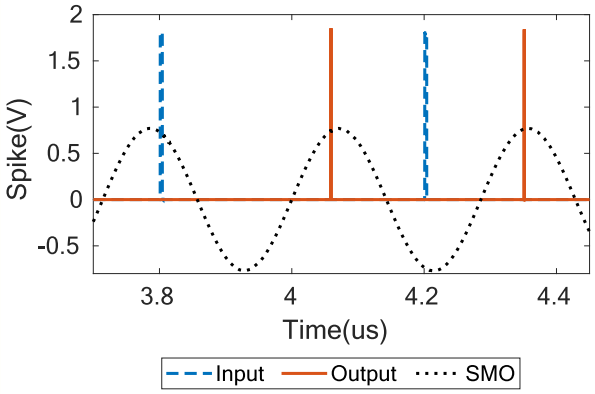


Fig. 11. Illustration of gamma alignment.

the integrate-and-fire mechanism starts, and thus an ISI spike train is fired. Noticeably, the input current ranges from 100 to 700 nA. It can be observed that the time interval between each pair of spikes varies with the current value between the onset signal and its corresponding spike train due to the different integration rates of the charge.

C. Gamma Alignment

The gamma alignment is a critical module in the introduced multiplexing temporal encoder. It offsets spikes to their next local maximum determined by the SMO. In the circuit implementation shown in Fig. 10, a spike detector consisting of a diode-connected transistor and a capacitor is used. With the diode connection, M1 transfers the incoming spike to C1 and stops the charge on C1 from leaking after that spike. The held spike is then fed into the AND gate and recognized as digital "1." Since the SMO is often modeled by sine wave and its frequency needs to be larger than that of the corresponding neurons, a function generator is often used to generate SMO. To ensure that the SMO is recognized as digital "1" only at local maximums, the threshold voltage of the AND gate is adjusted precisely such that the maximum point of the SMO aligns with the gate's threshold voltage. When the local maximum of SMO arrives after the input spike, an output spike will be generated through the buffer. Meanwhile, the switch M2 helps with the reset mechanism. The buffer-stabilized spike will trigger M2, and the held voltage will decrease to its initial level, waiting for the next input spike. Fig. 11 shows the

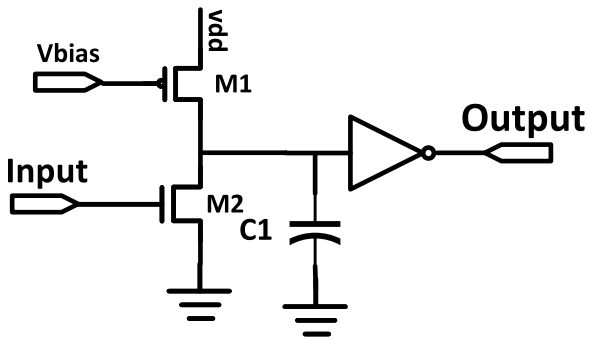


Fig. 12. Circuit schematic of the spike expander.

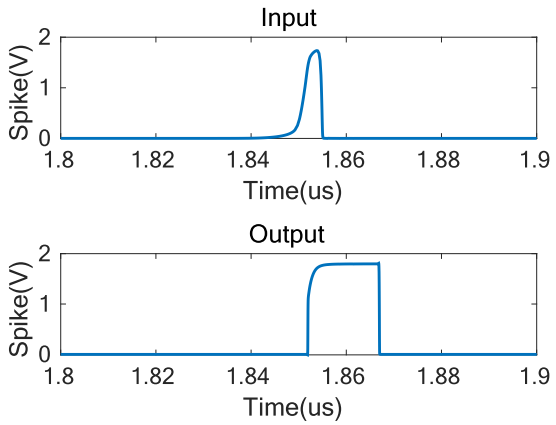


Fig. 13. Illustration of pulsewidth extension.

relationship between the input spikes, the SMO, and the shifted spikes. It can be observed that the outputted spikes of the ISI encoder are shifted to their corresponding local maximums of SMO.

D. Spike Expander

As mentioned in Section III, a lasting time of 10 ns for the incoming spike is required for the peak detector to work properly; else, the peak detector will be unable to charge the capacitor such that the voltage is recognized as digital "1." However, due to the reset mechanism, the ISI encoder can only output spikes that last for 1 ns. To this end, a spike expander is implemented to expand the pulsewidth of spikes to sidestep this issue, as illustrated in Fig. 12.

In the circuit implementation, an inverted spike is generated by the spike coming from the ISI encoder, and Vbias controls the inverted spike width since the gate voltage of the PMOS transistor controls the charging rate at the capacitor. Thus, the large capacitor to provide enough delay can be avoided. After that, an inverter is used to flip the inverted spike.

Fig. 13 depicts the property of spike time extension. It is noticeable that the expanded spike reaches 13 ns in pulsewidth while the input only lasts for 2 ns. Thus, the peak detector in the gamma alignment is capable of capturing those spikes and keeping them for the firing process of the AND gate.

With the help of internal reference frames, systems have outstanding robustness in noisy environments. Since the input

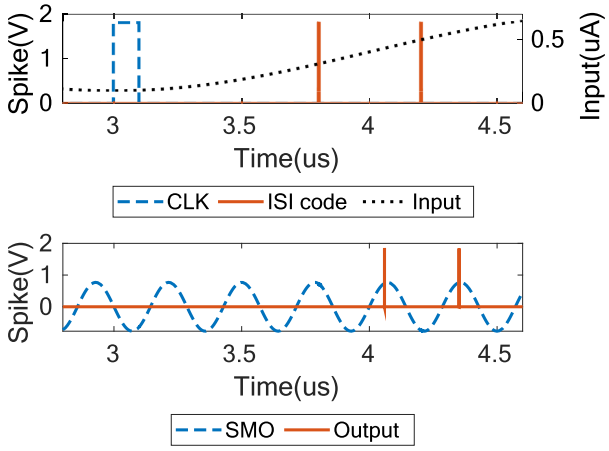


Fig. 14. Illustration of the signal flow.

data are interfered by noise, the temporal encoding part in the multiplexing temporal encoder generates output whose spike is shifted. However, due to the gamma alignment function of multiplexing encoding, the spike is shifted to its next local maximum determined by the SMO. As long as the noise-interfered spike is in the same gap between two SMOs' local maximums with the initial spike, the output spike of the multiplexing encoder remains unchanged, and thus, noises cannot affect the encoder's outputs. Therefore, the multiplexing temporal encoder can effectively improve the robustness of systems.

Fig. 14 illustrates the conversion of a current stimulus to a spike train using the introduced ISI-phase multiplexing encoder in post-layout simulation. The top part of the figure shows the ISI encoding functionality, where the input signal is transferred into a spike train. After that, with the help of a spike expander, the spikes in the original spike train are moved to their next maximum point of SMO correspondingly. Then the signal comes out of the encoder as a newly made spike train, which has high information density and robustness against noise.

V. Performance and Result Analysis

A. Performance Comparison Among Encoding Schemes

To demonstrate the effectiveness of different encoding schemes, SNNs with different encoders are implemented in PyTorch with the SpykeTorch simulator [21]. In this experiment, the MNIST dataset [22], the CIFAR-10 dataset [23], the SVHN dataset [24], and the spectrum sensing dataset [25] were used. For the MNIST dataset, 60 000 samples were used for training and 10 000 were used for testing. For the CIFAR-10 dataset, 50 000 samples were used for training and 10 000 samples were used for testing. For the SVHN dataset, 73 257 samples were used as training samples and 26 032 samples were used for testing. As for the spectrum sensing dataset, it was first introduced in a dynamic spectrum sharing system which resolves the spectrum scarcity in 5G systems [26]. While the *MIMO-OFDM* technologies would improve the spectral efficiency, the unused subcarriers still lead to suboptimal spectrum utilization efficiency. To sidestep this issue, the

TABLE II
PERFORMANCE COMPARISON OF CODE-LEVEL ENCODERS WITH THE MNIST, CIFAR-10, SVHN, AND SPECTRUM SENSING DATASETS

Encoder Type	Rate	TTFS	ISI	TTFS-Phase	ISI-Phase
MNIST	[27] – 83.0%	[27] – 85.0%	[28] – 90.0%	[this work] – 91.8%	[this work] – 93.8%
CIFAR-10	[28] – 79.7%	[29] – 77.4%	[31] – 83.7%	[this work] – 77.9%	[this work] – 83.8%
SVHN	[31] – 75.0%	[32] – 82.1%	[this work] – 82.8%	[this work] – 82.5%	[this work] – 86.4%
Spectrum Sensing	[this work] – 79.1%	[this work] – 85.0%	[this work] – 86.2%	[this work] – 86.3%	[this work] – 86.8%

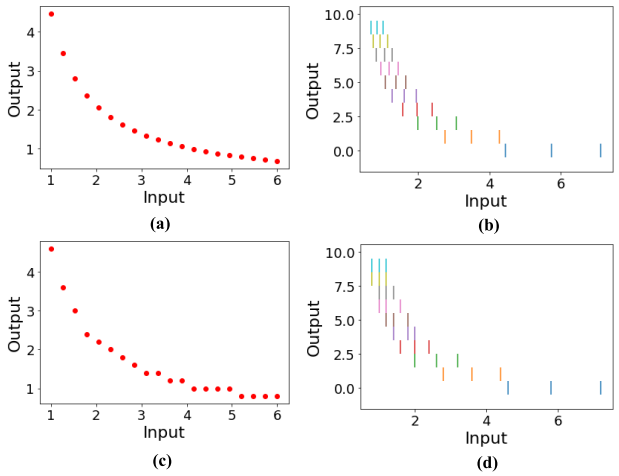


Fig. 15. Input–output relationship of encoders. Output of (a) TTFS encoder, (b) ISI encoder, (c) TTFS-phase encoder, and (d) ISI-phase encoder.

secondary users need to access the underused subcarriers, thus requiring to monitoring the spectrum utilization of primary users through spectrum sensing. It has been proven that the use of SNNs in predicting the underused spectrum band is an energy-efficient method [26].

In the first experiment, a rate encoder was implemented. Since the number of spikes is linearly proportional to the input in rate encoding, the output of the rate encoder in its mathematical model (using Python) was set to be a spike train with a spike number linearly proportional to the input. To realize TTFS encoding, the outputs of the TTFS encoder and inputs above the threshold value have an inverse logarithmic relationship. The larger the input, the smaller the output, as shown in Fig. 15(a). For instance, in the MNIST dataset, the gray scale value of each pixel was linearly assigned to the range from 0 to 6, and the output will be the first spike time.

In the second experiment, an ISI encoder was implemented. Multiple TTFS encoders with different thresholds were integrated, thus transferring the input data to a spike train with different time steps. As demonstrated in Fig. 15(b), the larger the input, the smaller the interspike intervals of the spike trains. With a similar prepossessing manner, the pixel value was first linearly assigned into the range from 0 to 8, and the output was represented by a vector array where each of them represents the time interval between spikes.

To integrate the multiplexing neural encoding methodology into both the TTFS encoder and the ISI encoder, gamma alignment is needed. The mathematical model of gamma alignment was achieved by updating the exact time of each spike to the next value in an arithmetic operation, where the frequency of SMO can be tuned by the common difference in the arithmetic operation. It is noticeable that the figures of the TTFS encoder and the TTFS-phase encoder are very similar. However, some inputs in the TTFS-phase encoder would lead to the same outputs because these outputs lie in the same period of SMO, which also occurs in the ISI-phase encoder, as illustrated in Fig. 15(c) and (d).

In verification, two various approaches were used to evaluate the performance of different encoding schemes according to the complexity of the dataset, for instance, a three-layer spiking convolution neural network was implemented for the MNIST and spectrum sensing datasets, while a ten-layer spiking convolutional neural network was implemented for the CIFAR-10 and SVHN datasets. All the models were trained with the spike-timing-dependent plasticity (STDP) training algorithm.

This performance was evaluated through a 12-GB NVIDIA Tesla K80 GPU with 13G RAM on Google Colab. Table II demonstrates the classification accuracy of different encoders for various datasets. For the MNIST dataset, the multiplexing encoding achieves 93.8% of accuracy, offering an improvement of up to 10.8% over alternative approaches. As for the CIFAR-10 dataset, the ISI-phase encoding scheme yields an accuracy of 83.8%, offering an improvement of up to 6%. For the SVHN dataset, the multiplexing temporal encoder offers 86.4% of accuracy while the rate encoder only reaches 75% of accuracy, which is 11.4% lower than that of the multiplexing encoder. Finally, for the spectrum dataset, the ISI-phase encoder achieves 86.8% accuracy, while the rate encoder only reaches 79.1%. In accounting for the complexity of the dataset and the neural network structure, the training/inferencing accuracy on each encoder behaves differently in different datasets. In short, the introduced multiplexing encoder has the potential to offer improved accuracy over alternative encoding schemes. Considering that the neural network used in this work is often smaller than these state-of-the-art works, the multiplexing temporal encoding scheme does help with higher classification accuracy. In conclusion, the introduced multiplexing temporal encoder can transform the datasets to be more classifiable for spike training and thus achieve higher accuracy for simple and complex problems.

B. Robustness, Power, and Area Analysis

In this section, the robustness, power consumption, and design area are examined based on the post-layout simulation.

TABLE III
POWER AND AREA COMPARISON OF MULTIPLEXING ENCODER BLOCKS

Blocks	ISI Encoding		Spike Expander	Gamma Alignment	Total
Power Consumption(μW)	Neuron / 1.22	2.50	0.01	0.07	2.58
Area(μm^2)	Neuron / 28×33	64×69	13×16	36.5×50.8	64×106

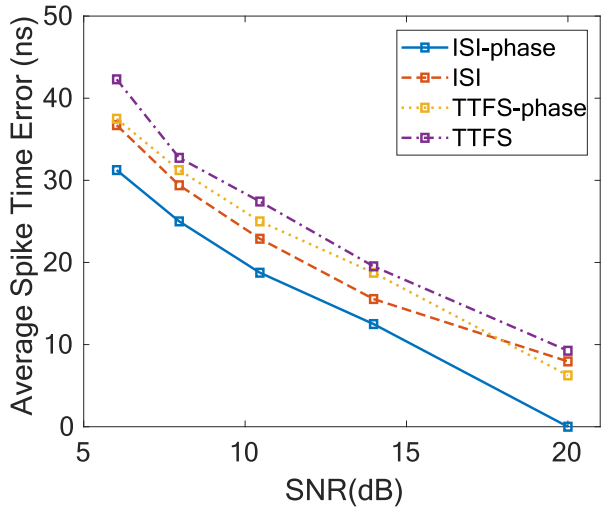


Fig. 16. Robustness comparison against noisy input of various encoders.

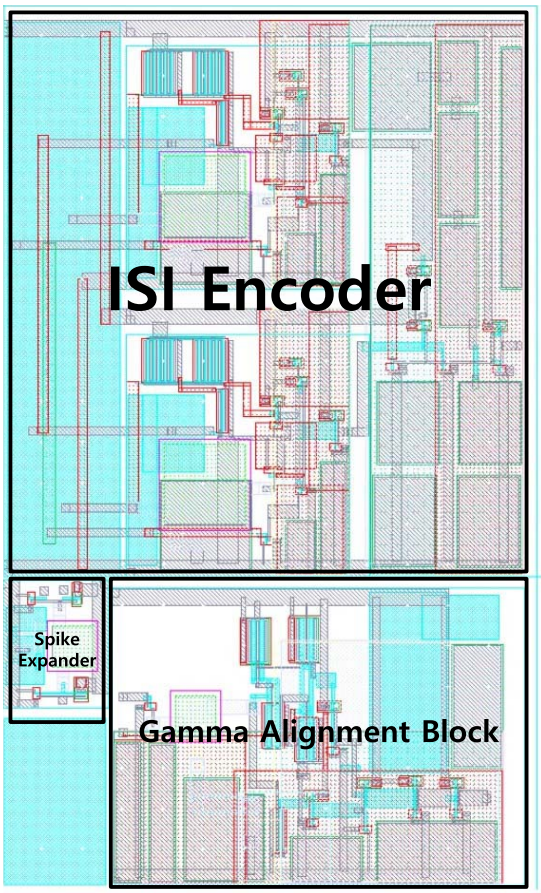


Fig. 17. Layout of the multiplexing temporal ISI-phase encoder.

To simulate the robustness, noises with various amplitudes were added to the inputs of the encoders. In this experiment, 16 output spikes were sampled and compared with the noise-free output, in which the average drifting error between two spike trains was recorded, as depicted in Fig. 16. It can be observed that the drifting error decreases as the input signal-to-noise ratio (SNR) increases. It can also be observed that the introduced multiplexing ISI-phase encoder demonstrates the lowest drifting error across all the SNR levels. Specifically, when the SNR was set to 20 dB, the input noise has zero impact on output spikes. In general, the average drifting error of the multiplexing ISI-phase encoder across all the SNR levels is over 25% lower than alternative approaches. Afterward, the robustness against supply voltage variation was examined. Our experimental results show that the final outcomes from our ISI-phase multiplexing encoder remain stable even when the supply voltage was reduced by 10%. As the reduction in supply voltage keeps increasing, defective spike trains would be generated while the spike width is significantly reduced/expand due to inaccurate SMO amplitude and triggering threshold at the logic AND gate. As a result, either additional spike expanders are needed to ensure the accuracy of later computations or more gamma alignment operations are required to remove the idle spikes.

Fig. 17 illustrates the layout of the introduced ISI-phase multiplexing encoder, occupying a total design area of 64×106 μm^2. An ISI encoder with two neurons takes up to 64×69 μm^2, where each neuron occupies 28×33 μm^2, while the spike expander and the gamma alignment take up to 13×16 μm^2 and 36.5×50.8 μm^2, respectively.

For ISI encoders, the number of neurons varies according to specific applications. It can be observed that the increasing number of neurons results in increased power consumption and area. The static power consumption and energy per spike of the introduced ISI-phase multiplexing encoder reach merely 2.58 μW and 95 fJ/spike, respectively, and the effective frame rate of the proposed circuit is 300 MHz. With the power-efficient design of the gamma alignment and spike expander, the reported power consumption is roughly twice as that of a single neuron.

Table III demonstrates the power consumption and design area of each function module and the entire ISI-phase multiplexing encoder. With two integrated neurons, the ISI encoder

TABLE IV
POWER AND AREA COMPARISON OF ENCODERS WITH OTHER WORKS

Encoder	Year	Process	Area(mm^2)	Power(μW)	VDD(V)
Rate[33]	2020	/	/	1240	2.5
TTFS	2022	180 nm	0.000924	1.22	1.8
TTFS[33]	2020	/	/	353.6	2.5
ISI	2022	180 nm	0.004416	2.5	1.8
ISI[34]	2022	45 nm	/	2.1	/
TTFS-Phase	2022	180 nm	0.004562	1.3	1.8
ISI-Phase	2022	180 nm	0.006784	2.6	1.8

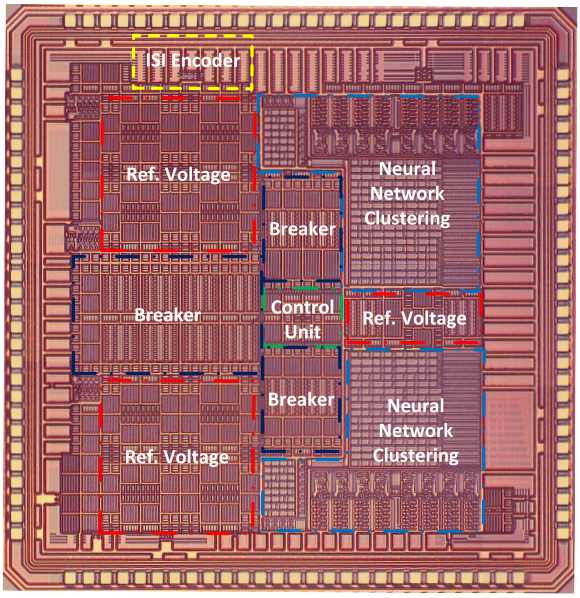


Fig. 18. Die micrograph of fabricated chip in CMOS 180-nm process.

consumes roughly 2.5 μW of power. Both the spike expander and gamma alignment ensure higher data capacity and better robustness against noise during the operation while only consuming 0.08 μW of power. The comparison of the introduced work and the state-of-the-art implementations are illustrated in Table IV. It is reasonable to conclude that our introduced ISI-phase multiplexing encoders consume less power than the state-of-the-art while a maintaining relatively reasonable silicon area.

C. Prototype of Multiplexing Temporal Encoder

The prototype of our introduced multiplexing encoder was built based on an on-chip ISI encoder that was fabricated in the 180-nm CMOS process, as illustrated in Fig. 18. The chip contains an ISI encoder with other basic function modules occupying 9 mm^2 of area while the ISI encoder takes up merely 0.024 mm^2 of silicon area. The ISI encoder consists of delay neurons occupying 924 μm^2 and an OR gate for the signal integration. The spike shifting function was then realized by the gamma alignment consisting of appropriate electronic components such as a diode, a capacitor, inverters, and an analog switch built on a printed circuit board (PCB).

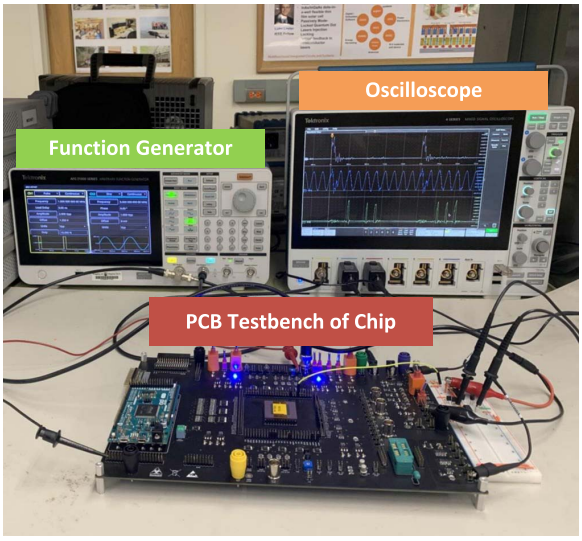


Fig. 19. Testbench for measurement.

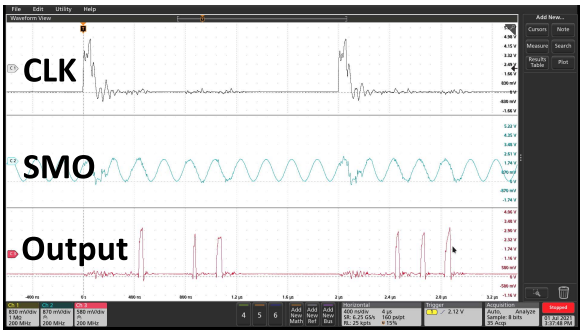


Fig. 20. Measurement result of the proposed multiplexing encoder.

The measurement testbench of the prototype is demonstrated in Fig. 19. A RIGOL DP832A dc power supply was used for the supply voltage and the constant voltage input of the proposed encoder. A Tektronix AFG31102 function generator was used to generate the global CLK and SMO signals. The global CLK signal is a pulse signal with 10% duty cycle at 1-MHz frequency, while the SMO is a 5-MHz sine wave. A Tektronix MSO46 mixed-signal oscilloscope was then used to verify the functionality of the introduced ISI-phase multiplexing encoder.

As illustrated in Fig. 20, the ISI encoder transfers the analog signals into spike trains with two parallel neurons. When a completed spike train was generated with a sampling window, the gamma alignment precisely offsets such a spike train to its next local maximum determined by the SMO signal. Such an experimental result demonstrates the successful implementation of the multiplexing encoder, in a way that is sufficient by integrating an ISI encoding and a phase encoding scheme.

VI. CONCLUSION

In this article, we presented a novel design of a multiplexing encoder that uses both the ISI encoding and phase encoding schemes. This encoder can transfer the current stimuli into the interspike interval, where each spike can then be offset to its next local maximum determined by the SMO. Therefore, the

intervals are precisely integer times of the SMO period. Our simulation results show that the multiplexing encoder achieves 93.8% accuracy on the MNIST dataset and 83.8% accuracy on the CIFAR-10 datasets, demonstrating improvements up to 10.8% and 6.4% over alternative approaches. Moreover, our multiplexing encoder also achieves 86.4% accuracy on the SVHN dataset, which is 11.4% more accurate than the others. With the spectrum sensing dataset, our multiplexing encoder is 7.7% more accurate than other encoders. To the best of our knowledge, our multiplexing encoder is the first IC implementation of the neural multiplexing ISI-phase encoder. The power consumption of our design is extremely low because of the power-efficient ISI encoder, spike expander and gamma alignment design. A prototype of the multiplexing temporal encoder is built based on an ISI encoder on a chip fabricated in the 180-nm CMOS technology. This work validated that it can transform data to a more classifiable structure by SNN and has improved robustness against noisy environments with very low power consumption and reasonable silicon area.

REFERENCES

[1] C. Mead, "Neuromorphic electronic systems," *Proc. IEEE*, vol. 78, no. 10, pp. 1629–1636, Oct. 1990.

[2] K. Bai and Y. Yi, *Opening the 'Black Box' of Silicon Chip Design in Neuromorphic Computing*. London, U.K.: IntechOpen, 2019.

[3] M. Davies et al., "Loihi: A neuromorphic manycore processor with on-chip learning," *IEEE Micro*, vol. 38, no. 1, pp. 82–99, Jan. 2018.

[4] F. Akopyan et al., "TrueNorth: Design and tool flow of a 65 mW 1 million neuron programmable neurosynaptic chip," *IEEE Trans. Comput.-Aided Desing Integr. Circuits Syst.*, vol. 34, no. 10, pp. 1537–1557, Oct. 2015.

[5] L. Deng et al., "Tianjic: A unified and scalable chip bridging spike-based and continuous neural computation," *IEEE J. Solid-State Circuits*, vol. 55, no. 8, pp. 2228–2246, Aug. 2020.

[6] S. Yang et al., "BiCoSS: Toward large-scale cognition brain with multigranular neuromorphic architecture," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 7, pp. 2801–2815, Jul. 2022.

[7] C. Zhao, K. Hamedani, J. Li, and Y. Yi, "Analog spike-timing-dependent resistive crossbar design for brain inspired computing," *IEEE J. Emerging Sel. Topics Circuits Syst.*, vol. 8, no. 1, pp. 38–50, Mar. 2017.

[8] C. Zhao, B. T. Wysocki, Y. Liu, C. D. Thiem, N. R. McDonald, and Y. Yi, "Spike-time-dependent encoding for neuromorphic processors," *ACM J. Emerg. Technol. Comput. Syst.*, vol. 12, no. 3, pp. 1–21, Sep. 2015.

[9] L. Wu, L. Yuan, and J. You, "Survey of large-scale data management systems for big data applications," *J. Comput. Sci. Technol.*, vol. 30, no. 1, pp. 163–183, Jan. 2015.

[10] S. Panzeri, N. Brunel, N. K. Logothetis, and C. Kayser, "Sensory neural codes using multiplexed temporal scales," *Trends Neurosci.*, vol. 33, no. 3, pp. 111–120, 2010.

[11] Z. Nadasdy, "Information encoding and reconstruction from the phase of action potentials," *Frontiers Syst. Neurosci.*, vol. 3, p. 6, Jul. 2009.

[12] K. Hamedani, "Energy efficient deep spiking recurrent neural networks: A reservoir computing-based approach," M.S. thesis, Bradley Dept. Elect. Comput. Eng., Virginia Tech., 2020.

[13] C. Liu et al., "A spiking neuromorphic design with resistive crossbar," in *Proc. 52nd Annu. Design Autom. Conf.*, Jun. 2015, pp. 1–6.

[14] C. Zhao et al., "Energy efficient spiking temporal encoder design for neuromorphic computing systems," *IEEE Trans. Multi-Scale Comput. Syst.*, vol. 2, no. 4, pp. 265–276, Apr. 2016.

[15] 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 *Proc. 35th Int. Conf. Comput.-Aided Design*, Nov. 2016, pp. 1–6.

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

[17] M. R. Rezaei, M. R. Popovic, S. A. Prescott, and M. Lankarany, "Synchrony-division neural multiplexing: An encoding model," *medRxiv*, pp. 1–28, Jan. 2021. [Online]. Available: https://www.medrxiv.org/content/early/2021/10/30/2021.10.29.21265658

[18] C. Kayser, M. A. Montemurro, N. K. Logothetis, and S. Panzeri, "Spike-phase coding boosts and stabilizes information carried by spatial and temporal spike patterns," *Neuron*, vol. 61, no. 4, pp. 597–608, 2009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0896627309000750

[19] T. Akam and D. M. Kullmann, "Oscillatory multiplexing of population codes for selective communication in the mammalian brain," *Nature Rev. Neurosci.*, vol. 15, no. 2, pp. 111–122, Feb. 2014.

[20] R. Naud and H. Sprekeler, "Sparse bursts optimize information transmission in a multiplexed neural code," *Proc. Nat. Acad. Sci. USA*, vol. 115, no. 27, pp. E6329–E6338, Jul. 2018.

[21] M. Mozafari, M. Ganjtabesh, A. Nowzari-Dalini, and T. Masquelier, "SpykeTorch: Efficient simulation of convolutional spiking neural networks with at most one spike per neuron," *Frontiers Neurosci.*, vol. 13, p. 625, Jul. 2019.

[22] L. Deng, "The MNIST database of handwritten digit images for machine learning research," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 141–142, Nov. 2012.

[23] A. Krizhevsky et al., "Learning multiple layers of features from tiny images," Univ. Toronto, Toronto, ON, Canada, Tech. Rep., 2009.

[24] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," Google, Mountain View, CA, USA, Stanford Univ., Stanford, CA, USA, Tech. Rep., 2011.

[25] O. Shears, K. Bai, L. Liu, and Y. Yi, "A hybrid FPGA-ASIC delayed feedback reservoir system to enable spectrum sensing/sharing for low power IoT devices ICCAD special session paper," in *Proc. IEEE/ACM Int. Conf. Comput. Aided Design (ICCAD)*, Nov. 2021, pp. 1–9.

[26] 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 *Proc. AAAI Conf. Artif. Intell.*, vol. 34, 2020, pp. 1292–1299.

[27] F. Nowshin and Y. Yi, "Memristor-based deep spiking neural network with a computing-in-memory architecture," in *Proc. 23rd Int. Symp. Quality Electron. Design (ISQED)*, Apr. 2022, pp. 1–6.

[28] V.-T. Nguyen, Q.-K. Trinh, R. Zhang, and Y. Nakashima, "STT-BSNN: An in-memory deep binary spiking neural network based on STT-MRAM," *IEEE Access*, vol. 9, pp. 151373–151385, 2021.

[29] Y. Cao, Y. Chen, and D. Khosla, "Spiking deep convolutional neural networks for energy-efficient object recognition," *Int. J. Comput. Vis.*, vol. 113, no. 1, pp. 54–66, May 2015.

[30] S. Park, S. Kim, H. Choe, and S. Yoon, "Fast and efficient information transmission with burst spikes in deep spiking neural networks," in *Proc. 56th Annu. Design Autom. Conf.*, Jun. 2019, pp. 1–6.

[31] Z. Wang, J. Liu, Y. Ma, B. Chen, N. Zheng, and P. Ren, "Perturbation of spike timing benefits neural network performance on similarity search," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 9, pp. 4361–4372, Sep. 2022.

[32] C. Ma, R. Yan, Z. Yu, and Q. Yu, "Deep spike learning with local classifiers," *IEEE Trans. Cybern.*, early access, Jul. 22, 2022, doi: 10.1109/TCYB.2022.3188015.

[33] S. Oh et al., "Hardware implementation of spiking neural networks using time-to-first-spike encoding," 2020, *arXiv:2006.05033*.

[34] V. S. Aadhitiya, J. B. Shaik, S. Singhal, S. M. Picardo, and N. Goel, "Design and mathematical modelling of inter spike interval of temporal neuromorphic encoder for image recognition," 2022, *arXiv:2205.09519*.

[35] T. Asai, Y. Kanazawa, and Y. Amemiya, "A subthreshold MOS neuron circuit based on the Volterra system," *IEEE Trans. Neural Netw.*, vol. 14, no. 5, pp. 1308–1312, Sep. 2003.

[36] K. Bai, Q. An, L. Liu, and Y. Yi, "A training-efficient hybrid-structured deep neural network with reconfigurable memristive synapses," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 28, no. 1, pp. 62–75, Oct. 2020.

[37] L. Camuñas-Mesa, B. Linares-Barranco, and T. Serrano-Gotarredona, "Neuromorphic spiking neural networks and their memristor-CMOS hardware implementations," *Materials*, vol. 12, no. 17, p. 2745, Aug. 2019.

[38] W. Gerstner and W. M. Kistler, *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. Cambridge, U.K.: Cambridge Univ. Press, 2002.

[39] X. Guo, X. Qi, and J. G. Harris, "A time-to-first-spike CMOS image sensor," *IEEE Sensors J.*, vol. 7, no. 8, pp. 1165–1175, Jun. 2007.

[40] K. Hynna and K. Boahen, "Space-rate coding in an adaptive silicon neuron," *Neural Netw.*, vol. 14, nos. 6–7, pp. 645–656, Jul. 2001.

[41] H. Jiang et al., "Cyclical sensing integrate-and-fire circuit for memristor array based neuromorphic computing," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2016, pp. 930–933.

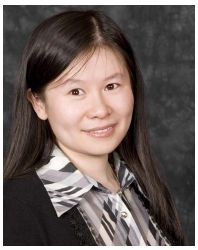
[42] M. Lankarany, D. Al-Basha, S. Ratté, and S. A. Prescott, "Differentially synchronized spiking enables multiplexed neural coding," *Proc. Nat. Acad. Sci. USA*, vol. 116, no. 20, pp. 10097–10102, May 2019.
[43] M. Lukosevicius, "Reservoir computing and self-organized neural hierarchies," Dept. Comput. Sci. Elect. Eng., Jacobs Univ., Bremen, Bremen, Germany, Tech. Rep., 2012.
[44] H. E. Michel, D. Rancour, and S. Iringentavida, "CMOS implementation of phase-encoded complex-valued artificial neural networks," in *Proc. ESA/VLSI Conf.*, 2004, pp. 551–557.
[45] S. E. Paraskevopoulou and T. G. Constandinou, "A sub-1 μW neural spike-peak detection and spike-count rate encoding circuit," in *Proc. IEEE Biomed. Circuits Syst. Conf. (BioCAS)*, Nov. 2011, pp. 29–32.
[46] B. Schrauwen, D. Verstraeten, and J. Van Campenhout, "An overview of reservoir computing: Theory, applications and implementations," in *Proc. 15th Eur. Symp. Artif. Neural Netw.*, 2007 pp. 471–482.
[47] C. Zhao, "Spike processing circuit design for neuromorphic computing," M.S. thesis, Bradley Dept. Elect. Comput. Eng., Virginia Tech., 2019.

Honghao Zheng (Student Member, IEEE) received the B.S. degree from the Dalian University of Technology, Dalian, China, in 2016, and the M.S. degree in electrical engineering from Boston University, Boston, MA, USA, in 2019. He is currently working toward the Ph.D. degree at the Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA, USA.

He joined Virginia Tech in September 2020. His research interests are currently focused on the analog/mixed-signal implementation of neuromorphic components and systems.

Kang Jun Bai (Member, IEEE) received the B.S. degree in electrical engineering and the M.S. degree in embedded electrical and computer systems from San Francisco State University, San Francisco, CA, USA, in 2014 and 2017, respectively, and the Ph.D. degree in electrical engineering from Virginia Tech, Blacksburg, VA, USA, in 2021.

He is currently a Research Electronics Engineer at the Air Force Research Laboratory Information Directorate (AFRL/RI), Rome, NY, USA. His research interests include mixed-signal integrated circuit design and nanoelectronic devices for bio-inspired computing systems, model emulation for deep learning benchmarks, and emerging applications' development for mobile edge computing.



Yang Yi (Senior Member, IEEE) is currently an Associate Professor with the Bradley Department of Electrical Engineering and Computer engineering, Virginia Tech, Blacksburg, VA, USA. Her research interests include VLSI circuits and systems, computer-aided design (CAD), neuromorphic architecture for brain-inspired computing systems, and low-power circuits' design with advanced nano-technologies for high-speed wireless systems.