ANP-I: A 28-nm 1.5-pJ/SOP Asynchronous Spiking Neural Network Processor Enabling Sub-0.1-μJ/Sample On-Chip Learning for Edge-AI Applications

Jilin Zhang, *Member, IEEE*, Dexuan Huo, *Student Member, IEEE*, Jian Zhang, *Graduate Student Member, IEEE*, Chunqi Qian, *Student Member, IEEE*, Qi Liu, *Student Member, IEEE*, Liyang Pan, *Senior Member, IEEE*, Zhihua Wang, *Fellow, IEEE*, Ning Qiao, *Member, IEEE*, Kea-Tiong Tang, *Senior Member, IEEE*, and Hong Chen, *Senior Member, IEEE*

***Abstract*— Reducing learning energy consumption is critical to edge-artificial intelligence (AI) processors with on-chip learning since on-chip learning energy dominates energy consumption, especially for applications that require long-term learning. To achieve this goal, we optimize a neuromorphic learning algorithm and propose random target window (TW) selection, hierarchical update skip (HUS), and asynchronous time step acceleration (ATSA) to reduce the on-chip learning power consumption. Our approach results in a 28-nm 1.25-mm^2 asynchronous neuromorphic processor (ANP-I) with on-chip learning energy per sample less than 15% of inference energy per sample. With all weights randomly initialized, this processor enables on-chip learning for edge-AI tasks such as gesture recognition, keyword spotting, and image classification, consuming sub-0.1 μJ of learning energy per sample at 0.56 V and 40-MHz frequency while maintaining >92% accuracy for all tasks.**

***Index Terms*— Application-specified integrated circuit (ASIC), asynchronous circuits, neuromorphic computing, on-chip learning, spiking neural network (SNN).**

I. Introduction

THE field of artificial intelligence has seen remarkable advancements in recent years, with deep neural networks (DNNs) playing a key role in various domains [1], including image classification [2], [3], [4], voice recognition [5], [6], and autonomous driving [7]. Despite their remarkable capabilities, however, the dense computation required by DNNs results in high power consumption, limiting their deployment in edge applications. Spiking neural network (SNN) is a promising alternative, as it leverages spatiotemporal sparse spikes for information transfer and event-based models for processing. The event-driven nature of SNNs, coupled with their sparse data acquisition and processing capabilities, makes them highly suitable for edge-AI applications.

Numerous studies have focused on developing energy-efficient neuromorphic processors. For instance, TrueNorth [8], which integrates 1M neurons and 256M synapses, consumes 65 mW of power in a typical application. However, its large die area (4.3 cm^2) makes it economically unfeasible to be deployed at the edge. Similarly, Tianjic [9] implements a unified architecture that combines neuromorphic computing and deep learning. It integrates 40k neurons and 10M synapses but consumes nearly 950 mW of power on random input. On the other hand, DYNAPs [10] implement 1k analog neurons and 64k analog synapses and consume only 0.85 mW at 1.3 V. However, the low-precision computation of analog neurons and synapses limits DYNAPs to simple tasks such as the classification of poker card symbols.

The aforementioned neuromorphic processors are designed solely for inference, meaning that the SNNs are trained off-chip. However, on-chip learning has been garnering more research interest in the field as it enables on-the-fly adaptation to changing environments. Given the varied environments that edge-AI devices will face, on-chip learning can help avoid performance degradation by facilitating fast adaptation. In addition, on-chip learning eliminates the need to send personal data to the cloud for edge devices. Despite these desirable features, implementing on-chip learning in neuromorphic processors for edge-AI applications is exceedingly challenging.

Loihi [11] is a neuromorphic processor that supports programmable on-chip learning and integrates 130k neurons and 130M synapses. However, its power consumption of around 300 mW makes it unsuitable for edge-AI applications.

Manuscript received 17 June 2023; revised 2 October 2023 and 5 December 2023; accepted 12 January 2024. Date of publication 30 January 2024; date of current version 25 July 2024. This article was approved by Associate Editor Sanu K. Mathew. This work was supported in part by the National Science and Technology Major Project from the Minister of Science and Technology, China, under Grant 2018AAA0103100; and in part by the National Natural Science Foundation of China under Grant 92164110 and Grant 62334014. *(Corresponding author: Hong Chen.)*

Jilin Zhang, Dexuan Huo, Jian Zhang, Chunqi Qian, Qi Liu, Liyang Pan, Zhihua Wang, and Hong Chen are with the School of Integrated Circuits, Tsinghua University, Beijing 100084, China (e-mail: zhangjil22@mails.tsinghua.edu.cn; hdx20@mails.tsinghua.edu.cn; jian-zha22@mails.tsinghua.edu.cn; qcq21@mails.tsinghua.edu.cn; qliu19@mails.tsinghua.edu.cn; panly@tsinghua.edu.cn; zhihua@tsinghua.edu.cn; hongchen@tsinghua.edu.cn).

Ning Qiao is with SynSense, Chengdu 610000, China.

Kea-Tiong Tang is with the Department of Electrical Engineering, National Tsing Hua University, Hsinchu 30013, Taiwan (e-mail: kttang@ee.nthu.edu.tw).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/JSSC.2024.3357045.

Digital Object Identifier 10.1109/JSSC.2024.3357045

ROLLS [12], on the other hand, integrates 256 analog neurons and 128k analog synapses and uses spike-driven synaptic plasticity (SDSP) learning rules to perform on-chip learning for simple tasks, such as classifying cars and motorbikes. Its analog design results in only 4-mW power consumption. However, its analog design limits it to perform more complex tasks in real-world environments. ODIN [13] is a digital neuromorphic processor consisting of 256 neurons and 64k synapses using SDSP learning rules. With fewer synapses than ROLLS, ODIN achieves an accuracy of 84.5% on the MNIST [14] dataset consuming only 0.477 mW power. ReckOn [15] realizes a spiking recurrent neural network (SRNN) with 272 neurons and 132k synapses with e-prop learning rules. It achieves an accuracy of 87.3% on the IBM DVS-Gesture dataset [16].

It is worth noting that the power consumption mentioned above for on-chip learning neuromorphic processors is only for inference. In [17], a classifier with on-chip SGD-based training capability is implemented. The learning energy overhead is 61.9%. Amravati et al. [18] propose a time-domain mixed-signal neuromorphic accelerator with on-chip reinforcement learning, and the learning energy overhead is 117.4%. Park et al. [19] introduce a neuromorphic image classification processor that realizes on-chip learning through a direct spike-only feedback algorithm with 7.5% learning energy overhead.

In order to realize a low-power neuromorphic processor enabling on-chip learning with low learning energy overhead for edge-AI applications, we propose a 28-nm 1.25-mm^2 asynchronous neuromorphic processor (ANP-I) [20] with 8-b/10-b weight precision that enables on-chip learning for edge-AI tasks in this article. ANP-I uses a hierarchical update skip (HUS) mechanism to reduce learning energy and a randomly selected target window (TW) to reduce the number of spikes used in learning. We adopted asynchronous design methodology, which enables event-driven computation in ANP-I, and fabricated a prototype chip to verify our design. We also provide two demonstrations to showcase the abilities of the ANP-I processor. The contributions of this article are listed as follows.

1) A randomly selected TW is proposed in the learning process, which reduces over 90% of the spike number in the learning sample without noticeable accuracy degradation.
2) A set of optimization techniques is put forward to reduce learning and inferencing power consumption, including HUS used to reduce the learning energy cost and asynchronous time step acceleration (ATSA) to reduce the latency and power consumption in both learning and inferencing processes.
3) A prototype chip is fabricated to verify our design, which presents advanced performance over a wide range of applications, such as image classification, gesture and physiological signal recognition, and keyword spotting.

Besides, two real-life application examples, gesture recognition with spikes from a dynamic vision sensor (DVS) and signals from a surface electromyography (sEMG) sensor, are demonstrated to further verify the performance of ANP-I in real-life applications.

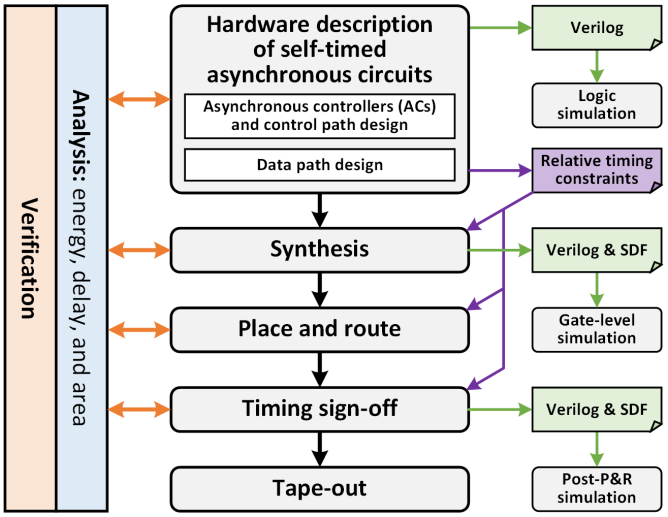


Fig. 1. Design flow of the self-timed asynchronous circuit.

The rest of this article is organized as follows. Section II describes the design flow of self-timed asynchronous logic. Section III introduces the working flow and overall architecture of ANP-I. In Section IV, the learning algorithm and hardware implementation of ANP-I are explained in detail, along with the proposed key features. Section V shows the measurement results, and Section VI discusses the semi-supervised learning process and others. Finally, Section VII concludes this article.

II. Design Flow of Self-Timed Asynchronous Logic

The potential benefits of self-timed asynchronous logic, such as increased speed, lower power consumption, and higher modularity, make it an attractive alternative to synchronous logic. However, the lack of EDA tools limits the usage of asynchronous logic. In our work, commercial EDA tools are adopted to design the self-timed asynchronous circuits, and the design flow is illustrated in Fig. 1. First, we use hardware description language, such as Verilog or very high-speed integrated circuit hardware description language (VHDL), to describe self-timed asynchronous circuits. With the design of asynchronous controllers (ACs), control paths, and data paths, relative timing constraints (RTCs) need to be exacted by designers. Then, synthesis is performed by commercial EDA tools, such as design compiler (DC). The RTCs help DC optimize the asynchronous circuits. Similarly, the RTCs are used for optimization in place and route and timing sign-off steps. Note that the analysis, verification, and simulation steps are the same as those for synchronous circuits. The differences in the design flow mainly lie in ACs and RTCs. The following subsections will discuss the differences in detail.

A. Asynchronous Controllers (ACs)

Instead of the global clocks in synchronous logic, handshake signals are adopted in asynchronous logic to realize

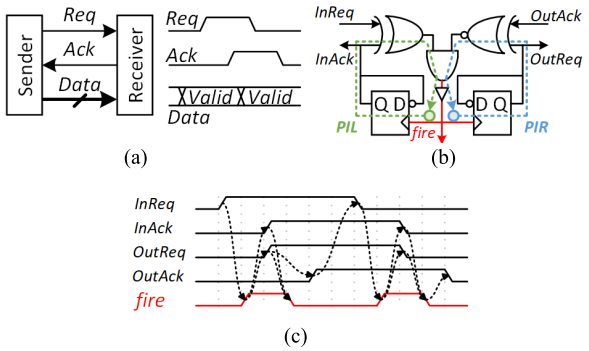


Fig. 2. (a) Two-phase bundled data protocol, (b) phase-decoupled Click, and (c) logic waveform of phase-decoupled Click.

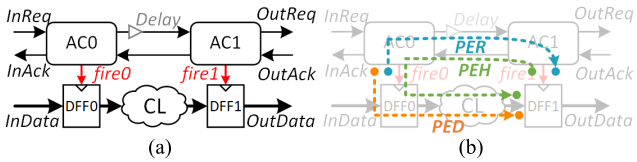


Fig. 3. (a) Asynchronous pipeline and (b) timing path of the asynchronous pipeline.

Path type	Check type	Relative timing constraints
Internal	Min-pulse width	*min(Delay(PIL), Delay(PIR)) > TPW*
External	Setup	*Delay(PER) - Delay(PED) > TSU*
	Hold	*Delay(PEH) < THO*

Fig. 4. RTCs of phase-decoupled Click-based self-timed asynchronous circuit.

synchronization. As shown in Fig. 2(a), request (Req) and acknowledge (Ack) handshake signals are used for data transmission between the sender and the receiver following a two-phase bundled data protocol. The sender inverts the Req signal when data are available to be sent. When the receiver detects the inversion of the Req signal, it captures the data and inverts the Ack signal, informing the sender that the receiver has captured the data. Then, the sender sends new data, and Req is inverted only after the data are valid and stable.

Here, we choose Click [21], an AC that follows a two-phase bundled data protocol, to build the control path. Although Click has higher hardware overhead than other ACs such as C-element because of the DFFs in Click, as illustrated in Fig. 2(b), the DFFs separate combinational loops. The phase-decoupled Click has one input channel and one output channel. We denote the request and acknowledge signals in input and output channels as InReq, InAck, OutReq, and OutAck. An input channel is full when its InReq and InAck signals have different logic values. An output channel is empty when its OutReq and OutAck signals have the same logic values. As shown in Fig. 2(c), when the InReq inverts, indicating a valid input event, the input channel becomes full and the fire signal is pulled up. The posedge of the fire signal inverts the OutReq and InAck signals by the DFFs, which resets phase-decoupled Click to an initial state.

B. Control and Data Paths Design

Control and data paths are designed using hardware description language (Verilog/VHDL). Compared with synchronous circuits, the only difference in self-timed asynchronous logic is that the control path is built by ACs without any global clock. Fig. 3(a) shows a two-stage asynchronous pipeline. The output channel of AC0 is connected with the input channel of AC1. When InData is valid and stable, the InReq of AC0 is inverted. AC0 generates a pulse of the fire0 signal, which is used to control DFF0 to capture InData. In the meantime, the InAck of AC0 is inverted to the signal that InData is captured by DFF0 safely. Preceding AC will invert the InReq of AC0 only after it has received InAck from AC0.

C. Relative Timing Constraints (RTCs)

RTCs are important for correct operation in asynchronous logic. A comprehensive RTC's taxonomic tree is presented in [22]. In our design, the RTCs are simplified as our ACs are based on phase-decoupled Clicks. The RTCs in our design flow are illustrated in Fig. 4, where only three types of RTCs need to be satisfied.

The first timing constraint is that the internal pulsewidth satisfies min(Delay(PIL), Delay(PIR)) > TPW. PIL and PIR stand for the path inside phase-decoupled Click shown in Fig. 2(b). PIL starts from the left DFF, through the XOR and AND gates, and ends at the output of delay gate. PIR starts from the right DFF, through the XNOR and AND gates, and ends at the output of delay gate. Delay(PIL) stands for the delay in path PIL, and TPW is the minimum pulsewidth required by D-flip-flop (DFF). If the constraint is not satisfied, delays should be added at the output of the AND gate. Fig. 3(b) illustrates the timing path of an asynchronous pipeline. Path PER starts from fire0, through the delay gate in the request line, and ends at fire1. Path PED starts from fire0, through combinational logic in the data path, and ends at the D-port of DFFs controlled by fire1. Path PEH starts from fire2, through the acknowledge line between AC1 and AC0 and combinational logic in the data path, and finally ends at the D-port of DFFs controlled by fire1. To ensure correct computation, the OutReq from AC0 is delayed and sent to AC1 as its InReq to satisfy the external timing constraints, which include setup and hold timing constraints. The setup timing constraint requires Delay(PER) − Delay(PED) > TSU, and the hold timing constraint requires Delay(PEH) > THO. TSU and THO stand for setup and hold time required by DFFs, respectively.

III. Hardware Implementation

ANP-I is a fully asynchronous design following the two-phase handshake protocol without a global clock. Fig. 5(a) shows the dataflow of ANP-I. ANP-I receives spike events from various sensors, such as the DVS, the dynamic audio sensor (DAS), and other sensors. The hidden layer receives input spike events and generates hidden spike events, which are then processed in the output layer. Errors are generated using output spike events and sent directly to both hidden layer and output layer. Only spike events within TW are used

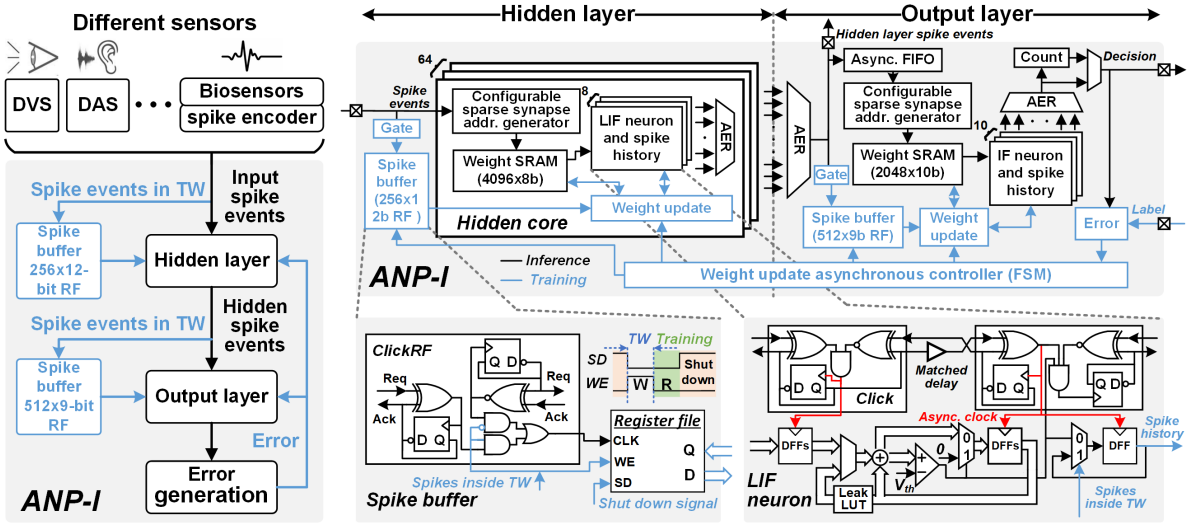


Fig. 5. Dataflow (a) and overall architecture (b) of ANP-I.

for training. Therefore, spike events within the TW are the only ones stored in the spike buffer for learning.

Fig. 5(b) illustrates the overall architecture of ANP-I. The hidden layer consists of 64 parallel cores. Each core contains eight leaky integrate-and-fire (LIF) neurons with configurable sparse connectivity between the input and hidden layers through 8-bit weights stored in a 4-kB SRAM (8 neurons $\times$ 512 synapses $\times$ 8 bits $=$ 4 kB). For each spike event, the corresponding synaptic weights are read and sent to one or more (up to four) LIF neurons, depending on the configured sparse connectivity. The LIF neurons work in parallel with shared leakage and threshold parameters. Spike events from the hidden layer are converted to spike addresses by an address event representation (AER) circuit and sent to an asynchronous FIFO. The spike events in an output layer are processed as long as the asynchronous FIFO is not empty.

The output layer consists of ten integrate-and-fire (IF) neurons, which are identical to the LIF neurons, except for the absence of the leaky mechanism. The spike buffers in the hidden and output layers are used to store spike events inside the TW and remain shut down during inference to reduce neuron, as shown in the lower half of Fig. 5(b). The detailed implementation of the spike buffer and LIF ANP-I is controlled by the ACs. As a result, each module only works when and where it is needed, thus improving the energy efficiency of ANP-I.

ANP-I supports an SNN with a maximum size of 1024 (input)-512-10. This scale of SNN is capable of adapting to the demands of many relatively complex tasks and can achieve over 90% accuracy on multiple tasks. When more complex tasks need to be performed, larger SNNs can be achieved by connecting multiple ANP-I chips, thereby achieving higher accuracy on complex tasks. To strike a balance between on-chip learning accuracy and memory usage, we set the bit width of the hidden layer weights to 8 bit. This choice allows us to achieve high accuracy while minimizing the memory required for weight storage. In addition, the output layer has a greater influence on the overall on-chip learning accuracy but requires fewer weights compared to the hidden layer. Therefore, we set the bit width of output layer weights to 10 bit, enhancing accuracy without significant memory overhead.

In the hidden layer, we divide 512 neurons into 64 hidden layer cores, that is, each core has eight neurons. This will increase the update skip rate and reduce training energy consumption. The entire learning process of a hidden core can be skipped when no neuron is spiked in the hidden layer core. The fewer neurons in a single hidden core, the higher the probability that the training process within that core will be skipped, resulting in a more efficient on-chip learning process. Moreover, 64 parallel hidden cores will increase computation parallelism and reduce inferencing and learning latency.

The blue blocks in Fig. 5 represent the learning modules, which receive requests and work only during the learning process and remain idle during the inference process. As illustrated in Fig. 6, when spike events from the TW are received by the input layer, they are forwarded to the hidden neurons and stored in the spike buffer in the hidden layer. The spike events generated by neurons in the hidden layer are passed to the output layer, and when they are forwarded to the output neurons, they are stored in the spike buffer in the output layer. After all the spike events are processed, the spike events from the spike buffer are read simultaneously by the hidden and output layers, and the training process begins. Each hidden core, as well as the output layer, has an ΔW calculation circuit. As a result, the ΔW calculations are running in parallel, and the training time is reduced.

IV. Energy-Efficient On-Chip Learning Processor

A. Sparse Target Propagation Algorithm

Here, we propose a sparse target propagation (S-TP) algorithm, which is a classification algorithm that uses a fully connected feedforward multilayer structure and supervised

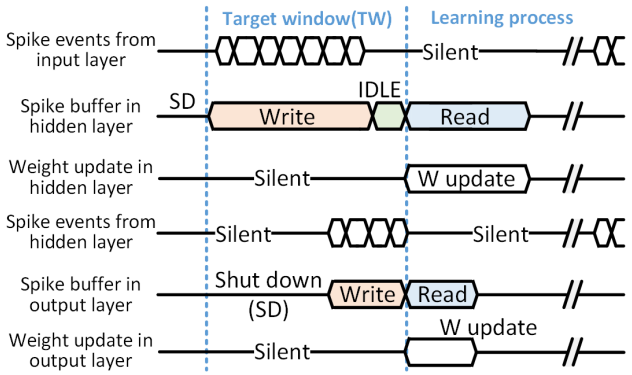


Fig. 6. Control flow for on-chip learning in ANP-I.

learning. Similar to the feedback alignment algorithm [23], [24], the fixed and random asymmetric feedback weights are used to solve the weight transport problem. The GLSNN algorithm [25], such as the S-TP algorithm, also uses global feedback alignment and local plasticity. The difference lies in the forward pass. In the S-TP algorithm, only spike events inside TW are forward before the backward pass. In the GLSNN algorithm, learning happens only after the entire forward pass. Compared with S-TP, the GLSNN algorithm requires more memory to store spike events and has a higher latency because more spike events are used in the forward pass. Compared with difference target propagation (DTP) [26], the S-TP algorithm omits the reconstruction loss for updating feedback weights in DTP. All feedback weights are first randomly initialized and then fixed during training. Moreover, the S-TP algorithm is an event-driven algorithm, which means that no weight updates will happen on the synapse of a silent neuron. In the S-TP algorithm, TW is used to capture the necessary information used for training. TW is a period of time during which spike events are captured and saved in buffers, then used for training. TW offers three advantages.

1) Only the spikes in TWs are used for training, and memory overhead for training is significantly reduced (0.3% in ANP-I) as a result.
2) The number and length of TW are configurable, which offers flexibility in the tradeoff between accuracy and training energy consumption.
3) TW allows fewer spike events for training and fewer updates during training, which further saves training energy overhead.

However, only the spike events in a few time steps are involved in the forward pass with the TW method. Therefore, the TW method is not suitable for applications with rate coding spike events, which require many time steps to represent the data accurately. However, the TW method is suitable for other coding schemes, such as time-to-first spike coding and rank-order coding. Previous works [27], [28] have also proposed the selective use of time steps for training to conserve energy. However, there are notable differences between the previous works and our approach. For instance, Meng et al. [27] employ backpropagation on every randomly selected time step, while our method employs backpropagation only once at the end of selected to our approach, which are less than 1/10 of the total length. Wang et al. [28] proposed an adaptive training window selection method to determine the input segment size and influences feature extraction of the interest area. The slight change in training window size in [28] has a great impact on the performance of the network model, while the TW selection approach is more robust to the change of TW length. Our benchmarks demonstrate that the TW method works well for applications with spike events from neuromorphic sensors (such as DVS and DAS) and bioelectrical signals such as EMG, ECG, and EEG.

B. Target Generation and ΔW Calculation

In our S-TP algorithm, the output neuron corresponding to the label is regarded as a target neuron. As shown in Fig. 7(a), when the target neuron spikes in TW, the target is 0, otherwise 1. When a non-target output neuron spikes in TW, its target is -1, otherwise 0. The target is encoded in 20 bits. Targets[9:0] stores the sign of target and targets[19:10] stores the value of target. The calculations of ΔW in the output layer and hidden layer are different, as described in (1) and (2), respectively. In (1), $\Delta W_o^{(n,m)}$ is the ΔW of synapse that connects the hidden neuron m and the output neuron n, lr stands for the learning rate, $S_{o-1}^m(t_{0,1})$ represents the number of spikes emitted by presynaptic neuron m in layer $o-1$ (previous layer of layer o), and T_n is the target of neuron n. In (2), $\Delta W_h^{(n,m)}$ is the ΔW of the synapse that connects hidden neuron m in layer $h-1$ (the previous layer of layer h) and hidden neuron n in layer h, $\mathrm{Sign}(S_h^n(t_{0,1}))$ is the sign function of $S_h^n(t_{0,1})$, $S_{h-1}^m(t_{0,1})$ is the number of spikes emitted by presynaptic neuron m in layer $h-1$, $W_f^{(o,h)}$ represents the fixed random weight between hidden neuron h and output neuron o, and T_o stands for the target assigned to output neuron o

$$\Delta W_o^{(n,m)} = lr \cdot S_{o-1}^m(t_{0,1}) \cdot T_n \tag{1}$$

$$\Delta W_h^{(n,m)} = \mathrm{Sign}\left(S_h^n(t_{0,1})\right) S_{h-1}^m(t_{0,1}) \sum_o \left(W_f^{(o,h)} T_o\right). \tag{2}$$

The weight update circuit implementing the functions in (1) and (2) is shown in Fig. 7(b). The fixed random weight is saved in a 240-bit LUT, which remains unchanged after configuration. Each hidden layer core has 10×8 (output neuron number $\times$ neuron number in the hidden layer core) feedback weights with 3-bit precision. Therefore, a 240-bit LUT is sufficient to store the feedback weights. Ten Sel circuits are designed to choose the 3-bit feedback weights from the LUT according to the generated targets, and all their outputs are summed up in the adder tree to get ΔW.

C. Target Window Selection

The location of the TW is chosen according to the number of spikes in a time step. This process can be performed both on-chip and off-chip. For off-chip selection, time steps in which the spike numbers are over a preset threshold (threshold is selected based on the prior knowledge of the target task) are selected as the TW candidates [as shown in Fig. 8(a)]. Then, a TW is randomly selected from the TW candidates. The reason for random selection is to improve the accuracy

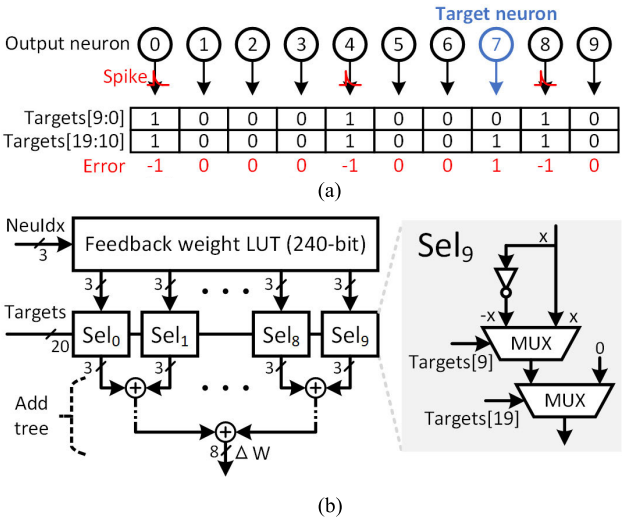


Fig. 7. (a) Example of target generation and (b) ΔW calculation circuit.

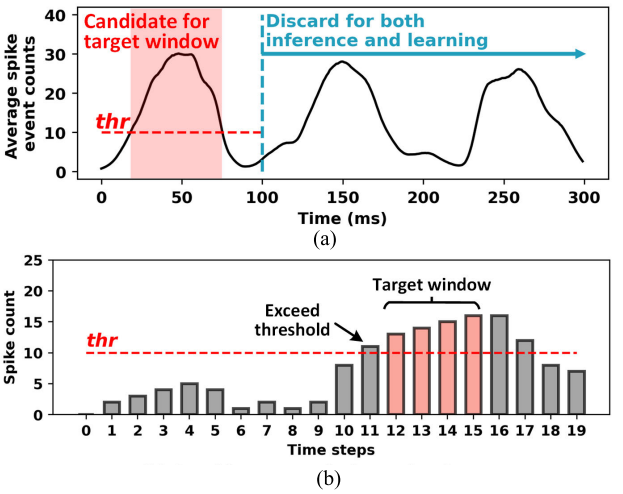


Fig. 8. (a) Off-chip and (b) on-chip target window selection.

of our approach by introducing variability during the training process. When ANP-I is deployed on edge devices, on-chip TW selection [see Fig. 8(b)] is adopted to determine the location of TW in real time. If the spike count in a time step is bigger than a pre-defined threshold, the next time step is the beginning of the TW. In ANP-I, the threshold value is set to 10 and the TW length is set to four time steps to achieve the highest on-chip learning accuracy on the N-MNIST dataset. The TW length and threshold are selected based on initial simulations on GPU. As shown in Fig. 9, ANP-I achieves the highest on-chip learning accuracy when the TW length is set to four on the N-MNIST dataset. With other TW lengths, the on-chip learning accuracy realized by ANP-I ranges from 94.5% to 95.7%.

The length of the TW is different for different tasks. For instance, in the training of the N-MNIST dataset, the highest accuracy is achieved when the TW length is set to four time steps. In the training of the DVS-Gesture dataset, the highest accuracy is obtained when the TW length is set to 12 time steps. Fig. 9 shows the TW length versus on-chip learning

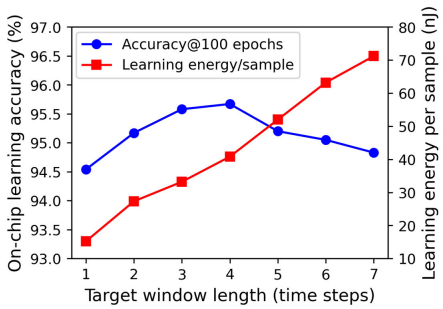


Fig. 9. Measured target window length versus on-chip learning accuracy and energy when on-chip learning on the N-MNIST dataset.

accuracy and energy when learning on the N-MNIST dataset. Note that the learning energy increases nearly linearly with the TW length, and the accuracy peaks at four time steps. In ANP-I, ΔW is calculated based on (1) and (2). $\text{Sign}(S_h^n(t_{0,1}))$ in (2) means that ANP-I only records whether a neuron spikes or not in TW, no matter how many spikes the neuron generates in the TW. Therefore, during on-chip learning, $\text{Sign}(S_h^n(t_{0,1}))$ in (2) is the same no matter how many spike inputs. The longer the length of TW, more neurons will generate more than one spike in the TW. However, ΔW is the same no matter whether there are multiple generated spikes or only one spike in the TW as input, which introduces bias during on-chip learning, resulting in lower accuracy.

As only the spikes in the TW are used for training in our algorithm, 96.3% of the spikes on N-MNIST, 98.1% of the spikes on DVS-Gesture, 94.5% of the spikes on N-TIDIGIT, and 92% of the spikes on SeNic are discarded. The software simulation results show that SNN trained without TW has a 0.83% accuracy improvement compared with SNN trained with TW when learning on the N-MNIST dataset.

D. Hierarchical Update Skip Mechanism

As mentioned above, we put forward an HUS mechanism, which skips updates hierarchically to speed up the training process and reduce training energy. The HUS mechanism includes three steps: ChipSkip, CoreSkip, and H/OSkip. SkipChip is the top step, in which all updates are skipped. As shown in Fig. 10(a), when only the target neuron in the output layer spikes in TW, all the values of the zero flags in targets are zeros. This means that no spike in the spike buffer will be sent out for training. Meanwhile, the spike buffer enters shutdown mode, resulting in all updates being skipped in all layers. CoreSkip is the middle step, in which updates in a single core will be skipped. When no neuron in a core spikes in TW, all the spike history of the core will be zeros, which will result in spike events stopped at the CoreGate module, as depicted in Fig. 10(b). Thus, all updates in this hidden core are skipped. H/OSkip is the bottom step, in which all the weight updates for the synapses corresponding to this spike event will be skipped. H/OSkip includes HidSkip and OutSkip. In the HidSkip step [see Fig. 10(c)], a synapse address is generated by the sparse address generation circuit. In the hidden cores, if the spike history corresponding to the synapse is zero, the weight update for this synapse will be skipped. HidSkip works the same as OutSkip, with the difference that HidSkip targets the hidden

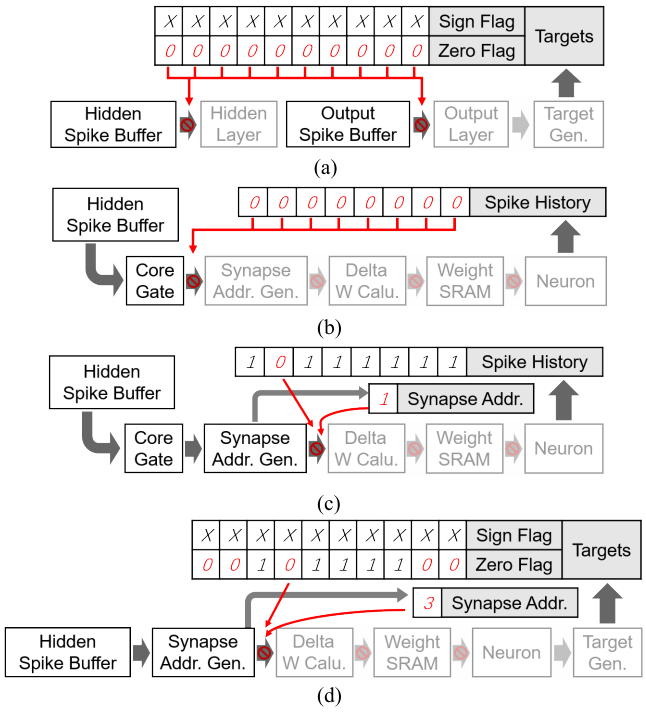


Fig. 10. Detail implementation of hierarchical update skip mechanism. (a) ChipSkip: skipping all updates. (b) CoreSkip: skipping updates in hidden core. (c) HidSkip: skipping updates for one spike on hidden layer. (d) OutSkip: skipping updates for one spike event on output layer.

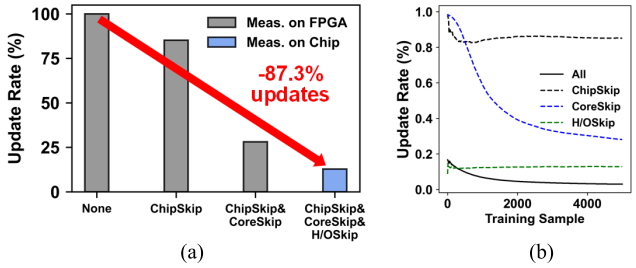


Fig. 11. (a) Update rate under different skip modes. (b) Update rate as a function of training sample.

cores and OutSkip targets the output core. The update rate in different skip modes and the update rate as a function of the training sample when learning on the N-MNIST dataset are shown in Fig. 11(a) and (b), respectively. From this, we find that 14.8% of updates in the ChipSkip step, 57.1% of updates in the CoreSkip step, and 15.3% of updates in the HidSkip and OutSkip steps are reduced; 87.3% of update operations are reduced in total, resulting in a significant reduction of on-chip learning energy overhead.

From the measurements on ANP-I, we find that the update rate in the CoreSkip step reduces rapidly at the first 5000 training examples. It is worth mentioning that the original weight is not used until all the skip steps are finished. Thus, the SRAM access is reduced significantly, and the training energy is saved greatly as a result.

E. Asynchronous Time Step Acceleration

We propose an ATSA to decrease latency between the hidden and output layers at the end of each time step to remove

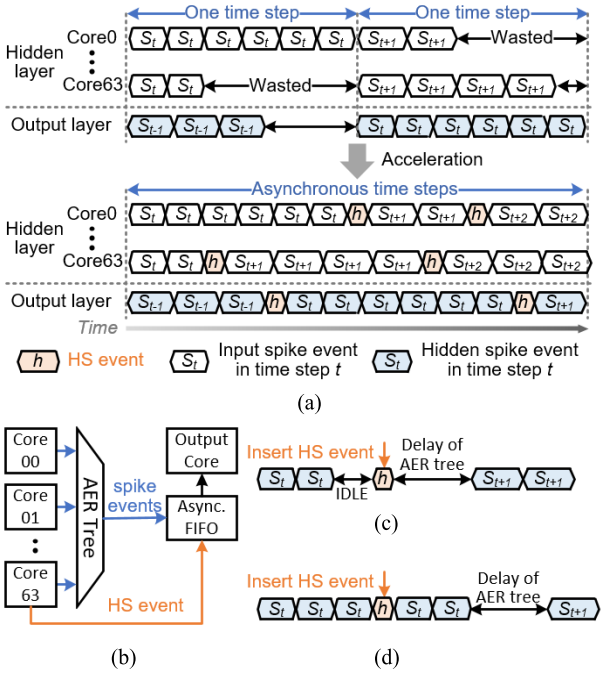
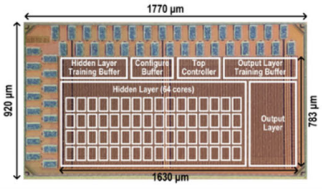


Fig. 12. (a) Asynchronous time step acceleration, (b) event flow for spike events and handshake event, (c) inserting handshake event when spike events are sparse, and (d) inserting handshake event when spike events are dense.

idle time. As shown in Fig. 12(a), at the end of each time step, a handshake event h is sent to each hidden core. Then, the hidden core moves to the next time step without waiting for other cores to synchronize. In order to advance the output layer to the next time step, not all the cores in the hidden layer send a handshake event to the output layer. Instead, only the hidden core 63 is allowed to forward a handshake event h directly to the output layer, as shown in Fig. 12(b). This approach reduces the required handshake latency to advance to the next time step, but it may lead to time step misalignment. In Fig. 12(c), when spike events generated from the hidden layer are sparse, this method can ensure that all the spike events are in the correct time step. However, when spike events generated from the hidden layer are dense, the spike events may fall in an incorrect time step. The measurement results demonstrate that the impact of these incorrect time steps on the inference accuracy can be ignored. This is because the spike events in ANP-I are sparse during almost all the training time. With ATSA, 30%, 58%, and 40% of the processing time is saved on the N-MNIST, DVS-Gesture, and N-TIDIGIT datasets, respectively.

V. Measurement Results

Fig. 13 shows ANP-I's metrics and chip photo, which was fabricated with a 28-nm process. The measurement setup is shown in Fig. 14(a), and the demonstration setup for sEMG- and DVS-based gesture recognition is illustrated in Fig. 14(b) and (c), respectively. The post-P&R simulated power breakdown of the ANP-I processor for both inference and training modes is shown in Fig. 15.



[1] The equivalent frequency for asynchronous is defined as the frequency of fastest asynchronous clock.
[2] The SOP energy includes leakage and idle power.

Technology	28nm CMOS	
Die size	0.92 x 1.77 mm^2	
Core size	0.78 x 1.63 mm^2	
On-chip memory	266.5kB SRAM	
Supply voltage	0.56V~0.9V	
Equivalent frequency[1]	40MHz~210MHz	
Weight precision	8-bit (hidden),10-bit (output)	
Power consumption	2.91mW@40MHz, 0.56V 56.8mW@210MHz, 0.9V	
Energy efficiency[2]	1.49pJ/SOP@40MHz, 0.56V 4.16pJ/SOP@210MHZ, 0.9V	
On-chip learning energy/sample (Accuracy with on-chip learning)	N-MNIST	40.8nJ(96.0%)
	DVS-Gesture	52.1nJ(92.0%)
	N-TIDIGIT	30nJ(92.6%)
	SeNic	87.3nJ(95.7%)

Fig. 13. Chip microphotograph and summary table.

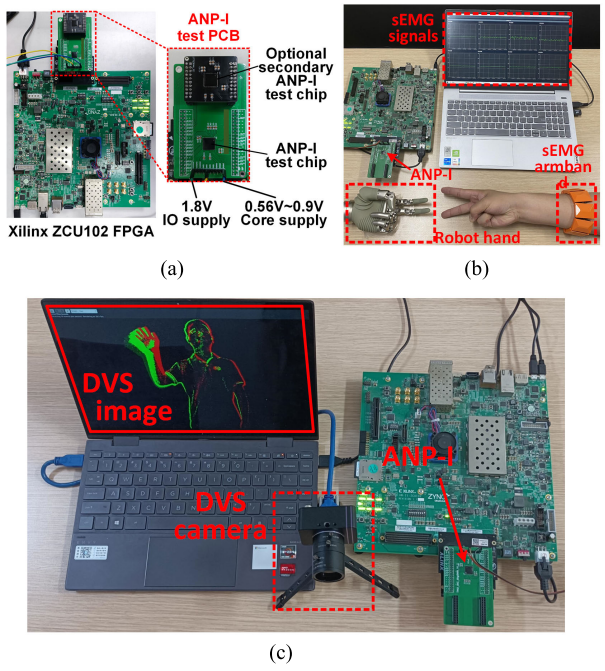


Fig. 14. (a) Measurement setup, (b) sEMG-based gesture recognition demo setup, and (c) DVS-based hand gesture recognition demo setup.

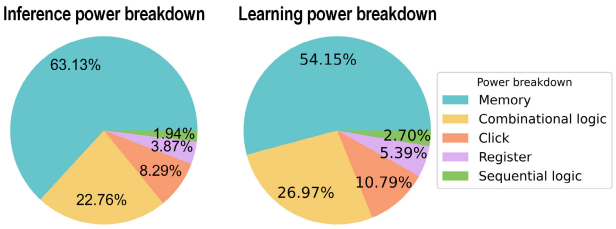


Fig. 15. Simulated power breakdown of ANP-I on inferencing and learning modes.

A. Image Classification

We demonstrate the image classification ability of ANP-I on the N-MNIST dataset, which is the MNIST dataset captured using a DVS, with each sample spanning 300 ms and consisting of 34 × 34 pixels. Each pixel has two types of spikes, representing the increasing and decreasing light intensity. This dataset poses a greater challenge than MNIST due to the added complexity of saccadic motion. All 60 000 training samples and 10 000 testing samples are used for on-chip learning. The samples are downsized to 17 × 17 pixels, and only the first 100 ms out of 300 ms are used for both learning and inferencing. The length of time step is set to 1 ms. Only four time steps are sent to ANP-I for on-chip learning. Therefore, 96.3% of the spikes on N-MNIST are discarded during on-chip learning. Combined with the HUS mechanism (87.3% unnecessary updates are skipped), ANP-I achieves 40.8-nJ on-chip learning energy per sample with 95.3% accuracy after 100 training epochs and 96% accuracy within 200 training epochs. The ANP-I is trained from randomly initialized weights, and the error rate as functions of the training epoch is illustrated in Fig. 16(a).

B. DVS-Based Gesture Recognition

In our work, the IBM DVS Gesture dataset is used for gesture recognition, which comprises ten different gestures executed by 29 subjects under three distinct lighting conditions. These gestures include arm roll, hand clap, and so on. All the data was collected using DVS128 cameras. Each captured gesture lasts approximately 6 s. We adopted the original split of the dataset, where 23 subjects were used for training and six subjects were used for inferencing. We use 1079 samples for on-chip learning and 264 samples for inferencing. The samples were downsized to 14 × 14 pixels and subjected to five temporal filters to transform temporal information into spatial information. The length of one time step is set to 25 ms. Spikes are accumulated inside one time step, and spikes are sent to ANP-I for training and inferencing only when the spike count of each pixel is greater than four. Only ten time steps are sent to ANP-I for on-chip learning. Therefore, 98.1% of the spikes on DVS-Gesture are discarded during on-chip learning. Combined with the HUS mechanism (89.8% of unnecessary updates are skipped), ANP-I achieves 52.1-nJ on-chip learning energy per sample with 90.43% accuracy after 100 training epochs and 92% within 200 training epochs. The ANP-I is trained from scratch with randomly initialized weights, and the error rate as functions of the training epoch is illustrated in Fig. 16(b).

C. Keyword Spotting

N-TIDIGIT [32] is spike-based dataset recorded by playing the audio files of the TIDIGIT dataset to a DAS. The dataset includes single digits and digit sequences of male or female speakers. Digits “0” to “9” in N-TIDIGIT are used in this work, with a total of 2464 training samples and 2486 testing samples. Each sample has 64 channels, and samples with the same label and different IDs have similar firing patterns. We use the first 1024 ms of the training set and test set in the N-TIDIGIT dataset. Eight temporal filters are applied to convert temporal information to spatial information. For the keyword spotting task, the target versus filter word ratio is 1:1. The ANP-I is trained from scratch with randomly initialized weights, and Fig. 16(c) demonstrates that ANP-I achieves 92.6% average accuracy on ten digits with on-chip learning.

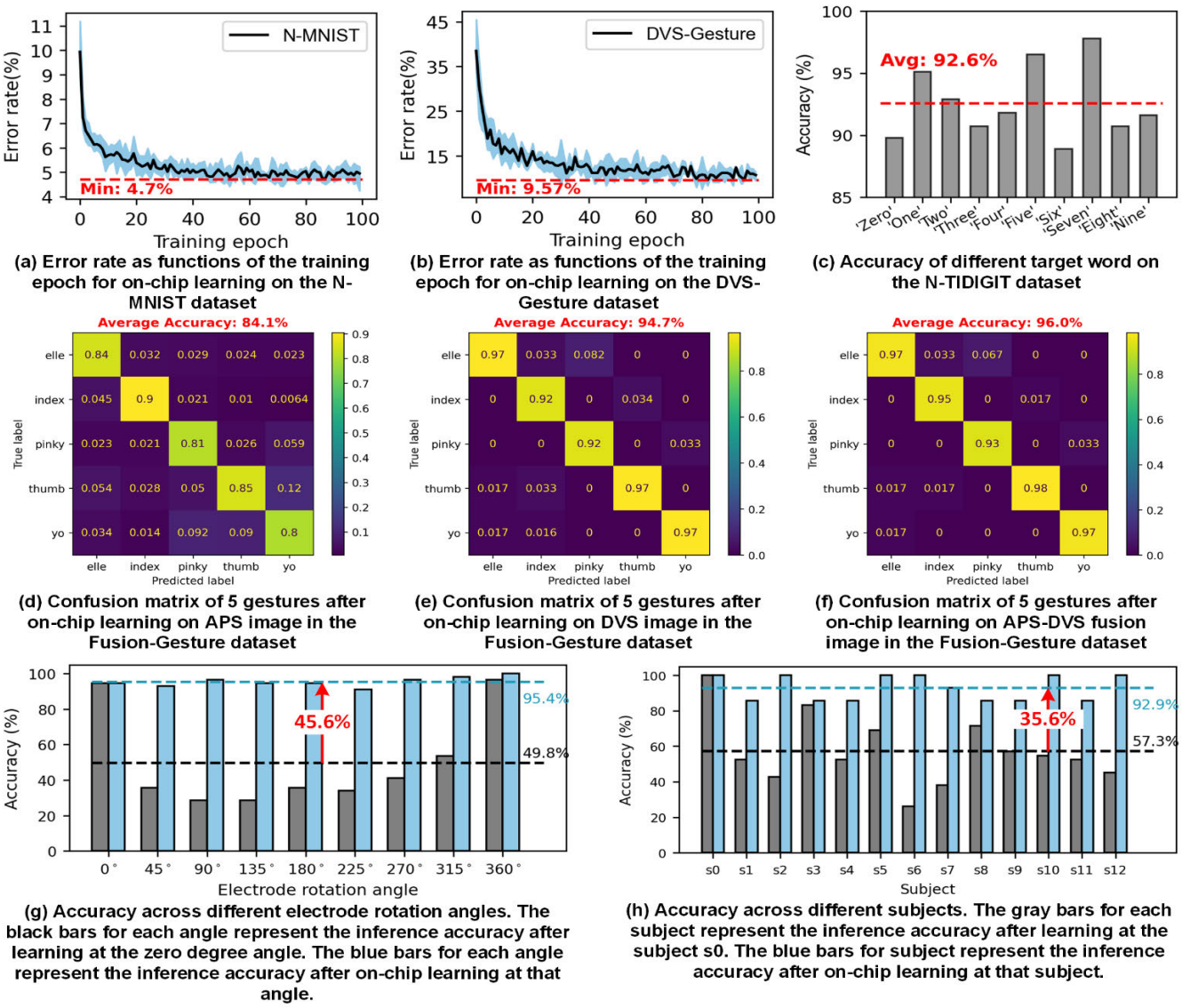


Fig. 16. On-chip learning result on different datasets. Error rate as functions of the training epoch for on-chip learning on (a) N-MNIST dataset and (b) DVS-Gesture dataset. (c) Accuracy of different target words on the N-TIDIGIT dataset. Confusion matrix of five hand gestures after on-chip learning on APS image (d), DVS image (e), and APS-DVS fusion image (f) in the Fusion-Gesture dataset. (g) Accuracy across different electrode rotation angles. (h) Different subjects.

D. Fusion-Gesture Recognition

Fusion-Gesture dataset [34] is a collection of five hand gestures: "pinky," "elle," "yo," "index," and "thumb." The dataset contains recordings from 21 subjects: the subject performs five hand gestures five times in each session and repeats for three sessions. The original dataset is simultaneously recorded with three types of sensors: active pixel sensor (APS), DVS, and EMG. In this task, we use the frame- and event-based data recorded from APS and DVS sensors, respectively. In the experiments, the hand gestures of the first 17 subjects are used for training, and the hand gestures of the remaining four subjects are used for testing. The spikes in the event-based data are accumulated every 100 ms to match the frame-based data. Similar to that in [34], the frame- and event-based data are cropped to the size of 40 × 40. The confusion matrixes of five hand gestures after on-chip learning on APS image, DVS image, and APS-DVS fusion image are depicted in Fig. 16(d)–(f), respectively. ANP-I is able to learn the APS-DVS fusion image and achieves 11.9% or 1.3% accuracy improvement compared to learning only the APS or DVS image.

E. sEMG-Based Gesture Recognition

The sEMG-based gesture recognition system helps the disabled enjoy a better life. However, in real-life circumstances, the performance of an sEMG-based gesture recognition system is affected by electrode shift and subject differences, resulting in a 20%–70% accuracy drop. ANP-I addresses these issues by one-shot on-chip learning with sub-second adaptation time. The SeNic [33] dataset is selected to demonstrate the ability of ANP-I. SeNic is an open-source dataset for sEMG-based gesture recognition in non-ideal conditions. It contains sEMG signals of seven hand gestures gathered from 36 intact-abled subjects (24.6 ± 2.2 years old, 62.8 ± 12.0 kg, 170 ± 8.1 cm,

TABLE I
COMPARISON TABLE OF STATE-OF-THE-ART ON-CHIP LEARNING PROCESSOR

	This work	ISSCC'22 [15]	JSSC'20 [19]	JSSC'19 [29]	VLSI'15 [30]
Technology Implementation Core area Memory Energy efficiency	28nm Digital 1.25mm^2 266.5kB 1.5pJ/SOP [a]	28nm FDSOI Digital 0.45mm^2 138kB 5.3pJ/SOP [b]	65nm Digital 10.1mm^2 353kB 0.29pJ/SOP	10nm Digital 1.72mm^2 896kB 3.8pJ/SOP	65nm Digital 1.8mm^2 37.6kB 5.7pJ/pix
Network type # Neurons # Synapses (width)	SNN (1024)-512-10 258k (8,10-bit)	Spiking RNN (256)-256-16 132k (8-bit)	Binary NN (784)-200-200-10 194k (14-bit)	Multicore SNN 64*64 1M (7-bit)	Spiking LCA 4x64 83k (4,5,14-bit)
On-chip learning -algorithm -multilayer	✓ S-TP ✓	✓ Mod. stoch. e-prop ✓	✓ Mod. SD ✓	✓ STDP ×	✓ SGD ×
Task	Image classification Hand gesture classification Keyword spotting	Navigation Hand gesture classification Keyword spotting	Image classification	Image classification	Image classification
Dataset	N-MNIST [d] IBM DVS Gestures [e] N-TIDIGIT [f] SeNic [g] Fusion-Gesture [i]	Delayed cue integration IBM DVS Gestures Spiking Heidelberg Digits	MNIST	MNIST	MNIST
Accuracy with on-chip learning	N-MNIST: 96.0%@10classes Gest: 92.0%@10classes KWS: 92.6%@1word SeNic: 95.7%@7classes Fusion-Gest: 96.0%@5classes	Nav. 96.4%@2decision Gest: 87.3%@10classes KWS: 90.7%@1word	97.8%@ 10classes	89%@ 10classes	84%-90%@ 10classes
Energy per sample (inference/learn)	NMNIST: 343nJ / 40.8nJ @0.56V, 40MHz [c] Gest: 3.9µJ / 52.1nJ @0.56V, 40MHz [c] KWS: 6.1µJ / 30nJ @0.56V, 40MHz [c] SeNic: 582nJ / 87.3nJ @0.56V, 40MHz [c]	Nav.: 1.4µJ / 3.4µJ @0.5V, 13MHz Gest: 46.1µJ / 112µJ @0.5V, 13MHz KWS: 4.4µJ / 18.5µJ @0.5V, 13MHz	236nJ / 254nJ @0.8V, 20MHz	1.0µJ/ N/A @0.53V, 20MHz	27-162nJ / 94.7µJ @ 0.43V, 40MHz

[a] At 0.56V 40MHz. Static power consumption is included. [b] At 0.5V, 13MHz. [c] Equivalent frequency for asynchronous design. [d] From [31], downscaled to 2x17x17, 10 classed. [e] From [16], downscaled to 16x16 and applied 5 temporal filter, 10 classed. [f] From [32], applied 8 temporal filter, target vs. filter word ration 1:1. [g] From [33], delta-modulator algorithm is used to transform a continuous sEMG signal into spike events. [i] From [34], aps and DVS image are used.

11 females) using Myo armband. Each gesture is held for 4–7 s with a 2-s rest state before the next gesture. The SeNic dataset contains data from five unideal conditions: 1) electrode shift; 2) individual difference; 3) muscle fatigue; 4) inter-day difference; and 5) arm postures. ANP-I is used to solve the two most important variations: electrode shifts and individual difference.

For the electrode shift problem, data gathered from 24 subjects are used for on-chip learning and inferencing. For each subject, ANP-I is trained on all the data at 0° angle. Once the electrodes are shifted, one of the gestures from the shifted angle will be used during on-chip learning. This allows ANP-I to learn the new features and improve inferencing accuracy. Fig. 16(g) shows the accuracy across different electrode rotation angles for 24 subjects. The accuracy is improved by 45.6% using on-chip learning. For the individual difference problem, data from 12 subjects in the 0° are used. ANP-I is trained on subject0 and tested on other subjects. The inferencing accuracy decreases because of the individual difference. However, ANP-I performs on-chip learning from one sample on a new subject. The inferencing accuracy for the new subject is recovered by 35.6%, as shown in Fig. 16(h).

We have developed a demo platform, as shown in Fig. 14(b), in which ANP-I can learn and recognize hand gestures through one-shot on-chip learning on sEMG signals gathered by the

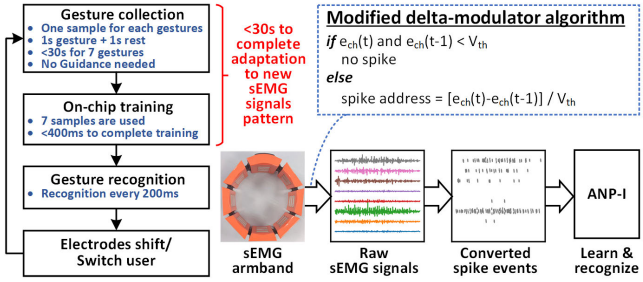


Fig. 17. Pre-process of sEMG signals.

sEMG armband. As shown in Fig. 17, a modified delta-modulator algorithm is proposed to convert raw sEMG signals into spikes, which are sent to the ANP-I for training and inference. The recognition results are used to control the robot arm.

VI. DISCUSSION

In this section, we discuss how on-chip learning is conducted on ANP-I in the case that the labeled data are not available. For most edge applications, it is challenging to obtain labeled data for supervised learning. As shown in Fig. 18, ANP-I uses pseudo-labeling techniques to generate

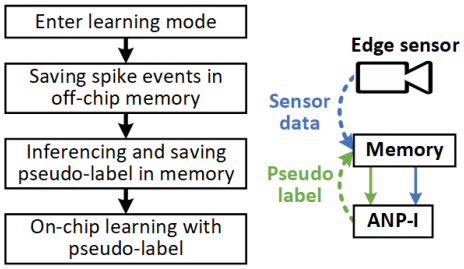


Fig. 18. Semi-supervised learning for on-chip training with unlabeled data.

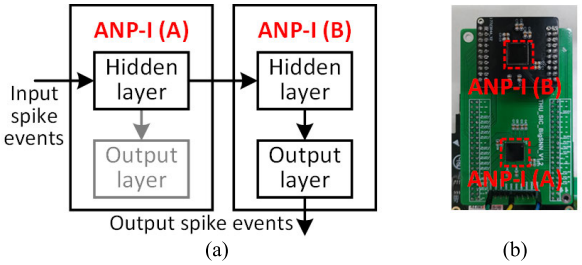


Fig. 19. (a) Schematic and (b) PCB board of two-chip system.

pseudo-labels for on-chip semi-supervised learning. An example application is sEMG-based gesture recognition. sEMG signals suffer from the inter-day difference, which is the variability in sEMG signals from the same subject under the same data acquisition conditions on different days. It often encompasses disturbances from two sources. One is the user's adaptation or motor learning as they become accustomed to the sEMG-based control system, particularly for individuals undergoing rehabilitation training [35]. The other source includes factors such as electrode displacement or shift caused by donning and doffing between different days. With semi-supervised learning, the accuracy of sEMG-based gesture recognition can be improved.

In addition, we further discuss in this section the greatest challenge in enhancing on-chip learning accuracy in ANP-I, which is the limited weight precision. While low precision (4 bits or lower) weights are often sufficient for inference-only neural networks, higher weight precision is required during on-chip learning to achieve acceptable accuracy. However, overfitting and convergence problems are still a major issue in improving on-chip learning accuracy, even with 8-bit weights. Existing solutions include the weight update accumulation mechanism, which computes and accumulates weight update values before applying them to the real weights [19]. This method allows for the accumulation of small weight updates into larger values, which can then be added to the real weights with limited precision. However, additional memory (3.2 kb in [19]) is required, resulting in more area and energy consumption during both inferencing and training. Another solution is the stochastic weight update mechanism, where weights are randomly selected and updated, as opposed to the classical ordered update, where all weights are updated. Previous works [36], [37], [38] demonstrate the effectiveness of stochastic weight updates in enhancing on-chip learning accuracy for neural networks with limited weight precision.

Furthermore, ANP-I allows the direct export of spike events from the hidden layer, enabling the connection of multiple ANP-I chips to create larger neural networks, as depicted in Fig. 19(a). Fig. 19(b) showcases a test board featuring two ANP-I chips, ANP-I (A) and ANP-I (B), which collectively implement a four-layer SNN (including the input layer) with 1024 (input layer)-512-512-10 neurons.

VII. Conclusion

The proposed ANP-I processor integrates 522 asynchronous neurons and 517k synapses for neuromorphic computation. ANP-I supports a wide range of applications, including image classification, DVS and sEMG-based gesture recognition, and keyword spotting. ANP-I achieves above 92% accuracy for all the benchmarks with sub-0.1-μJ learning energy per sample. Table I compares the ANP-I with the state-of-the-art on-chip training processors for edge-AI applications. Our chip achieves the lowest on-chip training energy consumption on all tasks without pre-training. The ANP-I achieves 2150× and 56× on-chip training energy savings on hand gesture recognition and KWS tasks with 4.7% and 1.9% accuracy improvements over [15], respectively. Compared with [19], the ANP-I saves 83.3% on-chip training energy at a cost of 1.8% accuracy on the handwritten digit classification task. Among the on-chip training processors in the comparison table, ANP-I achieves the lowest on-chip learning energy consumption and the highest inference accuracy on different tasks.

References

[1] E. Akleman, "Deep learning," *Computer*, vol. 53, no. 9, p. 17, Sep. 2020.
[2] J. Park, J. Kwon, J. Oh, S. Lee, J.-Y. Kim, and H.-J. Yoo, "A 92-mW real-time traffic sign recognition system with robust illumination adaptation and support vector machine," *IEEE J. Solid-State Circuits*, vol. 47, no. 11, pp. 2711–2723, Nov. 2012.
[3] D. Bankman, L. Yang, B. Moons, M. Verhelst, and B. Murmann, "An always-on 3.8 μ J/86% CIFAR-10 mixed-signal binary CNN processor with all memory on chip in 28-nm CMOS," in *IEEE J. Solid-State Circuits*, vol. 54, no. 1, pp. 158–172, Jan. 2019.
[4] M. Kim and J.-S. Seo, "An energy-efficient deep convolutional neural network accelerator featuring conditional computing and low external memory access," *IEEE J. Solid-State Circuits*, vol. 56, no. 3, pp. 803–813, Mar. 2021.
[5] M. Price, J. Glass, and A. P. Chandrakasan, "A low-power speech recognizer and voice activity detector using deep neural networks," *IEEE J. Solid-State Circuits*, vol. 53, no. 1, pp. 66–75, Jan. 2018.
[6] W. Xiong et al., "The Microsoft 2016 conversational speech recognition system," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Mar. 2017, pp. 5255–5259.
[7] K. Matsubara et al., "A 12-nm autonomous driving processor with 60.4 TOPS, 13.8 TOPS/W CNN executed by task-separated ASIL d control," *IEEE J. Solid-State Circuits*, vol. 57, no. 1, pp. 115–126, Jan. 2022.
[8] F. Akopyan et al., "TrueNorth: Design and tool flow of a 65 mW 1 million neuron programmable neurosynaptic chip," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 34, no. 10, pp. 1537–1557, Oct. 2015.
[9] 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.
[10] S. Moradi, N. Qiao, F. Stefanini, and G. Indiveri, "A scalable multicore architecture with heterogeneous memory structures for dynamic neuromorphic asynchronous processors (DYNAPs)," *IEEE Trans. Biomed. Circuits Syst.*, vol. 12, no. 1, pp. 106–122, Feb. 2018.
[11] M. Davies et al., "Loihi: A neuromorphic manycore processor with on-chip learning," *IEEE Micro*, vol. 38, no. 1, pp. 82–99, Jan. 2018.

[12] N. Qiao et al., "A reconfigurable on-line learning spiking neuromorphic processor comprising 256 neurons and 128K synapses," *Frontiers Neurosci.*, vol. 9, Apr. 2015, Art. no. 141.

[13] C. Frenkel, M. Lefebvre, J.-D. Legat, and D. Bol, "A 0.086-mm^2 12.7-pJ/SOP 64k-synapse 256-neuron online-learning digital spiking neuromorphic processor in 28-nm CMOS," in *IEEE Trans. Biomed. Circuits Syst.*, vol. 13, no. 1, pp. 145–158, Feb. 2019.

[14] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[15] C. Frenkel and G. Indiveri, "ReckOn: A 28nm sub-mm2 task-agnostic spiking recurrent neural network processor enabling on-chip learning over second-long timescales," in *Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC)*, vol. 65, San Francisco, CA, USA, Feb. 2022, pp. 1–3.

[16] A. Amir et al., "A low power, fully event-based gesture recognition system," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7388–7397.

[17] S. K. Gonugondla, M. Kang, and N. Shanbhag, "A 42pJ/decision 3.12TOPS/W robust in-memory machine learning classifier with on-chip training," in *Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC)*, Feb. 2018, pp. 490–492.

[18] A. Amravati, S. B. Nasir, S. Thangadurai, I. Yoon, and A. Raychowdhury, "A 55nm time-domain mixed-signal neuromorphic accelerator with stochastic synapses and embedded reinforcement learning for autonomous micro-robots," in *Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC)*, Feb. 2018, pp. 124–126.

[19] J. Park, J. Lee, and D. Jeon, "A 65-nm neuromorphic image classification processor with energy-efficient training through direct spike-only feedback," *IEEE J. Solid-State Circuits*, vol. 55, no. 1, pp. 108–119, Jan. 2020, doi: 10.1109/JSSC.2019.2942367.

[20] J. Zhang et al., "22.6 ANP-I: A 28nm 1.5pJ/SOP asynchronous spiking neural network processor enabling sub-O.1 μ J/sample on-chip learning for edge-AI applications," in *Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC)*, San Francisco, CA, USA, Feb. 2023, pp. 21–23, doi: 10.1109/ISSCC42615.2023.10067650.

[21] A. Peeters, F. T. Beest, M. de Wit, and W. Mallon, "Click elements: An implementation style for data-driven compilation," in *Proc. IEEE Symp. Asynchronous Circuits Syst.*, May 2010, pp. 3–14.

[22] G. Gimenez, J. Simatic, and L. Fesquet, "From signal transition graphs to timing closure: Application to bundled-data circuits," in *Proc. 25th IEEE Int. Symp. Asynchronous Circuits Syst. (ASYNC)*, Hirosaki, Japan, May 2019, pp. 86–95, doi: 10.1109/ASYNC.2019.00020.

[23] T. P. Lillicrap, D. Cownden, D. B. Tweed, and C. J. Akerman, "Random synaptic feedback weights support error backpropagation for deep learning," *Nature Commun.*, vol. 7, no. 1, p. 13276, Nov. 2016.

[24] A. Nøkland, "Direct feedback alignment provides learning in deep neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 2016, pp. 1037–1045.

[25] D. Zhao, Y. Zeng, T. Zhang, M. Shi, and F. Zhao, "GLSNN: A multi-layer spiking neural network based on global feedback alignment and local STDP plasticity," *Front. Comput. Neurosci.*, Nov. 2020, Art. no. 576841, doi: 10.3389/fncom.2020.576841. PMID: 33281591; PMCID: PMC7689090.

[26] D. H. Lee, S. Zhang, A. Fischer, Y. Bengio, "Difference target propagation," in *Proc. Joint European Conf. Mach. Learn. Knowl. Discovery Databases*, Cham, Switzerland: Springer, Sep. 2015, pp. 498–515, doi: 10.1007/978-3-319-23528-8_31.

[27] Q. Meng, M. Xiao, S. Yan, Y. Wang, Z. Lin, and Z.-Q. Luo, "Towards memory- and time-efficient backpropagation for training spiking neural networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Feb. 2023, pp. 6166–6176.

[28] J. Wang et al., "Short-Dataset-Driven prediction on area electricity consumption with adaptive training window selection," in *Proc. IEEE Int. Conf. Internet Things (iThings) IEEE Green Comput. Commun. (GreenCom) IEEE Cyber, Phys. Social Comput. (CPSCom) IEEE Smart Data (SmartData) IEEE Congr. Cybermatics (Cybermatics)*, Espoo, Finland, Aug. 2022, pp. 648–653.

[29] G. K. Chen, R. Kumar, H. E. Sumbul, P. C. Knag, and R. K. Krishnamurthy, "A 4096-neuron 1M-synapse 3.8-pJ/SOP spiking neural network with on-chip STDP learning and sparse weights in 10-nm FinFET CMOS," *IEEE J. Solid-State Circuits*, vol. 54, no. 4, pp. 992–1002, Apr. 2019.

[30] J. K. Kim, P. Knag, T. Chen, and Z. Zhang, "A 640M pixel/s 3.65 mW sparse event-driven neuromorphic object recognition processor with on-chip learning," in *Proc. Symp. VLSI Circuits (VLSI Circuits)*, Kyoto, Japan, Jun. 2015, pp. C50–C51.

[31] G. Orchard, A. Jayawant, G. K. Cohen, and N. Thakor, "Converting static image datasets to spiking neuromorphic datasets using saccades," *Frontiers Neurosci.*, vol. 9, p. 437, Nov. 2015.

[32] J. Anumula, D. Neil, T. Delbruck, and S.-C. Liu, "Feature representations for neuromorphic audio spike streams," *Frontiers Neurosci.*, vol. 12, p. 23, Feb. 2018.

[33] B. Zhu, D. Zhang, Y. Chu, Y. Gu, and X. Zhao, "SeNic: An open source dataset for sEMG-based gesture recognition in non-ideal conditions," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 30, pp. 1252–1260, 2022.

[34] E. Ceolini et al., "Hand-gesture recognition based on EMG and event-based camera sensor fusion: A benchmark in neuromorphic computing," *Frontiers Neurosci.*, vol. 14, p. 637, Aug. 2020.

[35] M. Ison, I. Vujaklija, B. Whitsell, D. Farina, and P. Artemiadis, "High-density electromyography and motor skill learning for robust long-term control of a 7-DoF robot arm," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 24, no. 4, pp. 424–433, Apr. 2016.

[36] J. Koščak, R. Jakša, and P. Sinčák, "Stochastic weight update in the backpropagation algorithm on feed-forward neural networks," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Barcelona, Spain, Jul. 2010, pp. 1–4, doi: 10.1109/IJCNN.2010.5596870.

[37] C. Frenkel, J.-D. Legat, and D. Bol, "MorphIC: A 65-nm 738k-synapse/mm2 quad-core binary-weight digital neuromorphic processor with stochastic spike-driven online learning," in *IEEE Trans. Biomed. Circuits Syst.*, vol. 13, no. 5, pp. 999–1010, Oct. 2019, doi: 10.1109/TBCAS.2019.2928793.

[38] C. Frenkel, J.-D. Legat, and D. Bol, "A 28-nm convolutional neuromorphic processor enabling online learning with spike-based retinas," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Oct. 2020, pp. 1–5, doi: 10.1109/ISCAS45731.2020.9180440.

Jilin Zhang (Member, IEEE) received the bachelor's degree in microelectronics science and engineering from Lanzhou University, Lanzhou, China, in 2019. He is currently pursuing the Ph.D. degree with the School of Integrated Circuit, Tsinghua University, Beijing, China.

His research interests focus on asynchronous circuits and neuromorphic computing, including asynchronous circuit design methodology, neuromorphic computing architecture and algorithm, and asynchronous chip and systems.

Dexuan Huo (Student Member, IEEE) received the B.S. degree in microelectronics science and engineering from Jilin University, Changchun, China, in 2017, and the M.S. degree in integrated circuit engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2020. He is currently pursuing the Ph.D. degree in electronic science and technology with Tsinghua University, Beijing, China.

His Ph.D. project is focused on the design of digital asynchronous spiking neural network accelerator for gas recognition. His research interests include spiking neural networks, machine olfaction, and digital asynchronous circuit design.

Jian Zhang (Graduate Student Member, IEEE) received the bachelor's degree from the School of Physics and Technology, Wuhan University, Wuhan, China, in 2022. He is currently pursuing the master's degree with the School of Integrated Circuit, Tsinghua University, Beijing, China.

His research interests primarily concentrate on system-level simulation and asynchronous circuits, including system-level simulator design for asynchronous neuromorphic hardware and hardware architecture search methodologies.

Chunqi Qian (Student Member, IEEE) received the bachelor's degree in microelectronics science and engineering from the Beijing University of Technology, Beijing, China, in 2021. She is currently pursuing the master's degree with the School of Integrated Circuit, Tsinghua University, Beijing.

Her areas of research include asynchronous neuromorphic circuit design and asynchronous circuit design methodology.

Qi Liu (Student Member, IEEE) received the bachelor's degree in chemical science and engineering from Tsinghua University, Beijing, China, in 2023, where he is currently pursuing the master's degree with the School of Integrated Circuit.

His research interests focus on spiking neural networks and asynchronous circuits.

Liyang Pan (Senior Member, IEEE) received the B.S. degree in microelectronics from the Hefei University of Technology, Hefei, China, in 1996, the M.S. degree in microelectronics from Zhejiang University, Zhejiang, China, in 1999, and the Ph.D. degree in microelectronics from Tsinghua University, Beijing, China, in 2003.

Since 2003, he has been with the Institute of Microelectronics, Tsinghua University. His research interests include memory devices, circuits, and systems.

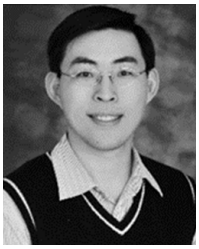
Zhihua Wang (Fellow, IEEE) received the B.S., M.S., and Ph.D. degrees in electronic engineering from Tsinghua University, Beijing, China, in 1983, 1985, and 1990, respectively.

From 1992 to 1993, he was a Visiting Scholar with Carnegie Mellon University, Pittsburgh, PA, USA. From 1993 to 1994, he was a Visiting Scholar with KU Leuven, Leuven, Belgium. Since 1997, he has been a Full Professor with Tsinghua University, where he has been the Deputy Director of the Institute of Microelectronics since 2000. Since 2011, he has been the Director of the Laboratory of Integrated Circuits and Intelligence Systems, Research Institute of Tsinghua University in Shenzhen (RITS-ICIS), Shenzhen, China. From September 2014 to March 2015, he was a Visiting Professor with The Hong Kong University of Science and Technology (HKUST), Hong Kong. He has coauthored 13 books/chapters, over 225 (569) papers in international journals (conferences), and over 251 (29) articles in Chinese journals (conferences). He holds 130 Chinese and ten U.S. patents. His research mainly focuses on CMOS radio frequency integrated circuits (RFIC) and biomedical applications, involving radio frequency identification (RFID), phase-locked loop (PLL), low-power wireless transceivers, and smart clinic equipment combined with leading-edge RFIC and digital image processing techniques.

Dr. Wang was an AdCom Member of the IEEE SSCS from 2016 to 2019. He was a Technology Program Committee Member of the IEEE ISSCC from 2005 to 2011. Since 2005, he has been a Steering Committee Member of the IEEE A-SSCC. He has served as the Chairperson for the IEEE SSCS Beijing Chapter from 1999 to 2009. He was the Technical Program Chair of A-SSCC 2013. He was a Guest Editor of IEEE Journal of Solid-State Circuits (JSSC) Special Issue in December 2006, December 2009, and November 2014. From 2019 to 2020, he was the Associate Editor-in-Chief of IEEE Open Journal of Circuits and Systems. He was an Associate Editor of IEEE Transactions on Circuits and Systems—I: Regular Papers from 2016 to 2019, IEEE Transactions on Circuits and Systems—II: Express Briefs from 2010 to 2013, and IEEE Transactions on Biomedical Circuits and Systems (BioCAS) from 2008 to 2015, and held other administrative/expert committee positions in China's national science and technology projects. From 2018 to 2019, he was an IEEE Solid-State Circuits Society (SSCS) Distinguished Lecturer. Since 2020, he has been an IEEE CASS Distinguished Lecturer.

Ning Qiao (Member, IEEE) received the bachelor's degree in microelectronics and solid-state electronics from Xi'an Jiaotong University, Xi'an, China, in 2006, and the Ph.D. degree in microelectronics from the Institute of Semiconductors, Chinese Academy of Sciences, Beijing, China, in 2012, with a focus on ultra-low-power low-noise mixed-signal circuits in SOI process.

He joined the Institute of Neuroinformatics, University of Zurich and ETH Zürich, Zürich, Switzerland, as a Post-Doctoral Researcher, in 2012, where he is currently involved in developing mixed-signal multicore neuromorphic VLSI circuits and systems. His research interests include ultra-low-power subthreshold mixed-signal neuromorphic VLSI circuits and systems, parallel neuromorphic computing architectures, and fully asynchronous event-driven computing and communication circuits and systems.



Kea-Tiong Tang (Senior Member, IEEE) received the B.S. degree in electrical engineering from the National Taiwan University, Taipei, Taiwan, in 1996, and the M.S. and Ph.D. degrees in electrical engineering from the California Institute of Technology, Pasadena, CA, USA, in 1998 and 2001, respectively.

From 2001 to 2006, he was a Senior Electrical Engineer with Second Sight Medical Products, Inc., Sylmar, CA, USA. He designed mixed-signal ASIC for retina prosthetic devices. In 2006, he joined the Faculty of Electrical Engineering, National Tsing Hua University, Hsinchu, Taiwan, where he is currently a Professor. His research interests include neuromorphic SoC design, bio/chemical sensing systems, analog and mixed-signal integrated circuits (IC) design, and biomedical SoC design.

Hong Chen (Senior Member, IEEE) received the Ph.D. degree from the Department of Electronic Engineering, Tsinghua University, Beijing, China, in 2005.

From 2005 to 2007, she worked at the Institute of Microelectronics of Tsinghua University (IMETU) as a Post-Doctoral Fellow. Since 2007, she has been working with the School of Integrated Circuits, Tsinghua University, where he is currently a Full Professor. From February 2006 to June 2006, she worked at the Medical Center, Nebraska University, Lincoln, NE, USA. She has been a Visiting Scholar with Gatech Tech from March 2016 to April 2017. She has published more than 120 journal, Atlanta, GA, USA, and conference papers. She holds one U.S. patent and 32 Chinese patents. Her research interests include bio-inspired on-chip learning asynchronous neuromorphic circuits, biomedical prosthetic circuits and systems, and so on.

Dr. Chen has served as an Associate Editor for IEEE Transactions on Biomedical Circuits and Systems, a Guest Editor for *Tsinghua Science and Technology*, a Technical Program Committee Member for The IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC), IEEE International Symposium on Circuits and Systems (ISCAS), and IEEE Biomedical Circuits and Systems Conference (BioCAS), and the General Chair for IEEE ASYNC 2023.