A 18.7 TOPS/W Mixed-Signal Spiking Neural Network Processor with 8-bit Synaptic Weight On-chip Learning that Operates in the Continuous-Time Domain

SEIJI UENOHARA¹,², (Member) and KAZUYUKI AIHARA³ (Nonmember)

¹Institute of Industrial Science, The University of Tokyo 4-6-1 Komaba, Meguro-ku, Tokyo 153-8505, Japan (e-mail: s.uenohara@gmail.com)
²Graduate School of Life Science and Systems Engineering, Kyushu Institute of Technology, 2-4 Hibikino, Wakamatsu-ku, Kitakyushu-shi, Fukuoka, 808-0196, Japan (e-mail: uenohara@brain.kyutech.ac.jp)
³International Research Center for Neurointelligence, University of Tokyo 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8654, Japan (e-mail: kaihara@g.ecc.u-tokyo.ac.jp)

Corresponding author: S. Uenohara (e-mail: s.uenohara@gmail.com)

This work was supported by the VLSI Design and Education Center (VDEC) at the University of Tokyo in collaboration with Cadence Design Systems, Inc., AMED under Grant Number JP20dm0307009, JST Moonshot R&D Grant Number JPMJMS2021, JSPS KAKENHI Grant Number JP20H05921, and the BrainMorphic AI (BMAI) project at the University of Tokyo’s Institute of Industrial Science in collaboration with NEC Corporation.

ABSTRACT We present a mixed-signal spiking neural networks processor with 8-bit synaptic weight on-chip learning in 40 nm CMOS that consists of a 10k mixed-signal synapse circuit and 100 analog leaky integrate-and-fire (LIF) neuron circuits. The processor has no clock signal except in peripheral circuits for I/O, and neuron and synapse circuits can operate asynchronously in the continuous-time domain, just like biological neurons. We demonstrate the energy efficiency of 6.24–18.7 TOPS/W in a multitarget spike learning task.

INDEX TERMS ReSuMe, on-chip learning, compute-in-memory, CIM, spiking neural networks, SNN

I. INTRODUCTION Transistor shrinking is approaching its physical limits, so three-dimensional (3D) integration technologies are being studied for next-generation semiconductor devices.

With 3D integration technologies, it is expected that new applications can be realized by stacking dies fabricated using different technologies, such as complementary metal-oxide-semiconductors (CMOS), micro-electro-mechanical systems (MEMS), and dynamic random-access memory (DRAM). Such technologies will also reduce the delay and power consumption for communication with other chips and system areas. However, a concern is that thermal problems will be more serious than those encountered with single-die integrated circuits. Excessive heat should be considered because heat is a more serious problem in 3D integrations than in thin dies and thus limits the number of stacked layers per volume [1]. To realize highly stacked systems, it is important to develop a highly efficient arithmetic scheme that can void thermal problems. In hardware research for machine learning (ML), mixed-signal hardware based on compute-in-memory (CIM) architectures has been proposed to realize high-efficiency application-specific integrated circuits (ASIC) [2]–[11].

CIM architectures are used to reduce power consumption for multiply-accumulate (MAC) operations. In CIM, MAC operations are carried out using analog current and voltage, and processors employing CIM architectures have demonstrated high energy efficiency [2]–[8]. Moreover, CIM processors based on resistive random-access memory (ReRAM) have been proposed to achieve higher energy efficiency [9]–[11]. The CIM approach has been shown to work effectively in the ultra-deep submicron regime [8]. CIM architectures can potentially allow AI processors to directly process sensor data without analog-to-digital conversion, thereby realizing extremely high-efficiency 3D integration in intelligent processors for data output from MEMS sensors. However, CIM circuits are more sensitive to fabrication mismatches than conventional digital circuits. On-chip learning can potentially
reduce the influence of mismatches [12], but hardware based on CIM processors is mostly inference hardware that has synaptic weights of 1–4 bits to reduce the footprint and energy consumption of the digital-to-analog converter (DAC), and implementation of hardware using the CIM approach with on-chip learning of synaptic weights exceeding 4 bits is a challenge.

Besides thermal problems, 3D integration has a global clock distribution problem. Because it is difficult to synchronize global operations among several chips using a common clock signal, it is important to select a configuration that does not require synchronization between chips in a 3D stacked circuit. Spiking neural networks (SNNs) have been proposed as an asynchronous operation model. SNN hardware has already been implemented as digital circuits with a clock signal [13]–[17] and analog or mixed-signal circuits without any clock signal [18]–[23]. Analog SNN hardware operates in the continuous-time domain without a clock signal, eliminating the power consumption that would be necessary for clock signal distribution. Furthermore, system scaling by chip stacking is easy.

With the aim of realizing a component for scalable ML systems using 3D stacking technology, we propose an SNN that satisfies three important criteria: high-efficiency computing with a CIM architecture, asynchronous operation without clock signals, and on-chip learning with a synaptic weight exceeding 4 bits. We designed a prototype using a TSMC 40 nm CMOS that operates in the continuous-time domain, the same as biological neurons. We employed the remote supervised method (ReSuMe) [24] as a supervised algorithm.

The remainder of this paper is organized as follows: In Section 2, we describe the learning algorithm implemented in our circuit. Section 3 describes the proposed circuit and implementation of the synapse and neuron circuits. Section 4 presents experimental results for the proposed circuit, and Section 5 concludes.

II. LEARNING ALGORITHM

The remote supervised method (ReSuMe) [24] shown in Fig. 1 is a supervised-learning algorithm for SNNs, in which algorithm weight updates are based on the i-th presynaptic spike train \( S_{pre,i} (t) \), the postsynaptic spike train \( S_{post,j} (t) \) output from the j-th neuron, and the target spike train \( S_{tgt,j} (t) \) for the j-th neuron. This algorithm can learn multi-target models, and can also be applied to various spiking neuron models, including leaky integrate-and-fire (LIF) [25], and the Hodgkin-Huxley [26] and Izhikevich [27] neuron models. This algorithm is expressed as

\[
\frac{dw_{ij}(t)}{dt} = [S_{tgt,j}(t) - S_{post,j}][a_d + \int_{0}^{\infty} f_{ij}(s_{ij})S_{pre,i}(t-s_{ij})dt],
\]

(1)

where \( t \) is the continuous time, \( a_d \) is a non-Hebbian term, and \( s_{ij} \) is the delay between the \( S_{pre,i}(t) \) and \( S_{tgt,j}(t) \) firings \((s_{ij} = t_{pre,i} - t_{tgt,j})\). The exponential kernel \( f_{ij}(s_{ij}) \) is

\[
f_{ij}(s) = \begin{cases} A_R \exp(-s_{ij}/\tau_R) & \text{if } s_{ij} \geq 0, \\ 0 & \text{if } s_{ij} < 0, \end{cases}
\]

where \( A_R \) is the amplitude of long-term potentiation, and \( \tau_R \) is the time constant of exponential decay. In our circuit, we set \( A_R = A^+ = A^- \).

III. PROPOSED CIRCUIT

A. CHIP ARCHITECTURE

The proposed circuit is implemented based on the compute-in-memory architecture shown in Fig. 2 to achieve high-efficiency MAC operations. This architecture consists of a mixed-signal synapse circuit and an analog leaky integrate-and-fire (LIF) neuron circuit. The synapse and neuron circuits have no clock signal and operate asynchronously in the continuous-time domain, the same as actual neural systems. The neuron-synapse array macro performs a MAC operation when a pre-spike arrives. Thus, processor power consumption depends on the frequency of the pre-spike input and the values of the voltage sources. SNN hardware is stored by localized flip-flops in the synapse circuit, and each synapse outputs an analog current weighted by the synaptic weight. The macro in the fabricated chip consists of the column circuit shown in Fig. 2.

Figure 3 shows the architecture of the SNN processor, which consists of a 100 × 100 mixed-signal synapse array and a 100 × 1 analog LIF circuit array. Input spikes (pre-spikes) and output spikes (post-spikes) are parallel input and output using a decoder and an encoder with 7 bits, respectively. Each decoder has 100 output nodes for pre-spike inputs. Target spikes for supervised learning using ReSuMe are input by a serial-to-parallel convertor (S2P). By restricting the operating neuron circuits, the processor can select between 100-input mode and 1,000-input mode. The mode is changed by a 1-bit selection signal SL. Subsection III-B describes the method for restricting neuron circuits.
B. NEURON CIRCUIT

Figure 4 shows details of the neuron circuit, which consists of a pulse generator (PG), a synapse circuit, and a neuron circuit. The PG realizes threshold processing for generating a spike pulse using an inverter. Bias voltages $V_{lb}$ and $V_{ln}$ adjust the threshold voltages of the inverter for threshold processing by restricting the current for charging/discharging gate capacitance in the next stage. Transistors $M_{ip}$ and $M_{in}$ supply bias current to reduce time variation of $V_{x,j}(t)$ induced by leak current from the synapse array. Membrane capacitor $C_{x,j}$ consists of a MOM capacitor and a parasitic capacitor for the synapse array. The design value for $C_{x,j}$ is 32.7 fF. Note that bias voltages $V_{ip}, V_{in}, V_{lk},$ and $V_{ln}, V_{lp}$ and reset voltage $V_{xrst}$ are common to all neuron circuits.

Three registers in the neuron circuit change the number of synapse circuits per neuron circuit. The first register sets the neuron circuit to active or inactive. Membrane potential $V_{x,j}(t)$ connects the next membrane potential $V_{x,j+1}(t)$ and the output node of synapse circuits via switches $SW_2$ and $SW_1$. The ON/OFF state of $SW_1$ and $SW_2$ are controlled by the second and third registers, respectively. For example, in the case of 100 synapses per neuron, the values of the first, second, and third registers are 1, 1, and 0, respectively. In the case of increasing the number of synapse circuits per neuron, the registers of an inactive neuron are set to 0, 0, and 1, and a metal line-connected membrane capacitor $C_{x,j}$ is shared with the next neuron.

C. SYNAPSE CIRCUIT

Figure 5 shows a block diagram of the synapse circuit. The synapse consists of a delay-line array and update signal generator (DLA&USG), 8-toggle flip-flops (T-FFs), and a DAC. The DLA&USG generates update signals for the synaptic weights held in flip-flops. The DAC outputs analog current according to the synaptic weight when pre-spike $S_{pre,j}$ is input. Voltages $V_{bDAC}, V_{bUSGA}, V_{lk}, V_{lpp}, V_{rsn}$, and $V_{rsp}$ are analog bias voltages, the roles of which are described below.

1) Discretization of Kernel Functions for Circuit Implementation

Because synaptic weights are held in flip-flops in our circuit, if kernel functions are expressed as analog continuous waveforms, an analog-to-digital converter (ADC) is required because $f_{ij}(s_{ij})$ has an analog value. To avoid using an ADC, we discretize the kernel function into five digital time windows consisting of five digital pulses $S_{D1}(t)–S_{D5}(t)$, as shown in Fig. 6(a). By this modification, $f_{ij}(s_{ij})$ and $\tau_R$ are respectively expressed as $S_{D1}(s_{ij})$ and $\sum_{q=1}^{1} T_{wq}$. Pulse widths $T_{w1}$–$T_{w5}$ are adjusted by $V_{b1}$–$V_{b5}$, respectively. The non-Hebbian term $a_d$ was not implemented in our circuit.

Synaptic weights are varied when a spike pulse of $S_{post,j}(t)$ or $S_{tg,j}(t)$ is included in any one of $S_{D1}(t)–S_{D5}(t)$. The value for $dw/dt$ and $f_{ij}(s_{ij})$ depends on the time-window index $q$. In the case of a positive update, one is added to the $q$th flip-flop when $S_{tg,j}(t)$ is included in the $q$th time window. In the case of a negative update, one is added to all flip-flops except the $q$th, then one is added to LSB when $S_{post,j}(t)$ is included in the $q$th time window. Note that negative updates are achieved as complements of the number 2. Figure 6(b) shows examples of flip-flop updating during ReSuMe execution.

2) Details of DLA&USG

Figures 7(a)–(c) show details of the DLA&USG, the delay-line (DL) circuit symbol, and the DL circuit, respectively. The DLA&USG consists of five DL circuits and one USG. The digital time window $S_{D,q}(t)$, having pulse width $T_{wq}$, is output from each DL. The DL includes a transistor biased on $V_{bias}$ (see Fig. 7(c)). This transistor suppresses the rising slope of $V_A$ generated at the trailing edge of $S_{in}$ (see Fig. 7(d)). This suppression is generated by a current limit for charging the parasitic capacitor at the drain node of the biased transistor, with $V_{bias}$ adjusting the slope. Varying the slope changes the time needed to reach threshold voltage $V_{invth}$ of an inverter. As a result, $T_{wq}$ is varied.

3) Flip-flops in Detail

Figure 8 shows the details of the T-FF. The T-FF is inverted at the trailing edge of an update signal. To achieve asynchronous addition, subtraction, and carry, we employed a circuit comprising a T-FF and an XOR. By connecting the XOR to the output stage of a T-FF, the T-FF is inverted when an adjacent lower bit switches from High to Low (carry) or when $S_{UD,n}$ switches from High to Low, where $n$ is the index of the T-FF. Calculation results for synaptic weight can be unstable in subtraction if $S_{UD,n}$ are input at almost the same
time. To avoid this problem, we shifted the timing of $S_{UD,n}$ so that signals arrive in order from the MSB to the LSB.

4) DAC

Figures 9(a)–(c) show a block diagram of the DAC, details of the digital block (DB), and details of the analog block (AB), respectively. The DAC operates while input node $S_{in}$ is High.
The DB consists of seven AND gates, seven OR gates, and one NOT gate, and generates switching signals for current sources in the AB. \(S_{1p} - S_{7p}\) and \(S_{1n} - S_{7n}\) are connected to PMOS/NMOS switched-current sources (SCSs).

The AB consists of a current mirror block (shaded area) and an NMOS/PMOS transistor acting as the SCS. The current mirror block generates gate voltages for the SCSs. The generated voltages depend on \(V_{DAC}\), which is set to the source-drain current values of \(M_{7p}\) and \(M_{7pb}\). We can obtain a gate voltage for which the source-drain current value of \(M_{6nb}\) will be half that of \(M_{7nb}\) by setting aspect ratio \(W/L\) of \(M_{7nb}\) to twice the size of \(M_{6nb}\). The gate voltage corresponding to the current value of the lower bits can be generated by the same procedure. Tables 1 and 2 show \(W/L\) ratios of transistors comprising the AB when the \(W/L\) of \(M_{7pb}\) and \(M_{7nb}\) are defined as unity, respectively. Current ratios in these tables are designed values when the source-drain current of \(M_{7pb}\) is defined as unity.

D. SUPERVISED LEARNING OPERATION

Figure 10 shows waveforms of each nodes during ReSuMe learning. The process of supervised learning in the designed circuit is summarized as follows:

1) Pre-spike \(S_{pre,i}(t)\) is input and the membrane potential \(V_{x,j}(t)\) changes.

2) Discretized kernel function \(f^{D}_{ij}(t)\) starts at the trailing edge of \(S_{pre,i}(t)\).

3) Synaptic weight \(W_{i,j}(t)\) is changed if post-spike \(S_{post,j}(t)\) or target spike \(S_{tgt,j}(t)\) is included in the kernel.

The sign of the synaptic weight is positive if it is greater...
TABLE 1. PMOS

| Name    | W/L | Current ratio |
|---------|-----|---------------|
| M7pb    | 1   | 1/2^1        |
| M6pb    | 2   | 1/2^2        |
| M5pb    | 1   | 1/2^3        |
| M4pb    | 2   | 1/2^4        |
| M3pb    | 1   | 1/2^5        |
| M2pb    | 2   | 1/2^6        |
| M1pb    | 1   | 0.5          |

TABLE 2. NMOS

| Name    | W/L | Current ratio |
|---------|-----|---------------|
| M7nb    | 1   | 1/2^1        |
| M6nb    | 0.5 | 1/2^2        |
| M5nb    | 1   | 1/2^3        |
| M4nb    | 0.5 | 1/2^4        |
| M3nb    | 1   | 1/2^5        |
| M2nb    | 0.5 | 1/2^6        |
| M1nb    | 1   | 0.5          |

than or equal to 127 and negative otherwise.

E. FABRICATED CHIP

We designed and fabricated the proposed circuit using TSMC 40-nm (1-poly, 8-metal) CMOS technology. Figures 11(a) and (b) show a whole-chip microphotograph and the single-synapse circuit layout, respectively.

IV. RESULTS OF CIRCUIT EXPERIMENTS

The prototype chip has nodes for observation and experiments. We can measure the membrane potential \( V_{x,100}(t) \) via a source follower buffer, and post-spike pulses \( S_{\text{post},10}(t) \), \( S_{\text{post},20} \), ..., \( S_{\text{post},100} \) without going through the encoder, but those are output via onboard level-shift circuits. Unless otherwise indicated, bias voltages and source voltages were as follows: synapse array voltage source \( V_{\text{SY}S} = 0.56 \) V, neuron array voltage source \( V_{\text{NU}} = 0.8 \) V, I/O voltage source \( V_{\text{IO}} = 1 \) V, \( V_{b1} = V_{b2} = V_{b3} = 0.45 \) V, \( V_{b4} = 0.4 \) V, \( V_{b5} = 0.4 \) V, \( V_{b6} = 0.1 \) V, \( V_{ip} = 0.6 \) V, \( V_{in} = 0.33 \) V, \( V_{ik} = 0 \) V, and \( V_{xrst} = 0 \) V.

A. DA CONVERSION CHARACTERISTICS

Figure 12 shows measurement results for synaptic weight versus change in membrane potential when the synapse receiving simultaneous input was changed from \#1 to \#4. Note that this membrane potential waveform is that of the 100th neuron obtained through a source follower.

The characteristics shown Fig. 12 are equivalent to DA conversion characteristics and thus should ideally be linear. However, as the figure shows, the characteristics were sigmoidal, nonlinear, and very noisy. Table 3 shows the slopes and intercepts obtained by linear fitting. The slope for \#2 is about twice as large as that for \#1, but slopes for \#3 and \#4 are not three and four times the slope for \#1. This is attributed to the transistor characteristic that the current value decreases as the drain-source voltage decreases.

B. MULTI-TARGET SPIKE LEARNING TASK

To demonstrate functionality for high-efficiency on-chip learning, we conducted a multi-target spike learning task. In

FIGURE 9. Block diagram for (a) DAC, (b) digital block in detail, and (c) analog block in detail.
100-input mode, one learning period was set to 80 µs (pre-spike train input: 60 µs; wait: 20 µs), and all synaptic weights were set to 255 (= (11111111)_2). Pre-spikes were input in order from \(S_{\text{pre1}}\) to \(S_{\text{pre100}}\) every 0.6 µs. Triple spikes were used as target spikes. Firing times were set to 1,960 ns, 2,930 ns, and 4,420 ns. Target spikes were set to the same spike train for all neurons.

Figure 13 shows the voltage waveforms of \(S_{\text{tgt100}}(t)\), \(S_{\text{post100}}(t)\), and \(V_{M}(t)\) during the task. The neuron fires at high frequency when the number of iterations \(p\) is unity, as shown in panels (a) and (b). We can see that the number of spikes decreases as learning progresses (see panel (b)), but there is nearly no decrease in the case of no learning (see panel (a)). The firing times of \(S_{\text{post100}}\) almost converged to the firing times of \(S_{\text{tgt100}}\) after thirty learning cycles.

**FIGURE 10.** Waveforms of pre-spikes, target spikes, post-spikes, discretized kernel functions, synaptic weight changes, and membrane potential during ReSuMe learning.

**FIGURE 11.** VLSI layout results: (a) chip microphotograph and (b) single-synapse circuit layout.

**FIGURE 12.** DA conversion characteristics obtained for the fabricated chip.

**TABLE 3.** Slopes and intercepts obtained by linear fitting

|   | Slope   | Intercept |
|---|---------|-----------|
| #1 | \(1.1 \times 10^{-4}\) | 0.39      |
| #2 | \(1.9 \times 10^{-4}\) | 0.38      |
| #3 | \(2.2 \times 10^{-4}\) | 0.38      |
| #4 | \(2.1 \times 10^{-4}\) | 0.38      |
C. COMPUTATIONAL EFFICIENCY

Figure 14 shows power consumption in the standby, learning, and inference states. Power consumption in the neuron array is low in the standby state, because the neuron circuits do not fire. The power consumptions in the neuron array and I/O during learning and inference were nearly the same, but those in the synapse array during learning were not the same. This difference was likely due to DLA&USG and T-FF updating.

V. DISCUSSION

CMOS ANN hardware for inference has already achieved high energy efficiency of more than 600 TOPS/W with MAC operations based on a CIM architecture [29] by limiting the bit-width of the synaptic weights as shown in Table 4. The energy efficiency is higher than that of ANN hardware using ReRAM [10], [28]. Thus, if we focus only on the efficiency of the MAC operation, it would seem to be no advantage of adopting non-CMOS memory such as ReRAM and phase change memory (PCM) for synapses in exchange for the risk of higher manufacturing cost and lower yield.

However, the physical characteristics of non-CMOS memory that allow analog information to be input and output as an analog signal can be a basic element of information processing without an ADC. Such an element is suitable for realizing information processing devices in which a sensor is directly connected to the information processing unit. In such a system, the power required for ADC can be reduced, and thus highly efficient information processing can be expected when viewing the system as a whole.

As Table 4 shows, the energy efficiency of ASICs with learning is only a few TOPS/W, which is lower than that of ASICs without learning. This is presumably due to the von Neumann-type architecture, in which different blocks are used for the memory and weight update blocks. In this study, we sacrificed integration and achieved high energy efficiency of up to 18.7 TOPS/W during learning by distributing the memory and weight update circuits in each synaptic circuit. This efficiency during learning was higher than the 15.4 TOPS/W during inference because the number of operations was larger than the 2 OP of the MAC operation (see Note C in Table 4), and this operation is executed efficiently by using the time domain.

The general flow of learning is to calculate the difference between the target value and the output of the neuron, and then to apply a function to the difference to determine the weight update amount. By expanding the information in the time domain, the difference and the function can be calculated simultaneously using a time window function. SNNs can naturally handle temporal information and thus are suitable for implementing efficient on-chip learning hardware.

Network configurations and learning algorithms that can take advantage of the characteristics of SNNs are still in the exploratory stage. Loihi [15] and SpiNNaker 2 [34] are designed to allow a flexibility in the learning algorithm and network configuration. In this study, we limited the learning algorithm to ReSuMe and restricted the flexibility, which resulted in high energy efficiency during learning as shown in Table 4. This result is one example of high energy-efficient on-chip learning hardware that can be realized even with a learning operations with 8-bit synaptic weights. These results show that even when using conventional CMOS technology, SNN hardware with on-chip learning can achieve very high energy efficiency when combined with CIM and an asynchronous architecture without a clock signal.
| Ref. | Technology [nm] | Die area [mm²] | Bitwidth | Synapse type | Power [µW] | On-chip training | MAC operation | Efficiency [TOPS/W] |
|------|-----------------|----------------|----------|--------------|------------|-----------------|----------------|-------------------|
| [19] | 65              | 10.1           | 410      | N/A          | 23.6       | N/A             | N/A            | 3.42              |
| [20] | 28              | 8.97           | N/A      | N/A          | 7.29-45.0  | N/A             | N/A            | 121(94)          |
| [20] | 40              | 9.41           | 100      | N/A          | 87         | Yes             | Analog (charge) | 3.99@91          |
| [20] | 40              | 2.71           | 512      | N/A          | 22k        | Yes             | Analog (charge) | 3.4@1.2 V       |
| [20] | 40              | 2.71           | 512      | N/A          | 22k        | No              | Analog (charge) | 3.4@1.2 V       |
| [21] | 68              | 10.1           | 410      | N/A          | 22k        | Yes             | Analog (charge) | 3.4@1.2 V       |
| [21] | 68              | 10.1           | 410      | N/A          | 22k        | Yes             | Analog (charge) | 3.4@1.2 V       |
| [23] | 40              | 1.51           | 332      | N/A          | 87         | Yes             | Analog (charge) | 3.42              |
| [31] | 65              | 10.1           | 410      | N/A          | 22k        | Yes             | Analog (charge) | 3.42              |
| [31] | 65              | 10.1           | 410      | N/A          | 22k        | Yes             | Analog (charge) | 3.42              |
| [34] | 22              | 10.1           | 410      | N/A          | 22k        | Yes             | Analog (charge) | 3.42              |
| [34] | 22              | 10.1           | 410      | N/A          | 22k        | Yes             | Analog (charge) | 3.42              |
| This work | 40              | 9.41           | 100      | 10k          | 93 x 10⁻³  | Yes             | Analog (charge) | 3.42              |
| This work | 40              | 9.41           | 100      | 10k          | 93 x 10⁻³  | Yes             | Analog (charge) | 3.42              |

TABLE 4. Performance results and comparison table


a) This value was calculated from “Energy per synaptic spike op (min) = 23.6 pJ,” shown in Ref. [15], when 1 MAC is 2 OP.
b) These values were obtained from the macro, which is the synapse circuit array.
c) A learning operation consists of pre-spike × weight and its summation (2 OP), post-spike × time window and its subtraction from the weight (2 OP), and target spike × time window and its addition to the weight (2 OP). One learning operation is thus generally 6 OP, but not always, because of the case where the neuron circuit does not fire when there is pre-spike input.
d) To increase the range of conductance available in the synapse, multiple PCs were used for a given magnitude of conductance update.
e) Executing 8-bit matrix multiplications from local SRAM in the 16 × 4 MAC accelerator.

The energy consumption per synapse circuit was 129.8 [fJ]. (5)

An inference operation consists of pre-spike × weight and its summation (= 2 OP). Thus, energy efficiency during inference was

$$2 \text{ OP}/129.8 \text{ [fJ]} \approx 15.40 \text{ [TOPS/W]}. \quad (6)$$

B. LEARNING

Power consumption using over learning, including standby, was 67.08 µW. Thus, the power consumption without standby was

$$67.08 \text{ [µW]} - 27.02 \text{ [µW]} = 40.06 \text{ [µW]}. \quad (7)$$

The energy consumption of synapse array during learning was calculated by

$$40.06 \text{ [µW]} \times 80 \text{ [µs]} \approx 3.204 \text{ [nJ]}. \quad (8)$$

The energy consumption per synapse circuit was

$$3.204 \text{ [nJ]} / 10,000 = 320.4 \text{ [fJ]}. \quad (9)$$

APPENDIX A

CALCULATIONAL PROCEDURE FOR ENERGY EFFICIENCY

We explain the calculational procedure for the energy efficiency of our fabricated chip during inference and learning. These values of energy efficiency were calculated from the power consumption of the synapse array without the standby power consumption (= 27.02 µW). The energy consumption was calculated from the power consumption when running an 80 µs operation sequence in which spikes were input into 10,000 synaptic circuits.

A. INFERENCE

Power consumption using over inference, including standby, was 43.25 µW. Thus, the power consumption without standby was calculated by

$$43.25 \text{ [µW]} - 27.02 \text{ [µW]} = 16.23 \text{ [µW]}. \quad (3)$$

Energy consumption refers to the product of power and time. The energy consumption of the synapse array during inference was calculated by

$$16.23 \text{ [µW]} \times 80 \text{ [µs]} \approx 1.298 \text{ [nJ]}. \quad (4)$$

The energy consumption per synapse circuit was

$$1.298 \text{ [nJ]} / 10,000 = 129.8 \text{ [fJ]}. \quad (5)$$

An inference operation consists of pre-spike × weight and its summation (= 2 OP). Thus, energy efficiency during inference was

$$2 \text{ OP}/129.8 \text{ [fJ]} \approx 15.40 \text{ [TOPS/W]}. \quad (6)$$

The work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/
A learning operation consists of pre-spike $\times$ weight and its summation (2 OP), post-spike $\times$ time window and its subtraction from the weight (2 OP), and target spike $\times$ time window and its addition to the weight (2 OP). One learning operation is thus generally 6 OP. Thus, energy efficiency during learning was

$$\frac{6 \text{ OP}}{320 \text{ TOPS/W}} \approx 18.72 \text{ [TOPS/W].}$$

However, depending on the time location of pre-spikes, post-spikes, and target spikes, learning may not occur. In this case, only MAC operation is conducted. Thus, energy efficiency was

$$\frac{2 \text{ OP}}{320 \text{ TOPS/W}} = 6.242 \text{ [TOPS/W].}$$

From the above, the energy efficiency during learning was between $6.242 \text{ [TOPS/W]}$ and $18.72 \text{ [TOPS/W]}$.

REFERENCES

[1] J. H. Lau and T. G. Yue, “Thermal management of 3D IC integration with TSV (through silicon via),” in 59th ECTC. IEEE, 2009, pp. 635–640.
[2] D. Bankman, L. Yang, B. Moons, M. Verhelst, and B. Mummert, “An always-on 3.8 $\mu$J/86% CIFAR-10 mixed-signal binary CNN processor with all memory on chip in 28nm CMOS,” in ISSCC. IEEE, 2018, pp. 222–224.
[3] W.-S. Khwa, J.-J. Chen, J.-F. Li, X. Si, E.-Y. Yang, X. Sun, R. Liu, P.-Y. Chen, Q. Li, S. Yu et al., “A 65nm 4k algorithm-dependent computing-in-memory SRAM unit-macro with 2.3 ns and 55.8 TOPS/W fully parallel product-sum operation for binary DNN edge processors,” in ISLPED. IEEE, 2018, pp. 496–498.
[4] S. K. Gonugondla, M. Kang, and N. Shanbhag, “A 42pJ decision 3.12 TOPS/W robust in-memory machine learning classifier with on-chip training,” in ISLPED. IEEE, 2018, pp. 490–492.
[5] S. Okumura, M. Yabuuchi, K. Hijioka, and K. Nose, “A ternary based bit scalable, 8.80 TOPS/W CNN accelerator with more-core processing-in-memory architecture with 896k synapses/mm$^2$,” in VLSC. IEEE, 2019, pp. C248–C249.
[6] X. Si, J.-J. Chen, Y.-N. Tu, W.-H. Huang, J.-H. Wang, Y.-C. Chiu, W.-C. Wei, S.-Y. Xu, X. Sun, R. Liu et al., “A twin-8T SRAM computation-in-memory macro for multiple-bit CNN-based machine learning,” in ISLPED. IEEE, 2019, pp. 396–398.

[7] J.-W. Su, X. Yi, Y.-C. Chou, T.-W. Chang, W.-H. Huang, Y.-N. Tu, R. Liu, P.-J. Lu, P.-J. Liu, H.-J. Wang et al., “A 28nm 64kb inference-training two-way transpose mulbit 6T SRAM compute-in-memory macro for AI edge chips,” in ISSCC. IEEE, 2020, pp. 240–242.
[8] Q. Dong, M. E. Sinangil, B. Erbagic, D. Sun, W.-S. Khwa, H.-J. Liao, Y. Wang, and J. Chang, “A 351 TOPS/W and 372.4 GOPS compute-in-memory SRAM macro in 7nm FinFET CMOS for machine-learning applications,” in ISSCC. IEEE, 2020, pp. 242–244.
[9] Q. Liu, B. Gao, P. Yao, D. Wu, J. Chen, Y. Pang, W. Zhang, Y. Liao, C.-X. Xue, W.-H. Chen et al., “A fully integrated analog ReRAM based 78.4 TOPS/W compute-in-memory chip with fully parallel MAC computing,” in ISSCC. IEEE, 2020, pp. 500–502.
[10] R. Mochida, K. Kouno, Y. Hayata, M. Nakayama, T. Ono, H. Suwa, R. Yasuhara, K. Katayama, K. Tsukawa, and Y. Gohou, “A 4M synapses integrated analog ReRAM based 66.5 TOPS/W neural-network processor with cell current controlled writing and flexible network architecture,” in VLSI. IEEE, 2018, pp. 175–176.
[11] C.-X. Xue, W.-H. Chen, J.-S. Liu, J.-F. Li, W.-Y. Lin, W.-E. Lin, J.-H. Wang, W.-C. Wei, T.-W. Chang, T.-C. Chang et al., “A 1Mb multibit ReRAM computing-in-memory macro with 14.6 ns parallel MAC computing time for CNN based AI edge processors,” in ISSCC. IEEE, 2019, pp. 388–390.
[12] Y. Sakemi, K. Morino, T. Morie, and K. Aihara, “A supervised learning algorithm for multilayer spiking neural networks based on temporal coding toward energy-efficient VLSI processor design,” arXiv preprint arXiv:2001/05348, 2020.
[13] T. Schoenauer, S. Atasoy, N. Mehrta, and H. Klar, “Neuropipe-chip: A digital neuro-processor for spiking neural networks,” IEEE Trans. Neural Netw., vol. 13, no. 1, pp. 205–213, 2002.

J. H. Lau and T. G. Yue, “Thermal management of 3D IC integration with TSV (through silicon via),” in 59th ECTC. IEEE, 2009, pp. 635–640.

D. Bankman, L. Yang, B. Moons, M. Verhelst, and B. Mummert, “An always-on 3.8 $\mu$J/86% CIFAR-10 mixed-signal binary CNN processor with all memory on chip in 28nm CMOS,” in ISSCC. IEEE, 2018, pp. 222–224.

W.-S. Khwa, J.-J. Chen, J.-F. Li, X. Si, E.-Y. Yang, X. Sun, R. Liu, P.-Y. Chen, Q. Li, S. Yu et al., “A 65nm 4k algorithm-dependent computing-in-memory SRAM unit-macro with 2.3 ns and 55.8 TOPS/W fully parallel product-sum operation for binary DNN edge processors,” in ISLPED. IEEE, 2018, pp. 496–498.

S. K. Gonugondla, M. Kang, and N. Shanbhag, “A 42pJ decision 3.12 TOPS/W robust in-memory machine learning classifier with on-chip training,” in ISLPED. IEEE, 2018, pp. 490–492.

S. Okumura, M. Yabuuchi, K. Hijioka, and K. Nose, “A ternary based bit scalable, 8.80 TOPS/W CNN accelerator with more-core processing-in-memory architecture with 896k synapses/mm$^2$,” in VLSC. IEEE, 2019, pp. C248–C249.

X. Si, J.-J. Chen, Y.-N. Tu, W.-H. Huang, J.-H. Wang, Y.-C. Chiu, W.-C. Wei, S.-Y. Xu, X. Sun, R. Liu et al., “A twin-8T SRAM computation-in-memory macro for multiple-bit CNN-based machine learning,” in ISLPED. IEEE, 2019, pp. 396–398.

J.-W. Su, X. Yi, Y.-C. Chou, T.-W. Chang, W.-H. Huang, Y.-N. Tu, R. Liu, P.-J. Lu, P.-J. Liu, H.-J. Wang et al., “A 28nm 64kb inference-training two-way transpose mulbit 6T SRAM compute-in-memory macro for AI edge chips,” in ISSCC. IEEE, 2020, pp. 240–242.

Q. Dong, M. E. Sinangil, B. Erbagic, D. Sun, W.-S. Khwa, H.-J. Liao, Y. Wang, and J. Chang, “A 351 TOPS/W and 372.4 GOPS compute-in-memory SRAM macro in 7nm FinFET CMOS for machine-learning applications,” in ISSCC. IEEE, 2020, pp. 242–244.

Q. Liu, B. Gao, P. Yao, D. Wu, J. Chen, Y. Pang, W. Zhang, Y. Liao, C.-X. Xue, W.-H. Chen et al., “A fully integrated analog ReRAM based 78.4 TOPS/W compute-in-memory chip with fully parallel MAC computing,” in ISSCC. IEEE, 2020, pp. 500–502.
SEIJI UENOHARA received a B.S. degree from Tokyo Denki University in 2011. He received M.S. and Ph.D. degrees in engineering from the Kyushu Institute of Technology, Fukuoka, Japan, in 2013 and 2016, respectively. He was an industry–government–academia researcher in the Center for Innovative Integrated Electronic Systems (CIES), Tohoku University in 2016. He was a project research associate in the Brain Morphic Artificial Intelligence (BMAI) social cooperation programs of the University of Tokyo from 2017 to 2020. He is currently Project Assistant Professor in the Graduate School of Life Science and System Engineering, Kyushu Institute of Technology, Japan. His research interests include the hardware design of neural networks, nonlinear dynamical systems, and time-series data analysis.

KAZUYUKI AIHARA received a B.E. degree in electrical engineering and Ph.D. degree in electronic engineering from the University of Tokyo in 1977 and 1982, respectively. He was a Professor with the Institute of Industrial Science at the University of Tokyo from 2003 to 2020. He led the Exploratory Research for Advanced Technology (ERATO) Aihara Complexity Modelling project by the Japan Science and Technology Agency (JST) from 2003 to 2008 and the FIRST Innovative Mathematical Modelling project by the Japan Society for the Promotion of Science (JSPS) through the Funding Program for World-Leading Innovative R&D Science and Technology (FIRST) program from 2010 to 2014. Currently, he is University Professor, Professor Emeritus and Deputy Director of International Research Center for Neurointelligence at the University of Tokyo. His research interests include mathematical modeling of complex systems, advanced control theory of complex systems, complex network theory, nonlinear data analysis, and data-driven modeling.