sBSNN: Stochastic-Bits Enabled Binary Spiking Neural Network With On-Chip Learning for Energy Efficient Neuromorphic Computing at the Edge

Minsuk Koo, *Member, IEEE*, Gopalakrishnan Srinivasan, Yong Shim, *Member, IEEE*, and Kaushik Roy, *Fellow, IEEE*

***Abstract*— In this work, we propose stochastic Binary Spiking Neural Network (sBSNN) composed of stochastic spiking neurons and binary synapses (stochastic only during training) that computes probabilistically with one-bit precision for power-efficient and memory-compressed neuromorphic computing. We present an energy-efficient implementation of the proposed sBSNN using *'stochastic bit'* as the core computational primitive to realize the stochastic neurons and synapses, which are fabricated in 90nm CMOS process, to achieve efficient on-chip training and inference for image recognition tasks. The measured data shows that the *'stochastic bit'* can be programmed to mimic spiking neurons, and stochastic Spike Timing Dependent Plasticity (or sSTDP) rule for training the binary synaptic weights without expensive random number generators. Our results indicate that the proposed sBSNN realization offers possibility of up to 32× neuronal and synaptic memory compression compared to full precision (32-bit) SNN and energy efficiency of 89.49 TOPS/Watt for two-layer fully-connected SNN.**

***Index Terms*— Stochastic bit, stochastic binary SNN, stochastic STDP, memory compression, neuromorphic computing.**

I. Introduction

IN THE current era of ubiquitous autonomous intelligence, there is a growing need for moving Artificial Intelligence (AI) to the edge to cope with the ever increasing demand for autonomous systems like drones, self-driving cars, and smart wearable devices. Energy-efficient neuromorphic systems are henceforth necessary to process the massive amount of data generated by the resource-constrained battery-powered edge devices. Furthermore, it is highly desirable to embed on-chip intelligence using low-complexity learning rules, which enable the edge devices to learn from real-time inputs. Real-time on-chip learning capability precludes the need for offline training in the cloud, which can otherwise lead to higher latency and security concerns for real-time applications.

Spiking Neural Networks (SNNs), on the account of event-driven computing capability and hardware-friendly local learning using Spike Timing Dependent Plasticity (STDP), offer a promising solution for realizing energy-efficient neuromorphic systems with on-chip intelligence. In fact, researchers in [1] demonstrated that SNN running on event-driven neuromorphic hardware like Intel *Loihi* [2] incurs the minimum energy per inference relative to similarly sized analog neural network executed on CPU/GPU while providing equivalent inference accuracy for a latency-critical keyword spotting task. Recent works on deep SNNs indicate that energy efficiency significantly increases with network depth due to exponential drop in the spiking activity across successive SNN layers [3], [4]. In this regard, prior works proposed energy-efficient implementations of SNN using CMOS [2], [5], [6] and emerging device technologies such as Resistive Random Access Memory (RRAM) [7], [8], Conductive Bridge RAM (CBRAM) [9], and Magnetic Tunnel Junctions (MTJs) [10]. However, SNNs composed of deterministic neuronal and synaptic models require multi-bit precision to store the parameters governing their dynamics. As a result, the computational complexity and neuronal/synaptic memory requirements increase with network size, leading to reduction in the overall power- and area-efficiency.

We propose and implement *'stochastic bits'* enabled binary SNN (sBSNN) that computes probabilistically with one-bit precision for energy- and memory-efficient neuromorphic computing at the edge. The core building block of the sBSNN is a *'stochastic bit'*, which switches between its logic low and high states with a probability that varies in a sigmoidal manner based on the input. We realize the stochastic neurons, referred to as sNeurons, and synapses (stochastic only during training) using the proposed *'stochastic bit'* as explained below. The

Manuscript received August 3, 2019; revised November 21, 2019, January 8, 2020, and February 23, 2020; accepted March 4, 2020. Date of publication March 16, 2020; date of current version July 31, 2020. This work was supported in part by the Center for Brain Inspired Computing (C-BRIC), one of the six centers in JUMP, in part by the Semiconductor Research Corporation (SRC) Program sponsored by the DARPA, in part by Semiconductor Research Corporation, in part by the National Science Foundation, in part by Intel Corporation, and in part by the DoD Vannevar Bush Fellowship. This article was recommended by Associate Editor A. James. *(Corresponding author: Minsuk Koo.)*

Minsuk Koo, Gopalakrishnan Srinivasan, and Kaushik Roy are with the Department of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: koom@purdue.edu; srinivg@purdue.edu; kaushik@purdue.edu).

Yong Shim is with Intel Corporation, Hillsboro, OR 97125 USA.

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSI.2020.2979826

sNeuron receives the weighted sum of the input spikes with the synaptic weights, and spikes probabilistically depending on the weighted input sum. The firing probability of the sNeurons, similar to the switching dynamics of the *'stochastic bit'*, has sigmoidal relationship with the weighted input sum. The binary synapse interconnecting a pair of input (pre) and output (post) neurons is similarly emulated using the *'stochastic bit'* during training. The binary synaptic weight is trained using the stochastic-STDP (sSTDP) algorithm presented in [11], where the synaptic weight is potentiated/depressed with a probability that depends on the degree of correlation between the spike times of the pre- and post-neurons. The trained binary synaptic weights are then used deterministically during inference to predict the class of a test input. The proposed sBSNN, with event-driven computing capability enabled by state-less sNeurons and memory-efficient on-chip learning capability enabled by the hardware-friendly localized sSTDP rule, offers a promising solution for building the next generation of autonomous intelligent systems.

To that effect, we propose an energy-efficient realization of sBSNN, fabricated in 90nm CMOS technology, to achieve on-chip training and inference for visual image recognition tasks. The proposed *'stochastic bit'* is composed of a cross-coupled inverter with PMOS header and NMOS footer transistors for obtaining the sigmoidal switching probability characteristics. We interface the CMOS *'stochastic bit'* with the appropriate peripheral circuitry to realize the sNeurons and synapses. The energy and memory efficiency of the proposed implementation stems from three key factors. First, the power consumed by the sNeuron for generating a spike is comparable to that consumed in a single transition of a cross-coupled inverter, which is typically in the order of few μW. In addition, the *'stochastic bit'* design also leverages power gating technique [12] with header and footer transistors between the supply and ground rails for reducing the leakage power consumption. Second, the spiking dynamics of the sNeuron depend only on the current input and not on the integrated sum of the current and past inputs, which precludes the need for storing the neuron state (typically known as the membrane potential) as is common in deterministic spiking neurons like the leaky integrate-and-fire neuron. Further, the synapses need only one-bit storage to record the respective binary states. Last, the weighted sum of the inputs with the synaptic weights, which is typically a series of multiply and accumulate (MAC) operations in analog neural networks, is transformed to AND operations followed by pulse count in the proposed sBSNN, thereby reducing the computational energy significantly. Our analysis using a two-layer fully-connected SNN of 400 neurons indicates that the proposed realization offers high energy efficiency of 89.49 TOPS/Watt, which renders it a potential candidate for enabling the next generation of intelligent devices.

In summary, we make the following contributions:

- We proposed the *'stochastic bit'* as the core computational primitive to realize the stochastic neurons and binary synapses, which are implemented in 90nm CMOS process.
- We proposed and evaluated the *'stochastic bit'* enabled sBSNN that computes probabilistically with one-bit precision for power-efficient and memory-compressed neuromorphic computing.
- We proposed and demonstrated one of the first works on all-CMOS realization of stochastic SNNs. Our proposal provides reconfigurable on-chip learning that is suitable for the real-time and resource constrained edge devices.

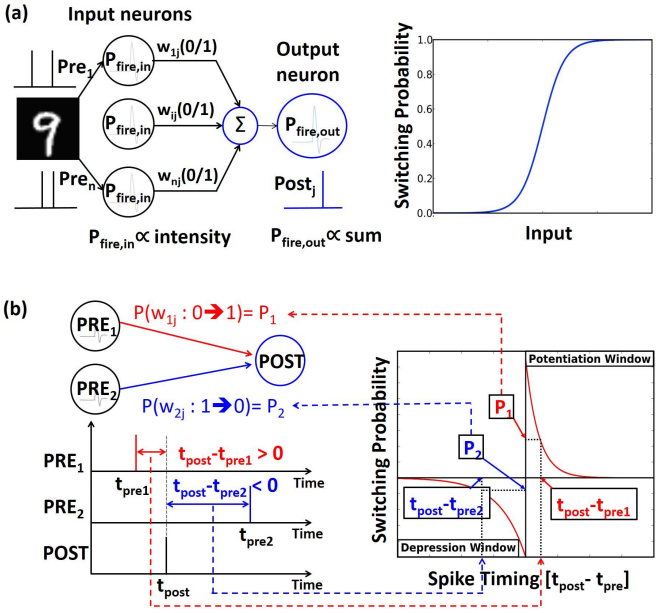


Fig. 1. (a) SNN composed of stochastic input and output neurons interconnected via binary synaptic weights. The output sNeurons fire with a probability that has sigmoidal relationship with the weighted input sum. (b) Stochastic-STDP learning rule for binary synaptic weights, where the synaptic switching probability depends on the time difference between input (pre) and output (post) spikes. The binary weight is probabilistically potentiated (depressed) for positive (negative) timing correlation between pre- and post-spikes.

The rest of the paper is organized as follows. Section II details the proposed sBSNN and sSTDP training rule. Section III describes the *'stochastic bit'* circuit design and the required peripherals for implementing the sNeurons and synapses. The system-level implementation of the sBSNN is also detailed in this section. Section IV presents the measured characterization results of the sNeurons and synapses, and the accuracy and energy efficiency of our sBSNN realization. Finally, section V concludes the paper.

II. Background

A. Stochastic Binary Spiking Neural Network (sBSNN)

The core building block of the proposed sBSNN is a set of input (pre) neurons connected to an output (post) neuron via binary weights. The input neurons, which represent the image pixels for a visual object recognition task, generate Poisson-distributed spikes at a rate proportional to the corresponding pixel intensities. At any given time, the input pre-spikes get modulated by the interconnecting synaptic weights to produce resultant current into the output neuron.

Several previous works have explored the hardware implementations for these core building blocks of stochastic SNNs, using emerging technologies like CBRAMs and MTJs [11], [13] and built-in blocks in FPGA board [14]. We proposed a *'stochastic bit'* as the core building block for neuron and synapse (training) to achieve on-chip learning with compressed memory. We model the output neuron using the *'stochastic bit'*, which spikes probabilistically based on the input current (or weighted input sum) during both training and inference. The spiking probability of the output sNeuron has sigmoidal dependence on the input current as illustrated in Fig. 1(a). It is important to note that the sNeuron is state-less since the stochastic spiking dynamics depend only on the instantaneous input current and not on the integrated sum of current and past input currents as is typical in deterministic neuron models, thereby eliminating the multi-bit precision requirement for the neuron state (or membrane potential). The stochastic synapses (stochastic only during training) are similarly emulated using the *'stochastic bit'*, where the synaptic switching probability depends on the time difference between the pre- and post-spikes as explained in the following subsection II-B.

B. Stochastic-STDP (sSTDP)

Spike Timing Dependent Plasticity (STDP) is a bio-inspired local learning mechansim, which has been experimentally observed in the rat hippocampus [15]. STDP postulates that the change in the weight of a multi-level synapse interconnecting a pair of pre- and post-neurons depends on the correlation between the respective spike times. If the pre-neuron spikes before the post-neuron, the synaptic weight increases (synaptic potentiation), while it decreases if the pre-neuron spikes after the post-neuron (synaptic depression). Binary synapses, on the contrary, require a probabilistic learning rule to prevent rapid switching of the weights between the high and low levels, which would otherwise render the synapses memory-less. We use the sSTDP learning algorithm proposed in [11] to train the binary synaptic weights, where the synaptic switching probability has exponential dependence on spike timing difference as illustrated in Fig. 1(b) and described by

$$P_{L\rightarrow H} = \gamma_{pot} \cdot e^{\frac{-\Delta t}{\tau_{pot}}} where\ \Delta t = t_{post} - t_{pre} > 0 \quad (1)$$

$$P_{H\rightarrow L} = \gamma_{dep} \cdot e^{\frac{\Delta t}{\tau_{dep}}} where\ \Delta t = t_{post} - t_{pre} < 0 \quad (2)$$

where $P_{L\rightarrow H}$ and $P_{H\rightarrow L}$ are the probability of potentiation and depression, respectively. In other words, the weight of a synapse changes based on the temporal correlation between the spike time of pre- and post-neurons. For example, if a pre- (post-) neuron fires before a post- (pre-) neuron does, it is positively (negatively) correlated with the input pattern [16]. Consequently, potentiation (depression) occurs probabilistically in the positive (negative) timing window of the sSTDP algorithm. The corresponding switching probability is determined by the spike timing difference between pre and post spikes as described in the above equations. The peak switching probability and time constant for potentiation (γ_{pot}, τ_{pot}) and depression (γ_{dep}, τ_{dep}) determine the synaptic learning efficacy. The sSTDP hyperparameters have to be chosen carefully to ensure right balance between the potentiation and depression weight updates, and achieve efficient learning. Once the training is complete, the learnt binary weights are used deterministically during inference. The presented sBSNN requires only one-bit precision for the neurons and synapses, leading to visual image recognition with compressed memory requirement.

III. sBSNN Design and Implementation

In this section, we first detail the design and implementation of the proposed *'stochastic bit'*, which is the core computing primitive of the sBSNN. We then present the design of sNeuron and synapse (stochastic only during training). Finally, we detail the system-level realization of two-layer fully-connected sBSNN for visual image recognition.

A. CMOS 'Stochastic Bit' Design

As mentioned in section I, controllable stochastic behavior is the central characteristic of the *'stochastic bit'*. In CMOS-based designs, stochastic behavior is largely dependent on the characteristics of the random noise source. Thermal noise is one of the commonly used entropy sources in CMOS process, which stems from the channel fluctuations induced by random Brownian motion of electrons. The power spectral density of thermal noise across a resistor is given by $V^2 = 4kTR$, where k is the Boltzman constant, T is the temperature in Kelvin, and R is the resistance in ohms. Accordingly, thermal noise induced stochasticity is only affected by the device resistance and operating temperature. Thermal noise has been used as the source of randomness in many True Random Number Generator (TRNG) designs [17], [18]. Also, metastability-based TRNG designs using cross-coupled inverters have been reported to achieve high operating frequency and power efficiency [19]. This motivated us to investigate the possibility of harnessing the metastable behavior of bi-stable circuits to implement the *'stochastic bit'*.

The proposed *'stochastic bit'* is realized using cross-coupled inverter with PMOS header transistors and NMOS footer transistors as depicted in Fig. 2(a). The operation of the *'stochastic bit'* is divided into two different modes, namely, pre-charge and evaluation, which are gated by the 'EN' (enable) signal as shown in Fig. 2(b). In the pre-charge mode (when 'EN' is low), the cross-coupled nodes A and B are pre-charged to the same voltage by leakage current, while the header and the footer transistors are turned off. Note that, the inherent power gating enabled by the PMOS header transistors and the NMOS footer transistors causes the leakage current of the proposed design to be lower than the gate leakage current of a 6T SRAM bitcell [12]. The switching probability depends on asymmetry in the effective strength of left- and right-wing PMOS transistors, which can be modulated using the input that is represented as 6-bit code in our implementation and activates different binary weighted PMOS switch transistors. The NMOS footer transistors connected to ground are controlled symmetrically in strength using the same input code, which is represented with 3-bit precision in our implementation, to modulate the shape of the probability curve. The shape of the switching probability versus the PMOS digital code is

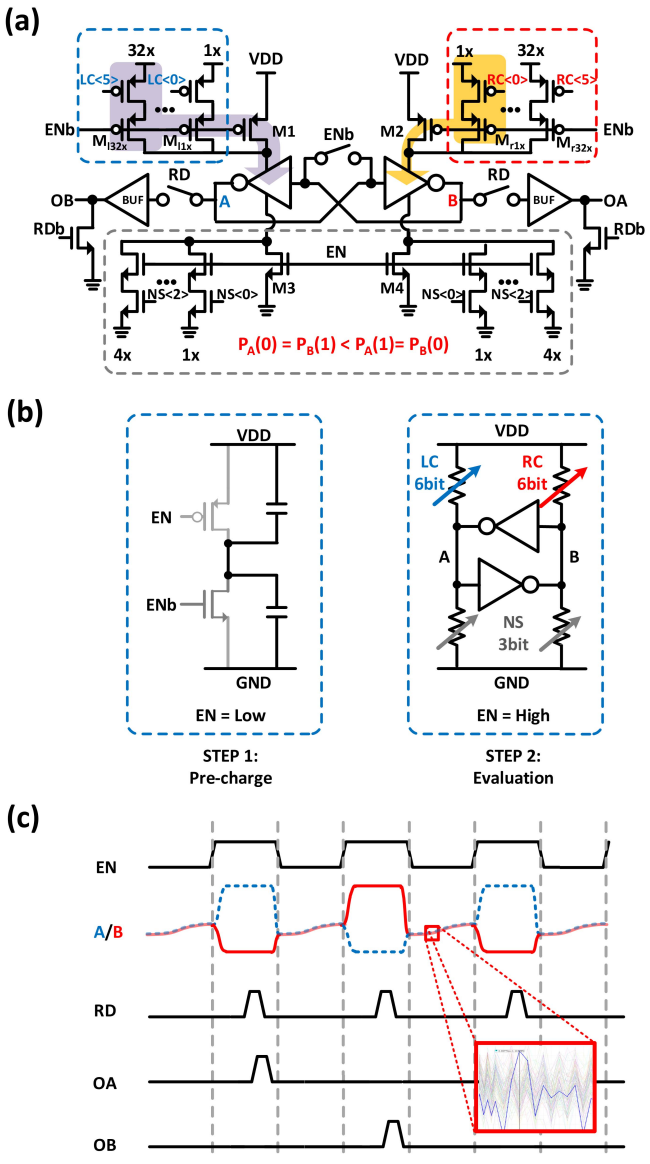


Fig. 2. (a) Schematic of 6-bit *'stochastic bit'* core. (b) Illustration of the pre-charge and evaluation modes of operation of the *'stochastic bit'*. (c) Timing diagram illustrating the operation of the *'stochastic bit'*.

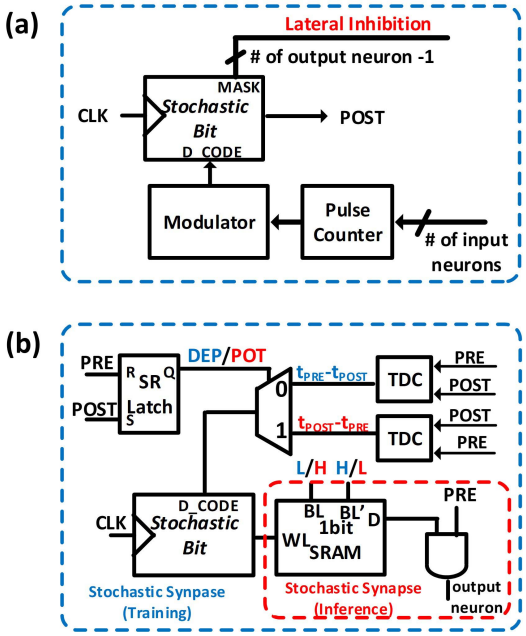


Fig. 3. Design of 'Stochastic bit' enabled (a) spiking neuron, and (b) binary synapse (stochastic during training and deterministic during inference).

sigmoidal as will be shown in the results section IV. The shape and the covered range of probability is programmable and can be reconfigured on-chip. It is worth noting that, the *'stochastic bit'* consumes only leakage power during the pre-charge mode, and charging/discharging power for nodes A and B during the evaluation mode. In addition, the speed of operation is based on the speed of 'EN' signal. Therefore, the proposed design becomes more power efficient and faster as CMOS process scales. Also, the PMOS and NMOS sizing, and bit-precision for the respective codes can be tuned based on the application requirements.

B. Stochastic Neuron (sNeuron)

We now describe how the *'stochastic bit'* is used to realize stochastic input and output neurons forming the sBSNN. The input neurons map the image pixel intensities to spike trains, where each neuron fires probabilistically at a rate proportional to the corresponding pixel intensity. The *'stochastic bit'* can inherently realize an input sNeuron by mapping the pixel intensity to PMOS code that controls its switching probability. On the contrary, the *'stochastic bit'* is interfaced with counter and modulator circuit (shown in Fig. 3(a)), which generates and modulates the weighted input, for realizing the output sNeuron that spikes with the desired probability. Also, the spiking activity of the sNeuron can be suppressed by masking the 'EN' signal of the *'stochastic bit'*, which is used for implementing lateral inhibition that facilitates competitive learning as will be explained in subsection III-D. The generated spikes from the input and the output sNeuron (PRE and POST) are applied to the stochastic binary synapses for synaptic updates as explained below.

C. Stochastic Binary Synapse

The stochastic binary synapse (during training) is realized by interfacing the *'stochastic bit'* with 6T SRAM as depicted in Fig. 3(b). Based on the sign of the spike timing difference, $t_{post} - t_{pre}$, the synaptic weight update event is determined as potentiation (depression) when the sign is positive (negative). Then, the spike timing difference, measured as the number of clock pulses using time to digital converter (TDC), feeds the *'stochastic bit'* to selectively turn on the PMOS header transistors, effectively causing it to produce an output pulse with the appropriate probability depending on spike timing. Note that TDC can be realized using a counter for potentiation (depression) that resets when PRE (POST) is high. TDC is shared by stochastic synapses that are activated by the same PRE/POST signal. The generated pulse activates the wordline of the 6T SRAM cell while the bitline is driven to VDD (ground) for synaptic potentiation (depression) update. Once the stochastic

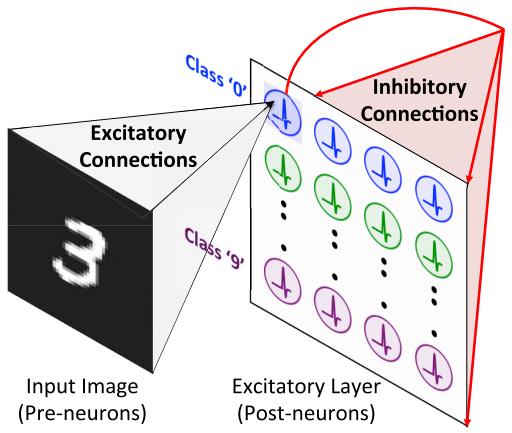


Fig. 4. Architecture of two-layer fully-connected sBSNN, with lateral inhibition, for object recognition. The input pixels (pre-neurons) are fully-connected via stochastic binary synapses to excitatory post-neurons, which are trained using the sSTDP learning algorithm. The excitatory neurons are subdivided into different groups, where each group is trained on a unique class of image patterns.

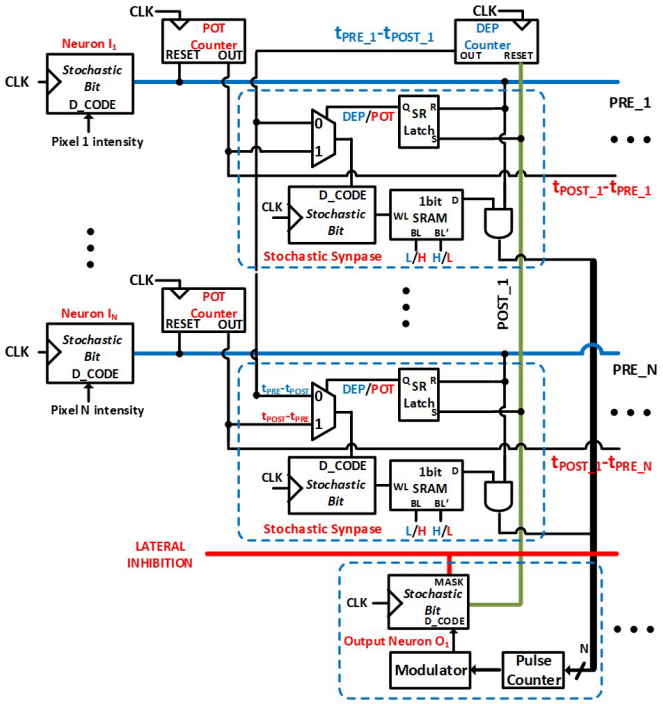


Fig. 5. System-level realization of two-layer fully-connected sBSNN with lateral inhibition.

training process is complete, the *'stochastic bit'* is powered off and the learnt binary weight stored in the corresponding SRAM cell is deterministically used during inference as shown in Figure 3(b). Note that, during both training and inference, the computation of the weighted input sum reduces to AND operations followed by pulse count since both the inputs and synaptic weights are binary. Hence, the sBSNN provides much higher computational energy efficiency relative to analog neural networks with real-valued (32-bit) inputs and synaptic weights, which require MAC (multiply-and-accumulate) units, and SNNs with real-valued weights and binary inputs that need accumulators for computing the weighted input sum.

D. sBSNN System-Level Implementation

1) On-Chip Training: We demonstrate the efficacy of the proposed sNeuron and synapse using a two-layer fully-connected sBSNN depicted in Fig. 4. Fig. 5 illustrates the system-level implementation of the two-layer sBSNN. The input sNeurons representing the image pixels are fully-connected via binary weights to output (post) sNeurons. At every time-step, the weighted sum of the input spikes with the synaptic weights are modulated and fed to the *'stochastic bit'* in the respective post-neurons, causing them to fire probabilistically. The weighted sum received by each post-neuron is calculated by counting the number of pulses from the output of the AND gates in the corresponding column of synapses as depicted in Fig. 5. The pulses are only generated when both inputs of the AND gate are high. Accordingly, power is only dissipated when there are transitions in the AND gate. As a result, the weighted input sum computation in sBSNN consumes significantly lower power compared to full precision (32-bit) SNN. In the event of a post-spike, the spiking neuron inhibits the remaining post-neurons, as illustrated in Fig.4, by masking the respective enable (EN) inputs as explained in subsection III-B to uniquely learn the presented pattern. The synapses connecting the input to the spiking post-neuron are probabilistically potentiated based on spike timing. The spike timing difference, $t_{post} - t_{pre}$ ($t_{pre} - t_{post}$) in the number of clock pulses, is measured using the POT (DEP) counter shown in Fig. 5, which is reset at every pre-spike (post-spike) and decremented by unity at successive time-steps. The elapsed count of POT (DEP) counter is sampled upon a post-spike (pre-spike) for potentiation (depression) weight update. The spike timing difference is fed to the *'stochastic bit'* in the synapses (depicted in Fig. 3(b)), which in turn probabilistically programs the SRAM as detailed in subsection III-C. The sSTDP-based probabilistic weight updates enable each excitatory neuron to learn a complete representation of the input pattern in the input to excitatory synaptic weights. In order to ensure that each excitatory neuron learns unique input representations, we divided the excitatory neurons into different clusters and trained each cluster of neurons on a distinct class of input patterns as proposed in [11]. Fig. 6 shows the MNIST digit representations learnt by a two-layer fully-connected sBSNN of 400 excitatory neurons using the sSTDP-based training methodology.

2) On-Chip Inference: At the end of training, each post-neuron learns to spike for a unique input class by encoding a general input representation in the input to output synaptic weights as shown in Fig. 6. Once training is completed, we disable the clock signal of the *'stochastic bit'* in the synapses, thereby fixing the weights for the inference phase. The learnt binary weights, stored in the SRAM cells, are used deterministically during inference. A test pattern is predicted to belong to the class learnt by the group of neurons with the highest average spike count over the time period for which the test input is presented. The proposed sBSNN implementation, by virtue of using simpler weighted input sum computation

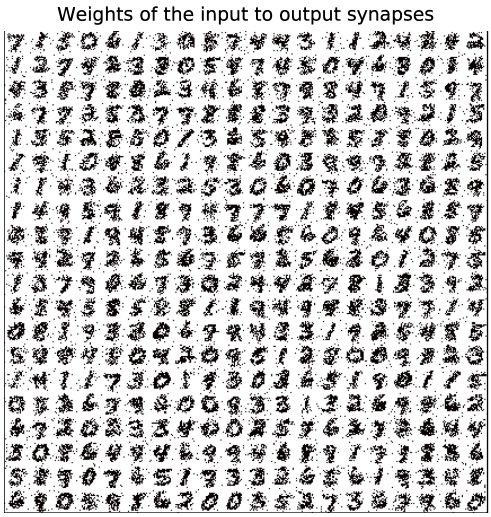


Fig. 6. MNIST digit representations (28 × 28 in dimension) learnt by a two-layer fully-connected sBSNN of 400 excitatory neurons (organized in 20 × 20 grid).

and state-less stochastic neurons, can provide high energy efficiency during inference as will be shown in section IV.

IV. Results

In this section, we first present the measured results of the sNeuron and synapse, which are fabricated in 90nm CMOS process. We subsequently show the simulation results of our sBSNN implementation (detailed in subsection subsection III-D) using the measured neuronal and synaptic dynamics on the MNIST dataset.

A. 'Stochastic Bit' Characterization

Fig. 7(a) illustrates the setup for characterizing the CMOS *'stochastic bit'* design (detailed in section III). The on-chip timing controller generates sufficient number of enable (EN) pulses, which is set to 768 in our experiments, for obtaining reasonable estimate of the *'stochastic bit'* switching probability for a specific configuration of PMOS and NMOS codes. The number of resultant output pulses at OA (refer to Fig. 2(a)) is recorded by a 15-bit on-chip counter to determine the switching probability for the chosen PMOS and NMOS codes. For every set of input codes, we performed the switching probability measurement 1000 times. Fig. 7(b) shows that the switching probability of the *'stochastic bit'* varies roughly in a sigmoidal manner with the PMOS code. The measured switching probability ranges from 11.6% to 90.1% with less than 5% standard deviation at a supply voltage of $1.4V$. In addition, we varied the NMOS code and found that it controls the shape of the switching probability curve as illustrated in Fig. 7(c). The variation in the switching probability dynamics with the NMOS code can be attributed to the change in the respective transistor sizes relative to the PMOS transistor sizes. Note that, the ratio of minimum to maximum switching probability is determined by the bit-precision of the PMOS code and the relative sizing (widths) of the PMOS and NMOS transistors, which need to be fixed at design-time based on the application requirements.

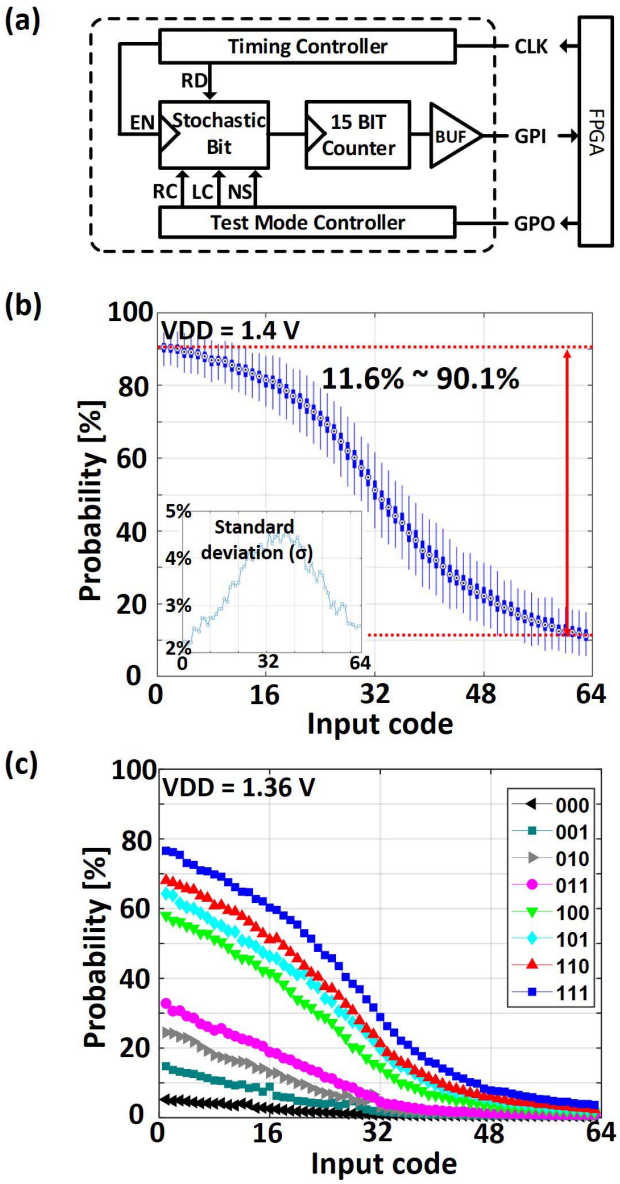


Fig. 7. (a) Measurement setup for the *'stochastic bit'* design, which is interfaced with a FPGA board for generating the on-chip clock/inputs and monitoring the outputs via the CLK and GPIO ports, respectively. (b) The measured box plots of switching probability versus the input (PMOS) digital code, and its standard deviation, σ (refer to the inset). In each box, the central mark indicates the median, the ends of the vertical blue boxes indicate the 25th and 75th percentiles, respectively, and the lines indicate the min and max value. (c) Switching probability dynamics for different 3-bit NMOS codes.

B. Stochastic Binary Synapse

The sSTDP dynamics required for training a binary synaptic weight are obtained by feeding the spike timing difference to the on-chip pulse generator, which generates the pre- and post-spikes as shown in Fig. 8(a). The Time-to-Digital Converter (TDC) measures the spike timing difference and produces the PMOS code for the *'stochastic bit'*, which probabilistically activates the SRAM wordline. The SRAM cell is then probed for potentiation (depression) event to estimate the sSTDP characteristics for the positive (negative) timing window. We adopted a methodology similar to that used for the *'stochastic bit'* characterization for measuring the sSTDP

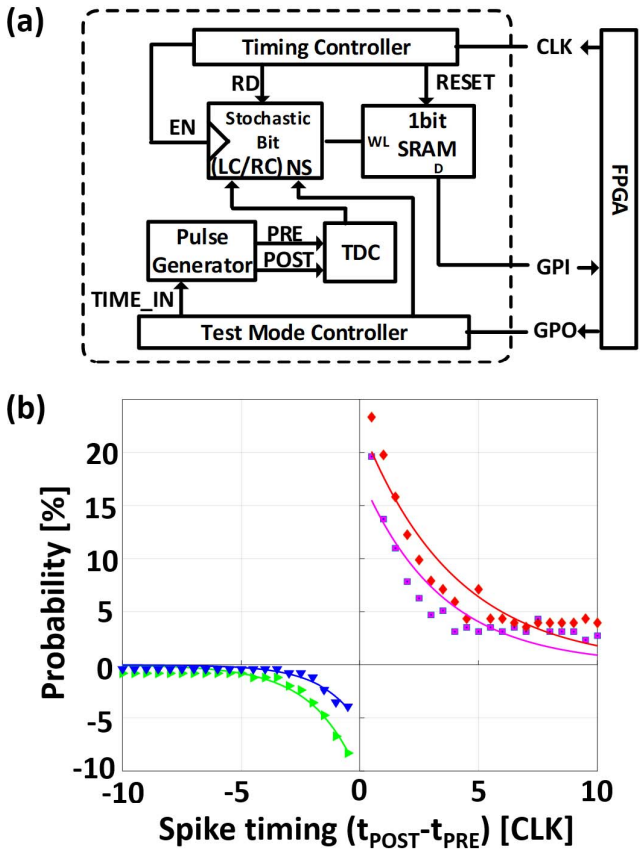


Fig. 8. (a) Measurement setup for the sSTDP dynamics needed to train a stochastic binary synapse, which is interfaced with an FPGA board for generating the clock and inputs (spike timing, TIME_IN), and monitoring the outputs (state of SRAM cell). (b) The measured sSTDP curve for different NMOS codes.

dynamics as explained below. For every value of spike timing within the sSTDP window, TIME_IN in Fig. 8(a), we generated sufficient number of enable pulses (set to 768 as explained in subsection IV-A) for the *'stochastic bit'* constituting the binary synapse. We then probed the 6T SRAM for a change in the cell state to determine the corresponding switching probability. We repeated the switching probability measurement 1000 times for every value of spike timing. Fig. 8(b) shows the measured sSTDP dynamics, where the synaptic switching probability has roughly exponential dependence on spike timing, which conforms to the sSTDP rule depicted in Fig. 1(b). The sSTDP dynamics can be tuned on-chip by programming the NMOS code controlling the footer transistor sizes in the *'stochastic bit'* as explained in subsection III-A. Note that the Time-to-Digital Converter (TDC in Fig. 8(a)) and pulse generators are used only for measurements. The binary synapse is composed of only the 6T-SRAM and the *'stochastic bit'* during training, where the pre- and post-spikes are generated by the input and output sNeurons, respectively, constituting the sBSNN. Also, the spike timing difference is estimated using a counter per pre-/post-neuron as described in subsection III-D.

C. sBSNN for MNIST Digit Recognition

The sBSNN implementation was functionally trained and evaluated using the measured neuronal and synaptic dynamics shown in Figs. 7(b) and 8(b), respectively, on the MNIST digit recognition dataset. The accuracy on the test dataset is 65.88% for an SNN of 400 excitatory neurons trained on 900 MNIST digit patterns, which was sufficient for all the neurons to learn general input representations as depicted in Fig. 6. Any more increase in the number of training patterns could deteriorate the learnt representations, leading to further loss in accuracy. The accuracy can be improved by increasing the number of excitatory neurons and/or by incorporating an additional fully-connected classification layer trained on a larger fraction of the dataset. We augmented the SNN with a softmax readout layer of 10 neurons corresponding to the 10 classes in the MNIST handwritten digit recognition task, where each readout neuron is fully-connected to all the excitatory neurons. For a given input pattern, the spike count of the excitatory neurons are estimated using the sSTDP trained sBSNN, and subsequently fed to the softmax readout layer, which predicts the test pattern to belong to the category represented by the readout neuron with the highest activation. We trained the readout layer on the entire training dataset using the Adam optimizer [26], which is a popular gradient-based supervised training algorithm, and cross-entropy loss function with learning rate of 0.001 for 8 epochs. We obtained higher accuracy of 92.30% on the entire MNIST test dataset of 10,000 images.

sBSNN offers possibility of up to 32× neuronal and synaptic memory compression relative to similarly sized full precision (32-bit) SNN with accuracy loss that can be minimized for larger SNNs. The energy of the sNeuron with the measurement blocks (refer to the sNeuron measurement setup in Fig. 7) is measured to be 8.4pJ/spike. The standalone neuronal energy is estimated to be 1.84pJ/spike as detailed in Table I. In addition, Table I also indicates that the proposed implementation offers lower neuronal energy consumption compared to related works in 90nm CMOS process.

TABLE I
COMPARISON WITH RELATED WORKS

	This Work	2016 VLSI [20]	2015 IEDM [21]	2017 VLSI Report [22]	2015 TCAS II [23]
Learning Rules	Stochastic STDP	Stochastic STDP	STDP	Modulated STDP [24]	STDP
STDP Timing Window	267ns (10 time steps) @37.5MHz	10ms	100us	N/A	3.5us
Stochastic deviation	<5%	N/A	N/A	N/A	N/A
On-chip reconfigurable	YES	N/A	YES	YES	YES
Energy /spike/neuron	8.4pJ* /1.84pJ**	N/A	N/A	11.9 μW***	9.3pJ /3.6pJ***
System Configuration	Stochastic Neuron/Synapse	RRAM Synapse IF neuron	PCM Synapse LIF neuron	Stochastic Neuron/Synapse	RRAM Synapse IF neuron
Accuracy	92.30% (784 ×400 × 10) Trained on 60k MNIST digits	86% (784 ×10) Trained on 50k MNIST digits	N/A	≈ 88% (784 ×500 × 10) Trained on 50k MNIST digits	N/A
Technology	90nm	Non-CMOS	Non-CMOS	Non-CMOS	180nm

* Measured power: 'stochastic bit' + 15b counter + etc. = 226μA*1.4V*26.7ns = 8.4pJ
** Estimated neuron power: 226μA * (33.3/153.2)*1.4V*26.7ns = 1.84pJ
(Post-layout simulated current: 153.2μ = 'stochastic bit'[33.3μ]+ others[119.9μ])
*** Peak power with a single spike duration of ≈ 10 μs
**** Normalized power [25] from 180nm to 90nm: 9.3pJ *(90/180)*1.4/1.8=3.61pJ

D. Energy Efficiency

Finally, we estimate the energy efficiency of the two-layer sBSNN implementation composed of 784 input and 400 output

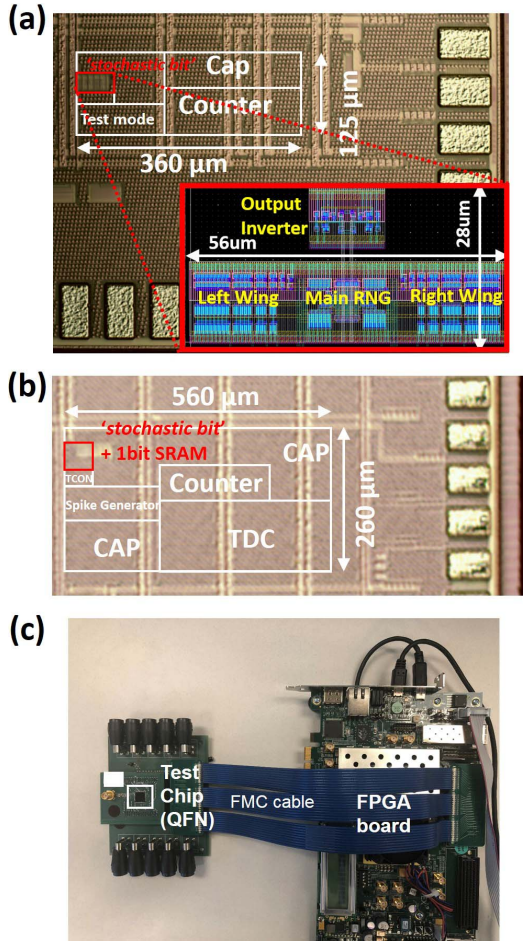


Fig. 9. (a) Die photo of the *'stochastic bit'* and its layout (refer to the inset). (b) Die photo of the stochastic binary synapse composed of the *'stochastic bit'* and 6T-SRAM bitcell. (c) Test chip measurement setup using FPGA.

sNeurons in terms of Tera-operations (TOPS) per Watt. Our functional simulations indicated that the average number of transitions in the AND gate of stochastic synapses is ∼700 out of 784 × 400 total possible transitions. The average power consumed by the AND gate per transition in 90nm CMOS process is estimated as $0.80\mu W$, which totals to 0.56mW per time step. Every output sNeuron requires a 10-bit ones counter for accumulating the maximum weighted input sum of 784, and the *'stochastic bit'* to spike probabilistically. The average weighted input sum received by the output sNeurons is functionally determined to be 21. The average power consumed by the 10-bit ones counter is estimated to be 0.558mW per sNeuron while that of the *'stochastic bit'* is measured to be 0.033mW per sNeuron. The total output neuronal power is 236mW (0.558mW+0.033mW × 400) while that of the input neurons is 25.87mW (0.033mW × 784). The proposed implementation performs 23.52 TOPS (784×400×2×37.5MHz) while consuming 262.8mW, leading to energy efficiency of 89.49TOPS/Watt. The high energy efficiency can be attributed to binary dot product computations and the inherent sparsity in the neuronal spiking activity offered by SNNs. Figs. 9(a)-(b) show the die shot of the sNeuron, synapse, and the layout of the *'stochastic bit'* core (inset of Fig. 9(a)). For measurements, we interfaced an FPGA

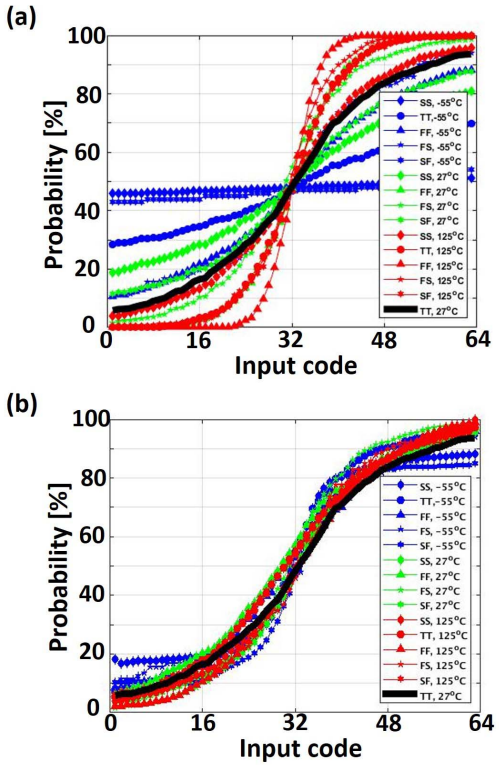


Fig. 10. (a) The switching probability curves with process (FF, TT, SS, FS, SF) and temperatures ($-55\ ^{\circ}C$, $27\ ^{\circ}C$, $125\ ^{\circ}C$) variations. (b) The compensated switching probability curves for all corners presented in (a).

to the QFN packaged chip on a custom PCB as depicted in Fig. 9(c).

E. Process and Temperature Variation

Fig. 10(a) shows the simulated switching probability curves affected by process and temperature variations. The black solid line represents the baseline of our design and the other lines represent variations caused by the different combinations of process corners (FF, TT, SS, FS, SF) and temperatures ($-55°C$, $27°C$, $125°C$). The (SS, $-55°C$) corner shows less than 10% change in probability due to decreased temperature and current, decreasing noise or the source of the randomness. The variations can be easily compensated by having variable size of M1 and M2 transistors of Fig. 2(b) in the same way we size M3 and M4 transistors. The size ratio between M1/M2 transistors and M_{l1x}/M_{r1x} transistors determines the unit step change of probability and thus, the slope of the probability curve. Fig. 10(b) shows the compensated switching probability curves from all corners presented in Fig. 10(a). In addition to the variation compensation, this approach also allows us to control the shape and slope of the probability curve at the cost of area required for sizing M1 and M2 transistors. Further, the probability range can also be controlled through the NMOS codes applied to M3 and M4 transistors as shown in Fig. 7(c).

V. Conclusion

We proposed *'stochastic bit'* enabled Binary SNN (sBSNN), composed of stochastic spiking neurons (sNeurons) and binary

synaptic weights, for energy- and memory-efficient neuromorphic computing at the edge. sBSNN computes probabilistically with only one-bit precision for both the constituting sNeurons and synapses, leading to on-chip visual image recognition with compressed memory requirement. We presented an energy-efficient implementation of two-layer fully-connected sBSNN using *'stochastic bit'* as the core computational primitive to realize the sNeurons and binary synapses (stochastic during training and deterministic during inference) fabricated in 90nm CMOS process. We demonstrated memory-efficient on-chip training of the binary synaptic weights using the stochastic-STDP (sSTDP) training algorithm. The proposed implementation, by virtue of sparse event-driven computing enabled by state-less sNeurons and binary weights, offered high energy-efficiency of 89.49 TOPS/Watt. We believe that sBSNN can provide a promising solution for building the next generation of intelligent devices capable of real-time on-chip learning.

REFERENCES

[1] P. Blouw, X. Choo, E. Hunsberger, and C. Eliasmith, "Benchmarking keyword spotting efficiency on neuromorphic hardware," 2018, *arXiv:1812.01739*. [Online]. Available: http://arxiv.org/abs/1812.01739

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

[3] A. Sengupta, Y. Ye, R. Wang, C. Liu, and K. Roy, "Going deeper in spiking neural networks: VGG and residual architectures," *Frontiers Neurosci.*, vol. 13, p. 95, Mar. 2019.

[4] C. Lee, S. Shakib Sarwar, and K. Roy, "Enabling spike-based backpropagation in state-of-the-art deep neural network architectures," 2019, *arXiv:1903.06379*. [Online]. Available: http://arxiv.org/abs/1903.06379

[5] 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.*, vol. 34, no. 10, pp. 1537–1557, Oct. 2015.

[6] K. Cheung, S. R. Schultz, and W. Luk, "NeuroFlow: A general purpose spiking neural network simulation platform using customizable processors," *Frontiers Neurosci.*, vol. 9, p. 516, Jan. 2016.

[7] B. Linares-Barranco and T. Serrano-Gotarredona, "Memristance can explain spike-time-dependent-plasticity in neural synapses," *Nature Precedings*, p. 1. Mar. 2009.

[8] V. Milo *et al.*, "Demonstration of hybrid CMOS/RRAM neural networks with spike time/rate-dependent plasticity," in *IEDM Tech. Dig.*, Dec. 2016, pp. 8–16.

[9] Y. Shi *et al.*, "Neuroinspired unsupervised learning and pruning with subquantum CBRAM arrays," *Nature Commun.*, vol. 9, no. 1, p. 5312, Dec. 2018.

[10] A. Sengupta, A. Banerjee, and K. Roy, "Hybrid spintronic-CMOS spiking neural network with on-chip learning: Devices, circuits, and systems," *Phys. Rev. Appl.*, vol. 6, no. 6, Dec. 2016, Art. no. 064003.

[11] G. Srinivasan, A. Sengupta, and K. Roy, "Magnetic tunnel junction based long-term short-term stochastic synapse for a spiking neural network with on-chip STDP learning," *Sci. Rep.*, vol. 6, no. 1, Sep. 2016, Art. no. 29545.

[12] S. Mutoh, T. Douseki, Y. Matsuya, T. Aoki, S. Shigematsu, and J. Yamada, "1-V power supply high-speed digital circuit technology with multithreshold-voltage CMOS," *IEEE J. Solid-State Circuits*, vol. 30, no. 8, pp. 847–854, Aug. 1995.

[13] M. Suri *et al.*, "CBRAM devices as binary synapses for low-power stochastic neuromorphic systems: Auditory (cochlea) and visual (retina) cognitive processing applications," in *IEDM Tech. Dig.*, Dec. 2012, pp. 3–10.

[14] A. Yousefzadeh, E. Stromatias, M. Soto, T. Serrano-Gotarredona, and B. Linares-Barranco, "On practical issues for stochastic STDP hardware with 1-bit synaptic weights," *Frontiers Neurosci.*, vol. 12, p. 665, Oct. 2018.

[15] G.-Q. Bi and M.-M. Poo, "Synaptic modifications in cultured hippocampal neurons: Dependence on spike timing, synaptic strength, and postsynaptic cell type," *J. Neurosci.*, vol. 18, no. 24, pp. 10464–10472, Dec. 1998.

[16] S. Lowel and W. Singer, "Selection of intrinsic horizontal connections in the visual cortex by correlated neuronal activity," *Science*, vol. 255, no. 5041, pp. 209–212, Jan. 1992.

[17] J. Holleman, S. Bridges, B. P. Otis, and C. Diorio, "A 3 μW CMOS true random number generator with adaptive floating-gate offset cancellation," *IEEE J. Solid-State Circuits*, vol. 43, no. 5, pp. 1324–1336, May 2008.

[18] C. S. Petrie and J. A. Connelly, "A noise-based IC random number generator for applications in cryptography," *IEEE Trans. Circuits Syst. I. Fundam. Theory Appl.*, vol. 47, no. 5, pp. 615–621, May 2000.

[19] S. K. Mathew *et al.*, "2.4 Gbps, 7 mW all-digital PVT-variation tolerant true random number generator for 45 nm CMOS high-performance microprocessors," *IEEE J. Solid-State Circuits*, vol. 47, no. 11, pp. 2807–2821, Nov. 2012.

[20] S. Ambrogio *et al.*, "Novel RRAM-enabled 1T1R synapse capable of low-power STDP via burst-mode communication and real-time unsupervised machine learning," in *Proc. IEEE Symp. VLSI Technol.*, Jun. 2016, pp. 1–2.

[21] S. Kim *et al.*, "NVM neuromorphic core with 64k-cell (256-by-256) phase change memory synaptic array with on-chip neuron circuits for continuous in-situ learning," in *IEDM Tech. Dig.*, Dec. 2015, pp. 1–17.

[22] M. Jerry, A. Parihar, B. Grisafe, A. Raychowdhury, and S. Datta, "Ultra-low power probabilistic IMT neurons for stochastic sampling machines," in *Proc. Symp. VLSI Technol.*, Jun. 2017, pp. T186–T187.

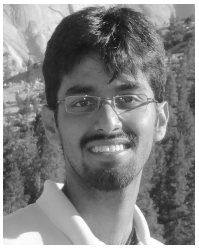
[23] X. Wu, V. Saxena, K. Zhu, and S. Balagopal, "A CMOS spiking neuron for brain-inspired neural networks with resistive synapses and in situ learning," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 62, no. 11, pp. 1088–1092, Jul. 2015.

[24] E. Neftci, S. Das, B. Pedroni, K. Kreutz-Delgado, and G. Cauwenberghs, "Event-driven contrastive divergence for spiking neuromorphic systems," *Frontiers Neurosci.*, vol. 7, p. 272, Jan. 2014.

[25] O. T.-C. Chen and R. R.-B. Sheen, "A power-efficient wide-range phase-locked loop," *IEEE J. Solid-State Circuits*, vol. 37, no. 1, pp. 51–62, Jan. 2002.

[26] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

Minsuk Koo (Member, IEEE) received the B.S. degree in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2007, and the M.S. degree in electrical engineering from Seoul National University, Seoul, South Korea, in 2009. He is currently pursuing the Ph.D. degree in electrical and computer engineering with Purdue University, West Lafayette, IN, USA. From 2009 to 2012, he was a Senior Engineer with RadioPulse Inc., Seoul, South Korea, where he was involved in the development of ZigBee transceiver and SoC products. His research interests include circuits and systems for neural networks and associative computing using CMOS and emerging devices.



Gopalakrishnan Srinivasan received the B.Tech. degree in electrical and electronics engineering from the National Institute of Technology, Calicut, India, in 2010, and the master's degree in computer engineering from North Carolina State University, Raleigh, NC, USA, in 2012. He is currently pursuing the Ph.D. degree in electrical engineering with Purdue University, under the guidance of Prof. K. Roy. His primary research interests include investigating brain-inspired spiking neural network architectures and training methodologies, and their energy-efficient implementation using CMOS and post-CMOS (spintronic) technologies.

Yong Shim (Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from Korea University, Seoul, South Korea, in 2004 and 2006, respectively, and the Ph.D. degree from Purdue University in August 2018. He currently works as a SRAM Circuit Designer with Intel Corporation, Hillsboro, OR, USA. In 2006, he joined Samsung Electronics Company, Ltd., Hwasung, South Korea, where he has been involved in designing circuits for memory interface. He also worked as a Graduate Research Intern with Circuit Research Labs, Intel Labs, in 2015. His research interests include the implementation of unconventional computing models such as neural networks, in-memory computing models, and optimization problem solvers based on the conventional CMOS circuits and emerging devices.

Kaushik Roy (Fellow, IEEE) received the B.Tech. degree in electronics and electrical communications engineering from IIT Kharagpur, India, and the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Illinois at Urbana–Champaign, Champaign, IL, USA, in 1990. He was with the Semiconductor Process and Design Center of Texas Instruments, Dallas, TX, USA, where he worked on FPGA architecture development and low-power circuit design. He joined the Faculty of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA, in 1993, where he is currently an Edward G. Tiedemann Jr. Distinguished Professor. He is also the Director of the Center for Brain-Inspired Computing funded by the SRC/DARPA. He has authored or coauthored more than 700 papers in refereed journals and conferences, and supervised 75 Ph.D. dissertations. He holds 18 patents, and he is the coauthor of two books on low power CMOS VLSI design (Wiley and McGraw Hill). His research interests include neuromorphic and emerging computing models, neuromimetic devices, spintronics, device-circuit-algorithm codesign for nanoscale silicon and non-silicon technologies, and low-power electronics. He was the recipient of the National Science Foundation Career Development Award in 1995, the IBM Faculty Partnership Award, the ATT/Lucent Foundation Award, the 2005 SRC Technical Excellence Award, the SRC Inventors Award, the Purdue College of Engineering Research Excellence Award, the Humboldt Research Award in 2010, the 2010 IEEE Circuits and Systems Society Technical Achievement Award (Charles Doeser Award), the Distinguished Alumnus Award from IIT Kharagpur, India, the Fulbright-Nehru Distinguished Chair, the DoD Vannevar Bush Faculty Fellow from 2014 to 2019, and the Semiconductor Research Corporation Aristotle Award in 2015, the best paper awards from the 1997 International Test Conference, the IEEE 2000 International Symposium on Quality of IC Design, the 2003 IEEE Latin American Test Workshop, the 2003 IEEE Nano, the 2004 IEEE International Conference on Computer Design, and the 2006 IEEE/ACM International Symposium on Low Power Electronics and Design, the 2005 IEEE Circuits and System Society Outstanding Young Author Award (Chris Kim), the 2006 IEEE Transactions on VLSI Systems Best Paper Award, the 2012 ACM/IEEE International Symposium on Low Power Electronics and Design Best Paper Award, and the 2013 IEEE TRANSACTIONS on VLSI Best Paper Award. He was a Faculty Scholar with Purdue University from 1998 to 2003. He was a Research Visionary Board Member of Motorola Labs in 2002 and held the M. Gandhi Distinguished Visiting Faculty with IIT Bombay, and the Global Foundries Visiting Chair with the National University of Singapore, Singapore. He has served on the Editorial Board of *IEEE Design and Test*, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS, the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, and the IEEE TRANSACTIONS ON ELECTRON DEVICES. He was a Guest Editor for the special issue on low-power VLSI for the IEEE DESIGN AND TEST in 1994, the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS in June 2000, the *IEE Proceedings—Computers and Digital Techniques* in July 2002, and the IEEE JOURNAL ON EMERGING AND SELECTED TOPICS IN CIRCUITS AND SYSTEMS in 2011.