This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2020.2965079, IEEE Transactions on Emerging Topics in Computing

Probabilistic Interpolation Recoder for Energy-Error-Product Efficient DBNs with p-bit Devices

Hossein Pourmeidani*, Shadi Sheikhfaal*, Ramtin Zand†, and Ronald F. DeMara*
*Department of Electrical and Computer Engineering
University of Central Florida, Orlando, FL 32816
†Department of Computer Science and Engineering
University of South Carolina, Columbia, SC 29208

***Abstract*—In this paper, a probabilistic interpolation recoder (PIR) circuit is developed for deep belief networks (DBNs) with probabilistic spin logic (p-bit)-based neurons. To verify the functionality and evaluate the performance of the PIRs, we have implemented a $784 \times 200 \times 10$ DBN circuit in SPICE for a pattern recognition application using the MNIST dataset. The PIR circuits are leveraged in the last hidden layer to interpolate the probabilistic output of the neurons, which are representing different output classes, through sampling the p-bit's output values and then counting them in a defined sampling time window. The PIR circuit is proposed as an alternative for conventional interpolation methods which were based on using a resistor-capacitor tank to integrate each neuron's output, followed by an analog-to-digital converter to generate the digital output. The circuit simulation results of PIR circuit exhibit at least 54%, 81%, and 78% reductions in power, energy, and energy-error-product, respectively, compared to previous techniques, without using any of the area-consuming analog components in the interpolation circuit. In addition, PIR circuits provide an inherent single stuck-at fault tolerant feature to mitigate both transient and permanent faults at the circuit's output. Reliability properties of the PIR circuits for single stuck-at faults are shown to be enhanced relative to conventional interpolation without requiring hardware redundancy.**

***Index Terms*—Deep Belief Network (DBN), magnetic tunnel junction (MTJ), probabilistic spin logic device (p-bit), analog-to-digital converter (ADC), MRAM.**

I. Introduction and Related Work

The Restricted Boltzmann machine (RBM) is one of the well-known classes of unsupervised learning approach [1]. A set of RBMs connected hierarchically can be utilized to create deep belief networks (DBNs) with outstanding learning abilities such as natural language understanding for various applications [2]. Most of the research on RBM and DBN has focused on software implementations. Albeit the software implementation of DBNs on current von-Neumann-based platforms (e.g. CPU, GPU, FPGA) provides flexibility, it incurs significant power dissipation and high latency due to inherent data communication costs, a.k.a. the "memory wall" issue. There are various hardware implementations for RBMs such as FPGAs [3] and CMOS multi-core processors [4] aiming to tackle existing software limitations.

Recently, processing-in-memory based solutions using emerging non-volatile memories (NVMs) such as resistive RAM (RRAM) [5] and phase change memory (PCM) [6] are set forth to be used within the DBN architecture. NVMs provide the capability of performing logic beyond data storage by bringing an intrinsic computation parallelism alleviating the data transfer bottleneck. NVMs are typically used as weighted connections interconnecting building blocks in RBMs.

The existing FPGA-based acceleration solutions show 25-145$\times$ speedup compared to software implementations [3]. However, these designs have noticeable limitations such as constrained clock frequencies, routing congestion, and resource deficiencies due to the significant embedded memory utilization for weighted connections and activation functions. In [7] optimization methods to reduce memory requirements for weights and biases are proposed. However, in order to implement each of the activation functions, a random number generator (RNG), dedicated piecewise linear approximator (PLA), and comparators are still required which increases area and energy consumption per neuron. As an alternative method, the stochastic CMOS-based RBM implementation have been set forth [8] that takes full advantage of low-complexity of the stochastic CMOS designs to improve area- and energy-efficiency. On the other hand, such implementation seeks extremely-long bit-stream that could lead to more energy consumption and longer latencies. Besides, it requires a significant amount of Linear Feedback Shift Registers (LFSRs) to generate the uncorrelated input and weight bit-streams. Both the FPGA and stochastic CMOS implementations leverage parallel Boolean circuits such as pseudo-random number generators, adder, and multipliers to improve the performance. Such designs impose significant area and energy overheads compared with leveraging the physical behaviors of emerging devices to perform the computation intrinsically.

Within the NVM domain, Bojnordi et al. [5] proposed to leverage resistive RAM (RRAM) devices to implement vector-matrix multiplication with up to 100$\times$ speedup and 10$\times$ energy savings over single-threaded cores. In the same way, Eryilmaz et al. [6] has used resistive memories with CMOS activation function that ultimately imposes excessive area and power consumption overheads. Recently, spintronic devices with low energy barrier nanomagnets such as spin orbit torque-Magnetic Tunnel Junctions (SOT-MTJs) and embedded magnetoresistive random access memory (MRAM) devices are leveraged as a natural building block to provide probabilistic sigmoidal activation functions for RBMs, as studied in [9] and [10], respectively. These devices have realized significant energy and area improvements compared to previous RBM hardware implementations. Thus, we will investigate various circuit implementations to interpolate the stochastic output

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2020.2965079, IEEE Transactions on Emerging Topics in Computing

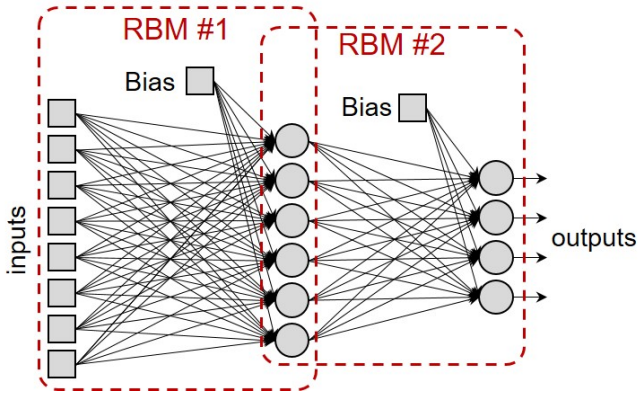


Fig. 1: An example of DBN structure including a visible layer and two hidden layers.

of the probabilistic spin logic devices (p-bit) proposed in [11]. In particular, inspired by a technique that is used to create an analog-to-digital converter [12], we will develop two CMOS-based probabilistic interpolation recoder (PIR) circuits, which leverage a sampling methodology to provide a digital output corresponding to the probabilistic output of the p-bit based neurons. The proposed circuits achieve significant improvements in terms of resource utilization and energy consumption compared to conventional integration followed by analog-to-digital conversion methods.

II. Background

A. Deep Belief Network (DBN)

DBN can be easily realized by stacking Restricted Boltzmann machines (RBMs), which are classes of recurrent stochastic neural networks, in which state of the network, k, has an energy expressed by (1), determined by the connection weights between nodes and the node bias, where s_i^k denotes the state of node i in k, b_i represents the bias, or intrinsic excitability of node i, and w_{ij} is the weight of connection between nodes i and j [13].

$$E(k) = -\sum_i s_i^k b_i - \sum_{i<j} s_i^k s_j^k w_{ij} \tag{1}$$

The probability of each node in a RBM to be in state one is determined based on (2), where σ denotes the sigmoid function. RBMs can reach a Boltzmann distribution in which the system probability to be in state $\boldsymbol{v}$ is represented by (3), and $\boldsymbol{u}$ could be any possible system state. Therefore, given sufficient time, the system moves towards the states with the lowest associated energy.

$$P(s_i = 1) = \sigma(b_i + \sum_j w_{ij} s_j) \tag{2}$$

$$P(v) = \frac{e^{-E(v)}}{\sum_u e^{-E(u)}} \tag{3}$$

RBM consists of two fully-connected layers called the *visible layer* and the *hidden layer*, as shown in Fig.1. Crossbar architecture is a widely-explored method to implement such networks.

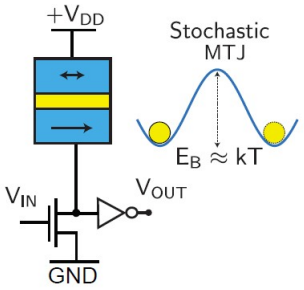


Fig. 2: The diagram of the embedded MRAM-based neuron (p-bit) as a building block for RBMs [11].

B. Embedded MRAM-Based Neuron

In this subsection, we show how a recently-proposed building block based on embedded MRAM technology can realize a neuron with probabilistic sigmoidal activation function [11]. The MRAM-based stochastic device (p-bit) structure is shown in Fig. 2. It consists of a magnetic tunnel junction (MTJ), which is a 2-terminal device with two possible resistive levels based on the orientation of its ferromagnetic (FM) layers, i.e. *fixed layer* and *free layer*. The fixed layer has a fixed magnetic orientation, while the free layer's magnetization orientation can be switched. In conventional MRAM cells, free layer of the MTJ is manufactured with a thermally-stable nanomagnet with a large energy barrier with respect to the thermal energy (kT). Accordingly, the fixed layer works as a non-volatile storage. Recently, in search of functional spintronic paradigms, thermally-unstable MTJs based on superparamagnetic materials have been theoretically and experimentally explored [14], [9], [15], [16], [17].

In this work, we use a thermally-unstable MRAM device with a low energy-barrier nanomagnet ($E_B \ll 40kT$) [11]. The MTJ resistance of this device randomly fluctuates between the two possible resistive states. This leads to a fluctuating output voltage at the drain of the NMOS transistor connected to a CMOS inverter. The inverter amplify such voltage deviation from the threshold voltage and generate a stochastic output modulated by the input voltage. Particularly, by reducing the drain-source resistance (r_{ds}) through increasing the input voltage (V_{IN}), the voltage at the drain of the NMOS transistor is shorted to the ground. Alternatively, it can get to V_{DD} by increasing the r_{ds} through decreasing V_{IN}. Such device operation is formulated considering the MTJ conductance [11]:

$$G_{MTJ} = G_0 \left[1 + m_z \frac{TMR}{(2 + TMR)}\right] \tag{4}$$

where m_z is the free layer magnetization, G_0 denotes the average MTJ conductance, $(G_P + G_{AP})/2$, and TMR represents the tunneling magnetoresistance ratio. The drain voltage can be written as:

$$V_{DRAIN}/V_{DD} = \frac{(2 + TMR) + TMR\ m_z}{(2 + TMR)(1 + \alpha) + TMR\ m_z} \tag{5}$$

where α is the ratio of the transistor conductance (G_T) to the average MTJ conductance (G_0).

The p-bit device uses a circular nanomagnet with near-zero energy barrier without shape anisotropy. The free layer

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2020.2965079, IEEE Transactions on Emerging Topics in Computing

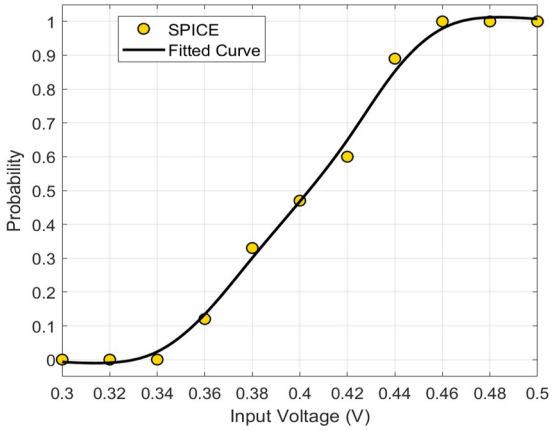


Fig. 3: Output probability of MRAM-based neuron vs. its input voltage.

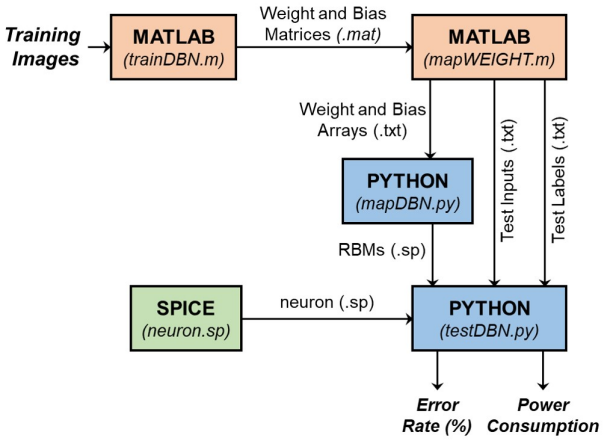


Fig. 4: The block diagram of PIN-Sim framework including five main modules [10].

magnetization for the MTJ conductance discussed in Equation 4 is given by the stochastic Landau-Lifshitz-Gilbert (LLG) equation:

$$(1+\alpha^2)\frac{d\hat{m}}{dt} = -|\gamma|\hat{m}\times\vec{H} - \alpha|\gamma|(\hat{m}\times\hat{m}\times\vec{H}) + 1/qN(\hat{m}\times\vec{I}_S\times\hat{m}) + \Big(\alpha/qN(\hat{m}\times\vec{I}_S)\Big) \quad (6)$$

where α is the damping coefficient of the nanomagnet, γ is the electron gyromagnetic ratio, q denotes the electron charge, and $\vec{I}_S$ is the spin current incident to the free layer. Fig. 3 shows the correlation between the probability of output being in state "1" and V_{IN}. A close observation shows that $V_{IN} = \frac{V_{DD}}{2} = 400\text{mV}$ produces an output probability of 50%.

Some of the most recent hardware implementations of DBNs are listed in Table I. In [3], FPGAs are utilized to achieve speedups of 25-145 in comparison with software implementations, however they still suffer from constrained clock frequencies and routing congestion along with substantial resource deficiencies because of the significant embedded memory utilization for both weighted connections and activation functions. The design presented in [8], benefits the low-complexity characteristics of stochastic CMOS-based arithmetic for implementing RBMs with reduced area and power consumption but the increased latencies in this design significantly restrains the energy savings due to the enormous number of linear feedback shift registers (LFSRs) that are required to generate the long input and weight bit-streams. In [5] and [6], the crossbar arrays have been employed with emerging technologies such as resistive RAM (RRAM) and phase change memory (PCM) to implement matrix multiplication within RBMs. In [5], Bojnordi et al. have employed RRAM devices as weighted connections to achieve 100-fold and 10-fold improvement with respect to operation speed and energy consumption, respectively, relative to single-threaded cores. The CMOS-based circuits such as multipliers and RNGs are employed in all the aforementioned designs to realize the probabilistic behavior of activation functions, which results in significant area and energy overheads. In [9], Zand et al. have achieved substantial area and energy reductions by employing low energy barrier spin-orbit torque (SOT) MTJs to implement the probabilistic sigmoidal activation function. Nevertheless, this design requires weighted connections with very large resistance values which results in considerable area overhead and fabrication complexity. Moreover, the current-mode behavior of the SOT-MTJ devices imposes considerable power consumption to the activation functions. Voltage-driven embedded MRAM-based neuron with low energy barrier (p-bit) has been proposed to take advantage of intrinsic thermal noise to generate sigmoidal probabilistic activation functions required for RBMs [10]. As listed in Table I, the p-bit based RBM implementation can attain approximately three orders of magnitude energy reduction relative to the previous energy-efficient CMOS-based implementations, as well as at least 90-fold decrease in the CMOS device count.

C. Probabilistic Inference Network-Simulator (PIN-Sim)

Herein, we use the Probabilistic Inference Network-Simulator (PIN-Sim) proposed in [10] to realize a circuit-level implementation of DBNs using memristive crossbars as weighted connections and embedded MRAM-based neurons as activation functions. As shown in Fig. 4, PIN-Sim is a hierarchical simulation framework that consists of five main modules: *(1) trainDBN:* a MATLAB-based module used for training various DBN topologies [18] *(2) mapWeight:* a module developed in MATLAB that converts the trained weights and biases to their corresponding resistance values, *(3) mapDBN:* a python-based module which provides a circuit-level implementation of the restricted Boltzmann machine using the obtained weight and bias resistances, *(4) neuron:* a SPICE model of the MRAM-based stochastic neuron [11], *(5) testDBN:* the main module developed in Python that executes test evaluations to assess the error rate and power consumption using the other modules in PIN-Sim.

III. Proposed Probabilistic Interpolation Recoder

In this paper, we use a $784 \times 200 \times 10$ DBN for MNIST pattern recognition tasks. Fig. 5 indicates the output voltages of the neurons for a sample digit of "4" in the last hidden layer whereas each neuron represents an output class. Fig.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2020.2965079, IEEE Transactions on Emerging Topics in Computing

TABLE I: Various hardware implementations for DBN architecture.

Design	[3]	[8]	[5]	[6]	[9]	[10]
Weighted Connection	Embedded multipliers	- LFSR - AND/OR gates	RRAM	PCM	SOT-DWM	Memristive Devices
Activation Function	-2-kB BRAM - PLA - RNG	- LFSR - Bit-wise AND - tree adder - FSM-based $tanh$	- 64×16 LUTs - Pseudo Random Number Generator - Comparator	Off-chip	near-zero energy barrier SOT-MTJ	Embedded MRAM-based stochastic neuron
Energy per neuron	$\sim 10-100nJ$	$\sim 10-100nJ$	$\sim 1-10nJ$	N/A	$\sim 1-10fJ$	$\sim 10-30fJ$
Normalized area per neuron	$\sim 3000\times$	$\sim 90\times$	$\sim 1250\times$	N/A	$\sim 1.25\times$	$1\times$

5(a) shows the probabilistic outputs of the p-bit devices while the outputs of their corresponding integrator circuits is demonstrated in Fig. 5(b). The outputs of the integrators are connected to the proposed PIR circuits described in this section to interpolate the probabilistic outputs of the neurons representing each class in the MNIST dataset.

A. Sample and Count based PIR (SC-PIR)

Conventional methods for designing an interpolation circuit for probabilistic neurons involve using an integrator circuit, e.g. resistor-capacitor (RC) circuit, along with an analog-digital-converter (ADC) to convert the probabilistic outputs of the neurons to a digital output, as shown in Fig. 6a. Interpolation circuits such as ADCs, which are required for a completely operational network, are being investigated as an emerging topic in computing. These are identified as useful targets to further reduce energy and area demands [19], [20], [21], [22], [23]. For instance in [20], significant reduction of the ADC energy and area overhead is achieved by using bit-slice sparsity since the power-hungry ADCs prevent the practical deployment of Resistive Random-Access Memory (ReRAM)-based DNN accelerators on end devices with limited chip area and power budget. In [21], they painstakingly attempted to reduce the overhead of ADCs by partitioning the input into several segments which are fed sequentially into the crossbar. An alternate technique is presented in [22] to reduce the overhead of ADCs in ReRAM neuromorphic computing systems by normalizing and quantizing data. In [23], it is an explicit focus to considerably decrease the overhead of the peripheral circuit to reduce the total design area and power consumption by quantizing the weights to fewer bits. Herein, we propose a CMOS-based probabilistic interpolation recoder (PIR), which is directly connected to the p-bits to generate a discrete n-bit output for each of the neurons in the last layer of the network. Fig. 6b shows the circuit structure of 3-bit SC-PIRs.

In the proposed SC-PIR circuits, the probabilistic output of the embedded MRAM-based neuron ($Neuron_{OUT}$) is sampled at the positive edge of each clock (clk), and the sampled outputs are accumulated through a counter. A $ctrl$ signal is utilized to reset the counter and control the PIR circuit's sampling time window. An n-bit PIR circuit counts the sampled outputs for $2n-1$ clocks and then returns the accumulated value in the form of an n-bit output ($OUT_{n-1}-OUT_0$). Fig. 7a exhibits the transient response of the proposed 3-bit SC-PIR circuits, while the input of the p-bit based neuron is set to $V_{IN} = \frac{V_{DD}}{2} = 400mV$. When the $ctrl$ signal is "0", the counter is reset and the output of the PIR circuit will be connected to GND, i.e. ($OUT_2 - OUT_0 = 000$). When the $ctrl = 1$, the counter is activated, and the output of the neuron is sampled at every positive edge of the clock signal. If the output of the RC integrator circuit connected to the neuron is greater than $V_{DD}/2$ during the sampling time, the PIR circuit will increment the counter, else the counter remains unchanged. For instance, in Fig. 7a, the counter is incremented from 000 to 001 at the fourth positive edge of the clock since ctrl signal is equal to "1" and the voltage of the $Neuron_{OUT}$ is greater than $V_{DD}/2 = 400mV$. An n-bit SC-PIR circuit continues this process for 2^n clock periods and after the 2^n-th period, the output of the counter is used as the interpolated output of the probabilistic neuron.

B. Sample and Shift based PIR (SS-PIR)

In this paper, we develop another alternative implementation of PIR circuits that is called sample and shift based PIR (SS-PIR) in the interest of improving energy consumption while obtaining a comparable error rate. In the proposed SS-PIR circuit, the sampled outputs are interpolated through a bidirectional shift register at the positive edge of clock (clk). The SS-PIR circuit shifts by one position the bit array stored in it, shifting in $Neuron_{OUT}$ and shifting out the last bit in the array at each transition of the clock input. The shift register in the SS-PIR circuit must be shifted right or left if the sampled output voltage of the neurons integrator ($Neuron_{OUT}$) is less than or greater than $V_{DD}/2$, respectively. In other words, the bit array that is stored in shift register multiplies or divides by 2 if $Neuron_{OUT}$ is less than or greater than $V_{DD}/2$, respectively. A $ctrl$ signal is utilized to reset the shift register and control the SS-PIR circuit's sampling time window. An n-bit SS-PIR circuit counts the sampled outputs for n clock periods and then returns the shifted value in the form of an n-bit output ($OUT_{n-1}-OUT_0$). Fig. 7b exhibits the transient response of the proposed 3-bit SS-PIR circuits while the input of the p-bit based neuron is set to $V_{IN} = \frac{V_{DD}}{2} = 400mV$. For instance, as shown in the figure, when the $ctrl$ signal is "1", the value stored in the shift register changes from 000 to 001 at the third positive edge, and from 001 to 011 at the fourth positive edge of the clock since $Neuron_{OUT} > (\frac{V_{DD}}{2} = 0.4V)$.

IV. PIR for Spiking Neural Networks

With some minor changes in the PIR circuit design, they can be utilized in the Spiking Neural Network (SNN) architectures as well. There are various implementations of spiking neurons,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2020.2965079, IEEE Transactions on Emerging Topics in Computing

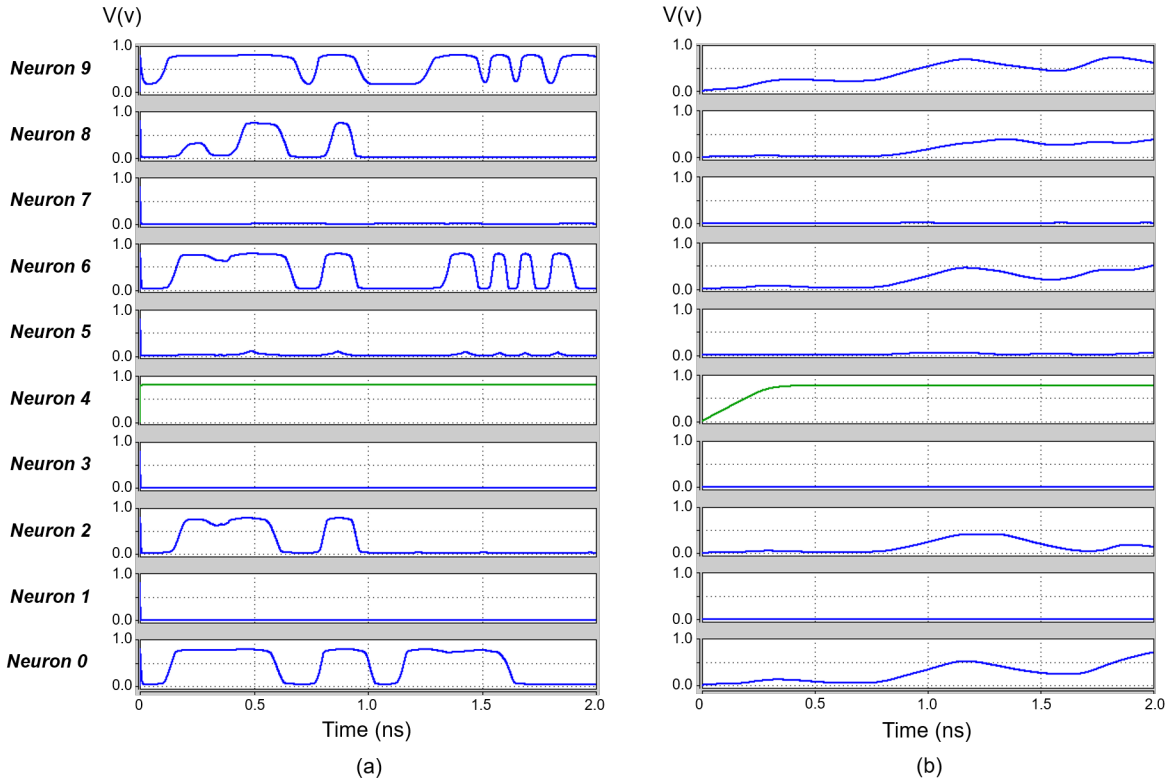


Fig. 5: Output voltages of a $784 \times 200 \times 10$ DBN for a sample digit of "4": (a) Probabilistic output of the p-bit devices, (b) Output of the integrator circuit [10].

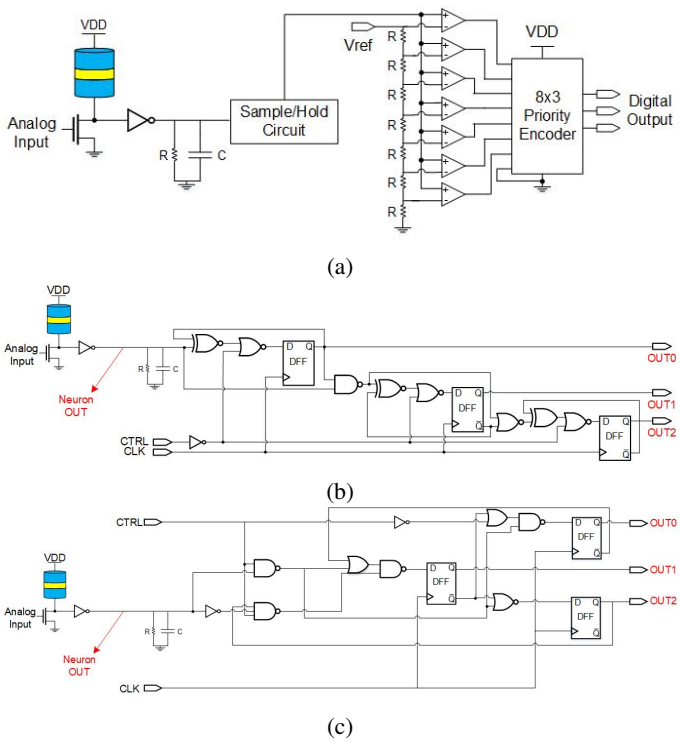
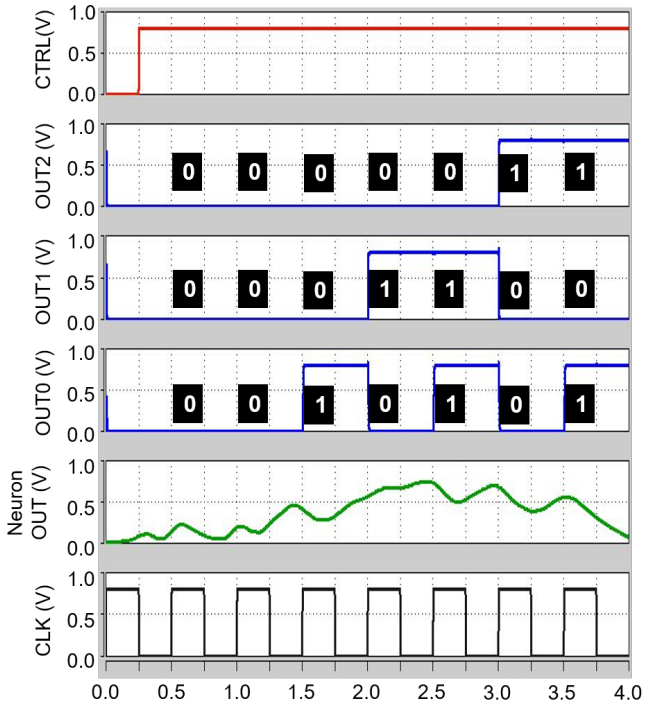


Fig. 6: (a) 3-bit ADC circuit, (b) 3-bit SC-PIR circuit, and (c) 3-bit SS-PIR circuit.

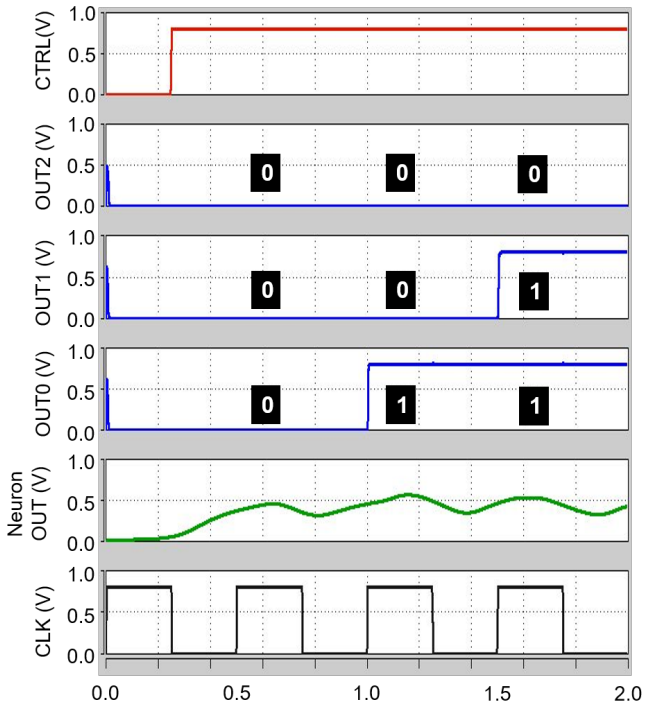
whereby some require a compatible counting and sampling while others do not utilize such techniques. In Seo et al. [24], each SNNs neuron circuit has its own 16-bit adder, Op-amp comparator, and a 4:1 mux to integrate all presynaptic weights and determine firing activity, which essentially imposes a large power overhead to the design. In [25], the neurons readout block includes a column ADC, which contains a summing amplifier, a sample-and-hold circuit and a high-resolution ADC, which again shows a large area-overhead. Wang et al. [26] exploit a capacitive accumulator and then a comparator as well as a flip-flop to readout the data from a SNN-based RRAM crossbar. On the other hand, in a recent work [27], the authors present an efficient three-step memristive-based SNNs neuron; or reference [28] presents an all-spin SNNs by using a domain wall-based neuron, where neither of these designs need adder/comparator-based techniques.

The proposed sequential PIR circuit can be modified to a combinational circuit which instead of sampling the output of the neuron at the positive edges of the clock, would increment the counter or shifts the shift register in SC-PIR and SS-PIR circuits, respectively. This occurs when the input voltage of the circuit (i.e. output of the neuron in SNN) is greater than a specific voltage threshold. However, the proposal of our PIR circuit is particularly important for DBNs whereas listed in Table 1, the p-bit based neurons achieve orders of magnitude energy and area reduction compared to their CMOS-based counterpart, but they require an efficient interpolation circuit to fully-leverage their advantages. Thus, in this paper, we have

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2020.2965079, IEEE Transactions on Emerging Topics in Computing



(a)



(b)

Fig. 7: Timing waveforms of (a) 3-bit SC-PIR circuit and (b) 3-bit SS-PIR circuit.

TABLE II: Parameters used for modeling and simulation [11].

Parameters	Value
Saturation magnetization (CoFeB) (M_s)	1100 emu/cc [31]
Free Layer diameter, thickness	22nm, 2nm
Polarization	0.59 [32]
TMR	110% [32]
MTJ RA-product	$9\Omega - \mu m^2$ [32]
Damping coefficient	0.01 [31]
Temperature	26.85°C

focused on developing energy and area efficient interpolation circuits for DBN architectures.

V. Simulation Results

In order to assess the performance of the proposed PIR circuits, we have utilized them within the structure of a $784 \times 200 \times 10$ DBN circuit implemented by the PIN-Sim framework for MNIST digit recognition application. As shown in Fig. 8, the PIR circuits are connected to the output layer to interpolate the probabilistic output of the neurons which represent the 0-9 digit classes of the MNIST dataset. Moreover, we employ a circular disk magnet that have been fabricated and characterized in [29], [30], and [17] with near-zero energy barrier without any shape anisotropy. Table II shows the device parameters that are used in the simulations in this paper [11]. It should be emphasized that the results are not considerably influenced by the current that is flowing at the midpoint ($V_{IN} = V_{DD}/2$) for the selected parameters with a circular free layer with an in-plane anisotropy, and any pinning at higher input voltages takes advantage of switching operation of the device. By verifying the functionality and efficiency of the PIR circuits for MNIST dataset, their efficiency for larger datasets will be validated as well. This is because the PIR circuits are only used to interpolate the probabilistic output of the last layer in the network, while the accuracy of the network for various datasets rely on other factors such as number of hidden layers and number of nodes in each hidden layer which is not the focus of this work. Thus, once it is shown that PIR circuits can properly interpolate the output of the network for MNIST dataset, it is also verified that they can interpolate the outputs of different DBN topologies for different datasets.

A. Accuracy Analyses

Herein, 100 images from MNIST dataset are selected, which induced the most discrepancy in recognition accuracy when classified using ADC and PIR circuits. Output classes are selected according to the binary values given to them by the ADC-based or PIR-based interpolation circuits. For instance, Table III exhibits the binary values generated for each output classes in the $784 \times 200 \times 10$ DBN for a sample digit "2" from the selected images of the MNIST dataset. The output class(es) with the largest binary value represents the first class(es) selected by the interpolation circuit. As listed in Table III, the 3-bit and 5-bit SC-PIR circuits produced similar output binary values for digit classes 2,3 and 3 respectively as its top selections, which is an incorrect recognition, while other circuits successfully selected the correct output class.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2020.2965079, IEEE Transactions on Emerging Topics in Computing

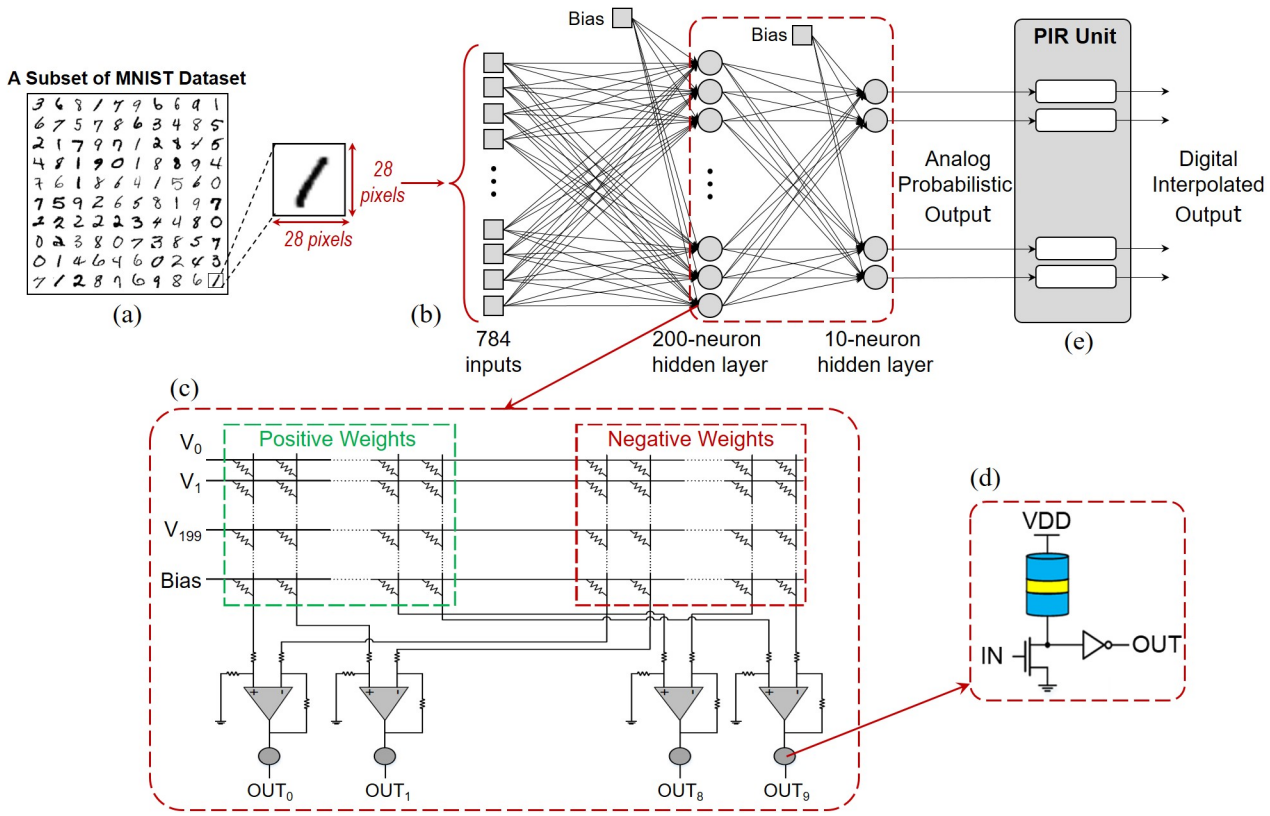


Fig. 8: Simulation framework utilized for application-level simulations. (a) subset of MNIST dataset with 100 test images, (b) a $784 \times 200 \times 10$ DBN developed for MNIST pattern recognition application, (c) hardware implementation of the $784 \times 200 \times 10$ DBN using PIN-Sim tool, (d) stochastic MRAM-based neuron (p-bit), and (e) PIR unit used to interpolate the probabilistic output of the p-bit based output neurons to digital output.

Table IV provides a recognition accuracy comparison between DBN circuits with 3-bit ADC and DBNs with 3-bit, 4-bit and 5-bit PIR in their structure. As listed, the 3-bit PIR circuits could obtain a comparable error rate with 3-bit ADC circuit, which led to a top-2 error rate of 0.23 and 0.24 for SC-PIR and SS-PIR respectively. This is mainly due to the low number of samples in the sampling time window for the 3-bit PIR circuits, i.e. only 7 and 3 samples for SC-PIR and SS-PIR respectively, which results in giving the same value to different classes. On the other hand, 4-bit SC-PIR and 5-bit SS-PIR circuits could achieve better error rate than 3-bit ADC circuit as shown in Fig. 9. It is worth emphasizing that the network topology, weights, and neurons in each of these DBN implementations are similar, even a similar random seed is utilized in the SPICE simulations to generate the probabilistic behavior of the p-bit based neurons, thus the discrepancy in the recognition accuracy is only induced by the difference in the interpolation circuits and no other factors are involved.

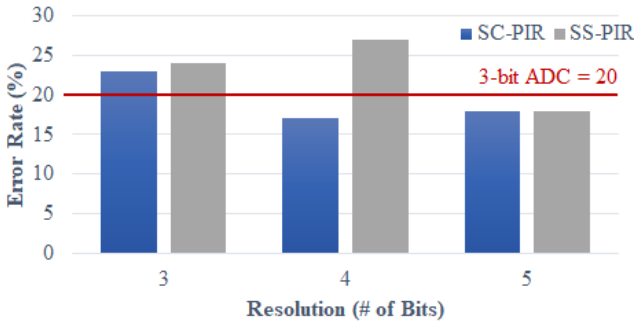


Fig. 9: Error Rate for 3-bit, 4-bit and 5-bit SC-PIR and SS-PIR.

B. Performance Analyses

In this work, the authors expected an increase in the error rate by replacing the ADCs with PIR circuits since a continuous integration operation followed by a sample-and-hold operation, and analog-to-digital conversion is replaced by a simple sample and accumulation method that is implemented only by CMOS transistors. Thus, to better comprehend the advantages of our proposed circuits, we have defined a metric called *energy-error-product (EEP)* as follows, which incorporates the energy costs to achieve a particular accuracy:

$$EEP = N \times E \times err \tag{7}$$

where N is the number of output neurons, E is the energy consumption of the PIR circuit, and err is the error rate of the network.

Table IV provides a comparison between the 3-bit ADC and 3-bit, 4-bit and 5-bit PIR circuits in terms of resource utilization, power/energy consumption, and EEP values. A

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2020.2965079, IEEE Transactions on Emerging Topics in Computing

TABLE III: The binary outputs generated by ADC-based and PIR-based interpolation circuits for an input digit "2" from the MNIST dataset of handwritten digits.

Output Class	3-bit ADC	3-bit SC-PIR	3-bit SS-PIR	4-bit SC-PIR	4-bit SS-PIR	5-bit SC-PIR	5-bit SS-PIR
Digit-0	001	100	001	0011	0000	10001	00000
Digit-1	001	000	000	0001	0000	00111	00000
Digit-2	110	111	111	1110	1111	11110	11111
Digit-3	010	111	011	0110	0011	11111	00000
Digit-4	001	001	000	0001	0000	01000	00000
Digit-5	000	000	000	0010	0000	00010	00000
Digit-6	000	001	000	0011	0000	01011	00000
Digit-7	000	000	000	0000	0000	00000	00000
Digit-8	001	100	000	0111	0000	10101	00000
Digit-9	000	000	000	0010	0000	00010	00000
1st Selected Digit Class	2	2,3	2	2	2	3	2
2nd Selected Digit Class	3	0,8	3	8	3	2	-
3rd Selected Digit Class	0,1,4,8	4,6	0	3	-	8	-

TABLE IV: Performance comparison between 3-bit, 4-bit and 5-bit SC-PIR and SS-PIR circuits.

Design		3-bit ADC	3-bit SC-PIR	3-bit SS-PIR	4-bit SC-PIR	4-bit SS-PIR	5-bit SC-PIR	5-bit SS-PIR
Resource Utilization	OP-AMP	9	-	-	-	-	-	-
	Capacitor	2	1	1	1	1	1	1
	Resistor	22	1	1	1	1	1	1
	Transistor	94	114	90	156	128	208	152
Required Number of clocks		-	8	4	16	5	32	6
Error Rate		0.20	0.23	0.24	0.17	0.27	0.18	0.18
Power Consumption (μW)		70.3	39.2	32	38.4	43.3	42.6	39.5
Energy Consumption (fJ)		351.5	156.8	64	307.2	108.25	681.6	118.5
Energy-Error-Product		702.6	360.6	153.6	522.2	292.2	1226.8	213.3

TABLE V: Power and energy consumption of weighted array, activation function and interpolation circuits for several DBN topologies.

Topology	Power Consumption (mW)				Energy Consumption (pJ)			
	Weighted Array	Activation Function	Interpolation Circuits		Weighted Array	Activation Function	Interpolation Circuits	
			3-bit ADC	3-bit SS-PIR			3-bit ADC	3-bit SS-PIR
784×10	4.146	0.194	0.703	0.32	8.292	0.388	3.515	0.64
$784 \times 200 \times 10$	80.4	5.6	0.703	0.32	321.6	22.4	3.515	0.64
$784 \times 200 \times 200 \times 10$	117.57	10.5	0.703	0.32	705.42	63	3.515	0.64

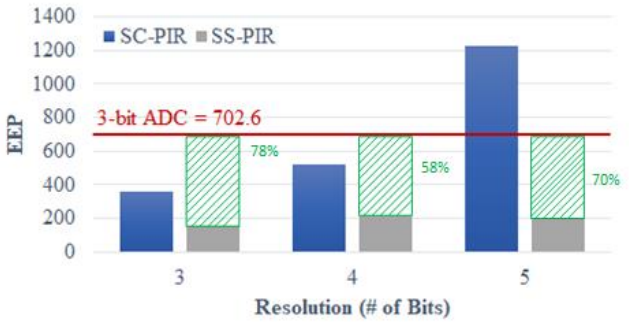


Fig. 10: EEP for 3-bit, 4-bit and 5-bit SC-PIR and SS-PIR.

comparison between 3-bit ADC and PIR circuits shows a significant improvement in the effectiveness of resource utilization. In the PIR circuits, all of the area-consuming elements in the conventional circuits such as operational amplifiers (op-amps), resistors, and capacitors are removed and for example, only 58 MOS transistors are increased for 5-bit SS-PIR compared to the 3-bit ADC circuit. Moreover, more than 54% and 81% reductions in power and energy are achieved, respectively, whereas EEP reduction is 78% for 3-bit SS-PIR circuit compared to 3-bit ADC as shown in Fig. 10. The results obtained verify the advantage of our proposed circuit in terms of the individual and combined metrics of accuracy and energy consumption.

Table V lists the power and energy consumption of the weighted array, activation function, and interpolation circuits for several DBN topologies. In smaller networks, such as the 784×10 DBN, energy consumption of the ADC-based interpolation circuit is approximately 9-fold greater than the energy that is consumed in the activation functions, while it constitutes almost 28% of the total energy consumption of the entire network. On the other hand, the proposed SS-PIR circuit achieves more than 5-fold energy consumption reduction compared to ADC-based circuit, which significantly reduces the contribution of the interpolation circuit to the total energy consumption of the network from 28% to only 6%. By enlarging the size of the network, the activation function and interpolation circuit will be minority sources of energy consumption, which is partially realized by the considerable

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2020.2965079, IEEE Transactions on Emerging Topics in Computing

energy reductions achieved by utilizing the p-bit devices as neurons and proposed PIRs as interpolation circuits.

C. Area Analysis

One of the major challenges of ADC circuits are their significant area consumption, which is mainly induced by the large analog components existing in their structure such as Op-Amps. Herein, we have used a Flash ADC, which uses a linear voltage ladder with Op-Amp based comparators and an encoder circuit to interpolate the probabilistic output of the circuit and compared its energy and area consumption with our proposed PIR circuits. For the Op-Amp circuits we have used the CMOS-based design proposed in [33], which reports an area consumption of approximately $250\mu m^2$ for $130nm$ CMOS technology, scaling it down to $14nm$ nodes using the scaling method proposed in [34] results in an approximate area consumption of 2.9 μm^2 for each Op-Amps utilized in the ADC circuits. On the other hand, the layout design results of MRAM-based neuron demonstrate that the area consumption of the MRAM-based neuron is approximately equal to $32\lambda \times 32\lambda$, where $\lambda = 14nm/2 = 7nm$ for $14nm$ FinFET technology, thus leading to the approximate area consumption of $0.05\mu m^2$ per neuron [10].

Herein we have used the area consumption of the p-bit neuron as the baseline and all the other estimated area values are normalized according to the p-bit area consumption. For instance, the area required to implement the RC circuit with 100 $K\Omega$ resistor and $20fF$ capacitor is almost three times larger than that of the p-bit [10], i.e. RC_{Area}=3X, i.e. 3×(p-bit neuron area). On the other hand, we have used the well-known 1T-1R structure for each weight in the weighted array, which allocates one transistor to each weight and the resistive devices are fabricated on top of the MOS transistors thus incurring no area overhead. Therefore, the estimated area consumption for each weight is approximately 0.02 μm^2 =0.4X. Table VI provides the normalized area consumptions for weighted arrays, activation functions, and interpolation circuits for various network topologies. As it is listed in table, the area consumption of the activation function and interpolation circuits constitute a significantly smaller portion of the entire networks area, when the DBNs become larger which is in part realized by significant area reductions achieved by p-bit devices and PIR circuits.

D. Fault Analysis

High performance integrated circuits must be protected against either transient or permanent faults. The most commonly fault model is the single stuck-at fault, in which faults are modeled in a way that only one circuit node is permanently connected to either 0 (stuck-at 0) or 1 (stuck-at 1). When a node is stuck-at 0 or 1, the value is still readable, but can not be altered. In a write operation, the stuck-at node is faulty if the desired value is not equal to the stuck-at value but if the two values are equal, the node is not faulty. In order to do a fault simulation, it is necessary to execute two simulations: one for the fault-free circuit and another for the faulty circuit with some faults. In this way, when using the single stuck-at model, the fault injection includes a node that is permanently set either to 0 or 1. By comparing the output of the two simulations, if the simulation results are different for the same input, it is concluded that the circuit is faulty [35]. In this paper, we evaluate ADC and PIR circuits in terms of reliability to achieve more efficient DBNs. For a circuit with n outputs, $2n$ single stuck-at faults would be possible since each output can set to 0 or 1. In 4-bit and 5-bit circuits, 8 and 10 single stuck-at faults can transpire respectively as shown in Tables VII and VIII. The 'X' shows the states that a faulty bit causes faulty output and the blank state illustrates that output is still correct despite a faulty bit. For example, the output of SS-PIR will be faulty when the desired output must be 31 just in a case that most significant bit becomes stuck at 0 (O_4/0). We calculate the faulty rate of each circuit by dividing the number of states that cause faulty outputs by all possible stuck-at fault states for each circuit. The faulty rate for 4-bit ADC and SC-PIR circuits is 50% because a bit flip for each output causes faulty output. In SS-PIR, the fault rate is 33% which is achieved by providing some dropouts between all possible outputs. To better comprehend the reliability advantages of SS-PIR circuits, we have defined a metric called *energy-error-faulty-product (EEFP)* as follows, where F is the fault rate:

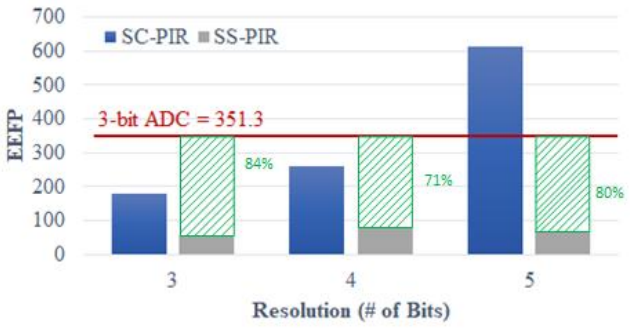


Fig. 11: EEFP for 3-bit, 4-bit and 5-bit SC-PIR and SS-PIR.

$$EEFP = N \times E \times err \times F \tag{8}$$

This incorporates the energy and reliability costs to achieve a particular accuracy. As shown in Fig. 11, all PIR-based circuits have better EEFP than 3-bit ADC up to 84% reduction except 5-bit SC-PIR. The SS-PIR can offer better performance also in the matter of reliability in comparison to ADC and SC-PIR.

VI. Conclusion

The concept of using sampling and count operations to interpret the probabilistic output of a p-bit based neuron offers an intriguing approach to realize a CMOS-based probabilistic interpolation recoder (PIR) for a spin-based stochastic binary neuron. Herein, we proposed a PIR circuit as a replacement for an analog-based approach to interpolate the output of the p-bit based activation functions in the last layer of a DBN circuit. The conventional method involved: first, using an RC circuit to continuously integrate the analog output of the p-bit, next an op-amp based sample and holder is

TABLE VI: Area of weighted array, activation function and interpolation circuits for several DBN topologies relative to the area occupied by a single p-bit-based neuron.

Topology	Normalized Area			
	Weighted Array	Activation Function	Interpolation Circuits	
			3-bit ADC	3-bit SS-PIR
784×10	$2600\times$	$10\times$	$4400\times$	$330\times$
$784 \times 200 \times 10$	$52000\times$	$2000\times$	$4400\times$	$330\times$
$784 \times 200 \times 200 \times 10$	$66000\times$	$400000\times$	$4400\times$	$330\times$

TABLE VII: Stuck-at fault table for 4-bit SC-PIR.

Bit	Stuck-at	Output = $O_3O_2O_1O_0$															
		1111	1110	1101	1100	1011	1010	1001	1000	0111	0110	0101	0100	0011	0010	0001	0000
O_3	0	X	X	X	X	X	X	X	X								
	1									X	X	X	X	X	X	X	X
O_2	0	X	X	X	X					X	X	X	X				
	1					X	X	X	X					X	X	X	X
O_1	0	X	X			X	X			X	X			X	X		
	1			X	X			X	X			X	X			X	X
O_0	0	X		X		X		X		X		X		X		X	
	1		X		X		X		X		X		X		X		X

TABLE VIII: Stuck-at fault table for 5-bit SS-PIR.

Bit	Stuck-at	Output = $O_4O_3O_2O_1O_0$					
		11111	01111	00111	00011	00001	00000
O_4	0	X					
	1		X	X	X	X	X
O_3	0		X				
	1			X	X	X	X
O_2	0			X			
	1				X	X	X
O_1	0				X		
	1					X	X
O_0	0					X	
	1						X

used to sample the output of the RC circuit, finally the analog sampled output is converted to a digital value through an op-AMP based ADC circuit and a priority encoder. Our proposed CMOS-based PIR circuit removes the need for all of area- and energy-consuming analog components existing in conventional circuits such as resistors, capacitors, and op-amps, and performs the interpolation operation only by using MOS-transistor based Boolean gates and flip-flops. In addition, the PIR circuits have an inherent single stuck-at fault tolerant features to tolerate either transient or permanent faults at the circuit's output without redundancy or active refurbishment overhead.

In order to verify the functionality and assess the performance of our PIR circuits, we used PIN-sim framework and SPICE circuit simulation tool to realize a circuit implementation of a $784 \times 200 \times 10$ DBN for the MNIST digit recognition application. During the development of the PIR circuits, we expected to observe a deterioration in the recognition accuracy of the DNBs, since we were replacing a continuous integration operation with a sampling and count operation. On the other hand, we also achieved a significant improvement in terms of resource utilization, and energy consumption. Thus, we have defined a performance metric called *Energy-Error-Product (EEP)* to exhibit the advantages of our proposed design. The simulation results exhibit 78% reduction in the EEP values for PIR-based DBN circuit compared to the conventional ADC-based method, which validates the efficiency of our design. Finally, the beneficial reliability properties of PIR circuits have been demonstrated for single stuck-at faults, relative to conventional interpolation, without requiring hardware redundancy while reducing fault rate from 50% to 33%.

ACKNOWLEDGMENT

This work was supported in part by the Center for Probabilistic Spin Logic for Low-Energy Boolean and Non-Boolean Computing (CAPSL), one of the Nanoelectronic Computing Research (nCORE) Centers as task 2759.006, a Semiconductor Research Corporation (SRC) program sponsored by the NSF through CCF 1739635.

REFERENCES

[1] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[2] R. Sarikaya, G. E. Hinton, and A. Deoras, "Application of deep belief networks for natural language understanding," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 22, no. 4, pp. 778–784, 2014.

[3] D. Le Ly and P. Chow, "High-performance reconfigurable hardware architecture for restricted boltzmann machines," *IEEE Transactions on Neural Networks*, vol. 21, no. 11, pp. 1780–1792, 2010.

[4] N. Lopes, B. Ribeiro, and J. Goncalves, "Restricted boltzmann machines and deep belief networks on multi-core processors," in *Neural Networks (IJCNN), The 2012 International Joint Conference on*. IEEE, 2012, pp. 1–7.

[5] M. N. Bojnordi and E. Ipek, "Memristive boltzmann machine: A hardware accelerator for combinatorial optimization and deep learning," in *High Performance Computer Architecture (HPCA), 2016 IEEE International Symposium on*. IEEE, 2016, pp. 1–13.

[6] S. B. Eryilmaz, E. Neftci, S. Joshi, S. Kim, M. BrightSky, H.-L. Lung, C. Lam, G. Cauwenberghs, and H.-S. P. Wong, "Training a probabilistic graphical model with resistive switching electronic synapses," *IEEE Transactions on Electron Devices*, vol. 63, no. 12, pp. 5004–5011, 2016.

[7] B. Yuan and K. K. Parhi, "Vlsi architectures for the restricted boltzmann machine," *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 13, no. 3, p. 35, 2017.

[8] A. Ardakani, F. Leduc-Primeau, N. Onizawa, T. Hanyu, and W. J. Gross, "Vlsi implementation of deep neural network using integral stochastic computing," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 10, pp. 2688–2699, 2017.

[9] R. Zand, K. Y. Camsari, S. D. Pyle, I. Ahmed, C. H. Kim, and R. F. DeMara, "Low-energy deep belief networks using intrinsic sigmoidal spintronic-based probabilistic neurons," in *Proceedings of the 2018 on Great Lakes Symposium on VLSI*, ser. GLSVLSI '18, 2018, pp. 15–20.

[10] R. Zand, K. Y. Camsari, S. Datta, and R. F. Demara, "Composable probabilistic inference networks using mram-based stochastic neurons," *J. Emerg. Technol. Comput. Syst.*, vol. 15, no. 2, pp. 17:1–17:22, Mar. 2019. [Online]. Available: http://doi.acm.org/10.1145/3304105

[11] K. Y. Camsari, S. Salahuddin, and S. Datta, "Implementing p-bits with embedded mtj," *IEEE Electron Device Letters*, vol. 38, no. 12, pp. 1767–1770, 2017.

[12] W. H. Choi, Y. Lv, H. Kim, J.-P. Wang, and C. H. Kim, "An 8-bit analog-to-digital converter based on the voltage-dependent switching probability of a magnetic tunnel junction," in *2015 Symposium on VLSI Technology (VLSI Technology)*. IEEE, 2015, pp. T162–T163.

[13] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, "A learning algorithm for boltzmann machines," *Cognitive science*, vol. 9, no. 1, 1985.

[14] B. Sutton, K. Y. Camsari, B. Behin-Aein, and S. Datta, "Intrinsic optimization using stochastic nanomagnets," *Scientific reports*, vol. 7, p. 44370, 2017.

[15] A. Sengupta, P. Panda, P. Wijesinghe, Y. Kim, and K. Roy, "Magnetic tunnel junction mimics stochastic cortical spiking neurons," *Scientific reports*, vol. 6, p. 30039, 2016.

[16] A. Sengupta, M. Parsa, B. Han, and K. Roy, "Probabilistic deep spiking neural systems enabled by magnetic tunnel junction," *IEEE Transactions on Electron Devices*, vol. 63, no. 7, pp. 2963–2970, July 2016.

[17] V. Ostwal, P. Debashis, R. Faria, Z. Chen, and J. Appenzeller, "Spin-torque devices with hard axis initialization as stochastic binary neurons," *Scientific reports*, vol. 8, no. 1, p. 16689, 2018.

[18] M. Tanaka and M. Okutomi, "A novel inference of a restricted boltzmann machine," in *2014 22nd International Conference on Pattern Recognition*, Aug 2014, pp. 1526–1531.

[19] S. Jain, A. Ankit, I. Chakraborty, T. Gokmen, M. Rasch, W. Haensch, K. Roy, and A. Raghunathan, "Neural network accelerator design with resistive crossbars: Opportunities and challenges," *IBM Journal of Research and Development*, vol. 63, no. 6, pp. 10–1, 2019.

[20] J. Zhang, H. Yang, F. Chen, Y. Wang, and H. Li, "Exploring bit-slice sparsity in deep neural networks for efficient reram-based deployment," *arXiv preprint arXiv:1909.08496*, 2019.

[21] Y. Long, X. She, and S. Mukhopadhyay, "Design of reliable dnn accelerator with un-reliable reram," in *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2019, pp. 1769–1774.

[22] X. Ma, G. Yuan, S. Lin, C. Ding, F. Yu, T. Liu, W. Wen, X. Chen, and Y. Wang, "Tiny but accurate: A pruned, quantized and optimized memristor crossbar framework for ultra efficient dnn implementation," *https://arxiv.org/abs/1908.10017*, 2019.

[23] G. Yuan, X. Ma, C. Ding, S. Lin, T. Zhang, Z. S. Jalali, Y. Zhao, L. Jiang, S. Soundarajan, and Y. Wang, "An ultra-efficient memristor-based dnn framework with structured weight pruning and quantization using admm," in *2019 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*. IEEE, 2019, pp. 1–6.

[24] J.-s. Seo, B. Brezzo, Y. Liu, B. D. Parker, S. K. Esser, R. K. Montoye, B. Rajendran, J. A. Tierno, L. Chang, D. S. Modha *et al.*, "A 45nm cmos neuromorphic chip with a scalable architecture for learning in networks of spiking neurons," in *2011 IEEE Custom Integrated Circuits Conference (CICC)*. IEEE, 2011, pp. 1–4.

[25] Y. Kim, Y. Zhang, and P. Li, "A reconfigurable digital neuromorphic processor with memristive synaptic crossbar for cognitive computing," *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 11, no. 4, p. 38, 2015.

[26] Y. Wang, T. Tang, L. Xia, B. Li, P. Gu, H. Yang, H. Li, and Y. Xie, "Energy efficient rram spiking neural network for real time classification," in *Proceedings of the 25th edition on Great Lakes Symposium on VLSI*. ACM, 2015, pp. 189–194.

[27] P. Wijesinghe, A. Ankit, A. Sengupta, and K. Roy, "An all-memristor deep spiking neural computing system: A step toward realizing the low-power stochastic brain," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 5, pp. 345–358, 2018.

[28] A. Sengupta, A. Ankit, and K. Roy, "Performance analysis and benchmarking of all-spin spiking neural networks (special session paper)," in *2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2017, pp. 4557–4563.

[29] R. P. Cowburn, D. K. Koltsov, A. O. Adeyeye, M. E. Welland, and D. M. Tricker, "Single-domain circular nanomagnets," *Phys. Rev. Lett.*, vol. 83, pp. 1042–1045, Aug 1999. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.83.1042

[30] P. Debashis, R. Faria, K. Y. Camsari, and Z. Chen, "Design of stochastic nanomagnets for probabilistic spin logic," *IEEE Magnetics Letters*, vol. 9, pp. 1–5, 2018.

[31] J. C. Sankey, Y.-T. Cui, J. Z. Sun, J. C. Slonczewski, R. A. Buhrman, and D. C. Ralph, "Measurement of the spin-transfer-torque vector in magnetic tunnel junctions," *Nature Physics*, vol. 4, no. 1, p. 67, 2008.

[32] C. Lin, S. Kang, Y. Wang, K. Lee, X. Zhu, W. Chen, X. Li, W. Hsu, Y. Kao, M. Liu *et al.*, "45nm low power cmos logic compatible embedded stt mram utilizing a reverse-connection 1t/1mtj cell," in *2009 IEEE International Electron Devices Meeting (IEDM)*. IEEE, 2009, pp. 1–4.

[33] N. Gougol, "Cmos operational amplifier design," *University of California at Berkeley Technical Report No. UCB/EECS-2016-223*, December 31, 2016.

[34] A. Stillmaker and B. Baas, "Scaling equations for the accurate prediction of cmos device performance from 180 nm to 7 nm," *Integration*, vol. 58, pp. 74–81, 2017.

[35] M. Habibi and H. Pourmeidani, "A hierarchical defect repair approach for hybrid nano/cmos memory reliability enhancement," *Microelectronics Reliability*, vol. 54, no. 2, pp. 475–484, 2014.

Hossein Pourmeidani is currently working towards Ph.D. degree in Computer Engineering at the University of Central Florida (UCF), Orlando, FL. He received double Master's degrees, M.Sc. in Computer Engineering from Islamic Azad University in 2012, M.Sc. in Computer Science from University of Mississippi in 2018, and a Bachelor's degree, B.Sc. in Computer Engineering from Islamic Azad University in 2010. His research interests include Machine Learning, Neuromorphic Computing Architectures, and Fault Tolerant Circuits.

Shadi Sheikhfaal received her B.Sc. degree in computer engineering from Azad University, Ardebil, Iran, in 2012 and her M.Sc. degree in computer engineering and computer systems architecture from Science and Research Branch of Azad University, Tehran, Iran, in 2014. She is currently pursuing her Ph.D. degree in computer engineering at University of Central Florida, Orlando, FL, USA. Her current research interests include biologically inspired computing, Neuromorphic computing and Spin-based computing.

Ramtin Zand received his Ph.D. degree in Computer Engineering from University of Central Florida (UCF), Orlando, FL, USA in 2019. He is a full-time faculty member at the Computer Science and Engineering department of the University of South Carolina, Columbia, SC. His research interests include Machine Learning and Neuromorphic Computing, Emerging Nanoscale Electronics including Spin-based Devices, Reconfigurable and Adaptive Computer Architectures, and Low-Power and Reliability-Aware VLSI Circuits.

Ronald F. DeMara (S87-M93-SM04) has been a full-time faculty member of ECE at the University of Central Florida since 1993. His interests are in computer architecture, post-CMOS devices, and reconfigurable fabrics with applications to intelligent and neuromorphic systems, on which he has published approximately 300 articles and holds one patent. He is a Senior Member of IEEE and served six terms as a Topical and/or Associate Editor of various Transactions of the IEEE Computer Society including *TC*, *TETC*, and *TVLSI*, as well as the Technical Program Committees of various IEEE conferences. He currently serves on the *IEEE Spectrum* Editorial Advisory Board. He has been Keynote Speaker of *IEEE RAW* and *IEEE ReConFig* conferences, and Guest Editor of *IEEE Transactions on Computers* 2017 Special Section on Innovation in Reconfigurable Fabrics and 2019 Special Section on Non-Volatile Memories. He received the *Joseph M. Biedenbach Outstanding Engineering Educator Award* from IEEE in 2008.