A Pulse-width Modulation Neuron with Continuous Activation for Processing-In-Memory Engines

Shuhang Zhang[1,2], Bing Li[1], Hai (Helen) Li[2,3], Ulf Schlichtmann[1]
[1]Chair of Electronic Design Automation, [2]Institute for Advanced Study, Technical University of Munich (TUM), Munich, Germany
[3]Department of Electrical and Computer Engineering, Duke University, Durham, NC, United States
Email: {shuhang.zhang, b.li, ulf.schlichtmann}@tum.de, hai.li@duke.edu

***Abstract*—Processing-in-memory engines have successfully been applied to accelerate deep neural networks. For improving computing efficiency, spiking-based designs are widely explored. However, spiking-based designs quantize inter-layer signals naturally, leading to performance loss. In addition, the spike mismatch effect makes digital processing necessary, impeding direct signal transfer between layers and thus resulting in longer latency. In this paper, we propose a novel neuron design based on pulse width modulation, avoiding the quantization step and bypassing spike mismatch via the continuous activation. The computation latency and circuit complexity can significantly be reduced due to the absence of quantization and digital processing steps, while keeping a competitive performance. Simulation results show that the proposed neuron design can achieve $> 100\times$ speedup compared with spiking-based designs. The area and power consumption can be reduced up to 74.87% and 25.63%.**

I. Introduction

In recent years, *deep neural networks* (DNNs) have achieved substantial breakthroughs in many applications, such as pattern recognition and natural language processing. To improve the execution speed and computing efficiency, neural network accelerators have extensively been studied [1]–[3]. Among these designs, *processing-in-memory* (PIM) that leverages memory structures for computation naturally bridges the performance gap between data processing and memory accesses, and thus, has attracted much interest. Various PIM designs based on traditional memories such as DRAM [4] and SRAM [5] as well as emerging technologies like *resistive memory* (RRAM) [6] have been proposed. Particularly, RRAM-based PIM designs stand out due to their dense data storage, natural support of *matrix-vector multiplication* (MVM), high computing efficiency and scalability [7]–[9].

Today's exploration on RRAM-based PIM designs spans from device to circuit and architecture. For circuit implementation, RRAM crossbar array where MVM is executed is the most important. At present, *one-transistor-one-RRAM* (1T1R) structure [10] is widely adopted to minimize the impact of sneak paths and improve the programming efficiency. Neuron circuit is another critical component that determines the area and efficiency of PIM designs. As depicted in Fig. 1, there are two types of neuron circuits: level-based [3] and spiking-based [11] designs. The former type uses voltage or current amplitude for data representation. This type of design usually requires complex components with large area and high power consumption, such as *trans-impedance amplifier* (TIA) and *analog digital converter* (ADC). The spiking-based designs encode data into spike trains. Such a digital format is naturally compatible with the surrounding designs. Simple circuitry like *integrate-and-fire circuit* (IFC) and counter are sufficient, achieving remarkable area and energy efficiency.

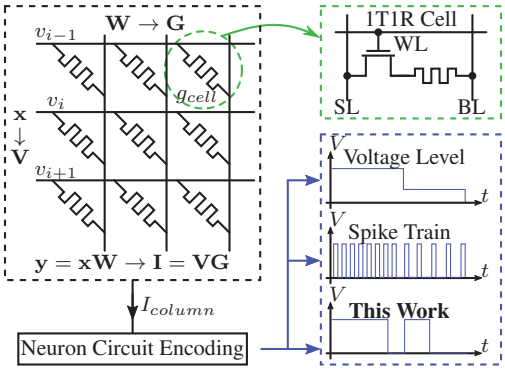

Fig. 1. RRAM array and various types of neuron formats.

However, there are two major challenges in spiking-based designs: *inter-layer signal quantization* and *spike mismatch*. First, the spikes passing through layers indeed are digitized signals due to data quantization, which inevitably induces information loss. Increasing the quantization precision, e.g., prolonging the computing time and thus increasing the number of spikes, helps alleviate the situation but also leads to longer latency [12]. Second, it is difficult to maintain the same spike width for various designs and under different operating conditions. For example, rate-coding based design uses the spike frequency to denote the activation strength. A strong activation can be translated into a train of narrow spikes while a weak signal induces a few wide spikes [6]. Such spike trains cannot be passed to the following layer directly, because the spike width also affects activation strength received by the following layer, but require an additional regulation circuitry.

To improve the inter-layer precision and avoid the spike mismatch, in this work, we propose a novel neuron design based on pulse width modulation. Unlike the previous work [13] that adopts the pulse width modulation to generate only the input signals to RRAM crossbars and uses IFCs to process the output currents, we aim at a neuron design that can provide a natural signal transfer across layers: the output currents will be encoded into pulse widths, which will then be fed into the subsequent layer. The simplified design also reduces the area and power consumption.

Moreover, when deploying DNNs on RRAM crossbars, data and weights need to be mapped to physical parameters, e.g., spike frequency and RRAM conductance. As these physical parameters can only provide positive values, a differential implementation using four crossbars [14] is usually adopted.

978-3-9819263-4-7/DATE20/©2020 EDAA

Such an approach increases the hardware cost to $4\times$ without including the additional circuitry to integrate the computation results from these crossbars. As an alternative solution, *shift operations* are often conducted which first shifts the entire data range to the positive domain. As the shift operations cause the output results to deviate from their original values, we need to adjust the difference once the computation is completed. For example, Chen and Li [15] include a digital circuity to cut off a fixed number from the output spikes. Our proposed pulse-width modulation neuron design integrates a shift elimination circuit to reduce the complexity of the digital interface.

We implemented and simulated the proposed neuron design at UMC 130nm technology node and applied the design to typical classification tasks using three different neural networks. Compared with traditional spiking-based designs [6], [11], [13], our proposed neuron circuit can achieve more than $100\times$ speedup as well as 74.87% and 25.63% reductions on area and power consumption, respectively.

The rest of this paper is organized as follows. In Section II, we explain the background of RRAM-based spiking PIM designs. The motivation of the proposed methods is presented in Section III. In Section IV, we elaborate the concept and circuit of the proposed pulse-width neuron. Simulation results and discussion are presented in Section V. At last, we conclude this work in Section VI.

II. Background

Many PIM engine designs have been explored based on various memory technologies. Among these solutions, RRAM is taken as one of the most promising candidates for its high computing efficiency and scalability. Therefore, we design and verify the neuron circuits based on RRAM-based computing platforms. This section introduces the background knowledge.

A. RRAM-based PIM Design

RRAM denotes a large group of nonvolatile memory technologies that use resistance states to store and represent data information. Commonly used devices include TiO_x-, TaO_x- and HfO_2-based thin-film structures. An RRAM device can be programmed between its *low resistance state* (LRS) and *high resistance state* (HRS). The resistive property of RRAM can also be leveraged to process MVM operation that is widely used in DNNs. The highly parallel computation in analog format remarkably improves the computing efficiency. Especially as DNN models become lager and more complicated, the data transfer between the computing cores and memory emerges as a performance bottleneck while the data processing in RRAM crossbar bridges the gap.

Deploying MVM on RRAM-based PIM requires mapping signed input data and weights to positive physical values. Previously Liu *et al.* [14] use four crossbars connected by routers to deal with the four combinations of the multiplications of signed input data and signed weights. To scale down the hardware cost, signed values are required to be shifted to positive values. Two methods are commonly adopted. The differential [13] applies shift operations on input data only and uses two crossbars to represent signed weights. This design offers more optimization space for tolerating potential defects or faults in crossbars after manufacturing. Shift operations can be applied on both data and weights, which requires only one crossbar [3].

Aiming to reduce hardware cost, we adopt the latter approach with a single crossbar. As shown in Fig. 1, an input data $\mathbf{x}$ can be mapped to a vector of voltage signals $\mathbf{V}$ as

$$\mathbf{V} = \alpha_1 \mathbf{x} + \boldsymbol{\beta_1}, \tag{1}$$

where α_1 is the scaling factor and $\boldsymbol{\beta_1}$ is the shifting bias. These voltage signals are applied to drive the rows of the crossbar. We also utilize shift operations to a weight matrix $\mathbf{W}$ and map its weight parameters to RRAM conductance. Here, the maximum weight (w_{max}) will be mapped to the LRS (g_{max}) and the minimum weight (w_{min}) will be mapped to the HRS (g_{min}). As such, $\mathbf{W}$ is represented by a crossbar with the following conductance array

$$\mathbf{G} = \alpha_2(\mathbf{W} - \mathbf{W_{min}}) + \boldsymbol{\beta_2}, \tag{2}$$

where $\alpha_2 = \frac{g_{max}-g_{min}}{w_{max}-w_{min}}$, $\mathbf{W_{min}} = w_{min} \cdot \mathbf{J}$, $\boldsymbol{\beta_2} = g_{min} \cdot \mathbf{J}$ and $\mathbf{J}$ is an all-ones matrix.

The voltages applied on the rows will generate currents flowing through RRAM cells. According to Kirchoff's current law, the output current can be expressed as follows

$$\mathbf{I} = \mathbf{VG}. \tag{3}$$

At the end of the columns, neuron circuits encode the set of summed currents into voltage signals and then transfer them into the digital domain for further processing, such as shifting or scaling. The processed results will then be used to generate the inputs to the next layer.

B. Shift Degradation

Due to the existence of shift operations, the output results contain additional components which need to be removed. More specifically, when realizing an MVM computation $\mathbf{y} = \mathbf{xW}$, the output $\mathbf{I}$ contains four components as follows:

$$\mathbf{I} = \alpha_1\alpha_2\mathbf{xW} + \alpha_1\mathbf{x}\boldsymbol{\beta_2} + \alpha_2\boldsymbol{\beta_1}\mathbf{W} + \boldsymbol{\beta_1}\boldsymbol{\beta_2}. \tag{4}$$

The first term $\alpha_1\alpha_2\mathbf{xW}$ represents the scaled original result. The latter three represent the degradation induced by the shift operations. Existing PIM architectures [3], [13] usually first transfer output currents into the digital domain and then use digital signal processing to remove the shift degradation. This is not an ideal solution as the digital processing breaks the signal coherence between layers, not even mentioning the reduced computing efficiency due to additional hardware cost and time.

III. Motivation

Although spiking-based design requires simple neuron circuit and achieves great power efficiency, there are two unsolved issues when deploying network models with multiple layers: inter-layer signal quantization and spike mismatch.

A. Inter-layer Signal Quantization

In spiking-based systems, the inter-layer signal is encoded into a number of spikes via specific circuits, e.g., IFC. This is indeed a quantization operation for inter-layer signals. The reduced activation precision potentially degrades the neural network performance [12]. Our preliminary simulation results in Fig. 2 present the scenario. Especially, the impact is more substantial to more complex neural networks. To mitigate the

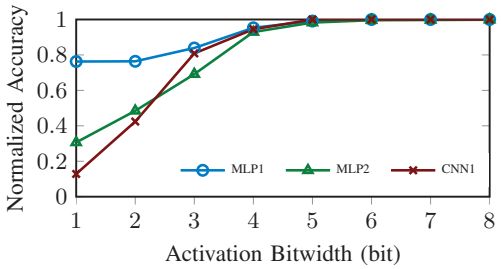

Fig. 2. Accuracy loss caused by neuron quantization. (MLP1, MLP2 evaluated on MNIST and CNN1 evaluated on Cifar10)

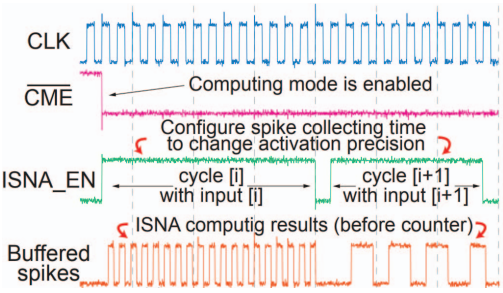

Fig. 3. Spikes generated by a spiking neuron under different input values [6].

quantization-caused performance loss, sufficient precision is required for activation, which usually indicates more output spikes and longer computing latency [6].

B. Spike Mismatch

An ideal spiking neuron should generate spike trains with the following features: (1) the frequency of spikes has a linear relationship to input activation; and (2) the spike width keeps constant under different activation. Unfortunately, a real spiking neuron cannot meet these requirements. For example, Fig. 3 presents the measured output spikes of a spiking neuron in [6]. It contains two computing cycles with different activation values. The activation value in the first cycle is larger, thus a higher frequency spike train is generated. Very importantly, the spike widths of the two cycles differ significantly. In the second computing cycle when a smaller activation value is applied, fewer spikes are produced while the widths of these spikes are much longer compared to those in the first cycle. If these spike trains are fed directly into the next layer, the current generated in a specific time period does not reflect the original input values correctly. Therefore, in spiking-based PIMs, counters are mandatory to transform these output spikes into the digital domain.

IV. Pulse-width Modulation Neuron Design

In this work, we propose a pulse-width modulation based neuron design to overcome the drawbacks of spiking-based neurons and get rid of the complex digital interface design. Our objective is to accelerate the computing speed and reduce power consumption and design area. In addition, this neuron can implement non-linear activation functions between layers and eliminate the shift degradation.

A. Design Concept

In traditional designs, the output current signal is usually encoded into the spike frequency. Our design first converts the current to the charge amount, which then determines the output pulse width.

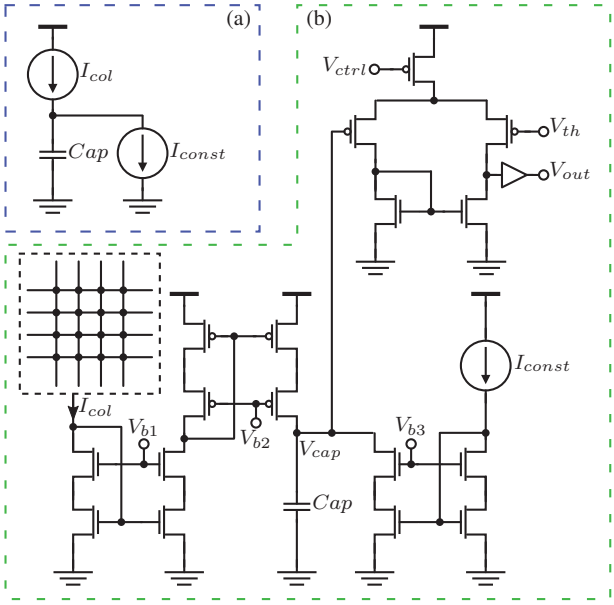

Fig. 4. (a) Concept of the proposed neuron. (b) Partial circuit of the proposed neuron.

The operation of the proposed neuron can be separated into two steps, as depicted in Fig. 4(a). In the first step, the input data is encoded to pulses of different widths and they are applied at rows of the crossbar, so the capacitor under each column will be charged by a variable current. This variable column current is represented by the current source I_{col} in this figure. The total accumulated charge during the charging time t_c in the first step can be represented as

$$Q = \int_{t_0}^{t_0+t_c} I_{col}(t)dt. \tag{5}$$

In the second step, the variable current source is cut off and the total charge accumulated during the first step will be discharged linearly via a constant current source I_{const}. The discharging time t_d represents the encoded pulse width that can be approximated as

$$t_d = \frac{Q}{I_{const}}. \tag{6}$$

The pulse-width modulation neuron overcomes the most critical problems of spiking-based neurons. The output current is converted to the total charge amount and the pulse width is linear with the total charge amount, which is a continuous value, avoiding the quantization problem. Additionally, generated pulses' widths can represent original values, bypassing the spike mismatch problem. So these pulses can be fed directly into the next layer. Also, in this neuron design, only partial circuit is activated in each step, charging or discharging part of the whole neuron circuit. This design avoids frequent switching behaviors so that this circuit can achieve better power efficiency than spiking-based designs.

B. Pulse-wdith Modulation Neuron Circuit

In Fig. 4(b), a partial circuit of the proposed neuron is presented. In the first charging step, the current from one column in the crossbar I_{col} will be connected to a current mirror and charge the capacitor. The current mirror under the crossbar will divide the current by setting different transistor sizes, so that

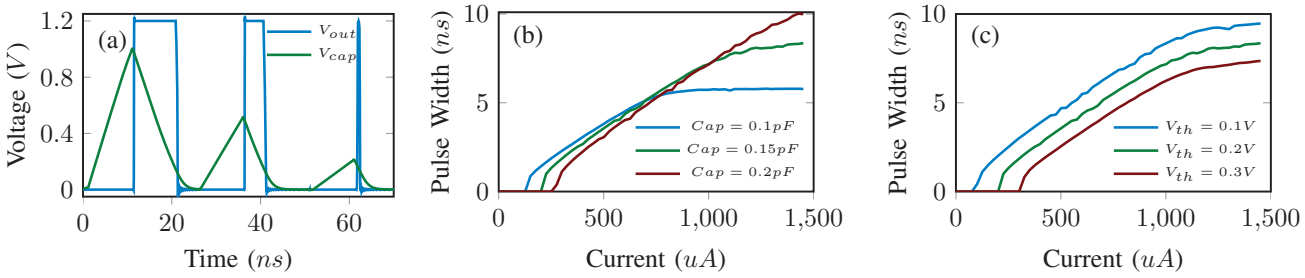

Fig. 5. (a) Simulated waveforms of V_{cap} and V_{out}. (b) Simulated pulse width with $V_{th} = 0.2V$. (c) Simulated pulse width with $Cap = 0.15pF$.

a smaller current will be generated in the following circuits, leading to the reduction of power consumption and total circuit area. The discharging current mirror and operational amplifier do not function by controlling the biasing voltages. In the discharging step, pulses are not applied on the rows of the crossbar, so that no current will flow from the crossbar, which automatically turns off these current mirrors used to charge the capacitor. Meanwhile, the current mirror used for discharging is turned on and a constant current is mirrored to discharge the capacitor.

In Fig. 5(a), the waveform of a capacitor with different charging currents is drawn. We observe that the voltage across the capacitor is similar to a triangle wave instead of a rectangular pulse, so we need a comparator and a buffer to transform this signal to a pulse signal shown in the same figure. Due to the existence of the comparator, the discharging time in Eq. (6) has to be modified to

$$t_d^{'} = \frac{Q}{I_{const}} - \frac{C \cdot V_{th}}{I_{const}}. \tag{7}$$

The second term caused by the threshold voltage can be used as an in-situ ReLU function. In Eq. (7), we see that the threshold discharging time is determined by capacitor size C and comparator threshold voltage V_{th}. So we can modify these two design parameters to adjust the threshold discharging time.

In Fig. 5(b), the pulse width of the proposed neuron circuit under different capacitor sizes is shown. With the increased charging current, the discharging time increases to a specific value and becomes saturated because of the limitation of the transistor size in current mirrors. In addition, a large capacitor can store more charge and become saturated later. In Fig. 5(c), the pulse width of the proposed neuron circuit under different threshold voltages is shown. With a higher threshold voltage, the neuron will generate valid pulses later. In this case, the pulse width will also be saturated because of the limitation of the capacitor size.

The capacitor size and threshold voltage affect the discharging time threshold, which can be also used to reduce the shift error caused by data/weight shift operations. The capacitor size cannot be modified easily in a manufactured chip, but the threshold voltage can be selected to achieve the best performance.

C. Shift Degradation Elimination

In spiking-based designs, the shift degradation is eliminated via digital processing, but digital processing breaks the coherence of different layers in neural networks. In pulse-width modulation neuron design, we aim to find a way to deal with the shift degradation without digital processing.

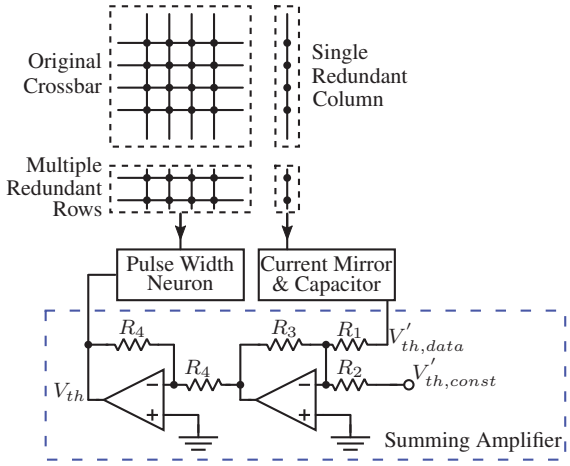

Fig. 6. Concept of shift degradation elimination.

As shown in Eq. (4), the shift degradation can be separated into two groups, *data-related* and *data-unrelated*. The second shift term in Eq. (4) is related to the input data, and the last two terms are not related to the input data. Therefore, the threshold voltage should also be composed of two parts as shown in Eq. (9), where $F(\cdot)$ is used to reduce the degradation caused by input data, and $G(\cdot)$ can be used to reduce the constant shift degradation.

$$V_{th} = V_{th,data} + V_{th,const} \tag{8}$$

$$= F(\mathbf{x}\boldsymbol{\beta_2}) + G(\alpha_2 \boldsymbol{\beta_1}\mathbf{W} + \boldsymbol{\beta_1}\boldsymbol{\beta_2}). \tag{9}$$

For the data-related shift degradation, it requires an additional column to resolve, because this shift is different for each input data. The RRAM conductance of this column is set to one value, so the charge accumulated in the capacitor represents scaled data-related shift degradation. The voltage across the redundant capacitor can be used to reduce the data-caused shift degradation.

For constant shift errors, the third term $\alpha_2\boldsymbol{\beta_1}\mathbf{W}$ in Eq. (4) is related to the weight matrix and will cause different $V_{th,const}^{'}$ for different columns, which requires multiple summing amplifiers, causing unaffordable circuit effort. To reduce the circuit effort, in this work, we use redundant rows to reduce the weight matrix caused degradation by

$$\mathbf{y_{supp}^{'}} = \alpha_1\alpha_2\mathbf{x}^{'}\mathbf{W}^{'} + \alpha_1\mathbf{x}^{'}\boldsymbol{\beta_2} + \alpha_2\boldsymbol{\beta_1}\mathbf{W}^{'} + \boldsymbol{\beta_1}\boldsymbol{\beta_2}. \tag{10}$$

In Eq. (10), $\mathbf{x}^{'}$ is the input of redundant rows and $\mathbf{W}^{'}$ is the weight matrix of redundant rows, which make the column sum

of $\mathbf{W}$ and $\mathbf{W}^{'}$ equal to zero. Therefore, the item $\alpha_2\boldsymbol{\beta_1}\mathbf{W}$ and $\alpha_2\boldsymbol{\beta_1}\mathbf{W}^{'}$ will be eliminated. In addition, if we set $\mathbf{x}^{'}$ equal to zero, we only include a constant $\boldsymbol{\beta_1\beta_2}$ to our final result.

This degradation can also be reduced in the algorithm training step. After normal quantization training [16], the absolute value of the column sum in a weight matrix could be large and will require large number of redundant rows. To reduce the number of redundant rows, the bias retraining step can be modified by changing the cost function in Eq. (11)

$$Cost = CrossEntropy + \lambda \sum ColumnSum. \tag{11}$$

After the modified bias retraining step, the column sum is reduced, which not only reduces the shift degradation, but also reduces the number of redundant rows required in this design.

The detailed implementation is shown in Fig. 6. The current mirror and capacitor circuit are similar to the circuit in the pulse width neuron. Afterwards, the data-caused threshold voltage $V_{th,data}^{'}$ and a constant threshold voltage $V_{th,const}^{'}$ are then summed as the threshold voltage V_{th} used in the pulse width neuron. The resistors used in the summing amplifier should be selected to recover the scaled voltage, so the V_{th} can be expressed as

$$V_{th} = \frac{R_1}{R_3} \cdot V_{th,data}^{'} + \frac{R_2}{R_3} \cdot V_{th,const}^{'}. \tag{12}$$

V. Simulation Results and Discussion

A. Simulation Setup

We implement the proposed neuron using UMC 130nm PDK. The RRAM cell in the crossbar adopts the 1T1R structure. The LRS and HRS of RRAM are set to $50k\Omega$ and $1M\Omega$, respectively [11]. The IR-drop in crossbar is set based on [17]. The capacitor size of the proposed neuron is set to $17fF$, which is the minimum MIM capacitor provided by this PDK. The charging and discharging time is set to $1ns$, providing enough encoding space for the pulse generators used for the first layer [18].

To evaluate the efficiency of the proposed neuron circuit, we deploy three networks (MLP1, MLP2 and CNN1) on MNIST [19] and Cifar10 [20]. CNN1 is developed based on [16], which has fewer kernels in convolutional layers and achieves similar classification accuracy. The network training is conducted in PyTorch on a workstation with an Intel 3.6 GHz CPU and an NVIDA GeForce GTX 1080Ti graphics card.

B. Accuracy and Latency Comparison

In Table I, the accuracy in software ("Original") and hardware level implementations, and computing latency in different scenarios are presented. Applying weight quantization ("Weight-quantized") makes the accuracy decrease slightly compared to the original accuracy. The accuracy decreases 0.07%, 1.06% and 2.91% on these three networks, respectively. The following two columns show the accuracy of the ideal spiking-based neural network that applies both weight and activation quantization. A high activation precision ("8-bits") can maintain accuracy, but a low activation precision ("2-bits") can degrade the accuracy dramatically. Our work applies the pulse-width modulation neuron design and considers the weight quantization and physical limitations, such as IR drop, and non-linearity between the neuron input current and output pulse width. It shows only marginal accuracy degradation.

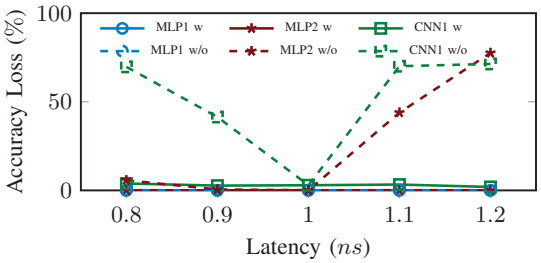

Fig. 7. Accuracy under different computing periods

Table I also summarizes the computing latency of one layer. Our design requires only $2ns$ to finish one layer computation. The latency data of spiking-based neural network is from [6]. We observe that 8-bit activation precision needs $200ns$ for one layer, although the computing time for digital signal processing is not included into the consideration. Therefore, this work can achieve more than $100\times$ latency reduction. If the spiking-based neural networks operate at a similar speed, the activation precision should be downgraded to 2-bits and the latency needs to be at approximately $3.1ns$, but the accuracy is much lower than what is demonstrated in this work.

C. Computing Latency Analysis

The nominal charging and discharging time for our neuron design is set to $1ns$. In this section, we also test different charging and discharging time to evaluate the sensitivity of the current working point. In Fig. 7, five different periods are used. Under different working points, the accuracy loss of these three neural networks are different. For the simple MLP1, the accuracy is almost identical to the original accuracy, but for more complex neural networks, MLP2 and CNN1 show higher accuracy loss due to the error accumulation through layers. To deal with the accuracy loss caused by deviation from the nominal working point, we need to calibrate these neurons by controlling the threshold voltage V_{th}. With calibration, the accuracy can be rescued to the original accuracy, which is shown in the same figure. Besides, the power under different charging and discharging time is almost the same at approximately $119uW$, because the power is only determined by the current flowing from the crossbar and the computation period will not affect the power.

D. Process Variations

Neuron performance under process variations is discussed in this section. We apply Monte-Carlo simulations to the proposed neuron circuit and take 1000 samples. In this simulation, we mainly consider the effect of transistor threshold voltage $V_{th,tran}$, electron mobility u_0, and transistor channel length L. In Fig. 8, the accuracy of hardware implemented neural networks decreases with increasing of process variations. In addition, we notice that a more complex neural network experiences more accuracy degradation, because the error caused by process variations will be accumulated and transferred to the following layers. Therefore, to cope with the degraded accuracy, we need to calibrate the neuron circuit by controlling the threshold voltage. In the same figure, we see that the rescued accuracy is almost the same with the original accuracy.

TABLE I
ACCURACY AND LATENCY COMPARISON

			Accuracy Comparison					Latency Comparison		
NN	Dataset	Size	Original	Weight-quantized	8-bits	2-bits	This Work	8-bits	2-bits	This Work
MLP-1	MNIST	1 FC (400, 10)	90.55%	87.36%	87.36%	66.74%	87.29%	$200ns$	$3.1ns$	$2ns$
MLP-2	MNIST	2 FC (400, 512, 10)	96.26%	95.20%	95.18%	37.85%	94.14%			
CNN-1	Cifar10	2 Conv (32, 64 filters) + 2 FC (1600, 512, 10)	83.96%	81.36%	81.20%	33.26%	78.45%			

TABLE II
POWER AND AREA COMPARISON

	[6]*	[13]	[11]*	[21]	This Work
Mode	Spiking-based			Level-based	Pulse-width-based
Tech node	150nm	130nm	130nm	65nm	130nm
Results	Measurement	Simulation	Simulation	Measurement	Simulation
LRS/HRS	–	$50k\Omega/1M\Omega$	$50k\Omega/1M\Omega$	–	$50k\Omega/1M\Omega$
Area	$37.96kF^2$	–	$10.37kF^2$	$1.66MF^2$	$9.54kF^2$
Power	–	$146uW$	$160uW$	$2.3mW$	$119uW$

* The area and power of counters are not considered.

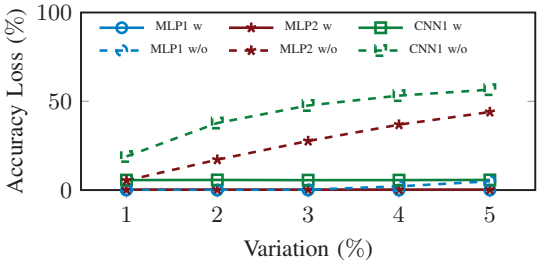

Fig. 8. Accuracy under different process variations.

E. Area and Power Comparison

In Table II, we compare the area and power of this work with some of the state-of-the-art designs. For spiking-based designs, our design replaces the spiking neuron and counter. This work can achieve up to 74.87% and 25.63% area and power reduction. If the area and power of the counter is included, our design can achieve more reduction. Among selected spiking-based designs, we focus on [11] and [13], which also adopt 130nm PDK and use the same RRAM parameters for schematic simulations. When compared with [11] and [13], our design still can achieve 8% and 25.63% area and power reduction. For level-based design, we mainly consider the power and area of ADCs, which dominates in level-based designs. Compared with this 8-bit ADC, our design can achieve 99.43% and 94.83% area and power reduction.

As mentioned in Section IV, to implement shift degradation elimination, several redundant rows and one redundant column are required for one RRAM crossbar, resulting in extra RRAM cells. In our work, we also examine the RRAM cells overhead, and only 18.38% more cells are required on average, which achieves great area efficiency over two or four crossbars implementations.

VI. CONCLUSIONS

In this work, a novel neuron based on pulse width modulation is presented. This neuron design avoids the problems of traditional spiking-based designs by encoding the current into continuous pulse width, excluding the complex digital interface designs. Therefore, the computing latency can significantly be reduced. The simulation results show the proposed design can achieve similar accuracy compared with high precision spiking-based designs, and latency is reduced dramatically. This design also achieves great area and power reduction compared with some state-of-the-art designs.

ACKNOWLEDGMENT

This work was supported by Technical University of Munich Institute for Advanced Study, funded by the German Excellence Initiative and the European Union Seventh Framework Programme under grant agreement n° 291763.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.
[2] C. Farabet, Y. LeCun, K. Kavukcuoglu, E. Culurciello, B. Martini, P. Akselrod, and S. Talay, "Large-scale FPGA-based convolutional networks," *Scaling up Machine Learning: Parallel and Distributed Approaches*, pp. 399–419, 2011.
[3] M. Hu, J. P. Strachan, Z. Li, E. M. Grafals, N. Davila, C. Graves, S. Lam, N. Ge, J. J. Yang, and R. S. Williams, "Dot-product engine for neuromorphic computing: programming 1T1M crossbar to accelerate matrix-vector multiplication," in *Proc. Design Autom. Conf.*, 2016, pp. 1–6.
[4] S. Angizi and D. Fan, "Graphide: A graph processing accelerator leveraging in-DRAM-computing," in *Proceedings of Great Lakes Symposium on VLSI*, 2019, pp. 45–50.
[5] R. Liu, X. Peng, X. Sun, W.-S. Khwa, X. Si, J.-J. Chen, J.-F. Li, M.-F. Chang, and S. Yu, "Parallelizing SRAM arrays with customized bit-cell for binary neural networks," in *Proc. Design Autom. Conf.*, 2018, pp. 1–6.
[6] B. Yan, Q. Yang, W.-H. Chen, K.-T. Chang, J.-W. Su, C.-H. Hsu, S.-H. Li, H.-Y. Lee, S.-S. Sheu, M.-S. Ho *et al.*, "RRAM-based spiking nonvolatile computing-in-memory processing engine with precision-configurable in situ nonlinear activation," in *Symposium on VLSI Technology*, 2019, pp. T86–T87.
[7] N. Raghavan, R. Degraeve, A. Fantini, L. Goux, S. Strangio, B. Govoreanu, D. Wouters, G. Groeseneken, and M. Jurczak, "Microscopic origin of random telegraph noise fluctuations in aggressively scaled RRAM and its impact on read disturb variability," in *International Reliability Physics Symposium*, 2013, pp. 5E.3.1–5E.3.7.
[8] M. Zhao, H. Wu, B. Gao, X. Sun, Y. Liu, P. Yao, Y. Xi, X. Li, Q. Zhang, K. Wang *et al.*, "Characterizing endurance degradation of incremental switching in analog RRAM for neuromorphic systems," in *International Electron Devices Meeting*, 2018, pp. 20.2.1–20.2.4.
[9] C.-H. Cheng, A. Chin, and F. Yeh, "Novel ultra-low power RRAM with good endurance and retention," in *Symposium on VLSI Technology*, 2010, pp. 85–86.
[10] P.-Y. Chen and S. Yu, "Compact modeling of RRAM devices and its applications in 1T1R and 1S1R array design," *IEEE Trans. Electron Devices*, vol. 62, no. 12, pp. 4022–4028, 2015.
[11] C. Liu, B. Yan, C. Yang, L. Song, Z. Li, B. Liu, Y. Chen, H. Li, Q. Wu, and H. Jiang, "A spiking neuromorphic design with resistive crossbar," in *Proc. Design Autom. Conf.*, 2015, pp. 1–6.
[12] F. Liu and C. Liu, "Towards accurate and high-speed spiking neuromorphic systems with data quantization-aware deep networks," in *Proc. Design Autom. Conf.*, 2018, pp. 1–6.
[13] H. Jiang, K. Yamada, Z. Ren, T. Kwok, F. Luo, Q. Yang, X. Zhang, J. J. Yang, Q. Xia, Y. Chen *et al.*, "Pulse-width modulation based dot-product engine for neuromorphic computing system using memristor crossbar array," in *Proc. Int. Symp. Circuits and Syst.*, 2018, pp. 1–4.
[14] X. Liu, M. Mao, B. Liu, B. Li, Y. Wang, H. Jiang, M. Barnell, Q. Wu, J. Yang, H. Li *et al.*, "Harmonica: A framework of heterogeneous computing systems with memristor-based neuromorphic computing accelerators," *IEEE Trans. Circuits Syst. I*, vol. 63, no. 5, pp. 617–628, 2016.
[15] F. Chen and H. Li, "Emat: an efficient multi-task architecture for transfer learning using reram," in *Proc. Int. Conf. Comput.-Aided Des.*, 2018, p. 33.
[16] Y. Wang, W. Wen, L. Song, and H. H. Li, "Classification accuracy improvement for neuromorphic computing systems with one-level precision synapses," in *Proc. Asia and South Pacific Des. Autom. Conf.*, 2017, pp. 776–781.
[17] B. Liu, H. Li, Y. Chen, X. Li, T. Huang, Q. Wu, and M. Barnell, "Reduction and IR-drop compensations techniques for reliable neuromorphic computing systems," in *Proc. Design Autom. Conf.*, 2014, pp. 63–70.
[18] J.-C. Liu, C.-J. Huang, and P.-Y. Lee, "A high-accuracy programmable pulse generator with a 10-ps timing resolution," *IEEE Trans. VLSI Syst.*, vol. 26, no. 4, pp. 621–629, 2018.
[19] Y. LeCun, C. Cortes, and C. Burges, "MNIST handwritten digit database," *AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist*, 2010.
[20] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Citeseer, Tech. Rep., 2009.
[21] K. Ohhata, "A 2.3-mw, 950-mhz, 8-bit fully-time-based subranging ADC using highly-linear dynamic VTC," in *Symposium on VLSI Circuits*, 2018, pp. 95–96.