An Energy-Efficient One-Shot Time-Based Neural Network Accelerator Employing Dynamic Threshold Error Correction in 65 nm

Luke R. Everson, *Student Member, IEEE*, Muqing Liu, *Student Member, IEEE*, Nakul Pande, *Student Member, IEEE*, and Chris H. Kim, *Fellow, IEEE*

***Abstract*— As neural networks continue to infiltrate diverse application domains, computing will begin to move out of the cloud and onto edge devices necessitating fast, reliable, and low-power (LP) solutions. To meet these requirements, we propose a time-domain core using one-shot delay measurements and a lightweight post-processing technique, dynamic threshold error correction (DTEC). This design differs from traditional digital implementations in that it uses the delay accumulated through a simple inverter chain distributed through an SRAM array to intrinsically compute resource intensive multiply-accumulate (MAC) operations. Implemented in 65-nm LP CMOS, we achieve an energy efficiency of 104.8 TOp/s/W at 0.7-V with 3b resolution for 19.1 fJ/MAC.**

***Index Terms*— Machine learning (ML), neuromorphic computing, time-domain computing, time-to-digital converter (TDC).**

I. Introduction

THE ever-increasing demand for higher performance and energy efficiency in machine learning (ML) applications has driven an impressive range of application-specified integrated circuits (ASICs) [1]–[8], [11]–[13] aimed at meeting the challenge. Digital SoCs [3]–[5], [7] have found success by restricting the weight resolution [8], changing memory access structures [4], and guarding operations when the input is zero [3]. However, all require large registers to store intermediate results and complex multiplier blocks.

SRAM memory-based current summation designs have been proposed as well [9], [10]. In [9], SRAMs are used to store weights. Interspersed in the array are local analog moving average blocks, the control unit that implements the charge sharing across bitlines. This design makes use of low-power (LP) analog techniques to drive down power but relies on charge sharing and utilizes a time-dependent pre-charge scheme to implement the input. These two techniques limit the scalability of the design and as such they are limited to convolutional operations that have reduced input lengths due to the filter size. As a result of using charge sharing, the SRAMs cannot be directly connected together which is why 10T bitcells are utilized instead of the conventional, denser 6T bitcells. Reference [10] also leverages charge sharing between SRAMs in order to implement signed multiplication. They demonstrate that using in-memory computing can reduce power consumption by as much as 4.5×. However, when used as an analog device, process variation from SRAMs necessitates on-chip learning. Models learned on one chip and applied to another can cause a 43% drop in accuracy. This requires a massive increase in overhead, reducing the efficiency of the core.

One of the key limitations of using SRAMs is the volatile storage. This requires the power supply to be connected constantly drawing static leakage power. If it is disconnected, an overhead will be incurred to reprogram the array to prepare the array for computation. Nonvolatile storage devices such as ReRAM [11] and eFlash [12] present opportunities to have persistent weight storage. The arrays work much in the same manner as SRAM crossbar arrays where the accessed bitline currents are summated to implement the multiply-accumulate (MAC). ReRAM is considered an ideal candidate due to not only the low access latency and energy but also the small footprint could enable very dense arrays. ReRAM can be thought of as a programmable, analog, and nonvolatile resistor. However, due to the nonuniform analog resistance states, [11] asserts that it can cause errors in the convolution. They work around this by using ReRAM as a digital device that has benefits including: better programming accuracy, binary voltages applied to wordline (WL) is scalable due to lower IR-drop in large arrays, and it does not require a large ON/OFF-resistance ratio previously required in analog ReRAM [11]. The key limitation is the lack of a capable commercially available process and even [11] does not have the measurement results to support their claims. eFlash arrays utilize multi-level storage element and are logic compatible to reduce cost and available in all processes [12]. The main drawback is the large cell size due to the I/O devices required to limit gate leakage on the storage node, on the order of 4× larger than an SRAM even accounting for the eFlash multi-level cell storage.

An emerging trend [1], [2] has been to employ time-domain circuits to implement dot-products, the main kernel for ML applications. Fig. 1 details how dot-products are computed in the time domain and in conventional digital implementations. In time domain, the delay is modulated by the

Manuscript received January 14, 2019; revised March 11, 2019 and April 25, 2019; accepted April 29, 2019. This paper was approved by Guest Editor Chen-Hao Chang. This work was supported in part by the National Science Foundation under Award CCF-1763761 and in part by IGERT under Grant DGE-1069104. *(Corresponding author: Chris H. Kim.)*

The authors are with the Electrical and Computer Engineering Department, University of Minnesota, Minneapolis, MN 55455 USA (e-mail: evers193@umn.edu; chriskim@umn.edu).

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

Digital Object Identifier 10.1109/JSSC.2019.2914361

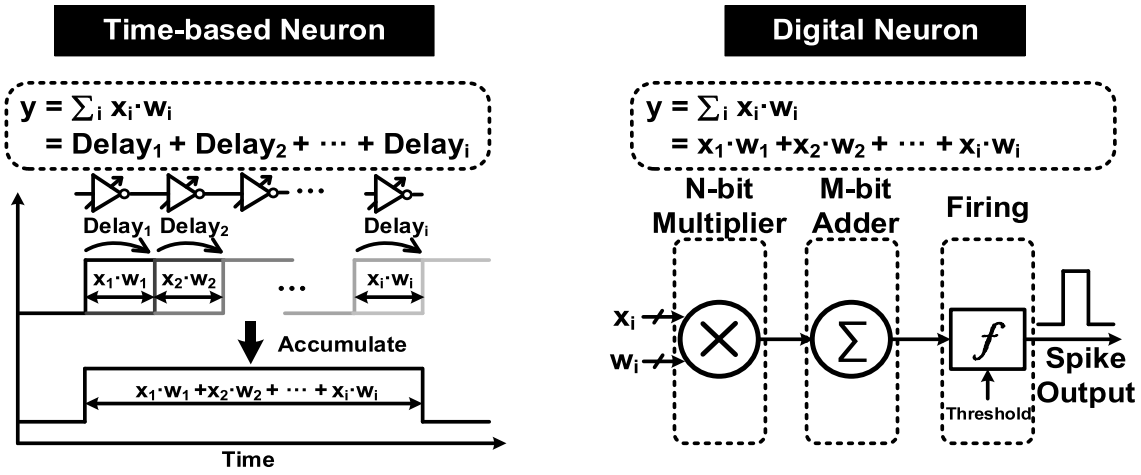


Fig. 1. Time-based neurons utilize the delay through basic circuit elements such as inverters to implement the dot product. Digital neurons use conventional Boolean logic for arithmetic operations. Both architectures can be mapped to deep learning applications.

application inputs and weights to generate proportional delays. These delays are accumulated and can be routed to a time-to-digital converter (TDC) or counter to be processed for use in the deep learning application. Alternatively, the digital approach relies on many multiplier blocks and wide merging adders, typically in an arraylike structure, to generate dot-products. The primary benefit of time-domain circuits is that the accumulate portion of the MAC is intrinsic to the architecture. Furthermore, the processing unit can be realized as a collection of inverters making the area and active power consumption very low. Digital methods can leverage existing IP-blocks for multipliers and adders and do not require calibration, unlike the time-domain circuits. In addition, digital circuits can handle higher bit operations more effectively due to the binary encoding.

Previous time-domain neuromorphic chips have several limitations. In [2], a digitally controlled oscillator was used to modulate the frequency, by switching capacitor loads representing the weights, while the number of cycles in a set sampling period was counted. While this closed-loop structure has the benefit of canceling temporal noise, it must oscillate for many cycles to generate a reliable result. Reference [1] is also a delay line-based approach, but the outputs and weights are restricted to binary. More critically, their design has twice the area overhead due to the fact that they utilize local reference delay lines instead of a global reference and can limit the potential scalability of the architecture. In this paper, we have addressed the shortcomings of the previous designs by implementing digitally controlled delay lines (DDLs) that are compared to a shared reference delay to compute multi-bit MACs.

This paper is organized as follows. Section II provides a detailed narrative on-chip implementation, the one-shot and how to apply time-based circuits to neuromorphic computing. The performance of the implemented two-bit TDC is analyzed in Section III. A novel accuracy boosting technique, dynamic threshold error correction (DTEC), is explained in Section IV. Measurement results from the chip and the performance on the target application are discussed in Section V. Finally, conclusions are framed in Section VI. The conference version of this paper was published in [13].

II. One-Shot and Time-Based Neuromorphic Concept

Conventionally, Boolean computations are used to realize arithmetic operations in hardware. However, time-domain circuits can also be used at an advantage of lower area and power per processing unit and reduced design complexity. The kernel of all ML algorithms can be distilled into a dot-product; $y = \sum xw + b$, where x is an input vector, w a weight matrix, and b is a bias, or offset vector. Our high-level architecture is shown in Fig. 2. An input pulse is presented on the left side of the core and the delay of each stage is modulated based on the application inputs. We describe it as one-shot because each pulse gets evaluated once. Each stage has eight delay units (DUs) with output taps which the pulse travels through as shown in Fig. 3. The number of DU enabled depends on the one-hot encoded weight, w, stored locally in SRAM cells, and the input pixel, x, which is applied across the array on the bitlines, BL. Each DU has two inverters to retain consistent polarity between stages. This is critical in the event that the rising and falling propagation delays are not matched, as well as ensuring correct polarity at the TDC. The output tap is realized as a complex tristate gate and the functionality is described in Fig. 4. The first column shows the circuit schematic and corresponding connections between different DUs. The right four columns show the activated paths, shown with black lines, depending on the values of the input, x, and weight, w. DU_5 is the nominal stage delay and is activated through the right branch of the circuit when the input bit is off, representing "zero delay." The right table shows the mapping between the algorithm-trained weights and the delays realized in the chip at each stage. When the input, x, is present, the left branch is enabled in the DU corresponding the weight bit of the stage. Larger positive weights map to shorter delays relative to the reference DDL, and conversely negative weights correspond to longer delays. The accumulation in the MAC is achieved naturally as the

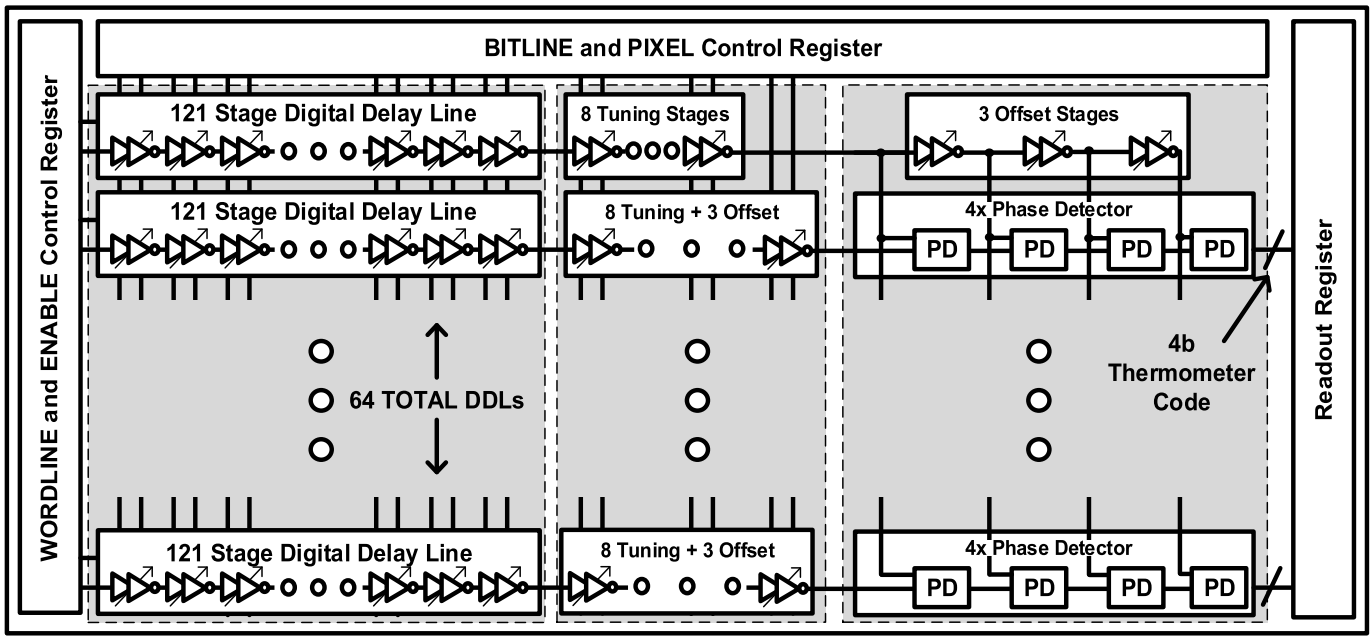


Fig. 2. Top-level schematic of the time-based neuromorphic core. Layout is based on SRAM array. The core contains 64 DDLs each with 129 stages and each has a 4b PD that is compared to the reference DDL.

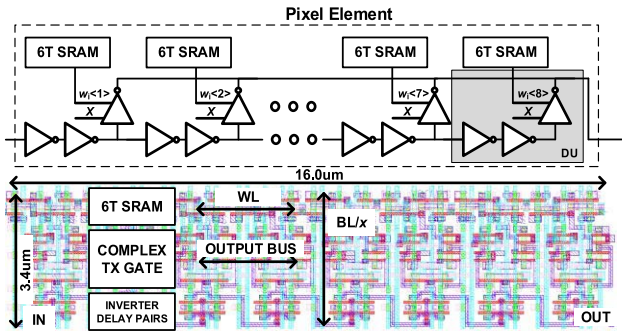
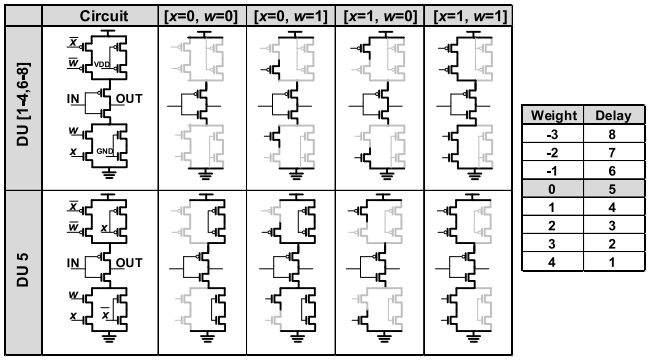


Fig. 3. Schematic of pixel stage (top). Complex tristates shown in Fig. 4 drive output bus. Layout of pixel stage (bottom).



Weight	Delay
-3	8
-2	7
-1	6
0	5
1	4
2	3
3	2
4	1

Fig. 4. Complex tristate connections used to implement dot product based on input and weights (left). 3b weight-delay mapping (right).

pulse passes sequentially through the DDL, stage by stage. The layout of each DU in the stage is pitch matched to a 6T SRAM so the layout is regular, compact, and scalable. The bias vector is applied in the same way for the last eight units. In addition, it can be used to tune process variation, so that during evaluation, those pixels are always activated. Fig. 5 shows the relationship between the time-domain computation in the chip and the expected arithmetic output. The phase detector (PD) output maps roughly to the rectified linear unit (ReLU) transfer function. When the reference pulse beats the neuron rising edge, all four thermometer bits are zero, regardless of the magnitude. The transfer function between the four bits is linear and then clips, or saturates, once the neuron pulse is faster than all the offsets.

III. TDC Performance Analysis

The delay of the time-based circuits can be tuned to cancel out inter-DDL process variation. Measured one-time calibration results are shown in Fig. 6. Calibration was performed by evaluating each DDL and measuring the DDL PD output. After each evaluation, the reference DDL bias bits (eight tuning in top DDL shown in Fig. 2) of the reference was increased and the process repeated. At each bias point, ten additional evaluations were run due to quantify trial-to-trial temporal noise, seen as the slope between TDC levels. No other measurements are averaged in the following discussion. The reference bias point at which the PD of each DDL trips is applied to the tuning bits of the respective bias to align all DDLs, thus, compensating process variation. The average of the tentrials is plotted in Fig. 6. The inter-DDL spread before calibration is approximately 21 reference bias steps, or tuning steps. After calibration, the spread was reduced to less than three tuning steps. The curves are mostly monotonically decreasing which is expected even though there is meta-stability when the phase of the output and reference DDLs is nearly matched. This supports the effectiveness of the proposed time-based MAC methodology.

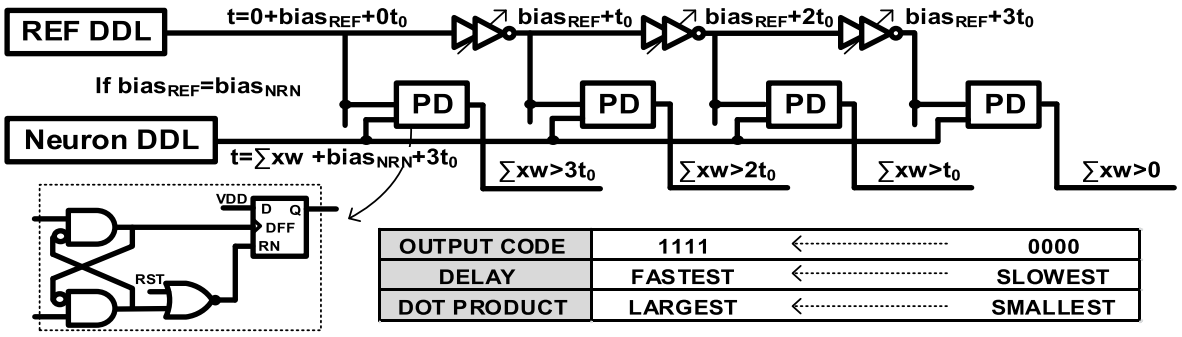


Fig. 5. Timing and details of delay to dot product relationship. Inset: circuit diagram of the PD.

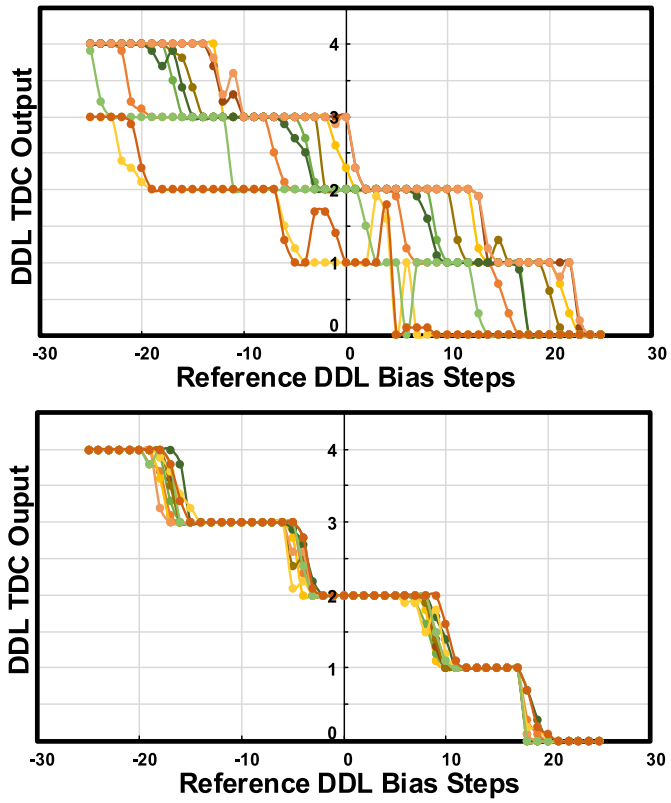


Fig. 6. Measured results from chip calibration. Ten DDL outputs are plotted for a tuning curve before (top) and after (bottom) calibration. Each data point is an average of ten evaluations.

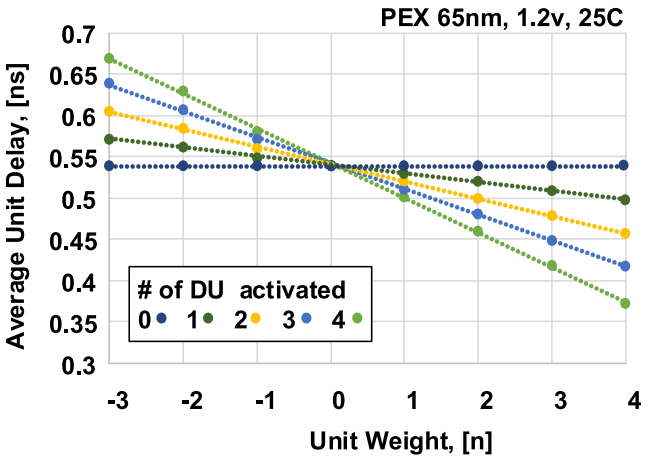


Fig. 7. Postlayout simulated linearity of DU. Average delays of chains of length four with 0, 1, 2, 3, and 4 units enabled are shown. The strength of the delay chain design is the linearity.

Fig. 7 shows the simulated average unit delay as a function of the weight in each unit. In this simulation, the extracted layout of a four-stage DDL was used to measure the delay of a single weight change. Each stage has the weight programmed from $[-3, 4]$, corresponding to the x-axis, and the number of active stages is swept from none to all four, corresponding to different series. This confirms that there are no systematic biases between different weights.

Using both measured and simulated tuning curves, it is possible to quantify the TDC performance [14]. It should be noted that while the TDC performance is important, the trade-offs between area, power, and application performance are paramount. For each bit of the increased TDC resolution, the area and power double. This incentivizes the designer to use a minimalistic design to keep overhead low while still managing to meet the requirements of the application, discussed further in Section IV. The TDC gain is the slope of the output code to input code. The ideal slope is $k_{\text{TDC}} = (1/T_{\text{LSB}})$, where T_{LSB} is the minimum time interval that can be measured, which in this case would be 12 tuning bits. The gain error describes the difference in the last output code to the expected result based on the gain, quantified as $E_{\text{gain}} = (1/T_{\text{LSB}})(T_{1111} - T_{0001}) - (2^N - 2)$ [14]. In this paper, we will estimate $T_{\text{LSB}} = 12$ tuning bits based on the calibrated tuning curve, thus $E_{\text{gain}} = -(1/4)$. Returning to the TDC gain, we can now use the gain error to accurately estimate the actual gain error as $k_{\text{TDC}} = (1/T_{\text{LSB}})(1 - (E_{\text{gain}}/N_{\text{levels}} - 2)) = (3/32)$. This is 12.5% steeper than the ideal gain due to the reduced phase window at output code 0001. This could be due in part to the reduced load seen after the third reference buffer and rectified in the future work by adding a dummy load to better match the delays of each branch of the TDC.

The previous paragraph studied the performance metrics that affect the linear performance. Next, we will attempt to quantify the nonlinear performance due to process variation and noise. The total delay can be described as $t_n = nT + \sum_{i=1}^{n} \varepsilon_i$, where ε_i is the delay error caused by process variation at stage i. In Table I, μ is equivalent to the nT term, where n is the chain length and T is the DU delay. If all the DUs are

TABLE I
DDL DELAY TABLE

PEX 65nm, 1.2V, 25C

t_{fall}			
Chain Length	μ	σ	σ/μ
4	2.25n	34.6p	0.0154
16	9.24n	68.3p	0.0074
128	72.92n	194.6p	0.0027
t_{rise}			
Chain Length	μ	σ	σ/μ
4	2.25n	37.4p	0.0167
16	9.24n	67.1p	0.0073
128	73.00n	200.5p	0.0027

*Estimated via square root law [11]

independent but derived from the same distribution, the standard deviation of the total time is $\sigma(t_n) = \sigma(\varepsilon)\sqrt{n}$. This is supported by Table I, where the chains of lengths 4 and 16 were simulated after parasitic extraction for 100 Monte Carlo samples. The distribution of the delays was normally distributed and the standard deviation follows the square root law. Chain lengths of 128 were estimated by the square root law. This has two consequences; the first being a shorter delay chain will have less variation. This is better; however, it has lower efficacy because there are fewer elements that can be multiplied at once reducing the throughput and increasing power. The second consequence is that the rate of increase decreases as more stages are added to the delay chain. This means that an increase of 8× stages only results in an increase of 2.8× in the standard deviation. In Fig. 7, it is estimated that the tuning step delay is 10.5 ps, which makes the standard deviation equal to roughly 18.5 tuning steps or 1.53 output codes. This could be reduced by increasing the transistor size, W, to reduce the Johnson–Nyquist noise, where in saturation, the power spectral density of the drain current is $S_i = 4kT(2/3)(W/L)\mu C_{ox}(V_{GS} - V_T)$ [15]. Reducing noise comes at a cost of higher power consumption. As voltage increases, it can be seen that the current will increase. These shifts would be seen at the global level since all DDLs share the same voltage supply. It is possible that local variations in the power supply grid could cause deviations in the current which would negatively impact functionality. In addition, increasing temperature will decrease the current. However, the likelihood of a significant temperature gradient across our small, dense array would be unlikely. If there was a global temperature shift, it would affect all DDLs together. Another method could utilize a closed-loop ring oscillator that integrates the noise over multiple cycles which reduces the total error at a cost of lower throughput and higher power per prediction [2]. With these tradeoffs identified, the proposed circuit strikes a balance between performance and a lightweight solution.

IV. DYNAMIC THRESHOLD ERROR CORRECTION

In this design, we opt for a 2-bit TDC due to the optimal tradeoff between small area and LP, and strong architecture

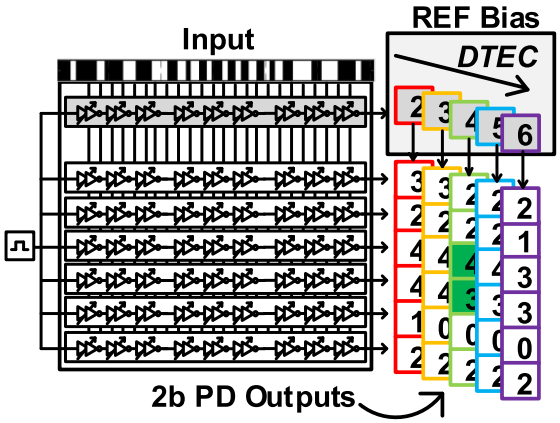


Fig. 8. DTEC concept. Reference DDL bias is increased to elucidate the strongest DDL. The number of steps can be limited based on power and speed requirements.

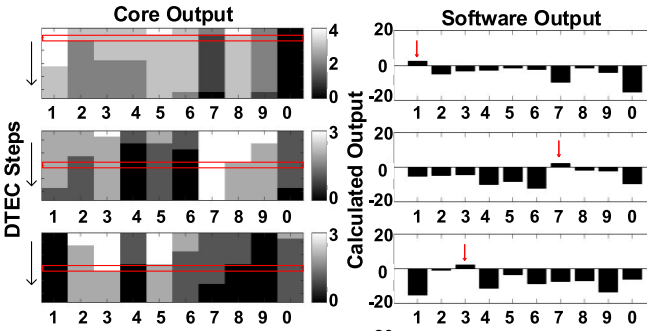


Fig. 9. Colormaps (left) show the outputs from core after DTEC. Each row corresponds to an evaluation from incremented values of the bias. The red rectangle highlights where DTEC has identified the dominant output. Bar plots (right) show the expected results from software.

performance. In networks with "winner-take-all" topologies, such as the last stage of classification networks, ambiguous predictions can occur. Unclear outputs in this paper can stem from limited resolution between PD trip-points or activity outside of the range of the PD. To mitigate this issue, we propose a DTEC technique that increases the effectiveness of the 2-bit TDC. As shown in Fig. 8, when two or more DDLs have the same output, DTEC works by increasing the threshold bias delay which moves the trip point of the PDs. DTEC is dynamic due to the fact that the bias sweep would be terminated after the third evaluation, when the dominant DDL was identified from the PDs. In addition, DTEC can be stopped after a fixed number of steps if no dominant DDL emerges to conserve power. In Fig. 9, the top row of colormaps shows the ambiguous predictions from the core, while successive rows show the output as DTEC is applied. Red rectangles highlight where DTEC has successfully identified the target.

Fig. 10 plots the distributions of the outputs from the intermediate layers in a two-layer dense neural network with 30 hidden units and 10 output units for all 10 000 test images in the MNIST benchmark [16]. The left column corresponds to the output with full precision weights and the right column corresponds to our rounded 3-bit weights. The first row shows the network model used for the analysis. The second row

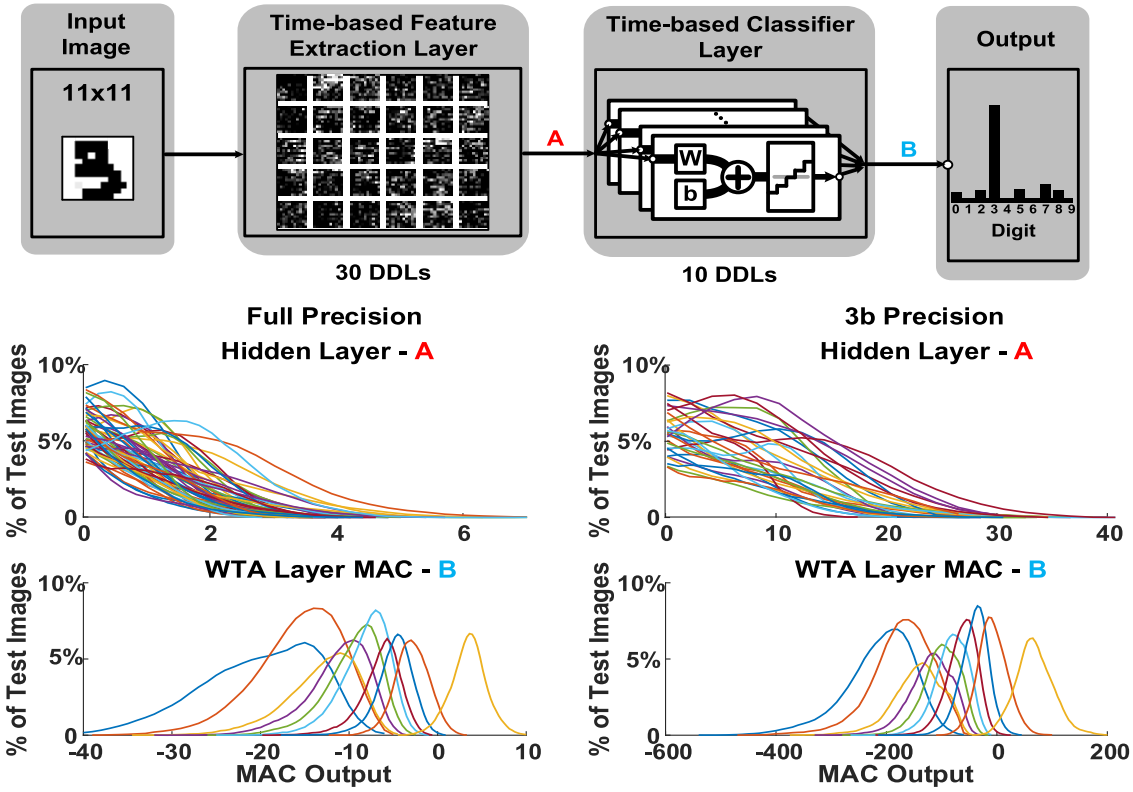


Fig. 10. Distribution of outputs in a two-layer fully connected neural network (top) with 30 hidden units and 10 output units. Full precision weights and right column have three-bit weights (left column). Curves are smoothed to simplify analysis.

displays the distribution of the MAC output of all 30 hidden units after the ReLU transfer function. The third row shows the winner take all (WTA) output of network, but each curve plots the sorted output instead of each unit (i.e., the correct outputs for the ten cases are grouped to one unit). The x-axis scales are not normalized to a unit weight. In the 3b precision network, the full precision weights have been scaled up to match the DU range (i.e., $[-3, 4]$). These curves support the assumption that a 2-bit TDC can cover the entire output range because according to Fig. 6, the width of the PD is 40 units on the x-axis. The hidden layer output would be contained inside that range. In the hidden layer, the results are approximately zero-centered prior to the ReLU activation but have a large range. Units in neural networks must have zero-centered activations, otherwise the predictions would be biased resulting in reduced learning capacity. If the TDC had a unit step of 1 tuning bit (equal to 1 step of the x-axis), this would require at least a 6-bit TDC for each DDL. The area overhead would render this solution infeasible. In addition, due to the effectiveness of the training, the correct prediction output histogram has very little overlap with the remaining predictions. We are able to leverage this outcome because in the majority of the cases, a high precision TDC will not provide additional information when the only relevant outcome is which unit has the highest activation. Another observation is that full precision and fixed point traces match closely. There is a modest amount of spread between the fixed and full precision hidden layer outputs. Nearly, all hardware implementations

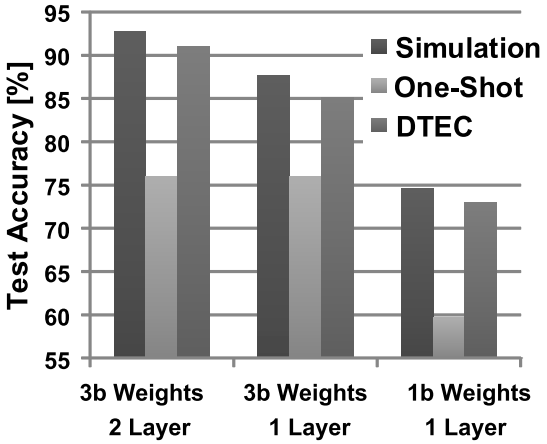


Fig. 11. Results on 11 × 11 MNIST for 3b two layer, 3b single layer, and 1b [8] single layer.

utilize fixed-point weights and this is an acceptable transform as the curves match.

Analysis of results from the 3b single-layer application (Section V) shows that by applying just two DTEC steps 81.64% of the correctible errors are recovered. This comes at a cost of just 41% additional evaluations per image. After the one-shot evaluation, 73% of all images have a dominant output. The remaining 2668 images begin DTEC. After the first step 46% are resolved and 37% after the second step leaving less than 1000 images ambiguous. Thus, 4108 DTEC evaluations improves the total accuracy from 69.16% to

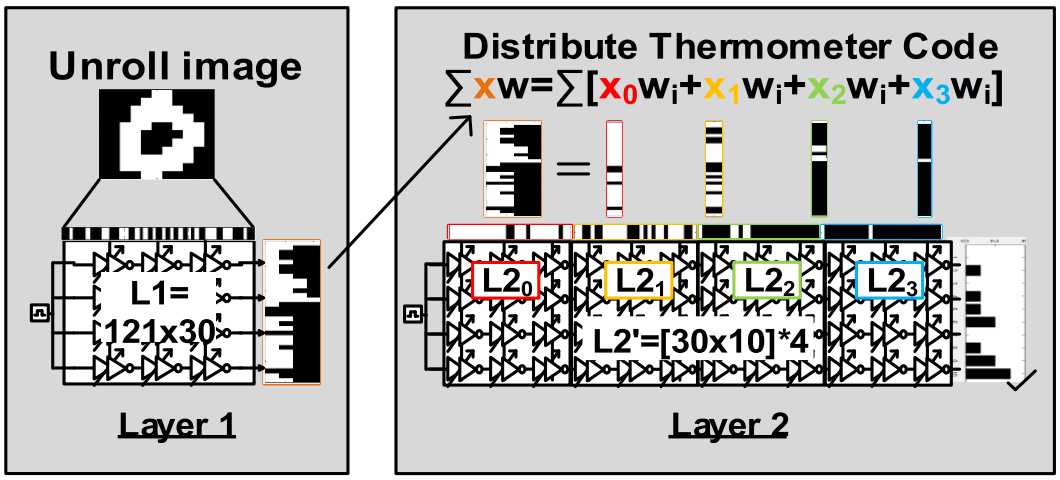


Fig. 12. Dataflow for multi-layer time-based deep neural network demonstrated in this paper.

TABLE II
COMPARISON TABLE

	This Work		A-SSCC'16 [1]	CICC'17 [2]	ISSCC'17 [3]	ISSCC'17 [4]	ISSCC'16 [5]	ISSCC'16[6]	Science'14[7]
Chip Architecture	Time-Based		Time-Based	Time-Based	Digital	Digital	Digital	Sw. Cap	Digital
Algorithm Target	FCDNN & CNN		FCDNN & CNN	FCDNN & CNN	FCDNN & CNN	FCDNN & FFT	CNN	CNN & SGD	FCDNN & CNN
Technology [nm]	65		65	65	28 FDSOI	40	65	40	28
Chip Area [mm^2]	0.644		3.61	0.24	1.87	7.1	12.25	0.012	430
Precision* [b]	[B,T,2,3]		B	3	[4-16]	[6-32]	16	3	[B,T]
On-Chip SRAM [kB]	8.06		20	3	144	270	181.5	[-]	256MB
VDD [V]	1.2 (Nom.)	0.7 (E_{MAx})	1	1.2	0.6	0.65	0.82	1	0.85
Frequency [MHz]	1700	285	23041	792	200	19.3	250	1000	0.001
Energy Efficiency** [TSop/s/W]	36.2	52.4	48.2	2.47	5.0	0.19	.18	3.86	0.04
Hardware Efficiency [GE/PE][1]	38.4		76.5	33.2	7456	18269	50637	288	6.5

*B=Binary, T=Ternary **Synaptic Op=MAC

82.14%. If three DTEC steps are applied, 88.8% of errors can be recovered at an overhead of 51%, demonstrating the dynamic scalability of the technique. Hardware results show that DTEC is an economical and scalable approach to significantly improve application performance.

V. TEST CHIP MEASUREMENT AND APPLICATION DETAILS

We evaluate the core on the MNIST benchmark [16]. Fig. 11 shows the comparison of classification accuracy on an 11 × 11 image for single- and two-layer networks between expected simulated software results, one-shot evaluation, and DTEC. To reduce the 28 × 28 grayscale images to 11 × 11 binary images, 3 pixels are sliced from all four sides of the image. Then, a fixed resizing command is applied, and finally, the pixels are binary thresholded. Fig. 12 shows how the core can be used in a multi-layer deep neural net application. Each bit of the thermometer code is expanded as the input in the next layer. The input is divided into four segments, and the weight matrix is copied four times ($L2_0$–$L2_3$), which gives each bit equal weighting. In the example shown in Fig. 12, 30 neurons in layer 1 yield a 120-bit input to layer 2.

By applying DTEC, the ambiguous results are almost completely recovered and the slight loss in accuracy is due to output differences smaller than a single tuning bit. Fig. 13 shows the tradeoff between power consumption and nominal stage

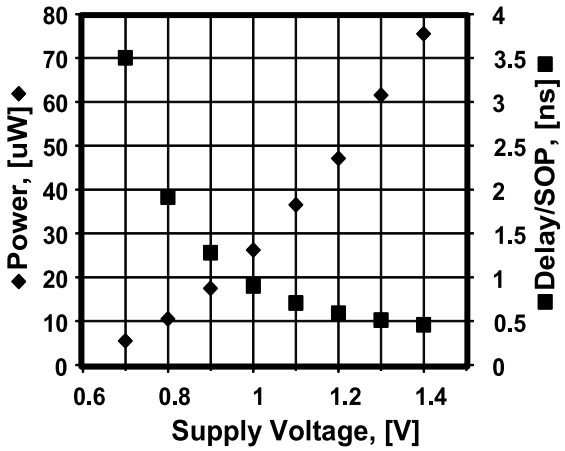
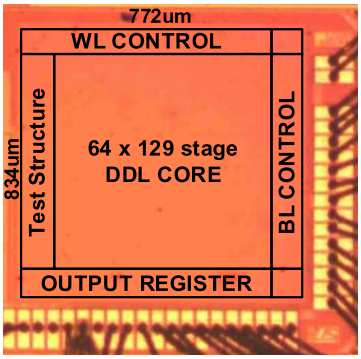


Fig. 13. Power consumption of a DDL and delay/stage versus VDD.

delay for various supply voltages. Power is kept exceptionally low because rarely are more than two stages switching at a time in a DDL. A wide operating voltage range is enabled due to the all-digital time-based design choices. If the design incorporated pipelining, it could achieve even greater throughput. That is, multiple pulses could be pushed into the DDL and the input could shift as well. This is ideally suited for convolutional nets where a weight filter slides across an image. In this case, the image could slide across the weights while input pulses are applied to the DDL. Die photograph and



Process	65nm LP CMOS
Core Area	0.644mm^2
VDD	0.7-1.4V
# of Neurons	64
# of Synapses	8256 (8K)
Throughput	1.7Gpixels/s/DDL
Power	47.1uW/DLL
Energy/SOP	27.6fJ

Fig. 14. Die photograph and chip summary with reported metrics at 1.2-V.

design specs are highlighted in Fig. 14. Table II shows the competitive performance compared with state of the art. All comparisons are made at the highest reported energy efficiency operating point. In our work, we report 1 SOP as a 1b input $\times$ 3b weight MAC without DTEC. Compared to [1] and [17], our energy efficiency would be 3$\times$ higher than reported. In addition, Moons *et al.* [3] reported the peak energy efficiency at 4b with 30%–60% sparsity which they claim is present in the convolutional neural networks. We report very economical energy efficiency although we have tuned the supply voltage to show 8.7% improvement, and modest gate equivalent count for each processing unit coming in at half the size of [1]. The gate efficiency compared is similar compared to [2]. This is interesting because the capacitive weights and binary encoded weights stored in local SRAMs are only slightly smaller than one hot encoded SRAMs, linearly unrolled inverters, and tristate output gates. One hot is less compact but does not require decoding, which causes overhead to control the capacitive connections in [2]. Our chip is scalable in voltage, weight resolution, and is versatile in that it is able to tackle fully connected deep networks as well as convolutional nets.

VI. Conclusion

We described a time-based neuromorphic core based on one-shot DDLs in 65-nm LP CMOS and proposed an error recovery technique, DTEC. It uses inverter delays to compute the dot-product kernel, making it ideally suited for ML applications. The proposed core is validated on the MNIST data set and achieves near simulated prediction accuracy on single- and multi-layer networks after applying our error correction technique, DTEC. The maximum energy efficiency of 54.2TSOPs/s/W with 3b resolution at 0.7-V makes the proposed architecture attractive for edge devices.

References

[1] D. Miyashita, S. Kousai, T. Suzuki, and J. Deguchi, "Time-domain neural network: A 48.5 TSOp/s/W neuromorphic chip optimized for deep learning and CMOS technology," in *Proc. IEEE Asian Solid-State Circuits Conf. (A-SSCC)*, Toyama, Japan, Nov. 2016, pp. 25–28.

[2] M. Liu, L. R. Everson, and C. H. Kim, "A scalable time-based integrate-and-fire neuromorphic core with brain-inspired leak and local lateral inhibition capabilities," in *Proc. IEEE Custom Integr. Circuits Conf. (CICC)*, Austin, TX, USA, Apr./May 2017, pp. 1–4.

[3] B. Moons, R. Uytterhoeven, W. Dehaene, and M. Verhelst, "14.5 Envision: A 0.26-to-10TOPS/W subword-parallel dynamic-voltage-accuracy-frequency-scalable convolutional neural network processor in 28 nm FDSOI," in *IEEE ISSCC Dig. Tech. Papers*, San Francisco, CA, USA, Feb. 2017, pp. 246–247.

[4] S. Bang *et al.*, "14.7 A 288 μW programmable deep-learning processor with 270KB on-chip weight storage using non-uniform memory hierarchy for mobile intelligence," in *IEEE ISSCC Dig. Tech. Papers*, San Francisco, CA, USA, Feb. 2017, pp. 250–251.

[5] Y.-H. Chen, T. Krishna, J. Emer, and V. Sze, "14.5 Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks," in *IEEE ISSCC Dig. Tech. Papers*, San Francisco, CA, USA, Jan./Feb. 2016, pp. 262–263.

[6] E. H. Lee and S. S. Wong, "24.2 A 2.5 GHz 7.7 TOPS/W switched-capacitor matrix multiplier with co-designed local memory in 40 nm," in *IEEE ISSCC Dig. Tech. Papers*, San Francisco, CA, USA, Jan./Feb. 2016, pp. 418–419.

[7] P. A. Merolla *et al.*, "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Science*, vol. 345, no. 6197, pp. 668–673, Aug. 2014.

[8] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks," in *Proc. NIPS*, Barcelona, Spain, 2016, pp. 4107–4115.

[9] A. Biswas and A. P. Chandrakasan, "Conv-RAM: An energy-efficient SRAM with embedded convolution computation for low-power CNN-based machine learning applications," in *IEEE ISSCC Dig. Tech. Papers*, San Francisco, CA, USA, Feb. 2018, pp. 488–490.

[10] S. K. Gonugondla, M. Kang, and N. Shanbhag, "A 42pJ/decision 3.12TOPS/W robust in-memory machine learning classifier with on-chip training," in *IEEE ISSCC Dig. Tech. Papers*, San Francisco, CA, USA, Feb. 2018, pp. 490–492.

[11] L. Ni, Z. Liu, H. Yu, and R. V. Joshi, "An energy-efficient digital ReRAM-crossbar-based CNN with bitwise parallelism," *IEEE J. Explor. Solid-State Computat. Devices Circuits*, vol. 3, pp. 37–46, 2017.

[12] M. Kim *et al.*, "A 68 parallel row access neuromorphic core with 22K multi-level synapses based on logic-compatible embedded flash memory technology," in *IEDM Tech. Dig.*, San Francisco, CA, USA, Dec. 2018, pp. 1–4.

[13] L. R. Everson, M. Lui, N. Pande, and C. H. Kim, "A 104.8 TOPS/W one-shot time-based neuromorphic chip employing dynamic threshold error correction in 65 nm," in *Proc. IEEE Asian Solid-State Circuits Conf. (A-SSCC)*, Tainan, Taiwan, Nov. 2018, pp. 273–276.

[14] S. Henzler, *Time-to-Digital Converters*. New York, NY, USA: Springer, 2010.

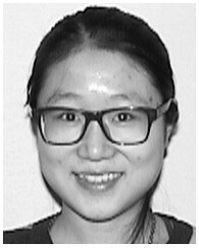
[15] Y. Tsividis, *Operation and Modeling of the MOS Transistor*. Oxford, U.K.: Oxford Univ. Press, 1999.

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

[17] D. Miyashita, S. Kousai, T. Suzuki, and J. Deguchi, "A neuromorphic chip optimized for deep learning and CMOS technology with time-domain analog and digital mixed-signal processing," *IEEE J. Solid-State Circuits*, vol. 52, no. 10, pp. 2679–2689, Oct. 2017.

Luke R. Everson (S'15) received the B.S.E.E. and M.S.E.E. degrees from the University of Minnesota, Minneapolis, MN, USA, in 2015 and 2016, respectively, where he is currently pursuing the Ph.D. degree with VLSI Research Laboratory.

In 2017, he was a Visiting International Scholar with IMEC, Leuven, Belgium, where he applied machine learning to biomedical signals. In 2018, he was an Electrical Engineering Intern with Medtronic Neuromodulation, Research and Core Technology, Fridley, MN, USA, developing an implantable peripheral prosthesis controller. His research focuses on time-based architectures for diverse applications including machine learning, neural signal recording, and graph computing.



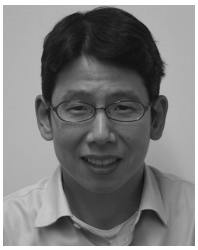
Muqing Liu (S'15) received the B.S. degree in applied physics from Tongji University, Shanghai, China, in 2013, and the M.S. degree in electrical engineering from Columbia University, New York, NY, USA, in 2015. She is currently pursuing the Ph.D. degree in electrical engineering with the University of Minnesota, Minneapolis, MN, USA.

In 2015, she joined the VLSI Research Laboratory, University of Minnesota, with a focus on neuromorphic circuits design and hardware security, such as physical unclonable functions and counterfeit electronics sensor design.

Ms. Liu was a recipient of the Best Paper Award in 2017 ACM/IEEE International Symposium on Low-Power Electronics and Design (ISLPED).

Nakul Pande (S'17) received the B.Tech. degree in electronics and communication engineering from the Institute of Engineering and Technology, Lucknow, India, in 2010, and the M.Tech. degree in electrical engineering from IIT Bombay, Mumbai, India, in 2013. He is currently pursuing the Ph.D. degree in electrical engineering with the University of Minnesota, Minneapolis, MN, USA.

From 2013 to 2015, he worked in the areas of design for testability and memory design with Qualcomm Inc., Bengaluru, India. In 2018, he was an Intern with Cisco Systems, Inc., Bloomington, MN, USA. His current research focuses on the different aspects of integrated circuit reliability, ranging from wearout mechanisms at the device level to interconnect reliability and radiation-induced soft errors.



Chris H. Kim (M'04–SM'10–F'19) received the B.S. and M.S. degrees from Seoul National University, Seoul, South Korea, and the Ph.D. degree from Purdue University, West Lafayette, IN, USA.

In 2004, he joined the University of Minnesota, Minneapolis, MN, USA. His group has expertise in digital, mixed-signal, and memory IC design, with special emphasis on circuit reliability, hardware security, memory circuits, radiation effects, time-based circuits, and beyond-CMOS computing.

Dr. Kim was a recipient of the University of Minnesota Taylor Award for Distinguished Research, the SRC Technical Excellence Award for his Silicon Odometer research, the Council of Graduate Students Outstanding Faculty Award, the NSF CAREER Award, the Mcknight Foundation Land-Grant Professorship, the 3M Non-Tenured Faculty Award, the DAC/ISSCC Student Design Contest Award (two times), the IBM Faculty Partnership Award (three times), the IEEE Circuits and Systems Society Outstanding Young Author Award, the ICCAD Ten Year Retrospective Most Influential Paper Award, the ISLPED Low-Power Design Contest Award (four times), and the ISLPED Best Paper Award (two times).