Open access • Journal Article • DOI:10.1109/TVLSI.2014.2365458

Energy Efficient Approximate Arithmetic for Error Resilient Neuromorphic Computing

Yongtae Kim, Yong Zhang, Peng Li

Institutions: Texas A&M University

Published on: 01 Nov 2015 - IEEE Transactions on Very Large Scale Integration Systems (IEEE)

Topics: Neuromorphic engineering, Word error rate, Energy consumption, Very-large-scale integration and Efficient energy use

Title
Energy efficient approximate arithmetic for error resilient neuromorphic computing

Authors
Kim, Yongtae
Zhang, Yong
Li, Peng

Publication Date
2014

Peer reviewed

Energy Efficient Approximate Arithmetic for Error Resilient Neuromorphic Computing

Yongtae Kim, Yong Zhang, and Peng Li

***Abstract*— This brief proposes a novel design scheme for approximate adders and comparators to significantly reduce energy consumption while maintaining a very low error rate. The considerably improved error rate and critical path delay stem from the employed carry prediction technique that leverages the information from less significant input bits in a parallel manner. The proposed designs have been adopted in a VLSI-based neuromorphic character recognition chip with unsupervised learning implemented on chip. The approximation errors of the proposed arithmetic units have been shown to have negligible impact on the training process while archiving good energy efficiency.**

***Index Terms*— Approximate adder and comparator, carry skip, energy efficiency, error resilience, neuromorphic computing.**

I. Introduction

Approximate computing allows remarkable power and energy savings by relaxing computation accuracy while achieving an acceptable processing quality [1]. The key observation is that many applications, such as digital signal processing (DSP) and neuromorphic systems, have inherent error resilience and hence 100% precision in computation is not required. Particularly, the core of many DSP and neuromorphic applications lies in processing specific kernel functions. For example, spiking neural networks heavily perform the leaky integrate-and-fire (LIF) operation to mimic neuron behavior [2]. Obviously, adders are one primary component for building these arithmetic kernel functions. In addition, comparators are indispensable to determine firing activities in the LIF operation of digital neurons. To this end, it is particularly attractive to design approximate arithmetic units for considerable energy saving in neuromorphic computing.

The approximate adder of [3] leverages a limited number of less significant input bits for carry speculation to increase the overall speed. The critical drawback of this approach is the use of a considerable number of carry generators, which gives rise to large area and high power dissipation. The error tolerant adder I (ETAI) [4] and the lower part OR adder (LOA) [5] are split into an accurate part for higher output bits and an inaccurate part, which utilizes a modified XOR (ETAI) or an OR function (LOA) to approximately compute the remaining lower bits. These approaches are limited by high error rates. The segment-based approximate adders are presented in [6] and [7], which are named the error tolerant adder II (ETAII) and the variable latency carry selection adder 1 (VLCSA-1), respectively. The carry for each k-bit segment is predicted from the lower k-bit inputs to reduce the delay of carry propagation. Similarly, the accuracy-configurable adder (ACA) [8] adopts a number of $2k$-bit sub-adders (SAs) and leverages only the k most significant bit (MSB) outputs of the SAs to achieve approximate additions. Unfortunately, these adders have high error rates for carry generation, particularly for two's complement signed additions of small numbers. In addition, the use of carry selection in VLCSA-1 and middle SAs in ACA results in high-power consumption and area overhead. In [9], the approximation errors for less significant bits are reduced using conditional bounding logic with dithering, which causes area and power overheads. Recently, a general model for approximate array-based arthritic units is presented in [10].

Manuscript received June 3, 2014; revised September 11, 2014; accepted October 22, 2014. Date of publication November 20, 2014; date of current version October 21, 2015.

Y. Kim was with the Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77843 USA. He is now with Intel Corporation, Santa Clara, CA 95054 USA (e-mail: yongtae.kim@intel.com).

Y. Zhang was with the Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77843 USA. He is now with Cadence Design Systems, San Jose, CA 95134 USA (e-mail: zyong@cadence.com).

P. Li is with the Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77843 USA (e-mail: pli@tamu.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/TVLSI.2014.2365458

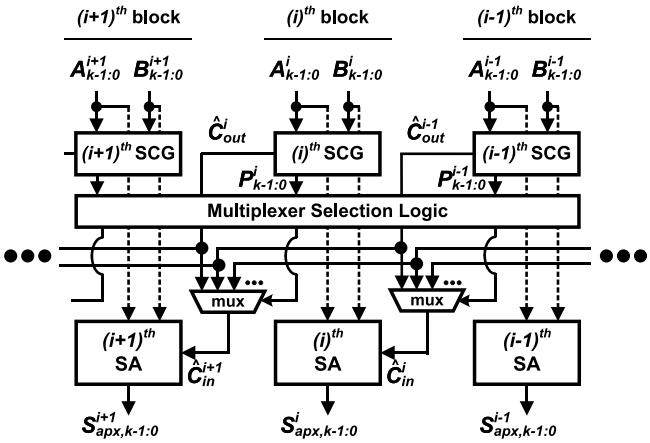

Fig. 1. Block diagram of the proposed approximate adder.

In this brief, we propose a novel approximate adder with a parallel carry-skip scheme. While an approximate adder design has been presented in our preliminary work [11], in this brief, we extend our approximate scheme to comparator design and extensively compare our designs with a large number of existing accurate and approximate adders and comparators, and show the large improvements in area, power, energy, timing, and error rate brought by our design technique. The performance of our approximate units for neurocomputing applications is demonstrated based upon a VLSI-based spiking neural network with over a thousand silicon neurons for character recognition.

II. Proposed Approximate Adder

Our main focuses in the design of approximate arithmetic units include a significant reduction of the error rate by the carry-skip scheme enabling carry speculation in a parallel manner and its application to the adder and comparator design. In particular, the carry for each k-bit segment is predicted by only the lower kv bits and ignoring the rest $n-kv$ bits to reduce the long carry propagation delay. While our preliminary work leverages only $2k$ (i.e., $v = 2$) bits for the carry prediction [11], we generalize the parallel carry-skip scheme by introducing an additional variable v. We adopt the same mathematical notations with [11] to express equations throughout this brief.

Fig. 1 shows the block diagram of the proposed approximate n-bit adder, which is divided into several k-bit sized blocks. Each block contains a k-bit SA and a k-bit sub-carry generator (SCG), which create a partial summation and partial carry-out signal, respectively. Note that the SAs can be implemented by any accurate adders,

such as ripple-carry adder (RCA) and carry-lookahead adder (CLA). At the beginning of an addition operation, all the SCGs simultaneously create the partial carry-out signals $(\ldots, \widehat{C}^{i+1}_{\text{out}}, \widehat{C}^{i}_{\text{out}}, \widehat{C}^{i-1}_{\text{out}}, \ldots)$ using only their k-bit inputs. Then, the SAs' carry-in signals $(\ldots, \widehat{C}^{i+1}_{\text{in}}, \widehat{C}^{i}_{\text{in}}, \widehat{C}^{i-1}_{\text{in}}, \ldots)$ are also concurrently speculated from the υ (≥ 2) preceding k-bit SCGs with a multiplexer. Finally, the SAs work with the speculated carries and produce the partial summations $(\ldots, S^{i+1}_{\text{apx},k-1:0}, S^{i}_{\text{apx},k-1:0}, S^{i-1}_{\text{apx},k-1:0}, \ldots)$. When all the propagate signals of the $(i-1)$th block are true (i.e., $P^{i-1}_{k-1:0} = 1$), the carry-out of a larger number of preceding blocks are required for more accurate carry prediction for the (i)th SA. To obtain the (i)th carry-in $\widehat{C}^{i}_{\text{in}}$, the multiplexer selects $\widehat{C}^{u}_{\text{out}}$, where $i - \upsilon \leq u < i$ if any propagate signal of the (u)th block is false. If all the propagate signals of the υ preceding blocks are true, it chooses $\widehat{C}^{i-\upsilon}_{\text{out}}$. As in Fig. 1, the multiplexer selection logic investigates the group propagate signals $(\ldots, P^{i+1}_{k-1:0}, P^{i}_{k-1:0}, P^{i-1}_{k-1:0}, \ldots)$ generated from the SCGs and decides the carry-in signal of each SA through the multiplexer.

The carry prediction is incorrect if all the propagate signals of more than υ consecutive blocks are true and a carry is generated in the preceding block, making a carry chain of a length greater than $k\upsilon$. Assuming that the adder inputs A and B are bitwise independent, then the propagate and generate signals are bitwise independent as well. We denote the event that the carry-in prediction of the (i)th SA is mistaken due to a carry propagation path of a length between $k\upsilon$ and $k(\upsilon+1)-1$ by E^{i}_{cin}

$$E^{i}_{\text{cin}} = P^{i-1}_{k-1:0} P^{i-2}_{k-1:0} \cdots P^{i-\upsilon}_{k-1:0} G^{i-\upsilon-1}_{k-1:0}. \tag{1}$$

The proposed adder produces an error if any error event E^{i}_{cin} occurs for any of the SAs except for the $\upsilon+1$ least significant ones. Thus, the error rate is expressed by

$$\mathbf{P_{err}}(n,k,\upsilon) = \mathbf{P}\left(E^{\lceil \frac{n}{k} \rceil - 1}_{\text{cin}} + E^{\lceil \frac{n}{k} \rceil - 2}_{\text{cin}} + \cdots + E^{\upsilon+2}_{\text{cin}} + E^{\upsilon+1}_{\text{cin}}\right). \tag{2}$$

By the inclusion–exclusion principle and the independence of the error events [11], the overall error rate of the proposed adder under random inputs is

$$\begin{aligned}
&\mathbf{P_{err}}(n,k,\upsilon) \\
&= \sum_{r=1}^{m-\upsilon-1} (-1)^{r+1} \left(\sum_{\substack{\upsilon < i_1 < \cdots < i_r < m, \\ \forall q: i_q - i_{q-1} > \upsilon}} \mathbf{P}\left(E^{i_r}_{\text{cin}}\right) \cdots \mathbf{P}\left(E^{i_1}_{\text{cin}}\right) \right) \\
&= \sum_{r=1}^{m-\upsilon-1} (-1)^{r+1} \left(\sum_{\substack{\upsilon \leq i_1 < \cdots < i_r < m, \\ \forall q: i_q - i_{q-1} > \upsilon}} \left(\frac{1}{2^{k\upsilon+1}} \left(1 - \frac{1}{2^k}\right)\right)^r \right) \\
&\qquad \text{where} \quad m = \lceil n/k \rceil.
\end{aligned} \tag{3}$$

III. Proposed Approximate Comparator

Basically, a comparator determines the larger of two inputs A and B, and can be implemented using a subtraction. After subtracting two inputs $A - B$, a comparison is readily done by checking the sign bit (i.e., MSB) of the result. In short, $A < B$ when the MSB $= 1$, otherwise $A \geq B$. Note that subtraction is achieved by addition of two's complement (i.e., $A - B = A + \overline{B} + 1$). Thus, we exploit the same idea of the parallel carry-skip scheme to improve the timing and energy efficiency of the comparator.

Fig. 2 illustrates the block diagram of the proposed approximate comparator. The n-bit comparator consists of a 1-bit full adder (FA) and υ (≥ 2) k-bit SCGs that are identical to the ones in the proposed adder. It is worth noting that the proposed approximate comparator exploits only $k\upsilon + 1$ MSBs of the n-bit inputs, resulting in area

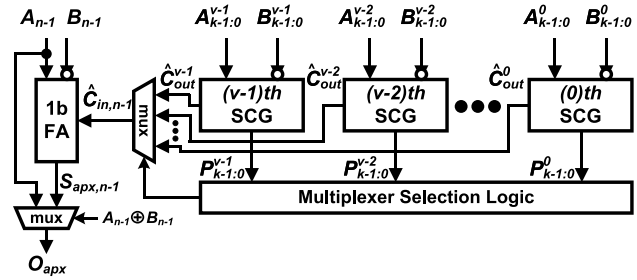

Fig. 2. Block diagram of the proposed approximate comparator.

and power reductions. Importantly, the input B is inverted to achieve subtraction operation. Since implementing two's complement necessitates an additional incrementor, we employ one's complement to further reduce area and energy with sacrificing an error rate, but still achieving a very low error rate. The FA generates the sign bit $S_{\text{apx},n-1}$ (MSB output) of the subtraction between the two inputs by leveraging the speculated carry-in signal $\widehat{C}_{\text{in},n-1}$ and the MSB of the two inputs. The speculated carry-in signal is obtained in the same parallel way by the υ SCGs and multiplexer selection logic. When the two inputs have the different signs (i.e., $A_{n-1} \oplus B_{n-1} = 1$), the comparison result is readily obtained by the input MSB without the FA. Therefore, the output multiplexer selects the MSB of the input A_{n-1} if the signs of two inputs are different from each other, otherwise, it chooses the FA output $S_{\text{apx},n-1}$.

The proposed comparator fails when the signs of the inputs are the same and the carry prediction for the FA is incorrect. The carry speculation is incorrect when all the propagate signals in the υ SCGs used for carry prediction are true and a carry generated from the $n - k\upsilon - 1$ least significant bits (LSBs). It is important to note that, we should consider all the propagate signals of the $n - k\upsilon - 1$ LSBs are true since the proposed comparator adopts one's complement, instead of two's complement, for the subtraction. We assume that the inputs A and B are bitwise independent. Then, the event of the carry prediction error for the FA is given by

$$E_{\text{cin},n-1} = P^{\upsilon-1}_{k-1:0} P^{\upsilon-2}_{k-1:0} \cdots P^{0}_{k-1:0} \left(G_{n-k\upsilon-2:0} + P_{n-k\upsilon-2:0}\right). \tag{4}$$

Then, the overall comparator error rate by the carry-skip scheme under random inputs is

$$\begin{aligned}
\mathbf{P_{err,cmp}}(n,k,\upsilon) &= \mathbf{P}(\overline{A_{n-1} \oplus B_{n-1}})\mathbf{P}(E_{\text{cin},n-1}) \\
&= \frac{1}{2^{k\upsilon+2}} \left(1 + \frac{1}{2^{n-k\upsilon-1}}\right).
\end{aligned} \tag{5}$$

IV. Simulation Results

The proposed approximate arithmetic units were designed in Verilog hardware description language and synthesized with a commercial 90-nm CMOS technology and standard cell library using Synopsys Design Compiler. In addition, the gate-level netlists were translated into transistor-level to perform HSPICE simulations.

Table I summarizes the performance comparison. While our preliminary brief [11] includes six previously presented approximate adder, which are Lu's Adder (LUA) [3], LOA [5], ETAI [4], ETAII [6], VLCSA-1 [7], and ACA [8], we also consider dither approximate adder (DAA) [9] in this brief. The same configurations as in [11] are set to these adders. Collectively, these adders are compared in terms of energy-delay-product (EDP), energy-delay-area product (EDAP), as well as more basic metrics such as area, delay, power, energy, error rate, mean absolute error (MAE), and mean squared error (MSE). Importantly, we use the energy-delay-error rate product (EDERP) to jointly evaluate the energy and error rate of an approximate arithmetic design. While the RCA is more energy

TABLE I
COMPARISON WITH OTHER 16-bit ADDERS

Design	*Area* (μm^2)	*Delay* (ps)	*Power* (mW)	*Energy* (pJ)	*Error Rate* (%)	*MAE*	*MSE*	*EDP* ($pJ \cdot ps$)	*EDAP* ($pJ{\cdot}ps{\cdot}\mu m^2$)	*EDERP* ($pJ{\cdot}ps{\cdot}\%$)
RCA	334	856	0.343	0.294	N/A	N/A	N/A	251	83936	N/A
CLA	514	407	0.922	0.375	N/A	N/A	N/A	155	81613	N/A
LUA	609	234	0.908	0.212	16.68	1363.7	$4.47e^7$	50	30210	827
LOA (8-8)	200	450	0.420	0.189	43.75	111.3	$4.16e^6$	85	16976	3719
ETAI (8-8)	234	435	0.470	0.204	90.00	178.3	$8.34e^6$	89	20711	7980
ETAII	374	254	0.564	0.143	5.86	239.9	$7.86e^6$	36	13583	213
VLCSA-1[1]	673	277	1.337	0.370	5.86	239.9	$7.86e^6$	103	69159	602
ACA[1]	472	374	0.666	0.249	5.86	239.9	$7.86e^6$	93	44010	546
DAA (8-8)	370	435	0.566	0.246	25.00	74.7	$4.19e^6$	107	39509	2671
Proposed[2]	466	359	0.600	0.215	0.18	7.5/14.1[3]	$4.91e^5$/$4.61e^{53}$	77	35959	14

[1] *without EDC* [2] *with the error magnitude reduction [11]* [3] *without the error magnitude reduction*

TABLE II
COMPARISON WITH OTHER 16-bit COMPARATORS

Design	*Area* (μm^2)	*Delay* (ps)	*Power* (mW)	*Energy* (pJ)	*Error Rate* (%)	*EDP* ($pJ \cdot ps$)	*EDAP* ($pJ{\cdot}ps{\cdot}\mu m^2$)	*EDERP* ($pJ \cdot ps \cdot \%$)
RCC	146	834	0.090	0.074	N/A	62	9086	N/A
CLC	423	323	0.258	0.083	N/A	27	11342	N/A
Proposed	121	215	0.111	0.024	0.098	5	620	0.50

and area efficient than CLA, over 2× longer delay degrades its EDP and EDAP. The carry prediction approach in LUA makes the errors to be able to occur in higher significant bits, which leads to the highest MAE and MSE among the approximate adders. The error rate of ETAI (8-8) reaches 90%, which may limit its practical use, due to lack of carry prediction for the accurate part (i.e., the carry is fixed to zero) and worsens EDERP remarkably. On the other hand, because of the simple carry speculation scheme of LOA (8-8), which is achieved by ANDing two MSBs of each operand of the inaccurate part, the error rate is improved to 43.75%, which is still fairly large. The proposed dithering control for the inaccurate part in DAA (8-8) further improves the error rate as well as the MAE/MSE with area and power overheads. Fortunately, ETAI (8-8), LOA (8-8), and DAA (8-8) have fairly low MAE and MSE despite of the high error rates since approximation errors are concentrated on lower significant bits (i.e., inaccurate parts). Among the adders having the same error rate of 5.86%, the ETAII is the most efficient in terms of all the metrics. As a result of the use of carry selection in VLCSA-1, it dissipates the highest power, which is up to 3.9× more than the others and incurs EDP, EDAP, and EDERP degradations. The proposed adder is 2.4× faster and 3.3× EDP efficient than RCA. The carry-skip scheme allows it to have the lowest error rate of 0.18% and EDERP of 14 among the approximate adders. Our design is comparable to ACA with respect to area, delay, power, and energy while having much lower error rate, MAE, MSE, and EDERP because of carry-skip.

From the best of our knowledge, unfortunately, no approximate comparator is presented to date. So, we compare the proposed 16-bit comparator with $k = 4$, $v = 2$ to the two accurate comparators, which are ripple carry based comparators (RCC) and carry lookahead based comparators, respectively. The implemented results are summarized in Table II. The proposed comparator demonstrates the best performance in all the aspects except that it consumes more power than RCC. It is up to 18× more efficient than the other designs in terms of EDP and EDAP with an extremely low error rate (< 0.1%), which is well suitable for error tolerant applications.

V. APPLICATION OF THE PROPOSED ARITHMETIC UNITS IN NEUROMORPHIC COMPUTING

We use the neuromorphic application and its evaluation environment described in [11] to systematically examine the impacts of adder and comparator errors of several designs. While Kim *et al.* [11] take into account only adders, in this brief, we include comparators for the LIF operations to further reduce the energy dissipation. In addition, we extend the application to have over a thousand of silicon neurons for character recognition to accommodate 32×32 pixel input patterns.

A. Impacts of Approximation Errors on the Neuromorphic Application

Fig. 3 shows the input character patterns A to Z for the training and the receptive fields of all excitatory output neurons after the training with the various adders. It is worth noting that the accurate 16-bit comparator (RCC) is used for the digital LIF neurons to compare the threshold voltage with the membrane potentials to generate neurons' firing activities. The corresponding error rates and MAEs during the learning process are listed in Table III. The receptive fields with the accurate adders (RCA and CLA) as in Fig. 3(b) are trained well to respond to the inputs from A to Z. This means that every letter appears once at least in the receptive fields. The results in Fig. 3(b) serves as a golden reference for the approximate adders. The proposed adder has an error rate of merely 0.18% with an MAE of 0.03 for the LIF computations during the training. Fortunately, because of the error resilience of the neuromorphic system, Fig. 3(c) shows that the receptive fields are trained successfully to recognize all the letters and the approximation errors have negligible effect on the training process of the character recognition system. We also test the various approximate adders with the network. All these approximate adders have an error rate of more than 13% during the learning process. As seen in Fig. 3(d)–(j), the approximate adders produce a set of receptive fields with random synaptic weights. These high error rates give rise to failures in training the network since the approximation errors cause the neurons to either fire randomly or cease to fire. In particular, the two's complement signed additions of small numbers frequently occur during leaky operations. In this case, the LUA, ETAII, VLCSA-1, and ACA produce many wrong carry predictions, incurring an error rate of more than 14% and a high MAE over 200 during the learning process and unacceptable performance degradation. This result suggests the carry speculation with only 4-bit of less significant inputs in these 16-bit adders might be insufficient for this application.

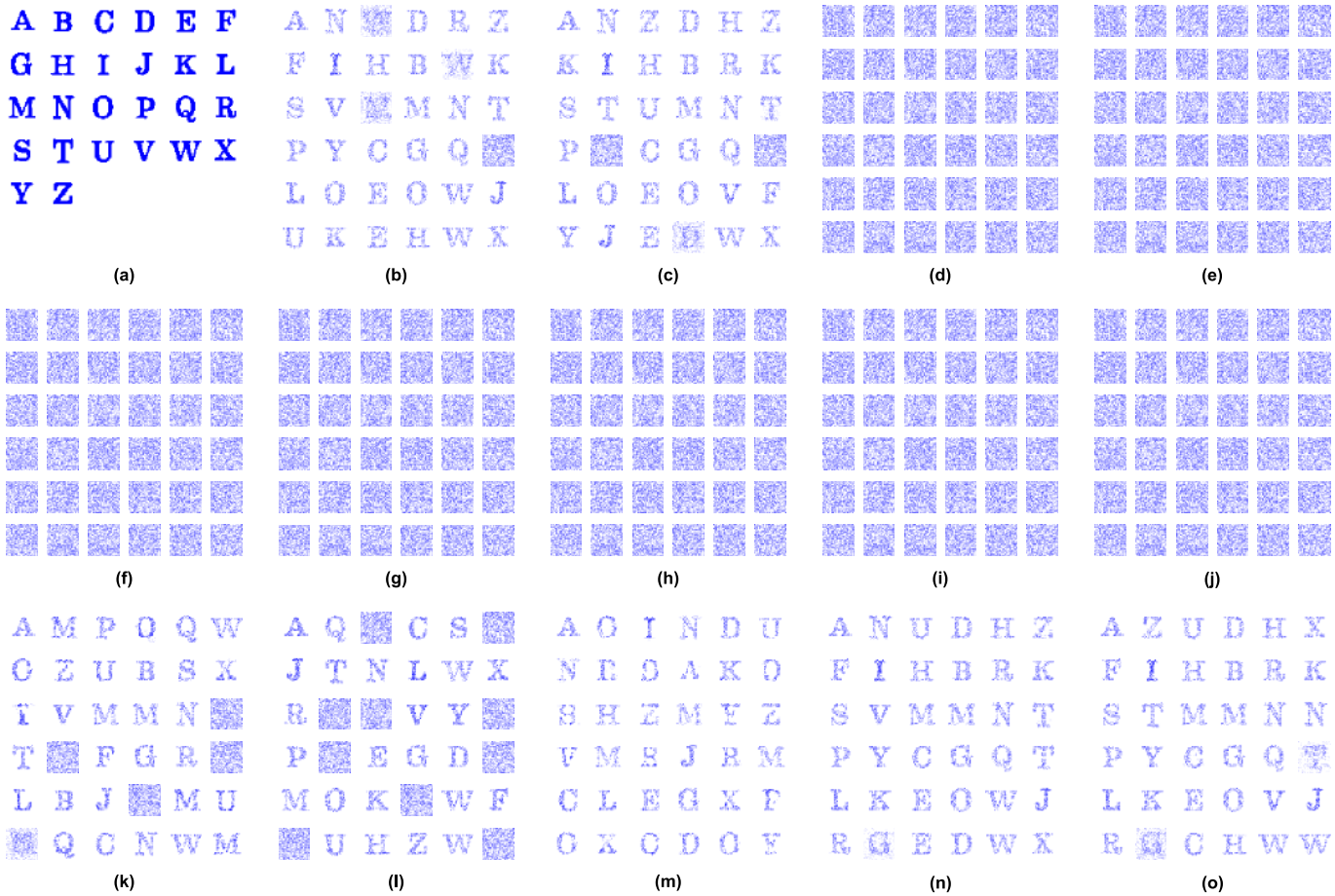

Fig. 3. (a) Input character patterns (32 × 32 pixel) and receptive fields with 16-bit. (b) Accurate adders. (c) Proposed approximate adder. (d) LUA. (e) LOA (8-8). (f) ETAI (8-8). (g) ETAII. (h) VLCSA-1. (i) ACA. (j) DAA (8-8). (k) LOA (13-3). (l) ETAI (15-1). (m) DAA (11-5). (n) Accurate adder with proposed comparator. (o) Proposed adder with proposed comparator.

TABLE III
ERROR RATES AND MAEs OF VARIOUS ADDERS DURING TRAINING PROCESS

Design	*Error Rate (%)*	*MAE*
LUA	14.24	200.47
LOA (8-8)	61.05	11.82
ETAI (8-8)	14.32	8.81
ETAII	14.24	544.11
VLCSA-1	14.24	544.11
ACA	14.24	544.11
DAA (8-8)	13.04	7.68
LOA (13-3)	60.95	1.59
ETAI (15-1)	17.50	0.17
DAA (11-5)	23.25	0.59
Proposed	0.18	0.03

To shed more light on this, we increase the accuracy of LOA, ETAI, and DAA by expanding the accurate part of the adder at the cost of increased delay and energy dissipation. When the LOA, ETAI, and DAA have 13-, 15-, and 11-bit accurate parts, respectively, the network starts to perform better. Although the LOA (13-3) still has a relatively high error rate of 60.95%, the corresponding receptive fields as in Fig. 3(k) are trained such that all alphabets except for D, E, H, I, and K can be identified. Due to the expansion of the accurate part of the adder, the errors now concentrate more on LSBs and the MAE is reduced from 11.82 to 1.59. Similarly, the inclusion of a 15-bit accurate part in ETAI (15-1) allows the network to be trained for all letters except for B and I as illustrated in Fig. 3(l). The DAA (11-5) achieves a relatively low MAE of 0.59 because of the dithering scheme, however, its high error rate hinders the network from training all the letters as seen in Fig. 3(m). Clearly, our design outperforms all other approximate adders.

Next, to see the impacts of the errors of the proposed approximate comparator on the neuromorphic computing, we replace the accurate comparator by the proposed approximate one with $k = 4$ and $v = 2$ in the neuron circuits. Fig. 3(n) is the receptive fields with the accurate adder and the proposed approximate comparator. The proposed comparator allows the trained network to recognize all the letters by its virtue of the extremely low error rate of 0.45% and the error resilience of neuromorphic computing. Additionally, when trained with both our adder and comparator, the network also creates good receptive fields that correspond to all letters as in Fig. 3(o).

B. Energy Efficiency of LIF Neuron With Adders and Comparators in Scaled Supply

Fig. 4 plots the energy comparison of one LIF operation with the neurons with the different adders under scaled power supply levels. The energies are normalized against the neuron with RCA and RCC. Neurons with LUA, ETAII, or VLCSA-1 can operate at a supply voltage of 1 V. The ETAII is the most energy efficient adder design while having a much larger error rate than the proposed adder and

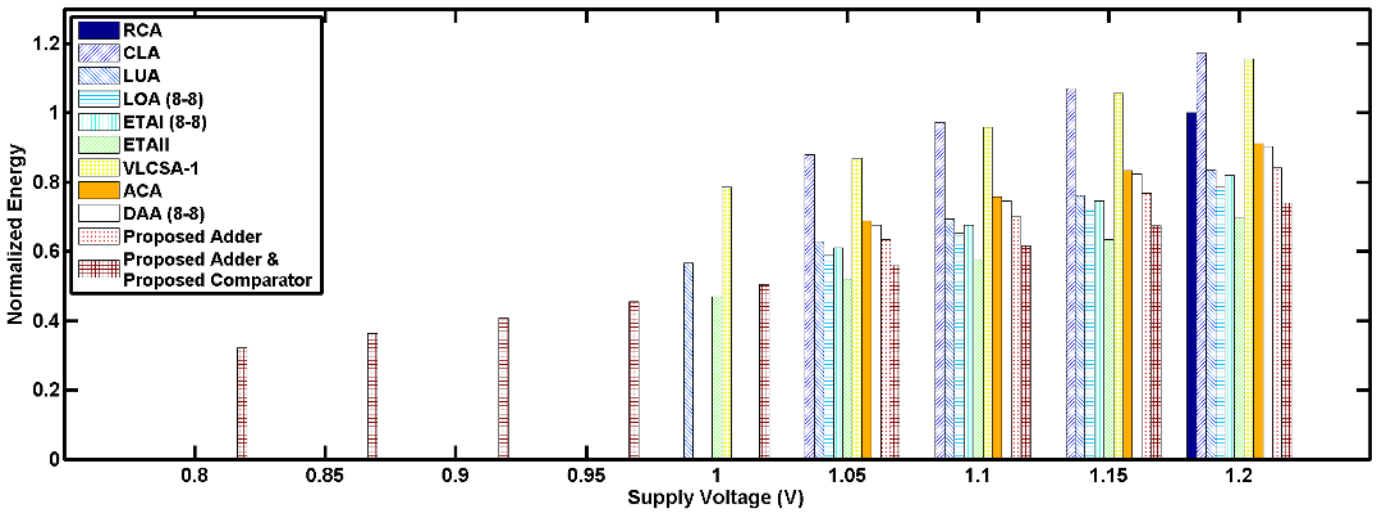

Fig. 4. Normalized energies of one digital LIF neuron with various adders with supply voltage scaling.

leading to poor learning performance [Fig. 3(g)]. Regretfully, the high power consumption from the carry selection in VLCSA-1 is an obstacle to attain energy efficiency. At the supply of 1.05 V, the LUA, ETAI, and proposed adder show the similar energy efficiency and the proposed adder consumes ∼8% and 6% less energy than ACA and DAA (8-8), respectively. Our design achieves the energy savings of up to 36.6% and 27.9% over RCA and CLA, respectively, in the scaled supply. It can be seen that our adder has the most competitive energy and error tradeoff among all these designs. We also compare a neuron leveraging both the proposed adder and comparator to the others. This neuron allows the supply voltage to decrease to 0.8 V. They enable the neuron to be 1.97×, 2.73×, and 3.11× energy efficient over the neuron adopting the proposed adder, CLA and RCA with the accurate comparator RCC, respectively, in the scaled supply without performance degradation [Fig. 3(o)]. Our comparator also provides a great energy saving with very low error rate for the neuromorphic computing.

VI. Conclusion

Novel approximate adder and comparator designs to considerably reduce energy consumption with a very moderate error rate has been presented for energy efficient neuromorphic VLSI systems. The proposed adder is 2.4× faster and 43% more energy efficient over traditional adders and our comparator achieves up to 71% energy saving over the conventional counterparts. We have proven that the approximation errors of our adder and comparator affect the training performance negligibly under an unsupervised learning-based VLSI neuromorphic character recognition chip. Moreover, the proposed approximate units improve energy efficiencies of up to 3.11× over traditional adders and comparators for the digital LIF operation with scaled supply voltage levels. Accordingly, the proposed design approach is applicable to energy efficient neuromorphic computing.

References

[1] R. Hegde and N. R. Shanbhag, "Soft digital signal processing," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 9, no. 6, pp. 813–823, Dec. 2001.

[2] Y. Kim, Y. Zhang, and P. Li, "A digital neuromorphic VLSI architecture with memristor crossbar synaptic array for machine learning," in *Proc. IEEE Int. Syst.-Chip Conf.*, Sep. 2012, pp. 328–333.

[3] S.-L. Lu, "Speeding up processing with approximation circuits," *Computer*, vol. 37, no. 3, pp. 67–73, Mar. 2004.

[4] N. Zhu, W. L. Goh, W. Zhang, K. S. Yeo, and Z. H. Kong, "Design of low-power high-speed truncation-error-tolerant adder and its application in digital signal processing," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 18, no. 8, pp. 1225–1229, Aug. 2010.

[5] H. Mahdiani, A. Ahmadi, S. Fakhraie, and C. Lucas, "Bio-inspired imprecise computational blocks for efficient VLSI implementation of soft-computing applications," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 57, no. 4, pp. 850–862, Apr. 2010.

[6] N. Zhu, W. L. Goh, and K. S. Yeo, "An enhanced low-power high-speed adder for error-tolerant application," in *Proc. 12th Int. Symp. Integr. Circuits*, Dec. 2009, pp. 69–72.

[7] K. Du, P. Varman, and K. Mohanram, "High performance reliable variable latency carry select addition," in *Proc. Design, Autom., Test Eur.*, Mar. 2012, pp. 1257–1262.

[8] A. B. Kahng and S. Kang, "Accuracy-configurable adder for approximate arithmetic designs," in *Proc. 49th IEEE/ACM Design Autom. Conf.*, Jun. 2012, pp. 820–825.

[9] J. Miao, K. He, A. Gerstlauer, and M. Orshansky, "Modeling and synthesis of quality-energy optimal approximate adders," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design*, Nov. 2012, pp. 728–735.

[10] B. Shao and P. Li, "A model for array-based approximate arithmetic computing with application to multiplier and squarer design," in *Proc. Int. Symp. Low Power Electron. Design*, Aug. 2014, pp. 9–14.

[11] Y. Kim, Y. Zhang, and P. Li, "An energy efficient approximate adder with carry skip for error resilient neuromorphic VLSI systems," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design*, Nov. 2013, pp. 130–137.