Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Perspective:

Cross-layer efforts for energy-efficient computing: towards peta operations per second per watt*

Xiaobo Sharon HU‡, Michael NIEMIER

Department of Computer Science and Engineering, University of Notre Dame, Notre Dame, IN 46556, USA

E-mail: shu@nd.edu; mniemier@nd.edu

Received Aug. 5, 2018; Revision accepted Sept. 9, 2018; Crosschecked Oct. 10, 2018

Abstract: As Moore's law based device scaling and accompanying performance scaling trends are slowing down, there is increasing interest in new technologies and computational models for fast and more energy-efficient information processing. Meanwhile, there is growing evidence that, with respect to traditional Boolean circuits and von Neumann processors, it will be challenging for beyond-CMOS devices to compete with the CMOS technology. Exploiting unique characteristics of emerging devices, especially in the context of alternative circuit and architectural paradigms, has the potential to offer orders of magnitude improvement in terms of power, performance, and capability. To take full advantage of beyond-CMOS devices, cross-layer efforts spanning from devices to circuits to architectures to algorithms are indispensable. This study examines energy-efficient neural network accelerators for embedded applications in this context. Several deep neural network accelerator designs based on cross-layer efforts spanning from alternative device technologies, circuit styles, to architectures are highlighted. Application-level benchmarking studies are presented. The discussions demonstrate that cross-layer efforts indeed can lead to orders of magnitude gain towards achieving extreme-scale energy-efficient processing.

Key words: Moore's law; Energy-efficient computing; Neural network accelerators; Beyond-CMOS devices
https://doi.org/10.1631/FITEE.1800466 **CLC number:** TP183; TP302

1 Introduction

It is widely accepted that we are approaching the limit of CMOS technology scaling. This not only applies to the feature size scaling of the transistors but more importantly to performance and power scaling. The golden days of exponential growth in transistor density are over! This trend has long been recognized by the research community as well as funding agencies. Significant research efforts have been devoted to exploring new technologies that may be able to replace CMOS. A number of new device concepts and associated material advances have been introduced. However, according to the benchmarking efforts led by the Semiconductor Research Corporation (SRC) (Nikonov and Young, 2015; Pan and Naeemi, 2017a), the results are not encouraging. While the jury is still out as to whether we will eventually find a device that can put us back to the exponential growth path that we have enjoyed for the past 50 years, none of the beyond-CMOS devices investigated by SRC sponsored research efforts (and others) seem to be able to offer sufficient performance and energy efficiency advantages to serve as drop-in replacements for CMOS.

In contrast to the slow-down of CMOS technology scaling, the demands on information processing capabilities are growing ever faster as we enter the Internet of Things (IoT) and artificial intelligence

‡ Corresponding author

* Project supported by the Center for Low Energy Systems Technology (LEAST), one of the six centers of STARnet, a Semiconductor Research Corporation Program sponsored by MARCO and DARPA

ORCID: Xiaobo Sharon HU, http://orcid.org/0000-0002-6636-9738; Michael NIEMIER, http://orcid.org/0000-0001-7776-4306

© Zhejiang University and Springer-Verlag GmbH Germany, part of Springer Nature 2018

(AI) age. For example, by 2020, the installed base of the IoT devices is forecast to grow to almost 31 billion worldwide, doubling the IoT device (https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/). Application domains for such IoT devices span from smart cities and homes, connected cars, to connected health and wearables. A direct consequence of this rapid increase in IoT devices is the enormous amount of data generated by these devices and the tremendous computation power required. Advances in other technology sectors such as sensing, imaging, and communication similarly lead to accelerated growth in data and their processing demands. For all these and many other applications, energy-efficient, cost-effective, and performance-attentive electronics are the foundation for their growth. Clearly, the slow-down of CMOS scaling can be a big road block.

Note that many new application areas must deal with exponentially growing unstructured data but differ in terms of the specific types of information processing needed. Some applications such as searching and string/text matching can be more memory-intensive, others (such as solving constrained optimization problems) are more computation-intensive, still others such as multi-layer neural networks have regular processing patterns, and others may not have any distinctive features. Such variations can be a double-edged sword. On one hand, homogeneous, many-core processors with a "one-size-fits-all" approach cannot achieve desired energy/cost/performance goals. On the other hand, it opens the possibility of including domain-specific accelerators in processor architectures. Therefore, besides the basic device technology, architectures and algorithms are facing tremendous challenges and opportunities.

Recognizing the above trends, the research community and funding agencies have started concerted efforts to promote vertically integrated research agendas. For example, U.S. Defense Advanced Research Projects Agency (DARPA), in collaboration with SRC, has initiated the Joint University Microelectronics Program (JUMP), which promotes application-domain driven research spanning from materials/devices all the way to applications. The goal of JUMP is "to catalyze innovations for increasing the performance, efficiency, and overall capabilities of broad classes of electronics systems for both commercial and military applications". A high bar has been set by DARPA, i.e., achieving 3200 tera operations per second per watt (TOPS/W) (Salmon, 2017). As another example, the U.S. National Science Foundation (NSF) and SRC are actively supporting new research that brings programmers, system architects, circuit designers, chip processing engineers, material scientists, and computational chemists together to explore new co-design paths towards minimizing the energy impacts of processing, storing, and moving data within future computing systems. These new programs have sparked great interest in the research community to collaboratively tackle the technical challenges due to the slow-down of CMOS technology scaling.

Two frequently asked questions about vertically integrated, cross-layer efforts are (1) how much benefit such approaches can actually offer and (2) whether they can really achieve the energy efficiency goals set by, say DARPA. In this study, we use a representative case study to demonstrate the potential for such vertically integrated, cross-layer efforts in achieving orders of magnitude improvements in energy, performance, and/or cost. The case study focuses on various approaches for designing neural network accelerators for solving the handwritten recognition problem based on the MNIST dataset (LeCun et al., 1998). The case study demonstrates how a co-design effort that spans from algorithms and architectures to circuits and devices can move us closer to the 3200 TOPS/W goal.

The rest of the paper is organized as follows. We first articulate the needs for cross-layer design by examining the state of the art benchmarking data of beyond-CMOS devices, and discuss why such design approaches can be beneficial. We then present a case study on using the cellular neural network (CeNN) computing model to implement convolutional neural networks (CoNNs). The case study demonstrates how device, circuit, architecture, and algorithm level efforts can be judiciously combined to achieve orders of magnitude improvements in energy and energy efficiency. Last, we comment on the path forward and key research.

2 Need for cross-layer efforts

There have been significant research efforts targeting beyond-CMOS technologies to build

devices to address the challenges brought upon by the CMOS scaling limit (Ionescu and Riel, 2011) (http://least.nd.edu). These devices vary in terms of state variable representation (e.g., spin or charge), physical operating principles (e.g., magnetic coupling or tunneling), materials used, etc. To quantify how these new devices might perform in the context of basic functions such as an inverter with fan-out-of-4 and 32-bit adder, Nikonov and Young at Intel organized comprehensive, device-level benchmarking efforts (Nikonov and Young, 2013, 2015). The benchmarking work (referred to as BCBv3 for beyond-CMOS benchmarking version 3) introduces an analytical methodology by which all device parameters would be derived from the same uniform assumptions (e.g., lithography-enabled feature sizes). Fig. 1 extracted from Aziz et al. (2018) is a sample output of BCBv3, showing the 32-bit ripple carry adder (RCA)'s dynamic switching energy vs. delay of CMOS and various beyond-CMOS device technologies including tunnel and ferroelectric transistors, magnetoelectric and spin torque devices. The benchmarking data suggests that only a few devices (i.e., a subset of tunnel transistors) could outperform (by less than one order of magnitude) a high-performance CMOS (CMOS HP) or low-power CMOS (CMOS LV) slated for 2018.

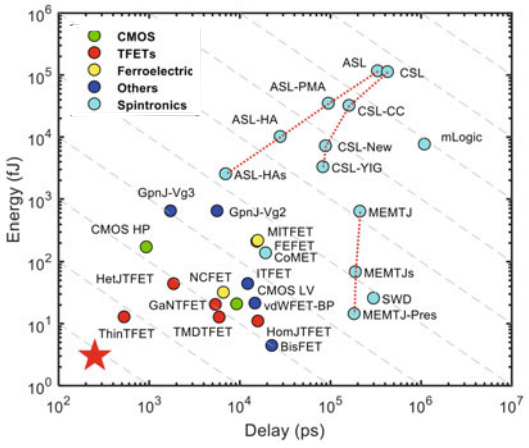


Fig. 1 Energy vs. delay of a 32-bit adder for representative beyond-CMOS spin- and charge-based devices (Reprinted from Aziz et al. (2018), Copyright 2018, with permission from IEEE). The two green dots correspond to 15-nm CMOS high-performance and low-voltage devices (References to color refer to the online version of this figure)

One question one may raise is how these devices would fare in multi-core processors that are widely used today. To project the performance of multi-core processors based on beyond-CMOS devices, we introduced an analytical architectural-level benchmarking model for beyond-CMOS devices (Perricone et al., 2016). Our model unifies architectural-level benchmarking (Esmaeilzadeh et al., 2011, 2013) with device-level benchmarking (Nikonov and Young, 2013, 2015) to provide insight as to how these devices could sustain Moore's law performance scaling trends. Fig. 2 shows the speedup results of four tunnel transistors and two CMOS (15-nm CMOS HP and CMOS LV) transistors relative to 45-nm CMOS HP transistors. The data are for the Mediabench and PARSEC benchmark suites, and all processors are subject to 5 W thermal design power (TDP). The results indicate that for highly parallel programs such as swaptions and JPEG, speedups of 13X and 7.5X can be expected. Though these speedups are more encouraging, they are still not enough for justifying a technology as a CMOS replacement. Furthermore, for less parallel programs, speedups are significantly lower (all less than 4X).

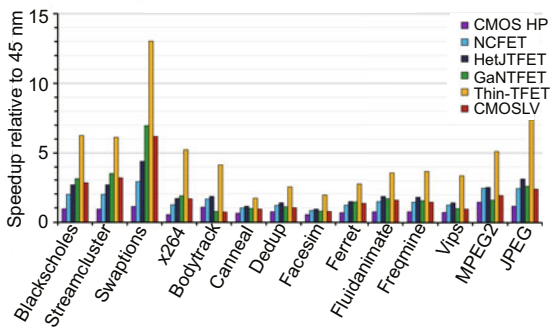


Fig. 2 Speedups relative to 45-nm CMOS for 15-nm CMOS and steep slope devices with a low (5 W) TDP. Results include two Mediabench benchmarks: MPEG2 and JPEG encoding (Reprinted from Perricone et al. (2016), Copyright 2016, with permission from the Association for Computing Machinery)

Even though CMOS scaling is slowing down and beyond-CMOS devices seem to be ready to replace CMOS, the demand for lower power, higher performance, and more cost-effective information processing has not diminished but rather is accelerating. This is fueled mostly by many emerging applications, such as IoT, AI, and autonomous driving. In a recent program

call from DARPA (https://www.darpa.mil/about-us/timeline/jump), Linton Salmon outlined the upside goal that DARPA has set for its performers (Fig. 3), i.e., 3200 TOPS/W or 0.3 fJ/op. If we consider 32-bit add as the operation, according to the benchmarking results from BCBv3, the lowest energy emerging device, BISFET, achieves about 0.6 fJ/op but its delay is much larger than that of CMOS devices. Note that the energy per operation is typically much higher when executing real programs due to overhead in memory, etc.

It is clear that none of the beyond-CMOS devices being investigated can offer compelling improvements over CMOS devices if used as a pure drop-in replacement. However, one key observation on beyond-CMOS devices sheds light on possible directions in reaching the goals set by DARPA and many applications. Specifically, many beyond-CMOS devices exhibit fundamentally different and unique characteristics that if exploited properly could offer significant performance and/or energy benefits. This direction is particularly powerful with the rise of domain-specific accelerators being deployed in many applications.

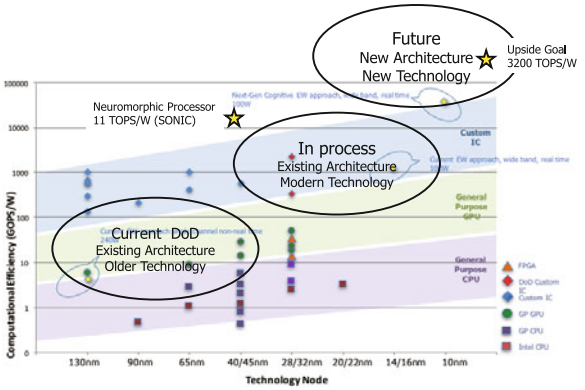


Fig. 3 DARPA's needs for future generations of systems on chip (Salmon, 2017)

As one example, a cellular neural network (CeNN) kernel has been benchmarked to implement in both CMOS and a number of beyond-CMOS devices (Pan and Naeemi, 2017b). CeNNs can be effectively exploited in applications such as associative memory and image processing (Szolgay et al., 1997; Scheutz et al., 2004; Horváth et al., 2017; Xu et al., 2017). Both digital and analog CeNN circuits can be realized with transistors, while spin-based beyond-CMOS devices such as magnetic tunneling junction (MTJs) can also be used to implement the CeNN function. Fig. 4 summarizes the benchmarking results considering all three types of designs: digital, analog, and spintronic. It can be readily seen that devices that do not fare as well for Boolean logic and simple arithmetic operations can be orders of magnitude better than CMOS equivalents in terms of energy delay product (EDP). Most TFET-based CeNNs consume less energy due to their steep threshold slope, low supply voltage, and superior analog figures of merits (FoMs) (Sedighi et al., 2015). CeNNs implemented by digital CMOS perform worse compared to their analog counterpart due to the large energy and delay from multiplying and adding synapse weights. Spintronic devices seem to offer significant advantages in implementing CeNN.

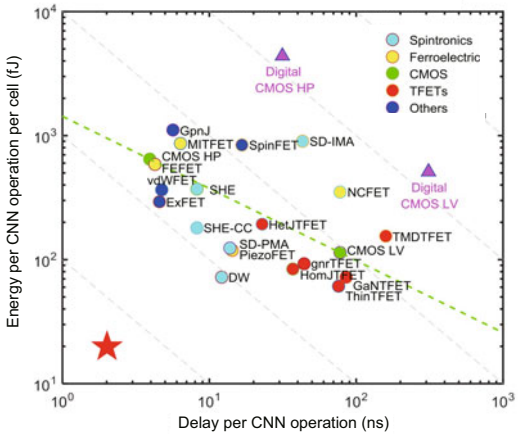


Fig. 4 Energy vs. delay for various beyond-CMOS technologies. Triangle and circular points of charge-based devices represent the digital and analog CeNN implementations, respectively. For the text labels of spintronic CeNN implementation, SD, SHE, and DW stand for spin diffusion, spin Hall effect, and domain wall motion, respectively, and CC represents the copper collector (Reprinted from Pan and Naeemi (2017b), Copyright 2017, with permission from IEEE) (References to color refer to the online version of this figure)

As another example, ferroelectric FETs (FeFET) (beyond-CMOS devices being studied actively by both industry and academia) exhibit hysteresis in their drain current vs. gate voltage (Salahuddin and Datta, 2008; George et al., 2016b) as shown in Fig. 5. The property that a single transistor can be used as both a switch and a storage element opens the door for energy-efficient realization of many interesting fine-grained logic-in-memory concepts (George

et al., 2016a; Yin et al., 2016a, 2017; Chen et al., 2018). Different architectures and applications (e.g., in-memory computing for big-data applications) can benefit greatly from such logic-in-memory modules.

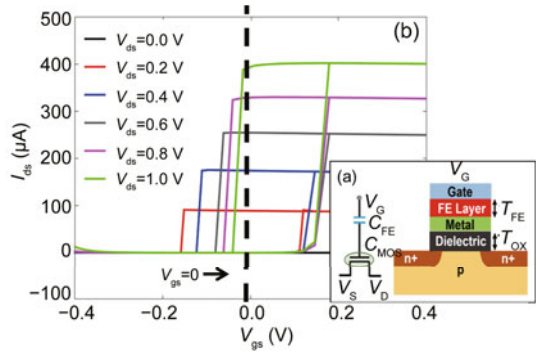


Fig. 5 FeFET structure and its equivalent circuit representation showing ferroelectric capacitance and the capacitance of the underlying MOSFET (a) and FeFET *I-V* curves with tunable hysteresis (b) (Reprinted from Yin et al. (2016a), Copyright 2016, with permission from IEEE) (References to color refer to the online version of this figure)

The above examples show that beyond-CMOS devices may be much more competitive than CMOS devices if their unique characteristics are effectively exploited in circuit and/or architectural designs. Such cross-layer efforts spanning from devices and circuits to architectures and applications are particularly valuable for developing domain-specific accelerators which are effective in tackling energy concerns.

Conducting cross-layer research from devices all the way up to applications, though straightforward conceptually, is challenging in practice. Some of the critical questions to be addressed include how to evaluate and compare different designs (e.g., is an analog convolutional neural network (CoNN) accelerator better than a digital one and over what figures of merit?), what benchmark suites should be used to evaluate designs, and how to assess the contribution from each layer. In the remainder of this study, we use a case study to show our initial attempts to answer the above questions.

3 CeNN for CoNN: a case study on device-circuit-architecture-algorithm co-design

In this section, we discuss the development and evaluation of a CeNN-friendly CoNN for solving the MNIST digit recognition problem. This study uses features of beyond-CMOS devices to construct low-power analog CeNN cells (device-to-circuit effort). A CeNN-friendly CoNN algorithm and a corresponding CeNN array based architecture are developed to leverage the low-power CeNN cells (algorithm to architecture to circuit effort). Evaluation of various design alternatives and comparison with existing work on solving the MNIST digital recognition problem are discussed to analyze the benefits of such cross-layer efforts. Below, we start with a brief introduction of the CeNN and tunnel FETs, a class of beyond-CMOS transistors. We then elaborate our cross-layer design effort on designing CeNN-friendly CoNNs. Last, we compare our designs with other accelerators for solving the MNIST problem. The key rationale of picking the MNIST problem as the target is the availability of a large number of existing implementations.

3.1 Cellular neural network and tunnel FET basics

A spatially invariant CeNN array (Chua and Roska, 2002) consists of an $M \times N$ array of identical cells (Fig. 6a). Each cell C_{ij}, $(i,j) \in \{1,2,\ldots,M\} \times \{1,2,\ldots,N\}$, has identical connections with adjacent cells in a predefined neighborhood of radius $r>0$. The size of the neighborhood is $m=(2r+1)^2$. A conventional analog CeNN cell consists of one resistor, one capacitor, $2m$ linear voltage controlled current sources (VCCSs), one fixed current source, and one specific type of non-linear voltage controlled voltage source (Fig. 6b). The input, state, and output of a given cell C_{ij} correspond to the nodal voltages u_{ij}, x_{ij}, and y_{ij}, respectively. VCCSs controlled by the input and output voltages of each neighbor deliver feedforward and feedback currents to a given cell, respectively. The dynamics of a CeNN are captured by a system of $M \times N$ ordinary differential equations, each of which simply follows the Kirchhoff current law (KCL) at the state nodes of the corresponding cells per Eq. (1):

$$C\frac{\mathrm{d}x_{ij}(t)}{\mathrm{d}t} = -\frac{x_{ij}(t)}{R} + \sum_{c_{kl}\in N_r(i,j)} a_{ij,kl}y_{kl}(t) + \sum_{C_{kl}\in N_r(i,j)} b_{ij,kl}u_{kl} + Z. \quad (1)$$

CeNN cells typically employ a non-linear

sigmoid-like transfer function at the output to ensure fixed binary output levels (Chua and Yang, 1988). The parameters $a_{ij,kl}$ and $b_{ij,kl}$ serve as weights for the feedback and feedforward currents from cell C_{kl} to cell C_{ij}, respectively. $a_{ij,kl}$ and $b_{ij,kl}$ are space-invariant and are denoted by two $(2r+1)\times(2r+1)$ matrices (If $r=1$, they are captured by 3×3 matrices). The matrices of a and b parameters are typically referred to as feedback template $\boldsymbol{A}$ and feedforward template $\boldsymbol{B}$, respectively. Design flexibility is further enhanced by the fixed bias current Z that provides a means to adjust the total current flowing into a cell. A CeNN can solve a wide range of image processing problems by carefully selecting the values of the $\boldsymbol{A}$ and $\boldsymbol{B}$ templates (as well as Z).

The most costly part of CeNN hardware is the VCCSs. Various circuits including inverters, Gilbert multipliers, and operational transconductance amplifiers (OTAs) (Wang et al., 1998; Molinar-Solis et al., 2007) can be used to realize VCCSs. For work to be discussed in this study, we use the OTA design from Lou et al. (2015) for both CMOS and beyond-CMOS implementations. OTAs provide a large linear range for voltage to current conversion, and can implement a wide range of transconductances allowing for different CeNN templates.

The specific example of beyond-CMOS devices that we will consider is tunnel FET (TFET). TFET promises to offer sub-threshold swing below 60 mV/dec and can operate at very low voltages (Seabaugh and Zhang, 2010; Kam et al., 2012). For example, TFETs have shown to be extremely power-efficient for logic circuits (Khatami and Banerjee, 2009). Previous research also demonstrated the use of TFETs in efficiently designing analog circuits such as DRAM, amplifier, and ADC (Liu et al., 2014; Sedighi et al., 2015). Furthermore, TFETs conduct current through band-to-band tunneling which has very weak sensitivity to temperature variation (Seabaugh and Zhang, 2010). As a result, their operations are robust in the presence of thermal noise (an important design consideration for analog circuits). Fig. 7 shows the I-V characteristics of two example III-V TFET devices: InAs homojunction TFET (HomTFET) and GaSb-InAs heterojunction TFET (HetTFET) (Avci et al., 2011; Zhou et al., 2012).

In Lou et al. (2015), we introduced TFET-based CeNN cell designs to reduce VCCS overhead in general and to realize nonlinear template operations in an efficient manner. A nonlinear template uses a function instead of a constant to define template values (i.e., the values of $a_{ij,kl}$ and $b_{ij,kl}$). Nonlinear CeNN templates allow some important functions to be fullfilled much more economically compared with linear templates. For example, with a nonlinear template, finding the maximum (or minimum) value among all the CeNN cells, i.e., GLOBMAX (or GLOBMIN), takes only one single CeNN step, while it would take 15 steps if linear templates are used! TFETs demonstrate small drain currents when $V_{\mathrm{ds}}<0$, which is helpful for realizing non-linear functionality. By combining the use of a nonlinear template and TFET devices, improvements of up to 2.5X and 6X in energy dissipation are observed for a TFET-based nonlinear solution when compared to a CMOS-based equivalent or TFET-based linear solution, respectively. Besides energy and delay improvements, TFET-based designs lead to smaller errors (Lou et al., 2015). Fig. 8 shows a TFET-based linear and nonlinear OTA design for realizing linear and nonlinear templates, respectively. The actual template value is simply G_mR, where R is the in-cell resistor and is fixed, and thus the G_m of the OTA is used to specify/realize a convolution template value per the discussion in the next subsection. By tuning the G_m of the OTA, we can realize different template values.

3.2 Realizing CoNN operations with CeNNs

The previous subsection briefly reviews how beyond-CMOS devices, specifically TFETs, can benefit analog CeNN cell designs (i.e., through device and circuit efforts). This subsection focuses on how CeNNs can be exploited to implement key operations found in CoNNs (i.e., circuit and architecture efforts). At a very high level, both CoNNs and CeNNs are capable of replicating certain aspects of human vision such as alternating simple/complex cells of the V1 area of the primary visual cortex and the spatial response of these cells (similar to alternating layers of pooling and convolution operations). Given that CeNNs (1) operate in the analog domain, which could result in lower power consumption/improved energy efficiency (Kim et al., 2008), (2) are Turing complete (Chua and Roska, 2002), and (3) can provide a richer library of functionality than typically associated with CoNNs, the topographic, highly

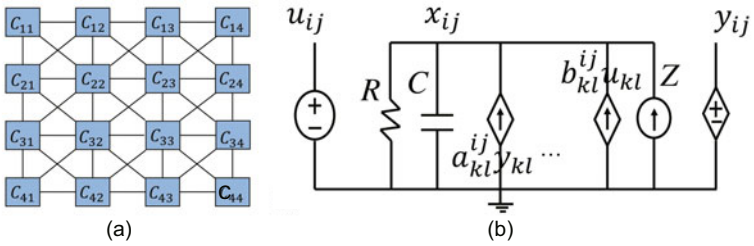


Fig. 6 CeNN architecture (a) and circuitry in a CeNN cell (b)

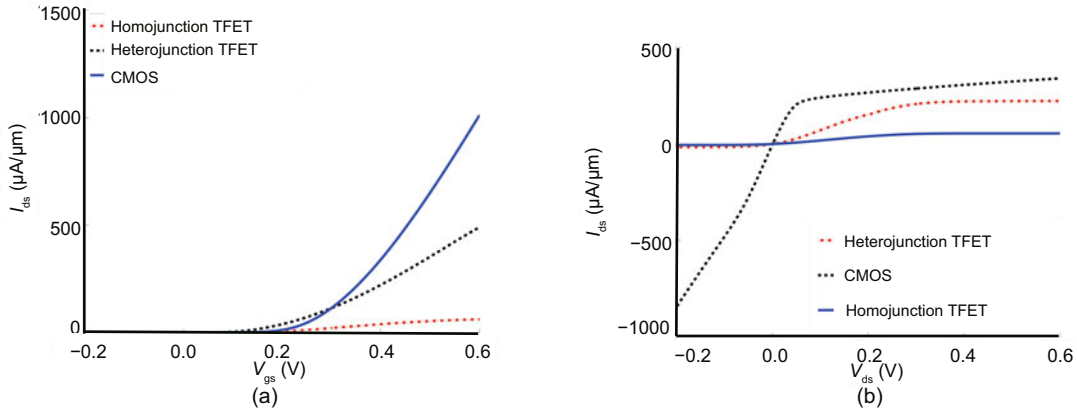


Fig. 7 I_{ds} versus V_{gs} when V_{ds}=0.6 V (a) and I_{ds} versus V_{ds} when V_{gs}=0.4 V (b)

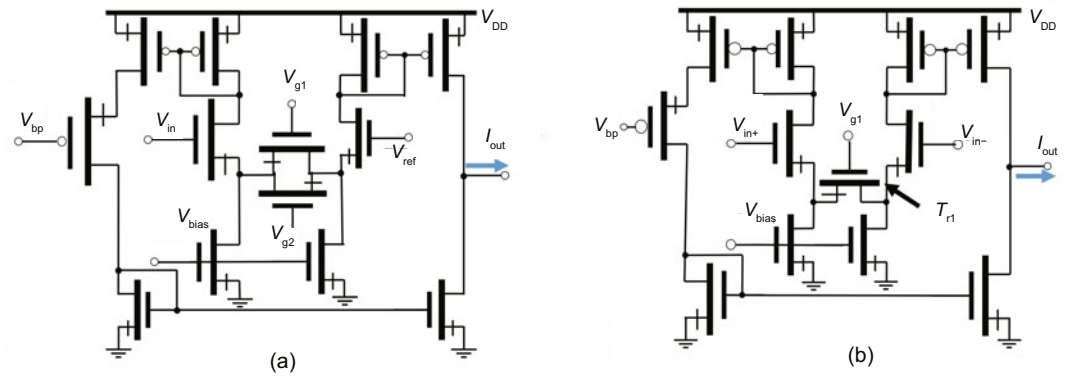


Fig. 8 TFET-based linear circuit (a) and TFET-based nonlinear circuit (b) (Reprinted from Lou et al. (2015), Copyright 2015, with permission from the Association for Computing Machinery)

parallel CeNN architecture has great potential in efficiently implementing operations in deep neural networks/CoNNs.

CeNNs are typically composed of a single layer of processing elements (PEs). While most CeNN hardware implementations lack the layered structure of CoNNs, by using local memory (commonly available on every realized CeNN chip), a cascade of said operations can be realized by re-using the result of each previous processing layer (Chua and Roska, 2002). Alternatively, one could simply build a cascade of CeNNs to realize the layered structure of a CoNN. This lends itself well to deep learning algorithms, which are built as cascades of different layers of non-linear processing elements, where every layer implements a simple operation that might include: (1) convolution, (2) non-linear operations (usually a rectifier), (3) pooling operations, and (4)

fully connected layers (although sometimes support vector machines (SVMs) are used instead). Below we sketch how the key operations in CoNN can be realized by CeNN operations. For more details, readers are referred to Horváth et al. (2017).

Convolution layers in CoNN are used to detect and extract different feature maps on input data as the summation of the point-wise multiplication of two feature maps. One map is the input image f, and the other map, often referred to as kernel, encodes a desired feature (g) to be detected by some operation. The convolution operation (the key element in deep learning architectures) is defined as

$$[f * g](i,j) = \sum_{k,l=-\infty}^{\infty} f(i-k, j-l)g(k,l). \quad (2)$$

In deep learning architectures, convolution serves as a simple change detector. These are the only layers with parameters that are changed online, and the exact convolutional kernels are optimized during training. As can be seen from Eq. (1), by applying the feedforward template (denoted as $b_{ij,kl}$), CeNNs can implement convolutional kernels in a straightforward manner.

As CoNNs are built for recognition and classification tasks, non-linear operations are required. One of the most commonly used non-linearity in deep learning architectures (Dahl et al., 2013) is the rectified linear unit (ReLU) as shown in Eq. (3), which thresholds every value below zero:

$$R(x) = \begin{cases} 0, & x \le 0, \\ x, & x > 0. \end{cases} \quad (3)$$

In a CeNN, the ReLU operation can be implemented using a nonlinear template. The template implementing the nonlinear function of the ReLU operation can be written as $D(x_{i,j}) = \max\{0, x_{i,j}\}$, which sets all negative values to zero and leaves the positive values unchanged; thus, it directly implements Eq. (3). When considering actual hardware implementations, to date, real hardware such as the ACE16k chip (Rodriguez-Vázquez et al., 2004) or the SPS 02 Smart Photosensor from Toshiba (http://www.toshiba-teli.co.jp/en/products/industrial/sps/sps.htm) applies standard CeNN nonlinearity on the cells:

$$y_{k,l} = \frac{1}{2}|x_{k,l}+1| - \frac{1}{2}|x_{k,l}-1|. \quad (4)$$

The ReLU operation can also be rewritten as two linear operations by applying the templates given below:

$$\boldsymbol{B}_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, Z = -1. \quad (5)$$

$$\boldsymbol{B}_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, Z = 1. \quad (6)$$

We will compare these two designs later.

Pooling operations are employed to decrease the amount of information between consecutive layers in a deep neural network to compensate for the effects of small translations. Max pooling operation selects the maximum element in a region around every value, i.e.,

$$P(i,j) = \max_{k,l \in S} f(i-k, j-l). \quad (7)$$

This max pooling operation can be implemented by the nonlinear, GLOBMAX template, which can be found in the standard CeNN template library (http://cnn-technology.itk.ppke.hu/Template_library_v3.1.pdf). The GLOBMAX operation selects the maximum value in the neighborhood of a cell in a CeNN array and propagates it through the array. By setting the execution time of the template accordingly, one can easily set how far the maximum values can propagate and which regions the maximum values can fill. This nonlinear template can also be implemented by using the non-linear function as given in Eq. (8):

$$D(x_{i,j}) = \begin{cases} -x/8, & x \le 0, \\ 0, & x > 0. \end{cases} \quad (8)$$

As before, the max pooling operation can be realized with a sequence of linear operations. However, in this case, a total of 16 CeNN steps are needed (Horváth et al., 2017), which can be quite expensive. An alternative to max pooling is average pooling, which can be easily realized with CeNNs. Specifically, to perform an average pooling operation in a 2×2 grid, one can simply employ the following $\boldsymbol{B}$ templates ($Z = 0$):

$$\boldsymbol{B}_{2\times 2} = \begin{bmatrix} 1/4 & 1/4 & 0 \\ 1/4 & 1/4 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \quad (9)$$

To accomplish classification in a CoNN, one needs to convert the feature maps extracted from previous layers into a scalar, index value associated with the selected class. A common approach for classification is to employ a fully connected (FC) layer and associated neurons. An FC layer can be achieved by a dot product between a weight map and the input data, or as a large convolution between the two maps. This product represents how strongly the data belongs to a class and is calculated for every class independently. The index of the largest weight can be selected and associated with the input data. Unfortunately, the multiplication of the feature map and a large weight map, i.e., point-wise calculation, cannot be efficiently implemented with CeNN template operations due to the large r (neighborhood size). In our study, we perform the FC layer operations on a digital processing element (conducting multiply and add functions) when the size of the matrices is greater than 3×3.

3.3 Design of CeNN-based CoNNs for MNIST

We now elaborate how CeNN arrays can be exploited to realize popular CoNNs (i.e., through architecture and algorithm efforts) based on the CeNN implementations of key CoNN operations discussed above. In particular, we consider using CeNN circuits and architecture to realize a CoNN algorithm for solving the MNIST problem (Horváth et al., 2017). In the MNIST handwritten digit classification task (LeCun et al., 1998), images of handwritten digits (0–9) represented by a 28×28-pixel black and white image must be classified. There are 60 000 images in the training set, and 10 000 images in the test set. According to the discussion in Section 3.2, all template operations for the convolution, ReLU, and pooling steps are feed-forward templates ($\boldsymbol{B}$). The feedback template ($\boldsymbol{A}$) is not used in any of the feature extracting operations (i.e., per Eq. (1), all values in $\boldsymbol{A}$ would simply be 0). As such, the training of the network can also be done with a CeNN with backpropagation methods such as stochastic gradient descent (Bottou, 2010).

In developing the CeNN-based CoNN, we restrict all computational kernels to a CeNN-friendly size of 3×3. Though larger kernels (e.g., 5×5, 7×7, or larger that may be used in CoNNs) are supported by the CeNN theory (i.e., per Section 3.2, a neighborhood's radius r could easily be larger than 1), due to increased connectivity requirements, such larger kernels are infrequently realized in hardware. That said, this is not necessarily a restriction. Larger kernels can be estimated using a series of 3×3 kernels, and it is a common practice to substitute larger kernels with 3×3 operations. Furthermore, Szegedy et al. (2016) suggested that smaller kernels can lead to fewer parameters and higher accuracy during training, which supports the use of implementation-friendly CeNN hardware.

Given the above, we devised a 5-layer, CeNN-friendly network, referred to as Design 1, which is depicted in Fig. 9 (Horváth et al., 2017). The first and third layers each consists of four convolution and four ReLU operations, the second and fourth layers each consists of four pooling layers, as indicated by the four stacked large squares. The last layer is a 20×20 FC layer for classification. Each layer (except the FC layer) employs four sets of different CeNN templates for extracting four different features. If only linear templates are allowed, a total of 68 CeNN template operations are required. If non-linear template operations are adopted, this number reduces to 38. As we have pointed out earlier, the FC layer operating on large matrix sizes can be costly for CeNN-based hardware implementation due to rapidly increasing interconnects. Based on the discussion in Section 3.2, we leverage a digital processing unit to realize this FC layer operation, which also requires an analog-to-digital conversion (ADC) unit.

Another method to overcome the FC layer challenge is to eliminate the need for such an FC layer by gradually reducing the feature map sizes to smaller sizes such as 3×3. Then the final FC layer needs only to deal with this small feature map. Based on this principle, we introduce an alternative network, referred to as Design 2, in which the original FC layer in Fig. 9 is replaced with an FC layer that needs only to deal with 3×3 feature maps, which is readily implementable with CeNN hardware (Fig. 10). This new network, however, has two additional layers, one convolution+RelU layer and one pooling layer. One question to answer is how this new network compares with the original network.

We omit the discussions on the actual architectural design. Interested readers can refer to Lou et al. (2018) for details.

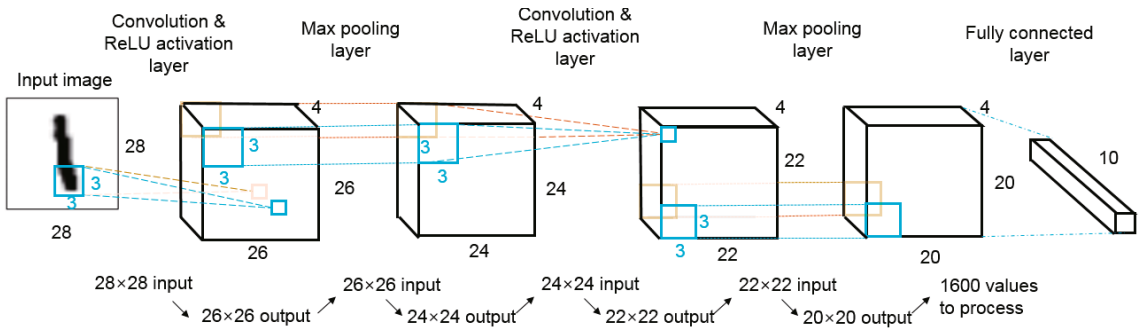


Fig. 9 CeNN-friendly CoNN with a fully connected layer

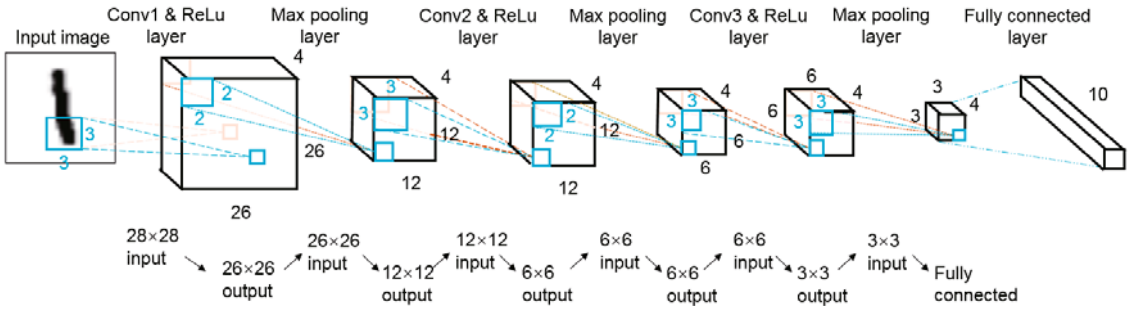


Fig. 10 CeNN-friendly CoNN without the fully connected layer layer

3.4 Evaluation of CeNN-based CoNNs for MNIST

To evaluate our CeNN-friendly CoNN designs obtained from our cross-layer efforts, we examine energy efficiency and accuracy. For our CeNN-based CoNNs in solving the MNIST problem, we used the 60 000 images in the training set to train the networks, and 10 000 images for inferencing to obtain inference accuracy. We consider six different CeNN-based CoNN designs below:

1. Design 1 (Fig. 9): (1) baseline (max pooling); (2) average pooling; (3) nonlinear templates.

2. Design 2 (Fig. 10): (1) baseline (max pooling); (2) average pooling; (3) nonlinear templates.

Inference accuracy depends on the actual network designs. Furthermore, precision in representation also impacts accuracy. We have developed both 4- and 8-bit precision CeNN-based designs. For detailed accuracy comparison, readers are referred to Horváth et al. (2017). The various components in CeNN-based designs are evaluated via SPICE simulation using the Arizona State University Predictive Technology Model for high-performance MOSFET devices at the 32-nm and 14-nm technology nodes (Zhao and Cao, 2006).

We first compare the six CeNN-based designs employing different technologies in terms of energy consumption and accuracy (equivalently misclassification rate) (Fig. 11). The beyond-CMOS devices considered include HomJTFET, HetJTFET, GaNT-FET, ThinTFET, TMDTFET, NCFET, FEFET, MITFET, vdWFET-BP, GpnJ-Vg3, and GpnJ-Vg2, and their associated energy and delay data are from Pan and Naeemi (2017a). Though interconnect parasitics, clocking and control circuits are not included in our evaluation, they should not change the overall trends seen in Fig. 11, since the energy of these elements would not be orders of magnitude larger and the major energy consumption is in the data movement and the computation as discussed in Chen et al. (2017). A few observations can be made from the figure. First, the 8-bit designs exhibit the best overall accuracy but highest energy. For example, Design 2 with average pooling for 8-bit design has energy of 23 fJ and accuracy of 97.4%, while Design 2 with average pooling for 4-bit design has 3 fJ energy and 94.3% accuracy. Second, comparing max pooling, average pooling, and nonlinear templates, we can see that algorithm- and circuit-level efforts together can achieve up to 2.5X energy improvements, and average pooling is most desirable

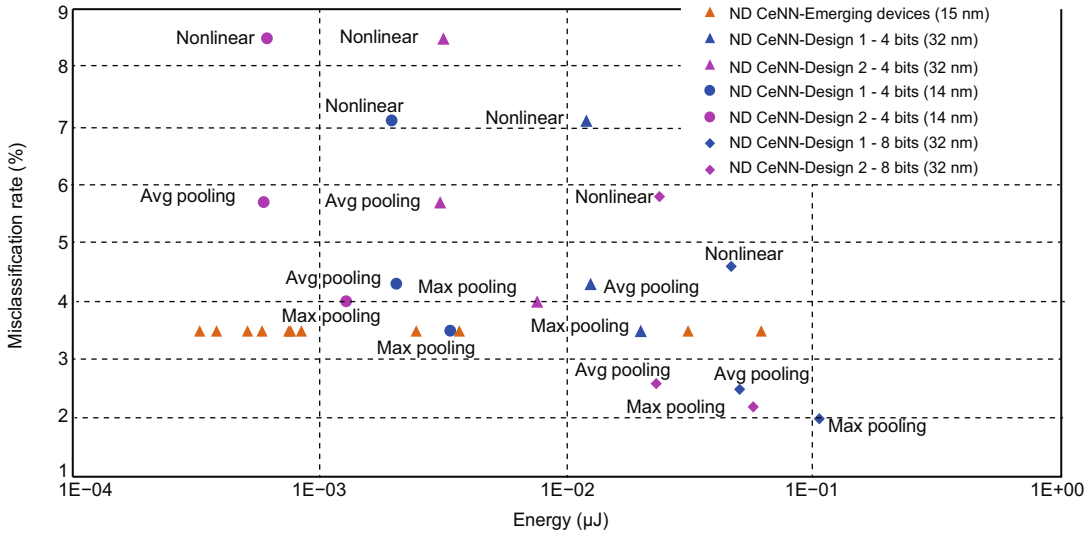


Fig. 11 Misclassification rate vs. energy for CeNN-based designs in solving the MNIST problem (References to color refer to the online version of this figure)

in terms of energy and accuracy tradeoff. Third, for the same algorithm/architecture/circuit design (hence the same accuracy), some beyond-CMOS devices are most energy-efficient with up to 10X energy improvement for similar technology nodes. The best performing beyond-CMOS device is ThinTFET (the left most triangle) (Li et al., 2016).

Next we examine the "competitiveness" of our CeNN-based designs with respect to existing accelerators for solving the MNIST problem. We are particularly interested in the ratio of energy to the number of operations so as to gauge how far we are to the goal set by DARPA, i.e., 0.3 fJ/op. Fig. 12 summarizes the misclassification rate vs. energy per operation (in log scale) for a number of published accelerator designs together with our CeNN-based designs. The operation here refers to multiply-add. The published accelerator designs that we have considered include the quantized ($\leq$8-bit precision) Minerva design (Reagen et al., 2016), digital (16-bit precision) DNN engine (Whatmough et al., 2017), digital (16-bit precision) KU Leuven ASIC chip (Moons and Verhelst, 2016), digital (16-bit precision) Eyeriss chip (Chen et al., 2017), RRAM crossbar based implementations (quantized with binary weights and inputs, and digital with 8-bit precision) from Tsinghua (Tang et al., 2017), and CPU/GPU-based solution of the DropConnect approach (Wan et al., 2013). Though there exist other accelerator designs for MNIST, they do not report the number of operations. Hence, the ratio of energy to the number of operations cannot be readily extracted. We have omitted these designs in this plot. Note that the CPU/GPU-based solution is the most accurate approach for MNIST to date, and the data are obtained from actual measurement via Intel i7-5820K with 32 GB DDR3 and Nvidia Titan. To make a fair comparison, for the existing NN accelerators based on CMOS technology nodes different from 32 nm, we scale their energy data to the 32-nm technology node using the ITRS data according to Perricone et al. (2016).

We can draw several conclusions from Fig. 12. The CPU/GPU design achieves the highest accuracy but has the highest energy per operation as expected. Among the considered CMOS-based designs, the Minerva MLP design offers the best energy per operation (the diamond point). Interestingly, though this design point has the lowest energy per operation, its actual energy consumption is not the lowest among the existing CMOS designs considered here. The lowest energy design is a DNN engine design with the lowest accuracy, i.e., the upper most orange circle with 40 fJ energy and 95.4% accuracy. The reason is that this DNN engine design, although having the lowest energy consumption compared to other CMOS designs (including other DNN engine designs implementing different networks), has a much lower

operation count and hence higher energy per operation (Whatmough et al., 2017). This reveals that lower energy per operation alone may not always lead to lower energy design due to variations in the number of operations.

Compared with the lowest energy DNN engine design, CeNN-based designs, for the same technology node (32-nm) and a slightly better accuracy, achieve a close to 19X energy/operation improvement. In terms of actual energy consumption, the CeNN designs achieve over 5X improvement. Among all the available energy/operation data, the Minerva MLP design, having the lowest energy/operation, leads to about 1 pJ/op, more than 3000X away from the goal of 0.3 fJ/op. The best one among all the CeNN-based designs (left most triangle) achieves 5.28 fJ/op, which is about 18X away from the goal of 0.3 fJ/op. However, the most energy-efficient CeNN design does sacrifice about 2% accuracy. We have pointed out that lower energy per operation does not necessarily mean lower energy. Nevertheless, the CeNN designs with better energy per operation happen to have lower energy. Specifically, comparing a 32-nm CeNN-based design with a 28-nm DNN design (with the lowest energy) in terms of iso-accuracy, the CeNN-based design offers 4X energy improvement.

4 Discussion and path forward

The case study presented in Section 3 demonstrates that through effective exploitation of unique characteristics of beyond-CMOS devices, significant energy efficiency improvements can be obtained. Such exploitation benefits greatly from vertically integrated cross-layer efforts spanning from devices and circuits all the way to architectures and algorithms. Specifically, we have observed that through the combined effort of algorithm, architecture, circuit, and device, over 220X improvement in energy can be achieved with about 0.5% accuracy degradation.

Though the case study focuses on leveraging TFETs and the CeNN architecture for designing DNN accelerators, the same observations can be made with other technologies applied to various applications. For example, we mentioned in Section 2 that FeFETs, being capable of functioning as both a switch and a storage element, can be used to build fine-grained compute-in-memory modules. As some examples, novel FeFET logic-in-memory elements such as basic logic gates and adders with storage capability were proposed by Yin et al. (2016b). Several FeFET-based ternary content-addressable-memory (TCAM) designs and a FeFET binary CoNN were introduced in Yin et al. (2017) and Chen et al. (2018). All of them effectively exploit the unique I-V characteristics of FeFETs, which exhibit (1) hysteresis, (2) a built-in access transistor, and (3) a high I_{on} to I_{off} ratio. In terms of quantitative advantages, for the TCAM example, besides layout area reduction, FeFET-based TCAMs can offer a maximum benefit of 7.5X/149X in energy-delay-product (EDP) versus a ReRAM/MTJ design (assuming 64 rows).

In our recent work, we also presented a FeFET-based compute-in-memory (CiM) architecture (Reis et al., 2018). The CiM design can not only serve as a general-purpose random access memory (RAM), but also perform Boolean operations ((N)AND, (N)OR, X(N)OR, INV) and addition (ADD) between two words in memory. The unique properties of FeFETs allow combined voltage- and current-based sense amplifiers, leading to more compact designs and lower power. The device-circuit-architecture efforts for the FeFET-CiM design achieve speedup (and energy reduction) of 119X (1.6X) and 1.97X (1.5X) over ReRAM and STT-RAM CiM designs, respectively, with respect to in-memory addition of 32-bit words. Furthermore, our approach offers an average speedup of 2.5X and energy reduction of 1.7X when compared to a conventional (not in-memory) approach across a wide range of application-level benchmarks.

Cross-layer design is most beneficial for domain-specific accelerators, which play a key role towards achieving peta operations per second per watt. Through cross-layer design, circuits and architectures can be developed by considering specific patterns (e.g., computation-intensive or memory-intensive, types of operations) of the algorithms. More importantly, unique characteristics (such as storage capability and analog behavior) of devices can be exploited through judicious design of circuits and architectures as well as algorithms.

Conducting cross-layer design does face quite some barriers. As we have shown in our case study, the algorithm selected impacts the underlying architecture and circuit, while circuit and architecture choices rely heavily on the devices to be used. Finding the best design across all these layers demands

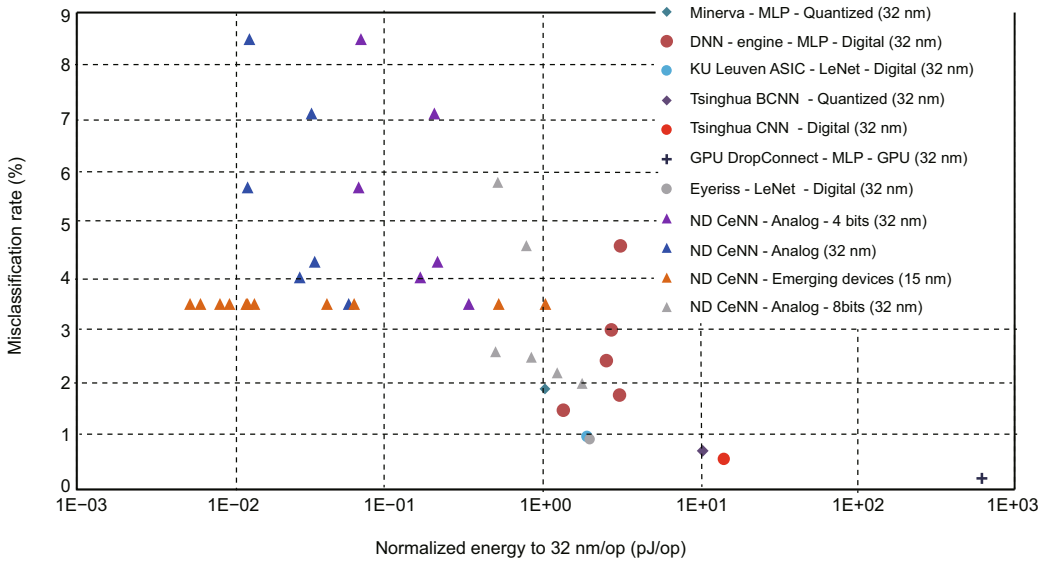


Fig. 12 Misclassification rate vs. energy/operation for various neural network accelerators in solving the MNIST problem. The data points for designs published by others are obtained from the literature. For each type of CeNN-based CMOS design, the six data points correspond to the six variations stated at the beginning of Section 3.4. The CeNN-based emerging technology designs all implement Design 1 (max pooling) (References to color refer to the online version of this figure)

the exploration of a huge design space as well as deep domain- and layer-specific knowledge. To aid such cross-layer efforts, proper abstraction and modeling tools are indispensable. Taking the DNN accelerator design as an example, one high-level top-down framework is shown in Fig. 13. The model spans five layers including (1) application domain (e.g., medical image segmentation, natural language processing, and robotics control), (2) benchmarking datasets from the application domain (e.g., ImageNet for image applications), (3) neural network description (e.g., number of layers, neurons and weights, operation types, and number of bits), (4) circuit and architecture description (e.g., composition of basic operations, memory hierarchy, sizes, and bandwidth), and (5) device- and technology-related modeling (e.g., energy and delay of basic operations). The goal of such a modeling framework is to predict figures of merits such as accuracy, performance, and energy. Proper evaluation tools are needed to support rapid exploration of the design space so as to narrow down the choices to a few promising solutions for further investigation.

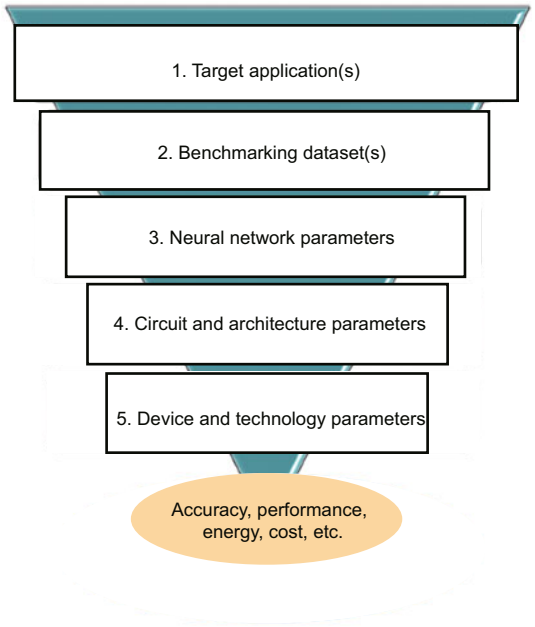


Fig. 13 A top-down modeling framework for designing DNN accelerators

In summary, isolated research efforts in any single layer seem not possible to take us to the goal of peta operations per second per watt. Cross-layer endeavor will be critical towards reaching the goal. Research advances in design tools to support such cross-layer endeavor are in great needs and will open new possibilities for device-circuit-architecture-algorithm co-optimization.

Acknowledgements

The authors would like to thank Mr. Qiu-wen LOU and Dr. Robert PERRICONE who have contributed several graphs and experimental data used in this study.

References

Avci UE, Rios R, Kuhn K, et al., 2011. Comparison of performance, switching energy and process variations for the TFET and MOSFET in logic. Symp. on VLSI Technology, Digest of Technical Papers, p.124-125.

Aziz A, Breyer ET, Chen A, et al., 2018. Computing with ferroelectric FETs: devices, models, systems, and applications. Proc Design, Automation & Test in Europe Conf Exhibition, p.1289-1298. https://doi.org/10.23919/DATE.2018.8342213

Bottou L, 2010. Large-scale machine learning with stochastic gradient descent. Proc 19th Int Conf on Computational Statistics, p.177-186. https://doi.org/10.1007/978-3-7908-2604-3_16

Chen XM, Yin XZ, Niemier M, et al., 2018. Design and optimization of FeFET-based crossbars for binary convolution neural networks. Proc Design, Automation & Test in Europe Conf Exhibition, p.1205-1210. https://doi.org/10.23919/DATE.2018.8342199

Chen YH, Krishna T, Emer JS, et al., 2017. Eyeriss: an energy-efficient reconfigurable accelerator for deep convolutional neural networks. *IEEE J Sol-State Circ*, 52(1):127-138. https://doi.org/10.1109/JSSC.2016.2616357

Chua LO, Roska T, 2002. Cellular Neural Networks and Visual Computing: Foundations and Applications. Cambridge University Press, New York, NY, USA.

Chua LO, Yang L, 1988. Cellular neural networks: theory. *IEEE Trans Circ Syst*, 35(10):1257-1272. https://doi.org/10.1109/31.7600

Dahl GE, Sainath TN, Hinton GE, 2013. Improving deep neural networks for LVCSR using rectified linear units and dropout. Proc IEEE Int Conf on Acoustics, Speech and Signal Processing, p.8609-8613. https://doi.org/10.1109/ICASSP.2013.6639346

Esmaeilzadeh H, Blem E, St. Amant R, et al., 2011. Dark silicon and the end of multicore scaling. Proc 38th Annual Int Symp on Computer Architecture, p.365-376. https://doi.org/10.1145/2024723.2000108

Esmaeilzadeh H, Blem E, St. Amant R, et al., 2013. Power challenges may end the multicore era. *Commun ACM*, 56(2):93-102. https://doi.org/10.1145/2408776.2408797

George S, Aziz A, Li XQ, et al., 2016a. Device circuit co design of FeFET based logic for low voltage processors. Proc IEEE Computer Society Annual Symp on VLSI, p.649-654. https://doi.org/10.1109/ISVLSI.2016.116

George S, Ma KS, Aziz A, et al., 2016b. Nonvolatile memory design based on ferroelectric FETs. Proc 53rd Annual Design Automation Conf, Article 118. https://doi.org/10.1145/2897937.2898050

Horváth A, Hillmer M, Lou QW, et al., 2017. Cellular neural network friendly convolutional neural networks—CNNs with CNNs. Proc Design, Automation & Test in Europe Conf & Exhibition, p.145-150. https://doi.org/10.23919/DATE.2017.7926973

Ionescu AM, Riel H, 2011. Tunnel field-effect transistors as energy-efficient electronic switches. *Nature*, 479(7373):329-337. https://doi.org/10.1038/nature10679

Kam H, Liu TJK, Alon E, 2012. Design requirements for steeply switching logic devices. *IEEE Trans Electron Dev*, 59(2):326-334. https://doi.org/10.1109/TED.2011.2175484

Khatami Y, Banerjee K, 2009. Steep subthreshold slope n- and p-type tunnel-FET devices for low-power and energy-efficient digital circuits. *IEEE Trans Electron Dev*, 56(11):2752-2761. https://doi.org/10.1109/TED.2009.2030831

Kim K, Lee S, Kim JY, et al., 2008. A 125 GOPS 583 mW network-on-chip based parallel processor with bio-inspired visual attention engine. *IEEE J Sol-State Circ*, 44(1):136-147. https://doi.org/10.1109/JSSC.2008.2007157

LeCun Y, Bottou L, Bengio Y, et al., 1998. Gradient-based learning applied to document recognition. *Proc IEEE*, 86(11):2278-2324. https://doi.org/10.1109/5.726791

Li MO, Yan RS, Jena D, et al., 2016. Two-dimensional heterojunction interlayer tunnel FET (Thin-TFET): from theory to applications. Proc IEEE Int Electron Devices Meeting, p.504-507. https://doi.org/10.1109/IEDM.2016.7838451

Liu HC, Datta S, Shoaran M, et al., 2014. Tunnel FET-based ultra-low power, low-noise amplifier design for bio-signal acquisition. Proc IEEE/ACM Int Symp on Low Power Electronics and Design, p.57-62. https://doi.org/10.1145/2627369.2627631

Lou QW, Palit I, Horváth A, et al., 2015. TFET-based operational transconductance amplifier design for CNN systems. Proc 25th Edition on Great Lakes Symp on VLSI, p.277-282. https://doi.org/10.1145/2742060.2742089

Lou QW, Pan CY, McGuinness J, et al., 2018. A mixed signal architecture for convolutional neural networks. To appear in arXiv.

Molinar-Solis JE, Gomez-Castaneda F, Moreno-Cadenas JA, et al., 2007. Programmable CMOS CNN cell based on floating-gate inverter unit. *J VLSI Signal Process Syst Signal Image Video Technol*, 49(1):207-216. https://doi.org/10.1007/s11265-007-0056-7

Moons B, Verhelst M, 2016. A 0.3–2.6 TOPS/W precision-scalable processor for real-time large-scale ConvNets. Proc IEEE Symp on VLSI Circuits, p.1-2. https://doi.org/10.1109/VLSIC.2016.7573525

Nikonov DE, Young IA, 2013. Overview of beyond-CMOS devices and a uniform methodology for their benchmarking. *Proc IEEE*, 101(12):2498-2533. https://doi.org/10.1109/JPROC.2013.2252317

Nikonov DE, Young IA, 2015. Benchmarking of beyond-CMOS exploratory devices for logic integrated circuits. *IEEE J Explor Sol-State Comput Dev Circ*, 1:3-11. https://doi.org/10.1109/JXCDC.2015.2418033

Pan CY, Naeemi A, 2017a. Beyond-CMOS device benchmarking for Boolean and non-Boolean logic applications. http://cn.arxiv.org/abs/1711.04295

Pan CY, Naeemi A, 2017b. Beyond-CMOS non-Boolean logic benchmarking: insights and future directions. Proc Design, Automation & Test in Europe Conf & Exhibition,

p.133-138.
https://doi.org/10.23919/DATE.2017.7926971

Perricone R, Hu XS, Nahas J, et al., 2016. Can beyond-CMOS devices illuminate dark silicon? Design, Automation Test in Europe Conf Exhibition, p.13-18.

Reagen B, Whatmough P, Adolf R, et al., 2016. Minerva: enabling low-power, highly-accurate deep neural network accelerators. Proc ACM/IEEE 43rd Annual Int Symp on Computer Architecture, p.267-278.
https://doi.org/10.1109/ISCA.2016.32

Reis D, Niemier M, Hu X, 2018. Computing in memory with FeFETs. Proc IEEE/ACM Int Symp on Low Power Electronics and Design, p.1-6.
https://doi.org/10.1145/2627369.2627631

Rodriguez-Vázquez A, Liñán-Cembrano G, Carranza L, et al., 2004. Ace16k: the third generation of mixed-signal SIMD-CNN ACE chips toward VSoCs. *IEEE Trans Circ Syst I*, 51(5):851-863.
https://doi.org/10.1109/TCSI.2004.827621

Salahuddin S, Datta S, 2008. Use of negative capacitance to provide voltage amplification for low power nanoscale devices. *Nano Lett*, 8(2):405-410.
https://doi.org/10.1021/nl071804g

Salmon L, 2017. A DARPA Perspective. https://www.src.org/calendar/e006128/agenda/salmon-darpa.pdf

Scheutz M, McRaven J, Cserey G, 2004. Fast, reliable, adaptive, bimodal people tracking for indoor environments. Proc IEEE/RSJ Int Conf on Intelligent Robots and Systems, p.1347-1352.
https://doi.org/10.1109/IROS.2004.1389583

Seabaugh AC, Zhang Q, 2010. Low-voltage tunnel transistors for beyond CMOS logic. *Proc IEEE*, 98(12):2095-2110.
https://doi.org/10.1109/JPROC.2010.2070470

Sedighi B, Hu XS, Liu HC, et al., 2015. Analog circuit design using tunnel-FETs. *IEEE Trans Circ Syst I*, 62(1):39-48. https://doi.org/10.1109/TCSI.2014.2342371

Szegedy C, Vanhoucke V, Ioffe S, et al., 2016. Rethinking the inception architecture for computer vision. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.2818-2826.
https://doi.org/10.1109/CVPR.2016.308

Szolgay P, Szatmari I, Laszlo K, 1997. A fast fixed point learning method to implement associative memory on CNNs. *IEEE Trans Circ Syst I*, 44(4):362-366.
https://doi.org/10.1109/81.563627

Tang TQ, Xia LX, Li BX, et al., 2017. Binary convolutional neural network on RRAM. Proc 22nd Asia and South Pacific Design Automation Conf, p.782-787.
https://doi.org/10.1109/ASPDAC.2017.7858419

Wan L, Zeiler M, Zhang S, et al., 2013. Regularization of neural networks using dropconnect. Proc 30th Int Conf on Machine Learning, p.1058-1066.

Wang L, de Gyvez JP, Sanchez-Sinencio E, 1998. Time multiplexed color image processing based on a CNN with cell-state outputs. *IEEE Trans VLSI Syst*, 6(2):314-322. https://doi.org/10.1109/92.678895

Whatmough PN, Lee SK, Lee H, et al., 2017. 14.3 A 28nm SoC with a 1.2GHz 568nJ/prediction sparse deep-neural-network engine with >0.1 timing error rate tolerance for IoT applications. Proc IEEE Int Solid-State Circuits Conf, p.242-243.
https://doi.org/10.1109/ISSCC.2017.7870351

Xu XW, Lu Q, Wang TC, et al., 2017. Edge segmentation: empowering mobile telemedicine with compressed cellular neural networks. Proc 36th Int Conf on Computer-Aided Design, p.880-887.
https://doi.org/10.1109/ICCAD.2017.8203873

Yin XZ, Aziz A, Nahas J, et al., 2016a. Exploiting ferroelectric FETs for low-power non-volatile logic-in-memory circuits. Proc IEEE/ACM Int Conf on Computer-Aided Design, p.1-8.
https://doi.org/10.1145/2966986.2967037

Yin XZ, Sedighi B, Niemier M, et al., 2016b. Design of latches and flip-flops using emerging tunneling devices. Proc Design, Automation & Test in Europe Conf & Exhibition, p.1150-1155.
https://doi.org/10.3850/9783981537079_0669

Yin XZ, Niemier M, Hu XS, 2017. Design and benchmarking of ferroelectric FET based TCAM. Proc Design, Automation & Test in Europe Conf & Exhibition, p.1448-1453. https://doi.org/10.23919/DATE.2017.7927219

Zhao W, Cao Y, 2006. New generation of predictive technology model for sub-45 nm early design exploration. *IEEE Trans Electron Dev*, 53(11):2816-2823.
https://doi.org/10.1109/TED.2006.884077

Zhou G, Li R, Vasen T, et al., 2012. Novel gate-recessed vertical InAs/GaSb TFETs with record high ION of 180 μA/μm at V_{DS}=0.5 V. Proc Int Electron Devices Meeting, p.777-780.
https://doi.org/10.1109/IEDM.2012.6479154