Please cite as Hamza Al Maharmeh, Nabil J. Sarhan, Chung-Chih Hung, Mohammed Ismail, Mohammad Alhawari. Compute-in-Time for Deep Neural Network Accelerators: Challenges and Prospects. In Proceedings of the IEEE 63rd International Midwest Symposium on Circuits & Systems (MWSCAS 2020), pages 990-993, August 9-12, 2020. DOI: 10.1109/MWSCAS48704.2020.9184470.

Compute-in-Time for Deep Neural Network Accelerators: Challenges and Prospects

Hamza Al Maharmeh[1], Nabil J. Sarhan[1], Chung-Chih Hung[2], Mohammed Ismail[1], and Mohammad Alhawari[1]
[1]*Dept. of Electrical and Computer Engineering, Wayne State University*. Detroit, USA
[2]*National Chiao Tung University.* Hsinchu, Taiwan
[1]{hamza.m,nabil.sarhan,ismail,alhawari}@wayne.edu
[2]cchung@mail.nctu.edu.tw

***Abstract*—Time-domain (TD) accelerators leverage both digital and analog features, thereby enabling energy-efficient computing and scaling with CMOS technology. This paper reviews state-of-the-art TD accelerators and discusses system considerations and hardware implementations, including the spatially unrolled and recursive TD architectures. Additionally, the paper analyzes the energy and area efficiency of the TD architectures for varying input resolutions and network sizes. This analysis provides insight for designers into how to choose the appropriate TD approach for a particular application.**

Keywords—Time-Domain (TD) Accelerators, Time-Domain Computation, Analog Domain, Spatially Unrolled, Recursive.

I. Introduction

The rapid deployment of low-power IoT devices requires highly efficient computing. Besides, the increasing demand for high performance and energy efficiency in Artificial Neural Networks (ANNs) and Deep Learning (DL) has driven a wide range of application-specific integrated circuits (ASICs) [1, 2].

An emerging trend in implementing machine learning-based (ML) accelerators is utilizing the *time domain* (TD) to compute multiply-accumulate (MAC) operations, which are the core of ML applications [5-10]. In TD computing, pulse width modulation (PWM) is used to map digital values into TD domain and then each MAC operation computes the output in terms of time delay. TD ANN can achieve excellent energy and hardware efficiency [5-10]. Although analog computation [3, 4] is efficient in terms of energy and area, it has limited accuracy and technology scaling [10]. The digital approach is more versatile and offers the best use of technology scaling, but it is not as efficient as the analog approach [3, 4, 10]. The TD approach leverages both features because it is energy efficient and can be scaled with CMOS technology.

This paper discusses the hardware implementation of time-based accelerators, including spatially unrolled and recursive architectures. Moreover, it analyzes the effect of varying the number of input bits and the size of the neural network on the area and energy efficiency for both TD architectures.

The rest of this paper is organized as follows. Section II compares different hardware accelerator implementations: analog, digital, and TD. Section III analyzes the different TD approaches. Section IV analyzes and compares the spatially unrolled and recursive TD cores. Finally, conclusions are drawn in Section V.

II. Comparison among Different Hardware Accelerator Implementations

The digital implementation is versatile and can be scaled with CMOS technology. However, since the data are represented as a multi-bit digital vector, the number of MAC units and operations increases with the number of bits, resulting in high dynamic switching capacitance, and consequently more power and area overhead [1, 2, 10].

In the analog domain, data are represented as a continuously varying voltage signal. Several analog-based accelerators have been proposed to implement MAC operations using charge manipulation schemes and ADCs [1, 2]. Analog approaches compute MAC operation in the analog voltage domain using a static random access memory (SRAM) array, capacitors, and data converters. In these approaches, input pixel data are either encoded as a PWM signal or a pulse- amplitude-modulated (PAM) signal. The MAC operation is performed by summing up the read current of simultaneously accessed bit-cells. This approach is susceptible to process variation, noise, bit-flips, and weak line corruption. Although analog computations are efficient in terms of energy (OPS/W) and area (OPS/mm2), they have limited accuracy and technology scaling [10] because of the finite voltage headroom.

The data in the TD are represented as a pulse with variable width or time difference in rising/falling edges to generate variable delays. The TD approach combines the advantages of the digital and analog approaches; it can scale with technology and computation is energy-efficient. In addition, unlike the analog-based computation, which requires the use of an analog circuit design flow, TD circuits can utilize the digital IC design flow, thereby enabling large-scale integration. TD cores can surpass digital implementations of ANN only if the number of input bits is 6 or fewer [10]. In TD ANN, calibration is necessary because of the analog nature of the delay signal that is more prone to noise and process variation. Moreover, the TD approach requires additional time-to-digital converters (TDCs) and digital-to-time converters (DTCs). However, DTC and TDC are still more energy- and area-efficient than DAC and ADC [11]. TD computing is better suited for applications that require low resolution and have stringent power requirements, such as edge devices. Table I summarizes the pros and cons of the aforementioned approaches. The comparison does not consider the reuse of blocks.

Phase-domain (PD) ANN is similar to TD, but it utilizes the phase shift to implement the dot product [12]. The main issues are requiring multiple clock sources and the dependence of the toggle activity on the input magnitude.

TABLE I. COMPARING DIFFERENT ACCELERATORS

Approach	Digital	Analog	Time
Data Representation	Multi-bit digital vector	Continuous voltage signal	Pulse with modulation
Technology Scaling	Yes	No	Yes
Immunity to noise	High	Low	Moderate
Input Resolution	High	Low/Moderate	Low/Moderate
Energy Efficiency	Low	Very High	High
Throughput	High	Very High	Moderate
Area	Large	Moderate	Moderate

III. TIME-DOMAIN ACCELERATORS

A TD ANN can be implemented using a *spatially unrolled* (SU) [5, 6, 8] or a *recursive* (REC) [9, 10] architecture. In the SU architecture, the inputs and weights are stored in spatially distributed storage elements with a dedicated processing element (PE). Thus, *N* number of DTCs are required for *N* input neurons, as shown in Fig. 1.a. The main advantage is that the addition operation comes for free as the pulse propagates from one delay element to the other. Fig. 1.b shows the recursive architecture, which reuses the same DTC and TDC blocks to perform each MAC, thereby reducing the chip area. SU is fast, but it is area-inefficient when it comes to scaling the number of input bits. As such, SU is limited to a small number of bits. In contrast, REC is slow, as each input is fed serially to a DTC, then to a time register/counter, and then the same process is repeated for the next input using the same blocks. However, the design can be scaled to large numbers of input bits because multi-bits digital inputs can be represented easily in PWM.

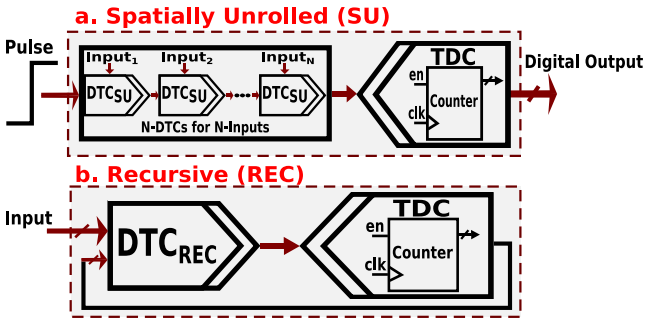

Fig. 1. TD Architectures: **a.** Spatially unrolled and **b.** Recursive

In [7], TD signal processing techniques are proposed, including shared time generator, median filter (MF), and winner-take-all (WTA) circuits. The MAC is performed in the digital domain, while WTA and MF are implemented in TD. The implementation of WTA and MF requires TD arithmetic, such as min., max., compare, and add using a NAND gate, a NOR gate, 2 NAND gates, and a series of inverters with variable delay, respectively. A WTA classifier classifies the generated feature vector. All comparison results are finally converted back to digital to find the final winner. The main disadvantage of this approach is that the only parts that are related to TD computation are the MF and WTA. Filtering happens first and WTA occurs last, and the processing in between (MAC) is done in digital, thereby requiring a 4x conversion time from digital to time and then time to digital. Besides, the power consumption is 75mW which is very high when compared with the typical TD approach.

A. Spatially Unrolled Architecture

In [8], a digitally controlled oscillator (DCO) is used to modulate the frequency by switching capacitor loads representing the weights, while the number of cycles is counted in a set sampling period. Each input stage of the DCO is composed of an inverter and binary-weighted MOSFET capacitors controlled by an input pixel and a 3-bit weight, which are stored in SRAM cells. Input pixels determine whether a stage is activated or not, and weights determine how many capacitors are turned-on as load in that stage. Delay of all stages accumulates naturally in the DCO loop and is modulated to an oscillation frequency, which is fed to an 8-bit counter. The counter increments every DCO cycle, and when the counter value reaches a specific count, a spike is generated and the counter self-resets. The main drawbacks can be summarized as follows. (1) The DCO must oscillate for many cycles to generate a reliable result. (2) Since many DCOs are used, the mismatch can cause a severe problem. (3) Implementation of binary-weighted capacitor banks makes the design sensitive to parasitic diffusion capacitance of the DCO and increases the chip area. (4) Adding a capacitor at every node of a continuously running oscillator increases the total capacitance and, as a result, the power consumption.

An approach based on delay lines is presented in [6], but the outputs and weights are restricted to binary. The proposed design uses the time difference of rising edges of two nodes for representing a value. The polarity is considered by monitoring which edge arrives first. The weights control the result of multiplication, whether it is negative or positive, using a variable resistor. Hence, the proposed PE is a series of inverters and two variable resistors. Binary-weighted NMOS transistors are used as variable resistors. The main drawback of this design is that the outputs and weights are restricted to binary. Besides, the design has twice the area overhead because of utilizing local reference delay lines instead of a global reference.

In [5], a TD core using one-shot delay measurements and an error correction technique is proposed. That work implements digitally controlled delay lines that are compared to a shared reference delay to compute the MAC operations. A chain of variable delay elements is cascaded, where each element has a delay value that depends on the product of the input and the weight. An input pulse is applied at the first element, where the output pulse width will eventually represent the MAC operation needed for the neuron. A low-resolution 2-bit TDC is used to convert the analog signal into digital and also to implement a rectified linear unit (ReLU) simultaneously, which makes the use of dynamic threshold error correction (DTEC) essential to get acceptable results. The use of DTEC can result in more than 51% of the time and power overhead. Moreover, completing the

computation of the hidden layer, the digital output of TDC must be encoded in a thermometer. Additionally, the one-hot encoding of 3-bit weights into 8-bit is necessary because of the used digitally controlled delay cell. Therefore, eight SRAM cells are required to represent each 3-bit weight. Furthermore, the accelerator works with only binary input.

B. Recursive Architecture

In [9], a TD CNN engine using bi-directional memory delay lines is proposed. The digital inputs are sequentially encoded as PWM signals using a DTC. Then, the time signal is multiplied by the corresponding weight bit using an AND gate. Subsequently, these pulses are sent to a time register to add them up. Finally, an up-down counter is used to convert the resultant MAC signal into a digital value. For CNN implementation, the authors implemented a max-pooling operation using three 8-bit comparators. Although this approach supports multi-bit resolution for the inputs (but not the weights), it results in lower throughput due to its sequential operation, even when compared with other TD approaches.

In [10], a TD neuromorphic accelerator with reinforcement learning is proposed. It is used by a mobile robot to avoid obstacles. The input is already represented as an 8-bit delay pulse because it comes from ultra-sonic sensors for measuring the distance from obstacles. The output is a control message sent to a Raspberry PI, which in turn sends the control command to the motor controllers. The MAC operations are computed in TD and accumulated in a 15-bit counter. A digital word, consisting of the first seven MSBs of the counter, is fed to a digital-to-pulse converter (DPC), which also implements a ReLU. Weights are represented in 6-bit signed-magnitude format. An input pulse controls the select bit of the multiplexor, and it is also used as the counter enable signal. The main drawback of this approach is the use of a DCO, which needs settling time. It needs to count for a long time to have the correct output. Also, the enable signal for the counter and the local DCO clock are asynchronous and can lead to a 1 LSB of error. Additionally, generating 32 frequencies for the DCO will result in many issues related to linearity and calibration. Since the design is mostly digital, the power consumed was 47% less than the digital, but in typical TD the savings should be better. In summary, table II summarizes the aforementioned approaches.

TABLE II. COMPARISON BETWEEN DIFFERENT TD CORES

Reference	[8]	[6]	[5]	[9]	[7]	[10]
Architecture	SU	SU	SU	REC	Digital-Time	REC
Technology [nm]	65	65	65	40	55	55
Chip Area [mm^2]	0.24	3.61	0.644	0.124	0.64	3.4
Precision [bit] Input/weight	1/3	1/1	1/3	4/1	MF=8[a] WTA=6[a]	*/6
VDD [V]	1.2	1	1.2	0.537	1.2	0.4-1
Frequency [MHz]	792	23041	1700	24	1330	780**
Energy Efficiency [TOPS/s/w]	2.47	48.2	36.2	12.08	NR^b	3.12**
Hardware Efficiency [GE/PE]	33.2	76.5	38.4	NR^b	NR^b	NR^b

[a] Weights are not needed since the TD calculations are winner take all (WTA), and median filter (MF).

[b] NR: Not Reported.

*The input is already represented in the time domain since it comes from an ultrasonic sensor.

** Frequency reported at 1V, while power efficiency reported at 0.4V.

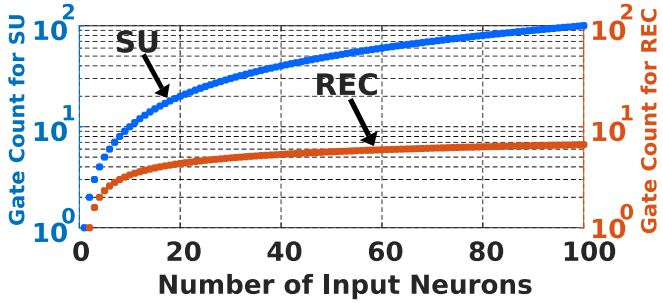

Fig. 2. Approximated gate count for SU and REC architectures.

IV. ANALYSIS OF SU AND REC ARCHITECTURES

In this section, we will analyze and compare the SU and REC architectures in terms of area and energy efficiency for varying numbers of input bits and input neurons. A counter-based TDC and an inverter-based DTC are considered in all analyses. The DTC used in REC architecture is usually more complicated than the one used in SU, but for simplicity, we assume both are identical.

A. Chip Area Analysis

In SU, the MAC operation requires multiple DTCs to realize the accumulation, and thus the gate count for DTC (GC_{DTC}) increases in proportion to the number of input neurons (x_n). The number of inverters in the DTC is proportional to 2 exponent the number of input bits ($2^{D_{in}}$). The flip-flops count for TDC (GC_{TDC}) is proportional to the logarithmic of the number of input bits (D_{in}): $log_2(D_{in})$. Hence, the total gate count (TGC) can be given by

$$TGC_{SU} = GC_{DTC} \times x_n + GC_{TDC}. \quad (1)$$

For REC, The total gate count can be given by

$$TGC_{REC} = GC_{DTC} + GC_{TDC}. \quad (2)$$

We can develop TGC_{SU} as a function of TGC_{REC} as follows:

$$TGC_{SU} = TGC_{REC}\left[\frac{GC_{DTC}\times x_n + GC_{TDC}}{GC_{DTC}+GC_{TDC}}\right] \approx TGC_{REC}\left[\frac{2^{D_{in}}\times x_n + log_2(D_{in})}{2^{D_{in}} + log_2(D_{in})}\right] \quad (3)$$

Notice that $\frac{2^{D_{in}}\times x_n + log_2(D_{in})}{2^{D_{in}} + log_2(D_{in})}$ is always greater than 1 for $D_{in}, x_n > 1$.

As can be seen in Fig. 2, the area for REC is always less than SU, with a ratio of approximately 10/100 when $x_n = 100$.

B. Energy Efficiency Analysis

Let us first analyze the dynamic and static energy consumed by TDC and DTC. For TDC, the dynamic and static energy consumption can be given by

$$E_{TDC\_Dynamic} = T_{in} \times W \times f_{TDC\_min} \times \frac{1}{2} C_{counter\_one\_bit} V^2, \text{ and } (4)$$

$$E_{TDC\_Static} = I_{leakage\_one\_gate} \times V(T_{period} - T_{active}) \times GC_{TDC}, \quad (5)$$

where T_{in} is the input represented in the time domain, and w is the weight, f_{TDC_min} is the minimum frequency of the TDC which in this case is assumed to be counter-based, $C_{counter\_one\_bit}$ is the total capacitance of 1 bit

TDC, $I_{leakage\_one\_gate}$ is the leakage current for one gate, V is the supply voltage, T_{period} is the total period, and T_{active} is the period at which the TDC is active. For DTC, the dynamic and static energy consumption can be given by

$$E_{DTC\_Dynamic} = 2^{D_{in}} \times \frac{1}{2} C_{inverter} V^2, \text{ and} \quad (6)$$

$$E_{DTC\_Static} = I_{leakage\_inverter} \times V \times (T_{period} - T_{active}) \times 2^{D_{in}}, \quad (7)$$

where $2^{D_{in}}$ represents the number of inverters in the DTC.

In SU, the dynamic, static, and total energy consumption for a MAC with x_n set of D_{in} and w can be given by

$$E_{SU\_Dynamic} = E_{one\_pulse} + E_{DTC\_Dynamic} \times x_n + E_{TDC\_Dynamic}, \quad (8)$$

$$E_{SU\_Static} = x_n(x_n - 1)E_{DTC\_Static} + E_{TDC_{Static}}, \text{and} \quad (9)$$

$$E_{SU\_total} = E_{SU\_Dynamic} + E_{SU\_Static}. \quad (10)$$

Notice that $x_n(x_n - 1)E_{DTC\_Static}$ is valid because each DTC is active only once, and for the rest of the clock period it is just waiting for the other DTCs to finish.

In REC, the dynamic, static, and total energy consumption for a MAC with x_n set of D_{in} and w can be given by

$$E_{REC\_Dynamic} = (E_{DTC\_Dynamic} + E_{TDC\_Dynamic}) \times x_n, \quad (11)$$

$$E_{REC\_Static} = (E_{DTC\_Static} + E_{TDC\_Static}) \times x_n, \text{ and} \quad (12)$$

$$E_{REC\_total} = E_{REC\_Dynamic} + E_{REC\_Static}. \quad (13)$$

Fig. 3 shows the consumed energy at SU and REC architectures for different digital input bits (*Din*) and input neurons. In Fig. 3.a, the digital input is 4 bits (*Din=4*). The SU architecture is more efficient if the number of input neurons is fewer than 45,000. For larger ANNs, REC is more efficient. Once the number of input bits is doubled (i.e. *Din=8*), the cutoff point is 5,500 input neurons.

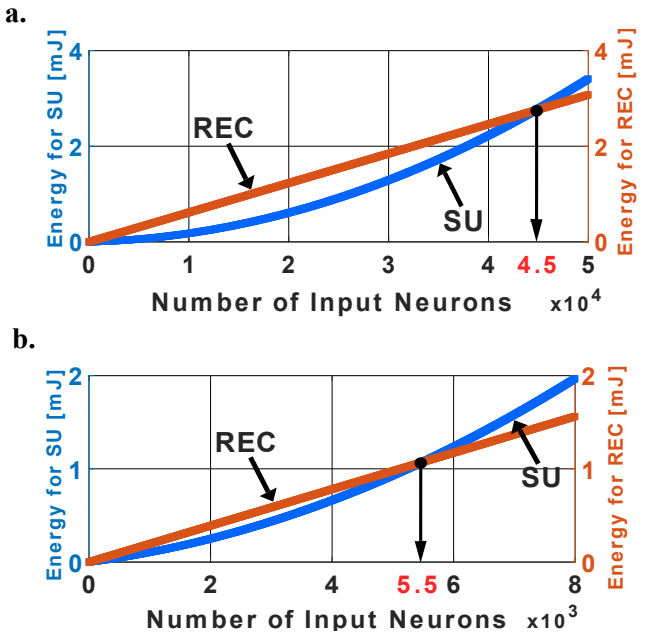

Fig. 3. Approximated consumed energy for SU and REC architectures for a. input bits (Din) = 4 and b. Din = 8.

V. Conclusion

The TD approach for ANN accelerators aims at combining the advantages of the digital and analog approaches. It can easily scale with technology while being energy-efficient. We have discussed the two hardware implementations of TD accelerators: *spatially unrolled* (SU) and *recursive* (REC). We have also analyzed the effect of varying the number of input bits and the size of the neural network on the area and energy efficiency for both architectures. REC has a smaller area, but SU is more energy-efficient for small ANN design.

References

[1] A. Andreopoulos, R. Alvarez-Icaza, A. S. Cassidy, and M. D. Flickner. "A low-power neurosynaptic implementation of local binary patterns for texture analysis." In 2016 International Joint Conference on Neural Networks (IJCNN), pages 4308–4316, July 2016.

[2] A C. Wang, W. Lou, L. Gong, L. Jin, L. Tan, Y. Hu, X. Li, and X. Zhou. "Reconfigurable hardware accelerators: Opportunities, trends, and challenges." CoRR, abs/1712.04771, 2017.

[3] Y. C. Xiang, P. Huang, Z. Zhou, R. Z. Han, Y. N. Jiang, Q. M. Shu, Z. Q. Su, Y. B. Liu, X. Y. Liu, and J. F. Kang. "Analog deep neural network based on nor flash computing array for high speed/energy efficiency computation." In 2019 IEEE International Symposium on Circuits and Systems (ISCAS), pages 1–4, May 2019.

[4] A. Tripathi, M. Arabizadeh, S. Khandelwal, and C. S. Thakur. "Analog neuromorphic system based on multi input floating gate mos neuron model." In 2019 IEEE International Symposium on Circuits and Systems (ISCAS), pages 1–5, May 2019.

[5] L. R. Everson, M. Liu, N. Pande and C. H. Kim, "An Energy-Efficient One-Shot Time-Based Neural Network Accelerator Employing Dynamic Threshold Error Correction in 65 nm," *IEEE Journal of Solid-State Circuits*, vol. 54, no. 10, pp. 2777-2785, Oct. 2019.

[6] 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 Journal of Solid-State Circuits (JSSC),* vol. 52, no. 10, pp. 2679 - 2689, Oct. 2017.

[7] Z. Chen and J. Gu, "A Time-Domain Computing Accelerated Image Recognition Processor With Efficient Time Encoding and Non-Linear Logic Operation," *IEEE Journal of Solid-State Circuits (JSSC),* vol. 54, no. 11, pp. 3226 - 3237, Nov. 2019.

[8] 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.

[9] A. Sayal, S. S. T. Nibhanupudi, S. Fathima and J. P. Kulkarni, "A 12.08-TOPS/W All-Digital Time-Domain CNN Engine Using Bi-Directional Memory Delay Lines for Energy Efficient Edge Computing," *IEEE Journal of Solid-State Circuits*, vol. 55, no. 1, pp. 60-75, Jan. 2020.

[10] A. Amaravati, S. B. Nasir, J. Ting, I. Yoon and A. Raychowdhury, "A 55-nm, 1.0–0.4V, 1.25-pJ/MAC Time-Domain Mixed-Signal Neuromorphic Accelerator With Stochastic Synapses for Reinforcement Learning in Autonomous Mobile Robots," *IEEE Journal of Solid-State Circuits*, vol. 54, no. 1, pp. 75-87, Jan. 2019.

[11] D. Miyashita *et al*., "An LDPC Decoder With Time-Domain Analog and Digital Mixed-Signal Processing," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 1, pp. 73-83, Jan. 2014.

[12] Y. Toyama, K. Yoshioka, K. Ban, S. Maya, A. Sai, and K. Onizuka. "An 8 bit 12.4 tops/w phase-domain mac circuit for energy-constrained deep learning accelerators." *IEEE Journal of Solid-State Circuits*, 54(10):2730–2742, Oct 2019.