Please cite as Hamza Al Maharmeh, Nabil J. Sarhan, Chung-Chih Hung, Mohammed Ismail, and Mohammad Alhawari. A Comparative Analysis of Time-Domain and Digital-Domain Hardware Accelerators for Neural Networks. In Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS), May 2021. DOI: 10.1109/ISCAS51556.2021.9401758.

A Comparative Analysis of Time-Domain and Digital-Domain Hardware Accelerators for Neural Networks

Hamza Al Maharmeh[1], Nabil J. Sarhan[1], Chung-Chih Hung[2], Mohammed Ismail[1], and Mohammad Alhawari[1]
[1]WINCAS Research Center, Dept. of Electrical and Computer Engineering, Wayne State University. Detroit, USA
[2]National Chiao Tung University. Hsinchu, Taiwan

***Abstract*—This paper presents a comprehensive analysis of hardware accelerators for neural networks in both the digital and time domains, where the latter includes *spatially unrolled* (SU) and *recursive* (REC) architectures. All accelerators are implemented and synthesized in a 65nm CMOS technology. An identical neural network model is implemented in the digital and time domain for comparative purposes in terms of throughput, power consumption, area, and energy efficiency. Post-synthesis results show that SU achieves the highest energy efficiency of 145 $TOp/s/W$ with a throughput of 4 GOp/s. The digital core is the fastest among other cores, whereas REC is the slowest but is the most area-efficient, occupying 0.114 mm^2. SU is more suited for applications with stringent power constraints and average performance, while REC is better suited for applications where the area is the most important requirement and the throughput is less significant. In contrast, the digital core is preferable for large neural networks and critical applications that require high performance.**

***Index Terms*—Digital-Domain Accelerators, Time-Domain Accelerators, Time-Domain Computation, Analog Domain, Spatially Unrolled, Recursive.**

I. Introduction

The tremendous growth of Internet-of-Things (IoT) and energy-constrained devices has urged the need for energy-efficient computing. Artificial Neural Networks (ANNs) and Deep Learning (DL) are the most widely used approaches due to the unprecedented achieved accuracy in image classification, object recognition/detection, and speech recognition. Conventional hardware implementation for DL uses GPUs to execute a a huge number of multiply-accumulate (MAC) operations [1], [2], which consume a tremendous amount of power that is not suited for energy-constrained devices.

An emerging trend is to utilize time-domain (TD) to perform MAC operations by representing the data as pulses with variable delays, and then converting the result back to digital domain [6]–[11]. TD cores require time-to-digital converters (TDCs) and digital-to-time converters (DTCs). However, DTC and TDC can be more energy and area efficient than digital-to-analog (DAC) and analog-to-digital converters (ADC), respectively [12]. Time-based accelerators can achieve superior performance while being energy efficient [6]–[11], [13]. Analog computation is another approach that is efficient in terms of area and energy [3]–[5]; however, it suffers from technology scaling and limited accuracy [11]. The digital approach has the best use of technology scaling, but it is not as efficient as the analog approach [3], [4], [11]. Time-based computation can take advantage of both approaches, analog and digital, as it is energy efficient and can be scaled with CMOS technology.

TD cores can be implemented using two different architectures, including *spatially unrolled* (SU) [6], [7], [9] and *recursive* (REC) [10], [11]. In the SU architecture, inputs and weights are stored in spatially distributed storage elements with a dedicated processing element (PE). Although SU uses many DTC and TDC blocks, the summation comes for free as the pulse propagates through the delay elements. In REC architecture, only one DTC and TDC blocks are used per neuron to perform each MAC, thereby reducing the chip area. Previous studies show that TD computation is more efficient than digital as long as the size of the neural network is relatively small [13] and the number of input bits is fewer than 7 in the case of REC architecture [11]. However, more studies are needed to compare the three architectures: digital, TD-SU, and TD-REC, while considering identical network size. Further, reported studies are based on simulation results and do not consider a hardware-based comparison among time and digital architectures [13].

The main contributions of this paper can be summarized as follows. First, unlike the work proposed in [6], [7], [9], [11], where analog design flow is used or custom analog blocks are needed, such as DTC and TDC, we implement TD cores for both SU and REC architectures using the digital flow synthesized in a 65nm CMOS technology. We also implement an identical digital core using the same technology for comparative purposes. Second, we present a detailed analysis of the results by comparing the performance of the three architectures in terms of throughput, power consumption, area, and energy efficiency.

The rest of this paper is organized as follows. Section II introduces TD computation and compares it with the digital counterpart. Section III discusses the implementation of all-digital TD-SU and TD-REC cores, as well as the digital core. Section IV analyzes and compares the implementation results. Finally, conclusions are drawn.

II. Digital and Time-Domain Architectures

In the digital domain, data are represented as multi-bit digital vector. An example of digital implementation is GPU that computes MAC operations by using a large number of

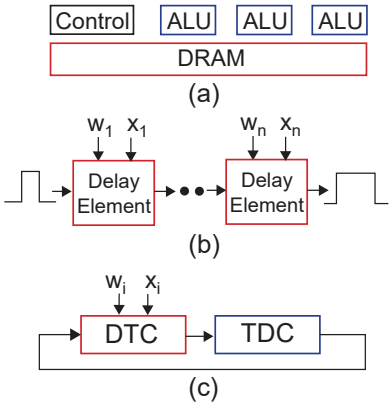

Fig. 1. (a) GPU architecture used to implement digital DNNs (b) Illustration of TD-SU architecture (c) Illustration of TD-REC architecture.

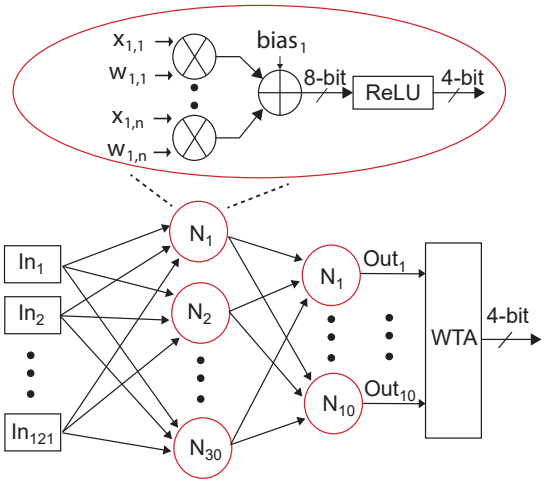

Fig. 2. Digital ANN architecture.

Arithmetic Logic Units (ALU) with the help of memory (DRAM) that stores the weights, as shown in Figure 1(a).

The data in TD are represented as variable pulse widths or time difference in rising/falling edges to generate variable delays. Unlike the analog-based computation, TD circuits can utilize the digital IC design flow, thereby enabling large-scale integration. TD computing is better suited for applications that require low resolution and have stringent power requirements. Figure 1(b) represents the basic implementation of TD-SU neuron. As depicted in the figure, a chain of variable delay elements are cascaded where each element has a delay value that depends on the product between each input and weight bits. An input pulse is applied at the first element, where the output pulse width will eventually represent the MAC operation. One of the main advantages of SU approach is that the addition operation comes for free as the pulse propagates from one delay element to the other.

In REC, only one DTC and TDC are needed per neuron, as shown in Figure 1(c). At each clock cycle, one input and its weight are fed to the DTC to convert them into time and then convert and accumulate the result in the TDC. The most common implementation is to convert the inputs into pulses using a DTC, which will act as an enable to a TDC (counter). The weights are represented as variable frequencies using a digitally controlled oscillator (DCO), which controls the clock of the TDC. The REC architecture 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 scaling the number of input bits and thus 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 TDC, and then the same process is repeated but it is area-efficient. Therefore, SU and REC approaches should be chosen based on the application.

III. Implementation of Digital and Time-Domain Cores

In this section, the implementations of digital, TD-SU, and TD-REC Artificial Neural Network (ANN) cores are presented. A feedforward fully-connected architecture is used with 121 binary inputs, 30 hidden neurons, and 10 output neurons for the three architectures.

A. Digital Accelerator

A digital core is implemented with 1-bit input, and 4-bit signed weights ranging from -3 to 4. In digital cores, if we have n-inputs, then we need n number of processing elements (PEs) per neuron. The ANN has 121 binary inputs, 30 hidden neurons, and 10 output neurons, as shown in Figure 2. In the hidden neuron, each input is multiplied by its weight, and the product terms are added and stored in an 8-bit register. A rectified linear unit (ReLU) is then used, which outputs the 4 most significant bits (MSBs) of the previous neuron output.

The output neuron is identical to the hidden neuron, but implements 4-bit MAC. Finally, winner take all (WTA) function is utilized to select the maximum output among the 10 outputs from the output layer.

B. Spatially Unrolled Accelerator

SU requires n number of PEs for n inputs per neuron. The implemented SU core is similar to the work in [6], however, DTC is implemented using a chain of inverters, as shown in Figure 3(a). The inputs are binary, but the weights are represented in 3-bit from -3 to 4, which are mapped to 0-7 to cover all the cases in 3-bit.

As depicted in Figure 3(b), each neuron has 121-DTCs and 1-TDC, where DTCs are used to implement digitally controlled delays, while the TDC compares the final delay with a reference delay to provide the final digital output. Thus, DTCs act as a multiplier, and as the pulse passes from one element to another, the product terms will be added. When a pulse is applied to the first DTC in the chain, the signal propagates and each DTC will add a delay, based on the values of the weight and the input, where positive weights will result in faster delays.

A 4-bit thermometer code TDC is used to convert the delay signal into digital and implement ReLU. The TDC is a phase

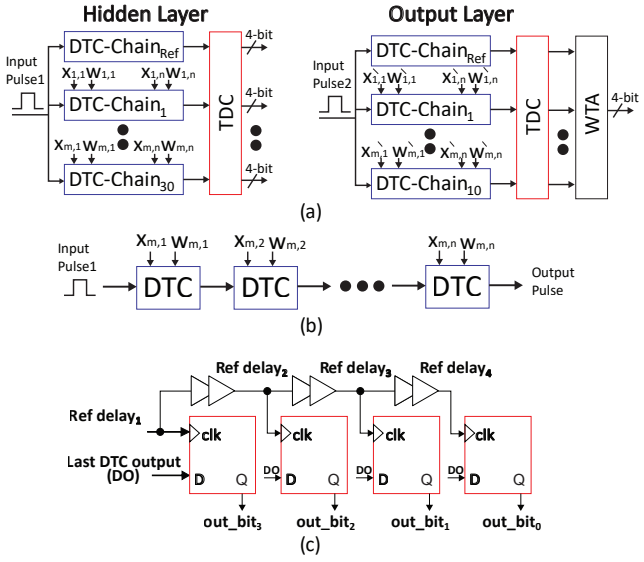

Fig. 3. (a) SU ANN architecture (b) Neuron implementation using cascaded DTCs (c) Implementation of phase detector as TDC.

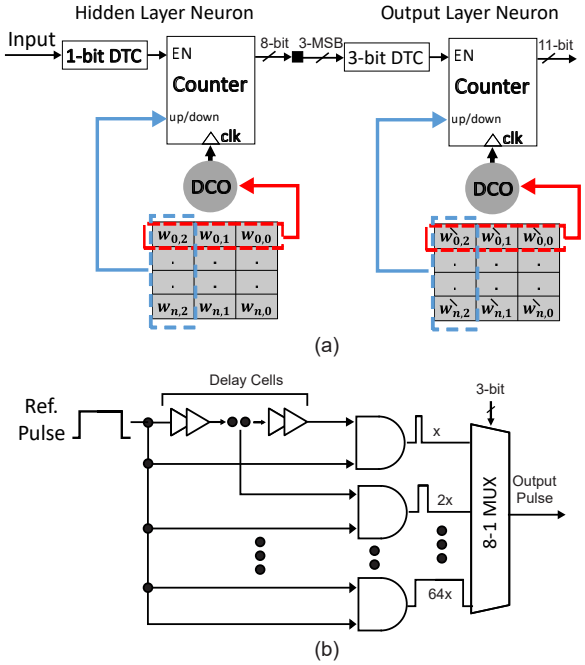

Fig. 4. (a) Implementation of hidden and output neurons of REC architecture (b) Implementation of 3-bit DTC.

detector that will compare the arrival time between 4 reference edges and the neuron output, as shown in Figure 3(c). The reference edges come from a reference delay line that is identical to a normal neuron delay line but with zero inputs. If the neuron output arrives before the fastest reference edge, then all the 4-bits are ones, and if the output arrives after the slowest reference edge, the output will be zero. For the next layer, since we have 30, 4-bit thermometer code inputs, then the output neuron will have 120 DTCs and 1 TDC. Therefore, we apply a second pulse to the first DTC in the line, as shown in Figure 3(a). The delay line in the output layer can be divided into 4 quarters, where we connect the MSBs of all inputs to the first quarter, then the second MSBs to the second quarter, and so on, until we connect the LSBs to the fourth quarter. Lastly, WTA is applied to find the final output.

C. Recursive Accelerator

In REC architecture, each neuron has three main components; DTC, TDC, and DCO, as shown in Figure 4(a). In the hidden neuron, a 1-bit input is converted into time using a DTC. The output of the DTC enables a counter that acts as TDC. Initially, the counter output is set to the binary code 10000000, which is the mid value of the 8-bit counter. The weights are represented in 3-bit digital code, and stored in registers. Each weight is encoded using 3-bit, such that if the MSB is 1, then the weight is positive, and negative if MSB is 0. The MSB will determine whether the counting direction is up or down, if the MSB is 1, then the counter will count up, otherwise it will count down. The 3-bit weight will control the input frequency of the counter by generating different frequencies using a DCO, including f, $2f$, $4f$, and $8f$, where the fastest clock is 800 MHz. The counter also implements a ReLU, by monitoring the MSB of the counter output, if it is 1, then the output is positive, and if it is 0, then the output is negative and can be set to 0. In this way, the MAC operations are computed in TD and accumulated in an 8-bit counter. In other words, the input will control how long the counter will count, and the weight will control the direction and the frequency of the counter. In the output neuron, a digital word, consisting of the first 3 MSBs after the sign bit of the counter, is fed to another DTC. Similar to the hidden neuron, the 3-input bit will be converted to a pulse using a 3-bit DTC, as shown in Figure 4(b). The output of the DTC will be used as an enable signal to the counter, and the final output will be accumulated in a 11-bit counter.

IV. Results and Analysis

The performance of the three architectures digital, TD-SU, TD-REC, is evaluated using the MNIST dataset [14], which consists of grayscale 28×28 pixel images. The images are handwritten digits from $0-9$, and the comparison is based on the classification accuracy. We binarize the images, crop 3 pixels from each edge, then resize them to 11×11 pixels. The neural network has 121 inputs, 30 hidden neurons, and 10 output neurons, and WTA is used to find the final output. The network is trained in MATLAB, and quantized 3-bit weights from -3 to 4 are used for inference. Digital, TD-SU, and TD-REC architectures are synthesized using 65nm standard cells with $1.2V$ supply voltage. Figure 5 shows comparative results of classification accuracy between software and synthesized digital, TD-SU, and TD-REC architectures. The digital approach achieves the highest accuracy of about 89%, very close to the software model which has 90% accuracy. TD-SU has slightly better classification performance than REC due to truncation when moving from one layer to another in REC. In

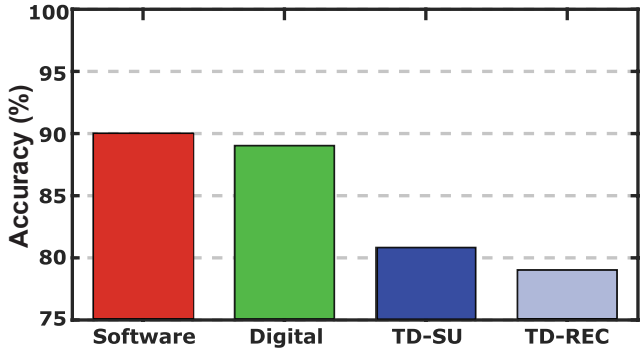

Fig. 5. Classification accuracy of software, digital, TD-SU, and TD-REC.

SU architecture, the phase detector allows to group the values by range, which results in better accuracy.

To analyze the performance of the three architectures, different performance metrics are considered, including frame rate, power consumption, energy efficiency, throughput and area, as shown in Table I. All the architectures are designed to work around their maximum frequency, and then we lower the frequency of the digital core to match SU and REC for comparative purposes. Similarly, SU is designed to work on $16.7MHz$, and then we lower the frequency to be the same as REC. The operating frequency of REC is limited due to the counter resolution, which is controlled by the DCO frequencies. The performance metrics are reported for all operating frequencies, as shown in Table I. The digital core is the fastest, and the least complicated since it does not have delay elements, and thus the timing analysis is much simpler. REC is the most area-efficient since one PE is needed per neuron. However, it has the lowest clock frequency because each MAC operation is done in one smaller clock cycle using the counter in which its clock frequency has to be much faster than the system clock. The reported frame rate in Table I is the clock needed to process each image. SU is a mid-range core that is energy-efficient, and has good processing performance, but this comes at the expense of area, since delay elements need to be cascaded in every neuron. Static and switching power are also reported for the three architectures. If we consider the ratio of switching power to the total power at $1.5MHz$, REC will have the highest ratio of 98%, which means that switching power contributes to 98% of the total power consumption. The digital architecture comes second with about 73%, then SU with 64%. REC has the highest switching power because of the used DCOs in every neuron, where the clock frequency is high and variable. The delay elements in SU are inactive most of the time, and thus it has the lowest percentage of switching power.

To better evaluate the results of the three architectures, energy efficiency and throughput are considered. TD-SU is the most energy efficient at 145.2 $TOp/s/W$ with average throughput. The throughput of the digital core is the highest with 60.5 GOp/s, whereas REC is the slowest. Regarding energy efficiency, the digital and SU architectures provide

TABLE I
DIGITAL VS. TD-SU VS. TD-REC

Architecture	Digital			SU		REC
Frame Rate* (MHz)	250	16.7	1.5	16.7	1.5	1.5
Total Power Consumption (mW)	50.737	3.488	0.424	1.111	0.147	1.545
Static Power (mW)	0.113	0.113	0.113	0.052	0.052	0.027
Switching Power (mW)	50.624	3.375	0.311	1.059	0.095	1.518
Area (mm^2)	0.196			0.331		0.114
Throughput $(GOp/s)^a$	60.50	4.03	0.372	4.03	0.372	0.372
Energy Efficiency $(TOp/s/W)^b$	47.7	46.3	35.1	145.2	101.5	14.2

*Frame rate is the frequency needed to process the next image
[a]Giga (10^9) Operations per Second
[b]Tera (10^{12}) Operations per Second per Watt

better performance at lower frequencies; however, the area efficiency of REC is outstanding. On the other hand, REC consumes 10 times more power than SU and approximately 3.5 times higher than the digital at the same clock frequency. Therefore, the SU architecture is better suited for applications that require stringent power constraints. REC is the best for applications where the area is the primary requirement. In addition, REC is more flexible when it comes to the extension of the input bits since the PE is designed once, and then used every clock cycle; however, for SU the PE needs to be repeated for every input, limiting its scalability. Finally, the SU and REC accelerators are implemented as all-digital, and their performance can be improved if custom-designed analog delay cells, DTC, and TDC are used similar to the work in [6], [7], [9]–[11].

V. CONCLUSION

This paper has implemented a digital core, and two TD cores: spatially unrolled (SU) and recursive (REC), using digital design flow in a 65nm CMOS. A detailed analysis is provided for all architectures using the same network topology and dataset. The performance of the implemented cores is evaluated using different metrics, including throughput, area, power, and energy efficiency. Results of the synthesized networks show that digital is the fastest but has average energy efficiency. REC is the most area-efficient, but it is the least efficient in terms of speed and energy. SU is the most energy-efficient with average throughput, but it has the largest area.

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] M. D. Edwards, H. Al Maharmeh, N. J. Sarhan, M. Ismail and M. Alhawari, "A Low-Power, Digitally-Controlled, Multi-Stable, CMOS Analog Memory Circuit," 2020 IEEE 63rd International Midwest Symposium on Circuits and Systems (MWSCAS), Springfield, MA, USA, 2020, pp. 872-875.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] H. Al Maharmeh, N. J. Sarhan, C. -C. Hung, M. Ismail and M. Alhawari, "Compute-in-Time for Deep Neural Network Accelerators: Challenges and Prospects," 2020 IEEE 63rd International Midwest Symposium on Circuits and Systems (MWSCAS), Springfield, MA, USA, 2020, pp. 990-993.

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