Energy-Efficient Pulse-based Convolution Engine for Near-Sensor Processing

M. Hassan Najafi*, S. Rasoul Faraji†, Kia Bazargan†, and David Lilja†
najafi@louisiana.edu, faraj008@umn.edu, kia@umn.edu, lilja@umn.edu
*School of Computing and Informatics, University of Louisiana at Lafayette, Lafayette, LA, USA
†Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN, USA

ABSTRACT

Near-sensor convolution engines have many applications in Internet-of-Things. Pulsed unary processing has been recently proposed for high-performance and energy-efficient processing of data using simple digital logic. In this work, we propose a low-cost, high-performance, and energy-efficient near-sensor convolution engine based on pulsed unary processing. The proposed near-sensor engine removes the necessity of using costly analog-to-digital converters. Synthesis results show that the proposed pulse-based design significantly improves the hardware cost and energy consumption compared to the conventional fixed-point binary-radix and also to stochastic computing-based designs.

KEYWORDS

Unary processing, near-sensor processing, convolution engine, time-based encoding, energy-efficient design.

1 MOTIVATION

Near-sensor computing has received considerable attention in the era of Internet-of-Things (IoT). By integrating some of the processing circuits with the sensing device, we can avoid significant overheads of memory and network communication costs. This often results in considerable hardware area and power cost saving. Convolution is a widely used function in different applications, particularly in neural networks (NNs). Conventional fixed-point binary radix-based and also stochastic computing (SC)-based implementations of convolution engines have been previously proposed. Fixed-point binary designs are fast and accurate but complex and costly. SC-based designs are low-cost but incur very long latencies and so very high energy consumption. They also lack the accuracy of fixed-point designs. These drawbacks beside the high hardware cost of the analog-to-digital (ADC) and analog-to-stochastic (ASC) converters make the traditional designs inefficient for near-sensor processing. In this work, we propose a mixed-signal design methodology for energy-efficient near-sensor design of the convolution engines. We exploit the idea of pulsed unary processing [5, 7] in efficient implementation of convolution functions.

2 RELATED WORK

Stochastic computing (SC) [2], an unconventional paradigm processing random bit-streams, has been used in low-area and low-power design of arithmetic functions. Independent of the length, the ratio of the number of 1s to the length of the bit-stream determines the value of a bit-stream. Multiplication as a complex and common operation in convolutions can be implemented using simple standard AND gates in stochastic domain. Authors in [3] propose a hybrid stochastic-binary NN for near-sensor computing. They use SC to implement the first convolutional layer of the NN. An analog-to-stochastic converter (ASC) is used to convert the sensor data directly to bit-stream representation. While the proposed design of [3] shows promising results compared to prior SC-based designs, the random fluctuations in generating bit-streams and the cost of ASCs limit its application for efficient near-sensor design of convolution engines.

Authors in [1] propose a hybrid deterministic bit-stream-binary design to remove the random fluctuation problem of [3]. By converting data to low-discrepancy (LD) bit-streams they perform accurate multiplications and achieve the same result as the conventional fixed-point binary design. Similar to the design of [3], the bit-stream processing is only used in the first convolutional layer of the NN. A requirement for this design is that the input data must be in the digital binary format to compare with LD random numbers to generate bit-streams. In case of near-sensor processing, therefore, the analog input data from sensor must first be converted to digital binary format using costly ADCs and then be converted to LD bit-streams.

Pulse-based processing is a hybrid mixed-signal computation technique combining the advantages of analog and digital designs. We point to [4] as a study of early works in this area. Multiplication of input data represented using pulsed signals was always an important challenge in early pulse-based designs. Pulsed (or time-encoded) unary processing was introduced recently for high-performance processing of data using low-cost SC circuits [5]. The designs inherit the low-cost advantages of SC but produce deterministic and accurate results in a significantly shorter time compared to the conventional digital bit-stream-based SC designs. Multiplication, scaled addition, and absolute value subtraction of pulsed unary signals are discussed in [6]. Maximum and minimum value functions based on pulsed unary processing are

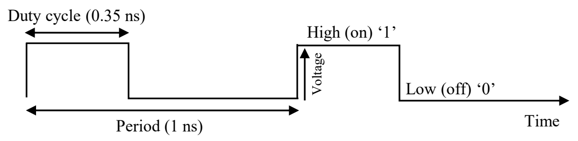

Figure 1: A periodic PWM signal. The value represented is the fraction of the time that the signal is high in each cycle–in this case, 0.35.

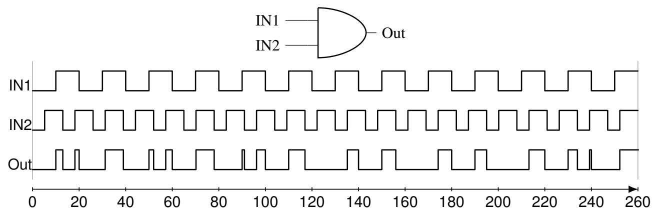

Figure 2: Example of multiplying two (unipolar) PWM signals using an AND gate. IN1 represents 0.5 with a period of 20 ns, and IN2 represents 0.6 with a period of 13 ns. The output signal represents 0.30 (78 ns/260 ns), the expected value from multiplication of the inputs.

discussed in [7] for high-performance and energy-efficient design of sorting networks circuits.

3 PULSED UNARY PROCESSING

Unary processing [8] is a hybrid information processing technique that has characteristics common to both the conventional binary and to SC. It is deterministic, but borrows the concept of averaging from stochastic methods [7]. Input data is encoded uniformly by a sequence of one value (say 1) followed by a sequence of the other value (say 0) in a stream of 0's and 1's. Similar to stochastic bit-streams, the value of a unary bit-stream is defined by the ratio of 1's in the bit-stream. For example, 1100 and 111000 are two unary bit-streams representing 0.5. The representation of numbers in unary processing is not limited to digital serial bit-streams. A time-based interpretation of numbers is also possible by pulse modulation of data as shown in Figure 1. Pulse-width modulated (PWM) signals can be treated as the inputs of unary processing circuits with values defined by their duty cycle.

Multiplication of pulsed unary data has been recently discussed in [6] by converting data into inharmonic PWM signals and logical ANDing the generated signals. The duty cycle of the PWM signals is set to the value represented. Relatively prime periods (inharmonic frequencies) are selected for the input signals and the operation is run for the product of the periods to produce deterministic and accurate result. Figure 2 shows an example of multiplying two input data in the pulsed unary domain.

Standard OR gates can be used for addition of pulse signals. The OR-based addition, however, is distorted by pulse overlap and can only be used when adding a small number

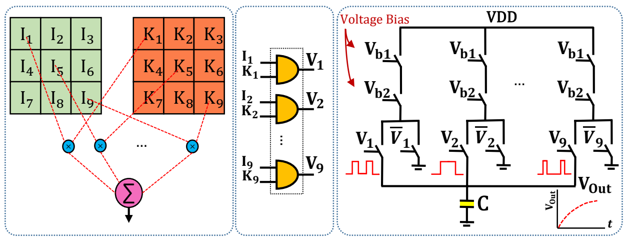

Figure 3: A 3x3 Convolution Engine: (left) convolution process (middle) multiplications using AND gates (right) accumulation using active integrator.

of inputs [4]. Current summation via an active integrator is a better choice to the OR-based addition. It does not incur any loss of activity information and does support adding a larger number of inputs. We should, however, consider that the output from current summation is no longer a pure pulse signal; it usually is averaged, and must be connected to a pulse-width modulator to regenerate a PWM signal [4].

While pulsed unary processing is deterministic, it comes at the cost of a slight accuracy loss. The frequency of analog-to-time-converters (ATCs) and so the frequency of the generated pulses affects the effective number of bits (ENOB) in representing and processing data [6]. The lower the frequency, the higher the ENOB. Imperfect generation of PWM signals and error in measuring the output signals are the main sources of inaccuracy in pulsed unary processing.

4 PROPOSED DESIGN

An $N \times N$ convolution engine consists of N^2 multiplication operations and a summation operation accumulating the results. In our proposed near-sensor convolution engine, we first convert all input data to pulse signals. This can be done using an ATC such as the PWM signal generator proposed in [6]. Two inharmonic frequencies are selected for the input signals, each for one input of every multiplication. Depending on the desired accuracy, we can adjust these frequencies. Outputs of AND gates are connected to an active integrator accumulating the multiplications results. Figure 3 shows the proposed convolution design for $N = 3$. One frequency will be used for all "I" values, and another one for all "K" values.

The proposed near-sensor engine includes three blocks. The first block is a PWM signal generator which converts the analog input data to a PWM signal with corresponding duty cycle. The second block includes some AND gates to do the multiplication operations in the pulsed unary domain. The third block is a time-to-voltage converter which accumulates and integrates the output signals over time and generates an analog voltage. If necessary, the generated voltage can then be converted to a digital format using an ADC for further processing in the digital domain.

In the developed convolution engine, we use a low-cost and high-performance integrator to integrate the outputs

Table 1: The parameters of the proposed integrator

Convolution Design	V_{b1}	V_{b2}	$W(nm)$	$L(nm)$
3×3	0.6	0.5	35×45	2×45
5×5	0.575	0.48	40×45	2×45
7×7	0.57	0.448	50×45	2×45

of the AND gates in the analog domain. The integrator first converts the output signals to their corresponding currents and then integrates them over time in a capacitor. The integrator uses the same size current source for all its inputs. Each input sinks a current into the capacitor based on the length of its high parts. We used a cascode structure with two PMOS transistors to implement each current source. The two PMOS transistors are used to route the current from the source to the capacitor and also to reduce the effect of clock feed-through. In the high phase of the output signal, one of these transistors sinks the current into the capacitor and in the low phase, the other transistor sinks the current into the ground. With this technique, we keep a voltage at the output of the current source, increase the linearity, and reduce the effect of clock feed-through on the capacitor.

The developed circuit can work linearly only for a specific part of the input range. As a result, it cannot produce a full range of output. The output, however, can be amplified using a simple linear amplifier to feed the next stage. The circuit parameters can be adjusted accordingly based on the application specification (e.g., for 3×3, 5×5, or 7×7 convolution windows). The proposed circuit is designed for two clock periods of 5 ns and 6 ns as the periods of the input PWM signals. Therefore, the output can be captured every 30 ns by the next stage. Table 1 shows the proposed integrator parameters for three different convolution windows.

5 EVALUATION

We evaluate the efficiency of the proposed pulse-based convolution engine in terms of area footprint, latency (critical path delay×number of cycles), power, and energy consumption compared to the conventional digital binary and also to the SC-based designs. Designs are compared for N=3, 5, and 7; We assume that the input data is coming from sensor and are in analog voltage/current format. For the conventional digital binary approach we implement an 8-bit precision fixed-point design. For this case the input data must be converted to digital binary format using an ADC. For the SC-based design we perform multiplication operations in the unipolar bit-stream domain. Similar to [1] the output bit-streams are accumulated using regular binary adders. The input data can be converted to stochastic bit-streams using an ASC, or an ADC plus a digital binary-to-stochastic converter. We convert the input data to 64-bit long LD bit-streams by comparing them in digital binary-radix format to the first 64 numbers from the MATLAB's built-in first two LD Sobol sequences. For the proposed pulse-based design we convert the input data to PWM signals with 5 ns and 6 ns periods. The outputs of multiplications are therefore ready after 30 ns. We use HSPICE pulse generator to generate the PWM signals. The outputs are accumulated using an active integrator as shown in Figure 3. The fixed-point binary and the SC-based designs are synthesized using the Synopsys Design Compiler v2018.06-SP2 with a 45-nm standard cell library. The same library is also used to estimate the cost of the pulsed unary design. Performance and power consumption of the pulse-based designs are evaluated using the Synopsys HSPICE v2018.09. We do not include the cost of senor data conversion in our evaluation. Prior work has shown that ASCs are as costly as regular ADCs [3]. ADCs are more costly than ATCs. If we were to incorporate this cost, the proposed pulse-based method would have shown even further improvements.

Table 2: Synthesis Results

Convolution Design		Area (μm^2)	Delay (ns)	Power (mW)	Energy (pJ)	MAE (%)
3×3	8-bit Fixed-Point	9892	1.4	8.78	12.7	0.0
	Stochastic-128	671	78.1	1.21	94.7	1.87%
	Pulsed Unary	107	30	0.09	2.8	0.85%
5×5	8-bit Fixed-Point	29482	1.7	20.70	35.1	0.0
	Stochastic-128	877	99.8	1.29	128.5	3.81%
	Pulsed Unary	607	30	0.38	11.4	1.14%
7×7	8-bit Fixed-Point	57651	1.9	33.71	64.0	0.0
	Stochastic-128	1177	116.5	1.61	187.3	17.23%
	Pulsed Unary	782	30	0.99	29.9	1.67%

Table 2 shows the synthesis, and the performance evaluation results. As can be seen, the proposed pulsed unary designs have the minimum hardware area cost. The area saving in the pulsed unary designs compared to the stochastic designs is due to using an active integrator in accumulating the multiplications results. The power and energy consumption are also significantly improved. We evaluated the performance of each design by finding the mean absolute error (MAE) of 1000 trials performing convolution on 1000 random sets of input data. For the SC and pulsed designs we set the expected results of greater than 1.0 to 1.0. The last column of Table 2 shows the MAE of each design. The proposed designs have shown a lower accuracy compared to the fixed-point but a better accuracy compared to the SC designs. Figures 4, 5, and 6 show the error analysis of the evaluated pulsed designs for the 1000 random input sets. The x-axis shows the expected output voltage while the y-axis shows the absolute error between the expected and the measured values.

With more and more sensors providing time-encoded outputs, a large number of the proposed pulse-based convolution engine can be used in parallel near sensor to immediately process data. This eliminates the need for costly ADCs/ASCs and also avoids wasting resources on memory and network communications.

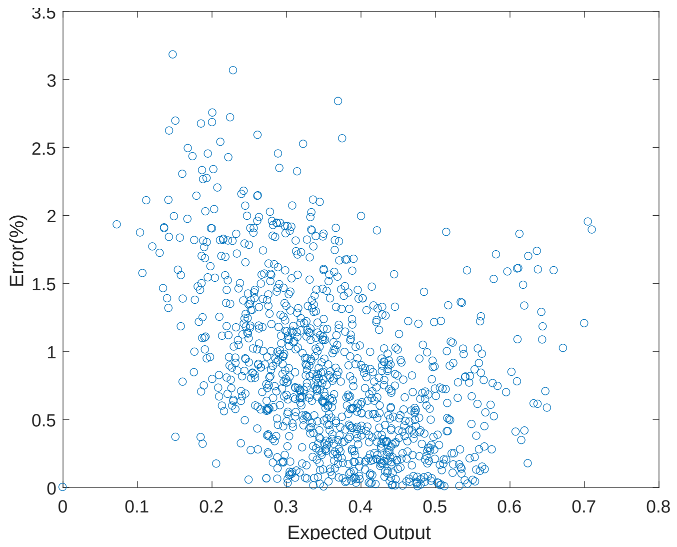

Figure 4: Error analysis of 9-input the proposed integrator. The MAE is 0.855%**.**

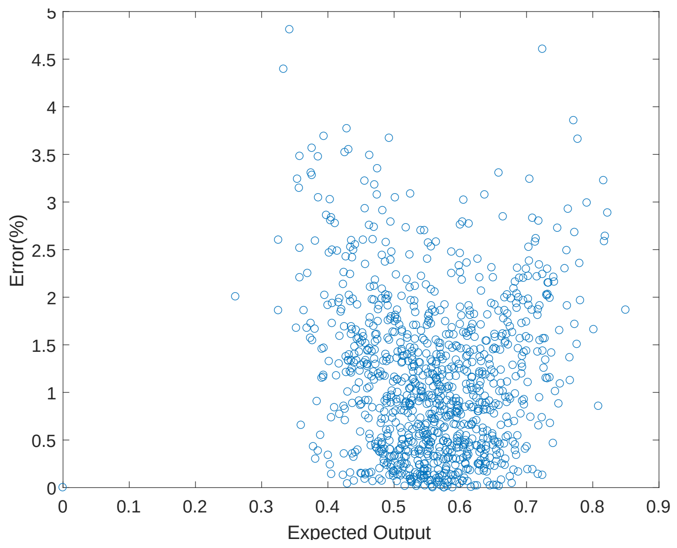

Figure 5: Error analysis of 25-input the proposed integrator. The MAE is 1.14%**.**

6 CONCLUSION

Pulsed unary processing combines an analog time-based representation of data with digital processing using simple logic gates. In this work, we proposed a low-cost, high-performance, and energy-efficient near-sensor convolution engine based on pulsed unary processing. The proposed design is compatible with the data provided by the sensors avoiding costly ADCs. The down-side is some inaccuracies due to the mixed-signal nature of the design. The inaccuracy, however, can be masked and tolerated by the application (e.g., neural networks).

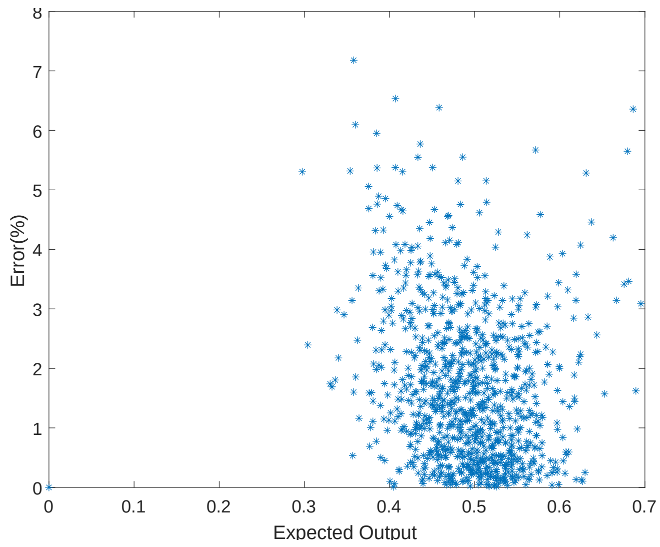

Figure 6: Error analysis of 49-input the proposed integrator. The MAE is 1.675%**.**

REFERENCES

[1] S. Rasoul Faraji, M. Hassan Najafi, Bingzhe Li, Kia Bazargan, and David J. Lilja. 2019. Energy-Efficient Convolutional Neural Networks with Deterministic Bit-Stream Processing. In *Design, Automation, and Test in Europe (DATE), 2019.*

[2] B.R. Gaines. 1969. Stochastic Computing Systems. In *Advances in Information Systems Science.* Springer US, 37–172. http://dx.doi.org/10.1007/978-1-4899-5841-9_2

[3] V. T. Lee, A. Alaghi, J. P. Hayes, V. Sathe, and L. Ceze. 2017. Energy-efficient hybrid stochastic-binary neural networks for near-sensor computing. In *Design, Automation Test in Europe Conference Exhibition (DATE), 2017.* 13–18.

[4] Wolfgang Maass and Christopher M Bishop. 2001. *Pulsed neural networks.* MIT press.

[5] M. Hassan Najafi, Shiva Jamali-Zavareh, David J. Lilja, Marc D. Riedel, Kia Bazargan, and Ramesh Harjani. 2017. An Overview of Time-Based Computing with Stochastic Constructs. *IEEE Micro* 37, 6 (November 2017), 62–71.

[6] M. H. Najafi, S. Jamali-Zavareh, D. J. Lilja, M. D. Riedel, K. Bazargan, and R. Harjani. 2017. Time-Encoded Values for Highly Efficient Stochastic Circuits. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 25, 5 (May 2017), 1644–1657.

[7] M. Hassan Najafi, D. J. Lilja, M. D. Riedel, and K. Bazargan. 2018. Low-Cost Sorting Network Circuits Using Unary Processing. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 26, 8 (Aug 2018), 1471–1480.

[8] W.J. Poppelbaum, A. Dollas, J.B. Glickman, and C. O'Toole. 1987. Unary Processing. In *Advances in Computers.* Vol. 26. Elsevier, 47 – 92.