A Time-Domain Current-Mode MAC Engine for Analogue Neural Networks in Flexible Electronics

Matthew Douthwaite*, Fernando García-Redondo †, Pantelis Georgiou* and Shidhartha Das†
*Centre for Bio-Inspired Technology, Dept. Electrical and Electronic Engineering, Imperial College London, SW7 2AZ, UK
†ARM Research, Cambridge, UK
Email: mdouthwaite@imperial.ac.uk, fernando.garciaredondo@arm.com, pantelis@imperial.ac.uk, sdas@arm.com

***Abstract*—Flexible electronics is becoming more prevalent in a wide range of applications, particularly wearable biomedical devices. These devices would greatly benefit from in-built intelligence allowing them to process data and identify features, in order to reduce transmission and power requirements. In this work, we present a novel time-domain multiply-accumulate (MAC) engine architecture that can act as the basic block of an artificial analogue neural network. The design does not require analogue voltage buffers, making them easier to realise in flexible technologies and consumes less power than conventional methods. The research could be used in future to construct a low power classifier for a low cost, flexible wearable biomedical sensor.**

***Index Terms*—Flexible Electronics, MAC Operation, Neural Networks, Analogue Signal Processing, Wearable Sensors**

I. Introduction

For over a decade, flexible electronics have become increasingly popular in applications requiring wearable biomedical sensors [1]. The use of flexible electronics enables a more conformal skin-sensor interface, which allows them to be worn for a longer period of time and provide more accurate results. Additionally, they are also generally very low cost and hence can be disposable [1]. While many manufacturers now provide flexible PCBs that can house off-the-shelf integrated circuits (ICs), the truly conformal devices are thin-film electronics or conductive substances which are printed onto polymers. In wearable applications, these designs are often referred to electronic 'skin' or 'tattoos' [2].

While there are many examples of flexible sensors [3], [4], these devices could benefit greatly from efficient data processing that reduces power and transmission requirements. There have been several works showing CPUs implemented in thin-film technology on flexible substrates since 2000 [5], [6], and the first SoC (System on Chip, 32-bit CPU with memory and peripherals) research prototype on plastic was announced by *Arm Ltd.* and *PragmatIC* in 2015 [7], [8]. However, these technologies are in relatively early stages of development and non-complementary (Pseudo-CMOS). With resistive [8] or NMOS [6] pull-ups, they are susceptible to high leakage currents. While incorporating full-scale SoCs would provide adequate processing, wearable flexible sensors are typically short-life and single-use, meaning that these energy hungry, complex systems would be too costly to use in this application. In contrast, designing a bespoke classifier to identify patterns from the sensor data automatically could save significant power, area and cost [9].

A conventional classifier using standard digital cells will still suffer from leakage. Conversely, analogue computing has long been known to provide advantages of energy efficiency and real-time computation [10]. Furthermore, bio-inspired neural networks are increasingly being used to classify sensors data and a number of groups have presented analogue-based neural networks in recent years [11]–[13].

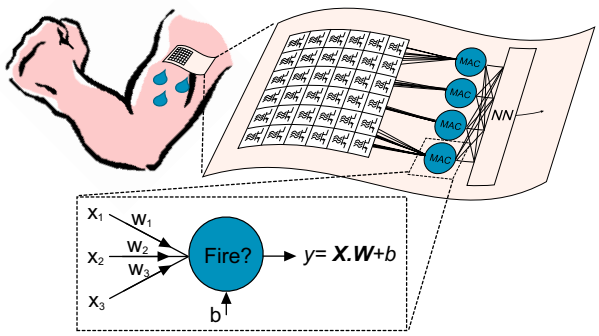

Fig. 1. A concept image of a neural network on flexible electronics interfacing to a chemical sensing array. The operation of a single neuron is shown, X is a vector of N inputs, W is a vector of N weights, b is a bias term and y is the output value.

Implementing analogue circuits in pseudo-CMOS technology can be challenging, in particular due to difficulty in buffering signals. In this research, we propose utilising time-domain encoding to avoid these challenges. Previously, we have used time encoding in electrochemical sensor front-ends to create robust, low-power sensor arrays [14]. By creating a time-domain analogue neural network in flexible technology, it is feasible to integrate these two systems to create a bio-inspired neural network with electrochemical sensor inputs for a wearable and disposable biomedical device, as illustrated in Fig. 1.

Thus, this paper presents a novel design of a multiply-accumulate (MAC) engine, the core building-block of a neural network, which uses time-domain inputs to create a robust architecture applicable to wearable flexible biomedical devices. We demonstrate a 10 times reduction in power consumption over the digital equivalent implementation by eliminating leakage currents inherent to pseudo-CMOS technologies, while acheiving close to ideal accuracy and using fewer gates and overall all less area.

In Section II we cover some fundamental principles of artificial neural networks. Section III then covers challenges specific to analogue neural networks, before Section IV introduces our proposed architecture. Section V presents the results of this architecture before we conclude and discuss future work in Section VI.

II. Neural Network Fundamentals

A neural network typically consists of three layers; an input layer, a number of fully connected hidden layers and an output layer. Each layer consists of a certain number of nodes, or 'neurons'. In the input and output layers, the number of neurons is dictated by the number of inputs and outputs to the neural network respectively. The number of neurons in each hidden layer, and quantity of hidden layers overall, gives an indication of the complexity of the network. This concept takes inspiration from biology, where the inputs to a neuron determine how likely it is to fire. The artificial neuron, shown in Fig. 1, takes in a set of inputs, multiplies each by a weight, and adds the total sum to a bias. This is called a multiply accumulate (MAC) operation, the result of which represents the importance of that particular neuron to the network. Finally, a non-linear - or 'activation' - function, is applied to the result of the MAC operation to give the neural network non-linearity. A MAC operation or set of parallel MAC operations followed by a decision, without the activation function, can be referred to as a linear classifier, and hence the MAC engine is a basic building block of a neural network. For a single class (output), where a decision is made based upon a threshold, a linear classifier can be represented by a simple equation, as shown in Fig 1. Typically, if there is just one class, the output y is a single value and the input data is said to be part of that class if y is greater than 0, and not otherwise. This model can then be extended to multiple classes, as shown in (1).

$$\mathbf{Y}_{1\times M} = \mathbf{X}_{1\times N}.\mathbf{W}_{N\times M} + \mathbf{B}_{1\times M} \tag{1}$$

In this case there are M classes, so X is still a vector of N inputs, W is now an N x M matrix of weights, B is a vector of M biases and Y is a vector of M outputs. The correct class can be determined by the maximum value in Y.

III. Analogue Computation for Neural Networks

A digital implementation of a MAC operation with instantiated digital multipliers and adders requires a large number of gates per operation which grows with network size. More recently, methods of using the inherent physics of electrical components (for example, Ohm's law and Kirchoff's Current Law) have been reported to carry out these multiply and add operations in the analogue domain for an area and energy efficient solution. Furthermore, with the outputs of many biological sensors in the analogue domain, it can be more efficient to process in analogue before converting to digital.

A. Crossbars

A typical architecture for performing matrix multiplication with analogue components is by using a crossbar network [12], [15]. A crossbar network encodes the weights as a matrix of conductances (resistors) which connect each input row and output column. If the column potential is assumed to be 0 V, and the inputs are applied as an analogue voltage on the rows, then the current flowing out of each column is given by (2). This architecture is now particularly being used for memristor-based networks [12]. Thus, by simply arranging these passive components in a crossbar network, a real-time multiply accumulate engine is created.

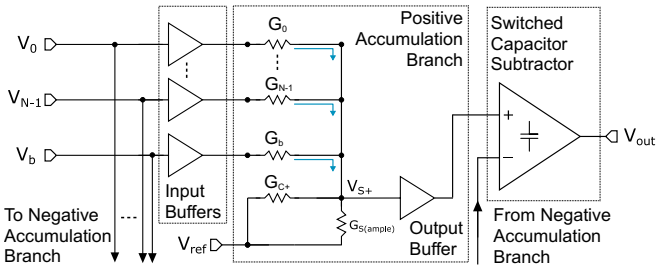

Fig. 2. The crossbar architecture for magnitude-encoded inputs. Implemented for comparison purposes in this work.

In many neural network models, inputs and weights can also be negative. To obtain an equivalent result to (2), we split the positive and negative input-weight products into two accumulations and the difference is then calculated. To implement this, we duplicate the crossbar network to create two branches for positive and negative weights respectively. Each branch uses the absolute values of the weights. We then use a switched capacitor subtractor to compute the final result. Negative inputs also need to be considered, as this changes which accumulation the input-weight product should be added to. To solve this, we raise the potential of the crossbar output column to a virtual ground which corresponds to a zero input. Consequently, negative inputs below this potential cause currents to leave the branch and reduce the accumulation. Fig. 2 shows a schematic of this solution. Additional features of this schematic are discussed in Section III-B.

$$I_{out} = V_{in,0}G_0 + V_{in,1}G_1 + ... + V_{in,N-1}G_{N-1} \tag{2}$$

B. Crossbar Challenges

One of the main challenges in an analogue crossbar network is the loading effect that occurs when inputs are shared between multiple output classes. Each subsequent branch adds additional load impedance to the input row, affecting the input magnitude and reducing the accuracy of the overall network. This is known as sneak current [11]. Therefore for a larger network, either compensation schemes or many buffers are required for each branch and input. To be confident that an input signal is reaching a branch without degradation, N x M buffers are required, which is infeasible in a large system. Furthermore, in pseudo-CMOS technologies it is difficult to create a conventional analogue buffer. A basic source follower (SF) is a functioning alternative, but suffers from non-linearity and loading effects introducing further errors to the network. Consequently, to ensure the architecture shown in Fig. 2 operates correctly, an SF buffer is used at every input in both branches. A compensation conductance, G_{C+}, is used to balance the overall branch impedance with that of the negative branch, ensuring the buffer gain is matched.

To overcome the challenges of analogue buffering, this work proposes to encode inputs in the time-domain. This approach, presented in the following section, improves robustness by removing the dependence on the amplitude of the signal, hence the loading effect of a large resistive array is not critical.

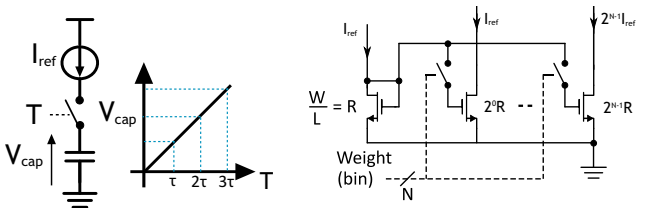

Fig. 3. The capacitor voltage V_{cap} increases linearly with a constant current *a)*, which can be controlled with the binary weighted current mirror *b)*.

IV. The Proposed Architecture

Time encoding of data is a popular technique whereby the signal is quantised as a pulse width. However, sampling in time requires a capacitive storage which, if implemented in a crossbar architecture with constant voltage inputs, would induce a non-linear charge curve. Instead, our solution departs from the typical crossbar architecture to charge a capacitor linearly using a constant current, as shown in Fig. 3a, which we can achieve using current mirrors provided the output device remains in saturation. In this section, we describe how to map inputs and weights to this design to obtain equivalent operation to the crossbar architecture to emulate a MAC engine.

A. Representing Inputs and Weights in Current Mode

To encode an input in time, it is straightforward to represent the magnitude as a pulse width. To represent weights, we can use the property of current mirrors that output to input ratio is dependent on the W/L ratios of the devices used. Thus, we can use weighted current mirrors to represent a full range of weights, as shown in Fig. 3b. With the outputs connected together, these weighted mirrors create a current proportional to the sum of the neuron's weights, given by (3). By connecting the input pulses to the gate switches of the corresponding output device, we obtain a current profile that when integrated gives a total charge equivalent to the weighted sum of inputs.

$$I_{out}(t) = I_{ref}\frac{G_0}{G_U} + I_{ref}\frac{G_1}{G_U} + ... + I_{ref}\frac{G_{N-1}}{G_U} \quad (3)$$

We now connect the output current to a capacitor, which is reset by connecting both terminals to VDD. The charge delivered to the capacitor during the calculation window gives it a final voltage that is a weighted sum of the input-weight products, given by (4). This is equivalent to multiplying (2) by a sampling resistance, 'R'.

$$V_{c-out} = \frac{1}{C}\sum_{i=0}^{N-1} I_{out,i}T_{in,i} \quad (4)$$

We also need to consider negative inputs and weights. The latter can be dealt with by using two branches and a subtractor, as explained in Section III-A, but negative inputs must be handled differently, since the direction of current flow can't be changed in a current mirror. In the proposed architecture, shown in Fig. 4, is a capacitor in each branch, one accumulating the positive input-weight products, the other the negative. Inputs can be directed to the appropriate accumulation using a demultiplexer, which directs the input pulses to the correct branch. If the weights are fixed, the input control signal, X_{SW}, can be connected for each input depending on the sign of the corresponding weight.

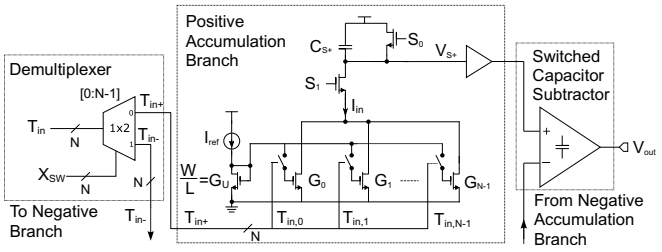

Fig. 4. Final architecture of the current mode, time encoding architecture.

V. Simulation Results

The time encoding architecture (TEA) proposed in Section IV was simulated and results were compared to a crossbar design described in Section III-A, henceforth known as magnitude encoding architecture (MEA). Both architectures were designed and simulated in *PragmatIC*'s sub-micron process [8]. The MEA uses source follower buffers and a virtual ground to handle negative inputs. The key metrics evaluated were accuracy and power consumption. For comparison, an equivalent MAC engine was synthesised from VHDL code.

A. Data Sets

Accuracy was tested using a set of 1000 generated data points, each with 8 'properties', corresponding to inputs, and assigned to one of three output classes. A model was then created with a set of weights and a bias to give an ideal accuracy, allowing comparison of the MAC engine model accuracy with a benchmark. The data points were converted into voltages or pulse widths, and the voltage output of each neuron were compared through a 'maximum' function to determine the chosen class.

B. Accuracy Results

As shown in Table I, the accuracy of the MEA is very dependent on the value set for the virtual ground (i.e. the potential of the column on the crossbar network). While the inputs are scaled such that 2.25V corresponds to '0', there is a 10.6% accuracy loss compared to the model. This is due to the loading effect on the buffers described in section III-B. By setting this virtual ground to 2.13 V, the value of a zero input into the non-ideal loaded buffer, the accuracy improves to 89.3%, a loss of 0.6%. The solution of adjusting the virtual ground is not practical however, as it would have to be tuned for each new set of weights.

The TEA, for the same size network, achieves an 89.6% accuracy, a loss of 0.3%. Increasing the time allowed for the subtraction capacitor to sample the two accumulation values actually causes the percentage accuracy to increase above that of the model by 0.2%, which is believed to be a feature of the data used. The accuracy of this architecture is also affected by the balance between the size of the storage capacitor and the maximum input pulse width. If the capacitor is too small when the maximum input is applied, the voltage stored will increase to the level where the current mirrors drop out of saturation, causing the charging current to decrease.

The two architectures were also compared in a different model with two additional classes. The new model had a different set of weights and new expected accuracy of 76.7%. It

TABLE I
ACCURACY RESULTS FOR 8 X 3 NETWORKS

Model	Condition	Accuracy
Ideal Model	8 Inputs, 3 Outputs	**89.9 %**
Magnitude encoding	Virtual ground of 2.25V	79.3 %
	Virtual ground of 2.13V	**89.3 %**
Time encoding	Sampling time of 20µs	89.6 %
	Sampling time of 70µs	**90.1 %**

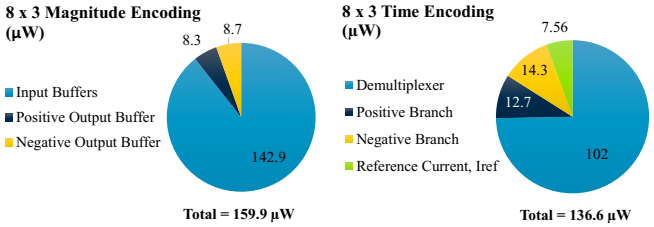

Fig. 5. Power consumption of the presented analogue architectures.

was found that the TEA maintained it's high accuracy without adjusting any parameters, achieving 78.6%. In contrast the MEA accuracy dropped significantly to 20.3%. This is due to the loading effects on the buffers, showing that the network would need to be re-tuned for each new set of weights.

C. Power Consumption and Performance Comparison

The total power consumption for the two 8 x 3 architectures was simulated and averaged over a computation cycle. A breakdown is shown in Fig. 5. The TEA shows a lower power consumption of 136.6 µW, as it does not require the analogue buffers which account for the bulk power loss in the MEA. It can also be seen that the majority of power (almost 75%) in the TEA is consumed by the demultiplexers.

An equivalent digital implementation of the same 8x3 MAC engine array was synthesised for comparison with the analogue architectures. The comparison is shown in Table II. Six-bit values were used for input and output signals and the weights were fixed. Once again, due to the pseudo-CMOS nature of the technology, leakage dominates power consumption, accounting for 98.5% of the all-digital total, although there was no optimisation of the design to reduce this. Overall, these simulations show that in this technology analogue architectures consume significantly lower power (98.8% and 99% less for the MEA and TEA respectively) compared to the equivalent digital implementation.

Finally, a figure of merit (FoM) summarises the performance of the architectures as million operations per second per watt (MOPS/W), and highlights the power efficiency of the analogue approaches. The choice between the MEA and TEA will be dictated by the application requirements for speed, area and accuracy. As described in Section III-B, problems requiring larger networks are expected to scale better using the proposed TEA approach.

VI. CONCLUSION

This paper has presented a time-domain current-mode architecture for an analogue neural network MAC engine on flexible electronics. By using this modality, errors due to non-ideal buffers and loading effects are negated, as the weights

TABLE II
A COMPARISON OF ANALOGUE AND DIGITAL IMPLEMENTATIONS SIMULATED IN THE *PragmatIC* PROCESS

Attribute	All-Digital Implementation	Magnitude Encoding	Time Encoding
Accuracy	89.9 %	89.3 %	90.1 %
Power (µW)	195 µW (Switching) 13 mW (Leakage)	159.9	136.6
Compute Time (µs)	10	50	150
FoM (MOPS/W)	364	6004	2343

of each branch do not affect the input currents. The proposed architecture achieves close to model accuracy and consumes 137 µW on average, lower than a conventional analogue implementation and an order of magnitude lower than an equivalent digital implementation in this technology.

As a result, this technique could provide a viable method of implementing neural networks in flexible electronics, enabling intelligence in sensing nodes. This could particularly be applied to wearable chemical sensors where there is a large amount of information available in which patterns of physiological changes could be identified.

ACKNOWLEDGEMENTS

The authors thank PragmatIC and Emre Ozer, Andy Kufel and James Myers of ARM Ltd. for their support. This research is supported by EC Horizon 2020 Research and Innovation Program through MNEMOSENE project under Grant 780215.

REFERENCES

[1] Y. Liu *et al.*, "Lab-on-Skin: A Review of Flexible and Stretchable Electronics for Wearable Health Monitoring," *ACS Nano*, vol. 11, no. 10, pp. 9614–9635, 2017.
[2] A. J. Bandodkar *et al.*, "Tattoo-Based Wearable Electrochemical Devices: A Review," *Electroanalysis*, vol. 27, no. 3, pp. 562–572, 2015.
[3] J. T. Smith *et al.*, "Flexible ISFET biosensor using IGZO metal oxide TFTs and an ITO sensing layer," *IEEE Sens. J.*, vol. 14, no. 4, pp. 937–938, 2014.
[4] S. Nakata *et al.*, "A wearable pH sensor with high sensitivity based on a flexible charge-coupled device," *Nat. Electron.*, vol. 1, no. 11, pp. 596–603, 2018.
[5] N. Karaki *et al.*, "A flexible 8b asynchronous microprocessor based on low-temperature poly-silicon TFT technology," *ISSCC. 2005 IEEE Int. Dig. Tech. Pap. Solid-State Circuits Conf. 2005.*, pp. 272–274, 2005.
[6] K. Myny *et al.*, "An 8-bit, 40-instructions-per-second organic microprocessor on plastic foil," *IEEE J. Solid-State Circuits*, vol. 47, no. 1, pp. 284–291, 2012.
[7] J. Happich, "Starting all Over Again on Plastic: ARM," *EE Times*, 2015.
[8] PragmatIC, "FlexIC Production." [Online]. Available: https://www.pragmatic.tech/technology
[9] E. Ozer *et al.*, "Bespoke Machine Learning Processor Development Framework on Flexible Substrates," in *Int. Conf. Flex. Printable Sensors Syst.*, Glasgow, 2019, pp. 1–3.
[10] G. Indiveri *et al.*, "Neuromorphic silicon neuron circuits," *Front. Neurosci.*, vol. 5, no. MAY, pp. 1–23, 2011.
[11] W. Bae *et al.*, "A crossbar resistance switching memory readout scheme with sneak current cancellation based on a two-port current-mode sensing," *Nanotechnology*, vol. 27, no. 48, pp. 1–12, 2016.
[12] C. Yakopcic *et al.*, "Extremely parallel memristor crossbar architecture for convolutional neural network implementation," *Proc. Int. Jt. Conf. Neural Networks*, vol. 2017-May, pp. 1696–1703, 2017.
[13] H. Tsai *et al.*, "Recent progress in analog memory-based accelerators for deep learning," *J. Phys. D. Appl. Phys.*, vol. 51, no. 28, 2018.
[14] M. Douthwaite *et al.*, "A Thermally Powered ISFET Array for On-Body pH Measurement," *IEEE Trans. Biomed. Circuits Syst.*, vol. PP, no. 99, pp. 1–11, 2017.
[15] A. Shafiee *et al.*, "ISAAC : A Convolutional Neural Network Accelerator with In- Situ Analog Arithmetic in Crossbars," in *2016 ACM/IEEE 43rd Annu. Int. Symp. Comput. Archit.*, Seoul, 2016, pp. 14–26.