Low Power Neuromorphic Analog System based on Sub-Threshold Current Mode Circuits

Sarthak Gupta, Pratik Kumar, Kundan Kumar, Satrajit Chakraborty, Chetan Singh Thakur
Indian Institute of Science, Bengaluru, India
Email: {sarthakgupta, pratikkumar, kundankumar, satrajitc, csthakur}@iisc.ac.in

***Abstract*—Hardware implementation of brain-inspired algorithms such as reservoir computing, neural population coding and deep learning (DL) networks is useful for edge computing devices. The need for hardware implementation of neural network algorithms arises from the high resource utilization in form of processing and power requirements, making them difficult to integrate with edge devices. In this paper, we propose a non-spiking four quadrant current mode neuron model that has a generalized design to be used for population coding, echo-state networks (uses reservoir network), and DL networks. The model is implemented in analog domain with transistors in sub-threshold region for low power consumption and simulated using 180nm technology. The proposed neuron model is configurable and versatile in terms of non-linearity, which empowers the design of a system with different neurons having different activation functions. The neuron model is more robust in case of population coding and echo-state networks (ESNs) as we use random device mismatches to our advantage. The proposed model is current input and current output, hence, easily cascaded together to implement deep layers. The system was tested using the classic XOR gate classification problem, exercising 10 hidden neurons with population coding architecture. Further, derived activation functions of the proposed neuron model have been used to build a dynamical system, input controlled oscillator, using ESNs.**

Keywords—Neuromorphic Engineering, Hardware Accelerators, Cognitive Systems, Sub-threshold Analog VLSI

I. INTRODUCTION

Neural network algorithms are useful for tasks such as audio-visual classification [1]-[3] and learning dynamic control [4]. Hardware implementation of such learning algorithms can improvise performance in the field of robotics and edge devices. Sub-threshold analog design of such systems leads to efficient power and area characteristics compared to digital design making them suitable for larger architectures and power-crunch areas like edge devices. Neural population coding is inspired from various cortical regions [5]-[10]. By considering the response from an ensemble of neurons [11], classification and regression tasks can be performed. In echo-state networks (ESNs), a reservoir of neurons is used to process temporal data [12]. Moreover, architectures like population coding and ESNs uses random and fixed weights in initial layers, which reduces the amount of memory required to store these weights [13]-[15], hence, making them more hardware friendly. Several deep learning architectures have also evolved over a period of time [16], [17] and different variations of these have been proposed to make them more reliable and efficient [18]–[22]. To cater to these evolving architectures, we propose a hardware model of the neuron, which can be generalized and adapted to variations in architectures. Various works on neuron models [23]–[27] exist. Also, there are existing works which utilize random device matches for random and fixed weights in population coding [13], [14]. Our design is a four quadrant current mode, which can be cascaded together for deep learning architectures. The activation function of the proposed neuron model approximates the '*tanh*' curve, and can be controlled. This imparts flexibility to our proposed model which helps the architecture to learn better, especially in the case of population coding and ESNs, where randomness arising from only device mismatches may not be enough.

II. CURRENT MODE NEURON MODEL

Figure 1(a) shows the design of a two input neuron sub-system model. This sub-system is divided into three parts: weighted input block (analogous to dendrites in biological neuron), summation block (analogous to soma in biological

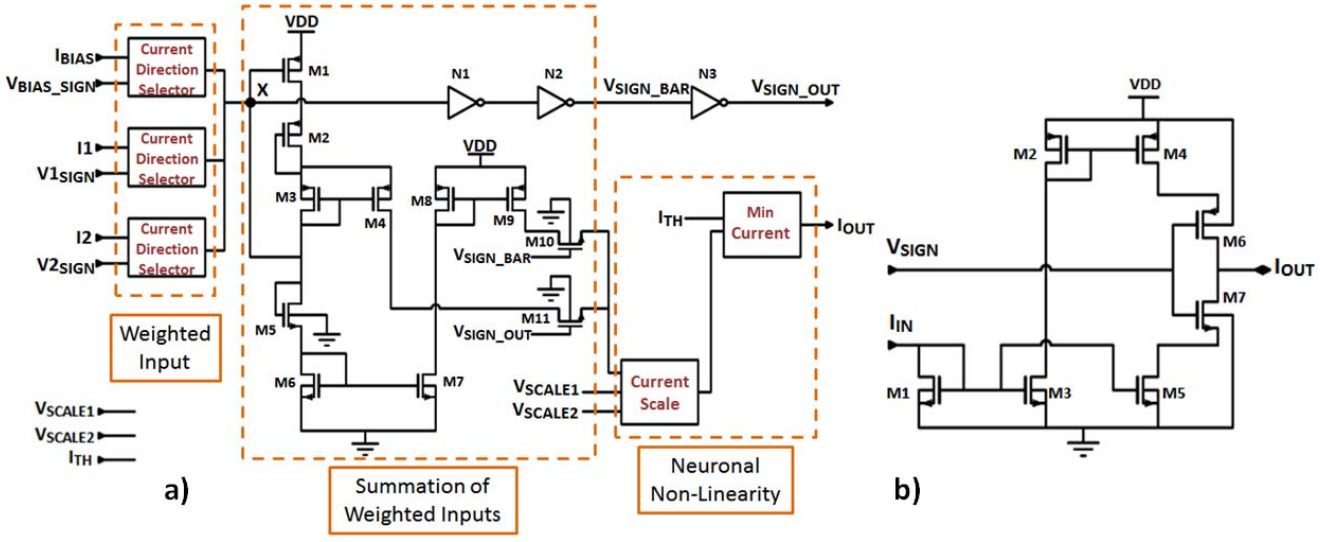

Figure 1. Schematic of proposed two input neuron model. (a) Four quadrant current mode design of two input neuron sub-system model. W/L $_{(M1\text{-}11)}$ = (0.42um)/(0.18um). (b) Current Direction Selector block – based on input sign voltage (V_{SIGN}), input current (I_{IN}) is sourced or sank at output (I_{OUT}). W/L $_{(M1\text{-}4,\ M6\text{-}7)}$ = (0.42um)/(0.18um), W/L $_{(M5)}$ = (0.84um)/(0.18um).

neuron) and neuronal non-linearity block (analogous to synapse in biological neuron). The input to the sub-system includes neuron inputs (I1, $V1_{SIGN}$, I2, $V2_{SIGN}$), and parameters: bias (I_{BIAS}; V_{BIAS_SIGN}), scale voltages (V_{SCALE1}; V_{SCALE2}) and threshold current (I_{TH}). The outputs of the sub-system are current, I_{OUT}, and voltage, V_{SIGN_OUT}, which represent the magnitude and sign of the output, respectively.

A. *Weighted Input Block*

Each input of the neuron whether bias or neural inputs, is given through the 'Current Direction Selector' (CDS) block as shown in Figure 1. Each input has one current input and one voltage input, which represent the magnitude and sign of the input, respectively. The input current direction is sink into the neuron sub-system model. In Figure 1(a), I1 and I2 currents correspond to the magnitudes of two given inputs, and $V1_{SIGN}$ and $V2_{SIGN}$ voltages correspond to the signs of the two given inputs. I_{BIAS} and V_{BIAS_SIGN} correspond to the magnitude and sign of bias, respectively, for the neuron model. Each of these inputs is fed into the respective CDS block. Figure 1(b) shows the schematic of the CDS block, where the direction of input current, I_{IN}, is set to source or sink based on the input sign voltage, V_{SIGN}. Current mirrors using transistors M1-M5 are used to replicate I_{IN}. Transistor M6 and M7 are used to set the direction of output current. If voltage V_{SIGN} is high, the direction of output current (I_{OUT}) is sink w.r.t. to the CDS block, and if voltage V_{SIGN} is low, the direction of output current (I_{OUT}) is source w.r.t. to the CDS block. The body terminal of M6 is connected to VDD, to increase the threshold voltage of M6 and hence compels M6 to remain in the sub-threshold region. Due to device mismatches, the magnitudes of I_{IN} and I_{OUT} will be different, and this will be useful in population coding and ESNs as these require random and fixed scaling of inputs.

B. *Summation Block*

The output currents from multiple CDS blocks (whether source or sink) are added at node X, as shown in Figure 1(a). The net current after summation at node X can be either source or sink. If the net current is source, the parasitic capacitance at X will get charged to VDD and transistor M1 will be in cut-off region. Hence, net source current will flow through M5 and M6. Further, voltage V_{SIGN_OUT} will become low or zero. If net current is sink, then first parasitic capacitance at X will discharge and transistor M1 will turn on. This will enable current to flow from supply through M1-M3. Diode connected M2 and M5 provides high output impedance to provide additional voltage drop to keep transistors M3 and M6 in the sub-threshold region. When net current is source, transistors M6-M9 are used to replicate and switch direction of current. In contrast, when net current is sink transistors M3 and M4 are used to replicate the net current. Transistors M10 and M11 are used as switches to select either sink or source current based on the voltage developed at point X. Multiple cascaded inverters N1, N2 and N3 are used to improve rise-time, fall-time and fan-out. Net source current of transistors M10 and M11 is passed to the current scale block.

C. *Neuronal Non-Linearity Block*

In neuronal non-linearity block as shown in Figure 1(a), the current scale block and minimum (min) current block are cascaded together. A schematic of the current scale block is shown in Figure 2(a). Here, current mirror (using M3 and M4) is used to replicate input current, I_{IN}, as bias current, I_B, for the differential pair M1 and M2. I1 and I2 are currents in the respective branches of M1 and M2. M5 and M6 are active loads. M6 and M7 are used to replicate current I2 as output current I_{OUT}. Differential voltages V_{SCALE1} and V_{SCALE2} are used to control the scaling factor for input current I_{IN}. In our design, V_{SCALE2} is fixed to a particular voltage and, V_{SCALE1} is varied for different neuron sub-systems. The current-voltage transfer characteristic of the differential pair is described as follows [28]:

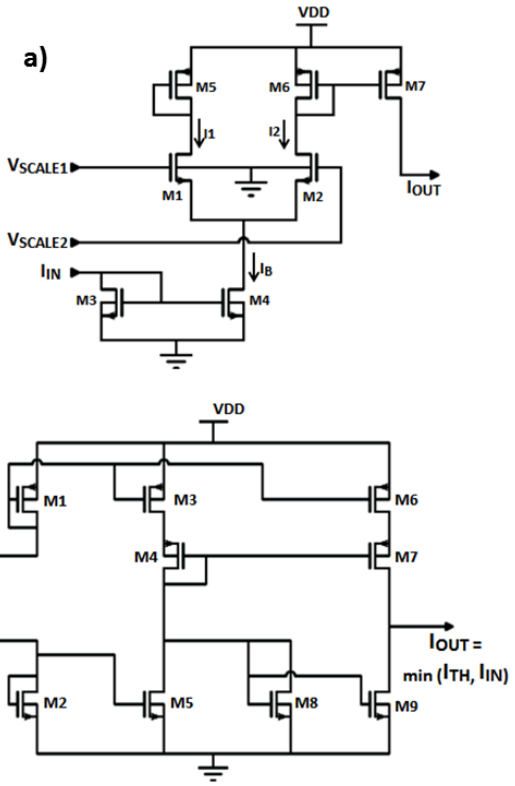

Figure 2. Schematic of sub-modules in neuronal non-linearity block. (a) Current scale block schematic. $(W/L)_{M1\text{-}7}$ = (0.42um)/(0.18um). (b) Min (Minimum) current block schematic. $(W/L)_{M1\text{-}9}$ = (0.42um)/(0.18um)

$$I1 = I_B \frac{\exp(\frac{V_{SCALE1}}{\eta U_T})}{\exp(\frac{V_{SCALE1}}{\eta U_T}) + \exp(\frac{V_{SCALE2}}{\eta U_T})} \quad (1)$$

$$I2 = I_B \frac{\exp(\frac{V_{SCALE2}}{\eta U_T})}{\exp(\frac{V_{SCALE1}}{\eta U_T}) + \exp(\frac{V_{SCALE2}}{\eta U_T})} \quad (2)$$

Here, U_T is the thermal voltage and in the weak-inversion region, the slope factor η of the transistors ranges from 1.1 to 1.5 [29]. Hence, the current I_{OUT} is proportional to input current I_{IN} and the proportionality constant (or scaling factor) can be varied by varying V_{SCALE1}. Current I_{OUT} of the current scale block is connected to I_{IN} of the min current block, as shown in Figure 2(b). If the current I_{IN} exceeds the threshold current I_{TH}, the output current I_{OUT} of the block will get fixed to I_{TH} [30]. To remove the offset current at output due to mismatches at M3 and M6, transistors M4 and

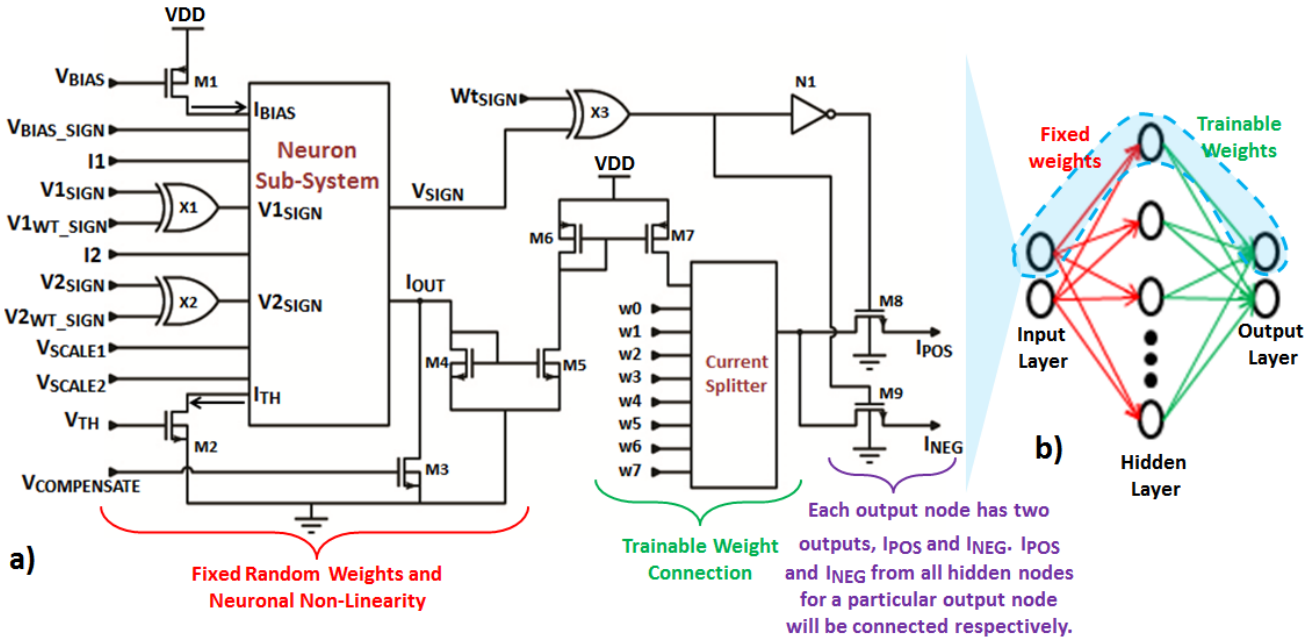

Figure 3. System-level implementation of population coding architecture. (a) System design for the highlighted portion of the population coding architecture, using proposed neuron model. $(W/L)_{M1\text{-}9}$ = (0.42um)/(0.18um). (b) Population Coding Architecture. Fixed weights are random, and trainable weights in our design are learned offline using batch gradient descent.

M7 are added for cascode current mirror. The output current of this block is the magnitude of the neuron output.

D. Complete System Design

Figure 3(a) represents the system design of single neuron from input to output in population coding architecture, using the proposed neuron model. Figure 3(b) represents the population coding architecture. An XOR gate is used to calculate the output sign of multiplication of two numbers. $V1_{SIGN}$ and $V2_{SIGN}$ are sign voltages for the corresponding two inputs. $V1_{WT\_SIGN}$ and $V2_{WT\_SIGN}$ are sign voltages for weights of the connection between input and neuron. XOR gates X1 and X2 are used to calculate the net sign of the weighted input. Three parameters control the non-linearity of the neuron: bias, threshold current of min current block, and scaling factor of current scale block. The scaling factor can be varied by varying voltage, V_{SCALE1}. Providing different voltages [15] to different neuron blocks is much easier than providing different currents. Hence, to control the magnitude of bias current and threshold current, M1 and M2 transistors are used. Voltage, $V_{COMPENSATE}$ is used to further decrease the offset current of min current block output (in our design it is less than 1nA) using M3 and is same for all neurons and controlled externally. Current mirrors using M4-M7 are used to replicate the output current of neuron model as sink current input to current splitter (CS) block [31], [32]. The output of CS block is either passed to I_{POS} or I_{NEG} branch using M8 and M9 as switches. This current selection is done by considering the sign voltage (V_{SIGN}) of neuron output and the sign of weight for the corresponding connection to output node using XOR, X3 and inverter, N1. Each output node is represented by two current outputs, I_{POS} and I_{NEG}. For multiple outputs, CS blocks corresponding to each output node will be present for every hidden neuron output. Further, inputs to these CS blocks will be provided by extending the current mirror M6 and M7. Moreover, I_{POS} and I_{NEG} branches of the respective output nodes are connected together. Voltages V_{BIAS}, V_{BIAS_SIGN}, V_{SCALE1}, and V_{TH} are varied externally for all neurons, using resistive polyline as potentiometer approach stated in existing work, [13]-[15].

III. RESULTS

For designed two input neuron model, I2 is made zero and I1 is swapped from -75nA to 75nA. Additionally, parameters like V_{BIAS}, V_{BIAS_SIGN}, V_{SCALE1}, and V_{TH} are varied in 882 combinations. For such combinations, the output is observed at I_{POS} and I_{NEG} using V_{SIGN}. During measurements, all weights of the CS block are set to high. Wt_{SIGN} is set low, so that the output of X3 is the same as V_{SIGN}. The net output current is plotted in Figure 4. It can be inferred that slope, and vertical and horizontal shifts of activation function can be controlled by varying the above parameters.

We tested our model with full system design for classic XOR classification using population coding architecture

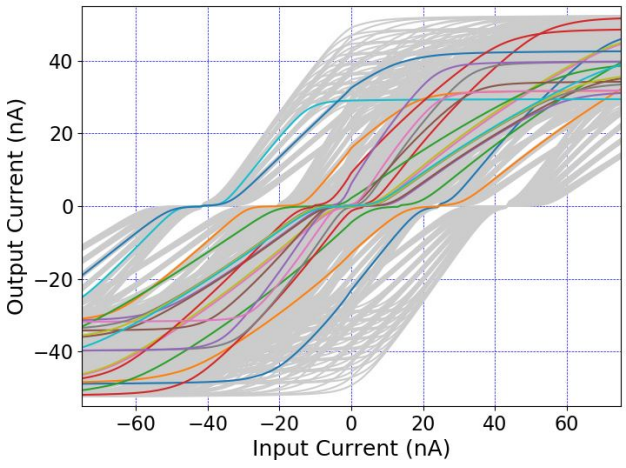

Figure 4. Different activation functions of various single input Neuron Model.For better visual depiction only 20 out of 882 neuron model curves are highlighted in color.

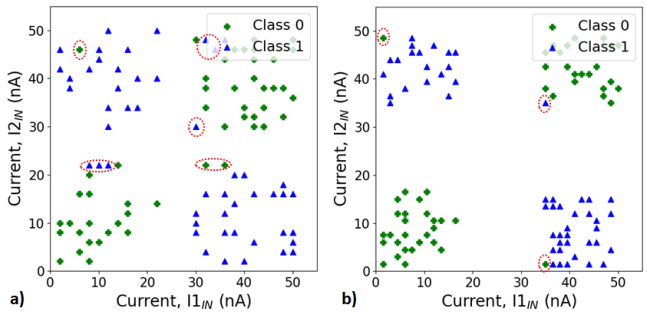

Figure 5. Classic XOR classification. (a) With margin of 8nA (per neuron) for adjacent classes, accuracy = 91.157%. (b) With margin of 18.5nA (per neuron) for adjacent classes, accuracy = 95.454%.

with 10 hidden neurons, two input nodes and one output node. The activation functions of all hidden neurons are configured using above mentioned parameters. Optimized weights are calculated offline, and these weights are then quantized into 8-bit levels for the CS block. We used two different sets of synthetic data, based on margin between the two adjacent classes, as shown in Figure 5(a) and (b). Misclassified points are encircled in red. Currents $I1_{IN}$ and $I2_{IN}$ are the input currents of each hidden neuron.

Further, we implemented ESN architecture for an input-controlled oscillator with y1 and y2 as state variables and μ as the user input to control the frequency of oscillations as mentioned in the following equation:

$$\frac{dy1}{dt} = \mu * y2; \frac{dy2}{dt} = -\mu * y1 \quad (3)$$

The reservoir contains 1000 neurons with 20% sparse connectivity. The activation functions of these neurons are selected from set 100 different activation functions, shown in Figure 4. The architecture of ESN is shown in Figure 6(a). State variables y1 and y2 are the system output and input node corresponds to user input, μ. While training, the golden output (or desired output) is fed back with unit delay to the reservoir. The network is trained for 3200 seconds. Read-out weights are optimized using batch gradient decent. Optimized weights are quantized into 12-bits. In testing phase the system output, instead of the golden output, is fed-back to the reservoir with unit delay. Figure 6(b) shows the input control frequency oscillator during the testing phase. The layout design of the proposed two input neuron model using 180nm technology node is shown in Figure 7.

IV. CONCLUSION

The proposed model generates configurable and versatile activation functions. This is achieved by use of minimum

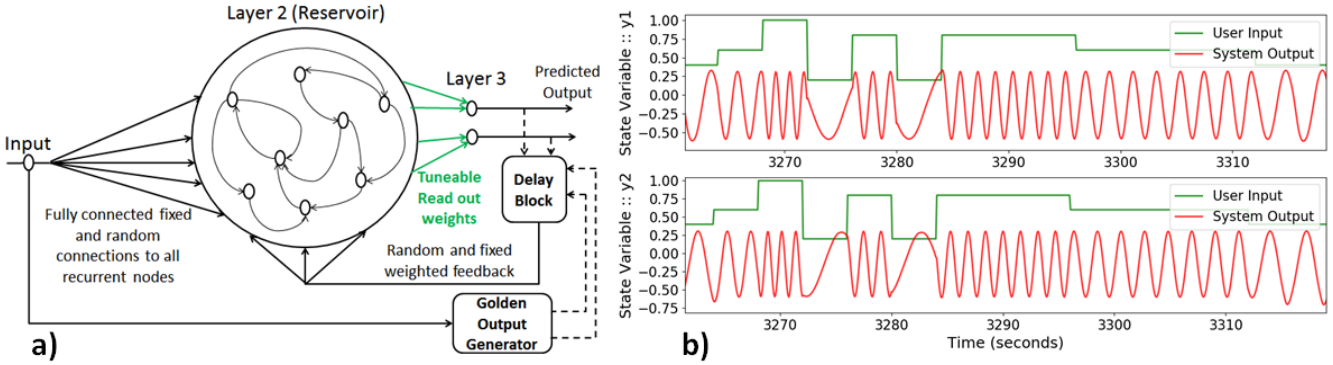

Figure 6. Echo-State Network (ESN) Implementation. (a) Architecture of echo-state network. (b) Voltage controlled frequency oscillator using activation functions of the proposed neuron model.

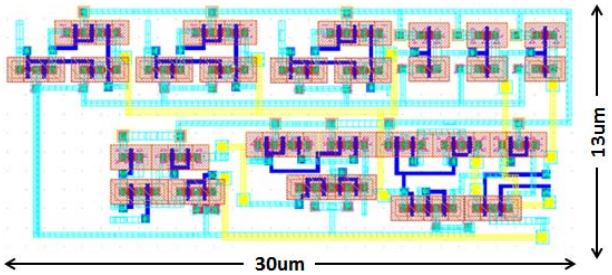

Figure 7. Layout of two input neuron sub-system model.

current block, current scale block and bias input, compared to standard differential pair for implementing hyperbolic tangent as activation function. By providing different parameters V_{BIAS}, V_{BIAS_SIGN}, V_{SCALE1}, and V_{TH} for different neurons, these neurons become more versatile in terms of non-linearity in activation function. Compared to existing works, [13]-[15], proposed model is current input and current output, giving it advantage to be extended to ESN or cascade into multiple layers for DL architectures. In case of DL architectures or in current splitter circuitry where random device mismatches need to be minimum, techniques like using longer channel lengths or cascode current mirror can be used. Moreover, the proposed neuron sub-system model can easily adapt to multiple architectures and variations in architectures, as well. As proof of concept for multiple inputs, we have demonstrated the system using a two input model. Specifications for the two input neuron sub-system model are tabulated in Table I. For ESNs, the output current at time $(t-1)^{th}$ instant needs to be stored for the time t^{th} instant. This can be achieved with current to voltage converter and sample and hold (S&H) block. Here, the voltage corresponding to current at $(t-1)^{th}$ instant can be stored using the S&H block, and corresponding proportion of current can be generated at time t^{th} instant. Because feedback weights in ESNs are random and fixed, the proportionality factor can be used to our advantage. Hence, our proposed neuron model can be extended to implement population coding, ESNs, and DL networks.

TABLE I. PROPOSED NEURON (2- INPUT) SUB-SYSTEM MODEL SPECIFICATIONS.

Type	Tech. Node	Domain	Area
Programmable	180nm	Analog (Sub-threshold)	390 um^2

ACKNOWLEDGMENT

Research facilities for this work were supported by the Pratiksha trust grant PratikshaYI/2017-8512.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. Hinton, " ImageNet classification with deep convolutional neural networks," In Proc. Advances in Neural Information Processing Systems 25, 1090–1098 (2012).

[2] G. Hinton, et al., "Deep neural networks for acoustic modeling in speech recognition," IEEE Signal Processing Magazine 29, 82–97 (2012).

[3] C. S. Thakur, R. M. Wang, S. Afshar, T. J. Hamilton, et al., "Sound stream segregation: a neuromorphic approach to solve the 'cocktail party problem' in real-time," Front. Neurosci., vol. 9, pp. 1– 10, 2015.

[4] K. S. Narendra, and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," IEEE Trans. Neural Networks, vol 1, no. 1, pp. 4-27, 1990.

[5] A. P. Georgopoulos, and A. F. Carpenter, "Coding of movements in the motor cortex," Curr. Opin. Neurobiol, 33 (2015) 3439.

[6] D. Sparks, "Population coding of saccadic eye movements by neurons in the superior colliculus," Nature 332 (1988) 357360.

[7] R. Quian Quiroga, and S. Panzeri, "Extracting information from neuronal populations: information theory and decoding approaches," Nat. Rev. Neurosci. 10, 173185 (2009).

[8] A. Pouget, P. Dayan, and Z. Zemel, "Information processing with population codes," Nat. Rev. Neurosci. 1: 125-132 (2000).

[9] K. Kang, R. M. Shapley, and H. Sompolinsky, "Information tuning of populations of neurons in primary visual cortex," J. Neurosci. 24 (15) (2004) 37263735.

[10] A. Pasupathy, and C. E. Connor, "Population coding of shape in area V4," Nat. Neurosci. 5 (12) (2002) 13321338.

[11] B. B. Averbeck, P. E. Latham, and A. Pouget, "Neural correlations, population coding and computation," Nat. Rev. Neurosci. 7:358366 (2006).

[12] H. Jaeger, "The echo state approach to analysing and training recurrent neural networks," Technical report GMD Report 148. German National Research Center for Information Technology, 2001

[13] C. S. Thakur, R. Wang, T. J. Hamilton, J. Tapson, and A. van Schaik, "Low power trainable neuromorphic integrated circuit that is tolerant to device mismatch," IEEE Trans. Circuits Syst. I 63, 211221 (2016).

[14] C.S. Thakur, R. Wang, T. J. Hamilton, R. Etienne-Cummings, J. Tapson, and A. van Schaik, "An analogue neuromorphic co-processor that utilizes device mismatch for learning applications," IEEE Trans. Circuits Syst. I, 65, 111 (2017).

[15] C. S. Thakur, T. J. Hamilton, R. Wang, J. Tapson, and A. van Schaik, "A neuromorphic hardware framework based on population coding," International Joint Conference on Neural Networks (IJCNN) pp. 18 (2015).

[16] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature 521, 436444 (2015).

[17] J. Schmidhuber, "Deep learning in neural networks: An overview. Neural Networks," 61:85117, 2015. doi: 10.1016/j.neunet.2014.09.003. Published online 2014; based on TR arXiv:1404.7828.

[18] M. Courbariaux, Y. Bengio, and J.P. David, "BinaryConnect: Training Deep Neural Networks with binary weights during propagations," In Advances in Neural Information Processing Systems. 31053113 (2015).

[19] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," arXiv preprint arXiv:1503.02531 (2015).

[20] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," J. Machine Learning Res. 15, 19291958 (2014).

[21] C. S. Thakur, R. Wang, S. Afshar, T. J. Hamilton, J. Tapson, and A. van Schaik, "An online learning algorithm for neuromorphic hardware implementation," arXiv:1505.02495, p. 8, May 2015.

[22] A. Nokland, "Direct feedback alignment provides learning in deep neural networks", NIPS, 2016.

[23] C. Mead, "Analog VLSI and Neural Systems," Addison-Wesley, 1989.

[24] M. Mahowald, and R. Douglas, "A silicon neuron," Nature 354, 515518 (1991).

[25] J. V. Arthur, and K. Boahen, "Silicon-neuron design: A dynamical systems approach," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 58, no. 5, pp. 10341043, May 2011.

[26] G. Indiveri, B. Linares-Barranco, T. J. Hamilton, A. van Schaik, et al., "Neuromorphic silicon neuron circuits," Front. Neurosci., vol. 5, 2011, DOI: 10.3389/fnins.2011.00073.

[27] C. S. Thakur, J. L. Molin, G. Cauwenberghs, G. Indiveri, K. Kumar, et al., "Large-Scale Neuromorphic Spiking Array Processors: A Quest to Mimic the Brain," Front. Neurosci., vol 12, Dec. 2018.

[28] A. Wang, B. H. Calhoun, and A. P. Chandrakasan, "Analog circuits in weak inversion, maximum transconductance-to-current ratio, differential pair, Sub-threshold design for ultra low-power systems," ISBN:0-387-34501-9, Springer.

[29] E. A. Vittoz, "Weak inversion for ultra low-power and very low-voltage circuits," In Proc. IEEE Asian Solid-State Circuits Conf., 2009, pp. 129132.

[30] E. A. Vittoz, "Analog VLSI signal processing: Why, where, and how?," J. VLSI Signal Process. 8(1):2744 (1994).

[31] G. Bult, and G. Geelen, "An inherently linear and compact MOST-only current division technique," IEEE Journal of Solid-State Circuits, vol. 27 pp. 1730-1735, 1992.

[32] T. Delbruck, and A. van Schaik, "Bias current generators with wide dynamic range," Analog Integr. Circuits Signal Process., vol. 43, no. 3, pp. 247268, Jun. 2005.