MRAM-based Analog Sigmoid Function for In-memory Computing

Md Hasibul Amin, Mohammed Elbtity, Mohammadreza Mohammadi, Ramtin Zand
Department of Computer Science and Engineering, University of South Carolina, Columbia, SC, 29201, USA

ABSTRACT
We propose an analog implementation of the transcendental activation function leveraging two spin-orbit torque magnetoresistive random-access memory (SOT-MRAM) devices and a CMOS inverter. The proposed analog neuron circuit consumes $1.8 \sim 27 \times$ less power, and occupies $2.5 \sim 4931 \times$ smaller area, compared to the state-of-the-art analog and digital implementations. Moreover, the developed neuron can be readily integrated with memristive crossbars without requiring any intermediate signal conversion units. The architecture-level analyses show that a fully-analog in-memory computing (IMC) circuit that use our SOT-MRAM neuron along with an SOT-MRAM based crossbar can achieve more than $1.1 \times$, $12 \times$, and $13.3 \times$ reduction in power, latency, and energy, respectively, compared to a mixed-signal implementation with analog memristive crossbars and digital neurons. Finally, through cross-layer analyses, we provide a guide on how varying the device-level parameters in our neuron can affect the accuracy of multilayer perceptron (MLP) for MNIST classification.

CCS CONCEPTS
- Hardware → Spintronics and magnetic technologies. Analog and mixed-signal circuits; - Computing methodologies → Machine learning.

KEYWORDS
SOT-MRAM Neuron, Multilayer Perceptron (MLP), Analog Neuron, In-Memory Computing

ACM Reference Format:
Md Hasibul Amin, Mohammed Elbtity, Mohammadreza Mohammadi, Ramtin Zand. 2022. MRAM-based Analog Sigmoid Function for In-memory Computing. In Proceedings of the Great Lakes Symposium on VLSI 2022 (GLSVLSI ’22), June 6–8, 2022, Irvine, CA, USA. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3526241.3530376

1 INTRODUCTION
In-memory computing (IMC) systems, as an alternative for von Neuman architectures, aim to eliminate the expensive data movement between processor and memory in data-intensive applications such as machine learning (ML) by implementing computation where the data exist. IMC architectures leverage massive parallelism in memristive crossbars and analog computation to aggressively reduce the computational time complexity of matrix-vector multiplication (MVM) operations in ML workloads. However, there are still a wide range of computational tasks in ML workloads that cannot be performed in crossbars, e.g., activation functions and subsampling layers. The solution offered in most of the existing IMC architectures such as ISAAC [1] and PRIME [2], is to design digital circuits for some of these functions, and place them near crossbars. Although this approach reduces the off-chip data transfer overhead between processor and memory for commonly-used functions, it requires analog-to-digital converter (ADC) and digital-to-analog converter (DAC) units to transfer the data between crossbars and digital functional units. According to a research by HP Labs [3], the signal converters and peripheral circuitry in IMC circuits can contribute to up to 90% and 95% of the total power consumption and area occupation, respectively. Therefore, reducing the need for signal conversion units in an IMC architecture can potentially lead to significant energy improvements.

Transcendental activation functions such as sigmoid and hyperbolic tangent (tanh) provide the desired nonlinear behavior which can be useful for a variety of ML models such as multilayer perceptron (MLP) and Long Short-Term Memory (LSTM). However, due to their hardware implementation challenges, alternative nonlinear activation functions such as rectified linear unit (ReLU), first proposed in [4], have become popular in modern deep learning models such as convolutional neural networks (CNNs) [5, 6]. Hardware implementation of sigmoid and tanh functions have been widely investigated in the digital domain. Direct implementations based on high-order polynomials, Taylor’s series, and Maclaurin series offer high precision at the expense of high power consumption [7]. On the other hand, CORDIC-based hardware implementation [8] can achieve high accuracy and reduced area, however they have a high latency. In the PWL technique the non-linear function is separated into several linear segments to reduce implementation complexity [9]. Higher precision can be accomplished by increasing the number of linear segments at the cost of more hardware resource utilization. Among the several function approximation approaches, LUT methods are the fastest [10], but it requires large amount of memory to provide high accuracy.

Although digital hardware can provide an exact implementation of the ideal sigmoid function and is more robust to noise, its high power, area, and latency overheads have motivated the research on analog transcendental activation functions. In [11], three NMOS and three PMOS transistors are carefully sized to approximate a sigmoid activation function with only one reference voltage to bias the transistors. In [12], an NMOS and a PMOS transistors with large width/length ratios are utilized along with a resistor load to realize tanh-like activation function. While, both of these
design achieve significant reductions in power and area compared to digital implementations, they still require large transistors. In this paper, we leverage spin-orbit torque magnetoresistive random-access memory (SOT-MRAM) devices and a simple CMOS inverter to develop a sigmoid activation function that is compatible with memristive crossbars. We exhibit the direct and indirect energy and performance benefits of our proposed analog neuron for the IMC architectures through the circuit implementation of a fully-analog MLP circuit and comparing it with existing mixed-signal designs.

## 2 SOT-MRAM TECHNOLOGY

MRAM device is one of the promising memristive technologies that have been widely investigate for the IMC circuits and architectures [13, 14]. The magnetic tunnel junctions (MTJs) are the main building block in MRAM devices. MTJs include two ferro-magnetic (FM) layers that are separated by a thin oxide layer as shown in Figure 1. The magnetization direction of electrons in one of the FM layers (reference layer) is fixed, whereas the direction of electrons in the other FM layer (free layer) can be switched. The resistance levels of MTJ are determined by the angle ($\theta$) between the magnetization orientation of the FM layers. The following equations can be used to calculate the resistance of the MTJ in parallel (P) and antiparallel (AP) magnetization configurations [15, 16]:

$$ R(\theta) = \frac{2R_{MTJ}(1 + TMR)}{2 + TMR(1 + \cos \theta)} = \begin{cases} R_P = R_{MTJ}, & \theta = 0 \\ R_A = R_{MTJ}(1 + TMR), & \theta = \pi \end{cases} \quad (1) $$

$$ TMR(T, V_b) = \frac{TMR_0}{100} \left( \frac{V_b}{V_0} \right)^2 \quad (2) $$

where $R_{MTJ} = \frac{RA}{Area}$. The MTJ’s resistance-area product (RA) value is determined by the material composition of its layers. TMR stands for tunneling magnetoresistance, and it is determined by temperature (T) and bias voltage ($V_b$). $TMR_0$ is a material-dependent constant, and $V_0$ is a fitting parameter.

SOT-MRAM device [17] is a class of MRAM technology, in which the magnetization direction of the free layer can be changed by means of a spin-polarized current produced by a charge current passing through a heavy metal (HM), as shown in Fig. 1. The resistance of the HM can be calculated using the below equation:

$$ R_{HM} = \rho_{HM}I_{HM}/w_{HM}l_{HM} \quad (3) $$

where $\rho_{HM}, I_{HM}, w_{HM}, l_{HM}$ are the resistivity, length, width, and thickness of the HM, respectively. Here, we use the aforementioned MTJ and HM equations to create a Verilog-A model of the SOT-MRAM device for circuit simulations.

## 3 MRAM-BASED SIGMOID NEURON

Our proposed SOT-MRAM-based sigmoid neuron incorporates two SOT-MRAM devices (MRAM1 and MRAM2) and a CMOS-based inverter, as shown in Figure 2. The SOT-MRAM devices are configured in opposite states where MRAM1 is in parallel state ($R_P$) and MRAM2 is in anti-parallel state ($R_A$). The SOT-MRAM devices form a voltage divider which reduces the slope of the inverter’s linear region leading to a smooth transition from VDD to VSS in the output node (OUT), when the input voltage node (IN) is swept from VSS to VDD. Herein, to verify that such circuit can realize a transcendental activation function between the input voltage ($V_{IN}$) and output voltage ($V_{OUT}$), we conducted a detailed circuit analysis as described in the followings.

First, we apply the Kirchhoff’s current law at OUT node resulting in the following relation:

$$ I_{IN} = I_{DSP} + I_{DSN} \quad (4) $$

where $I_{IN}$ is the current from node $IN$ to node $OUT$ which passes through the MRAM devices (Fig. 2 (a)), and $I_{DSP}$ and $I_{DSN}$ are the drain-to-source currents for the PFET (or PMOS) device $MP$ and NFET (or NMOS) device $MN$, respectively. Considering that no current flows towards the gates of the FinFETs, the $I_{IN}$ can be calculated as follows based on the Ohm’s law:

$$ I_{IN} = \frac{V_{IN} - V_{OUT}}{R_P + R_A} \quad (5) $$

where $R_P$ and $R_A$ can be obtained from equations (1) and (2). Replacing $I_{IN}$ in (4) with (5), we can calculate $V_{OUT}$ as:

$$ V_{OUT} = V_{IN} - [(R_P + R_A)(I_{DSP} + I_{DSN})] \quad (6) $$

The different regions of operation for our neuron can be identified based on the different operating regions of the $MN$ and $MP$ transistors. To find the operating region of the transistors, we need to calculate their gates’ voltage, i.e., $V_{INV}$ in Figure 2 (a). Since no current flows towards the gate of the transistors, $V_{INV}$ can be calculated using the below equation:

$$ V_{INV} = V_{IN} - V_T $$

where $V_T$ is the threshold voltage of the transistor.
Table 1: Drain-to-source currents for different operating regions of MP and MN.

| Cut-off | Triode | Saturation |
|---------|--------|------------|
| $I_{DS-P}$ | 0 | $\beta_P((V_{IN} - V_{DD} - V_{TP})(V_{OUT} - V_{DD}) - \frac{(V_{OUT} - V_{DD})^2}{2})$ |
| $I_{DS-N}$ | 0 | $\beta_N((V_{IN} - V_{SS} - V_{TN})(V_{OUT} - V_{SS}) - \frac{(V_{OUT} - V_{SS})^2}{2})$ |

Table 2: Conditions for different operating regions of MP and MN with $V_{TP}$ and $V_{TN}$ threshold voltages, respectively.

| Cut-off | Triode | Saturation |
|---------|--------|------------|
| MP | $V_{IN} > V_{DD} + V_{TP}$ | $V_{IN} < V_{DD} + V_{TP}$ | $V_{IN} < V_{DD} + V_{TP}$ |
| MN | $V_{IN} < V_{SS} + V_{TN}$ | $V_{IN} > V_{SS} + V_{TN}$ | $V_{IN} > V_{SS} + V_{TN}$ |

Figure 3: VTC of the proposed sigmoid neuron.

$$V_{IN} = V_{IN} = \frac{V_{IN} - V_{OUT}}{R_P + R_{AP}} \times R_P$$ (7)

By using equation (7) and the relations listed in Table 2, we can find the operating region of the transistors, and consequently the operating regions of the proposed neuron, based on the input voltage $V_{IN}$. Once the regions are identified, we can use equation (6) and the $I_{DS-P}$ and $I_{DS-N}$ relations listed in Table 1 to find the relation between $V_{OUT}$ and $V_{IN}$. Here, we utilized the MATLAB solver to obtain the aforementioned relations, based on which the neuron has five operating regions. Figure 3 shows the neuron’s voltage transfer characteristic (VTC) plotted based on the equations obtained from the solver for $V_{DD}=0.8V$ and $V_{SS}=0$ as the nominal voltages for the 14nm High-Performance PTM-MG FinFET model [18] with $V_{TP}=-0.2V$ and $V_{TN}=0.2V$ threshold voltages. As shown, the proposed neuron can provide a transcendental-like activation function that can approximate a $V_{OUT} = sigmoid(-V_{IN})$ function. An actual $V_{OUT} = sigmoid(V_{IN})$ function can also be provided by inverting the $V_{IN}$ before applying it to the proposed neuron. However, as described in the next section, it would not be necessary if the neural network is trained with the inverse sigmoid function.

4 RESULTS AND DISCUSSION

4.1 Device-level Analysis

To explore the impacts of device-level parameters such as TMR and RA on the shape of the sigmoid function, we implement the proposed neuron in SPICE circuit simulator using the Verilog-A model developed for the SOT-MRAM devices along with 14nm FinFET transistor model. Figure 4 shows how changing RA and TMR values in the SOT-MRAM can change the sigmoid function.

To analyze the effect of RA on our neuron, we vary the value of RA to 5, 10, 15, and 20 $\Omega \mu m^2$, while fixing the TMR to 200 (Fig. 4(a)). For analyzing the effect of TMR, we keep the value of RA fixed to 15 $\Omega \mu m^2$ and sweep the value of TMR to 100, 200, 300, and 400, and plot the results in Figure 4 (b). The results show that changing TMR imposes larger variations in the shape of the function. However, to investigate how the shape of the sigmoid function can impact the entire network’s accuracy, we trained a binarized 400×120×84×10 MLP model to classify MNIST [19] handwritten images with 20×20 pixels. We used $sigmoid(-\chi)$ activation function during the training to be compatible with the proposed neuron’s sigmoidal shape. First, the model is trained for 20 epochs achieving an accuracy of roughly 97%. Next, the trained weights are mapped to an SOT-MRAM based crossbar. Finally, the proposed SOT-MRAM neuron
Table 3: Comparison between the SOT-MRAM based neuron and previous digital and analog implementations.

| Domain    | [21] | [9] | [11] | [12] | This work |
|-----------|------|-----|------|------|-----------|
| Power (µW) | 6.72 × 10^7 | 493.4 | 74.21 | 32.16 | 18.04     |
| Area (µm²) | 9.3095 | 680.5 | 0.4975 | 0.3505 | 0.138     |

is integrated with the crossbar and the corresponding IMC circuit is simulated by SPICE. Here, we do not explain the crossbar circuit implementation since it has been already well-investigated in the literature [3, 20]. Figure 5 (a) shows how the changes made to the shape of the sigmoid activation function caused by varying the TMR and RA values can impact the overall accuracy of the implemented IMC circuit. The results show that higher RA values consistently lead to better accuracy. However for TMR, the accuracy increases when changing its value from 100 to 200 but it significantly drops when TMR value is increased to 300 and 400. Thus, a maximum accuracy of 95.83% is achieved with TMR=200 and RA=20 Ωµm², which is roughly 1% less than what was realized by ideal software implementation. The change in TMR and RA values can also affect the power consumption of the neuron as shown in Fig. 5 (b), according to which, higher RA and higher TMR values can lead to reduced power consumption.

4.2 Circuit-level Analysis

To compare our SOT-MRAM based sigmoid neuron with previous digital [21, 9] and analog [11, 12] implementation of transcendental neurons, we conduct SPICE circuit simulations using 14nm technology node [22] and verilog-A model of the SOT-MRAM. For the area comparison, we designed the layout of our neuron as shown in figure 2 (b), which demonstrates an area occupation of 0.138 µm². However, as the area occupation values of the previous neurons were measured based on older technology nodes, we use the scaling factors from [23] to scale-down their area to 14nm node to provide a fair comparison. For the power consumption comparison, we implemented the previous analog neurons [11, 12] and our SOT-MRAM neuron with 14nm technology node. For our neuron, we swept the input voltage from -2V to 2V and measure the average power consumption. However, the previous analog neurons [11, 12] take current as input unlike our proposed sigmoid neuron. For fair calculation of the average power consumption, we only consider the active range of their activation function, which we found to be from -200µA to 200µA. For the digital neurons, we scaled down their power consumption to 14nm technology using the scaling factors provided in [23]. The comparison results are listed in Table 3, according to which our neuron achieves 27× and 1.8× power reduction and 4931× and 2.5× area reduction compared to the most power- and area-efficient digital and analog neurons, respectively.

Table 4: Power consumption and latency breakdowns of the fully-analog and mixed-signal implementations of MLP.

| Layer | Power (mW) | Latency (Clocks) | Power (mW) | Latency (Clocks) |
|-------|------------|------------------|------------|------------------|
| Layer 1 | DAC | 3.463 | Crossbar | 221.965 |
| | Neuron | 59.208 | ADC | 0.727 |
| Layer 2 | DAC | 1.039 | Crossbar | 12.225 |
| | Neuron | 41.466 | ADC | 4.934 |
| Layer 3 | DAC | 0.42 | Crossbar | 0.42 |

Table 5: Total energy, power, and latency comparison between fully-analog and mixed-signal MLP implementations.

| Power(W) | Latency(ns) | Energy(nJ) | TOPS/W |
|----------|-------------|------------|--------|
| Mixed-Signal | 0.355 | 48 | 17.94 | 3.41 |
| Fully-Analog | 0.242 | 4 | 0.968 | 60.86 |

The fully-analog IMC architecture is a DAC unit in the first layer to convert the digital inputs from a CPU to analog signals, and one ADC unit to convert the analog output of the IMC to digital values to be transferred to the CPU. We compare our design with a conventional mixed-signal IMC architectures which use analog crossbars along with digital neurons, thus imposing the need for DAC and ADC units in every layer. The schematic of both designs for implementing the 400 × 120 × 84 × 10 MLP model is shown in Figure 6, in which the number of neurons, DACs, ADCs and the size of the crossbars in each layer is identified separately.

The fully-analog implementation saves power through the elimination of the intermediate conversion units, connecting all crossbars, and the use of power-efficient analog neurons, which are all achieved by using the SOT-MRAM based sigmoid neuron proposed herein. The total power, latency, energy, and performance in terms of tera operations per second per Watt (TOPS/W) of fully-analog and mixed-signal IMC architectures are listed in table 5. The results obtained show approximately 13× improvement in terms of the TOPS/W.
5 CONCLUSION

We proposed a power- and area-efficient SOT-MRAM based neuron circuit realizing an analog sigmoid activation function. Through a hierarchical simulation process, we show the characteristics and benefits of our proposed neuron across device, circuit, and architecture levels. Besides the direct performance and power benefits that can be achieved by the proposed neuron compared to the state-of-the-art, it is shown that it can be readily integrated with analog memristive crossbars without requiring any signal conversion that obviates the need for power-hungry DAC and ADC units. Simulation results exhibited that a fully-analog IC architecture using our proposed neuron achieves more than 13× improvement in TOPS/W compared to the mixed-signal IMC implementations that utilize analog crossbars along with digital neurons.

REFERENCES

[1] Ali Shafiee, Anirban Nag, Naveen Muralimanohar, Rajeev Balasubramonian, John Paul Strachan, Miao Hu, B. Stanley Williams, and Vivek Srivatsa. 2016. Isaac: a convolutional neural network accelerator with in-situ analog arithmetic in crossbars. In ISCAS ’16. IEEE Press, Seoul, Republic of Korea, 14–26.

[2] Ping Chi, Shuangchen Li, Cong Xu, Tao Zhang, Jishen Zhao, Yonggan Liu, Yu Wang, and Yuan Xie. 2016. Prime: a novel processing-in-memory architecture for neural network computation in reram-based main memory. ACM SIGARCH Computer Architecture News, 44, 3, 27–39.

[3] Miao Hu, John Paul Strachan, Zhao Li, Emmanuelle M. Grafals, Noraica Davila, Catherine Graves, Sity Lam, Ning Ge, Jianhua Joshua Yang, and R. Stanley Williams. 2016. Dot-product engine for neuromorphic computing: programming 1T1M crossbar to accelerate matrix-vector multiplication. In 2016 53nd ACM/EDAC/IEEE Design Automation Conference (DAC), 1–6.

[4] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. ImageNet classification with deep convolutional neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition. IEEE, 1106–1110.

[5] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[6] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition, 1–9.

[7] B. Lee and N. Burgess. 2003. Some results on taylor-series function approximation on fpga. In The Thirty-Seventh Asilomar Conference on Signals, Systems, Computers, 2003. Volume 2, 2198–2202 Vol.2.

[8] Hui Chen, Lin Jiang, Yuanyong Luo, Zhonghai Lu, Yuxiang Fu, Li Li, and Zongguang Yu. 2020. A cordic-based architecture with adjustable precision and flexible scalability to implement sigmoid and tanh functions. In 2020 IEEE International Symposium on Circuits and Systems (ISCAS). 1–5.

[9] Guido Baccelli, Dimitrios Stathis, Ahmed Hemani, and Maurizio Martina. 2020. Nacu: a non-linear arithmetic unit for neural networks. In 2020 57th ACM/IEEE Design Automation Conference (DAC), 1–6.

[10] Pramod Kumar Meher. 2010. An optimized lookup-table for the evaluation of sigmoid function for artificial neural networks. In 2010 16th IEEE/IFIP International Conference on Very Large Scale Integration (VLSI) and System-on-Chip, 91–95.

[11] G. Khodabandelhoo, M. Mirhassani, and M. Ahmadi. 2012. Analog implementation of a novel resistive-type sigmoidal neuron. IEEE Trans. Very Large Scale Integr. (VLSI) Syst., 20, 4, 750–754.

[12] J. Shamsi, A. Amirsoleimani, S. Mirakouchaki, A. Ahmada, S. Alirezaee, and M. Ahmadi. 2015. Hyperbolic tangent passive resistive-type neuron. In IEEE International Symposium on Circuits and Systems (ISCAS).

[13] Ramtin Zand, Kerem Yuzus Camsari, Steven D. Pyle, Ibrahim Ahmed, Chris H. Kim, and Ronald F. DeMarra. 2018. Low-energy deep belief networks using intrinsic sigmoidal spinnoronic-based probabilistic neurons. In Proceedings of the 2018 on Great Lakes Symposium on VLSI (GLSVLSI ’18). Chicago, IL, USA, 15–20.

[14] Mohammed Elbillah, Abhishek Singh, Brendan Reidy, Xiaoconen Guo, and Ramtin Zand. 2021. An in-memory analog computing co-processor for energy-efficient cnn inference on mobile devices. 2021 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), 2021.

[15] Y. Zhang, W. Zhao, Y. Lakys, J. O. Klein, J. V. Kim, D. Ravelosona, and C. Chappert. 2012. Compact modeling of perpendicular-anistotropy cofe/mgo magnetic tunnel junctions. IEEE Transactions on Electron Devices, 59, 3, 819–826.

[16] Ramtin Zand, Armanpooya, and Ronald F. DeMarra. 2018. Fundamentals, modeling, and application of magnetic tunnel junctions. Nanoscale Devices: Physics, Modeling, and Their Application, 317.

[17] L. Liu, C.F. Pai, Y. Li, H. W. Tseng, D. C. Ralph, and R. A. Buhrman. 2012. Spintorque switching with the giant spin hall effect of tantalum. Science, 336, 6081, 555–558.

[18] 2012. Predictive technology model. http://ptm.asu.edu. (2012).

[19] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. 1998. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86, 11, 2278–2324.

[20] Md Hadiul Amin, Mohammed Elbillah, and Ramtin Zand. 2022. Interconnect parasites and partitioning in fully-analog in-memory computing architectures. In 2022 IEEE International Symposium on Circuits and Systems (ISCAS).

[21] Gunjan Rajput, Gopal Raut, Mahesh Chandra, and Santosh Kumar Vishwakarma. 2021. Vlsi implementation of transcendental function hyperbolic tangent for deep neural network accelerators. Microprocessors and Microsystems, 84, 104270.

[22] Saurabh Sinha, Greg Yeric, Vikas Chandra, Brian Cline, and Yu Cao. 2012. Exploring sub-20nm findel design with predictive technology models. In DAC Design Automation Conference 2012, 283–288.

[23] Aaron Stillmaker and Bevan Baas. 2017. Scaling equations for the accurate prediction of CMOS device performance from 180 nm to 7 nm. Integration, 58, (June 2017), 74–91.

[24] Jevrani Kommarady. 2019. 10-bit c2c dac design in 65nm cmos technology. In Browse all Theses and Dissertations. 2112. https://corescholar.libraries.wright.edu/etd_all/2112.