Single-Poly Floating-Gate Memory Cell Options for Analog Neural Networks

Maksym Paliy[1*], Tommaso Rizzo[1], Piero Ruiu[1], Sebastiano Strangio[1], Giuseppe Iannaccone[1]

**E-mail address of corresponding author (*):* maksym.paliy@ing.unipi.it

[1]Dipartimento di Ingegneria dell'Informazione, Università di Pisa, Pisa, 56122, Italy

Abstract

In this paper, we explore the use of a 180 nm CMOS single-poly technology platform for realizing analog Deep Neural Network integrated circuits. The analysis focuses on analog vector-matrix multiplier architectures, one of the main building blocks of a neural network, implementing in-memory computation using Floating-Gate multi-level non-volatile memories. We present two memory options, suited either for current-mode or for time-domain vector-matrix multiplier implementations, with low-voltage charge-injection program and erase operations. The effects of a limited accuracy are also investigated through system-level simulations, by accounting for the temperature dependence of the stored weights and the corresponding impact on the network error rate.

Keywords: DNN, single-poly FG, Floating-Gate, Vector-Matrix Multiplier.

1. Introduction

The widespread diffusion of artificial intelligence functionality in electronic systems is calling for dedicated technologies and specialized hardware to realize Deep Neural Networks (DNNs), aiming at higher performance per unit energy with respect to processors based on the Von Neumann paradigm [1], [2]. The most frequently recurring building block of a DNN is the vector-matrix multiplier (VMM), for which a block diagram – which can also be interpreted as a parallel architecture – is illustrated in **Figure 1**. An efficient VMM computation is crucial for the overall performance of the neural network. During the inference phase, data input vectors X=[x_1, …x_i,… x_M] are multiplied by a matrix of programmable weights ($w_{i,j}$), which have been previously determined during the training phase, based on an available dataset of labelled data.

The recurring arithmetic operations implemented in a VMM cannot be efficiently performed by a general-purpose CPU, thus various degrees of parallel and/or logic in-memory architectures need to be implemented to boost overall DNN performance and efficiency [2], [3]. In this context, together with digital approaches such as GPU-based hardware and ASIC accelerators [4], computation in the analog domain is gaining momentum as a longer term solution [5]–[10], on the basis of the observation that limited equivalent arithmetic precision in the inference phase is sufficient for a DNN to achieve a high classification accuracy

[11]. Analog architectures can reach a very high computation efficiency, exploiting fundamental circuit laws and intrinsic device properties to execute in-memory arithmetic operations: addition, for example, can be simply obtained by exploiting Kirchhoff's current law in summing several currents coming from various branches and injected into the same node (**Figure 1**). On the other hand, it is well known that analog processing blocks can be affected by circuit non-idealities such as noise, non-linearity and process variations [5], [9], [10] which can be counteracted at the cost of increased area occupation.

In this paper we explore the implementation of VMM blocks in the analog domain, using a 180 nm CMOS single-poly process technology. We present the design of analog non-volatile memory cells to store the matrix weights, fabricated with a single-poly Floating-Gate (FG) structure. We discuss two memory options that can be obtained through layout optimization only, with no need to modify the standard process flow. Experimental results confirm the possibility to achieve multi-level storage capability. We describe the implementation of two analog VMM approaches based on the proposed memory cells, with a comparison of their main Figures of Merit (FOMs), namely Energy Efficiency (EE), latency time (T_{LAT}) and area occupation. Finally, we discuss the dependence of the analog weights upon the operating temperature and the resulting impact on the DNN inference accuracy.

2. Analog VMM implemented with single poly memory options

Figure 2 depicts two possible architectures that can be implemented to realize an analog VMM. In **Figure 2(a)**, a current-mode approach is shown, which exploits programmable current mirrors. The scalar product is implemented by multiplying the input signal, encoded as an input current, with the stored current-mirror magnification factor: the output currents of the cells of the same column are summed at the same node to perform the vector-matrix multiplication. We performed a detailed analysis of the current-mode VMM and of the programmable mirror cells in [5]. In a time-domain approach (shown in **Figure 2(b)**) [13] the input signals are encoded as voltage pulse widths, which are applied to the matrix rows and activate the corresponding cells for a given time. The programmability of the conductance of each cell enables a tunable on-state current (i.e., the weight) for a fixed voltage bias. The cell total charge is the result of the multiplication between the input pulse width and the cell current, while the whole charge is collected to the same node. Finally, a charge-to-voltage conversion is performed by the integrator block.

We designed and fabricated two single-poly memory cell options which can be used to implement the introduced VMM approaches using a standard CMOS technology node. In the following sub-sections, we will first discuss the experimental results performed on the two memory cells, and then we will present a system-level assessment of the various FOMs of a current-mode versus a time-domain technique.

A. Single-poly Floating-Gate cell options

In general, a conventional figure of merit for an analog DNN is the equivalent-number-of-bits (ENOB), which enables a quick comparison with a digital version of the network. In an analog function, the ENOB is limited by device noise and circuit transfer-function non-linearity, affecting the accuracy of the performed operation. Although neuromorphic algorithms are resilient to noise and limited accuracy, it has been shown that ENOB values of 5 bits can be acceptable for simple networks [5], but an ENOB of a at least 7 bits is needed for more complex networks such as AlexNet [12]. FG memory cells with multi-level programmability are then needed to store weights with a sufficient precision and dynamic range.

The programmable current mirror basic cell was designed by adding a p-type MOS capacitor (pCAP) in series to the gate of a n-type transistor, thus sharing the same non-accessible poly (layout shown in **Figure 3(a)**) and realizing the FG. The obtained cell has an equivalent circuit model similar to the one of a double-poly flash memory, but the external control gate (CG) is the n-well of the pCAP instead of the terminal on the top poly. The lower cost of the single poly technology is traded off against a larger area occupied on the chip. The I_D-V_{CG-S} transfer characteristics of the resulting device is close to the one of the intrinsic nMOS, but with a slightly degraded slope due to the non-ideal coupling factor of the CG to the channel. This issue is critical because the poor electrostatics can impact the output resistance of the current mirror cell, thus limiting the linearity of the operation. We realized different cells for various pCAP/nMOS area ratios (i.e., coupling), and measurements demonstrated the possibility to reach the linearity corresponding to an ENOB close to 5 for cells with area ratios larger than 50× and a nMOS channel-length ≥ 0.5 µm. In Figure **3(b)**, various I_D-V_{CG-S} curves are displayed for different stored weight conditions, measured after progressive program operations that result in an increased V_{th}. Cell programming is performed by exploiting a self-convergence scheme, where the cell state target is achieved when concurrent injections of holes and electrons compensate each-others. Thus, for long pulses, the final state depends only on the applied voltages and can be modulated with a fine tuning of the CG voltage for a fixed drain voltage. In particular, the V_{th} is increased by applying appropriate voltage values around ~6 V to the drain and values in the 4 - 6 V range to the CG, activating impact-ionized hot-electron injection, following the scheme reported in [5]. Lowering the CG voltage to ~1.5 V favors impact-ionized hot holes injection, leading to a V_{th} decrease.

The considered alternative approach consists in using a single transistor, without the need of a pCAP, therefore realizing a two-terminal FG cell: this cell suits well for implementation in a time-domain VMM, where one terminal is connected to the integrator input, kept at a constant reference voltage, while the time pulse is applied to the other terminal. In this case, the CG role is played by the drain itself, through the coupling provided by the gate-to-drain capacitance [14]. With the objective of targeting minimum area, a minimum size n-type MOSFET is used to realize the FG cell: the layout of a 2×2 array is shown in **Figure 4(a)**, reporting a x- and y-pitch of 1.12 and 1.54 µm, respectively. Program is obtained by applying progressive V_{DS} pulses of ~5 V, activating hot electron injection phenomena, until the target state is reached. On the other side, an erase operation is performed using a single 6 V pulse, favoring the injection of holes. Measured I_D-V_D curves at different programmed levels, displayed in **Figure 4(b)**, show a high degree of programmability, with an I_{MAX}/I_{min} ratio of more than 3 orders of magnitude for a read voltage of 1 V.

TABLE I
VMM CHARACTERISTICS

	CM-based FG-cell	*Two terminal FG-cell*
Area (mm^2)	*1.383*	*0.038*
Latency time (µs)	*100*	*3*
Throughput/area (GOPs/(s·mm^2))	*0.144*	*174.6*
EE (TOPs/J)	*36.8*	*14.5*

B. Figures of Merit Evaluation for the proposed VMMs

Characteristics of designed VMMs heavily impact the overall performance of a DNN. Since all data inputs are processed in parallel, then the latency time T_{LAT}, defined as the time to perform one MAC (multiply and accumulate) operation, represents the computation time of the entire VMM. An M×N sized VMM can perform (2M-1)×N elementary operations (i.e. sums or multiplications), while the energy required to perform these operations defines the energy efficiency (EE), expressed as TOPs/J. The comparison between the CM and TD approaches was evaluated by circuit level (Cadence) and system level simulations (MATLAB) performed on a 100×100 fully-connected layer of a DNN, by appropriately sizing the transistor cells and the circuit implementation of the VMMs to reach an ENOB of 6-bit. Results were compared in TAB.I. The need for a large pCAP to realize the memory cells used in the CM based approach leads to a 100×100 VMM layout area of 1.383 mm^2, and to a latency time T_{LAT} = 100 µs, corresponding to an EE of 36.8 TOPs/J. On the other hand, in the TD approach we can rely on minimum sized transistors, leading to a layout area of 0.038 mm^2. The latency time is evaluated on the basis of the maximum input time, i.e. the period T, which for a target ENOB of 6 bits is 1.5 µs. Therefore, we will obtain $T_{LAT} = 2T = 3$ µs, since two periods are needed for each operation (one for integration and one for sampling and reset). As regards the EE, its value of 14.5 TOPs/J is mainly constrained by the power dissipated by the integrator. Although the TD approach has a slightly lower EE compared to the CM case, the smaller area and lower latency time leads to a much better throughput/area figure-of-merit of *174.6 GOPs/(s·mm^2)*, which is 3 order of magnitude higher than the one of the CM based solution, which is *0.144 GOPs/(s·mm^2)*.

3. Impact of temperature on the weights and on DNN inference accuracy

In both VMM architectures, the weight is determined by the net charge injected in the FG, which in turn results in a shift of the threshold voltage. In the CM based multipliers, this shift determines the current magnification factor between the input and the output current, whereas in the two-terminal FG cells it determines the equivalent on-resistance. Since both cases operate in the sub-threshold region, small variations of the threshold voltage result in an exponential variation of the drain current (and therefore of the corresponding weight) that can be easily evaluated, resulting in the following:

$$w(T)_{\mathrm{CM}} = e^{\frac{\Delta V_{th}}{\frac{kT}{q}}} = \mathrm{w}(T_0)^{\frac{T_0}{T}} \qquad \text{CM cells}$$

$$w(T)_{\mathrm{TD}} = e^{\frac{V_{GS}-V_{th}(T)}{\eta\frac{kT}{q}}} = w(T_0)e^{\alpha\frac{T-T_0}{T}} \qquad \text{two-terminal cells}$$

where w(T$_0$) is the weight at the room temperature (T$_0$ = 300 K) at which the VMM is nominally operating, *k* is the Boltzmann constant, *q* is the elementary charge and $\alpha > 0$ is constant with T. We assume that the network weights have been trained at room temperature, and then that w(T$_0$)$_{(i,j)}$ are the nominal weights.

First, a system level simulation study has been performed using weights trained at T$_0$. Then, the DNN inference accuracy has been evaluated by accounting for the dependence on temperature. As benchmark, the well-known AlexNet [12] has been trained with 1000 images distributed in 50 classes from ImageNet [15] dataset for a total of 60 epochs.

For the two approaches, the inference operation performed by AlexNet at room temperature resulted in a classification accuracy of 95% [5]. The impact of temperature variation on the inference accuracy is shown in **Figure 5**. One should note that the error rate of the CM based AlexNet is below 10% in the range from ~-7 °C to ~ 40 °C, while for the TD the range is restricted to only ~ 20 °C to ~ 30 °C. The temperature impact on the classification accuracy can be mitigated with different approaches. For instance, one could rely on differential weights to partially counteract the weight sensitivity to temperature. In addition, considering the asymmetric shape of the error as a function of the temperature variation, a slightly different set of trained weights could be used to center the error shape around the nominal temperature.

4. Conclusions

The opportunity to realize single-poly non-volatile floating-gate cells for analog neural networks has been explored. UMC 180 nm technology has been used for the first test vehicle. The following results have been demonstrated: (1) realization of two FG cell options for different vector-matrix multiplier approaches, (2) successful multi-level analog storage capability experimentally demonstrated, (3) relatively low voltage program and erase operations. The main FOMs have been evaluated for both the CM and TD approaches, demonstrating that the CM architecture has a slightly higher EE, while the TD one can perform more operations per unit area. Moreover, a system-level simulation has been performed on AlexNet to analyze the impact of temperature variations on the inference capability, demonstrating that a DNN network implemented with proposed approaches can severely be impacted by temperature variations.

Acknowledgment

This work was partially supported by the European Commission through the QUEFORMAL H2020 Project (contract no. 829035).

References

[1] V. Sze, Y. Chen, J. Emer, A. Suleiman and Z. Zhang, “Hardware for machine learning: Challenges and opportunities,” 2017 IEEE Custom Integrated Circuits Conference (CICC), Austin, TX, 2017, pp. 1-8, doi: 10.1109/CICC.2017.7993626.

[2] X. Xu et al., “Scaling for edge inference of deep neural networks,” Nat. Electron., vol. 1, no. 4, pp. 216–222, Apr. 2018, doi: 10.1038/s41928-018-0059-3.

[3] Lukas Cavigelli, Michele Magno, and Luca Benini. 2015. “Accelerating real-time embedded scene labeling with convolutional networks”. In Proceedings of the 52nd Annual Design Automation Conference (DAC '15). Association for Computing Machinery, New York, NY, USA, Article 108, 1–6. DOI:https://doi.org/10.1145/2744769.2744788

[4] Y. Chen, T. Krishna, J. S. Emer, and V. Sze. “Eyeriss: An Energy-E-cient Reconfigurable Accelerator for Deep Convolutional Neural Networks”. In: IEEE Journal of Solid-State Circuits 52.1 (2017), pp. 127–138.

[5] M. Paliy, S. Strangio, P. Ruiu, T. Rizzo, and G. Iannaccone, “Analog Vector-Matrix Multiplier Based on Programmable Current Mirrors for Neural Network Integrated Circuits,” IEEE Access, vol. 8, pp. 203525–203537, 2020, doi: 10.1109/ACCESS.2020.3037017.

[6] K. Berggren et al., “Roadmap on emerging hardware and technology for machine learning,” Nanotechnology, vol. 32, no. 1, p. 012002, Jan. 2021, doi: 10.1088/1361-6528/aba70f.

[7] J. Lu, S. Young, I. Arel, and J. Holleman, “A 1 TOPS/W analog deep machine-learning engine with floating-gate storage in 0.13 µm CMOS,” IEEE J. Solid-State Circuits, vol. 50, no. 1, pp. 270–281, 2015, doi: 10.1109/JSSC.2014.2356197.

[8] J. Binas, D. Neil, G. Indiveri, S.-C. Liu, and M. Pfeiffer. “Precise neural network computation with imprecise analog devices”. 2016. arXiv: 1606.07786 [cs.NE].

[9] M. Judy et al., “A Digitally Interfaced Analog Correlation Filter System for Object Tracking Applications,” in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 65, no. 9, pp. 2764-2773, Sept. 2018, doi: 10.1109/TCSI.2018.2819962.

[10] M. M. Hasan and J. Holleman, “Implementation of Linear Discriminant Classifier in 130nm Silicon Process,” in 2018 IEEE International Symposium on Circuits and Systems (ISCAS), 2018, vol. 2018-May, no. 3, pp. 1–5, doi: 10.1109/ISCAS.2018.8351829.

[11] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio. ”Quantized Neural Networks: Training Neural Networks with Low Precision Weights and Activations”. In: J. Mach. Learn. Res. 18.1 (Jan. 2017), 6869–6898.

[12] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks”, Proceedings of the 25th International Conference on Neural Information Processing Systems, pp.1097–1105, 2012.

[13] M. Bavandpour, M. R. Mahmoodi and D. B. Strukov, “Energy-Efficient Time-Domain Vector-by-Matrix Multiplier for Neurocomputing and Beyond”, IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 66, no. 9, pp. 1512-1516, Sept. 2019. DOI: 10.1109/TCSII.2019.2891688.

[14] Danial, L., Pikhay, E., Herbelin, E. et al., “Two-terminal floating-gate transistors with a low-power memristive operation mode for analogue neuromorphic computing”, Nature Electronics, 2, pp. 596–605, 2019. DOI: https://doi.org/10.1038/s41928-019-0331-1

[15] The ImageNet database. [Online] Available: http://image-net.org/

Figures:

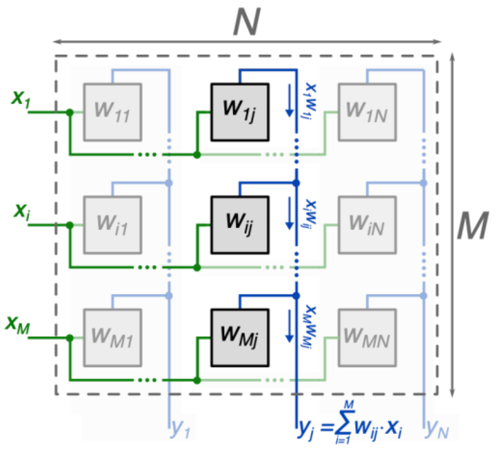

Figure 1: Vector-matrix multiplier block diagram.

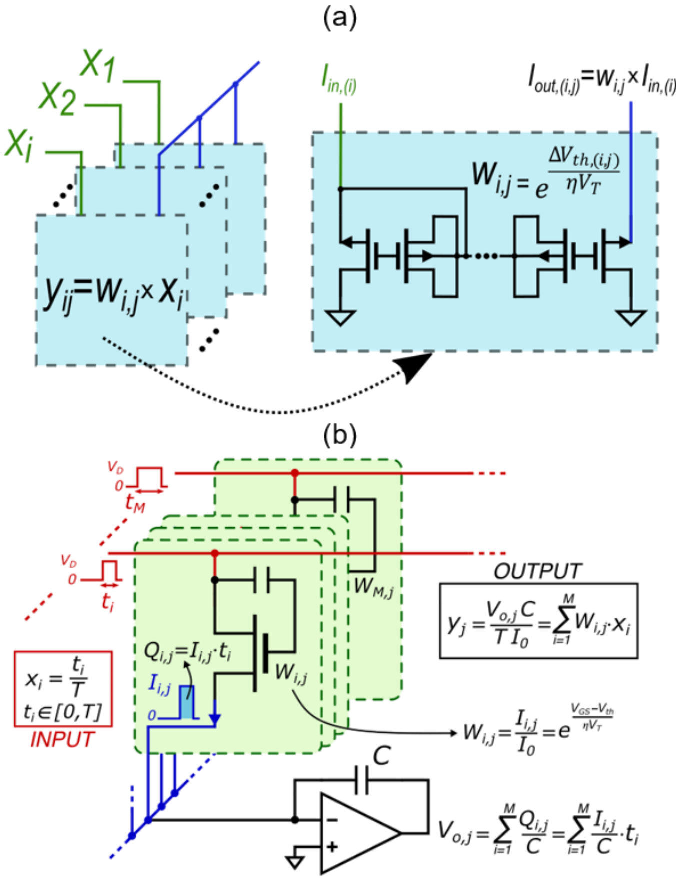
(a)
X_1
X_2
X_i
$y_{ij} = w_{i,j} \times x_i$
$I_{in,(i)}$
$I_{out,(i,j)} = w_{i,j} \times I_{in,(i)}$
$W_{i,j} = e^{\frac{\Delta V_{th,(i,j)}}{\eta V_T}}$
(b)
V_D
0
t_M
t_i
$W_{M,j}$
$Q_{i,j} = I_{i,j} \cdot t_i$
$I_{i,j}$
$W_{i,j}$
$x_i = \frac{t_i}{T}$
$t_i \in [0,T]$
INPUT
OUTPUT
$y_j = \frac{V_{o,j} C}{T I_0} = \sum_{i=1}^{M} W_{i,j} \cdot x_i$
$W_{i,j} = \frac{I_{i,j}}{I_0} = e^{\frac{V_{GS} - V_{th}}{\eta V_T}}$
C
−
+
$V_{o,j} = \sum_{i=1}^{M} \frac{Q_{i,j}}{C} = \sum_{i=1}^{M} \frac{I_{i,j}}{C} \cdot t_i$

Figure 2: Implementation of an analog VMM using (a) a current mode and (b) a time-domain approach.

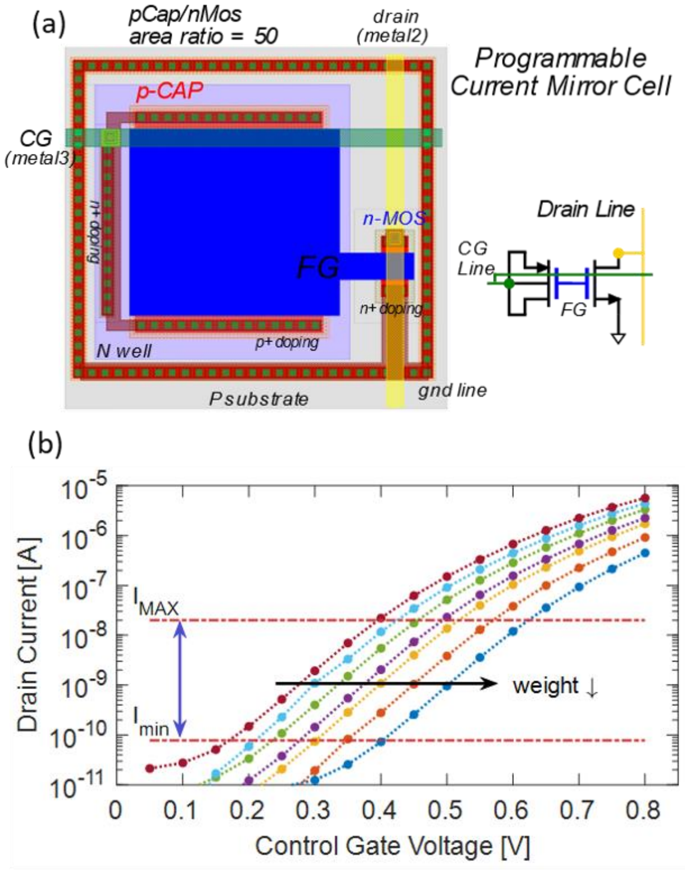

Figure 3: FG programmable Current-Mirror cell: (a) layout and schematic representation; (b) measured I_D-V_{CG-S} characteristic for various programming levels.

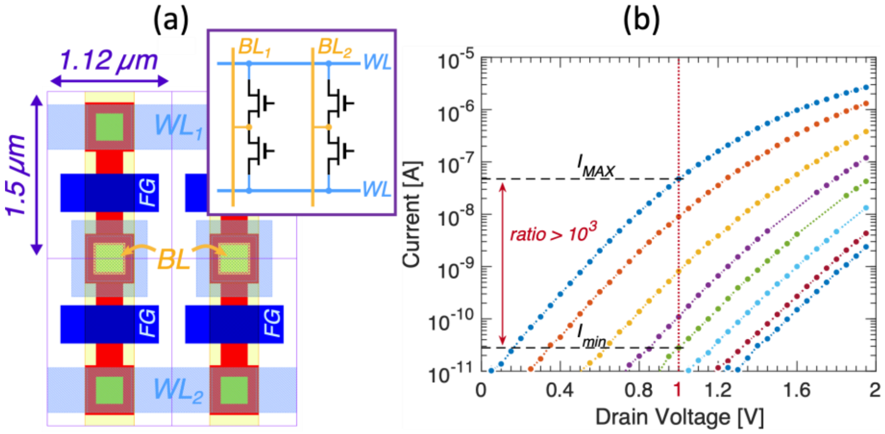

Figure 4: Two terminal FG cell: (a) layout of a 2×2 array composed of two pairs of mirrored cells, with the inset showing the corresponding electrical scheme; (b) measured I_D-V_D charactertistic for various programming levels.

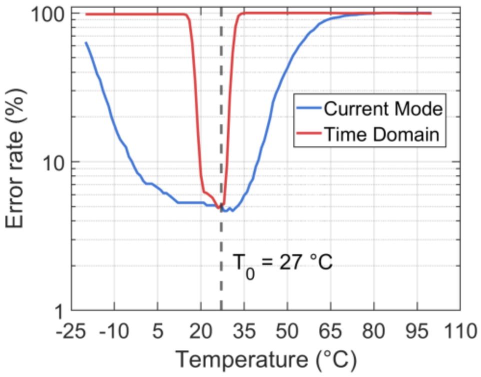

Figure 5: Impact of the temperature on the precision of AlexNet for both the TD and CM techniques.