Spin Orbit Torque-based Crossbar Array for Error Resilient Binary Convolutional Neural Network

Kamal Danouchi, Guillaume Prenat, Lorena Anghel

Spin Orbit Torque-based Crossbar Array for Error Resilient Binary Convolutional Neural Network

Kamal Danouchi, Guillaume Prenat, Lorena Anghel, *Senior, IEEE*

***Abstract*—Convolutional Neural Network (CNN) is one of the most important Deep Neural Networks (DNN) classes that helps solving many tasks related to image recognition and computer vision. Their classical implementations by using conventional CMOS technologies and digital design techniques are still considered very energy-consuming. Floating point CNN relies primarily on MAC (Multiply and ACcumulate) operation. Recently, cost-effective Bite-wise CNN based on XNOR and bit-counting operations have been considered as a possible hardware implementation candidate. However, the Von-Neumann bottleneck due to intensive data fetching between memory and the computing core limits their scalability on hardware. XNOR-BITCOUNT operations can be easily implemented by using In Memory Computing (IMC) paradigms executed on a memristive crossbar array. Among emerging memristive devices, the Spin-Orbit Torque Magnetic Random Access Memory (SOT-MRAM) offers the possibility to have a higher ON resistance that allows reducing the reading current, since all the crossbar array is read in parallel. This could contribute to a further reduction of energy consumption, paving the way for much bigger crossbar designs. This study presents a crossbar architecture based on SOT-MRAM with very low energy consumption; we study the impact of process variability on the synaptic weights and perform Monte-Carlo simulations of the overall crossbar array to evaluate the error rate. Simulation results show that this implementation has lower energy consumption with respect to other memristive solutions with 65.89 fJ per read operation. The design is also quite robust to process variations, with very low reading inaccuracies up to 10 %.**

I. Introduction

The deluge of data generated by the digital world has seen the emergence of tools to extrapolate useful information out of it, such as Deep Neural Network (DNN). Nowadays these networks complete several human tasks, such as facial recognition, voice recognition, or language processing [1] [2] to name just a small number of them. However, the hardware implementation of these networks still raises many questions about energy consumption, mostly due to the Von Neumann bottleneck [3], the continuous movements between the processing core and the memory being the main cause of energy over-consumption for Artificial Neural Network (ANN). To tackle this issue, one of the solutions is to perform the calculation inside the memory (In Memory Computing - IMC) avoiding data fetching between the processing unit and the memory. This paradigm relies on the use of crossbar arrays, that performs the computation of NN operations in an analog fashion and in one cycle. This architecture stores the weights in a Non-Volatile memory (NVM) based on memristive devices. Inputs will be applied in parallel under the form of voltages to the crossbar matrix then weighted by the resistive synaptic devices and summed along the column thanks to Kirchhoff's laws. However, encoding weights on synaptic devices is not straightforward as we are limited by the number of discernible conductance states of the synaptic devices.

In the past few years, many studies have proposed quantization methods to avoid using floating point parameters which are energy and memory intensive such as Ternary Neural Network (TNN) or Binary Neural Network (BNN) [4] [5] [6] [7]. Binary networks could be seen as the lowest level of precision or quantification, weights and activation have only binary values (+1[1] and -1[0]). This feature simplifies the weight mapping, reduces memory access and the overall energy consumption.

On the other hand, memristive devices used to store synaptic weights are subjected to non-idealities due to process variation, temperature, and parasitic effects [8]. This, coupled with the loss of information due to the binarization can result in a loss of accuracy in hardware implementation. In the present work, a binary crossbar architecture is implemented using Spin-Orbit Torque Magnetic Random Access Memory (SOT-MRAM) devices in which we evaluate the errors that can induce accuracy loss in the network. Then, we study the variability impact on the output of the crossbar through intensive simulations. We show the high energy efficiency of the SOT-MRAM crossbar array and its robustness against process variations.

II. Preliminary

A. Related Work

Our research focuses on the implementation of BNNs on hardware for IMC accelerators. In this scope, there is a significant number of studies on the implementation of these networks, particularly on Resistive Random Access Memory (RRAM) devices with the conventional 2T-2R bit-cell. In [9] [10], the authors emphasize the reliability of such structures against errors. For a crossbar implementation, the most prominent study is XNOR-RRAM [11], where the authors implemented XNOR and POPCOUNT operations in parallel as for conventional DNN implementation. The study in [12] presents a 4T2R bitcell robust against device variations affecting the inference operations. In spintronics technology, [13] unveiled a BNN implementation where the authors replaced

The authors are with the Univ. Grenoble Alpes, CEA, CNRS, Grenoble INP, IRIG-Spintec, 38000 Grenoble, France (e-mail: kamal.danouchi@cea.fr)

978-1-6654-5707-1/22/$31.00 ©2022 IEEE

the traditional summation of currents in a crossbar with a summation of resistances thus reducing the energy consumption of Spin-Transfer-Torque Magnetic Random-access Memory (STT-MRAM). Nonetheless, the main limit of this approach is the number of devices that could be implemented in the array due to analog noise. In addition, the use of time-to-digital converters in their design induces a higher latency. In this paper, we present a 4T-2R SOT-MRAM crossbar implementation for BNN inference. The SOT-MRAM separates the read and write operation, therefore the write operation is not limited by the Magnetic Tunnel Junction (MTJ) resistance level and its resistance can be tunned up to dozen of MΩ. The SOT can therefore be implemented in a traditional crossbar scheme. Moreover, the SOT-MRAM offers also two stable states and it is less affected by drift and retention issues than RRAM and Phase Change Memory (PCM) devices [14].

B. Convolutional Neural Network

Convolutional Neural Network (CNN) is a bio-inspired Artificial Neural Network (ANN), mainly used for image recognition [15]. It consists of convolution and max pooling layers used to extract feature maps and a Fully-Connected (FC) layer used for classification. The objective of the network is to classify with a higher or lower probability which image is fed into the network using Multi-Perceptron Layers (MLP). Since convolution is the most computation-intensive operation in CNN, several studies have proposed a bitwise CNN and specific memory access aware strategies. [16], [17].

C. Binary Neural Networks

Due to their extreme compression of the operands used in the computation, BNNs have been considered for implementation on embedded devices [18]. Unlike classical neural networks, BNN encodes the weights as well as the neuron activation values in binary format, rather than floating point formats. To allow this, during training the real values of the variables are compared to a threshold, accordingly to that the network will generate either a +1[1] or -1[0] as can be seen below:

$$sign(x) = \begin{cases} +1 & x > 0 \\ -1 & x < 0 \end{cases} \tag{1}$$

Here x represents a floating point value calculated during training, after passing through the sign function, x is binarized. Unlike the Multiplication and ACcumulation (MAC) operations performed by conventional networks, binary networks perform a logical XNOR operation followed by a pop-count operation. This facilitates its implementation on the hardware since these operations are not very energy consuming. As in most DNNs today, a batch normalization (BN) layer is added to stabilize and accelerate the training phase [19]. This layer is also simply implemented by a threshold operation:

$$\tau = \mu_\beta - \frac{\beta * \sqrt{\sigma_\beta^2 + \varepsilon}}{\gamma} \tag{2}$$

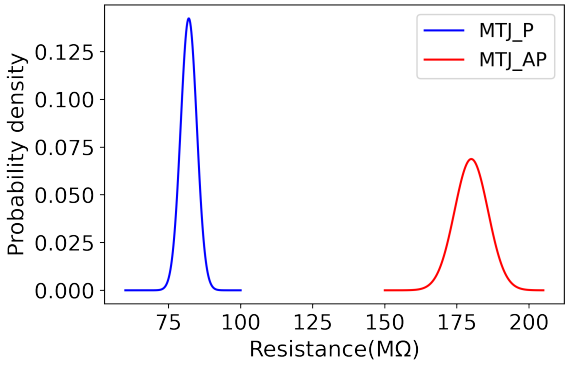


Fig. 1: Resistance distribution of R_P and R_{AP} of SOT-MRAM

Where γ and β parameters are computed during the training and used to scale and shift the normalized value. While μ_B and σ_B^2 represent the mean and the variance of the mini-batch B. The algorithmic-generated thresholds will later on be used at the output of the crossbar as activation.

D. Spin-orbit torque magnetic random access memory

In recent years, many resistive synaptic devices have been proposed to implement DNN [14], such as PCM, RRAM or STT-MRAM. STT-MRAM devices offer advantages in terms of writing speed (few ns), endurance ($\geq 10^{12}$ cycles), and low power consumption (around $10fJ$) [14], [20]. Thus for memory applications, a lot of companies have invested in STT-MRAM with already products on the shelves [21] [22]. Notwithstanding its strong performance for memory implementation, STT-MRAM is not a suitable candidate for conventional crossbar implementation due to its low resistance level (i.e. few $k\Omega$). Indeed, to perform Matrix-Vector Multiplication (MVM) in the array, all the bitcells are simultaneously read in parallel for an analog computation. As a result, this low resistance level will lead to a significant power draw in the array. To overcome this concern, SOT-MRAM is proposed [23]. According to the authors, the low resistance state of this memory could be tuned up to 100 MΩ. The SOT-MRAM is a three-terminal device that consists of an MTJ mounted on a heavy metal substrate, with the MTJ being the main core of all MRAM devices. This nanostructure is composed of two ferromagnetic layers separated by a tunnel barrier, with one of the layers being the Free Layer (FL) and the other one the Reference Layer (RL). The resistance value depends on the relative orientations of the magnetizations in the two layers, parallel (P) or anti-parallel (AP). The P state exhibits a low resistance state and the AP state a high resistance state, the resistance can be then measured through the tunnel magneto-resistance effect (TMR). The writing is performed thanks to a current passing in the writing line below the MTJ, inducing the SOT effect. The SOT-MRAM separates the read and write paths avoiding accidentally switching the MTJ while reading. This separation of the write and read paths allows tuning the read and write parameters of the device independently, allowing a large resistance of the stack. For

this purpose, two access transistors are added for read and write operations controlled by a Write Word-Line (WWL) and a Read Word-Line (RWL). Given its high resistance, SOT-MRAM may be an important candidate for the implementation of neural networks. Nevertheless, SOT-MRAM like other emerging memristive solutions is also subject to process variation that could impact several characteristics of the MTJ such as the writing time or the TMR. Such discrepancies can alter the desired result. To catch these defects, global variation ($+/-3\sigma$) are added in the SOT-MRAM model on the TMR and the Resistance-Area product (RA), see Fig. 1. In addition to the variations of resistance, the crossbar is also subject to the variation that can impact the selection transistors and periphery like sense amplifiers. We propose in this work to study all those variations and see how they impact the design and the performance of the crossbar array.

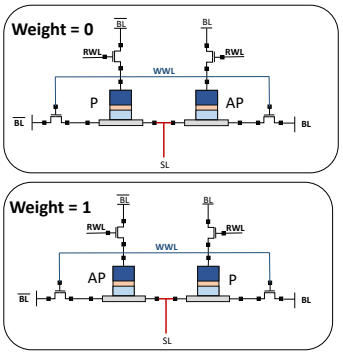
III. Implementation

A. SOT-MRAM bit-cell design

In BNNs, the basic computation consists in an XNOR-bitcount operation. Some studies proposed to implement it with purely digital operation [24] [12] [25] and other studies proposed to stay on the implementation of crossbar by exploiting Ohm and Kirchhoff's laws [26]. In these implementations of BNN, two complementary cells are used to store a synaptic weight [3]. In the proposed implementation of Fig. 2, the synpatic weight of '1' is stored with a AP and P configuration state, for '0', the configuration is (P-AP). The inputs of the cell (BL) are still differential to allow XNOR operation. In contrast to the 2T-2R implementation [26], the SOT-MRAM is a 3-terminal device where another NMOS transistor is added to separate the read and write paths. This has the advantage of avoiding read disturb, minimizing errors. To illustrate the XNOR operation in the proposed bit-cell, let's assume that the input is '1' and the wanted weight is '0', in this case AP path will be activated leading to a low current , the output will be equal to AP (0). In the scenario where '1' is the weight and the input is also '1', the P path is chosen leading to a larger current than AP, the output will be '1'(P). The current generated by all the cells in a column will be summed up and compared to a threshold voltage by a sense amplifier (SA) [27], avoiding the use of an analog to digital converter (ADC). This particular feature of BNN will allow reducing the area as well as the power consumption of the ADC, since the periphery is the block that consumes most of the energy in a crossbar architectures [3], [28].

B. SOT-MRAM-based crossbar array

In this study, a convolution layer implementation is proposed on a crossbar architecture. Prior to that, an algorithmic study with a bitwise CNN was performed on the MNIST dataset. The network architecture is presented in Table I. This model contains 2 convolutional layers, 2 max-pooling layers and 1 FC layer . The output represents here the feature maps.



XNOR OPERATION

Input (BL)	Weight	Output (SL)
0(-1)	0	P (1)
0(-1)	1	AP(0)
1(+1)	0	AP(0)
1(+1)	1	P(1)

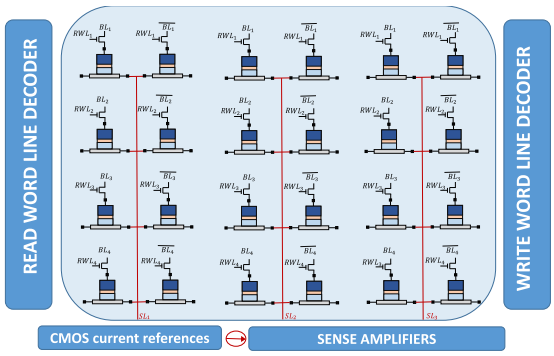
Fig. 2: Bitcell containing two complementary SOT-MRAM cells to perform XNOR operation in memory with the truth table

TABLE I: Binary CNN configuration for MNIST dataset

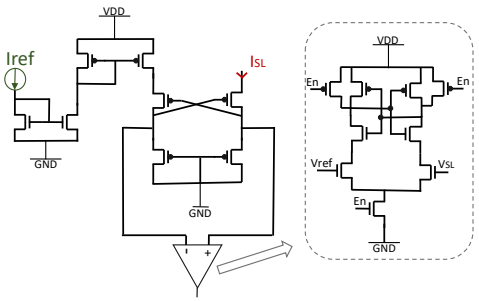
Layer name	Kernels	Output
Convolution-1	5×5×32	24×24×32
Max Pooling-2	3×3	8×8×32
Convolution-3	5×5×16	4×4×16
Max Pooling-4	2×2	2×2×16
Fully Connected	64×10	10

In this paper only Convolution-1 is treated. This layer performs 32 convolutional operations with kernels of 5×5 size. To compute the convolution in an analog fashion, it has to be mapped on a physical NVM-based array. However, mapping a convolutional layer is more challenging than FC layer, single shot mapping is no longer possible. Yet, the convolution realises a dot product between the kernel weights and a specific portion of the input image, several times. Hence a simple method of mapping the convolutional operation is to unroll all the kernels in 1D and arrange them in each column of the crossbar array [29], [30]. For the first convolutional operation, the array contains then 25 inputs (5×5) for 32 outputs. During the inference, only one writing operation of all weights is performed, since they do not change with time.Thus, the weights obtained during the training stage are loaded into the crossbar array.

After the writing step of the weights, several reading steps are carried out for the inference. Indeed, to obtain all the feature maps of the first convolutional layer, 24×24 reading operation per column are required. In order to perform these operations, it is first necessary to encode the MNIST image into reading signals. To do so, the image is sliced in multiple waveform that encode the value of the pixel in the amplitude of the signal, "VDD" for the black pixel and the "GND" for the white one. For both, a reading time of 20 ns with 1 V of amplitude is used to sum all the weights in crossbar and perform the bit-counting operation. Here, the bitwise batch normalization layer explained in Sec. II-C is also added to the crossbar array. The computed thresholds during the training phase (i.e. $\tau = \mu_\beta - \frac{\beta * \sqrt{\sigma_\beta^2 + \varepsilon}}{\gamma}$) are mapped at the circuit level, this will provides several activation levels for the array. To accomplish this mapping, a first estimation of the maximum



(a) SOT crossbar array during reading operation



(b) CMOS current reference and sense amplifier

Fig. 3: SOT Crossbar array implementation

TABLE II: Value of the CMOS current references

Reference number	Value (nA)
0	19
1	38
2	57
3	76
4	95
5	114
6	133
7	152
8	190

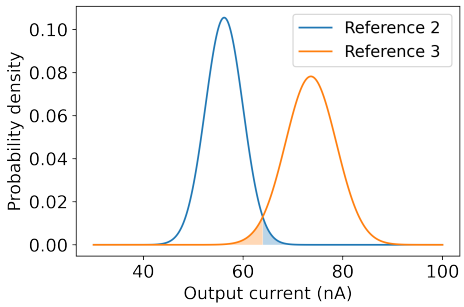


Fig. 4: CMOS current reference overlap

current generated by the crossbar given the weights in the array is realized. After that, a normalization between the maximum current produced by the array with the maximum threshold obtained through the training is performed.

These thresholds are simply represented by a CMOS current reference [31], with which each column output of the crossbar is compared thanks to a sense amplifier [27]. In Fig. 3, the crossbar architecture and the CMOS current reference are shown. For clarity reasons, we represent only reading path transistors in the crossbar. In this application, the batch normalization thresholds for the 32 kernels are 9 in total (some kernels have similar values). It is therefore necessary to design 9 different CMOS current references (see Table II).

Nonetheless, process variations can induce an overlap between two references that are close to one another. Thus, two distinct references can generate the same current leading to some reading inaccuracies. An example is shown in Fig. 4 where Monte Carlo (M-C) simulations show an overlap between reference 2 and reference 3. To further evaluate the impact of this overlap, 100 M-C simulations have been performed on each CMOS current reference and the design was accordingly adapted with accurate sizing of the Width/Length (W/L) of the transistors. The aim is to reduce the overlap to the minimum, making it possible to discriminate between the current values. Fig. 5 shows the accuracy of the different references generated for the reading of the crossbar. It indicates the percentage of being in the desired current range according to the chosen current reference (and not in the neighboring current distribution). The studied CMOS references have a low dependency and low variability over process.

C. Crossbar evaluation

To evaluate the accuracy of the proposed crossbar array architecture over manufacturing variations, several reading operations (i.e 200) have been performed based on an image from the MNIST dataset. The objective is to assess the robustness of the crossbar against variations and the impact of accuracy on the desired output. To do so, 100 MC simulations were conducted on the aforementioned reading operations. First, only the CMOS current references were considered. Fig. 6 shows that the output is just slightly affected and we observe only 4% of error over the whole reading operations. Therefore, the robustness of the current references used at the output of the crossbar is confirmed. Secondly, to assess how the distribution of resistance showed in Fig. 1 can impact the desired output, a similar test to the one before is carried out. This time, only the crossbar is taken into consideration. Compared to the previous study, we observe a little increase in the error. This can be explained by the fact that in some worst-case corners, all the bitcells will be impacted in such a way to drastically modify the generated reading current, inducing even more reading errors. Finally, M-C simulations on a combination of the crossbar and the current references were achieved. This time, the error is even more important with a bigger deviation. This case could be seen as the worst case of the study since a decorrelation between the reference current and the current produced by the crossbar array can occur. Although in the latter case the errors are larger, the

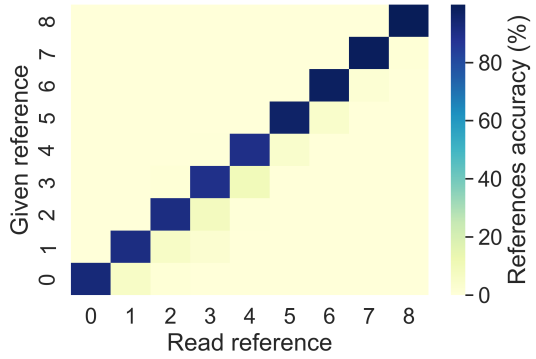


Fig. 5: Accuracy of the generated references

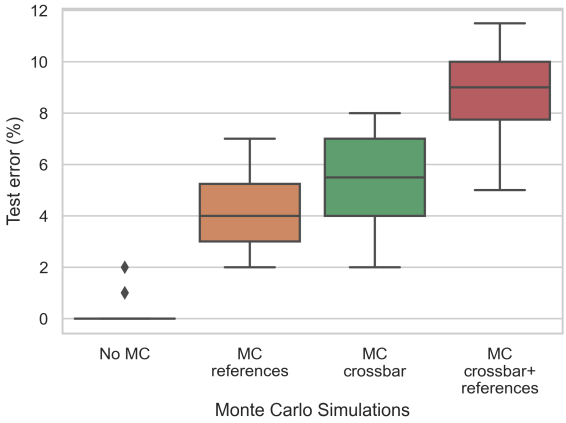


Fig. 6: Test error evaluated for MNIST dataset

overall trend is on average less than 10%.

In Fig. 7, we see the power consumption of the proposed architecture, with a 28 nm technology. The first thing to point out is the low power draw at the crossbar level. By the extremely high resistance of the SOT memory, it is possible to drastically reduce the power consumption in memory. The principal sources of power consumption in this architecture are the decoders and the peripherals. In fact, it is known that the periphery is the part that consumes the most energy in a crossbar architecture. However, the implementation of the activation by means of a current reference and a sense amplifier reduces the energy consumption compared to an ADC [3].

IV. Discussion

In this study, a 1600-component crossbar array was designed using SOT-MRAM for a BNN. To do so, we implemented the XNOR bitcell and performed the BNN computation in memory. In addition, the Batch Normalisation layer was also mapped with current references and sense amplifiers. Then, several M-C simulations were performed to evaluate the impact of process variations on the proposed SOT crossbar array, whether at the bitcell level or on the periphery. For inference implementation, [9] presented an RRAM array where they achieved 25 nJ of energy consumption with 2000

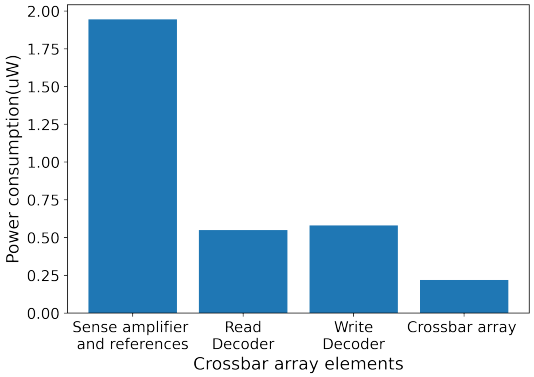


Fig. 7: Power consumption figures for the Full Crossbar array implementation

RRAM devices. For PCM technology, [32] obtained an energy consumption of 56 nJ (with respect to the number of devices). Concerning STT-MRAM technology, [13] with a 64×64 crossbar array, obtained a power dissipation of 42 μW. In our implementation, the total power is 3.3 μW and the energy consumption is 65.89 fJ. The implementation presented here allows us to have a consumption significantly lower than what is done in the literature, with a percentage of error remaining similar to other studies [10].

All these studies allow us to consider the viability of using the SOT memory to implement BNN. Indeed, this implementation shows robustness to process variations with a very low percentage of error, 10% on average. More importantly, this memory allows us to circumvent the bottleneck of the STT memory in the conventional crossbar implementation by exploiting its very high resistance [13].

V. Conclusion

In this paper, we present for the first time the implementation of a BNN with SOT-MRAM technology. We illustrate that this implementation is very energy efficient with a low error rate. For this purpose, we present a binary cell design based on a 4T-2R structure. An evaluation of the energy consumption is performed on the crossbar and the periphery. To show the robustness of the approach, several Monte Carlo simulations have been performed. Nevertheless, the implementation of the crossbar is not complete, the goal here was to evaluate the errors at the circuit level for a future more complete implementation. This will lead to a complete simulation and design flow for evaluation of application.

References

[1] Y. LeCun, Y. Bengio *et al.*, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, no. 10, p. 1995, 1995.

[2] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, Nov. 2012, conference Name: IEEE Signal Processing Magazine.

[3] S. Yu, "Neuro-inspired computing with emerging nonvolatile memorys," *Proceedings of the IEEE*, vol. 106, no. 2, pp. 260–285, Feb. 2018, conference Name: Proceedings of the IEEE.

[4] C. Zhu, S. Han, H. Mao, and W. J. Dally, "Trained Ternary Quantization," *arXiv:1612.01064 [cs]*, Feb. 2017, arXiv: 1612.01064.

[5] F. Li, B. Zhang, and B. Liu, "Ternary Weight Networks," *arXiv:1605.04711 [cs]*, Nov. 2016, arXiv: 1605.04711. [Online]. Available: http://arxiv.org/abs/1605.04711

[6] D. Zhang, J. Yang, D. Ye, and G. Hua, "LQ-Nets: Learned Quantization for Highly Accurate and Compact Deep Neural Networks," *arXiv:1807.10029 [cs]*, Jul. 2018, arXiv: 1807.10029.

[7] M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1," *arXiv preprint arXiv:1602.02830*, 2016.

[8] I. Chakraborty, M. Ali, A. Ankit, S. Jain, S. Roy, S. Sridharan, A. Agrawal, A. Raghunathan, and K. Roy, "Resistive Crossbars as Approximate Hardware Building Blocks for Machine Learning: Opportunities and Challenges," *Proceedings of the IEEE*, vol. 108, no. 12, pp. 2276–2310, Dec. 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9141337/

[9] M. Bocquet, T. Hirztlin, J.-O. Klein, E. Nowak, E. Vianello, J.-M. Portal, and D. Querlioz, "In-memory and error-immune differential rram implementation of binarized deep neural networks," in *2018 IEEE International Electron Devices Meeting (IEDM)*. IEEE, 2018, pp. 20–6.

[10] M. Ezzadeen, A. Majumdar, M. Bocquet, B. Giraud, J.-P. Noël, F. Andrieu, D. Querlioz, and J.-M. Portal, "Low-overhead implementation of binarized neural networks employing robust 2t2r resistive ram bridges," in *ESSCIRC 2021-IEEE 47th European Solid State Circuits Conference (ESSCIRC)*. IEEE, 2021, pp. 83–86.

[11] X. Sun, S. Yin, X. Peng, R. Liu, J.-s. Seo, and S. Yu, "Xnor-rram: A scalable and parallel resistive synaptic architecture for binary neural networks," in *2018 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2018, pp. 1423–1428.

[12] E. Giacomin, T. Greenberg-Toledo, S. Kvatinsky, and P.-E. Gaillardon, "A robust digital rram-based convolutional block for low-power image processing and learning applications," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 2, pp. 643–654, 2018.

[13] S. Jung, H. Lee, S. Myung, H. Kim, S. K. Yoon, S.-W. Kwon, Y. Ju, M. Kim, W. Yi, S. Han, B. Kwon, B. Seo, K. Lee, G.-H. Koh, K. Lee, Y. Song, C. Choi, D. Ham, and S. J. Kim, "A crossbar array of magnetoresistive memory devices for in-memory computing," *Nature*, vol. 601, no. 7892, pp. 211–216, Jan. 2022. [Online]. Available: https://www.nature.com/articles/s41586-021-04196-6

[14] Z. Wang, H. Wu, G. W. Burr, C. S. Hwang, K. L. Wang, Q. Xia, and J. J. Yang, "Resistive switching materials for information processing," *Nature Reviews Materials*, vol. 5, no. 3, pp. 173–195, 2020.

[15] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan, "Review of deep learning: Concepts, cnn architectures, challenges, applications, future directions," *Journal of big Data*, vol. 8, no. 1, pp. 1–74, 2021.

[16] T. Simons and D.-J. Lee, "A review of binarized neural networks," *Electronics*, vol. 8, no. 6, p. 661, 2019.

[17] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "Xnor-net: Imagenet classification using binary convolutional neural networks," in *European conference on computer vision*. Springer, 2016, pp. 525–542.

[18] C. Yuan and S. S. Agaian, "A comprehensive review of binary neural network," *CoRR*, vol. abs/2110.06804, 2021. [Online]. Available: https://arxiv.org/abs/2110.06804

[19] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *CoRR*, vol. abs/1502.03167, 2015. [Online]. Available: http://arxiv.org/abs/1502.03167

[20] B. Dieny, I. L. Prejbeanu, K. Garello, P. Gambardella, P. Freitas, R. Lehndorff, W. Raberg, U. Ebels, S. O. Demokritov, J. Akerman, A. Deac, P. Pirro, C. Adelmann, A. Anane, A. V. Chumak, A. Hirohata, S. Mangin, S. O. Valenzuela, M. C. Onbaşlı, M. d'Aquino, G. Prenat, G. Finocchio, L. Lopez-Diaz, R. Chantrell, O. Chubykalo-Fesenko, and P. Bortolotti, "Opportunities and challenges for spintronics in the microelectronics industry," *Nature Electronics*, vol. 3, no. 8, Aug. 2020. [Online]. Available: https://www.nature.com/articles/s41928-020-0461-5

[21] Q. Dong, Z. Wang, J. Lim, Y. Zhang, M. E. Sinangil, Y.-C. Shih, Y.-D. Chih, J. Chang, D. Blaauw, and D. Sylvester, "A 1-mb 28-nm 1t1mtj STT-MRAM with single-cap offset-cancelled sense amplifier and in situ self-write-termination," vol. 54, no. 1, pp. 231–239, conference Name: IEEE Journal of Solid-State Circuits.

[22] K. Lee, K. Yamane, S. Noh, V. B. Naik, H. Yang, S. H. Jang, J. Kwon, B. Behin-Aein, R. Chao, J. H. Lim, S. K, K. W. Gan, D. Zeng, N. Thiyagarajah, L. C. Goh, B. Liu, E. H. Toh, B. Jung, T. L. Wee, T. Ling, T. H. Chan, N. L. Chung, J. W. Ting, S. Lakshmipathi, J. S. Son, J. Hwang, L. Zhang, R. Low, R. Krishnan, T. Kitamura, Y. S. You, C. S. Seet, H. Cong, D. Shum, J. Wong, S. T. Woo, J. Lam, E. Quek, A. See, and S. Y. Siah, "22-nm FD-SOI embedded MRAM with full solder reflow compatibility and enhanced magnetic immunity," in *2018 IEEE Symposium on VLSI Technology*, pp. 183–184, ISSN: 2158-9682.

[23] J. Doevenspeck, K. Garello, B. Verhoef, R. Degraeve, S. Van Beek, D. Crotti, F. Yasin, S. Couet, G. Jayakumar, I. A. Papistas, P. Debacker, R. Lauwereins, W. Dehaene, G. S. Kar, S. Cosemans, A. Mallik, and D. Verkest, "SOT-MRAM Based Analog in-Memory Computing for DNN Inference," in *2020 IEEE Symposium on VLSI Technology*, Jun. 2020, pp. 1–2, iSSN: 2158-9682.

[24] T. Hirtzlin, M. Bocquet, B. Penkovsky, J.-O. Klein, E. Nowak, E. Vianello, J.-M. Portal, and D. Querlioz, "Digital Biologically Plausible Implementation of Binarized Neural Networks With Differential Hafnium Oxide Resistive Memory Arrays," *Frontiers in Neuroscience*, vol. 13, p. 1383, Jan. 2020. [Online]. Available: https://www.frontiersin.org/article/10.3389/fnins.2019.01383/full

[25] M. Natsui, T. Chiba, and T. Hanyu, "Design of mtj-based nonvolatile logic gates for quantized neural networks," *Microelectronics journal*, vol. 82, pp. 13–21, 2018.

[26] X. Sun, X. Peng, P.-Y. Chen, R. Liu, J.-s. Seo, and S. Yu, "Fully parallel RRAM synaptic array for implementing binary neural network with (+1, 1) weights and (+1, 0) neurons," Jan. 2018, pp. 574–579, iSSN: 2153-697X.

[27] T. Kobayashi, K. Nogami, T. Shirotori, Y. Fujimoto, and O. Watanabe, "A current-mode latch sense amplifier and a static power saving input buffer for low-power architecture," in *1992 Symposium on VLSI Circuits Digest of Technical Papers*. Seattle, WA, USA: IEEE, 1992, pp. 28–29. [Online]. Available: http://ieeexplore.ieee.org/document/229252/

[28] B. Li, L. Xia, P. Gu, Y. Wang, and H. Yang, "Merging the interface: Power, area and accuracy co-optimization for rram crossbar-based mixed-signal computing system," in *Proceedings of the 52nd annual design automation conference*, 2015, pp. 1–6.

[29] T. Gokmen, M. Onen, and W. Haensch, "Training deep convolutional neural networks with resistive cross-point devices," *Frontiers in Neuroscience*, vol. 11, 2017. [Online]. Available: https://www.frontiersin.org/article/10.3389/fnins.2017.00538

[30] W. Haensch, T. Gokmen, and R. Puri, "The next generation of deep learning hardware: Analog computing," *Proceedings of the IEEE*, vol. 107, no. 1, pp. 108–122, 2018.

[31] S. singh Chouhan and K. Halonen, "A 352nw, 30ppm/°c all mos nano ampere current reference circuit," *Microelectronics Journal*, vol. 69, pp. 45–52, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0026269217304378

[32] S. Ambrogio, P. Narayanan, H. Tsai, R. M. Shelby, I. Boybat, C. di Nolfo, S. Sidler, M. Giordano, M. Bodini, N. C. P. Farinha, B. Killeen, C. Cheng, Y. Jaoudi, and G. W. Burr, "Equivalent-accuracy accelerated neural-network training using analogue memory," *Nature*, vol. 558, no. 7708, pp. 60–67, Jun. 2018. [Online]. Available: https://www.nature.com/articles/s41586-018-0180-5