EE746 Project Stage 2
FeFET based Resistive Processing Units

Koustav Jana (170070051), Mihir Kavishwar (17D070004), Prashant Kurrey (17D070057)

November 29, 2021

Link to presentation and code: Click Here

1 Introduction

Resistive Processing Units (RPUs), which are based on the principle of in-memory computing, offer a promising solution for energy-efficient and fast training of Deep Neural Networks (DNNs). RPUs can perform tasks such as Matrix-Vector-Multiplication (MVM) and vector-vector outer product in a highly parallel fashion ($\mathcal{O}(1)$ time complexity). An RPU is made up of resistive crossbars, which have Non-Volatile Memory (NVM) element at every intersection to store a weight. Previous works have used varies NVMs such as Resistive RAM, Phase Change Memory and Charge Trap Flash. In this project we want to explore if FeRAM can be potentially used as a NVM in RPUs.

2 Theory

In this work, we use Ferroelectric FETs as the NVM element in the RPU. FeFETs have ferroelectric HfO_2 in the gate-stack whose polarization switching helps achieve conductance modulation. These are particularly attractive for RPUs because of their compact 1T structure, fast and low voltage switching and CMOS compatibility.

The ferroelectric materials apart from their high and low-V_T states can take intermediate values of V_T due to partial switching of poalrization domains. For large area FETs it is possible to achieve a continuum of intermediate V_T states, making them particularly useful for synaptic weights in RPUs. Also the equivalence between accumulative and one-shot switching shows that the FE nuclei generation does not easily decay with time, thus can achieve desired V_T switching using pulse-trains similar to that of a single pulse. This is highly convenient because on-chip generation of single pulses with varying pulse-width is difficult.

3 Methods

We implemented an RPU in MATLAB for an image classification task on the MNIST dataset. The NN model was same as that in [5] - a fully connected network with two hidden layers consisting of 256 and 128 neurons respectively. ReLU activation was used in the hidden layers while softmax activation was used for the output layer. The model hyperparameters and simulation results are discussed on the next page. We ran the simulations with two different device datasets - CTF and FeFET.

Hyperparameter	**Value**
Update step size (α)	0.01
Weight scaling factor (k)	600α
Initial device conductance ($g_{1,0}, g_{2,0}$)	$\sim \mathcal{U}(0,1)$
Pulse train length PL	10
Input scaling factor C	$\frac{\alpha}{PL.\Delta^{+}g(c).k}$

Table 1: Hyperparameters

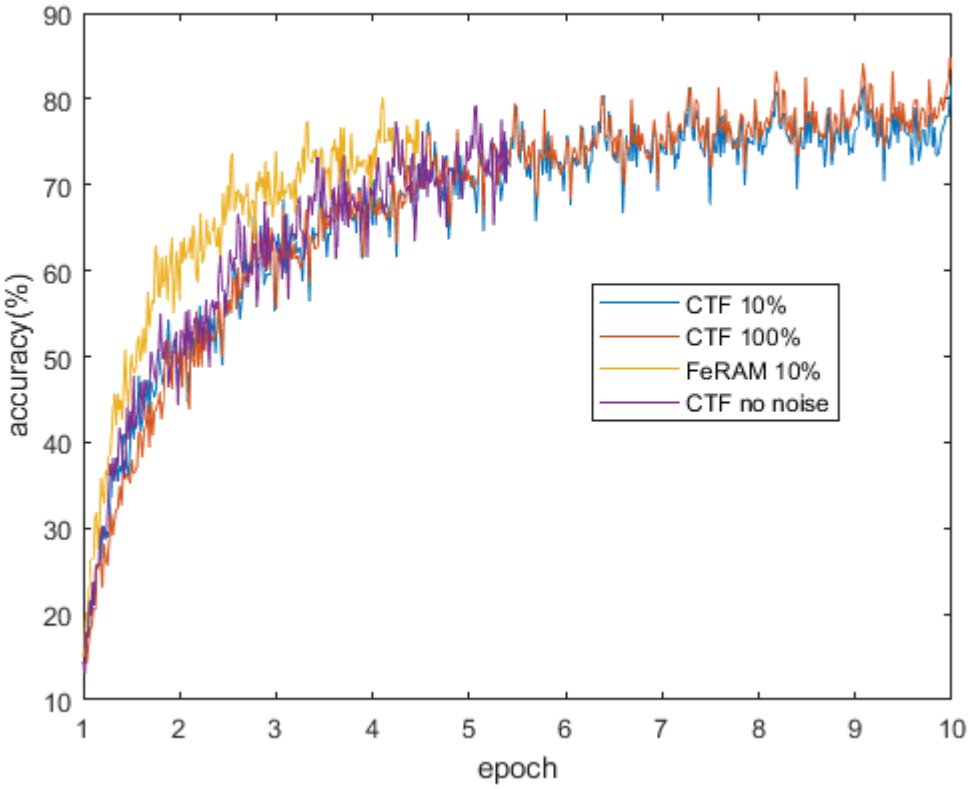

Figure 1: Training Accuracy

4 Results and Discussions

Figure 1 shows the dependence of training accuracy on the number of epochs for the cases of CTF(baseline-no noise, $\sigma/\mu = 10\%$, $\sigma/\mu = 100\%$) and FeFET($\sigma/\mu = 10\%$). It is observed that the training performance is insensitive to noise even on increasing its strength, with not much difference between results of FeFET and CTF. The testing accuracy is around 75% and the poor training accuracy is due to selection of hyperparamter. The accuracy can be increased by changing the Neural net model and the parameter.

5 Future Work

Because of the randomness associated with the switching of multiple domains within FeFET, the device has an intrinsic noise. If we can somehow exploit this inherent stochasticity to achieve the required stochastic multiplication, our RPU would not need any external circuitry for stochastic pulse-train generation.

However, to achieve a stochastic multiplication of δ^i and x^i we need to control the stochasticity of the FeFETs based on δ^i and x^i. This is not easy for a large-area FeFET where the noise is gaussian. Howver, if we consider small-area FETs, we obtain an abrupt switching between the high and low-V_T states which is intrinsically stochastic. Thus the V_T switching in this case is Bernoulli with a switching probability. For a given pulse amplitude, the switching probability between the high and low-V_T states can be controlled via the pulse-width.

This provides the motivation to use the following architecture (Fig.2) of ultrascaled FeFETs at each crosspoint. Here the 2 FeFETs in series have there input pulse-widths set such that the whole branch is ON

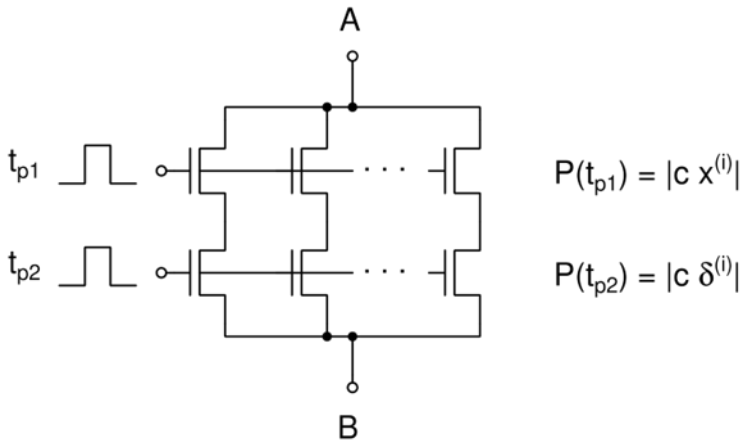

Figure 2: Proposal to harness inherent stochasticity of ultrascaled FeFETs.

with a probability proportional to $\delta^i.x^i$. For a large enough N we can get $\Delta g \propto \delta^i.x^i$

6 References

[1] Zhen Fan, Jingsheng Chen, and John Wang. "Ferroelectric HfO 2 -based materials for next-generation ferroelectric memories". In: *Journal of Advanced Dielectrics* 6 (May 2016), p. 1630003. DOI: 10.1142/S2010135X16300036.

[2] Tayfun Gokmen and Yurii Vlasov. "Acceleration of Deep Neural Network Training with Resistive Cross-Point Devices: Design Considerations". In: *Frontiers in Neuroscience* 10 (2016), p. 333. ISSN: 1662-453X. DOI: 10.3389/fnins.2016.00333. URL: https://www.frontiersin.org/article/10.3389/fnins.2016.00333.

[3] Seyoung Kim et al. "Analog CMOS-based resistive processing unit for deep neural network training". In: *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*. 2017, pp. 422–425. DOI: 10.1109/MWSCAS.2017.8052950.

[4] Halid Mulaosmanovic, Thomas Mikolajick, and Stefan Slesazeck. "Random Number Generation Based on Ferroelectric Switching". In: *IEEE Electron Device Letters* 39.1 (2018), pp. 135–138. DOI: 10.1109/LED.2017.2771818.

[5] Varun Bhatt et al. "Software-Level Accuracy Using Stochastic Computing With Charge-Trap-Flash Based Weight Matrix". In: *2020 International Joint Conference on Neural Networks (IJCNN)*. 2020, pp. 1–8. DOI: 10.1109/IJCNN48605.2020.9206631.

[6] Halid Mulaosmanovic et al. "Investigation of Accumulative Switching in Ferroelectric FETs: Enabling Universal Modeling of the Switching Behavior". In: *IEEE Transactions on Electron Devices* 67.12 (2020), pp. 5804–5809. DOI: 10.1109/TED.2020.3031249.