EE746 Project:
FeFET based Resistive Processing Units

Koustav Jana (170070051)

Mihir Kavishwar (17D070004)

Prashant Kurrey (17D070057)

Outline

- RPU Introduction and Motivation
- CTF based RPU Methodology
- FeFET based RPU Methodology
- MATLAB Simulation Results
- Exploiting inherent stochasticity of FeFETs
- Conclusion and Future Work

RPU: Introduction and Motivation

- Training of large Deep Neural Networks (DNN) is a time consuming and computationally intensive task that demands datacenter-scale computational resources recruited for many days
- Resistive Processing Units can potentially accelerate DNN training by orders of magnitude while using much less power.
- State-of-the-art (SotA) RPU devices can store and update the weight values locally thus minimizing data movement during training and allowing to fully exploit the locality and the parallelism of the training algorithm

RPU: Introduction and Motivation

We can use the RPU for two applicaions:

- **Inference**: Forward pass to calculate output of Neural Network with present weights
- **Training**: Updating the present weights (conductances) as per backpropogation algorithm

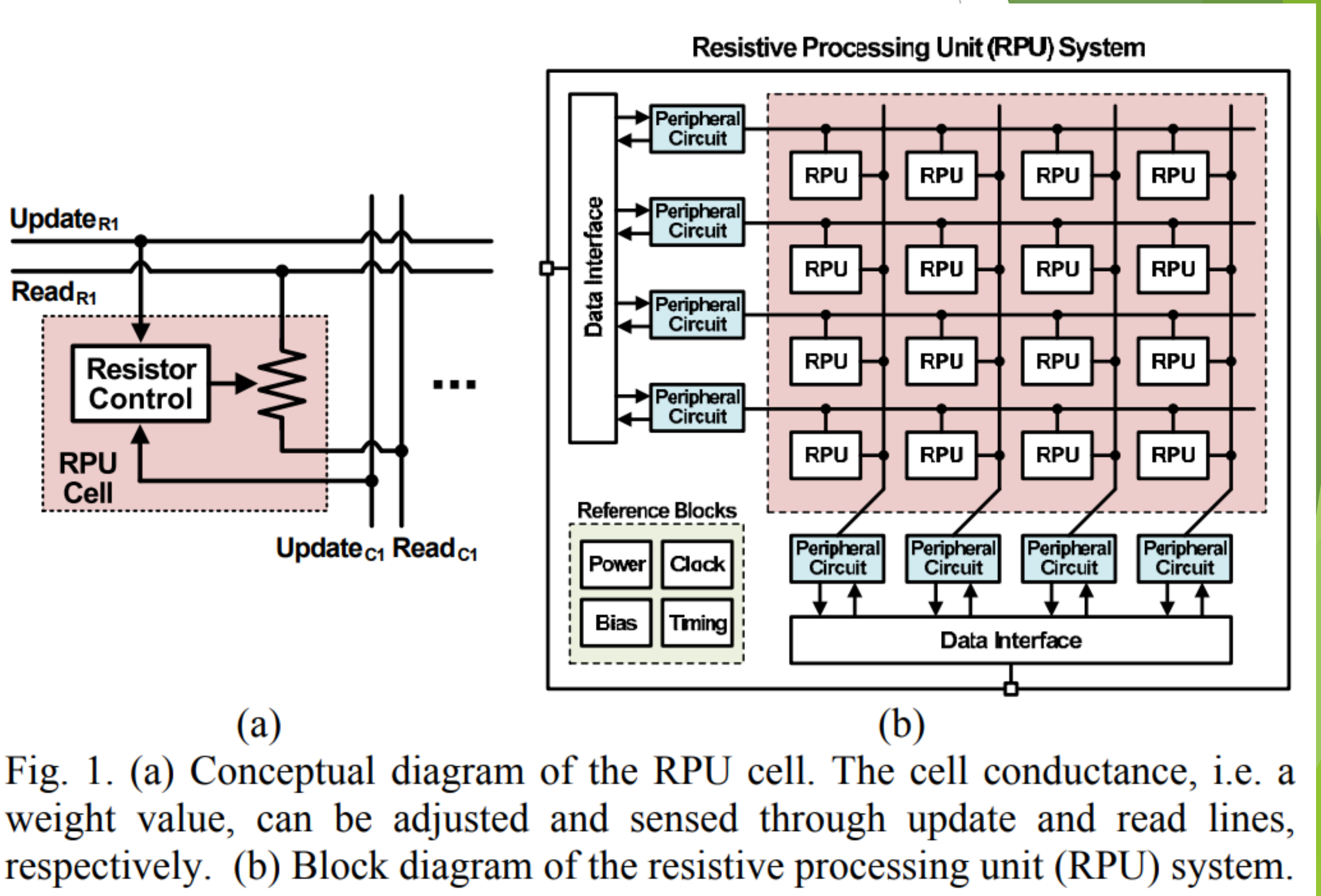


Fig. 1. (a) Conceptual diagram of the RPU cell. The cell conductance, i.e. a weight value, can be adjusted and sensed through update and read lines, respectively. (b) Block diagram of the resistive processing unit (RPU) system.

Kim, Seyoung et al. “Analog CMOS-based resistive processing unit for deep neural network training.” *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)* (2017): 422-425.

CTF based RPU: Methodology

- We implemented a Charge Trap Flash (CTF) based RPU in MATLAB for an image classification task on the MNIST dataset, by refering to the following paper:

  V. Bhatt, S. Shrivastava, T. Chavan and U. Ganguly, "Software-Level Accuracy Using Stochastic Computing With Charge-Trap-Flash Based Weight Matrix," *2020 International Joint Conference on Neural Networks (IJCNN)*, 2020, pp. 1-8, doi: 10.1109/IJCNN48605.2020.9206631.

- In this work, analog multiplication is performed using **stochastic pulse trains** to update weights
- We carried out simulations for three different cases:
  1. No device noise
  2. AWGN with variance = 10% of mean
  3. AWGN with variance = 100% of mean

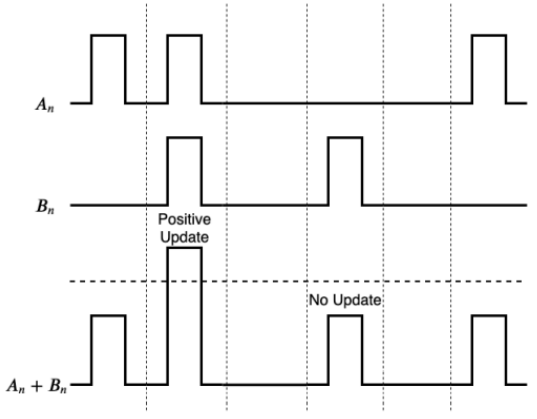
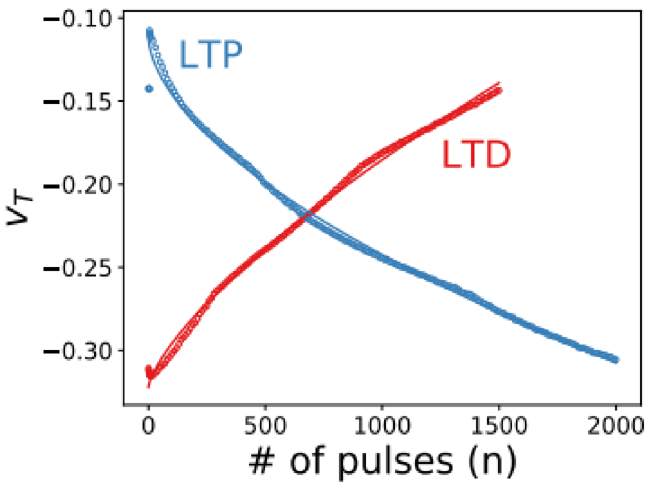


Fig. 1. Analog multiplication using Stochastic pulse trains in RPU: Analog numbers are represented by a stochastic pulse train of identical pulses where the probability of high voltage in trains A_n and B_n is proportional to $x^{(i)}$, $\delta^{(i)}$ respectively. Updates occur at the coincidences, i.e., AND(A_n, B_n), which have a probability proportional to $x^{(i)}\delta^{(i)}$. Polarity is reversed for negative updates.

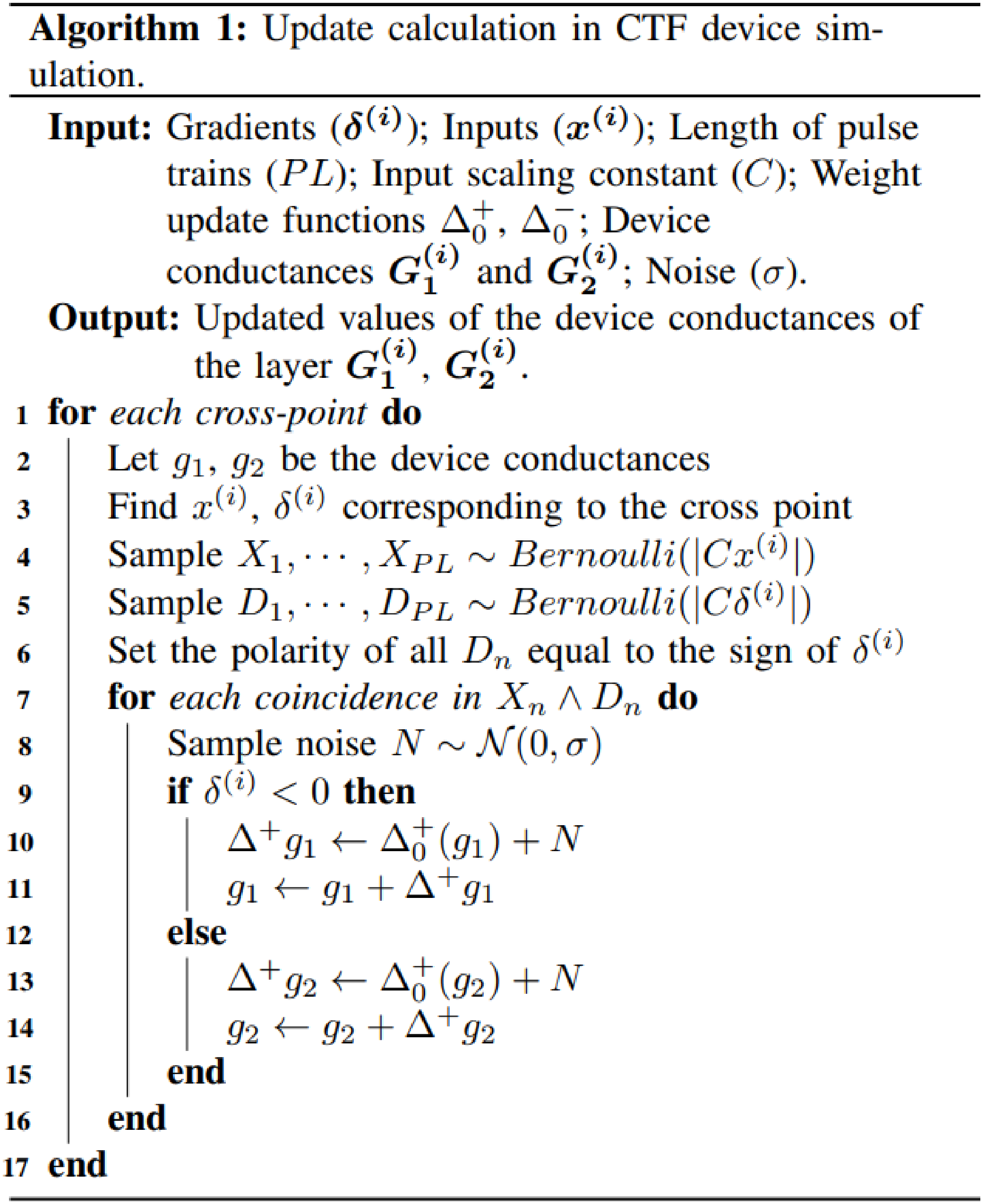


CTF based RPU: Methodology

- We used the same algorithm and parameters as those given in the paper
- Define $\Delta^{+}_{0}(g)$ as the positive change in V_T when V_T= g (using LTD data) and $\Delta^{-}_{0}(g)$ as the negative change in V_T when VT = g (using LTP data)
- The results of the curve fit gives the following equations

$$\Delta_0^+(g) = 4.50(g+0.32)^{-0.39} \times 10^{-5}$$

$$\Delta_0^-(g) = -1.74(-g-0.11)^{-0.72} \times 10^{-5}$$



Algorithm 1: Update calculation in CTF device simulation.

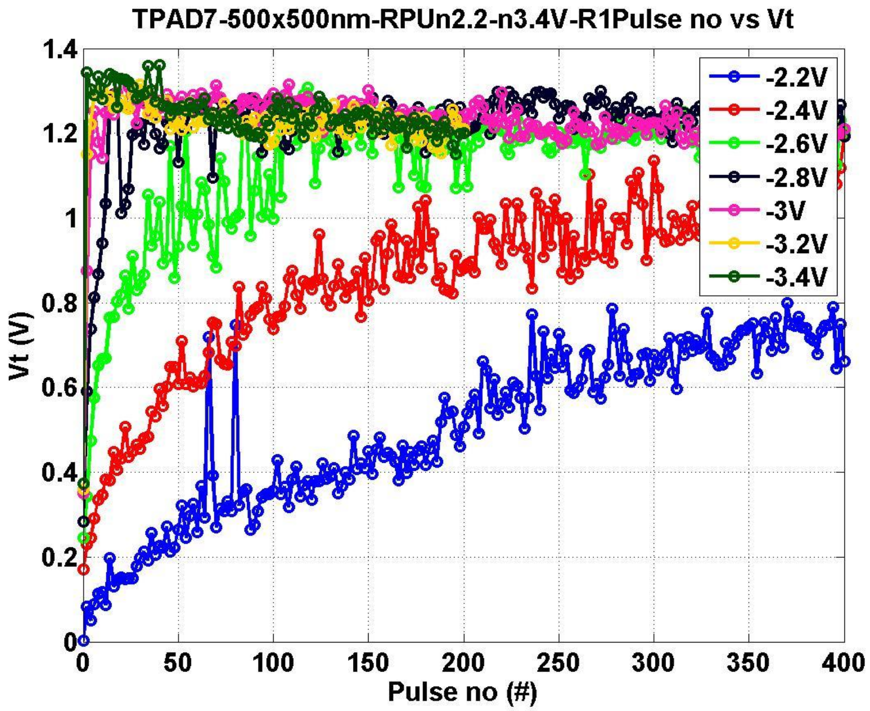
Input: Gradients ($\boldsymbol{\delta}^{(i)}$); Inputs ($\boldsymbol{x}^{(i)}$); Length of pulse trains (PL); Input scaling constant (C); Weight update functions Δ_0^+, Δ_0^-; Device conductances $\boldsymbol{G}_1^{(i)}$ and $\boldsymbol{G}_2^{(i)}$; Noise (σ).

Output: Updated values of the device conductances of the layer $\boldsymbol{G}_1^{(i)}$, $\boldsymbol{G}_2^{(i)}$.

```
for each cross-point do
    Let $g_1$, $g_2$ be the device conductances
    Find $x^{(i)}$, $\delta^{(i)}$ corresponding to the cross point
    Sample $X_1, \cdots, X_{PL} \sim Bernoulli(|Cx^{(i)}|)$
    Sample $D_1, \cdots, D_{PL} \sim Bernoulli(|C\delta^{(i)}|)$
    Set the polarity of all $D_n$ equal to the sign of $\delta^{(i)}$
    for each coincidence in $X_n \wedge D_n$ do
        Sample noise $N \sim \mathcal{N}(0, \sigma)$
        if $\delta^{(i)} < 0$ then
            $\Delta^+ g_1 \leftarrow \Delta_0^+(g_1) + N$
            $g_1 \leftarrow g_1 + \Delta^+ g_1$
        else
            $\Delta^+ g_2 \leftarrow \Delta_0^+(g_2) + N$
            $g_2 \leftarrow g_2 + \Delta^+ g_2$
        end
    end
end
```

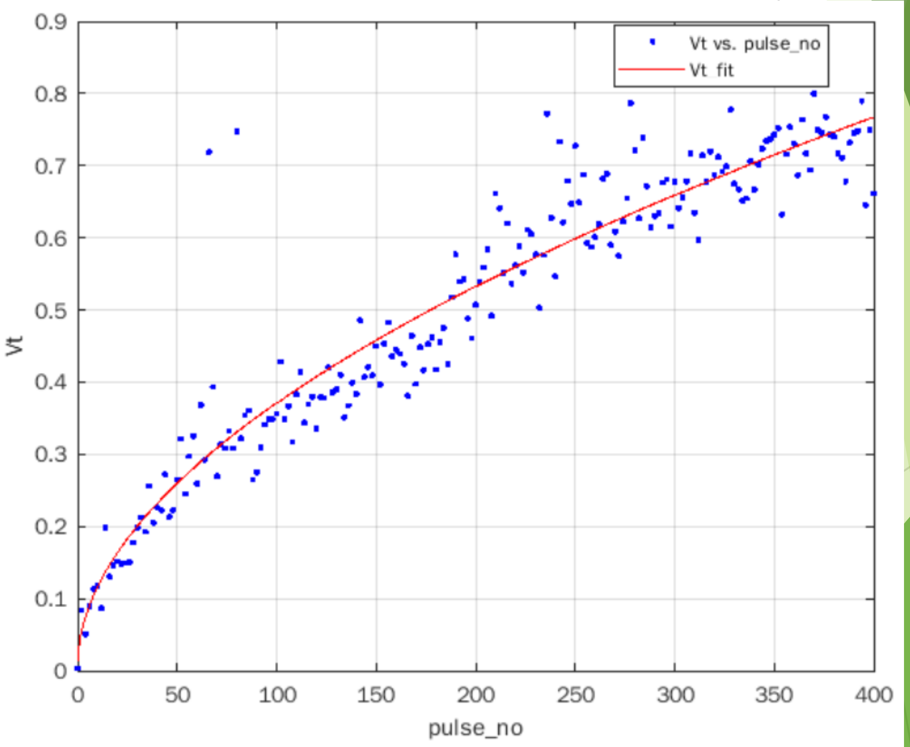
FeFET based RPU: Methodology

- The same methodology was followed but now using experimental data of FeFET device



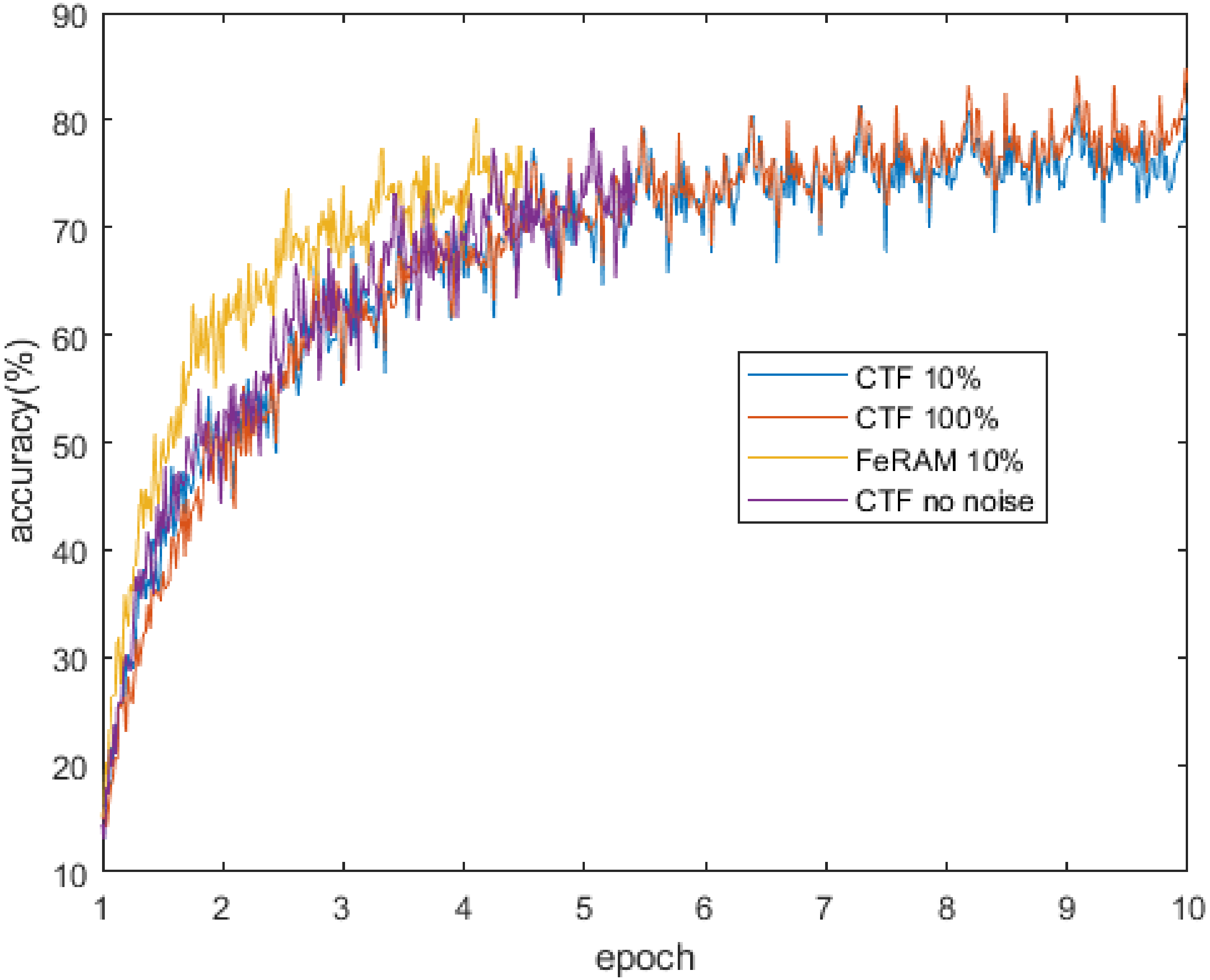
Vt vs Pulse no. data for FeFET device for different pulse amplitudes

$$y = 0.02985*x^{(0.5387)} + 0.01404$$



Best fit curve of Vt vs Pulse no. data for pulse amplitude = -2.2V

MATLAB Simulation Results



Training
accuracy = 80 %

Can we exploit the inherent stochasticity of FeFET devices?

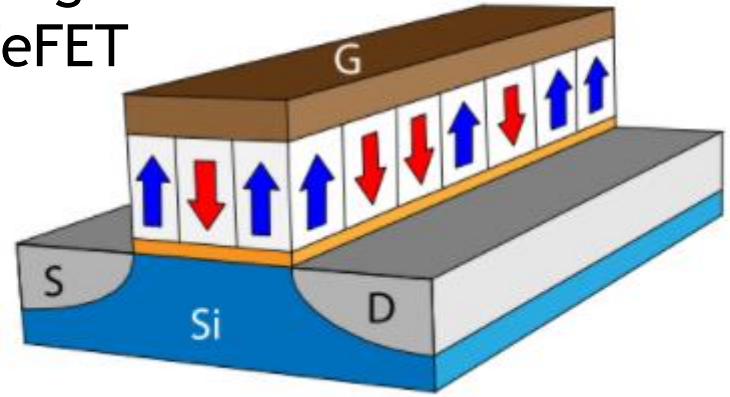
Advantages Offered

- No external circuitry needed for stochastic pulse-train generation, deterministic pulse train will do
- Device noise not a nuisance, but RESOURCE!!!

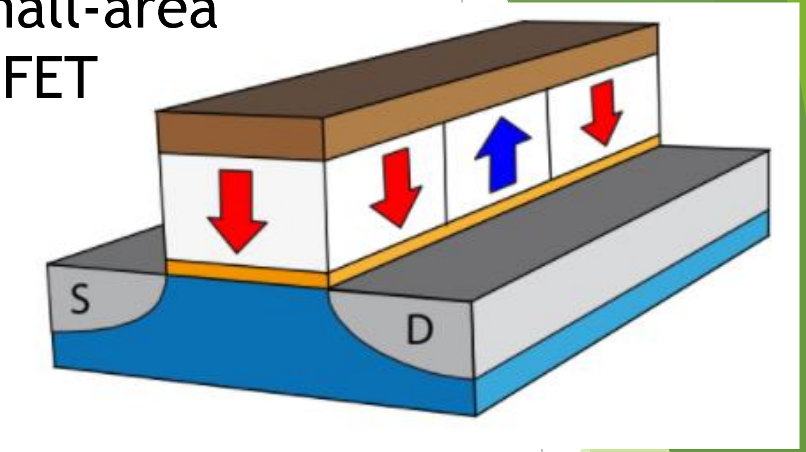
Main Idea: To enable a stochastic multiplication of $x^{(i)}$ and $\delta^{(i)}$, need to have two independent stochastic events with probabilities proportional to $x^{(i)}$ and $\delta^{(i)}$ respectively. Then the probability of the two events happening simultaneously will be proportional to the product $x^{(i)}.\delta^{(i)}$

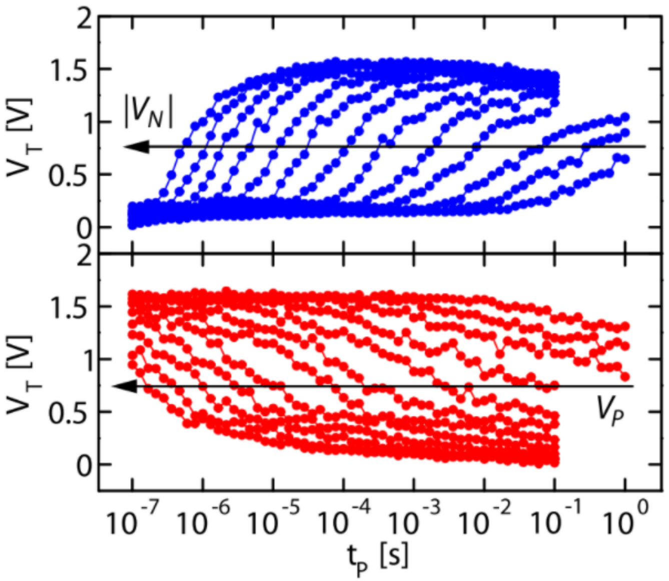
Large vs Small-Area FeFETs

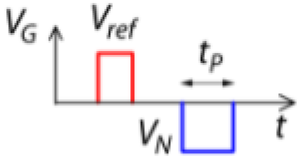
Large-area FeFET

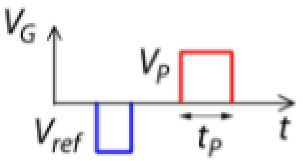


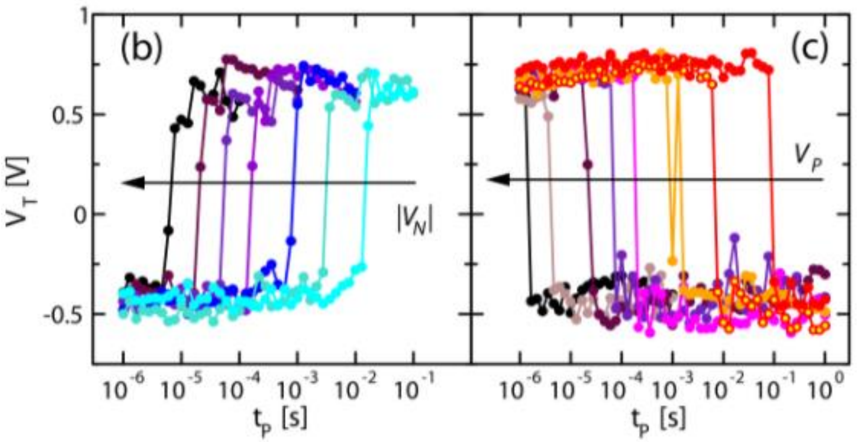
Small-area FeFET







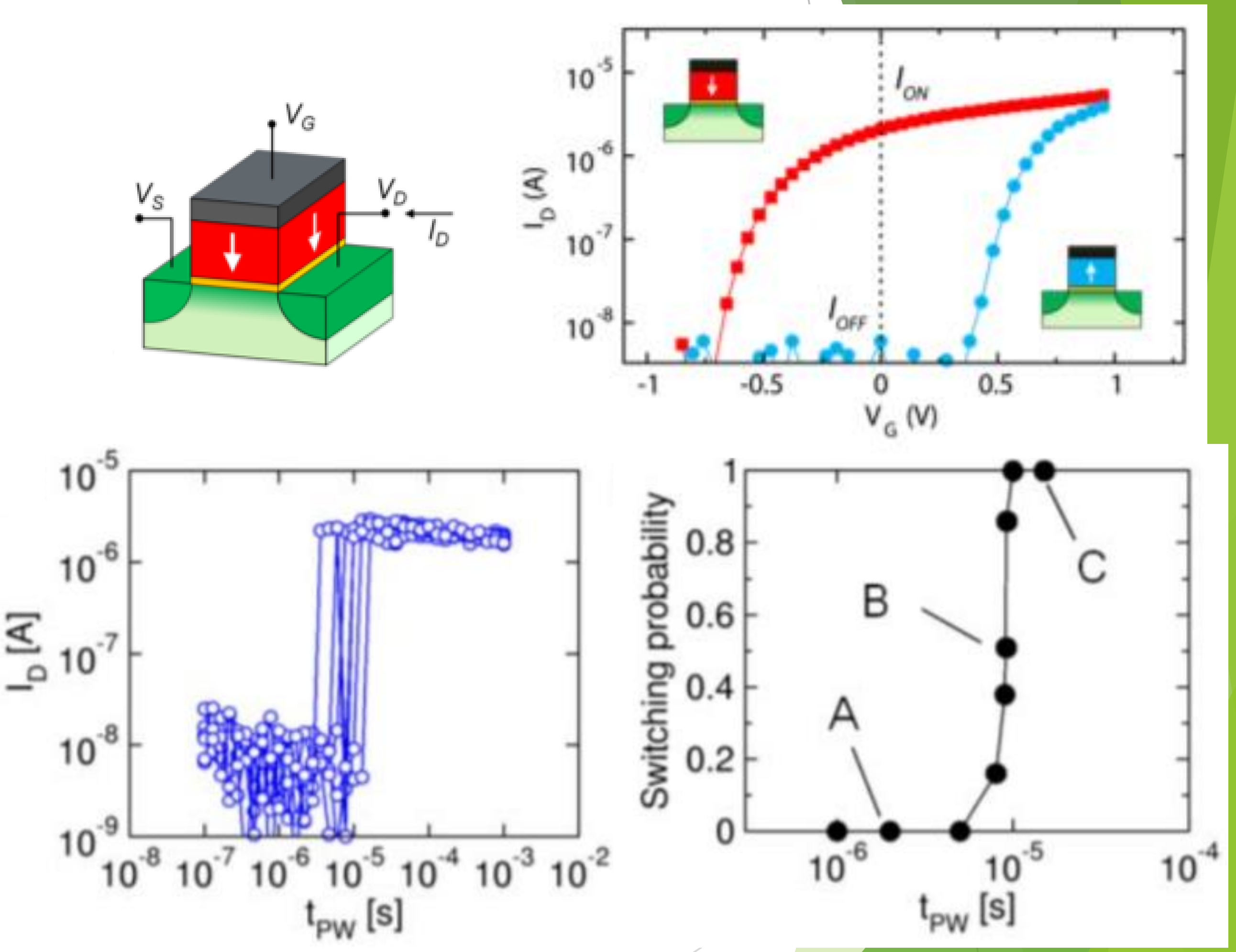




H. Mulaosmanovic et al., "Investigation of Accumulative Switching in Ferroelectric FETs: Enabling Universal Modeling of the Switching Behavior," in IEEE Transactions on Electron Devices, vol. 67, no. 12, pp. 5804-5809, Dec. 2020, doi: 10.1109/TED.2020.3031249.

Single domain ultrascaled FeFETs

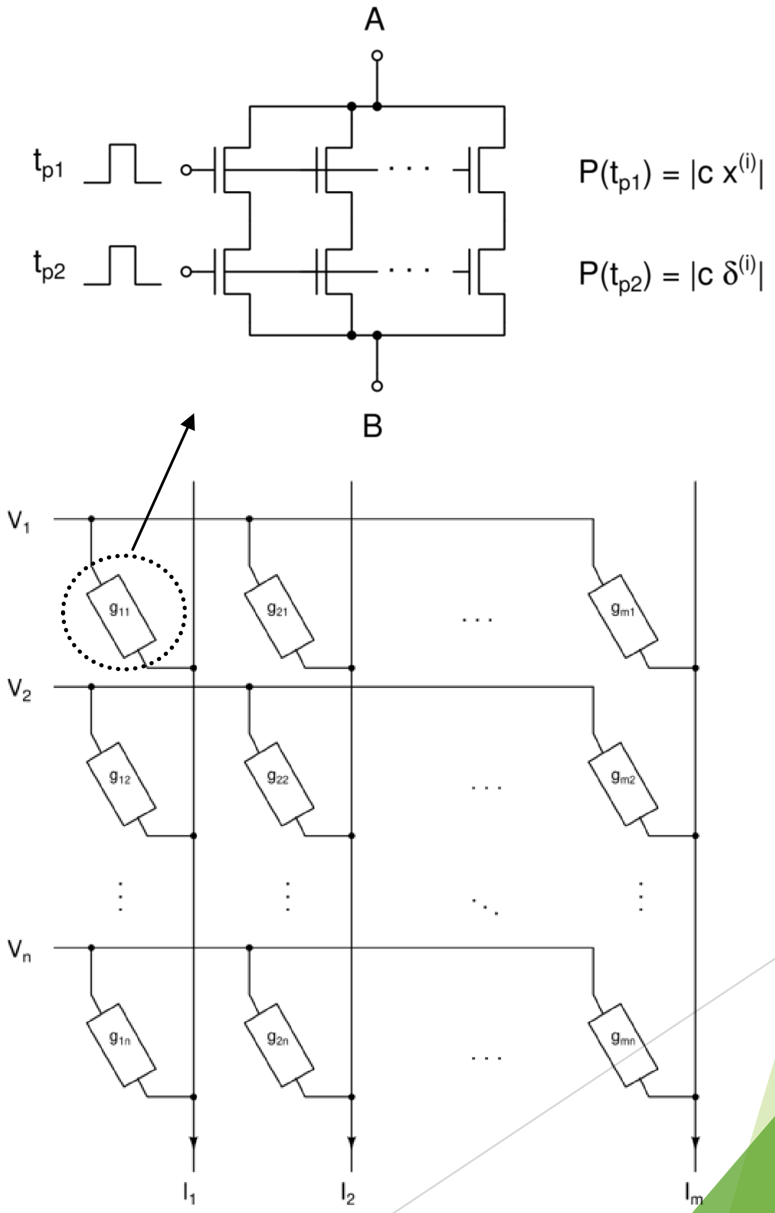
- In presence of several domains it is possible to get a continuum of values of V_T, whereas for a single domain we only have 2 states of V_T
- On application of a gate voltage pulse, a switching maybe possible with the probability dependent on the pulse width and the pulse amplitude
- Possible to control the inherent stochasticity of single-domain FeFET through pulse width/amplitude



H. Mulaosmanovic, T. Mikolajick and S. Slesazeck, "Random Number Generation Based on Ferroelectric Switching," in IEEE Electron Device Letters, vol. 39, no. 1, pp. 135-138, Jan. 2018, doi: 10.1109/LED.2017.2771818.

Architecture of each cross-point

- Apply t_{p1} and t_{p2} such that the switching probabilities of the FeFETs in series are $|Cx^{(i)}|$ and $|C\delta^{(i)}|$ respectively
- Each branch conducts with a probability $\propto x^{(i)}.\delta^{(i)}$
- For a large enough N, we can get $\Delta g \propto x^{(i)}.\delta^{(i)}$ for the entire system
- Idea similar to that of coincidence between stochastic pulse trains, here the coincidence between stochastic switching of two FeFETs in series does the job



Feasibility and Limitations

- Only discrete set of values possible for the synapse conductance, thus compromised precision of the crossbar weights (ranging from 0 to $\mathrm{N}.\, g_{ON}$ in steps of g_{ON}, where g_{ON} is conductance when both the FeFETs are ON)
- Requires 4N FeFETs per cross-point as opposed to 2 for the case using stochastic pulse trains, although the single-domain FeFETs are much smaller in size than the multi-domain ones
- The relation between pulse-width and switching probability needs to be carefully analyzed and modelled (From the data in literature, transition seems quite abrupt)

Conclusion and Future Work

- CTF and FeFET based RPUs were simulated in MATLAB for MNIST classification, and the the training accuracy was analyzed
- To improve the training accuracy, hyperparameters need to be tuned appropriately
- An idea was proposed to exploit inherent stochasticity of FeFET devices in a RPU system, further investigation is required to check feasibility