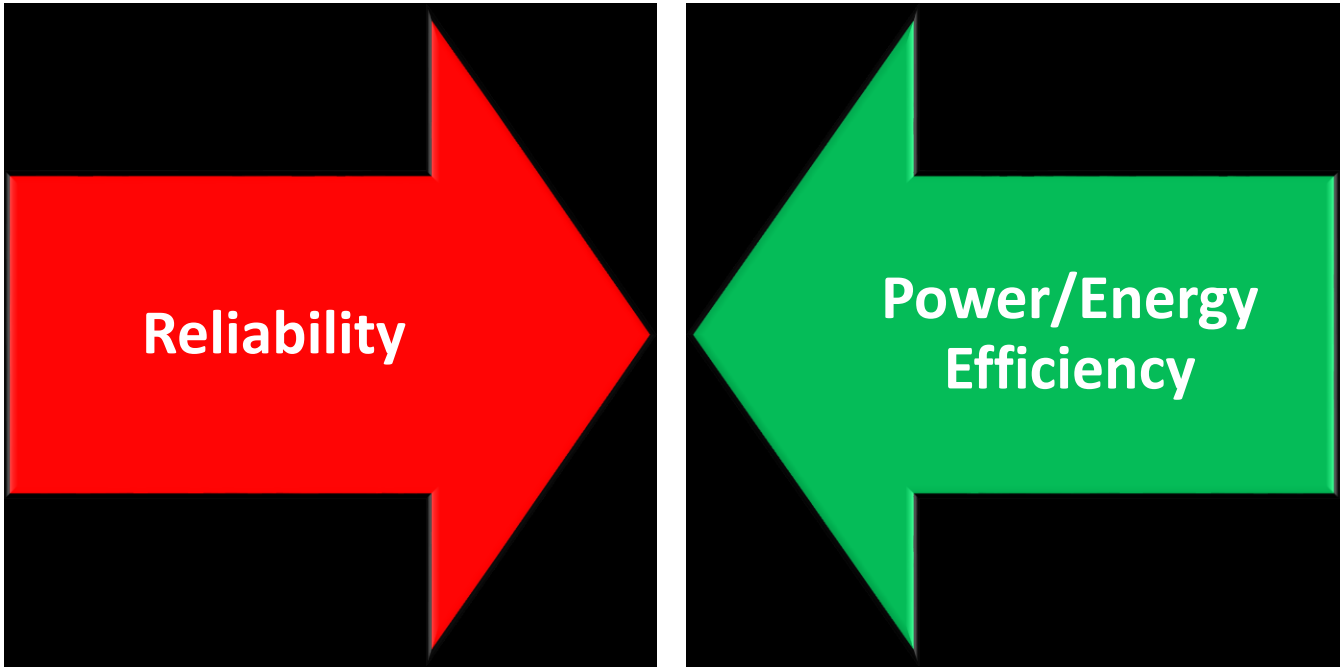
On the Resilience of Deep Learning for reduced-voltage FPGAs

Kamyar Givaki*, Behzad Salami**, Reza Hojabr*, S. M. Reza Tayaranian*, Ahmad Khonsari*, Dara Rahmati***, Saeid Gorgin****, Adrian Cristal**, Osman S. Unsal**.

* University of Tehran, Tehran, Iran,
** Barcelona Supercomptuing Center (BSC),Barcelona, Spain,
*** Institute for Research in Fundamental Sciences (IPM), Tehran, Iran,
**** Iranian Research Organization for Science and Technology (IROST), Tehran,Iran.

Aggressive Undervolting

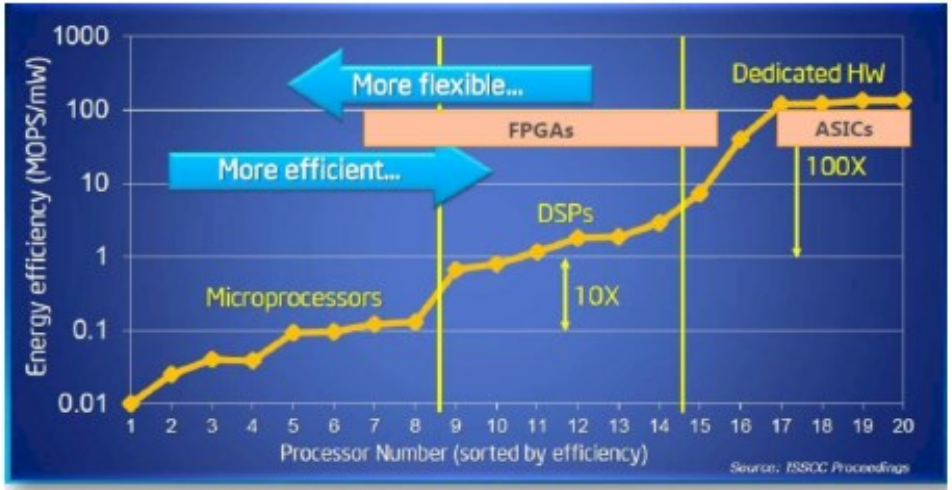
- ***Aggressive undervolting-*** Underscaling the supply voltage below the nominal and safe level:
  - ***Power/Energy Efficiency***: Reduces dynamic and static power quadratically and linearly, respectively.
  - ***Reliability***: Increases the circuit delay and in turn, causes timing faults.



Undervolting on FPGAs: Motivation

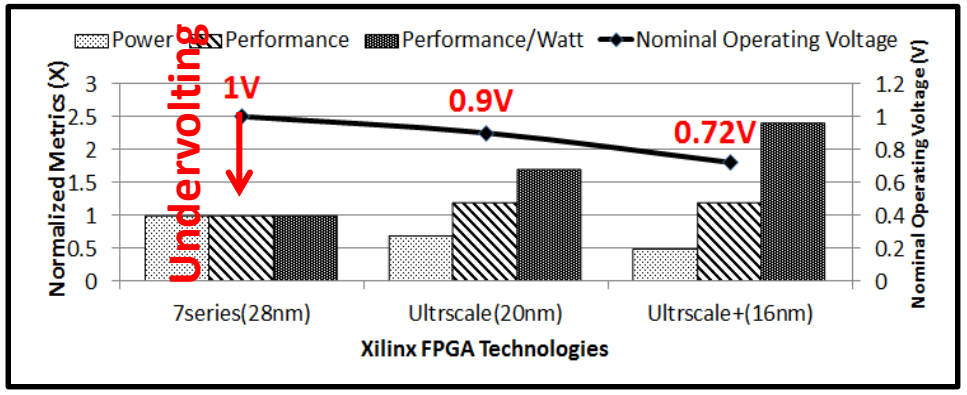
Contribution of FPGAs in large data centers is growing, expected to be in ***30%*** of datacenter servers by 2020 (Top500 news).

- In comparison to ASICs, energy efficiency of FPGAs is a serious concern, *i.e.*, 10X-100X less-efficient.



Source: Bob Broderson, Berkeley Wireless group [Intel/Altera]

- Nominal voltage reduction of FPGAs is naturally applied for different generations.



[Xilinx]

Contributions

- Real Fault Maps
  - ❑ DNN training is inherently robust for undervolting-related faults, evaluated on the fault maps of real FPGA fabrics that are publicly available.

- Synthetic Fault Maps
  - ❑ Fault rate of at least 25% can significantly affect the DNN accuracy.

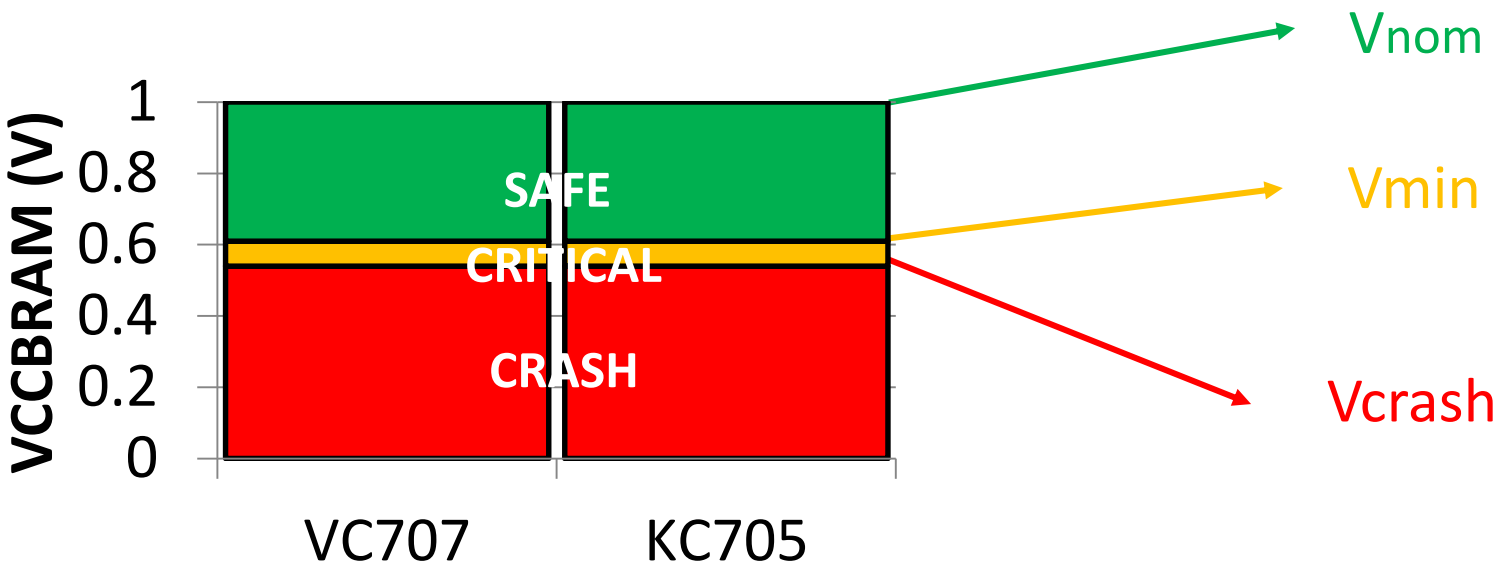
Overall Voltage Behavior

SAFE
- No observable fault
- Voltage Guardband below Vnom

CRITICAL
- Faults manifest
- Below Vmin, min safe voltage

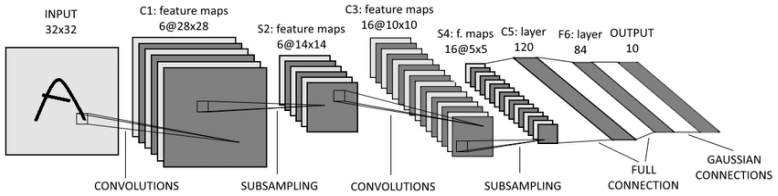
CRASH
- FPGA stops operating below Vcrash, min operating voltage

- ***Voltage guardband***: to ensure the worst-case environmental and process technologies.
- ***Experimental conditions***: At ambient temperature and maximum operating frequency.



Summary of the work

- Simulation of the effect of voltage underscaling related faults on the training phase of NNs.
  - **Datasets:**
    - **LeNet-5 => MNIST**

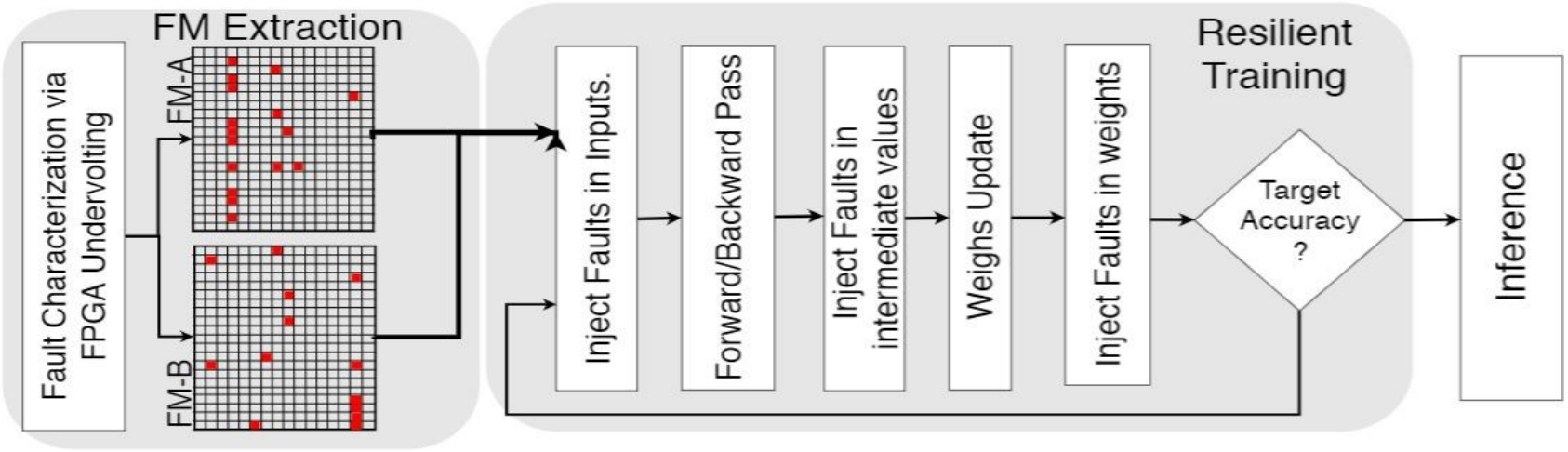


    - **A Customized Network based on LeNet-5 => Cifar-10**

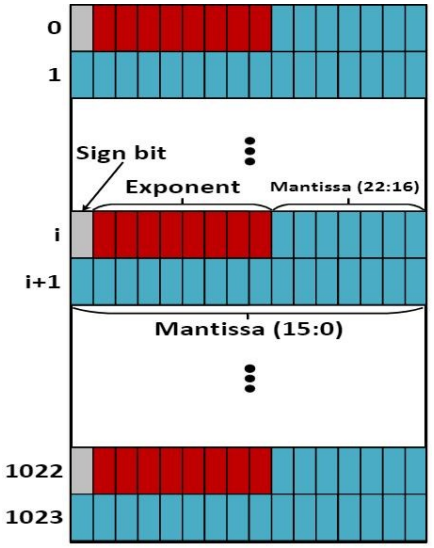
Layer type	Kernel size	Stride	# of Channels	Activation function
Conv	3×3	1	32	Relu or Tanh
Conv	3×3	1	32	Relu or Tanh
Max pooling	2×2	–	–	–
0.3 Dropout	–	–	–	–
Conv	3×3	1	64	Relu or Tanh
Conv	3×3	1	64	Relu or Tanh
Max pooling	2×2	–	–	–
0.4 Dropout	–	–	–	–
FC layer	–	–	–	Sigmoid
FC layer	–	–	–	Softmax

  - **Different activation functions:**
    - Hyperbolic Tangent (**Tanh**)
    - Rectified Linear Unit (**RELU**)

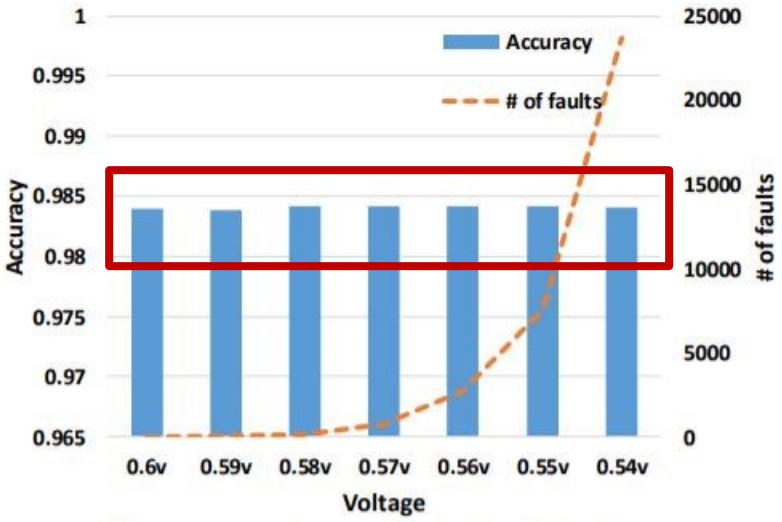
Experimental Methodology



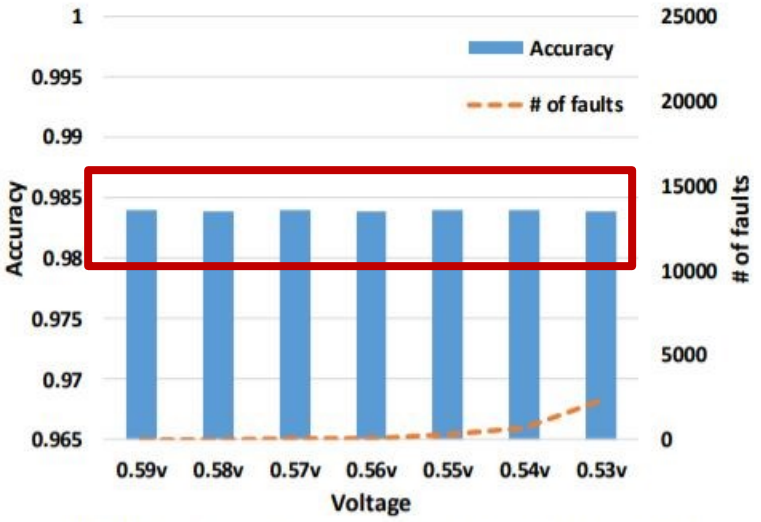
- Training with faults
  - Input image is written into memory.
  - Faults are injected to the input image based on the position of the image in the memory.
  - After each iteration, some intermediate values are generated.
  - Faults are injected to the intermediate values based on their Position in the memory.
  - Update weights.
  - Inject fault in the updated weights based on their position.
- Single-precision floating-point numbers (32-bits) to represent input, weights, and intermediate variables;



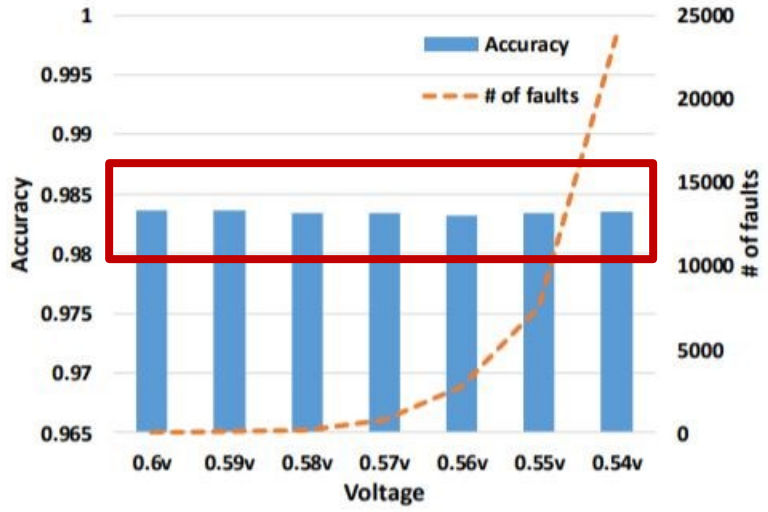
Training accuracy of the LeNet-5 network in the classification of MNIST



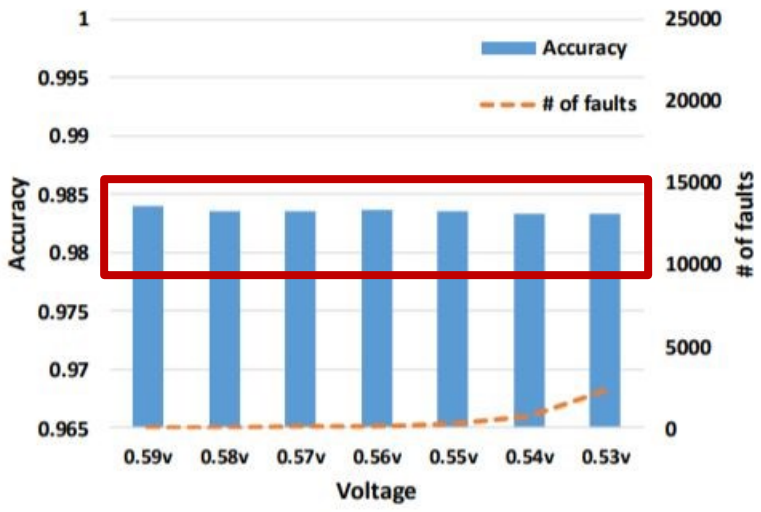
(a) (Activation Function, FPGA)= (RELU,VC707)



(b) (Activation Function, FPGA)= (RELU,KC705)

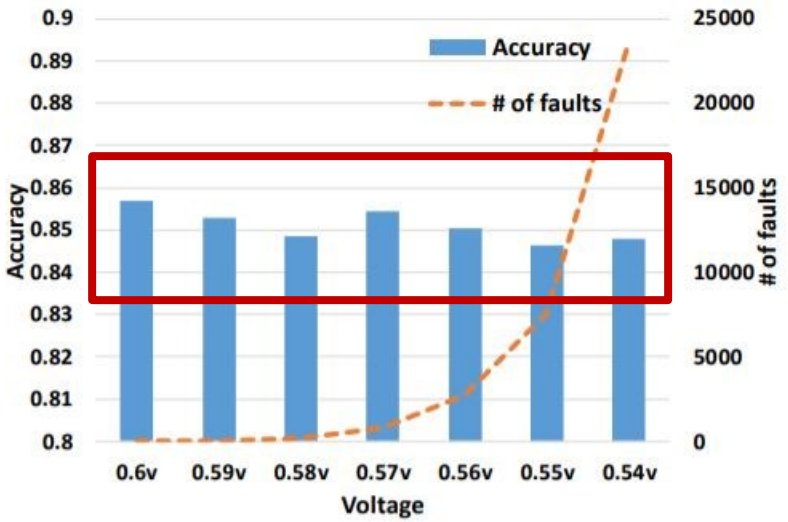


(c) (Activation Function, FPGA)= (Tanh,VC707)

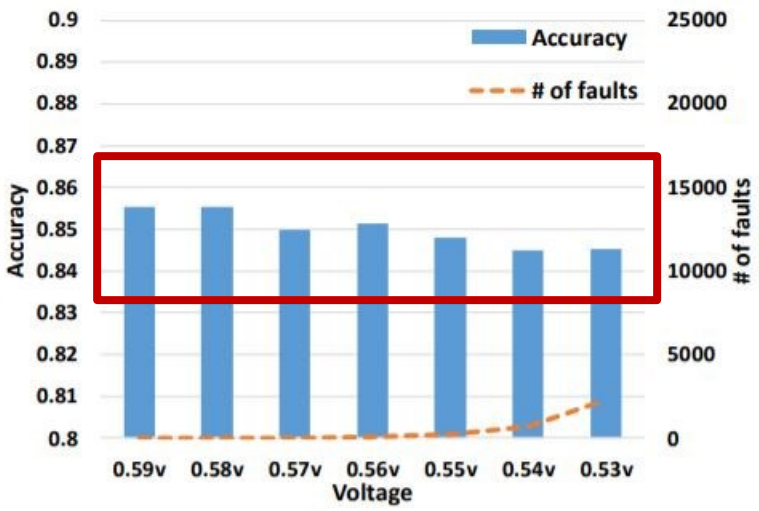


(d) (Activation Function, FPGA)= (Tanh,KC705)

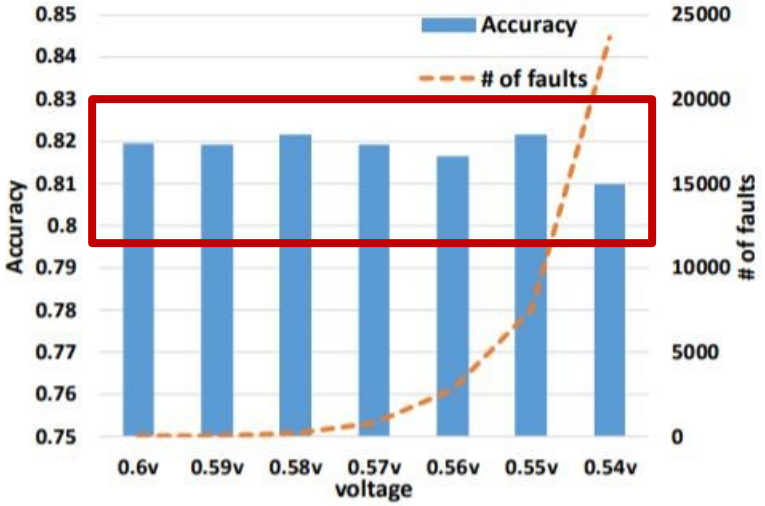
Training accuracy of the network for the classification of Cifar-10



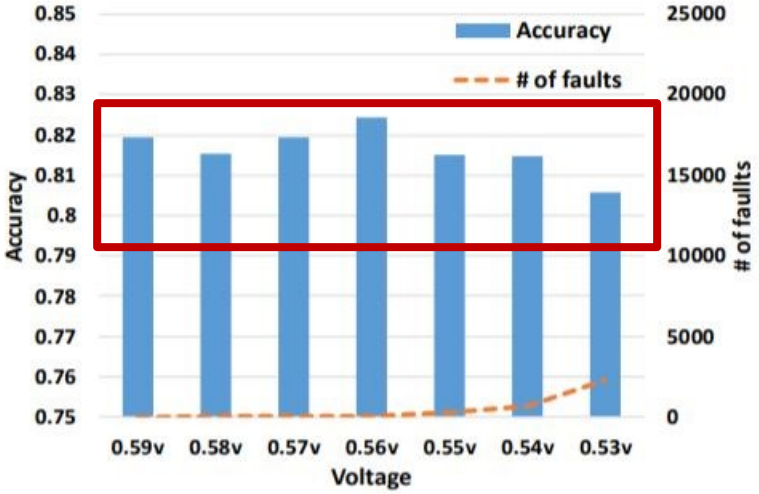
(a) (Activation Function, FPGA)= (RELU,VC707)



(b) (Activation Function, FPGA)= (RELU,KC705)



(c) (Activation Function, FPGA)= (Tanh,VC707)



(d) (Activation Function, FPGA)= (Tanh,KC705)

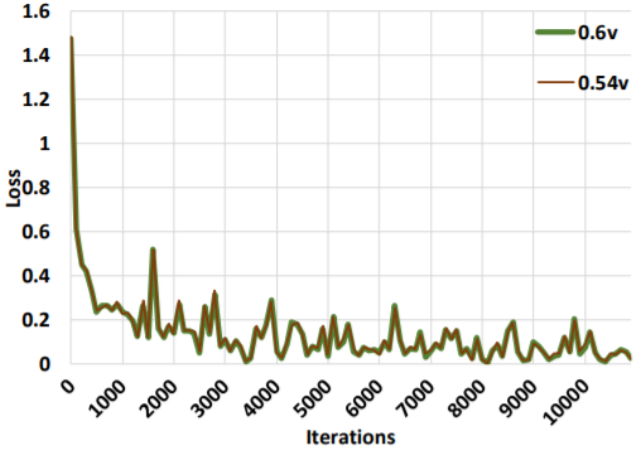
Convergence rate

Voltage ⬇

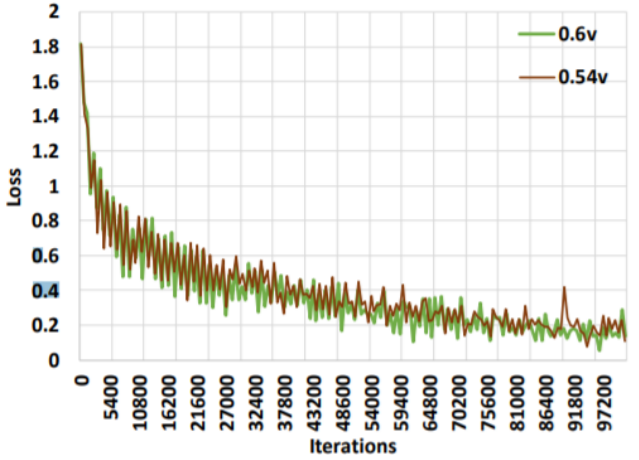
Faults ⬆

Convergence rate ⬇

Dataset	Reference accuracy	Device	Voltage	Iterations
MNIST (Relu)	98%	VC707	V_{min}=0.6V	4950
			V_{crash}=0.54V	5200
		KC705	V_{min}=0.59V	4900
			V_{crash}=0.53V	5050
CIFAR-10 (Tanh)	80%	VC707	V_{min}=0.6V	47800
			V_{crash}=0.54V	51200
		KC705	V_{min}=0.59V	43800
			V_{crash}=0.53V	61000



(a) (Dataset, Activation Function)= (MNIST, RELU)



(b) (Dataset, Activation Function)= (CIFAR-10, RELU)

Key Points of the First Contribution

- **The fault rate for real FPGA fabrics is under 1%.**
- **In the same number of iterations :**
  - **The reduction in classification accuracy is negligible.**
- **To have equal accuracy**
  - **On average, 10% more iterations are required**

Contributions

- Real Fault Maps
  - ❑ DNN training is inherently robust for undervolting-related faults, evaluated on the fault maps of real FPGA fabrics that are publicly available.
- Synthetic Fault Maps
  - ❑ fault rate of at least 25% can significantly affect the DNN accuracy.

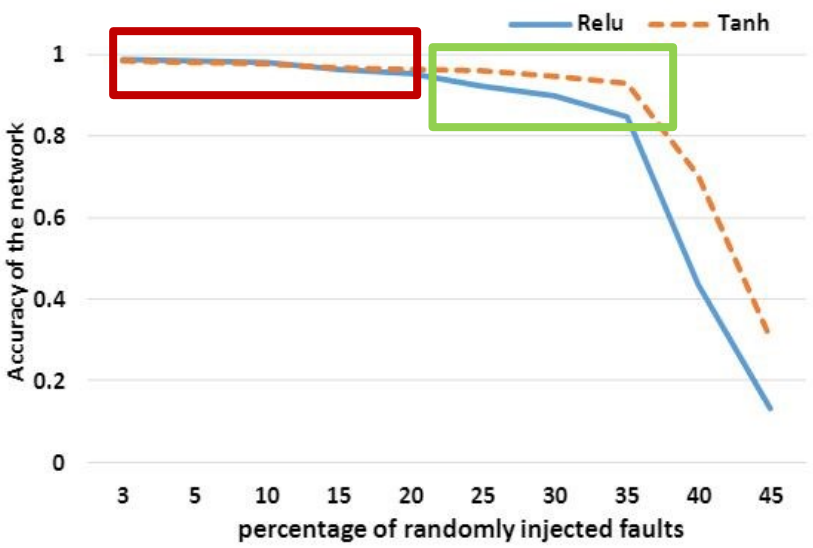
Random fault injection

Injecting faults with rates up to 50%

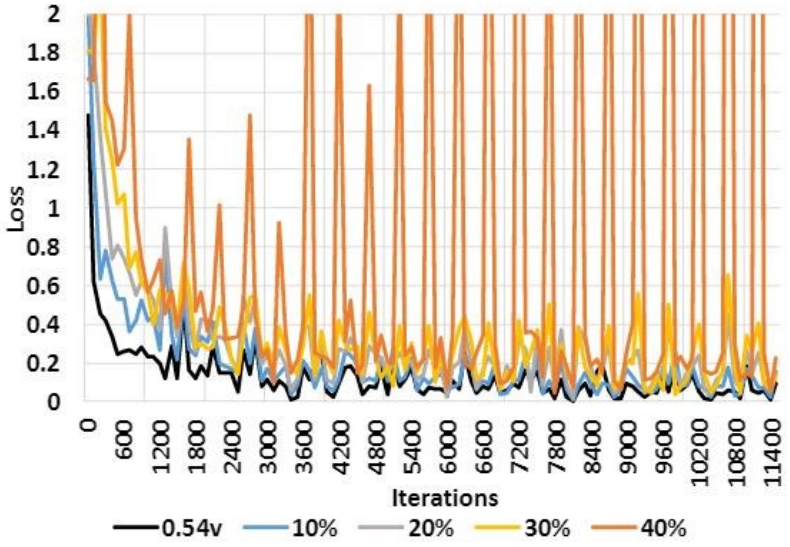
Uniform distribution

- Dataset: MNIST
- Device: VC707
- Network: LeNet-5
  - Relu
  - Tanh

Accuracy of training with generated faults



The accuracy comparison of Relu and Tanh activation functions for randomly generated fault maps



Comparison of the network loss for several random generated fault maps with 0.54V real fault map for VC707and LeNet-5 when the activation function is Relu.

Key Points of the Second Contribution

- **The accuracy of the network for fault rates lower than 10% is similar to accuracy for no-fault scenarios.**

- **There is no difference between the accuracy of Relu and Tanh activation functions for fault rates lower than 20%.**

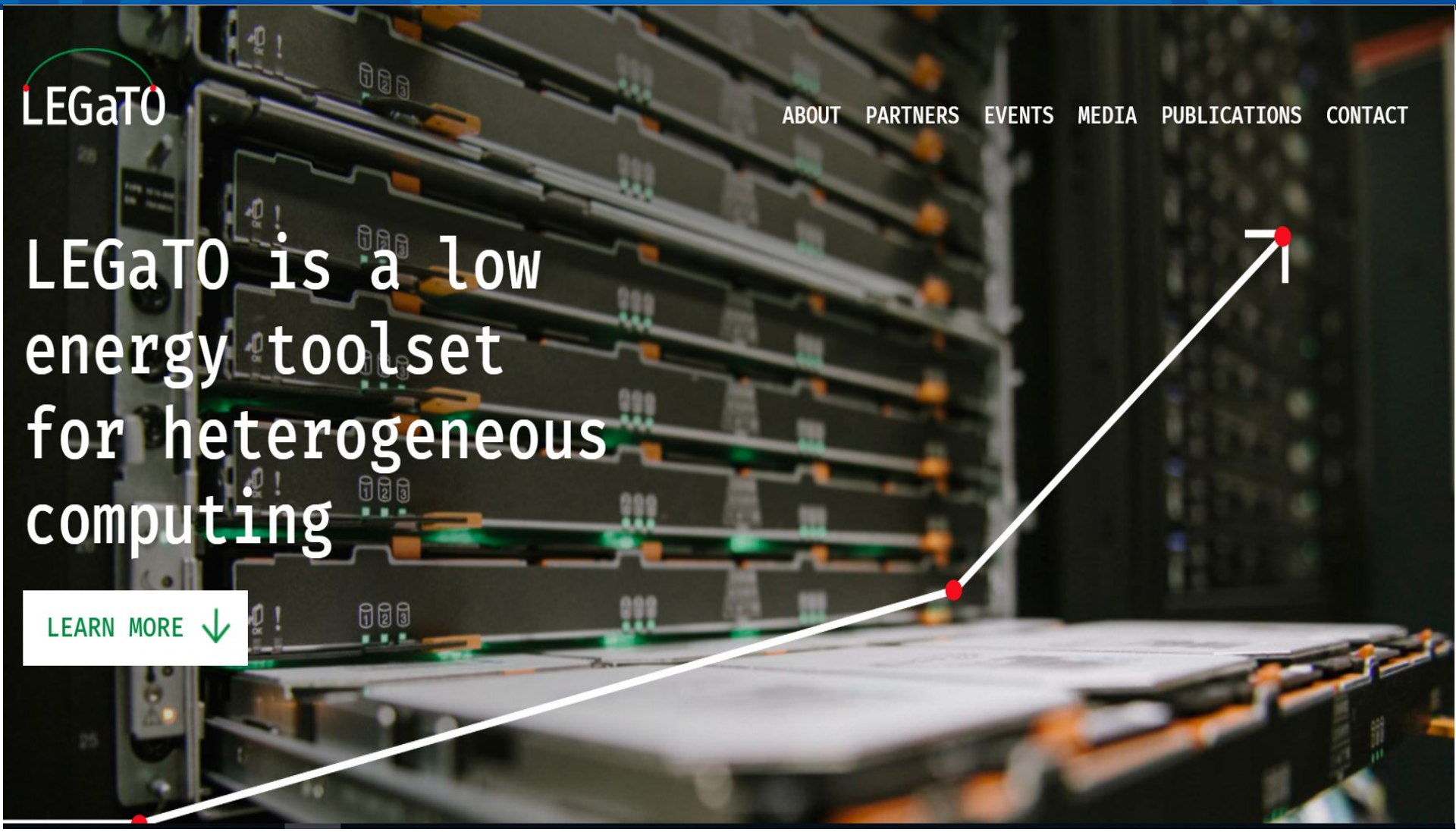
- **Tanh has better performance (in terms of classification accuracy) in the presence of a high rate of injected faults.**

- **Injecting 25% random faults**
  - **Relu: 6.25% lower than the training with no faults**
  - **Tanh: 2.75% lower than the training with no faults**

Conclusion

- We experimentally evaluate the effect of aggressive voltage underscaling of FPGA block RAMs on the training phase of deep neural networks.
- We have found that modern FPGAs are robust enough in extremely low-voltage levels and that low-voltage related faults can be automatically masked within the training iterations, so there is no need for costly software- or hardware-oriented fault mitigation techniques like ECC.
- Approximately 10% more training iterations are needed to fill the gap in the accuracy.
- The training process is even resilient to fault rate more than the fault rate of real FPGAs.
- We are working on repeat our model on real FPGAs.

LEGaTO



https://legato-project.eu/

Thanks!

Any Question/Comment?

Contact:
Kamyar Givaki
Kamyar.givaki@bsc.es