Hardware Accelerator of Neural Network Inference Supporting Fault Injections

Author: Filip Masár

Supervisor: Ing. Vojtěch Mrázek Ph.D.

Overview

Embedded systems can be sensitive to faults, and in some areas, such as autonomous driving or automotive systems, a fault can cause critical failure. Therefore, the fault tolerance of such systems is worth investigating. In the literature, this property is typically analysed in simulators that lack simulation quality or speed [1] [2] [3]. The aim of this work is to create a real NN inference accelerator in hardware that supports the emulation of fault injections.

Architecture

The NVIDIA Deep Learning Accelerator (NVDLA) [6] was our choice for the upgrade. It is a configurable accelerator based on a grid of processing elements (PEs) that perform the multiply-accumulate (MAC) operation that is crucial for NN inference. The architecture of the accelerator is shown below:

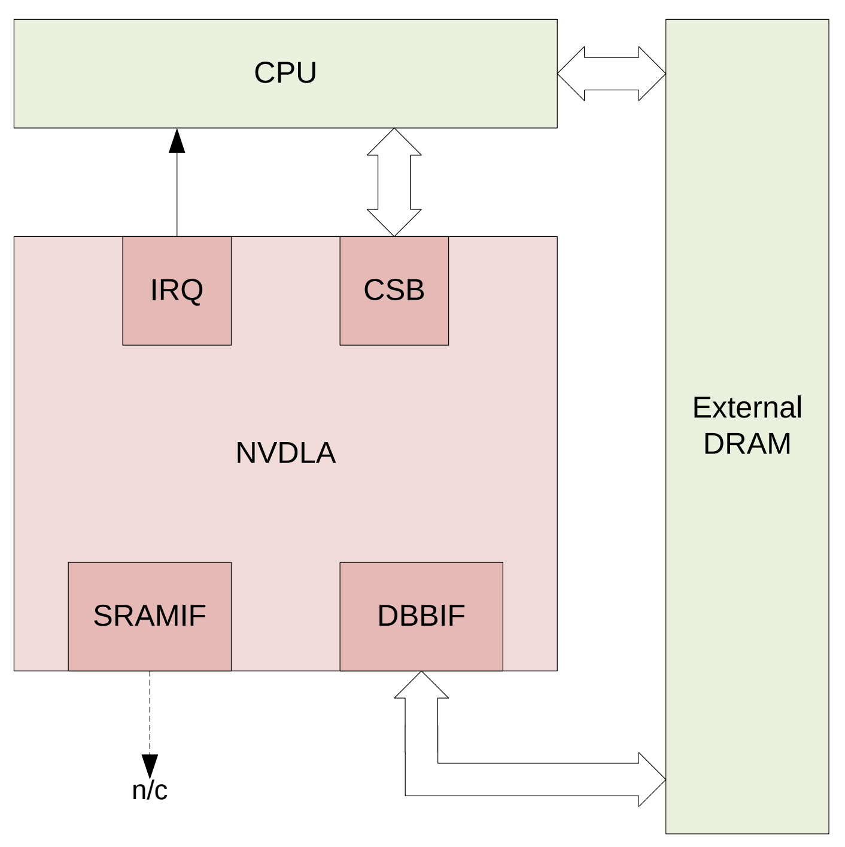

Figure 1. Small NVDLA system [6]

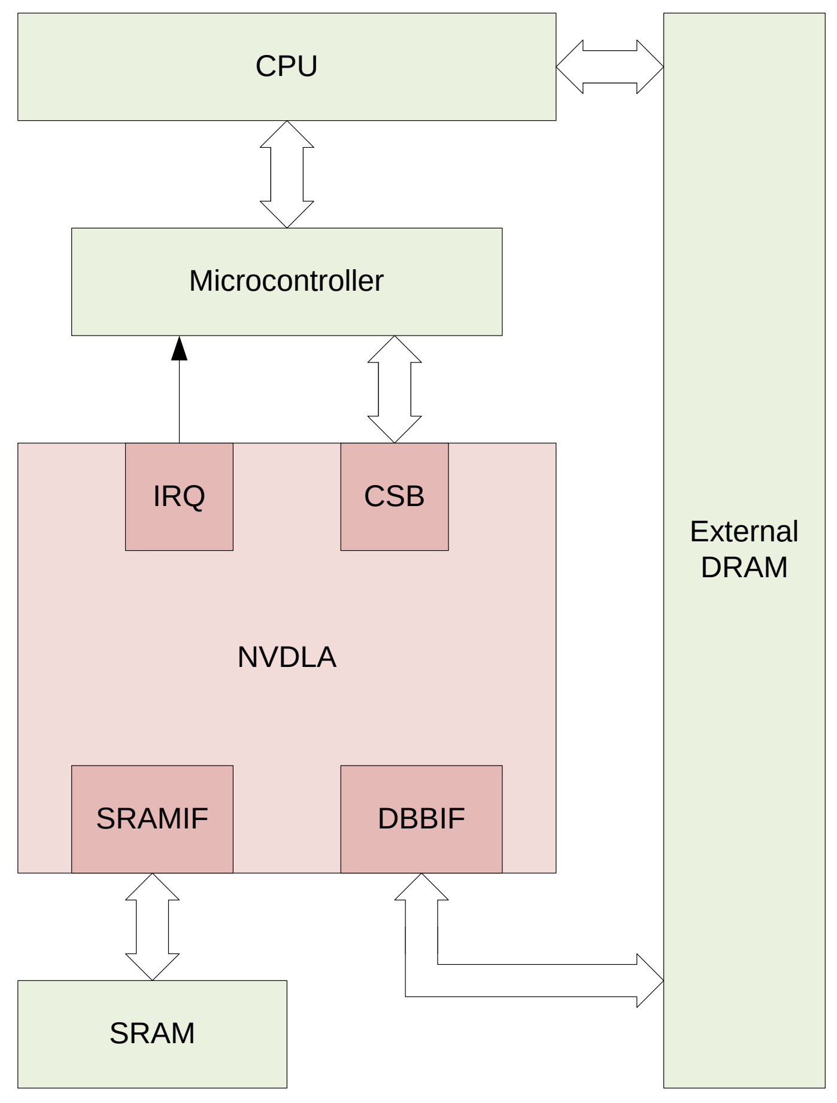

Figure 2. Large NVDLA system [6]

The target platform chosen was the Zynq UltraScale+ XCZU7EV SoC. This hardware limits us to the small configuration with 8-bit data precision.

The original NVDLA design targets the ASIC flow, and some modifications have been made to support configurable FPGAs. The most advanced part of the accelerator design was a software stack for the Linux subsystem. We also used the Tengine framework [7] to compile the pre-trained neural network coming from the Caffe tool.

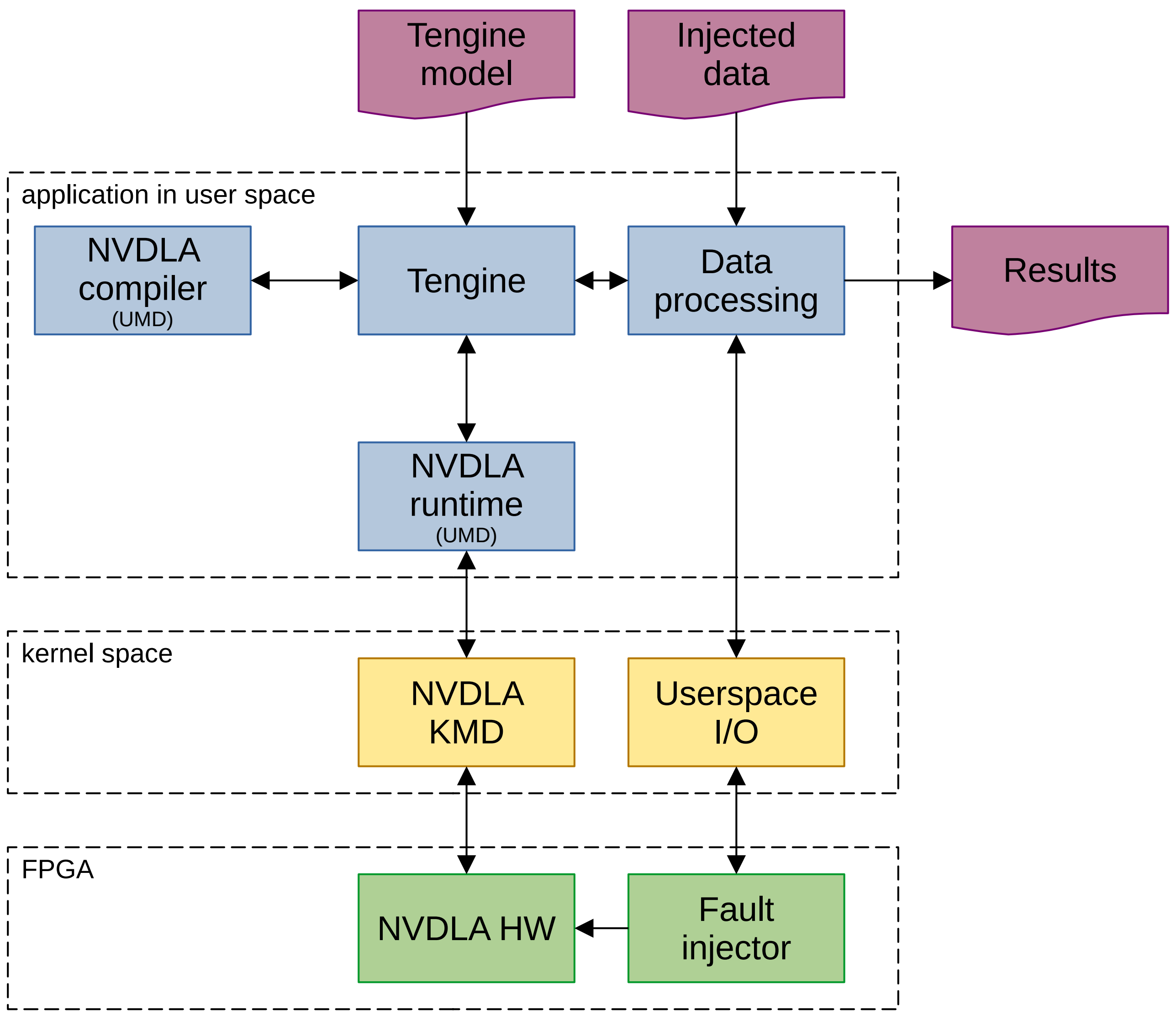

Figure 3. Final system architecture

Device	Threads	Image processing time (ms)
ARM Cortex-A53	1	22.68
ARM Cortex-A53	4	14.12
NVDLA small	64 MACs	4.59

Table 1. Evaluation time of the ResNet-18 (CIFAR-10 dataset).

Fault Injection

We decided to extend the architecture of the PEs to integrate the fault injector. There is a grid of 8x8 multipliers that can produce faulty output. The level of fault injection can be controlled using special configuration wires (red wires in Figure 4).

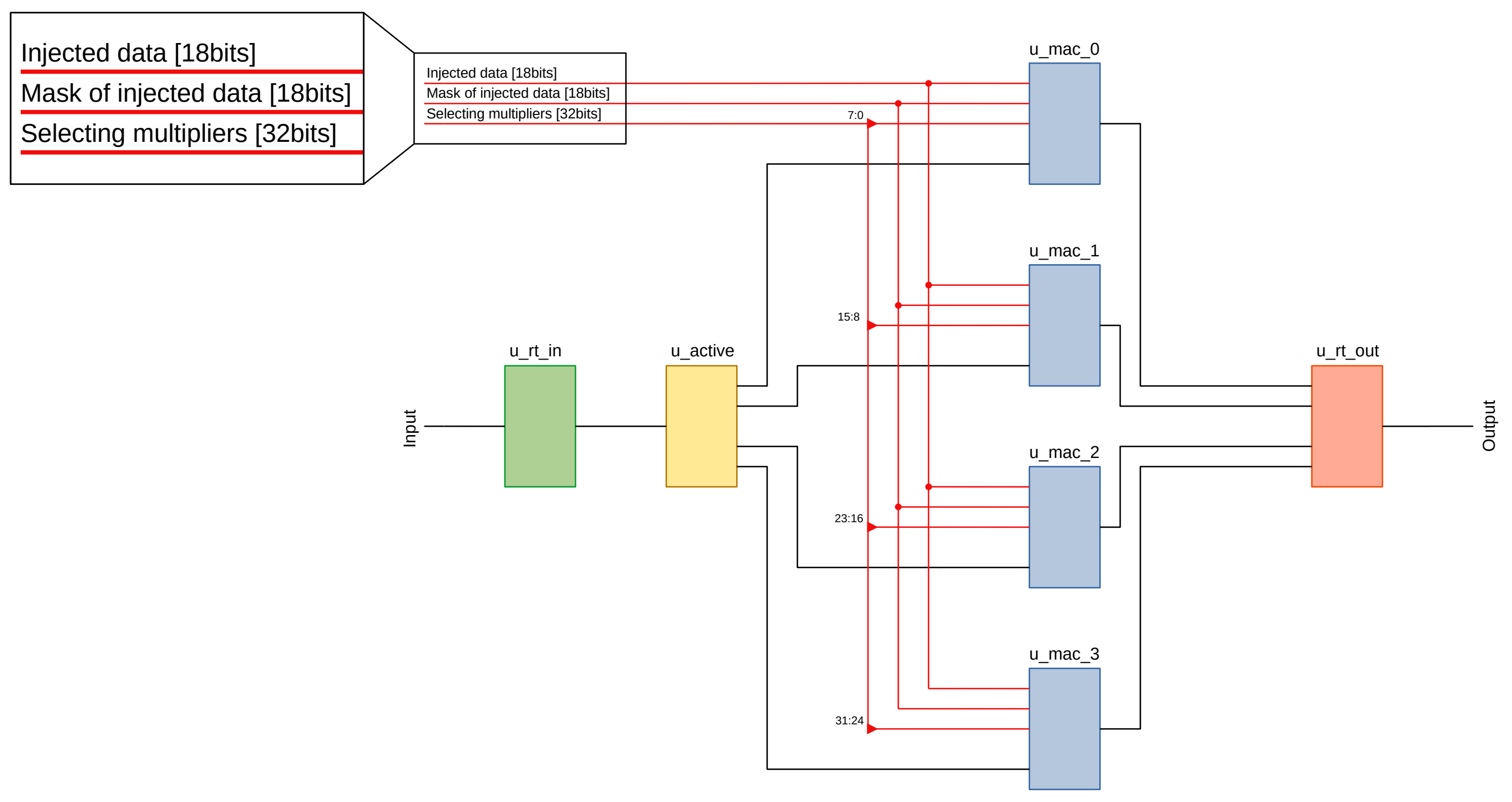

Figure 4. Computation partition with fault injection buses connected to MACs

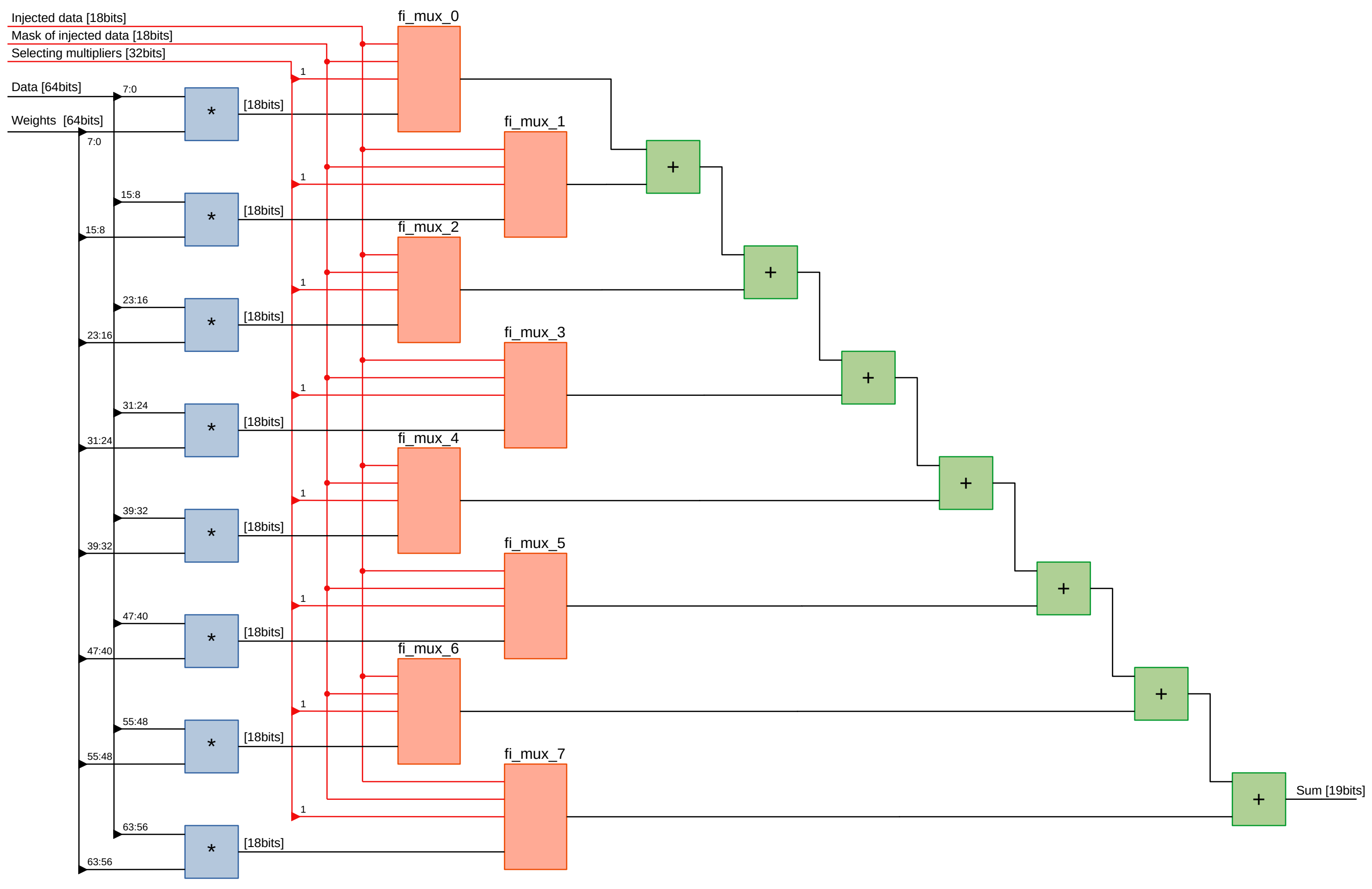

Figure 5. Fault injection inside MAC

Results

The ResNet-18 trained on CIFAR-10 was used and was trained with an accuracy of 75.1%. The Figures 6 shows the loss of accuracy as a function of the number of multipliers affected. The faults can only change individual bits at the output of multipliers (Figure 7).

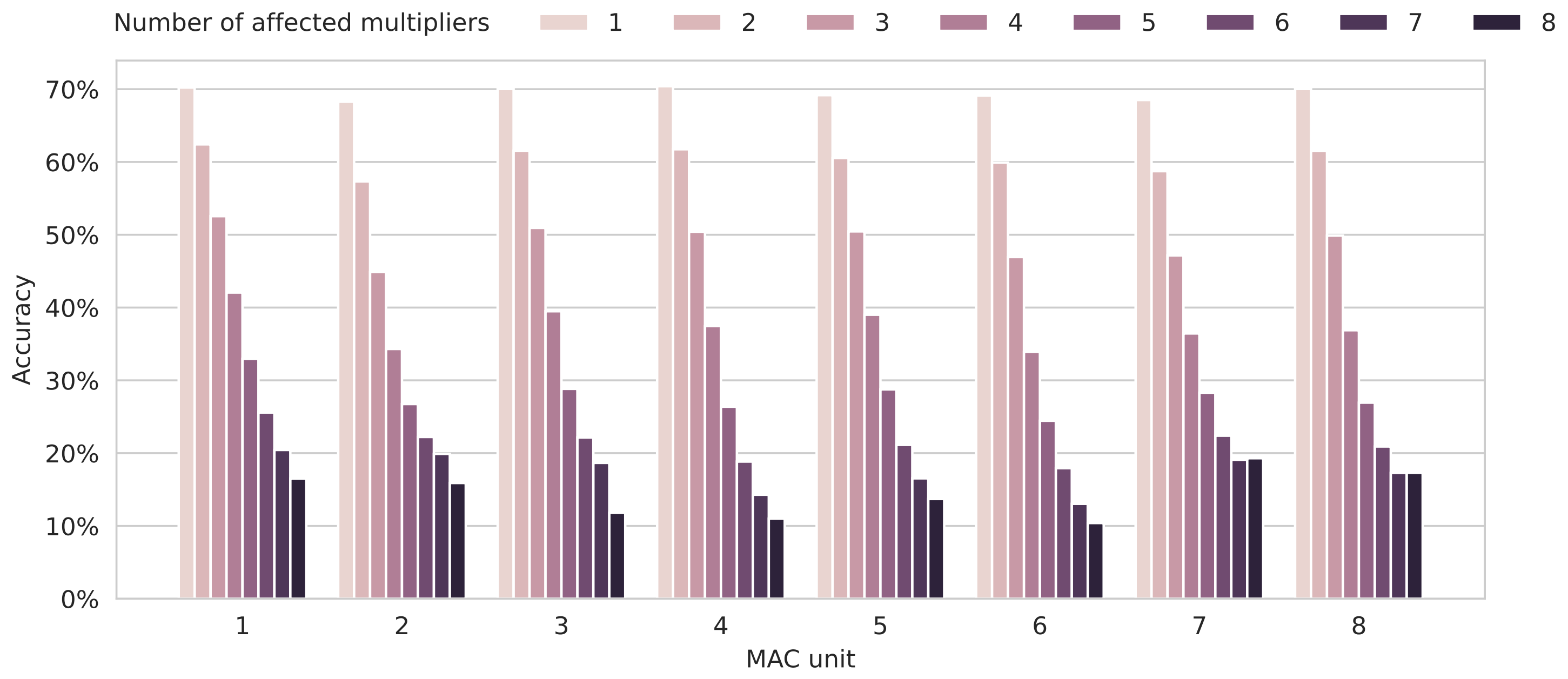

Figure 6. Comparison of the loss of accuracy for a given number of injected multipliers.

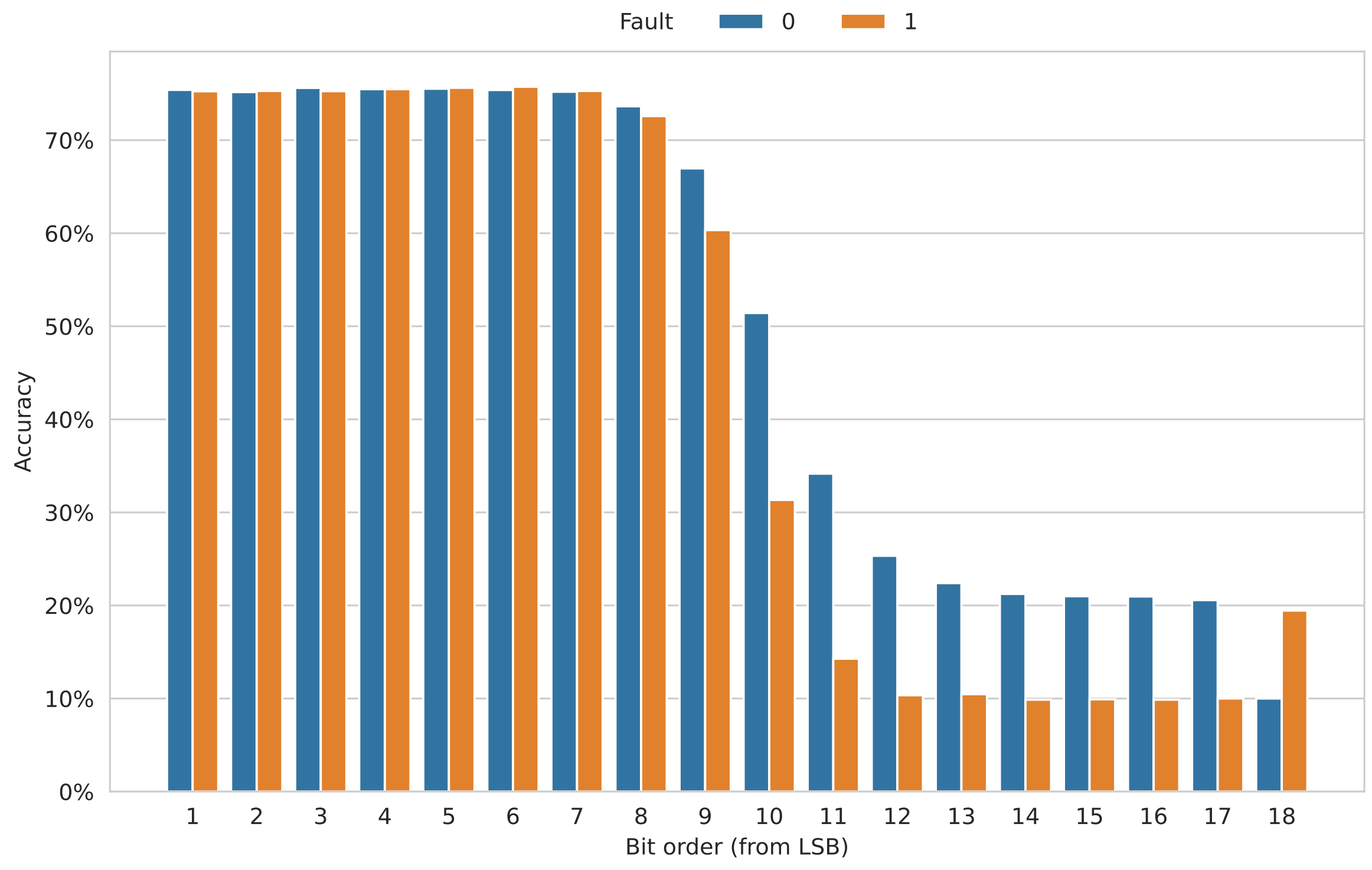

Figure 7. Comparison of accuracy in the injection of individual bits

Conclusion

The proposed accelerator supports real-time fault injection into the NN inference in real hardware. It allows researchers to investigate the parameters of the NNs and can help them to make the NNs more robust.

BRNO UNIVERSITY OF TECHNOLOGY FACULTY OF INFORMATION TECHNOLOGY

Excel@FIT 2024