Evaluation of Xilinx Deep Learning Processing Unit under Neutron Irradiation

Dimitris Agiakatsikas, Nikos Foutris, Aitzan Sari, Vasileios Vlagkoulis, Ioanna Souvatzoglou, Mihalis Psarakis, Mikel Luján, Maria Kastriotou and Carlo Cazzaniga

Abstract—This paper studies the dependability of the Xilinx Deep-Learning Processing Unit (DPU) under neutron irradiation. It analyses the impact of Single Event Effects (SEEs) on the accuracy of the DPU running the resnet50 model on a Xilinx Ultrascale+ MPSoC.

Index Terms—DPU, Data-center, Radiation-test, Neutrons, Reliability, DNN, AI

I. INTRODUCTION

FIELD Programmable Gate Arrays (FPGAs) have evolved from glue-logic devices to sophisticated heterogeneous computing platforms that pave the way to the new Artificial Intelligence (AI) wave.

As a result, datacenter market leaders increasingly integrate FPGA System-on-Chip (SoC) in their infrastructure to target complex AI applications. Companies like Microsoft, Amazon Web Services, and Baidu scale up AI and high-performance applications on hundreds of thousands of Intel and Xilinx FPGA devices [1].

A popular application of AI is Convolutional Neural Networks (CNNs). A growing body of research shows that deploying optimised CNN models on FPGAs achieves a higher performance per watt than CPU and GPU solutions [2]. However, modern FPGAs are vulnerable to Single Event Upsets (SEUs) due to their reliance on SRAM memory to store their configuration and application data. SEUs in SRAM FPGA-SoCs are not destructive but cause various failure modes, such as Silent Data Corruption (SDC), application crashes and kernel panics when an OS is used.

Previous works have explored the Architectural Vulnerability Factor (AVF) of custom FPGA CNN designs with fault-injection campaigns and irradiation experiments [3], [4]. These works have targeted relative simple CNN case studies for edge computing that neither require an Operating System (OS) nor have complex CNN topologies. Simple case studies serve well when focusing on analysing the reliability tradeoffs of various CNN configurations, e.g., the reliability of a CNN under different quantisation and model-compression schemes. Exploring the reliability of large-scale datacenter CNN applications, however, requires a different testing paradigm. The case studies should test additional aspects of a system, such as the operating system’s stability, while SEUs are occurring in resources of the CPU, e.g., L1 and L2 caches, on the on-chip Ethernet controller and in custom logic implemented with Programmable Logic (PL).

To this extent, this work analyses the dependability of the whole computational stuck of a commercial CNN inference solution, namely the Xilinx Vitis AI Deep Learning Processing Unit (DPU). In more detail, we implemented the DPU on a Zynq Ultrascale+ XCZU9EG MPSoC, which executed image classification with the resnet50 CNN model. By performing Neutron Irradiation experiments, we observed that the FPGA-SoC OS did not crash due to errors in L1 and L2 caches of the CPU, while the DPU application had a very low AVF.

II. BACKGROUND

A. Vitis AI and the Deep Learning Processing Unit (DPU)

Xilinx has introduced a rich ecosystem of tools and Intellectual Property (IP) cores to ease the development of AI applications. In more detail, Xilinx provides the Vitis AI development environment that encompasses 1) AI frameworks (e.g., Tensorflow), 2) pre-optimised AI models, 3) quantization and model compression tools, and 4) the DPU, with all necessary Linux drivers to seamlessly deploy a CNN application on Xilinx devices.

The DPU is a convolution neural network accelerator IP offered by Xilinx for Zynq-7000 SoC and Zynq Ultrascale+ MPSoC devices. The DPU is implemented with PL and is tightly interconnected via the AXI bus to the SoC processing system (PS), as shown in Fig. 1. The DPU executes special instructions that are generated by the Vitis AI compiler. A typical Vitis AI development flow involves 1) the optimisation and compilation of a CNN model to DPU instructions and 2) the compilation of software running on the Application Processing Unit (APU), i.e., CPU.

The APU pre- and post-processes DNN data, controls the DPU, and orchestrates the movement of instructions and data between the DPU, the CPU, and the off-chip DDR memory.
The DPU consists of an instruction scheduler and up to three on-chip BRAM buffers and computing engines. The instruction scheduler fetches and decodes DPU instructions from off-chip memory and controls the on-chip memories and computing engines. The DPU is available in eight configurations, i.e., B512, B800, B1024, B1152, B1600, B2304, B3136, and B4096. Each configuration utilises a different number of computing engines and on-chip memories in order to target different sized devices and support various DPU functionalities, e.g., ReLU, RELU6, or Leaky ReLU.

III. EXPERIMENTAL SETUP

A. ChipIr Neutron Beam

ChipIr is an ISIS neutron and muon facility instrument at the Rutherford Appleton Laboratory (UK), designed to deliver an atmospheric-like fast neutron spectrum to test radiation effects on electronic components and devices [5], [6]. The ISIS accelerator provides a proton beam of 800 MeV, 40 µA, 10 Hz, impinging on the tungsten target of its Target Station 2, where ChipIr is located. The spallation neutrons produced illuminate a secondary scatterer which optimises the atmospheric-like spectrum arriving at ChipIr, with an acceleration factor of up to $10^9$ for ground-level applications. With a frequency of 10 Hz, the beam pulses consist of two 70 ns wide bunches separated by 360 ns. The beam fluence at the position of the Device Under Test (DUT) was continuously monitored by a silicon diode, while the beam flux of neutrons above 10 MeV during the experimental campaign was $5.6 \times 10^6$ neutrons/cm$^2$/s. The beam size was set through the two sets of the ChipIr jaws to 7 cm x 7 cm.

B. Design Under Test (DUT): DPU-B4096

The Vivado DPU targeted reference design (TRD) [7] provided by Vitis AI v1.3.1 was implemented with Vivado 2020.2 for a ZCU102 development board. The ZCU102 board features the Zynq UltraScale+ XCZU9EG MPSoC. The B4096 configuration of the DPU was synthesised with default settings, i.e., with RAM_USAGE_LOW, CHANNEL_AUGMENTATION_ENABLE, DWCV_ENABLE, POOL_AVG_ENABLE, RELU_LEAKYRELU_RELU6, Softmax.

The design was implemented with Vivado’s Performance_ExplorePostRoutePhysOpt* run strategy because Vivado’s default run strategy resulted in time violations. Table I shows the resource utilisation and operating frequency of the DPU TRD. Due to the high resource utilisation of the TRD, Vivado reported a relatively high percentage ($41.45\%$) of essential bits – 59281993 out of the 143015456 total configuration bits were essential bits. Please recall that essentials bits are configuration bits that, when corrupted have the potential to cause functional errors. Two important notes can be made for Table I.

1st note: All resources in the DPU operate at 325 MHz except for the DSPs, which run at $2 \times 325 \text{ MHz} = 650 \text{ MHz}$. This is because the DPU design applies a double data rate technique on DSP
resources. Since DSPs are able to operate in a much higher frequency than other PL resources, one can perform $N$ times more computation by running the DSPs with $N$ times the frequency of the surrounding logic while multiplexing and de-multiplexing their input and output data, respectively.

2nd note: The design utilises 319, 55, 405, 4 and 1 LUT, LUTRAM, Flip-Flops (FF), BRAM and DSP more resources, respectively, than the baseline DPU-TRD design. This is because we included the Xilinx Soft Error Mitigation (SEM) controller in the design to perform fault injection and validate our experimental setup prior to the radiation tests. The clock of the SEM controller was gated off during beamtime to replicate a simple out-of-the-box implementation scenario.

We used Petalinux 2020.2 to generate a Linux OS image for the ZCU102 by using the default Board Support Package (BSP) provided by the DPU-TRD, except 1) the nfs-utils package which was additionally added in RootFS to mount a Network File Sharing (NFS) folder on Linux, and 2) the u-boot bootloader that mounted an external SD EXT4 file system instead of INITRD RAM disk.

The CNN application that run on the DPU was the resnet50.xmodel, which was also provided by the Vitis AI DPU-TRD. This resnet50 model is neither compressed nor quantised but serves nice as a baseline application for comparison with more optimised models that we aim to implement and test in future work.

C. Test procedure

Fig. 3 shows the test setup of the radiation experiment. A laptop in ChipIr’s control room orchestrated the test procedure of the DUT. The ZCU102 development board (hosting the DUT), an Ethernet-controlled Power Supply Unit (PSU), and a USB device that remotely reset the ZCU102 (i.e., by electrically shorting the SRTS_B and POR_B buttons of the board) was located in the beam room.

The test of the DUT took place as follows. 1) An Experiment Control Software (ECS) running on the laptop remotely resets the DUT and waits for the DUT to boot, 2) the DUT Linux OS restarts, and 3) after a successful kernel boot, an /etc/init.d/startup.sh script executes the following sub-tasks: 4a) the DUT connects on an NFS folder located on the laptop, 4b) the DUT writes a sync.log file in the shared NFS folder to notify the ECS of a successful boot, 4c) an initial resnet50 classification takes place to warm-up the CPU caches, 4d) the sync.log is updated to notify ECS that it is ready to start image classifications, 4e) the /etc/init.d/startup.sh enters an infinite loop where it continuously runs DPU classifications and stores the results in the NFS folder to be checked by the ECS. The result checking (i.e. by the ECS) of each classification iteration is synchronised with the DUT via a mutex stored in the shared sync.log file. The ECS remotely resets the DUT when it detects a boot timeout, a Critical Error (see Sec. IV) or a result timeout. It is worth noting that for each classification iteration, the DUT saves the classification result and the Linux dmesg.log for post-analysis.

IV. Experimental Results

In this section, we discuss the impact of neutron radiation effects on the reliability of the DPU accelerator. Given that the DPU comprises of a heterogeneous architecture including the ARM SoC and the FPGA fabric, we first present the cross-sections of the memories of the PS part (i.e. CPU caches) and then discuss how the SEUs in the PL configuration memory affect the behaviour of the system. Please note that the neutron radiation experiments took place at ChipIr on May 2021.

Table II presents the cross-sections of the 32KB Level-1 Data (L1-D) Cache, the 32KB Level-1 Instruction (L1-I) Cache, and the translation lookaside buffer (TLB) – a two-level TLB with 512 entries that handles all translation table operations of the CPU. Table III presents the cross-sections of the 1MB Level-2 (L2) Cache and the Snoop Control Unit (SCU). The SCU has duplicate copies of the L1 data-cache tags. It connects the APU cores with the device’s accelerator coherency port (ACP) to enable hardware accelerators issue coherent accesses to the L1 memory space. The upsets in the data and tag arrays in both the L1 and L2 caches have been separately identified. The cross-sections of the tag arrays have been calculated based on the tag sizes of the caches, e.g., a 16-bit tag in the 16-way set associative 1MB L2 cache. The cross sections have been calculated for a total fluence of $5.5 \times 10^{10}$ neutrons/cm$^2$ on a more than 3-hour radiation experiment. The results show that the cross-sections of the tag arrays are slightly lower than those of the data arrays. Our cross-section calculations for all caches (i.e., L1 and L2) are very close to those reported in [4].

Fig. 4 presents the proportion of detected upsets during cache accesses per CPU core. As shown in the figure, the upsets in the L1 caches are balanced between the four cores, while in the L2 cache, more upsets were observed in Core 3. In future work, we aim to save the utilisation of all MPSOC cores and OS running processes to better understand the unbalanced distribution of detected upsets per core in L2.

All MPSOC caches are protected against SEUs with Error Correction Code (ECC) mechanisms, e.g., L1-D and L2 caches
### Table IV: Critical and Tolerable Errors in the DPU

| Error Type | Core 1 | Core 2 | Core 3 | Core 4 |
|------------|--------|--------|--------|--------|
| Critical   | 0.27%  | 0.36%  | 0.54%  | 0.48%  |
| Tolerable  | 0.15%  | 0.17%  | 0.21%  | 0.19%  |

---

### Table V: Cross-Section of Error Detection (SECD) with Single Error Correction (SEC)

| Cross-Section | Critical (C) | Tolerable (T) |
|---------------|--------------|---------------|
| L1 Cache      | 5.01E-08     | 3.04E-08      |
| L2 Cache      | 1.39E-07     | 8.54E-08      |

---

### Fig. 4: Cache upsets per CPU Core

- Core 1: 12 upsets
- Core 2: 25 upsets
- Core 3: 13 upsets
- Core 4: 46 upsets

---

### REFERENCES

1. A. M. Keller and M. J. Wirthlin, "Impact of Soft Errors on Large-Scale FPGA Cloud Computing," in ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (FPGA), New York, NY, USA, 2019.
2. E. Wang, J. J. Davis, R. Zhao, H.-C. Ng, X. Niu, W. Luk, V. C. Frost, and P. Rech, "Selective Hardening for Neural Networks in FPGAs," IEEE Transactions on Nuclear Science, vol. 66, no. 1, pp. 216–222, 2019.
3. F. Libano, B. Wilson, J. Anderson, M. Wirthlin, C. Cazzaniga, C. D. Czirr, and A. C. Manzella, "Performance and Reliability Comparison of Soft Error Mitigation Schemes in the Xilinx UltraScale+ FPGA," in ACM/IEEE International Symposium on Field-Programmable Gate Arrays (FPGA), New York, NY, USA, 2019.
4. V. C. Frost, and P. Rech, "Selective Hardening for Neural Networks in FPGAs," IEEE Transactions on Nuclear Science, vol. 66, no. 1, pp. 216–222, 2019.
5. E. Wang, J. J. Davis, R. Zhao, H.-C. Ng, X. Niu, W. Luk, V. C. Frost, and P. Rech, "Selective Hardening for Neural Networks in FPGAs," IEEE Transactions on Nuclear Science, vol. 66, no. 1, pp. 216–222, 2019.

---

**Conclusion:**

The neutron radiation experiment demonstrated that the ECC and the interleaving schemes integrated into the UltraScale MPSoC protect the software stack from SEUs. The Xilinx UltraScale+ ADPU, which is the most vulnerable part of the UltraScale MPSoC, was shown to be resistant to SEUs when implemented in the FPGA. The results obtained during the neutron radiation experiment indicated that the UltraScale MPSoC is highly resilient to SEUs, which is a crucial aspect of its reliability in high-reliability applications. Further improvements in error detection and mitigation strategies are necessary to improve the overall system reliability.
[6] C. Cazzaniga, R. G. Alía, M. Kastriotou, M. Cecchetto, P. Fernandez-Martinez, and C. D. Frost, “Study of the Deposited Energy Spectra in Silicon by High-Energy Neutron and Mixed Fields,” IEEE Transactions on Nuclear Science, vol. 67, no. 1, pp. 175–180, 2020. [Online]. Available: https://doi.org/10.1109/TNS.2019.2944657

[7] “Zynq UltraScale MPSoC DPU TRD V3.3 Vivado 2020.2,” Xilinx Inc. [Online]. Available: https://github.com/Xilinx/Vitis-AI/blob/v1.3.1/dsa/DPU-TRD/prj/Vivado/README.md