# Lightweight Active Fences for FPGAs

Anis Fellah-Touta, Lilian Bossuet, Vincent Grosso, Carlos Andres Lara-Nino

# Lightweight Active Fences for FPGAs

Anis Fellah-Touta[1], Lilian Bossuet[1],
Vincent Grosso[1], and Carlos Andres Lara-Nino[2]
1. UJM St-Etienne, CNRS, LabHC UMR 5516, SAINT-ETIENNE, France.
2. Universitat Rovira i Virgili, DEIM, TARRAGONE, Spain.

September 8, 2024

**Abstract**

Power analysis can compromise the security of computing platforms by leveraging the energy consumption derived from the processing of cryptographic routines. Algorithmic protections against this threat seek to obfuscate the operation of such procedures to minimize information leakage. However, these solutions are ad-hoc for each algorithm. In turn, the protection costs are independent and they accumulate as more protected algorithms are implemented on the device. This area of research is critical due the recent surge in proposals for remote power analysis attacks (RPA) that target FPGAs when they are spatially shared among multiple tenants. In this attack model, an adversary will employ on-board circuitry to implement on-chip voltage sensors and retrieve the samples remotely. The use of active fences has been proposed as a protection against remote power analysis attacks. This countermeasure relies on reserving a reconfigurable space within the FPGA which will separate it into sub-regions. These “fences" will then generate some electrical interference to hinder the performance of an attack. As FPGAs can be configured in multiple ways, there are different approaches for connecting the hardware inside the fence. In this work, we describe a LUT-based configuration which can achieve the same instantaneous power drop as a ring oscillator bank with less LUTs. This contributes to reducing the hardware costs of active fences.

**Keywords:** Active Fence, Power Waster, Side Channels Countermeasure.

**Please cite as:**

```
@InProceedings{FBGL24,
title = {{Lightweight Active Fences for FPGAs}},
author = {Fellah-Touta, Anis and Bossuet, Lilian and Grosso, Vincent and Lara-Nino, Carlos
Andres},
booktitle = {Proceedings of the 2024 IFIP/IEEE International Conference on Very Large Scale
Integration (VLSI-SoC)},
pages = {1--4},
year = {2024},
publisher = {IEEE},
location = {Tanger, Morocco}}
```

# 1 Introduction

Integration of FPGA accelerators into cloud computing environments has made cloud FPGA platforms popular since they can significantly boost the performance of specific applications [LHZ21; Che+14]. On demand hardware acceleration required for high-performance computing tasks is provided by cloud FPGAs. In their infrastructure, major cloud providers such as Alibaba Cloud, Amazon Web Services and Microsoft Azure have implemented FPGA-based accelerators. In most cases these types of platform offer FPGA instances as a service which allows users to directly deploy their designs or applications onto the cloud. For example, F1 instances in which users can run their own FPGA designs are offered by Amazon Web Services while Microsoft's Project Catapult uses FPGAs to accelerate Bing search rankings among other things. Multi-tenant cloud FPGAs refer to situations where multiple users share one physical FPGA chip at once. The aim behind this idea is to increase hardware usage and reduce costs, whereby a user pays only for part of an FPGA instead of the entire one. This can be achieved by either dynamically reconfiguring some parts of an FPGA or statically compiling many users' modules into one hardware image that will load onto an FPGA. However, security is still an issue because they are vulnerable to attacks such as remote power analysis attack (RPA). Recent studies show that cloud FPGAs could be remotely attacked through power analysis [ZS18; Sch+21]. These attacks are done without physically accessing the FPGA by using internal sensors to measure voltage fluctuations and indirectly retrieve valuable data. This is a significant concern since cloud FPGAs are used in machine learning workloads and cryptographic operations which process sensitive information. The fluctuations in power consumption are detectable through the Power Distribution Network (PDN) of the FPGA. An attacker can indirectly measure these power consumption variations by implementing on-chip sensors on the FPGA fabric such as Time-to-Digital Converters (TDCs) and Ring Oscillators (ROs). These sensors are designed to sense the changes in delay caused by voltage drops, which are induced by the switching activity of the logic under attack.

Remote Power Analysis (RPA) with Time-to-Digital Converters (TDCs) and Ring Oscillators (ROs) is an emerging threat against FPGAs. Several countermeasures against these threats have been proposed, which include masking [ISW03], shuffling [Vey+12] and other hiding techniques [Kra+19] as well as straight up preventing the deployment of voltage-sensing circuits [Gna+18]. We are interested in "hidding" methods due their simplicity and generality. One of the most promising of such strategies involves the use of active fences. Active fences are composed of a ring of oscillator circuits placed between the victim circuit and the rest of the FPGA. These oscillators add noise to the power consumption patterns, thereby reducing the signal-to-noise ratio (SNR) that makes it harder for attackers to extract useful information.

Remote Power Analysis (RPA) is an emerging threat against FPGAs [ZS18; Sch+21]. In this paradigm, an adversary employs internal sensors to remotely study the power of the system. Several countermeasures have been proposed against this attack. They include cryptographic methods like masking and shuffling, software methods like bitstream checking, and physical methods like generating electric interference. In this work, we are interested in the latter methods given their simplicity and generality. One of the most promising of such

strategies involves the use of active fences [Kra+19; Gla+23]. These circuits are composed of LUT-based ring oscillators placed between the victim circuit and the rest of the FPGA. These oscillators add noise to the power consumption patterns, thereby reducing the signal-to-noise ratio (SNR) and reducing the ability of an adversary for retrieving useful information about the victim.

Masking of neural networks increases the area by around 2-3 times, as compared to the neural network in its original form. This leads to a significant resource overhead on the FPGA and makes active fences more suitable to protect neural networks implemented on shared FPGAs, since this approach can provide strong protection against RPA without the significant resource overhead incurred by masking techniques. Krautter et al. demonstrated the effectiveness of active fences by placing an RO fence of approximately the same size as the victim AES circuit between the AES module and the rest of the FPGA, which increased the number of traces needed to break a byte of the secret key using Correlation Power Analysis (CPA) by approximately $\times 60$. We present a new LUT-based power waster to build active fences. This design consumes the same instantaneous voltage drop as the RO-based grid, but it uses fewer LUTs. Given that neural networks are one of the usual workloads on cloud FPGAs, we have applied the active fence to protect a neural network and evaluate the effectiveness of the proposed configuration. Our proposed LUT-based waster is effective in reducing the SNR, offering a promising countermeasure against side-channel attacks while providing optimal resource use in shared FPGA environments.

In this paper, we present a new LUT-based power waster circuit useful to build active fences. The proposed design can generate an equivalent instantaneous voltage drop as a conventional ring oscillator array, but with fewer LUTs. Given that neural networks (NN) are one of the usual victims of RPA, we have applied the proposed active fence to protect one of such implementations and evaluate the effectiveness of the proposed configuration. Our proposed lightweight fence is effective in reducing the SNR, offering a promising countermeasure against side-channel attacks while providing improved resource utilization.

The rest of the paper is structured as follows. In Section 2 we review some relevant works from the literature. Section 3 describes the proposed active fence design and analyzes its characteristics. Section 4 describes our evaluation methodology. In Section 5 we provide and discuss experimental results. Lastly, Section 6 concludes the paper.

## 2 Background and related Work

The RPA on shared FPGAs pose a serious threat not only to cryptographic modules but also to neural networks, which have increasingly been used in cloud computing [Tia+23; MGT22; Moi+21; Zha+21]. These attacks can exploit the shared power distribution network (PDN) of the FPGA to observe power variations caused by different computational activities, even without physically accessing the hardware. By monitoring these variations, attackers can infer sensitive information such as neural network weights and architecture, which are crucial for maintaining the integrity and confidentiality of the data processed by these networks. The main sensors for such attacks include TDCs and ROs. These sensors operate by measuring changes in logic delay, which vary with the

supply voltage, rather than directly measuring the voltage itself. TDCs typically consist of a tapped delay line implemented using a chain of CARRY elements, allowing them to capture voltage fluctuations over very short intervals. This makes TDCs particularly effective for high-frequency sampling required in RPA. In the other hand, a ROs consists of a ring osillator whose output is connected to a counter. The counter measures the number of oscillations from the ring over a fixed sampling period, which can then be converted to the oscillation frequency. Changes in the oscillation frequency correspond to fluctuations in the supply voltage, which can be correlated to the power consumption of any sensitive operations on the FPGA. Given that RPA can potentially result in compromising either cryptographic operations or neural network operations on shared FPGAs, there must be developed strong countermeasures.

Preemptive security against RPA attacks can be provided through bitstream scanning. This strategy seeks to prevent the deployment of bitstreams containing malicious circuits that could enable RPA [Gna+18]. One means of doing this is through identifying LUT-based oscillator designs in the configuration bitstream. FPGA vendor tools already give critical warnings in case combinatorial loops are detected during the bitstream generation. Other tools, like FPGADefender [La+20], go further and warn against the use of other potentially malicious circuits like non-combinatorial loops and time-to-digital converters (TDC). This is a virus-scanner-like tool specifically designed for FPGAs. It scans the bitstreams for the presence of self-oscillator circuits used in RPA, such as ROs. Once such elements are found, the tool blocks the loading of the bitstream onto the FPGA, hence effectively stopping possible attacks in the early stages of deployment. The main drawback of these tools is that, like antivirus software, they must be periodically updated to catch up with emerging internal sensor designs [FBL24].

Other RPA countermeasures focus on enhancing the resilience of the victim against side channel analysis. This is primarily achieved through masking [ISW03] and shuffling [Vey+12]. These techniques break the relationship between the sensitive data and the side channels, effectively thwarting the attack even if the adversary manages to retrieve some measurements. However, applying these protections to NN implementations is complicated and resource-consuming [Bro+24; Dub+21]. Neural networks usually require a great number of computations and data transfers, and the addition of masking or shuffling can reduce the performance of the NN. Additionally, these solutions must be tailored to every particular NN architecture.

Another approach for protecting neural networks relies on disrupting the side-channels of the device. These strategies aim at reducing the SNR, making it difficult for the attacker to extract useful information from the leakages. Some of these approaches include de-synchronization [Bre+23], addition of spurious computations [Cha+22], hardware reconfiguration [Ahm+23], and noise generation through hardware circuits [YCZ23]. A rather common technique is to place a grid of ring-oscillators as an "active fence" between the victim and the attacker. The ring-oscillators are randomly activated to induce noise and increase the number of measurements required to perform an attack. The main advantage of this approach is that it does not require changes in the victim implementation itself, making it a generic form of protection.

# 3 Proposed lightweight active fence

Power wasters are a type of hardware trojan sometimes used in fault injection attacks. However, they can be used in the construction of active fences. By activating power wasters randomly it is possible to introduce electrical noise into the side channels. This can obfuscate the data-associated leakages of the device. The activation pattern can be generated with internal circuits, for example a pseudo-random number generator (PRNG). Optimizing the cost of the active fence involves designing power wasters that achieve the desired level of voltage drop with minimal resource cost. The magnitude of the instantaneous drop should be restricted by the operational threshold of the FPGA (0.9-1.1V). Therefore, what we seek are smaller power wasters. In Fig. 1 we illustrate the power waster proposed in this work. The circuit is a series of inverters driven by the output of a ring oscillator. In an FPGA, the ring oscillator and the inverters can be implemented using LUTs. We refer to both components as logic elements (LE).

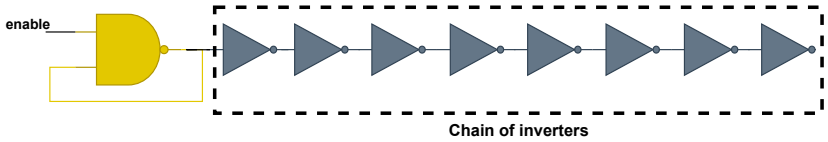



Figure 1: Proposed LUT-based power waster design.

The voltage drop induced by the proposed power waster depends on the number of inverters connected to the ring oscillator. We tested multiple configurations to determine the optimal chain length which maximized the instantaneous voltage drop. We fixed an upper bound of 1,100 LUTs to build an active fence and implemented multiple "copies" of the power waster to fit this area. This upper bound was selected based on the results from [Kra+19]. The NewAE CW305 test board, featuring an AMD-Xilinx Artix-7 FPGA (XC7A100-2TFTG256), was used to implement the active fences. A digital oscilloscope with a sampling rate of 10GSpS was used to observe the power supply of the FPGA. The AMD-Xilinx 2020.2 toolchain was used to generate all the bitstreams and to configure the test board. The results for this experiment are shown in Fig. 2.

In our active fences, a single power waster consists of a ring oscillator and N inverters. Therefore, with one LE the chain length is zero and with 16 LE we have one ring oscillator and 15 chained inverters. The reader should note that the 01 LE configuration is equivalent to the active fences from [Kra+19]. From Fig. 2 we observe that adding inverters results in a higher voltage drop compared to the configuration consisting solely on ring oscillators. However, the relationship between the number of inverters and the magnitude of instantaneous voltage drop appears to be non-linear. For example, the configuration with eight LE has in a larger voltage drop than the configuration with 16 LE. Fig. 2 shows that the maximum voltage drop (0.91 V) was achieved with a configuration of 9 LE (one ring oscillator and eight chained inverters), suggesting this as the optimal setup for active fence applications.

With the same experimental setup we studied the placement impact of the inverter chains of the power wasters. We deployed an active fence with 123 opti-

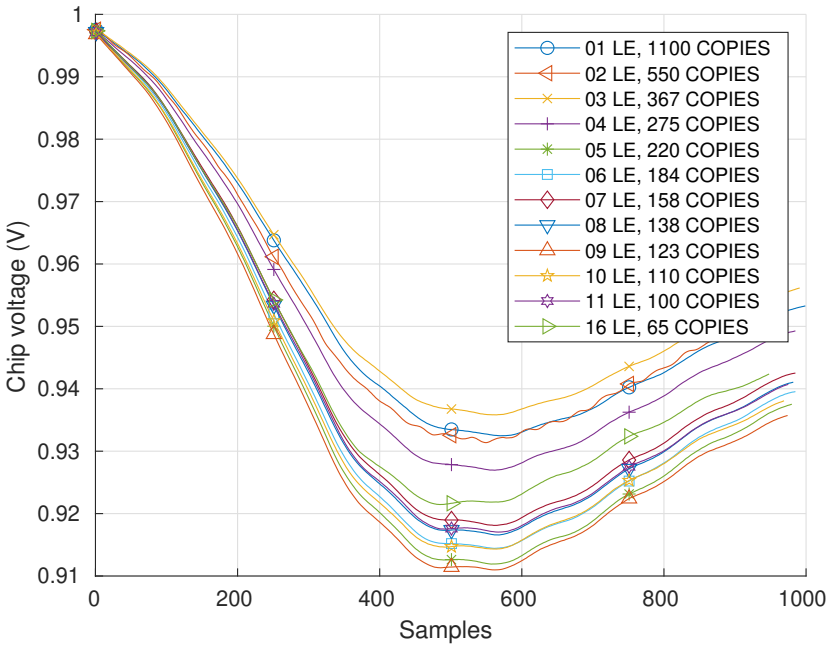



Figure 2: Instantaneous voltage drop generated with active fences of at most 1,100 LUTs. Each fence consists of several copies of the proposed power waster.

mal power wasters (9LE) and compared the produced instantaneous power drop when the circuits are manually placed in the FPGA versus the case where the toolchain handles the placement. Fig. 3 shows the results for this experiment. This analysis demonstrates that the voltage drop is greater when the inverters are manually placed compared to when they are placed by the toolchain. Manual placement allows for optimized spatial distribution and connectivity of the inverters, which may lower parasitic capacitance. A decrease in parasitic capacitance results in an increase in switching activity of the inverters, thus increasing the current draw and consequently the instantaneous voltage drop.

# 4 Experimental evaluation

We devised a second experimental setup to evaluate the effectiveness of the proposed active fence. We used a Trenz TE0802 development board, equipped with an AMD-Xilinx Zynq UltraScale+ SoC FPGA (XCZU2CG-1SBVA484E). The AMD-Xilinx Vivado 2020.2 toolchain was used to generate the FPGA bitstreams and launch applications through Vitis.

The aim of this experiment was to estimate the SNR of a NN design when protected with the proposed countermeasure. The SNR can serve as a metric for estimating the difficulty of an attack by evaluating the strength of the exploitable signal in comparison to the background noise. This estimation was conducted both in the presence and absence of the countermeasure, in order to assess the utility of the proposed design. A handwritten-number recognition

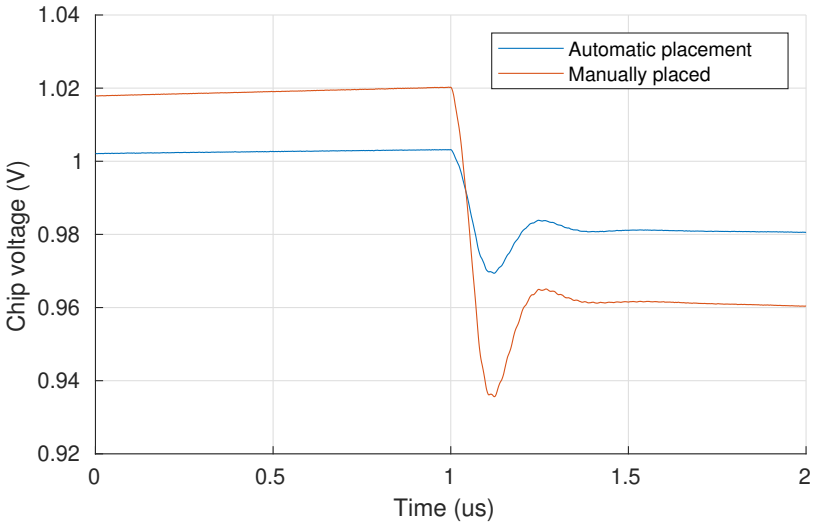



Figure 3: Impact of manual vs. automatic placement on the voltage drop.

application was used as the victim architecture. We employed an internal sensor based on a TDC array with a sampling rate of 200 MSpS to emulate an optimal configuration for RPA [Sch+21]. The active fence was positioned between the victim and the sensor and controlled by a PRNG implemented using a 6-bit linear feedback shift register (LFSR). Fig. 4 illustrates the physical layout of the experimental setup.

We conducted a series of three experiments, each involving the calculation of the SNR. The first experiment served as a baseline, where the countermeasure was disabled. In the second experiment we enabled a RO-based fence as a point of comparison. Finally, in the third experiment we evaluated the performance of the active fence proposed in this work.

## 4.1 NN implementation

We implemented the handwritten-number recognition NN architecture following the approach described in [MSP19]. The implemented network has 400 inputs (receiving a 20x20 pixel image), one hidden layer with 25 nodes, and 10 outputs, following the [400, 25, 10] configuration. It was tested on a subset of the MNIST database and achieved an accuracy of 99%. The NN IP was created using Vivado HLS 2020.2. On the Trenz TE0802 development board it occupies 12% of the total available area with an operating frequency of 50 MHz.

## 4.2 Active fence implementation

Following the approaches described in [Kra+19; Gla+23], the active fence was designed to match the resource utilization of the circuit under protection. In our case this amounts to 5,760 LUTs. The fence was divided into 64 banks, with each bank containing 10 instances of the proposed optimal power waster configuration (9 LE). Within each bank, all wasters were controlled by a single

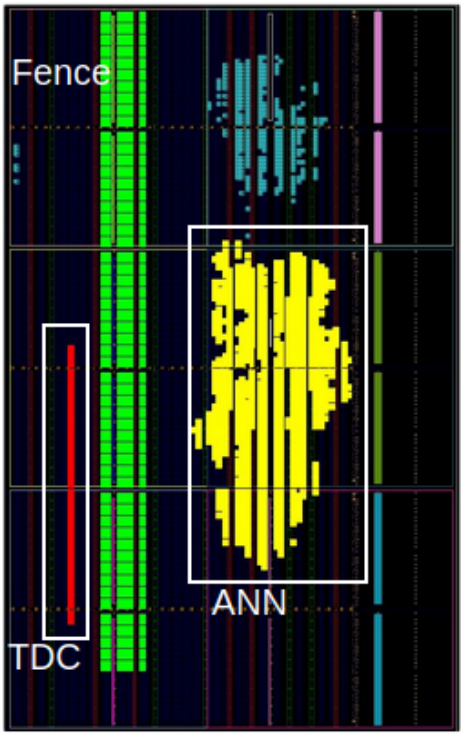



Figure 4: Floor-planning used in the experimental setup

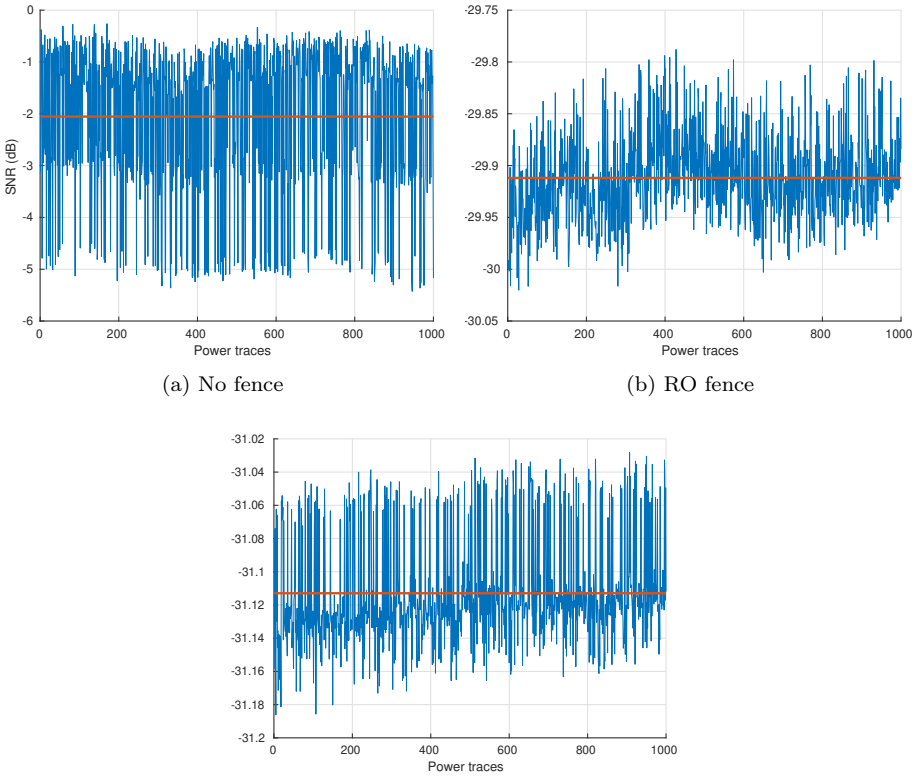
dedicated enable signal. However, each bank could be independently enabled. This was achieved by using a PRNG to generate the enable signals that drove the banks. For the conventional active fence, we follow to the same organization of 64 banks, with each bank containing 90 ROs implemented using NAND gates. This configuration ensures that both fences consume a total of 5,760 LUTs, facilitating a fair comparison of their effectiveness in mitigating power side-channel attacks. The reader should note that this setup allows for a fair comparison, however the proposed active fence could be implemented with less resources.

## 5 Results and Discussion

We evaluated the impact of different active fences on the SNR during the operation of the NN. For each configuration we collected one thousand power traces corresponding to the NN operation. Mathematically, the SNR was derived using the following formula:

$$\mathrm{SNR}_{dB} = 10 \log_{10} \frac{E[S^2]}{E[N^2]}$$

Where $S$ represents the data-dependent leakages and $N$ is the noise. The former was computed as the average of unprotected power traces using the same input. The latter was computed by subtracting $N$ from all the observations. The




(a) No fence

(b) RO fence

(c) LUT fence

Figure 5: SNR plots for the three scenarios

comparative analysis of the SNR values can be observed in Fig. 5.

Without any fence applied the NN implementation shows an average SNR of $-2dB$, indicating a huge amount of leakage of exploitable information from the system. The average SNR was reduced to $-30dB$ with the RO-based fence. The proposed fence configuration managed a further reduction of the SNR, bringing its average down to $-31dB$ when using the same hardware amount as the RO-based fence. The reader should note that a reduction of $1dB$ implies a ratio of $10^{1/10}$ between the state of the art and the proposed fence. A lower value of the SNR indicates an increased complexity for an attack. Therefore, the SNR results show a clear improvement in terms of the performance of the proposed active fence countermeasure with respect to the conventional active fences in terms of the leakage of exploitable information from the victim architecture.

Analyzing the SNR provides valuable insights into the effectiveness of a countermeasure. However, it does not guarantee a perfect mitigation. A significant reduction in SNR, such as dividing the original SNR without the fence by a significant factor, would point to a substantial decrease in exploitable leakage. It hence makes the amount of information an attacker can gain from it considerably reduced, thereby enhancing the security. Nonetheless, it is important to recognize that a lower SNR does not completely eliminate the risk of side-channel attacks. It only makes the attacks harder. A reduced SNR should

hence not be considered the sole indicator of improved security, but it should be accompanied by other security measures and constant monitoring to ensure full protection.

# 6 Conclusions

In this paper, we have proposed a novel power waster design which allows to build lightweight active fences. These circuits can be used as a countermeasure against remote power side-channel attacks on FPGAs. The proposed active fences offer the same instantaneous drop in voltage as conventional fences while reduced the amount of hardware resources in the circuit. The key advantages of this design lie in its optimized resource efficiency and effective noise generation to obfuscate power consumption patterns, thereby mitigating side-channel leakages and reducing the exploitable SNR for potential attacks. Future work could focus on investigating the scalability of this approach for larger FPGA designs and exploring its integration with other security measures for increased protection.

# Acknowledgment

This work has been supported by the French government through the *Agence Nationale de la Recherche* in the framework of the *France 2030* initiative under project ARSENE (ANR-22-PECY-0004).

# References

[Ahm+23] Mahya Morid Ahmadi, Lilas Alrahis, Ozgur Sinanoglu and Muhammad Shafique. "FPGA-Patch: Mitigating remote side-channel attacks on FPGAs using dynamic patch generation". In: *ISLPED*. 2023, pp. 1–6.

[Bre+23] Jakub Breier, Dirmanto Jap, Xiaolu Hou and Shivam Bhasin. "A Desynchronization-Based Countermeasure Against Side-Channel Analysis of Neural Networks". In: *CSCML*. 2023, pp. 296–306. ISBN: 978-3-031-34671-2.

[Bro+24] Manuel Brosch, Matthias Probst, Matthias Glaser and Georg Sigl. "A Masked Hardware Accelerator for Feed-Forward Neural Networks With Fixed-Point Arithmetic". In: *IEEE Trans. Very Large Scale Integr. Syst.* 32.2 (2024), pp. 231–244. DOI: 10.1109/TVLSI.2023.3340553.

[Cha+22] Hervé Chabanne, Jean-Luc Danger, Linda Guiga and Ulrich Kühne. "Parasite: Mitigating Physical Side-Channel Attacks Against Neural Networks". In: *SPACE*. 2022, pp. 148–167. ISBN: 978-3-030-95085-9.

[Che+14] Fei Chen et al. "Enabling FPGAs in the cloud". In: *Computing Frontiers*. 2014. ISBN: 9781450328708. DOI: 10.1145/2597917.2597929. URL: https://doi.org/10.1145/2597917.2597929.

[Dub+21] Anuj Dubey, Afzal Ahmad, Muhammad Adeel Pasha, Rosario Cammarota and Aydin Aysu. "ModuloNET: Neural Networks Meet Modular Arithmetic for Efficient Hardware Masking". In: *IACR Trans. Cryptogr. Hardware Embedded Syst.* 2022.1 (2021), pp. 506–556. DOI: 10.46586/tches.v2022.i1.506-556. URL: https://tches.iacr.org/index.php/TCHES/article/view/9306.

[FBL24] Anis Fellah-Touta, Lilian Bossuet and Carlos Andres Lara-Nino. "A Lightweight Non-Oscillatory Delay-Sensor for Remote Power Analysis". In: *HOST*. 2024, pp. 343–348. DOI: 10.1109/HOST55342.2024.10545353.

[Gla+23] Ognjen Glamočanin, Anđela Kostić, Staša Kostić and Mirjana Stojilović. "Active wire fences for multitenant FPGAs". In: *DDECS*. 2023, pp. 13–20.

[Gna+18] Dennis R. E. Gnad, Sascha Rapp, Jonas Krautter and Mehdi B. Tahoori. "Checking for Electrical Level Security Threats in Bitstreams for Multi-tenant FPGAs". In: *FPT*. 2018, pp. 286–289. DOI: 10.1109/FPT.2018.00055.

[ISW03] Yuval Ishai, Amit Sahai and David Wagner. "Private Circuits: Securing Hardware against Probing Attacks". In: *CRYPTO*. 2003, pp. 463–481. DOI: 10.1007/978-3-540-45146-4_27.

[Kra+19] Jonas Krautter, Dennis R.E. Gnad, Falk Schellenberg, Amir Moradi and Mehdi B. Tahoori. "Active Fences against Voltage-based Side Channels in Multi-Tenant FPGAs". In: *ICCAD*. 2019, pp. 1–8. DOI: 10.1109/ICCAD45719.2019.8942094.

[La+20] Tuan Minh La, Kaspar Matas, Nikola Grunchevski, Khoa Dang Pham and Dirk Koch. "FPGADefender: Malicious Self-Oscillator Scanning for Xilinx UltraScale + FPGAs". In: *ACM Trans. Reconfigurable Technol. Syst.* 13.3 (2020). ISSN: 1936-7406. DOI: 10.1145/3402937.

[LHZ21] Miriam Leeser, Suranga Handagala and Michael Zink. "FPGAs in the Cloud". In: *Comput. Sci. Eng.* 23.6 (2021), pp. 72–76. DOI: 10.1109/MCSE.2021.3127288.

[MGT22] Vincent Meyers, Dennis Gnad and Mehdi Tahoori. "Reverse engineering neural network folding with remote FPGA power analysis". In: *FCCM*. 2022, pp. 1–10.

[Moi+21] Shayan Moini, Shanquan Tian, Daniel Holcomb, Jakub Szefer and Russell Tessier. "Power side-channel attacks on BNN accelerators in remote FPGAs". In: *IEEE J. Emerging Sel. Top. Circuits Syst.* 11.2 (2021), pp. 357–370.

[MSP19] Harsh Mittal, Abhishek Sharma and Thinagaran Perumal. "FPGA Implementation of Handwritten Number Recognition using Artificial Neural Network". In: *GCCE*. 2019, pp. 1010–1011. DOI: 10.1109/GCCE46687.2019.9015236.

[Sch+21] Falk Schellenberg, Dennis R. E. Gnad, Amir Moradi and Mehdi B. Tahoori. "An Inside Job: Remote Power Analysis Attacks on FPGAs". In: *IEEE Des. Test* 38.3 (2021), pp. 58–66. DOI: 10.1109/MDAT.2021.3063306.

[Tia+23] Shanquan Tian, Shayan Moini, Daniel Holcomb, Russell Tessier and Jakub Szefer. "A Practical Remote Power Attack on Machine Learning Accelerators in Cloud FPGAs". In: *DATE*. 2023, pp. 1–6.

[Vey+12] Nicolas Veyrat-Charvillon, Marcel Medwed, Stéphanie Kerckhof and François-Xavier Standaert. "Shuffling against Side-Channel Attacks: A Comprehensive Study with Cautionary Note". In: *ASIACRYPT*. 2012, pp. 740–757. DOI: 10.1007/978-3-642-34961-4_44.

[YCZ23] Xiaobei Yan, Chip Hong Chang and Tianwei Zhang. *Defense against ML-based Power Side-channel Attacks on DNN Accelerators with Adversarial Attacks*. Preprint 2312.04035. arXiv, 2023.

[Zha+21] Yicheng Zhang, Rozhin Yasaei, Hao Chen, Zhou Li and Mohammad Abdullah Al Faruque. "Stealing neural network structure through remote FPGA side-channel analysis". In: *IEEE Trans. Inf. Forensics Secur.* 16 (2021), pp. 4377–4388.

[ZS18] Mark Zhao and G. Edward Suh. "FPGA-Based Remote Power Side-Channel Attacks". In: *S&P*. 2018, pp. 229–244. DOI: 10.1109/SP.2018.00049.