Stealthy Logic Misuse for Power Analysis Attacks in Multi-Tenant FPGAs

– Extended Version* –

Vincent Meyers *, Dennis R. E. Gnad *, Nguyen Minh Dang *,
Falk Schellenberg †, Amir Moradi ‡, and Mehdi B. Tahoori *
*Institute of Computer Engineering, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany
†Max Planck Institute for Security and Privacy (MPI-SP), Bochum, Germany
‡Horst Görtz Institute for IT Security, Ruhr-Universität Bochum (RUB), Bochum, Germany
*{vincent.meyers, dennis.gnad, mehdi.tahoori}@kit.edu , minh.dang@alumni.kit.edu ;
†falk.schellenberg@mpi-sp.org ; ‡amir.moradi@rub.de

***Abstract*—FPGAs have been used in the cloud since several years, as accelerators for various workloads such as machine learning, database processes and security tasks. As for other cloud services, a highly desired feature is virtualization in which multiple tenants can share a single FPGA to increase utilization and by that efficiency. By solely using standard FPGA logic in the untrusted tenant, on-chip logic sensors allow remote power analysis side-channel and covert channel attacks on the victim tenant. However, such sensors are implemented by unusual circuit constructions, such as ring oscillators, delay lines, or unusual interconnect configuration, which might be easily detected by bitstream and/or netlist checking. In this paper, we show that such structural checking methods are not universal solutions as the attacks can make use of any normal circuits, which mean they are "benign-looking" to any checking method. We indeed demonstrate that – without any additional and suspicious implementation constraints – standard circuits intended for legitimate tasks can be misused as a sensor thereby monitoring instantaneous power consumption, and hence conducting key-recovery attacks. This extremely stealthy attack is a threat that can originate from the application layers, i.e. through various high-level synthesis approaches.**

I. Introduction

Power analysis attacks have been lifted from pure physical attacks that need device access to threats that can be deployed remotely through software or firmware. This has been shown both in Field Programmable Gate Arrays (FPGAs) [2–4] as well as Systems on Chip (SoCs) [5–7], and PC systems [8–10]. As FPGAs are getting more widespread adoption in the cloud from companies such as Amazon, Alibaba, and Telekom, a highly desired feature is virtualization and sharing FPGAs between multiple tenants [4, 11–13].

One of the obstacles for virtualizing FPGAs among multiple tenants are powerful side-channel attacks that can be performed by realizing voltage sensors with standard FPGA logic [2–4, 14, 15], as well as fault attacks working in a similar way [16–20]. Most of these circuits have unusual properties that are not found in ordinary digital circuits, such as feeding a clock as a data signal [2, 4, 14, 15], or using combinational loops [3, 16]. Consequently, various attempts of checking bitstreams emerged, which analyze FPGA bitstreams and check the resulting netlist for malicious patterns that might be used for fault or side-channel attacks, before allowing them to be loaded into the device [21, 22].

In this paper, we show how benign logic in existing bitstreams or netlists can be misused as a voltage sensor. Our results show that such sensors are indeed potent enough to be used for standard power analysis attacks. Notably, this way, the sensors are entirely stealthy to any feasible bitstream-checking attempts. This is because the circuit is not altered and still performs its intended meaningful task when not exploited by the attacker. By applying specific data patterns to critical path endpoints at elevated clock rates, they become sensitive to voltage fluctuations. Using post-processing, their results can be used as another type of improvised voltage sensor. Except for their potential timing violations with a secondary clock, measuring voltage in this manner is thus entirely stealthy.

In short, this paper makes the following contributions:

- Misusing existing logic of a normal FPGA design, such that voltage estimates can be measured without the need of previously-used specialized circuits. That makes it harder to detect.
- Post-processing data from path endpoints to estimate voltage fluctuations for performing a successful power analysis attack.

Adversary Model: The adversary model in this paper follows what has been proposed for multi-tenant or cloud FPGAs [2–4, 16, 20, 21]. In this model, the FPGA is utilized as a computing accelerator in a cloud environment, where no local physical access is possible to the board. The FPGA is split in separated regions that are logically isolated, and each user can access only their dedicated region through a hypervisor and partial reconfiguration. Because of the shared Power Distribution Network (PDN) on the whole FPGA, different users are still connected on the electrical level. Through that, an adversary (malicious user) can try to perform power analysis side-channel attacks on a victim user.

We further tighten this model by also assuming the integration of bitstream checking techniques or even manual

* This paper is an extended version of [1]

inspection that would not allow the unusual and conspicuous circuits to be uploaded to the device. Therefore, the attacker is only allowed to use existing circuits that fulfill some benign and meaningful tasks in order to perform any attack.

Paper Outline: The rest of the paper is organized as follows. Section II gives some necessary information on background and related work. Section III explains our methodology of manifesting sensors using the logic of an existing circuit, which is then implemented in a proof-of-concept in the experimental setup explained in Section IV. The results are presented in Section V and its implications are discussed in Section VI while the paper is concluded in Section VII.

II. Background and Related Work

Power analysis side-channel attacks have been an issue for cryptographic circuits, continuously since the introduction of Differential Power Analysis (DPA) in the seminal work by Kocher et al. in 1999 [23]. The power consumption of a circuit can depend on the data and operations it executes, and thereby reveal secret data that is being processed. Typically, differential attacks are performed, which use multiple measurements of an encryption with the same secret key [23]. Since power consists of electrical current and voltage, monitoring just one of them can be sufficient to perform power analysis, as has been used in on-chip attacks inside FPGAs or SoCs [2–5, 7, 14]. Fault attacks can also be performed inside FPGAs, by causing massive voltage fluctuations leading to timing faults [16–20]. However, this paper concentrates on side-channel attacks.

On-chip power analysis attacks are possible, because most integrated circuits are supplied by a common PDN for the entire chip, or a common power supply on the same Printed Circuit Board (PCB) [24]. By that, electrical connections between attacker and victim components of the chip exist. Since PDNs are not ideal, power consumption of individual components will lead to voltage fluctuations, which can be observed through the entire PDN. Thus, when the victim is performing cryptographic operations, a difference in power consumption will also lead to voltage fluctuations in the fraction of the chip controlled by the attacker. If any part of the circuit can be observed by the attacker that is sensitive to voltage fluctuations, there is a chance that power analysis attacks become feasible inside the chip, without any dedicated test or measurement equipment. In FPGAs, it has been shown that such attacks are indeed feasible by using the existing FPGA primitives to create specialized circuits capable of voltage sensing [2, 4, 14, 25].

Various types of circuits have been used to measure voltage, mainly loop-based Ring Oscillators (ROs) and delay-line based Time-to-Digital Converters (TDCs). The working principle for both of them is to make voltage fluctuations visible in a digital circuit, by measuring the resulting change in the circuit delay. For ROs, a combinational loop is used, whose oscillation frequency is inversely proportional to its delay. By counting the amount of oscillations in a fixed time window asynchronously, an estimate of voltage fluctuations can be gained, which has been used for low speed power analysis attacks [3], shown left in Figure 1. A faster approach is to use a TDC, which is a delay line out of buffers through which a signal propagates in a fixed time window [2, 4, 14] shown on the right side of Figure 1. Between the buffers, registers are added, such that different levels of voltage can be sensed. Both for defining the time window as well as for the signal itself, the standard system clock is used. Depending on the – supply voltage-dependent – speed of the buffers, more or less registers show a ‘0’ or ‘1’, correlated to voltage fluctuations. Using the values in these registers are sufficient to perform an on-chip power analysis attack [2, 4, 14]. More recently, delay-based sensors were implemented by just using the delay of the FPGA interconnect itself, instead of using FPGA primitives for the buffers [15].

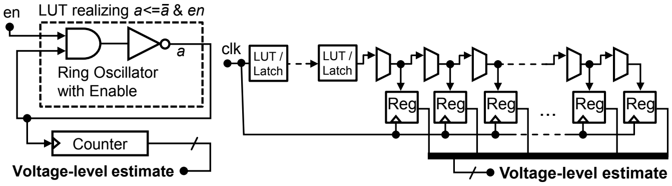


Fig. 1: Specialized circuits used as voltage fluctuation sensors for power anaysis attacks in FPGAs, taken from [21]. **Left:** Sensor based on RO; **Right:** Sensor based on a delay line.

As countermeasures to power analysis attacks, *hiding* and *masking* schemes have been used since the first introduction of these attacks [23, 26], and have also been dedicated towards cloud FPGAs [27, 28]. Another approach which is exclusively proposed for cloud FPGAs is bitstream checking. There, bitstreams/netlists are analyzed for malicious circuits, similarly to how software is checked for viruses [21]. Using that approach, many circuits for power analysis and also fault attacks can be detected thus far [21, 22]. These approaches search for known constructs that improvise voltage sensors. In this paper we will show that this is not sufficient.

III. Re-using Benign Circuits as Sensors

Asynchronous ROs or other combinational loops used so far as SCA sensors [3] can usually be detected by bitstream checking, without rejecting benign circuits [21]. However, in this paper, we show how existing circuits can be repurposed as TDCs, sufficient to perform Correlation Power Analysis (CPA) and extract AES keys. That makes the attack extremely stealthy over previous approaches [2, 4, 14]. Moreover, unlike TDCs, the misused logic does not depend on placement and routing constraints, thereby simplifying the deployment of the sensor.

By operating at timing critical conditions, any path in a circuit can become sensitive to voltage fluctuations. Since each complex circuit has various endpoints, different transistor delays will result in single output sensor bits of different sensitivity to fluctuations in voltage. Under normal situations this behavior can not be exploited, but running the circuit at higher clock rates will. This effectively reduces the timing margin of the circuit, leading to incorrect outputs when the critical path

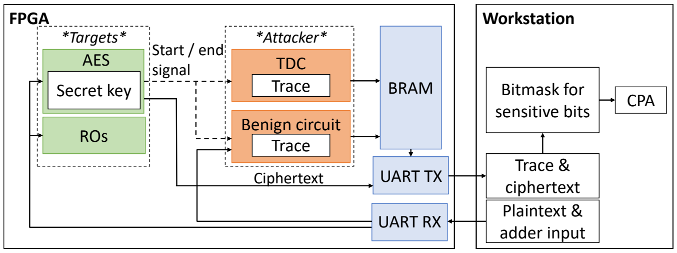


Fig. 2: Overview of the experimental setup

is activated by proper input patterns, and depending on the momentary voltage fluctuations.

In order to observe transitions in critical path endpoints that we want to use as sensor bits, they need to be stimulated with the proper inputs to the circuit. For that, we use two clock cycles. In the first, we reset the logic to a known value, in the second clock cycle the output is then used as a measurement value. Otherwise, a bit that might not have flipped because of insufficient propagation speed would remain set and no switch from 1 to 0 – or vice versa – would be noticed. As a result, the circuit has to alternate between two modes in the consecutive clock cycles, the "reset" mode and "measure" mode. These modes are activated by proper input stimuli of the circuit.

As an example, let A and B be input signals of an n-bit ripple carry adder of an ALU. A is set to the value of $2^n - 1$ and B to 1. The correct result for C is 2^n with the carry out bit set. The carry signal sets all bits of C to 0 and the carry out to 1. By overclocking the circuit the result is read and stored before the carry bit propagates through all stages. Because the carry bit passes through two gates per fulladder, it is susceptible to gate delays. Consequently, voltage fluctuations can be extracted from the result C, depending on how far the carry signal can propagate. Additionally, other signals than the carry signal might be sensitive and lead to bits not toggling, leaving gaps of 0s in higher order bits of C. This can be generalized to other circuits by providing the right input stimuli that will also lead to similar propagation through the circuit.

IV. Experimental Setup

For the implementation and experiments we use a Xilinx Zynq XC7Z020 that contains two ARM CPUs which we do not use, and an Artix-7 FPGA fabric. The FPGA has an external 125MHz reference clock and four Multi-Mode Clock Managers (MMCMs) which can be used to generate clocks.

An overview of our experimental setup is shown in Figure 2. The data transmission to and from the board is realized through a simple UART TX and RX. With the workstation, the UART can be used to send the input to AES and a benign circuit, as well as receive the ciphertext and recorded sums. The AES module encrypts the plaintext, sent from a workstation, while the benign circuit receives two values as the input for two consecutive clock cycles, which it alternates repeatedly during the encryption process. Each result of the benign circuit is saved in BRAM and returned to the workstation as a trace along with the ciphertext. In our experiments, the benign circuit is either an ALU including an 192-bit Adder, or two parallel ISCAS-85 C6288 circuits, each containing a 32-bit multiplier [29].

Furthermore, we expand the design by two more components: an array of 8000 ROs and a TDC sensor. While ROs are used as a controlled surrogate for voltage fluctuation generation, the TDC is used as an established way of measuring differences in voltage levels for side-channel attacks. For each of the two benign circuits (either ALU or C6288), it is possible to operate the implementation in any combination of Target and Attacker without reimplementing the design and possible influences from mapping heuristics. On the workstation, a python script is responsible for transmitting, receiving and storing traces and tuples of plaintexts and ciphertexts. In addition to the raw data, a separate file with traces only containing relevant bits for the CPA is stored.

For our setup, either benign circuit was synthesized for 50 MHz, while in the overclocked mode it is supplied with a 300 MHz clock. The TDC-based sensor is designed to operate at 100 MHz. The AES is synthesized and running at 100 MHz and has a 32-Bit datapath so that four SBoxes are evaluated in parallel. To evaluate whether the captured side-channel data indeed contains enough information for a successful attack we perform textbook CPA using a single bit mask model before the final SBox computation as hypothesis, such as in [2].

On the right side of Figure 3 we show the floorplan of the mapped ALU. All yellow elements are part of the ALU, while the red elements are all the sensitive bits/endpoints. This shows that the circuit is quite scattered, and not as clear as a TDC circuit that is specifically designed for sensing, for which we show the floorplan in the left excerpt in Figure 3.

In Figure 4 we show another floorplan of two mapped C6288 (multiplier) circuits. Similarly, the two C6288 circuits are marked in yellow and sensitive endpoints (w.r.t. voltage fluctuations from ROs) marked in red.

V. Results

Using our experimental setups, we first perform some preliminary experiments looking at the outputs from the ALU while two numbers are added and activating the critical path as described in Section III. We test its performance by activating ROs in the system to cause a strong amount of voltage fluctuations. We look at the result of the ALU adder, and if its output can be used to reconstruct a similar voltage fluctuation trace as those seen with a TDC sensor. Then we proceed to perform CPA on AES using the ALU. We compare these results against a dedicated TDC for voltage sensing. After the initial experiments with our custom ALU, we proceed by analyzing if similar results can be reproduced with the ISCAS-85 C6288 circuit. Furthermore, we evaluate if a single path endpoint of the ALU/C6288 or single bit of the TDC is also sufficient to perform CPA.

A. Preliminary: RO and AES influence on TDC and ALU

As a preliminary experiment, we look at the response of the output of the ALU Adder when all ROs are activated, and the

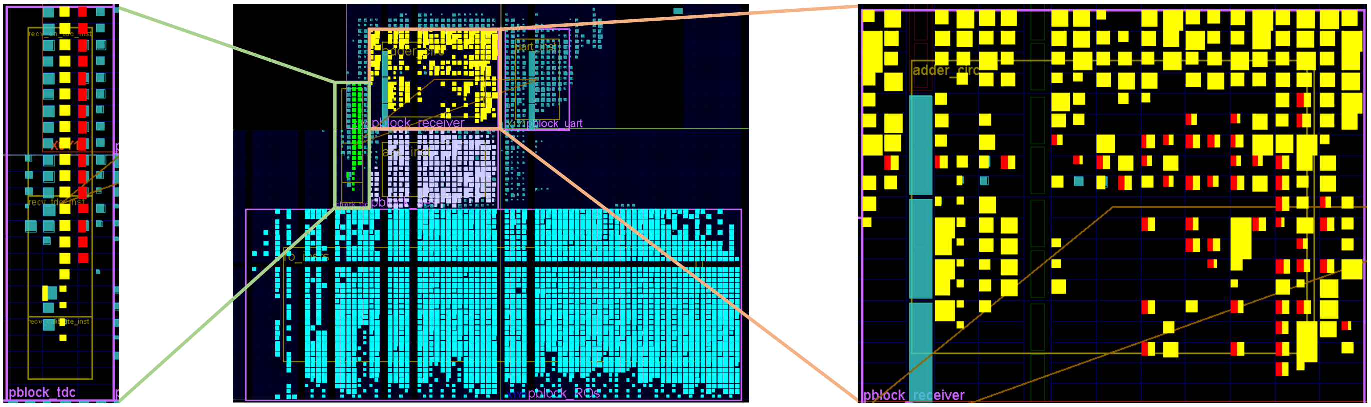


Fig. 3: ALU experimental setup. View of the relevant part of the floorplan. The ALU logic is displayed in yellow, TDC circuit in green, AES in lilac and ROs in light blue. The left and right excerpts are detailed views on the TDC circuit (left) and ALU (right). Yellow parts are the main logic of each circuit; sensitive endpoints marked red.

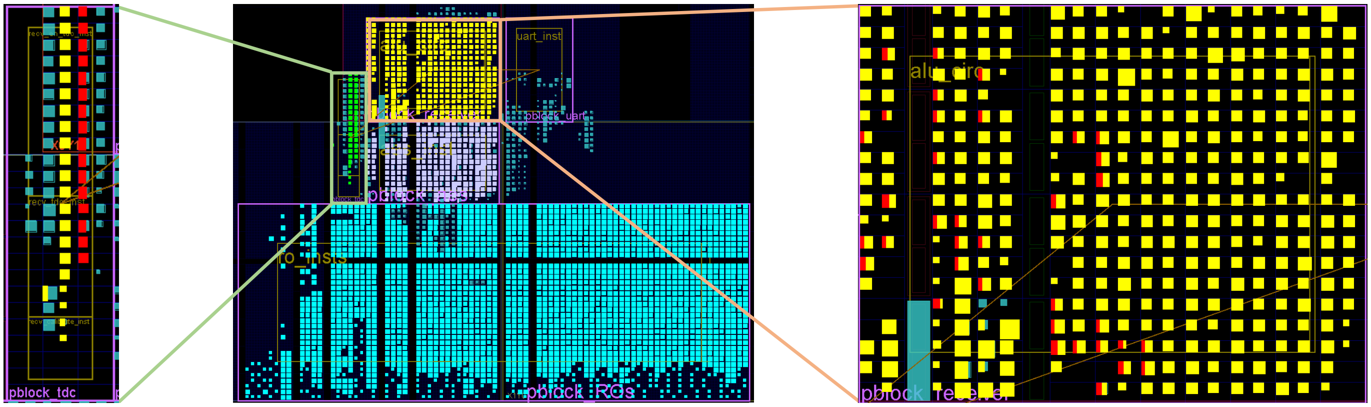


Fig. 4: C6288 experimental setup. View of the relevant part of the floorplan. The C6288 circuit is displayed in yellow, TDC circuit in green, AES in lilac and ROs in light blue. The left and right excerpts are detailed views on the TDC circuit (left) and C6288 (right). Yellow parts are the main logic of each circuit; sensitive endpoints marked red.

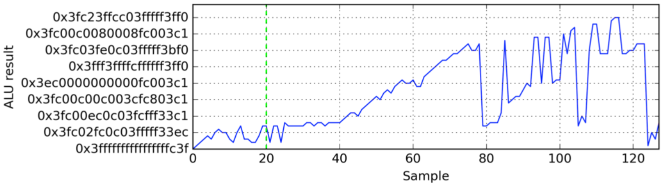


Fig. 5: Absolute value of the toggling ALU bits under influence of 8000 ROs. The ALU runs at 300 MHz and the result of every second clock cycle is shown. The dashed vertical green line indicates the point in time at which the ROs get enabled.

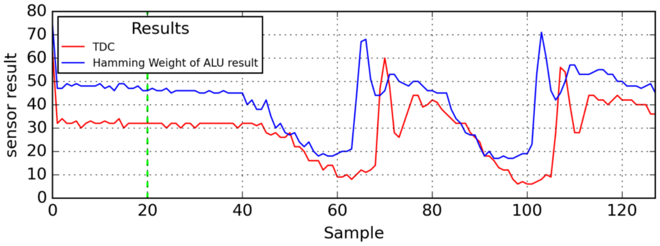


Fig. 6: Influence of 8000 ROs causing two consecutive voltage drops. Results of the TDC sampled at a frequency of 150MHz shown in red. Hamming weight of the toggling sensitive ALU bits is shown in blue. The ALU runs at 300 MHz where the result of every second clock cycle is shown. The time shift between TDC and ALU is due to additional buffer registers inside the circuit. The dashed vertical green line indicates when the ROs get enabled.

ALU is overclocked at 300 MHz, and compare it against the results of the TDC-based sensor of the same design. The ROs are turned on and off in a frequency of 4 MHz, where they are gradually enabled and suddenly disabled. We first look at the raw output of the ALU in Figure 5, which shows a rather random output after the ROs get enabled after around Sample 20.

We post-process this output by selecting all bits of the ALU that fluctuate, and then apply the Hamming weight. That result is compared against the output of the TDC in Figure 6. The ROs are gradually enabled from around Sample 40. From that, the TDC output (**red**) goes down from around 30 to 10 on the Y-axis (TDC), which is indicating increased transistor delays inside the TDC from a voltage drop, while for the post-processed ALU result (**blue**), a similar change is observed with minor offsets in sample or sensor result. When all ROs are disabled at around Sample 70, an overshoot occurs, reducing transistor delay that leads to outputs of 60 and 70. A similar

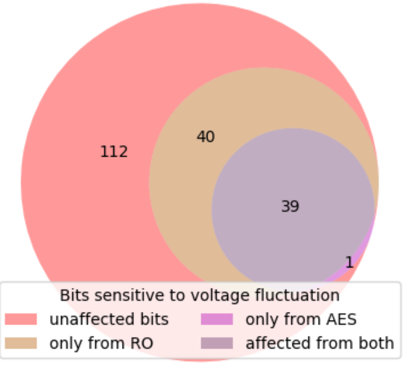


Fig. 7: Shows the amount of ALU bits sensitive to voltage fluctuation from different sources and 112 bits not being affected.

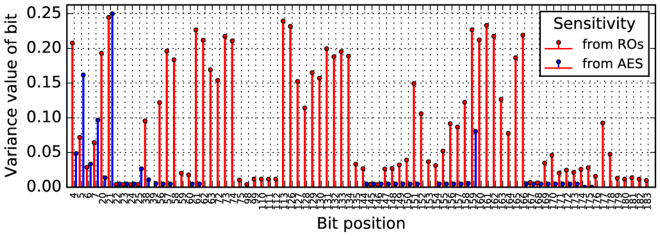


Fig. 8: Variance of each sensitive bit of the ALU under voltage fluctuations from 8000 ROs and AES respectively.

behavior is repeated from around Samples 80 to 120. We thus assume, the ALU can be used in this mode for further experiments, such as performing CPA on the AES. However, its sensitivity to minor fluctuations is not yet known from this experiment.

For the ALU, not all endpoints are relevant for measuring differences in voltage levels and thus retaining their values during the experiments. This observation is displayed in Figure 7. When activating ROs, 79 of the 192 ALU output bits are sensitive to voltage fluctuations.

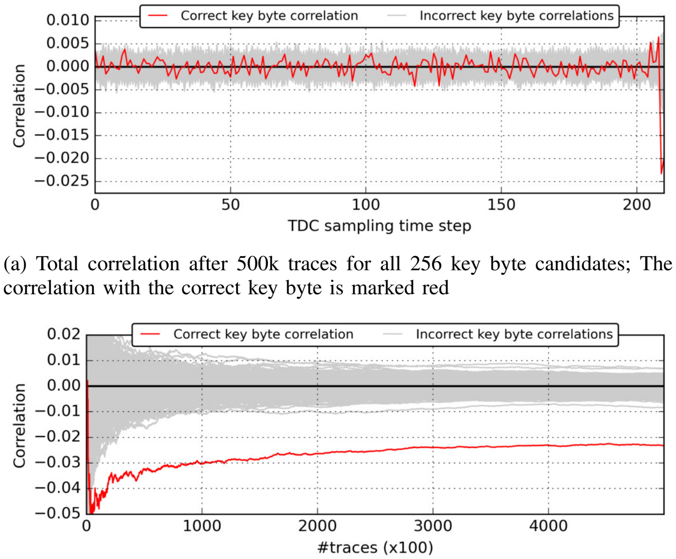
We also repeated this experiment when running AES in a similar way, and looked at the bits affected in the ALU. Most of those bits are a subset of those sensitive to voltage fluctuations from the ROs. On the other hand, when running the AES module in a similar way, only 40 bits toggle, where 39 of them are a subset of those affected by the ROs.

With this information, we can reduce the ALU results to *bits of interest*, shown in detail in Figure 8. The sensitivity of each endpoint can be expressed by its variance. Bits with a higher variance toggle more often and therefore carry more information about the activity on the FPGA. In our implementation Bit 21 of the ALU has the highest variance.

B. Correlation Power Analysis on the AES

Since the preliminary experiment proves the ALU sensitive to voltage fluctuations from ROs, we proceed to use them to perform a power analysis attack on AES, and compare it to results from measurements with a TDC sensor.

In Figure 9 we show a baseline of the CPA results achievable when using the TDC. Because of its linear behavior, just a few hundred traces are needed to clearly distinguish the correct secret key byte (**red**) from all incorrect ones (**gray**).



(a) Total correlation after 500k traces for all 256 key byte candidates; The correlation with the correct key byte is marked red

(b) Correlation progress over 500k traces for all 256 key byte candidates; The correlation with the correct key byte is marked red

Fig. 9: Measurements with a TDC-based sensor at 150 MHz. CPA attack on the 1st bit of the 4th byte of the last secret round key of AES. After the first **few hundred traces**, the correct secret key is already clearly distinguished.

Since the ALU has almost similar performance in replicating the RO measurements of the TDC, we compare how well the ALU can be used to perform CPA, shown in Figure 10. In this regard, it does not perform as fast as the TDC to recover the key, but still recovers the correct secret key byte with (**red**) with about 150k traces.

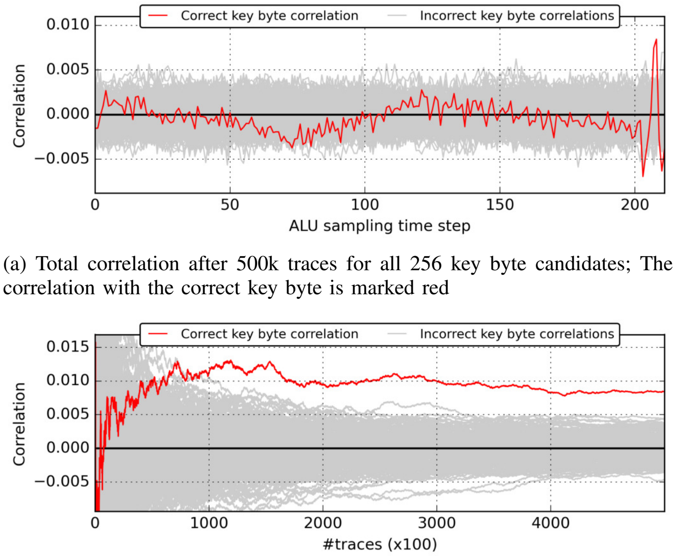
C. Correlation Power Analysis with Single Bits

This attack is much more potent if even single critical path can be misused as a sensor. Thus, here we analyze if even a single bit or path endpoint can be used for CPA, further reducing the preconditions for the attack. For choosing that single path, we use the bit with the highest variance, for the ALU that is shown in Figure 8. For the TDC we use the highest variant bit 32 close to the idle value, and for the ALU, bit 21 (c.f. Figure 8). Please note that this analysis is entirely offline and easily repeated with another device.

For a TDC sensor, using all bits versus only one bit does not make a noticeable difference in key recovery effort, which again just needs a few hundred clock cycles, as shown in Figure 11. We also just need to increase from about 150k to 200k traces for one case when a single endpoint inside the overclocked ALU is used, as shown in Figure 12, which proves that even a single critical path can lead to a security breach. We repeated this for an alternate bit (bit 6) of the ALU, where also just about 150k traces are needed (Figure 13).

D. Results on C6288 ISCAS-85 circuit

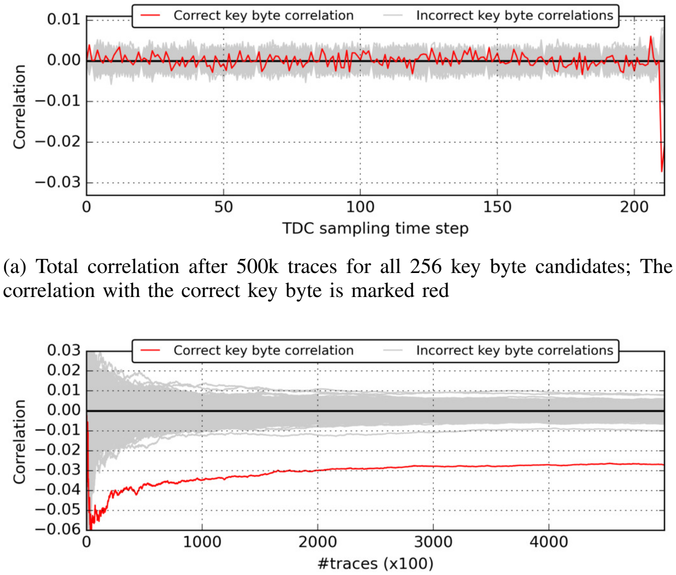
So far we have shown that a voltage sensor can be realized using only standard FPGA logic using an ALU circuit and



(a) Total correlation after 500k traces for all 256 key byte candidates; The correlation with the correct key byte is marked red

(b) Correlation progress over 500k traces for all 256 key byte candidates; The correlation with the correct key byte is marked red

Fig. 10: Measurements with traces derived from the ALU clocked at 300 MHz with an effective sampling rate of 150 MHz. CPA attack on the 1st bit of the 4th byte of the last secret round key of AES. The correct secret key is revealed after about **150k traces**.



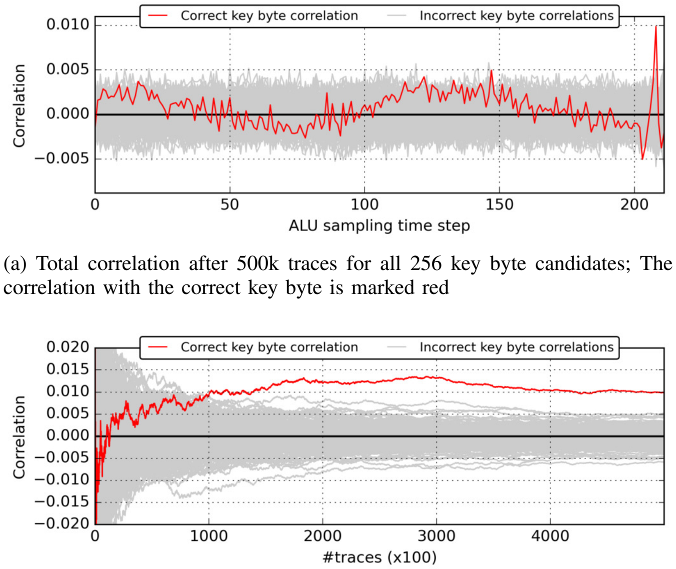
(a) Total correlation after 500k traces for all 256 key byte candidates; The correlation with the correct key byte is marked red

(b) Correlation progress over 500k traces for all 256 key byte candidates; The correlation with the correct key byte is marked red

Fig. 11: Measurements with only a single output (**bit 32**) of a TDC-based sensor at 150 MHz. CPA attack on the 1st bit of the 4th byte of the last secret round key of AES. Even when just using a single bit, the correct secret key is already clearly distinguished after a **few hundred traces**.

compared it against a TDC. We extend our initial experiment by demonstrating how another known benign circuit can be utilized as a hidden voltage sensor by using two instances of the C6288 multiplier circuit, similarly overclocked to 300 MHz.

As for the ALU, we require a reset input pattern between



(a) Total correlation after 500k traces for all 256 key byte candidates; The correlation with the correct key byte is marked red

(b) Correlation progress over 500k traces for all 256 key byte candidates; The correlation with the correct key byte is marked red

Fig. 12: Measurement traces from a single path endpoint (**bit 21**) of the ALU; clocked at 300 MHz with an effective sampling rate of 150 MHz. CPA attack on the 1st bit of the 4th byte of the last secret round key of AES. The correct secret key is revealed after about **200k traces**.

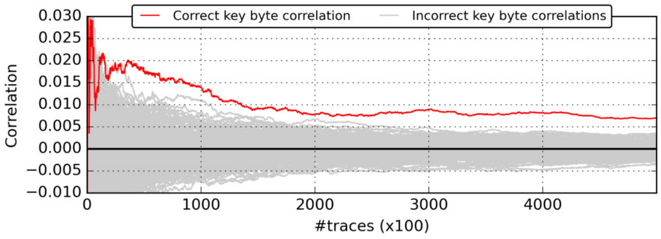


Fig. 13: Correlation progress from a single path endpoint (**bit 6**) of the ALU; clocked at 300 MHz with an effective sampling rate of 150 MHz. CPA attack on the 1st bit of the 4th byte of the last secret round key of AES. The correct secret key is revealed after about **150k traces**.

measurements to observe transitions in the path endpoints, resulting in an effective sampling rate of 150MHz. Each instance has 32 output bits, totaling 64 bits, which will be observed in the following experiments.

Analogue to the experiments with the overclocked ALU adder, we investigate sensitivity of the circuits to voltage fluctuations. Figure 14 shows the influence of 8000 ROs on the results of the C6288 circuits. Here, we observe the same behavior that occurs for the adder sensor and thus, conclude that the overclocked circuit can be utilized for voltage measurements. We find that of the total 64 bits, 49 are sensitive to voltage fluctuation from RO activity.

Next, we investigate the effect of an AES module on the overclocked C6288 circuit. In Figure 15 we further show that 32 of those 49 bits are affected by the less intensive fluctuations resulting from the AES module. All the bits toggling for the AES are also influenced by the ROs. Additionally, we find that 50% of endpoints can potentially be used as sensor

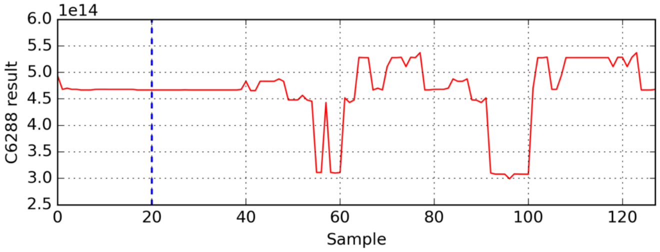


Fig. 14: Absolute value of the toggling C6288 circuit bits under influence of 8000 ROs. The C6288 circuit runs at 300 MHz and the result of every second clock cycle is shown. The dashed vertical blue line indicates the point in time at which the ROs get enabled.

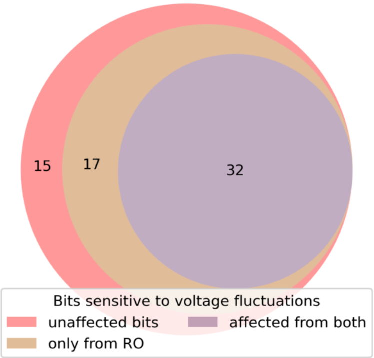


Fig. 15: Shows the amount of C6288 bits sensitive to voltage fluctuation from different sources and 15 bits not being affected. All bits affected by AES are also affected by activity from the ROs.

bits for attacking AES, while with the ALU it is only around 20% of endpoints. We also show the variance of the sensitive bits in Figure 16, allowing to reduce the C6288 results to *bits of interest*, as we did previously for the ALU.

For AES, using only one instance of the circuit did not yield sufficient results even with 500k traces. However, by adding the Hamming weight of the multipliers' results at each time step, we achieve performance similar to the overclocked ALU and successfully retrieve a key byte as shown in Fig. 17. The 32-bit outputs of the multipliers are concatenated into a 64-bit number on which the Hamming weight is applied.

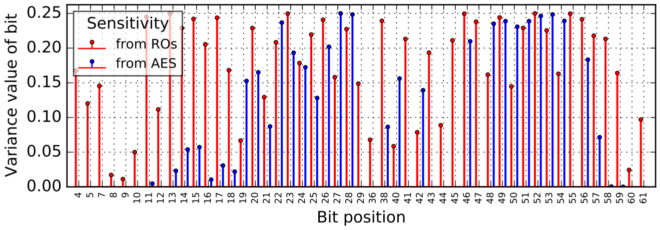
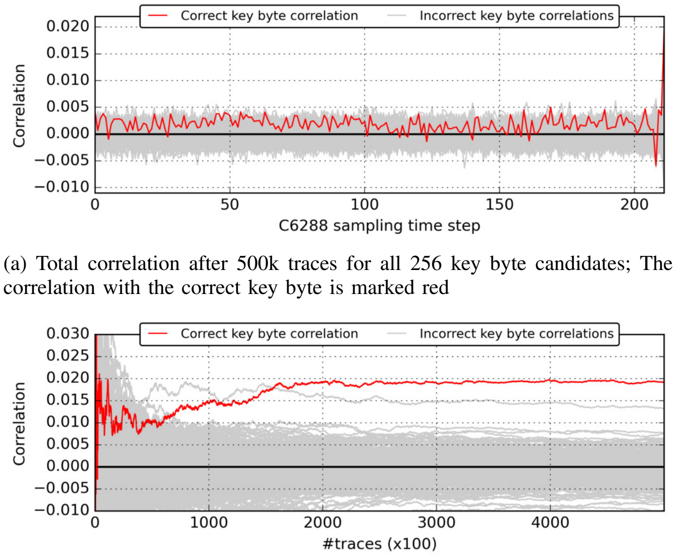


Fig. 16: Variance of each sensitive bit of the C6288 circuit under voltage fluctuations from 8000 ROs and AES respectively.



(a) Total correlation after 500k traces for all 256 key byte candidates; The correlation with the correct key byte is marked red

(b) Correlation progress over 500k traces for all 256 key byte candidates; The correlation with the correct key byte is marked red

Fig. 17: Measurements with traces derived from the C6288 circuit clocked at 300 MHz with an effective sampling rate of 150 MHz. CPA attack on the 1st bit of the 4th byte of the last secret round key of AES. The correct secret key is revealed after about **200k traces**.

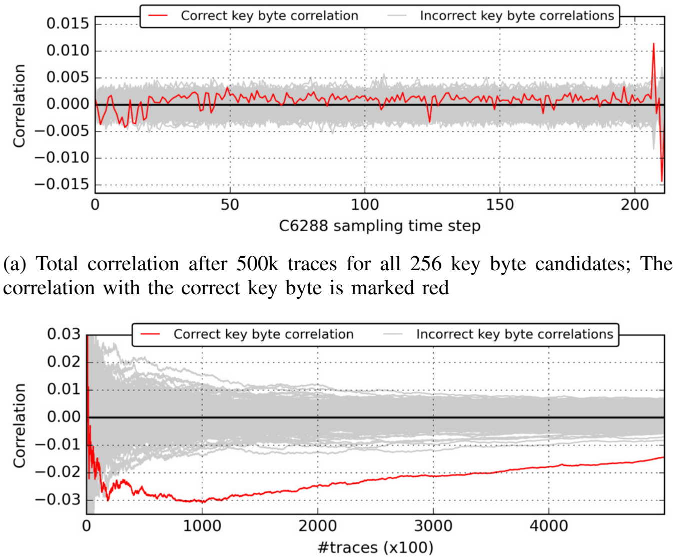
The correct key is retrieved after about 200k traces, which is slightly more than the amount required with the ALU adder, which takes about 150k traces. This can be explained by the number of output bits of the different sensors, which is 192 for the adder and a combined 64 for the multiplier. Therefore, the adder has a higher resolution. The resolution can be increased by adding more instances of the C6288, at the cost of higher resource utilization. Please note, that a full realistic design might be much larger and therefore allow for an even better sensor, with the increased difficulty of finding the right input patterns.

Furthermore, CPA was also performed on the output of a single path endpoint of the C6288 multiplier. Fig.18 displays that when considering only a single bit (bit 28) of the combined output of the two C6288 circuits, the correct key byte can be recovered with about 100k traces. Bit 28 was chosen, considering the variance of individual bits as seen in Fig. 16. This particular bit therefore lead to a slightly better result than combining it with the other bits.

VI. Discussion

The shown results prove that arbitrary arithmetic circuits, such as an ALU or a multiplier can be re-purposed to sense relative voltage fluctuations. By that, power analysis side-channel attacks on cryptographic modules can be performed that previously required special circuits such as TDCs [2, 4, 14], ROs [3], or RDS sensors [15].

Because even a single bit (path endpoint) of the tested circuits can be used for a successful CPA, more complex structures like a carry-chain from a ripple carry adder are not necessary. In our experiments, they merely facilitated the stimuli we had to choose to activate all critical path endpoints. In a more complex circuit, Automatic Test Pattern Generation (ATPG) tools and path delay testing can be used to find such stimuli to activate the critical paths. However, finding the stimuli for a single bit might be rather easy, even if done



(a) Total correlation after 500k traces for all 256 key byte candidates; The correlation with the correct key byte is marked red

(b) Correlation progress over 500k traces for all 256 key byte candidates; The correlation with the correct key byte is marked red

Fig. 18: Measurement traces from a single path endpoint (**bit 28**) of the C6288 circuit; clocked at 300 MHz with an effective sampling rate of 150 MHz. CPA attack on the 1st bit of the 4th byte of the last secret round key of AES. The correct secret key is revealed after about **100k traces**.

manually. For instance, any path longer than those for control flow in a CPU might be used as a sensor.

The approach of bitstream checking uses a strict timing analysis that would indeed detect the approach presented in this paper. For that to work, a mechanism would need to be established, such that a circuit can not select a faster clock than timing analysis suggests. However, we believe that level of strictness is very unrealistic to apply in complex real-world FPGA designs. In a larger design, there are typically many false paths or non-functional paths that are ignored during timing closure, since they have no impact on the correct functionality of the circuit. Even some FPGA vendor IP modules require the use of false path constraints. However, such constraints can potentially hide logic that can be used as sensors for power analysis attacks. As of now, it is thus very hard to prevent the implementation of the presented stealthy sensors, and they can pose a security threat.

We believe that even beyond FPGAs, parts of a circuit can be used for on-chip power analysis attacks, but need documentation or reverse engineering of the respective chip to find the right critical path endpoints. Furthermore, a certain level of clock control might be necessary. We suggest future research can look into this topic.

VII. Conclusion

Using FPGAs as multi-tenant devices is highly interesting to further increase computing efficiency, with their security being an important aspect if deployed in cloud computing platforms. Some of the recent attacks have shown that this operation mode might not be ready for practical use yet, but countermeasures are being worked on. In this paper we have shown that even more stealthy attacks are feasible which will be very hard to detect under normal circumstances. By using any benign-looking circuit under false timing assumptions, some critical path endpoints can be used as sensors sensitive to voltage fluctuations, which this paper proves to be sufficient to perform on-chip power analysis attacks, and key recovery on an AES module. In the future, bitstream checking will need to be more aware of such possible security threats and perform very specific timing checks to prevent side-channel attacks.

References

[1] D. R. E. Gnad, V. Meyers, N. M. Dang *et al.*, "Stealthy Logic Misuse for Power Analysis Attacks in Multi-Tenant FPGAs," in *Design, Automation & Test in Europe (DATE)*, 2021.

[2] F. Schellenberg, D. R. E. Gnad, A. Moradi, and M. B. Tahoori, "An Inside Job: Remote Power Analysis Attacks on FPGAs," in *Design, Automation & Test in Europe (DATE)*. IEEE, 2018.

[3] M. Zhao and G. E. Suh, "FPGA-Based Remote Power Side-Channel Attacks," in *Symposium on Security and Privacy (S&P)*, IEEE, 2018.

[4] O. Glamočanin, L. Coulon, F. Regazzoni, and M. Stojilović, "Are cloud FPGAs really vulnerable to power analysis attacks?" in *Design, Automation & Test in Europe (DATE)*, IEEE, 2020.

[5] D. R. E. Gnad, J. Krautter, and M. B. Tahoori, "Leaky Noise: New Side-Channel Attack Vectors in Mixed-Signal IoT Devices," *Transactions on Cryptographic Hardware and Embedded Systems (TCHES)*, 2019.

[6] C. O'Flynn and A. Dewar, "On-Device Power Analysis Across Hardware Security Domains," *Transactions on Cryptographic Hardware and Embedded Systems (TCHES)*, 2019.

[7] J. Gravellier, J.-M. Dutertre, Y. Teglia, and P. L. Moundi, "Sideline: How delay-lines (may) leak secrets from your soc," in *International Workshop on Constructive Side-Channel Analysis and Secure Design (COSADE)*, Springer, 2021.

[8] M. Lipp, A. Kogler, D. Oswald *et al.*, "PLATYPUS: Software-based power side-channel attacks on x86," in *Symposium on Security and Privacy (S&P)*, IEEE, 2021.

[9] Y. Wang, R. Paccagnella, E. T. He *et al.*, "Hertzbleed: Turning Power Side-Channel Attacks Into Remote Timing Attacks on x86," in *USENIX Security Symposium (USENIX Security)*, 2022.

[10] C. Liu, A. Chakraborty, N. Chawla, and N. Roggel, "Frequency throttling side-channel attack," in *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2022.

[11] S. A. Fahmy, K. Vipin, and S. Shreejith, "Virtualized FPGA Accelerators for Efficient Cloud Computing," in *CloudCom*. IEEE Computer Society, 2015.

[12] A. Khawaja, J. Landgraf, R. Prakash *et al.*, "Sharing, Protection, and Compatibility for Reconfigurable Fabric with AmorphOS," in *USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, 2018.

[13] J. M. Mbongue, A. M.-I. Shuping, P. Bhowmik, and C. Bobda, "Architecture support for FPGA multi-tenancy in the cloud," in *International Conference on Application-specific Systems, Architectures and Processors (ASAP)*, IEEE, 2020.

[14] J. Gravellier, J.-M. Dutertre, Y. Teglia *et al.*, "Remote Side-Channel Attacks on Heterogeneous SoC," in *International Conference on Smart Card Research and Advanced Applications (CARDIS)*, Nov. 2019.

[15] D. Spielmann, O. Glamočanin, and M. Stojilović, "RDS: FPGA Routing Delay Sensors for Effective Remote Power Analysis Attacks," *IACR Transactions on Cryptographic Hardware and Embedded Systems (TCHES)*, Mar. 2023.

[16] J. Krautter, D. R. E. Gnad, and M. B. Tahoori, "FPGAhammer: Remote Voltage Fault Attacks on Shared FPGAs, suitable for

DFA on AES," *IACR Transactions on Cryptographic Hardware and Embedded Systems (TCHES)*, 2018.

[17] T. Sugawara, K. Sakiyama, S. Nashimoto *et al.*, "Oscillator without a Combinatorial Loop and its Threat to FPGA in Data Center," *Electronics Letters*, 2019.

[18] D. Mahmoud and M. Stojilović, "Timing violation induced faults in multi-tenant FPGAs," in *Design, Automation & Test in Europe (DATE)*, IEEE, 2019.

[19] M. M. Alam, S. Tajik, F. Ganji *et al.*, "RAM-Jam: Remote Temperature and Voltage Fault Attack on FPGAs using Memory Collisions," in *Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)*, Aug 2019.

[20] G. Provelengios, D. Holcomb, and R. Tessier, "Characterizing power distribution attacks in multi-user FPGA environments," in *Field Programmable Logic and Applications (FPL)*, IEEE, 2019.

[21] J. Krautter, D. R. E. Gnad, and M. B. Tahoori, "Mitigating Electrical-Level Attacks towards Secure Multi-Tenant FPGAs in the Cloud," *ACM Trans. Reconfigurable Technol. Syst. (TRETS)*, Aug. 2019.

[22] T. M. La, K. Matas, N. Grunchevski *et al.*, "FPGADefender: Malicious Self-Oscillator Scanning for Xilinx UltraScale + FPGAs," *ACM Trans. Reconfigurable Technol. Syst. (TRETS)*, 2020.

[23] P. Kocher, J. Jaffe, and B. Jun, "Differential Power Analysis," in *Advances in Cryptology*. Springer Berlin Heidelberg, 1999.

[24] F. Schellenberg, D. R. E. Gnad, A. Moradi, and M. B. Tahoori, "Remote Inter-Chip Power Analysis Side-Channel Attacks at Board-Level," in *International Conference on Computer-Aided Design (ICCAD)*, Nov 2018.

[25] K. M. Zick, M. Srivastav, W. Zhang, and M. French, "Sensing Nanosecond-scale Voltage Attacks and Natural Transients in FPGAs," in *Symposium on Field Programmable Gate Arrays (FPGA)*. ACM, 2013.

[26] S. Chari, C. S. Jutla, J. R. Rao, and P. Rohatgi, "Towards Sound Approaches to Counteract Power-Analysis Attacks," in *Advances in Cryptology*. Springer Berlin Heidelberg, 1999.

[27] J. Krautter, D. R. E. Gnad, F. Schellenberg *et al.*, "Active Fences against Voltage-based Side Channels in Multi-Tenant FPGAs," in *International Conference on Computer-Aided Design (ICCAD)*, ACM, Nov. 2019.

[28] O. Glamočanin, A. Kostić, S. Kostić, and M. Stojilović, "Active Wire Fences for Multitenant FPGAs," in *International Symposium on Design and Diagnostics of Electronic Circuits and Systems (DDECS)*, IEEE, 2023.

[29] M. C. Hansen, H. Yalcin, and J. P. Hayes, "Unveiling the ISCAS-85 benchmarks: A case study in reverse engineering," *IEEE Design & Test of Computers*, 1999.