Side-Channel Exploitation of DRAM Access Patterns for Fingerprinting FPGA-CPU Environments

Eliott Quéré
Univ Rennes, Inria, CNRS, IRISA
eliott.quere@irisa.fr

Maria Méndez Real
Univ Bretagne-Sud
Lab-STICC - UMR 6285
maria.mendez-real@univ-ubs.fr

Alessandro Palumbo
CentraleSupélec, Inria, CNRS, IRISA
alessandro.palumbo@inria.fr

Lilian Bossuet
Univ Lyon, UJM-Saint-Etienne, CNRS
Laboratoire Hubert Curien UMR 5516 F-42023
lilian.bossuet@univ-st-etienne.fr

Rubén Salvador
CentraleSupélec, Inria, CNRS, IRISA
ruben.salvador@inria.fr

***Abstract*—The widespread adoption of FPGA-accelerated computing in embedded and cloud environments introduces new side-channel threats due to shared hardware resources. This work investigates DRAM access patterns as a leakage source to fingerprint CPU activity, examining both SoC-FPGA and cloud-based co-processor models. In SoC environments, cache-miss-induced DRAM activity generates measurable power fluctuations that can be remotely observed. While previous research has detected these fluctuations using external electromagnetic probes for side-channel-based disassembly, we assess whether embedded FPGA sensors can achieve similar results, enabling attackers to infer CPU operations without physical access. However, in cloud-based co-processor models, where FPGA-CPU interactions occur over PCIe and RDMA, large-scale power management appears to significantly lower the Signal-to-Noise Ratio (SNR), potentially making power side channels more challenging to exploit compared to SoC-FPGAs. Given this uncertainty, we investigate the feasibility of power-based leakage while also exploring timing-based side channels leveraging PCIe contention and RDMA latency variations, which have been shown to reveal workload characteristics. By evaluating both power and timing leakage across these architectures, we comprehensively assess side-channel risks in FPGA-accelerated platforms and emphasize the need for stronger isolation mechanisms.**

***Index Terms*—FPGA security, side-channel attacks, DRAM access patterns, cache-miss leakage, power analysis, PCIe contention, RDMA timing attacks, CPU fingerprinting, embedded sensors, cloud computing security.**

I. Introduction

Field-Programmable Gate Arrays (FPGAs) have become essential components in modern high-performance computing, offering flexible acceleration capabilities across various domains, from machine learning to cryptographic processing. As their adoption expands to cloud environments, new security concerns arise due to the shared infrastructure in which multiple users operate on the same hardware[1]. Unlike traditional embedded FPGA deployments, cloud-based architectures expose shared resources, such as Peripheral Component Interconnect Express (PCIe), DRAM, and power delivery networks (PDNs), to potential adversaries, creating new avenues for remote side-channel attacks.

A key concern in FPGA security is the potential for information leakage through shared memory and communication interfaces. In a SoC-FPGA, where the CPU and FPGA fabric are tightly integrated on the same die, cache misses from the CPU trigger DRAM accesses and generate distinct power fluctuation patterns visible from the FPGA. Prior work [2] has demonstrated that these patterns, visible through electromagnetic (EM) emanations, can be exploited for side-channel disassembly. However, whether these cache-miss patterns can be observed through embedded power sensors[3] such as Time-to-Digital Converters (TDCs)[4] or Routing Delay Sensors (RDS)[5] remains an open question. If confirmed, this approach would introduce a novel method for cache attacks, leveraging embedded sensors to monitor specific DRAM access patterns and fingerprint CPU activity.

Beyond vulnerabilities in SoC-FPGAs, cloud deployments introduce additional complexities. In a co-processor model, where FPGAs communicate with CPUs via PCIe and Remote Direct Memory Access (RDMA), DRAM access is mediated through memory controllers and the I/O Memory Management Unit (IOMMU). Unlike tightly integrated SoC-FPGAs, cloud infrastructures rely on large-scale Power Supply Units (PSUs) feeding both CPUs and FPGAs[6]. This increased scale significantly lowers the SNR of power side channels, as the aggregation of multiple power domains introduces noise that attenuates fine-grained fluctuations. The resulting low-SNR conditions challenge the feasibility of power-based attacks [7], making signal extraction more complex and potentially less exploitable. However, a systematic evaluation is needed to determine whether advanced filtering or statistical techniques could improve leak observability. Alternatively, timing-based side channels that use PCIe contention and RDMA latency

variations offer a more resilient attack vector under these conditions, as previous research has shown their effectiveness in workload inference [8]. We aim to study the exploitability and security implications of these leakages in cloud FPGA environments by evaluating power and timing-based side channels.

Building on these insights, we examine the feasibility of leveraging DRAM access patterns for CPU fingerprinting in various FPGA deployment models. Our contributions are as follows.

- We evaluate how cache-miss-induced DRAM activity generates power fluctuations in the PDN of SoC-FPGAs, assessing its impact as a side-channel leakage source.
- We assess the feasibility of power-based leakage in cloud FPGA co-processor models and, if limited by low SNR, investigate timing-based exploitation via PCIe and RDMA contention.
- We have initiated preliminary experimentation using a TDC sensor on an SoC-FPGA (Zybo Z7-20) to validate our approach against previously identified EM-based cache-miss patterns [2].

By demonstrating that DRAM access patterns leak across both embedded and cloud FPGA deployments, our work underscores the need for stronger isolation mechanisms and security-aware FPGA designs. The insights presented in this paper highlight the urgency of addressing side-channel risks in FPGA-accelerated computing, particularly as these architectures become increasingly central to modern high-performance and cloud-based workloads.

II. Background

The emergence of FPGA-accelerated cloud computing has introduced new security challenges due to the remote threat model, where adversaries lack physical access but can exploit shared infrastructure to mount side-channel attacks. Unlike traditional FPGA security concerns, which primarily focus on bitstream confidentiality [9], hardware Trojans [10], and fault injection attacks [11], cloud-based deployments expose FPGAs to multitenant environments where isolation mechanisms are imperfect. In this context, attackers can leverage shared resources—such as PCIe, DRAM, and PDNs to infer sensitive computations from co-located workloads.

FPGAs in the cloud are typically deployed under an Acceleration-as-a-Service (AaaS) or FPGA-as-a-Service (FaaS) model [12]. These services integrate FPGA boards into cloud environments using different deployment configurations, each with distinct security and resource-sharing implications. This work focuses on two of the three primary deployment setups [13]: the co-processor and the SoC models. Each model employs a distinct resource-sharing arrangement, leading to unique avenues for side-channel exploitation (see Fig. 1). While the co-processor model allows power-based leakage exploitation via shared PSU [6], its effectiveness may be constrained by low SNR. In such cases, timing-based attacks through PCIe contention could offer a more viable alternative for side-channel exploitation. In contrast, with its tighter integration between the FPGA and CPU, the SoC model may exhibit even more pronounced power-based leakage due to localized PDN effects, making power side channels the primary focus of investigation.

A. Co-processor and PCIe Contention-Based Threats

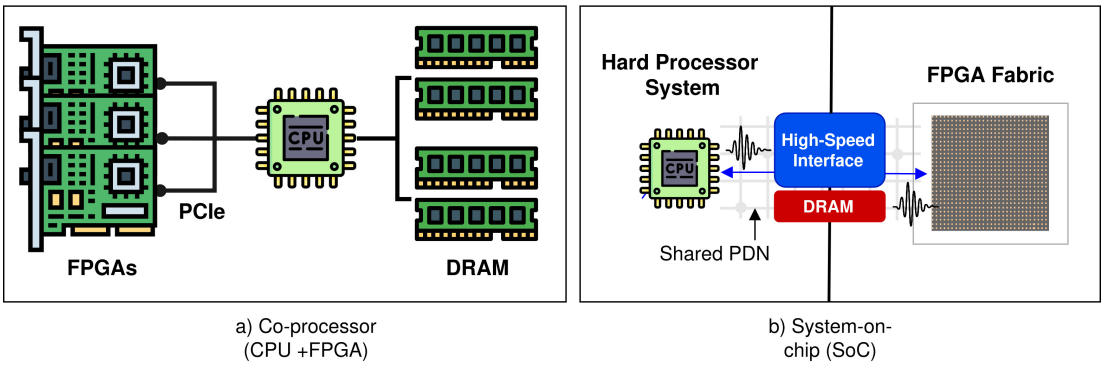
In the co-processor deployment model, an FPGA and CPU share a node and communicate through a dedicated PCIe interface. While this setup ensures high-throughput data transfers, it also introduces shared-resource contention. PCIe is a point-to-point full-duplex communication standard designed for high-speed data transfers among multiple devices. It interconnects devices such as GPUs, FPGAs, NICs, and SSDs through switches and Platform Controller Hubs (PCHs). In high-density configurations, scheduling inefficiencies in these switches can lead to performance degradation and open side-channel vulnerabilities.

Early research demonstrated [14] that variations in PCIe bandwidth, for instance, those influenced by different driver implementations, can disclose the co-location of virtual machines by revealing distinct resource usage patterns. In subsequent work [15], PCIe congestion-induced delays were used as a side channel. In particular, attackers probing an RDMA Network Interface Controller (NIC) that shared a PCIe switch with a GPU managed to infer the GPU's workload on a remote server.

More recent research [8] examines PCIe contention in FPGA-accelerated cloud environments and demonstrates cross-VM attacks leveraging PCIe traffic for covert channels. By analyzing bandwidth signatures, attackers inferred operational details such as cross-NUMA node co-locations and identified interference arising from network and SSD activities. Additionally, the study highlighted how PCIe contention can degrade host-FPGA communication. Another work [16] extends these insights by designing an attack circuit to measure PCIe bandwidth and collect side-channel traces, then training machine learning models to classify victim accelerators based on their unique communication signatures.

Additionally, RDMA presents another attack vector within PCIe-based infrastructures. DMA enables efficient data transfers between peripherals and memory without processor intervention, a key mechanism in high-throughput cloud environments. In FPGA-based systems, frameworks like Intel's Open Programmable Acceleration Engine (OPAE)[17] request contiguous physical memory blocks accessible by the FPGA. At the same time, Intel Virtualization Technology for Directed I/O (VT-d) assigns Input/Output Virtual Addresses (IOVA) to PCIe devices, ensuring the FPGA has a contiguous view of allocated memory. However, PCIe as the underlying interface for DMA also introduces vulnerabilities in RDMA-enabled environments [18].

Specifically, the IOMMU walker mechanism translates memory accesses for PCIe I/O devices, such as Remote Network Interface Cards (RNICs) and GPUs [19]. During intensive GPU workloads, these walkers can become saturated, delaying RDMA operations and increasing response times.



a) Co-processor (CPU +FPGA)

b) System-on-chip (SoC)

Fig. 1. Different FPGA deployment models in cloud computing. a) The co-processor model utilizes PCIe connections to link FPGA boards with CPUs in data centres. b) The SoC model integrates the FPGA and CPU on the same chip die. Adapted from [13].

Such timing variations create a measurable side channel, allowing attackers to infer GPU activity and extract sensitive workload information. On platforms like AWS EC2 F1, DMA operations map DRAM channels to FPGA memory via PCIe using designs such as CL_DRAM_DMA [20]. As previously noted [8], contention-based side channels arising from PCIe traffic and DMA timing can be exploited to profile and fingerprint co-tenant activity in cloud environments.

B. Power Analysis in SoC-FPGA Environments

SoC-FPGA devices integrate microprocessors with FPGA fabric on a single chip. This tight coupling reduces latency and power consumption between the CPU and FPGA but also exposes a shared PDN that can be exploited for power-based side-channel attacks [21]. By monitoring subtle voltage fluctuations in the PDN, adversaries can extract sensitive information, even in the presence of logical isolation or when the SoC architecture suggests physical separation between processing elements.

While traditional power analysis attacks required physical probes, recent advancements have demonstrated the feasibility of remote power monitoring through embedded on-chip sensors [3]. Among the most studied techniques, Time-to-Digital Converters (TDCs) [22] and Ring Oscillators (ROs) [23, 24] provide a means to capture power variations without external access. TDCs measure propagation delays within uniform buffer chains, correlating voltage fluctuations with power consumption, while ROs rely on frequency modulation induced by supply voltage changes. Although ROs are easier to implement, TDCs generally offer higher sensitivity and resolution, enabling key recovery and other advanced attacks.

Recent sensor designs prioritize integration and stealthiness to blend seamlessly into the FPGA fabric. For instance, VITI [25] uses look-up tables (LUTs) in place of dedicated delay elements, while MULT-8 [26] repurposes DSP primitives to reduce detectability. Routing Delay Sensors (RDS) [5] exploit the inherent sensitivity of FPGA routing resources to PDN voltage changes, achieving higher precision with minimal resource overhead.

SideLine [27] demonstrates how existing modules in a SoC, like delay-locked loops (DLLs), typically used to mitigate jitter [28] and synchronize high-speed memory buses, can be repurposed as remote power sensors in high-end SoC memory controllers. While DLL-based sensing is easy to implement via memory-mapped registers, its limited resolution requires many traces for successful side-channel analysis. However, given the widespread integration of DLLs and programmable delay blocks in modern SoCs, their evolving role in high-speed memory interfaces is likely to enhance their power-sensing capabilities, improving sampling rates with next-generation RAM technologies and thereby increasing the risk of remote power side-channel attacks in SoC-FPGA environments.

C. Memory and Cache Attacks

In cloud environments, the increasing demand for peripheral performance places significant strain on the memory subsystem of modern processors. In some cases, DRAM throughput is insufficient to handle data traffic from high-performance peripherals such as network cards. To address this shortfall, modern Intel processors implement Direct Cache Access (DCA) instead of traditional DMA [29], performing I/O operations directly on the Last Level Cache (LLC) to minimize latency and optimize performance. However, this shared LLC, accessible by both the CPU and accelerators like FPGAs, creates opportunities for cache-based side-channel attacks [30].

Previous research has shown that cache attacks exploit timing differences to infer sensitive information [31]. By measuring memory access latencies, adversaries can deduce whether specific addresses are cached or identify the cache level used. Modern hierarchical cache architectures, composed of faster, smaller caches (L1 and L2) and larger, slower caches (LLC), further facilitate these attacks. Such vulnerabilities have been extensively studied in cloud computing [32]. A prominent example is NetCAT [33], which introduced the first network-based PRIME+PROBE attack targeting the LLC of a remote machine, demonstrating vulnerabilities in both cooperative and adversarial network scenarios.

Subsequent studies revealed hardware vulnerabilities in hybrid FPGA- CPU systems, particularly in cloud deployments where FPGAs and CPUs belong to distinct security domains [34]. The memory interfaces, including hardware-level drivers and logical interfaces, can be exploited to infer CPU cache activity, allowing adversaries to deduce memory access patterns and extract sensitive information. For instance, FPGA-based accelerators have been shown able to spy on neighbouring peripherals through the I/O Translation Lookaside Buffer (IOTLB), a cache specialized in DMA address translations [35]. Using Flush+Reload and PRIME+PROBE techniques, malicious FPGAs can snoop on PCIe peripherals such as smart NICs or GPUs, all sharing the same DMA remapping engine.

Parallel work leveraged EM emanations from DRAM accesses as a cache side-channel [2], highlighting that DRAM leakage can reveal cache access patterns. Although physical EM measurements are impractical in typical cloud FPGA settings, power measurements offer a viable alternative for remote leakage exploitation through embedded sensors. This approach adapts the principles of EM-based cache attacks by analyzing voltage fluctuations, aligning with the two deployment scenarios discussed in this article.

In co-processor models, attackers may indirectly exploit PCIe and RDMA to infer cache misses and memory access patterns. In contrast, the shared DRAM and PDN enable more direct monitoring of cache-related power fluctuations in SoC environments, facilitating CPU fingerprinting and cache-based attacks.

III. Threat Model

The growing adoption of FPGA-accelerated computing in embedded and cloud environments introduces new attack surfaces for remote side-channel analysis. We consider an adversary capable of deploying custom FPGA logic and software probes to extract sensitive information from CPU operations by monitoring power fluctuations, timing variations, or memory access patterns. The adversary lacks physical access but can leverage shared hardware resources to mount these attacks.

In SoC-FPGA environments, where the CPU and FPGA fabric are tightly integrated, shared components such as DRAM controllers and the PDN become leakage vectors. Cache-miss-induced DRAM activity can generate observable power variations, which adversaries may exploit despite logical isolation mechanisms. In cloud-based co-processor deployments, where FPGA-CPU interactions occur over PCIe and RDMA, power-based attacks may be limited by low SNR. However, adversaries can exploit timing-based side channels by analyzing PCIe contention and RDMA latency fluctuations to infer CPU workloads.

The primary risk across both models is that logically separated security domains, such as an FPGA user partition and an isolated CPU workload, may still physically share key hardware components. DRAMs, PDNs, and PCIe interfaces create unintended leakage channels, enabling attackers to infer memory access patterns and extract sensitive CPU computations

IV. Ongoing Experimentations and Future Work

Our research investigates side-channel leakage from DRAM access patterns in SoC-FPGA and cloud co-processor FPGA environments. To assess the feasibility of such attacks, we conduct controlled experiments leveraging embedded TDC sensors within the FPGA fabric. Using the SCABox framework [21], we integrate well-calibrated TDC sensors, stream the collected measurements into internal BRAM-based FIFOs, and post-process the extracted traces. On the CPU side, we execute bare-metal workloads, starting with TinyAES [36] to evaluate the sensor's sensitivity to fine-grained power variations. We validate leakage exploitability through Correlation Power Analysis (CPA) and extend our study to general DRAM activity inference, comparing traces from idle execution with those collected during structured memory-intensive workloads. We aim to identify DRAM access signatures that reveal CPU workload characteristics by analyzing these traces in the time and frequency domains. Since SCABox was initially designed for the Zybo Z7-20, used in previous EM-based cache-miss studies [2], we perform our initial experiments on this platform before transitioning to a Xilinx UltraScale+ FPGA, where we will assess the scalability of our approach in high-performance environments and explore potential countermeasures.

Beyond SoC environments, we aim to extend our study to cloud FPGA deployments, where FPGA-CPU interactions occur via PCIe and RDMA. Power-based side channels are more challenging in these settings due to large-scale power management and lower SNR. To address this, we will investigate whether cache-miss-related DRAM access patterns remain observable when memory transactions are mediated through PCIe and RDMA mechanisms. We will also explore timing-based side channels by monitoring latency variations in the IOMMU and memory controllers, as such anomalies could indicate concurrent memory accesses or CPU-driven contention, revealing workload characteristics even in cloud environments. By systematically analyzing both power-based leakage (in SoCs) and timing-based leakage (in co-processor models), we aim to extend cache-miss detection methodologies to the cloud. Ultimately, these findings contribute to a unified framework for side-channel analysis in FPGA-accelerated computing, exposing new security risks and emphasizing the need for stronger isolation mechanisms at both the hardware and system levels to safeguard next-generation heterogeneous computing platforms.

References

[1] C. Jin et al. "Security of Cloud FPGAs: A Survey". In: *CoRR* abs/2005.04867 (2020).

[2] J. Maillard et al. "Cache Side-Channel Attacks Through Electromagnetic Emanations of DRAM Accesses:" in: *Proceedings of the 21st International Conference on Security and Cryptography*. SCITEPRESS - Science and Technology Publications, 2024, pp. 262–273.

[3] F. Schellenberg et al. “An inside job: Remote power analysis attacks on FPGAs”. In: *2018 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. ISSN: 1558-1101. 2018, pp. 1111–1116.

[4] S. Moini et al. “Voltage Sensor Implementations for Remote Power Attacks on FPGAs”. In: *ACM Trans. Reconfigurable Technol. Syst.* 16.1 (2022).

[5] D. Spielmann, O. Glamočanin, and M. Stojilović. “RDS: FPGA Routing Delay Sensors for Effective Remote Power Analysis Attacks”. In: *IACR Transactions on Cryptographic Hardware and Embedded Systems* (2023), pp. 543–567.

[6] I. Giechaskiel, K. B. Rasmussen, and J. Szefer. “C3APSULe: Cross-FPGA Covert-Channel Attacks through Power Supply Unit Leakage”. In: *2020 IEEE Symposium on Security and Privacy (SP)*. 2020, pp. 1728–1741.

[7] M. des Noes. “Distribution of Signal to Noise Ratio and Application to Leakage Detection”. In: *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2024.2 (2024), pp. 384–402.

[8] I. Giechaskiel, S. Tian, and J. Szefer. “Contention-Based Threats Between Single-Tenant Cloud FPGA Instances”. In: *Security of FPGA-Accelerated Cloud Computing Environments*. Ed. by J. Szefer and R. Tessier. Cham: Springer International Publishing, 2024, pp. 137–172.

[9] M. Moraitis. “FPGA Bitstream Modification: Attacks and Countermeasures”. In: *IEEE Access* 11 (2023), pp. 127931–127955.

[10] R. S. Chakraborty et al. “Hardware Trojan Insertion by Direct Modification of FPGA Configuration Bitstream”. In: *IEEE Design & Test* 30.2 (2013), pp. 45–54.

[11] L. Zussa et al. “Power supply glitch induced faults on FPGA: An in-depth analysis of the injection mechanism”. In: *2013 IEEE 19th International On-Line Testing Symposium (IOLTS)*. 2013, pp. 110–115.

[12] C. Bobda et al. “The Future of FPGA Acceleration in Datacenters and the Cloud”. In: *ACM Trans. Reconfigurable Technol. Syst.* 15.3 (2022).

[13] M. Kawser Ahmed et al. “Multi-Tenant Cloud FPGA: A Survey on Security, Trust and Privacy”. In: *ACM Trans. Reconfigurable Technol. Syst.* (2025).

[14] I. Giechaskiel, S. Tian, and J. Szefer. “Cross-VM Covert- and Side-Channel Attacks in Cloud FPGAs”. In: *ACM Trans. Reconfigurable Technol. Syst.* 16.1 (2022).

[15] M. Tan et al. “Invisible Probe: Timing Attacks with PCIe Congestion Side-channel”. In: *2021 IEEE Symposium on Security and Privacy (SP)*. 2021, pp. 322–338.

[16] C. Fang et al. “Gotcha! I Know What You Are Doing on the FPGA Cloud: Fingerprinting Co-Located Cloud FPGA Accelerators via Measuring Communication Links”. In: *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*. CCS ’23. ACM, 2023, pp. 2024–2037.

[17] P. Colangelo et al. “Application of convolutional neural networks on Intel® Xeon® processor with integrated FPGA”. In: *2017 IEEE High Performance Extreme Computing Conference (HPEC)*. 2017, pp. 1–7.

[18] R. Neugebauer et al. “Understanding PCIe performance for end host networking”. In: *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*. SIGCOMM ’18. Budapest, Hungary: Association for Computing Machinery, 2018, pp. 327–341.

[19] H. Kim and J. Hur. “PCIe Side-channel Attack on I/O Device via RDMA-enabled Network Card”. In: *2022 13th International Conference on Information and Communication Technology Convergence (ICTC)*. 2022, pp. 1468–1470.

[20] A. W. Services. *CL_DRAM_DMA Custom Logic Example*. https://github.com/aws/aws-fpga/tree/master/hdk/cl/examples/cl_dram_dma. Accessed: 3 February 2025. 2021.

[21] J. Gravellier et al. “Remote Side-Channel Attacks on Heterogeneous SoC”. In: *Smart Card Research and Advanced Applications*. Cham: Springer International Publishing, 2020, pp. 109–125.

[22] F. Schellenberg et al. “An Inside Job: Remote Power Analysis Attacks on FPGAs”. In: *IEEE Design & Test* 38.3 (2021), pp. 58–66.

[23] M. Zhao and G. E. Suh. “FPGA-Based Remote Power Side-Channel Attacks”. In: *2018 IEEE Symposium on Security and Privacy (SP)*. 2018, pp. 229–244.

[24] J. Gravellier et al. “High-Speed Ring Oscillator based Sensors for Remote Side-Channel Attacks on FPGAs”. In: *2019 International Conference on Reconfigurable Computing and FPGAs (ReConFig 2019)*. Cancun, Mexico, 2019.

[25] B. Udugama et al. “VITI: A Tiny Self-Calibrating Sensor for Power-Variation Measurement in FPGAs”. In: *IACR Transactions on Cryptographic Hardware and Embedded Systems* (2022), pp. 657–678.

[26] A. Fellah-Touta, L. Bossuet, and C. A. Lara-Nino. “A Lightweight Non-Oscillatory Delay-Sensor for Remote Power Analysis”. In: *2024 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*. ISSN: 2765-8406. 2024, pp. 343–348.

[27] J. Gravellier et al. “SideLine: How Delay-Lines (May) Leak Secrets from Your SoC”. In: *Constructive Side-Channel Analysis and Secure Design*. Springer International Publishing, 2021, pp. 3–30.

[28] X. Wang. “Reduction of Power Supply Induced Jitter with Applications to DDR Controllers”. PhD thesis. Carleton University, 2017.

[29] M. Wang, M. Xu, and J. Wu. “Understanding I/O Direct Cache Access Performance for End Host Networking”. In: *Proc. ACM Meas. Anal. Comput. Syst.* 6.1 (2022).

[30] L. Bossuet and E. M. Benhani. “Security Assessment of Heterogeneous SoC-FPGA: On the Practicality of Cache Timing Attacks”. In: *2021 IFIP/IEEE 29th In-*

ternational Conference on Very Large Scale Integration (VLSI-SoC) (2021), pp. 1–6.

[31] Y. Lyu and P. Mishra. "A Survey of Side-Channel Attacks on Caches and Countermeasures". In: *Journal of Hardware and Systems Security* 2.1 (2018), pp. 33–50.

[32] M. S. İnci et al. "Cache Attacks Enable Bulk Key Recovery on the Cloud". In: *Cryptographic Hardware and Embedded Systems – CHES 2016*. Ed. by B. Gierlichs and A. Y. Poschmann. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 368–388.

[33] M. Kurth et al. "NetCAT: Practical Cache Attacks from the Network". In: *2020 IEEE Symposium on Security and Privacy (SP)*. 2020, pp. 20–38.

[34] T. Tiemann et al. "Microarchitectural Vulnerabilities Introduced, Exploited, and Accelerated by Heterogeneous FPGA-CPU Platforms". English. In: *Security of FPGA-Accelerated Cloud Computing Environments*. Security of FPGA-Accelerated Cloud Computing Environments. Springer International Publishing, 2024, pp. 203–237.

[35] T. Tiemann et al. "IOTLB-SC: An Accelerator-Independent Leakage Source in Modern Cloud Systems". In: *Proceedings of the 2023 ACM Asia Conference on Computer and Communications Security*. ASIA CCS '23. Melbourne, VIC, Australia: Association for Computing Machinery, 2023, pp. 827–840.

[36] K. Kokke. *Tiny AES in C*. https://github.com/kokke/tiny-AES-c. Accessed: 3 February 2025. 2018.