Coil-Based Detection and Concurrent Error Correction Against EMFI

An Experimental Case-Study on a Prototype ASIC

Felix Uhle[1], Nicolai Müller[1], Thorben Moos[2], Philipp Günther[3] and Amir Moradi[4]

[1] Horst Görtz Institute for IT Security, Ruhr-Universität Bochum, Bochum, Germany
firstname.lastname@rub.de
[2] UCLouvain, Louvain-la-Neuve, Belgium
firstname.lastname@uclouvain.be
[3] Langer EMV-Technik GmbH, Bannewitz, Germany
firstname.lastname@langer-emv.de
[4] Technische Universität Darmstadt, Darmstadt, Germany
firstname.lastname@tu-darmstadt.de

Abstract. Fault injection attacks target cryptographic primitives implemented in hardware by deliberately inducing faults, leading to abnormal behavior that can be exploited to extract sensitive information. To mitigate these threats, practical and low-overhead countermeasures, whether at the algorithmic or physical level, are essential. However, the real-world effectiveness of these countermeasures remains uncertain, as it is not always clear whether, or to what extent, their underlying security assumptions hold in practice. Therefore, a thorough evaluation under realistic attack scenarios, including recent techniques such as Electromagnetic Fault Injection (EMFI), is crucial. In this work, we demonstrate the resistance of a protected real-world target chip against EMFI. Specifically, our fabricated 65 nm CMOS ASIC employs concurrent error correction based on the techniques described in Impeccable Circuits II as an algorithm-level countermeasure. At the physical level, the chip integrates multiple coils of varying sizes and positions that serve as sensors for electromagnetic fields. We employ a practical and affordable attack setup featuring a commercial fault-injection probe mounted on an XYZ stage for precise positioning over the ASIC. This setup allows us to investigate the effects of various attack parameters, including the probe's position, pulse polarity, and voltage level. Our results highlight that the coils serve as a lightweight and effective countermeasure for the practical detection of EMFI attempts. In contrast, for concurrent error correction, a gap between theory and practice emerges: the protection overhead actually makes such designs more susceptible to EMFI in real-world scenarios.

Keywords: ASIC · Fault Analysis · EMFI · Impeccable Circuits · Coil · Sensor · Error Detection · Error Correction

1 Introduction

Secure and efficient implementations of cryptographic primitives are essential for preserving the confidentiality of sensitive data in emerging technologies such as the Internet of Things and Industry 4.0. In this context, such implementations must be resilient against physical attacks that target the implementation rather than the design of the cryptographic primitive itself. This challenge significantly complicates the development of secure systems. More specifically, without dedicated countermeasures, nearly every cryptographic primitive

becomes vulnerable to physical attacks at the implementation level. Physical attacks primarily fall into two categories: side-channel attacks (SCA) and fault injection attacks (FIA), both of which become particularly powerful when potential adversaries have physical access to the device. Side-Channel Analysis (SCA) exploits unintended information leakage emitted via physical characteristics, including timing behavior [Koc96], power consumption [KJJ99], and electromagnetic radiation [GMO01]. When an adversary can capture side-channel measurements with sufficient precision, statistical analysis can be employed to extract sensitive information. Fault Injection Analysis (FIA) involves deliberately disrupting hardware to induce faulty behavior, which can then be exploited to extract sensitive information [BDL97]. Common methods for inducing faults include clock glitching, which involves shortening the duration of specific clock cycles [ADN+10]; voltage glitching, where transient changes are introduced to the supply voltage [SGD08]; Laser Fault Injection (LFI), which ionizes transistors using focused laser beams [SA02]; and EMFI, which induces parasitic currents by varying the magnetic field near the chip's surface. Recently, EMFI has gained prominence due to its relatively low equipment cost compared to laser-based methods, its ability to induce multiple localized faults, and its applicability without requiring chip decapsulation, making it both *practical* and *highly efficient*. If an adversary successfully injects faults, a range of concrete attacks may become applicable, including the well-known Differential Fault Analysis (DFA) [BS97] and the more recent Statistical Fault Analysis (SFA) [FJLT13] and Statistical Ineffective Fault Analysis (SIFA) [DEK+18]. DFA involves precisely injected faults that disrupt specific intermediate values within a circuit, resulting in altered system outputs. By comparing these faulty outputs to fault-free references, an attacker can infer sensitive information. In contrast, SFA and SIFA relax the requirement for precise fault injection by analyzing the statistical distribution of faults in intermediate values, enabling information extraction even when the fault location or effect is not strictly controlled. To counter FIA, various countermeasures have been proposed, differing in both the abstraction level at which they are applied and whether they aim to detect or correct injected faults. At the physical level, the chip can be protected by adding shielding layers that act as physical barriers to block injected faults. Additionally, dedicated sensors can be implemented to detect physical signs of fault injection attempts and trigger appropriate responses, such as erasing all sensitive information from the chip. Related work includes, among other approaches, the use of various types of sensors to detect indirect effects caused by EMFI [ELH+12, DDRT12, ZDT+14, EBRM16, MNH+16, HBB+16b, KSS+18]. Sensors are also used to continuously monitor the magnetic field near the chip to detect probes used to perform SCA [MFT+14] or SCA and FIA [HHM+17], or to detect hardware trojans [HGM+20, WPZ+24]. Such countermeasures are typically evaluated through experimental testing, where a set of attack vectors is applied in a lab environment to assess the effectiveness of the protection. Notably, there is no formal adversary model to abstractly quantify the security gain provided by shielding or sensing techniques. At the algorithmic level, methods from coding theory can be employed to detect or correct up to a fixed number t of simultaneously injected faults. Specifically, such methods introduce redundancy into the circuit, either by replicating the circuit multiple times or by generating additional parity bits. Faults can then be detected or corrected by evaluating this redundant information, for example, using majority voting or error-detecting [AMR+20] and error-correcting codes [SRM20, BKHL20]. While formal adversary models, such as those proposed in [RSG23, DN23, BFG+24], can abstract the security of algorithmic countermeasures, their relevance and applicability to real-world scenarios remain uncertain. For instance, countermeasures that guarantee the correction of up to t simultaneously injected faults typically assume that an adversary can inject only a very limited number of faults. However, as shown in [BBM+22], even highly localized techniques like LFI can affect multiple transistors with a single shot. This raises concerns about the practical effectiveness of such countermeasures, especially against less precise injection methods

such as EMFI. Hence, even for algorithmic countermeasures with provable security under a formal adversary model, experimental analysis remains essential to verify whether the assumptions of the model align with the capabilities of real-world adversaries. From this perspective, physical and algorithmic countermeasures appear to be complementary. Algorithmic countermeasures can guarantee a certain level of security in a clearly defined adversary model, e.g., the detection of up to t faulted wires. Physical sensors do not engage in formal arguments regarding the security level they offer; rather, they rely on engineering intuition and empirical evaluation to substantiate their efficacy. However, this does not mean that the latter are less effective or important in practice. Depending on how much the true adversary deviates from the abstract model assumed by the algorithmic countermeasure, sensor-based solutions can help to catch faults that would otherwise fall outside the coverage. Approaches to combine the two worlds in a theoretically sound manner include the recent Uniform Random Fault Model [CDM+25]. However, the effectiveness of this combined protection must be validated through experimental analysis.

1.1 State of the Art

1.1.1 EMFI

EMFI is based on the principles of electromagnetic induction, a phenomenon discovered by Michael Faraday in the early 19th century. Electromagnetic induction refers to the generation of an electromagnetic force (emf) in a conductor when it is exposed to a changing magnetic field. This principle is fundamental to the operation of EMFI. Electromagnetic induction can be both emitted and received by a coil. When an electric current flows through a coil, it generates a magnetic field. Conversely, when a coil is exposed to a changing magnetic field, a current is induced by the changing field in that coil. In modern Integrated Circuit (IC), any wire might be susceptible to electromagnetic induction.

1.1.2 Physical countermeasures

Owing to its low cost, non-invasiveness, and precision, EMFI has become a prevalent fault injection technique. Consequently, numerous studies propose physical countermeasures that detect *indirect* effects caused by electromagnetic (EM) fields inside the circuit, rather than the fields themselves. Such sensors demand a thorough and accurate understanding of the on-chip effects induced by EMFI. Most existing solutions assume that EMFI manifests as either clock-tree glitches or Power Delivery Network (PDN) fluctuations, effects that can be detected through the methods described below. A major advantage of these countermeasures focusing on *indirect* effects is that they can be implemented on Field Programmable Gate Arrays (FPGAs) and are not Application-Specific Integrated Circuit (ASIC) exclusive.

Voltage Glitch Detector To the best of our knowledge, the first *indirect* countermeasure specifically targeting EMFI (only evaluated on an FPGA) was proposed in [DDRT12, ZDT+14], adapting a glitch detector originally designed for global clock or voltage faults [ELH+12]. Its core mechanism relies on a *guardian delay* T_{gd}, constrained such that

$$T_{\mathrm{crit}} < T_{\mathrm{gd}} < T_{\mathrm{clk}},$$

where T_{crit} is the maximal propagation time of a signal along the critical path, and T_{clk} is the period time of the clock signal. The detection mechanism employs a logic-gate delay chain (implementing T_{gd}) and a monitoring register, incurring an area overhead of 0.3% per sensor in an Advanced Encryption Standard (AES)-128 implementation. Zussa *et al.* [ZDT+14] do not provide any data regarding the overhead in power consumption. When an EMFI-induced parasitic clock edge shortens T_{clk} to T'_{clk}, the countermeasure triggers if $T'_{\mathrm{clk}} < T_{\mathrm{gd}}$.

This condition arises before T'_{clk} exceeds the critical timing threshold T_{crit}, since $T_{\text{crit}} < T_{\text{gd}}$. Therefore, timing violations caused by parasitic clock edges are effectively prevented by this detector. Conversely, voltage droops slow signal propagation, increasing the effective guardian delay to T'_{gd}. With an unaltered T_{clk}, this yields $T_{\text{clk}} < T'_{\text{gd}}$, enabling detection of certain voltage and clock glitches before a timing violation occurs. However, this countermeasure does not address all possible physical effects. In particular, if T_{crit} increases while T_{gd} and T_{clk} remain unchanged, the condition $T_{\text{gd}} < T_{\text{clk}} < T_{\text{crit}}$ may arise undetected. This scenario can occur when EMFI induces only *localized* effects-as observed in [OGSM15, DLM21]-where the sensor logic (governing T_{gd} and T_{clk}) resides in a region unaffected by EMFI, while the target logic remains vulnerable. More generally, non-trivial bypass cases may exist, underscoring the need for a nuanced understanding of intra-IC physical interactions to ensure the efficacy of such *indirect* countermeasures. A critical limitation of this approach, as noted in [EBRM16], is its reliance on *a priori* knowledge of the protected circuits timing characteristics to tune the delay elements. While post-silicon trimming may mitigate this issue for moderate-scale ICs (e.g., smart cards), albeit at the cost of time-consuming manual calibration, this method becomes impractical for advanced-node System on a Chips (SoCs) due to heightened process variations. The challenge is further exacerbated in designs employing Dynamic Voltage and Frequency Scaling (DVFS), a power-management technique that invalidates fixed-timing assumptions. Fundamentally, the detector's effectiveness hinges on static timing behavior, constraining its broader applicability. To address these limitations, [EBRM16] introduces an alternative *indirect* sensor based on the sampling fault model [OGT+14, OGM17, DLM19, DLM21], detailed in the following paragraph. Another *indirect* sensor, known as a Time-to-Digital Converter (TDC), detects anomalous operating conditions, such as PDN fluctuations caused by EMFI or temperature manipulation, as proposed in [EAD+20]. Its effectiveness against LFI is further demonstrated in [EMV+22]. However, this approach must still be adapted to the operating frequency of the target system. Neither [EAD+20] nor [EMV+22] quantifies power or area overhead.

Sampling Fault Detector The authors of [EBRM16] introduce a countermeasure based on the sampling fault model [OGT+14, OGM17, DLM19, DLM21], which assumes that EMFI disrupts register switching processes, thereby inducing faults. Their proposed solution implements a purely digital countermeasure that, being independent of timing characteristics, can be automatically integrated into any digital circuit. The sensor is claimed to incur low power and area overhead, though no quantitative data is provided. The design employs register pairs that switch on opposite clock edges to complementary values: the first register transitions from 0 to 1 on the rising edge, while its pair transitions from 1 to 0 on the falling edge. A second register pair, initialized with opposite states, covers the remaining transitions ($1 \rightarrow 0$ on rising edges and $0 \rightarrow 1$ on falling edges), thus monitoring all vulnerable switching scenarios predicted by the sampling fault model. However, Nabhan *et al.* [NDR+23] demonstrate that this countermeasure fails to detect all EMFI-induced effects through FPGA-based verification, thereby exposing fundamental limitations of the sampling fault model. This underscores how non-generic, imperfect models may yield inefficient *indirect* sensors under certain conditions.

Ring-Oscillator-Based Detector A Ring Oscillator (RO) offers another approach to mitigate the limitations discussed in Section 1.1.2. It consists of an odd number of inverters connected in a closed loop, oscillating at a frequency determined by the cumulative delay of its components. Fluctuations in the PDN alter inverter propagation delays, resulting in frequency shifts or phase changes. This renders ROs effective for detecting EMFI-induced PDN manipulations when paired with capture logic. However, if an attacker is able to inject a parasitic clock edge (e.g., at low frequencies, as observed in [NDR+23]) without

exceeding the detection threshold, this *indirect* countermeasure loses efficacy.

Several capture logic implementations exist, each with trade-offs. A counter-based approach, as presented in [HAS09], faces two key challenges, as mentioned in [HBB16a]: (1) it requires a wide counter to cover the full operation of the protected hardware and detect minor frequency deviations (high sensitivity), and (2) faults are only detected with a potentially high latency. Alternatively, a Phase Locked Loop (PLL) can monitor the RO's reference clock for frequency changes, as done in [MNH+16], while [HBB+16b] extends this to phase detection. A purely digital capture circuit, as introduced in [HBB16a], avoids PLLs (often scarce or unavailable) using just two registers and an `AND` gate, though it demands manual delay tuning. This design enables multiple instantiations across an IC, improving coverage, and achieves, as reported in [HBB16a], higher detection rates than PLL-based solutions. On the targeted FPGA platform, the design requires only five Look Up Tables (LUTs) and two D-type Flip Flops (DFFs); however, power overhead remains unquantified. The perfect placement of such RO-based sensors and TDC sensors (see Section 1.1.2) (to protect against LFI) is analyzed in [HSCM23, HSM25].

Khairallah *et al.* [KSS+18] quantify the overhead of the RO-based digital-capture sensor in a Plantlet [MAM17] implementation, reporting an increase (+112%) in power consumption to 212% [KSS+18, Table 3]. This penalty is attributed to the ROs's continuous switching activity.

Coil-Based Detector Miura *et al.* [MFT+14] introduced a sensor using coils to detect variations in the EM field surrounding a protected core, originally designed to detect microprobes in SCA attacks. Later work [HHM+17] extended this approach to FIA, particularly EMFI. Both works are implemented and studied on an ASIC platform. However, the sensor's effectiveness relies not only on the coils (routed above the core) but also on auxiliary components: ROs, LC oscillators, and additional logic circuitry beyond the detection unit. The LC oscillators interface with the coils, while the detection logic monitors shifts in oscillation frequency caused by adversarial probes. Thus, this design remains *partially indirect*, as it infers attacks from frequency deviations, like the RO-based detector discussed in Section 1.1.2, induced by changes in the local magnetic field. However, this sensor comes only with an overhead of 9% in power consumption over an AES implementation [HHM+17, Table 1]. On-chip coils have also been explored for hardware Trojan detection [HGM+20, WPZ+24], though these approaches neither evaluate EMFI resilience nor incorporate on-chip detection logic.

Other Physical Countermeasures Beyond sensor-based detection, physical countermeasures can leverage a circuit's physical properties to prevent fault *exploitation* rather than detect injection attempts. For instance, Khairallah *et al.* [KSS+18] introduce a heuristic method that thwarts successful faults by analyzing attack prerequisites (e.g., spatial constraints) and strategically placing components to invalidate these conditions.

Summary To our knowledge, existing physical countermeasures against EMFI detect only *indirect* effects, such as clock glitches, PDN noise, or sampling errors, rather than the EM field itself. This indirect detection offers an advantage: generality across attack vectors. For instance, a voltage glitch detector may identify EMFI-induced faults alongside voltage droop or clock manipulation. However, these approaches exhibit critical trade-offs:

- *Model limitations*: Some solutions (e.g., sampling fault detector) rely on incomplete fault models, risking undetected attack paths [NDR+23].
- *Deployment challenges*: Most sensors remain FPGA-validated, with unproven ASIC efficacy [ZDT+14, EBRM16, HBB+16b, HBB16a, EAD+20] and compatibility issues with modern techniques like DVFS [ZDT+14, EMV+22, EAD+20].

- *Overhead*: The sensors introduce overhead in area, power, and work. Most noticeable is the power overhead of oscillator-based designs, ranging from 9% (coil-LC oscillators [HHM+17]) to 212% (RO-based designs [KSS+18]), while others require extensive calibration (overhead in deployment/work)[ZDT+14, EAD+20, EMV+22].

These trade-offs motivate our approach: a sensor that *directly* measures the EM field underlying EMFI, avoiding the limitations of indirect detection methods.

1.1.3 Algorithmic Countermeasures

Algorithmic countermeasures with the ability to detect or correct faults in a cryptographic implementation often rely on redundant information. These range from straightforward techniques, such as running parallel cipher instances with deviation checks or majority voting, to more sophisticated approaches based on error-detecting and error-correcting codes. Specifically, binary linear codes are applied to encode a message $X \in \mathbb{F}_2^k$ with message size k into a codeword $Y \in C$ with $C \subset \mathbb{F}_2^n$ codeword size n, while a so-called $[n, k, d]$-code with *minimum distance* $d = \min_{\forall Y \in C} \mathsf{HW}(Y)$[1] can detect all faulty codewords with less than d faulty bits, while all faulty codewords with less than $\frac{d}{2}$ faulty bits can be corrected. To protect a cryptographic primitive against SIFA, error-correction methods are necessary. The approach proposed in *Impeccable Circuits II* employs binary, linear, and systematic codes that embed the message into a codeword along with $n - k$ parity bits, allowing the correction of a fixed number of simultaneously injected faults, even in the presence of fault propagation within the circuit [SRM20]. The required changes to transform an unprotected circuit into a circuit protected by error-correcting codes are as follows:

- All primary inputs that are not control signals are grouped into k-bit chunks and encoded into their corresponding codewords. Each control signal is treated as a separate k-bit chunk by padding it with $k - 1$ leading zeros and is likewise encoded into its respective codeword.
- The round function is modified to process the codewords instead of the messages, and the propagation of faults through it is limited in a way that making t bits faulty inside the round function results in at most t faulty output bits.
- Moreover, at least one correction point is integrated into the circuit's data flow to correct faulty codewords. Each correction point corrects the circuit's state using syndrome decoding. To enable parallel correction, multiple syndrome decoders, each capable of correcting a single codeword, are instantiated. Importantly, all decoders are implemented to ensure that fault propagation remains limited.

To the best of our knowledge, no practical evaluation of Impeccable Circuits II has been conducted on either ASIC or FPGA platforms. However, Bartkewitz *et al.* [BBM+22] conducted an experimental evaluation of *Impeccable Circuits*, a scheme that employs $[n, k, d]$-codes for error detection. In their work, multiple protected instances of Skinny [BJK+16] were fabricated as ASICs, and the security of these implementations was assessed under LFI attacks to quantify the achieved level of protection. It has been shown that codes with non-injective mappings from messages to codewords do not meet the required security level. Therefore, in this work, we exclusively consider codes with injective mappings.

1.2 Our Contributions

This work presents an experimental security analysis of a real-world 65 nm CMOS ASIC subjected to EMFI. The ASIC integrates multiple AES architectures, including variants

[1] $\mathsf{HW}(Y)$ denotes the Hamming weight of codeword Y.

protected with Concurrent Error Correction (CEC) schemes based on Impeccable Circuits II [SRM20], as well as a coil-based sensor matrix for physical error detection. Our analysis yields four novel key contributions:

- A *low-overhead coil sensor* (evaluated on-silicon) that detects EMFI *directly* via electromagnetic induction, unlike prior indirect-detection approaches (Section 1.1.2).
- The *first silicon-validated evaluation* of AES implementations protected with Impeccable Circuits II [SRM20], conducted on a 65 nm CMOS ASIC under realistic EMFI conditions.
- A quantitative assessment of *hybrid countermeasures*, showing that combining physical and algorithmic protections offers no synergistic security benefits.
- Experimental evidence that theoretical fault models might *fail to accurately predict* real-world adversarial effectiveness.

Through this analysis, we demonstrate that a distributed network of coils that vary in size and position across the ASIC provides effective physical detection of EMFI. Our results reveal two findings:

- Even a minimal coil subset (with low area overhead) detects a large share of EMFI attempts, though perfect detection remains elusive.
- Algorithmic countermeasures designed to correct up to two simultaneous faults are readily bypassed via EMFI, with protected designs exhibiting *reduced* security due to increased circuit complexity. This exposes a fundamental disconnect between theoretical protections and practical outcomes.

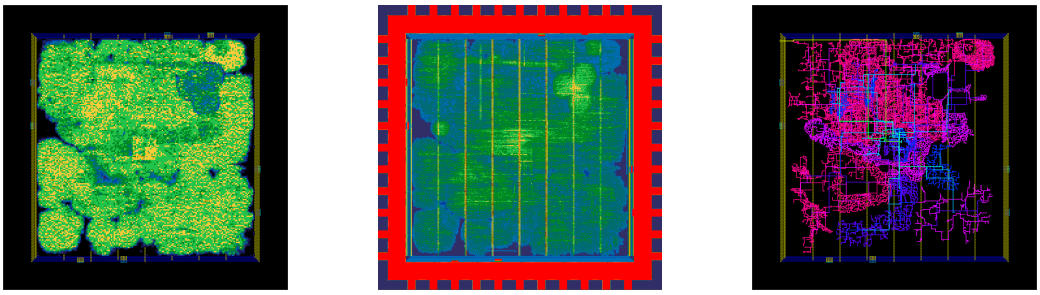
2 ASIC

The target ASIC is manufactured in 65 nm Complementary Metal-Oxide-Semiconductor (CMOS) technology by a leading foundry. The designed area of the ASIC without sealring is 1968.0 μm $\times$ 1967.8 μm. The outer pad ring carrying the bond pads is 71 μm wide and the inner pad ring carrying the IO (Input/Output) cells is 120 μm wide. The standard cell area is 1496.00 μm $\times$ 1496.13 μm large. The technology option we used features a maximum of 9 metal layers (M1 to M9), but in our design, the uppermost layer only contains a logo and no functional routing. Between the standard cell area and the inner pad ring, four power rings (two for V_{DD} and two for GND) of 6 μm width each are routed on M7 in horizontal and M8 in vertical direction. The power distribution is also supported by 7 vertical power straps on M8 and supplied through a Core-V_{DD} and a Core-GND cell, respectively, on each side of the IO ring. In total, the ASIC features 44 bond pads, 11 on each side (left, right, top, and bottom), with 7 on each side being logical IO cells and 4 being power delivery cells for the core and the IO ring, respectively. 11 of the 28 logical IOs are direct digital outputs of the coil-based EMFI detection countermeasure described in the next subsection. With this pinout, our ASIC design perfectly fits the common JLCC44 package we have chosen. When a die is bonded within this package for experimental analysis, approximately the outer 71 μm on each side are occupied by bond wires, leaving a maximum area of 1897.00 μm $\times$ 1896.80 μm that can be scanned with a probe from the frontside (in reality less, as the probe has its own diameter and additional safety margins are advisable to not make accidental connections). Figure 1 contains additional layout information in the form of (heat) maps, showing the cell density, the wire utilization, and the clock-tree distribution over the chip area. The ASIC contains 15 different cryptographic cipher cores, 3 of which will be analyzed in this work, and has been implemented for a clock frequency of $\geq$ 50 MHz under worst-case operating

Table 1: Post-layout implementation figures for the targeted AES designs.

Target	Cell Count #			Area [GE]			Power	Delay
	Total	Comb.	Sequ.	Total	Comb.	Sequ.	[mW]	[ns]
$\mathcal{T}_1$	2405	1005	274	3966.00	2203.00	1763.00	0.0371	4.32
$\mathcal{T}_2$	17763	15338	825	34523.00	29143.50	5379.50	0.0942	7.09
$\mathcal{T}_3$	78763	77018	607	158980.00	154662.00	4318.00	0.1651	8.03

conditions (high temperature, low voltage). Under typical operating conditions, higher clock frequencies are expected to be supported. Configuration registers in the top-level module can be used to select an execution profile. All cores, controllers, data input and output registers, and local parts of the clock tree are clock-gated to be able to uniquely activate individual components without being affected by undesired noise from neighboring units.



(a) Cell Density Map (b) Wire Utilization Map (c) 9-Level Clock Tree

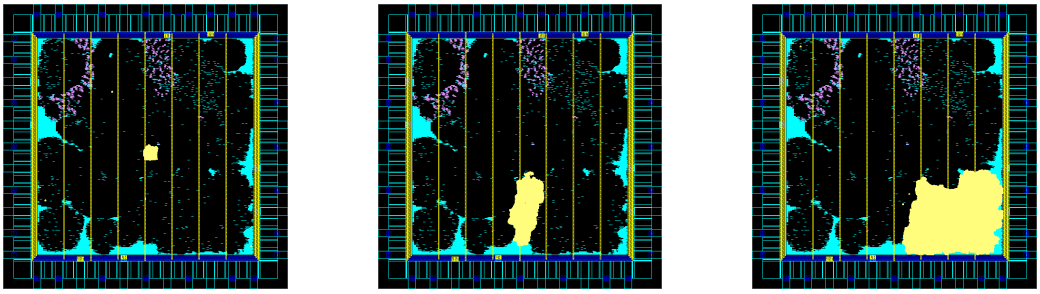
Figure 1: Heatmaps showing cell density, wire utilization and the full clock tree.

2.1 Algorithmically Protected Designs

The ASIC integrates two AES cores, each protected by the algorithmic countermeasure described in Section 1.1.3. All cores were automatically generated using the AGEFA tool, which transforms an unprotected, annotated gate-level Verilog netlist, along with the code specification, such as message size k and target minimum distance d, into an algorithmically protected design [MM24]. We selected the codes based on the results reported in [MM24], choosing designs that exhibited smaller circuit sizes compared to those using alternative code specifications. These codes are lexicographic codes [CS86, Con90], which AGEFA can automatically determine using a greedy algorithm [MM24, Algorithm 3]. Further, the following settings were applied:

- We configured AGEFA to select the code C with the minimum parity size by disabling the code optimization procedure described in [MM24, Section 3.2.4]. In our case, this resulted in slightly smaller circuit sizes compared to an equivalent design using an optimized code. We obtain a $[3, 1, 3]$-code capable of correcting a single fault and a $[16, 8, 5]$-code that can correct up to two simultaneously injected faults.

- We decomposed the circuit into a linear and a non-linear part and instructed AGEFA to insert two correction points, protecting each part separately. This decomposition reduces the overhead required to limit fault propagation in both parts, leading to slightly smaller circuit sizes compared to an equivalent design without decomposition.



(a) Target $\mathcal{T}_1$. (b) Target $\mathcal{T}_2$. (c) Target $\mathcal{T}_3$.

Figure 2: Floorplan locations of the different byte-serial AES cores analyzed in this work marked in yellow. The AGEFA-generated cores with concurrent error correction are parametrized with minimum distance d and message size k.

The approximate physical locations of the targeted AES cores are depicted in Figure 2. Post-layout implementation details are presented in Table 1. Cell counts and area are given both in total and separately for combinational logic and sequential cells. Power is estimated at 10 MHz operation, and delay is estimated at typical operating conditions (room temperature, nominal voltage). Overall, it can be observed that the error detection countermeasures come at a very significant overhead in the combinational logic area, in particular more than one up to almost two orders of magnitude. The increased combinational logic overhead also leads to a significant rise in power consumption: $\mathcal{T}_2$ consumes 254% the power of $\mathcal{T}_1$, whereas $\mathcal{T}_3$ consumes 445% that of $\mathcal{T}_1$.

2.1.1 Target Designs

In this work, we present the evaluation results for three distinct targets, each offering different levels of protection in theoretical models such as the f-fault model [IPSW06] and the (general) random-fault model [DN23, BFG+24]. Table 2 gives an overview of the implemented targets $\mathcal{T}$. The corresponding physical properties of the targets are given in Table 1. In the f-fault model [IPSW06], the first-order protected ($f = 1$) target

Table 2: Overview of evaluated AES designs.

Target	**Protection Order** (f-fault model)	**Description**
$\mathcal{T}_1$	$f = 0$	Unprotected AES
$\mathcal{T}_2$	$f = 1$	AGEFA AES $[3, 1, 3]$-code
$\mathcal{T}_3$	$f = 2$	AGEFA AES $[16, 8, 5]$-code

$\mathcal{T}_2$ offers more protection than the unprotected target $\mathcal{T}_1$ but less protection than the second-order protected ($f = 2$) target $\mathcal{T}_3$.

2.2 Physical EMFI Countermeasure – Coils

We have implemented a total of 11 different EMFI detection coils on 3 different metal layers: M2, M4, and M7. The locations, sizes, and shapes of the 11 coils are depicted in Figure 3. Each of the three metal layers considered here has a thickness of 0.22 μm, and the coils are routed with a width of 0.1 μm. The height from substrate is 0.945 μm, 1.735 μm, and 2.92 μm for M2, M4, and M7, respectively. Figure 4 shows a schematic of the working principle of the coil-based EMFI detection countermeasure. Essentially, one end of each coil is connected to logical zero (GND) via a tie-low cell. The other end is

connected to the clock input of three flip-flops, one with low threshold voltage, one with standard threshold voltage and one with high threshold voltage. The input pin (D) of the flip-flop is connected to logical one (V_{DD}) via a tie-high cell. A multiplexer can then be used to decide which of the flip-flop outputs (Q) is read. Once a changing magnetic field induces a voltage in the conducting metal coil, the flip-flops, depending on whether their threshold voltage is overcome, may react to this apparent logical change in input signal by saving the value of signal D (constant logical one) to output Q. Hence, if one of the flip-flops contains a logical one, it means that an electromagnetic induction event has been detected since its last reset to logical zero. The flip-flops obviously require an asynchronous reset signal (RST); otherwise, their content cannot be set back to logical zero without induced current. After a reset, the coil is prepared to detect the next potential induction event. The idea behind the countermeasure is that an EMFI probe exhibits a strong electromagnetic impulse, which is expected to induce sufficient voltage for detection in the coils, depending on location, size, and polarity. We comprehensively test this hypothesis in this work.

A potential issue with sensors that directly detect electromagnetic fields is the occurrence of false positives, i.e., the sensor incorrectly identifies a fault attempt when no EMFI is actually taking place. While a comprehensive and sound study to definitively prove the absence of false positives is beyond the scope of this work, we conduct preliminary experiments.

2.2.1 Overhead

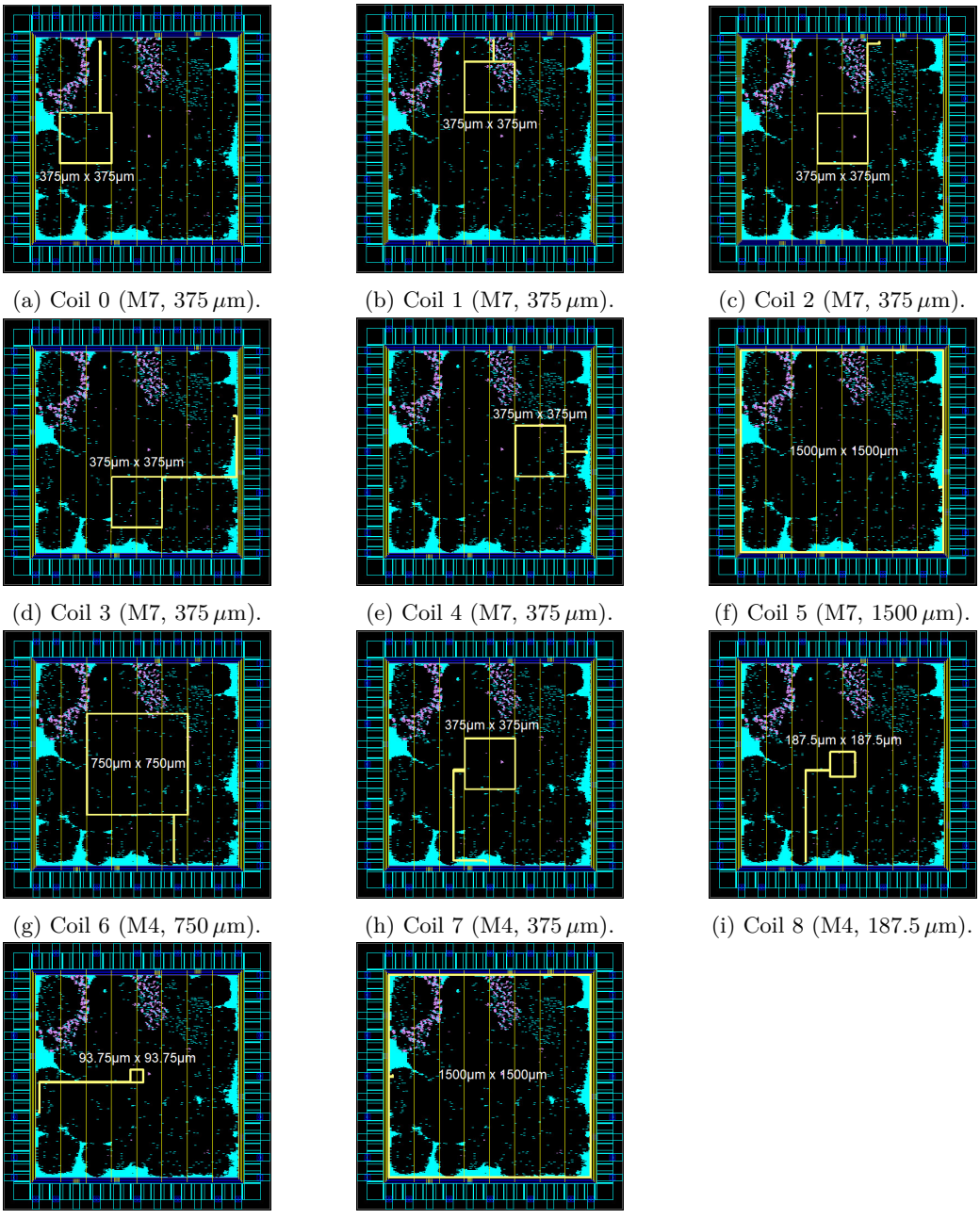
The proposed sensor introduces minimal logical gate overhead, requiring just three flip-flops and a multiplexer (Figure 4). The multiplexer is included only for countermeasure evaluation and can be replaced or omitted based on the desired fault-response mechanism. Routing overhead varies depending on the coil's implementation layer and size. For example:

- Coil 5 (Figure 3f), or Coil 10 (Figure 3k) introduce *negligible overhead*, as their traces are routed in less congested regions of the layout.
- Central coils on lower metal layers, e.g., Coil 7 (Figure 3h), increase routing complexity due to higher wire density in core areas of the IC.

The sensor's power overhead is equally minimal. A single coil (including all detection logic) consumes just 1.92×10^{-5} mW, while the complete 11-coil implementation requires 2.11×10^{-4} mW. These figures represent only 0.052% and 0.569% of $\mathcal{T}_1$'s power consumption (the most power-efficient of our three evaluated targets (Table 1)). This represents a three-orders-of-magnitude improvement over the 112% overhead of RO-based detectors [KSS+18] (Section 1.1.2) and even a higher efficiency than the 9% overhead of coil-LC oscillators [HHM+17] (Section 1.1.2). The sensor thus achieves lower power overhead than our algorithmic countermeasure (Table 1), highlighting its efficiency as a *low-overhead* solution.

3 Evaluation

For our EMFI measurements, we use a custom setup consisting of a XYZ stage to move the EM field emitter to different positions over the ASIC described in Section 2.1. The stage offers an accuracy of 100 μm and allows us to place the EM field emitter in fine-grained steps at the positions of interest. The ASIC is hosted by a custom Printed Circuit Board (PCB) designed to establish communication between the ASIC and a Control-FPGA. The Control-FPGA is built using a *Digilent Cmod A7-15T* FPGA board, which is programmed with a bitstream that enables all communication required for our experiments. The setup



(a) Coil 0 (M7, 375 μm). (b) Coil 1 (M7, 375 μm). (c) Coil 2 (M7, 375 μm).

(d) Coil 3 (M7, 375 μm). (e) Coil 4 (M7, 375 μm). (f) Coil 5 (M7, 1500 μm).

(g) Coil 6 (M4, 750 μm). (h) Coil 7 (M4, 375 μm). (i) Coil 8 (M4, 187.5 μm).

(j) Coil 9 (M4, 93.75 μm). (k) Coil 10 (M2, 1500 μm).

Figure 3: Sizes and spatial arrangement of the various coils on the ASIC.

is shown in Figure 5a, and a closer view of the ASIC and the EM field emitter is given in Figure 5b. The setup is completed by a desktop computer that communicates via UART with the Control-FPGA and stores the measurement results and instructs the Control-FPGA which measurements to perform.

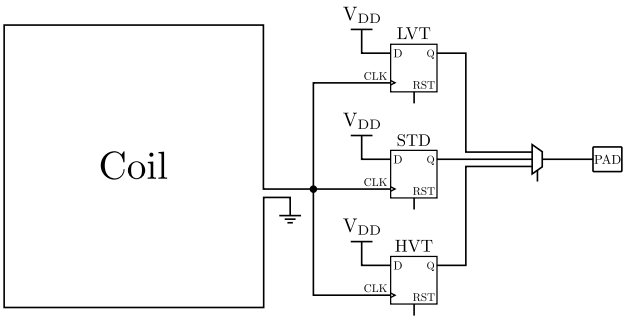
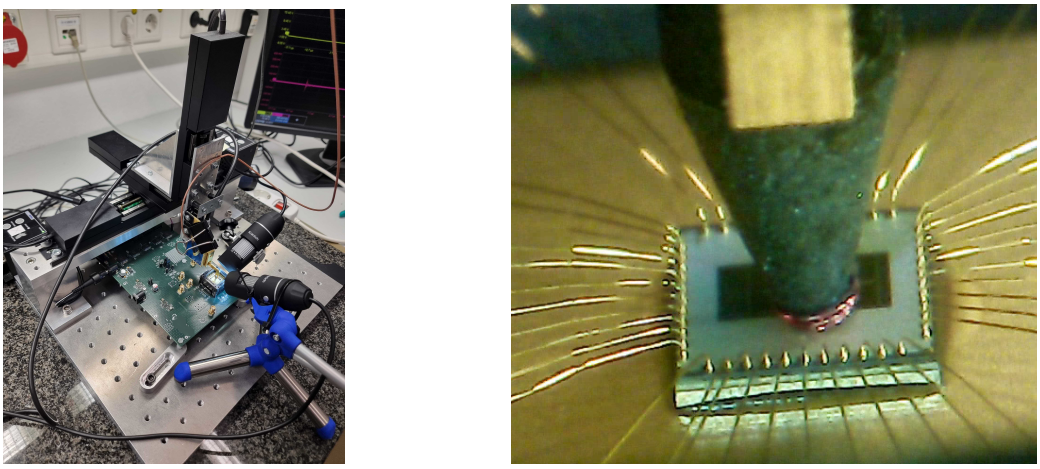


Figure 4: Schematic of the coil-based EMFI detection countermeasure based on flip-flops with three different threshold voltages V_t (LVT = low V_t, STD = standard V_t, HVT = high V_t).



(a) Overview Setup.

(b) EM field emitter over ASIC close-up.

Figure 5: Our setup used to perform all evaluations within the paper.

3.1 Pulse Generator and Injector

To conduct EMFI experiments, we utilize the ICI HH500-15 set from Langer EMV-Technik GmbH. This set comprises two main devices: the BPS 202 high-voltage supply and the ICI HH500-15 L-EFT pulse magnetic field source, henceforth referred to as the *emitter*. The high-voltage supply is configurable to produce a constant voltage within a range of 50 V to 500 V, adjustable in increments of 10 V. The pulse is generated directly within the ICI probe from this high voltage. Additionally, the polarity of the pulses can be set via remote control to be negative, positive, or alternating, eliminating the need for a tip change during polarity switches. For synchronizing the disturbance pulses with the Device Under Test (DUT), the EMFI system features a trigger input and a low-jitter delay line. The trigger input can be delayed by a counter value in steps of approximately 0.7 ns, with a minimum delay of 100 ns due to hardware constraints. The total maximum delay achievable is about 450 ns, with a jitter of less than 1 ns. Given our capability to generate a trigger signal synchronous to the negated edge of the clock of the ASIC, we can configure trigger events in steps of $T_{\text{asic}} = 83$ ns. Fine-tuning of the delay is accomplished using the low-jitter delay line in the pulse probe. A crucial feature for the fine-grained evaluation of the *coils* implemented in the ASIC, as detailed in Section 2.2, is the ability to induce highly localized electromagnetic fields. This is facilitated by the ICI probe, which has a

probe tip with a coil diameter of only 500 μm. The voltage, polarity, and delay variables can be set via a computer, which also instructs the controller to perform the measurements, enabling complete automation of the evaluation process.

3.2 Evaluation Procedure

We investigate the impact of several parameters on the effectiveness of frontside EMFI against our target ASIC. We did not perform backside EMFI due to resource constraints. This approach requires extensive preparation and specialized equipment for backside chip preparation. Moreover, the preparation procedure carries a high risk of damaging the chip, which is unacceptable given the limited supply of fabricated ASIC samples. Therefore, we exclusively execute frontside EMFI. This corresponds with the fundamental motivation of EMFI: to achieve a balance between the effectiveness of an attack and the amount of preparation required. A complex backside preparation process would undermine this tradeoff.

The parameters considered for our extensive EMFI evaluation procedure include the emitters location ℓ, the polarity p of the emitted EM field, the voltage level v of the pulse generated by the pulse generator, and the low-jitter delay d after which the pulse is applied. To estimate ℓ, we model the chip as a 12×12 grid, allowing the emitter to take 11 steps along both the X and Y axes. This resolution offers a compromise between spatial granularity and the overall time required to perform the experiments. To position the emitter at a specific location, such as one tile of the grid, the stage can perform 11 equally spaced steps per axis across the surface of the ASIC between the bonding wires. Each step corresponds to a 100 μm displacement, with our setup exhibiting an average deviation of 15 μm upon repeated experiments. The used coil has a base diameter of 1.8 mm and tapers down to a pointed tip with a diameter of $\delta_{\text{tip}} = 500\,\mu\text{m}$. The distance between two opposite bonding wires is 1825 μm, while the emitter can be moved by a maximum of $\Delta_{\text{eval}} = 1825\,\mu\text{m} - 650\,\mu\text{m} = 1175\,\mu\text{m}$, due to the approximate coil diameter of $\delta_{\text{contact}} = 650\,\mu\text{m}$ at the height where it would otherwise collide with the bonding wires. An additional clearance of approximately 27.5 μm is left on both inner sides of the bonding wires to ensure that the coil does not come into contact with them. This results in a step width of $\Delta_{\text{step}} = 100\,\mu\text{m}$. We focus our investigation on the area between the bonding wires, as the objective of this work is to study the effects of EMFI on protected AES cores and evaluate the effectiveness of the implemented coil-based countermeasure. Injecting a parasitic pulse directly into a specific set of bonding wires, particularly the power or clock rails, is effectively equivalent to performing clock or voltage glitching. As a result, the bonding wire area is considered out of scope for this study. The locations ℓ of the EM field emitter under investigation are given by

$$\ell \in L = \{(x, y) \mid x, y \in [0, 11]\}.$$

The ASIC provides 11 pins that allow direct access to one of the three detection registers associated with each coil, as described in Section 2.2. The specific type of register connected to the pins can be selected by configuring the variable $t \in T$, where $T = \{\text{lvt}, \text{std}, \text{hvt}\}$. The characteristics of the emitted EM field depend on the polarity o, which can be either positive (+) or negative (−). Although the EM field emitter can be configured to alternate between polarities, this mode is not relevant to our work, as we focus solely on the effects of individual pulses rather than frequency-specific or burst-based injections. Consequently, we restrict our selection to $o \in O = \{-, +\}$. As explained in Section 3.1, the EM field emitter can be configured to generate fields with voltages ranging from a minimum of 50 V to a maximum of 500 V, adjustable in increments of ten volts. Since a primary goal of this work is to investigate the sensitivity of the implemented coils, for which a fine granularity of voltage levels is essential, we evaluate all resulting $|V| = 46$ voltage levels

given by $v \in V = \{50, 60, 70, \ldots, 500\}$. The ASIC under investigation is clocked at a frequency of $f_{\text{clk}} = 12\,\text{MHz}$, as explained in Section 2. This corresponds to a clock period of $T_{\text{clk}} = \frac{1}{12}\,\text{ns} = 83.333\,\text{ns}$. The clock signal is generated by the Control-FPGA and is the inverted version of the internal clock used by the Control-FPGA itself. This setup enables us to generate a clock-synchronous trigger via the Control-FPGA. Since the trigger is asserted on the negative edge of the ASIC clock, while the internal logic of the ASIC is clocked on the positive edge, the low-jitter delay d allows us to shift the trigger timing toward the subsequent rising edge. By exploring all delays $d \in D = \{15, 20, 25, 30, 35\}$, we can delay the trigger relative to the negative clock edge of the ASIC by $100 + d \cdot 0.7\,\text{ns}$, using the delay adjustment integrated into the emitter as described in Section 3.1. The maximum tested delay, $d = 35$, corresponds to 124.5 ns, allowing us to shift the trigger approximately on the rising edge of the next clock cycle of the ASIC. We determined the set D through a preliminary evaluation by selecting random positions on the surface of the ASIC, setting the voltage to $v = 500$, and testing both polarities $p \in P$. We then incremented d in steps of 5, starting from 15 (since 13 is the minimal delay) up to 35, to ensure the trigger occurred before the next clock cycle. For each value of d, we observed whether faulty ciphertexts could be generated without consistently causing timeouts. This evaluation was performed at three randomly chosen positions near the center of the ASIC. At higher delay values, such as $d = 35$, we encountered constant timeouts and no usable faulty ciphertext for the positions tested. Based on the previous description, we define an EMFI setting S, hereafter referred to simply as a setting, according to Definition 1.

Definition 1 (Setting S)**.** A setting is a tuple $S \coloneqq (\ell, t, o, v, d)$ where

- $\ell \in L$ with $L = \{(x, y) \mid x, y \in [0, 11]\}$ is the location of the emitter.
- $t \in T$ with $T = \{\text{lvt}, \text{std}, \text{hvt}\}$ is the type of the considered coil status registers.
- $o \in O$ with $O = \{-, +\}$ is the polarity of the induced magnetic field.
- $v \in V$ with $v \in V = [50, 60, 70, \ldots, 500]$ is the voltage level that generates the field.
- $d \in D$ with $d \in D = \{15, 20, 25, 30, 35\}$ is the low-jitter trigger delay.

For each fault injection attempt under a specific setting, we record the ciphertext c produced by the target for a fixed plaintext p and secret key k. Additionally, we observe the output of all 33 detection logic registers (11 coils, each with 3 registers, as explained in Section 2.2), which are transmitted to the Control-FPGA via a custom four-bit parallel communication interface. This transmission requires the communication logic of the ASIC to remain functional after the EMFI happens. If this communication faces a timeout, i.e., if the target does not respond within 500 cycles, it is no longer possible to access the detection flags raised by the coils through the interface. Instead, a dedicated timeout flag e is set by the Control-FPGA. A timeout may result from a fault in the target circuit, the communication logic, or other peripherals. To still gain insight into the detection behavior in case the Control-FPGA detects a timeout, one register type from each of the eleven coils can be read directly via the pins, which are connected to the Control-FPGA. The set of flags stored in all 33 detection registers is denoted by r. Specifically, we denote the flag associated with a register of type $t \in T$ and a coil j with $j \in [0, 10]$ with r_j^t. Furthermore, we denote the set of flags supplied via the pins by p. Analogous to r, we use r_j^t to refer to the flag corresponding to register type t of coil j, where $j \in [0, 10]$. We summarize the four outputs observed from the target during a single EMFI event under setting S as a measurement, as defined in Definition 2.

Definition 2 (Measurement M)**.** A measurement is a tuple $M := (c, r, e, q)$ where:

- c is the ciphertext produced by the target.
- r is the output of the 33 detection logic registers.
- e is the timeout flag.
- q is the output of one detection register per coil (same type for all coils).

All settings S are investigated for 10 sequential measurements. Consequently, an experiment is defined by a measurement M, the corresponding setting S, and a repetition index i, as given in Definition 3.

Definition 3 (Experiment E)**.** An *experiment* E is a tuple $E := (S, M, i)$ where:

- S is a setting.
- M is the resulting measurement.
- $i \in [0, 9]$ is the repetition index.

For a single E, the register type t is identical for all q_j^t, where $j \in [0, 10]$. A value of $q_j^t = 1$ or $r_j^t = 1$ indicates that detection register j of type t has registered an EMFI attempt. We do not store the plaintext and key used in each experiment, as the same values are used throughout all experiments. Specifically, we use the test vector defined in Federal Information Processing Standards Publication (FIPS) 197, shown in Table 3. This choice enables straightforward detection of an *effective fault* by comparing the ciphertext output c of the target, given (p, k), with the reference ciphertext c_{ref} from Table 3. Keeping

Table 3: FIPS 197 AES standard example plaintext-key pair.

Plaintext p	`32 43 f6 a8 88 5a 30 8d 31 31 98 a2 e0 37 07 34`
Key k	`2b 7e 15 16 28 ae d2 a6 ab f7 15 88 09 cf 4f 3c`
Ciphertext c_{ref}	`39 25 84 1d 02 dc 09 fb dc 11 85 97 19 6a 0b 32`

the (p, k) constant within all experiments reduces the complexity of the analysis, since one potential parameter is eliminated from the evaluation. However, sticking to one (p, k)-pair might lead to incorrect conclusions when arguing about the security of a target in terms of the algorithmic countermeasure, explained in Section 2.1. However, since our primary focus is on evaluating the coil-based sensing, whose detection behavior is independent of the processed data, we consider this acceptable. For each target under investigation, we perform an experiment for every possible S. Therefore, we define the set of all possible experiments as an evaluation
$mathcal E$ in Definition 4.

Definition 4 (Evaluation)**.** The evaluation is the set of all possible experiments where:

$$\mathcal{E} := \{\, (S, M, i) \mid S \text{ is a setting},\ i \in [0, 9],\ M \text{ is the corresponding measurement} \}.$$

This results in $|\mathcal{E}| = |L| \cdot |T| \cdot |P| \cdot |V| \cdot |D| \cdot |I| = 1\,987\,200$ measurements being performed, as illustrated in Algorithm 1. Algorithm 1 is executed independently for each target under evaluation. The EM field emitter is positioned on the surface of the ASIC to ensure that $\delta_{\text{tip}} \ll \Delta_{\text{c2s}}$, where Δ_{c2s} denotes the distance from the coil tip to the surface of the ASIC. This setup ensures a localized EM field with a normally distributed amplitude beneath the coil, as described in [DLM21]. Settings are selected (cf. Lines 27) in a manner that optimizes efficiency. In particular, the loops are ordered such that the overhead introduced at each level decreases with increasing loop depth. For example, moving the stage (cf.

Algorithm 1 Evaluation Process.

Require: Plaintext p, Key k
Ensure: Evluation
$mathcal E$

1: $\mathcal{E} \leftarrow \emptyset$
2: **for** $\forall \ell \in L$ **do** ▷ *Place EM field emitter at location ℓ.*
3: **for** $\forall t \in T$ **do** ▷ *Select cell type t to connect to pins.*
4: **for** $\forall o \in O$ **do** ▷ *Set polarity of the field to o.*
5: **for** $\forall v \in V$ **do** ▷ *Set voltage level to v.*
6: **for** $\forall d \in D$ **do** ▷ *Set trigger delay to d.*
7: $S \leftarrow (\ell, t, o, v, d)$ ▷ *Specify settings.*
8: **for** $i \in [0, 9]$ **do**
9: $(c, r, e, q) \leftarrow \mathsf{Fault}(S, p, k)$ ▷ *Try to inject faults.*
10: $M \leftarrow (c, r, e, q)$
11: $E \leftarrow (S, M, i)$
12: $\mathcal{E} \leftarrow \mathcal{E} \cup \{E\}$
13: **return** $\mathcal{E}$

Line 2) incurs the highest overhead, whereas selecting the cell type t (cf. Line 3) can, and must, be changed dynamically during each ASIC reset, which occurs before every experiment. Choosing the delay (cf. Line 6) or repeating a measurement for the same setting introduces no additional overhead. In Line 9, the fault injection is modeled by the Fault function, which receives the inputs of the cryptographic primitive along with the selected settings. This function initiates the processing of the cryptographic primitive, waits for the trigger, and then injects the EM field. It returns a measurement, which is subsequently added to the experiment and included in the evaluation.

4 Evaluation Results

In this section, we analyze the detection behavior of the implemented coils, as described in Section 2.2, and evaluate the practical security level offered by the three target designs introduced in Section 2.1.1. The evaluation for each target $\mathcal{T}_i$, where $i \in [1, 3]$, is denoted by $\mathcal{E}_i$.

In addition to the comprehensive evaluation procedure described in Section 3.2, we conducted a preliminary series of experiments to assess the potential for false positives caused by electromagnetic emissions from electronic devices, such as smartphones or computers. During these experiments, the sensor was placed near the ASIC while tasks known to generate EM emissions (such as initiating phone calls or downloading large files over WLAN) were performed. No false positives were detected by the sensor in any of these scenarios. However, given the limited scope of these experiments, we cannot definitively exclude the possibility of false positives under certain conditions. Further research is needed to fully address this issue.

4.1 Coils

The ASIC under investigation offers a set of coils designed to detect EM fields produced by the EM field emitter. Since the coils work independently of the target under investigation, we merge the results of the evaluation of the different cores we have conducted. Specifically, the results presented in the following are retrieved by analyzing the data set $\mathcal{E} = \mathcal{E}_1 \cup \mathcal{E}_2 \cup \mathcal{E}_3$. This independence also allows us to ignore the received ciphertext $c \in M$ for all experiments $E \in \mathcal{E}$. Since a fault injected into the communication module of the ASIC prevents reliable

reading of the detection registers, this section focuses on the results reported through the output pins. For that purpose, we grouped all experiments $E \in \mathcal{E}$ by the cell type $t \in T$ with which E was performed. We denote the set of all experiments performed with the *lvt* detection registers connected to the output pins by $\mathcal{E}_{\text{lvt}}$, while $\mathcal{E}_{\text{std}}$ and $\mathcal{E}_{\text{hvt}}$ denote the measurements with *std* and *hvt* connected to the pins.

4.1.1 Local Detection Behavior of Coils and the Influence of Cell Type

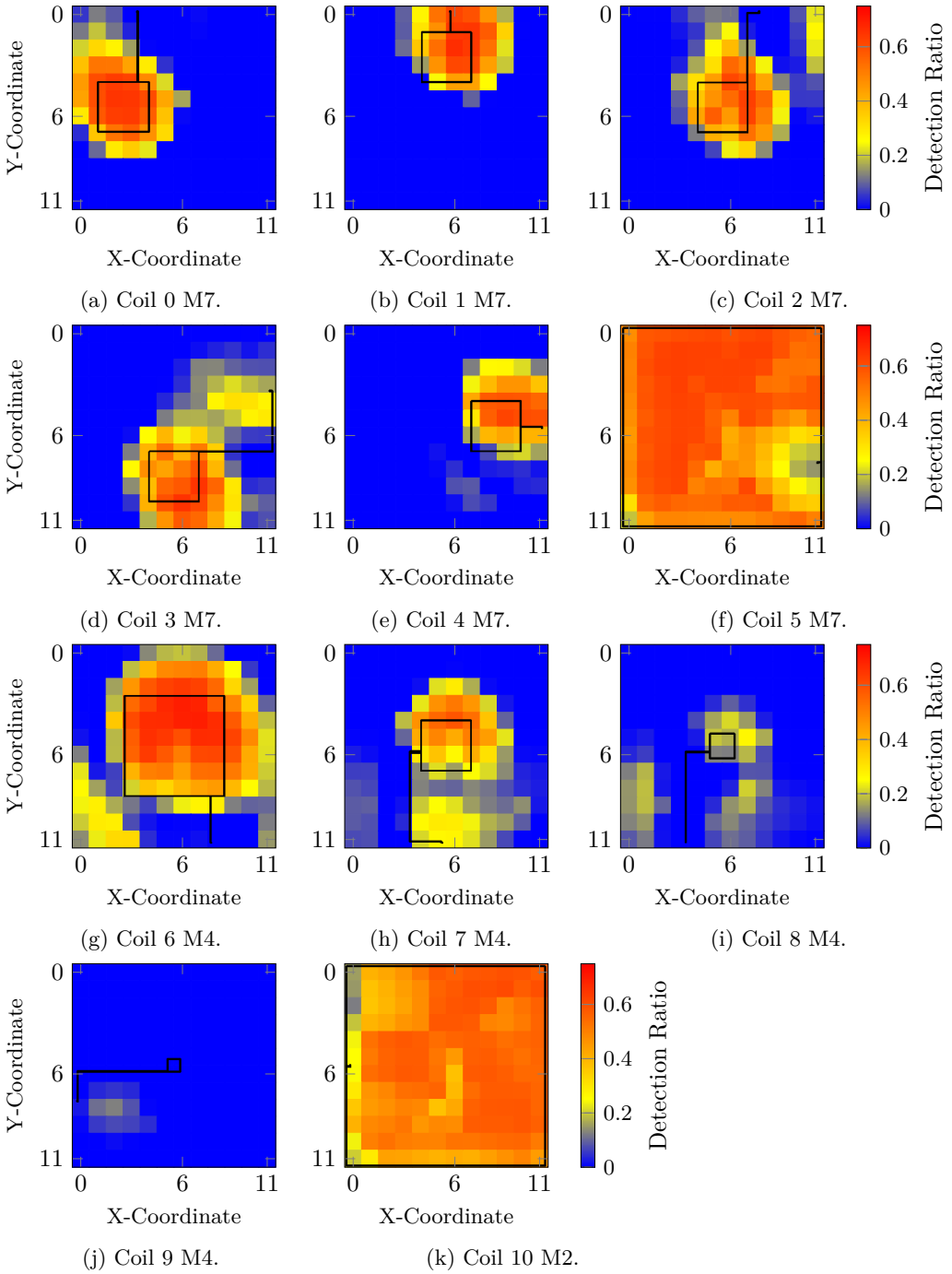
The initial hypotheses regarding the locality of the detection behavior of the coils are as follows:

Hypothesis 1. *Coils that encompass the entire area of the ASIC, specifically coils 5 and 10, are expected to detect the highest number of experiments $E \in \mathcal{E}$.*

Hypothesis 2. *Smaller coils are anticipated to exhibit more localized detection behavior, responding primarily when the EM field emitter is positioned in close proximity.*

Hypothesis 3. *Given that coil 5 is located closer to the surface of the ASIC compared to coil 10, it is expected to detect more experiments $E \in \mathcal{E}$, regardless of the cell type used in the detection register.*

To verify the correctness of these hypotheses, we analyze the detection rate of the coils as reported via the output pins. Figure 6 illustrates the reaction coils by checking all eleven *lvt* detection registers, denoted as r_j^{lvt} for $j \in [0, 10]$. For each position, a total of $|\mathcal{E}_{\text{lvt}}| = 13800$ experiments are conducted. The individual plots show the ratio of experiments at each position that trigger a reaction of a given coil. The maximum detection ratio of 0.69855 per experiment at a single location $\ell \in L$ is observed for Coil C6. As shown in Figure 6g, the experiments with the highest detection ratio are concentrated near the center of this coil. We attribute this to the interplay between the size and shape of the EM field emitter and the coil, which enables the detection of even lower voltage levels, such as $v = 50$, that are not detected by larger coils. However, across all locations $\ell \in L$, the majority of experiments are detected by C5, with 1040715 detections out of a total of 1669800 experiments corresponding to a detection ratio of 0.62326. This result supports Hypothesis 1. When focusing on coils apart from Coil 10 and Coil 5, which do not cover the entire area of the ASIC, a localized detection behavior can be observed, as illustrated in Figure 6a. This observation is not contradicted by Figure 6j, where the coil's response does not correspond to its physical location. However, this discrepancy arises from the coil's small size. Rather than capturing the EM field through the coil loop itself, we assume that the coil picks up the EM field from the wire connecting it to the detection registers. This effect is also evident in, for example, Figure 6h, where the coil detects approximately 30 percent of the experiments $E \in \mathcal{E}_{\text{lvt}}$. This observation supports Hypothesis 2. When focusing exclusively on $\mathcal{E}_{\text{lvt}}$, it is confirmed that Hypothesis 3 holds true. While Coil 5 on M7 detects 1 040 715 of the conducted experiments, Coil 10 on M2 detects only 984 038 experiments. Contrary, when focusing on $\mathcal{E}_{\text{hvt}}$, Coil 5 detects only 1 016 108 experiments, fewer than for $\mathcal{E}_{\text{lvt}}$, while Coil 10 detects 1 054 118 experiments across all locations, exceeding not only Coil 5's detections on the output of r_5^{lvt} but also those of r_5^{hvt}. This result disproves Hypothesis 3 and legitimates our design shown in Figure 4, since it demonstrates that relying solely on the *lvt* cells is insufficient. Furthermore, we conclude from the observation that coils on different metal layers, such as Coil 5 and Coil 10, supplement each other, since it does not always hold true that the coils closer to the surface of the ASIC detect more faults. Note that the plots for $\mathcal{E}_{\text{std}}$ and $\mathcal{E}_{\text{hvt}}$ look quite similar to those depicted in Figure 6 and do not contradict Hypothesis 1 and Hypothesis 2 but provide further evidence that Hypothesis 3 is not true. In summary, coils such as Coil 5 and Coil 10, which encapsulate the entire standard cell area, provide a high level of detection while introducing minimal overhead. Since the coil wires are



(a) Coil 0 M7. (b) Coil 1 M7. (c) Coil 2 M7.

(d) Coil 3 M7. (e) Coil 4 M7. (f) Coil 5 M7.

(g) Coil 6 M4. (h) Coil 7 M4. (i) Coil 8 M4.

(j) Coil 9 M4. (k) Coil 10 M2.

Figure 6: Relative number of experiments $E \in \mathcal{E}_{\text{lvt}}$ per location $\ell \in L$ that lead to $q_j^{\text{lvt}} = 1$, i.e., relative number of fault attempts that where detected by the *lvt* detection register of coil $j \in [0, 10]$, with coil j depicted in black.

routed outside the standard cell area, they do not interfere with the routing of the logic on the ASIC, resulting in near-zero overhead from the coil wiring. The only marginal overhead arises from the detection register construction shown in Figure 4. However, in a non-prototype design, the multiplexer wiring that connects one of the registers to the ASIC IO pins, used solely for analysis purposes, can be omitted. Smaller coils, such as Coil 6, which maximize fault detection at specific positions, offer strong localized detection and can be used to protect particularly vulnerable circuit areas. However, these smaller coils have a greater impact on routing since they occupy the routing resources on their respective metal layers. This can pose challenges in densely routed designs that already heavily utilize all available metal layers. Furthermore, employing a design that integrates cell types with different threshold voltages, as illustrated in Figure 4, and placing coils on multiple metal layers enhances EMFI coverage across varying parameters. It should be noted that if coils become too small, their ability to detect EMFI faults using a modern setup is limited (cf. Figure 6h).

4.1.2 Influence of Winding and Polarity

Hypothesis 4. *If the coil winding and the polarity $o \in O$ of the EM field are aligned, the coil can detect EM fields induced at lower voltage levels compared to when they are misaligned.*

To validate Hypothesis 4, we analyze the set of experiments $\mathcal{E}^{\text{intime}}$, which includes all experiments where the Control-FPGA did not report a timeout (i.e., $e \neq 1$ for each e in M). This selection allows us to observe the register value r instead of the output pin value q, enabling a broader range of experiments to be analyzed. Additionally, it removes the dependency on the specific cell type $t \in T$, allowing us to draw more general conclusions, even though we cannot evaluate the cases where the ASIC experiences a timeout. We begin by clustering all experiments $E \in \mathcal{E}^{\text{intime}}$ according to the polarity $o \in O$. Experiments with $o = +$ are grouped into the set $\mathcal{E}_+$, and those with $o = -$ into $\mathcal{E}_-$. Each of these sets is then further subdivided based on the voltage level $v \in V$ used during the experiment. This results in subsets $\mathcal{E}_{o,v}$ for each combination of $o \in O$ and $v \in V$. Next, we refine these subsets by identifying which coil detected each experiment. Specifically, $\mathcal{E}^{\text{detected}}_{o,v,j}$ contains all experiments from $\mathcal{E}_{o,v}$ for which the detection register $r^t_j = 1$ for any $t \in T$. This analysis is performed for all coils $j \in [0, 10]$. The resulting groups are used to generate Figure 7a.

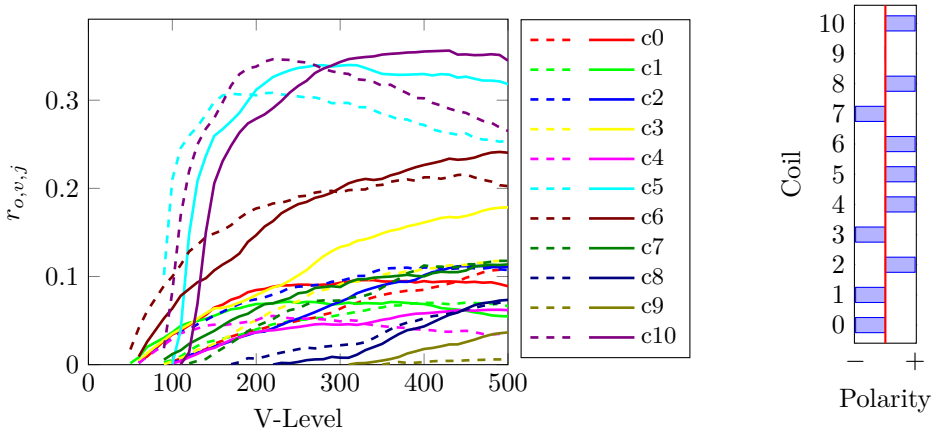
On the y-axis of this graph, the ratio of the detected cases $r_{o,v,j} = \frac{|\mathcal{E}^{\text{detected}}_{o,v,j}|}{|\mathcal{E}_{o,v}|}$ is depicted, while the different voltage levels $v \in V$ are given on the x-axis. The different polarities are illustrated by solid and dashed lines. The solid lines depict $r_{o,v,j}$ for $o = -$, while the dashed lines represent $o = +$. The different colors indicate different coils $j \in [0, 10]$. By examining whether the dashed ($o = +$) or solid ($o = -$) line appears at lower voltage levels $v \in V$, we can infer which coils are specifically sensitive to a given polarity. In Figure 7b, we highlight the polarity that triggers each coil at lower voltages. Excluding Coil 9, the observed behavior aligns with the winding direction of the coils, as summarized below:

Clockwise: Coils 0, 1, 3, 7

Counterclockwise: Coils 2, 4, 5, 6, 8, 9, 10

Coil 9 is excluded from this analysis because, as shown in Figure 6j, it is too small to reliably detect any experiment.

In the worst case, experiments with a polarity that does not match the winding of the coils are only detected at voltage values that are up to 50 V higher than those that match the winding of the coil. For example, Coil 8 detects the matching polarity at $v = 170$, whereas the opposite polarity is only detected at $v = 220$. This observation confirms



(a) Coil reaction behavior depending on the polarization of the EM-field.

(b) Maximum sensitivity depending on polarization.

Figure 7: Dependency of the winding of coils and the polarity of the EM-field.

Hypothesis 4. Therefore, to ensure reliable detection across all polarities, coils with both winding directions should be implemented, or the detection mechanism must be redesigned to operate independently of the coil's winding orientation. Additionally, Figure 7a confirms the observation from Section 4.1.1 that Coil 6 detects the highest number of experiments at the lowest voltage level, $v = 50$. Only Coil 1 also detects at $v = 50$, but its detection ratio is significantly lower, with $r_{50,1} = 0.00237$, compared to $r_{50,6} = 0.01321$ for Coil 6. The best-performing coils in terms of overall detection across the entire surface of the ASIC (Coils 5 and 10) only detect for some positions at $v = 90$ a fault attempt. This suggests that relying solely on large coils with near-zero overhead may not be sufficient if faults can be injected at voltage levels below $v = 90$ or at positions where the coils start to detect only for higher voltages. Whether such fault injection is feasible is analyzed in Section 4.2. It is also important to note that the situation could worsen with the use of even smaller EM field emitters, which produce EM fields with smaller diameters. These fields may still affect the target at higher voltages while remaining undetected by the coils.

4.2 Targets and Coils

In this section, we analyze the effectiveness of combining the implemented physical and algorithmic countermeasures. To this end, we discuss the results of the experiments conducted on the three targets presented in Table 2.

In the following, the protection levels of the three targets ($\mathcal{T}_i$ for $i \in \{1, 2, 3\}$) are compared based on the number of *effective faults* occurring in their corresponding evaluations $\mathcal{E}_i$. An effective fault is observed when the output ciphertext c (from $\mathcal{T}_i$) deviates from the reference ciphertext c_{ref}. We denote the set of experiments in $\mathcal{E}_i$ containing effective faults as

$$\mathcal{E}_i^{c \neq c_{\text{ref}}} = \{E \in \mathcal{E}_i \mid c \neq c_{\text{ref}}\},$$

which includes timeouts. The set $\mathcal{E}_i^{e \neq 1}$ contains experiments that *do not* result in a timeout. The intersection $\mathcal{E}_i^{c \neq c_{\text{ref}}} \cap \mathcal{E}_i^{e \neq 1}$ identifies effective faults, excluding timeouts. The number of elements of such intersections are denoted with $N_i^{\text{ef}} = |\mathcal{E}_i^{c \neq c_{\text{ref}}} \cap \mathcal{E}_i^{e \neq 1}|$. These numbers are used to assess the protection levels of the individual targets. Following theoretical assumptions, the unprotected target $\mathcal{T}_1$ is expected to have a lower security level than $\mathcal{T}_2$,

formalized as

$$N_1^{\text{ef}} > N_2^{\text{ef}}.$$

Extending this relation to $\mathcal{T}_3$ yields Hypothesis 6, encoding the intuition that better-protected cores should exhibit fewer effective faults. The theoretical protection levels of the targets (under the f-fault model [IPSW06]) are summarized in Table 2. To study also the security provided by the used hybrid (physical + algorithmic) approach, we also investigate the number of effective and undetected faults ($N_i^{\text{euf}} = |\mathcal{E}_i^{r\neq\mathbf{0}^{33}} \cap \mathcal{E}_i^{c\neq c_{\text{ref}}} \cap \mathcal{E}_i^{e\neq 1}|$). As with the previously discussed numbers, a lower value of N_i^{euf} corresponds to a higher security level for the combination of $\mathcal{T}_i$ and the presented physical detection mechanism. To analyze the security-area tradeoff in AES implementations, we compute the effective undetected fault susceptibility density $\rho_i = \frac{N_i^{\text{euf}}}{a_i}$ for each target, where a_i is the area in Gate Equivalent (GE). A lower ρ indicates that the implementation leverages its area more effectively to prevent undetected effective faults. For example, if $\rho_1 > \rho_2$ for $a_1 < a_2$ (cf. Table 1), the larger design ($\mathcal{T}_2$) achieves better fault detection and correction *per unit area*, suggesting its added hardware provides some security benefits. However, since an adversary can target any point of the ASIC, the previous statement requires careful consideration. When $\rho_2 > \rho_1$, a clearer implication is that more complex targets introduce vulnerabilities not only from the increased area but also from secondary effects caused by higher circuit complexity.

To verify not only Hypothesis 6 but also Hypothesis 5 and Hypothesis 7, we analyze the experiments conducted for the individual targets.

Hypothesis 5. *A target is vulnerable to faults induced by EMFI only when the emitter is placed in close to the target circuit.*

Hypothesis 6. *For $\mathcal{T}_1$, more effective faults (independent of detection logic) occur than for $\mathcal{T}_2$, while for $\mathcal{T}_3$ even more effective faults occur than for $\mathcal{T}_2$.*

Hypothesis 7. *The observed timeouts occur at the location where the control logic of the ASIC is located.*

4.2.1 AES Byte Serial Unprotected

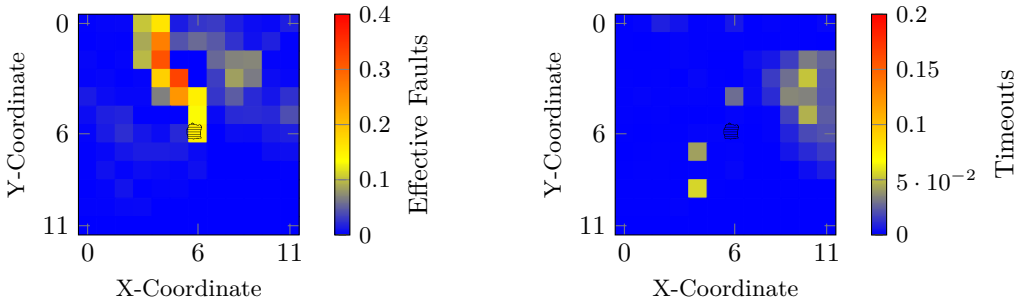
An overview of the results when performing the experiments in $\mathcal{E}_1$ is given in Table 4. While the table columns are partially clarified in Section 4.2, we repeat their definitions below to provide a complete explanation. The first column of the table reports the total

Table 4: Result overview for $\mathcal{T}_1$.

$\lvert\mathcal{E}_1\rvert$	$\lvert\mathcal{E}_{1,\ell}\rvert$	$\lvert\mathcal{E}_1^{r\neq\mathbf{0}^{33}}\rvert$	$\lvert\mathcal{E}_1^{c\neq c_{\text{ref}}}\rvert$	$\lvert\mathcal{E}_1^{r\neq\mathbf{0}^{33}} \cap \mathcal{E}_1^{c\neq c_{\text{ref}}}\rvert$	N_1^{ef}	N_1^{euf}	ρ_1
1 987 200	13 800	1 250 211	58 567	6005	49 308	527	0.1328

number of experiments $|\mathcal{E}_1|$ that were conducted. The second column indicates the number of experiments per position $|\mathcal{E}_{1,\ell}|$. The third column focuses on the coil-based detection mechanism and indicates how many of the $|\mathcal{E}_1|$ experiments were successfully detected by the coils. This is denoted by $|\mathcal{E}_1^{r\neq\mathbf{0}^{33}}| = |\{E \in \mathcal{E}_1 \mid r \neq \mathbf{0}^{33}\}|$. In the next column, the number of experiments $|\mathcal{E}_1^{c\neq c_{\text{ref}}}| = |\{E \in \mathcal{E}_1 \mid c \neq c_{\text{ref}}\}|$ that resulted in an effective fault, including those that caused a timeout of the ASIC, are given. This column is followed by the number of experiments $|\mathcal{E}_1^{r\neq\mathbf{0}^{33}} \cap \mathcal{E}_1^{c\neq c_{\text{ref}}}|$ that are undetected effective faults (including timeouts). Next we report the count of effective faults where a valid output is produced (rather than triggering a timeout), formally defined as $N_1^{\text{ef}} = |\mathcal{E}_i^{c\neq c_{\text{ref}}} \cap \mathcal{E}_i^{e\neq 1}|$. This is followed by the count of *undetected* effective faults, given by $N_1^{\text{euf}} = |\mathcal{E}_1^{r\neq\mathbf{0}^{33}} \cap \mathcal{E}_1^{c\neq c_{\text{ref}}} \cap \mathcal{E}_3^{e\neq 1}|$.

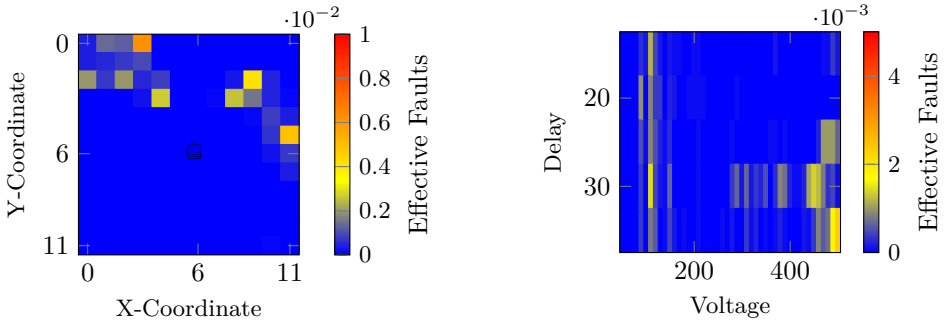
The last column specifies the effective undetected fault susceptibility density $\rho_1 = \frac{N_1^{\text{euf}}}{a_1}$ where a_1 is the area in GE. Since $N_1^{\text{euf}} > 0$, it is indeed possible to fault $\mathcal{T}_1$ without triggering the detection mechanism depicted in Figure 4. To verify Hypothesis 5, we compare a heatmap of the effective faults plotted in Figure 8a with the location of the target visualized in Figure 2a. The heatmap shows the ratio of experiments per location



(a) Effective Faults (no timeouts).

(b) Timeouts.

Figure 8: Effective faults with 13 800 experiments per location $\ell \in L$ for target $\mathcal{T}_1$ with $\mathcal{T}_1$ shaded and contoured in black.



(a) Per Position with $\mathcal{T}_1$ shaded in black.

(b) Per Delay and Voltage.

Figure 9: Effective and undetected Faults for $\mathcal{T}_1$.

$\ell \in L$ that causes the core to compute a faulty output but produces no timeout, i.e., $e \neq 1$ and $c_{\text{ref}} \neq c$. We observe a clear overlap between the location of the core and the region where effective faults are induced. However, the area where the highest number of experiments result in effective faults does not directly correspond to the location of the target. Instead, this region of the ASIC includes several components of the global clock tree, which are connected to all cores. This contradicts Hypothesis 5 and provides evidence that EMFI can induce faults through interference with the global clock tree. This finding is consistent with recent observations by Nabhan *et al.* [NDR+23] on an FPGA platform, where they reported that, in certain configurations, the electromagnetic field couples with the clock tree. Additionally, Figure 8b visualizes the locations where experiments result in timeouts. A distinct cluster of timeouts can be observed near the upper half of the right edge, with a few additional faults occurring along the vertical line at $x = 4$. We note a slight correlation between these timeouts and the wire utilization, as shown in Figure 1b. Specifically, in the region where $x = 4$ and $y \in [7, 9]$, there are multiple connections to the ASICs power straps, highlighted by red dots in Figure 1b. We attribute the concentration of timeouts along the upper right border to the dense accumulation of wires near the second

power strap from the right. However, there is no evidence for Hypothesis 7: the timeouts do not correlate with the position of the control register. Figure 9 serves as a reference for analyzing how experiments that are not detected by the coils can still affect the target core. In particular, Figure 9a illustrates the proportion of undetected experiments that successfully manipulate the target $\mathcal{T}_1$ at each position $\ell \in L$. Our indicate show that undetected faults do not occur at any location in the center of the ASIC, indicating that the core is difficult to fault without detection in this central region. We attribute this to the presence of multiple overlapping coils distributed in the center of the ASIC, as shown in Figure 3. An important observation is that the majority of the experiments that bypass the detection logic are performed at higher voltage levels, as shown in Figure 8b. This suggests that the EM fields generated by our high-voltage setup are highly localized in nature and do not necessarily trigger Coil 5 or Coil 10, even though these coils encompass the entire area of the ASIC.

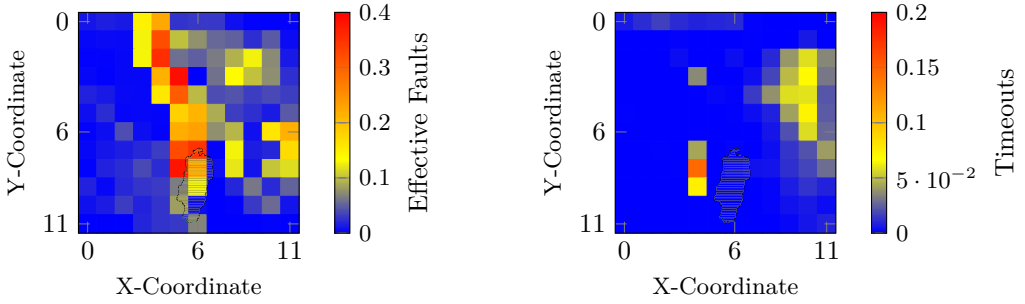
4.2.2 AGEFA AES Byte Serial [3, 1, 3]

For target $\mathcal{T}_2$, we observe more than 2.39 times the number of effective faults and timeouts compared to $\mathcal{T}_1$, as shown in Table 5 (column definitions are provided in Section 4.2.1). This is expected, as Table 1 shows that $\mathcal{T}_2$ is an order of magnitude larger than $\mathcal{T}_1$. Moreover,

Table 5: Result overview for $\mathcal{T}_2$.

$\lvert\mathcal{E}_2\rvert$	$\lvert\mathcal{E}_{2,\ell}\rvert$	$\lvert\mathcal{E}_2^{r\neq\mathbf{0}^{33}}\rvert$	$\lvert\mathcal{E}_2^{c\neq c_{\text{ref}}}\rvert$	$\lvert\mathcal{E}_2^{r\neq\mathbf{0}^{33}} \cap \mathcal{E}_2^{c\neq c_{\text{ref}}}\rvert$	N_2^{ef}	N_2^{euf}	ρ_2
1987200	13800	1270852	140123	13452	121184	1546	0.0448

as discussed in Section 4.2.1, the region where $\mathcal{T}_2$ is located is particularly susceptible to EM field interference. Consequently, the increased fault count can be attributed to the larger number of experiments that can be conducted within this vulnerable area. The number



(a) Effective Fault (no timeouts).

(b) Timeout.

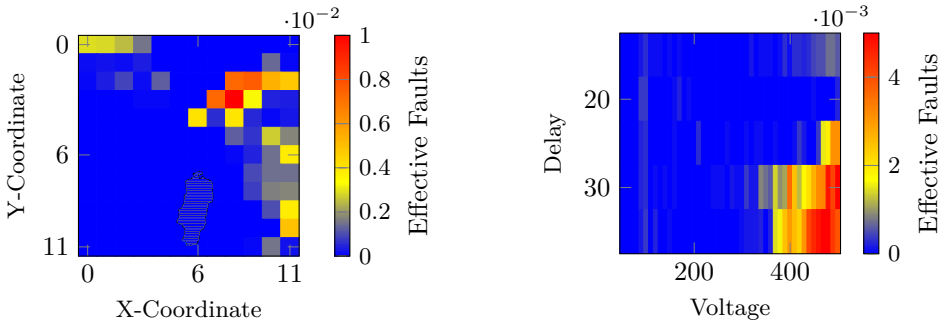
Figure 10: Effective faults with 13 800 experiments per location $\ell \in L$ for target $\mathcal{T}_2$ with $\mathcal{T}_2$ shaded and contoured in black.

of *purely effective faults* (excluding timeouts) N_i^{ef} increases by a factor of 2.46, while the number of *undetected effective faults* (also excluding timeouts) N_i^{euf} rises even more sharply, by a factor of 2.9. This increase alone is sufficient to reject Hypothesis 6. It indicates that the better protected cores are vulnerable to more differently parametrized attacks, reducing the requirements for the adversary to cause an effective fault. They are therefore valid to be exploited by an adversary. It appears that the area used to implement the algorithmic error correction does not justify the limited level of fault correction it provides. However, this observation is somewhat relativized when comparing ρ_1 (cf. Table 4) with ρ_2 (cf. Table 5), which shows that although the area of the countermeasure increases,

the effective undetected fault-to-area ratio decreases. This aligns with the discussion in Section 4.2, indicating that the algorithmic countermeasure does provide some fault correction capabilities. Nevertheless, it remains critical to note that an adversary faces no artificial limitations on fault injection positions. Thus, ρ_i is only meaningful for evaluation if the complexity of the applied strategies further reduces the security level beyond what the increased area alone would suggest.

The fact that it is possible to alter the target and produce a faulty output from outside the target area is further supported by Figure 10a. This visualization provides additional evidence that the occurrence of timeouts correlates with regions of high wire utilization.

Interestingly, as shown in Figure 11b, not only does the number of locations where an undetected experiment can cause an effective fault increase (as seen in Figure 11a), but the voltage window encompassing the majority of voltage levels that induce effective faults also increases. This provides evidence that $\mathcal{T}_2$ is not only more susceptible to faults due



(a) Per Position with $\mathcal{T}_2$ shaded in black.

(b) Per Delay and Voltage.

Figure 11: Effective and undetected faults for $\mathcal{T}_2$.

to the higher number of susceptible locations but also because individual locations are more prone to effective yet undetected faults. We attribute this either to the higher power consumption or to the greater number of registers dependent on the clock signal, as shown in Table 1.

4.2.3 AGEFA AES Byte Serial [16, 8, 5]

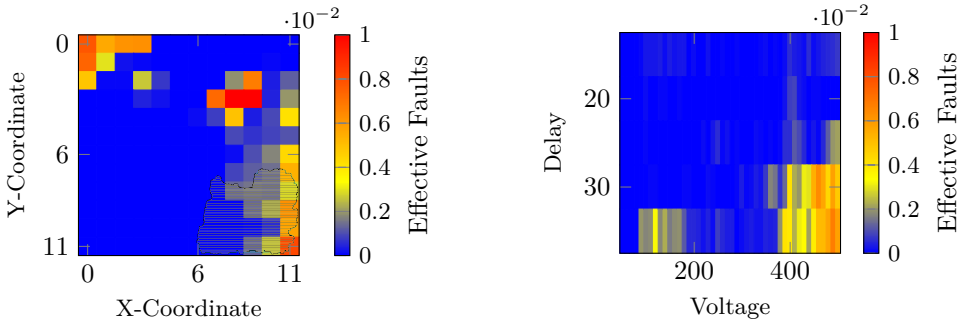
The situation regarding the algorithmic countermeasures implemented for target $\mathcal{T}_3$ is similar to that of $\mathcal{T}_2$. Compared to $\mathcal{T}_1$, $\mathcal{T}_3$ exhibits a significant increase in both the number of effective faults and timeouts, as well as the total number of effective faults (N_3^{ef}) and effective undetected faults (N_3^{euf}). Specifically, these metrics are higher by factors of 4.8, 5.3, and 4, respectively (cf. Table 6 and Table 4; column definitions are provided in Section 4.2.1). The increase can again be attributed to the countermeasure's area,

Table 6: Result overview for $\mathcal{T}_3$.

$\lvert\mathcal{E}_3\rvert$	$\lvert\mathcal{E}_{3,\ell}\rvert$	$\lvert\mathcal{E}_3^{r\neq\mathbf{0}^{33}}\rvert$	$\lvert\mathcal{E}_3^{c\neq c_{\text{ref}}}\rvert$	$\lvert\mathcal{E}_3^{r\neq\mathbf{0}^{33}} \cap \mathcal{E}_3^{c\neq c_{\text{ref}}}\rvert$	N_3^{ef}	N_3^{euf}	ρ_3
1 987 200	13 800	1 264 266	284 189	16 177	263 732	2157	0.0135

which is two orders of magnitude larger, suggesting that the increased area does not justify the level of protection provided. Since the figures illustrating the locations of timeouts and effective faults reveal no new insights for this target, we omit them here. A novel observation exclusive to $\mathcal{T}_3$ is the appearance of a new voltage window between $v = 90$ and $v = 180$ where effective faults can be induced without detection. This further supports the conclusion that undetected but effective faults are not only more likely due to the increased

number of experiments made possible by the target's larger area, but also because faults can occur at certain locations with lower voltage levels due to higher circuit complexity.



(a) Per Position with $\mathcal{T}_3$ shaded in black.

(b) Per Delay and Voltage.

Figure 12: Effective and undetected Faults for $\mathcal{T}_3$.

5 Conclusion

We successfully designed, implemented, and fabricated an ASIC hosting various AES cores protected with different algorithmic correction countermeasures and additional physical detection countermeasures. One core serves as a baseline with no algorithmic countermeasures, while the others are protected with different algorithmic correction countermeasures. All cores are protected with a coil-based physical detection mechanism. The algorithmic countermeasures, based on the principles of Impeccable Circuits II [SRM20], can correct up to two faults. The physical countermeasures aim to detect EMFI attempts through a detection logic using coils designed to sense EM fields. The results of our practical evaluations, conducted with a state-of-the-art EMFI setup, are twofold. Firstly, we observe that the majority of EMFI experiments could be detected by coil-based detection mechanisms. However, even with all 11 coils and sensitivities utilized, some parameters could bypass this detection. Despite this, we gained significant insights into the behavior of the coils under differently parameterized EM fields. Notably, we observed that the coil winding and the polarity of the EM field significantly impact detection capability. Secondly, we identified a discrepancy between theory and practice: the core with the highest theoretical protection is the most fault-prone, while the unprotected core is the most resilient. This phenomenon has two primary causes. First, the additional area required for protection mechanisms increases the number of positions where EMFI can be successfully performed, providing more opportunities for an adversary to fine-tune attacks and bypass the coil-based detection mechanism. Second, larger cores are more susceptible to faults at individual positions, further aiding an adversary in bypassing physical countermeasures. We hypothesize that this increased susceptibility could be due to less stable power consumption or more registers, both of which raise the likelihood of violating timing slack and causing timing faults.

Acknowledgments

The work described in this paper has been supported in part by the Federal Ministry of Research, Technology and Space of Germany through the Project KOSEF (16KIS1597). Thorben Moos is a postdoctoral researcher of the Belgian Fund for Scientific Research (F.R.S.-FNRS).

References

[ADN+10] Michel Agoyan, Jean-Max Dutertre, David Naccache, Bruno Robisson, and Assia Tria. When clocks fail: On critical paths and clock faults. In *CARDIS 2010*, volume 6035 of *LNCS*, pages 182–193. Springer, 2010.

[AMR+20] Anita Aghaie, Amir Moradi, Shahram Rasoolzadeh, Aein Rezaei Shahmirzadi, Falk Schellenberg, and Tobias Schneider. Impeccable circuits. *IEEE Trans. Computers*, 69(3):361–376, 2020.

[BBM+22] Timo Bartkewitz, Sven Bettendorf, Thorben Moos, Amir Moradi, and Falk Schellenberg. Beware of Insufficient Redundancy An Experimental Evaluation of Code-based FI Countermeasures. *IACR TCHES*, 2022(3):438–462, 2022.

[BDL97] Dan Boneh, Richard A. DeMillo, and Richard J. Lipton. On the importance of checking cryptographic protocols for faults (extended abstract). In *EUROCRYPT 1997*, volume 1233 of *LNCS*, pages 37–51. Springer, 1997.

[BFG+24] Sonia Belaïd, Jakob Feldtkeller, Tim Güneysu, Anna Guinet, Jan Richter-Brockmann, Matthieu Rivain, Pascal Sasdrich, and Abdul Rahman Taleb. Formal Definition and Verification for Combined Random Fault and Random Probing Security. In Kai-Min Chung and Yu Sasaki, editors, *Advances in Cryptology - ASIACRYPT 2024 - 30th International Conference on the Theory and Application of Cryptology and Information Security, Kolkata, India, December 9-13, 2024, Proceedings, Part VII*, volume 15490 of *LNCS*, pages 167–200. Springer, 2024.

[BJK+16] Christof Beierle, Jérémy Jean, Stefan Kölbl, Gregor Leander, Amir Moradi, Thomas Peyrin, Yu Sasaki, Pascal Sasdrich, and Siang Meng Sim. *The SKINNY Family of Block Ciphers and Its Low-Latency Variant MANTIS*, pages 123–153. Springer Berlin Heidelberg, 7 2016.

[BKHL20] Jakub Breier, Mustafa Khairallah, Xiaolu Hou, and Yang Liu. A countermeasure against statistical ineffective fault analysis. *IEEE Trans. Circuits Syst.*, 67-II(12):3322–3326, 2020.

[BS97] Eli Biham and Adi Shamir. Differential fault analysis of secret key cryptosystems. In *CRYPTO 1997*, volume 1294 of *LNCS*, pages 513–525. Springer, 1997.

[CDM+25] Gaëtan Cassiers, Siemen Dhooghe, Thorben Moos, Sayandeep Saha, and François-Xavier Standaert. Fly away: Lifting fault security through canaries and the uniform random fault model. *IACR Cryptol. ePrint Arch.*, page 863, 2025.

[Con90] John H. Conway. Integral lexicographic codes. *DM*, 83(2-3):219–235, 1990.

[CS86] John H. Conway and Neil J. A. Sloane. Lexicographic codes: Error-correcting codes from game theory. *IEEE TIT*, 32(3):337–348, 1986.

[DDRT12] Amine Dehbaoui, Jean-Max Dutertre, Bruno Robisson, and Assia Tria. Electromagnetic transient faults injection on a hardware and a software implementations of AES. In Guido Bertoni and Benedikt Gierlichs, editors, *2012 Workshop on Fault Diagnosis and Tolerance in Cryptography, Leuven, Belgium, September 9, 2012*, pages 7–15. IEEE Computer Society, 2012.

[DEK+18] Christoph Dobraunig, Maria Eichlseder, Thomas Korak, Stefan Mangard, Florian Mendel, and Robert Primas. SIFA: exploiting ineffective fault inductions on symmetric cryptography. *IACR TCHES*, 2018(3):547–572, 2018.

[DLM19] Mathieu Dumont, Mathieu Lisart, and Philippe Maurine. Electromagnetic fault injection : How faults occur. In *2019 Workshop on Fault Diagnosis and Tolerance in Cryptography, FDTC 2019, Atlanta, GA, USA, August 24, 2019*, pages 9–16. IEEE, 2019.

[DLM21] Mathieu Dumont, Mathieu Lisart, and Philippe Maurine. Modeling and simulating electromagnetic fault injection. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.*, 40(4):680–693, 2021.

[DN23] Siemen Dhooghe and Svetla Nikova. The Random Fault Model. In *SAC 2023*, volume 14201 of *LNCS*, pages 191–212. Springer, 2023.

[EAD+20] Mohammad Ebrahimabadi, Md Toufiq Hasan Anik, Jean-Luc Danger, Sylvain Guilley, and Naghmeh Karimi. Using digital sensors to leverage chips' security. In *2020 IEEE Physical Assurance and Inspection of Electronics (PAINE)*, pages 1–6. IEEE, 12 2020.

[EBRM16] David El-Baze, Jean-Baptiste Rigaud, and Philippe Maurine. A fully-digital em pulse detector. In Luca Fanucci and Jürgen Teich, editors, *2016 Design, Automation & Test in Europe Conference & Exhibition, DATE 2016, Dresden, Germany, March 14-18, 2016*, pages 439–444. IEEE, 2016.

[ELH+12] Sho Endo, Yang Li, Naofumi Homma, Kazuo Sakiyama, Kazuo Ohta, and Takafumi Aoki. An efficient countermeasure against fault sensitivity analysis using configurable delay blocks. In Guido Bertoni and Benedikt Gierlichs, editors, *2012 Workshop on Fault Diagnosis and Tolerance in Cryptography, Leuven, Belgium, September 9, 2012*, pages 95–102. IEEE Computer Society, 2012.

[EMV+22] Mohammad Ebrahimabadi, Suhee Sanjana Mehjabin, Raphael Viera, Sylvain Guilley, Jean-Luc Danger, Jean-Max Dutertre, and Naghmeh Karimi. Detecting laser fault injection attacks via time-to-digital converter sensors. In *2022 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*. IEEE, 6 2022.

[FJLT13] Thomas Fuhr, Éliane Jaulmes, Victor Lomné, and Adrian Thillard. Fault attacks on AES with faulty ciphertexts only. In Wieland Fischer and Jörn-Marc Schmidt, editors, *2013 Workshop on Fault Diagnosis and Tolerance in Cryptography, Los Alamitos, CA, USA, August 20, 2013*, pages 108–118. IEEE Computer Society, 2013.

[GMO01] Karine Gandolfi, Christophe Mourtel, and Francis Olivier. Electromagnetic analysis: Concrete results. In *CHES 2001*, volume 2162 of *Lecture Notes in Computer Science*, pages 251–261. Springer, 2001.

[HAS09] Ghaith Hammouri, Kahraman Akdemir, and Berk Sunar. *Novel PUF-Based Error Detection Methods in Finite State Machines*, pages 235–252. Springer Berlin Heidelberg, 2009.

[HBB16a] Wei He, Jakub Breier, and Shivam Bhasin. *Cheap and Cheerful: A Low-Cost Digital Sensor for Detecting Laser Fault Injection Attacks*, pages 27–46. Springer International Publishing, 11 2016.

[HBB+16b] Wei He, Jakub Breier, Shivam Bhasin, Noriyuki Miura, and Makoto Nagata. Ring oscillator under laser: Potential of pll-based countermeasure against laser fault injection. In *2016 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)*, pages 102–113. IEEE, 8 2016.

[HGM+20] Jiaji He, Xiaolong Guo, Haocheng Ma, Yanjiang Liu, Yiqiang Zhao, and Yier Jin. Runtime trust evaluation and hardware trojan detection using on-chip EM sensors. In *57th ACM/IEEE Design Automation Conference, DAC 2020, San Francisco, CA, USA, July 20-24, 2020*, pages 1–6. IEEE, 2020.

[HHM+17] Naofumi Homma, Yu-ichi Hayashi, Noriyuki Miura, Daisuke Fujimoto, Makoto Nagata, and Takafumi Aoki. Design methodology and validity verification for a reactive countermeasure against em attacks. *Journal of Cryptology*, 30:373–391, 4 2017.

[HSCM23] Shungo Hayashi, Junichi Sakamoto, Masaki Chikano, and Tsutomu Matsumoto. Effective layout design for laser fault sensor on fpga. In *CCS '23: ACM SIGSAC Conference on Computer and Communications Security*, pages 103–112. ACM, 11 2023.

[HSM25] Shungo Hayashi, Junichi Sakamoto, and Tsutomu Matsumoto. Design methodology of digital sensors for detecting laser fault injection attacks in fpgas. *Journal of Cryptographic Engineering*, 15, 6 2025.

[IPSW06] Yuval Ishai, Manoj Prabhakaran, Amit Sahai, and David A. Wagner. Private circuits II: keeping secrets in tamperable circuits. In Serge Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings*, volume 4004 of *Lecture Notes in Computer Science*, pages 308–327. Springer, 2006.

[KJJ99] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In *CRYPTO 1999*, volume 1666 of *LNCS*, pages 388–397. Springer, 1999.

[Koc96] Paul C. Kocher. Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. In *CRYPTO 1996*, volume 1109 of *LNCS*, pages 104–113. Springer, 1996.

[KSS+18] Mustafa Khairallah, Rajat Sadhukhan, Radhamanjari Samanta, Jakub Breier, Shivam Bhasin, Rajat Subhra Chakraborty, Anupam Chattopadhyay, and Debdeep Mukhopadhyay. Dfarpa: Differential fault attack resistant physical design automation. In *2018 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1171–1174. IEEE, 3 2018.

[MAM17] Vasily Mikhalev, Frederik Armknecht, and Christian Müller. On ciphers that continuously access the non-volatile key. *IACR Transactions on Symmetric Cryptology*, 2017.

[MFT+14] Noriyuki Miura, Daisuke Fujimoto, Daichi Tanaka, Yu-ichi Hayashi, Naofumi Homma, Takafumi Aoki, and Makoto Nagata. A local em-analysis attack resistant cryptographic engine with fully-digital oscillator-based tamper-access sensor. In *Symposium on VLSI Circuits, VLSIC 2014, Digest of Technical Papers, Honolulu, HI, USA, June 10-13, 2014*, pages 1–2. IEEE, 2014.

[MM24] Nicolai Müller and Amir Moradi. Automated Generation of Fault-Resistant Circuits. *IACR TCHES*, 2024(3):136–173, 2024.

[MNH+16] Noriyuki Miura, Zakaria Najm, Wei He, Shivam Bhasin, Xuan Thuy Ngo, Makoto Nagata, and Jean-Luc Danger. Pll to the rescue. In *DAC '16: The 53rd Annual Design Automation Conference 2016*, pages 1–6. ACM, 6 2016.

[NDR+23] Roukoz Nabhan, Jean-Max Dutertre, Jean-Baptiste Rigaud, Jean-Luc Danger, and Laurent Sauvage. A tale of two models: Discussing the timing and sampling EM fault injection models. In *Workshop on Fault Detection and Tolerance in Cryptography, FDTC 2023, Prague, Czech Republic, September 10, 2023*, pages 1–12. IEEE, 2023.

[OGM17] Sébastien Ordas, Ludovic Guillaume-Sage, and Philippe Maurine. Electromagnetic fault injection: the curse of flip-flops. *J. Cryptogr. Eng.*, 7(3):183–197, 2017.

[OGSM15] S. Ordas, L. Guillaume-Sage, and Philippe Maurine. Em injection: Fault model and locality. In *2015 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)*, pages 3–13. IEEE, 9 2015.

[OGT+14] Sébastien Ordas, Ludovic Guillaume-Sage, Karim Tobich, Jean-Max Dutertre, and Philippe Maurine. Evidence of a larger em-induced fault model. In Marc Joye and Amir Moradi, editors, *Smart Card Research and Advanced Applications - 13th International Conference, CARDIS 2014, Paris, France, November 5-7, 2014. Revised Selected Papers*, volume 8968 of *Lecture Notes in Computer Science*, pages 245–259. Springer, 2014.

[RSG23] Jan Richter-Brockmann, Pascal Sasdrich, and Tim Güneysu. Revisiting fault adversary models - hardware faults in theory and practice. *IEEE Trans. Computers*, 72(2):572–585, 2023.

[SA02] Sergei P. Skorobogatov and Ross J. Anderson. Optical fault induction attacks. In *CHES 2002*, volume 2523 of *LNCS*, pages 2–12. Springer, 2002.

[SGD08] Nidhal Selmane, Sylvain Guilley, and Jean-Luc Danger. Practical setup time violation attacks on AES. In *EDCC-7 2008*, pages 91–96. IEEE Computer Society, 2008.

[SRM20] Aein Rezaei Shahmirzadi, Shahram Rasoolzadeh, and Amir Moradi. Impeccable circuits II. In *57th ACM/IEEE DAC 2020*, pages 1–6. IEEE, 2020.

[WPZ+24] Hanqiu Wang, Max Panoff, Zihao Zhan, Shuo Wang, Christophe Bobda, and Domenic Forte. Programmable EM sensor array for golden-model free run-time trojan detection and localization. In *Design, Automation & Test in Europe Conference & Exhibition, DATE 2024, Valencia, Spain, March 25-27, 2024*, pages 1–6. IEEE, 2024.

[ZDT+14] Loic Zussa, Amine Dehbaoui, Karim Tobich, Jean-Max Dutertre, Philippe Maurine, Ludovic Guillaume-Sage, Jessy Clediere, and Assia Tria. Efficiency of a glitch detector against electromagnetic fault injection. In *Design Automation and Test in Europe*, pages 1–6. IEEE Conference Publications, 3 2014.