# LASER-BASED LOGIC STATE ANALYSIS IN HARDWARE SECURITY

## Threats and Opportunities

vorgelegt von
M. Sc.
Thilo Krachenfels
ORCID ID: 0000-0002-8569-2020

an der Fakultät IV - Elektrotechnik und Informatik
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften
- Dr.-Ing. -

genehmigte Dissertation

Promotionsausschuss:

Vorsitzender: Prof. Dr. Stefan Schmid
Gutachter: Prof. Dr. Jean-Pierre Seifert
Gutachter: Prof. Dr.-Ing. Christof Paar
Gutachter: Prof. Patrick Schaumont, Ph.D.

Tag der wissenschaftlichen Aussprache: 23. August 2023

Berlin 2023

Dedicated to my family

# ABSTRACT

Integrated circuits (ICs) are used in virtually every technical product, from consumer devices to hardware in more critical applications, such as infrastructure management and the military. At the same time, sophisticated side-channel analysis (SCA) attacks using chip failure analysis (FA) techniques can be leveraged to extract logical states from within the chip. So-called logic state imaging techniques allow the extraction of, for instance, data from on-chip memory cells by optical inspection through the chip's backside. If used maliciously, this endangers the chip's secret keys, user data, and intellectual property (IP). Consequently, countermeasures must be in place to protect from these powerful attacks. Chip manufacturers, however, have not yet put the highest effort into developing and integrating innovative countermeasures for a few reasons. First, it is expected that measures like masked implementations developed to protect against traditional SCA can also prevent attacks using FA tools. Second, it may be seductive to think that these attacks are too complex in terms of reverse engineering, which can prevent attacks. Third, it is believed that the high costs of suitable setups hinder potential adversaries.

This work demonstrates that schemes designed to protect against traditional SCA do not protect against single-trace optical techniques. We show that the individual key shares can be extracted using laser logic state imaging (LLSI) – a technique applied for security investigations for the first time. Furthermore, we show that applying machine learning techniques can significantly reduce the reverse engineering effort. In this regard, we show how to extract secrets using convolutional neural networks (CNNs) automatically. Concerning the argument of high setup costs, we show that attacks can be conducted much cheaper than expected. Our setup for thermal laser stimulation (TLS) is cheaper by a factor of ten compared with conventional FA tools. The mentioned techniques, though, can not only be utilized for attacking devices. In this regard, we use LLSI to spot malicious modifications of hardware implementations on field-programmable gate arrays (FPGAs) and show how tiny and dormant hardware Trojans (HTs) can be detected reliably.

Finally, this work discusses future research directions and challenges for laser-based hardware security investigations. Future directions include the new class of active SCA techniques, the development of countermeasures, and the simulation of optical probing techniques. The main future challenges discussed in this thesis are increasing logic densities, new transistor designs, 3D packaging, and backside power delivery networks.

# ZUSAMMENFASSUNG

Integrierte Schaltungen (ICs) werden in praktisch jedem technischen Produkt verwendet, von Verbrauchergeräten bis hin zu Hardware in kritischeren Anwendungen, wie z.B. im Infrastrukturbetrieb und beim Militär. Zugleich können komplexe Angriffe mit Seitenkanalanalyse-Techniken (SCA) und Methoden aus der Chip-Fehleranalyse (FA) ausgenutzt werden, um logische Zustände aus Chips zu extrahieren. Sogenannte Logic State Imaging-Techniken ermöglichen beispielsweise die Extraktion von Daten aus Speicherzellen durch optische Inspektion durch die Chiprückseite. Bei böswilliger Nutzung gefährdet dies geheime Schlüssel, Benutzerdaten und geistiges Eigentum (IP) auf dem Chip. Folglich müssen Gegenmaßnahmen umgesetzt werden, um sich vor diesen wirkungsvollen Angriffen zu schützen. Aus meheren Gründen haben die Chip-Hersteller jedoch noch nicht die größten Anstrengungen in der Entwicklung und Integration von innovativen Gegenmaßnahmen unternommen. Zum einen wird erwartet, dass Maßnahmen wie Masking-Implementierungen, die zum Schutz gegen traditionelle SCA entwickelt wurden, auch Angriffe mit FA-Tools verhindern können. Zweitens könnte man meinen, dass optische Angriffe bezüglich des nötigen Reverse Engineering zu komplex sind und somit Angriffe verhindert werden. Schließlich wird auch angenommen, dass die hohen Kosten für geeignete Setups potenzielle Angreifer fernhalten.

Diese Arbeit zeigt, dass Algorithmen, die zum Schutz gegen herkömmliche SCA entwickelt wurden, keinen Schutz gegen optische Single-Trace-Techniken bieten. Wir zeigen, dass die einzelnen Schlüsselanteile mit Hilfe von Laser Logic State Imaging (LLSI) extrahiert werden können - eine Technik, die zum ersten Mal für Sicherheitsuntersuchungen eingesetzt wird. Darüber hinaus zeigen wir, dass die Anwendung von Techniken des maschinellen Lernens den Reverse-Engineering-Aufwand erheblich reduzieren kann. In diesem Zusammenhang zeigen wir, dass man mithilfe von Convolutional Neural Networks (CNNs) automatisch kryptografische Schlüssel extrahieren kann. Was das Argument der hohen Setup-Kosten anbelangt, so zeigen wir, dass Angriffe viel kostengünstiger durchgeführt werden können als bisher angenommen. Unser Setup für die thermische Laserstimulation (TLS) ist im Vergleich zu herkömmlichen FA-Tools um den Faktor zehn günstiger. Die genannten Techniken können jedoch nicht nur für Angriffe verwendet werden. In diesem Zusammenhang verwenden wir LLSI, um bösartige Modifikationen von Hardware-Implementierungen auf Field-Programmable Gate Arrays (FPGAs) zu detektieren und zei-

gen, wie winzige und inaktive Hardware-Trojaner (HT) zuverlässig erkannt werden können.

Schließlich diskutiert diese Arbeit zukünftige Forschungsrichtungen und Herausforderungen für laserbasierte Hardware-Sicherheitsuntersuchungen. Zu den zukünftigen Richtungen gehören eine neue Klasse aktiver SCA-Techniken, die Entwicklung von Gegenmaßnahmen und die Simulation von Probing-Techniken. Die in dieser Arbeit diskutierten zukünftigen Herausforderungen sind steigende Logikdichten, neue Transistordesigns, 3D-Packaging, und rückseitige Stromversorgungsnetze.

# PUBLICATIONS

The publishers' versions of the following peer-reviewed publications are fully included in this thesis:

**T. Krachenfels**, H. Lohrke, J.-P. Seifert, E. Dietz, S. Frohmann, and H.-W. Hübers, "Evaluation of Low-Cost Thermal Laser Stimulation for Data Extraction and Key Readout," *Journal of Hardware and Systems Security*, vol. 4, no. 1, pp. 24–33, Mar. 2020. DOI: 10.1007/s41635-019-00083-9.

**T. Krachenfels**, F. Ganji, A. Moradi, S. Tajik, and J.-P. Seifert, "Real-World Snapshots vs. Theory: Questioning the t-Probing Security Model," in *2021 IEEE Symposium on Security and Privacy (SP)*, IEEE Computer Society, 2021, pp. 1955–1971. DOI: 10.1109/SP40001.2021.00029.

**T. Krachenfels**, T. Kiyan, S. Tajik, and J.-P. Seifert, "Automatic Extraction of Secrets from the Transistor Jungle using Laser-Assisted Side-Channel Attacks," in *30th USENIX Security Symposium (USENIX Security 21)*, USENIX Association, Aug. 2021, pp. 627–644, ISBN: 978-1-939133-24-3.

**T. Krachenfels**, J.-P. Seifert, and S. Tajik, "Trojan awakener: Detecting dormant malicious hardware using laser logic state imaging (extended version)," *Journal of Cryptographic Engineering*, May 2023. DOI: 10.1007/s13389-023-00323-3.

Along with the above items, the following publications of research data have been released:

**T. Krachenfels**, "Laser logic state images of masked AES implementations from registers on a Cyclone IV FPGA," Sep. 2020. DOI: 10.14279/depositonce-10440.

**T. Krachenfels**, T. Kiyan, S. Tajik, and J.-P. Seifert, "Images from on-chip memories captured using the laser-assisted side-channel techniques LLSI and TLS," Feb. 2021. DOI: 10.14279/depositonce-11354.

The following additional peer-reviewed publications were authored by Thilo Krachenfels ("*" denotes that both authors contributed equally to the corresponding work):

O. Bittner*, **T. Krachenfels***, A. Galauner, and J.-P. Seifert, "The Forgotten Threat of Voltage Glitching: A Case Study on Nvidia Tegra X2 SoCs," presented at the 2021 Workshop on Fault Detection and Tolerance in Cryptography (FDTC), IEEE Computer Society, Sep. 2021, pp. 86–97. DOI: 10.1109/FDTC53659.2021.00021.

T. Kiyan, **T. Krachenfels**, E. Amini, Z. Shakibaei, C. Boit, and J.-P. Seifert, "Extraction of Secrets from Allegedly Secret-free IoT Sensors using Artificial Intelligence," in *Proceedings of the IEEE International Symposium on the Physical and Failure Analysis of Integrated Circuits (IPFA)*, Sep. 2021.

**T. Krachenfels**, J.-P. Seifert, and S. Tajik, "Trojan Awakener: Detecting Dormant Malicious Hardware Using Laser Logic State Imaging," in *Proceedings of the 5th Workshop on Attacks and Solutions in Hardware Security*, ser. ASHES '21, New York, NY, USA: Association for Computing Machinery, Nov. 2021, pp. 17–27. DOI: 10.1145/3474376.3487282.

R. Buhren, H.-N. Jacob, **T. Krachenfels**, and J.-P. Seifert, "One Glitch to Rule Them All: Fault Injection Attacks Against AMD's Secure Encrypted Virtualization," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '21, New York, NY, USA: Association for Computing Machinery, Nov. 2021, pp. 2875–2889. DOI: 10.1145/3460120.3484779.

S. Parvin, **T. Krachenfels**, S. Tajik, F. S. Torres, J.-P. Seifert, and R. Drechsler, "Toward Optical Probing Resistant Circuits: A Comparison of Logic Styles and Circuit Design Techniques," in *2022 27th Asia and South Pacific Design Automation Conference (ASP-DAC)*, Jan. 2022, pp. 429–435. DOI: 10.1109/ASP-DAC52403.2022.9712518.

N. Kühnapfel, R. Buhren, H. N. Jacob, **T. Krachenfels**, C. Werling, and J.-P. Seifert, "EM-Fault It Yourself: Building a Replicable EMFI Setup for Desktop and Server Hardware," in *2022 IEEE Physical Assurance and Inspection of Electronics (PAINE)*, Oct. 2022. DOI: 10.1109/PAINE56030.2022.10014927.

S. Parvin, M. Goli, **T. Krachenfels**, S. Tajik, J.-P. Seifert, F. S. Torres, and R. Drechsler, "LAT-UP: Exposing Layout-Level Analog Hardware Trojans Using Contactless Optical Probing," presented at the IEEE Computer Society Annual Symposium on VLSI (ISVLSI 2023), Manuscript accepted for publication, Jun. 2023.

The following publications were invited papers:

C. Boit, T. Kiyan, **T. Krachenfels**, and J.-P. Seifert, "Logic State Imaging From FA Techniques for Special Applications to One of the Most Powerful Hardware Security Side-Channel Threats," in *2020 IEEE International Symposium on the Physical and Failure Analysis of Integrated Circuits (IPFA)*, Jul. 2020. DOI: 10.1109/IPFA49335.2020.9261000.

E. Amini, K. Bartels, C. Boit, M. Eggert, N. Herfurth, T. Kiyan, **T. Krachenfels**, J.-P. Seifert, and S. Tajik, "Special Session: Physical Attacks through the Chip Backside: Threats, Challenges, and Opportunities," in *2021 IEEE 39th VLSI Test Symposium (VTS)*, Apr. 2021. DOI: 10.1109/VTS50974.2021.9441006.

# CONTRIBUTIONS

The presented cumulative dissertation is based on multiple published co-authored works. In the following, I list my own contributions according to Section 2 (4) of the Doctoral Regulations. I had leading textual, editorial, and content responsibilities in all four publications. The list was approved by all co-authors.

## PUBLICATION 1

**T. Krachenfels**, H. Lohrke, J.-P. Seifert, E. Dietz, S. Frohmann, and H.-W. Hübers, "Evaluation of Low-Cost Thermal Laser Stimulation for Data Extraction and Key Readout," *Journal of Hardware and Systems Security*, vol. 4, no. 1, pp. 24–33, Mar. 2020. DOI: 10.1007/s41635-019-00083-9.

My contributions are as follows:

- I developed the idea for the evaluation together with Heiko Lohrke.
- I did the practical implementation, programming, and execution of the measurements. Heiko Lohrke, Enrico Dietz, and Sven Frohmann assisted in the case of technical problems with the experimental setup.
- The manuscripts's text was mainly written by myself, but in collaboration with Heiko Lohrke.

## PUBLICATION 2

**T. Krachenfels**, F. Ganji, A. Moradi, S. Tajik, and J.-P. Seifert, "Real-World Snapshots vs. Theory: Questioning the t-Probing Security Model," in *2021 IEEE Symposium on Security and Privacy (SP)*, IEEE Computer Society, 2021, pp. 1955–1971. DOI: 10.1109/SP40001.2021.00029.

My contributions are as follows:

- I developed the idea for the work in cooperation with Shahin Tajik, Amir Moradi, and Fatemeh Ganji.
- I planned and realized the experimental setup. Specifically, I prepared the implementation of the masking schemes on the FPGA, designed the experimental electrical setup, and automated the measurements with the Phemos-1000.
- The experiments on the Phemos-1000 were performed by me. Furthermore, I analyzed and evaluated the resulting image data. Fatemeh Ganji advised me on the evaluation using computer vision techniques.

- The text of the manuscript was written by me in collaboration with Fatemeh Ganji, Amir Moradi, and Shahin Tajik.

### PUBLICATION 3

**T. Krachenfels**, T. Kiyan, S. Tajik, and J.-P. Seifert, "Automatic Extraction of Secrets from the Transistor Jungle using Laser-Assisted Side-Channel Attacks," in *30th USENIX Security Symposium (USENIX Security 21)*, USENIX Association, Aug. 2021, pp. 627–644, ISBN: 978-1-939133-24-3.

My contributions are as follows:

- The concept of automated extraction of secret keys using Deep Learning techniques was developed by me, building on the basic idea of Heiko Lohrke to perform automated hardware analysis on the Phemos-1000 using Machine Learning.
- I built the experimental setup, implemented the devices under test, and executed the automated measurements. Furthermore, I processed and evaluated the image data and designed and trained the neural networks.
- Most of the manuscripts's text was written by me. Tuba Kiyan and Shahin Tajik assisted me in writing single parts of the discussion and in proofreading.

### PUBLICATION 4

**T. Krachenfels**, J.-P. Seifert, and S. Tajik, "Trojan awakener: Detecting dormant malicious hardware using laser logic state imaging (extended version)," *Journal of Cryptographic Engineering*, May 2023. DOI: 10.1007/s13389-023-00323-3.

My contributions are as follows:

- I developed the idea for the work together with Shahin Tajik.
- I built the experimental setup, performed the experiments, and evaluated and presented the results.
- The text of the manuscript was written mainly by myself. Shahin Tajik supported me in the textual fine-tuning.

# CONTENTS

# LIST OF FIGURES

# LIST OF ABBREVIATIONS

| | |
|---|---|
| AES | Advanced Encryption Standard |
| ASIC | Application-specific integrated circuit |
| BBRAM | Battery-backed RAM |
| BGA | Ball grid array |
| BPD | Backside power delivery |
| BPR | Buried power rail |
| CC | Common Criteria |
| CLB | Configurable logic block |
| CMOS | Complementary metal-oxide-semiconductor |
| CNF | Conjunctive normal form |
| CNN | Convolutional neural network |
| DPA | Differential power analysis |
| DUT | Device under test |
| EM | Electromagnetic |
| EOFM | Electro-optical frequency mapping |
| EOP | Electro-optical probing |
| FA | Failure analysis |
| FD-SOI | Fully depleted silicon-on-insulator |
| FET | Field-effect transistor |
| FF | Flip-flop |
| FIB | Focused ion beam |
| FinFET | Fin field-effect transistor |
| FPGA | Field-programmable gate array |
| FWHM | Full width at half maximum of the intensity |
| GAA | Gate-all-around |
| GND | Ground |
| GPU | Graphics processing unit |
| HIL | High-power incoherent light source |
| HT | Hardware Trojan |
| IC | Integrated circuit |
| IDE | Integrated development environment |
| IP | Intellectual property |

| | |
|---|---|
| LAB | Logic array block |
| LC | Logic cluster |
| LE | Logic element |
| LFI | Laser fault injection |
| LIO | Liquid immersion lens |
| LLSI | Laser logic state imaging |
| LSM | Laser scanning microscope |
| LUT | Lookup table |
| LVI | Laser voltage imaging |
| LVP | Laser voltage probing |
| NA | Numerical aperture |
| NIR | Near-infrared |
| ML | Machine learning |
| NMOS | N-channel metal-oxide-semiconductor |
| MOSFET | Metal-oxide-semiconductor field-effect transistor |
| MUX | Multiplexer |
| NVM | Non-volatile memory |
| OTP | One-time programmable |
| PCB | Printed circuit board |
| PDN | Pull-down network |
| PEM | Photon emission microscopy |
| PICA | Picosecond imaging circuit analysis |
| PLS | Photonic laser stimulation |
| PMOS | P-channel metal-oxide-semiconductor |
| PUF | Physically unclonable function |
| PUN | Pull-up network |
| RAM | Random-access memory |
| SAT | Boolean satisfiability problem |
| SCA | Side-channel analysis |
| SEM | Scanning electron microscopy |
| SIL | Solid immersion lens |
| SNR | Signal-to-noise ratio |
| SoC | System on a chip |
| SOI | Silicon-on-insulator |
| SPEA | Simple photonic emission analysis |
| SRAM | Static random-access memory |

| | |
|---|---|
| STI | Shallow trench isolation |
| SWIR | Short-wave infrared |
| TLS | Thermal laser stimulation |
| TPM | Trusted platform module |
| TSV | Through-silicon via |
| VCC | Supply voltage |
| VDD | Supply voltage |
| VLSI | Very large-scale integration |

# 1 MOTIVATION

At first glance, it creates a shiny, mirror-like, but almost unimposing impression. I am talking about a piece of silicon called a die or, more colloquially, a chip or an integrated circuit (IC), as shown in Fig. 1.1. It contains electrical circuitry comprised of billions of transistors in an area typically no larger than a thumbnail. Almost every technical product nowadays contains one or more of these chips, be it a washing machine, smartphone, car, or autonomous vehicle. The increasing integration and shrinkage of structures, and therefore, a price reduction of manufacturing, has enabled this development. Depending on the use case, functionality is traditionally achieved by running software on a microprocessor or designing application-specific integrated circuits (ASICs). The advantage of the latter is higher performance and power efficiency. Reconfigurable hardware, such as field-programmable gate arrays (FPGAs), offers a third possibility. As the name indicates, the configuration of FPGAs can be changed in the field, making hardware patchable and allowing faster times to market. In addition, manufacturers are combining microprocessors, FPGAs, and other specialized hardware such as graphics processing units (GPUs) and machine learning (ML) accelerators in one chip to create powerful reprogrammable systems on a chip (SoCs).

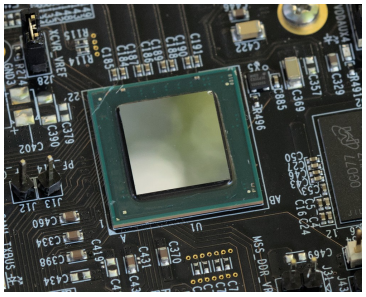


Fig. 1.1: Flip-chip packaged IC.

*Failure analysis tools for debugging integrated circuits*

Due to the complex structures in modern ICs, manufacturers and chip vendors widely use failure analysis (FA) tools to debug their circuits. Modern ICs are comprised of many metal layers on the front side of the chip, making access to the active silicon area, where the transistors reside, almost impossible. Therefore, current FA techniques access the chip through the backside. Since silicon is transparent to near-infrared (NIR) light, optical techniques can be used without physical contact with the device. Fig. 1.2 depicts the previously shown chip, imaged with an infrared camera. It reveals the inner structures of the IC, pre-

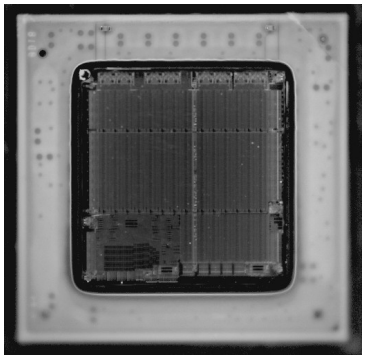


Fig. 1.2: Infrared image of the chip shown above.

viously invisible to the human eye. Different techniques exploiting the transparency of silicon to NIR light are used in practice. One class of techniques captures the photons emitted by the circuit, for instance, due to transistor switching activities. Another class used to investigate the inner structure and operations of the chip is laser-based FA tools. Using a laser scanning microscope (LSM), a laser can be positioned precisely on the location of interest on the IC or scanned over a region. Such a setup can inspect the chip without influencing it electrically. For instance, thermal laser stimulation (TLS) can establish a side channel in the device's current consumption by creating a local heat gradient. On the other hand, the laser light reflected from the device is influenced by the electrical properties of the chip and thus yields information about internal voltage levels. The corresponding methods are called optical probing techniques. All FA techniques that aim to extract the inner logic states of circuits by creating a 2-dimensional image can be classified as *logic state imaging* techniques. This work focuses on *single-trace* logic state imaging techniques, meaning that the logic state of a circuit of interest can be captured with a single measurement. Although FA techniques were initially developed for debugging ICs during the development and production phase, they can also be used by malicious entities to attack devices.

*ICs are used in critical applications*

Look at news articles from the last few years, and you will find several warnings and reports about cyber attacks on critical infrastructures. While writing this thesis, Russia wages war against Ukraine, likely using cyber attacks as a tool in modern warfare [1]. The increasing number of cyber attacks against companies and critical infrastructures rouses governments to protect their infrastructure better [2]. Vital infrastructure like power grids, train lines, mobile networks, and hospitals rely on IT infrastructure based on embedded systems containing ICs. Since most of the reported attacks are caused by weaknesses in software, the industry is focusing on securing software implementations. However, even if the software is hardened against attacks, the system is still insecure as long as the hardware is vulnerable. Especially edge computing systems or moving targets like autonomous machines are physically much more exposed than, for instance, servers in a data center. Once an adversary has access to such a system, she can perform hardware attacks at will. Consequently, the threat of hardware attacks can not be neglected for embedded systems and must be explored by industry and academia.

*Threats: extraction of secrets and malicious modifications*

The deployed devices typically contain cryptographic keys, user data, and intellectual property (IP). Once extracted, a potential adversary might gain access to protected systems and secret information or can clone devices. Furthermore, malicious entities can modify the hardware's functionality before or after manufacturing to, for instance, cause information leakage or create a backdoor or kill switch. Such modifications, called hardware Trojans (HTs), are a big topic in the

area of trustworthy electronics. The U.S. Department of Defense, for instance, has recently adopted a zero-trust approach to buying microelectronics [3]. It states that every newly acquired piece of hardware must be validated before it is used. The reason is that through untrusted or hacked manufacturers, vulnerable hardware can end up being used in critical devices, enabling an outsider to control the system. For FPGAs, an adversary can not access or change the hardware configuration during the production of the IC, as it will only be programmed by a trusted entity or even the end user. On the one hand, this can prevent IP extraction and malicious modifications. However, contrary to ASICs, the FPGA's implementation is even modifiable in the field. Therefore, if not protected well, an adversary can insert an HT at any time during a device's usage.

*Flip-chip: easy access to the chip's backside*

Due to manufacturing and performance reasons, many ICs are brought to the device in flip-chip packages today, see Fig. 1.1. In such a package, the chip's backside faces up, allowing easy access to the active area. This allows testing engineers to non-invasively analyze the chip regarding malicious modifications, such as HTs. On the other hand, adversaries can more easily deploy optical FA tools for attacks. Therefore, industry and society must deal with this double-edged sword by knowing the potential capabilities of attackers and trying to protect against them.

*Problem statement of this thesis*

Recent work has demonstrated that LSM-based FA tools can be leveraged to extract data from security primitives [4, 5] and an FPGA's configuration data [6] and decryption keys [7, 8]. Nevertheless, most IP designers and chip manufacturers do not integrate countermeasures against this class of attacks into their designs and devices. Besides, parts of the security community might think that theoretical models developed for classical side-channel resistance hold for all kinds of side-channel attacks, including optical FA attacks. On top of that, it is believed that tamper-proof memories, where the direct extraction of content is not easily possible, can reliably protect secrets stored on the device. During usage, though, the secrets will typically be loaded into volatile memories, which can be vulnerable to extraction using FA techniques. Finally, chip vendors believe that attacks using FA equipment are costly and complex and, therefore, can not be carried out by many entities.

*Research question and scientific contribution*

Together with my co-authors, in this thesis, I am examining the following main research question: *What are the threats and opportunities of laser-based logic state analysis concerning the security of modern integrated circuits?* In this regard, we investigate if theoretical models that ensure side-channel resistance still hold for optical attacks. We implement protected versions of a cryptographic core on an FPGA and apply a single-trace optical probing technique to investigate the effectiveness of the protection. This is the first application of laser logic state imaging (LLSI) in the hardware security field. Furthermore, we study how

computer-aided secrets extraction can be accomplished and how practical automated reverse-engineering approaches are. In this respect, we first conduct automated single-trace measurements on memory elements of microcontrollers and FPGAs. Then, we apply deep learning techniques to the obtained images to investigate whether the secret keys contained in these memories can be extracted automatically. In addition, we investigate if tools are available for lower prices than expected by building and testing a low-cost setup for laser stimulation attacks. Finally, we investigate how logic state imaging techniques can detect tampering with the hardware design. In this regard, we implement small configuration changes on an FPGA and compare the results of unmodified and modified hardware. Additionally, we test the approach on HT benchmarks available online.

*Thesis structure*

This work follows a cumulative format, meaning that the related publications are reprinted as originally published. Therefore, their layout clearly and by requirement from the university differs from the main layout of this thesis. After Chapter 2, which provides background information on ICs and laser-based circuit analysis, each of the following four subsequent chapters (3-6) presents and discusses one of the publications listed at the beginning of page ix. One downside of a cumulative dissertation is that partial doubling of background information between the publications can not be avoided, which might require the reader to jump over already-known descriptions and explanations. The advantage is that each chapter is self-contained and can be read independently.

Accordingly, Chapter 3 shows that the t-probing security model must be revisited when considering optical logic state imaging techniques. In Chapter 4, we evaluate if it is possible to skip the tedious IC reverse-engineering and to directly extract secrets from the chip in an automated fashion. Chapter 5 presents a low-cost setup for TLS that can extract secrets from ICs. In Chapter 6, we show how to apply logic state imaging techniques for detecting HTs on FPGAs. Chapter 7 is devoted to discussing the overall findings of this thesis by considering newly published literature, giving insights into potential future work, and introducing future challenges for laser-based logic state extraction. Finally, Chapter 8 summarizes and concludes this thesis.

# 2 BACKGROUND

This chapter gives background information not captured in detail in the scientific publications but may be of interest to readers outside the fields of electrical engineering and hardware security. It covers the composition of digital ICs and laser-based analysis of devices.

## 2.1 VERY LARGE-SCALE INTEGRATION

Transistors are the main building blocks of any digital IC. Today's chips integrate billions of transistors that implement different functionalities. The process of creating such complex ICs is called very large-scale integration (VLSI). In standardized, automated processes, a logic design is translated into a chip layout definition, which is then used as a floor plan to manufacture the chip. Many steps are required to create a functional device. Typically beginning with a uniformly doped silicon wafer[1], numerous sequential photolithography, ion implantation, deposition, etching, oxidation, and polishing steps are performed [9, pp. 180 ff.]. The result is a wafer with field-effect transistors on its top, see Fig. 2.1. This area with the transistors is called the active area because the actual switching activity and logic functionality is located here. After the transistors have been created, a stack of metal layers is added that connects the individual transistors with each other and the chip's front side, where pads are created to route the signals to the outside.

Multiple instances of a design are placed next to each other on a larger wafer. After manufacturing, this wafer is cut into pieces, and the individual dies are put into a package. The package connects the

1 Doping will be explained in the following section.

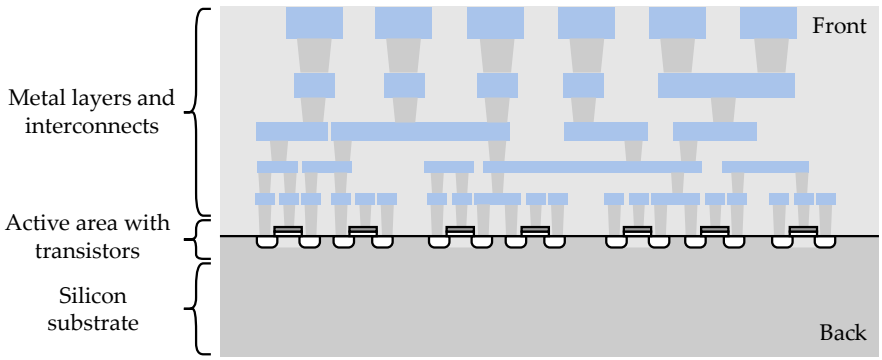
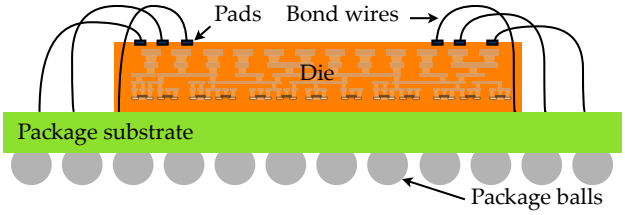
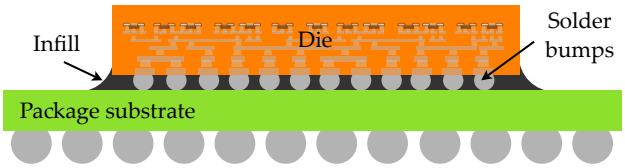



Fig. 2.1: Schematic of a chip cross-section.




(a) Wire bonding: chip's backside facing down




(b) Flip-chip: chip's backside facing up

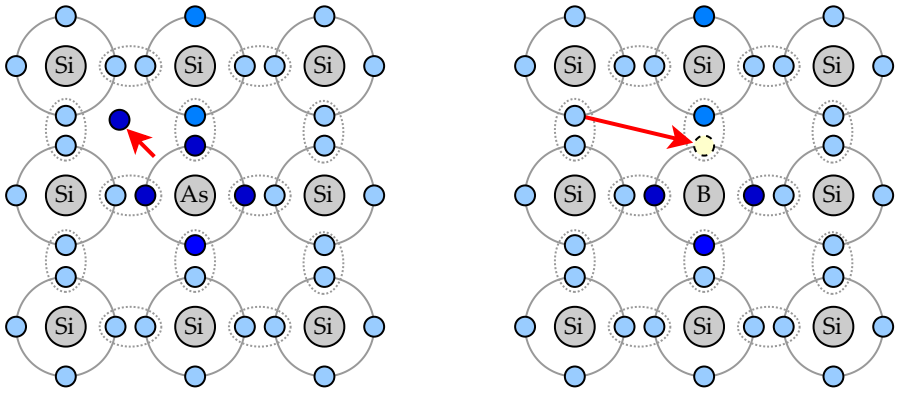
Fig. 2.2: Comparison of wire and flip-chip bonding in a ball grid array (BGA) package.

delicate pads to a larger footprint which can then be mounted on a printed circuit board (PCB). Depending on the application, different package types are used. Traditionally, chips were wire-bond to the package, meaning that the front side faces up and thin copper wires connect the pads with the package pins, see Fig. 2.2a. The die is molded into a plastic case afterward. Modern chips are often manufactured in flip-chip packages, meaning the silicon backside faces up, and the die is connected using solder bumps to the package, see Fig. 2.2b. Due to performance, size, cost, and better compatibility, flip-chip packages are becoming more popular and widely used [10]. The advantage of this package type for laser-based analysis is that the chip backside is often directly accessible.

### 2.1.1 *Field-Effect Transistors*

*Silicon doping: introducing impurities*

The essential building blocks in a VLSI circuit are metal-oxide-semiconductor field-effect transistors (MOSFETs). Two types of transistors are typically used to implement digital CMOS circuits: n-channel metal-oxide-semiconductor (NMOS) and p-channel metal-oxide-semiconductors (PMOSs) transistors. They are manufactured using negatively diffused silicon rich in electrons (n-doped) or positively diffused silicon rich in positively charged holes (p-doped). Doping means that impurities are introduced into the silicon. Silicon atoms have four valence electrons that create a crystal lattice by forming covalent bonds between their valence electrons. N-doped silicon is created by implanting donor atoms with one valence electron more than silicon, such as phosphorus or arsenic. In this way, one electron of the donor stays weakly bonded. If enough energy is provided, e.g., by an electric field, the electron can be liberated and freely move to act as a charge




(a) n-type silicon where donor impurities add free electrons that can move through the silicon

(b) p-type silicon where acceptor impurities create free holes that attract electrons from nearby bonds

Fig. 2.3: Silicon crystal structure for n- and p-doped substrates.

carrier, see Fig. 2.3a. These electrons are referred to as free electrons. Similarly, p-doped silicon is created by implanting acceptor atoms with one valence electron less than silicon, for instance, boron or aluminum. Therefore, one bond between the acceptor and silicon atoms stays unsatisfied. The missing electron is called an electron hole, which attracts electrons from nearby covalent bonds, see Fig. 2.3b. When an electron moves, this will create another hole, starting a chain-like process, effectively moving the hole around the silicon. Consequently, the holes can freely move and act as charge carriers.

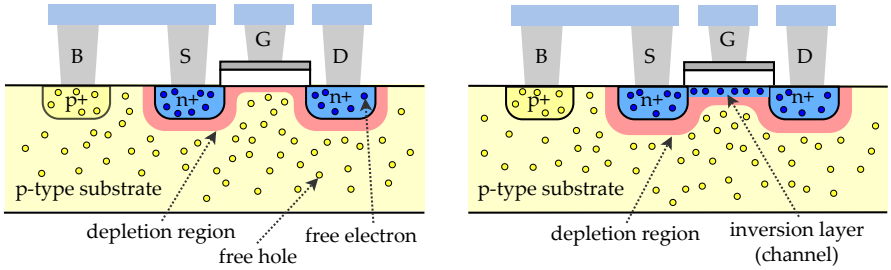
*The structure of a field-effect transistor*

A MOSFET is implemented as a channel area with drain and source connections at each end, see Fig. 2.4. The drain and source areas are highly doped, as the "+" indicates. The channel is influenced by the electric field generated by a gate electrode made from polysilicon or metal, which is electrically isolated from the channel. Depending on the gate-source voltage, carriers are attracted or depleted from the channel region under the gate. For digital circuits, typically enhancement field-effect transistors (FETs) are used, meaning that the channel is high-ohmic or non-conducting when a zero gate-source voltage is applied.

An NMOS transistor consists of an area of p-type silicon as the channel that separates two sections of n-type silicon: the drain and source, see Fig. 2.4. Due to the p-n junctions between the source and drain and the channel, free carriers are depleted from their boundaries even if no voltage is applied. This is because free electrons from the n+ doped area diffuse into the p-substrate to fill a hole. Conversely, free holes diffuse into the n+ doped area where the bond is satisfied with a free electron. The area is called the depletion region, as no free electrons or holes are present in that area that could cause a current flow.

*Subthreshold region: no free carriers in the channel*

If a small positive gate-source voltage is applied, electrons are




(a) Subthreshold region (high-ohmic channel, $V_{GS} < V_{th}$)

(b) Linear region (low-ohmic channel, $V_{GS} \leqslant V_{th}$, $V_{DS} < V_{GS} - V_{th}$)

Fig. 2.4: MOSFET operation modes relevant for steady-state analysis. B: Bulk, S: Source, G: Gate, D: Drain, $V_{th}$: Threshold voltage.

attracted toward the gate and fill the holes, effectively pushing the free holes in the p-type substrate further away from the gate. Consequently, no free carriers are in the gate area, meaning the depletion regions have expanded and continuously reach from drain to source, see Fig. 2.4a. The transistor is then said to be in the subthreshold region, where it is switched off, and theoretically, no current can flow between the drain and the source. Nevertheless, some electrons at the source can enter the channel and flow to the drain, which causes a small leakage current flow. The gate-source voltage has an exponential influence on that leakage current.

*Linear region: free charge carriers in the channel*

At a specific gate-source voltage, the threshold voltage ($V_{th}$), more electrons from the bulk silicon have been attracted so that the p-type channel area has been changed to n-type. The channel is said to be inverted. At this point, enough carriers have been accumulated that a channel can form, which begins to operate as a low-ohmic resistor and a current can flow. Due to the highly doped source area, more carriers can move into the channel region. The transistor is then said to be in the linear region because the gate voltage, relative to both the drain and source voltages, has a linear influence on the current flow.

*Saturation region: channel is pinched off*

The third operating region of the MOSFET is saturation, where mainly the gate voltage controls the current flowing through the channel. It occurs if the drain-source voltage exceeds the saturation voltage ($V_{DS} > V_{GS} - V_{th}$). In the saturation region, no channel exists near the drain region anymore; the channel is said to be pinched off. Nevertheless, through the high electric field, a current continues to flow with the charge carriers being more spread into the substrate. In digital logic, transistors are used as a switch, meaning that the drain-source voltage is approximately zero when the transistor is on, as no load is driven. Consequently, the saturation region is irrelevant when studying static states of digital logic, as in this work. Nevertheless, it should be noted that the transistors shortly enter the saturation region when switching to another logic state because the internal capacities of the logic gate are charged or discharged. Readers interested in details

on the static conditions in a MOSFET may be directed to [11, pp. 92 ff.] or [9, pp. 140 ff.].

For the PMOS, the behavior can be explained analogously to the NMOS transistor with the difference that holes are the charge carriers. Its channel is made of n-type silicon, and the drain and source area is from p-type silicon. The electric field attracts holes toward the gate if a negative gate-source voltage is applied. The electrons are pushed away from the channel so that more atoms with unsatisfied covalent bonds remain. Once the applied voltage is strong enough, a conductive channel forms. Due to the different mobility between electrons and holes, the electrical characteristics are different between NMOS and PMOS transistors. To compensate for that, the dimensions of the transistors, such as the channel length, can be adapted.

It should be noted that the transistor has a fourth terminal, which is connected to the bulk silicon. On that terminal, typically, the same voltage as to the source of the transistor is applied to avoid a forward biasing of the diodes from the drain/source to the substrate and well. Consequently, the bulk connection is required to make the transistor operate as intended.

### 2.1.2 *CMOS Technology*

*Complementary transistors: low static power consumption*

Most digital ICs are nowadays manufactured in a complementary metal-oxide-semiconductor (CMOS) technology. It uses pairs of NMOS and PMOS transistors to implement logic gates with different functionality, such as inverters, NAND and NOR gates, and memory cells. The transistors connected between the output of the logic gate and the supply voltage (VDD)[2] are summarized as pull-up network (PUN), typically implemented by PMOS transistors. The complementary part typically consists of NMOS transistors connected between the output of the logic gate and ground (GND) and is called pull-down network (PDN). The result of this and the main characteristic of CMOS logic is that, theoretically, only a current can flow between VDD and GND while switching, when the transistors in both the PUN and PDN are conducting. Therefore, in a static state, no power is consumed. Compared to non-complementary logic styles like NMOS logic, CMOS has dramatically reduced power consumption and thus was already used since the 1980s. Most of today's chips, above 95 % as of 2011, are manufactured using CMOS technologies [9]. In order to prevent a leakage current flow between neighboring transistors, today's processes implement shallow trench isolation (STI). It separates transistors by a trench that reaches into the bulk silicon, filled with dielectric (electrically insulating) material such as silicon dioxide.

*Different process varieties and wafer types*

Another reason for the dominance of CMOS technologies is that they can be manufactured with a low number of defects on a small

2 VDD and VCC are typically used interchangeably for the positive supply voltage.

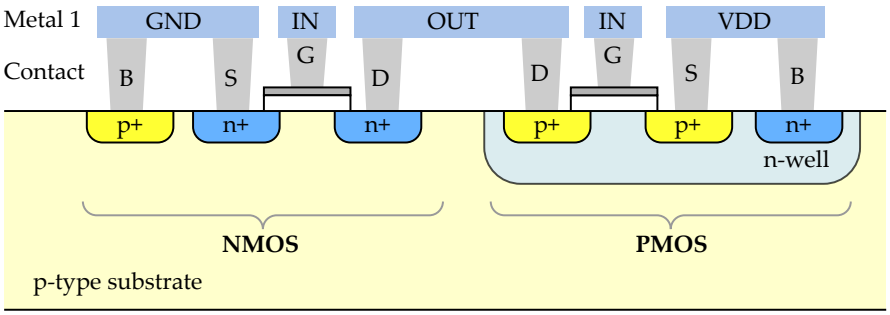



Fig. 2.5: Simplified cross-section of a CMOS inverter manufactured in an n-well process. B: Bulk, S: Source, G: Gate, D: Drain.

chip area. This has allowed the continuous shrinking of transistors until today. One aspect of its good manufacturability is that NMOS and PMOS transistors can be integrated into a single bulk wafer that is uniformly doped. Most of today's chips are built on a p-doped wafer, whereas the NMOS transistors can be easily integrated by adding n-doped drain and source areas. For the PMOS transistors, an additional area of n-doped silicon has to be added, called the n-well, see Fig. 2.5. Note that CMOS can also be implemented on an n-type substrate with p-wells or a twin- or triple-well [9]. The latter two allow separate optimization for n-type and p-type transistors [12]. However, due to the lower cost of wafers and fewer process steps, the n-well process is preferred in most modern technologies. It should further be noted that instead of a bulk silicon wafer, also heavily doped silicon wafers with an epitaxially grown layer to avoid latch-ups can be used [9, p. 180]. Furthermore, a silicon-on-insulator (SOI) wafer that electrically separates the transistors from the bulk silicon can be used to improve performance. The applicability of laser-based analysis on these wafer types will be discussed in Section 2.2.1 and Section 7.4.2. Finally, it should be noted that the previous description does not contain all details of the CMOS technology. Its purpose is instead to give the reader the most important principles and facts.

*Technology node generations*

Driven by the demand for more computational power and higher efficiency, CMOS technologies have evolved over the years. Higher performance was in the past mainly achieved by increasing the logic density in the chip, involving a downscaling of the transistors. This continued shrinkage has been reflected in the so-called technology nodes, named after length units. Until approximately the 45 nm technologies, the name indicated the transistor sizes. More precisely, it referred to the contacted poly pitch, which gave information about the gate length. Since the channel length of the transistors could not be reduced further, other aspects than the channel length were shrunk to achieve higher transistor densities. In the most recent technology nodes, the node name became an artificial marketing number, basically still trying to follow Moore's law that predicted that the transistor den-

sity would double every 24 months. However, transistor shrinkage can no longer fulfill Moore's law, and the shrinkage is slowing down [13]. Additionally, technologies with the same name from different companies do not have comparable transistor sizes or densities. The actual sizing of current and future technologies and their influence on the applicability of laser-based analysis of ICs will be discussed in more detail in Section 7.4.1. In general, the small structures are stressing the physical limits of IC manufacturing more and more, requiring high debugging efforts. Consequently, the small technology sizes are also challenging the FA community to come up with debugging tools offering higher resolutions.

*A word on 3D transistor designs*

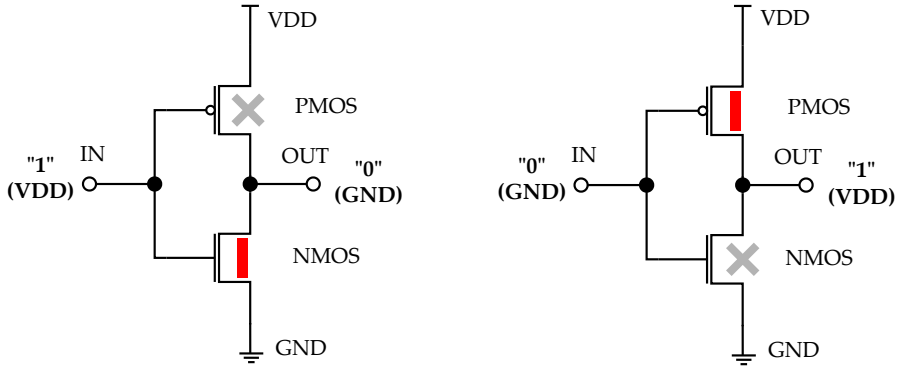
The transistor structure described previously is called a planar CMOS technology, meaning that the gate is placed as one flat block onto the channel. Today's chips manufactured with 20 nm technologies and larger are typically still produced in planar CMOS technology. However, due to the continuous shrinkage of transistor sizes, the leakage current flowing while the transistor is switched off became a severe problem. Therefore, in smaller technologies, there is a shift towards 3D transistor structures to achieve further shrinkage while keeping the leakage current of the transistors in a tolerable range. These advanced technology nodes can only be manufactured by a few companies worldwide, like TSMC, Intel, and Samsung. For further information on 3D transistor designs, please refer to Section 7.4.2. Note that all devices investigated in this work were manufactured in planar CMOS technology.

*CMOS inverter and combinatorial logic*

The inverter is the most simple logic gate in the CMOS technology, consisting of one PMOS in the PUN and one NMOS in the PDN. Fig. 2.6 shows the circuit diagrams for the two static logic states. When the input is logic `1`, the PMOS has a high-ohmic channel and the NMOS a low-ohmic channel, effectively pulling the output to GND (logic `0`). If logic `0` is applied to the input, the states are flipped around, and the output is pulled to VDD (logic `1`). Two consecutive inverters are called buffer gate, which is used to convert the potentially degraded input voltage levels to a strong `0/1` or to provide a higher driving strength at the output of the buffer. These logic gates are called combinatorial logic, as their functioning is not influenced by a clock signal, and the output changes immediately when the input changes. Other combinatorial logic gates, like NAND and NOR gates, contain more than one transistor in the PUN and PDN.

*CMOS memory cells and sequential logic*

Combinatorial logic gates can not store a value over time. Memory cells like static random-access memory (SRAM) cells and flip-flops (FFs) are used for that purpose. By implementing two cross-coupled inverters, where the output of the first inverter is connected to the input of the second inverter and vice versa, the logic value is retained in the cell as long as it is powered. For writing new values and reading the current value, further transistors are needed. Memory elements are




(a) Input = `1`, Output = `0`

(b) Input = `0`, Output = `1`

Fig. 2.6: Circuit diagrams of a CMOS inverter in the two different steady states.

typically only updated upon the edge of a clock signal, and therefore, they are referred to as sequential logic. Since CMOS memory cells will be described in detail in the publications contained in this work, please refer for more information to the publication in Chapter 3 on page 28 or in Chapter 4 on page 47.

*How the state of a transistor can be extracted*

Logic state imaging techniques, as presented in the following section, can extract the state of a transistor, i.e., can detect whether it is in the on- or off-state. Since CMOS logic consists of complementary pairs of transistors, the only information required to extract the logic state is which transistors are switched on and which are switched off. The techniques used in this thesis can, on the one hand, heat a transistor using a laser, which influences the power consumption of the device depending on the transistor state. On the other hand, the number of free carriers in the transistor's channel can be measured optically, which makes it possible to distinguish between the different states of a transistor.

## 2.2 LASER-BASED ANALYSIS OF INTEGRATED CIRCUITS

This section covers the principles of laser-based analysis of ICs using FA microscopes and gives an overview of the main setup used in this work.

### 2.2.1 *Interaction of Light with Silicon*

*Reflection, transmission, and absorption of light*

When using microscopy to investigate semiconductors, different effects have to be considered. A light beam that hits a surface can be reflected or refracted. Usually, a portion of the light is reflected and a portion is refracted, depending on the properties of the involved mediums. Refraction can be split into two parts; the transmission through the

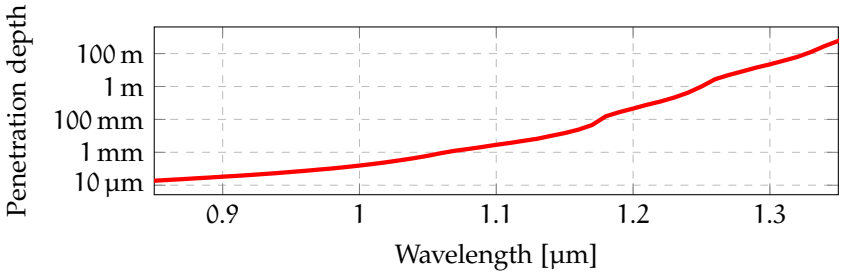



Fig. 2.7: Penetration depth in intrinsic (undoped) silicon at 300K. Data from [17].

medium and the absorption by the medium. The material properties and the wavelength of the light with the connected photon energy dictate the portion of each effect. The intensity of a light beam propagating in a medium decreases typically exponentially with increasing distance $d$ [14]:

$$I(d) = I_0 \cdot e^{-\alpha(\omega)d} , \tag{2.1}$$

where $I_0$ is the intensity of the light before it enters the material and $\alpha(\omega)$ is the absorption coefficient that depends on the angular frequency of the wave. Note that $\omega = 2\pi f$ and $\lambda = v(\lambda)/f$ where $\lambda$ is the wavelength, $f$ is the frequency, and $v$ is the propagation speed or phase velocity of light in the medium. This variation of the speed of light depending on the wavelength is called dispersion.

The ratio between the speed of light in vacuum $c$ and the phase velocity $v$ is defined as a medium's absolute refractive index $n = c/v$. In a nutshell, the refractive index defines how much the path of the light is bent when entering one medium from another and how much of the light is reflected. Note that the wavelength of a light source is typically given for vacuum or air, where $n = 1$ and $n \approx 1$, respectively. Due to the change in phase velocity, the wavelength in a medium will change with the medium's index of refraction as $\lambda = \lambda_0/n$, where $\lambda_0$ is the wavelength of the light in vacuum [15].

Coming back to the absorption, one derived number from $\alpha(\omega)$ is the penetration depth which is given by

$$\delta = \frac{1}{\alpha(\omega)} , \tag{2.2}$$

and specifies the depth where the light intensity has decreased to $1/e$ (37 %) of its original value [16]. This number indicates the required sample preparation concerning the thickness of the bulk silicon.

*The bandgap of silicon as an important factor*

Given the above formulae, the wavelength, which is inversely proportional to the photon energy of the light, is a vital aspect to consider when relying on the propagation of light in a material. A distinctive point in the wavelength-dependent absorption characteristics of silicon is connected to its semiconducting property. Silicon has a band gap

at around 1.12 eV, which corresponds to the photon energy of light with a wavelength of around 1.1 µm. Photons with energies above the silicon band gap can be absorbed by the silicon by liberating an electron that can then act as a charge carrier. Light with photon energies smaller than the silicon band gap ($\lambda > 1.1\,\mu m$) interacts mainly thermally with the silicon. While the injection of photocarriers is a desired effect for fault injection techniques, optical side-channel techniques preferably rely on light with photon energies smaller than the silicon bandgap to not unintentionally change the behavior of the device under test (DUT).

In general, silicon is transparent to light in the near-infrared (NIR) region. Especially wavelengths above the silicon bandgap can pass silicon very well with a penetration depth of a few tens of meters for a wavelength of around 1.3 µm [17], see Fig. 2.7. When coming closer to the bandgap, the absorption increases, and the penetration depth drops to a few millimeters. In the visible light spectrum, the corresponding penetration depth reduces to only a few micrometers. Depending on the applied wavelength, thinning the silicon backside of the chip under investigation can be necessary, as the typical thickness of the bulk silicon is a few hundred micrometers.

*Influence of doping on the absorption*

Besides the wavelength, also the doping concentration influences the absorption [18]. Since the doping concentration of the bulk silicon is relatively low, NIR light with wavelengths above 1.1 µm can pass the silicon well. However, for very high doping concentrations, silicon becomes almost opaque in the NIR range [19]. Since wafers with an epitaxially grown silicon layer are typically based on a highly doped bulk wafer (cf. Section 2.1.2), the optical analysis of devices using this type of wafers might not be possible without aggressive thinning of the silicon backside. Nevertheless, this kind of wafer does not seem to be broadly used, as all devices investigated in this work were built on lightly doped bulk silicon.

### 2.2.2 *Laser Scanning Microscope*

Since the chip is transparent in the NIR region, lasers in that range of wavelengths can be used to analyze the device. This is typically achieved with a laser scanning microscope (LSM), which incorporates laser light sources that can be focused through a microscope objective on the sample [20]. A scanner, typically consisting of galvanometric mirrors, can scan the laser beam over the field of view. Fig. 2.8 shows a schematic of such a system, where the different lenses focus and defocus the laser light, and the beam splitter allows the light reflected from the device to enter a detector. This detector is typically an avalanche photodiode that converts the incoming light into electricity, which is used in the LSM to measure the reflected light's intensity. LSMs are typically designed so that only light that is within the focal plane can

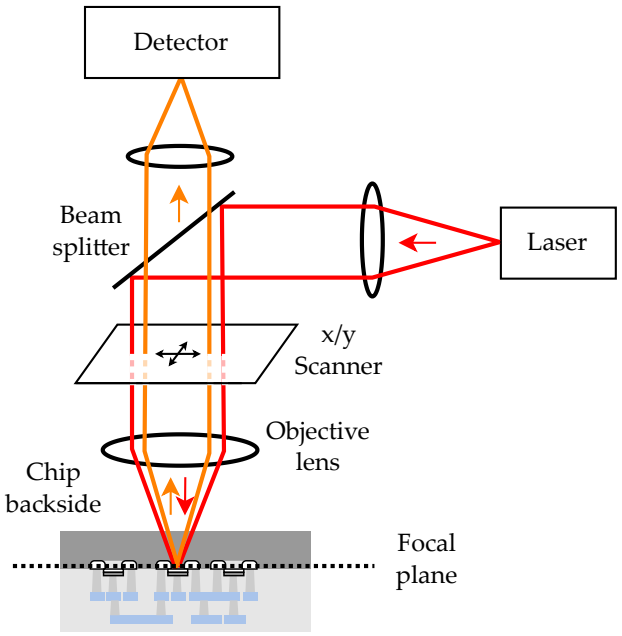



Fig. 2.8: Schematic of an LSM setup. Figure based on own illustration previously published in [22].

reach the detector, which is then called a confocal LSM [21]. The light has to pass so-called pin-holes that block light not coming from the focal plane.

*Optical resolution as a limiting factor for laser-based analysis*

One important aspect of microscopy is the optical resolution. The resolution R can be defined as the minimum distance at which two points can be distinguished from each other. For a microscope system, it can be defined by the Rayleigh criterion as

$$R = 0.61\lambda/NA \,, \tag{2.3}$$

where $\lambda$ is the wavelength of the light and NA is the numerical aperture. The NA depends on the angle of light accepted by a lens and the index of refraction of the medium the lens is operated in ($n = 1$ for air) [23]. Although different definitions for optical resolution exist, all of them contain the relationship between the wavelength and the NA as $R \sim \lambda/NA$.

An intuitive way to explain the optical resolution for an LSM is using the beam diameter. Depending on the optics and the wavelength of the light, the laser beam focused on the DUT can only reach a specific minimum diameter. Therefore, if the transistor density is so high that the beam can only cover more than a single transistor, the resolution is not high enough to resolve a single transistor. Due to the Gaussian distribution of the beam, the highest intensity is in the center of the beam. One possibility to calculate the laser beam diameter is calculating the Airy disk's diameter, which assumes a perfectly focused beam. By definition, the diameter coincides with the full width at half maximum of the intensity (FWHM), i.e., the distance

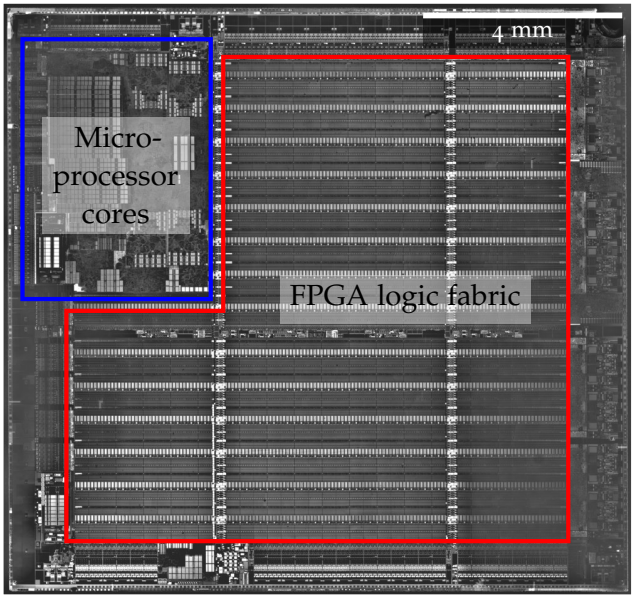



Fig. 2.9: Reflected light image of a Microsemi PolarFire SoC, consisting of multiple images acquired with a 5× lens. The regular structures of the logic fabric can be distinguished from the more irregular structures of the processor cores' synthesized logic.

from the beam's center where the intensity is 50 % of the intensity in the center. For a confocal microscope, the FWHM can be calculated as [23]:

$$\mathrm{FWHM} = 0.87\frac{\lambda}{\mathrm{NA}} \,. \tag{2.4}$$

For an illustration of the laser beam's distribution, see Fig. 7.1 in Section 7.3, where this definition is used to simulate optical probing measurements. Consequently, the optical resolution limits the applicability of laser-based analysis of ICs on small and densely-placed transistors. For a discussion on future challenges for FA and security analysis and an illustration of the available optical resolution versus the IC technology nodes, see Section 7.4.1 in the discussion.

*Reflected light images for analysis of functional blocks and navigation*

Due to the different materials and doping concentrations in the chip, an optical image that shows the internal structures of the chip can be acquired by analyzing the amount of reflected light over the scanning position. This image can be used to localize different functional blocks, such as memories or synthesized computation cores, and aid navigation on the chip. For instance, for the SoC shown in Fig. 2.9, the FPGA logic and microprocessor cores can be distinguished clearly. Apart from capturing an optical image from the chip, different techniques can be used to analyze the electrical signals within the chip, as discussed below.

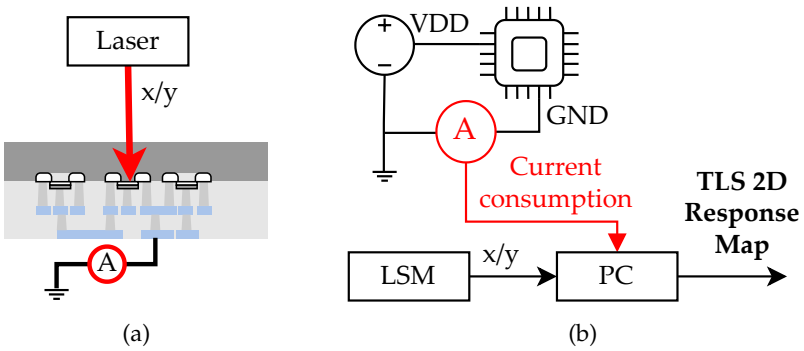



Fig. 2.10: Concept (a) and simplified electrical setup (b) for TLS. The laser is scanned over the DUT and the changes in the current consumption are measured, grayscale-encoded, and mapped onto the scanning position by PC software.

### 2.2.3 *Laser Stimulation*

One FA technique initially developed to detect faults such as short circuits or improper connections is laser stimulation. The general principle is to measure device parameters like voltage or current consumption while influencing the device using laser radiation. Scanning a region of interest with the laser can help to localize faults because the DUT behaves differently in the presence of a fault. Depending on the laser wavelength, either photocarriers are generated or mainly heat is induced by the laser beam, cf. Section 2.2.1. The former technique is called photonic laser stimulation (PLS), and the latter is thermal laser stimulation (TLS).

*Thermal laser stimulation to extract logic states*

Apart from localizing defects, laser stimulation can be used to create a data-dependent side channel in the measured device parameter. In other words, the laser's influence allows the extraction of internal logic states from the DUT [24]. To not influence the DUT's operation by the laser radiation, TLS is the preferred technique. Fig. 2.10 shows the principle of TLS where the device's current consumption is measured while scanning with a laser over the device. The measured values are grayscale-encoded and mapped onto the scanning position, which leads to a 2D response map.

*Seebeck effect causes the generation of a voltage drop*

The effect allowing to extract logic states is Seebeck voltage generation that occurs when a thermal gradient is created in silicon. Fig. 2.11 shows a MOSFET transistor in the on-state (with a low-ohmic channel) under thermal stimulation at the drain. According to the Seebeck effect, the temperature gradient created between the metal contact and the channel causes a diffusion of carriers that acts like a voltage source ($V_{Seeb.}$) [4, 25]. The same effect can be observed when stimulating the source of the transistor, though the voltage source will have the opposite sign [25]. If the transistor is in the off-state (with a high-ohmic

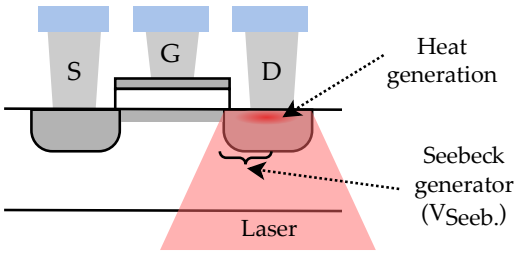



Fig. 2.11: Seebeck voltage generation by thermally stimulating the drain area of a MOSFET. Figure based on [7, 24].

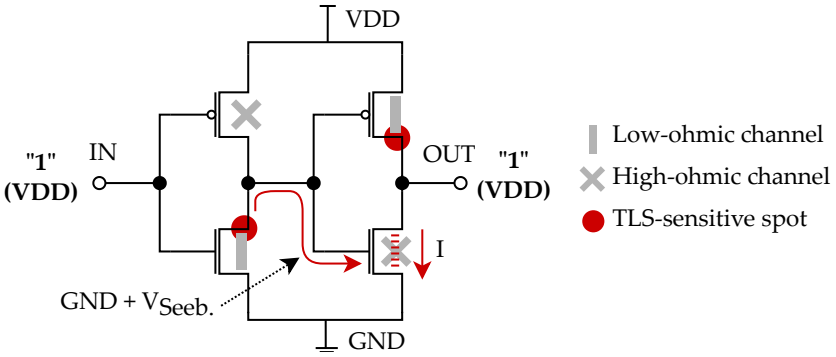



Fig. 2.12: Principle of how thermal stimulations leads to increased current consumption in a CMOS buffer, which can be measured externally. If the buffer is in the opposite state (with input `0`), the drains of the first inverter's PMOS and the second inverter's NMOS transistor are sensitive to TLS. Figure based on [4].

channel), one connection of the Seebeck generator is floating, and the voltage source is ineffective.

*Measuring the Seebeck voltage*

The voltage of the Seebeck generator is in the range of $0.2\,\mathrm{mV\,K^{-1}}$ to $0.4\,\mathrm{mV\,K^{-1}}$, which can be hard to detect at a pin of the IC, as an electrical path between the point of generation and the pin must exist [4]. In CMOS technologies, though, there is no static current path between VDD and GND because at least one transistor in that path is in the off-state (cf. Section 2.1.2). Therefore, a single inverter's state can not be extracted using TLS. However, the Seebeck voltage can be detected if applied to the gate of another transistor that is the only off-state transistor between VDD and GND. Then the changed gate voltage has, via sub-threshold operation, an exponential influence on the current flowing through the transistor. In the case of a CMOS memory cell or a buffer, it allows the direct extraction of the logic gate's state. Fig. 2.12 shows for a buffer gate how the stimulation at the drain of the first inverter's PMOS leads to an increased leakage current in the second inverter's NMOS transistor. This increase in current consumption can be measured in the device's power consumption. How a CMOS memory cell's content can be extracted using TLS will be shown in the publication in Chapter 4 on page 47.

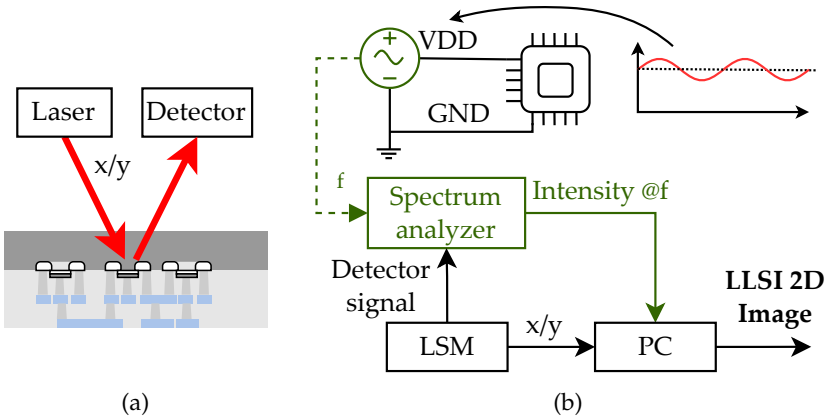



Fig. 2.13: Concept of optical probing (a) and simplified electrical setup for LLSI (b). The supply voltage is modulated, the laser is scanned over the DUT, and the reflected light is fed into a spectrum analyzer and inspected for the modulation frequency. Its output is mapped onto the scanning position.

### 2.2.4 *Optical Probing*

Apart from influencing the device with the laser and measuring a device parameter, the light reflected back from the chip can be analyzed to extract information about the voltage levels and logic states, see Fig. 2.13a. The corresponding techniques are referred to as optical (contactless) probing techniques. The origin of the optical probing signal can mainly be described by the effects of absorption and refraction due to free carriers [26, 27]. The influence of a change in the charge carrier density $\Delta N$ for the wavelength $\lambda$, the index of refraction $n$, and the absorption coefficient $\alpha$ can be calculated as [27]:

$$\Delta n = \sqrt{1 - \frac{(q\lambda)^2}{4(\pi c_0)^2 \epsilon_0} \cdot \frac{\Delta N}{m}} \quad \text{and} \tag{2.5}$$

$$\Delta \alpha = \frac{\lambda^2 q^3}{4\pi^2 c_0^3 \epsilon_0} \cdot \frac{\Delta N}{n m^2 \mu}\,, \tag{2.6}$$

where $q$ is the electron charge, $\epsilon_0$ is the permittivity of free space, $c_0$ is the speed of light in vacuum, $\mu$ is the mobility of the charge carrier, $m$ is the effective mass of the charge carrier, and $n$ is the index of refraction given as $n = n_0 + \Delta n$ with $n_0$ being the index of refraction taking dispersion into account. As can be seen from the equations, the light's absorption and refraction depend on the number of free carriers in the optical beam. Since the voltages applied to the terminals of transistors and capacitors change the number of charge carrier densities in the silicon, this explains why the voltage levels in the chip can be probed optically.

*Contactless probing of dynamic signals*

The modulation of the reflected light by different carrier concen-

trations in the chip can be used to probe the internal voltage levels at a transistor of interest. This functionality can be compared with acquiring a waveform using an oscilloscope. However, due to the weak modulation of the light, several hundred iterations of the same waveform must be integrated to achieve a reasonable signal-to-noise ratio (SNR). During such a measurement, the laser beam is parked at the position of interest, and multiple iterations of the waveform are integrated. The technique is called laser voltage probing (LVP), or when an incoherent light source is used, as in this work, electro-optical probing (EOP) [28].

*Localization of periodically switching transistors*

In order to localize areas of interest that should be probed, the beam can be scanned over the device. Then at each beam location, the reflected light is analyzed by a spectrum analyzer for a set frequency. The resulting amplitude at each scanning position is then grayscale-encoded and plotted over the position, resulting in a 2D map of the scanned area. In this way, all locations switching at a specific periodic frequency can be localized. The technique is called laser voltage imaging (LVI), or if an incoherent light source is used, electro-optical frequency mapping (EOFM) [29].

These two techniques, though, can only be used to extract the logic states from the chip if the same operation with the same data of interest can be repeated for many iterations. Consequently, memory cells' contents that are not preserved over multiple iterations can not be read out using the classical EOP and EOFM as described.

*Analysis of static transistor states*

Modulating the supply voltage creates another possibility to extract logic states. The technique is based on EOFM and is called laser logic state imaging (LLSI) [30]. Its idea is that the modulation of the supply voltage only heavily affects the carrier concentrations of low-ohmic transistors, i.e., that are in the on-state. Consequently, in this case, the logic states can be extracted without any switching activity in the circuit. However, the states of the transistors have to stay constant during the entire measurement. Fig. 2.13 shows the concept of LLSI, where a laser is scanned over the device and the reflected light is captured by a detector. The detector signal is fed into a spectrum analyzer set to the power rail's modulation frequency. The output of the spectrum analyzer is mapped onto the scanning position, which results in a 2D image of the scanned area. The supply voltage modulation is typically achieved by adding a sine wave with a peak-to-peak voltage in the range of tens to a few hundred millivolts to the normal supply voltage.

Fig. 2.14 shows how the logic state from a buffer can be extracted. Due to the modulation of the supply voltage, the concentrations of free carriers are modulated as well. This effect gives a strong LLSI signal only for on-state transistors, which allows deducing the logic state of the buffer [30].

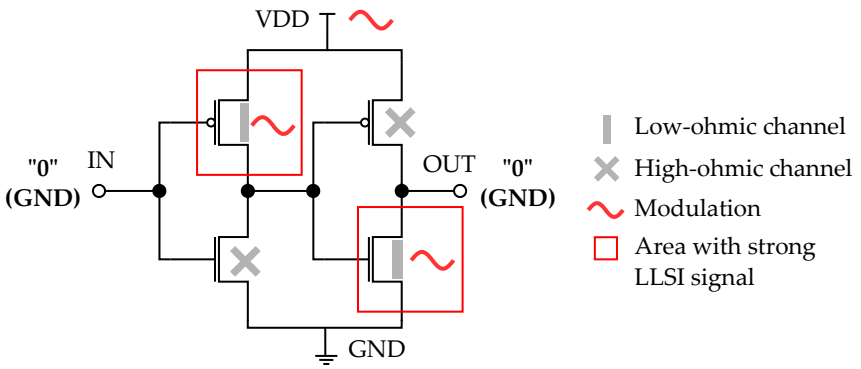
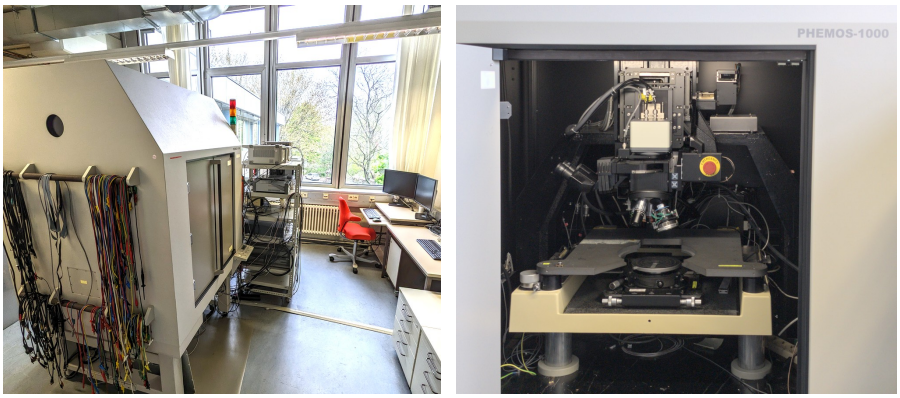



Fig. 2.14: Principle of LLSI logic state extraction on a CMOS buffer. Only on-state transistors show a clear signature in the LLSI image. Figure based on [30].

### 2.2.5 *Phemos-1000 Failure Analysis Microscope*

The LSM used in this work is a Hamamatsu Phemos-1000 failure analysis microscope. It incorporates different light sources, such as a 1.3 µm laser for thermal stimulation and a 1.3 µm high-power incoherent light source (HIL) for optical probing. Furthermore, it offers different lenses: a 0.5× macro lens, and 5×, 20×, 50×, and 100× magnification lenses. The 50× lens is optimized for optical probing and has a silicon thickness correction feature. The highest optical resolution can be achieved with our 50× lens, which has an NA of 0.71. Thus, when using a wavelength of 1.3 µm, the FWHM (Equation 2.4) of the laser spot is around 1.6 µm, which corresponds to an optical resolution according to the Rayleigh criterion (Equation 2.3) of 1.12 µm. Although the beam is limited in its minimum diameter, it can be scanned on a smaller grid over the device. The granularity of the scan can be controlled by setting the scanner zoom. A scanner zoom of 2× to 8× can be configured for the laser scanner, which results in a smaller area being scanned




(a) Overview of the Phemos-1000 lab (b) LSM inside the enclosure

Fig. 2.15: Hamamatsu Phemos-1000 setup used for this work.

with higher precision. Next to the LSM functionality, the Phemos also houses a camera for spatial photon emission measurements.

The entire setup is shown in Fig. 2.15a. A dark box enclosure contains the microscope itself, providing shielding from the laser radiation for the user and blocking light from the outside, see Fig. 2.15b. Samples can be placed under the microscope on a vacuum chuck. The operation of the Phemos can be fully controlled from a PC, which runs the control software. The racks next to the dark box contain, on the one hand, components that belong to the Phemos, such as the optical probing controller. On the other hand, there are further instruments to control the DUT, such as waveform generators, amplifiers, and an oscilloscope.

# 3

# REAL-WORLD SNAPSHOTS VS. THEORY

## 3.1 SIDE-CHANNEL ANALYSIS AND MASKING SCHEMES

Classical side-channel analysis (SCA) attacks have been known and used for many years. Researchers have explored different ways to extract secret information, such as keys from cryptographic operations. By observing the power consumption or the electromagnetic (EM) emanation, as the most prominent examples, it is possible to extract the secret directly from the captured traces or with the help of statistical analyses. In this regard, after recording millions of traces of the same algorithm, techniques such as differential power analysis (DPA) can leverage the weakest data dependencies to obtain the secret [31]. Due to the relevance of attacks for devices on the market, companies and academia have started developing countermeasures in the early 21st century [32, 33].

*Countermeasures against side channel attacks*

Most countermeasures were designed to remove exploitable data dependencies from the side channel. While so-called hiding countermeasures try to completely eliminate leakage caused by computation, masking countermeasures try to make the leakage independent from the processed data. Masking schemes make use of randomization and concepts known from multiparty computation. They split the computation in a randomized way so that observing the individual parts of the computation to reconstruct the secret becomes more challenging or even impossible [34]. Probing security models try to prove the effectiveness of such schemes by assuming limited capabilities of the attacker, caused by physical restrictions. An example and one of the earliest models used to assess the security of masking schemes against probing attacks is the t-probing model [34, 35]. The assumption is that attackers only have a limited number of independent probes available to observe the circuit's operation. Only by observing all shares in a statistically independent way, the secret can be reconstructed.

*Masking not yet standardized*

Masking schemes have become one of the most promising protections against SCA attacks. However, although masking is among the countermeasures employed in today's secure smartcards, such as banking cards, it is not yet standardized. The National Institute of Standards and Technology (NIST) in the U.S. is about to form a masked circuits library, but they are currently not taking any actions toward standardization [36]. Nevertheless, countermeasures against SCA attacks are part of the Common Criteria (CC) defined by certification bodies in several countries, for instance, by the National Cybersecurity Agency of France (ANSSI) and The Federal Office for

Information Security of Germany (BSI) [37]. Consequently, manufacturers increasingly integrate masking schemes into their recent devices but keep details about it confidential.

*Research question*

Considering the growing application of masking schemes, we started to investigate whether these schemes can protect against optical SCA attacks. It seems obvious that the required number of independent probes to observe the individual shares limits an adversary conducting power or EM analyses. For power analysis, typically only one point for measuring the consumed power is available. Nevertheless, statistical analyses also allow higher-order attacks and, therefore, potentially even attacks against masked implementations with a low number of shares [38]. EM analysis can allow the positioning of multiple probes on the IC. However, spatial limitations restrict the number of independent probes to a low number [39]. For optical logic state imaging techniques, however, these limitations might not be valid anymore. One technique of interest that had not yet been leveraged for security purposes is called LLSI (cf. Section 2.2.4). It is a single-trace logic state imaging technique. The idea is that when halting the clock, the logic states of, for instance, data registers can be read using optical probing in a single measurement. In the following work, we have investigated the threat of LLSI for masked implementations of cryptographic cores.

## 3.2 PUBLICATION

The publication is reprinted subsequently. It was presented at the *2021 IEEE Symposium on Security and Privacy (SP)* and published in [40]. The logic state images acquired in connection with this publication are available in [41].

# Real-World Snapshots vs. Theory: Questioning the $t$-Probing Security Model

Thilo Krachenfels*, Fatemeh Ganji†§, Amir Moradi‡, Shahin Tajik†§ and Jean-Pierre Seifert*
* Technische Universität Berlin, Chair of Security in Telecommunications, Germany
† Worcester Polytechnic Institute, Department of Electrical and Computer Engineering, USA
‡ Ruhr-Universität Bochum, Horst Görtz Institute for IT-Security, Germany

***Abstract*—Due to its sound theoretical basis and practical efficiency, masking has become the most prominent countermeasure to protect cryptographic implementations against physical side-channel attacks (SCAs). The core idea of masking is to randomly split every sensitive intermediate variable during computation into at least $t+1$ shares, where $t$ denotes the maximum number of shares that are allowed to be observed by an adversary without learning any sensitive information. In other words, it is assumed that the adversary is bounded either by the possessed number of probes (e.g., microprobe needles) or by the order of statistical analyses while conducting higher-order SCA attacks (e.g., differential power analysis). Such bounded models are employed to prove the SCA security of the corresponding implementations. Consequently, it is believed that given a sufficiently large number of shares, the vast majority of known SCA attacks are mitigated.**

**In this work, we present a novel laser-assisted SCA technique, called Laser Logic State Imaging (LLSI), which offers an unlimited number of *contactless* probes, and therefore, violates the probing security model assumption. This technique enables us to take *snapshots* of hardware implementations, i.e., extract the logical state of all registers at any arbitrary clock cycle with a single measurement. To validate this, we mount our attack on masked AES hardware implementations and practically demonstrate the extraction of the full-length key in two different scenarios. First, we assume that the location of the registers (key and/or state) is known, and hence, their content can be directly read by a single snapshot. Second, we consider an implementation with unknown register locations, where we make use of multiple snapshots and a SAT solver to reveal the secrets.**

***Index Terms*—EOFM, Hardware Security, LLSI, Masking, Optical Probing, Probing Model, Side-Channel Analysis**

## I. Introduction

Electronic embedded devices are an indispensable part of our today's connected systems. To ensure the confidentiality and integrity of processed data in these systems, strong cryptography is needed. But even in the presence of such cryptographic primitives, the security of deployed devices still can be compromised by attackers, who can gain access to these devices and thus launch physical attacks. Side-Channel Analysis (SCA) attacks are examples of such physical threats, which are hard to detect and mitigate due to their most often passive nature. SCA attacks exploit the inevitable influence of computation and storage on different measurable quantities on a device, such as timing [1], power consumption [2], Electro-Magnetic (EM) emanation [3], and photon emission [4].

Several countermeasures have been proposed to defeat SCA attacks. Among them, masking has been shown to be the most effective one that can be applied to most cryptographic schemes. Masking schemes are based on the principle of splitting the computation over several randomized and independent shares. To prove the security of the masked implementations, the $t$-probing model was first introduced in the seminal work of Ishai et al. [5]. In this model, the adversary is assumed to be limited by the number of $t$ probes available for observing the computation on wires. In such a scenario, we require to employ at least $t + 1$ shares to assure that the adversary cannot learn any sensitive information from $t$ observations. In practice, assuming such a limit is quite plausible.

For instance, due to the lack of spatial distance in case of invasive micro/nano-probing attacks or EM analysis, we expect the number of possible probes to be very limited. Moreover, the higher number of probes leads to a more expensive probe station, and hence, the cost of multi-probe stations is another limiting factor for the adversary. Currently, the most advanced commercially-available nano-probe station consists of at most eight needles [6]. Similarly for EM stations, the largest setup, which has been reported so far only in [7], makes use of three simultaneous probes. In the case of classical power analysis, typically only one physical probe is available. However, it captures the entire circuit's power consumption, including that of all shares of all sensitive variables at once (univariate) or at multiple time instances (multivariate). Therefore, higher-order statistical analyses dealing with such power measurements to some extent reflect the number of probes, for example, see [8]. Such higher-order analyses are, however, strongly affected by the noise level [9]. Consequently, it is believed that employing a sufficiently large number of shares can – in the presence of noise – avert classical SCA attacks.

On the other hand, more advanced photonic SCA attacks from the chip backside [10] enable the adversary to capture side-channel information of several transistors simultaneously, and hence, can provide a large number of probes. However, these attacks can only extract data during transitions. Moreover, due to the typically low Signal-to-Noise Ratio (SNR), the integration of leakages associated to many executions of the cryptographic algorithm with attacker-controlled inputs is necessary. Yet, the existing randomization in masking schemes makes measurement repetition and integration over the same data infeasible. While randomization has been mainly consid-

§These authors contributed to this work when they were with Technische Universität Berlin.

ered as a countermeasure against power/EM SCA attacks in the literature, optical attacks become ineffective as well due to their need for integration.

In response, an intriguing research direction dealing with single-trace SCA attacks has been formed, which mainly target the implementation of public-key algorithms requiring a large number of clock cycles [11]–[13]. Besides, there have been efforts to mount SCA with a minimum possible number of traces by profiling the target in advance, also known as template attacks [7], [14]. Unfortunately, these techniques are relevant only for specific cryptographic schemes and cannot be applied in general to all masked implementations. Furthermore, the profiling phase, in the case of template attacks, might be infeasible in real-world scenarios, where only one sample is available. Besides, it should be noted that profiling still does not guarantee the success of the SCA attack by a single-trace measurement and cannot easily scale with an increase in the number of shares. Driven by the limitations mentioned above, the following question arises: *Does a practical single-trace SCA technique exist that offers an unlimited number of probes while not being limited to specific cryptographic algorithms?*

**Our Contributions.** In this work, we indeed positively answer the above question. We present a novel laser-assisted SCA attack from the chip backside using a known Failure Analysis (FA) technique, called Laser Logic State Imaging (LLSI)[1]. By modulating the voltage supplying the transistors on the chip, the corresponding light reflection (originating from a laser scanning irradiation on these transistors) also becomes modulated. The resulting modulation is highly data dependent because only transistors in the on-state affect the reflection of the laser. We demonstrate how an adversary can deploy LLSI in a particular clock cycle to take a *snapshot* from the entire circuit and recover the state of all transistors, which form the gates and registers. Hence, it enables the adversary to have an unlimited number of *contactless* probes during a time period, which invalidates the central underlying assumption of the probing security model for masking schemes. Moreover, in contrast to other optical attacks or conventional SCA techniques, LLSI does not require any repeated measurements with the same data. Therefore, the existing randomness in masking schemes does not have any protective effect.

To validate our claims, we consider two attack scenarios. First, we assume that the location of the registers is known to the adversary; hence their content can be directly read out using a single snapshot. If this includes key and/or state registers of the underlying cipher, extracting the secret key is straightforward. In this case, the effort for the attacker grows linearly with the number of shares. Second, we demonstrate that even without knowing the location of the registers, the attacker can still recover the secret key by capturing a couple of snapshots at consecutive clock cycles, and making use of a SAT solver. Apart from several simulation-based investigations, to practically show the effectiveness of LLSI we mount snapshot attacks on masked AES designs implemented on a Field Programmable Gate Array (FPGA) manufactured with a 60 nm technology. As a result, we successfully break the security of the targeted masked implementations by extracting their full-length keys.

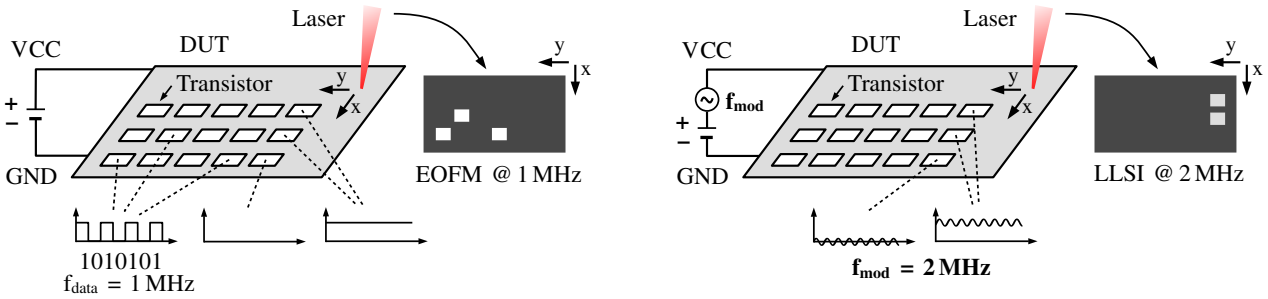
[1]It should be noted that conducting LLSI from the IC backside has been previously reported in the failure analysis community. We claim neither this technique nor our experimental setup as the contribution of this work. Our primary intention is to draw attention to the potential threat of this known but not well-researched technique as an attack tool.

## II. Background

### A. Masking Countermeasures and $t$-Probing Model

While several customized countermeasures (e.g., shielded hardware, current filtering, and dual-rail logic) have been designed to protect specific cryptographic implementations against SCA attacks, masking is known as the most widely studied one with sound theoretical and mathematical foundations. The main idea behind masking schemes is to make use of a couple of parties (order of the masking), and split the intermediate computations dealing with the secrets, i.e., multi-party computation and secret sharing. The input of the circuit (key and plaintext) should be represented in a shared form, and the final result (ciphertext) should be obtained by recombining the output shares while the entire computations are performed only on shares. The primary advantage of masking is that it can be assessed in formal security models. In Boolean masking, as the most common scheme, every random bit $x$ is represented by $(x_0, \ldots, x_d)$ in such a way that $x = x_0 \oplus \ldots \oplus x_d$. Based on formal analyses given in [15], a secret sharing with $d+1$ shares can at most defeat an adversary who is limited to the $d^{\text{th}}$ order SCA. Further, it has been demonstrated that measurements of each share $x_i$ are affected by Gaussian noise, and hence, the number of noisy traces required to recover $x$ grows exponentially with the number of shares [9]. Therefore, as a general knowledge, a higher number of shares would potentially diminish the feasibility of attacks.

On the other hand, the security of masking has been analyzed by the $t$-probing model, which was first introduced in [5]. In this model, it is assumed that the adversary has access to at most $t$ physical probes to observe the computation on wires of the circuit at each time period (e.g., one clock cycle). In such a scenario, at least $t+1$ shares are required to ensure that the adversary cannot learn any sensitive information from $t$ observations. Although we would like to consider an adversary with an unlimited number of probes, this task is generally impractical according to the impossibility of obfuscation [5], [16]. To unify the leakage models, and therefore, simplify the analysis of SCA countermeasures, it has been shown that the two aforementioned leakage models are related by reducing the security in one model to the security of the other one [8], [17]. In other words, a $d^{\text{th}}$-order noisy SCA is equivalent to placing $t = d$ physical probes on the wires of the target circuit. Based on such models and assumptions, several constructions have been introduced [18]–[23], and a couple of security proofs have been given [24]–[27]. Moroever, some (security) verification tools have been developed [28]–[32], and multiple implementations have been reported [33]–[39].




(a) Classical EOFM (b) LLSI

Fig. 1. Comparison of classical EOFM with LLSI. Classical EOFM can be applied to localize transistors switching with a known data-dependent frequency (here: 1 MHz), however, transistors carrying a static signal do not appear in the image. In contrast, for LLSI, the power supply is modulated with a known frequency (here: 2 MHz), thus transistors in the on-state can be localized.

In order to highlight the deployment of masking schemes in real-world products, we would like to mention that protection against side-channel attacks is among the criteria defined by certification bodies in several countries. Masking schemes are among the countermeasures which have been employed in, e.g., banking cards since more than a decade ago by smartcard vendors.

### B. Optical Backside Failure Analysis Techniques

Due to the increasing number of metal layers on the frontside of integrated circuits (ICs), optical FA techniques have been developed to access on-chip signals through the backside [40]. The main techniques are photon emission analysis, laser stimulation, and optical probing, which take advantage of the high infrared transmission of silicon for wavelengths above 1 µm. Although initially developed for FA purposes, these techniques are nowadays also used in the security domain [41]. FA labs are equipped with machines that incorporate all of the previously mentioned techniques in one device, which is typically a laser scanning microscope (LSM) equipped with a camera for photon emission analysis, a detector for measuring the reflected laser light, and laser sources of different wavelengths.[2]

Due to their high spatial resolution, optical FA techniques seem to be promising for conducting single-trace measurements. For instance, the analysis of Photon Emission (PE) with temporal resolution allows to detect the time of switching activities of single transistors. Related techniques are Picosecond Imaging Circuit Analysis (PICA) [42], and the more low-cost approach of Simple Photonic Emission Analysis (SPEA) [43], which has been used to attack unprotected implementations of, e.g., AES [43] and RSA [10]. However, the circuit has to be repeatedly stimulated for these techniques, since the emission probability is very low for a single switching event. This disqualifies time-resolved PE analysis from being a single-trace technique. Optical techniques that in principle can probe static signals are Thermal Laser Stimulation (TLS) [44] and spatial PE analysis of off-state leakage current [45], [46]. However, due to the requirements of low noise on the power line for TLS, and high static current for PE analysis, these techniques are restricted to specific applications and targets. In contrast, optical probing seems to be a more promising technique, and thus, it is discussed in more detail below.

*1) Optical Probing - EOP and EOFM:* For optical probing, a laser beam is focused by a microscope-based setup on the backside of the IC, and the reflected light is analyzed to find data dependencies. Since the refractive index and absorption coefficient within the silicon depend on the electrical properties present in the device [47], the laser light irradiating the IC is modulated and partially reflected. A detector processes the returning light and converts it to an electrical signal. Due to the transparency of the silicon to the wavelengths above 1.1 µm, optical probing can be carried out in a non-invasive manner on some devices [48] (see also Section VII-A5).

The laser can either be parked at a specific location, or scanned over a larger area of the chip. When the laser remains at a particular location, the waveform of the signal of interest can be extracted. This technique is called Electro-Optical Probing (EOP)[3]. To achieve a sufficiently-high SNR, many repetitions of the same waveform must be integrated. On the other hand, when the laser scans an area, the detected signal can be fed into a spectrum analyzer set to a narrowband filter for finding areas on the chip that operate with a specific frequency. This technique is known as Electro-Optical Frequency Mapping (EOFM)[3]. The result of an EOFM measurement is a 2-D image showing a signature at areas switching with the frequency of interest, see Fig. 1a.

Two crucial steps are involved in an attack scenario where the adversary tries to localize and probe a set of registers/memories using optical probing [48], [49]. First, the attacker induces a known frequency into the device (e.g., by supplying the clock or rebooting the chip at a specific frequency) to activate the target registers or memories, see Fig. 1a. Second, the device is operated in a loop, and EOP can be used to read out the values of each individual register. Note that if the sensitive data are processed in parallel, the content of the

[2]For a discussion on cost and availability of such FA machines, see Section VII-A3.

[3]When using a coherent light source, EOP is typically called Laser Voltage Probing (LVP), and EOFM is called Laser Voltage Imaging (LVI).

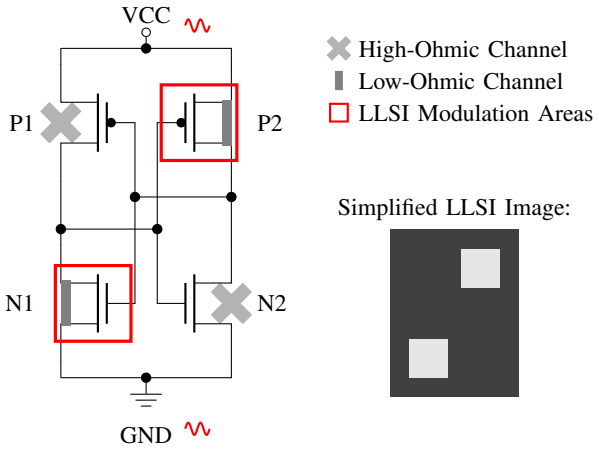



Fig. 2. Schematic of a CMOS memory cell and the expected 2-D LLSI image for the cell. For simplicity we omit the input transistors. Only the transistors in the on-state are expected to give a strong LLSI signal, therefore, the logic state of the memory cell can be deduced. Figure based on [50].

registers can be directly obtained from the EOFM image [49]. As a result, EOFM can be deployed to localize and probe the secret simultaneously on a cryptographic device. However, the downside of this approach is that only dynamic signals which are available for an arbitrary number of repetitions can be extracted. Therefore, classical EOP/EOFM cannot be used to extract static data, i.e., the state of memory elements that are only available once and at a certain point in time.

*2) LLSI:* Laser Logic State Imaging (LLSI) makes the read-out of static signals possible. The technique was introduced as an extension to EOFM to the failure analysis community [50]. For LLSI the supply voltage is modulated with a known frequency. Due to the modulation of the transistor channel's electric field caused by the supply voltage modulation, transistors in the on-state give clear signatures on the LLSI image, while this is not the case for transistors in the off-state, see Fig. 1b. This observation can be used to deduce the logical state of, for instance, a memory cell.

Fig. 2 shows a CMOS memory cell consisting of two cross-coupled inverters. Each inverter consists of one PMOS and one NMOS transistor, connected between VCC and GND. The input to the CMOS inverter directly dictates whether its NMOS is in the on-state and the PMOS transistor in the off-state, or vice-versa. In both cases, only one transistor per inverter is in the on-state. Consequently, when knowing the transistors' states, the value of the inverters' input can be derived. By modulating the power supply of the device, the channel's electric field of all transistors in the on-state modulates with the induced frequency and, as explained above, that can be detected using LLSI. In the example given in Fig. 2, the top right and bottom left transistors are in the on-state, and the expected simplified LLSI image shows a clear signature at those two locations. With the inverted input values, the other two transistors would be in the on-state, resulting in clear signatures on the top left and bottom right of the image. Hence, it can be concluded that all logic states can be extracted using one LLSI measurement.

## III. Threat Model

With our attack, we target hardware implementations of a block cipher protected by some masking countermeasure. While assuming here that the input (plaintext/ciphertext) and the key are shared by Boolean masking, we do not presume any specific masking scheme. Note that the key has to be stored in a masked format on the chip, and it has to be re-masked with fresh masks every time it is used. Otherwise, template attacks [51] or classical optical probing [49] on key or key schedule might be possible. The cipher might be implemented on an FPGA or realized as an ASIC. Following the common serialized or round-based design architecture, or as being enforced by the glitch-resilient masking schemes, the implementation should make use of registers to store the cipher's intermediate values.

We stress that in our technique, we are not making use of any specific construction or feature of any certain masking scheme. We just suppose that the state register (and key register) are masked, which is a general statement and does not deal with any particular technique to realize masking schemes in hardware, like TI [34], DOM [52], GLM [23], CMS [20], UMA [22], etc. Note that these different masking schemes define various techniques to realize non-linear functions (like the ciphers' Sboxes), but they all have in common that the state and key registers are masked. In short, even if the underlying Boolean masking scheme of the target device does not follow any of the known hardware masking schemes, our approach is still a valid attack vector.

Under the above assumptions, we consider a potential attacker, who can take snapshots of the hardware state using LLSI and extract the values stored in the registers. To read out the content of registers at a specific clock cycle, the attacker should either halt the clock or the content should remain in the registers and not get cleared after the algorithm has terminated (see Section VII-A1 for a detailed discussion on clock control). For the purpose of extracting the secret, the attacker could either directly target the (masked) key registers or some registers containing intermediate values of the cipher, from which the secret can be deduced. Which registers the attacker chooses to target, depends on her knowledge about the netlist and layout of the implementation. Regarding this, two scenarios can be discussed (see Fig. 3). **Scenario 1**: If the attacker knows where the key registers are located on the chip, possibly learned by reverse engineering, she could directly target them. Still, due to the underlying masking scheme, she has to target all shares of the key registers. We consider this as the most straightforward scenario and cover it in Section IV-A. **Scenario 2**: If the attacker does not know which registers on the chip contain the secret, some knowledge about the algorithm can help with the key extraction, as explained in Section IV-B. Related to this, we also propose a method to differentiate registers from other combinatorial gates on a chip, if the attacker does not even know the areas on which the registers of the design are placed.

***Real-World Targets.*** To demonstrate how an adversary might

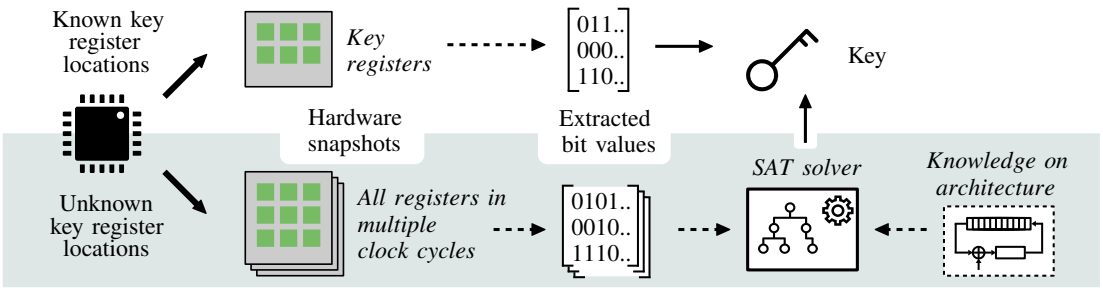



Fig. 3. Two approaches with different assumptions: known key register locations (Section IV-A) and unknown key register locations (Section IV-B).

benefit from such an attack in the real world, we provide some examples for the target devices. One example would be payTV smartcards, which are all programmed with the same key to decrypt the scrambled satellite signal in the receivers using some block cipher. By extracting the encryption key, the adversary can counterfeit the payTV cards and sell them in the black market. Consequently, extracting the secret from one device breaks the security of all devices in the field. Another example would be every microcontroller/microprocessor or FPGA that supports firmware or bitstream encryption, respectively. If the adversary can break this protection mechanism by extracting the key, she can decrypt the firmware/bitstream and clone, reverse-engineer, or tamper with the IP. Note that the adversary is not interested in the hardware itself, and hence, even if the chip gets unusable during the key extraction, the main assets, e.g., key or firmware, are still valuable for the adversary.

## IV. Approach

This section describes methods employed to launch our attack in Scenario 1 and Scenario 2 explained in Section III.

### A. Scenario 1: Known Register Locations

Here we assume that the location of the key registers (i.e., registers used to store key shares) on the chip is known to the adversary. In this case, at some point in time, a given secret key (in a shared form) is loaded in these key registers. Once the attacker knows the corresponding clock cycle, she can take snapshots of the chip using LLSI. The attacker, in principle, can learn the location of these registers by reverse-engineering the layout and netlist of the chip. In the case of an ASIC, this can be done by de-layering the chip and applying some tools to extract the netlist (e.g., ChipJuice [53]). Interestingly enough, the whole procedure is also available as a service, e.g., [54]. If the implementation platform is an FPGA, reverse-engineering the netlist from the bitstream is essential [55]–[57]. When the bitstream is available solely in an encrypted form, the attacker first needs to decrypt it. This is possible, as most cryptographic ASIC cores on mainstream SRAM-based FPGAs, responsible for decrypting the bitstream, are either not protected against SCA or contain other implementation vulnerabilities [44], [48], [58]–[61]. Moreover, it is worth mentioning that an attacker, who is involved in the development and fabrication process of the IC or has enough influence on those entities, might possess parts or entire information necessary to localize the (key) registers on the chip.

***Automatically extracting bit values from snapshots.*** To extract the values from the register snapshots, the attacker first has to discover the data dependency in the LLSI measurements. To this end, if she has control over the data written in the registers, she can take two snapshots of a register cell containing once the value `0` and another time `1`. By subtracting these LLSI images from each other, the attacker can clearly localize the data dependency. Upon knowing how to distinguish between `0` and `1`, she can extract the values in an automated fashion.

For this purpose, we propose an approach based on classical image processing techniques, namely image registration through cross-correlation, cf. [62]. For this, the corners of each register cell (containing one bit of data) should be known with sufficient precision so that the attacker can cut the snapshot of a single register cell from a potentially larger image. For selecting the cell boundaries on an FPGA, domain knowledge can help as the registers are expected to be arranged in regular structures. In the lack of such knowledge, boundaries can be determined by conducting image segmentation methods, e.g., the watershed transformation [63]. Besides, to reduce the impact of the noise, the two-dimensional (2-D) Wiener filter can be applied [64], which can remove the noise by applying a pixel-wise adaptive low-pass Wiener filter to grayscale images.

After these steps, the attacker can choose two snapshots of cells as reference samples (i.e., templates): one containing `0`, and the other one `1` (such two different images can be easily found). Afterward, the attacker applies the cross-correlation over all the snapshots of register cells. Note that, since the positions of the individual register cells are given to the algorithm, the cross-correlation function (instead of the normalized one) can be employed to conduct the image registration. The reference sample that fits best to the targeted register cell determines the bit value contained in the snapshot.

***Remark 1.*** If giving labels to register cells on a training device is not feasible, the adversary still ends up with two groups of cells labeled as (`0`, `1`) or (`1`, `0`). Consequently, she obtains two candidates for the secret key (verified by a single plaintext-ciphertext pair). Therefore, having access to a training device is not an essential fact.

***Remark 2.*** The adversary should not necessarily look for the key registers. Recovering the state of the cipher – either at

initial cipher rounds when the input is known or at final cipher rounds when the output is known – would suffice to reveal the key completely or partially, depending on the underlying cipher. For example, having the state of AES-128 encryption at the first round (after AddRoundKey, SubBytes, or MixColumns) is enough to recover the entire 128-bit key, but for AES-256, two consecutive rounds should be covered.

### B. Scenario 2: Unknown Register Locations

If the location of the registers in the underlying design is not known to the adversary, the attack seems to be nontrivial. Our proposal in such a case is to follow a two-step approach: i) distinguishing the registers from combinatorial cells, and ii) making use of a SAT solver to reveal the location of registers of interest, and finally, extracting the secret.

*1) Identifying register cells:* To localize all register cells of the design on a chip, we propose an approach that takes advantage of the difference between sequential and combinatorial logic. In synchronous designs – as the most common design architecture – every register is driven by the system clock[4]. Consequently, all register cells have a clock input transistor. In contrast, combinatorial logic is data driven, and thus has no clock input. By conducting a traditional EOFM measurement at the clock frequency, the adversary can localize those clock input transistors. The identified areas are the candidates for the location of register cells. Furthermore, in those areas, conducting LLSI experiments with different data might give hints on the existence of a register. In doing so, if the attacker finds at least one register cell, she can attempt to find similarities between its corresponding area and other candidate regions identified by an optical image or the LLSI image. Clearly after this step, the procedure of the automatic extraction of bit values from the snapshots, as explained in Section IV-A, can be followed.

*2) Using SAT solver:* Here, we suppose that the registers are distinguished from the other cells (e.g., through the technique given above), and their values can be recovered at multiple clock cycles, following the above instructions. We also suppose that the design architecture is known to the adversary, i.e., what is processed and stored at every clock cycle. However, it is not known to the adversary which recovered value belongs to which register cell. Having the above assumptions in mind, we propose to use a SAT solver to conduct the attack. It is noteworthy that SAT solvers have also been used to construct algebraic side-channel attacks [65], [66], where a SAT is written based on, e.g., the Hamming weight of the intermediate values recovered by a Template attack. We made use of CryptoMiniSat 5 [67], which, compared to other alternatives, can more easily deal with XOR clauses.

We first focus on a single snapshot at a certain clock cycle leading to binary observations denoted by $\{\omega_0, \ldots, \omega_{n-1} \in \mathbb{F}_2\}$ corresponding to $n$ registers of the design. Some registers belong to the control logic (finite-state machine), which are out of our interest. Therefore, we target $m \leq n$ registers according to the architecture of the underlying design. For example, $m = 256$ for an unprotected implementation of AES (128 bits for the state register and 128 bits for the key register). If we define variables $v_{i\in\{0,\ldots,m-1\}}$ for the value of targeted register cells at the selected clock cycle, we can write

$$v_i = c_0^i \omega_0 + \ldots + c_{n-1}^i \omega_{n-1}, \tag{1}$$

where with $c_j^i$ we denote binary coefficients. Since only one of the observations is associated to the $i$-th register cell, only one of the coefficients $c_{j\in\{0,\ldots,n-1\}}^i$ is 1, and the rest are 0. In other words, $\forall i, \sum_{\forall j} c_j^i = 1$. These are the first formulations that we require to include in the Boolean satisfiability problem (SAT), which are generated individually for each targeted register cell $v_{i\in\{0,\ldots,m-1\}}$, and are independent of the observations $\omega$ and the architecture of the circuit under attack.

We should also add the formulations for (1) for each $v_i$. Those observations $\omega_j$ that are 0 cancel out the corresponding coefficient $c_j$. Therefore, we can write

$$v_i \oplus \left( \sum_{\forall j, \omega_j = 1} c_j \right) = 0. \tag{2}$$

Having more snapshots at different clock cycles, the clauses for (2) should be repeated for $m$ distinct register variables $v_i$ based on the corresponding observations $\omega_j$. However, the coefficients $c_j^i$ stay the same, i.e., they are defined only once for the entire circuit independent of the number of snapshots.

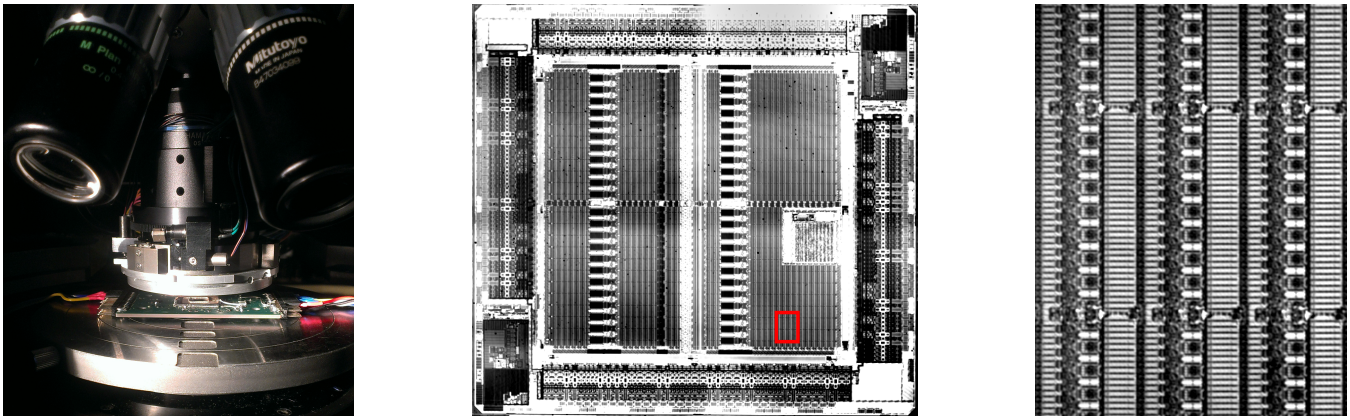
The remaining task is to link the variables $v_i$ (of targeted register cells) at different clock cycles. This is done based on the underlying design architecture of the circuit under attack and the functions it realizes. For example, in a round-based architecture, the state register cells store the output of the cipher round function, and the key register cells the round keys. In a serialized architecture, the content of the registers is shifted (e.g., in a byte-wise fashion), and certain operations (e.g., Sbox) are applied on particular registers at determined clock cycles. We will elaborate an example in Section VI-D.

For a masked implementation with $d+1$ shares, the number of targeted registers at each clock cycle is $m(d+1)$ (e.g., $512 \times 2$ for a first-order masked implementation of AES using the state and key registers with 2 shares). Therefore, the entire formulations given in (2) should be repeated $d+1$ times. In the next step, we define $m$ virtual variables $\nu_i = \bigoplus_{l=1}^{d+1} v_{i,l}$ (for each clock cycle), where $(v_{i,1}, \ldots, v_{i,d+1})$ represent variable $\nu_i$ with $d+1$ shares. The corresponding formulations should be also added to the SAT. The rest is similar to an unmasked implementation, i.e., the (unmasked) variables $\nu_i$ at different clock cycles are linked based on the design architecture of the circuit under attack. We give a detailed explanation how to write the clauses in Appendix B.

## V. Experimental Setup

To evaluate our proposed attack, we need a target device that can run masked AES implementations of different protection orders. In order to conduct LLSI, the power supply of the

[4]In case of clock gating, it should be made sure that the clock is propagated at the target cycle. A detailed discussion is given in Section VII-A2.



(a) DUT under the PHEMOS-1000 FA microscope

(b) Laser scan image of the DUT backside (5x lens)

(c) Zoom-in of the framed area containing the LABs (50x lens)

Fig. 4. Device under test (DUT): Intel Cyclone IV FPGA with part number EP4CE6E22C8N.

device must be modulated, and the backside of the chip must be optically accessible. Since snapshots of a large number of registers in multiple clock cycles have to be acquired, the automation of LLSI measurements would be beneficial.

### A. Device Under Test (DUT)

Our target device was an Intel Cyclone IV FPGA [68] (see Fig. 4). It is manufactured in a 60 nm technology and contains 392 logic array blocks (LABs), each consisting of 16 logic elements (LEs). The LEs mainly consist of a four-input look-up table (LUT) and a programmable register. Furthermore, in every LE, there is logic for loading and clearing data, routing, and clocking. To access the backside of the chip, we opened the package and thinned the bulk silicon to a remaining depth of around 25 µm.We soldered the prepared sample upside down to a custom Printed Circuit Board (PCB) to expose connections to input/output and power supply pins. To keep the power supply modulation for LLSI as unaffected as possible, we did not place capacitors on the PCB.

### B. Electrical and Optical Setup

As the setup (Fig. 5), we used a Hamamatsu PHEMOS-1000 FA microscope with optical probing capabilities. It is equipped with a 1.3 µm high-power incoherent light source (HIL) and 5x/0.14NA, 20x/0.4NA, and 50x/0.71NA objectives. An additional scanner-zoom of 2x, 4x and 8x is available. For EOFM/LLSI, the laser is scanned over the device using galvanometric mirrors, and the reflected light is separated by semi-transparent mirrors and fed into a detector. Its output is then fed into a bandpass filter set to the frequency of interest. The resulting amplitude at every scanning location is mapped to its position and displayed as a grayscale encoded 2-D image.

For LLSI, the supply voltage has to be modulated. Therefore, the internal core voltage ($V_{CCINT}$) of the DUT is supplied with 1.2 V by a Texas Instruments voltage regulator (TPS7A7001), whose feedback path is used to modulate the output voltage. The sine wave signal used for this purpose is generated by a Keithley 3390 function generator, and a

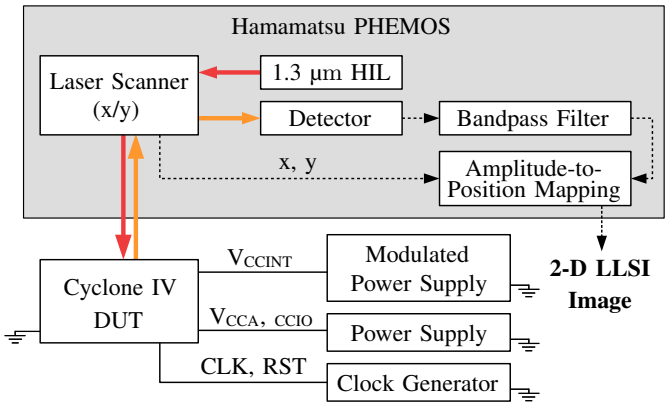



Fig. 5. Electrical and optical setup for conducting LLSI experiments.

Toellner laboratory power supply (TOE8732) provides the DC voltage. An LLSI peak-to-peak modulation amplitude up to 700 $mV_{pp}$ at 90 kHz is possible without disturbing the functionality of the device. The auxiliary voltage pin ($V_{CCA}$) and I/O voltage pin ($V_{CCIO}$) are supplied by the second channel of the TOE8732, which is set to 2.5 V. The clock for the DUT is externally supplied via a Rigol DG4162 function generator, which allows single-stepping and stopping the clock.

### C. Automation of LLSI Acquisition

To create snapshots of the registers in multiple clock cycles in an automated fashion, we use the CadNavi interface provided by the PHEMOS-1000 and the USB interface of the clock generator. The CadNavi interface gives access to functionalities of the PHEMOS, e.g., moving the microscope stage, adjusting the focus, and starting and stopping the measurements. Using the clock generator, the DUT can be reset, and clock cycles can be advanced in single steps. In the LabView programming environment, we implemented a scanning routine as follows. First, the device is stopped at the clock cycle of interest. The stage is then moved to a location of interest, where the focus is adjusted, and drift of the optical setup in *x*- and *y*-direction is corrected. For drift correction, we apply an elastic image registration on the current optical image

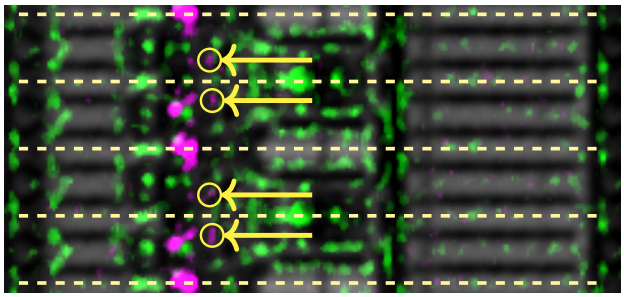


Fig. 6. Experiment for identifying the register cells. EOFM image at the clock frequency (magenta) and LLSI signature (green), overlaid onto an optical image and gathered in parallel while the device was running. LE boundaries indicated by dashed lines and potential clock transistors of registers by arrows.

and an image recorded before the first measurement. Finally, an optical image is taken and the LLSI snapshot is gathered. After the program has gathered snapshots of all locations of interest, the same procedure begins for the next clock cycle.

## VI. Results

### A. Data Dependency of LLSI Measurements

To find the approximate register locations on the FPGA, we first conducted an EOFM measurement at the clock frequency [69], while the device was operating normally. In the result shown in Fig. 6, we could identify several spots switching at the clock frequency. We presume that some of the spots are the actual clock buffers for the registers, and others are part of the clock routing buffers between the LEs. By comparing the chip layout from the FPGA design software with the optical image, we identified the horizontal boundaries between the LEs, as indicated with the dashed lines in Fig. 6. Note that every second LE seems to be flipped horizontally. We then identified clock activity spots that are at the same relative position for every LE, see marked spots in Fig. 6. Because every LE contains only a single-bit register, we expected the registers to reside in the vicinity of these spots.

To find a data dependency in the LLSI measurements and confirm the register location hypothesis, we targeted a single register cell. For this, we set all surrounding registers to `0` and took two LLSI snapshots, one with the targeted bit set to `1`, and one with `0`, see Fig. 7. We set the modulation of $V_{CCINT}$ to $530\,\mathrm{mV_{pp}}$ at $90\,\mathrm{kHz}$ and scanned using the 50x lens with 2x zoom and a pixel dwell time of $10\,\mathrm{ms/px}$. Note that we could see a signature on the LLSI measurements already with a lower modulation amplitude, but we chose these settings to increase the SNR, and hence, decrease the scanning time.

By subtracting the captured LLSI measurements, the areas with differences become visible. It can be observed that there is only one LE with differences, indicated by the yellow window in Fig. 7. The size of this area is about $7\,\mu\mathrm{m} \times 9\,\mu\mathrm{m}$, and located directly to the right of the potential clock buffer. Due to the number of different spots, we assume that the window contains more than just the register. Presumably, the in- and output transistors, as well as other logic, also contribute to the LLSI signature; however, this is irrelevant to our attack as its goal is to extract the bit values stored in the register cells. To demonstrate how arbitrary data from the LLSI images can be read out, we took a snapshot of 24 registers containing randomly chosen data. For an easier manual extraction, we have subtracted a reference snapshot with all registers set to `0`, see Fig. 10 in Appendix A. Consequently, if there is a clear difference for a cell, it contains the value `1`; otherwise `0`.

This leads us to the conclusion that the register inside the LAB and LE can be localized, and also the bit values `0` and `1` can be distinguished using a single LLSI measurement.

### B. Implementation Under Attack

We chose the AES-DOM implementation [52], which is available on GitHub [70]. It is a serialized AES encryption engine that is given the shares of 128-bit plaintext and key, shifted in byte-by-byte during the first 16 clock cycles. The code is written so that it allows the user to arbitrarily adjust the protection order (i.e., the number of shares), meaning that for a $d+1$ sharing scheme, it is expected to provide security against attacks up to $d$-th order by means of $d+1$ shares. It requires a high number of random masks refreshed at every clock cycle, i.e., $(d+1)(9d+10)$ bits for $d+1$ shares. Due to its serialized architecture, only one instance of the (masked) Sbox is instantiated. Since the Sbox has 4 stages of pipeline intermediate registers (essential for almost any hardware masked implementation), a complete SubBytes operation takes 16+4 clock cycles. MixColumns is also performed column-wise, requiring 4 clock cycles. However, due to an interleaved fashion (ShiftRows and MixColumns being applied in parallel to SubBytes), the entire encryption can be terminated after 200 clock cycles [52].

For the implementation on the FPGA, we restricted the AES-DOM core to be placed in a dedicated area on the FPGA using the logic fencing feature of the FPGA design software. Our wrapper module, which is responsible for providing all inputs to the AES core, can thus be excluded from the hardware snapshots. The highest protection, which we could fit on the FPGA (with our co-existing wrapper module), was of $4^{\text{th}}$ order, resulting in 5 shares.

### C. Key Extraction with Known Register Locations

In the first scenario, we target a $d+1 = 3$-share[5] and a $d+1 = 5$-share implementation of AES-DOM (as given in Section VI-B), resulting in $3 \times 128 = 384$ and $5 \times 128 = 640$ bits of key registers, respectively. We placed all key registers to known locations. To minimize the LLSI scanning time, we considered 3 and 5-share implementations occupying in total 24 and 40 LABs (each LAB with 16 register cells), respectively. As the input key shares are provided byte-by-byte to the AES-DOM core, after 16 clock cycles all key shares are stored inside the key registers; hence it is sufficient to extract the key register content only in the $16^{\text{th}}$ clock cycle.

We could achieve a reasonable SNR for the LLSI measurements with the 50x lens, 2x zoom, and a pixel dwell time of

[5]In the AES-DOM code [70], the protection order $d$ is shown by parameter $N = d$.

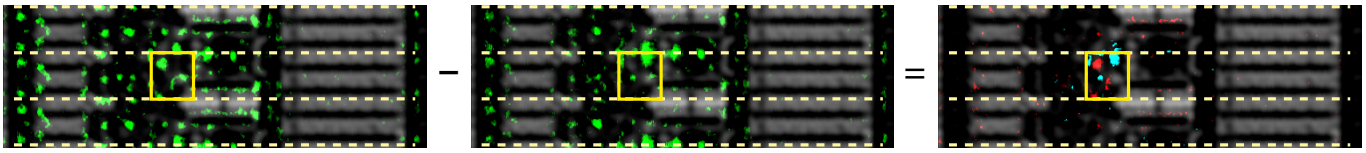



Fig. 7. LLSI measurement of 3 LEs (separated by dashed lines) with only the register of the centered LE (yellow window) set first to `1` and then to `0`, while keeping the other registers set to `0`. When subtracting images from each other, the result indicates the differences. Only the register at the centered LE shows a clear difference, indicating that the bit value has changed.

$3.3\,\mathrm{ms/px}$ with a $V_{CCINT}$ modulation of $640\,mV_{pp}$ at $90\,kHz$. Our scanning routine – including autofocus and drift correction – needs 2.7 minutes to scan one LAB (containing 16 register cells). Note that we scanned only the part of the LABs holding the register cells. Scanning all 3 and 5-share key registers took around 65 and 108 minutes, respectively.

We could easily read out the bit values from the LLSI measurements (even manually possible, for example, see Fig. 8). Subtracting a reference measurement when zero stored in the registers (recorded, e.g., directly after resetting the device) could potentially facilitate manual readout, as also already observed in Section VI-A. However, we used an automated correlation-based extraction scheme which does not require to take snapshots of all registers while they contain zeroes.

***Extracting bit values from snapshots.*** To extract the bit values from the LLSI images as described in Section IV, we applied off-the-shelf image processing algorithms provided in the Matlab software package [71]. First, we registered all the optical images that had been captured along with the snapshots using an elastic transformation. Note that here the process of registration refers to the transformation of the sets of data into one coordinate system, which should not be confused with the technique that we apply to identify the register values. The alignment enables us to cut every register cell according to the boundaries observed in Section VI-A from the snapshot images in an automated fashion. From the resulting cells, we chose two template snapshots of a single register cell for different bit values and subtracted them from each other to remove the signatures not representing the bit value. Then, as explained in Section IV-A, we applied noise reduction through adaptive filtering, and finally converted the templates to a binary mask, see Fig. 8. To extract the bit values, we calculated the 2-D cross-correlation between the snapshot and each template. For determining the value of the register cells, the template for which the maximum correlation is achieved is taken into account. In our experiment, we extracted the value of all registers from the snapshots with 100% accuracy. It is worth mentioning that for our approach, solely a pair of reference cells is required, which can be prepared straightforwardly. The efficiency of our technique should be evident when comparing it with machine learning methods that require a relatively large set of labeled cells.

Due to the underlying 2nd- and 4th-order Boolean masking scheme, by bit-wise XOR'ing all shares, the entire 128 bits of the AES key are trivially revealed (for the first key byte of the 3-share implementation, see Fig 9). The raw LLSI measurements and extraction scripts for all experiments are available online as open-access research data[6].

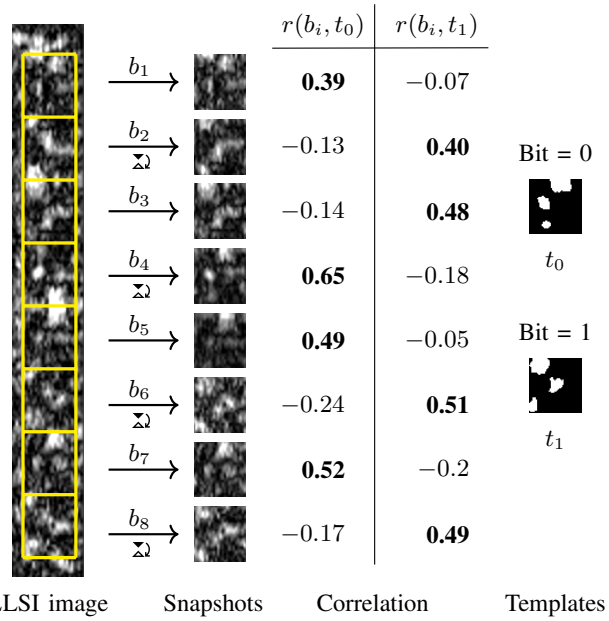



Fig. 8. Correlation-based data extraction mechanism from snapshots of half a LAB (8 bits). Due to the FPGA layout, every second cell has to be flipped horizontally. The correlation coefficient $r(a, b)$ between each cell and the templates for value `0` and `1` is calculated. The extracted bit value is determined by the template matching best.

### D. Key Extraction with Unknown Register Locations

In the second scenario, we selected a $d + 1 = 2$-share implementation of AES-DOM as the target. We adjusted the size of logic blocks so that nearly all 16 registers in each LAB are used, occupying in total 45 LABs. Note that these LABs cover the entire registers of the AES-DOM design, including the shared key registers, shared state registers, the intermediate masked Sbox registers, and those of finite-state machines. However, we do not have any knowledge about the exact location of each register cell and enforce no other placement rule rather than what is explained above. Using the scheme explained in Section IV-B1, we localized the physical area on the chip where the register cells are placed, see Fig. 6.

To conduct the attack, we first investigated the design architecture of the AES-DOM, being serialized with the state and key registers shifted byte-wise, as stated before. Table I represents the content of 32 registers (consisting of 8 bits each) stored in consecutive clock cycles for the first 36 clock cycles, whereas the order of rows in the table is not of our interest.

[6]http://dx.doi.org/10.14279/depositonce-10440

TABLE I
STATE OF THE REGISTERS OF THE AES-DOM DESIGN IN THE FIRST 36 CLOCK CYCLES, EACH ROW REPRESENTS A REGISTER BYTE, K: KEY BYTES, S: SUBBYTES OUTPUT, M: MIXCOLUMNS OUTPUT, K': 2ND ROUND KEY BYTES, S': 2ND-ROUND SUBBYTE OUTPUT.

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | K0 | K1 | K2 | K3 | S0 | M1 | M2 | M3 | S4 | M5 | M6 | M7 | S8 | M9 | M10 | M11 | S12 | M13 | M14 | M15 | K'0 |
| – | – | – | – | – | – | – | – | – | – | – | – | – | – | K0 | K1 | K2 | K3 | S0 | S5 | M2 | M3 | S4 | S9 | M6 | M7 | S8 | S13 | M10 | M11 | S12 | S1 | M14 | M15 | K'0 | K'1 |
| – | – | – | – | – | – | – | – | – | – | – | – | – | K0 | K1 | K2 | K3 | S0 | S1 | S10 | M3 | S4 | S9 | S14 | M7 | S8 | S13 | S2 | M11 | S12 | S1 | S6 | M15 | K'0 | K'1 | K'2 |
| – | – | – | – | – | – | – | – | – | – | – | – | K0 | K1 | K2 | K3 | S0 | S1 | S2 | S15 | S4 | S9 | S14 | S3 | S8 | S13 | S2 | S7 | S12 | S1 | S6 | S11 | K'0 | K'1 | K'2 | K'3 |
| – | – | – | – | – | – | – | – | – | – | – | K0 | K1 | K2 | K3 | S0 | S1 | S2 | S3 | S4 | S9 | S14 | S3 | S8 | S13 | S2 | S7 | S12 | S1 | S6 | S11 | K'0 | K'1 | K'2 | K'3 | S'0 |
| – | – | – | – | – | – | – | – | – | – | K0 | K1 | K2 | K3 | S0 | S1 | S2 | S3 | S4 | S9 | S14 | S3 | S8 | S13 | S2 | S7 | S12 | S1 | S6 | S11 | K'0 | K'1 | K'2 | K'3 | S'0 | S'1 |
| – | – | – | – | – | – | – | – | – | K0 | K1 | K2 | K3 | S0 | S1 | S2 | S3 | S4 | S5 | S14 | S3 | S8 | S13 | S2 | S7 | S12 | S1 | S6 | S11 | K'0 | K'1 | K'2 | K'3 | S'0 | S'1 | S'2 |
| – | – | – | – | – | – | – | – | K0 | K1 | K2 | K3 | S0 | S1 | S2 | S3 | S4 | S5 | S6 | S3 | S8 | S13 | S2 | S7 | S12 | S1 | S6 | S11 | K'0 | K'1 | K'2 | K'3 | S'0 | S'1 | S'2 | S'3 |
| – | – | – | – | – | – | – | K0 | K1 | K2 | K3 | S0 | S1 | S2 | S3 | S4 | S5 | S6 | S7 | S8 | S13 | S2 | S7 | S12 | S1 | S6 | S11 | K'0 | K'1 | K'2 | K'3 | S'0 | S'1 | S'2 | S'3 | S'4 |
| – | – | – | – | – | – | K0 | K1 | K2 | K3 | S0 | S1 | S2 | S3 | S4 | S5 | S6 | S7 | S8 | S13 | S2 | S7 | S12 | S1 | S6 | S11 | K'0 | K'1 | K'2 | K'3 | S'0 | S'1 | S'2 | S'3 | S'4 | S'5 |
| – | – | – | – | K0 | K1 | K2 | K3 | S0 | S1 | S2 | S3 | S4 | S5 | S6 | S7 | S8 | S9 | S10 | S1 | S6 | S11 | K'0 | K'1 | K'2 | K'3 | S'0 | S'1 | S'2 | S'3 | S'4 | S'5 | S'6 | S'7 | S'8 | S'9 |
| – | – | K0 | K1 | K2 | K3 | S0 | S1 | S2 | S3 | S4 | S5 | S6 | S7 | S8 | S9 | S10 | S11 | S12 | S1 | S6 | S11 | K'0 | K'1 | K'2 | K'3 | S'0 | S'1 | S'2 | S'3 | S'4 | S'5 | S'6 | S'7 | S'8 | S'9 |
| – | K0 | K1 | K2 | K3 | S0 | S1 | S2 | S3 | S4 | S5 | S6 | S7 | S8 | S9 | S10 | S11 | S12 | S13 | S6 | S11 | K'0 | K'1 | K'2 | K'3 | S'0 | S'1 | S'2 | S'3 | S'4 | S'5 | S'6 | S'7 | S'8 | S'9 | S'10 |
| K0 | K1 | K2 | K3 | S0 | S1 | S2 | S3 | S4 | S5 | S6 | S7 | S8 | S9 | S10 | S11 | S12 | S13 | S14 | S11 | K'0 | K'1 | K'2 | K'3 | S'0 | S'1 | S'2 | S'3 | S'4 | S'5 | S'6 | S'7 | S'8 | S'9 | S'10 | S'11 |
| – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | K0 | K1 | K2 | K3 | K0 | K1 | K2 | K3 | K4 | K5 | K6 | K7 | K8 | K9 | K10 | K11 | K12 | K13 | K14 | K15 | K'0 |
| – | – | – | – | – | – | – | – | – | – | – | – | – | – | K0 | K1 | K2 | K3 | K0 | K1 | K2 | K3 | K4 | K5 | K6 | K7 | K8 | K9 | K10 | K11 | K12 | K13 | K14 | K15 | K'0 | K'1 |
| – | – | – | – | – | – | – | – | – | – | – | – | – | K0 | K1 | K2 | K3 | K0 | K1 | K2 | K3 | K4 | K5 | K6 | K7 | K8 | K9 | K10 | K11 | K12 | K13 | K14 | K15 | K'0 | K'1 | K'2 |
| – | – | – | – | – | – | – | – | – | – | – | – | K0 | K1 | K2 | K3 | K0 | K1 | K2 | K3 | K4 | K5 | K6 | K7 | K8 | K9 | K10 | K11 | K12 | K13 | K14 | K15 | K'0 | K'1 | K'2 | K'3 |
| – | – | – | – | – | – | – | – | – | – | – | K4 | K5 | K6 | K7 | K4 | K5 | K6 | K7 | K4 | K5 | K6 | K7 | K8 | K9 | K10 | K11 | K12 | K13 | K14 | K15 | K'4 | K'5 | K'6 | K'7 | K'4 |
| – | – | – | – | – | – | – | – | – | – | K4 | K5 | K6 | K7 | K4 | K5 | K6 | K7 | K4 | K5 | K6 | K7 | K8 | K9 | K10 | K11 | K12 | K13 | K14 | K15 | K'4 | K'5 | K'6 | K'7 | K'4 | K'5 |
| – | – | – | – | – | – | – | – | – | K4 | K5 | K6 | K7 | K4 | K5 | K6 | K7 | K4 | K5 | K6 | K7 | K8 | K9 | K10 | K11 | K12 | K13 | K14 | K15 | K'4 | K'5 | K'6 | K'7 | K'4 | K'5 | K'6 |
| – | – | – | – | – | – | – | – | K4 | K5 | K6 | K7 | K4 | K5 | K6 | K7 | K4 | K5 | K6 | K7 | K8 | K9 | K10 | K11 | K12 | K13 | K14 | K15 | K'4 | K'5 | K'6 | K'7 | K'4 | K'5 | K'6 | K'7 |
| – | – | – | – | – | – | – | K4 | K5 | K6 | K7 | K8 | K9 | K10 | K11 | K8 | K9 | K10 | K11 | K8 | K9 | K10 | K11 | K12 | K13 | K14 | K15 | K'4 | K'5 | K'6 | K'7 | K'8 | K'9 | K'10 | K'11 | K'8 |
| – | – | – | – | – | – | K4 | K5 | K6 | K7 | K8 | K9 | K10 | K11 | K8 | K9 | K10 | K11 | K8 | K9 | K10 | K11 | K12 | K13 | K14 | K15 | K'4 | K'5 | K'6 | K'7 | K'8 | K'9 | K'10 | K'11 | K'8 | K'9 |
| – | – | – | – | K4 | K5 | K6 | K7 | K8 | K9 | K10 | K11 | K8 | K9 | K10 | K11 | K8 | K9 | K10 | K11 | K12 | K13 | K14 | K15 | K'4 | K'5 | K'6 | K'7 | K'8 | K'9 | K'10 | K'11 | K'8 | K'9 | K'10 | K'11 |
| – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | K12 | K13 | K14 | K15 | K12 | K13 | K14 | K15 | K'0 | K'1 | K'2 | K'3 | K'4 | K'5 | K'6 | K'7 | K'8 | K'9 | K'10 | K'11 | K'12 |
| – | – | – | – | – | – | – | – | – | – | – | – | – | – | K12 | K13 | K14 | K15 | K12 | K13 | K14 | K15 | K'0 | K'1 | K'2 | K'3 | K'4 | K'5 | K'6 | K'7 | K'8 | K'9 | K'10 | K'11 | K'12 | K'13 |
| – | – | – | – | – | – | – | – | – | – | – | – | – | K12 | K13 | K14 | K15 | K12 | K13 | K14 | K15 | K'0 | K'1 | K'2 | K'3 | K'4 | K'5 | K'6 | K'7 | K'8 | K'9 | K'10 | K'11 | K'12 | K'13 | K'14 |
| – | – | – | – | – | – | – | – | – | – | – | – | K12 | K13 | K14 | K15 | K12 | K13 | K14 | K15 | K'0 | K'1 | K'2 | K'3 | K'4 | K'5 | K'6 | K'7 | K'8 | K'9 | K'10 | K'11 | K'12 | K'13 | K'14 | K'15 |

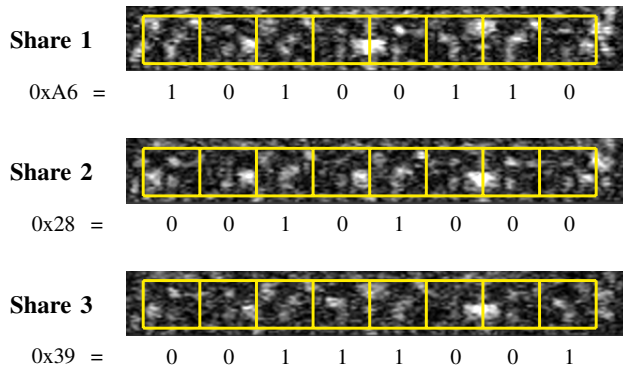



Fig. 9. Extracted values of the first byte of key register shares fo the 3-share implementation. XOR'ing the results $\texttt{0xA6} \oplus \texttt{0x28} \oplus \texttt{0x39} = \texttt{0xB7}$ reveals the first byte of the unshared key beginning with $\texttt{0xB7FCBFF83}\ldots$

For example, the first row shows that the register that stored K0 at clock cycle 16, will hold K1, K2, K3, S0, M1, M2, ... in the next clock cycles. We would like to highlight that it is a symbolic representation and independent of the masking order, e.g., K0 represents all $d+1$ shares of the first byte of the key.

It can be seen that in clock cycle 16, all registers are filled; a part of the state registers with SubBytes' output and the first quarter with 4 bytes of the key. The key register is also fully filled by the given key, which precisely justifies why we targeted this clock cycle for the attack in the first scenario, see Section VI-C. Here, we also started at clock cycle 16 and collected LLSI measurements of the entire 45 LABs in 12 consecutive clock cycles. Each full snapshot in a clock cycle took around 2 hours. Using the fully automated setup developed for this purpose, which applies drift correction mechanisms, we collected all snapshots in 24 hours without any manual interaction. Using the correlation-based extraction technique (see Section VI-C), we extracted the values stored in all registers during the 12 clock cycles.

***Using SAT solver.*** To extract the key, we made use of CryptoMiniSat 5 [67] and followed the technique explained in Section IV-B2. We developed a program in C++ which receives i) the architecture of the underlying design as in Table I, ii) the masking order $d$, iii) the number of covered clock cycles $n$, and iv) the value of registers extracted by snapshots at $n$ clock cycles. The program generates a Boolean satisfiability problem (SAT) to be solved by the SAT solver. For the above case (i.e., $d = 1$ and $n = 720$ register bits in 12 clock cycles), the SAT led to 3 650 048 clauses on 717 728 variables. The SAT solver required 1 hour and 47 minutes to solve the problem and successfully report the revealed key. Note that the SAT solver does not find a unique solution, but all of them lead to the same revealed key. This is due to the underlying masking scheme, i.e., when representing a variable $x$ by 2 shares, the SAT solver makes a distinction between $(x_1, x_2)$ and $(x_2, x_1)$, while both of them lead to the unique unmasked value $x$. This holds for all masked variables in the SAT. If there are $\lambda$ of such mask variables, the SAT solver can find $((d+1)!)^{\lambda}$ correct solutions.

***Extension.*** To examine the efficiency of this approach for different numbers of shares $d+1$ and different numbers of covered clock cycles $\eta$, we have conducted several investigations. We simulated the AES-DOM for $d \in \{0, \ldots, 6\}$ and extracted all register values at the first 36 clock cycles (see Table I). Note that we supplied the implementation with random masks (refreshed at every clock cycle), and did not

TABLE II
THE REQUIRED TIME FOR THE SAT SOLVER TO REPORT A SOLUTION, SUCCESSFULLY RECOVERING THE KEY, FOR DIFFERENT MASKING ORDER $d$ AND VARIOUS NUMBER OF COVERED CLOCK CYCLES BY SNAPSHOTS.

| **Masking** | Number of covered clock cycles starting from 16 | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **order** $d$ | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 |
| 0 | 1.5 h | 7 m | 2 m | 54 s | 46 s | 21 s | 19 s | 24 s | 19 s | 17 s | 19 s | 15 s | 9 s |
| 1 | - | - | - | 1.78 h | 14 m | 10 m | 8 m | 7 m | 8 m | 6 m | 6 m | 5 m | 6 m |
| 2 | - | - | - | - | 1.76 h | 56 m | 47 m | 38 m | 39 m | 30 m | 28 m | 26 m | 21 m |
| 3 | - | - | - | - | - | 5.4 h | 4.5 h | 2.5 h | 2.83 h | 2.15 h | 1.93 h | 1.8 h | 1.2 h |
| 4 | - | - | - | - | - | - | 9.5 h | 8.91 h | 7.71 h | 6.16 h | 5.65 h | 4.75 h | 4.71 h |
| 5 | - | - | - | - | - | - | 1.1 d | 20.61 h | 17.96 h | 16.08 h | 18.5 h | 21.55 h | 19.11 h |
| 6 | - | - | - | - | - | - | - | 1.8 d | 1.9 d | 1.75 d | 1.8 d | 1.49 d | 1.35 d |

consider the name/order of registers when extracting their values. Starting from clock cycles 16, we ran the SAT solver on SATs covering $\eta \in \{9, \ldots, 21\}$ clock cycles, i.e., from clock cycle 16 to clock cycle 24 up to 36. We repeated this experiment with 10 sets of different plaintext/key (and random masks). We found out that the SAT solver usually needs less time to find the solution when more clock cycles are covered by the SAT (expected, as it contains more information). We further recognized that there is a minimum number of required covered clock cycles depending on the number of shares. The averaged results obtained using a machine with a 2.6 GHz CPU and 256 GB RAM are shown in Table II. Note that multithreading is not beneficial here, as CryptoMiniSat 5 looks for different solutions by each thread. Besides, starting before the clock cycle 16 is not helpful since some registers do not contain meaningful data (see Table I).

We have also investigated other design architectures. In short, if the circuit does not allow the collection of enough snapshots per encryption/decryption (e.g., at most 10 in a round-based AES-128 encryption), snapshots for more inputs (plaintexts) can be collected. Although it becomes out of the single-trace feature of our attack, it still allows recovering the secrets by a few snapshots (corresponding to different inputs). As a general overview, a design which requires a higher number of clock cycles for each encryption/decryption would also exhibit more information in the snapshots. We should stress that due to their high area overhead, usually just one instance of some basic blocks (like Sbox) is instantiated in masked implementations, leading to a high number of clock cycles per encryption/decryption. This would potentially decrease the number of required snapshots in our attack.

## VII. DISCUSSION

### A. Attack Feasibility

*1) Clock control:* For taking a snapshot of registers in a region of interest, the registers' contents should not be updated by the clock signal. Therefore, the adversary either needs to halt the clock signal for every snapshot or find a time window, where the registers' contents remain constant for several clock cycles, sufficient for taking a snapshot. Depending on the hardware designer, the state of the (masked) registers might not be cleared after the termination of the encryption/decryption. The same observation has been reported in [72]. In such cases, there is no need to have any control over the clock. If the locations of the registers are known to the adversary, a snapshot from all key registers after the encryption/decryption can be taken to recover the key. However, as multiple snapshots from successive clock cycles are required for the scenario with unknown register locations, this method cannot be applied. Thus, controlling the clock signal is inevitable. To stop the clock, we have identified the two following possible scenarios.

***External clock.*** In the most uncomplicated scenario, the clock is supplied to the chip externally. Hence, the adversary can easily tamper with the clock signal before it enters the chip and keep it low/high at her desired periods to take a snapshot. Naturally, she can repulse the clock again to move one or several clock cycles further with encryption/decryption.

***Internal clock.*** The attack becomes more challenging if the clock is generated internally on the chip. Depending on the target platform (i.e., FPGA or ASIC), the attacker needs to apply more sophisticated techniques to tamper with the clock. If the target is an SRAM-based FPGA, the attacker can use laser fault injection to manipulate the clock source configuration (e.g., based on ring-oscillators) or its routing configuration to stop the clock signalling [73], [74]. To take a snapshot of registers, the adversary first needs to inject a fault into the clock circuitry at her desired cycle and then take a snapshot. However, the challenge would be to reactivate the clock for the next snapshots. Although rebooting the FPGA leads to the correct reconfiguration and reactivation of the clock circuitry, it will not be helpful for the next snapshots due to newly generated random masks. Although successive immediate fault injections are feasible in principle, it might be impractical due to laser setup limitations. Moreover, laser fault injection is not effective in case of an ASIC or a flash-based FPGA since only transient faults can be injected, which is usually not sufficient to halt the internal clock permanently.

A more realistic solution, applicable to all platforms, is circuit editing using Focused Ion Beam (FIB). Using FIB, the attacker can physically cut the metal lines responsible for clock signal delivery or damage the transistors of clock buffers to stop the clock. After disconnecting the internal clock from the

cipher, the attacker can provide her own controllable external clock signal by injecting pulses into clock lines using active nano-probe needles [6]. Even though FIB circuit editing is an invasive technique, it is a practically feasible approach [75]. Thus, we believe that an internal clock cannot stop the attacker from mounting snapshot attacks, although it increases the difficulties.

*2) Clock Gating:* In synchronous circuits, clock gating can be deployed to reduce dynamic power consumption by cutting the clock signal from flip-flops when they are not in use. In this case, since the clock signal is not continuously delivered to a specific group of registers, a question rises about the feasibility of conducting EOFM on an unknown layout to localize the registers. To ensure that all clock gated registers are receiving the clock signal during an EOFM measurement, the dwell time of the laser at each pixel has to be larger than full encryption/decryption time. As a result, we can be confident that the gated registers have been activated temporarily and received the clock signal. Note that while the clock signals for these gated registers might not be periodic anymore during the dwell time of the laser, they still contain the clock frequency component, however with a lower amplitude. Therefore, an EOFM measurement with the clock frequency reveals clock buffers of gated registers with different modulation intensities, i.e., stronger modulation for always active registers and weaker modulation for gated registers. For instance, assume that the cryptographic core is running with a $100\,\mathrm{MHz}$ clock, and the dwell time of the laser is $1\,\mathrm{ms\,px^{-1}}$. In this example, AES DOM requires about 200 clock cycles or $2\,\mu\mathrm{s}$ to complete an encryption. Hence, by keeping the cryptographic operation in a loop during an EOFM measurement, the AES circuit finishes the encryption 500 times while the laser beam is still at the same position. Upon the laser's movement to the next pixel, the same number of operations in the loop occurs until the entire die is scanned with the laser. Thus, by setting the correct relation between the clock frequency and the dwell time for the laser, all registers still can be localized while clock gating is in use. Note that gates involved in the combinatorial logic will not be falsely identified as clock buffers, because they are updated only on either the rising or falling edge of the clock signal while the clock buffers toggle on both edges. Therefore, the combinatorial gates – except those belonging to the clock tree – do not appear on the EOFM image.

*3) Time expenditure and Attack Cost:* One might argue that the time-consuming task of taking the snapshots discourages an adversary from mounting the attack, especially if all registers have to be covered in several clock cycles. For the 2-share implementation, it took $24\,\mathrm{hours}$ to capture snapshots of all registers in 12 clock cycles, see Section VI-D. The time fraction for a single LAB (16 registers) is $2.67\,\mathrm{min}$. Note that autofocus and drift correction significantly contribute to that time. However, the LLSI scan, which creates the actual snapshot of the registers, takes only around $65\,\mathrm{s}$. Therefore, using a more stable optical setup, the acquisition time could potentially be reduced by up to 60%. Furthermore, the registers on the used FPGA are spread over the device with much space in between. On an ASIC implementation, the registers are potentially placed closer together, and thus, a smaller area needs to be imaged by LLSI. Nevertheless, we consider the measurement time of our setup not as a hurdle for an attacker, because the measurements are fully automated and hence can run unsupervised without the presence of an operator. Therefore, we think that – concerning measurement time – our approach is practically feasible in a real scenario.

While laser scanning microscopes are not as cheap as typical oscilloscopes for power/EM analysis, they are common FA equipment. They can be rented for about $\$300$/h including an operator from different FA labs. Therefore, depending on the attack scenario, one can estimate the cost of such attacks based on the number of shares and the size of the die. For instance, the estimated cost to perform LLSI for the known layout of 3-share and 5-share masked AES implementations would be $\$325$ (65 min.) and $\$540$ (108 min.), respectively. Naturally, the cost for an unknown layout would increase, since several snapshots from the entire die have to be taken. However, the cost would increase only linearly by the number of registers on the chip. The estimated cost to mount LLSI attack against an unknown layout with 2-share masked AES implementation would be $\$7200$ (24 hours).

*4) Optical resolution and register size:* In the FA community, optical probing has been shown to be applicable even to the $10\,\mathrm{nm}$ technology node by using a Solid Immersion Lens (SIL), leading to an optical resolution of around $200\,\mathrm{nm}$ [40], [76]. For smaller technology nodes, a higher resolution can be achieved in the visible light regime [77], [78]. For our experiments, we did not use an SIL; hence, the resolution is $\approx 1\,\mu\mathrm{m}$ due to the wavelength of the laser. This resolution might seem low for the DUT manufactured in a $60\,\mathrm{nm}$ technology. However, unlike IC failure analysis, the security evaluation of ICs does not have to rely on targeting a single transistor; therefore, optical resolution requirements can be relaxed to a certain extent. The comparison of technology size and optical resolution often misleads to the assumption that optical probing is not possible for small technology sizes. This has already been shown wrong in [48], where extracting the bitstream from a $28\,\mathrm{nm}$ FPGA was demonstrated.

The size of the area which we used to extract the logic state of one register from, has a dimension of about $7\,\mu\mathrm{m} \times 9\,\mu\mathrm{m}$ for our DUT manufactured in a $60\,\mathrm{nm}$ technology. This area contains multiple transistors. For traditional optical probing techniques, like EOP, the distance between transistors is critical for being able to extract the waveform from exactly one transistor and not a mixture of different signals. However, for LLSI, it is not crucial whether the laser spot covers multiple transistors at a time or not. As long as different signatures for different logic states can be observed in the LLSI measurements, the stored data can be extracted successfully.

*5) Chip preparation and silicon access:* For our attack, we had to depackage the target chip and mount it upside-down on a customized PCB to establish access to the silicon backside. This makes the attack semi-invasive, and one might argue that the effort for chip preparation puts a too high hurdle on the

attacker. However, note that modern chips are increasingly manufactured in flip-chip packages, due to performance, size, cost, and environmental compatibility reasons [79]. Here the silicon backside is directly exposed to the attacker, and no chip preparation is necessary. Therefore, depending on the chip packaging, our attack can also be non-invasive, cf. [48].

### B. Theory vs. Practice

It is tempting to claim that our results rule out the application of the $t$-probing model as presented in [5]. In this regard, we highlight two main points. First, our attack falls only partially within that framework as it requires that the $t$ probes should not move within a time period. Second, but more interestingly, our results demonstrate that some of the assumptions made in [5] do not always hold in reality. More concretely, in [5] and its follow-up studies, the measure of the cost of a probing attack is associated with the value $t$, which is shown to be ineffective for our attack. For this purpose, for a practically feasible, yet more powerful adversary mounting our proposed attack, the spatial coverage and/or the resolution of the probe play a much more vital role. Moreover, it is claimed in [5] that, even in the presence of a fully adaptive adversary moving the probes within a clock cycle, the security is guaranteed as long as the total number of probes in each clock cycle does not exceed $t$. Conversely, we present a powerful new attacker, who is not limited by the number of probes as long as she can manipulate the usual functionality of the clock, which is very likely as explained above and also practically demonstrated by us. To sum this up, the existence of such powerful attackers suggests that the model presented in [5] should be revisited. Of course, the cost for such powerful attackers is higher than that for a classical SCA attack, and there is certainly a trade-off between the cost and the gain depending on the value of the secrets stored in the device.

### C. Potential Countermeasures

Our attack consists of four main steps, namely i) accessing the IC backside, ii) modulating the power supply, iii) scanning with a thermal laser, and iv) localizing the key/state registers. Possible countermeasures can be designed and integrated into the chip to prevent each step.

*1) Package-level countermeasures:* The optical access to the backside of the chip can be prevented after the fabrication and during the packaging of the die. For instance, active backside coatings [80] can make the backside of the chip opaque to the laser scanning microscopy. Since these coatings interact with the transistors, they can detect any tampering attempt. Unfortunately, passive coating layers are not effective since they can be removed mechanically without any consequences.

*2) Device-level countermeasures:* To take a snapshot from the hardware, the core voltage of the device needs to be modulated with a specific frequency during the laser irradiation on the transistors. For preventing the modulation of the supply voltage, internal voltage regulators can be integrated into the circuit to isolate the supply voltage of secure cores from the outside of the device and keep the core voltage constant. Such regulators have already been proposed to defeat power and EM SCAs [81]. As a side note, supplying a voltage regulator by a low voltage (close to its predefined output level) can lead to an unstable output or a transparency between input and output. While the former case already might be sufficient for LLSI, in the latter case, the adversary becomes able to modulate the internal supply voltage at her will. Moreover, distributed temperature sensors can be deployed on the die to detect local temperature variations resulting from the laser beam. However, it should be noted that such temperature sensors have to operate independently from the main system clock; otherwise, they will also be deactivated by halting the clock. Since the wavelength of the thermal lasers is larger than the bandgap of the silicon, no electron-hole pairs are generated upon the incident of photons, and therefore, conventional silicon-based light sensors do not trigger. Temperature sensors can be either realized by timing-sensitive circuits (e.g., ring-oscillators [82]) or specific materials with longer bandgap wavelengths.

*3) Circuit-level countermeasures:* A possible way to defeat our proposed attack is to change the location of registers dynamically. It cannot be done physically, but it seems to be possible logically. Suppose that every single bit is allowed to be stored in a set of $k$ registers. Having $n$ bits, $k \times n$ register cells are required. In addition to this overhead, a mechanism is required to assign one of such $k$ register cells to a single-bit value, dynamically selected at every clock cycle, and independent of other single-bit values. Indeed, we need to randomize the location of registers, independent of any masking scheme integrated to defeat classical SCA attacks. Realizing this might be possible by a form of reconfigurability. To the best of our knowledge, there is no such a scheme known to the hardware security community, and therefore, it is among our planned future works.

## VIII. Conclusion

Masking is the most effective protection for cryptographic implementations against (passive) SCA attacks. The mathematical proof of the probing security models, however, assumes a limited number of probes available to the attacker. This assumption holds for virtually all practically feasible SCA attacks reported so far. We introduced a new optical attack approach that can capture *hardware snapshots* of the IC's entire logic state. It is a single-trace technique offering a number of probes that is only bounded by the number of transistors on the chip. We showed that extracting the keys from 2-, 3- and 5-share AES-128 implementations is practically feasible, even when the exact register locations are not known to the attacker. Due to the practically unlimited number of probes in our attack, implementations with higher protection orders (i.e., with a high number of shares) are vulnerable as well. The complexity of the attack depends on the design architecture, the number of shares, and the knowledge of the adversary about the underlying implementation. The results confirm (again) that cryptography should not rely on complexity of physical attacks. Moreover, assumptions made in theoretical models can be invalidated through more advanced FA techniques, and

hence, one should not underestimate them. We believe that the integration of countermeasures to defeat our attack is not a trivial task. Nevertheless, we gave an overview of the potential countermeasures at different levels of abstraction.

## ACKNOWLEDGMENT

The work described in this paper has been supported in part by the Einstein Foundation in form of an Einstein professorship - EP-2018-480, and in part by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC 2092 CASA - 390781972. The authors would also like to acknowledge Hamamatsu Photonics K.K. Japan and Germany for their help and support on the PHEMOS system. The authors declare no other financial and non-financial competing interests.

## REFERENCES

[1] P. C. Kocher, "Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems," in *CRYPTO '96*, ser. LNCS, vol. 1109. Springer, 1996, pp. 104–113.

[2] P. C. Kocher, J. Jaffe, and B. Jun, "Differential Power Analysis," in *CRYPTO '99*, ser. LNCS, vol. 1666. Springer, 1999, pp. 388–397.

[3] D. Agrawal, B. Archambeault, J. R. Rao, and P. Rohatgi, "The EM Side-Channel(s)," in *CHES 2002*, ser. LNCS, vol. 2523. Springer, 2002, pp. 29–45.

[4] J. Ferrigno and M. Hlavác, "When AES blinks: introducing optical side channel," *IET Information Security*, vol. 2, no. 3, pp. 94–98, 2008.

[5] Y. Ishai, A. Sahai, and D. A. Wagner, "Private Circuits: Securing Hardware against Probing Attacks," in *CRYPTO 2003*, ser. LNCS. Springer, 2003, vol. 2729, pp. 463–481.

[6] Kleindiek Nanotechnik GmbH. (2020) Prober Shuttle (PS8). [Online]. Available: https://www.nanotechnik.com/ps8.html

[7] R. Specht, V. Immler, F. Unterstein, J. Heyszl, and G. Sigl, "Dividing the threshold: Multi-probe localized EM analysis on threshold implementations," in *HOST 2018*. IEEE Computer Society, 2018, pp. 33–40.

[8] A. Duc, S. Dziembowski, and S. Faust, "Unifying Leakage Models: From Probing Attacks to Noisy Leakage," *J. Cryptology*, vol. 32, no. 1, pp. 151–177, 2019.

[9] E. Prouff, M. Rivain, and R. Bevan, "Statistical Analysis of Second Order Differential Power Analysis," *IEEE Trans. Computers*, vol. 58, no. 6, pp. 799–811, 2009.

[10] E. Carmon, J.-P. Seifert, and A. Wool, "Photonic Side Channel Attacks Against RSA," in *2017 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*. IEEE, 2017, pp. 74–78.

[11] R. Primas, P. Pessl, and S. Mangard, "Single-Trace Side-Channel Attacks on Masked Lattice-Based Encryption," in *CHES 2017*, ser. LNCS, vol. 10529. Springer, 2017, pp. 513–533.

[12] K. Järvinen and J. Balasch, "Single-Trace Side-Channel Attacks on Scalar Multiplications with Precomputations," in *CARDIS 2016*, ser. LNCS, vol. 10146. Springer, 2016, pp. 137–155.

[13] M. Alam, H. A. Khan, M. Dey, N. Sinha, R. L. Callan, A. G. Zajic, and M. Prvulovic, "One&done: A single-decryption em-based attack on openssl's constant-time blinded RSA," in *USENIX Security 2018*. USENIX Association, 2018, pp. 585–602.

[14] L. Lerman, R. Poussier, G. Bontempi, O. Markowitch, and F. Standaert, "Template Attacks vs. Machine Learning Revisited (and the Curse of Dimensionality in Side-Channel Analysis)," in *COSADE 2015*, ser. LNCS, vol. 9064. Springer, 2015, pp. 20–33.

[15] S. Chari, C. S. Jutla, J. R. Rao, and P. Rohatgi, "Towards Sound Approaches to Counteract Power-Analysis Attacks," in *CRYPTO '99*, ser. LNCS, vol. 1666. Springer, 1999, pp. 398–412.

[16] B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich, A. Sahai, S. P. Vadhan, and K. Yang, "On the (Im)possibility of Obfuscating Programs," in *CRYPTO 2001*, ser. LNCS, vol. 2139. Springer, 2001, pp. 1–18.

[17] A. Duc, S. Dziembowski, and S. Faust, "Unifying Leakage Models: From Probing Attacks to Noisy Leakage," in *EUROCRYPT 2014*, ser. LNCS, vol. 8441. Springer, 2014, pp. 423–440.

[18] S. Nikova, V. Rijmen, and M. Schläffer, "Secure Hardware Implementation of Nonlinear Functions in the Presence of Glitches," *J. Cryptology*, vol. 24, no. 2, pp. 292–321, 2011.

[19] B. Bilgin, B. Gierlichs, S. Nikova, V. Nikov, and V. Rijmen, "Higher-Order Threshold Implementations," in *ASIACRYPT 2014*, ser. LNCS, vol. 8874. Springer, 2014, pp. 326–343.

[20] O. Reparaz, B. Bilgin, S. Nikova, B. Gierlichs, and I. Verbauwhede, "Consolidating Masking Schemes," in *CRYPTO 2015*, ser. LNCS, vol. 9215. Springer, 2015, pp. 764–783.

[21] J. Balasch, S. Faust, and B. Gierlichs, "Inner Product Masking Revisited," in *EUROCRYPT 2015*, ser. LNCS, vol. 9056. Springer, 2015, pp. 486–510.

[22] H. Groß and S. Mangard, "A unified masking approach," *J. Cryptographic Engineering*, vol. 8, no. 2, pp. 109–124, 2018.

[23] H. Groß, R. Iusupov, and R. Bloem, "Generic Low-Latency Masking in Hardware," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2018, no. 2, pp. 1–21, 2018.

[24] A. Duc, S. Faust, and F. Standaert, "Making Masking Security Proofs Concrete - Or How to Evaluate the Security of Any Leaking Device," in *EUROCRYPT 2015*, ser. LNCS, vol. 9056. Springer, 2015, pp. 401–429.

[25] S. Dziembowski, S. Faust, and M. Skorski, "Noisy Leakage Revisited," in *EUROCRYPT 2015*, ser. LNCS, vol. 9057. Springer, 2015, pp. 159–188.

[26] G. Barthe, F. Dupressoir, S. Faust, B. Grégoire, F. Standaert, and P. Strub, "Parallel Implementations of Masking Schemes and the Bounded Moment Leakage Model," in *EUROCRYPT 2017*, ser. LNCS, vol. 10210, 2017, pp. 535–566.

[27] O. Bronchain, J. M. Hendrickx, C. Massart, A. Olshevsky, and F. Standaert, "Leakage Certification Revisited: Bounding Model Errors in Side-Channel Security Evaluations," in *CRYPTO 2019*, ser. LNCS, vol. 11692. Springer, 2019, pp. 713–737.

[28] G. Barthe, S. Belaïd, F. Dupressoir, P. Fouque, B. Grégoire, and P. Strub, "Verified Proofs of Higher-Order Masking," in *EUROCRYPT 2015*, ser. LNCS, vol. 9056. Springer, 2015, pp. 457–485.

[29] G. Barthe, S. Belaïd, F. Dupressoir, P. Fouque, B. Grégoire, P. Strub, and R. Zucchini, "Strong Non-Interference and Type-Directed Higher-Order Masking," in *CCS 2016*. ACM, 2016, pp. 116–129.

[30] R. Bloem, H. Groß, R. Iusupov, B. Könighofer, S. Mangard, and J. Winter, "Formal Verification of Masked Hardware Implementations in the Presence of Glitches," in *EUROCRYPT 2018*, ser. LNCS, vol. 10821. Springer, 2018, pp. 321–353.

[31] V. Arribas, S. Nikova, and V. Rijmen, "VerMI: Verification Tool for Masked Implementations," in *ICECS 2018*. IEEE, 2018, pp. 381–384.

[32] G. Barthe, S. Belaïd, G. Cassiers, P. Fouque, B. Grégoire, and F. Standaert, "maskVerif: Automated Verification of Higher-Order Masking in Presence of Physical Defaults," in *ESORICS 2019*, ser. LNCS, vol. 11735. Springer, 2019, pp. 300–318.

[33] T. D. Cnudde, O. Reparaz, B. Bilgin, S. Nikova, V. Nikov, and V. Rijmen, "Masking AES with d+1 Shares in Hardware," in *CHES 2016*, ser. LNCS, vol. 9813. Springer, 2016, pp. 194–212.

[34] B. Bilgin, B. Gierlichs, S. Nikova, V. Nikov, and V. Rijmen, "Trade-Offs for Threshold Implementations Illustrated on AES," *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 34, no. 7, pp. 1188–1200, 2015.

[35] F. Wegener and A. Moradi, "A First-Order SCA Resistant AES Without Fresh Randomness," in *COSADE 2018*, ser. LNCS, vol. 10815. Springer, 2018, pp. 245–262.

[36] L. De Meyer, A. Moradi, and F. Wegener, "Spin Me Right Round Rotational Symmetry for FPGA-Specific AES," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2018, no. 3, pp. 596–626, 2018.

[37] A. Poschmann, A. Moradi, K. Khoo, C. Lim, H. Wang, and S. Ling, "Side-Channel Resistant Crypto for Less than 2,300 GE," *J. Cryptology*, vol. 24, no. 2, pp. 322–345, 2011.

[38] A. Moradi, A. Poschmann, S. Ling, C. Paar, and H. Wang, "Pushing the Limits: A Very Compact and a Threshold Implementation of AES," in *EUROCRYPT 2011*, ser. LNCS, vol. 6632. Springer, 2011, pp. 69–88.

[39] H. Groß, S. Mangard, and T. Korak, "An Efficient Side-Channel Protected AES Implementation with Arbitrary Protection Order," in *CT-RSA 2017*, ser. LNCS, vol. 10159. Springer, 2017, pp. 95–112.

[40] C. Boit, S. Tajik, P. Scholz, E. Amini, A. Beyreuther, H. Lohrke, and J.-P. Seifert, "From IC Debug to Hardware Security Risk: The Power of Backside Access and Optical Interaction," in *2016 IEEE*

*23rd International Symposium on the Physical and Failure Analysis of Integrated Circuits (IPFA)*. IEEE, 2016, pp. 365–369.

[41] M. T. Rahman, Q. Shi, S. Tajik, H. Shen, D. L. Woodard, M. Tehranipoor, and N. Asadizanjani, "Physical Inspection Attacks: New Frontier in Hardware Security," in *2018 IEEE 3rd International Verification and Security Workshop (IVSW)*, Jul. 2018, pp. 93–102.

[42] M. K. Mc Manus, J. A. Kash, S. E. Steen, S. Polonsky, J. C. Tsang, D. R. Knebel, and W. Huott, "PICA: Backside failure analysis of CMOS circuits using Picosecond Imaging Circuit Analysis," *Microelectronics Reliability*, vol. 40, no. 8, pp. 1353–1358, Aug. 2000.

[43] A. Schlösser, D. Nedospasov, J. Krämer, S. Orlic, and J.-P. Seifert, "Simple Photonic Emission Analysis of AES," in *Cryptographic Hardware and Embedded Systems – CHES 2012*, ser. Lecture Notes in Computer Science, E. Prouff and P. Schaumont, Eds. Berlin, Heidelberg: Springer, 2012, pp. 41–57.

[44] H. Lohrke, S. Tajik, T. Krachenfels, C. Boit, and J. Seifert, "Key Extraction Using Thermal Laser Stimulation A Case Study on Xilinx Ultrascale FPGAs," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2018, no. 3, pp. 573–595, 2018.

[45] F. Stellari, P. Song, M. Villalobos, and J. Sylvestri, "Revealing SRAM memory content using spontaneous photon emission," in *VTS 2016*. IEEE Computer Society, 2016, pp. 1–6.

[46] J. Couch, N. Whewell, A. Monica, and S. Papadakis, "Direct read of idle block RAM from FPGAs utilizing photon emission microscopy," in *HOST 2018*. IEEE Computer Society, 2018, pp. 41–48.

[47] U. Kindereit, G. Woods, J. Tian, U. Kerst, R. Leihkauf, and C. Boit, "Quantitative Investigation of Laser Beam Modulation in Electrically Active Devices as Used in Laser Voltage Probing," *IEEE Transactions on Device and Materials Reliability*, vol. 7, no. 1, pp. 19–30, 2007.

[48] S. Tajik, H. Lohrke, J. Seifert, and C. Boit, "On the Power of Optical Contactless Probing: Attacking Bitstream Encryption of FPGAs," in *CCS 2017*. ACM, 2017, pp. 1661–1674.

[49] H. Lohrke, S. Tajik, C. Boit, and J. Seifert, "No Place to Hide: Contactless Probing of Secret Data on FPGAs," in *CHES 2016*, ser. LNCS, vol. 9813. Springer, 2016, pp. 147–167.

[50] B. Niu, G. M. E. Khoo, Y.-C. S. Chen, F. Chapman, D. Bockelman, and T. Tong, "Laser Logic State Imaging (LLSI)," in *ISTFA 2014*. ASM International, 2014, p. 65.

[51] M. Wagner, S. Heyse, and C. Guillemet, "Brute-Force Search Strategies for Single-Trace and Few-Traces Template Attacks on the DES Round Keys of a Recent Smart Card," *IACR Cryptology ePrint Archive*, vol. 2017, p. 614, 2017.

[52] H. Groß, S. Mangard, and T. Korak, "Domain-Oriented Masking: Compact Masked Hardware Implementations with Arbitrary Protection Order," in *TIS@CCS 2016*. ACM, 2016, p. 3.

[53] TEXPLAINED. (2018) Hardware security software - CHIPJUICE - Recover any IC's internal architecture. [Online]. Available: https://www.texplained.com/about-us/chipjuice-software/

[54] TechInsights Inc. (2020) Access Our Reverse Engineering — TechInsights. [Online]. Available: https://www.techinsights.com/access-reverse-engineering

[55] K. D. Pham, E. L. Horta, and D. Koch, "BITMAN: A tool and API for FPGA bitstream manipulations," in *DATE 2017*. IEEE, 2017, pp. 894–897.

[56] T. Zhang, J. Wang, S. Guo, and Z. Chen, "A Comprehensive FPGA Reverse Engineering Tool-Chain: From Bitstream to RTL Code," *IEEE Access*, vol. 7, pp. 38 379–38 389, 2019.

[57] M. Ender, P. Swierczynski, S. Wallat, M. Wilhelm, P. M. Knopp, and C. Paar, "Insights into the mind of a trojan designer: the challenge to integrate a trojan into the bitstream," in *ASPDAC 2019*. ACM, 2019, pp. 112–119.

[58] A. Moradi, A. Barenghi, T. Kasper, and C. Paar, "On the vulnerability of fpga bitstream encryption against power analysis attacks: extracting keys from xilinx virtex-ii fpgas," in *Proceedings of the 18th ACM conference on Computer and communications security*, 2011, pp. 111–124.

[59] A. Moradi, D. Oswald, C. Paar, and P. Swierczynski, "Side-channel attacks on the bitstream encryption mechanism of altera stratix ii: facilitating black-box analysis using software reverse-engineering," in *Proceedings of the ACM/SIGDA international symposium on Field programmable gate arrays*, 2013, pp. 91–100.

[60] A. Moradi and T. Schneider, "Improved Side-Channel Analysis Attacks on Xilinx Bitstream Encryption of 5, 6, and 7 Series," in *COSADE 2016*, ser. LNCS. Springer, 2016, vol. 9689, pp. 71–87.

[61] M. Ender, A. Moradi, and C. Paar, "The unpatchable silicon: A full break of the bitstream encryption of xilinx 7-series fpgas," in *29th {USENIX} Security Symposium ({USENIX} Security 20)*, 2020.

[62] J. N. Sarvaiya, S. Patnaik, and S. Bombaywala, "Image Registration by Template Matching Using Normalized Cross-Correlation," in *ACT 2009*. IEEE, 2009, pp. 819–822.

[63] F. Meyer, "Topographic distance and watershed lines," *Signal processing*, vol. 38, no. 1, pp. 113–125, 1994.

[64] J. S. Lim, *Two-Dimensional Signal and Image Processing*. Prentice-Hall, Inc., 1990.

[65] M. Renauld and F.-X. Standaert, "Algebraic Side-Channel Attacks," in *International Conference on Information Security and Cryptology*. Springer, 2009, pp. 393–410.

[66] Y. Oren, M. Renauld, F.-X. Standaert, and A. Wool, "Algebraic Side-channel Attacks Beyond the Hamming Weight Leakage Model," in *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 2012, pp. 140–154.

[67] M. Soos, K. Nohl, and C. Castelluccia, "Extending SAT solvers to cryptographic problems," in *SAT 2009*, ser. LNCS, vol. 5584. Springer, 2009, pp. 244–257.

[68] Altera Corporation. (2016) Cyclone IV Device Handbook. [Online]. Available: https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/hb/cyclone-iv/cyclone4-handbook.pdf

[69] M. T. Rahman, S. Tajik, M. S. Rahman, M. Tehranipoor, and N. Asadizanjani, "The key is left under the mat: On the inappropriate security assumption of logic locking schemes." in *2020 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*. IEEE, 2020.

[70] H. Gross. (2016) DOM Protected Hardware Implementation of AES. [Online]. Available: https://github.com/hgrosz/aes-dom

[71] The MathWorks Inc., "MATLAB–The Language of Technical Computing," http://www.mathworks.com/products/matlab/.

[72] T. Moos, "Static Power SCA of Sub-100 nm CMOS ASICs and the Insecurity of Masking Schemes in Low-Noise Environments," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2019, no. 3, pp. 202–232, 2019.

[73] S. Tajik, H. Lohrke, F. Ganji, J. Seifert, and C. Boit, "Laser Fault Attack on Physically Unclonable Functions," in *FDTC 2015*. IEEE Computer Society, 2015, pp. 85–96.

[74] H. Lohrke, P. Scholz, C. Boit, S. Tajik, and J.-P. Seifert, "Automated Detection of Fault Sensitive Locations for Reconfiguration Attacks on Programmable Logic," in *ISTFA 2016*. ASM International, 2016, p. 6.

[75] C. Helfmeier, D. Nedospasov, C. Tarnovsky, J. S. Krissler, C. Boit, and J. Seifert, "Breaking and entering through the silicon," in *CCS 2013*. ACM, 2013, pp. 733–744.

[76] M. Von Haartman, S. Rahman, S. Ganguly, J. Verma, A. Umair, and T. Deborde, "Optical Fault Isolation and Nanoprobing Techniques for the 10 nm Technology Node and Beyond," in *Proceedings of the 41st International Symposium for Testing and Failure Analysis*, 2015, pp. 47–51.

[77] J. Beutler, J. J. Clement, E. I. Cole, J. Stevens, V. C. Hodges, S. Silverman, and R. Chivas, "Visible Light LVP on Bulk Silicon Devices," Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), Tech. Rep., 2015.

[78] C. Boit, H. Lohrke, P. Scholz, A. Beyreuther, U. Kerst, and Y. Iwaki, "Contactless Visible Light Probing for Nanoscale ICs through 10 $\mu$m Bulk Silicon," in *Proceedings of the 35th Annual NANO Testing Symposium (NANOTS 2015)*, 2015, pp. 215–221.

[79] H. Tong, Y. Lai, and C. Wong, *Advanced Flip Chip Packaging*. Springer US, 2013.

[80] E. Amini, A. Beyreuther, N. Herfurth, A. Steigert, B. Szyszka, and C. Boit, "Assessment of a Chip Backside Protection," *Journal of Hardware and Systems Security*, vol. 2, no. 4, pp. 345–352, 2018.

[81] M. Kar, A. Singh, S. K. Mathew, A. Rajan, V. De, and S. Mukhopadhyay, "Reducing Power Side-Channel Information Leakage of AES Engines Using Fully Integrated Inductive Voltage Regulator," *J. Solid-State Circuits*, vol. 53, no. 8, pp. 2399–2414, 2018.

[82] S. Tajik, J. Fietkau, H. Lohrke, J. Seifert, and C. Boit, "PUFMon: Security monitoring of FPGAs using physically unclonable functions," in *IOLTS 2017*. IEEE, 2017, pp. 186–191.

## APPENDIX A
## ADDITIONAL FIGURE

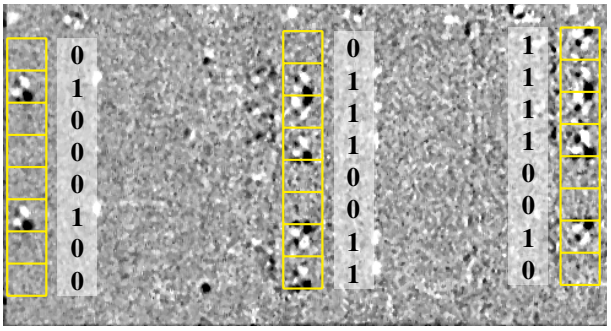



Fig. 10. Difference image of a snapshot covering three times eight registers, once filled with random data, and once with zeroes. A significant difference (black and white spots) for a register corresponds to bit value `1`.

## APPENDIX B
## SAT CLAUSES

We suppose that the registers are distinguished from the other cells (e.g., through the technique given in Section III), and their values can be recovered at multiple clock cycles, following the given instructions. We also suppose that the design architecture is known to the adversary, i.e., what is processed and stored at every clock cycle. However, the relation between the recovered values (through snapshots) and the register cells is unknown. In other words, it is not known to the adversary which recovered value belongs to which register cell.

Having the above assumptions in mind, we use CryptoMiniSat 5 [67] to conduct the attack, which, compared to other similar SAT solvers, can more easily deal with XOR clauses. We should highlight that in such SAT solvers, the problem should be written in Conjunctive Normal Form (CNF), or let say product of sums. Each clause is a sum (logical OR) of a couple of variables (or their invert). The product (logical AND) of all clauses should be `True`, hence every clause should be `True`. CryptoMiniSat allows us to easily define XOR-based clauses as well.

We first focus on a single snapshot at a certain clock cycle leading to binary observations denoted by $\{\omega_0, \ldots, \omega_{n-1} \in \mathbb{F}_2\}$ corresponding to $n$ registers of the design. Some registers belong to the control logic (finite-state machine), which are out of our interest. Therefore, we target $m \leq n$ registers according to the architecture of the underlying design. For example, $m = 256$ for an unprotected implementation of AES (128 bits for the state register and 128 bits for the key register). If we define variables $v_{i\in\{0,\ldots,m-1\}}$ for the value of targeted register cells at the selected clock cycle, we can write

$$v_i = c_0^i\omega_0 + \ldots + c_{n-1}^i\omega_{n-1}, \tag{3}$$

where with $c_j^i$ we denote binary coefficients. Since only one of the observations is associated to the $i$-th register cell, only one of the coefficients $c^i_{j\in\{0,\ldots,n-1\}}$ is 1, and the rest are 0. In other words, $\forall i, \sum_{\forall j} c_j^i = 1$. These are the first formulations that we require to include in the Boolean satisfiability problem (SAT). To this end, we break the addition into bit level by defining intermediate variables $t_{j\in\{2,\ldots,n-1\}}$ for each $i$ individually. Below, we drop the superscript $i$ for both $t$ and $c$ for simplicity. Adding $c_0$ and $c_1$ leads to result $t_2 = c_0 \oplus c_1$ and carry $c_0c_1$. Since the carry must be zero, we can add the following clauses to the SAT.

$$\overline{t_2} \oplus c_0 \oplus c_1 = 1, \qquad \overline{c_0} \vee \overline{c_1} = 1 \tag{4}$$

The same procedure is repeated for adding $c_2$ and the result of former addition $t_2$, i.e., $t_3 = c_2 \oplus t_2$ and $c_2t_2 = 0$. Generally, we can write

$$\forall j \in \{2, \ldots, n-2\}, \quad \overline{t_{j+1}} \oplus c_j \oplus t_j = 1, \qquad \overline{c_j} \vee \overline{t_j} = 1 \tag{5}$$

At the end, we add a clause $t_{n-1} \oplus c_{n-1} = 1$ to the SAT, defining that the final result of the addition should be 1. These clauses (which are independent of the observations $\omega$ and the architecture of the circuit under attack) are generated individually for each targeted register cell $i \in \{0, \ldots, m-1\}$.

We should also add the CNF of (3) for each targeted register cell. Those observations $\omega_j$ that are 0 cancel out the corresponding coefficient $c_j$. Therefore, we can write

$$v_i \oplus \left( \sum_{\forall j, \omega_j = 1} c_j \right) = 0.$$

This translates to

$$\overline{v_i} \vee \left( \bigvee_{\forall j, \omega_j = 1} c_j \right) = 1, \qquad v_i \vee \overline{\left( \bigvee_{\forall j, \omega_j = 1} c_j \right)} = 1. \tag{6}$$

The left equation can be easily added as a clause to the SAT (as it is already in CNF), but the right one should be split into multiple clauses as follows:

$$\forall j, \omega_j = 1, \qquad v_i \vee \overline{c_j} = 1. \tag{7}$$

Having more snapshots at different clock cycles, the clauses in (6) and (7) should be repeated for $m$ distinct register variables $v_i$ based on the corresponding observations $\omega_j$. However, the coefficients $c_j^i$ stay the same, i.e., they are defined only once for the entire circuit independent of the number of snapshots. Accordingly, the clauses in (4) and (5) are also not repeated.

The remaining task is to link the variables $v_i$ (of targeted register cells) at different clock cycles. This is done based on the underlying design architecture of the circuit under attack and the functions it realizes. For example, in a round-based architecture, the state register cells store the output of the cipher round function, and the key register cells the round keys. In a serialized architecture, the content of the registers is shifted (e.g., in a byte-wise fashion), and certain operations (e.g., Sbox) are applied on particular registers at determined clock cycles.

In case of a masked implementation with $d + 1$ shares, the number of targeted registers at each clock cycle becomes $m(d + 1)$ (for example, $512 \times 2$ for a first-order masked

implementation of AES making use of the state and key registers with 2 shares). Therefore, the entire clauses given in (6) to (7) should be repeated $d+1$ times. In the next step, we define $m$ virtual variables $\nu_i = \bigoplus_{l=1}^{d+1} v_{i,l}$ (for each clock cycle), where $(v_{i,1}, \ldots, v_{i,d+1})$ represent variable $\nu_i$ with $d+1$ shares. The corresponding clauses can be written as

$$\forall i \in \{1, \ldots, m\}, \qquad \overline{\nu_i} \oplus v_{i,1} \oplus \ldots \oplus v_{i,d+1} = 1.$$

The rest is similar to an unmasked implementation, i.e., the (unmasked) variables $\nu_i$ at different clock cycles are linked based on the design architecture of the circuit under attack.

# 4

# AUTOMATIC EXTRACTION OF SECRETS

## 4.1 REVERSE ENGINEERING AND IMAGE RECOGNITION

*Reverse engineering for data extraction*

In the previous chapter, we have shown that extracting the content from on-chip memories using logic state imaging techniques, such as LLSI, is possible. However, the work included manual reverse engineering to extract the content from the registers of the FPGA. The memory cells' locations, their boundaries, and how the signatures for the bit values `0` and `1` can be distinguished had to be found. Using this knowledge, we applied a tailored image processing algorithm to extract the content from the individual cells' images. Previous publications have also shown that manual reverse engineering is necessary. For example, the authors of [7] had to manually find the memory layout for extracting a full key. While a relatively small number of key bits (e.g., 256 in [7]) in an evenly distributed memory is still manageable, obfuscated, shuffled, or non-regular memory structures and longer sequences of data might not be easily extractable manually.

*Machine learning in hardware security*

At the same time, machine-learning techniques are on the rise. For instance, image recognition has advanced to observe the environment in real-time for controlling autonomous cars. Especially artificial neural networks have gained popularity due to their excellent object detection and classification capabilities without having to tune the algorithm manually. Therefore, these techniques have also gained popularity in the security field to break or protect devices and implementations. As one prominent application, deep learning can aid in extracting secrets from classical side-channel data like power or EM traces [42, 43]. As a matter of fact, the number of papers on deep learning-based SCA has increased almost exponentially over the past years [43]. Apart from leveraging deep learning algorithms to attack devices, they can be used to design implementations that are more robust to SCA, for instance, in the form of leakage compensation circuits [44]. Furthermore, deep learning can even be used to detect fault injection attacks during the runtime of the chip [45].

*Research question and potential threats*

Given the above observations, we asked ourselves to what extent the reverse engineering procedure needed to extract secrets from ICs using laser-based techniques can be automated. If reverse engineering and data extraction can be fully automated, attacks would be possible with much less manual labor than previously expected. We assumed two phases for such an approach: profiling and extraction. In the profiling phase, the attacker has access to a training device on which she can control the programmed secret data. She can capture as many

logic state images from the device as she needs. Afterward, she would feed the obtained images into a machine learning tool that learns how to extract the secret contained in the images. Once the attacker has finished the profiling, she can directly move to a device containing an unknown secret, capture one logic state image, and extract the contained data quickly – potentially within minutes.

This approach drastically reduces the manual labor and expertise required to extract the secret from a single device. However, this is not the only benefit an adversary would gain. The outlined procedure can also be of interest when she wants to extract secrets from multiple devices of the same type. The profiling would only need to be done once, and the obtained model can be re-used to speed up the extraction from multiple device instances. Furthermore, this method would allow adversaries with sufficient resources, e.g., nation-states, to profile different devices with a low amount of human labor involved. With such a set of models at hand, extracting secrets from different device types on demand would be possible.

In the following work, we tested this approach on three different devices: two FPGAs and a microcontroller. We have applied two different optical logic state imaging techniques, TLS and LLSI, and trained convolutional neural networks (CNNs) on the secret extraction from the memory structures.

## 4.2 PUBLICATION

The reprinted publication follows subsequently. It was originally presented at the *30th USENIX Security Symposium (USENIX Security 21)* [46]. The logic state images acquired in connection with this publication are available in [47].

# Automatic Extraction of Secrets from the Transistor Jungle using Laser-Assisted Side-Channel Attacks

Thilo Krachenfels*, Tuba Kiyan*, Shahin Tajik† and Jean-Pierre Seifert*‡
* *Technische Universität Berlin, Chair of Security in Telecommunications*
† *Worcester Polytechnic Institute, Department of Electrical and Computer Engineering*
‡ *Fraunhofer SIT*

## Abstract

The security of modern electronic devices relies on secret keys stored on secure hardware modules as the root-of-trust (RoT). Extracting those keys would break the security of the entire system. As shown before, sophisticated side-channel analysis (SCA) attacks, using chip failure analysis (FA) techniques, can extract data from on-chip memory cells. However, since the chip's layout is unknown to the adversary in practice, secret key localization and reverse engineering are onerous tasks. Consequently, hardware vendors commonly believe that the ever-growing physical complexity of the integrated circuit (IC) designs can be a natural barrier against potential adversaries. In this work, we present a novel approach that can extract the secret key without any knowledge of the IC's layout, and independent from the employed memory technology as key storage. We automate the – traditionally very labor-intensive – reverse engineering and data extraction process. To that end, we demonstrate that black-box measurements captured using laser-assisted SCA techniques from a training device with known key can be used to profile the device for a later key prediction on other victim devices with unknown keys. To showcase the potential of our approach, we target keys on three different hardware platforms, which are utilized as RoT in different products.

## 1 Introduction

For security applications, people rely on hardened hardware modules, like Trusted Platform Modules (TPMs), as the root-of-trust (RoT) for storing secret keys. Those keys ensure the functioning of complex and delicate systems like routers, servers, sensor systems, and cars by establishing secure communication channels, safeguarding trusted code execution, and protecting the intellectual property embodied in the device. Extracting secret keys managed by a RoT hardware would break the entire system's security. Possible motivations for attackers are the extraction of secret information, tampering with the design, or cloning the device.

Modern integrated circuits (ICs) and system-on-chips (SoCs) consist of billions of transistors, which makes the reverse engineering of the design and layout very challenging. Moreover, data extraction from various key storage technologies requires different measurement tools and expertise, making the attack costly and unscalable. This physical complexity might lead to a belief by vendors that the localization and extraction of assets/secrets on their products is a laborious task. In addition to that, the usage of the keys in diverse applications, such as firmware/bitstream decryption, asymmetric cryptographic operations, or logic deobfuscation, makes the generalization of an attack against RoTs infeasible.

There are companies like Techinsights [1] and Texplained [2] that invest lots of expertise and effort into fully reverse engineering ICs with destructive techniques and using sophisticated failure analysis (FA) tools, such as scanning electron microscopes (SEMs) and focused ion beams (FIBs) [3]. They can extract the IC's netlist, analyze its functioning, and therefore, find the location where the key is stored. While effective, this approach is very time consuming and expensive. On the other hand, researchers have shown that attacks on some specific devices only require partial reverse engineering. Applying SEM [4, 5], FIB [6], microprobing [7], and laser-assisted side-channel analysis (SCA) techniques using laser scanning microscopes (LSMs) [8, 9, 10, 11] are examples of such academic work. Nevertheless, these attacks have only been carried out in an experimental environment, where many details of the design were available beforehand or had to be gathered manually.

Considering the high amount of manual reverse engineering work, one might ask if machine learning techniques could be applied in the context of hardware security to reduce the required knowledge for key extraction. Indeed, the benefit of applying deep learning techniques on classical SCA attacks, like power and electromagnetic (EM) analysis, have already been discovered and studied extensively [12, 13, 14, 15]. At the same time, convolutional neural networks (CNNs) have become the default choice for image classification tasks, as they remove the need for manually tailoring the algorithm to

Reprinted from the original article published in the Proceedings of the 30th USENIX Security Symposium, 2021

its specific application. Consequently, CNNs could also be one suitable method for extracting a key from images captured by FA techniques from a complex chip with unknown design and layout. In other words, if an attacker combines image recognition techniques with sophisticated FA tools that are capable of capturing the logic state from inside the IC, a new threat dimension arises. Such an approach can antiquate the expensive reverse engineering portion of hardware attacks. On the positive side, such a tool, if automated, can also be used for security assessment of products.

**Our contribution.**

In this work, we develop an attack approach drawing the connection between image recognition techniques, profiling SCA, and sophisticated FA tools to extract the secret key from memory cells of an IC without requiring any knowledge about the chip's layout and its functioning.

To validate our claims, we conduct SCA using two different and well-known laser-assisted SCA methods, namely thermal laser stimulation (TLS) [10] and laser logic state imaging (LLSI) [11]. We apply these SCA techniques on three different hardware targets with various process technology sizes: the dedicated key memory of an 20 nm Field Programmable Gate Array (FPGA), the SRAM of a 180 nm microcontroller, and the registers of an 60 nm FPGA. All these platforms can be potentially part of an RoT implementation. To showcase the strength of our approach, we exemplarily deploy CNNs to create models out of obtained measurements from these devices. The results demonstrate that our trained models can extract an unknown secret key from the victim devices with high accuracy, even in the presence of largely irrelevant information and activities on the chip. Moreover, it is not required to know the location of targeted memory cells and how to interpret the bit values from the measurements. Note that our approach is not limited to optical SCA attacks, and can also be combined with SEM, FIB, or any other FA microscopy tools, which capture the activity of transistors.

While in this work we have applied deep learning due to its straight-forward nature for highlighting the threat of our approach, deploying other image recognition techniques is also conceivable. In this regard, we are open-sourcing the side-channel data to enable other researchers to improve data extraction using various techniques. Consequently, we would like to stress that the emphasis of this work is on showing that laser-based SCA can eliminate the reverse engineering step for extracting secret information, and not on applying deep learning techniques as profiling SCA tool.

## 2 Threat Model

### 2.1 Target

In hardware RoT applications, we can distinguish between two different kinds of keys. At least one *root key* must be stored in plaintext in a non-volatile memory (NVM). Other keys might be stored internally/externally in an encrypted form, decryptable by the root key. In the following, we will refer to them as *application keys*. In addition to its usage as key-decryption-key, the root key might also be deployed directly, e.g., for firmware or bitstream decryption. Since the root key is typically stored in a (tamper- and read-proof) NVM, such as flash memory, EEPROM, or ferroelectric RAM, the direct extraction of its content is not a straightforward task [5]. However, even if the NVM is considered secure, for being used, the contained key will be loaded into CMOS memory cells at some point in time, see Fig. 1. The same holds true for application keys after they have been decrypted.

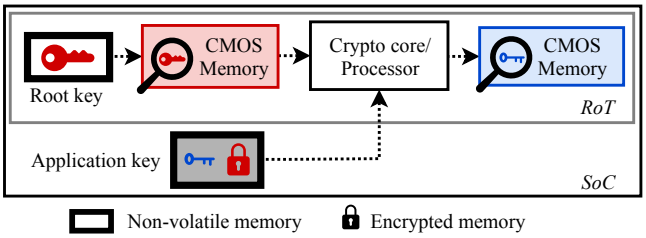
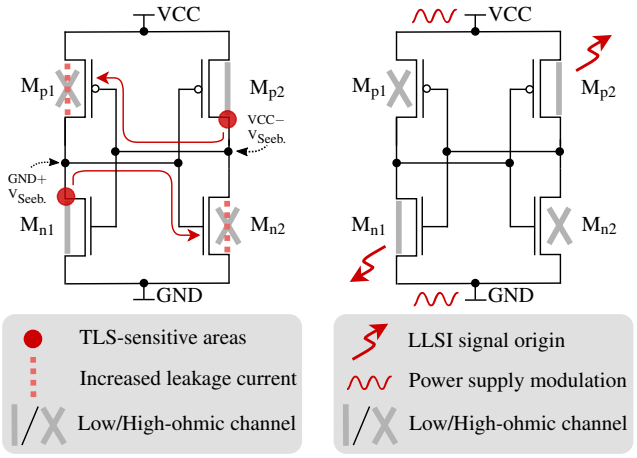



Figure 1: Extraction of the root key after it was loaded from a tamper- and read-proof non-volatile memory, or of an application key after it has been decrypted using the root key.

Previous work has shown that sophisticated non- and semi-invasive FA tools are capable of extracting logic states from CMOS logic gates [16] and memory cells [8, 10, 11]. These techniques typically produce an image (i.e., activity map or response image) which contains information about the logic state of the area of interest. Yet, extracting the actual memory content from these images can be a challenging task, even if the chip's layout is known, or at least understood to a certain degree. Although tools like SAT solvers [11] and image recognition techniques can aid the localization of the key, much prior knowledge of the memory cell's design, its geometry, and its exact location is required. Therefore, a potential attacker might be highly motivated to reduce the effort for extracting keys from the images.

### 2.2 Attacker's Motivation

We assume an adversary who has access to FA tools and has a strong motivation to avoid expensive reverse engineering of the whole IC for just extracting a single key out of it. One might ask why an adversary would invest that much effort into extracting the key from a single device. The primary motivation in many scenarios is that the same key is used for all devices, for instance, when firmware, bitstream, or logic encryption is used to protect the proprietary design of a system. The key is therefore programmed by the vendor before the product is shipped to the customer. Consequently, extracting the key from one device would break the security of all devices from the same family. Even if the key differs between




(a) TLS (b) LLSI

Figure 2: Schematic of a CMOS memory cell and how the two measurement techniques can extract the cell's logic state. Transistors for read and write access are omitted. Figures based on [19, 20].

devices, it should be kept in mind that all chips from a device family have the same layout. Therefore, the adversary can learn how to extract the key from a training device and use her knowledge to extract the key also from other devices of the same family.

# 3 Background

## 3.1 Optical Side-Channel Analysis Attacks

For being able to debug the active silicon of integrated circuits (ICs) in the presence of the many metal layers on the chip frontside, techniques have been developed to access on-chip signals through the IC backside [17]. The corresponding optical side-channel analysis (SCA) techniques take advantage of the high infrared transmission in silicon for wavelengths above 1 µm, basically allowing to "see through" the bulk silicon at the IC backside. Due to their availability in FA labs around the globe, related techniques like photon emission analysis, laser stimulation, and optical probing have been adopted by the hardware security field [18, 10, 9]. A typical setup consists of a laser scanning microscope (LSM) with laser sources of different wavelengths, a detector for measuring the reflected laser light, and optionally a camera for photon emission analysis.

The two relevant techniques for this work, including reported attacks in the literature, will be discussed below.

### 3.1.1 Thermal Laser Stimulation

Thermal laser stimulation (TLS) is an SCA technique that induces electrical perturbations on a target device by creating local temperature gradients when stimulating an area of interest with a laser beam. The laser beam's wavelength is above 1.1 µm, which does not have enough energy to generate electron-hole pairs, but thermal gradients. A temperature variation on a thermocouple can lead to a voltage generation, which is known as the Seebeck effect [21]. The Seebeck voltage can be leveraged to extract the logical states from CMOS memory cells [19].

A CMOS memory cell consists of two cross-coupled inverters, with one transistor per inverter being low-ohmic (conducting) and one being high-ohmic (nonconducting), see Fig. 2a. Hence, while storing a value, i.e., in the stable state, only a negligible current is flowing between VCC and GND. However, if a laser beam stimulates the drain-bulk junction of a transistor with low-ohmic channel, it generates a Seebeck voltage ($V_{Seeb.}$). This voltage is forwarded along the circuit to the gate of a transistor in the high-ohmic state. This transistor is slightly switched on and – via exponential sub-threshold operation–, the current drawn from the memory cell's power supply increases. If an area of interest on the device is scanned pixel-wise by a laser beam and the small power consumption variations are recorded along with the laser beam's location, the TLS response map of the scanned area can be obtained. The areas of the two sensitive transistors will show up brighter in the TLS response map, due to the slight increase in power consumption. For the opposite bit state, the other two transistors will appear on the response map, making the two different bit states of the memory cell distinguishable from each other.

TLS is a well-understood technique that has been used to read out SRAM memory on microcontrollers [19, 22] and extract the cryptographic key from the battery-backed RAM on an FPGA [10]. One scan over the area of interest can reveal the entire memory content, and therefore, TLS can be considered a single-trace SCA technique. Naturally, the memory content should stay constant during the scan. Recently it has been shown that TLS can be mounted with cheaper setups than previously expected – for around $100k [23].

### 3.1.2 Laser Logic State Imaging

Optical probing is an FA tool used for acquiring electrical information from inside the IC [24, 8, 9]. Electro-optical frequency mapping (EOFM) is an optical probing technique that allows creating a 2-D activity map of circuits, showing nodes that are switching at a particular frequency [25]. While light with wavelengths above 1 µm scans the IC backside pixel by pixel, it passes through the silicon substrate. The light is partially absorbed and partially reflected by structures such as metal layers and transistors, whereas the electrical field present at transistors influences the light's amplitude and phase. A portion of the reflected light leaves the IC through the backside where it is converted into a voltage and fed into a narrow-band frequency filter set to the frequency of interest. The resulting signal's amplitude and the position information

form the 2-D activity map on which areas modulating at the frequency of interest appear as bright spots.

For EOFM measurements, it is necessary to know the internal switching frequency of the circuit of interest to track the signals. This frequency can be hard to predict, and even worse, there is not necessarily any switching activity for memory cells if no read/write operation is carried out. This problem can be tackled by inducing a frequency, for example, by modulating the core voltage that supplies the circuit under test. The corresponding technique is called laser logic state imaging (LLSI) and has been introduced as an extension to EOFM [20]. LLSI makes the extraction of static logic states possible, e.g., from a CMOS memory cell, as illustrated in Fig. 2b. The low-ohmic transistors' electric fields oscillate with the power supply's modulation frequency, and hence, produce an EOFM signal. In contrast, off-state transistors do not produce a strong EOFM signal. Consequently, the logic state of the SRAM cell can be deduced. LLSI has been used to read out SRAM on a microcontroller [22] and the registers on an FPGA [11].

Note that LLSI can be used to extract not only the state of SRAM cells or registers, but also any cluster of transistors, such as buffers or logic gates. As long as the bit state of the logical element affects the involved transistors, the bit value can be extracted. Next to TLS, also LLSI can be considered a single-trace SCA technique, as one scan over the region of interest is sufficient to capture its entire logic state. Similarly, to perform LLSI, the memory content has to remain constant during the scan. One way to achieve this requirement is to halt the clock signal to prevent any update in the values of the memory [11]. However, in some applications, e.g., logic locking, the secret key has to be provided constantly to the locked circuit in order to keep it unlocked during runtime, and therefore, no clock control is needed. Moreover, it has been observed that some cryptographic accelerators do not necessarily clear key registers after encryption/decryption [26], and hence, the key remains in the registers as long as the device is powered on.

## 3.2 Deep Learning for Image Classification

Due to their high flexibility, convolutional neural networks (CNNs) [27, 28, 29] are a popular choice for many computer vision applications such as image recognition [30, 31]. Image recognition typically consists of two tasks: object classification (also called image-level annotation) and object detection (object-level annotation). While for classification only the presence of an object from a given set of classes is assessed – and not its position –, object detection is typically a more challenging task. In this work, we are only interested in the existence of a logic `0` or `1` in an image, and therefore, we will only cover object classification in the following.

CNNs are a subclass of deep neural networks and complement the fully-connected (FC) networks (also known as multilayer perceptrons) with trainable feature extractors, the so-called *convolutional layers*. A convolutional layer finds features in the image (e.g., corners, edges, etc.) using trainable filters that cover a certain receptive field. The resulting *feature maps* can be fed into subsequent convolutional layers to detect larger features. Intermediate subsampling steps – *pooling layers* – reduce the resolution of the feature maps to decrease the sensitivity to shifts and other distortions. Finally, after some repetitions of convolutional and pooling layers, the output is *flattened* and fed into the FC network to classify the images.

In the literature, different architectural designs for CNNs have been reported, e.g., LeNet-5 [28], AlexNet [32], and VGG [33]. The authors of the VGG architecture presented a generic design consisting of the repetitive application of filters with a very small receptive field (3×3 pixels), followed by a max-pooling over a 2×2 pixel window. The stack of convolutional layers is followed by FC layers with one neuron for each class in the output layer [33]. The structure of multiple small convolutional layers followed by a max-pooling layer is often referred to as *VGG-block* and has become a popular building block and starting point when designing a new model from scratch, like it will be required for the optical key extraction. Different concepts have been developed to reduce over-specialization on the training data (so-called overfitting) of CNNs, especially when only a small training dataset is available. For instance, a *dropout layer* can remove random nodes from the FC layers during training, which leads to the extraction of more robust features [34]. Furthermore, *data augmentation* can increase the number of training samples artificially, and therefore, reduce overfitting as well [35].

## 3.3 Related Work

This work builds on an approach that is known as profiled side-channel analysis [36], where a device under the adversary's control is used to create a leakage model, which is later used to extract the secret from a similar device [37]. In the literature, profiled SCA is typically applied to a cryptographic core by observing its operation, for instance, through power and EM side-channels. In the *profiling phase*, the behavior of the DUT is observed and incorporated in a leakage model using either statistical methods (a.k.a template attacks [38]) or machine learning techniques [39], such as support vector machines [40] and neural networks [12, 13, 14, 15]. In the *attack phase*, the extracted model is used to extract the unknown secret from the target device. Traditional SCA has limited applicability in some cases, e.g., when the key is not involved in active computations, or when countermeasures prohibit the capturing of a sufficient number of traces.

Next to side-channel analysis, machine learning is also used in many other applications in the field of hardware security [39], for instance, for hardware trojan detection [41] and reverse engineering [3, 42].

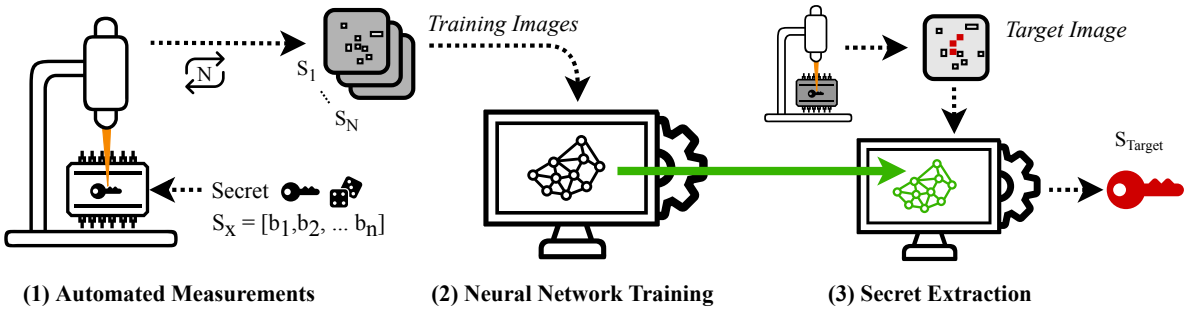



Figure 3: Schematic of the proposed three-step attack approach.

## 4 Attack Approach

Our attack approach has already been sketched in [43] and assumes that the adversary has access to a training device, for which she can control the contained secret at her will. However, she does not have any knowledge about the design of the chip and the location of the key storage. In this scenario, the approach for the attacker consists of three steps, see Fig. 3. In the first step, randomly chosen keys are programmed into the training device, and SCA images are captured from the IC backside for each key. Subsequently, neural networks are trained with the obtained images. These two steps can be specified as profiling phase. In the final step, the attack phase, the secret on the target device is revealed by one or a few measurements and the previously trained networks. Note that in this work, we chose to apply deep learning techniques for image recognition due to their ad-hoc adaptability to many problems with minimal tuning effort. For the secret extraction from the images, potentially also other machine learning or statistical methods can be applied. In the following, we discuss the three steps of our approach in more detail.

### 4.1 Automated Measurements

For gathering a training dataset, the adversary captures response images using TLS or LLSI from the training device containing different randomly chosen keys. Since capturing many high-resolution images from larger areas of the chip can be very time consuming, the attacker would first try to find candidate areas for the on-chip memory. Due to the repetitive and regular structure of memory arrays, such candidate areas often can be discovered by analyzing an optical image of the chip. If this is not the case, two response images (containing two different secrets) can be captured from the entire chip area. When subtracting the two images, the attacker can consider all areas showing a difference as candidate areas which should be covered by the automated measurements. Consequently, one sample in the training database consists of one or more response images and the programmed secret. After capturing some samples, the attacker can continue with step 2, that is, training CNNs with the database.

### 4.2 Neural Network Training

Before training CNNs with the response images, possible drift caused by mechanical instabilities of the setup should be corrected. For this, classical image registration techniques can be used, e.g., by calculating the offset between an optical image captured along with the response image and one fixed optical image. Subsequently, the response image can be transformed according to the calculated shift.

Furthermore, the programmed secret is split into its individual binary bits, which are assigned as multiple labels to each image – one label per bit. Once these preparatory steps are done, a CNN can be designed to learn the secret bits from the response images. More specifically, for each bit of the secret, the images are classified to contain either the binary bit value `0` or `1`. Note that each bit of the secret is handled independently from the other bits. To find out if the images depend on the secret at all, different network architectures should be investigated while trying to learn just a single bit of the secret. Following common practice, we propose to start with a simple model, containing only a few convolutional layers (one VGG-block, see Section 3.2).

To reduce the resources needed for training the model, the images can be split into smaller-sized sections, and a separate model can be trained on each section. As a side-effect, the attacker can find the secret's rough location. If the network does not reach a very high validation accuracy, but the secret bits can be learned to some degree, more measurements from the respective section might be required (supposedly also with higher resolution). The application of data augmentation techniques is likely to reduce the required number of measurements. Once single bits can be learned successfully, a multi-label classification can be attempted to reduce the training time. In other words, one network should learn more than one bit at the same time. This can be achieved by adding more output neurons to the FC network – one per bit of the key to be learned.

### 4.3 Secret Extraction

When all bits could be learned using the training dataset with a sufficiently high accuracy, the attacker knows the required

locations on the chip and measurement parameters for a successful extraction of the secret. She then can capture response images from the target device (containing an unknown secret) and let the obtained models predict the key from those images. Depending on the accuracy of the network, multiple images with slightly different parameters (like focus position) could be obtained for being able to apply a majority voting scheme on the predicted secret bits, and therefore, achieve a higher probability for predicting all bits of the secret correctly. In this work, we abstain from extracting the secret from a target device and instead rely on the test accuracy from the training phase as an indicator for the attack's success. However, we expect the inter-device differences to be lower than the noise introduced during different measurement runs and by data augmentation.

# 5 Experimental Setup and Target Devices

In the first part of this section we give details on our setup for conducting TLS and LLSI measurements. Then we briefly describe our setup for the learning part. Finally, we introduce the devices under test (DUTs) and present images of their memory structures captured with our setup.

## 5.1 Measurement Setup

### 5.1.1 Optical and Electrical Setup

The core of our setup is a Hamamatsu PHEMOS-1000 FA microscope. It is equipped with a 1.3 µm high-power incoherent light source (HIL) for optical probing and a 1.3 µm laser for thermal stimulation. In addition to the 5×, 20×, and 50× lenses, a scanner-zoom of 2×, 4×, and 8× is available. The light beam is scanned pixel-wise over the device using galvanometric mirrors. For acquiring optical images and conducting LLSI, the reflected light is separated by semi-transparent mirrors and fed into a detector. For LLSI, the detector's output is fed into a bandpass filter set to the frequency of interest. The PC software then produces a 2-D image containing the measured amplitude at each pixel. For conducting TLS measurements, the laser is scanned over the device, and its power consumption is measured using an external current preamplifier (Stanford Research Systems SR 570). The amplifier's output is fed into the PHEMOS PC software, which produces a response map of the locations sensitive to the thermal stimulation. The setup specific to the devices under test is described in Section 5.3.

### 5.1.2 Measurement Automation

For repeating the measurements with different secrets programmed into the target devices, we programmed a tool in the LabView programming environment. It can control the PHEMOS software (e.g., start and stop measurements, execute auto-focus, move the lens) and access the captured images for correcting horizontal and vertical drift. Furthermore, the tool can trigger the programming of a new secret into the DUT by communicating with a target-specific script running on another PC. In one iteration of the automated measurements, first a new secret is programmed. Then, after executing the auto-focus, an optical image is captured and saved. The drift between that image and the first image of the measurement series is calculated and the lens is moved accordingly. Finally, the TLS or LLSI measurement is conducted and the resulting image is saved along with the secret.

## 5.2 Learning Setup

For correcting drift in the final images, we made use of the MATLAB image processing toolbox. As machine learning toolbox, we used the Keras API for TensorFlow (version 2.3.0). We ran all our experiments on an Ubuntu 20.04.1 LTS machine with an Intel i7-6850K CPU @ 3.6 GHz, 128 GiB of system memory and a GeForce GTX 1080 Ti GPU with 11 GiB of memory. For all experiments, we made use of the TensorFlow GPU support.

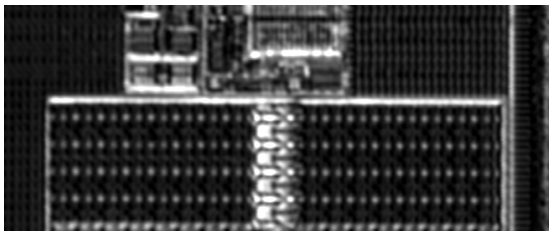
## 5.3 Devices under Test

We chose three different targets manufactured in different technology sizes and containing different kinds of volatile key memories for our evaluations.

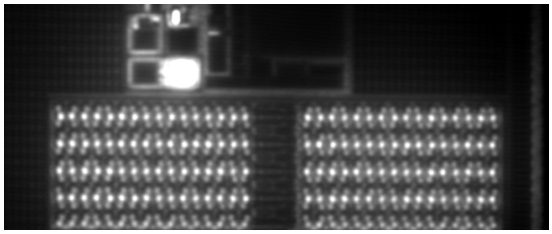
### 5.3.1 Xilinx Kintex Ultrascale BBRAM

As first and simple target we chose the battery-backed RAM (BBRAM) of a Xilinx Kintex Ultrascale FPGA, which is used for storing a 256-bit bitstream decrpytion key. In principle, BBRAM is identical to common SRAM – except that it is designed to be powered via battery over a long period. Therefore, BBRAM cells are susceptible to optical SCA attacks. In the literature it has been shown that the key from this device family can be extracted using TLS [10].

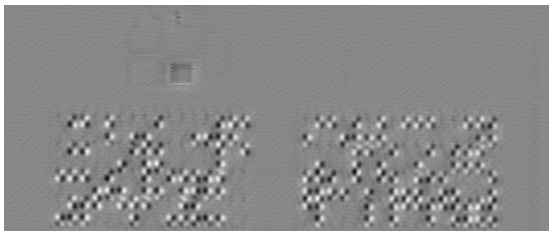
The FPGA, which is manufactured in a 20 nm technology, is mounted on an AVNET development board (AES-KU040-DB-G). The flip-chip package of the FPGA allows direct access to the silicon backside of the chip. For conducting TLS measurements, the current preamplifier is connected to the battery rails of the chip and the main power supply is switched off. The bias voltage of the amplifier supplies the BBRAM during the TLS measurement. For programming a new key, the FPGA has to be powered by the development board's power supplies. To fully automate the programming and measurement process, we made use of the supplies' PMBus interface, allowing to switch the power on and off programmatically via a microcontroller (using the TI PMBus library [44]). Consequently, for programming a new key, the power supplies are



(a) Optical image



(b) TLS response image with a random key programmed



(c) Difference between two TLS response images with different keys

Figure 4: Images of the Xilinx Ultrascale BBRAM.

switched on, a key is programmed via JTAG and the Xilinx Vivado TCL interface [45], and the power supplies are switched off again. Note that during the whole process, the BBRAM voltage is supplied by the current preamplifier. Fig. 4 showcases images of the BBRAM area captured with our setup. Although the chip is manufactured in a 20 nm technology, the size of one memory cell is around 2.8 µm × 3.1 µm, which can be explained by leakage current considerations [10].

### 5.3.2 Texas Instruments MSP430 SRAM

As second and more flexible target, we chose the freely programmable 1024-byte SRAM of a Texas Instruments MSP430 microcontroller. The chip is manufactured in a 180 nm technology with an SRAM cell size of approximately 2.5 µm × 1.9 µm [22]. The literature shows that the SRAM content of this device can be extracted using TLS and LLSI [22]. For our experiments, we chose to conduct LLSI measurements, as TLS is only possible while the device is in a low-power mode, which is not the case for LLSI. Hence, LLSI can be considered a more powerful technique in this case.

To access the chip backside, the device had to be opened and soldered backside-up on a custom PCB. Note that polishing or thinning the silicon backside was not necessary. For modulating the power supply of the SRAM memory, we made use of the VCORE pin, which provides access to the internally generated core voltage of the microcontroller. To this pin, we connected our modulator circuit, consisting of a voltage regulator whose feedback path is modulated using a laboratory frequency generator with a sinusoidal wave. For programming the SRAM content during the automated measurements, we used an Olimex JTAG debugger (MSP430-JTAG-TINY-V2), controlled by a Python script using the MSPDebug command line tool [46]. During the whole LLSI measurement, the debugger is left connected and switched on. Fig. 5 showcases images of the SRAM area captured using our setup.

### 5.3.3 Intel Cyclone IV Registers

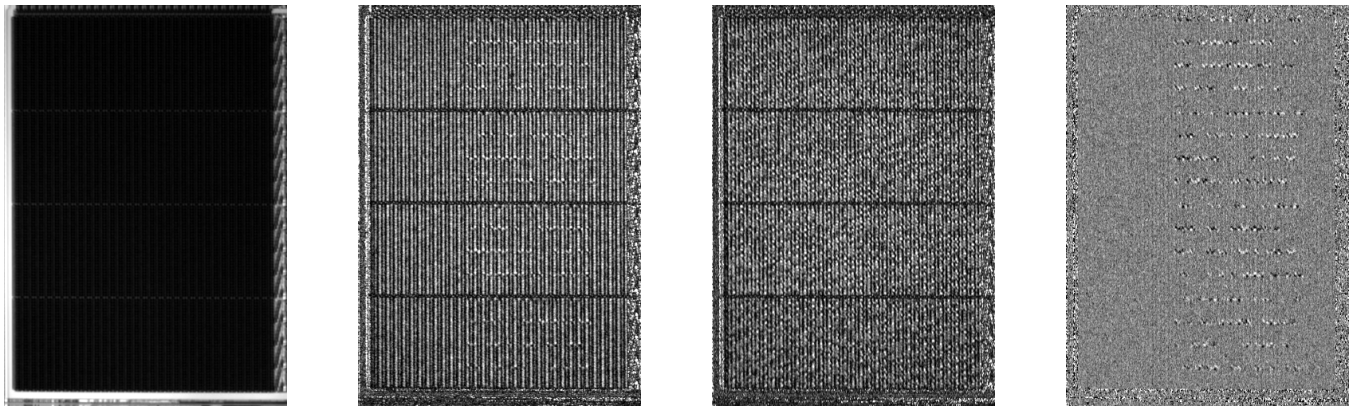
As the third target, we chose the registers of a Intel Cyclone IV FPGA. The FPGA consists of 392 identical logic array blocks (LABs), each comprised of 16 logic elements (LEs), whereas every LE contains one register cell. The chip is manufactured in a 60 nm technology. We had to open the package and solder the chip backside-up on a custom PCB for accessing the chip's backside. To modulate the supply voltage for conducting LLSI, we used a voltage regulator (TI TPS7A7001) and modulated its feedback path with a sinusoidal wave. We created a logic design that updates the register values when applying an external clock with precomputed randomly chosen values during the automated measurements.

By subtracting two LLSI images with different data, we found the LAB's area containing the registers. To reduce the measurement time, we covered only that area with the automated measurements. Consequently, one response image contains one logic array block, and therefore 16 registers, see Fig. 6. From the difference images, we could also estimate the memory cell size to around 7 µm × 9 µm.

## 6 Results

In this section we apply our deep learning based approach on the response images captured with the automated setup. For all experiments, we first reduced the drift – caused by mechanical instabilities of our setup – between the images in the dataset. For this, we calculated the offset between the optical image captured along with each response image and one fixed optical image by means of an elastic transformation using the MATLAB image processing toolbox. Then we applied the transformation to the corresponding response image. For the sake of simplicity, we will in the following refer to the response images only as "images". To encourage others working with our data, we made all images captured in the context of this work available online.[1]

[1] http://dx.doi.org/10.14279/depositonce-11354



(a) Optical image

(b) LLSI image (512 key bits, rest zeroized)

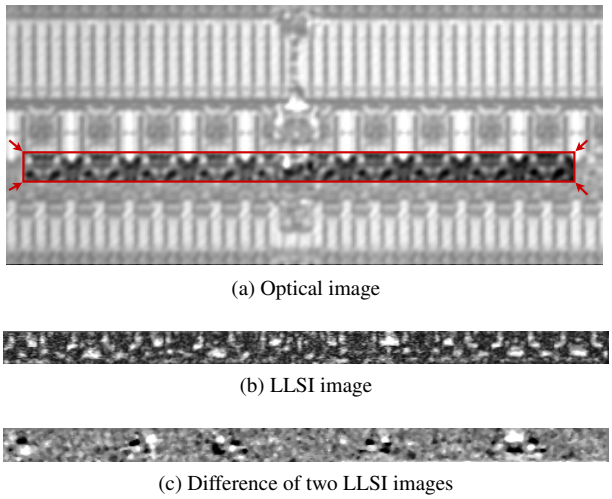
(c) LLSI image (512 key bits, rest randomized)

(d) Difference between two LLSI images (rest zeroized)

Figure 5: Images of the TI MSP430's 1024-byte SRAM area.

| Target | # Mem. bits | # Key bits | Technique | Image dimensions | Lens and scanner zoom | # Images | Time/Image (mm:ss) | Total time (hh:mm) |
|---|---|---|---|---|---|---|---|---|
| BBRAM | 288 | 256 | TLS | 985 px × 407 px | 50× (×2) | 578 | 02:02 | 19:35 |
| MSP430 (zeroized) | 8192 | 512 | LLSI | 503 px × 355 px | 50× | 433 | 13:00 | 93:49 |
| MSP430 (randomized) | 8192 | 512 | LLSI | 503 px × 355 px | 50× | 821 | 13:00 | 177:53 |
| FPGA Registers | 16 | 16 | LLSI | 509 px × 28 px | 50× (×2) | 568 | 2:40 | 25:17 |

Table 1: Overview of devices under test and the captured images in automated measurements.



(a) Optical image

(b) LLSI image

(c) Difference of two LLSI images

Figure 6: Images of one Intel Cyclone IV LAB containing 16 registers.

## 6.1 Key Extraction from BBRAM

Using the automated setup, we have captured over 500 TLS images of the BBRAM containing randomly chosen keys, see Tab. 1 for details. The memory cells' locations within the image become visible when subtracting two TLS images containing different keys, see Fig. 4c. The relatively large spots indicate that the memory cells cover many pixels, and therefore, we downsized the images with a factor of 0.4 before using them for training. We first investigated if it is possible to extract single key bits from the images (Section 6.1.1). Further, we examined ways for reducing the required time for learning (Section 6.1.2) and the number of images in the training dataset (Section 6.1.3). Finally, we constructed an optimized attack approach from our findings (Section 6.1.4).

### 6.1.1 Learning single bits

For the first experiments, we fed images containing the entire BBRAM area into the network (cf. Fig. 4). For the CNN, we used a simple VGG-like structure, consisting of just two convolutional layers, followed by a pooling layer, and a FC network with one hidden layer (512 neurons), a dropout layer (rate 0.2), and an output layer with one neuron. For the model summary, see Fig. 18 in the Appendix. For all experiments in this work, we used the Adam optimizer with an initial learning rate of 0.001, binary cross-entropy loss functions, and rectifier activation functions. We randomly split the available images into training (70%), validation (15%), and test (15%) datasets. Further, we applied a batch size of 8 images and set the number of steps per epoch to the number of images in the training dataset divided by the batch size. To deal with the relatively small datasets, we augmented the images by means of an affine transformation with a random rotation of

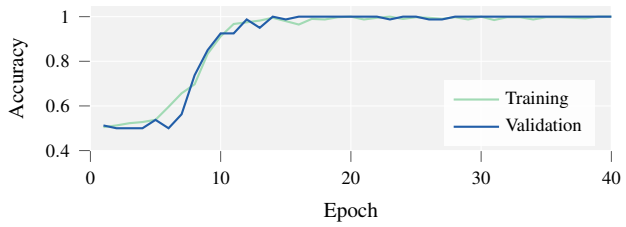



Figure 7: Training and validation accuracy when learning a single bit of the BBRAM key from the full image.

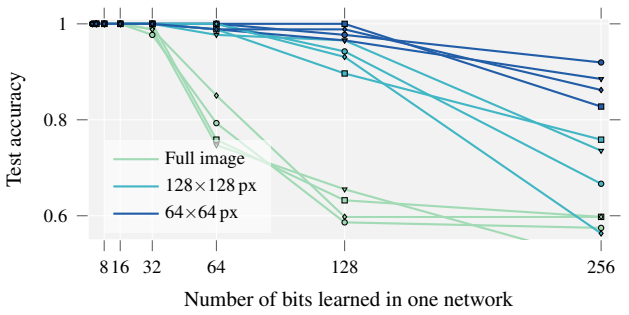



Figure 8: Test accuracies for four bits of the BBRAM key when trying to learn multiple bits in parallel with one network. Shown values depict the maximum out of 3 runs.

2 degrees, a width/height shift of 1 pixel, and a shear of 2 degrees.

The results show that the network can quickly learn one bit of the key, see Fig. 7. We repeated the experiment for 50 randomly chosen bit positions of the key, and recognized, that not all networks lead to a test accuracy of 100%. Therefore, we repeated the network training five times per key bit for different splits of the dataset. In most runs (at least 3 out of 5), we achieved a test accuracy of 100%. The reasons for some networks to perform better and some worse could be the relatively small number of training images and the random initialization of the networks' weights. To make predictions of the secret more reliable, an ensemble learning strategy can potentially be used, for instance, by considering the models from multiple runs in a majority voting fashion. Training a network for one bit took around 180 seconds per run, which – depending on the number of runs – can lead to a training time of some hours to a few days.

### 6.1.2 Learning bits in parallel

To speed up model training for all key bits, we added more output neurons to the network to learn multiple key bits in parallel. For this, we randomly chose bit positions from the key and checked if we can achieve a simultaneous test accuracy of 100% for all bits. This was the case for up to 4 key bits per network, when training for the same number of epochs as before on the full image, which leads to a 4× speedup in training time. Above 4 bits, the test accuracy was decreasing

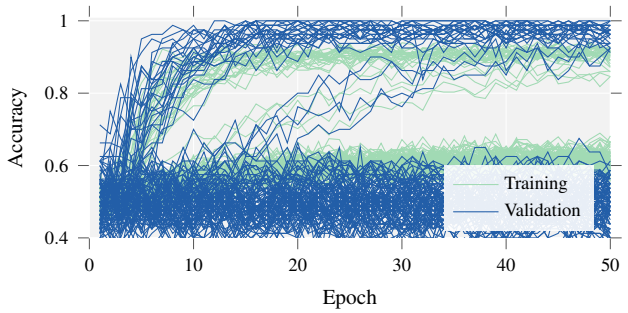



Figure 9: Training history when learning 128 bits of the BBRAM key on a 64 px × 64 px section. The bits contained in the section converge to 100% accuracy, and therefore, can be clearly separated from the others.

significantly. Increasing the number of convolutional and FC layers did not improve the prediction accuracy. Further, we noticed that the achieved performance depends on the spatial distance between the memory cells learned in parallel. When trying to learn cells in close vicinity, the per-bit accuracy is higher than with randomly chosen memory cells.

To further increase the number of bits learned in parallel, we reduced the network's data input dimensions by breaking the images into sections, and training one network for each section. Now not all key bits are contained within one section, and consequently, an accuracy of around 50% might indicate that the section does not contain the corresponding bit. Therefore, we picked four bits that are contained in a specific 128 px × 128 px and 64 px × 64 px section, and tried to learn up to 256 bits of the key in parallel from differently sized sections. The results confirm that a smaller section size leads to a higher accuracy. We could achieve a test accuracy of 100% for all four bits contained in the section when trying to learn up to 32 bits in parallel, see Fig. 8. Although not reaching a very high test accuracy, the network for learning 128 bits in parallel can clearly separate bits that are contained in the section from bits that are not, see Fig. 9. A few bits achieve a higher validation accuracy only in later epochs, presumably because they are not fully contained in the image section, and therefore, are harder to learn.

To sum up, this experiment has shown two things. Firstly, breaking the images into smaller sections can increase the achieved accuracy of the model. Secondly, the bits' rough locations on the image can be found very efficiently, by learning many bit positions of the key in parallel.

### 6.1.3 Reducing the number of required images

We expect the cost of using the FA microscope, i.e., for capturing the images, to be in orders of magnitude higher than the cost for training the CNNs. Therefore, we consider the required number of training images as the limiting factor regarding the attack costs. Consequently, we tried reduce the

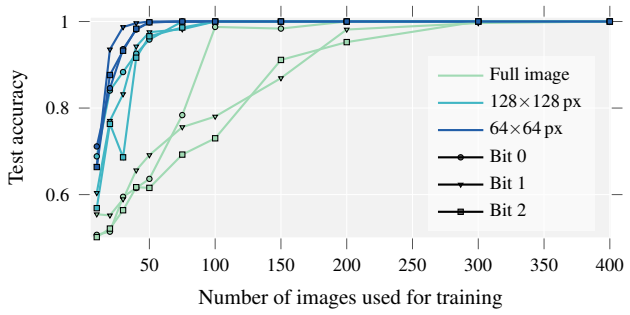



Figure 10: Learning one bit of the BBRAM key per network from differently sized sections, with respect to the number of images used for training. The experiment was repeated for three key bits.

number of samples used for training to a minimum, while still being able to extract the secret. For this, we again tried to learn only single bits per network, and repeated the experiment for three bits of the key on different image section sizes. In a nutshell, the results indicate that training on a smaller section size requires a smaller test dataset, see Fig. 10. Remarkably, to learn a single bit from a 64 px × 64 px section with 100% accuracy demands only 50 training images.

#### 6.1.4 Optimized attack approach

From the above findings, we can now develop an attack approach that is adaptable to constraints like the amount and quality of available images. We propose a two-step divide-and-conquer approach as follows. First, for finding the bits' coarse locations, networks are trained on many bits in parallel for small sections of the original image. Note that high test accuracies are dispensable in this case, since it is only of interest whether a bit is learnable or not. Once the coarse location of each bit is found, networks for each bit (or small groups of bits) can be trained on the corresponding sections.

**Localization** We chose to reduce the training dataset to only 150 images to better reflect a real attack scenario in which capturing time is expensive, resulting in a dataset acquisition time of 5 hours. We then trained networks for 128 bits in parallel on 64 px × 64 px sections of the images with 5 px overlap at every side, resulting in 21 sections, see Fig. 11. We ran every training three times and selected the most promising section for each bit by first filtering for test accuracies above 75% and then picking the section with the highest number of successful runs. For instance, some of the key bits between 0 and 127 could be learned in section 12, see Fig. 12. The algorithm found bit numbers 0-5, 32-37, 64-69, 96-101, 129, and 131-133[2]. This matches with the memory mapping already discovered in [10]. Note that the bits 128-133 seem to reside directly in the overlap region of sections 12 and 19, and therefore, some bits could be better learned in section 12, and some in 19. We could successfully find the corresponding section for every bit of the key. One training run took 133 seconds, which results in a total localization time of 4:42 hours.

Additionally, we reduced the dataset to 100 images and trained networks only for 64 bits in parallel on 64 px × 64 px sections. The experiment delivered the same localization results as before, with a slightly shorter training time (4:24 hours). Consequently, we believe that tweaks and optimizations can reduce the number of required images even further.

**Prediction** Once all bits' rough locations are known, at most one network training per key bit is necessary to predict all bits with high accuracy. The previous results indicate that there is a trade-off between training time and training dataset size. Training one network on a 64 px × 64 px image section for one key bit with a dataset consisting of 100 images takes around 30 seconds, resulting in a total training time for all bits of the key of 2:08 hours (for one run per model). To increase the bit prediction accuracy, multiple training runs can potentially be combined in an ensemble learning strategy with only a linear increase in training time.

### 6.2 Key Extraction from Microcontroller SRAM

On the microcontroller SRAM as our most flexible target, we evaluated two scenarios. In scenario 1 (Section 6.2.1), we programmed a randomly chosen key into 512 bits of the 1 kB (= 8192 bits) memory at the addresses `0x10 – 0x4f`, while keeping the rest of the memory zeroized. This scenario corresponds to the BBRAM target, except for the smaller memory cell sizes and the more distributed memory cells holding the key. In the scenario 2 (Section 6.2.2), the entire memory content is randomized. Again, we consider the same 512 bits of the memory to be the key which should be extracted. This scenario simulates a high amount of irrelevant information in the measurement, caused by other activities on the chip or intended obfuscation.

#### 6.2.1 Scenario 1: Rest zeroized

We captured over 400 images for this scenario, see Tab. 1 for details. Fig. 5d indicates that the images are not as clear as the BBRAM images. The reason is that we did not use an extra 2× scanner zoom like for the BBRAM, because we wanted to fit the whole memory into one image. Furthermore, the memory cells are slightly smaller than those of the BBRAM. The difference image of two different keys (see Fig. 5d) indicates that the key is distributed over large parts of the memory, and therefore, nearly the whole image must be considered for extracting the key bits. We first investigated the required number of images for learning one bit, see Fig. 13. The results show that around 100 images are sufficient to reach 100% test accuracy for a 64 px × 64 px section. For a 128 px × 128 px section, already around 400 images are required to achieve a

[2] Numbering with most significant bit first.

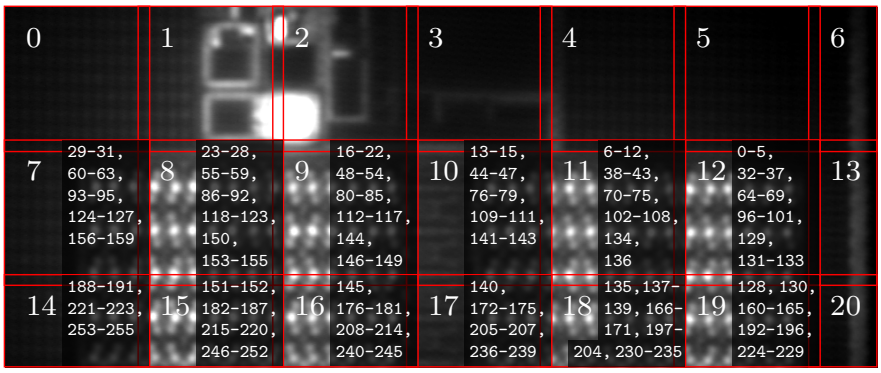



Figure 11: BBRAM memory area split into 64 px × 64 px sections. The small numbers indicate the localized key bits for each section (most significant bit = 0).

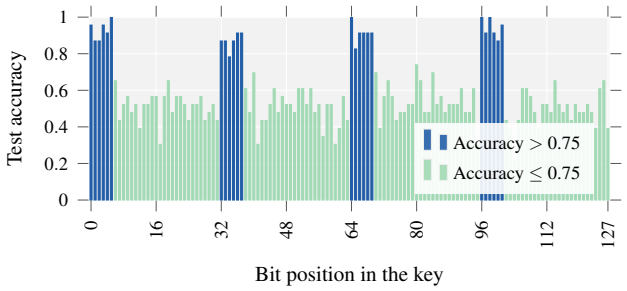



Figure 12: Trying to learn the first 128 bits of the BBRAM key in parallel on section 12 (see Fig. 11) with only 150 images used.

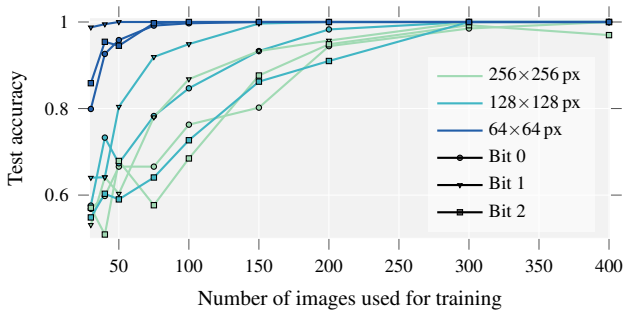



Figure 13: SRAM scenario 1 – Learning one key bit per network from differently sized sections with respect to the number of images used for training. The experiment was repeated for three bit positions.

test accuracy of 100%. The network architecture and setup working best is identical to the setup used for the BBRAM key extraction (Section 6.1.1).

For the localization step, we split the images into 64 px × 64 px sections, resulting in 54 sections, see Fig. 19 in the Appendix. For every section, we trained models on 128 key bits in parallel. We could localize all 512 key bits by using 300 images from the dataset. The results are shown in Tab. 2 in the Appendix. Note that the number of images can be reduced when accepting longer localization times – by learning less bits in paralel.

### 6.2.2 Scenario 2: Rest randomized

In the previous scenario, there was not much noise present in the images. However, on a real target, surrounding memory cells might not always hold the same value. Therefore, we randomized the entire memory content for scenario 2. The subtraction of two LLSI images shows that no longer any area of interest can be recognized, see Fig. 14. We assumed that this scenario is harder to learn, and therefore, captured over 800 images, see Tab. 1 for details.

As before, we first investigated how many images are required to extract single bits. The results show that – compared to scenario 1 – eight times more images are necessary to achieve a test accuracy of 100%, see Fig. 15. Interestingly, only one of the three bits achieves a test accuracy of 100% for 128 px × 128 px sections (Bit 1). Also for 64 px × 64 px sections, the other two bit positions (Bit 0 and Bit 2) show clearly worse accuracies. For the other bits and larger sections, the number of images seems to be insufficient to achieve a very high test accuracy.

When using 400 images for the localization and learning 128 bits per network, we could map 91% of the bits to the same sections as in scenario 1. In other words, 45 out of 512 bit positions were not found in their correct section. Therefore, we ran the same experiment using 800 images. Although still 12 bit positions were not mapped to the same sections as in scenario 1, they could be located in a neighboring section. The reason is that those bits seem to be located in the overlap region of the two sections. The results show that a high level of irrelevant information increases the amount of required images significantly. Nevertheless, extracting the key is still possible when spending enough time on measurements.

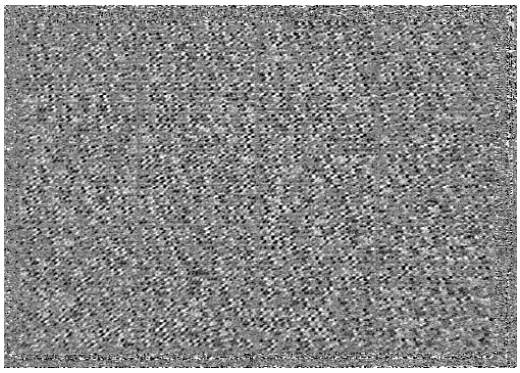


Figure 14: SRAM scenario 2 – Difference between two LLSI images with the entire memory randomized (image rotated clockwise by 90°).

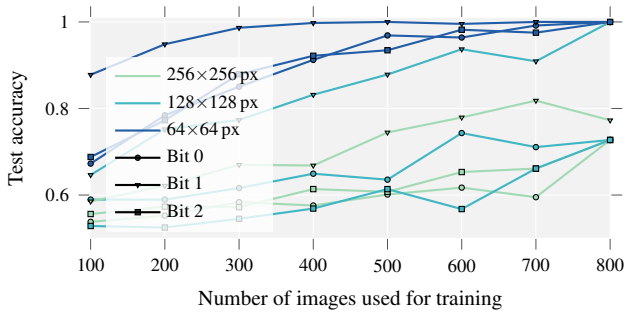



Figure 15: SRAM scenario 2 – Learning one bit per network from differently sized sections while the whole memory content is randomized. Experiment is repeated for three bit positions.

## 6.3 FPGA Register Content Extraction

Our dataset for this target consists of more than 500 images, each containing one logic array block (LAB) with 16 register bits, see Tab. 1 for details. Note that not all images show the physically same registers on the chip, but instead instances of the same logic layout. Therefore, if the bit values can be learned in our experiment, the resulting predictor can be used to extract data from all LABs distributed over the FPGA.

Like for the other targets, we investigated the influence of the training dataset size on the test accuracy when training networks on a single bit of the secret. The results indicate that – depending on the section size of the images – at most 150 images are required to achieve a test accuracy of 100%, see Fig. 16. Although the bits can already be learned from the full images with a low number of training samples, we further split the images into smaller sections to localize the individual bits in more detail. Fig. 17 shows the results for splitting the images into 8 sections, which already gives very precise information on the bits' position.

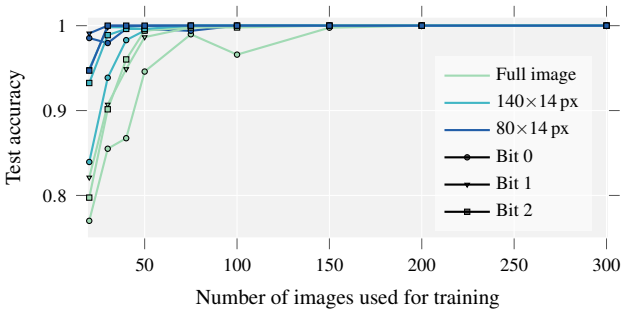



Figure 16: FPGA registers – Learning one bit per network from differently sized sections with respect to the number of images used for training and validation. The experiment was repeated for three bit positions.

# 7 Discussion

## 7.1 Scalability of Data Extraction

One important aspect is the scalability of our approach towards the extraction of larger chunks of data and implementations employing classical countermeasures against SCA attacks (e.g., Boolean masking).

In our experiments on the MSP430 microcontroller (Section 6.2), we have captured images of the full 1024-byte SRAM with randomly chosen content. We have defined 64 bytes in a fixed address range as the key bits, and have shown that all bits can be localized and extracted from the images. Since we could have chosen any other address range within the memory as key storage, it will also be possible to extract the entire memory content from the images with only a linear increase in extraction time. Consequently, we expect our approach to work also on larger chunks of data with only a linearly growing effort.

One might ask if the approach is also applicable when the key is not present in plaintext on the chip. Examples for implementations that do not require a key in plaintext are masked versions of cryptographic cores that work on shared forms of the key [47]. Previous work has already shown that all key shares can be extracted using laser-assisted SCA when all potential memory/register locations are known to the adversary: either by direct readout or with the help of a SAT solver [11]. In this work, we assume zero knowledge about the memory locations on the chip.

In preliminary experiments, we presume a 2-share Boolean masking of the key, meaning that the unmasked key can only be obtained by XOR'ing two values stored in the memory. We artificially created the masking on the available dataset by defining pairs of memory locations as the shares. In other words, on a memory snapshot containing $N$ bit values $b_0 \dots b_{N-1}$, one key bit $k$ for a 2-share masking is $k = b_x \oplus b_y$ $(0 \leq x, y < N, x \neq y)$. During the profiling phase, only the unmasked key $k$ is known to the adversary. We

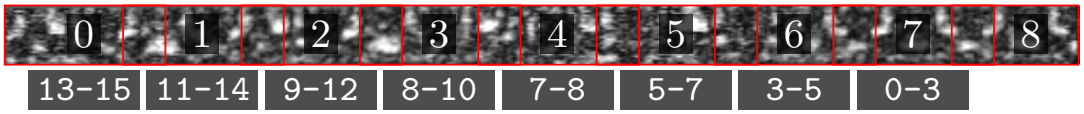



Figure 17: Sections of the FPGA register area for localizing the bits' rough positions. The number ranges indicate the bit positions of the secret localized in the respective section.

trained models on a 128 px × 128 px section of the BBRAM images containing $N = 47$ bits, and achieved 100% test accuracy for all exemplarily tested bit combinations (e.g., for $(x,y) \in \{(0,4),(1,8),(32,66)\}$, cf. Fig. 11).

Hence, the network has not only learned the memory locations of the individual shares, but also that the values have to be XOR'ed to obtain the unmasked key. We used the same neural network structure as in all the other experiments presented in this work and observed that the model needs to be trained for more epochs than for the direct key extraction. On the MSP430 microcontroller SRAM, we only had success on some bit combinations, and therefore, we believe that the network architecture will have to be adapted to work more reliably. A more thorough exploration of masked data extraction can be conducted in the future using the data collected in this work. In summary, also the unmasked key of a masked implementation can be extracted using our laser-assisted SCA approach.

## 7.2 Optical Resolution and Cell Size

Optical resolution is defined as the ability of an optical system to differentiate between two closely spaced objects. Because of constant decrease in feature sizes – now reaching down to the 5 nm node, optical resolution has been a growing concern for the FA community. Debugging the root cause of a failure can require to resolve adjacent minimum size transistors from each other, which might be challenging when we think of the most recent technology nodes. Tools such as the solid immersion lens (SIL) and visible light source systems [48, 49] have been introduced to overcome this problem. It has been shown that a SIL can improve the optical resolution down to approximately 200 nm, enabling optical probing even for 10 nm technology nodes [17, 50]. With our setup, we can achieve a laser spot diameter of approximately 1 µm without a SIL. Laser power at the center of the spot is the strongest and decreases exponentially through the edge.

The transistors in the SRAM cells are often designed to be larger than those used in the logic part of the chip to avoid off-leakage current related data loss. Although the DUTs in our experiments were manufactured in technology nodes down to 20 nm, the contained memory cells were larger than expected. Among the DUTs that we have used, the smallest cell size is 2.5 µm × 1.9 µm in MSP430 which is still larger than the 1 µm laser diameter. In the case of the Xilinx Ultrascale BBRAM, the cell size is even larger, although the technology size is much smaller. This shows that cell sizes do not always proportionally scale with the technology nodes, but cell size scaling also depends on many other parameters such as current leakage or supply voltage. The designers have to keep the transistor sizes bigger to maintain the circuit performance and the yield. In addition to that, while logic density continues to double in every technology generation, the memory cell size shrink cannot keep up the trend at the same pace. As a result, the memory density increase remains less than double at every new technology node [51]. The limiting factor appears to be lithography and the cost associated with it [52].

The question whether it is possible to extract logic states from memory cells that are smaller than the laser spot size can not be answered trivially. While for FA purposes it might be important to target only a single transistor, for our approach it is only important that the response image differs in some way between the logic states `0` and `1`. As a matter of fact, the distances and the positions of the opposite state transistors with respect to each other are more important than the transistor sizes. For our DUTs, we do not know the exact memory structure, and we have not tested our approach on memories other than presented in this work. However, this is among our future research interests.

For the optical SCA techniques used in this work, the laser beam is scanned over the device pixel-wise. When reducing the pixel size to values smaller than the laser spot diameter, for every pixel the superimposed signal/response originating from multiple transistors or memory cells will be captured. Consequently, the resulting response image will be noisy. We suppose that image processing tools like CNNs can be used to recover the logic state from the interfering signals. To the best of our knowledge, this has not been investigated in the hardware security community, and therefore, it is among our planned future works. In conclusion, the optical resolution might be a challenge when going to memories with smaller cell sizes and higher cell densities. However, we assume that optical contactless probing will continue to be present for a while due to the reasons mentioned above.

## 7.3 Chip Access

All the above mentioned SCA techniques are performed through the chip backside, which means that the attacker should have access to the bulk silicon. Since many modern ICs are manufactured in flip-chip packages, optical attacks are easy to conduct and often even do not require extra prepa-

ration steps. For instance, the Xilinx Kintex Ultrascale FPGA is shipped in a bare-die flip-chip package, and therefore, no preparation was needed for silicon access. In contrast, the packages of the other targeted devices had to be opened and soldered back-side up on a custom PCB for accessing the backside, which makes it a semi-invasive attack. Nevertheless, it should be noted that for technology nodes of 20 nm and below, flip-chip packages are becoming more prevalent due to performance, size and cost issues [53].

## 7.4 Attack Cost and Time Expenditure

Our investigations have shown that – depending on the area of interest on the chip and the imaging resolution – several hours to a few days have to be spent for automated measurements on the training device. This time is presumably the most costly period when conducting the proposed attack. This is not a challenge when the attacker owns a setup for conducting the attacks. The tools for conducting TLS and LLSI cost around $1M, whereas a setup for conducting only TLS can be acquired for around $100k [23]. Since a laser scanning microscope is common equipment in FA labs around the globe, a suitable setup can also be rented for about 300$/h including an operator. Consequently, we can calculate the costs for acquiring the images as given in Tab. 1 to $509 for the BBRAM (50 images), around $6.5k for scenario 1 (100 images) and $52k for scenario 2 on the microcontroller SRAM, and $667 for the FPGA registers (50 images). Note that due to the mostly automated measurements, which can also run unsupervised during the night, those fares could presumably be reduced. Furthermore, a more stable optical setup would avoid the need for a frequent auto-focus and drift correction, and therefore, can potentially reduce the measurement times according to our estimations by up to 50%. Although we agree that the costs are still high for some scenarios, we would like to stress that the gathered model is applicable to all devices of a device series, and can extract the secrets contained in multiple devices.

## 7.5 Key Control

One might argue that it is not always true that the adversary can program different keys into the NVM on a training device, for instance, when one-time programmable (OTP) memories like e-fuses or ROMs are used. We admit that such keys cannot be extracted using our approach. However, in many applications a OTP memory only stores a key-decryption-key, which is used to decrypt other application keys contained in reprogrammable NVMs. This makes the system more flexible and keys can be updated together with the device's firmware. Since the application keys will be decrypted by some cryptographic core on the device, they will in the end also be stored in registers on the chip. We have shown that this kind of application keys can be targeted using our approach.

## 7.6 Potential Countermeasures

When looking for potential countermeasures, one should keep in mind that potentially many different FA techniques can be used to read out the logic states of the device under attack. Therefore, a countermeasure should at best protects against all possible attack techniques. In other words, there exist various countermeasures that are effective against some FA-based attack techniques, but do not necessarily prevent other methods.

One technique proposed for protecting semiconductor intellectual property is IC camouflaging [54, 55]. Therefore, one might ask if camouflaging also can protect against memory readout. The idea behind camouflaging is to insert logic gates whose functionality cannot be extracted by delayering the chip and applying imaging techniques like SEM. However, since optical techniques rely on interactions with the actual transistors, they can still recognize the function of the camouflaged gates [54]. In other words, it would be possible to extract the logic states using activity maps of the circuit. Consequently, camouflaging does not seem to be an appropriate countermeasure.

The foremost requirement for our attack approach to succeed is access through the chip's backside. Active backside coatings [56] can prevent the optical access to the chip's silicon by adding an opaque coating layer. By actively checking the intactness of the coating, attempts to remove it can be detected. Since removing the silicon substrate from the chip backside is necessary for conducting SEM- and FIB-based attacks, an active coating can also help in these cases. However, to the best of our knowledge, there is no implementation of an active backside coating ready for mass production.

According to the preliminary results presented in Section 7.1 on masking implementations, Boolean masking seems to increase the effort for the attacker, but does not prevent laser-assisted SCA to a sufficient degree.

# 8 Conclusion

Hardware attacks using sophisticated FA tools are often seen as too costly and time-consuming to pose a severe threat to modern ICs and SoCs. Therefore, vendors usually rely on the complexity of the layout and tamper-proof memories to prevent key extraction. However, for being used, every key will be cached into memory cells that are vulnerable to probing techniques, such as optical SCA. In this work, we have shown that the automation of FA tools combined with deep learning techniques reduces the required effort by an adversary significantly. We carried out highly automated measurements on three different hardware targets holding an attacker-controlled secret in their memories. Besides, we have demonstrated how to fully extract the secret from the captured images without knowing the chip's layout, especially the memory cells' design, geometry, and exact location. We believe that our ap-

proach has the potential to antiquate the expensive reverse engineering part of hardware attacks by offering a very targeted and generic procedure for key extraction, which can also be applied in the presence of largely irrelevant information and activities on the chip. Hence, a great deal of attention has to be paid to this threat when designing new RoT devices for critical applications. While, in this work, we presented an offensive application of our approach, it also can be utilized to assess the vulnerability of the products in the early stages of the design, and consequently, assist in finding the right defense techniques.

## Acknowledgment

The work described in this paper has been supported in part by the Einstein Foundation in form of an Einstein professorship – EP-2018-480, in part by the Deutsche Forschungsgemeinschaft (DFG – German Research Foundation) under the priority programme SPP 2253 – 422730034, and in part by the German Ministry for Education and Research as BIFOLD – Berlin Institute for the Foundations of Learning and Data (ref. 01IS18025A). The authors would also like to acknowledge Hamamatsu Photonics K.K. Japan and Germany for their help and support on the PHEMOS system.

## References

[1] TechInsights Inc. *Semiconductor Analysis & IP Services*. 2020. URL: https://www.techinsights.com/.

[2] Texplained. *Hardware Security Insight*. 2021. URL: https://www.texplained.com/.

[3] S. E. Quadir, J. Chen, D. Forte, N. Asadizanjani, S. Shahbazmohamadi, L. Wang, J. Chandy, and M. Tehranipoor. "A Survey on Chip to System Reverse Engineering". In: *ACM Journal on Emerging Technologies in Computing Systems* 13.1 (2016), 6:1–6:34. DOI: 10.1145/2755563.

[4] C. Kison, J. Frinken, and C. Paar. "Finding the AES Bits in the Haystack: Reverse Engineering and SCA Using Voltage Contrast". In: *Cryptographic Hardware and Embedded Systems – CHES 2015*. Springer, 2015, pp. 641–660. DOI: 10.1007/978-3-662-48324-4_32.

[5] F. Courbon, S. Skorobogatov, and C. Woods. "Reverse Engineering Flash EEPROM Memories Using Scanning Electron Microscopy". In: *International Conference on Smart Card Research and Advanced Applications*. Springer, 2017, pp. 57–72. DOI: 10.17863/CAM.7164.

[6] C. Helfmeier, D. Nedospasov, C. Tarnovsky, J. S. Krissler, C. Boit, and J.-P. Seifert. "Breaking and Entering Through the Silicon". In: *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*. ACM, 2013, pp. 733–744. DOI: 10.1145/2508859.2516717.

[7] O. Kömmerling and M. G. Kuhn. "Design Principles for Tamper-Resistant Smartcard Processors". In: *Proceedings of the USENIX Workshop on Smartcard Technology (WOST'99)*. USENIX Association, 1999.

[8] H. Lohrke, S. Tajik, C. Boit, and J.-P. Seifert. "No Place to Hide: Contactless Probing of Secret Data on FPGAs". In: *Cryptographic Hardware and Embedded Systems – CHES 2016*. Springer, 2016, pp. 147–167. DOI: 10.1007/978-3-662-53140-2_8.

[9] S. Tajik, H. Lohrke, J.-P. Seifert, and C. Boit. "On the Power of Optical Contactless Probing: Attacking Bitstream Encryption of FPGAs". In: *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, 2017, pp. 1661–1674. DOI: 10.1145/3133956.3134039.

[10] H. Lohrke, S. Tajik, T. Krachenfels, C. Boit, and J.-P. Seifert. "Key Extraction Using Thermal Laser Stimulation". In: *IACR Transactions on Cryptographic Hardware and Embedded Systems* (2018), pp. 573–595. DOI: 10.13154/tches.v2018.i3.573-595.

[11] T. Krachenfels, F. Ganji, A. Moradi, S. Tajik, and J.-P. Seifert. *Real-World Snapshots vs. Theory: Questioning the t-Probing Security Model*. 2020. arXiv: 2009.04263 [cs].

[12] H. Maghrebi, T. Portigliatti, and E. Prouff. "Breaking Cryptographic Implementations Using Deep Learning Techniques". In: *Security, Privacy, and Applied Cryptography Engineering*. Springer, 2016, pp. 3–26. DOI: 10.1007/978-3-319-49445-6_1.

[13] R. Benadjila, E. Prouff, R. Strullu, E. Cagli, and C. Dumas. *Study of Deep Learning Techniques for Side-Channel Analysis and Introduction to ASCAD Database*. 2018. URL: https://eprint.iacr.org/2018/053.

[14] T. Kubota, K. Yoshida, M. Shiozaki, and T. Fujino. "Deep Learning Side-Channel Attack Against Hardware Implementations of AES". In: *2019 22nd Euromicro Conference on Digital System Design (DSD)*. 2019, pp. 261–268. DOI: 10.1109/DSD.2019.00046.

[15] S. R. Hou, Y. J. Zhou, and H. M. Liu. "Convolutional Neural Networks for Profiled Side-Channel Analysis". In: *Radioengineering* 27.3 (2019), pp. 651–658. DOI: 10.13164/re.2019.0651.

[16] M. T. Rahman, S. Tajik, M. S. Rahman, M. Tehranipoor, and N. Asadizanjani. "The Key Is Left under the Mat: On the Inappropriate Security Assumption of Logic Locking Schemes". In: *Proceedings of the IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*. 2020.

[17] C. Boit, S. Tajik, P. Scholz, E. Amini, A. Beyreuther, H. Lohrke, and J. P. Seifert. "From IC Debug to Hardware Security Risk: The Power of Backside Access and Optical Interaction". In: *Proceedings of the 23rd International Symposium on the Physical and Failure Analysis of Integrated Circuits (IPFA)*. IEEE, 2016, pp. 365–369. DOI: 10.1109/IPFA.2016.7564318.

[18] S. Tajik, D. Nedospasov, C. Helfmeier, J.-P. Seifert, and C. Boit. "Emission Analysis of Hardware Implementations". In: *17th Euromicro Conference on Digital System Design*. IEEE, 2014, pp. 528–534. DOI: 10.1109/DSD.2014.64.

[19] D. Nedospasov, J. P. Seifert, C. Helfmeier, and C. Boit. "Invasive PUF Analysis". In: *Proceedings of the 2013 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)*. IEEE, 2013, pp. 30–38. DOI: 10.1109/FDTC.2013.19.

[20] B. Niu, G. M. E. Khoo, Y.-C. S. Chen, F. Chapman, D. Bockelman, and T. Tong. "Laser Logic State Imaging (LLSI)". In: *Proceedings from the 40th International Symposium for Testing and Failure Analysis (ISTFA 2014)*. ASM International, 2014, p. 65.

[21] T. H. Geballe and G. W. Hull. "Seebeck Effect in Silicon". In: *Physical Review* 98.4 (1955), pp. 940–947. DOI: 10.1103/PhysRev.98.940.

[22] T. Kiyan, H. Lohrke, and C. Boit. "Comparative Assessment of Optical Techniques for Semi-Invasive SRAM Data Read-out on an MSP430 Microcontroller". In: *ISTFA 2018: Proceedings from the 44th International Symposium for Testing and Failure Analysis*. ASM International, 2018, p. 266.

[23] T. Krachenfels, H. Lohrke, J.-P. Seifert, E. Dietz, S. Frohmann, and H.-W. Hübers. "Evaluation of Low-Cost Thermal Laser Stimulation for Data Extraction and Key Readout". In: *Journal of Hardware and Systems Security* 4.1 (2020), pp. 24–33. DOI: 10.1007/s41635-019-00083-9.

[24] W. M. Yee, M. Paniccia, T. Eiles, and V. Rao. "Laser Voltage Probe (LVP): A Novel Optical Probing Technology for Flip-Chip Packaged Microprocessors". In: *Proceedings of the 1999 7th International Symposium on the Physical and Failure Analysis of Integrated Circuits (IPFA)*. 1999, pp. 15–20. DOI: 10.1109/IPFA.1999.791222.

[25] H. Zhang, P. Tian, X. Qian, and W. Wang. "Electro Optical Probing / Frequency Mapping (EOP/EOFM) Application in Failure Isolation of Advanced Analogue Devices". In: *2017 IEEE 24th International Symposium on the Physical and Failure Analysis of Integrated Circuits (IPFA)*. 2017. DOI: 10.1109/IPFA.2017.8060131.

[26] T. Moos. "Static Power SCA of Sub-100 Nm CMOS ASICs and the Insecurity of Masking Schemes in Low-Noise Environments". In: *IACR Transactions on Cryptographic Hardware and Embedded Systems* (2019), pp. 202–232. DOI: 10.13154/tches.v2019.i3.202-232.

[27] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. "Backpropagation Applied to Handwritten Zip Code Recognition". In: *Neural Computation* 1.4 (1989), pp. 541–551. DOI: 10.1162/neco.1989.1.4.541.

[28] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-Based Learning Applied to Document Recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324. DOI: 10.1109/5.726791.

[29] Y. LeCun, F. J. Huang, and L. Bottou. "Learning Methods for Generic Object Recognition with Invariance to Pose and Lighting". In: *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004.* Vol. 2. 2004. DOI: 10.1109/CVPR.2004.1315150.

[30] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. *ImageNet Large Scale Visual Recognition Challenge*. 2015. arXiv: 1409.0575 [cs].

[31] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen. "Deep Learning for Generic Object Detection: A Survey". In: *International Journal of Computer Vision* 128.2 (2020), pp. 261–318. DOI: 10.1007/s11263-019-01247-4.

[32] A. Krizhevsky, I. Sutskever, and G. E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*. 2012, pp. 1106–1114. DOI: 10.1145/3065386.

[33] K. Simonyan and A. Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2015. arXiv: 1409.1556 [cs.CV].

[34] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. *Improving Neural Networks by Preventing Co-Adaptation of Feature Detectors*. 2012. arXiv: 1207.0580 [cs].

[35] P. Simard, D. Steinkraus, and J. Platt. "Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis". In: *Seventh International Conference on Document Analysis and Recognition*. 2003, pp. 958–963. DOI: 10.1109/ICDAR.2003.1227801.

[36] L. Lerman, R. Poussier, G. Bontempi, O. Markowitch, and F.-X. Standaert. "Template Attacks vs. Machine Learning Revisited (and the Curse of Dimensionality in Side-Channel Analysis)". In: *Constructive Side-Channel Analysis and Secure Design*. Springer International Publishing, 2015, pp. 20–33. DOI: 10.1007/978-3-319-21476-4_2.

[37] F.-X. Standaert, F. Koeune, and W. Schindler. "How to Compare Profiled Side-Channel Attacks?" In: *Applied Cryptography and Network Security*. Springer, 2009, pp. 485–498. DOI: 10.1007/978-3-642-01957-9_30.

[38] O. Choudary and M. G. Kuhn. "Template Attacks on Different Devices". In: *Constructive Side-Channel Analysis and Secure Design*. Lecture Notes in Computer Science. Springer International Publishing, 2014, pp. 179–198. DOI: 10.1007/978-3-319-10175-0_13.

[39] R. Elnaggar and K. Chakrabarty. "Machine Learning for Hardware Security: Opportunities and Risks". In: *Journal of Electronic Testing* 34.2 (Apr. 1, 2018), pp. 183–201. DOI: 10.1007/s10836-018-5726-9.

[40] G. Hospodar, B. Gierlichs, E. De Mulder, I. Verbauwhede, and J. Vandewalle. "Machine Learning in Side-Channel Analysis: A First Study". In: *Journal of Cryptographic Engineering* 1.4 (2011), p. 293. DOI: 10.1007/s13389-011-0023-x.

[41] K. Hasegawa, M. Yanagisawa, and N. Togawa. "Hardware Trojans Classification for Gate-Level Netlists Using Multi-Layer Neural Networks". In: *2017 IEEE 23rd International Symposium on On-Line Testing and Robust System Design (IOLTS)*. 2017, pp. 227–232. DOI: 10.1109/IOLTS.2017.8046227.

[42] M. Chen and P. Liu. *Deep Learning-Based FPGA Function Block Detection Method Using an Image-Coded Representation of Bitstream*. July 20, 2020. arXiv: 2007.11434 [cs, eess].

[43] C. Boit, T. Kiyan, T. Krachenfels, and J.-P. Seifert. "Logic State Imaging From FA Techniques for Special Applications to One of the Most Powerful Hardware Security Side-Channel Threats". In: *2020 IEEE International Symposium on the Physical and Failure Analysis of Integrated Circuits (IPFA)*. 2020, pp. 1–7. DOI: 10.1109/IPFA49335.2020.9261000.

[44] Texas Instruments Inc. *MSP-PMBUS PMBus Software Library for MSP MCUs*. Version 1.0. 2015. URL: https://www.ti.com/tool/MSP-PMBUS.

[45] Xilinx Inc. *Vivado Design Suite Tcl Command Reference Guide (UG835 )*. 2019.

[46] D. Beer. *Dlbeer/Mspdebug*. 2020. URL: https://github.com/dlbeer/mspdebug.

[47] Y. Ishai, A. Sahai, and D. A. Wagner. "Private Circuits: Securing Hardware against Probing Attacks". In: *Advances in Cryptology - CRYPTO 2003*. Vol. 2729. LNCS. Springer, 2003, pp. 463–481. DOI: 10.1007/978-3-540-45146-4_27.

[48] J. Beutler, V. C. Hodges, J. J. Clement, J. Stevens, E. I. C. Jr, S. Silverman, and R. Chivas. *Visible Light LVP on Bulk Silicon Devices*. Tech. rep. Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), 2015.

[49] C. Boit, H. Lohrke, P. Scholz, A. Beyreuther, U. Kerst, and Y. Iwaki. "Contactless Visible Light Probing for Nanoscale ICs through 10 $\mu$m Bulk Silicon". In: *Proceedings of the 35th Annual NANO Testing Symposium (NANOTS 2015)*. 2015, pp. 215–221.

[50] M. Von Haartman, S. Rahman, S. Ganguly, J. Verma, A. Umair, and T. Deborde. "Optical Fault Isolation and Nanoprobing Techniques for the 10 Nm Technology Node and Beyond". In: *Proceedings of the 41st International Symposium for Testing and Failure Analysis*. 2015, pp. 47–51.

[51] D. Maheshwari. "6.1 Memory and System Architecture for 400Gb/s Networking and Beyond". In: *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*. 2014, pp. 116–117. DOI: 10.1109/ISSCC.2014.6757362.

[52] A. Keshavarzi, D. Maheshwari, D. Mattos, R. Kapre, S. Krishnegowda, M. Whately, and S. Gopalswamy. "Directions in Future of SRAM with QDR-WideIO for High Performance Networking Applications and Beyond". In: *Proceedings of the IEEE 2014 Custom Integrated Circuits Conference*. 2014, pp. 1–6. DOI: 10.1109/CICC.2014.6946029.

[53] H. Tong, Y. Lai, and C. Wong. *Advanced Flip Chip Packaging*. Springer US, 2013.

[54] B. Shakya, H. Shen, M. Tehranipoor, and D. Forte. "Covert Gates: Protecting Integrated Circuits with Undetectable Camouflaging". In: *IACR Transactions on Cryptographic Hardware and Embedded Systems, 2019(3)* (2019), pp. 86–118. DOI: 10.13154/tches.v2019.i3.86-118.

[55] J. Rajendran, M. Sam, O. Sinanoglu, and R. Karri. "Security analysis of integrated circuit camouflaging". In: *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*. ACM, 2013, pp. 709–720. DOI: 10.1145/2508859.2516656.

[56] E. Amini, A. Beyreuther, N. Herfurth, A. Steigert, B. Szyszka, and C. Boit. "Assessment of a Chip Backside Protection". In: *Journal of Hardware and Systems Security* 2.4 (2018), pp. 345–352. DOI: 10.1007/s41635-018-0052-3.

# Appendix

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
inputImage (InputLayer)      [(None, 158, 384, 1)]     0
_________________________________________________________________
conv2d (Conv2D)              (None, 158, 384, 32)      320
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 158, 384, 32)      9248
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 79, 192, 32)       0
_________________________________________________________________
flatten (Flatten)            (None, 485376)            0
_________________________________________________________________
dense (Dense)                (None, 512)               248513024
_________________________________________________________________
activation (Activation)      (None, 512)               0
_________________________________________________________________
dropout (Dropout)            (None, 512)               0
_________________________________________________________________
outputBit079 (Dense)         (None, 1)                 513
=================================================================
Total params: 248,523,105
Trainable params: 248,523,105
Non-trainable params: 0
_________________________________________________________________
```

Figure 18: CNN model summary for the BBRAM experiments, here for learning bit 79 of the key.

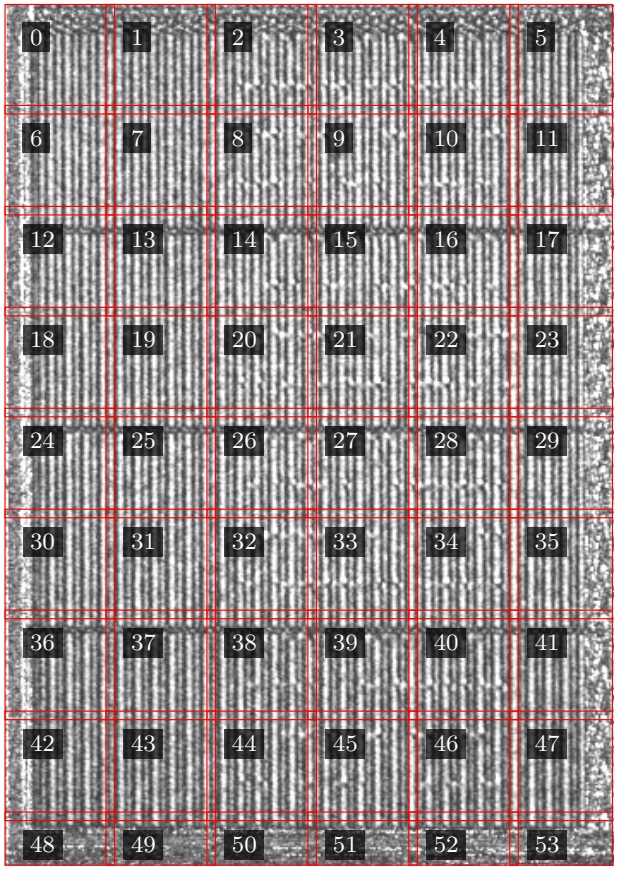



Figure 19: Sections of the MSP430's SRAM area used for localizing the bits.

| **Section** | **Bit positions** (most significant bit first) |
|---|---|
| 2 | 375, 383, 391, 399, 407, 415, 423, 431, 439, 447, 455, 463, 471, 479, 487, 495, 503, 511 |
| 3 | 183, 191, 199, 207, 215, 223, 231, 239, 247, 255, 263, 271, 279, 287, 295, 303, 311, 319, 327, 335, 343, 351, 359, 367 |
| 4 | 7, 15, 23, 31, 39, 47, 55, 63, 71, 79, 87, 95, 103, 111, 119, 127, 135, 143, 151, 159, 167, 175 |
| 8 | 374, 382, 390, 398, 406, 414, 422, 430, 438, 446, 454, 462, 470, 478, 486, 494, 502, 510 |
| 9 | 182, 198, 206, 214, 222, 230, 238, 246, 254, 262, 270, 278, 286, 294, 302, 310, 318, 326, 334, 342, 350, 358, 366 |
| 10 | 6, 14, 22, 30, 38, 46, 54, 62, 70, 78, 86, 94, 102, 110, 118, 126, 134, 142, 150, 158, 166, 174, 190 |
| 14 | 373, 381, 389, 397, 405, 413, 421, 429, 437, 445, 453, 461, 469, 477, 485, 493, 501, 509 |
| 15 | 181, 189, 197, 205, 213, 221, 229, 237, 245, 253, 261, 269, 277, 285, 293, 301, 309, 317, 325, 333, 341, 349, 357, 365 |
| 16 | 5, 13, 21, 29, 37, 45, 53, 61, 69, 77, 85, 93, 101, 109, 117, 125, 133, 141, 149, 157, 165, 173 |
| 20 | 372, 380, 388, 396, 404, 412, 420, 428, 436, 444, 452, 460, 468, 476, 484, 492, 500, 508 |
| 21 | 188, 196, 204, 212, 220, 228, 236, 244, 252, 260, 268, 276, 284, 292, 300, 308, 316, 324, 332, 340, 348, 356, 364 |
| 22 | 4, 12, 20, 28, 36, 44, 52, 60, 68, 76, 84, 92, 100, 108, 116, 124, 132, 140, 148, 156, 164, 172, 180 |
| 26 | 371, 387, 395, 403, 411, 419, 427, 435, 443, 451, 459, 467, 475, 483, 491, 499, 507 |
| 27 | 179, 195, 203, 211, 219, 227, 235, 243, 251, 259, 267, 275, 283, 291, 299, 307, 315, 323, 331, 339, 347, 355, 363, 379 |
| 28 | 3, 11, 19, 27, 35, 43, 51, 59, 67, 75, 83, 91, 99, 107, 115, 123, 131, 139, 147, 155, 163, 171, 187 |
| 32 | 370, 378, 386, 394, 402, 410, 418, 426, 434, 442, 450, 458, 466, 474, 482, 490, 498, 506 |
| 33 | 178, 186, 194, 202, 210, 218, 226, 234, 242, 250, 258, 266, 274, 282, 290, 298, 306, 314, 322, 330, 338, 346, 354, 362 |
| 34 | 2, 10, 18, 26, 34, 42, 50, 58, 66, 74, 82, 90, 98, 106, 114, 122, 130, 138, 146, 154, 162, 170 |
| 38 | 369, 377, 385, 393, 401, 409, 417, 425, 433, 441, 449, 457, 465, 473, 481, 489, 497, 505 |
| 39 | 185, 193, 201, 209, 217, 225, 233, 241, 249, 257, 265, 273, 281, 289, 297, 305, 313, 321, 329, 337, 345, 353, 361 |
| 40 | 1, 9, 17, 25, 33, 41, 49, 57, 65, 73, 81, 89, 97, 105, 113, 121, 129, 137, 145, 153, 161, 169, 177 |
| 44 | 368, 376, 384, 392, 400, 408, 416, 424, 432, 440, 448, 456, 464, 472, 480, 488, 496, 504 |
| 45 | 192, 200, 208, 216, 224, 232, 240, 248, 256, 264, 272, 280, 288, 296, 304, 312, 320, 328, 336, 344, 352, 360 |
| 46 | 0, 8, 16, 24, 32, 40, 48, 56, 64, 72, 80, 88, 96, 104, 112, 120, 128, 136, 144, 152, 160, 168, 176, 184 |

Table 2: Localization of the key in the MSP430's SRAM for the sections shown in Fig. 19.

# 5 LOW-COST THERMAL LASER STIMULATION

## 5.1 FAILURE ANALYSIS MICROSCOPES AND ALTERNATIVES

In the previously presented publications, we used the Hamamatsu Phemos-1000 [48] FA microscope designed to aid chip manufacturers and designers in debugging their devices. Alternative devices on the market with a comparable feature set are the Semicaps 1100 [49] or the Thermo Fisher Meridian 4 [50]. However, due to the relatively large feature set, such devices are expensive, with prices of around one million dollars. Although setups for debugging silicon are available for rent in labs worldwide for a few hundred dollars per hour, not every adversary might want to use these facilities or can get access there. Therefore, it is believed that the limited availability of suitable setups for logic state imaging techniques can prevent attacks. However, considering that a single technique like TLS or LLSI is sufficient to read out memory contents, it might be possible to gather a setup for a much lower price.

*Alternative setups for laser-based logic state imaging*

Both TLS and LLSI use the laser scanning feature of the microscope. However, optical probing techniques such as LLSI necessitate the reflected light to be captured and analyzed. That requires a specially optimized optical setup, a detector for the reflected light, and a spectrum analyzer to filter for the frequency of interest. Using something other than an existing LSM as a basis for the setup would thus not be easily possible. Consequently, in [16] it is shown that an old LSM can be retrofitted for optical probing techniques. The costs for such a setup are presumably much lower than for a recent FA microscope that supports optical probing. In contrast to LLSI, for TLS, the chip only has to be heated up by the laser, and the reflected light is irrelevant. Merely the DUT's current consumption has to be measured while scanning with the laser over the area of interest. The two-dimensional map of current consumption over the scanning position already constitutes the TLS logic state image.

*Research question*

A cheaper setup for TLS would show that the hurdle for conducting laser-based logic state imaging attacks is lower than expected. Given the relatively simple principle of TLS, we asked ourselves if a cheaper setup produces comparable results to a professional one. In this regard, we started to investigate the possibility of using a setup initially designed for laser fault injection (LFI) by adding the functionality needed for TLS ourselves. Suitable commercial setups for LFI are available from Riscure [51] or Alphanov [52]. They consist of a microscope objective through which a laser can be focused on the

DUT. Mechanical stages move the objective over the DUT. In the following publication, we used an Alphanov S-LMS station to investigate if it can serve as TLS platform and how it performs compared with a traditional FA microscope.

## 5.2 PUBLICATION

The original publication is reprinted in the following. It was presented at the *International Conference on Physical Assurance and Inspection of Electronics (PAINE 2019)* and published in the *Journal of Hardware and Systems Security* [53].

# Evaluation of Low-Cost Thermal Laser Stimulation for Data Extraction and Key Readout

**Thilo Krachenfels[1] · Heiko Lohrke[1] · Jean-Pierre Seifert[1] · Enrico Dietz[2] · Sven Frohmann[2] · Heinz-Wilhelm Hübers[2,3]**

Received: 5 June 2019 / Accepted: 2 October 2019 / Published online: 20 November 2019


**Abstract**

Recent attacks using thermal laser stimulation (TLS) have shown that it is possible to extract cryptographic keys from the battery-backed memory on state-of-the-art field-programmable gate arrays (FPGAs). However, the professional failure analysis microscopes usually employed for these attacks cost in the order of 500k to 1M dollars. In this work, we evaluate the use of a cheaper commercial laser fault injection station retrofitted with a suitable amplifier and light source to enable TLS. We demonstrate that TLS attacks are possible at a hardware cost of around 100k dollars. This constitutes a reduction of the resources required by the attacker by a factor of at least five. We showcase two actual attacks: data extraction from the SRAM memory of a low-power microcontroller and decryption key extraction from a 20-nm technology FPGA device. The strengths and weaknesses of our low-cost approach are then discussed in comparison with the conventional failure analysis equipment approach. In general, this work demonstrates that TLS backside attacks are available at a much lower cost than previously expected.

**Keywords** IC security · Optical attacks · Thermal laser stimulation · FPGA security

✉ Thilo Krachenfels
tkrachenfels@sect.tu-berlin.de

Heiko Lohrke
lohrke@sect.tu-berlin.de

Jean-Pierre Seifert
jpseifert@sect.tu-berlin.de

Enrico Dietz
enrico.dietz@dlr.de

Sven Frohmann
sven.frohmann@dlr.de

Heinz-Wilhelm Hübers
heinz-wilhelm.huebers@dlr.de

1 Chair of Security in Telecommunications, Technische Universität Berlin, Berlin, Germany

2 Institute of Optical Sensor Systems, Deutsches Zentrum für Luft- und Raumfahrt (DLR), Berlin, Germany

3 Department of Physics, Humboldt-Universität zu Berlin, Berlin, Germany



## 1 Introduction

Data extraction from integrated circuits (ICs) can pose a serious threat to the secrets stored within. Extraction of cryptographic keys, sensitive data stored in memory, or device fingerprint information, as used in physically unclonable functions (PUFs), allows attackers to break security features. Physical attacks, such as side-channel attacks, are one of the main approaches to extract data contained in embedded devices.

Thermal laser stimulation (TLS) is one such technique, which analyzes changes in the current consumption of the device in response to applied laser radiation. In the past, it has been used to read out the content of static random-access memory (SRAM) and thus allows the characterization of SRAM PUFs [1]. It was also applied to extract the key from the battery-backed random-access memory (BBRAM) contained within the decryption unit of a 20-nm technology field-programmable gate array (FPGA) [2]. Furthermore, TLS can be considered a suitable technique for the readout of microcontroller SRAM working memory [3]. Therefore,

it is a powerful data extraction tool for an attacker on the hardware level.

However, all previously mentioned experiments have been conducted using professional failure analysis (FA) equipment, more specifically a Hamamatsu Phemos-1000 laser scanning microscope (LSM). Such a system typically costs around 500k to 1M dollars, and even when renting, costs for the development of a TLS attack are still in the range of thousands of dollars [2]. As a consequence, even though TLS is a powerful attack technique, the connected costs might discourage attackers from applying it.

Yet, it needs to be kept in mind that FA equipment usually offers a lot more features than an attacker might actually need, for instance, support for wafer handling and automated testing equipment, very fast acquisition times, and integration of other measurement techniques, such as photon emission. In principle, however, all that is needed for a TLS attack is a way to move a laser spot over the device and simultaneously measure a current. This raises the question if attackers might be able to use simpler, more low-cost setups. If so, the threat posed by TLS techniques would be larger than expected so far. The main aim of this work is to determine if this is the case.

To evaluate this question, suitable alternatives to the usually employed FA systems need to be considered. One such candidate are commercially available setups used for evaluation of laser fault injection (LFI) which are by a factor of around 5 to 10 cheaper than FA LSMs. Such systems usually feature a laser with focusing optics and some mechanical means to move the laser spot on the device under test (DUT), e.g., via motorized stages. The only thing required to perform TLS with such a system would thus be a current preamplifier and a laser of suitable wavelength. Hence, it seems plausible that such a setup could be modified to perform TLS attacks at a low cost. However, it is unclear if the expected slower scanning speeds of motorized stages, as opposed to galvanometric mirrors usually used in FA solutions, might make attacks infeasible. Besides that, a drift in electronics and the mechanical system as well as a lower scan resolution might hinder an attack. An evaluation of the general possibility of developing such a setup thus seems to be beneficial. Consequently, this paper will evaluate if such a setup is generally feasible and what advantages and disadvantages it would bring for a potential attacker. This knowledge could then be used in the future to develop a more accurate TLS attacker model and thus better protected devices.

**Our Contribution** In this work, we demonstrate the feasibility of a low-cost TLS attack setup by retrofitting a commercial LFI setup with a suitable laser, amplifier, and software. For this, we only use commercially available components. We then evaluate two previously published attack types on the setup. The first one is the extraction of data from the SRAM of a microcontroller, as used for PUF characterization [1] and working memory data extraction [3]. For this attack, we showcase TLS scans of the whole memory area and also demonstrate that data can be extracted from the individual memory cells. The second evaluated attack is the readout of the decryption key from the BBRAM of an FPGA, as presented in [2]. We demonstrate that even with a low-cost setup, extraction of the full 256-bit AES key from the device is possible. Finally, we discuss and compare our results with the classical approach of using professional failure analysis equipment and highlight possible countermeasures.

## 2 Background

### 2.1 Thermal Laser Stimulation

Techniques from FA that use laser radiation to impact the DUT are referred to as laser stimulation techniques. Usually, the laser is scanned over the DUT while device parameters like the current consumption are monitored, grayscale-encoded and plotted over the scanning position (see Fig. 1). The resulting response map shows areas where laser radiation causes changes in the current consumption of the DUT. For TLS, the laser wavelength is chosen to have a photon energy smaller than the silicon bandgap, which consequently only causes local heating and no photocarrier generation.

When drain or source of a single metal-oxide semiconductor field-effect transistor (MOSFET) are thermally stimulated, effectively a voltage source between the corresponding metal contact and the channel is generated [1, 5]. This voltage source is also referred to as Seebeck generator, since it is caused by the Seebeck effect [6]. When the channel of the transistor is low ohmic, this generator is connected between drain and source. In contrast, when the channel is high ohmic, one connection of the generator is floating and the generated voltage is ineffective. The sign of the generated voltage depends on whether drain

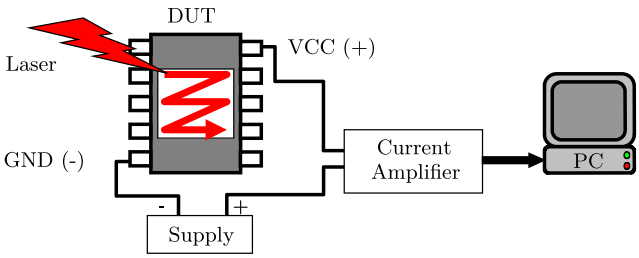



**Fig. 1** Scanning the laser over the DUT causes a change in current consumption due to thermal stimulation. Figure based on [4]

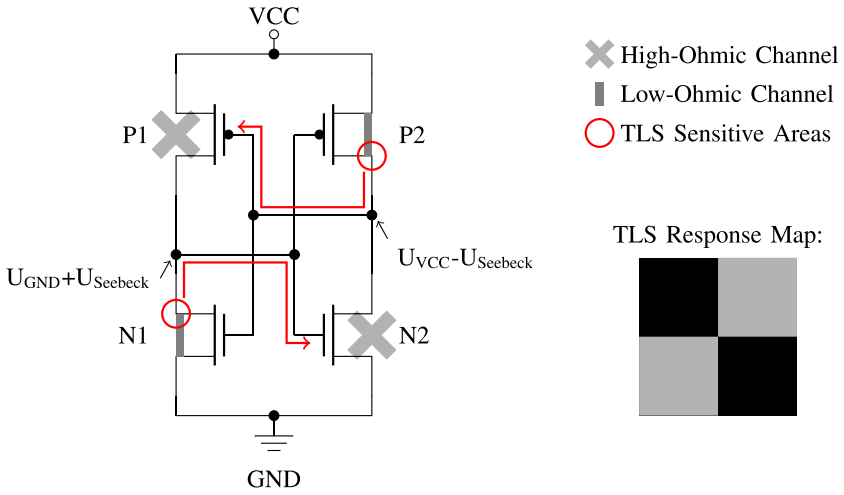



**Fig. 2** Memory cell under thermal stimulation. Figure based on [1]

or source are stimulated and on the type of the MOSFET (n- or p-type) [6].

A memory cell, as implemented in complementary metal-oxide semiconductor (CMOS) technology, basically consists of two cross-coupled inverters (see Fig. 2). The circuit stays in one of two stable states because of the cross-coupling. While being in such a stable state, for ideal transistors there is no current flow between VCC and GND, since one of the transistors in each connection from VCC to GND is high ohmic. However, under stimulation, the Seebeck generator causes the creation of a voltage ($U_{Seebeck}$), which is added to the existing voltage levels. When assuming 0 V as GND level and, for instance, the drain of transistor N1 is stimulated, effectively $U_{Seebeck}$ is applied to the gate of N2. Consequently, the resistance of N2 decreases via exponential sub-threshold operation, which in turn results in an increased current flow between VCC and GND. The same applies for transistor P1 when the drain of P2 is stimulated. The change in current consumption can be expected to be in the nanoampere range [1]. If a laser with a beam diameter approximately equal to the transistor size is scanned over the cell, a TLS response map as shown in Fig. 2 can be expected. Due to the increased current consumption, the sensitive transistors will be shown as brighter pixels. If the memory cell is in the inverted state, the other two transistors are sensitive. The cell's state can thus be deduced from the TLS response map.

Note that due to the chosen laser wavelength only thermal stimulation occurs, which can increase the leakage current of the memory cell but cannot change its state.

## 2.2 TLS for PUF Characterization and Data Extraction

The extraction of data stored in SRAM on microcontrollers can pose a threat to secrets stored within. For instance, the authors of [1] show that the extraction of data from SRAM memory on microcontrollers down to the 180-nm technology node is possible. More specifically, they demonstrate the characterization of a proof-of-concept SRAM-based PUF implementation on a microcontroller using TLS on professional FA equipment. Similarly, the authors of [3] show the potential to read out the whole working memory on a 180-nm technology microcontroller using TLS. It should be noted that for both attacks it was necessary to put the DUTs into a low-power mode, to reduce the noise of the system.

Such attacks on SRAM memory of microcontrollers are hereafter referred to as SRAM data extraction attacks.

## 2.3 TLS for Decryption Key Extraction

The authors of [2] demonstrate that the battery-backed random access memory (BBRAM) on a 20-nm technology field-programmable gate array (FPGA) can be read out with TLS using professional FA equipment. The BBRAM stores a 256-bit key used for bitstream decryption. To retain the key while the FPGA is powered off, the BBRAM is powered by a coin-cell battery. During the attack, the TLS signal is acquired by measuring the current consumption on this battery line. Since the BBRAM is the only circuit powered by the battery, the noise on this battery line is very low.

It should be noted that the attack was successful because the memory cell size is approximately 2.8 µm × 3.1 µm, which is about 10 times larger than the expected minimum size on a 20-nm technology device [2]. This can be explained by reliability, leakage, and low-current consumption considerations.

For their attack approach, the authors assume that the BBRAM is located close to the configuration logic. They consult the documentation to get an estimate of its location on the chip. By conducting a TLS scan over the candidate area, they can find the BBRAM. Afterward, they prove a data dependency in the measurements and create a mapping from memory cell locations to logical bits. Finally, they

show that a key stored in the BBRAM can be extracted using TLS in a manual or automated fashion within minutes.

Although the BBRAM is typically only battery-backed SRAM, this attack type is hereafter referred to as BBRAM key readout attack.

## 3 Setup

### 3.1 Laser Stimulation Setup

As the core of our setup, we use an ALPhANOV Single Laser Microscope Station (S-LMS), which was designed for laser fault injection purposes [7]. It is a microscope-based setup that allows the injection of different laser sources. In our case, we use a 1424-nm laser diode capable of delivering more than 300 mW in continuous waveform (CW) mode. The laser power can be controlled via PC software. The laser is focused through objectives, which are mounted on a manual turret, into the IC backside. For thermal stimulation, we use a 50×/0.65NA objective with silicon thickness correction, for optical images we additionally use 20×/0.5NA and 2.5×/0.1NA objectives. The whole microscope is mounted on XYZ motorized stages, which allow movements with a resolution of 50 nm. The stages are controlled via PC software or a joystick. The S-LMS is also equipped with infrared (IR) lighting and a short-wave infrared (SWIR) camera. This allows the user to monitor the laser spot position and perform optical navigation.

For measuring the current consumption during stimulation, we use a Stanford Research Systems ”SR570” current preamplifier, which has a bias voltage feature. The preamplifier outputs a voltage proportional to the current, which is digitized using a National Instruments ”PCI-6259” card.

To realize the scanning functionality and TLS response map creation, we developed a scanning software in the “LabView” programming environment from National Instruments. In this software, the scanning parameters, such as step size, step resolution, scanning speed, and number of samples per pixel, can be entered. The stage, and thus also the laser spot, is then moved continuously over the DUT while the preamplifier output is sampled. From this data, a TLS response map is created, in which higher current consumption of the DUT corresponds to brighter pixels. The whole setup is visualized in Fig. 3. Note that all results shown in this work have been achieved with this setup.

### 3.2 Devices Under Test

#### 3.2.1 DUT for SRAM Data Extraction

As mentioned in Section 2.2, reading out SRAM via TLS has been demonstrated down to the 180-nm technology node. Thus, a 180-nm Texas Instruments MSP430F5131 microcontroller is used in our experiments. It is equipped with 1 KB of SRAM with a cell size of approximately 2.5 × 1.9 µm [3]. For access to the silicon, the backside packaging material and the metal chip carrier were removed.

During the experiments, VCC of the DUT is supplied with 2.6 V via an auxiliary power supply. The core, which contains the SRAM, is supplied via the internally generated VCORE voltage, which is also available externally at a pin. To this pin, we connect the SR570 current preamplifier and set the bias voltage to 2.1 V, which is slightly above the VCORE voltage of 1.9 V. In this way, a significant amount of the core voltage is supplied by the SR570.

A JTAG debugging interface is connected to the device. This allows to directly write arbitrary data into the SRAM. For noise reduction on the VCORE net, the DUT is send to low-power mode 4 (LPM4) during the TLS scan.

For all experiments on the MSP430, the current amplification of the SR570 was set to 1 nA $V^{-1}$ and the input offset to 500 nA. The laser current was set to 600 mA, which corresponds to a total power of about 43 mW for the 50× lens. The silicon thickness correction of the objective was set to 350 µm.

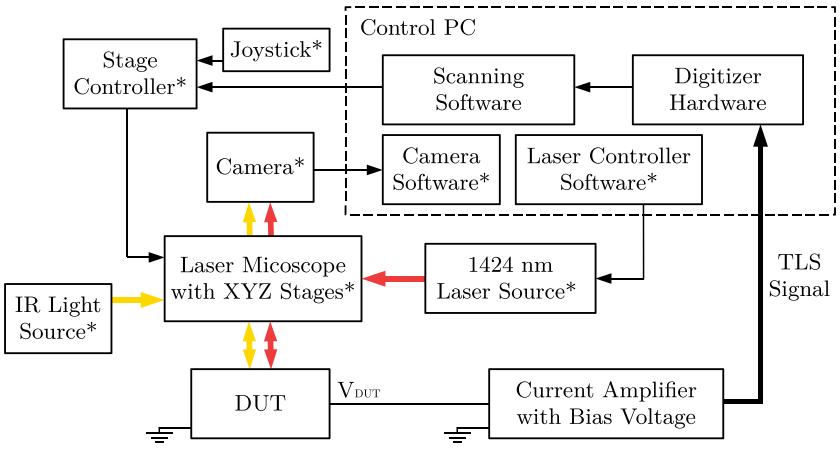



**Fig. 3** Block diagram of the setup. Components marked with an asterisk are part of the S-LMS [7]

#### 3.2.2 DUT for BBRAM Key Readout

The target platform for bitstream key extraction is a Xilinx Ultrascale FPGA development board from AVNET (model AES-KU040-DB-G). It contains a Xilinx Ultrascale XCKU040-1FBVA676 FPGA manufactured with 20-nm technology in a flip-chip ball grid array (BGA) package. Due to the flip-chip package, direct access to the silicon is available and no preparation is necessary. The thickness of the substrate is about 750 µm [2].

The 256-bit key used for bitstream decryption can be stored in a battery-backed RAM (BBRAM) which is programmed via a JTAG interface [8]. While the device is powered off, the BBRAM is supplied by a battery via the $V_{BATT}$ line. To measure the current consumption of the BBRAM during TLS, we soldered cables to the battery connector and connected them to the input of the SR570 current amplifier. During key programming, the board is powered via its external supply. During TLS experiments, however, the board is powered off and the $V_{BATT}$ voltage is supplied via the bias voltage feature of the SR570.

For all experiments on the Ultrascale FPGA, the current amplification of the SR570 was set to 2 nA $V^{-1}$ with no input offset. The laser current was set to 500 mA, which corresponds for the 50× lens to a total power of about 26 mW. The silicon thickness correction of the objective was set to 750 µm.

## 4 Measurement Results

### 4.1 SRAM Data Extraction

#### 4.1.1 SRAM Overview

To localize the SRAM optically, the camera of the setup was used (see Fig. 4). After zeroizing the whole SRAM via JTAG, the device is sent to low-power mode by code run from flash. A TLS scan with 0.5-µm scan step size was then acquired (see Fig. 5).

It can be seen that most of the memory shows a regular structure, except for some irregular vertical strips, mainly in the bottom left quadrant. Closer investigation revealed that this is data placed in SRAM by the code which enters the low-power mode. This already demonstrates that data dependencies can be observed. The SRAM seems to be subdivided into four blocks with a small offset of about one cell width in between, as already discovered in [3]. In addition, some cells seem to be more sensitive to TLS, as can be seen by some irregular bright spots. This can be explained by manufacturing variability. The TLS response becomes increasingly blurry in the right half of the scan, which can be explained by thermal and mechanical drift of the DUT due to the long scan duration of 43 min. This could potentially be avoided by scanning smaller areas and refocusing for each measurement.

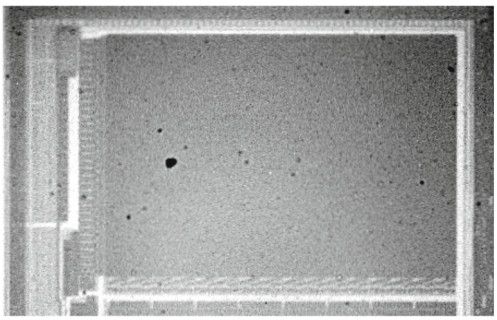


**Fig. 4** Optical image (20× lens) of the SRAM block

#### 4.1.2 Extraction of Single Bits

To demonstrate the extraction of single bits, we compare measurements of a small area of the SRAM (see Fig. 6). For the first measurement, the centered bit (framed in red) is set to `1`, while all other bits are `0`. For the second measurement, all bits, including the highlighted one in the center, are `0`. A pattern similar to the response map in Fig. 2 can be observed. For bit value `1`, the bottom left and top right of the cell are the most sensitive spots. In contrast, for bit value `0`, the sensitive spots are in the other corners of the cell. The subtraction of both response maps reveals the change more clearly.

These results show that the resolution of our setup is sufficient for extracting data from arbitrary SRAM cells on the MSP430 device. Consequently, an SRAM PUF implemented with a similar feature size could potentially be characterized with this setup. If the memory layout would

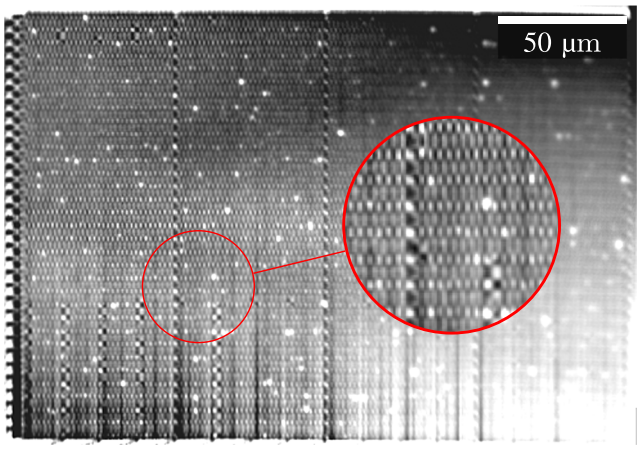



**Fig. 5** TLS overview scan of the SRAM. The scan direction is bottom-to-top (fast axis) and then left-to-right (slow axis)

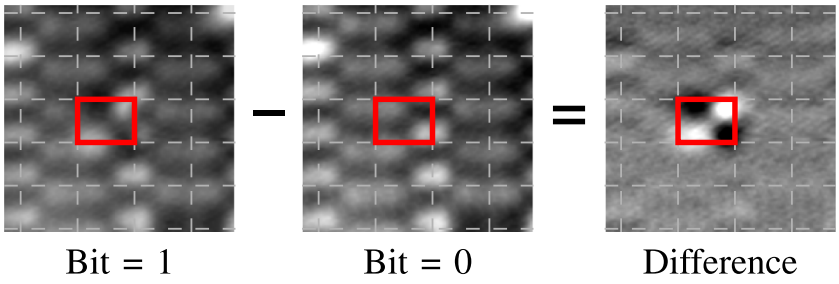



**Fig. 6** Data dependency of the measurements for a single SRAM bit which was first set to `1` and then to `0`, while all other bits in the area are set to `0`. The subtraction of both response maps reveals the change more clearly

be reverse-engineered using TLS, the full working memory of the MSP430 could be read out as well.

## 4.2 BBRAM Key Readout

### 4.2.1 Localization and Optical Overview

In the attack scenario of [2], the BBRAM was first localized inside the configuration area. For this, we performed a scan of that area with a pixel size of 5 µm and a stage speed of 2 mm $s^{-1}$ in about 5 min. The response map (see Fig. 7a), reveals two sensitive areas when the BBRAM is activated. If the BBRAM is deactivated, only one sensitive area remains (see Fig. 7b). Figure 8 shows an optical image of the area where TLS sensitivity occurred. The two highlighted block-like structures on the left correspond to the area where the TLS signal was dependent on BBRAM activation. These are the BBRAM block candidates which are already known from [2]. The structure on the right-hand side was always sensitive and thus can be disregarded. The results show that the BBRAM can be localized using our setup. In the next step, the detailed TLS response map has to be analyzed.

### 4.2.2 TLS Overview Scan

To observe data dependencies in the TLS response of the BBRAM, we first programmed a random key and an all-zeroes key and acquired TLS response maps (see Fig. 9a and b). It can be seen that different keys lead to different patterns in the response map.

To further investigate the key dependency of the TLS response, a single memory cell can be examined.

### 4.2.3 Extraction of a Single Bit

To identify a single bit in the TLS response map, we scanned a small area of the BBRAM with high resolution (pixel size 50 nm, stage speed 50 µm $s^{-1}$) with different bit values for one memory cell (see Fig. 10). While the sensitive spots for bit value `1` are on the top left and bottom right, the spots for value `0` are on the top right and bottom left of the cell (cf. Fig. 2). The subtraction of the two response maps clearly shows that the state and thus the bit value of the centered BBRAM cell differs in the two measurements.

Hence, this experiment proves that the optical resolution of our setup is sufficient for extracting the bit value stored in one BBRAM cell. The observed cell size is about 3.2 µm ×2.8 µm.

### 4.2.4 Key Extraction

Since we have already shown that data extraction of single bits from the BBRAM is possible with our setup and the mapping from physical to logical bit positions is known from [2], now a complete key can be extracted. For this, we subtract the response map of the all-zeroes key (Fig. 9b) from the response map of the random key (Fig. 9a). On the difference image (see Fig. 11), areas with large black and white spots correspond to bit value `1`, the others to value `0`. By adding a grid to optically show the SRAM cell size and position, the key bits can be easily extracted manually. Note that the top row is used to store security-relevant information, such as a configuration counter and error-detection bits [2]. The scan of the whole BBRAM for a pixel size of 250 nm and a stage speed of 50 µm $s^{-1}$ takes about 7 min.

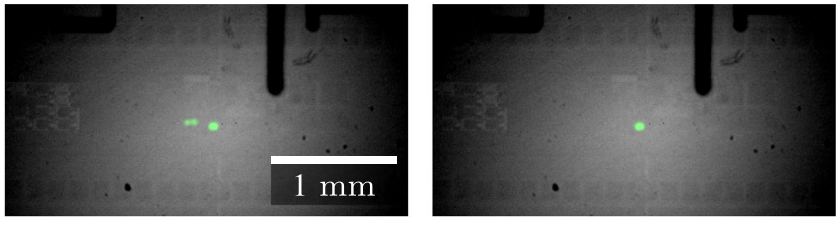



**Fig. 7** Localization scan with activated and deactivated BBRAM. The TLS signal is superimposed on an optical image (2.5× lens)

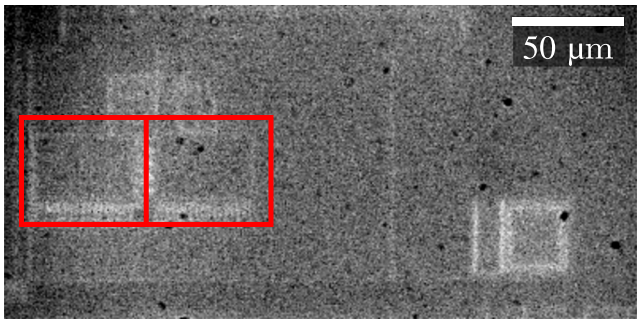



**Fig. 8** Optical image (20× lens) of the area sensitive to TLS with BBRAM candidate framed red

Given the above, the bitstream decryption key can be extracted from the BBRAM using our setup within minutes. This proves that the complete attack on the FPGA bitstream decryption key can be conducted with a much cheaper setup than previously expected.

## 5 Discussion

### 5.1 Low-Cost vs. FA Setup

#### 5.1.1 Acquisition Time

The experiments have shown that the time needed for acquisition is substantially longer when using the low-cost setup. This is due to the fact that for an FA laser scanning microscope (LSM) only small and light galvanometric mirrors are moved to scan the beam. In our case, though, the optical setup is moved on mechanical stages and the connected inertia poses a limit on the maximum scan speed. To give some exemplary numbers, for an FA LSM an acquisition time of 1.2 min can be expected for BBRAM key extraction [2]. With our setup, 7 min were needed. This is an increase by a factor of 5.8, but makes measurements due to the generally short duration still unproblematic.

For TLS SRAM data extraction, experiments performed by the authors have resulted in 4.8 min of acquisition time on an FA LSM. Using the low-cost setup, a complete SRAM scan on the MSP430 takes 43 min. This is an increase by a factor of 9. For such a long acquisition time, negative effects such as sample drift will lead to complications. This could already be observed in Fig. 5 (Section 4.1). It can thus be seen that with the approach demonstrated in this paper, attackers will have to trade cost for time. Additionally, procedures such as refocusing might be needed to prevent negative side effects, although these are relatively easy to implement.

For a low-cost approach, mechanical stages seem to be the obvious choice, since they are available in virtually any laboratory and microscope setup, as also in the used fault injection setup. However, it should be noted that galvanometric scan mirrors can be acquired at comparable prices to mechanical stages. Yet, the optical setup is more demanding, especially the requirements on the objective rise, since the field of view has to be sufficiently large.

#### 5.1.2 Resolution

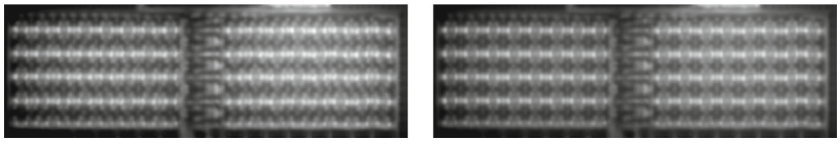
In terms of optical resolution, FA LSMs and the low-cost approach are virtually identical. This is due to the fact that the optical resolution is mainly determined by the wavelength and the numerical aperture of the objective lens. Using the same lens and wavelength should thus yield the same resolution in both setups. For the 50× lens and laser used in our setup, an optical resolution of about 1 µm can be expected.

For scan step resolution, the situation is different. The angular stepping resolution of an LSM's scanning mirror is translated by the objective lens into a spatial scanning resolution. This means that the scanning resolution can be increased by using larger magnification lenses. In contrast, the scan step resolution for the low-cost setup is simply the resolution of the stage — 50 nm in our case. Both the LSM's and the low-cost setup's scan resolutions are significantly lower than the optical resolution and can be expected to not be a limiting factor.

In general it can be said that our setup is not better or worse compared with an LSM in terms of resolution. However, it should be noted that FA LSMs can be equipped with a solid immersion lens (SIL), which can increase the resolution by a factor of around 4.3 in case of the Hamamatsu Phemos system [9].

#### 5.1.3 Cost

Our setup only consists of commercially available components. The core of the setup is a "S-LMS" station by



(a) Random key (b) All-zeroes key

**Fig. 9** TLS response maps of the whole BBRAM programmed with two different keys

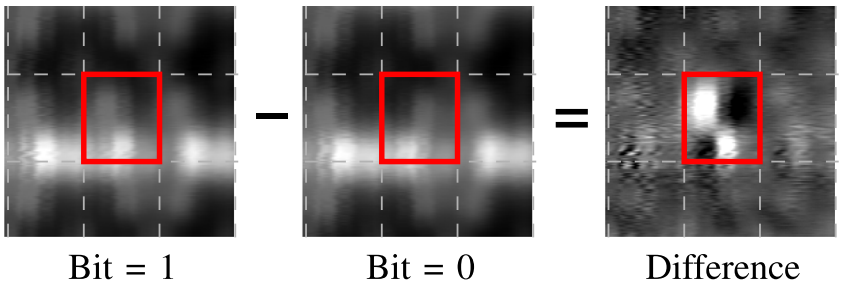



**Fig. 10** Data dependency of the measurements for a single BBRAM bit which was first set to `1` and then to `0`, while all other bits in the area were `0`. The subtraction of both response maps highlights the change more clearly

ALPhANOV. In a configuration suitable for retrofitting TLS, the system costs about 102k USD, including a 1.4 µm laser. Additionally, the SR570 current preamplifier for 2595 USD [10] and the NI-6259 digitizer card for 1940 USD [11] have to be acquired. Including the control PC and a LabView license, the price of the complete setup is expected to be below 110k USD.

Compared with a Phemos-1000 Failure analysis setup with a price between 500k and 1M USD, our setup is five to ten times cheaper. Furthermore, the setup can in principle be set up on a single desk and purchasing of the equipment is expected to require less effort.

### 5.2 Attack Feasibility and Limitations

The feasibility of TLS attacks on SRAM in general depends on the spatial distance between the sensitive transistors. Consequently, the limiting factor is the laser spot size. The minimum possible spot diameter is about 1 µm without SIL and 235 nm with a recent SIL, which corresponds to cell dimensions of 2 µm and 470 nm, respectively. Thus, the attack is expected to work at least down to these cell sizes.

Furthermore, it can be expected that with post-processing of the TLS signal, for instance, by deconvolution, an additional resolution enhancement by a factor of 2 is possible. However, to the best of our knowledge, the actual SRAM cell size limit for TLS attacks is unknown and should be subject of future research. It should also be noted that with the switch to new technologies, like FinFET, changes in the behavior of the stimulated cells might occur.

Next to the cell size, the attack is also limited by the amount of noise present in the TLS signal. Specifically, the leakage current of the transistor affected from stimulation should be higher than the fluctuations in the overall current consumption. In our experiments, this was fulfilled by the low noise on the battery line of the FPGA, and by sending the microcontroller to low-power mode.

Readers interested in more details regarding the attack feasibility are directed to [2].

### 5.3 Countermeasures

The possible countermeasures against TLS attacks from the chip backside can be divided into two categories. On the one hand, techniques could be applied to obstruct the access to the chip or the measurement signal, and on the other hand, active attack detection mechanisms could be employed.

An approach for the former class could be to reduce the resolution of the laser beam by scrambling the incoming light from the chip backside, and thus increasing the beam diameter within the silicon. In [12], this approach is applied against optical contactless probing. The authors introduce the usage of nanopyramid structures, which scramble the reflected light. However, adding the

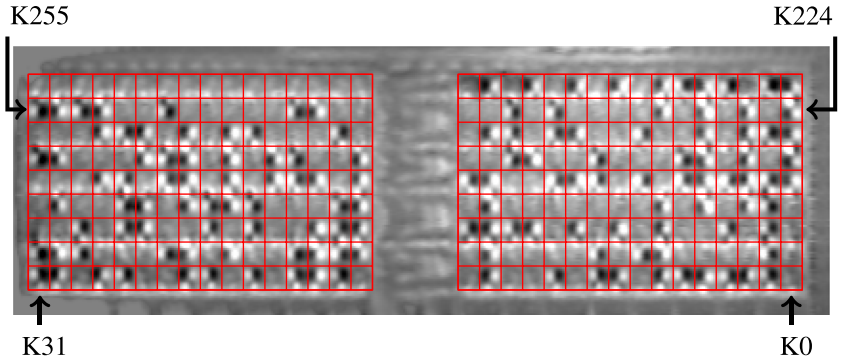



**Fig. 11** Subtraction of the TLS response maps of the random key and the all-zeroes key. The existence of black and white patterns in one cell corresponds to bit value `1`. Bit position K255 corresponds to the most significant bit of the key. The key programmed into the BBRAM is: `0xf20c28551d626c97c75932351b5dcebf-4de340562ca7f54ae34f42c2d9ae4b7e`

nanopyramids between chip backside and the transistors is only possible for bonding-based SOI devices. Furthermore, the countermeasure was not tested with respect to thermal stimulation. Yet, it might be an interesting approach for further research.

Another approach for the first category of countermeasures could address the destruction of the data dependency in the TLS signal. Since the attack relies on low noise in the current consumption of the target device, noise injection can be used for this purpose. In [13], a noise source has been successfully designed and integrated to protect an encryption core from power analysis attacks. This shows that on-chip noise-based mitigation techniques can work. Against TLS attacks on SRAM, a proof of concept countermeasure was presented in [2]. By injecting noise on the battery line of a BBRAM key storage, TLS data extraction can be made much harder or possibly even unfeasible. The authors show that this can be an effective mitigation technique, even with negligibly lower battery life time.

A more thorough approach is the protection of the chip backside by employing an opaque coating layer to obstruct optical access completely. However, a solely passive layer could be easily removed by polishing. As evaluated in [14], the integrity of the coating layer can be assured by in-silicon light emitters and sensors. This combines an obstruction approach (first category) with an active detection countermeasure (second category). Yet, due to the high power consumption of the photo sensors, this protection scheme can not protect devices with a very restricted power profile, such as the BBRAM key storage.

To actively detect the temperature changes induced by the laser radiation, temperature sensors could be useful [2]. However, it is questionable whether the very small temperature changes with a laser power of less than 50 mW can be detected with a small amount of false positives. Furthermore, the current consumption of temperature sensitive circuits, such as ring oscillators, is typically high [15]. Hence, for power-constrained devices like the BBRAM, this does not seem to be a feasible solution.

## 6 Conclusion

In this work, we have shown that constructing a low-cost setup for TLS is indeed feasible. By retrofitting an LFI setup with the necessary equipment, we have demonstrated a solution for TLS five to ten times cheaper than traditional FA equipment. Although with slower signal acquisition, we were still able to show that two state-of-the-art attacks, specifically against SRAM on a microcontroller and BBRAM on an FPGA, are possible in reasonable time. Consequently, the attacker model must be rethought and adapted to better reflect the lower-than-expected hurdle for an attacker to apply TLS. Therefore, better protection mechanisms against attacks from the chip backside will have to be deployed.

**Acknowledgments** The authors would like to thank ALPhANOV for providing a Single Laser Microscope Station (S-LMS) for the setup.

## References

1. Nedospasov D, Seifert JP, Helfmeier C, Boit C (2013) Invasive PUF analysis. In: 2013 Workshop on fault diagnosis and tolerance in cryptography (FDTC). IEEE, pp 30–38
2. Lohrke H, Tajik S, Krachenfels T, Boit C, Seifert JP (2018) Key Extraction Using Thermal Laser Stimulation. IACR Transactions on Cryptographic Hardware and Embedded Systems, pp 573–595
3. Kiyan T, Lohrke H, Boit C (2018) Comparative Assessment of Optical Techniques for Semi-Invasive SRAM Data Read-out on an MSP430 Microcontroller. In: ISTFA 2018: Proceedings from the 44th International Symposium for Testing and Failure Analysis. ASM International, pp 266–271
4. Beaudoin F, Desplats R, Perdu P, Lewis D (2001) Implementing Thermal Laser Stimulation in a Failure Analysis Laboratory. In: ISTFA 2001: Proceedings of the 27th International Symposium for Testing and Failure Analysis. ASM International, pp 151–160
5. Boit C, Helfmeier C, Nedospasov D, Fox A (2013) Ultra high precision circuit diagnosis through seebeck generation and charge monitoring. In: Proceedings of the 20th IEEE International Symposium on the Physical and Failure Analysis of Integrated Circuits (IPFA). IEEE, pp 17–21
6. Geballe TH, Hull GW (1955) Seebeck Effect in Silicon. Phys Rev 98(4):940–947
7. ALPhANOV (2019) Optical and Laser System for Fault Injection for IC Evaluation S-LMS: Single Laser Microscope Station. http://www.alphanov.com/40-optoelectronics-systems-and-microscopy-single-spot-laser-station.html, accessed 15 Apr 2019
8. Wilkinson K (2018) Using Encryption and Authentication to Secure an UltraScale/UltraScale+ FPGA Bitstream. https://www.xilinx.com/support/documentation/application_notes/xapp1267-encryp-efuse-program.pdf, accessed 02 Apr 2019
9. Hamamatsu Photonics KK (2015) NanoLens-SHR. https://www.hamamatsu.com/resources/pdf/sys/SSMS0053E_Nanolens-SHR.pdf, accessed 18 Apr 2019
10. Stanford Research Systems (2014) Low-Noise Current Preamplifier. https://www.thinksrs.com/downloads/pdfs/catalog/SR570c.pdf, accessed 02 Apr 2019
11. National Instruments (2019) NI PCI-6259. http://sine.ni.com/nips/cds/view/p/lang/en/nid/14128, accessed 16 Apr 2019
12. Shen H, Asadizanjani N, Forte D, Tehranipoor M (2018) Nanopyramid: An Optical Scrambler Against Backside Probing Attacks. In: ISTFA 2018: Proceedings from the 44th International Symposium for Testing and Failure Analysis. ASM International
13. Das D, Maity S, Nasir SB, Ghosh S, Raychowdhury A, Sen S (2017) High Efficiency Power Side-Channel Attack Immunity Using Noise Injection in Attenuated Signature Domain. In: 2017

IEEE International Symposium on Hardware Oriented Security and Trust (HOST), pp 62–67
14. Amini E, Beyreuther A, Herfurth N, Steigert A, Szyszka B, Boit C (2018) Assessment of a Chip Backside Protection. J Hardw Syst Secur 2(4):345–352
15. Tajik S, Fietkau J, Lohrke H, Seifert JP, Boit C (2017) PUFMon: Security Monitoring of FPGAs Using Physically Unclonable Functions. In: 2017 IEEE 23rd International Symposium on On-Line Testing and Robust System Design (IOLTS). IEEE, pp 186–191

# 6

# HARDWARE TROJAN DETECTION USING LLSI

## 6.1 MALICIOUS MODIFICATIONS OF HARDWARE

News stories like "The Big Hack" [54], claiming that state actors undermined the U.S. technology supply chain, have shaken up governments worldwide. In the story, the authors present how a small chip, secretly added to a PCB, can act as a stealthy backdoor. Although this story created many rumors and its authenticity was doubted, the threat of malicious modifications of ICs and PCBs is real. With DARPA's Trusted Integrated Circuits program, the U.S. military acknowledged the importance of trustworthy ICs more than fifteen years ago [55]. Due to the globalized value chain of semiconductor design, devices predominantly rely on third-party designs and are manufactured in foreign factories, potentially controlled by governments. Therefore, a malicious actor can slightly alter the design through harmful modifications such as inserting a backdoor, adding logic to leak sensitive information, or inserting a kill switch [56].

*Stealthy Trojans and detection approaches*

The presumably most stealthy way to insert such so-called hardware Trojans (HTs) is to embed them into the IC with the Trojan being inactive by default. Only a minimal trigger logic is active and waits for a condition to enable the Trojan payload. In such a scenario, electrical testing cannot detect the Trojan. Side-channel analysis, such as power, EM, or photonic emission analysis, might also not be able to detect the small trigger logic. Even if the HT leaves a trace in side-channel data, one needs some reference to compare the traces to, called a golden chip. Reverse engineering the device and comparing the results to the design plans can help to detect modifications. However, while non-destructive techniques exist on the PCB level [57–59], ASICs have to be delayered for precise analyses [60]. Therefore, the chip will be destroyed during analysis. Non-destructive methods that do not require a golden chip were proposed in the literature [61]. However, these approaches can detect dormant HTs of small sizes only with a high rate of false positives. Therefore, a non-destructive technique that can detect tiny modifications of the IC would be highly beneficial.

*Optical probing-based detection*

Optical laser-based FA techniques offer high resolutions while being non-destructive. One approach that uses optical probing, more specifically EOFM, was already presented in [62]. However, this approach is limited to detecting HTs that modify the sequential logic of the design. Any changes in the combinatorial logic stay undetected. Consequently, one might ask if laser-based logic state imaging techniques can fill this gap.

*Research question*

All other publications of this thesis focused on evaluating how a potential adversary might be able to extract secrets from the IC. Nevertheless, optical logic state imaging techniques can also be used for protecting devices and implementations against attacks. Regarding HTs, we started investigating how we can use LLSI to detect malicious modifications of hardware implementations. We decided to focus on re-programmable hardware, as their implementation is especially vulnerable to malicious alterations because of the inbuilt re-programmability. An insufficiently protected interface for programming an FPGA allows an adversary to change the hardware configuration at any time before or during usage of the device, as shown in several studies [6, 7, 63]. In the subsequent publication, we investigate how HTs inserted into the combinatorial and sequential logic on recent FPGAs can be detected.

## 6.2 PUBLICATION

The work was originally presented at the *5th Workshop on Attacks and Solutions in Hardware Security (ASHES 2021)* [64]. The version reprinted here is an extended version that contains, among other additions, new experimental results on a 20 nm FPGA. It was published in the *Journal of Cryptographic Engineering (JCEN)* [65].

# Trojan awakener: detecting dormant malicious hardware using laser logic state imaging (extended version)

**Thilo Krachenfels**[1] · **Jean-Pierre Seifert**[1,2] · **Shahin Tajik**[3]

Received: 13 May 2022 / Accepted: 26 April 2023


**Abstract**
The threat of (HTs) and their detection is a widely studied field. While the effort for inserting a Trojan into an (ASIC) can be considered relatively high, especially when trusting the chip manufacturer, programmable hardware is vulnerable to Trojan insertion even after the product has been shipped or during usage. At the same time, detecting dormant HTs with small or zero-overhead triggers and payloads on these platforms is still a challenging task, as the Trojan might not get activated during the chip verification using logical testing or physical measurements. In this work, we present a novel Trojan detection approach based on a technique known from (IC) failure analysis, capable of detecting virtually all classes of dormant Trojans. Using (LLSI), we show how supply voltage modulations can awaken inactive Trojans, making them detectable using laser voltage imaging techniques. Therefore, our technique does not require triggering the Trojan. To support our claims, we present three case studies on 28 nm and 20 nm SRAM- and flash-based (FPGAs). We demonstrate how to detect with high confidence small changes in sequential and combinatorial logic as well as in the routing configuration of FPGAs in a non-invasive manner. Finally, we discuss the practical applicability of our approach on dormant analog Trojans in ASICs.

**Keywords** Hardware security · Hardware Trojans · Optical side-channels · Hardware snapshots · LLSI

## 1 Introduction

Due to their reconfigurability, high performance, and a short time to market, programmable hardware, especially (FPGAs), has become the default solution in many fields. One of the main strengths of FPGAs compared with (ASICs) is that the hardware configuration can be updated and even reprogrammed during runtime. At the same time, the demand for security increases as more and more security-critical systems are based on electronics. Therefore, malicious modifications of the design, referred to as (HTs), endanger the security of many applications. On FPGAs, a Trojan might be inserted after manufacturing and testing, i.e., in the untrusted field [1, 2], for instance, by altering the entire configuration (known as bitstream) or via partial reconfiguration. Particularly if the chip foundry can be trusted, this depicts a much more powerful threat model than for ASICs. Most security-critical FPGAs rely on bitstream encryption and authentication to avoid such Trojan insertions. However, these protection schemes have shown to be vulnerable to various physical [3–6] and mathematical attacks [7], leaving them susceptible to tampering. Consequently, in critical applications, where the chip is deployed in an untrusted field or could be accessed by untrusted parties, it should be possible to check the integrity of the hardware.

Integrity checking of running applications on FPGAs in the field faces mainly two obstacles. First, while checking the configuration against a golden bitstream would reveal tampering (as proposed in [8]), it is not possible in many cases. In several defense/aerospace applications, where flash-based FPGAs [9] or SRAM-based FPGAs with preemptive decryption key zeroization [10] are deployed, no bitstream (encrypted or unencrypted) is available to the hardware testing engineer in the field for verification. In these cases, the configuration is stored inside the chip and bitstream readback

Thilo Krachenfels
tkrachenfels@sect.tu-berlin.de

Jean-Pierre Seifert
jpseifert@sect.tu-berlin.de

Shahin Tajik
stajik@wpi.edu

1 Chair for Security in Telecommunications, Technische Universität Berlin, Berlin, Germany

2 Fraunhofer SIT, Darmstadt, Germany

3 Worcester Polytechnic Institute, Worcester, USA

Reprinted from the original article published in the Journal of Cryptographic Engineering (JCEN), https://doi.org/10.1007/s13389-023-00323-3.

is not possible. Even if the bitstream is available, analyzing the unencrypted bitstream is not an option since the circuit and the secret keys for bitstream decryption should be unknown even to the testing engineer. Moreover, the same bitstream can be encrypted with various keys for different FPGAs, and therefore, comparing encrypted bitstreams to each other for tampering detection might also not be feasible.

Second, while early HTs had logic triggers that could be activated by logical testing [11] under some circumstances, recently proposed HTs are classified as *stealthy* or *dormant*. In other words, the Trojan payload reacts only under extremely rare conditions, for instance, in a particular temperature, supply voltage, or frequency range [12] or after a certain amount of specific events have occurred [13]. Furthermore, under operational and testing conditions, a dormant Trojan tries to hide from physical inspection or side-channel analysis, e.g., by leveraging analog components [13], manipulating only the dopant level of the chip [14], or changing only the routing configuration on programmable hardware [12].

Several approaches based on side-channel analysis (SCA) for detecting such dormant HTs have been proposed in the literature [15–22]. However, they all face severe limitations regarding resolution and the capability to detect all types of HTs. For instance, approaches using electromagnetic (EM) backscattering side-channels are naturally limited by their resolution and can only detect larger malicious design changes [18, 19]. Furthermore, these approaches can reliably detect dormant Trojans only with a high rate of false positives. One technique that provides higher resolution is optical probing, where the chip is scanned through its backside with a laser, and the reflected light is analyzed. However, the reported approach based on electro-optical frequency mapping (EOFM) [21] is limited to detecting malicious modifications only in the sequential logic, and thus, Trojans that solely consist of combinatorial logic stay undetected.

A new optical probing technique that has recently been leveraged in the hardware security field is called *laser logic state imaging (LLSI)* [23]. It is an optical probing technique that can extract the logic states of single transistors, and therefore, more complex logic gates or memory cells [24]. In LLSI, the chip's supply voltage is modulated, which causes the light reflection originating from a laser scanning irradiation to be modulated as well. The modulation amplitude is dependent on the carrier concentration present in the silicon, for instance, inside the channel of a transistor. Consequently, the LLSI signal is highly data-dependent and provides a practically unlimited number of electro-optical probes. Hence, it should be possible to extract the configuration of an FPGA's logic fabric using LLSI, especially because the configuration is held in memory cells distributed over the chip. The logic state of these cells controls the functioning of (LUTs), multiplexers (MUXes), and pass transistors in switch boxes. In this work, we try to clarify *if small dormant HTs on state-of-the-art FPGAs—consisting of combinatorial or sequential logic—can be detected by applying LLSI.*

*Our contribution* We indeed positively answer the above question. First, we present how LLSI allows us to capture the state of every transistor of the logic fabrics of SRAM- and flash-based FPGAs. Based on this, we demonstrate how to partially reverse-engineer the FPGA's configuration, including the detection of changes in a single LUT. Second, we show how this new approach can detect small and dormant HTs on FPGAs. Stimulating all transistors with the power supply modulation awakens maliciously modified hardware, from which we then can take a snapshot. Therefore, the Trojan can be inactive/dormant, as our approach does not rely on any switching activity on the chip. For detecting HTs, we first capture a reference snapshot of the FPGA's logic fabric in the trusted field—when the design is known to be Trojan-free. Later, to check if the design has been altered, we capture a snapshot of the logic fabric and compare it to the reference. We show that the high resolution of optical probing allows detecting small changes of the configuration, down to changes in a single combinatorial gate.

Our approach can be applied non-invasively since almost all current FPGAs are available in flip-chip packages allowing easy access to the silicon backside. To validate our claims, we present three case studies on SRAM- and flash-based FPGAs from Xilinx (28 nm and 20 nm technology) and Microchip (28 nm technology), respectively. Although our experiments are focused on FPGAs, we discuss why LLSI is applicable for analog HT detection on ASICs.

*Remarks on the extended version* The original version of this work has been presented in [25]. The version at hand contains the following additional and revised content: (i) the investigation of a new target device manufactured in a 20 nm technology, including setup, results, and discussions; (ii) a more thorough explanation and discussion of the experimental setup, especially regarding the LLSI modulation frequencies; (iii) a detailed discussion of how to prepare a real-world device that should be investigated using the presented HT detection approach; and (iv) additional figures depicting the experimental setup.

## 2 Background

### 2.1 Hardware Trojans

#### 2.1.1 Properties and taxonomy

The term hardware Trojan (HT) includes a wide range of malicious circuit modifications which, for instance, try to

leak sensitive information through side-channels, implement kill-switches and backdoors, or enforce faulty computations. HTs can be characterized by their physical properties (e.g., type and size of modifications), activation characteristics (i.e., trigger source and frequency), and action characteristics (i.e., which goal the HT serves) [26]. As diverse as the different types of HTs are, so are the potential entities that might introduce the malicious modifications [27]. During the development and production of (ICs), weak points include third-party intellectual property (IP) cores, malicious design tools, and mask layout or doping concentration modifications [28] by untrusted foundries. The platform TrustHub [29] provides several design-level HT benchmarks, primarily available as gate-level descriptions. TrustHub provides access to the automatically generated HT benchmarks presented in [30] that alter existing circuit designs by inserting malicious logic gates.

Programmable hardware devices, like FPGAs, are less prone to production-based HT insertion than ASICs. On the other hand, due to their reconfigurability, they provide the possibility for malicious modifications even after the product has been shipped to the user. It has been shown that the key used for encrypting the bitstream on recent SRAM-based FPGAs can be extracted using SCA techniques [3–6]. With the extracted key at hand, the bitstream can be decrypted, modified, and stored as a replacement for the original bitstream [17]. Although bitstream extraction from flash-based FPGAs might not be possible, the adversary could still be able to reprogram certain parts of the configuration or even replace the entire chip containing her malicious version of the design.

### 2.1.2 Hardware Trojans on FPGAs

While generic Trojans, such as backdoors, can be implemented on both ASICs and programmable hardware, a few HTs especially tailored to FPGAs have been proposed. For instance, Jacob et al. have proposed an approach that exploits shared resources between the programmable logic and the embedded microcontroller on an FPGAs system on a chip (SoC) [31, 32]. By hidden functionalities in an IP design block, the programmable logic can access and manipulate shared memory locations used for storing sensitive information like cryptographic keys. Ender et al. have proposed a Trojan that is solely based on minor timing modifications on the chip [12]. They show that by operating the chip with modified signal paths at a specific frequency, the data masking scheme protecting against side-channel analysis attacks is not functional anymore, allowing the extraction of the secret key used in the protected algorithm. They show that on an FPGA, longer signal paths can be realized by instantiating route-thru LUTs, or by modifying the routing in the switch boxes, which results in zero overhead in resource usage, and therefore, is hard to detect. In another effort, Roy et al. [2] showed that the reconfigurable LUTs could be exploited to realize HTs with zero payload overheads. Finally, Ng et al. [1] demonstrated that integrated sensors inside FPGAs could be deployed as Trojan triggers.

### 2.1.3 Detection of hardware Trojans—related work

As already mentioned in Sect. 1, HT detection on FPGAs cannot always be carried out by checking or comparing bitstreams. Therefore, most of the HT detection techniques use different kinds of physical measurements and side-channel information obtained from the chip. Optical chip backside reflectance imaging [22], scanning electron microscopy (SEM) imaging [33], or focused ion beam (FIB) imaging [34] are not suitable for detecting HTs on FPGAs, because the physical design and layout of the chip do not depend on the actual programmed functionality. SCA techniques, such as power analysis, EM analysis [20], or backscattering analysis [18, 19], can be used for all types of ICs. By applying different clustering algorithms, the Trojan-infected chips can be separated from the non-infected chips, often without the need of a golden chip, i.e., a chip which is known to be Trojan-free. However, these techniques only offer a limited resolution, which requires the Trojan trigger logic to consist of a minimum number of gates or being separated from its input signals to a certain extent [18]. Furthermore, the clustering does only work if the set of samples contains at least one non-infected device.

SCA techniques offering higher resolution include approaches that observe the chip's operation through the silicon backside, which is transparent to near-infrared (NIR) light. For instance, photon emission (PE) analysis can be used to compare dynamic and static emissions with the chip layout [16] or emissions from a golden chip [15]. Furthermore, adding oscillators with inputs from the design that act as beacons can facilitate the detection of tampering attempts, especially when cheaper infrared imaging is used [17]. However, such an approach increases the resource consumption of the design considerably in many cases and might not be able to detect all possible changes in LUT configurations. One approach providing higher resolution and better localization capabilities is optical probing. The authors of [21] have demonstrated that using an optical probing technique, all (FFs) used in the hardware design can be located and mapped to the intended design from the FPGA's integrated development environment (IDE). In this way, malicious changes in the sequential logic can be detected reliably and in a non-invasive fashion, if the chip is packaged as flip-chip. However, combinatorial logic cannot be detected using that approach, which is the major downside of the approach.

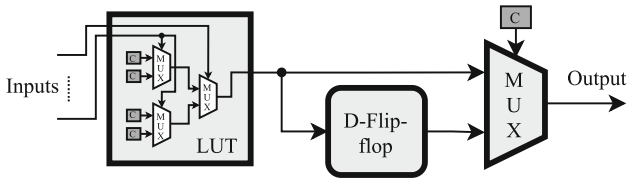



**Fig. 1** Simplified schematic of an FPGA logic block. LUTs and MUXes are controlled by configuration memory cells

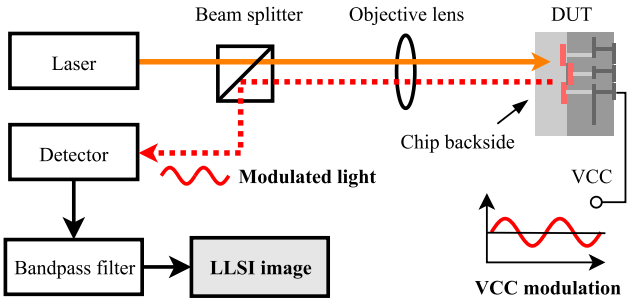



**Fig. 2** Schematic of LLSI image acquisition. The DUT is scanned with a laser through the chip backside; due to a power supply (VCC) modulation, the reflected light is modulated, which can be detected

## 2.2 Field-programmable gate arrays (FPGAs)

The heart of an FPGA is its configurable logic fabric, consisting of an array of small configurable logic elements containing lookup tables (LUTs) and flip-flops (FFs) for implementing combinatorial and sequential logic, respectively. Configurable routing resources interconnect these blocks. Together with on-chip memories and input/output capabilities, such as transceivers, the designer can implement virtually every functionality on the FPGA. To add the software configurability of processors to FPGAs, vendors offer soft processor cores, and recently even SoCs containing both ASIC processors and an FPGA logic fabric, connected by an effective interconnection network.

Although the logic fabric architecture differs between manufacturers, the building blocks are multi-input LUTs for combinatorial logic, FFs for sequential logic, and MUXes for signal routing, see Fig. 1. The two main configuration storage types for FPGAs are volatile SRAM-based and non-volatile flash-based memories.

### 2.2.1 SRAM-based

The dominating manufacturers for FPGAs are Xilinx (acquired by AMD) and Intel (formerly Altera), with a combined share of more than 85% [35]. Both of them focus on SRAM-based FPGAs. The advantage of using SRAM as memory technology is that the chip can be manufactured with cutting-edge chip technologies, which allows for higher logic densities. Due to the volatile nature of SRAM cells, the FPGA's configuration is lost after every power-down. Therefore, the configuration (the bitstream) must be stored in external memory and loaded upon every reboot by the FPGA's configuration fabric. This fabric decrypts the configuration and loads it into the distributed SRAM cells on the chip, which determine the behavior of LUTs, MUXes, and routing transistors. One advantage of the volatile configuration storage is the possibility to partially reconfigure the logic fabric during runtime.

### 2.2.2 Flash-based

Flash-based FPGAs are offered mainly by Microchip (formerly Microsemi) and Lattice Semiconductor, with a combined market share smaller than 12% [35]. The main advantage of flash-based FPGAs over SRAM-based FPGAs is their lower power consumption. Further, the configuration is stored in a non-volatile way in distributed flash cells. One reason for the lower power consumption is that flash cells consist of fewer transistors than SRAM cells and do not need to be powered for retaining their value.

## 2.3 Laser-based logic readout

### 2.3.1 Technique

Optical probing is a powerful approach known from IC failure analysis (FA). A laser is pointed on the chip's backside, and switching activity causes the reflected laser light to modulate. More specifically, mainly the concentration of free carriers distinguishes the refraction and absorption of the laser light in silicon. When the laser scans the device and the reflected signal is fed through a bandpass filter set to a frequency of interest, all areas on the chip switching at a frequency of interest can be detected. The corresponding technique is called electro-optical frequency mapping (EOFM) or laser voltage imaging (LVI).

Using classical EOFM, only periodically switching elements on the chip can be detected. The static logic state of circuits, however, can be captured using laser logic state imaging (LLSI), which was introduced as an extension to EOFM [24]. The main idea behind LLSI is to stop the clock and induce a periodic frequency into the entire logic by modulating the power supply, see Fig. 2. This causes the free carrier concentrations to vary periodically, e.g., in the channel of transistors or in capacitors. This, in turn, modulates the reflected light, which can be detected using EOFM. Transistors that are switched on (low-ohmic channel) can thus be

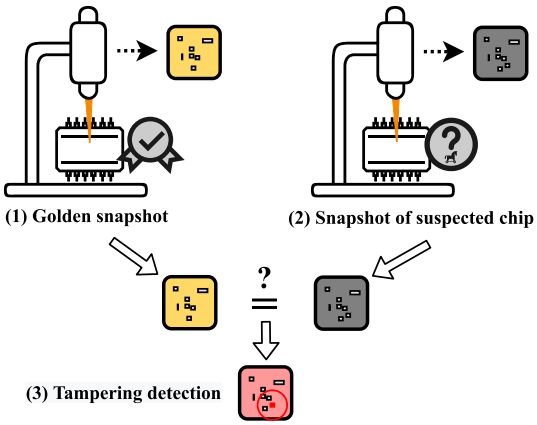



**Fig. 3** Approach for detecting tampering with the FPGA logic fabric configuration

distinguished from transistors that are switched off (high-ohmic channel).

### 2.3.2 Related work

LLSI has been used in the hardware security field to extract the values stored in SRAM cells or FFs. The authors of [23] demonstrated that the FF content of an FPGA manufactured in a 60 nm technology can be extracted using LLSI. Using classical image recognition techniques, they show that the content can be extracted in an automated fashion. In [36], the authors demonstrate that a key stored in the SRAM of a microcontroller can be extracted using LLSI combined with deep learning techniques without the need to reverse-engineer the chip's layout. To the best of our knowledge, LLSI has neither been used to extract an FPGA's logic fabric configuration nor to detect HTs.

## 3 Approach

In our scenario, the supply chain from the finished product to the field cannot be trusted. In other words, an adversary might replace or change the device's functionality after it has left the trusted design house. In such a scenario, the highest efforts are paid to detect malicious hardware, e.g., in military, space, and aircraft applications. Although LLSI can capture the states of transistors and memory cells in all ICs, our goal in this work is to apply LLSI for creating snapshots of the logic fabric in FPGAs. To do so, we need to modulate the supply voltage of the logic under test, in our case, of the logic fabric, see Sect. 2.3. Furthermore, we need to halt the clock of the FPGA. To test if the FPGA's configuration manifests in the hardware snapshots, we configure the logic fabric in different ways, for instance, by altering the configuration of LUTs and the routing. We then compare the snapshot images to see if the changed configuration can be detected and at which location the change has occurred.

Once different configuration changes can be detected, the knowledge can be used to also detect malicious modifications on the chip, see Fig. 3. In our approach, we create a snapshot of the original Trojan-free design, also known as golden design, in the trusted design house (1). It typically will be necessary to create multiple snapshots to cover the entire logic fabric area with high resolution. We then assume a malicious entity that inserts a Trojan into the FPGA configuration of the product. Before using the final product in a security-critical application, the integrity of the IC should be certified. For this, we create a snapshot of the suspected chip (2). To eliminate the chance of any tampering, we compare the golden snapshot with the current snapshot (3). For comparing the snapshots, subtracting the images might be helpful. If there are differences, this indicates that the configuration has been altered, and the chip is not trustworthy. It should be noted that the state of the FPGA in step (1) and (2) should be the same, i.e., the clock should be stopped in the same cycle. We expect our approach to work on both SRAM- and flash-based FPGAs.

*SRAM-based FPGAs* SRAM-based FPGA configuration takes place by configuring LUTs and global/local routing via SRAM cells. In the end, all configuration SRAM cells do control MUXes, which consist of pass transistors. Since LLSI can extract the logic states of CMOS transistors, the FPGA's entire configuration should be extractable—given a sufficiently high optical resolution.

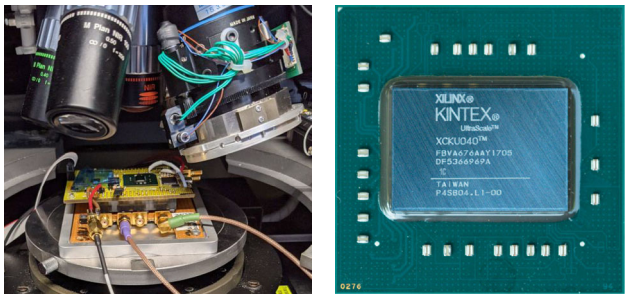
*Flash-based FPGAs* The configuration of flash-based FPGAs is stored in dedicated flash cells, which are distributed over the chip. They control the LUTs and global/local routing using multiplexers, which, like in SRAM-based FPGAs, consist of pass transistors. Therefore, also the configuration of flash-based FPGAs should be extractable using LLSI. If the flash cells are supplied by another voltage rail, it might be possible to see a configuration dependency by modulating that rail.

## 4 Experimental setup

This section first presents our measurement setup, followed by the devices under test (DUTs) and their setup for conducting LLSI.

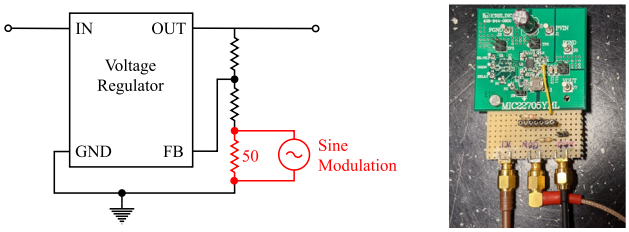
### 4.1 Measurement setup

As the setup for capturing the LLSI images, we use a Hamamatsu PHEMOS-1000 FA microscope, see Fig. 4a, equipped with a high-power incoherent light source (HIL) for optical probing. The microscope offers 5×, 20×, and 50× lenses and



(a) PHEMOS-1000 (b) Flip-chip package

**Fig. 4** Xilinx Kintex-7 target under the PHEMOS-1000 microscope with 5× lens in use (**a**) and photography of the Xilinx UltraScale device (**b**)




(a) Schematic (b) Modulator

**Fig. 5** LLSI modulation setup with the modulation regulator schematic (**a**) and the modified MIC22705YML-EV board (**b**)

an additional scanner zoom of ×2, ×4, and ×8. Due to the light source's wavelength of around 1.3 µm and the numerical aperture (NA) of our 50× lens of 0.71, the minimum beam diameter is around 1 µm. The step size of the galvanometric scan mirrors, however, is in the range of a few nanometers. For EOFM/LLSI measurements, the frequency of interest $f$, the bandpass bandwidth $\Delta f$, and the pixel dwell time $\Delta t_{\mathrm{px}}$ (in ms/px) can be configured in the PHEMOS software. To achieve LLSI measurements with an acceptable noise level, it is required to modulate the power rail of interest at more than around 80 kHz. LLSI image to the exact position on the chip, an optical light reflectance image can be captured alongside the measurement.

To better evaluate the LLSI signal differences and map them to a location on the optical image, we used the ImageJ application [37]. The pixel-wise subtraction of two LLSI images results in a mostly gray image with the differences displayed in white and black color. While this already shows the differences between the images clearly, the location of the changes is not intuitively visible. To superimpose the difference image on an optical image, we first remove noise by the "despeckle" functionality of ImageJ, and then merged the optical image and the difference image. To improve the visibility of the differences, we have remapped the black and white spots in the raw difference image to the colors yellow and green.

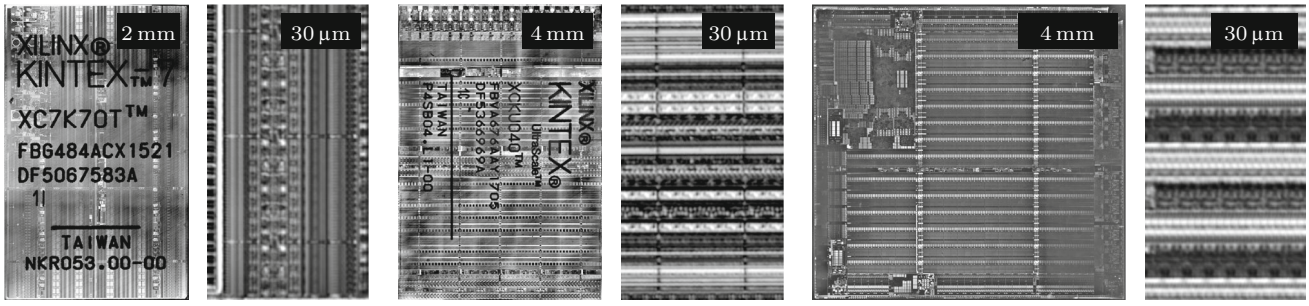
## 4.2 Devices under test

### 4.2.1 Xilinx Kintex-7 FPGA

As SRAM-based FPGA, we chose the Xilinx Kintex-7 XC7K70T, manufactured in a 28 nm technology. The chip is available in a ball grid array (BGA) bare-die flip-chip package on a Numato Systems Skoll development board. The FPGA can be programmed using the Xilinx Vivado IDE. In the Kintex-7 architecture [38], the logic fabric is comprised of (CLBs), so-called logic slices, and have a switch matrix for connecting to the global routing matrix. One slice consists of four 6-input LUTs (which can be configured as two 5-input LUTs with separate outputs each), eight FFs, as well as MUXes and arithmetic carry logic. While the slice naming uses $X$ and $Y$ coordinates (e.g., SLICE_X0Y0), the LUTs inside one slice are named from A5LUT/A6LUT to D5LUT/D6LUT, and the corresponding FFs from AFF/A5FF to DFF/D5FF. Next to the logic slices (2/3 of all slices), there are also memory slices usable as distributed RAM or shift registers.

To prepare the device for LLSI measurements, we disabled the onboard voltage regulator for VCC. Then, we soldered an SMA connector to the voltage rail for supplying the voltage externally via a power supply that can be modulated. For this purpose, we modified a MIC22705YML-EV voltage regulator evaluation board by replacing the resistor between the feedback pin and GND with a resistor to set the correct output voltage, in series with a 50 Ω resistor, see Fig. 5. In parallel to the latter, we connected a Keithley 3390 laboratory waveform generator to generate a sine wave. The regulator's output then provides a sine wave with a frequency of up to 300 kHz and a DC offset of the rated value for VCC of 1 V with a sufficient current drive strength. For higher frequencies, the regulator would stop functioning as intended. However, already when trying to modulate the DUT's voltage rail at low frequencies of a few kHz, no significant modulation can be measured on the printed circuit board (PCB)'s voltage rail. The reason for that is the existence of large decoupling capacitors, smoothing undesired peaks and fluctuations of the supply voltage. We desoldered all decoupling capacitors connected to VCC of 0.1 µF and larger using a hot air station to achieve a sufficiently high modulation amplitude. As a result, we could achieve a peak-to-peak modulation between 150 mV and 200 mV around the VCC offset of 1 V at a frequency $f$ of 80 kHz.

Figure 6a shows optical (light reflectance) images of the entire chip and a section of the logic fabric. A raw LLSI image from the Kintex-7 logic fabric indicates that the modulation of VCC influences the light reflection almost everywhere, see Fig. 7.




(a) Xilinx Kintex-7 (b) Xilinx UltraScale (c) Microchip PolarFire SoC

**Fig. 6** Laser reflection images of the DUTs: entire chip (left) and zoom into the logic fabric (right)

### 4.2.2 Xilinx UltraScale FPGA

As a second SRAM-based FPGA, we chose the Xilinx UltraScale XCKU040, manufactured in a 20 nm technology. The chip is available in a flip-chip bare-die package, see Fig. 4b, on an AVNET development board (model AES-KU040-DB-G). Similar to the Kintex-7 architecture (Sect. 4.2.1), the UltraScale logic fabric is comprised of CLBs. Each CLB contains one slice providing eight 6-input LUTs (which can also be configured as two 5-input LUTs with separate outputs), sixteen FFs, as well as MUXes and arithmetic carry logic. The slices are named using $X$ and $Y$ coordinates, whereas the LUTs and FFs are named with capital letters (A5LUT/A6LUT to H5LUT/H6LUT and AFF/AFF2 to HFF/HFF2). Next to the logic slices, there are memory slices that can be used as distributed RAM or shift registers. Figure 6b shows optical images of the entire chip and a section of the logic fabric.

To modulate the voltage rail of the UltraScale target, we used the same external modulation circuit as for the Kintex-7 (see Fig. 5). First, we disabled the onboard voltage regulator for VCC (0.95 V) by desoldering the coil at the regulator's output. Then, we soldered an SMA connector to the corresponding pad for supplying VCC externally. Furthermore, we desoldered all decoupling capacitors connected to VCC of 0.1 µF and larger from the PCB for being able to modulate the voltage rail at a sufficiently high frequency. For the experiments, we used a peak-to-peak modulation of around 150 mV at a frequency $f$ of 80 kHz with a VCC offset of 0.95 V.

### 4.2.3 Microchip PolarFire SoC FPGA

As flash-based FPGA, we chose the Microchip PolarFire SoC MPFS250T-FCVG484EES, manufactured in a 28 nm technology. The configuration is stored in distributed flash cells manufactured in Microchip's SONOS technology [39], consisting of two floating-gate transistors. The chip is available on the PolarFire SoC FPGA Icicle Kit in a BGA flip-chip

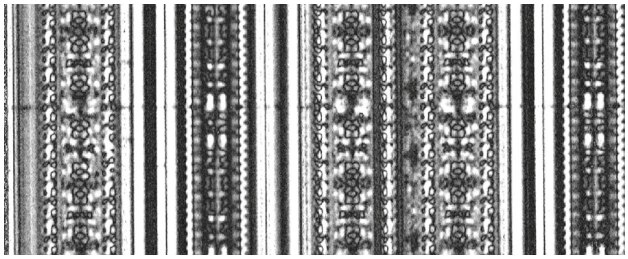


**Fig. 7** LLSI raw image from the logic fabric on the Kintex-7 FPGA. 50× (×2) zoom, $\Delta t_{\text{px}} = 2.1$ ms/px, $\Delta f = 300$ Hz

package with a lid. After cooling down the device in a typical household freezer, we could pry off the lid using a knife to access the chip backside. The FPGA can be programmed using the Microsemi Libero IDE. In the PolarFire architecture [40], the logic fabric is comprised of arrays of logic clusters (LCs) that are connected by interface logic (IL). Each LC consists of 12 logic elements (LEs), whereas each LE contains a 4-input LUT, a FF, and a MUX. Next to a connection to the IL, the individual LEs inside one LC are connected by a carry chain. Next to the LCs, there are other blocks, such as dedicated math and memory blocks, connected via the IL.

We could use the onboard MIC22705YML voltage regulator for modulating VDD of this target. Via a jumper, the resistor in the feedback path can be changed to create a 1.0 V or 1.05 V supply voltage. By removing the jumper and connecting our own resistors, we could create the same modulation capabilities as shown in Fig. 5a. To increase the LLSI signal's amplitude, we desoldered all decoupling capacitors connected to VDD of 0.1 µF and larger from the PCB. We used a peak-to-peak modulation of approximately 170 m V around the VDD offset of 1 V. A modulation frequency $f$ of 83.5 kHz led to the highest LLSI signal amplitude. Note that the SONOS cells are not supplied by VDD but VDD25, which is supplied by a 2.5 V regulator. To modulate the VDD25 voltage, we soldered a jumper to disable the onboard regulator and added an SMA connector to supply VDD25 via our external modulator circuit. However, as we could not detect any benefit over modulating VDD, we only used the

VDD modulation for the experiments presented in this paper. Figure 6b shows optical images of the entire chip and a part of the logic fabric.

## 5 Results

### 5.1 Detecting changes in the logic fabric

To investigate the capabilities of LLSI for detecting changes in the logic fabric configuration, we first tried to detect small changes within one logic element, i.e., changes in the LUT configurations and FF logic states. Although the number of different configurations is high, we aimed at creating a good coverage of detectable changes.

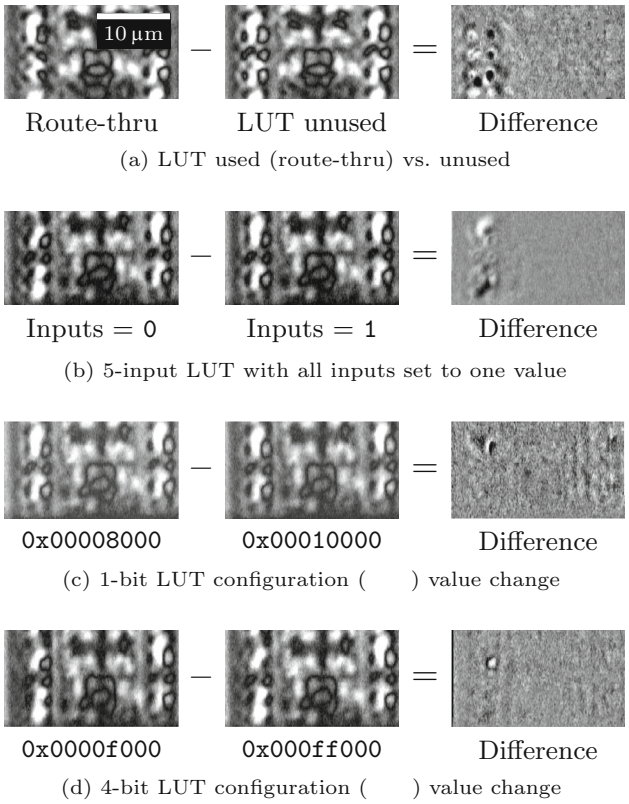
#### 5.1.1 SRAM-based (Kintex-7)

*LUT used versus unused* We compared implementations where once the LUT is unused and once a route-thru LUT is implemented. We assumed a route-thru LUT to be the configuration with minimal differences compared to the unused LUT, as the input of the LUT is directly routed to the output of the SLICE. Nevertheless, the differences can be clearly identified, see Fig. 8a.

*LUT inputs 0 versus 1* When changing the values of LUT inputs, which originate from the output of another LUT or a FF, the change is clearly visible as well, see Fig. 8b. As could be expected, we observed fewer changes if fewer input values are changed. Still, we could detect changes also if only one input value is changed.

*LUT configuration value changes* The smallest possible change we could imagine is the manipulation of single bits in the LUT configuration. We observed that the number of bits changed in the LUT configuration `INIT` value does not necessarily determine how significant the difference in the LLSI response is, see Fig. 8c, d. We assume that not the SRAM cell holding the configuration produces the LLSI signature, but the actual multiplexers and pass transistors. If a configuration change causes—due to the applied LUT inputs—more multiplexers to change their states (cf. Fig. 1), there will be a bigger difference between the LLSI images.

*FF value 0 versus 1* Finally, we designed a bit more complex design, which contains two FFs and one LUT residing in different logic slices, see Fig. 9. We have subtracted the LLSI images of two consecutive clock cycles. While the difference for the LUT is concentrated in a single small area, there are many different spots for the FFs. This might be explained by the fact that the input buffers, the actual memory cell, the output buffers, and the clock buffers have changed their values by advancing a clock cycle as well. Interestingly, although the two registers were instantiated in exactly the same way in the IDE, different changes can be observed between them.




(a) LUT used (route-thru) vs. unused

(b) 5-input LUT with all inputs set to one value

(c) 1-bit LUT configuration ( ) value change

(d) 4-bit LUT configuration ( ) value change

**Fig. 8** Kintex-7 LLSI results for different lookup-table configurations. 50× (×4) zoom, $\Delta t_{\text{px}} = 3.3\,\text{ms/px}$, $\Delta f = 100\,\text{Hz}$

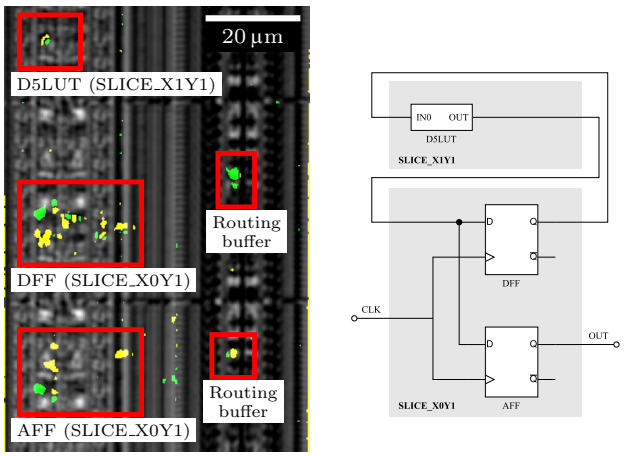
This might be caused by the different output configurations of the FFs or an asymmetric ASIC design of the CLB. For instance, the clock buffers or some intra-CLB routing capabilities, which are invisible in the IDE for the designer, might reside close to DFF. Finally, we could observe differences in the (assumed-to-be) routing areas, supposedly interconnecting the two slices X0Y1 and X1Y1.

#### 5.1.2 SRAM-based (UltraScale)

To investigate if similar results can be achieved on a DUT manufactured in a smaller technology, we conducted the same experiments on the UltraScale FPGA.

*LUT used versus unused* Although the technology node size of the UltraScale series is around 28% smaller than of the Kintex-7 series, the difference between a route-thru LUT and a completely unused LUT is clearly visible, see Fig. 10a. Due to the technology size reduction, the affected area is smaller but can still be resolved using our optical setup. Furthermore, the difference image looks more blurry than for the Kintex-7 FPGA. One explanation for this might be the lower modulation amplitude achievable on the UltraScale board.

*LUT inputs 0 versus 1* Flipping the LUT's inputs values can be detected reliably as well, see Fig. 10b. Interestingly, the affected area seems to be as large as in the previous exper-




(a) LLSI difference (b) Logic schematic

**Fig. 9** Kintex-7 LLSI difference superimposed over an optical image for FF values 0 versus 1 with CLB inputs and outputs connected. Yellow and green colors correspond to the black and white spots in the raw difference image. 50× (×2) zoom, $\Delta t_{\text{px}} = 2.1$ ms/px, $\Delta f = 300$ Hz

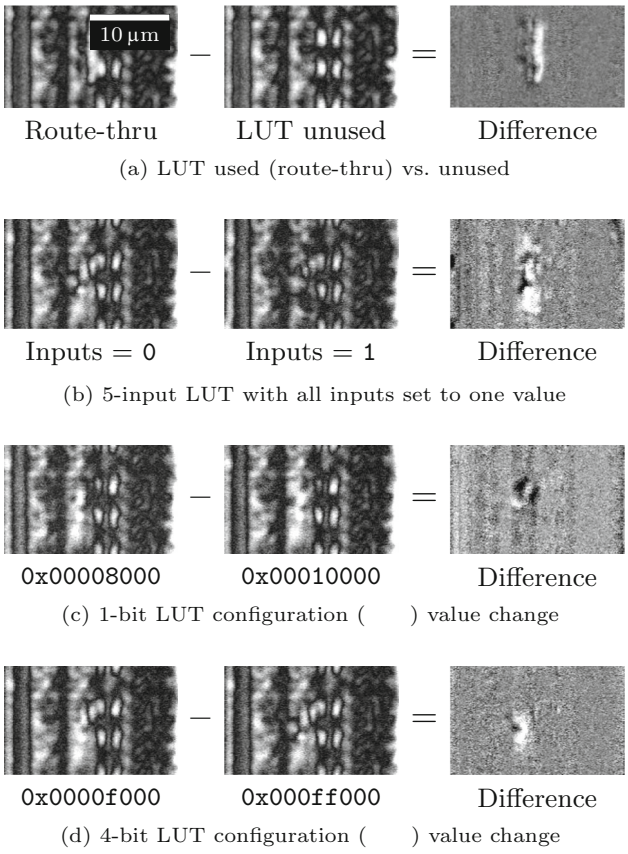
iment on used vs. unused LUT. The reason might be that we cannot control the routing of signals and which values are applied to unused inputs.

*LUT configuration value changes* We could clearly detect the same LUT configuration changes that we could detect on the Kintex-7, see Fig. 10c, d. For this target, the affected area neither reflects the number of bits changed in the configuration. This observation supports the hypothesis that the LUT's multiplexers and not the memory cells for the configuration contribute most to the LLSI signal.

*FF value 0 versus 1* When investigating an entire CLB with one LUT and two FFs in use, multiple areas with differences in the LLSI image can be observed, see Fig. 11. Again, we subtracted the LLSI images of two consecutive clock cycles. From the knowledge gained in the previous experiments, we could identify the changes in the LUT and map two areas with similar changes to the two FFs. Despite these distinctly allocable changes, many other areas with clear differences appear in the image. These changes seem to belong to the CLB's MUXes (left of the LUTs and FFs) and routing resources, such as buffers (right side of the image). However, since the chip's layout is unknown, these assumptions cannot be verified further.

### 5.1.3 Flash-based (PolarFire SoC)

To investigate whether configuration changes can also be detected on the flash-based FPGA, we conducted similar experiments on the PolarFire SoC FPGA.



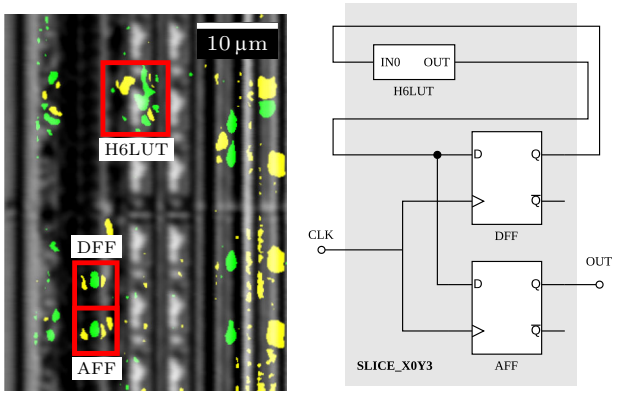

(a) LUT used (route-thru) vs. unused

(b) 5-input LUT with all inputs set to one value

(c) 1-bit LUT configuration ( ) value change

(d) 4-bit LUT configuration ( ) value change

**Fig. 10** UltraScale LLSI results for different lookup-table configurations. 50× (×4) zoom, $\Delta t_{\text{px}} = 2.1$ ms/px, $\Delta f = 300$ Hz




(a) LLSI difference (b) Logic schematic

**Fig. 11** UltraScale LLSI difference superimposed over an optical image for different FF values and LUT inputs. 50× (×4) zoom, $\Delta t_{\text{px}} = 2.1$ ms/px, $\Delta f = 300$ Hz

*LUT used versus unused* For this target, we compared the configuration for a route-thru LUT with an unused LUT as well, see Fig. 12a. The LLSI responses show a clear difference, although the corresponding area is smaller than on the Xilinx FPGAs. The reason might be that the LUTs on Kintex-7 and UltraScale have up to 6 inputs, while they only have 4 inputs on PolarFire, resulting in a significant difference in the number of contained MUXes.

*LUT inputs 0 versus 1* The area of differences when only the LUT inputs change are smaller than the differences between a used and unused LUT—as can be expected, see Fig. 12b.

*LUT configuration value changes* Changes in the LUT configurations can be detected as well. For a large change in the configuration, i.e., by flipping all bits, the change with the largest area is visible, see Fig. 12c. As for the other FPGAs, the reason might be the different number of MUXes affected by the configuration change, under the assumption that the inputs of the LUT stay constant. For a 2-bit change in the `INIT` value, a smaller difference is visible, see Fig. 12d. Moreover, we observed that when all LUT inputs are set to `0`, the difference for changed `INIT` values is larger than when all inputs are set to `1`. Since in our experiment the output of the LUT was not changed by applying the different inputs (due to the configured `INIT` value), we suppose that a different number of multiplexers changed their states depending on the LUT inputs.

*FF value 0 versus 1* Similar to the experiments on the SRAM-based FPGAs, we created snapshots of a larger area of the logic fabric, on the one hand, to observe the LLSI response differences for a FF, and on the other hand, to learn about the detectability of buffers and routing transistors. Figure 13 shows the difference of two LLSI responses captured in two consecutive clock cycles. The state change of the FF is clearly visible on the top right of the image. The three LUTs receive the output of the FF as inputs, and therefore, their responses differ, too. Differences can also be observed in between the rows of logic elements. These areas presumably belong to the routing logic, thus containing data and clock buffers.

## 5.2 Detecting changes in routing

The authors of [12] propose malicious modifications in the signal runtime on the FPGA by using either route-thru LUTs or manipulating the routing to take longer paths. We have already shown that the insertion of route-thru LUTs can be detected; see Sect. 5.1. To test the capability of our approach

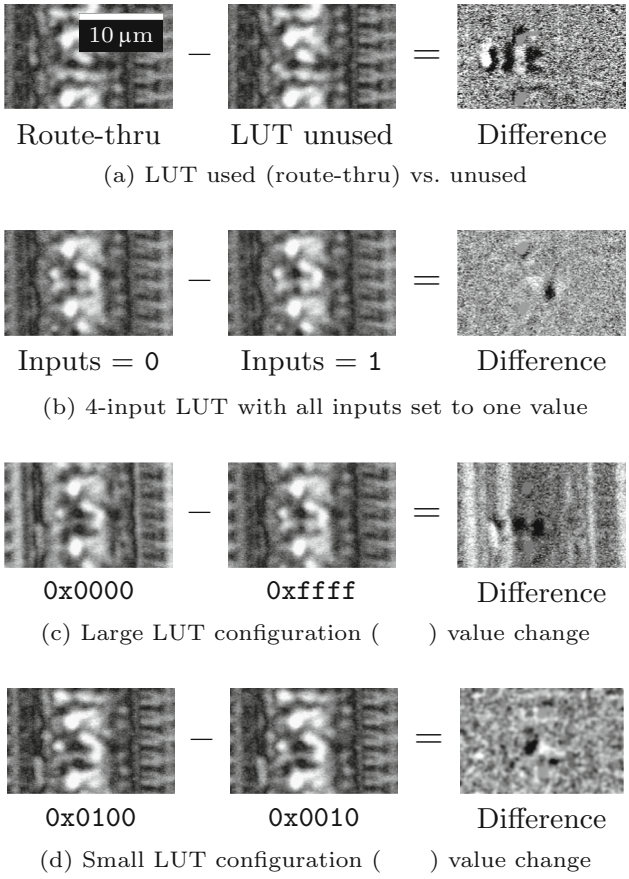
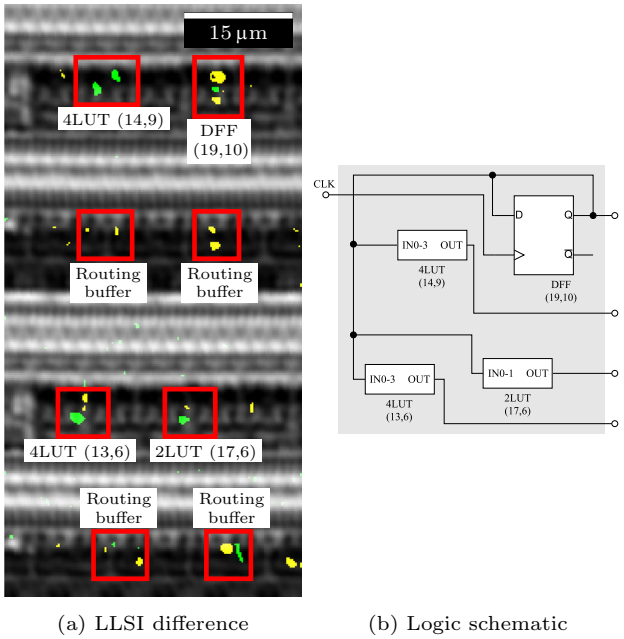
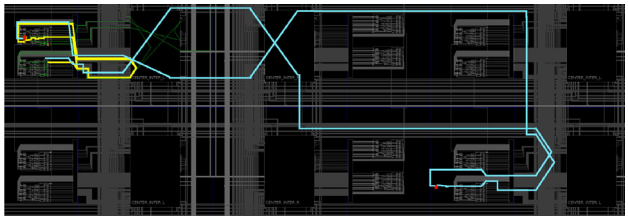



**Fig. 12** PolarFire SoC LLSI results. Images rotated by 90 degrees, 50× (×4) zoom, $\Delta t_{\text{px}} = 3.3$ ms/px, $\Delta f = 100$ Hz

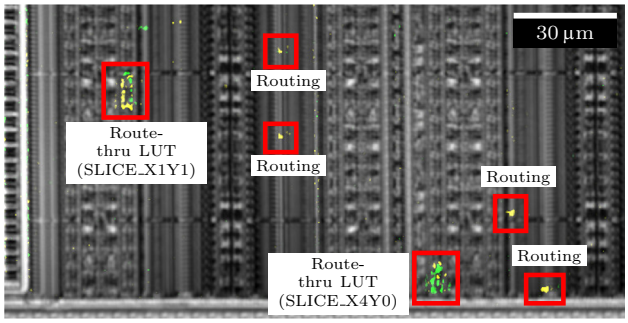



(a) LLSI difference (b) Logic schematic

**Fig. 13** PolarFire SoC LLSI difference superimposed over an optical image for different FF values and LUT inputs. 50× (×2) zoom, $\Delta t_{\text{px}} = 3.3$ ms/px, $\Delta f = 100$ Hz

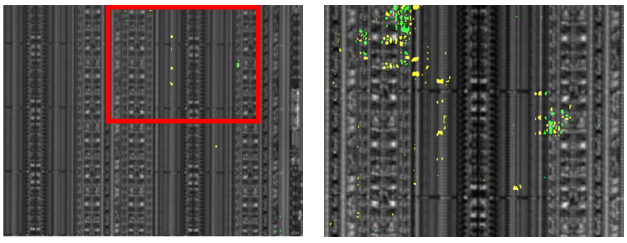


(a) Placement schematic of design with LUT in SLICE_X1Y1 (yellow) and in SLICE_X4Y0 (blue)



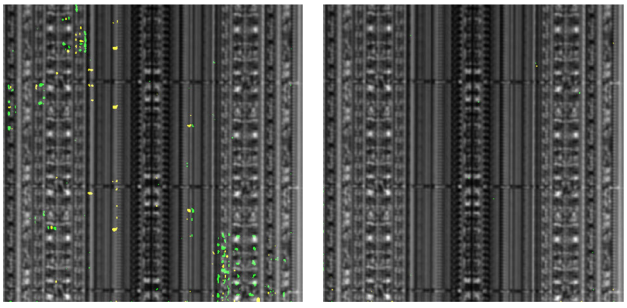

(b) LLSI difference

**Fig. 14** Difference in routing configuration on Kintex-7 when moving a route-thru LUT from SLICE X1Y1 to X4Y0 while keeping the signal source and destination in SLICE X1Y1 and X0Y1. 50× (×2) zoom, $\Delta t_{\text{px}} = 2.1$ ms/px, $\Delta f = 300$ Hz



(a) 50× zoom (b) 50× (×2) zoom

**Fig. 15** Combinatorial Trojan benchmark (c2670_T071) section on Kintex-7. (**a**) $\Delta t_{\text{px}} = 5ms/px$, (**b**) $\Delta t_{\text{px}} = 3.3$ ms/px, $\Delta f = 100$ Hz



(a) Free vs. Trojan (b) Free vs. Free, different runs

**Fig. 16** Sequential Trojan benchmark (s1423_T607) section on Kintex-7. 50× (×2) zoom, $\Delta t_{\text{px}} = 3.3ms/px$, $\Delta f = 100Hz$

to detect changes in the routing, we created a design for the Kintex-7 FPGA that contains one route-thru LUT, whose location we change between two measurements. Thereby, the signal is forced to be routed differently. For the first snapshot, the LUT is placed in SLICE_X1Y1, while for the second snapshot, it is placed in SLICE_X4Y0, see Fig. 14a. The signal source and sink are kept at the same location (in SLICE_X0Y1 and X1Y1). Figure 14b clearly shows not only the differences in the LLSI response for the changed LUT placement but also for the routing logic. Consequently, one can also detect changes in signal routing with our approach.

## 5.3 Trojan benchmarks

The previous results have already shown that small changes, down to single bit changes in the LUT configuration and small changes in the routing configuration, can be detected using our method. Therefore, we have demonstrated that LLSI can detect the malicious modifications proposed in [12] introducing changes in the signal path delays. To demonstrate that we can also detect other HTs proposed in the literature, we exemplarily implemented HT benchmarks generated using the TRIT framework [30], which can be found on TrustHub [29]. We implemented two benchmarks on the Kintex-7 DUT, one consisting only of combinatorial HT logic (from TRIT-TC) and one also containing sequential logic (from TRIT-TS). All provided benchmarks generated using TRIT introduce additional logic gates and/or FFs. We fixed the location and routing placement of all logic components and the routing that does not belong to the HT trigger or payload to keep the changes of the implementation minimal.

### 5.3.1 Combinatorial Trojan

The c2670_T071 HT benchmark introduces six additional logic gates. Figure 15 only shows a part of the logic fabric area consumed by the implementation. However, already in this section of the design, clear differences can be observed. As can be seen, zooming into an area with suspicious differences can highlight the changes more clearly.

### 5.3.2 Sequential Trojan

Next to combinatorial gates, the s1423_T607 benchmark contains a counter with 15 states implemented using FFs. Figure 16a indicates that many changes can be detected both in the CLBs and routing areas. As expected, when capturing two LLSI images of the same area from the Trojan-free design, no clear differences can be observed, see Fig. 16b. This proves that the previously observed differences are not only caused by noisy measurements.

# 6 Discussion

In this section, we first discuss further research directions continuing our approach. Subsequently, we talk about the applicability of our approach and discuss potential limitations.

## 6.1 Further research directions

### 6.1.1 Application to ASICs

Regarding the applicability of our approach to ASIC implementations, a few things have to be kept in mind. Generally, it should be possible to detect the locations of all transistors and then overlay the layout file. In this way, irregularities and deviations from the intended designs can be detected, even without having a golden chip. One drawback is that modifications that only affect the metal layers cannot be detected if the changes do not manifest in the light reflection. However, we think that detecting analog HTs, such as capacitor-based and dopant-level Trojans, should be possible using LLSI. Since these HTs use analog properties of the chip and are pre-silicon modifications, we could not investigate them. However, in the following, we explain why our approach should be able to detect such HTs.

*Detecting capacitor-based Trojans* Results from [24] indicate that decoupling capacitors can be imaged using LLSI. Since these capacitors are connected between VCC and GND, the power supply modulation will modulate the electric field and charge density of the capacitor, which influences the light reflection. Therefore, LLSI might also be applicable to detect HTs that only introduce changes in the capacitance to create a stealthy trigger mechanism (e.g., A2 Trojans [13]).

*Detecting dopant-level Trojans* The investigations in [41] and [42] show that the light reflection for optical probing depends on the doping level of the silicon. Therefore, malicious modifications in the doping concentration to alter the functionality of logic gates [14] might be detectable using LLSI.

### 6.1.2 Reverse-engineering the FPGA configuration

As already shown in this work, the configuration of the FPGA logic fabric is contained in the LLSI snapshots. Although the resolution seems to be insufficient to extract the exact configurations manually, machine learning approaches might be able to solve that task. The advantages of employing deep learning techniques have already been demonstrated in [36] for data extraction from dedicated on-chip memories. Such configuration extraction can also facilitate the structural and functional reverse engineering of bitstreams in proprietary formats.

## 6.2 Applicability of LLSI

We have shown that our approach using LLSI can detect a wide range of changes in the FPGA logic fabric configuration. In the following, we discuss the practical applicability of LLSI.

### 6.2.1 Chip access

For our approach, we need access to the silicon backside of the chip. Since all FPGAs used in this work are only available in flip-chip packages, this requirement can be easily met. Moreover, due to performance, size, cost, and environmental compatibility reasons, chips are predominantly delivered in flip-chip packages [43]. While many of such packages have a lid installed—which we could easily remove for the PolarFire SoC—there are also bare-die packages available, like the one of our Kintex-7 and UltraScale DUTs. Consequently, if a customer would like to have the opportunity to test the chip for HTs using an optical probing approach, he or she should choose a bare-die package to facilitate testing. Thinning or polishing the silicon backside is not necessary for optical probing, as shown in this work.

### 6.2.2 PCB modifications

In order to reach modulation frequencies of 80 kHz and higher, we had to replace the voltage regulator on the Kintex-7 and UltraScale DUTs with an external one. However, on the PolarFire DUT, we could leverage the on-PCB regulator for the modulation, requiring no modifications on the PCB. Consequently, by using a suitable voltage regulator on the PCB, there is no need to provide the modulated voltage from an external source.

During our investigations, we observed that a higher modulation of the supply voltage produces a clearer LLSI image, and consequently, a shorter pixel dwell time is sufficient. Moreover, a higher modulation frequency can further reduce the pixel dwell time, leading to faster scan times. The PCB and the die interposer PCB, however, are designed to compensate spikes and smooth undesired peaks and fluctuations of the supply voltage. For this purpose, decoupling capacitors of different sizes are connected between the supply voltage rail and ground, effectively acting as low-pass filters.

To achieve the desired modulation amplitude of the power rail at frequencies above 80 kHz, we had to remove the decoupling capacitors of 0.1 µF and larger from the PCB. Due to the existence of other capacitive and inductive elements in the circuit, a higher modulation frequency results in a lower modulation amplitude and, therefore, a lower LLSI signal level. Consequently, there is a tradeoff between the noise ratio in the LLSI images, the scan time, and the electrical preparation of the DUT. Due to practical reasons, we did not remove

smaller capacitors. Furthermore, we did not remove capacitors from the interposer PCB, as there is no documentation on potential effects available. Nevertheless, a device that is ready for use in a practical application must have installed all capacitors due to reliability and stability constraints. One way to still enable the measurements required by our approach is the installation of jumpers or other switches on the PCB to disable the capacitors on demand.

#### 6.2.3 Optical stability

In our experiments, we observed that the optical focus was slightly drifting during the LLSI measurements due to mechanical instabilities in the setup. Since the LLSI signal heavily depends on the focus position, there are small differences between LLSI images that are not caused by design modifications. However, the stability of our setup was sufficient to produce reliable and significant results for detecting malicious changes in the design. Nevertheless, the image quality will improve if the mechanical stability is enhanced, for instance, by operating the setup in a tempered room and a shock-absorbing building.

#### 6.2.4 Optical resolution

The optical resolution of laser-assisted side-channel techniques has been discussed extensively by the research community in numerous publications, e.g., in [4, 21, 44–47]. We discuss the most important and new insights in the following.

Both FPGAs used in this work were manufactured in 28 nm and even 20 nm technologies. Although the minimum width of our setup's optical beam is around 1µm, it should be kept in mind that the technology size does distinguish neither the minimum size of a transistor nor the typical distance between transistors. An important fact is that the laser scanner has a step size in the range of a few nanometers. Therefore, while scanning with the laser over the DUT, the beam covers one specific point on the chip multiple times. Consequently, if the beam covers multiple nodes of interest, the LLSI image shows a different position-dependent superposition of the same nodes at different adjacent pixel locations. However, due to the Gaussian intensity distribution of the beam, it might still be possible to extract the logic state. This explains why optical probing delivers meaningful results also on structures that are smaller than the beam diameter.

Moreover, a so-called solid immersion lens (SIL) can be used to increase the optical resolution down to 250 nm [48], which is sufficient to resolve individual transistors in a 14 nm technology [49]. Accordingly, Intel has shown that LLSI can be applied on very small devices, such as single inverters, on a test chip manufactured in a 14 nm technology [24].

Even if it might not be possible to resolve single SRAM cells used for configuration storage in future technologies, the FFs, MUXes, and other pass transistors are influenced by the configuration and contribute to the LLSI image as well. This is supported by the observation that even on the 20 nm FPGA, the different LUT configurations could be detected. Furthermore, typical HTs in benchmarks alter the design by inserting or modifying multiple logic gates or FFs, resulting in huge changes, which we could detect reliably.

## 7 Conclusion

Dormant hardware Trojans that introduce only tiny malicious hardware modifications pose a severe threat in security-critical applications. In this work, we have demonstrated a detection approach for dormant HTs using the laser-assisted optical probing method LLSI. By modulating the power supply of the chip, even inactive logic is visible on the logic snapshots. By awakening the potential Trojan in this way, no malicious modification of the FPGA's configuration stays undetected. We have demonstrated that our approach is applicable to recent SRAM- and flash-based FPGAs on the market in a non-invasive manner. It did not make a significant difference whether the FPGAs were manufactured in a 28 nm or 20 nm technology. Finally, we have explained why our framework should also be suitable for detecting stealthy HTs on ASICs.

**Acknowledgements** The authors would like to acknowledge Hamamatsu Photonics K. K. Japan and Germany for their help and support on the PHEMOS system.

**Funding** Open Access funding enabled and organized by Projekt DEAL. The authors from Technische Universität Berlin have been supported in part by the Einstein Foundation (EP-2018-480), and in part by the Deutsche Forschungsgemeinschaft (DFG—German Research Foundation) under the priority programme SPP 2253, Grant Number 439918011. The author from Worcester Polytechnic Institute has been supported in part by National Science Foundation (NSF) under the Grant Number 2117349 and in part by Massachusetts Technology Collaborative (MassTech).

**Data availability** The datasets generated during and analyzed during the current study are available from the corresponding author on reasonable request.

## References


1. Ng, X.T., et al.: Integrated sensor: a backdoor for hardware trojan insertions? In: 2015 Euromicro conference on digital system design (2015). https://doi.org/10.1109/DSD.2015.119
2. Roy, D.B., et al.: The conflicted usage of RLUTs for security-critical applications on FPGA. J. Hardw. Syst. Secur. **2**, 162–178 (2018). https://doi.org/10.1007/s41635-018-0035-4
3. Moradi, A., Schneider, T.: Improved side-channel analysis attacks on Xilinx bitstream encryption of 5, 6, and 7 series . In: International workshop on constructive side-channel analysis and secure design (2016)
4. Tajik, S., Lohrke, H., Seifert, J.-P., Boit, C.: On the power of optical contactless probing: attacking bitstream encryption of FPGAs . In: 2017 ACM SIGSAC conference on computer and communications security (CCS) (2017)
5. Lohrke, H., Tajik, S., Krachenfels, T., Boit, C., Seifert, J.-P.: Key extraction using thermal laser stimulation. In: Conference on cryptographic hardware and embedded systems (CHES) (2018)
6. Hettwer, B., Leger, S., Fennes, D., Gehrer, S., Güneysu, T.: Side-channel analysis of the Xilinx Zynq ultrascale+ encryption engine. In: Conference on cryptographic hardware and embedded systems (CHES) (2021)
7. Ender, M., Moradi, A., Paar, C. The unpatchable silicon: a full break of the bitstream encryption of Xilinx 7-series FPGAs. In: 29th USENIX security symposium (USENIX security 20) (2020)
8. Zhang, Z., Njilla, L., Kamhoua, C.A., Yu, Q.: Thwarting security threats from malicious FPGA tools with novel FPGA-oriented moving target defense. IEEE Trans. VLSI Syst. **27**(3), 665–678 (2019). https://doi.org/10.1109/TVLSI.2018.2879878
9. Microchip Technology, Inc. UG0753 User guide PolarFire FPGA security (2021)
10. Xilinx, Inc.: Developing tamper-resistant designs with Zynq UltraScale+ devices (2018)
11. Salmani, H., Tehranipoor, M., Plusquellic, J.: New design strategy for improving hardware trojan detection and reducing trojan activation time . In: 2009 IEEE international workshop on hardware-oriented security and trust (HOST) (2009)
12. Ender, M., Ghandali, S., Moradi, A., Paar, C.: The first thorough side-channel hardware trojan. In: Advances in Cryptology—ASIACRYPT 2017. **10624**, 755–780 (2017). https://doi.org/10.1007/978-3-319-70694-8_26
13. Yang, K., Hicks, M., Dong, Q., Austin, T., Sylvester, D.: A2: analog malicious hardware. In: 2016 IEEE symposium on security and privacy (SP) (2016). https://doi.org/10.1109/SP.2016.10
14. Becker, G.T., Regazzoni, F., Paar, C., Burleson, W.P.: Stealthy dopant-level hardware trojans. In: Conference on cryptographic hardware and embedded systems (CHES) (2013). https://doi.org/10.1007/978-3-642-40349-1_12
15. Song, P., et al.: MARVEL—malicious alteration recognition and verification by emission of light. In: 2011 IEEE international symposium on hardware-oriented security and trust (HOST) (2011). https://doi.org/10.1109/HST.2011.5955007
16. Stellari, F., et al.: Verification of untrusted chips using trusted layout and emission measurements. In: 2014 IEEE international symposium on hardware-oriented security and trust (HOST) (2014). https://doi.org/10.1109/HST.2014.6855562
17. Duncan, A., et al.: FLATS: filling logic and testing spatially for FPGA authentication and tamper detection. In: 2019 IEEE international symposium on hardware oriented security and trust (HOST) (2019). https://doi.org/10.1109/HST.2019.8741025
18. Nguyen, L.N., Cheng, C.-L., Prvulovic, M., Zajic, A.: Creating a backscattering side channel to enable detection of dormant hardware trojans. IEEE Trans. VLSI Syst. **27**(7), 1561–1574 (2019). https://doi.org/10.1109/TVLSI.2019.2906547
19. Adibelli, S., Juyal, P., Nguyen, L.N., Prvulovic, M., Zajic, A.: Near field backscattering based sensing for hardware trojan detection. IEEE Trans. Antennas Propag. (2020). https://doi.org/10.1109/TAP.2020.3000562
20. He, J., Ma, H., Liu, Y., Zhao, Y.: Golden chip-free trojan detection leveraging trojan trigger's side-channel fingerprinting. ACM Trans. Embed. Comput. Syst. **20**(1), 1–18 (2020). https://doi.org/10.1145/3419105
21. Stern, A., Mehta, D., Tajik, S., Farahmandi, F.,Tehranipoor, M.: SPARTA: a laser probing approach for trojan detection. In: 2020 IEEE international test conference (ITC) (2020)
22. Zhou, B., et al.: Hardware trojan detection using backside optical imaging. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. (2020). https://doi.org/10.1109/TCAD.2020.2991680
23. Krachenfels, T., Ganji, F., Moradi, A., Tajik, S., Seifert, J.-P.: Real-world snapshots vs. theory: questioning the t-probing security model . In: 2021 IEEE symposium on security and privacy (SP) (2021). https://doi.org/10.1109/SP40001.2021.00029
24. Niu, B., et al.: Laser logic state imaging (LLSI) (2014). In: 40th international symposium for testing and failure analysis ISTFA (2014)
25. Krachenfels, T., Seifert, J.-P., Tajik, S.: Trojan awakener: detecting dormant malicious hardware using laser logic state imaging. In: 5th workshop on attacks and solutions in hardware security (2021). https://doi.org/10.1145/3474376.3487282
26. Wang, X., Tehranipoor, M., Plusquellic, J.: Detecting malicious inclusions in secure hardware: challenges and solutions. In: 2008 IEEE international workshop on hardware-oriented security and trust (2008). https://doi.org/10.1109/HST.2008.4559039
27. Bhunia, S., Hsiao, M.S., Banga, M., Narasimhan, S.: Hardware trojan attacks: threat analysis and countermeasures. Proc. IEEE **102**(8), 1229–1247 (2014). https://doi.org/10.1109/JPROC.2014.2334493
28. Becker, G.T., Regazzoni, F., Paar, C., Burleson, W.P.: Stealthy dopant-level hardware trojans. In: Conference on cryptographic hardware and embedded systems (CHES) (2013)
29. Shakya, B., et al.: Benchmarking of hardware trojans and maliciously affected circuits. J. Hardw. Syst. Secur. **1**(1), 85–102 (2017). https://doi.org/10.1007/s41635-017-0001-6
30. Cruz, J., Huang, Y., Mishra, P., Bhunia, S.: An automated configurable trojan insertion framework for dynamic trust benchmarks (2018). In: 2018 design, automation & test in Europe conference & exhibition (DATE). https://doi.org/10.23919/DATE.2018.8342270
31. Jacob, N., Rolfes, C., Zankl, A., Heyszl, J., Sigl, G.: Compromising FPGA SoCs using malicious hardware blocks (2017). In: Design, automation test in Europe conference exhibition (DATE). https://doi.org/10.23919/DATE.2017.7927157
32. Jacob, N., Heyszl, J., Zankl, A., Rolfes, C., Sigl, G.: How to break secure boot on FPGA SoCs through malicious hardware. In: Conference on cryptographic hardware and embedded systems (CHES) (2017)
33. Vashistha, N., et al.: Trojan scanner: detecting hardware trojans with rapid SEM imaging combined with image processing and machine learning. In: 44th international symposium for testing and failure analysis (ISTFA) (2018)
34. Sugawara, T., et al.: Reversing stealthy dopant-level circuits. In: International workshop on cryptographic hardware and embedded systems (2014)
35. Doug Black.: Xilinx says its new FPGA is world's largest (2019). https://www.enterpriseai.news/2019/08/21/xilinx-says-its-new-fpga-is-worlds-largest/

36. Krachenfels, T., Kiyan, T., Tajik, S., Seifert, J.-P.: Automatic extraction of secrets from the transistor jungle using laser-assisted side-channel attacks. In: 30th USENIX security symposium (USENIX security 21) (2021)
37. Rueden, C.T., et al.: Image J2: ImageJ for the next generation of scientific image data. BMC Bioinform. **18**(1), 529 (2017). https://doi.org/10.1186/s12859-017-1934-z
38. Xilinx, Inc.: 7 series FPGAs configurable logic block user guide (UG474) (2016)
39. Microsemi Corporation: White paper: PolarFire non-volatile FPGA family delivers ground breaking value: cost optimized, Lowest Power, EU immunity, and high-security (2017). https://www.microsemi.com/document-portal/doc_download/1243174-polarfire-fpga-white-paper
40. Microchip Technology, Inc.: UG0680 user guide PolarFire FPGA fabric (2021)
41. Kindereit, U., et al.: Quantitative investigation of laser beam modulation in electrically active devices as used in laser voltage probing. IEEE Trans. Device Mater. Reliab. **7**(1), 19–30 (2007). https://doi.org/10.1109/TDMR.2007.898074
42. Kindereit, U.: Investigation of laser-beam modulations induced by the operation of electronic devices. Ph.D. thesis, Technische Universität Berlin (2009). https://depositonce.tu-berlin.de//handle/11303/2440
43. Tong, H., Lai, Y., Wong, C.: Advanced Flip Chip Packaging. Springer (2013)
44. Boit, C., et al.: From IC debug to hardware security risk: the power of backside access and optical interaction. In: 23rd international symposium on the physical and failure analysis of integrated circuits (IPFA) (2016). https://doi.org/10.1109/IPFA.2016.7564318
45. Lohrke, H., Tajik, S., Boit, C., Seifert, J.-P.: No place to hide: contactless probing of secret data on FPGAs. In: Conference on cryptographic hardware and embedded systems (CHES) (2016). https://doi.org/10.1007/978-3-662-53140-2_8
46. Rahman, M.T., et al.: Physical inspection attacks: new frontier in hardware security. In: 2018 IEEE 3rd international verification and security workshop (IVSW) (2018). https://doi.org/10.1109/IVSW.2018.8494856
47. Rahman, M.T., Tajik, S., Rahman, M.S., Tehranipoor, M., Asadizanjani, N.: The key is left under the mat: on the inappropriate security assumption of logic locking schemes. In: IEEE international symposium on hardware oriented security and trust (HOST) (2020)
48. Hamamatsu Photonics K.K.: NanoLens-SHR (2015). https://www.hamamatsu.com/resources/pdf/sys/SSMS0053E_Nanolens-SHR.pdf
49. Von Haartman, M., et al.: Optical fault isolation and nanoprobing techniques for the 10 nm technology node and beyond. In: 41st international symposium for testing and failure analysis (ISTFA) (2015)

# 7

# DISCUSSION & FUTURE WORK

This chapter discusses the publications of this thesis as a whole. Due to the cumulative nature of this thesis, the chapter also explores relevant related work published after the individual publications of this work. Finally, this chapter contains hints and ideas about potential future investigations and challenges.

## 7.1 ACTIVE SIDE-CHANNEL ANALYSIS

In this thesis, LLSI and TLS have proven to be powerful single-trace optical SCA techniques. As opposed to classical *passive* SCA, like power and EM analysis, they can be seen as *active* SCA techniques. For LLSI, the power supply must be actively modulated for the measurement to succeed. It can be said that the modulation actively establishes the side channel. Similarly, for TLS, the thermal stimulation actively influences the device to measure data dependencies in the device's power consumption. Again, only the active stimulation of the device opens the side channel. More such active SCA techniques can be found in the literature. An example is the backscattering approach presented by Nguyen *et al.* [66]. The principle is that an electromagnetic field stimulates the IC, and its reflection is analyzed. Similarly, such scattering measurements can be conducted electrically on PCB [67] or IC level [59]. The authors stimulate the device with signals of different frequencies and measure the frequency response. Although they initially used this approach to detect tamper events only, the technique has proven to be an effective SCA technique [68].

Regarding future work, one can expect to see more of these techniques actively stimulating the device to exploit a side channel. One example might be the photon emission side channel caused by the leakage current flow of transistors. Couch *et al.* show that this approach can extract data from on-chip SRAM cells [69]. However, one measurement takes many hours, which makes the technique almost unusable in practice. If one could actively increase the emitted photons caused by leakage current, measurement times could be reduced considerably.

## 7.2 ADDITIONAL COUNTERMEASURE APPROACHES

In the publications contained in this thesis, we have described existing and proposed new countermeasure concepts. This section presents and discusses newly developed approaches from the literature.

### 7.2.1 *Generic Optical Probing Countermeasures*

As discussed in this thesis' publications, a generic countermeasure approach to prevent optical attacks through the chip backside would be coating an opaque layer on the chip backside [70]. The intactness of this layer should be actively monitored to detect tampering attempts. However, such a countermeasure requires additional processing steps, and therefore, to the best of our knowledge, there is no implementation of an active backside coating ready for mass production.

Another recently studied approach tries to re-design the logic and logic gates for more resistance against optical probing. Rahman *et al.* place additional logic gates next to the gates to be protected [71]. These gates carry the opposite logic states with the goal that the effects on the reflected light cancel out each other and, therefore, the signal of interest cannot be probed. This approach has two significant issues. Firstly, it assumes that the optical resolution of the laser probe is sufficiently low so that it can only cover areas larger than the targeted gate together with the protection gates. However, even if this assumption does hold, while scanning over the DUT, the laser might be parked at a location where only partial canceling occurs, and the signal of interest can still be detected. Secondly, NMOS and PMOS transistors contribute differently to the reflected light. Consequently, there might be no perfect cancellation of the signals at all. Nevertheless, the approach can reduce the SNR of the optical probe.

We have presented a similar approach in [22]. In contrast to the previous approach, we investigated different logic styles concerning better cancellation effects. Dual-rail logic gates have been shown to provide better cancellation, which requires the attacker to achieve higher optical resolutions. Furthermore, the data-dependent reflection can be reduced by lowering the supply voltage and limiting the output swing of the logic. Since no experiments on real devices were conducted to confirm the assumptions, this will have to be part of future research.

### 7.2.2 *Countermeasures Against Single-Trace Attacks*

In the publications presented in this thesis, we have outlined that for single-trace optical attacks such as TLS or LLSI, the clock must be frozen, or the logic state must be retained during the entire scan. Therefore, detecting a frozen clock can be one measure to detect an LLSI measurement. Roy *et al.* propose a circuit-level countermeasure based on timing sensors [72]. The main idea is to create an internally-generated asynchronous sensor clock and compare it to the synchronous system clock during runtime. Since there is no electrical control over the sensor from the outside, the attacker can not easily disable it.

A second constraint for LLSI is that the power supply can be modulated. Similar to the previous clock freezing detection, a circuit-level sensor is proposed to detect such modulations [72]. The sensor is based on a frequency-to-voltage converter with pre- and postprocessing. The concept is that the modulation causes a control signal to change, which a comparator can detect.

Recently, Roy *et al.* proposed a fully digital circuit-level countermeasure based on polymorphic sensor gates [73]. The basic idea is to use polymorphic logic gates that change their output based on circuit characteristics and environmental conditions. The proposed sensor circuit is designed so that the output of the polymorphic gate follows the supply voltage modulation signal, i.e., the modulation is converted to a digital signal. A comparator can detect changes in that signal. To also account for clock freezing, the authors integrate a detection using a ring oscillator-based clock generation circuit that counts the system clock cycles and raises an alarm if cycles are missing.

All these circuit-level countermeasures can be low-cost approaches to protect against single-trace attacks. However, if the attacker has access to a circuit-editing tool like a focused ion beam (FIB) to alter the clock signal, as discussed in [40], she could try to disconnect the internal sensors as well. Furthermore, fault injection methods could be used to disable the sensors [72]. Nevertheless, as hinted by Roy *et al.*, this would require prior reverse engineering and a more complex setup.

## 7.3 OPTICAL PROBING SIMULATIONS

The main drawback of the Trojan detection using LLSI presented in Chapter 6 is that it requires a golden image of the unmodified hardware. One possibility to remove this constraint could use simulations of LLSI. In the FA community, researchers have developed possibilities to simulate LVI [23] and LVP [74, 75]. Using simple geometrical simulations, which ignore most of the physical parameters of the silicon, Ravikumar *et al.* could simulate LVI measurements of CMOS inverters on a chip manufactured in a 20 nm FinFET technology [23]. The idea is to model the laser beam as Gaussian distribution and the layout as polygons. Due to the complementary nature of the CMOS technology, NMOS and PMOS transistors are modeled with an opposite phase, which is reflected by an opposite sign in the polygon's value. Furthermore, different amplification factors are used for NMOS and PMOS due to the different modulation behavior of the two types. Ravikumar *et al.* assumed that the only modulation contributed by the gate is over the active channel area [23]. Furthermore, since the sources of an inverter's transistors are connected to a constant voltage, the drain will dominate the modulation due to the large size compared to the gate.

The measurement can be simulated by calculating the convolution of the laser beam and the layout polygons. The Gaussian intensity profile p of the laser is modeled by

$$p(r) = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot e^{\frac{-(r)^2}{2\sigma^2}}, \tag{7.1}$$

where $r$ is the distance from the center of the beam and $\sigma$ is the standard deviation which can be calculated as $\sigma = 0.37\lambda/NA$ for a confocal microscope [23]. Fig. 7.1a illustrates this distribution for our lab's best-resolving lens. Given the above formula and procedure, we could replicate the simulation results from [23] by implementing the simulator in Python.

*Own simulations of EOFM*

Subsequently, we performed simulations for a single inverter of a chip manufactured in a planar CMOS 250 nm technology[1], for which we had access to the layout, see Fig. 7.1b. We extracted the drain and channel diffusion areas from the layout file and assigned different amplification factors for the NMOS and PMOS (3 and 1.33, respectively) [26, p. 105]. Then we performed EOFM measurements on the chip. As can be seen in Fig. 7.1c, the simulation (Fig. 7.1b) reflects the measurement to a certain degree. The NMOS (at the bottom) shows a stronger signal than the PMOS, and the size of the spots approximately matches the simulation. However, we observed additional patterns in the shape of partial concentric circles that appear, especially around the PMOS area. When changing the focus position, more such patterns show up. These patterns can be explained by the reflection of the light at different metal layers in the chip. All reflected signals are overlaid, and due to phase shifts occurring at the different interfaces, the signal is canceled out at certain positions, creating concentric curves around the transistor. At the auto-focus level and above, the PMOS is split into two spots that cancel out at the intersection. Using the EOFM phase measurement mode, we could confirm that the two spots carry a signal with opposite phases. One explanation of this phenomenon could be the interaction of the laser beam with the n-well-to-p-substrate interface in the PMOS transistor.

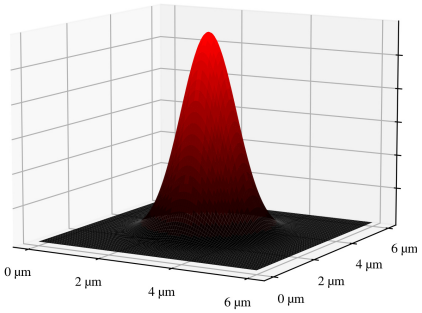
In [23], such interference effects were not observed. The reason could be that their setup's optical resolution and beam diameter are low compared with the transistor sizes. Therefore, observing such effects in detail on single transistors might not have been possible. Furthermore, they used a FinFET[2] technology, which might show fewer such effects.

*Comparing different lenses*

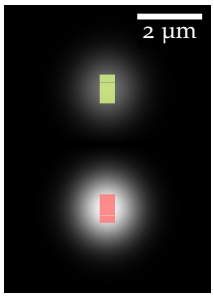
To better understand the relationship of the patterns concerning the laser spot size, we did experiments with another lens with a lower NA, see Fig. 7.1d. As expected, the results show a slightly larger diameter and a more blurry signal when using the poor-resolving lens. However,

1 The sample was kindly provided and supported by the Leibniz Institute for High Performance Microelectronics (IHP), Frankfurt (Oder), Germany.

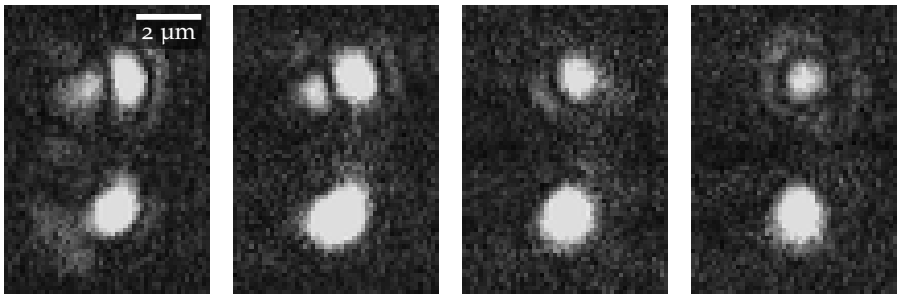
2 See Section 7.4.2 for more details on FinFETs.




(a) Laser intensity profile for simulation ($\lambda$ = 1300 nm, 0.71 NA)

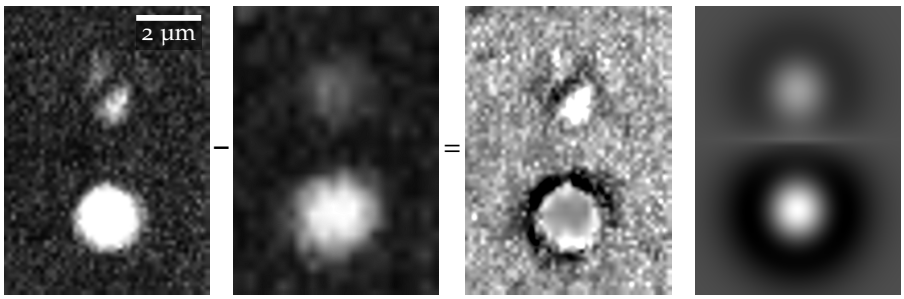



(b) Simulation for parameters shown in (a)




2 µm up | 0 µm (Auto focus) | 2 µm down | 4 µm down

(c) EOFM experimental result for different focus levels, 50× 0.71 NA lens, 4× scanner zoom




0.71 NA | 0.4 NA | Difference | Diff. Simulation

(d) EOFM experimental results for two lenses with different optical resolutions (50× 0.71 NA; 20× 0.4 NA), subtracted from each other, compared with the difference of the respective simulation results

Fig. 7.1: EOFM simulation and measurement results for a single inverter on a chip manufactured in a 250 nm technology, operated at a 1 MHz switching frequency. PMOS at the top, NMOS at the bottom. For the simulations, only the drain and channel areas were considered. All simulation and measurement images have the same scale.

when comparing the simulations for the respective NA values, there is a much wider spot for the 0.4 NA lens, whereas the spot size for the 0.71 NA lens coincides well with the measurements. This observation indicates that not the lens's optical properties alone but the entire optical path must be considered. One explanation could be that the confocal nature of the LSM, where the light passes a pinhole before entering the detector, can sharpen the beam and provides sharper spots than theoretically assumed.

*Better simulations with more knowledge about the device*

All the observed effects will have to be investigated in the future to provide more realistic simulations of optical probing. It should be noted that in [27], the author conducts experiments and simulations on a single transistor. The results explain some of the observed results, as the simulations take into account reflections and phase shifts at the different material interfaces in the device. However, the precise geometries and material properties must be known to perform such simulations. In practice, though, these parameters are typically not released by the foundries and are typically not available for researchers due to this reason.

*Why simulating LLSI is different*

During the previously described classical EOFM experiments, the transistors fully switch between on and off states. In the case of an inverter, the source of the transistor is always connected to a constant voltage, and only the gate is switched, leading to different voltage levels at the drain. Therefore, the simulator for classical EOFM assumes that a modulated signal will only occur over the channel and drain areas. For LLSI measurements, however, the transistors are not changing states, but only the supply voltage is modulated with a small amplitude. When a transistor is on (in the linear region), this modulation changes the concentration of free carriers in all transistor areas, as the drain voltage depends on the source voltage. Furthermore, the modulation can also influence the gate voltage because it might be connected to the output of another modulated transistor. On top of that, passive devices like decoupling capacitors will appear in the LLSI image because they are connected between VDD and GND, and their carrier concentration is also modulated.

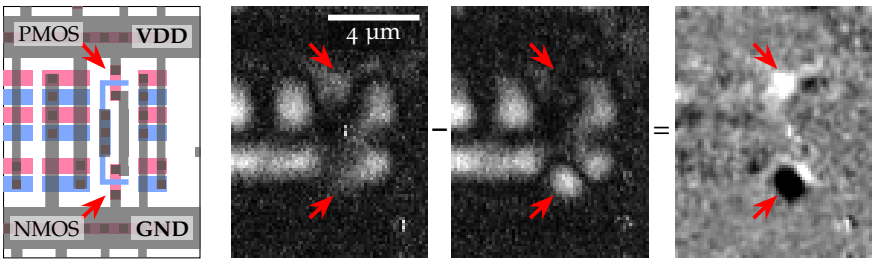



Fig. 7.2: LLSI measurement results for a single inverter on the 250 nm technology chip using the 50× lens, 4× scanner zoom, and a dwell time of 3.3 ms/px. Arrows shown for orientation.

Fig. 7.2 shows the layout of a single inverter with fill structures on its left and right sides. The PMOS is at the top and the NMOS at the bottom, whereas the diffusion areas are displayed in red, the polysilicon in blue, and metal connections and vias in gray. We took one LLSI measurement in each logic state of the inverter. It can be seen that there is also a signal in the area of the filler cells. Since these cells are connected to VDD and GND, they act as capacitors and, thus, are influenced by the modulation of VDD. When subtracting the two images, it becomes apparent that the signal in these areas is almost constant. One can see a big difference only in the transistors' regions, whereas the spot of the NMOS is more prominent than that of the PMOS. Since there are black and white areas (compared with the gray background) next to these two spots, the laser beam seems to be modulated at different regions of the transistors. From these observations, it can be concluded that the simulator for traditional EOFM must be adapted to consider other areas of the transistor, and even passive devices like capacitors need to be integrated to simulate LLSI measurements correctly.

## 7.4 FUTURE CHALLENGES

For the most recent chip generations, optical contactless fault isolation techniques are not reaching the desired resolution anymore and other specifics in chip design make failure analysis more challenging. Since we use FA tools for security analysis, most of the challenges also apply to security investigations. The following sections discuss the most prominent changes and resulting challenges for IC failure analysis.

### 7.4.1 *Limited Optical Resolution*

As introduced in Section 2.2.2, the optical resolution R depends on the light's wavelength and the optical system's NA with $R \sim \lambda/NA$. From this relationship, it can be concluded that the optical resolution can be increased by either reducing the wavelength or increasing the NA. The benefit of analyses in the NIR light spectrum is that no sample preparation in terms of backside thinning is required. However, since silicon is only highly transparent for wavelengths with photon energy smaller than the silicon bandgap ($\lambda > 1.1\,\mu m$), the wavelength for NIR light analysis is limited. The best resolution with NIR can be achieved with a wavelength of around 1064 nm, which requires a maximum remaining silicon thickness of around 50 µm to 100 µm due to the lower penetration depth for this wavelength (cf. Section 2.2) [76–78].

*Solid immersion lenses for NIR analysis*

Traditional microscope lenses are limited in their NA since the used light has to travel through air before entering and after leaving the silicon sample. Due to the refractive index of air of around 1, a lens with a maximum opening angle could only achieve an NA of at most

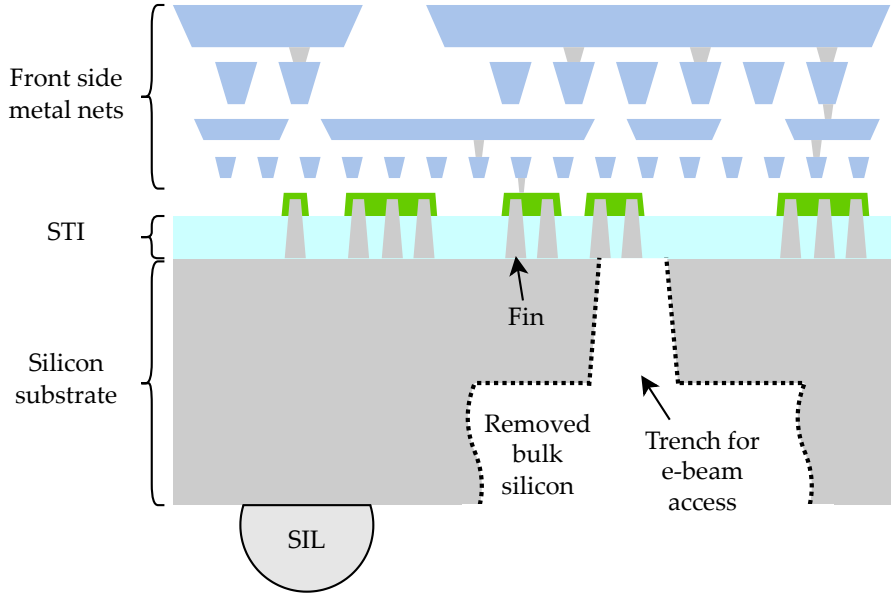



Fig. 7.3: Schematic of a chip with attached aplanatic SIL and trench for e-beam probing. Figure based on [83].

1. However, practical considerations limit the NA of modern lenses to around 0.85 [16]. To increase the NA, liquid immersion lenses (LIOs) have been used. The idea is to add oil on top of the silicon surface with a high refractive index. However, since no matching oil is available for silicon, only an NA of around 1.4 can be achieved [16, 76]. A higher NA can be accomplished by using a solid immersion lens (SIL), a lens in a hemispherical or aplanatic shape that is in contact with the chip backside, see Fig. 7.3. The maximum NA that can be achieved with such a lens is around 3.3 [79]. Consequently, NIR light analysis can be used down to a 10 nm technology node for fault isolation purposes [76].

*Visible light probing to improve resolution*

The second possibility to increase the resolution is to reduce the wavelength of the light used for optical probing. The so-called visible light probing uses light with wavelengths around 650 nm [80, 81]. Due to the opaqueness of silicon in this spectrum, increased sample preparation effort is required. The reported maximum remaining silicon thickness for visible light probing ranges from 10 µm [81] down to 1 µm [80]. Such global thinning can be challenging due to thermal effects that influence the surface topology of the silicon, which can reduce the effectiveness of a SIL [82]. Nevertheless, the highest possible resolution of optical probing can be achieved in combination with a SIL, see Fig. 7.4.

*Resolution requirements for FA*

For fault isolation purposes, the required resolution is connected to the technology node. While the node size itself does not give any insight into any actual length on the chip (cf. Section 2.1.2), the contacted gate pitch of a minimum-size transistor is the more relevant number. It also distinguishes how close neighboring nodes can be

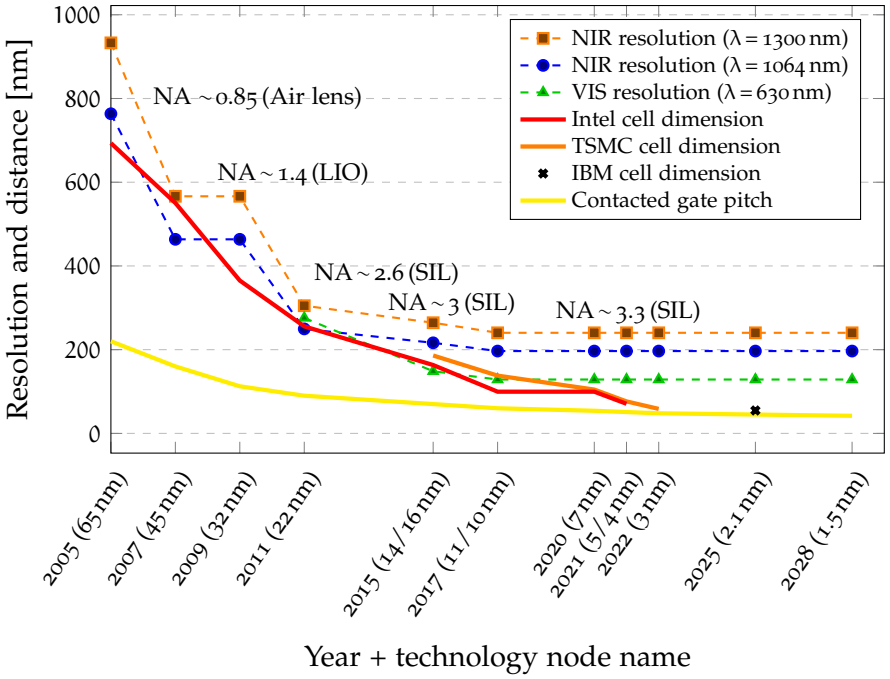



Fig. 7.4: Evolution of optical FA resolution over the years [76, 81, 84] compared with the inverse logic cell density (cell dimension) for Intel, TSMC, and IBM technologies [85, 86] and the minimum contacted gate pitch from the IRDS ground rules map for logic devices [13, 87, 88]. NA values of the lens generations are given for NIR analysis. VIS refers to visible light probing.

placed. As the pitch sizes are still shrinking, higher resolving analysis tools are required [76, 84]. Fig. 7.4 shows the evolution of optical FA techniques over time, compared with the average cell dimensions for Intel, TSMC, and IBM technology nodes. Since the wavelength and the numerical aperture have reached their limits for optical analysis, there is an increasing gap between the required and achieved resolution. This gap forces the FA community to think about new solutions because in FA, tiny defects that relate to a single transistor have to be localized.

*Relaxed resolution requirements for security applications*

When considering security applications, the resolution requirements are more relaxed since it is often unnecessary to resolve structures of minimum size on the chip [84]. For instance, data buses and bus drivers that carry sensitive data all over the chip are typically not realized with minimum-size transistors. In the publication presented in Chapter 6, we have shown that registers on a 20 nm technology FPGA can be probed with a 1.3 µm wavelength and a 0.71 NA lens, which corresponds to a resolution of R = 1117 nm (Equation 2.3). Furthermore, not the size of the transistor but the distance to nearby structures that could interfere with the beam are limiting the applicability of laser stimulation and optical probing [6, 84]. On top of that, unlike for FA applications where the exact localization of a fault is

essential, for security applications it is sufficient if the data-dependent signal of interest can be captured at any location. Nevertheless, also malicious applications will profit from higher resolutions, for instance, to extract sensitive information from minimum-size structures such as embedded memory arrays. The following paragraphs discuss an existing technique to overcome the resolution limits of optical NIR fault isolation.

*Beyond optical analysis: Electron-beam probing*

A higher resolution than with optical techniques can be achieved using electron-beam (e-beam) probing [89, 90]. E-beam probing includes the generation of scanning electron microscopy (SEM) images and the probing of waveforms from the chip. A high-energy electron beam (primary beam) probes the surface of the device of interest and produces secondary electrons that can be detected using a secondary electron detector. The number of detected electrons gives insights into the surface topography and the electrical field present at the locations of interest. Recent tools can achieve spot sizes down to a few tens of nanometers.

E-beam probing can be used to directly read signals from metal lines on the chip's front side. However, when individual transistors should be analyzed, backside access must be used. Furthermore, e-beam probing can only be performed with direct access to the STI level of the chip, i.e., with direct access to the transistors' drain, source, and channel regions. Therefore, an increased effort has to be put into sample preparation, see Fig. 7.3. The proposed procedure involves a mechanical thinning of the chip area of interest to less than 40 µm remaining silicon thickness [89, 91]. Then, a FIB or a laser is used to remove further substrate until the n-well level is reached. Finally, small openings to the STI level can be created and the e-beam prober can be used to extract logic states from the active area [92]. This procedure implies that the region of interest that should be probed needs to be known before chip preparation. For FA purposes, this is not a problem. For security applications, though, the chip's layout is often unknown. Consequently, optical techniques that do not require complex chip preparations should first be used to identify the areas that need to be accessed with higher resolution.

*Conclusion: threat will stay*

In summary, one can say that as long as smaller chip technologies are developed, there have to be FA techniques to debug them. Therefore, FA techniques will always pose a threat to ICs, no matter how small the structures on the chip will be. However, with an increased resolution of the setup, the effort for an attacker will become higher, for instance, in sample preparation. Nevertheless, an attacker might often not be interested in resolving the smallest structures on the chip and, thus, lower resolving setups are sufficient in many cases to extract secrets successfully.

### 7.4.2 *New Wafer Types and 3D Transistor Designs*

In order to allow further shrinkage of technology nodes, new wafer types and transistor designs have been developed that replace the classical planar transistors. In the process of shrinking the manufacturing technologies, the channel length of the transistors was reduced further and further. This results in so-called short-channel MOSFETs, where the channel length is approximately equal to the size of the depletion layer widths of the drain and source [93]. Multiple effects occur in the case of a short channel, such as threshold voltage roll-off and drain-induced barrier lowering. The threshold voltage is reduced because the depletion regions of the drain and source extend further into the channel area. In other words, drain and source areas help to make the channel more conductive by depleting the channel. Similarly, high drain voltages can depress the barrier between the source and channel, which also lowers the effect of the gate voltage on the current flow between the drain and the source. Another effect that occurs with a higher chance in short-channel transistors is gate depletion. Since the gate electrode is typically made of heavily doped polysilicon, high electric fields can deplete the gate, decreasing the gate capacitance and thus lowering the transistor's threshold voltage. Increasing the doping concentration and reducing the oxide thickness to counteract this effect have come to their limits. Consequently, there is a comeback of metal gates in combination with well-insulating (high-$\kappa$) dielectrics to increase the gate capacitance and, therefore, the gate's influence on the channel. In all cases, the reduced threshold voltage due to the short-channel effects leads to an increased leakage current flow. Consequently, the overall goal is to reduce leakage currents while still shrinking the area consumed by a transistor.

*Short-channel effects lead to increased leakage currents*

One approach is to add an insulating layer between the bulk silicon and the transistor or logic gate. The silicon material of the channel is a thin layer of fully depleted (undoped) silicon, meaning that it does not contain intrinsic charge carriers (free electrons or holes). The resulting wafer technology is called fully depleted silicon-on-insulator (FD-SOI) and allows much faster switching with less leakage current. The material used as an insulator is typically silicon dioxide ($SiO_2$) that is, with a refractive index n of around 1.5, transparent to NIR (for silicon, n is $\approx$ 3.5). Consequently, optical probing [94] and photon emission [95] experiments can still succeed without any changes. It should be noted that due to the relatively high manufacturing cost and the potentially high self-heating on high-performance chips (the insulator is a poor thermal conductor), the big manufacturers like TSMC and Intel have not yet adopted FD-SOI and still use bulk silicon wafers.

*Fully depleted silicon-on-insulator wafers*

The more widely used approach to allow further shrinking is the design of new transistor structures. Therefore, the next evolutionary

*FinFET transistors*

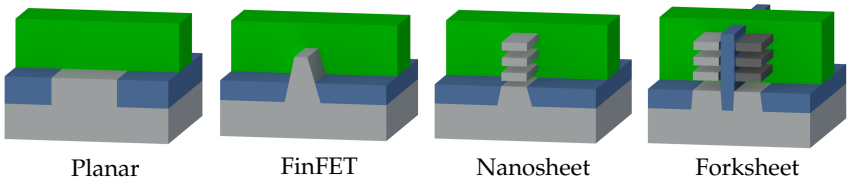
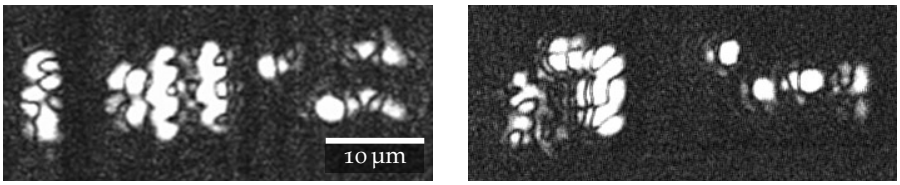



Fig. 7.5: Evolution of transistor structures. Gray: silicon, Blue: oxide, Green: gate. Note that these are simplified illustrations where, e.g., the gate insulation and the drain/source areas are omitted.




(a) Planar (b) FinFET

Fig. 7.6: Comparison of EOFM signal for a configurable logic block (CLB) implementing a register on a Xilinx Ultrascale FPGA (20 nm, planar) and a Xilinx UltraScale+ FPGA (16 nm, FinFET) with a dwell time of 1 ms/px, the 50× lens, and 4× additional zoom.

step after planar CMOS adopted by the industry was to implement the gate on three sides of the channel, making the drain and source areas appear as fins, see Fig. 7.5. Since the channel, drain, and source areas are directly connected to the bulk silicon, optical inspection from the chip's backside is still possible. It has been shown that FinFETs can be modeled analogously to planar transistors, and optical probing can be applied as before [96]. However, the signal can be more blurred due to the structure of the fins [97]. Nevertheless, due to the knowledge of planar technologies, the signal can be interpreted sufficiently to allow fault localization. TSMC integrated FinFETs starting from their 16 nm process node in 2015. We conducted measurements on FPGAs manufactured in this 16 nm technology and a 20 nm planar TSMC technology. Although the layout of the chips can be expected to be different, we can confirm that both chips show similar patterns with a mix of high-intensity spots and cancellations at the boundaries, see Fig. 7.6. Therefore, it can be concluded that FinFET devices are still vulnerable to laser-based logic state extraction techniques.

*Gate-all-around FETs*

Even FinFETs are reaching their limits regarding leakage current when shrinking the transistors further. Intel and TSMC have announced to move to gate-all-around (GAA) FETs starting from 2 nm or 3 nm nodes. The main idea is to implement a gate on all four sides of the channel and to split the channel into multiple so-called nanowires or nanosheets, see Fig. 7.5. Ganesh indicates that optical probing should still be able to succeed, though, with much more blurred signals [97]. In general, however, it might depend on the ac-

tual implementation of the transistors. Since a metal gate will cover the channel area from all four sides, the probing light will not be able to reach the channel's silicon. However, depending on the actual implementation, there might be a spacer between the gate and the drain/source regions that could allow optical access. Furthermore, the simulation of CMOS inverters in Section 7.3 has shown that next to the channel, the drain area is responsible for most of the signal-dependent modulation of the light. Since the drain and source areas can be expected to be still optically accessible from the bulk silicon, optical probing will still be possible. Laser stimulation can also be expected to work on such transistor structures, as the gate metal is a good thermal conductor that can transmit the induced heat into the channel region.

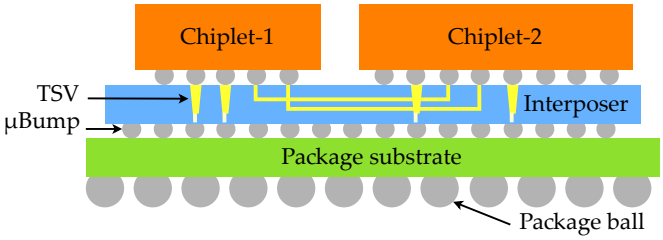
Another interesting development to further reduce the size of logic cells is to put the NMOS and PMOS transistors closer together. In this regard, so-called forksheet transistors have been proposed [98]. The idea is to use one shared gate structure for NMOS and PMOS nanosheets, see Fig. 7.5. The NMOS and PMOS transistors are either separated by a dielectric wall or put on top of each other. Especially the latter case could be challenging for optical probing applications, as the drain areas for NMOS and PMOS would then be stacked onto each other.

One can anyhow ask if optical techniques will be relevant in these technology nodes, as other challenges might already hinder optical access, as discussed in the remainder of this chapter.

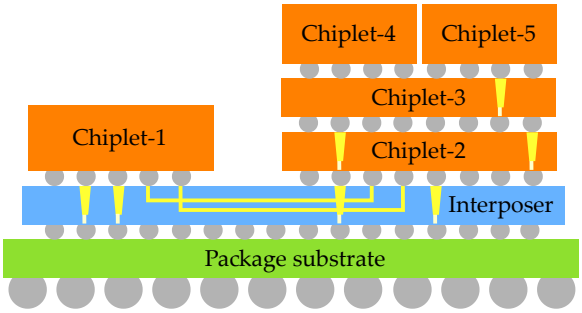
### 7.4.3 *3D Chip Stacking*

*Combining multiple stacked dies in one package*

For increasing the yield during the production of ICs and achieving higher design flexibility, chiplet-based designs have gained popularity during the past years. The idea is to split the functionality of a design into multiple functional circuit blocks, called chiplets, and combine multiple of them in one package [99]. For example, a base logic die can be combined with different compute, acceleration, memory, or radiofrequency chiplets to form a powerful SoC. Traditionally, the chiplets were placed next to each other on an interposer PCB that interconnects them, called 2D packaging. However, this connection type is not ideal regarding the latency and bandwidth of inter-chiplet communication. Chiplet-to-wafer bonding provides a better connection and integrates the chiplets by adding a silicon interposer, see Fig. 7.7a. This technique is used in recent packaging types, such as Intel's Foveros [100] and TSMC's CoWoS [101]. Note that both face-to-back and face-to-face assembly of the dies exist, meaning that the metal front side is in contact with the backside of the lower chip (interposer) or the two metal front sides are connected, respectively.




(a) Example for 2.5D packaging.




(b) Example for 3D packaging with face-to-back connected chiplets.

Fig. 7.7: Comparison of 2.5D and 3D chiplet packaging with TSVs and silicon interposer. Due to the many vendor-specific differences, this is not a complete picture of all existing types.

While these silicon interposer-based approaches are called 2.5D packaging, higher densities can be achieved with 3D chip stacking methods. Here multiple chiplets are stacked onto each other, see Fig. 7.7b. While wire bonding was used in the first place to interconnect the chips, 3D stacking based on through-silicon vias (TSVs) is used in the most recent packaging generations, such as TSMC's SoIC [102]. Furthermore, package-on-package technologies allow adding another packaged chip (e.g., a memory chip) on top of another package. Note that there are many different 3D stacking techniques that all do coexist.

*Fault isolation on 3D integrated chips*

While optical FA on 2.5D packages is still possible on the individual chiplets, 3D packages heavily complicate fault isolation. Only the top chiplet can be easily accessed because they typically face up with their backside. To access the lower chiplets, however, at least parts of the chips on top have to be removed. For FA purposes, this might be applicable because the engineers fully know the design and can open a trench at positions where the upper chips stay functional. An e-beam prober, see Section 7.4.1, could then be used to analyze the sample. For security purposes, however, the design is often unknown. Consequently, the upper chiplets will most likely be destroyed when trying to access a lower chiplet. Advanced techniques, such as X-ray might be applicable, see Section 7.4.4. Nevertheless, 3D integration can be seen as a natural countermeasure against attackers.

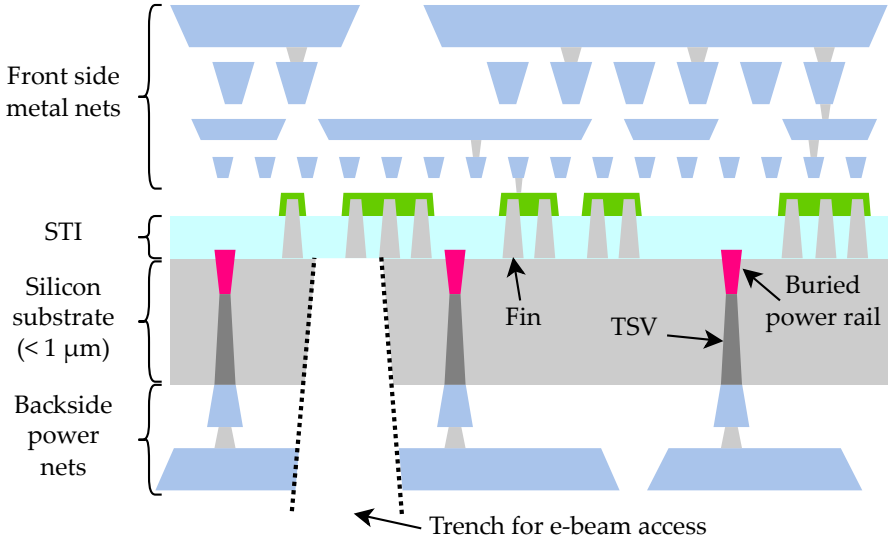



Fig. 7.8: Schematic of BPD with trench for e-beam probing. Figure based on [83].

### 7.4.4 *Backside Power Delivery Networks*

For future downscaling of technology nodes beyond 5 nm, some of the metal congestion from the chip front side will have to be removed. Among other measures, so-called buried power rails (BPRs) will be required to achieve further miniaturization [103]. The idea is to bury the power rails into the substrate area and remove two metal tracks for power rails from a standard cell's design. However, since the connection to the BPRs still has to be made from the chip's front side, the full benefits might only be reached by moving the power delivery network to the backside of the IC, see Fig. 7.8 [104]. This can be achieved by thinning down on the backside to a remaining silicon thickness below 1 µm and adding nano TSVs connecting to the BPRs. Then, metal layers are added to the backside of the chip. The industry will adopt the approach of backside power delivery (BPD) as, for instance, Intel has announced to integrate BPD into their 20A technology expected in volume production in 2024 [105].

*Challenging fault isolation on BPD chips*

For FA and security investigations, the existence of BPD means that the back surface of the chip is no longer accessible. Only small trenches through the power network can be opened to keep the chip operational during analysis. Therefore, placing a SIL for optical FA analyses might be impossible. As discussed above, e-beam probing is already a solution for FA on small technology nodes. Since it works through a small opening on the chip's backside, it might also be a suitable technique under the existence of BPD. However, if in the future even logic signals are moved to the backside metal plane, it might not be possible to access all locations of interest anymore. Another solution, which could enable analysis without opening a trench, is

micro X-ray fluorescence microscopy. Initial investigations have shown that the leakage current can be altered using X-ray [83]. This technique could be used similarly to laser stimulation, such as TLS, where a laser is used to locally influence the DUT for exhibiting internal device states and properties by a change in current consumption. However, this first study has shown that interpreting the obtained signals is not trivial. One of the main problems of X-ray analysis is also the device degradation, which might persist for multiple weeks or even permanently. Furthermore, the achieved beam diameter is between 15 µm and 20 µm, which is a much worse resolution than can be achieved with optical techniques (factor at least 15 µm/250 nm = 60). In summary, FA using X-ray microscopy is a potential solution that requires much more research to be applicable in daily operations.

# 8

# SUMMARY & CONCLUSION

This work has investigated threats and opportunities of laser-based logic state extraction from ICs. We used optical single-trace techniques known from chip FA. For most experiments, we used the PHEMOS-1000 LSM, capable of conducting laser stimulation and optical probing measurements.

In the first publication contained in this thesis, we investigated if theoretical models that ensure side-channel resistance still hold for optical attacks. In this regard, we implemented masked versions of AES with different numbers of shares on an FPGA. In case the memory locations of the individual key shares are known to the adversary, she can directly extract the key using LLSI. However, even if the memory locations of the key shares are unknown, capturing LLSI images from multiple clock cycles and applying a SAT solver can reveal the key. These results prove that the t-probing security model does not consider the presence of optical single-trace attacks and can not protect against powerful adversaries with FA equipment. Consequently, future security models must consider optical probing techniques to reflect the real threat of hardware attacks.

In the second publication, we studied how computer-aided extraction of secrets can be accomplished and how practical automated reverse-engineering approaches are. We captured logic state images from three different devices with randomly chosen keys programmed in their memories. Then, we used these images to train CNNs on the key. The results show that we can successfully extract all key bits when presenting an image with an unknown key to the trained network. Therefore, the automatic extraction of secrets is a valid threat that should be kept in mind by chip vendors and users. Our study shows that less expertise and manual effort than commonly assumed is required to extract secret information from an IC.

In the third publication, we set up and tested a low-cost setup for laser stimulation attacks to show that tools are available for lower prices than previously expected. We used a bench originally designed for laser fault injection that is available at a price around ten times lower than an FA microscope. The impact of our work also has manifested in the fact that the vendor of the laser fault injection station now advertises and offers a laser stimulation option for customers. Consequently, one can assume that malicious parties can conduct optical hardware attacks with a much lower budget than previously expected.

Finally, in the fourth publication, we investigated how logic state imaging techniques can detect tampering with the hardware design. Specifically, we have shown how logic state imaging techniques can uncover HTs on FPGAs. We investigated this capability on recent flash- and SRAM-based FPGAs by acquiring LLSI images from different logic configurations and Trojan benchmarks. This work has shown that laser-based logic state extraction is not only an attack tool but offers the opportunity to improve the security of ICs.

The subsequent discussion chapter has elaborated on additional aspects that might be important for future research. First, by introducing the term active side-channel analysis, we create a better understanding of future techniques that stimulate the DUT to force side-channel leakage. Second, analyzing new countermeasure approaches against optical probing attacks has shown interesting new directions. Furthermore, the simulation of optical probing techniques can be a game changer in predicting the resistance to optical attacks already in the design phase. Finally, looking at current and future challenges in the IC design and failure analysis gave insights into upcoming trends that influence security investigations using optical techniques. New transistor designs, 3D chip stacking techniques, and power delivery networks on the chip backside will impact the applicability of optical attacks as used in this work.

To conclude, this work has shown that laser-based logic state extraction poses a threat to secret data processed by integrated circuits: Security models do not hold what might be expected, automation and machine learning can even replace human expertise in extracting data, and setups are available much cheaper than anticipated. Nevertheless, optical single-trace techniques can also improve the security of a system by detecting malicious design modifications. Since new generations of failure analysis techniques are arising, the threat of optical side-channel attacks will remain with future chip generations. Therefore, there is no reason to rest as the necessity to protect chips against the presented and future attacks will persist.

# BIBLIOGRAPHY


[1] A. Hope. "Ukraine Warns of Massive Russian Cyber Attacks on the Country's and Allies' Critical Infrastructure," CPO Magazine. (Oct. 7, 2022), [Online]. Available: https://www.cpomagazine.com/cyber-security/ukraine-warns-of-massive-russian-cyber-attacks-on-the-countrys-and-allies-critical-infrastructure/ (visited on 10/13/2022) (cit. on p. 2).

[2] J. Delcker. "Critical cybersecurity infrastructure: How can countries protect themselves against cyberattacks?" Deutschen Welle. (Sep. 30, 2022), [Online]. Available: https://www.dw.com/en/critical-cybersecurity-infrastructure-how-can-countries-protect-themselves-against-cyberattacks/a-63295176 (visited on 10/13/2022) (cit. on p. 2).

[3] C. Todd Lopez. "DOD Adopts 'Zero Trust' Approach to Buying Microelectronics," U.S. Department of Defense. (May 19, 2020), [Online]. Available: https://www.defense.gov/News/News-Stories/Article/Article/2192120/dod-adopts-zero-trust-approach-to-buying-microelectronics/ (visited on 10/13/2022) (cit. on p. 3).

[4] D. Nedospasov, J. P. Seifert, C. Helfmeier, and C. Boit, "Invasive PUF Analysis," in *Proceedings of the 2013 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)*, IEEE, Aug. 2013, pp. 30–38. DOI: 10.1109/FDTC.2013.19 (cit. on pp. 3, 17, 18).

[5] H. Lohrke, S. Tajik, C. Boit, and J.-P. Seifert, "No Place to Hide: Contactless Probing of Secret Data on FPGAs," in *Cryptographic Hardware and Embedded Systems – CHES 2016*, ser. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, Aug. 2016, pp. 147–167. DOI: 10.1007/978-3-662-53140-2_8 (cit. on p. 3).

[6] S. Tajik, H. Lohrke, J.-P. Seifert, and C. Boit, "On the Power of Optical Contactless Probing: Attacking Bitstream Encryption of FPGAs," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, ACM, 2017, pp. 1661–1674. DOI: 10.1145/3133956.3134039 (cit. on pp. 3, 76, 101).

[7] H. Lohrke, S. Tajik, T. Krachenfels, C. Boit, and J.-P. Seifert, "Key Extraction Using Thermal Laser Stimulation: A Case Study on Xilinx Ultrascale FPGAs," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2018, no. 3, pp. 573–595, Aug. 2018. DOI: 10.13154/tches.v2018.i3.573-595 (cit. on pp. 3, 18, 43, 76).

[8] S. Chef, C. Chua, J. Tay, and C. Gan, "Quantitative Study of Photoelectric Laser Stimulation for Logic State Imaging in Embedded SRAM," in *ISTFA 2021: Conference Proceedings from the 47th International Symposium for Testing and Failure Analysis*, Phoenix, Arizona, USA, Oct. 2021, pp. 154–162. DOI: 10.31399/asm.cp.istfa2021p0154 (cit. on p. 3).

[9] R. J. Baker, *CMOS: Circuit Design, Layout, and Simulation* (IEEE Press Series on Microelectronic Systems), 3rd ed. John Wiley & Sons, 2011, ISBN: 978-0-470-89117-9 (cit. on pp. 5, 9, 10).

[10] H.-M. Tong, Y.-S. Lai, and C. P. Wong, *Advanced Flip Chip Packaging*. Boston, MA: Springer US, 2013. DOI: 10.1007/978-1-4419-5768-9 (cit. on p. 6).

[11] J. M. Rabaey, A. P. Chandrakasan, and B. Nikolić, *Digital Integrated Circuits: A Design Perspective*. Pearson Education, Incorporated, 2003 (cit. on p. 9).

[12] N. H. E. Weste and K. Eshraghian, *Principles of CMOS VLSI Design: A Systems Perspective*, 2nd ed. Addison-Wesley, 1993 (cit. on p. 10).

[13] IRDS, "International Roadmap for Devices and Systems, 2022 Update, More Moore," IEEE, 2022. [Online]. Available: https://irds.ieee.org/images/files/pdf/2022/2022IRDS_MM.pdf (visited on 02/15/2023) (cit. on pp. 11, 101).

[14] "Interaction of Light with Matter," in *Semiconductor Optics*, Berlin, Heidelberg: Springer, 2007, pp. 37–72. DOI: 10.1007/978-3-540-38347-5_3 (cit. on p. 13).

[15] C. F. Klingshirn, *Semiconductor Optics* (Graduate Texts in Physics). Berlin, Heidelberg: Springer, 2012. DOI: 10.1007/978-3-642-28362-8 (cit. on p. 13).

[16] H. Lohrke, "Laser-Based Attacks on Secure Integrated Circuits," 2019. [Online]. Available: https://depositonce.tu-berlin.de/handle/11303/10277 (visited on 12/11/2019) (cit. on pp. 13, 63, 100).

[17] M. A. Green, "Self-consistent optical parameters of intrinsic silicon at 300K including temperature coefficients," *Solar Energy Materials and Solar Cells*, vol. 92, no. 11, pp. 1305–1310, Nov. 2008. DOI: 10.1016/j.solmat.2008.06.009 (cit. on pp. 13, 14).

[18] R. Soref and B. Bennett, "Electrooptical effects in silicon," *IEEE Journal of Quantum Electronics*, vol. 23, no. 1, pp. 123–129, Jan. 1987. DOI: 10.1109/JQE.1987.1073206 (cit. on p. 14).

[19] S. Basu, B. J. Lee, and Z. Zhang, "Infrared Radiative Properties of Heavily Doped Silicon at Room Temperature," *Journal of Heat Transfer-transactions of The Asme*, vol. 132, Feb. 1, 2010. DOI: 10.1115/1.4000171 (cit. on p. 14).

[20] W. J. Alford, R. D. VanderNeut, and V. J. Zaleckas, "Laser scanning microscopy," *Proceedings of the IEEE*, vol. 70, no. 6, pp. 641–651, Jun. 1982. DOI: 10.1109/PROC.1982.12362 (cit. on p. 14).

[21] R. H. Webb, "Confocal optical microscopy," *Reports on Progress in Physics*, vol. 59, no. 3, p. 427, 1996. DOI: 10.1088/0034-4885/59/3/003 (cit. on p. 15).

[22] S. Parvin, T. Krachenfels, S. Tajik, J.-P. Seifert, F. S. Torres, and R. Drechsler, "Toward Optical Probing Resistant Circuits: A Comparison of Logic Styles and Circuit Design Techniques," in *2022 27th Asia and South Pacific Design Automation Conference (ASP-DAC)*, Jan. 2022, pp. 429–435. DOI: 10.1109/ASP-DAC52403.2022.9712518 (cit. on pp. 15, 94).

[23] V. Ravikumar, G. Lim, J. Chin, K. Pey, and J. Yang, "Understanding spatial resolution of laser voltage imaging," *Microelectronics Reliability*, vol. 88–90, pp. 255–261, Sep. 2018. DOI: 10.1016/j.microrel.2018.07.051 (cit. on pp. 15, 16, 95, 96).

[24] C. Boit, C. Helfmeier, D. Nedospasov, and A. Fox, "Ultra high precision circuit diagnosis through seebeck generation and charge monitoring," in *Proceedings of the 20th IEEE International Symposium on the Physical and Failure Analysis of Integrated Circuits (IPFA)*, IEEE, Jul. 2013, pp. 17–21. DOI: 10.1109/IPFA.2013.6599119 (cit. on pp. 17, 18).

[25] T. H. Geballe and G. W. Hull, "Seebeck Effect in Silicon," *Physical Review*, vol. 98, no. 4, pp. 940–947, May 15, 1955. DOI: 10.1103/PhysRev.98.940 (cit. on p. 17).

[26] U. Kindereit, G. Woods, J. Tian, U. Kerst, and C. Boit, "Investigation of Laser Voltage Probing Signals in CMOS Transistors," in *2007 IEEE International Reliability Physics Symposium Proceedings. 45th Annual*, Apr. 2007, pp. 526–533. DOI: 10.1109/RELPHY.2007.369946 (cit. on pp. 19, 96).

[27] U. Kindereit, "Investigation of laser-beam modulations induced by the operation of electronic devices," May 8, 2009. [Online]. Available: https://depositonce.tu-berlin.de//handle/11303/2440 (visited on 06/05/2018) (cit. on pp. 19, 98).

[28] W. M. Yee, M. Paniccia, T. Eiles, and V. Rao, "Laser voltage probe (LVP): A novel optical probing technology for flip-chip packaged microprocessors," in *Proceedings of the 1999 7th International Symposium on the Physical and Failure Analysis of Integrated Circuits (IPFA)*, 1999, pp. 15–20. DOI: 10.1109/IPFA.1999.791222 (cit. on p. 20).

[29] Y. S. Ng, T. Lundquist, D. Skvortsov, J. Liao, S. Kasapi, and H. Marks, "Laser voltage imaging: A new perspective of laser voltage probing," in *Proceedings of the 36th International Symposium for Testing and Failure Analysis (ISTFA)*, ASM International, 2010, pp. 5–13 (cit. on p. 20).

[30] B. Niu, G. M. E. Khoo, Y.-C. S. Chen, F. Chapman, D. Bockelman, and T. Tong, "Laser Logic State Imaging (LLSI)," in *ISTFA 2014: Conference Proceedings from the 40th International Symposium for Testing and Failure Analysis*, ASM International, 2014, pp. 65–72. DOI: 10.31399/asm.cp.istfa2014p0065 (cit. on pp. 20, 21).

[31] P. Kocher, J. Jaffe, and B. Jun, "Differential Power Analysis," in *Advances in Cryptology — CRYPTO' 99*, ser. Lecture Notes in Computer Science, Berlin, Heidelberg: Springer, 1999, pp. 388–397. DOI: 10.1007/3-540-48405-1_25 (cit. on p. 23).

[32] C. Clavier, J.-S. Coron, and N. Dabbous, "Differential Power Analysis in the Presence of Hardware Countermeasures," in *Cryptographic Hardware and Embedded Systems — CHES 2000*, ser. Lecture Notes in Computer Science, Berlin, Heidelberg: Springer, 2000, pp. 252–263. DOI: 10.1007/3-540-44499-8_20 (cit. on p. 23).

[33] T. Popp, S. Mangard, and E. Oswald, "Power Analysis Attacks and Countermeasures," *IEEE Design & Test of Computers*, vol. 24, no. 6, pp. 535–543, Nov. 2007. DOI: 10.1109/MDT.2007.200 (cit. on p. 23).

[34] A. Covic, F. Ganji, and D. Forte, *Circuit Masking: From Theory to Standardization, A Comprehensive Survey for Hardware Security Researchers and Practitioners*, Jun. 29, 2021. DOI: 10.48550/arXiv.2106.12714. arXiv: 2106.12714 [cs] (cit. on p. 23).

[35] Y. Ishai, A. Sahai, and D. Wagner, "Private Circuits: Securing Hardware against Probing Attacks," in *Advances in Cryptology - CRYPTO 2003*, vol. 2729, Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 463–481. DOI: 10.1007/978-3-540-45146-4_27 (cit. on p. 23).

[36] National Institute of Standards and Technology (NIST). "Masked Circuits for Block-Ciphers," CSRC | NIST. (May 12, 2021), [Online]. Available: https://csrc.nist.gov/projects/masked-circuits (visited on 10/20/2022) (cit. on p. 23).

[37] Common Criteria. "Application of Attack Potential to Smartcards (CCDB-2013-05-002)," Common Criteria. (May 2013), [Online]. Available: https://www.commoncriteriaportal.org/files/supdocs/CCDB-2013-05-002.pdf (visited on 10/20/2022) (cit. on p. 24).

[38] A. Duc, S. Dziembowski, and S. Faust, "Unifying Leakage Models: From Probing Attacks to Noisy Leakage.," in *Advances in Cryptology – EUROCRYPT 2014*, ser. Lecture Notes in Computer Science, Berlin, Heidelberg: Springer, 2014, pp. 423–440. DOI: 10.1007/978-3-642-55220-5_24 (cit. on p. 24).

[39] R. Specht, V. Immler, F. Unterstein, J. Heyszl, and G. Sig, "Dividing the threshold: Multi-probe localized EM analysis on threshold implementations," in *2018 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, Washington, DC: IEEE, Apr. 2018, pp. 33–40. DOI: 10.1109/HST.2018.8383888 (cit. on p. 24).

[40] T. Krachenfels, F. Ganji, A. Moradi, S. Tajik, and J.-P. Seifert, "Real-World Snapshots vs. Theory: Questioning the t-Probing Security Model," in *2021 IEEE Symposium on Security and Privacy (SP)*, IEEE Computer Society, 2021, pp. 1955–1971. DOI: 10.1109/SP40001.2021.00029 (cit. on pp. 24, 95).

[41] T. Krachenfels, "Laser logic state images of masked AES implementations from registers on a Cyclone IV FPGA," Sep. 2020. DOI: 10.14279/depositonce-10440 (cit. on p. 24).

[42] L. Masure, C. Dumas, and E. Prouff, "A Comprehensive Study of Deep Learning for Side-Channel Analysis," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pp. 348–375, 2020. DOI: 10.13154/tches.v2020.i1.348-375 (cit. on p. 43).

[43] S. Picek, G. Perin, L. Mariot, L. Wu, and L. Batina, "SoK: Deep Learning-based Physical Side-channel Analysis," *ACM Computing Surveys*, Oct. 28, 2022. DOI: 10.1145/3569577 (cit. on p. 43).

[44] W. Shan, S. Zhang, J. Xu, M. Lu, L. Shi, and J. Yang, "Machine Learning Assisted Side-Channel-Attack Countermeasure and Its Application on a 28-nm AES Circuit," *IEEE Journal of Solid-State Circuits*, vol. 55, no. 3, pp. 794–804, Mar. 2020. DOI: 10.1109/JSSC.2019.2953855 (cit. on p. 43).

[45] R.-R. Shrivastwa, S. Guilley, and J.-L. Danger, "Multi-source Fault Injection Detection Using Machine Learning and Sensor Fusion," in *Security and Privacy*, ser. Communications in Computer and Information Science, Cham: Springer International Publishing, 2021, pp. 93–107. DOI: 10.1007/978-3-030-90553-8_7 (cit. on p. 43).

[46] T. Krachenfels, T. Kiyan, S. Tajik, and J.-P. Seifert, "Automatic Extraction of Secrets from the Transistor Jungle using Laser-Assisted Side-Channel Attacks," in *30th USENIX Security Symposium (USENIX Security 21)*, USENIX Association, Aug. 2021, pp. 627–644, ISBN: 978-1-939133-24-3 (cit. on p. 44).

[47] T. Krachenfels, T. Kiyan, S. Tajik, and J.-P. Seifert, "Images from on-chip memories captured using the laser-assisted side-channel techniques LLSI and TLS," Feb. 2021. DOI: 10.14279/depositonce-11354 (cit. on p. 44).

[48] Hamamatsu Photonics K.K. "PHEMOS-1000 Emission microscope C11222-16." (2022), [Online]. Available: https://www.hamamatsu.com/us/en/product/semiconductor-manufacturing-support-systems/failure-analysis-system/C11222-16.html (visited on 10/28/2022) (cit. on p. 63).

[49] SEMICAPS Pte Ltd. "SEMICAPS 1100 Upright Analytical System." (2019), [Online]. Available: https://semicaps.com/wp-content/uploads/2019/09/Product_S1100.pdf (visited on 10/28/2022) (cit. on p. 63).

[50] Thermo Fisher. "Meridian 4 System." (2022), [Online]. Available: https://www.thermofisher.com/de/en/home/electron-microscopy/products/electrical-failure-analysis-systems/meridian-iv.html (visited on 11/14/2022) (cit. on p. 63).

[51] Riscure. "Laser Station 2." (2022), [Online]. Available: https://www.riscure.com/products/laser-station-2/ (visited on 11/14/2022) (cit. on p. 63).

[52] ALPhANOV. "Single laser fault injection - S-LMS." (2022), [Online]. Available: https://www.alphanov.com/en/products-services/single-laser-fault-injection (visited on 11/14/2022) (cit. on p. 63).

[53] T. Krachenfels, H. Lohrke, J.-P. Seifert, E. Dietz, S. Frohmann, and H.-W. Hübers, "Evaluation of Low-Cost Thermal Laser Stimulation for Data Extraction and Key Readout," *Journal of Hardware and Systems Security*, vol. 4, no. 1, pp. 24–33, Mar. 2020. DOI: 10.1007/s41635-019-00083-9 (cit. on p. 64).

[54] J. Robertson and M. Riley, "China Used a Tiny Chip in a Hack That Infiltrated U.S. Companies," *Bloomberg.com*, Oct. 4, 2018. [Online]. Available: https://www.bloomberg.com/news/features/2018-10-04/the-big-hack-how-china-used-a-tiny-chip-to-infiltrate-america-s-top-companies (visited on 10/31/2022) (cit. on p. 75).

[55] D. R. Collins, "Trust in Integrated Circuits," Defense Advanced Research Projects Agency Arlington Va Microsystems Technology Office, 2008. [Online]. Available: https://apps.dtic.mil/sti/citations/ADA482032 (visited on 10/01/2022) (cit. on p. 75).

[56] M. Xue, C. Gu, W. Liu, S. Yu, and M. O'Neill, "Ten years of hardware Trojans: A survey from the attacker's perspective," *IET Computers & Digital Techniques*, vol. 14, no. 6, pp. 231–246, 2020. DOI: 10.1049/iet-cdt.2020.0041 (cit. on p. 75).

[57] N. Asadizanjani, S. Shahbazmohamadi, M. Tehranipoor, and D. Forte, "Non-Destructive PCB Reverse Engineering Using X-Ray Micro Computed Tomography," presented at the ISTFA 2015, ASM International, Nov. 1, 2015, pp. 164–172. DOI: 10.31399/asm.cp.istfa2015p0164. (visited on 05/04/2023) (cit. on p. 75).

[58] H. Pearce, V. R. Surabhi, P. Krishnamurthy, J. Trujillo, R. Karri, and F. Khorrami, "Detecting Hardware Trojans in PCBs Using Side Channel Loopbacks," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 30, no. 7, pp. 926–937, Jul. 2022. DOI: 10.1109/TVLSI.2022.3171174 (cit. on p. 75).

[59] T. Mosavirik, P. Schaumont, and S. Tajik, "ImpedanceVerif: On-Chip Impedance Sensing for System-Level Tampering Detection," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pp. 301–325, 2023. DOI: 10.46586/tches.v2023.i1.301-325 (cit. on pp. 75, 93).

[60] E. Sarkar and M. Maniatakos, "On automating delayered IC analysis for hardware IP protection," in *Proceedings of the International Conference on Omni-Layer Intelligent Systems*, ser. COINS '19, New York, NY, USA: Association for Computing Machinery, May 5, 2019, pp. 205–210. DOI: 10.1145/3312614.3312656 (cit. on p. 75).

[61] L. N. Nguyen and A. Zajic, "A Novel Golden-Chip-Free Clustering Technique Using Backscattering Side Channel for Hardware Trojan Detection," *2020 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, 2020. DOI: 10.1109/HOST45689.2020.9300127 (cit. on p. 75).

[62] A. Stern, D. Mehta, S. Tajik, F. Farahmandi, and M. Tehranipoor, "SPARTA: A Laser Probing Approach for Trojan Detection," in *2020 IEEE International Test Conference (ITC)*, Nov. 2020. DOI: 10.1109/ITC44778.2020.9325222 (cit. on p. 75).

[63] M. Ender, A. Moradi, and C. Paar, "The Unpatchable Silicon: A Full Break of the Bitstream Encryption of Xilinx 7-Series FPGAs," *29th USENIX Security Symposium (USENIX Security 20)*, pp. 1803–1819, Aug. 2020, ISSN: 978-1-939133-17-5. [Online]. Available: https://www.usenix.org/conference/usenixsecurity20/presentation/ender (cit. on p. 76).

[64] T. Krachenfels, J.-P. Seifert, and S. Tajik, "Trojan Awakener: Detecting Dormant Malicious Hardware Using Laser Logic State Imaging," in *Proceedings of the 5th Workshop on Attacks and Solutions in Hardware Security*, ser. ASHES '21, New York, NY, USA: Association for Computing Machinery, Nov. 2021, pp. 17–27. DOI: 10.1145/3474376.3487282 (cit. on p. 76).

[65] T. Krachenfels, J.-P. Seifert, and S. Tajik, "Trojan awakener: Detecting dormant malicious hardware using laser logic state imaging (extended version)," *Journal of Cryptographic Engineering*, May 2023. DOI: 10.1007/s13389-023-00323-3 (cit. on p. 76).

[66] L. N. Nguyen, C.-L. Cheng, M. Prvulovic, and A. Zajic, "Creating a Backscattering Side Channel to Enable Detection of Dormant Hardware Trojans," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 27, no. 7, pp. 1561–1574, Jul. 2019. DOI: 10.1109/TVLSI.2019.2906547 (cit. on p. 93).

[67] T. Mosavirik, F. Ganji, P. Schaumont, and S. Tajik, "ScatterVerif: Verification of Electronic Boards Using Reflection Response of Power Distribution Network," *ACM Journal on Emerging Technologies in Computing Systems*, vol. 18, no. 4, 65:1–65:24, Oct. 13, 2022. DOI: 10.1145/3513087 (cit. on p. 93).

[68] S. K. Monfared, T. Mosavirik, and S. Tajik. "LeakyOhm: Secret Bits Extraction using Impedance Analysis." (2023), [Online]. Available: https://eprint.iacr.org/2023/693 (visited on 05/21/2023), preprint (cit. on p. 93).

[69] J. Couch, N. Whewell, A. Monica, and S. Papadakis, "Direct read of idle block RAM from FPGAs utilizing photon emission microscopy," in *2018 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, Apr. 2018, pp. 41–48. DOI: 10.1109/HST.2018.8383889 (cit. on p. 93).

[70] E. Amini, R. Muydinov, B. Szyszka, and C. Boit, "Backside Protection Structure for Security Sensitive ICs," in *ISTFA 2017: Conference Proceedings from the 43rd International Symposium for Testing and Failure Analysis*, ASM International, 2017, pp. 279–284. DOI: 10.31399/asm.cp.istfa2017p0279 (cit. on p. 94).

[71] M. T. Rahman, N. F. Dipu, D. Mehta, S. Tajik, M. Tehranipoor, and N. Asadizanjani, "CONCEALING-Gate: Optical Contactless Probing Resilient Design," *ACM Journal on Emerging Technologies in Computing Systems*, vol. 17, no. 3, 39:1–39:25, Jun. 30, 2021. DOI: 10.1145/3446998 (cit. on p. 94).

[72] S. Roy, T. Farheen, S. Tajik, and D. Forte, "Self-timed Sensors for Detecting Static Optical Side Channel Attacks," in *2022 23rd International Symposium on Quality Electronic Design (ISQED)*, Apr. 2022. DOI: 10.1109/ISQED54688.2022.9806217 (cit. on pp. 94, 95).

[73] S. Roy, S. Tajik, and D. Forte, "Polymorphic Sensor to Detect Laser Logic State Imaging Attack," in *2023 24th International Symposium on Quality Electronic Design (ISQED)*, Apr. 2023. DOI: 10.1109/ISQED57927.2023.10129304 (cit. on p. 95).

[74] E. Abuayob, E. Nisenboim, A. Raveh, B. Niu, and T. Tong, "Complex Waveform Analysis for Advanced CMOS ICs," presented at the ISTFA 2016, Fort Worth, Texas, USA, Nov. 1, 2016, pp. 68–75. DOI: 10.31399/asm.cp.istfa2016p0068 (cit. on p. 95).

[75] R. Krishnan, S. Xuan, L. Gabriel, T. Abel, L. Winson, G. Ranganathan, P. Angelina, and C. Meng, "Pattern Search Automation for Combinational Logic Analysis," in *ISTFA 2018: Conference Proceedings from the 44th International Symposium for Testing and Failure Analysis*, Phoenix, Arizona, USA: ASM International, 2018, pp. 86–92. DOI: 10.31399/asm.cp.istfa2018p0086 (cit. on p. 95).

[76] M. von Haartman, S. Rahman, S. Ganguly, J. Verma, A. Umair, and T. Deborde, "Optical Fault Isolation and Nanoprobing Techniques for the 10 nm Technology Node and Beyond," in *ISTFA 2015: Conference Proceedings from the 41st International Symposium for Testing and Failure Analysis*, Portland, Oregon, USA: ASM International, 2015, pp. 52–56. DOI: 10.31399/asm.cp.istfa2015p0052 (cit. on pp. 99–101).

[77] V.-K. Ravikumar, A. Phoa, P. Sabbineni, D. Skvortsov, and T. Lundquist, "Continuous-Wave 1064 nm Laser for Laser Voltage Imaging and Probing Applications," in *ISTFA 2014: Conference Proceedings from the 40th International Symposium for Testing and Failure Analysis*, ASM International, Nov. 2014, pp. 335–339. DOI: 10.31399/asm.cp.istfa2014p0335 (cit. on p. 99).

[78] G. Ranganathan, V.-K. Ravikumar, A. Phoa, and C.-W. Teo, "Timing analysis case studies using 1064 nm Continuous Wave laser for Fault Isolation on scan failures," in *2015 IEEE 22nd International Symposium on the Physical and Failure Analysis of Integrated Circuits*, Jun. 2015, pp. 13–16. DOI: 10.1109/IPFA.2015.7224320 (cit. on p. 99).

[79] K. Agarwal, R. Chen, L. S. Koh, C. J. R. Sheppard, and X. Chen, "Crossing the Resolution Limit in Near-Infrared Imaging of Silicon Chips: Targeting 10-nm Node Technology," *Physical Review X*, vol. 5, no. 2, p. 021 014, May 6, 2015. DOI: 10.1103/PhysRevX.5.021014 (cit. on p. 100).

[80] J. Beutler, V. C. Hodges, J. J. Clement, J. Stevens, E. I. Cole, S. Silverman, and R. Chivas, "Visible Light LVP on Bulk Silicon Devices," in *ISTFA 2015: Conference Proceedings from the 41st International Symposium for Testing and Failure Analysis*, Portland, Oregon, USA: ASM International, Nov. 1, 2015, pp. 6–13. DOI: 10.31399/asm.cp.istfa2015p0006 (cit. on p. 100).

[81] H. Lohrke, P. Scholz, A. Beyreuther, C. Boit, E. Uhlmann, *et al.*, "Contactless Fault Isolation for FinFET Technologies with Visible Light and GaP SIL," in *ISTFA 2016: Conference Proceedings from the 42nd International Symposium for Testing and Failure Analysis*, Fort Worth, Texas, USA: ASM International, Nov. 1, 2016, pp. 19–26. DOI: 10.31399/asm.cp.istfa2016p0019 (cit. on pp. 100, 101).

[82] J. Li, E. Halteh, J. Elliott, H. L. Marks, and C. Richardson, "Thermal Exposure Effects of Backside Thinned Flip-Chip Device on Visible Light Probing," in *ISTFA 2018: Conference Proceedings from the 44th International Symposium for Testing and Failure Analysis*, Phoenix, Arizona, USA: ASM International, Nov. 1, 2018, pp. 191–195. DOI: 10.31399/asm.cp.istfa2018p0191 (cit. on p. 100).

[83] W. Lo, P. Gupta, R. Venkatesh, R. Schlangen, H. Marks, B. Cory, F. Su, B. Stripe, S. Lewis, and W. Yun, "X-Ray Device Alteration Using a Scanning X-Ray Microscope," in *STFA 2022: Conference Proceedings from the 48th International Symposium for Testing and Failure Analysis*, Pasadena, California, USA, Oct. 30, 2022, pp. 153–162. DOI: 10.31399/asm.cp.istfa2022p0153 (cit. on pp. 100, 107, 108).

[84] E. Amini, K. Bartels, C. Boit, M. Eggert, N. Herfurth, T. Kiyan, T. Krachenfels, J.-P. Seifert, and S. Tajik, "Special Session: Physical Attacks through the Chip Backside: Threats, Challenges, and Opportunities," in *2021 IEEE 39th VLSI Test Symposium (VTS)*, Apr. 2021. DOI: 10.1109/VTS50974.2021.9441006 (cit. on p. 101).

[85] I. Cutress. "Intel's 10nm Cannon Lake and Core i3-8121U Deep Dive Review." (Jan. 25, 2019), [Online]. Available: https://www.anandtech.com/show/13405/intel-10nm-cannon-lake-and-core-i3-8121u-deep-dive-review (visited on 03/05/2023) (cit. on p. 101).

[86] D. I. Cutress. "Intel's Process Roadmap to 2025: With 4nm, 3nm, 20A and 18A?!" (Jul. 26, 2021), [Online]. Available: https://www.anandtech.com/show/16823/intel-accelerated-offensive-process-roadmap-updates-to-10nm-7nm-4nm-3nm-20a-18a-packaging-foundry-emib-foveros (visited on 03/05/2023) (cit. on p. 101).

[87] IRDS, "International Roadmap for Devices and Systems, 2018 Update, More Moore," IEEE, 2018 (cit. on p. 101).

[88] IRDS, "International Roadmap for Devices and Systems, 2021 Update, More Moore," IEEE, 2021. [Online]. Available: https://irds.ieee.org/images/files/pdf/2021/2021IRDS_MM.pdf (visited on 02/15/2023) (cit. on p. 101).

[89] R. Schlangen, R. Leihkauf, U. Kerst, C. Boit, R. Jain, T. Malik, K. Wilsher, T. Lundquist, and B. Kruger, "Backside E-Beam Probing on Nano scale devices," in *2007 IEEE International Test Conference*, Oct. 2007. DOI: 10.1109/TEST.2007.4437627 (cit. on p. 102).

[90] T. Tong, H. J. Ryu, Y. Wang, W.-H. Chuang, J. Huening, P. Joshi, and Z. Ma, "Electron Beam Probing of Active Advanced FinFET Circuit with Fin Level Resolution," in *ISTFA 2018: Conference Proceedings from the 44th International Symposium for Testing and Failure Analysis*, Phoenix, Arizona, USA: ASM International, Nov. 1, 2018, pp. 345–348. DOI: 10.31399/asm.cp.istfa2018p0345 (cit. on p. 102).

[91] C. Boit, J. Jatzkowski, F. Altmann, M. DiBattista, S. Silverman, G. Zwicker, N. Herfurth, E. Amini, and J.-P. Seifert, "The IC Ultra-Thin Back Surface - A Field of Real Nanoscale Fault Isolation Opportunities Requiring a Skillful Sample Preparation," in *2022 IEEE International Symposium on the Physical and Failure Analysis of Integrated Circuits (IPFA)*, Jul. 2022. DOI: 10.1109/IPFA55383.2022.9915783 (cit. on p. 102).

[92] J. Huening, P. Joshi, S. Zhao, W.-h. Chuang, T. Tong, and Z. Ma, "E-beam Probing: A High-Resolution Technique to Read Volatile Logic and Memory Arrays on Advanced Technology Nodes," in *2021 IEEE Physical Assurance and Inspection of Electronics (PAINE)*, Nov. 2021. DOI: 10.1109/PAINE54418.2021.9707713 (cit. on p. 102).

[93] V. K. Khanna, "Short-Channel Effects in MOSFETs," in *Integrated Nanoelectronics: Nanoscale CMOS, Post-CMOS and Allied Nanotechnologies*, ser. NanoScience and Technology, New Delhi: Springer India, 2016, pp. 73–93. DOI: 10.1007/978-81-322-3625-2_5 (cit. on p. 103).

[94] V. K. Ravikumar, R. Wampler, M. Y. Ho, J. Christensen, and S. L. Phoa, "Laser voltage probing in failure analysis of advanced integrated circuits on SOI," in *2012 19th IEEE International Symposium on the Physical and Failure Analysis of Integrated Circuits*, Jul. 2012. DOI: 10.1109/IPFA.2012.6306297 (cit. on p. 103).

[95] F. Stellari, P. Song, J. Vickers, C. Shaw, S. Kasapi, and R. Ispasoiu, "Evaluating PICA Capability for Future Low Voltage SOI Chips," in *ISTFA 2008: Conference Proceedings from the 34th International Symposium for Testing and Failure Analysis*, Portland, Oregon, USA: ASM International, Nov. 1, 2008, pp. 407–416. DOI: 10.31399/asm.cp.istfa2008p0407 (cit. on p. 103).

[96] J. Fine, C. Young, C. Hobbs, G. Bersuker, T. Lundquist, and C.-C. Tsao, "Optical Probing of FinFETs," in *Proceedings of the 20th IEEE International Symposium on the Physical and Failure Analysis of Integrated Circuits (IPFA)*, Jul. 2013, pp. 28–31. DOI: 10.1109/IPFA.2013.6599121 (cit. on p. 104).

[97] U. Ganesh, "Laser Voltage Probing (LVP) – Its value and the race against scaling," *Microelectronics Reliability*, Proceedings of the 27th European Symposium on Reliability of Electron Devices, Failure Physics and Analysis, vol. 64, pp. 294–298, Sep. 1, 2016. DOI: 10.1016/j.microrel.2016.07.054 (cit. on p. 104).

[98] R. Ritzenthaler, H. Mertens, G. Eneman, E. Simoen, E. Bury, *et al.*, "Comparison of Electrical Performance of Co-Integrated Forksheets and Nanosheets Transistors for the 2nm Technological Node and Beyond," in *2021 IEEE International Electron Devices Meeting (IEDM)*, Dec. 2021. DOI: 10.1109/IEDM19574.2021.9720524 (cit. on p. 105).

[99] K. Sakuma, P. S. Andry, C. K. Tsang, S. L. Wright, B. Dang, *et al.*, "3D chip-stacking technology with through-silicon vias and low-volume lead-free interconnections," *IBM Journal of Research and Development*, vol. 52, no. 6, pp. 611–622, Nov. 2008. DOI: 10.1147/JRD.2008.5388567 (cit. on p. 105).

[100] D. B. Ingerly, S. Amin, L. Aryasomayajula, A. Balankutty, D. Borst, *et al.*, "Foveros: 3D Integration and the use of Face-to-Face Chip Stacking for Logic Devices," in *2019 IEEE International Electron Devices Meeting (IEDM)*, Dec. 2019. DOI: 10.1109/IEDM19573.2019.8993637 (cit. on p. 105).

[101] P. K. Huang, C. Y. Lu, W. H. Wei, C. Chiu, K. C. Ting, *et al.*, "Wafer Level System Integration of the Fifth Generation CoWoS®-S with High Performance Si Interposer at 2500 mm2," in *2021 IEEE 71st Electronic Components and Technology Conference (ECTC)*, Jun. 2021, pp. 101–104. DOI: 10.1109/ECTC32696.2021.00028 (cit. on p. 105).

[102] C. Hu, M. Chen, W. Chiou, and D. C. Yu, "3D Multi-chip Integration with System on Integrated Chips (SoIC™)," in *2019 Symposium on VLSI Technology*, Jun. 2019, T20–T21. DOI: 10.23919/VLSIT.2019.8776486 (cit. on p. 106).

[103] A. Mocuta, P. Weckx, S. Demuynck, D. Radisic, Y. Oniki, and J. Ryckaert, "Enabling CMOS Scaling Towards 3nm and Beyond," in *2018 IEEE Symposium on VLSI Technology*, Jun. 2018, pp. 147–148. DOI: 10.1109/VLSIT.2018.8510683 (cit. on p. 107).

[104] D. Prasad, S. S. Teja Nibhanupudi, S. Das, O. Zografos, B. Chehab, *et al.*, "Buried Power Rails and Back-side Power Grids: Arm® CPU Power Delivery Network Design Beyond 5nm," in *2019 IEEE International Electron Devices Meeting (IEDM)*, Dec. 2019. DOI: 10.1109/IEDM19573.2019.8993617 (cit. on p. 107).

[105] D. Schor. "Intel Announces 20A Node: RibbonFET Devices, PowerVia, 2024 Ramp," WikiChip Fuse. (Jul. 26, 2021), [Online]. Available: https://fuse.wikichip.org/news/5943/intel-announces-20a-node-ribbonfet-devices-powervia-2024-ramp/ (visited on 02/15/2023) (cit. on p. 107).