Laser-Based Attacks on Secure Integrated Circuits

Extracting and Protecting Sensitive Information

vorgelegt von
M. Eng.
Heiko Lohrke

von 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.-Ing. Friedel Gerfers
Gutachter: Prof. Dr.-Ing. Christian Boit
Gutachter: Prof. Dr. Jean-Pierre Seifert
Gutachter: Prof. Patrick Schaumont, Ph.D. (Virginia Tech, USA)

Tag der wissenschaftlichen Aussprache: 14. Dezember 2018

Berlin 2019

LASER-BASED ATTACKS ON SECURE INTEGRATED CIRCUITS

Extracting and Protecting Sensitive Information

HEIKO LOHRKE

Fachgebiet Halbleiterbauelemente
Fakultät Elektrotechnik und Informatik
Technische Universität Berlin

Sometimes things are hidden under the surface.
You just gotta know how to bring 'em out.

— Angus MacGyver

Dedicated to my friends, family, and the memory of my father.

ABSTRACT

To defend against attackers, knowledge about their approach is helpful. This work evaluates attacks on integrated circuits (ICs) using laser scanning microscopes (LSMs) in combination with semiconductor failure analysis (FA) techniques. It identifies likely attack paths, develops suitable setups, and tests attack feasibility. All attacks are performed through the silicon backside. Three main attack approaches are evaluated: automated laser fault injection location profiling, memory readout by laser stimulation, and reverse engineering as well as data extraction using optical contactless probing. Using these approaches, commercial application-specific integrated circuits (ASICs) down to 20 nm technology size are successfully attacked using an optical resolution of about 1 µm. Additionally, the use of visible light (VIS) and solid immersion lenses (SILs) for resolution improvement as well as low-cost approaches to VIS LSMs are assessed. Furthermore, concrete countermeasures are implemented and evaluated. Further mitigation approaches are also presented and discussed. The lack of backside protection is identified as a key factor in all attacks. Flip-chip devices are found to worsen this situation by removing the need for any preparation and giving direct backside access. More specifically, the findings are as follows.

Automated laser fault injection profiling is shown to determine suitable attack locations in a time span in the order of minutes. This is demonstrated by analyzing the configuration memory on 180 nm technology size Altera MAX V complex programmable logic devices (CPLDs). Using the profiling information, laser reconfiguration attacks are performed on proof-of-concept (POC) implementations of physically unclonable functions (PUFs). Additionally, an analysis of the underlying fault mechanism is carried out.

For laser stimulation, data extraction from static random access memory (SRAM) and battery-backed SRAM (BBRAM) key memory is demonstrated. Readout of the 1 KB SRAM of a 180 nm technology Texas Instruments MSP430 microcontroller is presented with error rates between 0.4% and 5.5%. BBRAM key recovery from the ASIC decryption core of a 20 nm technology Xilinx Kintex UltraScale field-programmable gate array (FPGA) is developed in 7 hours of lab work, with a single 256-bit key ultimately being extracted in 15 minutes.

For optical contactless probing, two attacks are presented. The first performs key extraction on a POC implementation of a PUF key storage concept by Xilinx implemented on a 60 nm Altera Cyclone IV FPGA. A contactless characterization of the employed ring oscillator PUF is also performed. The second attack enables reverse-engineering

and plaintext extraction on the decryption ASIC of a 28 nm technology Xilinx Kintex 7 FPGA. The plaintext gates are found in less than 10 working days and extraction of decrypted data is demonstrated.

For evaluation of resolution improvements, a suitable setup for comparison of visible light (VIS) and infrared (IR) illumination is developed. Additionally, a gallium phosphide solid immersion lens (GaP SIL) is designed, fabricated, incorporated into the setup, and experimentally verified. Application of the SIL improves resolution by 190% and 170% in IR and VIS respectively. Switching the illumination of the SIL from IR to VIS yields an improvement of 60%. Optical contactless probing techniques in VIS are performed by using the setup on 16/14 nm FinFET test devices. Additionally, a low-cost setup for VIS LSMs built from optical drive components is presented. The setup is capable of acquiring reflected light images with sub-micron resolution with a hardware cost of less than $100.

In connection with implemented countermeasures, two circuits are presented. The first one is shown to successfully prevent laser stimulation data extraction by injection of a noisy current. The second circuit combines a PUF with an attack detection circuit to prevent optical probing. Sensitivity to both 1.1 µm and 1.3 µm laser radiation is demonstrated while also providing PUF functionality. Further potential attack-specific and general countermeasures are discussed.

In conclusion, this work shows that employing LSM-based failure analysis techniques for attack development is a promising approach for attackers and a serious threat to defenders. Especially in the absence of backside protection, a multitude of attacks can be successfully launched. To properly secure integrated circuits in the future, the development of reliable, thorough, and cost-effective backside protection structures is indicated.

ZUSAMMENFASSUNG

Zur Abwehr von Angreifern ist Wissen über ihre Vorgehensweise hilfreich. Diese Arbeit untersucht Angriffe auf integrierte Schaltkreise (ICs) unter Verwendung von Laserscanmikroskopen (LSMs) in Kombination mit Halbleiterfehleranalysetechniken. Sie identifiziert wahrscheinliche Angriffswege, entwickelt geeignete Aufbauten und testet mit diesen die Durchführbarkeit der Angriffe. Alle Angriffe in dieser Arbeit werden über die Siliziumrückseite ausgeführt. Drei Hauptansätze werden evaluiert: automatisiertes Auffinden von geeigneten Positionen zur Laserfehlerinjektion, Speicherauslesen durch Laserstimulation sowie Reverse Engineering und Datenextraktion mittels optischem kontaktlosem Probing. Mit diesen Ansätzen werden kommerzielle anwendungsspezifische integrierte Schaltungen (ASICs) bis zu einer Technologiegröße von 20 nm mit einer optischen Auflösung

von etwa 1 µm erfolgreich angegriffen. Darüber hinaus werden die Verwendung von sichtbarem Licht (VIS) und Festkörperimmersionslinsen (SILs) zur Verbesserung der Auflösung sowie kostengünstige Ansätze für VIS LSMs erforscht. Weiterhin werden konkrete Gegenmaßnahmen implementiert und evaluiert. Weitere Möglichkeiten zum Schutz vor Angriffen werden ebenfalls vorgestellt und diskutiert. Der fehlende Rückseitenschutz wird als Schlüsselfaktor für alle Angriffe identifiziert. Flip-Chip-Gehäuse verschlechtern diese Situation, da keine Gehäusebearbeitung mehr erforderlich ist und ein direkter Zugriff auf die Rückseite erfolgen kann. Im Detail werden die folgenden Ergebnisse erzielt.

Es wird gezeigt, dass das automatisierte Auffinden von Laserfehlerinjektionspositionen es erlaubt geeignete Angriffsorte innerhalb von Minuten zu bestimmen. Dies wird durch die Analyse des Konfigurationsspeichers von komplexen programmierbaren Logikbausteinen (CPLDs) vom Typ Altera MAX V mit 180 nm Technologie demonstriert. Unter Verwendung der erhaltenen Informationen werden Laserrekonfigurationsangriffe auf proof-of-concept (POC) Implementierungen von physically unclonable functions (PUFs) durchgeführt. Zusätzlich erfolgt eine Analyse des zugrunde liegenden Fehlermechanismus.

Bei den Laserstimulationsangriffen wird die Datenextraktion aus static random access memory (SRAM) und sowie aus batteriegepuffertem SRAM Schlüsselspeicher (BBRAM) demonstriert. Das Auslesen von 1 KB SRAM eines 180 nm Technologie Texas Instruments MSP430-Mikrocontrollers mit Fehlerraten zwischen $0,4\%$ und $5,5\%$ wird gezeigt. Die Extraktion des geheimen Schlüssels aus dem BBRAM eines ASIC-Entschlüsselungskerns, der Teil eines 20 nm Technologie FPGAs (Xilinx Kintex UltraScale) ist, wird demonstriert. Der Angriff wird in 7 Stunden Laborarbeit entwickelt, wobei ein einzelner 256-Bit-Schlüssel innerhalb von 15 Minuten extrahiert werden kann.

Für das optische kontaktlose Proben werden zwei Angriffe vorgestellt. Der Erste führt eine Schlüsselextraktion an einer POC-Implementierung eines PUF-Schlüsselspeicherkonzepts der Firma Xilinx durch, welches auf einem 60 nm Altera Cyclone IV FPGA realisiert wurde. Eine kontaktlose Charakterisierung der verwendeten Ringoszillator-PUF wird ebenfalls durchgeführt. Der zweite Angriff ermöglicht Reverse Engineering und Klartextextraktion auf dem Entschlüsselungs-ASIC eines 28 nm Technologie Xilinx Kintex 7 FPGA. Die Klartextgatter werden in weniger als 10 Arbeitstagen gefunden und die Extraktion von entschlüsselten Daten wird demonstriert.

Im Rahmen der Versuche zu Auflösungsverbesserungen wird ein geeigneter Aufbau zum Vergleich von sichtbarer (VIS) und infraroter (IR) Beleuchtung entwickelt. Zusätzlich wird eine Galliumphosphidfestkörperimmersionslinse (GaP SIL) entworfen, hergestellt, in den Aufbau integriert und experimentell verifiziert. Die Anwendung der

SIL verbessert die Auflösung um 190% bzw. 170% im IR und VIS. Das Umschalten der Beleuchtung der SIL von IR auf VIS zeigt eine Verbesserung von 60%. Unter Verwendung des Aufbaus werden optische kontaktlose Probingverfahren im VIS auf 16/14 nm FinFET Testtransistoren durchgeführt. Darüber hinaus wird ein kostengünstiges Setup für VIS LSMs vorgestellt, welches aus den Komponenten optischer Laufwerke aufgebaut wird. Dieses System ist in der Lage optische Bilder mit einer Auflösung von weniger als einem Mikrometer zu erfassen, wobei die Hardwarekosten unter 100 US-Dollar liegen.

Im Rahmen der implementierten Gegenmaßnahmen werden zwei Schaltkreise vorgestellt. Der Erste zeigt, dass die Extraktion von Daten durch thermische Laserstimulation mittels Injektion eines Rauschstroms erfolgreich verhindert werden kann. Die zweite Schaltung kombiniert eine PUF mit einer Angriffsdetektionsschaltung, um kontaktlose optische Probingangriffe zu verhindern. Die Empfindlichkeit gegenüber sowohl 1,1 µm als auch 1,3 µm Laserstrahlung wird demonstriert, während gleichzeitig PUF-Funktionalität bereitgestellt wird. Darüber hinaus werden weitere mögliche angriffsspezifische und allgemeine Gegenmaßnahmen diskutiert.

Zusammenfassend zeigt diese Arbeit, dass die Anwendung von LSM-basierten Fehleranalysetechniken für die Angriffsentwicklung ein vielversprechender Ansatz für Angreifer und eine ernsthafte Bedrohung für Verteidiger darstellt. Insbesondere wenn kein Rückseitenschutz vorhanden ist, kann eine Vielzahl von Angriffen erfolgreich durchgeführt werden. Um integrierte Schaltungen in Zukunft schützen zu können, ist die Entwicklung von zuverlässigen, umfassenden und kostengünstigen Rückseitenschutzstrukturen angezeigt.

PUBLICATIONS

This constitutes a list of publications released in connection with this thesis. A "*" denotes that both authors contributed equally to the corresponding work.

C. Boit, **H. Lohrke**, P. Scholz, A. Beyreuther, U. Kerst, and Y. Iwaki. "Contactless Visible Light Probing for Nanoscale ICs through 10 μm Bulk Silicon." In: *Proc. 35th Annual NANO Testing Symposium (NANOTS)*. Invited Paper. 2015.

S. Tajik*, **H. Lohrke***, F. Ganji, J.-P. Seifert, and C. Boit. "Laser Fault Attack on Physically Unclonable Functions." In: *Proc. Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)*. 2015.

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: *Proc. IEEE 23rd Int. Symp. on the Physical and Failure Analysis of Integrated Circuits (IPFA)*. Invited Paper. 2016.

H. Lohrke*, S. Tajik*, C. Boit, and J.-P. Seifert. "No Place to Hide: Contactless Probing of Secret Data on FPGAs." In: *Proc. 18th Conference on Cryptographic Hardware and Embedded Systems (CHES)*. 2016.

H. Lohrke*, S. Tajik*, P. Scholz, C. Boit, and J.-P. Seifert. "Automated Detection of Fault Sensitive Locations for Reconfiguration Attacks on Programmable Logic." In: *Proc. 42nd Int. Symp. for Testing and Failure Analysis (ISTFA)*. 2016.

H. Lohrke*, P. Scholz*, A. Beyreuther, U. Ganesh, E. Uhlmann, S. Kühne, M. Jagodzinski, Y. Iwaki, R. Chivas, S. Silverman, et al. "Contactless Fault Isolation for FinFET Technologies with Visible Light and GaP SIL." In: *Proc. 42nd Int. Symp. for Testing and Failure Analysis (ISTFA)*. 2016.

H. Lohrke, H. Zöllner, P. Scholz, S. Tajik, C. Boit, and J.-P. Seifert. "Visible Light Techniques in the FinFET Era: Challenges, Threats and Opportunities." In: *Proc. IEEE 24th Int. Symp. on the Physical and Failure Analysis of Integrated Circuits (IPFA)*. Invited Paper. 2017.

S. Tajik, J. Fietkau, **H. Lohrke**, J.-P. Seifert, and C. Boit. "PUFMon: Security Monitoring of FPGAs Using Physically Unclonable Functions." In: *Proc. IEEE 23rd Int. Symp. on On-Line Testing and Robust System Design (IOLTS)*. 2017.

S. Tajik*, **H. Lohrke***, J.-P. Seifert, and C. Boit. "On the Power of Optical Contactless Probing: Attacking Bitstream Encryption of FPGAs." In: *Proc. Conference on Computer and Communications Security (CCS)*. CCS 2017 Best Paper Award Finalist, CSAW 2017 Europe Applied Research Award Winner. 2017.

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." In: *IACR Transactions on Cryptographic Hardware and Embedded Systems (TCHES)* (3 2018).

T. Kiyan*, **H. Lohrke***, and C. Boit. "Comparative Assessment of Optical Techniques for Semi-Invasive SRAM Data Read-Out on an MSP430 Microcontroller." In: *Proc. 44th Int. Symp. for Testing and Failure Analysis (ISTFA)*. 2018.

ACKNOWLEDGMENTS

First of all, I would like to thank Prof. Dr. Christian Boit for supervising this thesis. He gave me all the support I could wish for, even in difficult times. With the Semiconductor Devices group, he has managed to build one of the most friendly circles of colleagues I have come across so far. Many thanks also to Prof. Dr. Patrick Schaumont for agreeing to review this thesis and providing helpful feedback. I also thank Prof. Dr. Jean-Pierre Seifert for his review and for supporting the fruitful cooperation between the Security in Telecommunications and Semiconductor Devices groups as well as for insights into the more theoretical aspects of security. I furthermore thank the Helmholtz Research School on Security Technologies (HRSST) for supporting me during the first 3.5 years of my thesis.

For introducing me to the world of hardware security I would like to thank Dr. Clemens Helfmeier. I don't think this thesis would have been possible without our discussions and the knowledge you conveyed to me. Further thanks go to Dr. Shahin Tajik for the years of fruitful and fun cooperation and the many, many nights spent in the lab with an endless supply of Iranian snacks produced by people with too large mustaches. Further thanks to Dr. Fatemeh Ganji for cooperation on papers, nice conference trips, and the attempt of explaining the magic of machine learning to me. I also thank Dr. Philipp Scholz for all the hours spent on the design and fabrication of solid immersion lenses as well as for long discussions on intercontinental flights. I would also like to thank Anne Beyreuther for being the first person willing to try with me if visible light imaging might work on our equipment. Also mentioned should be Dr. Enrico Dietz, Helmar Dittrich, and Dr. Sven Frohmann for providing knowledge and equipment. I also thank all my colleagues at the Semiconductor Devices group. Andreas for sample preparation and technical support, Helmut for many insights into electronics and lab machinery, Norbert for always helping me out when I ran out of money for coffee, Orman for his emergency candy kiosk, Elham for teaching me some Farsi, Ivo for always providing an optimistic state of mind, Arek for interesting real-world failure analysis investigations, Tuba for the pleasant cooperation on the MSP430 paper, Silvia for help with all the bureaucratic things, and Uwe for support on lab and logistics. Big thanks also go to Dan and Philipp for proofreading.

Furthermore, I would like to thank Hamamatsu, Qualcomm, and Varioscale for their support in the various cooperation projects. Special thanks go to Yoshitaka Iwaki and Minh Vo for support with the Phemos system. Many thanks also to Ulrike Kindereit for the very

enjoyable cooperation during the visible light experiments. Hannes Zöllner I would like to thank for the cooperation on the low-cost LSM and many adventurous geocaching and lost place trips. I would also like to thank Brigitte Unger for cooperation and Iskander Tschinibaew and Axel Schönau for their bachelor's theses in connection with the descrambling and bit recognition for the MSP430. I also thank Prof. Dr. Katzenbeisser and Nikolas Anagnostopoulos for cooperation on the MSP430 SRAM readout software.

Last but not least I would like to thank my parents, my family, and my friends. They have all supported me through these past 4.5 years and have on many occasions not seen me as often as they and I would have liked. My father, I thank for all the things that he has taught me, I for sure would not be an engineer today without him. My mother, I thank for sharing her faith and optimism when I was struggling. My younger sister was always there for a talk and cheering me up, thank you. I also thank my older siblings for inviting me over to their houses and the joy they brought me through my nieces. My friends have always supported me through this long journey and took me out into the Berlin nightlife when needed, thank you, I could not have done this without you.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LISTINGS

`0b11...`	Binary Number
`0xFF...`	Hexadecimal Number
ADC	Analog-to-Digital Converter
AES	Advanced Encryption Standard
ASCII	American Standard Code for Information Interchange (Character Encoding)
ASIC	Application-Specific Integrated Circuit
BBRAM	Battery-Backed Random Access Memory
CBC	Cipher Block Chaining (AES Mode of Operation)
CCLK	Configuration Clock
CLK	Clock
CPLD	Complex Programmable Logic Device
CRP	Challenge-Response-Pair
DPA	Differential Power Analysis
DRV	Data Retention Voltage
DUT	Device Under Test
EAH	Ernst-Abbe-Hochschule Jena
eFuse	Electronically Programmed Fuse (Type of Memory)
EOFM	Electro-Optical Frequency Mapping
EOP	Electro-Optical Probing
FA	Failure Analysis
FFT	Fast Fourier Transformation (Algorithm)
FIB	Focused Ion Beam
FinFET	Fin Field-Effect Transistor
FOV	Field of View
FPGA	Field-Programmable Gate Array
GaP	Gallium Phosphide

GND Ground

GPIB General Purpose Interface Bus

GUI Graphical User Interface

HeNe Helium-Neon (Type of Gas Laser)

I/O Input/Output

IoT Internet of Things

IP Intellectual Property

IR Infrared

JTAG Joint Test Action Group (Debug and Test Interface)

LAB Logic Array Block (FPGA Structure)

LADA Laser Assisted Device Alteration

LE Logic Element (FPGA Structure)

LFI Laser Fault Injection

LIVA Laser Induced Voltage Alteration

LPM Low Power Mode

LS Laser Stimulation

LSM Laser Scanning Microscope

LUT Look-Up Table

LVI Laser Voltage Imaging

LVP Laser Voltage Probing

MATLAB Numerical Computing Environment Software

MISO Master In Slave Out (SPI Bus Signal)

ML Machine Learning

MOSFET Metal-Oxide-Semiconductor Field-Effect Transistor

MOSI Master Out Slave In (SPI Bus Signal)

NA Numerical Aperture

NFC Near Field Communication

NMOS N-Type MOSFET Transistor

NOP No Operation

NVM	Non-Volatile Memory
OBIC	Optical Beam Induced Current
OBIRCH	Optical Beam Induced Resistance Change
PEM	Photon Emission Microscopy
PLD	Programmable Logic Device
PLL	Phase-Locked Loop
PLS	Photoelectric Laser Stimulation
PMOS	P-Type MOSFET Transistor
PMT	Photomultiplier Tube
POC	Proof of Concept
PPM	Parts Per Million
PUF	Physically Unclonable Function
RO	Ring Oscillator
Scilab	Numerical Computational Software Package
SCLK	Serial Clock (SPI Bus Signal)
SDL	Soft Defect Localization
SDR	Software-Defined Radio
SEL	Single Event Latch-Up
SEM	Scanning Electron Microscopy
SEU	Single Event Upset
SFD	Summation of Frequency Differences
SIL	Solid Immersion Lens
SNR	Signal-to-Noise Ratio
SPA	Simple Power Analysis
SPI	Serial Peripheral Interface Bus
SRAM	Static Random-Access Memory
TIVA	Thermally Induced Voltage Alteration
TLS	Thermal Laser Stimulation
TRNG	True Random Number Generator

TUB	Technische Universität Berlin
UART	Universal Asynchronous Receiver Transmitter
VCC	Supply Voltage
VIS	Visible (Light)
XOR	Exclusive Or (Logic Function)

1 INTRODUCTION

Since the first simple integrated circuits (ICs) were presented in the late 1950s, ICs have demonstrated a triumphant advance into nearly all areas of modern life. Today, we rely on them in our computers, mobile phones, cars, passports, and numerous other devices. Some of these ICs store secret data such as the personal identification number (PIN) in our banking cards. Others, such as the ones in planes and power plants, are part of vitally important machinery. If the secrets from these devices can be extracted or if they can be attacked, this can lead to severe consequences. Therefore, prevention of attacks and data extraction is an important task. To defend against attackers, it is extremely helpful to know how they would proceed. Attacks thus need to be understood. The evaluation of attack approaches is the main aim of this thesis.

The relevance of protection against attacks on IC-based systems becomes evident when looking at the consequences. Readout of the PIN from the IC in your banking card would allow for stealing of your savings. Extracting the private keys from your hardware bitcoin wallet would enable transfer of the cryptocurrency to the attacker. Impersonating the remote key of your car will allow criminals to drive off with it in seconds [90]. Apart from these more personal examples, IC attacks are also relevant for more crucial structures. For example, as part of a 2015 cyberattack on the Ukrainian power grid, the firmware on critical devices was overwritten and corrupted to prevent operators from restoring normal functionality [48, 94]. The attack ultimately left approximately 225,000 customers without electricity [48]. An example of an extraction attack on data contained in critical ICs is the claimed readout and decryption of sensitive data from a US spy drone captured by Iran in 2011 [16]. When considering the relevance of attacks on IC-based systems, it should also be taken into account that the number of interconnected embedded devices and wireless connections is steadily rising. Recent developments such as the "internet of things" (IoT) and near field communication (NFC) are just two prominent examples of this trend. This naturally increases the number of exposed devices and thus the attack surface. From these examples, it can be seen that the defense against attacks on IC-based systems is an important aspect.

This directly leads to the question of how attacks could be prevented. As already stated, knowledge about attacks is helpful in the design of defenses. For that very reason, malicious attackers can usually not be expected to share details about their approaches. One

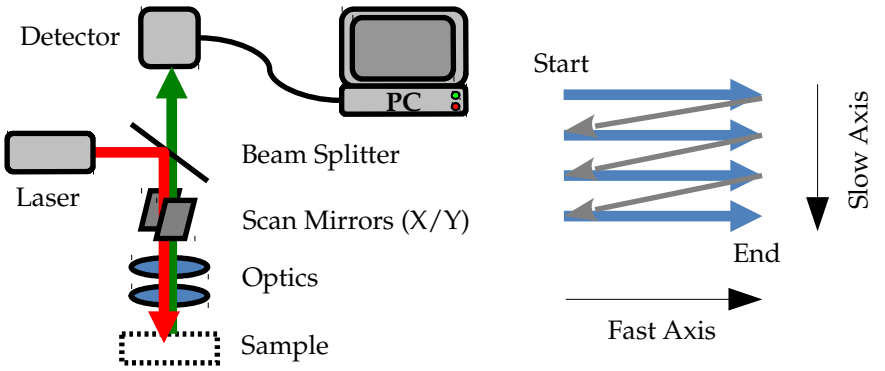


Figure 1.1: Simplified sketch of a laser scanning microscope (LSM) for reflected light image acquisition and the usually employed scan pattern. Figure based on [15].

option to gain attack knowledge would be to analyze performed attacks in retrospect to learn about the employed methods. A second option is to take the role of the attacker and evaluate different attacks on the target system. This is often referred to as a "white hat" attack. Such an approach allows to understand likely attack paths and approaches as well as their limitations, prerequisites, and required effort. The gained knowledge can then be used to design better countermeasures. It furthermore enables more sound decisions regarding the assessment of a given threat. This "white hat" attack approach will be pursued in this thesis.

There is a wide range of attack classes that can be launched against ICs. One set of attacks that is particularly worrisome are those derived from failure analysis (FA) techniques. IC failure analysis always needs techniques to extract information and parameters from the chips currently manufactured. Designers and failure analysis engineers require this information to be able to perform silicon debug and root cause of failure determination to be able to deliver a reliable product. However, the existence of these techniques leads to the question of how dangerous these or derived techniques would be in the hand of an attacker. If an attacker can either acquire FA equipment herself or rent it from an FA lab by the hour, she would have access to potentially very powerful techniques. Previous work has already demonstrated that FA techniques such as focused ion beam (FIB) and photon emission microscopy (PEM) can be used to reverse-engineer and attack ICs [28, 80]. Another common tool in failure analysis is the laser scanning microscope (LSM), see Fig. 1.1. LSMs are designed to quickly scan a laser beam across a device using galvanometric mirrors. Simultaneously, they can sample either reflected light or device parameter values, which can then be analyzed. The use of infrared wavelengths, to which the silicon is transparent, allows them to acquire images from inside the silicon, change device behavior by influencing the transistors, analyze internal device activity, and extract

data. As this class of techniques bears the potential for very powerful attacks, an evaluation of likely attack approaches seems beneficial. Because of this reason, the focus of this thesis was chosen to be the evaluation of LSM-based attacks on integrated circuits. The main research question is: what attacks could be performed and what approaches would likely be taken if an attacker had access to LSM-based FA equipment.

The approach to answering this question is to develop attacks based on LSM FA techniques, design suitable setups for them, and then perform the attacks to evaluate their feasibility. The thesis will in general focus on the engineering perspective of the evaluated attacks, i.e. on the approaches, setups, and procedures. Among the techniques evaluated will be laser fault injection, laser stimulation, and optical contactless probing. Additionally, ways to improve optical resolution and low-cost approaches will be examined. All attacks will be performed through the silicon backside, as modern devices use many metal interconnection layers on top of the transistors and these obstruct optical frontside access. The chapters of the thesis are designed to be self-contained. As a consequence, instead of a global background chapter, compact background information for each employed technique will be given at the beginning of each chapter. For readers interested in additional background information, suitable references are given where appropriate.

The outline of this thesis is as follows. Chapter 2 will combine scanning test approaches from FA with laser fault injection (LFI) to quickly identify fault-sensitive locations and perform precision LFI attacks. Chapter 3 will evaluate the use of laser stimulation for automated readout of data from internal IC memories. Chapter 4 will demonstrate the use of optical contactless probing techniques to reverse engineer circuits and extract secret data. Chapter 5 will present a setup using visible light and a solid immersion lens for resolution improvement. It will furthermore demonstrate a low-cost approach for building an LSM out of optical disk drive parts. Finally, Chapter 6 will present the results of implementing selected countermeasures and discuss additional potential attack mitigation strategies.

2 FAULT INJECTION ATTACKS

In the context of hardware security, fault injection attacks are a class of attack that seeks to disturb normal device operation in such a way that a faulty behavior is created. This faulty behavior can then be exploited to break certain security features of the device. Injected faults can be achieved in a multitude of ways: lowering the device voltage momentarily, manipulating the clock or applying electromagnetic pulses to the device. This chapter, however, will deal with so-called laser fault injection (LFI).

2.1 LASER FAULT INJECTION

Laser fault injection (LFI) uses an optical approach to disturb normal device operation: a laser beam with a photon energy larger than the silicon band gap energy is focused into the device and a laser pulse is triggered. The incident photons cause the generation of electron-hole pairs in the active area of the device. These injected carriers then cause transient changes in internal voltages and currents and might also change device parameters like switching speed [55]. If the influence is strong enough, this then leads to faulty behavior of the device. Changes in the speed of digital circuits might, for example, lead to incorrect latching of data values, which will then propagate through the rest of the circuit. If LFI is applied to flip-flops, registers, SRAM cells or similar memory structures, it can also change the value held by them. Consequently, attackers can use LFI to manipulate the behavior of a device and break its security features [70, 74, 89].

This chapter will assess the attack potential of LFI attacks under the assumption that the attacker has access to a standard failure analysis (FA) laser scanning microscope (LSM). In particular, the possibility of the automated discovery of LFI locations through laser scan techniques will be evaluated. The developed techniques will then be used to execute exemplary attacks against physically unclonable functions (PUFs). All attacks presented focus on faults injected into the configuration memory of programmable logic devices such as complex programmable logic devices (CPLDs) and field-programmable gate arrays (FPGAs). Additionally, an analysis of the underlying fault mechanism will be performed. Some of the results and figures were already published in [40, 79].

2.2 RECONFIGURATION ATTACKS

Reconfiguration attacks differ from classical laser fault injection in that they cause a semi-permanent change in the configuration of programmable logic devices. Instead of altering delays, changing memory contents or causing transient events, the targeted device will continue to operate with the faulty behavior until it is reset and reconfigured. For this reason, they constitute a powerful tool for attackers. The following sections will briefly explain the underlying principles and then move on to assess the potential threat posed by an attacker able to perform reconfiguration attacks using failure analysis equipment.

2.2.1 *Configuration of Programmable Logic Devices*

Programmable logic devices such as field-programmable gate arrays (FPGAs) and complex programmable logic devices (CPLDs) can be used to implement different hardware designs in a reconfigurable way. The tasks addressed by the implemented design can range in complexity from simple "glue logic" to complete microprocessors and cryptosystems. The logic gates inside these devices are configurable and can perform arbitrary logic functions. Additionally, routing structures allow to flexibly connect basic "building blocks" such as logic gates, memory and I/O structures inside the device to each other. With this, a reconfigurable hardware implementation of the desired tasks can be realized. For modern devices, the main architectural differences between CPLDs and FPGAs lie in their logic size and routing complexity. This section will briefly review the implementation of logic functions in programmable logic devices, focusing on the often used look-up table (LUT) concept.

The basic building blocks of CPLDs and FPGAs are programmable logic elements (LEs). These are the smallest unit of logic in a device and are designed to be a flexible primitive for the designer's implementation. To perform arbitrary combinational logic functions, LUTs are used as function generators in the LEs. An exemplary LUT consists of multiple inputs, one output, multiplexers, and memory cells to define its behavior. See Fig. 2.1 for an example of a 3-input LUT. In this case, A, B, and C are inputs of the LUT while Y is the output. The one-bit memory cells B0 to B7 hold the actual configuration and will also be referred to as the "LUT bits" here. The "select" inputs of the multiplexers are connected to the LUT inputs and will select their upper input when they receive a logic high or "1" signal and their lower input if they receive a logic low or "0" signal.

How the LUT works is best illustrated with an example: If a "1" is stored in B7 and all other bits are set to "0", the LUT will perform the logic function $Y = A \wedge B \wedge C$, which is equivalent to an AND

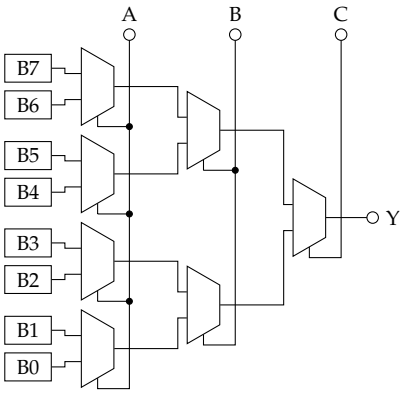


Figure 2.1: 3-input look-up table (LUT) example with inputs A/B/C, output Y, and 1-bit memory cells B0 to B7. This LUT can perform any 3-input combinational logic function depending on the programming of the memory cells.

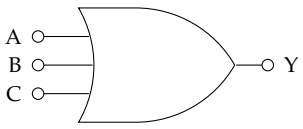
gate. Only if all inputs are "1" the multiplexers will select B7 and the output will also be "1". The configuration of the LUT bits is often also written down as a binary or hexadecimal representation of the memory cell contents. In this case, the AND gate configuration can be referred to as either `0b10000000` or `0x80` respectively. Inputs of the LUT can also be ignored, depending on the LUT configuration. For example, to perform the function of an inverter $Y = \neg A$, the LUT could be configured as `0b01010101`, causing inputs B and C to be "don't care" inputs.

LUTs can be made to accept an arbitrary number of inputs by extending the structure. For example, a 4-input LUT can be realized by combining two 3-input LUTs. In this case, both A, B and C inputs of the 3-input LUTs are connected together and their Y outputs are fed into an additional multiplexer whose select input is connected to the new "D" input. The output of this multiplexer then constitutes the output of the 4-input LUT. It can be seen by these examples that in the general case an n-input LUT can perform any n-input combinational logic function while requiring 2^n 1-bit memory cells. Such a LUT can then perform $2^{(2^n)}$ different combinational logic functions. In this context, the inputs of a LUT can also be seen as address inputs indexing into a 1-bit memory of size 2^n.

Apart from the LUTs, the LEs also contain programmable registers to perform sequential logic functions. Additionally, there are routing resources available. These consist of programmable switch matrices which can be used to connect different LEs to each other. The settings for these elements are also stored in memory cells inside the device.

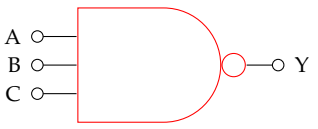
As many devices use volatile SRAM cells to store their configuration, the values of these cells have to be loaded at each power-on. In the case of CPLDs, the required non-volatile memory (NVM) is usually built into the device, while for FPGAs it is an external component.

A	B	C	Y	LUT bit
0	0	0	0	B0
1	0	0	1	B1
0	1	0	1	B2
1	1	0	1	B3
0	0	1	1	B4
1	0	1	1	B5
0	1	1	1	B6
1	1	1	1	B7



(a) Normal OR gate configuration.

A	B	C	Y	LUT bit
0	0	0	1	B0
1	0	0	1	B1
0	1	0	1	B2
1	1	0	1	B3
0	0	1	1	B4
1	0	1	1	B5
0	1	1	1	B6
1	1	1	0	B7



(b) Faulty NAND gate configuration.

Figure 2.2: Truth tables for a reconfiguration attack against a 3-input LUT. If the attacker manages to flip the bits B0 and B7, she can transform the OR gate to a NAND gate.

The binary data containing the setting of the configuration memory cells and therefore also the implemented design is commonly referred to as the "bitstream".

2.2.2 *Reconfiguration Attack Scenario*

Taking the design of LUTs in CPLDs and FPGAs into account, it is evident that a change in the memory cells will also change the function that the LUT performs. An attacker can exploit this to mount a targeted attack on the LUT bit memory cells. Using a high power laser beam she might be able to flip some bits in the configuration memory of the LUT. This will allow her to change the logic function of the gate and thus to deactivate or alter security-relevant functions which use this gate.

Fig. 2.2 provides an example of such an attack. In this case, a 3-input LUT has been configured as an OR gate, performing the function $Y = A \vee B \vee C$. The LUT bits are set to `0b11111110` or `0xFE` in this case. If the attacker manages to flip the bits B0 and B7, she can transform the OR gate to a NAND gate with a configuration of `0b01111111` or `0x7F`. The same approach can be used if bigger LUTs with a larger number of bits are to be attacked. Similarly, changes can also be introduced into the memory cells controlling the routing of signals.

However, fault-sensitive locations, as well as their influence on the circuit, vary with the layout [62]. Thus, as the attacker usually has no access to the layout, she will have to perform some kind of characterization of the fault-sensitive locations. If she does this manually, this

might be a very tedious task if she wants to characterize the whole chip area. On the other hand, if she just tests some areas that she suspects to be susceptible, she will risk missing sensitive locations.

Yet, if she has access to failure analysis equipment, it would be relatively straightforward for her to automate this task. She can, for example, use a laser scanning microscope (LSM) for beam positioning. LSMs are designed to quickly scan a beam across an area of interest using galvanometric mirrors and measure the reflected light simultaneously. A PC is then used to assemble the reflected light data into a 2D image of the examined chip. With such a system, she would be able to use a scanning approach for fault injection, potentially analyzing the whole chip within seconds. If she configures the setup to repeatedly shoot the laser with high power while the chip is scanned and simultaneously analyzes the device for changes, she will find all fault-sensitive locations in a small amount of time.

To allow her to determine if she has actually injected a fault at a certain location she has multiple options. The first is that changes in the implementation she seeks to attack directly lead to observable changes outside of the device. This might be a change in an output pin state or in a ciphertext or plaintext that is output by the device. If there is no such observable output available, she might be able to add it. This can be achieved by eavesdropping on the transmission of an unencrypted bitstream from an external memory to an FPGA, reverse engineering the bitstream and adding an extra output to the design by use of suitable tools [58]. If the bitstream is not available to her, she can also profile the fault-sensitive locations for generic logic primitives on a training device and later use photon emission analysis to find such primitives on the target [80] and attack them.

As it seems that with such an approach fast automated detection of fault-sensitive locations is feasible, an evaluation of the actual capabilities of such an approach is needed. The first step of this assessment would be the design of a suitable setup.

2.2.3 *Automated Detection of Reconfiguration Attack Locations*

In this section, a suitable setup for the automated mapping of fault-sensitive locations will be developed and tested.

2.2.3.1 *Hardware Setup*

To implement the attack scenario outlined in Sect. 2.2.2, it is required that the device is first reset and reconfigured, to ensure it is in a clean, fault-free state. Afterward, the laser spot will be positioned on the first location of interest and a high power shot will be triggered. Next, the device will need to be analyzed for changes in its configuration and the results saved. Finally, the beam will be moved to the next location and the procedure will be repeated.

The first step of resetting and reconfiguring the device can be done in a very straightforward way, by simply power cycling the device.

For the second step, the laser beam needs to be positioned fairly quickly, to be able to analyze an area of interest in a reasonable amount of time. If the attacker has access to a failure analysis laser scanning microscope (FA LSM), this is however already a given. FA LSMs are designed to quickly scan a laser beam across a DUT while simultaneously sampling either reflected light or device parameter values. Since these systems use galvanometric mirrors with very little inertia for beam control, their sampling and positioning speed can be in the order of 512 by 512 locations in two seconds.

To examine the device for changes in the third step, the attacker will use some form of analysis circuit. This might either be implemented directly into the device, if the attacker is able to modify the device programming, or might also be realized as an external circuit. For capturing the device analysis result, the sampling inputs of the already employed FA LSM can then be used.

An aspect which needs to be considered in this case is the point in time at which the analysis of the DUT is performed. If the device is analyzed during or directly after the laser shot, the DUT will still experience perturbations from the generated carriers, and these effects will be apparent in the analysis result. However, the effects which only stem from the generated carriers will be transient effects and will be temporary. Thus, if the DUT is examined a long enough time after laser irradiation has stopped, these transient effects will have subsided. Accordingly, an analysis at this point will only show faults which have been semi-permanently injected into the configuration memory. As a result, by changing the delay between laser shot and DUT analysis, an attacker will be able to choose if she only wants to map semi-permanent or also transient faults. Therefore, this delay should be implemented as an adjustable parameter in the setup. Finally, a suitable setup should also be able to supply some basic control signals to the DUT.

With these concepts in mind, it is relatively straightforward to develop a suitable hardware setup, which is shown in Fig. 2.3. The optical section of the setup consists of a Hamamatsu Phemos-1000 laser scanning microscope. A 20x/0.4NA Mitutoyo objective lens allows for long distance navigation. For fault injection and short distance navigation, a 50x/0.76NA Hamamatsu objective is available. A 1064 nm laser (Hamamatsu C9215) is used for fault injection. The laser can be operated in two modes: high power pulsed mode and low power mode. For navigation, the low power mode can be used, while the high power mode is solely used for fault injection. The low power mode supplies a maximum laser power of 200 mW at the beam source, while high power mode allows for 1 W of peak power with a

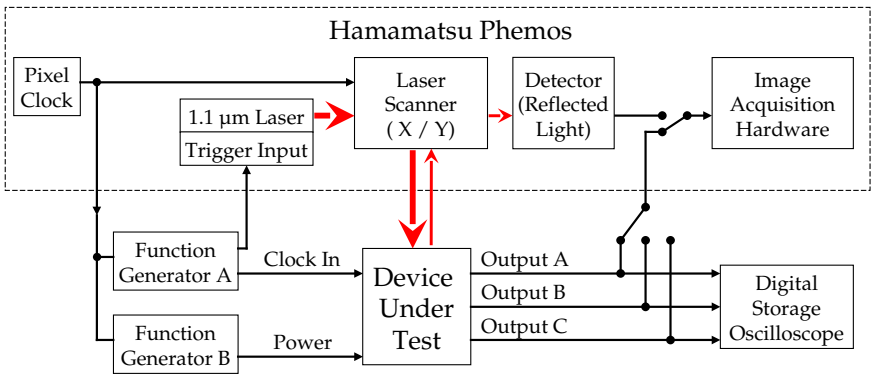


Figure 2.3: Block diagram of the optical and electrical setup for fault-sensitive location profiling. © 2015 IEEE. [79]

pulse duration of 200 ns. Adjustment of the laser power is possible in 0.5% steps from 2% to 100%.

The electrical setup consists of two function generators, designated "A" and "B" (Rigol DG4162 and Keithley 3390) and a digital storage oscilloscope (LeCroy WaveMaster 8620A). A pixel clock output that is synchronized with the scanning mirrors is provided by the Phemos. This output serves as the primary trigger for the function generators. Channel one of function generator A is then used to supply control signals (such as clock) to the DUT, while channel two is used to trigger the laser shot. Function generator B is used to control power to the device, to allow for triggering of a power-on reset event and device configuration. The digital storage oscilloscope is used to monitor the outputs of the DUT for testing and control, as well as acquire time domain output waveforms.

The analysis of the DUT functions during fault injection is done by connecting the DUT outputs to the pixel value sampling input of the image acquisition hardware. Depending on the type of analysis performed, different filters and circuits can also be inserted into this signal path. Additionally, for experiments using continuous power, a Toellner TOE8732 power supply is available.

In total, this concept delivers a flexible setup which can be used to perform different types of automated fault mapping analyses.

2.2.3.2 *Device Under Test*

Altera MAX V CPLDs with part number 5M80ZT100C5N and 180 nm technology size were selected as DUTs. The 100-pin TQFP package option was chosen and an Ultratec ASAP-1 polishing machine was used to bring the samples to 30 µm remaining silicon thickness. The samples were then inversely soldered to a custom PCB and placed in the Phemos, see Fig. 2.4. In these devices, the programmable logic is arranged in logic array blocks (LABs), with each LAB containing

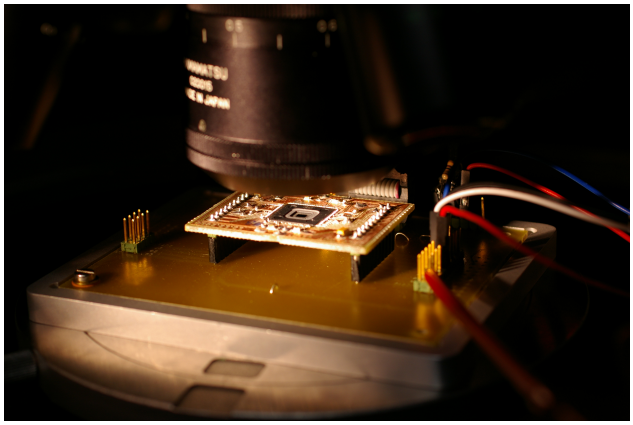


Figure 2.4: The DUT and a custom carrier board are placed in the Phemos laser scanning microscope. © 2015 IEEE. [79]

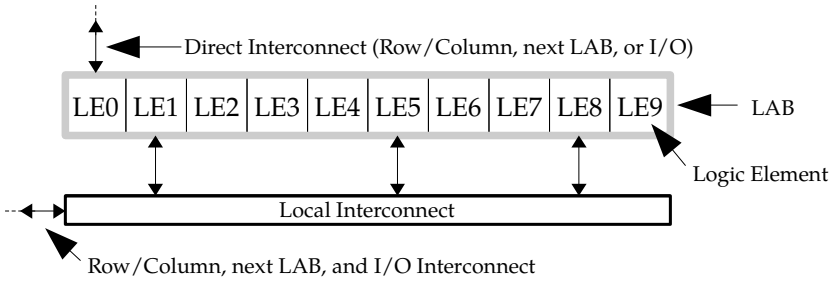


Figure 2.5: Simplified structure of a single logic array block (LAB) with 10 logic elements (LEs) and routing interconnects. The interconnects can be used to route signals to other LEs, LABs, and I/Os, either directly or through additional "row and column" interconnect structures. Figure based on [3].

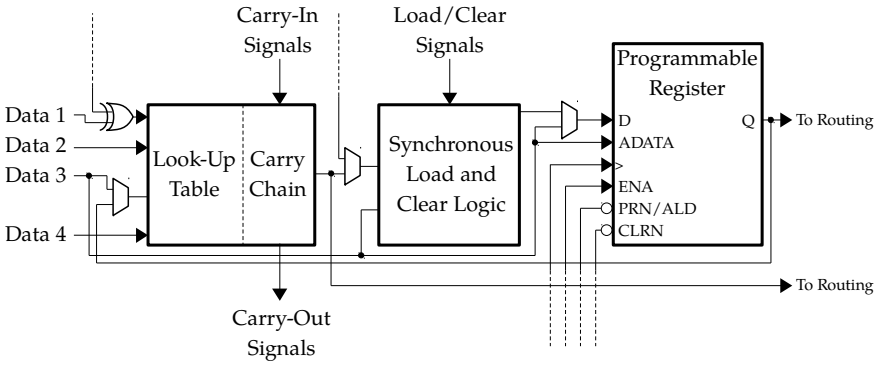


Figure 2.6: Simplified structure of a single logic element (LE). The look-up table (LUT) can be used for combinatorial logic while the programmable register can be used for sequential logic. The outputs of the LE are fed into routing and interconnect structures. Figure based on [3].

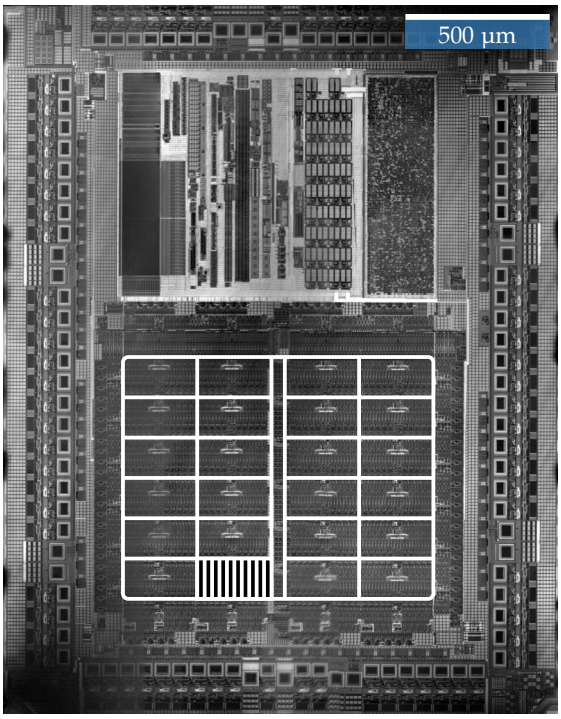


Figure 2.7: Backside reflectance image of the Altera MAX V 5M80ZT100C5N CPLD. The framed area contains the programmable logic cells. The grid corresponds to the placement of 4 by 6 LABs. Each LAB contains 10 LEs (only shown for one LAB). © 2015 IEEE. [79]

10 logic elements (LEs), see Fig. 2.5. The structure of an individual LE can be seen in Fig. 2.6. Each LE contains a 4-input programmable LUT for combinatorial logic functions. Additionally, each LE has a dedicated register, which allows implementing sequential logic functions as well. A laser scan image of the DUT with the LAB and LE positions annotated can be seen in Fig. 2.7. Both the V_{CCINT} and V_{CCIO} supply voltages of the device are powered with 1.8 V in all experiments, which is within the nominal voltage rating.

2.2.3.3 *Analysis Scenarios*

To illustrate the usage of the setup, different analysis scenarios are explained and tested on the selected DUT. The first case that is considered is the analysis of fault injection locations for the different bits of the LUTs of the MAX V CPLD. In this case, it is assumed that the attacker wants to profile the fault injection locations on a test device to gain knowledge for an attack on an actual target device of the same type later. To allow for output of the relevant LUT bits, she will implement some chosen logic function in a LUT for analysis and an additional counter, see Fig. 2.8. The counter in this example will be driven by the "clock in" signal of the setup, while the outputs of the counter will be connected to the inputs of the LUT. In this way, ev-

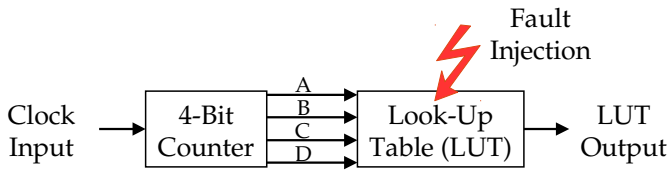


Figure 2.8: Logic implementation for look-up table fault-sensitive location profiling.

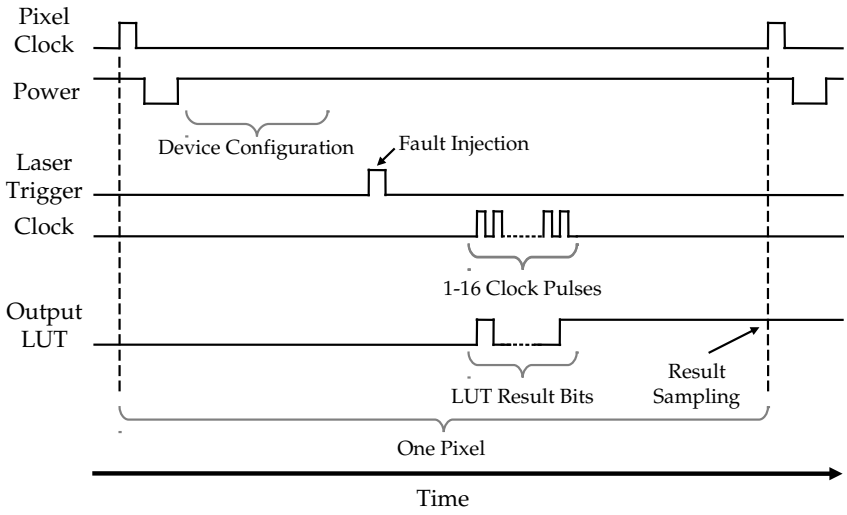
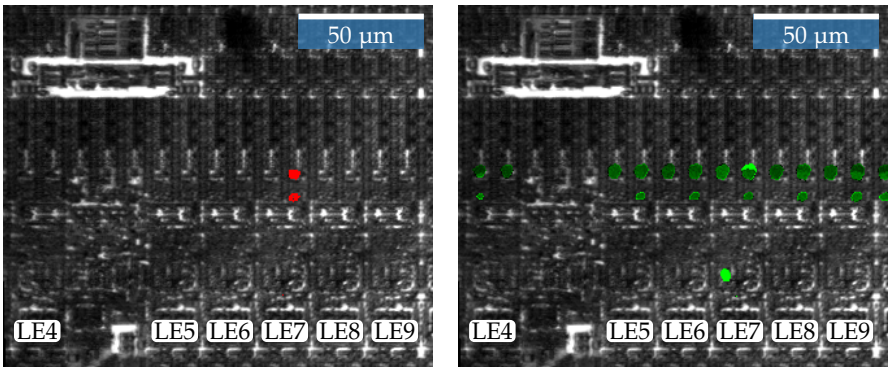


Figure 2.9: Signal flow during LUT bit fault injection analysis. © 2015 IEEE. [79]

ery pulse on the clock input will advance the counter and cause the next bit of the LUT to be present at the LUT output. By connecting the LUT output to a DUT output pin the attacker will then be able to bring the current value of a chosen LUT bit to the output pin, depending on how many clock pulses she supplies to the clock input of the DUT. To avoid influencing the counter circuit instead of the LUT during fault injection, she will have to take care to implement the LUT in an isolated region separated from the counter circuits. Using a suitable signal flow, see Fig. 2.9, she will then be able to analyze all fault locations for a chosen bit.

In this case, as soon as the pixel clock goes low, the DUT supply voltage will momentarily be lowered, causing a power-on reset. After this, the device is allowed enough time to perform configuration and reach a stable operational state. Afterward, a high power laser shot is triggered followed by an additional delay. After this, 1 to 16 clock pulses are supplied to the clock input of the counter, see Fig. 2.9. The number of clock cycles will then determine which bit value is present at the output of the LUT, compare Fig. 2.8. The output value of the LUT will then be sampled by the Phemos input, as soon as the pixel clock goes high. This procedure will then be repeated for the next



(a) LUT bit 11, normally low

(b) LUT bit 15, normally high

Figure 2.10: Fault-sensitive location mapping results overlayed on a reflected light image of the DUT for an AND gate implemented in a single LE. Red denotes a low-to-high fault, green a high-to-low fault. Figure 2.10b shows transient faults in dark green and semi-permanent faults in light green.

pixel. In this way, a map of the state of the selected LUT bit for every tested location is obtained, and fault injection locations which change this bit can easily be identified. By repeating the measurement with a different number of clock pulses, all LUT bits can be analyzed in this fashion.

As an example measurement, an AND logic gate implemented in a single 4-input LUT is considered. In this case, there are 16 possible input combinations, and therefore 16 bits in the LUT. To perform the AND function, the value `0x8000` will be programmed into the configuration memories of the LUT. For the counter, a four-bit-wide variant is used. The results of the automated fault injection location mapping can be seen in Fig. 2.10 and were acquired with 80% laser power in 240 s. In these images, the fault mapping results have been overlayed onto a reflected light image, to allow for better orientation. Additionally, the normal state of the bit output has been faded out, so that only changes in the output are visible as either red (high logic level) or green (low logic level) spots.

In Fig. 2.10a it is evident that there are two locations which cause bit 11 to be changed from zero to one. Analysis of the implementation shows that these locations lie in the area of the LE in which the AND function was implemented. All other locations can be seen to not alter the bit. Therefore, using this analysis, two locations for altering bit 11 of the LUT can be discovered. Note that in the case that the attacker has no knowledge about the implementation location, this method will also deliver the implementation's LE position to her.

In Fig. 2.10b the same measurement is performed for bit 15. In this case, there are two different types of fault injection locations visible: areas where the output voltage is slightly below the high level (dark green) and areas where it is at a clean low level of zero volts (light

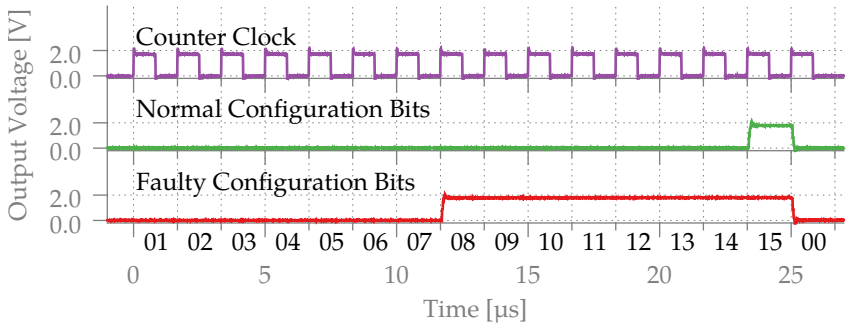


Figure 2.11: Normal and faulty configuration bits for a single LUT. The logic function is changed from an AND for all four inputs to a buffer for input D.

green). A detailed analysis in the time domain using the oscilloscope shows that the dark green areas are transient faults while the light green areas represent semi-permanent faults, i.e. LUT bit flips. The semi-permanent fault injection locations again lie in the LE in which the AND gate is implemented. Thus, two locations for manipulating bit 15 of the LUT are revealed. However, as opposed to bit 11, there are also transient faults visible in all other LEs, although these have not been used in the implementation. This can be explained in the following way: as bit 15 is normally one, the LUT output at the point of sampling is usually high, i.e. close to the supply voltage. As dark green areas represent a DUT output voltage slightly lower than the high level, this indicates that if the laser is shot at these locations, the output voltage of the DUT momentarily drops and does not recover fully until the time of sampling. This might indicate that there is a general structure in every LE that is sensitive to fault injection. However, this structure does not cause semi-permanent bit flips in the LUT, but instead a transient drop in supply and/or output voltage.

Gaining information about fault-sensitive locations in this way allows the attacker to proceed to analyze the effects on the logic function performed by the LUT in detail. For this, she can simply fire a single laser shot into a sensitive region of interest. She can then use the already mentioned counter and clock setup in combination with the oscilloscope to analyze the change in all LUT bits and consequently logic function. An example of such a measurement can be seen in Fig. 2.11. Please note that the bits are given in zero-based numbering. Additionally, because of the implementation of the counter, the output starts with the second bit (01) and wraps around after the 16th bit (15) to output the first bit (00).

In this case, the AND gate originally implemented by programming the bits `0x8000` into the LUT has been changed into a buffer for input D, equivalent to the bits `0xFF00`. It is evident from this that a single laser shot can change multiple bits of the LUT, a fact that also could already be seen by comparing Fig. 2.10a and Fig. 2.10b. Us-

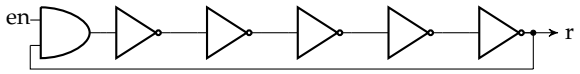


Figure 2.12: A ring oscillator (RO) circuit utilizing five inverters and an AND gate for enable control.

ing a setup like the one presented here, an attacker can easily profile fault-sensitive locations and their exact impact on the logic functions programmed into the device. This knowledge will then later allow her to precisely manipulate the programming of the target device she seeks to attack.

A second example of the usage of the setup for profiling concerns a more complex logic circuit: a ring oscillator (RO) spanning multiple LEs. A ring oscillator is a construct which consists of an odd number of inverters and an AND gate to enable or disable operation, see Fig. 2.12. If we assume that the enable ("en") input of the ring oscillator is initially low, it follows that the output of the AND gate is also low. Because of the odd number of inverters, it furthermore follows that the signal is high at the output of the last inverter and therefore also at the second input of the AND gate. This situation is static as long as enable stays low. However, if enable is set to high, the output of the AND gate will switch to high. This will also cause a change in logic state of all the following inverters, which will propagate through the inverter chain. When the signal has propagated, the output of the inverter chain will now be low. Because of the feedback line leading back to the AND gate, this will cause the AND output to go low again. This change in state will then once more propagate through the inverter chain and the whole process will repeat. As a consequence, the output of the RO will oscillate with a certain frequency.

It is assumed that this time the attacker is interested in simply finding fault injection locations to disable the RO circuit. To test this scenario, an RO was implemented using multiple LUTs which were configured to act as five inverters. In this case, the attacker will output the RO signal from the device and feed it through a low pass filter before connecting it to the image acquisition input of the Phemos. If the RO is running, this will then deliver a signal of half the supply voltage. If the injected fault causes the ring oscillator to stop, the low pass output will either be at ground or at the supply voltage. Employing a similar triggering configuration as used for the LUT bit analysis, the attacker can then acquire a map of all locations which stop the RO, see Fig. 2.13.

For this measurement, 75.5% laser power and 240 s scan time have been used. Again, the fault mapping results have been overlayed onto a reflected light image. The normal state of the RO (running) has been faded out and therefore only fault injection locations which cause the RO to stop show up. In this case, red means the RO stops with its

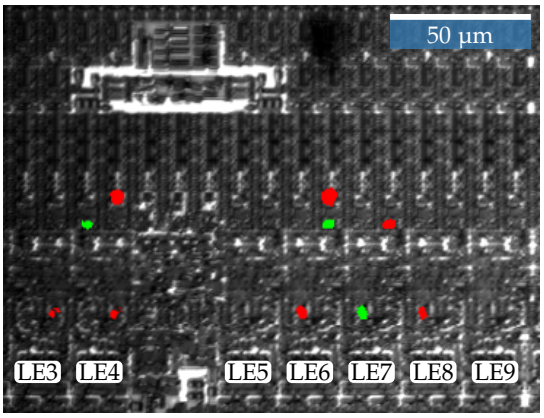


Figure 2.13: Fault-sensitive location mapping results overlayed on a reflected light image of the DUT for a multi-LE ring oscillator implementation. Red denotes a stuck-at-one fault, green a stuck-at-zero fault.

output high, while green means it stops with its output low. It can be seen from this example that the attacker not only gains knowledge about where to disable the RO but can also choose freely with which output value it should stop. It can also be seen that the RO circuit does not utilize LE5 and LE9. This is caused by the automated generation of the implementation by the synthesis software and shows that the attacker can directly identify which LEs are relevant to her target and which are not.

To conclude this section, it can be said that the presented setup has been shown to allow an attacker to easily profile fault injection locations and their precise influence in a short time span. However, since the setup has so far only been used on simple example cases, it is desirable to assess the potential of such an approach on actual security implementations.

2.2.4 *Reconfiguration Attacks Against PUFs*

In this section, the automated fault injection analysis scenarios from the previous section will be evaluated on proof-of-concept (POC) implementations of different security primitives using physically unclonable functions (PUFs) [22, 54] to assess the potential of such attacks. PUFs generate (virtually) unique outputs, so-called responses, to a given set of input bits, known as challenges. Since they exploit device manufacturing variability to generate their outputs, they should be hard to predict or clone. In a simplified way, this can be seen as generating a "silicon fingerprint" unique for every device. This feature of PUFs can be used for different tasks such as random number generation, authentication or key generation.

For assessing the feasibility of fault injection reconfiguration attacks, two PUF implementations were chosen: an XOR arbiter PUF

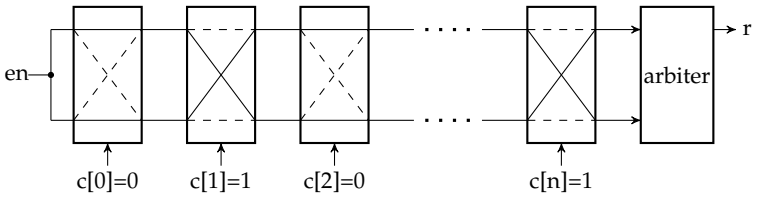


Figure 2.14: A basic arbiter PUF. © 2015 IEEE. [79]

and a ring oscillator (RO) PUF. For both, suitable POC implementations of the core elements were realized. In conjunction with the RO PUF, attacks on RO true random number generators (TRNGs) are also discussed.

2.2.4.1 *XOR arbiter PUF*

In an arbiter PUF, multiple path-selecting stages are interconnected with each other, see Fig. 2.14. Each of these stages has two inputs, two outputs, and a challenge input ("c"). The path that the two input signals take through these stages is selected by the respective challenge input. For each stage, regardless of the state of the challenge input, the propagation paths for both input signals are identical by design. However, due to manufacturing imperfections, they will have a slightly different propagation delay. At the end of the last stage, there is an arbiter element, which detects which signal arrives first, and outputs a binary response accordingly on its "r" output.

To evaluate the response of the PUF, the challenge bits are applied, and shortly after the enable ("en") input is set high. The enable signal now travels simultaneously on both paths through the arbiter PUF. As both paths are identical by design, there is a race condition for the arrival of the two signals. However, due to the manufacturing imperfections, one signal *will* arrive first and generate a zero or one response accordingly. Which signal path is faster will depend on the partial paths selected by the challenge bits. As the variations in propagation delay of the different elements are assumed to be random, a different response will be generated for every challenge. If the variations in propagation delay are random, it also immediately follows that the same implementation, run on a different device, will generate different responses to the applied challenges. A PUF like this can thus be used to authenticate a device [19]. In the naive case, while still at the factory, the PUF is exercised with a number of challenges and its responses are saved into a database. If the device is to be authenticated in the field, a challenge or a number of challenges is sent to the device and its response is compared to the one saved in the database.

For a straightforward implementation of an arbiter PUF on a CPLD, each stage could be built using two digital multiplexers. However, as these multiplexers would be realized by individual LUTs, routing

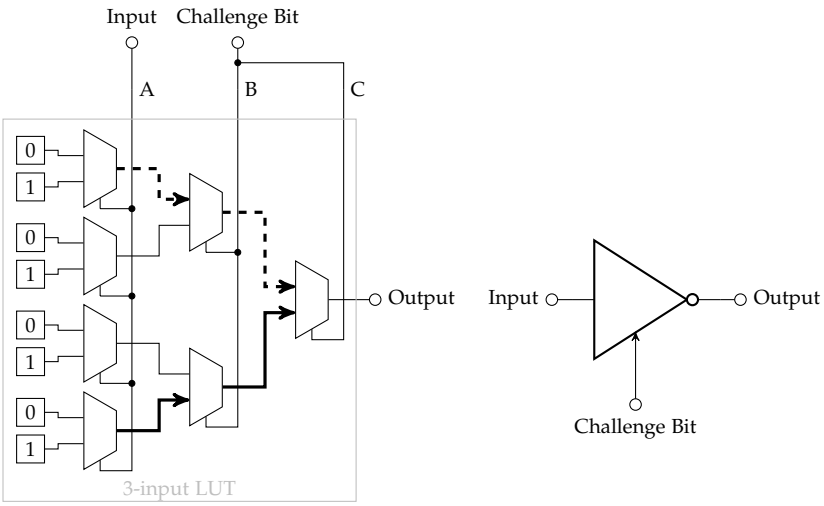


Figure 2.15: Switchable LUT propagation path structure based on [44] (left) and equivalent inverter representation (right). The challenge input can be seen as a select switch for the internal propagation path of the inverter.

constraints for the LUT-to-LUT connections would lead to delay imbalances for the two propagation paths. To address this problem, an approach based on [44] is used in this work.

The basic building block for this approach is an inverter-equivalent structure, whose internal propagation path is switchable, see Fig. 2.15. This structure is explained here with a 3-input LUT as an example, although it can be implemented on any kind of LUT. The LUT input A constitutes the input of the inverter, while all other LUT inputs are connected to the challenge bit, and the SRAM cells of the LUT are set to `0b01010101`. The output of the inverter is simply the output of the LUT. If the state of the input signal is changed, all multiplexers of the first stage will change their output to the inverted state accordingly. This change in output will then either travel along the thick dashed or the thick solid path, depending on the challenge bit. Therefore, the challenge bit decides through which elements the signal will propagate. This structure can thus be seen as an inverter whose internal propagation path can be switched using the challenge input, see Fig. 2.15. Note that it is possible to switch between more than two paths in the LUT, by connecting all inputs other than A to individual challenge bits. This is also the case for the original implementation presented in [44]. However, an implementation with a single challenge bit input is used here for simplicity.

Using this inverter structure, it is now possible to implement an arbiter PUF suitable for a CPLD, see Fig. 2.16. In this case, there are two inverter chains consisting of an even number of inverters connected to the data ("D") and clock ("CLK") inputs of a D flip-flop. Each inverter is realized using the structure of Fig. 2.15. Before exer-

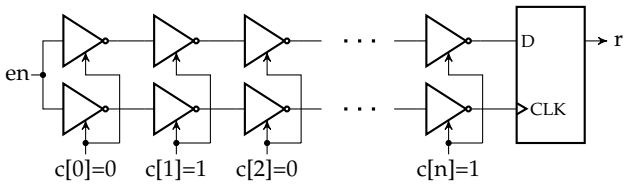


Figure 2.16: An arbiter PUF CPLD implementation. © 2015 IEEE. [79]

cising the PUF, the "en" input is at logic low level. The challenge bits are then applied to select the path elements to be compared. Afterward, the PUF is enabled with the "en" input. The signal now travels down the two inverter chains and arrives at the D flip-flop. If the top inverter chain is faster, the data input will be high when the rising clock edge arrives, and thus the "r" output of the flip-flop will also be high. If the lower inverter chain is faster, data will be low at the rising clock edge and r will therefore also be low. Thus, this structure implements the function of an arbiter PUF. It should also be noted that an implementation like this assures that if the propagation path delays inside the LUT are different by design (Dashed versus solid path in Fig. 2.15.), this does not have negative effects on the PUF, as it always compares the same LUT signal path in the inverters of the top and bottom inverter chain.

Unfortunately, simple arbiter structures like this have been shown to be vulnerable to machine learning (ML) attacks [21, 37]. Using ML attacks, the behavior of a PUF can be learned and modeled, by analyzing the challenges and responses of the PUF, the so-called challenge-response-pairs (CRPs). As a consequence, the responses of the PUF can be predicted and the PUF thus cloned in software. To combat these kinds of attacks, the concept of non-linear XOR arbiter PUFs has been introduced, impairing the effectiveness of pure ML attacks [75]. Such an XOR arbiter PUF consists of multiple ordinary arbiter PUF chains whose responses are fed into an XOR gate to derive the final response. An implementation of such a PUF based on the previously introduced structures can be seen in Fig. 2.17.

Although there are effective ML attacks against XOR arbiter PUFs with a small number of arbiter chains, large XOR arbiter PUFs can still be considered secure against them [64, 65]. Yet, if the potential of reconfiguration attacks to alter logic gates as demonstrated in Sect. 2.2.3 is taken into account, there is an obvious way to attack them: by disabling the inverters highlighted in Fig. 2.17. Using this approach, the clock input of the D flip-flop of the targeted arbiter chains will not receive a clock signal, and thus, all flip-flops except for the top one will always output a zero into the XOR gate. Consequently, the output of the PUF will be identical to the output of the first arbiter chain and this chain can now be learned individually. After learning the first arbiter chain, the attacker can then proceed to reset the device to restore it to normal operation and then learn the

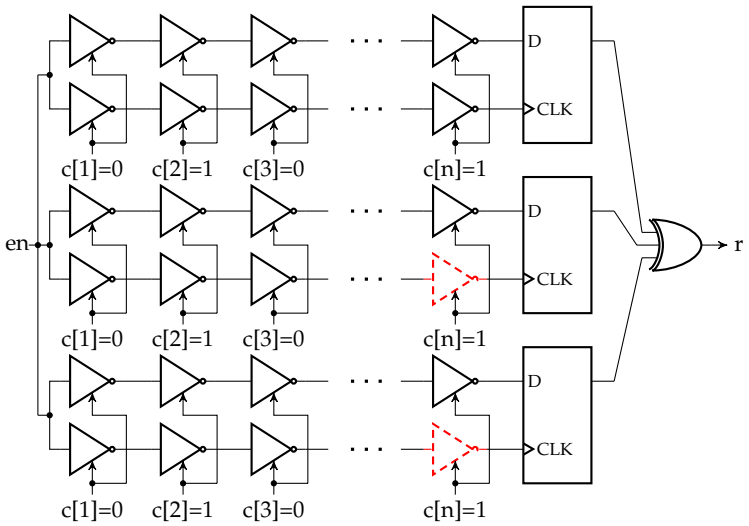


Figure 2.17: A CPLD implementation of an XOR arbiter PUF. The highlighted inverters are potential reconfiguration attack locations. © 2015 IEEE. [79]

second arbiter chain using the same approach. When she has managed to learn all individual arbiter chains this way, the combination of their outputs through an XOR function is trivial.

A detailed mathematical analysis in [79] shows the advantage of such an approach. To learn an XOR arbiter PUF with a maximum variation of delay values of M, n inverter stages and k parallel arbiter chains with an accuracy level of ε and a confidence level of δ, in the case of a combined ML and reconfiguration attack, the maximum number of needed CRPs is:

$$N = O\left(\left(1 + \frac{2}{\varepsilon}\ln(1/\delta)\right)knM^2 + \frac{2k}{\varepsilon}n^2M^4\right)$$

For an attack based solely on ML, the maximum number of CRPs is:

$$N_{XOR} = O\left(\left(1 + \frac{2}{\varepsilon}\ln(1/\delta)\right)n^kM^{2k} + \frac{2}{\varepsilon}n^{2k}M^{4k}\right)$$

It can thus be seen that the combined ML and reconfiguration attack will allow the XOR arbiter PUF to be learned with a significantly smaller number of CRPs for a large number of parallel chains (k), as opposed to the conventional approach.

To prove the feasibility of such an attack, one needs to prove that reconfiguration attacks are actually capable of selectively disabling the targeted inverter chain, while not affecting the rest of the implemented circuitry. It should be noted that for this attack to work, it is irrelevant which specific inverter in the chain is targeted, and what specific logic function it performs after the fault injection. As long as the attacker is able to change any inverter of the chain to any logic function which does not propagate the signal, her attack will be successful.

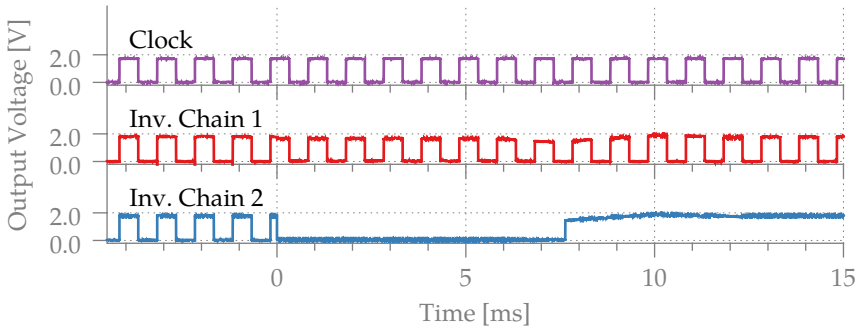


Figure 2.18: Selective deactivation of an inverter chain in the proof-of-concept implementation.

To demonstrate that such a modification is possible using the automated fault mapping approach, two inverter chains using the discussed implementations were realized on the Altera MAX V DUT already described in Sect. 2.2.3. Each chain consisted of eight inverters. As the DUT contains 4-input LUTs, the fourth ("D") input was additionally shorted with inputs B, C, and the challenge bit input, compare Fig. 2.15. The inputs of the inverter chains were then connected to function generator A of the setup, which supplied a 50% duty cycle 1 kHz square wave. The outputs of the inverter chains were then routed to pins of the device and monitored using the oscilloscope, compare Fig. 2.3. Using knowledge about fault-sensitive locations from a mapping approach similar to the one demonstrated in Fig. 2.10, it was then straightforward to inject faults into the LEs of the targeted inverter chain, causing its deactivation. For this, continuous power was supplied using the Toellner power supply with 1.8 V supply voltage and an 18 mA current limit, while the laser power was set to 75.5%. The results of the fault injection can be seen in Fig. 2.18. It is evident that inverter chain 2 is immediately deactivated when the laser is shot at 0 ms, while inverter chain 1 continues to propagate the input signal. Therefore, it is proven that it is possible to selectively deactivate a single inverter chain, which makes the described attack against the XOR arbiter PUF feasible. It should be noted that inverter chain 2 changes its logic state about 7.5 ms after the actual fault injection. A detailed analysis of this phenomenon and its underlying mechanisms will be given in Sect. 2.2.5. Nevertheless, this does not hinder the discussed attack, as the measurements have shown that inverter chain 2 will indeed stay inactive after this final logic level change until the device is reset. As a typical scan for the automated mapping for profiling the fault-sensitive locations for this attack takes only 240 seconds, this demonstrates the potential of fault injection reconfiguration attacks carried out using this approach.

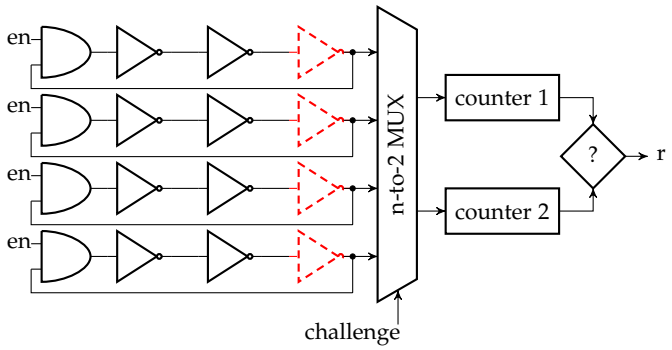


Figure 2.19: A ring oscillator (RO) PUF. The highlighted inverters are potential reconfiguration attack locations. © 2015 IEEE. [79]

2.2.4.2 *RO PUF and RO TRNG*

The second PUF that was considered for assessing the feasibility of fault injection reconfiguration attacks was a ring oscillator (RO) PUF [75]. Similarly to the arbiter PUF, the RO PUF also exploits intrinsic timing differences of circuit elements of the implementation device. However, instead of a race condition between two propagating signals, the RO PUF uses the frequencies of ring oscillators. An example of an RO PUF consisting of four ring oscillators can be seen in Fig. 2.19. On the left-hand side of this figure, four identical ring oscillator circuits are visible. As already discussed in Sect. 2.2.3.3, the outputs of the ring oscillators will change their state periodically, due to the feedback from the final inverter into the AND gate. The frequency of each RO will depend on the round trip propagation delay of the logic gates which make up the RO. Even though the elements of all ROs are designed identically, there will be small process variations, which cause each RO to run at a slightly different frequency. As these variations are expected to be random, the frequencies of the individual ROs can be expected to be characteristic for each manufactured device. To now evaluate these characteristics, the complete PUF also contains an n-to-two-multiplexer, two counters, and a comparison element, in addition to the already mentioned ROs, see Fig. 2.19. To generate the characteristic PUF response, the challenge bits applied to the multiplexer first cause the selection of two ROs for comparison. Through this, the selected ROs are connected to counter one and two respectively and are then enabled. After a certain amount of time has passed, the ROs are disabled and the values of the two counters are compared. Depending on which RO frequency was higher, counter one or counter two will have a higher value, and the result of the comparison is output accordingly. This bit then constitutes the PUF's response to the applied challenge bits.

A security primitive similar to the RO PUF is the RO true random number generator (TRNG) shown in Fig. 2.20. In this case, the output

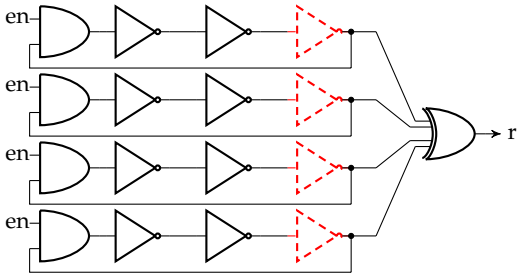


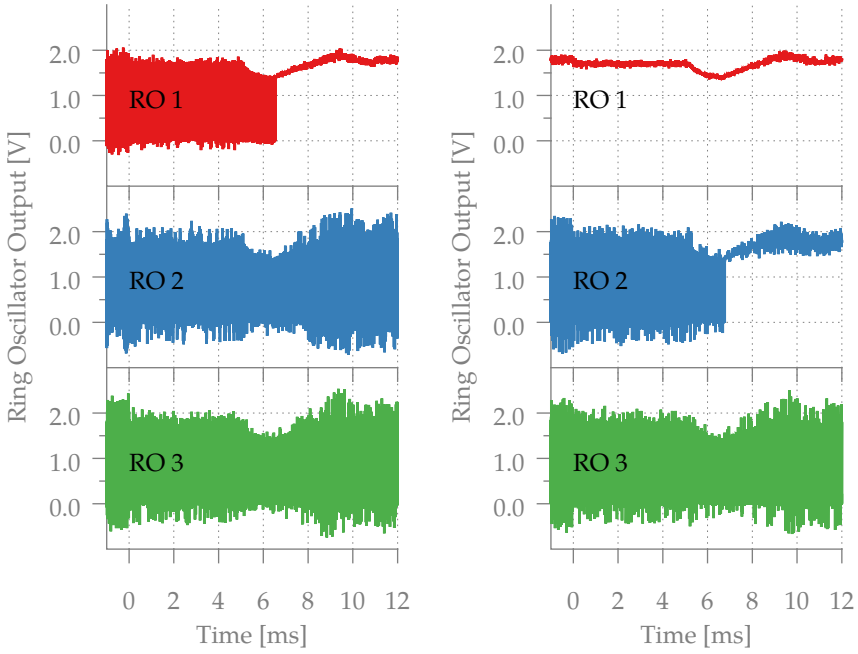
Figure 2.20: A ring-oscillator-based true random number generator. The output r is sampled regularly to generate a random stream of bits [76]. The highlighted inverters are potential reconfiguration attack locations.

of multiple ROs is combined by an XOR and sampled periodically to generate random bits [76].

A relevant metric for both circuits is the entropy of their output, which is important if they are used as the basis for cryptographic functions. If an attacker wants to selectively lower the entropy of one of these two circuits, she can do so via reconfiguration attacks. For the PUF, if she disables one of the inverters highlighted in Fig. 2.19, the corresponding RO will become inactive, i.e. have a frequency of zero. If multiple ROs are disabled, this will lower the entropy of the generated PUF response. To be precise, as discussed in [79], the entropy of the attacked PUF will be $log_2((N-i)!)$, with N being the original number of ROs and i the number of disabled ROs. In the case of the TRNG, the attacker can choose to only lower the entropy, or take an even more dramatic approach. If she disables all but one or even all of the ROs, this will make the output periodic or static respectively, allowing her to predict the output of the TRNG.

To prove that manipulation and selective RO deactivation is actually possible using the automated fault injection location mapping approach, three ROs were implemented in the MAX V DUT. The ROs consist of five inverters, of which each is implemented in a single LUT. The RO outputs were again routed to pins of the device and the RO activity could thus be monitored using the oscilloscope. The mapping and attack processes already described in Sect. 2.2.3 were then carried out and a shot was fired at the sensitive locations for the first RO with a laser power of 75.5%. Again, for this final fault injection, the DUT was supplied using continuous 1.8 V supply voltage and an 18 mA current limit.

Fig. 2.21a shows the effect of this shot on all ROs. It is evident that the first RO is disabled, while the other two ROs continue their operation. After this successful fault injection, the motorized DUT stage of the Phemos was used to navigate to the next RO. During navigation, the laser power was set to a level that just barely allowed a live image with the LSM to be acquired. This was done to not induce unintentional faults into the surrounding circuitry. After arrival at the



(a) First shot, targeting RO 1 (b) Second shot, targeting RO 2

Figure 2.21: Selective deactivation of ring oscillators in the proof-of-concept implementation.

second RO location, a second shot was fired. Fig. 2.21b shows that this successfully disables the second RO, while the third RO still oscillates. This proves that selective deactivation of ROs is possible using the automated mapping approach and therefore makes the discussed attacks against RO PUFs and RO TRNGs feasible. Performing the needed fault injection location mapping merely required 240 s of acquisition time per scan, allowing to map the sensitive locations of all ROs in a few minutes. It should be noted that while the actual laser shot is fired at 0 ms, the targeted ROs do not stop their operation until approx. 6.5 ms afterward. Additionally, there seems to be some influence on the output voltage of the ROs which are not targeted as well. A detailed discussion of this phenomenon will be given in Sect. 2.2.5. Nevertheless, the discussed attack can be considered successful.

2.2.5 *Analysis of Fault Mechanism*

This section will present a more in-depth analysis of the mechanisms involved in the fault injection in this particular device. As was already mentioned in Sect. 2.2.4, deactivation of the targeted logic sometimes happens several milliseconds after the actual laser shot, compare Fig. 2.21. Additionally, even if the logic is deactivated instantly, there might be logic changes several milliseconds later, compare Fig. 2.18. As the laser pulse duration is only 200 ns, it seems improbable that

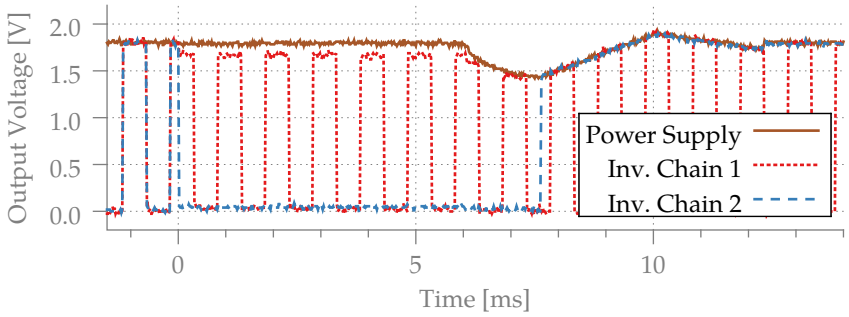


Figure 2.22: Comparison of the inverter chain outputs and the power supply voltage during a reconfiguration attack on inverter chain 2.

the changes are caused by the injected photocurrent alone, but are instead a result of a more complicated mechanism. To investigate this, the experimental data of Sect. 2.2.4 was revisited.

During this, it first stood out that the power supply voltage seems to be influenced by the fault injection, although with a considerable delay. Fig. 2.22 shows an example of this behavior, where fault injection into an inverter chain was performed. After the fault injection, which immediately disables inverter chain 2 at 0 ms, the supply voltage stays constant for about 6 ms and then drops with what looks like a capacitor discharge curve. As soon as the voltage has reached 1.44 V at 7.6 ms, inverter chain 2 changes its logic state, and the supply voltage starts to rise linearly. It then slightly overshoots the set voltage of 1.8 V to 1.9 V, goes down linearly to 1.73 V and afterward jumps directly to 1.8 V. After this, the power supply behaves normally.

Apart from this behavior, it can also be found that the output voltages of the device acquire an offset with regard to the supply voltage. For example, the output voltage of inverter chain 1 can be seen to immediately drop from 1.8 V to 1.67 V at 0 ms in Fig. 2.22. Interestingly, the low-level output voltages also acquire an offset, although it is smaller and can hardly be seen in the figure. In the case of inverter chain 1, the low-level voltage changes from 0 mV to 28 mV at 0 ms. When the power supply voltage starts to drop at about 6 ms, both of these offsets can be seen to be decreasing with regard to the power supply voltage. This is visible for the high-level voltage of inverter chain 1 in Fig. 2.22. A detailed analysis of the data shows that at the point where inverter chain 2 changes its logic state at 7.6 ms, these offsets immediately disappear.

In the case of the attack on the ring oscillators, the same behavior of the power supply and device output voltages can be observed. Additionally, all ROs decrease their frequency immediately after the laser shot. Fig. 2.23 shows this for the targeted RO 1 as well as for RO 2 and RO 3. While RO 1 will stop its operation after 6.5 ms, RO 2 and RO 3 will continue their operation normally after the disturbances caused by the fault injection. It should be noted that RO 2 and RO 3

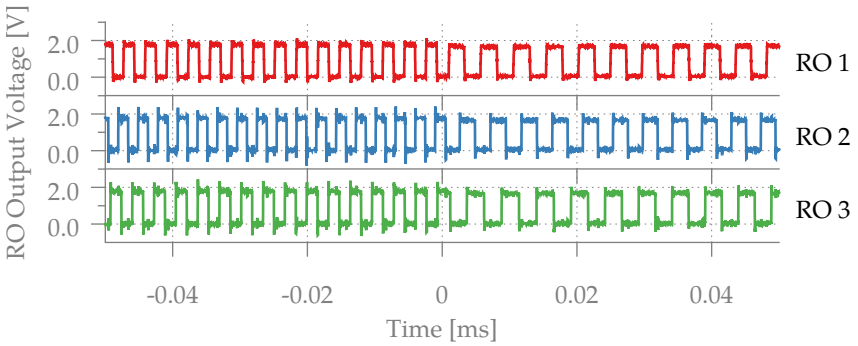


Figure 2.23: Detailed view of the activity of all ring oscillators at the moment of fault injection.

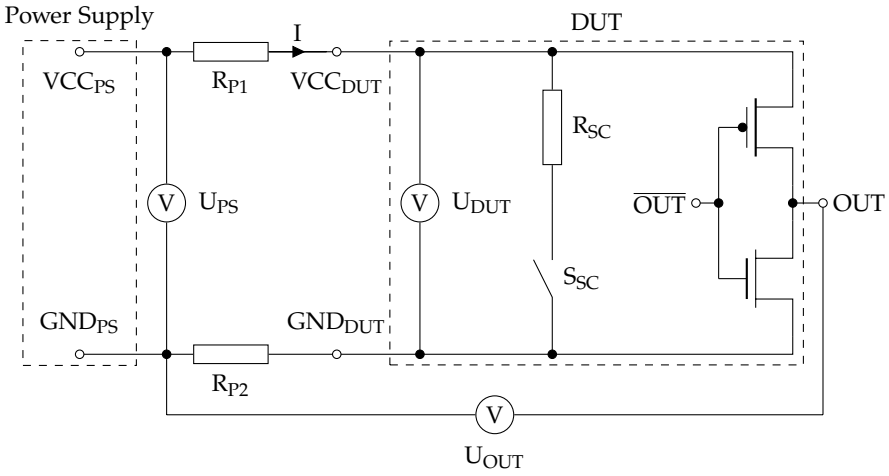


Figure 2.24: Schematic of the model employed for fault mechanism analysis.

are implemented at different physical locations inside the device and should therefore not receive any laser radiation.

A possible explanation of this behavior is that the laser shot creates a short circuit of some sort. This is suggested by the drop in supply voltage after 6.5 ms and especially the immediate creation of offsets in device output low-level and high-level voltages. If we assume that the power supply wiring has some parasitic resistance on ground as well as on the supply line, an increased current in the device will cause offset voltages to be generated on both lines. As the power supply is located outside of the microscope, and there are about three meters of wiring and several connectors required to connect the DUT, this seems to be a reasonable assumption.

To determine if a short circuit can explain the observed behavior, a corresponding model was developed. Taking the particular experiment setup into account, this leads to the equivalent circuit shown in Fig. 2.24. Here, the cable resistance is modeled by R_{P1}/R_{P2} and the digital output (OUT) of the device is represented by a CMOS in-

verter. The voltages U_{PS} and U_{OUT} represent the supply and output voltages as measured during the experiments, while U_{DUT} represents the internal device supply voltage. The short circuit itself is modeled by switch S_{SC} and the equivalent short circuit resistance R_{SC}.

When the short circuit is activated, the current I flows from VCC_{PS} through the parasitic resistance R_{P1}, into the short circuit and finally back via R_{P2} to GND_{PS}. Because of the current, there will be a voltage drop across both R_{P1} and R_{P2} according to Ohm's law, effectively lowering the device's internal supply voltage U_{DUT}, as

$$U_{DUT} = U_{PS} - I \cdot R_{P1} - I \cdot R_{P2}. \tag{2.1}$$

This can explain the reduced RO frequencies after the laser shot, as previous experiments have shown that a lower internal supply voltage causes the ROs to slow down. As the power supply voltage U_{PS} is measured directly at the supply, no change would be observed in the measurements at this point in time, although the internal supply and ground levels of the device would have changed. This agrees with the behavior seen during the experiments at 0 ms. However, with this model, the output voltage U_{OUT} of the device would reflect the changed internal levels. If the digital output OUT is low, the output inverter directly connects OUT to GND_{DUT}. Therefore, U_{OUT} will effectively measure the voltage drop across R_{P2}, as

$$U_{OUT_{Low}} = I \cdot R_{P2}. \tag{2.2}$$

If the output is high, OUT is connected to VCC_{DUT} and will acquire an offset equal to the voltage drop across R_{P1} as

$$U_{OUT_{High}} = U_{PS} - I \cdot R_{P1}. \tag{2.3}$$

In both cases, the offsets would appear at the output as soon as a short circuit causes an increased supply current. This agrees well with the behavior seen during the experiments.

The short circuit based model is also able to explain the lowering of the supply voltage at about 6 ms. This can be simply interpreted as an attempt of the overcurrent protection of the power supply to reduce the current. As the supply voltage drops, the model would predict the current through the short circuit to decrease. As a consequence, the voltage drops across the parasitic resistors R_{P1} and R_{P2} will be lowered. This would lead to decreased offsets for U_{OUT} during the lowering of the voltage, see Eq. 2.2 and 2.3, which is exactly what can be observed in the experiment, see Fig. 2.22. The disappearance of the offsets at 7.6 ms would then suggest that here the short circuit condition is somehow resolved. This agrees well with the fact that the power supply starts to increase the voltage again at this point, see Fig. 2.22.

Therefore, the behavior seen during the experiments can be explained by this short circuit based model. There are two probable

mechanisms to explain the creation of such a short circuit: a simple latch-up and a short circuit caused by faulty configuration data. In the first case, the injected photocarriers activate a thyristor-like parasitic structure and a low resistance path between VCC_{DUT} and GND_{DUT} is created. In the second case, faults accidentally injected into adjacent configuration memory cells could create a connection between a high-level signal and a low-level signal through routing resources. This would also create a low-resistance connection between VCC_{DUT} and GND_{DUT}. Usually, the synthesis software generating the configuration data for programmable logic devices takes care of preventing such connections during the design phase. However, they are still possible at the hardware level and are known to be generated by faulty configuration data [1].

Both these causes can explain why the short circuit is resolved when the voltage drops to 1.44 V at 7.6 ms. In the first case, if a latch-up was the cause, at this point the current might be decreased below the hold current needed to sustain the latch-up condition. In the second case, it can be assumed that the low supply voltage of 1.44 V causes additional SRAM cells to flip until the short circuit is resolved because of routing or logic changes. An indication that this might be the case is that the device datasheet [3] states: "If there is a V_{CCINT} voltage sag below 1.4 V during user mode, the functionality of the device is not guaranteed [...]." This suggests that 1.4 V is the voltage at which the SRAM cells of the configuration memory fail.

Of these two explanations, the one involving SRAM bit flips seems more probable. Firstly, latch-ups can often only be resolved by removing power completely and are additionally often destructive. Secondly, modern CMOS devices have been designed to be robust against latch-ups caused by, for example, ionizing radiation. Furthermore, the minimum allowed voltage for the configuration memory and the voltage at which the resolving of the short circuit occurs match very closely. Additionally, the latch-up based model is not able to explain the output state change occurring in the inverter chain at exactly the point when the short circuit is resolved. The stopping of the ROs 6.5 ms after the actual laser shot can also only be explained by additional SRAM cells flipping because of the low supply voltage. In total, SRAM bit flips in the routing or logic configuration memory seem to be the more probable explanation for the creation and resolving of the short circuit condition. Consequently, this hypothesis is assumed through the rest of this section.

What remains to be explained is the characteristic shape of the power supply voltage curve during and after the short circuit condition. In particular, it needs to be determined if it is caused by the power supply overcurrent protection circuitry alone or if there are some other effects in the DUT which need to be accounted for. To determine this, a resistive load was applied to the power supply and

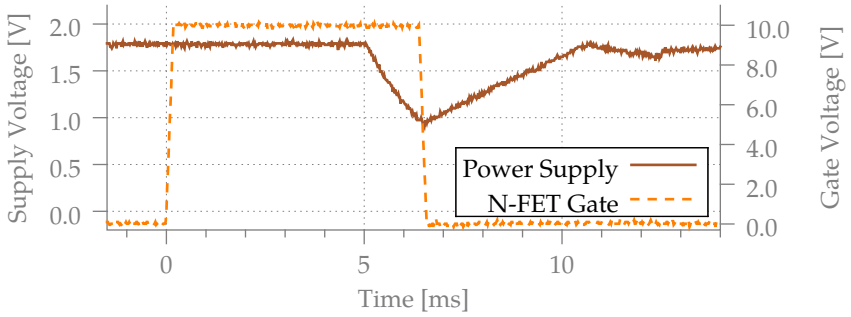


Figure 2.25: Power supply overcurrent behavior with a 17.5 Ω load, a set voltage of 1.8 V and 18 mA current limit. The load is enabled and disabled using a BUZ11 n-channel power MOSFET.

the resulting voltage curve was measured. To enable and disable the load, a BUZ11 n-channel power MOSFET was used. The results of this measurement can be seen in Fig. 2.25. For this, the supply was set to 1.8 V and 18 mA current limit, as in the fault injection experiments. The gate of the MOSFET was driven with 10 V for 6.5 ms. A load resistance of 17.5 Ω was empirically determined to give a behavior which most closely resembles the one seen in the experiments, compare Fig. 2.22. It is evident that the power supply requires 5 ms to show any reaction at all. After this, it starts the process of lowering the voltage, during which the short circuit is removed by disabling the MOSFET gate. The behavior seen previously of overshooting and settling to the set voltage is then also observed. Therefore, the characteristic shape of the power supply voltage curve in the experiments can be concluded to be caused solely by the overcurrent protection circuitry.

In view of this, the current model of the fault mechanism can be summed up as follows: when the laser is shot, it immediately causes some configuration memory cells to flip. This then creates a short circuit through a faulty routing configuration. In the case of the inverter chain, LUT bits seem to be affected as well, as the inverter chain is disabled immediately. In the case of the ROs, there seems to be no change in LUT bits at this point, as the ROs continue their operation. The power supply then reacts to the overcurrent condition caused by the short circuit by lowering the voltage. When the minimum data retention voltage (DRV) of the SRAM configuration memory cells is reached, additional configuration changes occur. These resolve the short circuit caused by the faulty routing through additional random routing changes. Furthermore, changes in the LUT bit memory cells then cause logic state changes in the case of the inverter chain, see Fig. 2.22, and disabling of RO 1 in the case of the ring oscillators, see Fig. 2.21a. The power supply then starts to recover from the overcurrent condition by increasing the voltage back to nominal levels and normal operation continues.

This model is able to explain the observed behavior well, although there remains one open question: if the voltage drops low enough for the SRAM configuration memory cells to fail, why is only the targeted structure affected? A global lowering of the voltage below the SRAM data retention voltage would suggest that changes could occur anywhere. However, as demonstrated in the previous sections, it is always the targeted structure that is altered, while the other implemented structures continue to operate. A possible explanation for this is a dependency of the DRV on temperature. As the short circuit will be created in the proximity of the laser spot, the device will heat up in this area because of the current. If the voltage at which the SRAM fails is higher for higher temperatures, the cells in the targeted area would fail first. As a result, the secondary faults caused by the low voltage would only be generated in the targeted area. Indeed, experimental results in [61] show an increase in DRV of about 8% when the temperature is raised from 27 °C to 100 °C in a 130 nm SRAM device, supporting this theory. However, other so far unrecognized effects which might still be lingering several milliseconds after the actual laser shot cannot be excluded as the cause without further experiments. In the future, a measurement of DRV over temperature for the MAX V DUT could bring additional hints regarding this aspect of the fault mechanism. However, due to the external inaccessibility of the configuration memory cells, these could only be performed indirectly via operational tests of a given implemented design.

2.3 CHAPTER CONCLUSION

This chapter has first outlined the potential of reconfiguration attacks using laser fault injection (LFI) on programmable logic devices (PLDs) and has then developed a suitable setup for the evaluation of such attacks. The development was carried out under the assumption that the attacker has access to an infrared laser scanning microscope (LSM). The setup presented is able to detect suitable fault injection locations in a fully automated way. The functionality of the setup was demonstrated by using it to profile the locations for reconfiguration attacks on simple logic gates and circuits, such as AND gates and ring-oscillators (ROs). The capabilities of the setup were then taken into account during the development of reconfiguration attack scenarios against physically unclonable functions (PUFs). Said attack scenarios were then carried out against proof-of-concept implementations, demonstrating attack feasibility against XOR arbiter PUFs, RO PUFs and RO true random number generators (TRNGs). It was shown that the automated mapping approach used can acquire the information needed for successful LFI attacks in a time span in the order of minutes. This demonstrates that the attack potential when the attacker has access to a suitable LSM is high.

Furthermore, a detailed analysis of the underlying fault mechanism was performed. This led to a suitable model being developed which agrees well with the observed behavior. It was shown that the simple single-bit-flip model usually associated with LFI does not hold for the experiments performed on this particular device. Instead, it seems that primary faults in routing lead to a short circuit condition, which causes additional faults because of an insufficient supply voltage during the short circuit. These secondary faults however only appear in the targeted area, presumably due to the influence of localized heating caused by the short circuit. A thermal increase of configuration memory data retention voltage (DRV) was determined to be the most probable cause for this behavior. However, to fully exclude other causes, additional measurements would need to be performed in future work.

3 STIMULATION ATTACKS

Laser stimulation is commonly used in failure analysis and is a mature technique for isolating faults. Laser stimulation uses laser radiation to influence device parameters and usually monitors the changed properties indirectly, for example via the current consumption. Analysis of the spatial stimulation response map then yields information about the fault cause and location.

However, in a security context, the same information can be used by an attacker to extract data or secrets from the device. In contrast to the fault injection approach discussed in Chapter 2, such attacks do not disrupt normal device operation and do not cause data changes. Therefore, they are much harder to detect.

This chapter will assess the attack potential of stimulation attacks under the assumption that the attacker has access to a standard failure analysis microscope. In particular, the extraction of memory contents from static random-access memory (SRAM) and key memory will be evaluated. Some of the results and figures are published or are pending publication in [34, 42].

3.1 LASER STIMULATION TECHNIQUES

In failure analysis (FA), a number of techniques have been developed which use laser radiation to influence the device under test (DUT). These are referred to as laser stimulation (LS) techniques and measure changes in device parameters in response to the incident laser radiation. This is usually performed by scanning a region of interest with a laser beam and monitoring the device parameters, see e.g. Fig. 3.1. As the stimulated area of the DUT behaves differently when it is affected by faults, this allows FA engineers to isolate the fault cause.

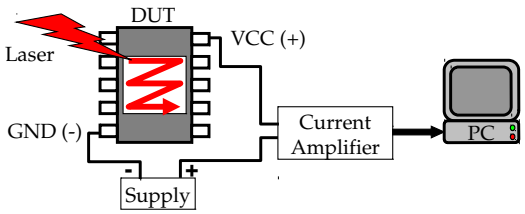


Figure 3.1: Supply current monitoring laser stimulation setup. Depending on the laser wavelength, thermal or photoelectric stimulation can be applied. Figure based on [6]. [42]

monitored parameter	stimulation effect	
	photocarriers (PLS)	thermal (TLS)
current	OBIC optical beam induced current	OBIRCH optical beam induced resistance change
voltage	LIVA laser induced voltage alteration	TIVA thermally induced voltage alteration
operating parameters (pass/fail)	LADA laser assisted defect localization	SDL soft defect localization

Table 3.1: Comparison of common laser stimulation techniques used in failure analysis.

The stimulation effect employed depends on the laser wavelength. If the photon energy is larger than the silicon bandgap, photocarriers will be generated in the semiconductor. This is referred to as photoelectric laser stimulation (PLS). If the energy is smaller, the effect of the stimulation is mainly localized heating. This is referred to as thermal laser stimulation (TLS). Both the photocarriers as well as the heating then influence the parameters of the device.

The monitored parameters are often simply voltage or current at a specific device pin. However, more complex monitoring such as the pass/fail result of tests performed on the device is also used. As this chapter will focus on current monitoring techniques, voltage and operating parameter monitoring techniques will not be discussed here. However, more information about them can be found in [12–14, 63]. For reference, some common laser stimulation techniques are given in Tab. 3.1, grouped by the stimulation effect as well as the monitored parameter.

A simplified example of a supply current monitoring laser stimulation setup is given in Fig. 3.1. In this setup, the device is biased with a specific supply voltage and the current between the supply pins is monitored via a current preamplifier. The exposed silicon backside is then scanned by a laser beam which penetrates the silicon and stimulates the structures inside the device. A PC simultaneously samples the current preamplifier output and plots it as a 2D map of the device's response to the stimulation. By choosing a suitable wavelength, either thermal or photoelectric stimulation can be applied.

An example of a current monitoring TLS technique is optical beam induced resistance change (OBIRCH), which can be used to localize shorts in a device. For this, the device is biased with a small voltage and the current flowing is monitored. When the thermal stimulation heats up a part of the device, its resistance will change due to the temperature change. If that area is carrying a significant part of the short-circuit current, this will change the overall current and can be detected. As the current change is often most pronounced at the cause of the short, it can easily be localized.

A current monitoring PLS example is the use of optical beam induced current (OBIC) to detect faulty connections at p-n junctions. Here, the device supply pins are biased at zero volts and the current is monitored. In this case, every p-n junction connected between the supply pins will effectively act as a photodiode when stimulated by photocarriers. That is, every connected p-n junction will be detectable by a current generated on the supply pins. A faulty connection to one of the p-n junctions will then manifest itself in the absence of such a current.

These and similar techniques have in the past mainly been used to localize defects. However, the authors of [67, 73] demonstrated that photoelectric LS using visible 639 nm/650 nm laser radiation is able to extract data from SRAM and analyze its operation. Yet, as this technique was applied on large technologies (e.g. 900 nm) through the frontside, it is unfeasible on modern devices, as these absorb or reflect the entering light in the numerous frontside metal layers. Nevertheless, more recent publications have demonstrated that when applying thermal LS through the device backside, the stimulation response of the SRAM is data dependent [9, 49]. Although these works discussed the possibility of data extraction, they did not interpret the resulting response patterns apart from single bits and did not demonstrate actual data recovery, especially not on a full memory scale. However, as such data recovery would pose the risk of extraction of key data and other secrets from the memory of ICs, an assessment of the feasibility of memory extraction using these techniques is desirable.

3.2 THERMAL LASER STIMULATION OF SRAM MEMORY

This section aims at giving the necessary background information on SRAM TLS and presenting a suitable setup for the full memory analysis attempted in later sections. The basic principles of thermal SRAM stimulation will first be reviewed and then a suitable setup for TLS memory analysis will be presented. Finally, exemplary TLS responses will be discussed to illustrate the data dependency of the stimulation response.

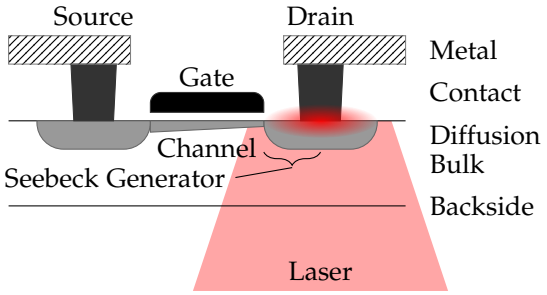


Figure 3.2: Seebeck voltage generation in a MOSFET transistor. Figure based on [9]. [42]

3.2.1 *Principle*

To illustrate the data dependency of TLS data from SRAM memory cells, it is first necessary to understand the behavior of a single MOSFET transistor under stimulation, which is shown in Fig. 3.2. In this case, the drain of the transistor is heated by the laser beam, creating a temperature gradient, which in turn causes a diffusion of carriers [23, 52]. As discussed in detail in [9, 49], this effectively leads to the generation of a voltage source between the drain's metal contact and the channel of the MOSFET, which is also referred to as a "Seebeck generator". If the transistor is on (low-ohmic channel), this Seebeck generator is basically connected between source and drain. If it is off (high-ohmic channel) one terminal of the Seebeck generator is effectively floating. A corresponding situation occurs if the source is stimulated. However, the voltage generated between source and drain will have an opposite sign, due to the changed orientation of the temperature gradient [49]. Due to the doping, the sign will also change for p- or n-type MOSFETs.

Therefore, it can be seen that the Seebeck generator can only act on devices connected to source or drain if there is a low-ohmic channel. Furthermore, the sign of the voltage will change depending on which side of the MOSFET is stimulated. The sign will also change with the type (PMOS/NMOS).

With these principles in mind, it is now possible to analyze the data dependence of the TLS response of a single SRAM cell. Fig. 3.3 shows a schematic of the basic memory element of a typical SRAM cell consisting of two cross-coupled inverters. For simplicity, the connections and transistors for read/write access have been omitted. It can be seen that this circuit will keep one of two states due to the cross-coupling. If, as shown in the schematic, the N1 NMOS transistor is on, it will pull the gates of the right-hand side transistors (P2 and N2) low. This will turn the P2 PMOS transistor on and the N2 NMOS transistor off, which pulls the gates of the left-hand side transistors (P1 and N1) to high level. This will keep P1 off and N1 on, thus keeping the cell's state stable. It is evident that by applying

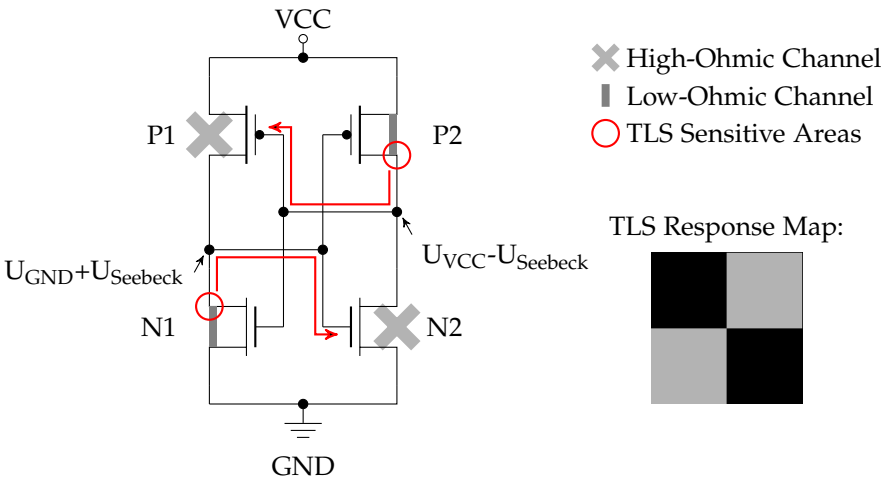


Figure 3.3: SRAM cell under thermal stimulation and expected simplified TLS response map under the assumption of transistor size ≈ beam diameter. Figure based on [9].

suitable voltages to the gates of the transistors, the cell can be flipped into its inverted state. In this case, P1 and N2 would be on and the cell's state would be kept stable by the same mechanism. These two states can then be used to store either a "1" or a "0" bit in the SRAM cell. Furthermore, for ideal transistors, there would be no current draw between VCC and GND when the cell is stable.

However, if thermal stimulation is applied to the cell shown in Fig. 3.3, Seebeck voltages will be generated at the transistors. The absolute voltage generated at the respective transistor is denoted here as $U_{Seebeck}$. Following the previous reasoning, for high-ohmic transistors the Seebeck generators will have no chance to act upon the circuit, as they are floating. However, the voltages generated at N1 and P2 can. For example, a stimulation at the highlighted drain of P2 will cause the voltage at its drain to decrease to $U_{VCC} - U_{Seebeck}$. This will then act upon the gate of P1 (red arrow), causing a slight decrease of its resistance via exponential sub-threshold operation. As N1 is already on, this will lead to a leakage current between VCC and GND. As the cell is symmetric, a similar behavior can be observed when stimulating the highlighted drain of N1. In this case, the decreased resistance of N2 would also cause an increased VCC-GND leakage current. This current can be expected to lie in the nanoampere range [49]. From this, it can be reasoned that if the area of the SRAM cell is scanned in a TLS setup and the current consumption is plotted over the X/Y position, a TLS response map similar to the one depicted in Fig. 3.3 can be expected. If the laser beam diameter is approx. equal to the transistor size, the areas of the sensitive transistors will be brighter due to the grayscale-encoded current consumption. The insensitive transistors, on the other hand, will be darker. From this TLS map, the

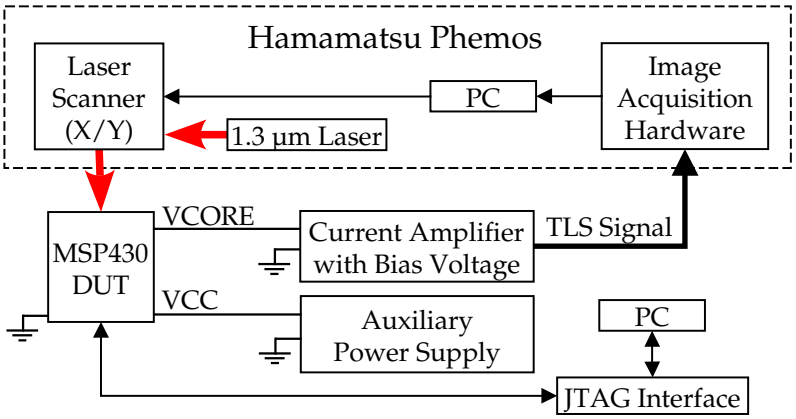


Figure 3.4: Block diagram of the setup used for thermal laser stimulation. Reprinted with permission of ASM International. All rights reserved. [34]

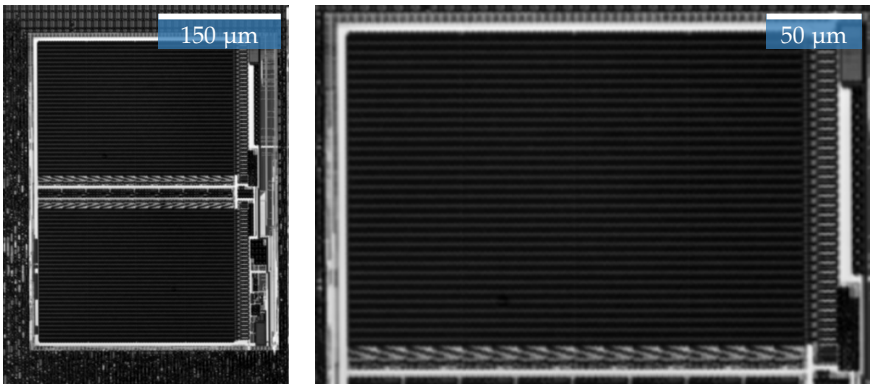
current bit state of the cell can be deduced, as the opposite bit state would have an inverted TLS response map.

Apart from single bit extraction, the 2D response maps generated by TLS stimulation of larger areas have the potential to extract the full content of SRAM memory blocks. As this surely poses a threat to sensitive data contained in RAM, an assessment of the feasibility and capabilities of such an approach seems to be beneficial. For such an evaluation, the development of a suitable experimental setup is needed first.

3.2.2 *Setup*

Fig. 3.4 shows the setup developed to allow for TLS signal acquisition. The setup uses a 1.3 µm laser to stimulate a Texas Instruments MSP430F5131 microcontroller using a Hamamatsu Phemos-1000 failure analysis laser scanning microscope. The laser has a nominal input power into the system of 100 mW. The MSP430 device under test (DUT) is equipped with 1 KB of SRAM and manufactured in 180 nm technology. To allow for optical access to the DUT, a semi-invasive backside approach has been taken. For this, only the backside packaging material and the metal chip carrier were removed and no thinning or polishing was applied.

Fig. 3.5 shows a reflected light image of the DUT SRAM area acquired with the Phemos. Later analysis revealed that only the top block of the SRAM area is active. This is due to the fact that all MSP430 devices of this family (MSP430F51x1 and MSP430F51x2 devices) seem to be based on a generic silicon die with all possible features present. For less-capable devices, some of these are then simply disabled before packaging and labeling. This assumption is supported by the fact that the die inside the MSP430F5131 package



(a) Full SRAM area (b) Active block only

Figure 3.5: Reflected light image of the SRAM of the MSP430. Only the top block containing 1 KB is active. Reprinted with permission of ASM International. All rights reserved. [34]

is actually labeled MSP430F5172, which is the most feature-rich chip of the device family. The experiments presented here thus solely take place in the active SRAM area of Fig. 3.5b.

The MSP430 is connected to a JTAG debugging interface to allow for SRAM content manipulation. This can either be done directly via debugging commands or a program can be sent to the flash memory of the device, which then manipulates the SRAM. Alternatively, the DUT memory can be left at its uninitialized startup values.

During TLS experiments, the DUT VCC rail is connected to an auxiliary power supply (Toellner TOE8732), see Fig. 3.4. VCC powers all parts of the device except for the core which contains the SRAM memory. This core is supplied via an internally generated VCORE voltage, which is also available externally at a pin. To this VCORE supply net, a Stanford Research Systems "SR570" current amplifier is connected. The SR570 bias voltage is set slightly higher than the internal VCORE voltage setpoint. This leads to a significant amount of the VCORE current consumption being supplied via the preamplifier and not internally via the VCC net. The SR570 then converts the VCORE current consumption into a proportional voltage, which constitutes the actual TLS signal. This signal is then fed into an auxiliary input of the Phemos image acquisition hardware.

During TLS response acquisition, the 1.3 µm laser source of the Phemos is focused through the silicon backside of the DUT to reach the active area. The beam is then moved over a region of interest using galvanometric scan mirrors while the TLS signal of the current preamplifier is sampled simultaneously. This data can then be assembled into a 2D representation of the DUT VCORE current consumption as a function of the laser beam position. This data constitutes the TLS response map of the device.

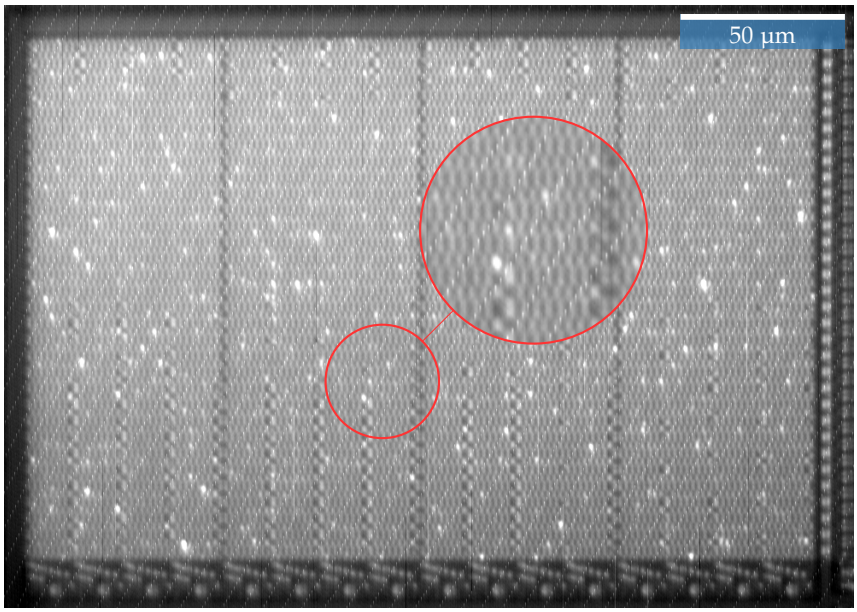


Figure 3.6: TLS response map of the full 1 KB SRAM of the MSP430. The memory has been initialized to `0xFF` before the measurement. Irregularities are caused by the data of the initialization program's variables. Reprinted with permission of ASM International. All rights reserved. [34]

The described setup can be used for different analysis scenarios. For example, if no code or memory initialization is executed, the initial startup values of the SRAM can be analyzed. This can be relevant for certain types of security concepts such as physically unclonable functions (PUFs) [22, 54], which can use the SRAM startup values to generate a "silicon fingerprint" of the device, see e.g. [29]. Alternatively, data that is present during program execution, such as keys or other sensitive data, can in principle be extracted. For this, the device clock needs to be halted, or its VCC voltage lowered under the operating threshold (brownout) after or during program execution. A brownout approach has the added benefit for an attacker that the device will be unable to perform defensive actions, such as memory zeroization. If a suitable brownout voltage is chosen, the CPU will not be able to operate. At the same time, the SRAM will still hold the data, as the SRAM fail threshold is usually lower than the lowest operational voltage. Thus, it can be seen that in connection with these scenarios the setup should allow for flexible attack evaluations.

3.2.3 *Response Map Data Dependency*

This section will showcase some results acquired with the setup described in Sect. 3.2.2 to illustrate the data dependence of the TLS response maps.

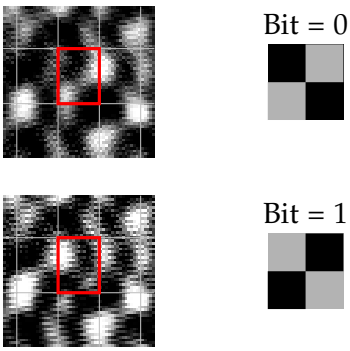


Figure 3.7: TLS response of a single SRAM cell at different bit values (assigned arbitrarily). The cell size is approx. 1.9 µm x 2.5 µm. Reprinted with permission of ASM International. All rights reserved. [34]

Fig. 3.6 shows TLS data of the full 1 KB of SRAM memory of the MSP430. In this case, SRAM initialization was performed prior to data acquisition. For this, the device was booted with the auxiliary power supply set to 2.6 V. After power-on, code was run from flash which sets all SRAM memory that is not needed for program execution to `0xFF`. The MSP430 was then sent to low power mode (LPM4) to reduce noise on the supply rail. In this mode, many parts of the device are deactivated, however, the memory remains powered [82]. For acquiring the TLS response, the current amplifier connected to VCORE was set to 2.1 V, slightly above the set point of the internal VCORE regulator (1.9 V). The amplification was set to 1 nA/V, the laser power to 100%, and the scan time to 240 s.

The TLS response of Fig. 3.6 closely resembles the reflected light images of the area, compare Fig. 3.5b. Because of the initialization, most of the response map shows a regular structure. The irregular vertical stripes that are present are most likely caused by the data of the SRAM fill program itself, for whose execution a small amount of SRAM needs to be reserved. That the uninitialized cells are clearly visible already demonstrates the data dependency of the TLS signal. The stimulation response also reveals that the memory seems to be divided into four sub-blocks, with a small gap in between. Counting of the regular structures reveals an amount of 64 by 32 in every sub-block, which, if interpreted as single bits, gives exactly 1024 bytes in total. Additionally, it is evident that some cells seem to be more sensitive to TLS stimulation (bright spots), which can be explained by manufacturing variability.

To illustrate the data dependency of a single bit, Fig. 3.7 presents TLS data of a single cell with enhanced contrast for different bit values. The surrounding cells contain arbitrary values. A grid corresponding to the cell boundaries has been overlaid onto the data to aid in orientation. It can be seen that the cell's TLS response follows the expected simplified TLS map discussed in Sect. 3.2.1, compare

also Fig. 3.3. The measurements show the top TLS spot of the cell to be larger than the lower one, which can be explained by different size PMOS and NMOS transistors in the cell. The TLS data clearly shows that the highlighted cell is the only one changing its bit state between the first and the second measurement. This proves the ability to deduce the bit state from TLS data.

These first measurements with the setup demonstrate that full memory extraction seems feasible. However, manual cell-by-cell extraction is a tedious task and therefore cell state recognition would need to be automated. Furthermore, the analysis showed that the physical placement of the SRAM cells does not follow its logical organization. For extraction of data, this logical-to-spatial mapping (so-called "scrambling") of the memory would need to be known. An assessment of the feasibility of these two tasks will thus be carried out in the following sections.

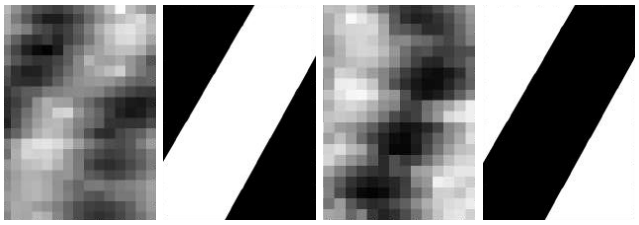
3.3 AUTOMATED MEMORY RECOVERY FROM STIMULATION DATA

This section will determine if automated extraction of memory contents is indeed possible. The basis for this will be the setup presented in Sect. 3.2.2 and measurement data produced with it. As mentioned in the previous section, there are two separate challenges present for an attacker seeking to extract memory contents: automated recognition of the cell state and reverse engineering of the mapping of physical cell positions to logical bits. To assess if these challenges can be overcome, these were handed out as topics for two bachelor's theses [71, 85]. As bachelor's theses are produced with a limited amount of time (three months nominally) by relatively inexperienced engineers, the feasibility of said methods can be assumed if the theses can overcome the mentioned challenges.

3.3.1 *Detection of the Memory Cell State*

In the first bachelor's thesis [85] different recognition methods were evaluated against TLS data to assess feasibility and error rates. For this purpose, a GUI-based MATLAB software was developed. For detecting the bit state, the TLS data was first split into individual cell data. Approaches to automatically detect the cell positions failed and instead manual specification of the edges of the memory area was necessary. To increase the accuracy of the cell extraction, distortions caused by thermal sample drift were compensated by the software. Example data of two extracted cells is presented in Fig. 3.8a and 3.8c.

Different filters and threshold algorithms were then applied to the individual cells to extract dark and bright regions. From this, a refined general TLS pattern could be developed, see Fig. 3.8b and 3.8d. This knowledge was then used to develop three different cell state



(a) Cell A (b) Pattern A (c) Cell B (d) Pattern B

Figure 3.8: TLS responses of two cells with different bit values and proposed analysis pattern. [85]

detection algorithms based on thresholding. The first two algorithms analyze the state of each of the three regions apparent in the patterns of Fig. 3.8b and 3.8d. The first method assigns states to each region (bright or dark) and is, therefore, also able to detect erroneous recognition. Taking cell A (Fig. 3.8a) and pattern A (3.8b) as an example, a cell where the regions are detected black-white-white instead of black-white-black can be marked as "not recognized". This method thus also delivers information about the certainty of cell state detection. The second method makes a decision in any case by only taking the central strip into account and determining if it is dark or bright. Lastly, the third method simply determines the largest area within a cell automatically and detects if it is dark or bright. This last method has the added benefit that the boundaries of the pattern regions (compare Fig. 3.8b and 3.8d) do not need to be known beforehand as with the first two methods.

All three methods were evaluated against TLS data acquired on a 24x32 bit area of the MSP430 SRAM as described in Sect. 3.2.2 and 3.2.3, and achieved detection rates between 87.6% (third method) and 100% (second method). The thesis thus proved that automated bit state detection is indeed feasible.

3.3.2 *Reverse Engineering of the Physical Layout*

The second bachelor's thesis [71] focused on determining the physical-position-to-logical-address mapping of the memory of the MSP430. This so-called "scrambling" was examined by writing known data to the device and then taking TLS measurements to identify where in the DUT changes occur when certain addresses or bits are changed. As already demonstrated in Sect. 3.2.3, changes in data content manifest as specific patterns in the TLS response, compare Fig. 3.6. Using these patterns, it was possible to deduce the physical-to-logical mapping of the device, which can be summed up as follows: The SRAM address space ranges from `0x1C00` to `0x1FFF`. Although a single byte can be addressed, the device basically operates on 16-bit words placed at every even address. The physical locations of the

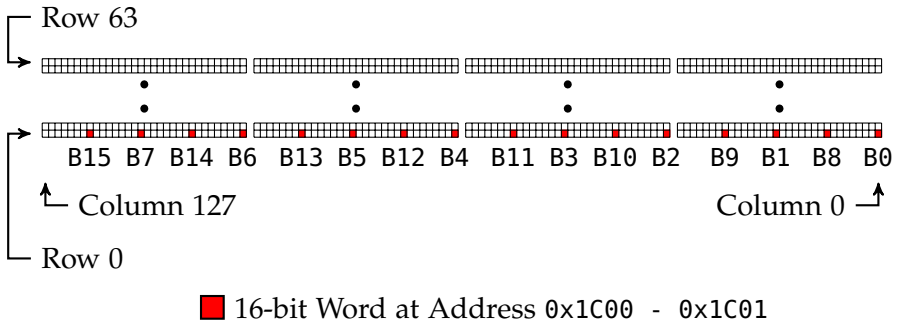


Figure 3.9: Bitwise scrambling of a 16-bit word. The bits of the two contained bytes are simply interleaved with each other and placed at every eighth column.

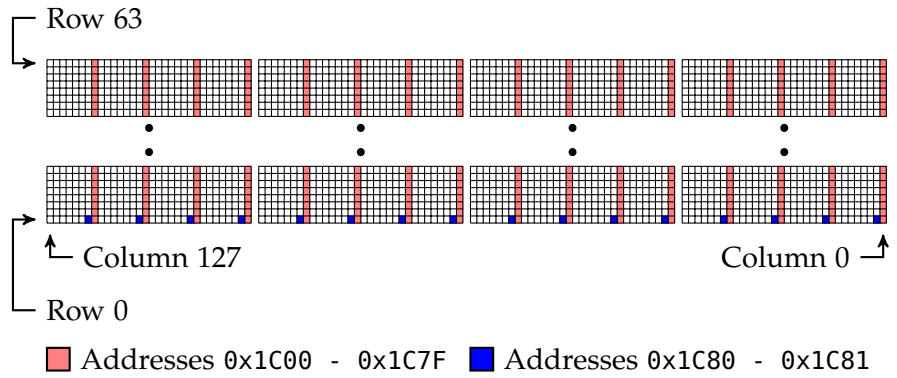
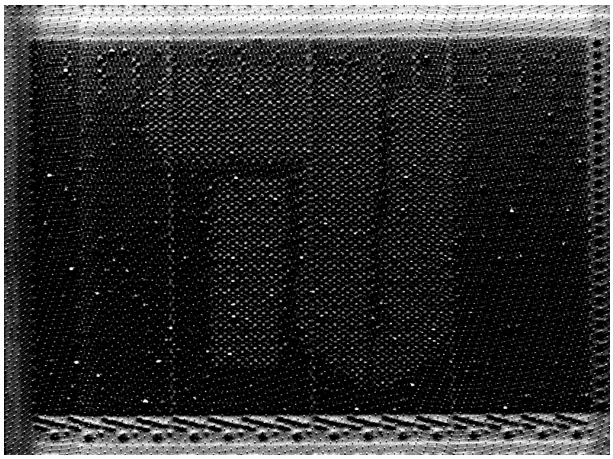


Figure 3.10: Address-wise scrambling of data words (16 bits per word). The SRAM is filled bottom to top from row 0 to row 63. After row 63 the filling wraps around to row 0 again while shifting one column to the left. This holds true for the whole memory except for an anomaly at addresses `0x1D80-0x1E7F`.

device are organized as 128 columns by 64 rows. Fig. 3.9 illustrates the scrambling of a single 16-bit word written at `0x1C00`. The orientation of Fig. 3.9 matches the one of Fig. 3.6. It can be seen that the individual bits of the word are written into every eighth column and that additionally the two bytes are interleaved with each other. Also apparent is that the memory is filled starting at row 0. This distribution of bits put in every eighth column in this specific pattern was determined to be consistent along the whole memory range.

Regarding the address-wise scrambling, it was determined that the memory is filled from row 0 to row 63, at which point it will wrap around to row 0 again and shift one column to the left. Fig. 3.10 illustrates this. This was also shown to be consistent along the whole memory range except for an anomaly in the range `0x1D80` to `0x1E7F`. In this area, the first word at the bottom of the memory is shifted *two* columns to the left when the wrap-around occurs, effectively skipping one column for every bit. When the top of the memory is

(a) Targeted physical bit distribution.



(b) TLS measurements of the data present in the MSP430. The data has been post-processed to enhance visibility.

Figure 3.11: Proof of validity of the descrambling algorithm. The target bit distribution is "reverse-scrambled" and then programmed into the MSP430 SRAM. The logical-to-physical mapping of the SRAM then leads to the recreation of the original pattern. [71]

reached, the previously skipped columns are then filled, i.e. the pattern is shifted one column to the *right*. Upon the next wrap-around, starting at `0x1E80`, the filling of the memory continues as it would have normally, i.e. at the columns that would have been expected without the anomaly. In other words, the anomaly behaves as if every fourth and fifth physical column of the memory were switched with each other. A detailed description of the anomaly can be found in [71]. Additionally, it should be mentioned that the cell orientation is mirrored for every row.

To allow for automated scrambling and descrambling, a MATLAB script was developed during the bachelor's thesis. This script was then used to verify the determined descrambling rules. For this, a target *physical* bit distribution was created in the form of the TU Berlin logo, see Fig. 3.11a. The script was then used to convert this data into a "reverse-scrambled" data file which could be programmed into the MSP430 SRAM. Only if the scrambling rules are indeed correct, the scrambling in the MSP430 should place the bits at the targeted physical positions, making the logo visible in the TLS measurements.

Fig. 3.11b reveals that this is indeed the case and the scrambling rules seem to have been determined correctly. Although some anomalies can be seen at the top of Fig. 3.11b, these could later be determined to be actual extra data placed in SRAM by the debugging tools used to program the data file.

The feasibility of quick and precise reverse engineering of physical-to-logical SRAM mapping using TLS is thus proven.

3.3.3 *Evaluation of Full Memory Recovery*

To evaluate the error rates when attempting full memory extraction, the code of Sect. 3.3.1 and 3.3.2 was combined. The initial code merge of the two existing MATLAB codes [71, 85] was conducted at Technische Universität Darmstadt at the Security Engineering group during a cooperation project. The resulting program was then handed back to Technische Universität Berlin, where it was adapted to work on the MSP430 memory layout and improved. While revising the code, an additional cell state detection method was added, which proved to be more robust than the ones discussed in Sect. 3.3.1. This method simply splits the cell into four quadrants and calculates the sum of TLS values in each of them to arrive at the total TLS response of each quadrant. The TLS responses of two diagonally opposing quadrants are then furthermore summed, arriving at two diagonal TLS sums. These sums are then subtracted from each other, with a positive difference indicating a "0" cell state and a negative difference indicating a "1" state. Additionally, if this difference does not exceed a certain threshold, the cell can be marked as "unsure/not detected".

This code was then used to evaluate full memory extraction. For this, the complete SRAM memory of the device was filled with known data and the device was sent to low power mode. Afterward, TLS measurements were conducted, see Fig. 3.6 as an example. The data was acquired using a 50x/0.71NA lens with 100% laser power (approx. 13 mW on the DUT), 72 s scan time, and 2 nA/V amplification. The stimulation response was acquired in four separate measurements using a 2x scanner zoom in combination with image stitching [60] to increase the effective pixel resolution. The data was then fed into the memory extraction software and the resulting binary data was output. Fig. 3.12 shows an excerpt of a hex dump of the data extracted via TLS for an experiment where the SRAM was filled with the string "Hello World! ". It can be seen in the ASCII representation on the right that the data can be recognized, although there are some incorrect bits, as evident in the hexadecimal representation.

To analyze TLS memory extraction more thoroughly, this experiment was repeated with different data sets and the binaries extracted via TLS were compared with binary images extracted directly from SRAM via the JTAG debugging interface. Tab. 3.2 shows the used fill

```
20 57 6F 72 6C 64 21 20 48 65 6C 64 6F 20 57 6F          World! Heldo Wo
72 6C 64 21 20 48 65 6C 6C 6F 20 57 6F 72 6C 64         rld! Hello World
21 20 48 65 68 6C 6F 20 5F 6F 72 6C 64 21 20 48         ! Hehlo _orld! H
65 6C 6C 6F 20 57 6F 72 6C 74 21 20 40 25 6C 6C         ello Worlt! @%ll
6F 20 57 6F 52 6C 64 21 20 48 65 6C 6C 6D 20 57         o WoRld! Hellm W
6F 72 6C 64 21 20 48 65 6C 6C 6F 20 5F 6F 72 6C         orld! Hello _orl
64 21 20 48 65 68 6C 6F 20 57 6F 72 6D 64 21 20         d! Hehlo Wormd!
48 65 6C 6C 6D 20 57 6F 72 6C 64 21 20 48 65 6C         Hellm World! Hel
68 EF 60 57 6F 72 6C 64 21 30 48 65 6C 6C 6F 20         h.‘World!0Hello
57 6F 72 6C 64 21 20 48 65 6C 6C 6F 20 57 6F 72         World! Hello Wor
6D 64 21 20 48 65 6C 6C 6F 20 57 6F 72 6C 64 21         md! Hello World!
20 48 65 6C 6C 6F 20 57 6F 72 64 64 21 20 48 6D          Hello Wordd! Hm
6C 6C 6F 20 57 6F 72 6C 24 21 20 48 65 6C 6C 6F         llo Worl$! Hello
20 57 6F 72 6C 64 21 20 48 65 6C 6C 6F 20 57 EF          World! Hello W.
72 EC 64 21 20 48 65 6C 6C 6F 20 53 6F 62 6C 64         r.d! Hello Sobld
21 20 48 65 6C 6C 6F 20 57 6F 72 6C 24 21 20 48         ! Hello Worl$! H
65 64 2C 6F 20 57 6F 72 6C 64 21 20 48 65 6C 6C         ed,o World! Hell
6F 20 57 6F 72 6C 64 21 20 48 65 6C 6C 6F 20 57         o World! Hello W
6F 7A 6C 64 21 20 48 65 6C 6C 6F 20 57 6F 72 6C         ozld! Hello Worl
64 21 20 48 65 6C 6C 6F 20 55 6F 72 6C 64 21 20         d! Hello Uorld!
48 65 6C 6C 6F 20 57 6F 72 7C 64 21 20 48 65 6C         Hello Wor|d! Hel
6D 6F 20 57 6F 72 6C 64 21 20 48 65 6C 4C 6F 20         mo World! HelLo
57 6F 72 6C 64 21 20 48 65 6C 6C 6F 20 57 4F 72         World! Hello WOr
6C 64 21 20 48 65 6C 6C 6F 30 57 6F 76 6E 64 21         ld! Hello0Wovnd!
20 48 65 6C 6C 6F 20 57 6D 76 6C 64 21 20 48 65          Hello Wmvld! He
6C 6C 6F 20 57 6F 72 6C 64 21 20 48 65 6C 6C 6F         llo World! Hello
20 57 6F 72 6C 64 21 20 48 65 6C 6C 6E 60 57 6F          World! Helln‘Wo
72 6C 64 21 20 48 65 6C 6C 6F 20 57 6F 76 6C 64         rld! Hello Wovld
```

Figure 3.12: Excerpt from a hex dump (hexadecimal and ASCII representation) of data extracted by thermal laser stimulation. The SRAM of the MSP430 was previously filled with the string “Hello World! “. The data, as well as bit errors, can be seen.

Fill Data	Number of Errors	Bit Error Rate [%]
`0x00`	57	0.70
`0xFF`	35	0.43
`0xAA`	76	0.93
`0x55`	32	0.39
“Hello World! “	123	1.50
random	452	5.52

Table 3.2: Bit error rates achieved using the memory extraction tool with different data written to the 1 KB (8192 bits) SRAM of the MSP430.

data, the number of errors, as well as the resulting bit error rate for these experiments.

It can be seen that between 99.6% and 94.5% of all bits are detected correctly. The `0x00`, `0xFF`, `0xAA`, and `0x55` experiments indicate that there is no outstanding difference in detection rate between "0" and "1" cells. Random data, on the other hand, seems to lead to higher error rates. This is probably caused by the increased influence of neighboring cells with different values on the currently analyzed cell. As a result, an optimization of the "four quadrants" detection method to disregard the cell border area might decrease this error rate in the future. Nevertheless, even the current worst-case detection rate of 94.5% makes data extraction feasible. Furthermore, it should be noted that if data of interest is found using this approach, the remaining bits can still be analyzed visually by the attacker.

3.3.4 *Summary of Results*

The previous sections have shown that a complete readout of SRAM memory is indeed feasible. The presented approach allows attackers to extract secrets such as keys and passwords from memory or, if they are in control of the clock, to analyze write access to SRAM for reverse engineering of algorithms. In general, it can be said that the effort required for TLS full memory extraction depends on the memory size as well as the detectability of the TLS signal. The detectability itself will be influenced by parameters like technology size, noise present on the power rail and similar factors. The results presented have shown that full extraction of SRAM memory contents is not only feasible but also that the necessary preliminary development can be carried out by bachelor students in a limited amount of time. This proves that TLS memory readout can be a powerful tool in the hands of an attacker. In the future, improvement of detection algorithms might increase the detection rates of between 99.6% and 94.5% achieved in the first attempt. The developed extraction tool can furthermore be used to evaluate attacks on real-world implementations in future work.

3.4 DECRYPTION KEY EXTRACTION FROM BBRAM

The experiments conducted so far in Sect. 3.3 were carried out on a general purpose microcontroller and not on a security-dedicated application-specific integrated circuit (ASIC). Furthermore, it is unknown how the TLS memory extraction technique will fare on devices with a technology size smaller than 180 nm. It is thus desirable to also evaluate TLS memory extraction against devices with dedicated security ASICs and a smaller technology size. For such a case study, a Xilinx Kintex UltraScale field-programmable gate array (FPGA) was

chosen. This device is manufactured in 20 nm technology and contains an advanced encryption standard (AES) decryption core ASIC with a 256-bit decryption key. The details of the necessity for decryption cores in FPGAs and their security features will not be detailed in this chapter, as only the feasibility of extraction of the AES key from memory is of interest here. However, Chapter 4 will later give a detailed discussion of FPGA security and evaluate laser-based attacks on them in more detail. In the case presented here, the device is simply used as a real-world target containing an AES key in memory.

One option for storage of the decryption key in this device is the so-called battery-backed RAM (BBRAM). This is simply a very low-power SRAM which is connected to a battery to keep the key in memory when the main power supply is removed. This constitutes an ideal scenario for an attacker using TLS for key extraction. As the BBRAM is the only structure connected to the battery power supply, a very low amount of noise on the supply rail can be expected. A further advantage is the ability to perform the key extraction with the board completely powered down, except for the battery supply. As no other components will be active, the noise will be further lowered and the FPGA will additionally be unable to take defensive actions. Apart from these aspects, the low number of bits makes it potentially feasible to extract the key manually, without the need to develop image recognition as in Sect. 3.3.1. However, the small cell size expected for a 20 nm device will be a challenge.

For the experiments, a development board (Digilent model AES-KU040-DB-G) was chosen which utilizes an XCKU040-1FBVA676 device in a bare-die flip-chip package. Because of this, no DUT preparation or thinning is necessary as the silicon is already exposed directly. Fig. 3.13 shows a die overview image assembled from multiple reflected light images acquired by the Phemos at 1.3 µm wavelength using a 5x/0.14NA lens. To generate the image, stitching software was used [60].

The approach to key extraction is rather simple: first, the BBRAM area needs to be located, then analyzed for data dependency of the TLS response, and finally examined for scrambling of the key bits. For this, the same principles and setups described in Sect. 3.2 can be employed.

To locate the key memory, the battery supply is connected to the current amplifier whose bias voltage is set to the nominal backup battery voltage (1.5 V). As TLS will influence the current through the BBRAM power supply terminal when stimulating the BBRAM area, this change can be detected by the current amplifier and located in the stimulation response map. The area for stimulation can be limited to the configuration and decryption logic area whose position can be deduced from information given in datasheets and similar documentation, see Fig. 3.13. Multiple structures were responsive

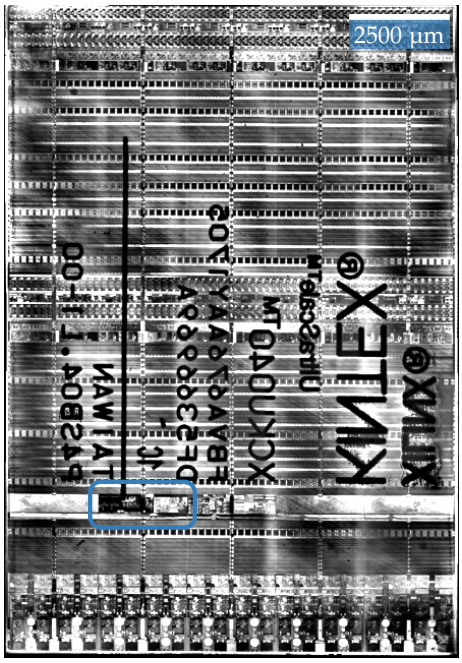


Figure 3.13: Overview reflected light image of the Xilinx UltraScale XCKU040 die. The area containing the configuration and decryption logic is highlighted. [42]

to TLS stimulation when the configuration area was examined using the 5x lens, see Fig. 3.14. These were analyzed in more detail with the 50x/0.71NA lens, which revealed the structures seen in Fig. 3.15a. Of these structures, only one showed dependency on deactivation of BBRAM key storage, compare Fig. 3.15b. This area was thus assumed to contain the BBRAM and examined further. The other structure was disregarded and assumed to be an electrostatic discharge protection structure for the battery power supply input. Fig. 3.16 shows a detailed reflected light image of the suspected BBRAM area. Its structure is similar to the one to be expected for standard RAM structures with word line and bit line access structures in the middle and bottom and cells seemingly distributed in two blocks.

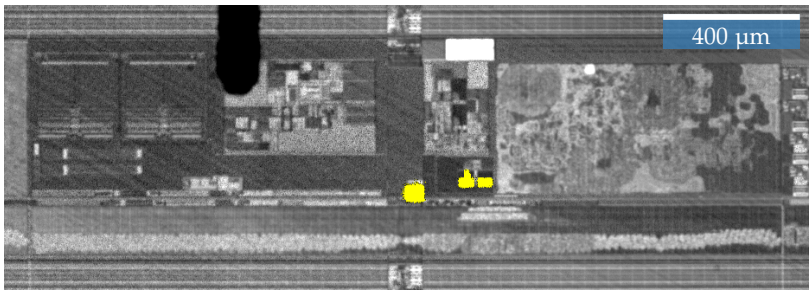
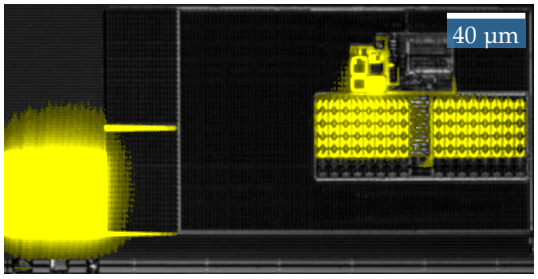
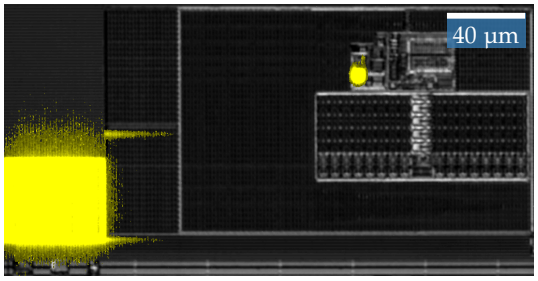


Figure 3.14: TLS measurements for BBRAM key memory localization using the 5x lens. The thresholded TLS data (yellow) is overlayed onto a reflected light image for orientation. [42]



(a) BBRAM key storage active.



(b) BBRAM key storage inactive.

Figure 3.15: TLS measurements for BBRAM key memory localization using the 50x lens. The thresholded TLS data (yellow) is overlayed onto a reflected light image for orientation. [42]

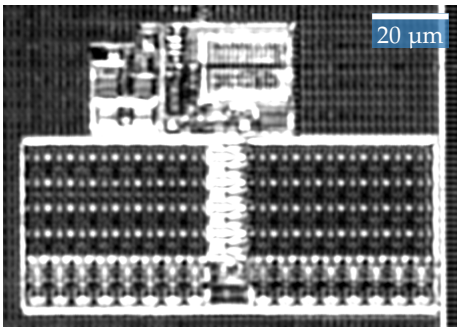


Figure 3.16: Reflected light image of the BBRAM AES key storage. [42]

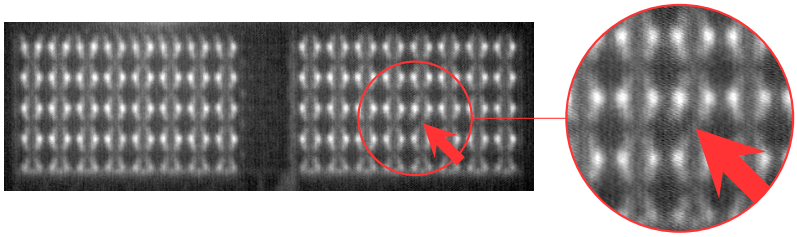


Figure 3.17: Data dependency of the TLS response. A single bit (bit 120) has been set in the BBRAM key data which manifests as an irregularity in the measurement result. [42]

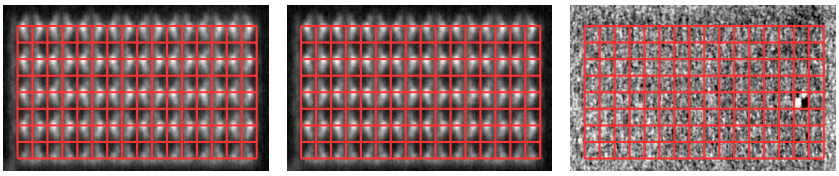


Figure 3.18: Difference calculation between an "all bits zero" TLS reference and measurement data quickly reveals which bits are set in the AES key. As an example, the right-hand half of the BBRAM with a single bit (bit 126) set is shown here. [42]

TLS measurements in these two blocks confirmed this theory and also revealed a data dependency of the TLS response map. Fig. 3.17 shows a measurement performed in this area with a single bit set in the key data. It can be seen that the set bit manifests as an irregularity in the TLS data. This change is even more evident if a reference measurement with an "all zeroes" key is subtracted from the TLS data for the current key. Fig. 3.18 shows an example of this procedure performed on only the right-hand half of the BBRAM. The single bit that is set can clearly be identified in the difference image.

Analysis of the memory area for scrambling revealed that there seems to be no scrambling present. The bits of the key are simply mapped from left to right and bottom to top, see Fig. 3.19. It also became apparent that there is an additional row of memory (32 bits) at the top of the BBRAM, compare also Fig. 3.18. The experiments determined that these 32 bits are used as additional memory for security-relevant features, such as the so-called "configuration counting" [57] differential power analysis (DPA) countermeasure feature of the device, see Fig. 3.19. This mapping already allows for manual extraction of the key and control data. Additionally, automatic extraction of the key using image recognition has been tested [42]. For this, five random keys, an all-zero, and an all-one key were used. All were recovered without error.

The experiments demonstrated in this section took a total of about 6-7 hours on the Phemos for locating and reverse engineering of the key memory. Using this knowledge and the automatic bit recognition,

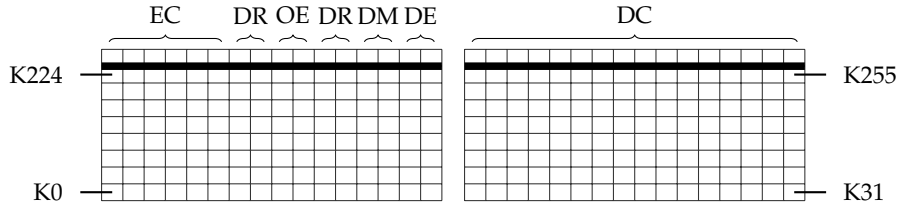


Figure 3.19: Mapping of the individual AES key bits (K) in the BBRAM. The bits are mapped from left to right and bottom to top. Additionally, other security-relevant data is saved at the top row: EC = error check, DR = DPA countermeasure reserved bits, OE = key obfuscation enable, DM = DPA countermeasure mode, DE = DPA countermeasure enable, DC = DPA countermeasure counter (8 bit, redundant). [42]

a key can now be extracted from the device in less than 15 minutes. As the rate for the equipment used in this work is about $300/hour, including the operator, the total cost of attack development would be $2100, with a single key extraction at $75. The vulnerability was reported to Xilinx and confirmed. It should be noted that the memory cell size (approx. 2.8 µm x 3.1 µm) was much bigger than expected for a 20 nm device, which is probably due to reliability and leakage/low current consumption considerations. As the key is vital for device operation, designers probably opt for a larger but more reliable memory cell which also delivers a longer battery life. Because of this aspect, it can be expected that BBRAM key storage solutions on other devices and from different vendors exhibit the same vulnerabilities. In total, these experiments showed that TLS is indeed a powerful attack technique for use in key extraction from real-world devices even on recent technology nodes and can do so in a very short amount of time.

3.5 CHAPTER CONCLUSION

This chapter has first given an overview of laser stimulation (LS) techniques common in failure analysis (FA). Following that, previous works which have shown the general feasibility of data extraction via photoelectric as well as thermal LS were briefly discussed. Among these, the authors of [9, 49] demonstrated the potential of secret data recovery as well as full memory extraction from the backside. As no actual data extraction has so far been conducted using thermal LS (TLS) from the backside, an assessment of its feasibility was pursued. For this, an overview of the findings regarding the principles of TLS SRAM data extraction by [9, 49] was first given. Based on these principles, a suitable setup for TLS SRAM data extraction was constructed and presented. Using this setup, TLS measurements were performed on a 180 nm technology Texas Instruments MSP430 microcontroller.

These measurements were used to analyze and illustrate the behavior of single SRAM cells under thermal stimulation and proved the ability to extract the bit state by TLS. Development of an automatic cell state detection via image recognition as well as the reverse engineering of the mapping of the physical cell positions to logical bits was then carried out in two bachelor's theses. The results from these theses were then used, partly in a cooperation with Technische Universität Darmstadt, to create a MATLAB software for full memory extraction. The software was then evaluated on the full 1 KB of SRAM memory of the MSP430 and achieved error rates between 0.4% and 5.5%.

To evaluate how well the technique would perform on an application-specific integrated circuit (ASIC) decryption core manufactured in a recent 20 nm technology, it was applied to the key memory of an AES-256 circuit used in a Xilinx Kintex UltraScale field-programmable gate array (FPGA). Surprisingly, the development of full extraction took merely around 7 hours, with a single key extraction taking about 15 minutes. The battery-backed SRAM cells (BBRAM) used in this device turned out to be much larger than the minimum cell size expected for a 20 nm device, probably due to reliability and leakage current concerns.

All in all, this chapter has demonstrated the potential of TLS for full memory extraction with low error rates as well as key recovery. Key recovery was even possible on very recent technology devices. Especially from the experiments performed on the FPGA, it was evident that it is more sensible to discuss actual memory cell size instead of technology size when evaluating the attack risk. With the current setup, TLS can be expected to work down to cell sizes of roughly 2 µm, 600 nm if the currently equipped resolution-enhancing solid immersion lens (SIL) is employed. With a more recent SIL, about 470 nm can be expected. It should be noted, that in the case of the FPGA, if the same ratio between technology size and BBRAM size as seen in the experiments is kept, BBRAM can be expected to remain vulnerable down to the 5 nm node with a recent SIL. However, it is unclear if the switch to FinFET and similar structures might bring changes in cell behavior under stimulation.

4

OPTICAL PROBING ATTACKS

Optical probing is used in failure analysis (FA) to identify areas carrying a specific signal (activity maps) and to analyze the signal present at individual transistors (waveform probing). This allows debugging silicon circuits in a similar fashion as one would with an oscilloscope at the component level. However, as optical probing gives access to internal signals of the device, it can also be used by attackers to extract sensitive data. When an attacker employs optical probing, she can use the generated activity maps to reverse-engineer the circuit and then extract data that is passing through the transistors using waveform probing. As the information is directly carried by the probing beam, there is close to no influence on the device, as opposed to fault injection, see Chapter 2, and laser stimulation, see Chapter 3. Probing attacks can therefore not be detected without dedicated sensor structures.

This chapter will assess to what extent these common FA techniques pose a threat to data which is present inside devices, in particular field-programmable gate arrays (FPGAs). For this, the decryption circuits implemented on the reconfigurable logic of FPGAs as well as application-specific integrated circuits (ASICs) present in these devices will be analyzed. Both extraction of generated keys and plaintext will be evaluated. It is again assumed that the attacker has access to a standard failure analysis microscope, in this case with an optical probing option. Some of the results and figures in this chapter were already published in [41, 81].

4.1 OPTICAL PROBING TECHNIQUES

In failure analysis (FA) several techniques for extracting electrical information from integrated circuits have been established. One class of techniques uses light to extract the relevant signals and is therefore referred to as "optical contactless probing". Most of these optical contactless probing techniques can be further divided into two subclasses. The first is the acquisition of electrical waveforms, while the second is the creation of 2D maps of active circuitry. Ready-to-use optical probing systems are supplied by a multitude of different manufacturers. Depending on the manufacturer, optical waveform probing is referred to as electro-optical probing (EOP), laser timing module (LTM), laser time probe (LTP) or laser voltage probing (LVP). Acquisition of 2D activity maps is similarly referred to as electro-optical frequency mapping (EOFM), signal mapping image (SMI) or laser

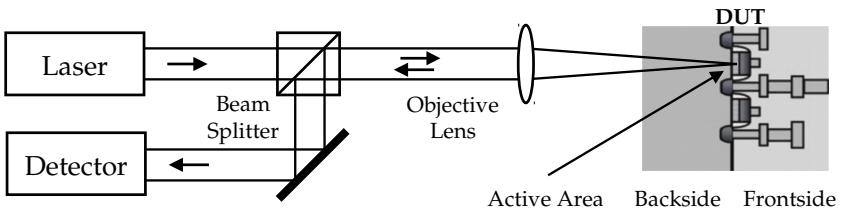


Figure 4.1: Simplified contactless optical probing setup. A laser illuminates the device under test (DUT), whose properties modulate the reflected light, which is then analyzed by a detector. Reprinted by permission from Springer Nature Customer Service Centre GmbH: Springer Nature, [41], © 2016.

voltage imaging (LVI). Although the names and implementation details differ from manufacturer to manufacturer, the underlying basic principles remain the same and will be briefly illustrated here. Fig. 4.1 shows a sketch of a simplified optical probing setup. Here, a parallel laser beam exits the light source, passes through a beam splitter and impinges onto an objective lens. The lens then focuses the beam into the active area of the device under test (DUT) through the device backside. Part of the light is then reflected off metal layers or other structures in the device and travels back through the lens and the beam splitter and is finally collected by a detector which evaluates its intensity. As the device backside consists of silicon, usually infrared light is employed for this scheme, as silicon becomes transparent at a wavelength of above 1.1 µm [24]. Inside the active DUT area, the waveform present at the electrical node of the device modulates the physical properties of the device. Some of these then modulate the refractive index n and absorption coefficient α_{abs}. For a detailed discussion of the causes for the n/α_{abs} modulation the interested reader is referred to [33] and the sources cited therein. These two parameters influence how much light is reflected back onto the detector. They do so either directly through modulating absorption or indirectly through (parasitic) interference effects caused by modulation of the optical path length. As a result, the amount of light reaching the detector is modulated by the electrical waveform present at the probed DUT area. Therefore, the electrical waveform can be recovered from the output signal of the light detector. However, since the relative change of light intensity is in the order of a hundred parts per million (ppm) [33], the signal is usually lost in the noise. A common approach to avoid this problem is to average multiple waveforms of the detector signal using a trigger signal while the device is run in a loop. These averaged waveforms will then deliver the electrical signal present at the probed device area. This technique will be referred to here as laser voltage probing (LVP).

An extension of this concept is the creation of 2D circuit activity maps. In this case, the detector signal is not averaged but directly

fed into a spectrum analyzer. The spectrum analyzer is then set to some frequency of interest and zero span, thus effectively acting as a very narrow frequency filter with a configurable bandwidth. The laser beam is then scanned across a rectangular area of interest on the device using galvanometric mirrors, while a computer simultaneously samples the output of the spectrum analyzer. If some areas of the device are switching at the set spectrum analyzer frequency, they will modulate the light with this frequency. As a consequence, the detector signal will be able to pass the filter, which will lead to an increased signal at the spectrum analyzer output. If a certain circuit area is inactive, or active at a different frequency, the spectrum analyzer output will be zero. After the whole area is analyzed in this fashion, the frequency filter output values can be assembled into a 2D representation by a PC. If the numerical data is then grayscale or color encoded, the resulting image will show active circuitry as bright pixels above a black background of non-activity. This technique will be referred to as laser voltage imaging (LVI) in this work.

It should be mentioned that for some circuit to show up on an LVI activity map, it is sufficient if some frequency *component* of its electrical signal is able to pass the filter. Therefore, circuits might also show up in this map even if their fundamental frequency is not at the filter frequency. For simple waveforms, like rectangular waves, this might happen due to higher harmonic components, while more complex signals might have an equally complex frequency spectrum containing such components. This also means that arbitrary waveforms can be examined using LVI, as long as their frequency components can be understood.

As LVP and LVI are specifically designed to extract information from integrated circuits, the question arises as to which extent they might constitute a threat to secure integrated circuits and the secrets contained within them. The following chapters will try to assess this threat in the context of field-programmable gate array (FPGA) devices. As opposed to custom integrated circuits in national ID documents, military hardware or other such secure circuits, commercial FPGAs are freely available. Nevertheless, FPGAs are used in highly critical applications such as cryptographic, aerospace, and military systems. Furthermore, the manufacturers of hardware which uses FPGAs have a strong interest to protect their intellectual property programmed into the FPGA, both for security and commercial reasons. Therefore, most modern FPGAs feature state-of-the-art encryption concepts and technology to secure the data loaded into the FPGA, which makes them an ideal device for the experiments in the following chapters.

4.2 GENERATED KEY EXTRACTION AND PUF CHARACTERIZATION

This section assesses the extraction of key data and characterization of physically unclonable functions (PUFs, see [22, 54] and Sect. 2.2.4) using optical probing techniques. In particular, probing of configuration data decryption keys from FPGAs is examined. Furthermore, characterization of a PUF used in generating these decryption keys is also evaluated, which then enables manual calculation of the key.

The structure of this section is as follows: the need for and concepts of FPGA configuration data encryption and decryption will be briefly explained in Sect. 4.2.1. Sect. 4.2.2 then sketches a proof-of-concept (POC) implementation of an advanced key generation which follows the concepts of the previous section. Afterward, Sect. 4.2.3 and 4.2.4 present the concepts of two optical probing attacks against said POC implementation. A suitable hardware setup for assessing these attacks is then developed in Sect. 4.2.5. Finally, Sect. 4.2.6 reveals the results of said attacks on the POC implementation.

4.2.1 *FPGA Bitstream Encryption Concepts*

FPGAs are devices which contain basic building blocks for logic circuits, such as logic gates, registers, SRAM blocks, multipliers, I/O buffers and so on. The FPGA allows for flexible setup and connection of these elements to form more advanced logic circuits. These can range in complexity from simple logic functions to complete microprocessors implemented on the FPGA gates. Some of these basic FPGA concepts have already been discussed in Sect. 2.2.1. The data that is used to configure the flexible parts of an FPGA is usually referred to as the "configuration data" or simply the "bitstream". For many current FPGAs, this configuration bitstream has to be loaded into the FPGA from external sources after power-up. The reason for this is that some FPGA vendors employ volatile memory cells in their configuration and routing circuits, whose states cannot be preserved during power-down. If the configuration bits which are transferred into the device are not protected, an attacker might be able to intercept these bits on the printed circuit board and therefore gain insight into the device configuration. This might then allow her to tamper with said configuration, reverse-engineer device operation from the configuration data, or simply steal intellectual property that is present in the bitstream. To combat this threat, FPGA vendors have introduced encryption of the device bitstream. There are different concepts for bitstream encryption, each with its own advantages and disadvantages. Some concepts which are relevant to this work will be explained in the following sections.

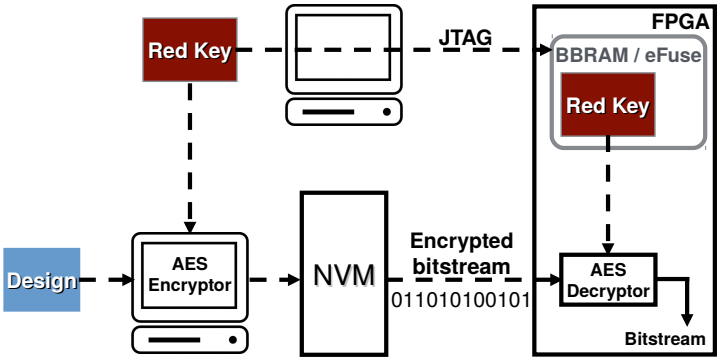


Figure 4.2: Simple bitstream encryption scheme using a secret so-called "red key" that is programmed into the FPGA device for decryption purposes. Reprinted by permission from Springer Nature Customer Service Centre GmbH: Springer Nature, [41], © 2016.

4.2.1.1 *Simple Bitstream Encryption*

A straightforward, simple bitstream encryption/decryption scheme is sketched in Fig. 4.2. In this case, a secret key is first generated, which is referred to here as the "red key". At the trusted manufacturing site, this red key is used to encrypt the design, which is then placed in the non-volatile memory (NVM) of the target system. In this example, the encryption is done using the "AES" (advanced encryption standard) algorithm. Furthermore, still at the trusted site, the red key is also transferred into the FPGA, see Fig. 4.2. In this example, a "JTAG" (joint test action group) debug and testing interface is used. Internally, the FPGA saves the red key either to a battery-backed RAM (BBRAM) or to electronically programmable fuses (eFuses). BBRAM can be reprogrammed, while eFuses are one-time programmable only. Both of these memories can never be read out through the FPGA interfaces. In the untrusted field, the FPGA will load the encrypted bitstream from the NVM and use the stored red key to decrypt it. The decrypted bitstream is then used to configure the FPGA.

This simple bitstream encryption has some drawbacks: if the red key is stored in eFuses, depending on the eFuse size, optical microscopy or scanning electron microscopy (SEM) can reveal the key bits, see Fig. 4.3. As the eFuse relies on physical material changes for bit storage, it will preserve its contents throughout device decapsulation and preparation for SEM. BBRAM, on the other hand, is harder to readout. To preserve the BBRAM memory content, a constant power supply needs to be maintained during the whole decapsulation and preparation process, which is needed for many conventional SRAM readout attacks. An exception is the flip-chip BBRAM stimulation attack demonstrated in Sect. 3.4. In this regard, BBRAM is more secure than eFuse key storage. However, since BBRAM also requires

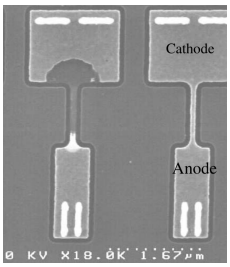


Figure 4.3: 90 nm technology eFuses in programmed (left) and unprogrammed state (right), as seen with a scanning electron microscope after decapsulation and thinning. The bit state of the individual eFuses can easily be determined. © 2008 IEEE. [83]

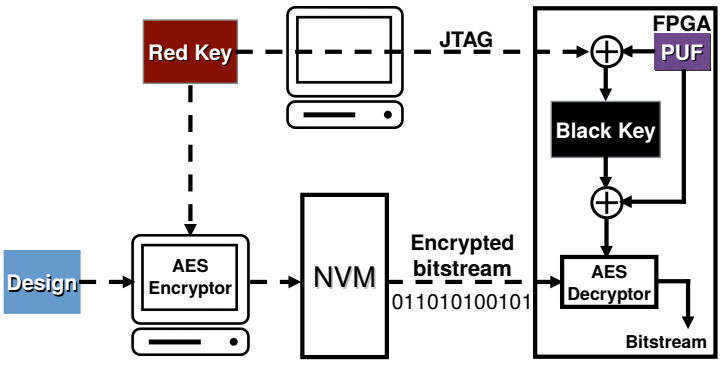


Figure 4.4: Advanced bitstream encryption scheme that makes use of a physically unclonable function (PUF) to wrap the secret "red key", thus never directly storing the red key inside the device. The red key is regenerated only momentarily during each power-on. Reprinted by permission from Springer Nature Customer Service Centre GmbH: Springer Nature, [41], © 2016.

a backup battery which lasts throughout the whole device lifetime, which can be tens of years, it might raise concerns about reliability.

4.2.1.2 *Bitstream Encryption Using PUFs*

Because of the drawbacks of the simple concepts described in the previous section, alternate approaches to key storage have been devised. One of these, which uses physically unclonable functions (PUFs), is presented in this section. Fig. 4.4 shows a simplified sketch of a concept employing a PUF for secure key storage as proposed by FPGA manufacturer Xilinx in [56]. In general, PUFs can be used for different tasks such as random number generation, authentication or key generation, see also Sect. 2.2.4. Since they exploit device manufacturing variability to generate their outputs, they should be hard to predict or clone. In the case of this concept, the PUF is used as a "silicon fingerprint", generating a number of bits which are unique to one specific device. Since the fingerprint is based on said manufacturing variability, simply copying the PUF implementation to a different

chip will yield a changed PUF response. This can now be used to achieve secure key storage.

As before, a secret "red key" is first generated and used to encrypt the design, which is then placed into NVM, see Fig. 4.4. While still at the trusted site, the PUF is run inside the FPGA, generating its device-specific response. This "device fingerprint" is then used to wrap the secret red key, for example by performing an XOR operation or an encryption, yielding a wrapped "black key". The red key can be regenerated from this black key only if the PUF response for that specific device is known. The black key can then also be placed into internal or external NVM. It should be stressed that this black key in itself is useless to an attacker, as she does not know the PUF response necessary to calculate the red key.

Now, when the device boots in the untrusted field, first the same PUF implementation is run inside the FPGA. Using the PUF's response and the black key, the FPGA is now able to calculate the secret red key and uses it to decrypt the bitstream. As soon as the decryption has finished, the red key, black key, and PUF implementation will be erased from memory and the device will be configured using the decrypted bitstream. The red key is therefore never permanently stored and only regenerated during bitstream decryption. It is furthermore only present as a volatile on-die-only signal. Additionally, the use of a PUF prohibits the use of the same encrypted bitstream on a different device (cloning), as the changed PUF response would result in an incorrectly calculated red key.

For implementing this concept there are two fundamental options. The first and straightforward one would be to implement the bitstream decryption as an application-specific integrated circuit (ASIC) block in the FPGA, which is also referred to as a "hard" decryption core. The second option is to use what is called a "soft" decryption core. In this case, the decryption is implemented on the configurable FPGA elements and is also often referred to as a "bootloader". Here, the bootloader code is first authenticated and loaded into the FPGA, which then loads and decrypts the bitstream and uses it to finally configure the device through a feature called "partial reconfiguration".

Although hard cores seem like the most straightforward way to implement such a bitstream decryption, their unalterability can become a major disadvantage. In the case of multiple FPGA families, simple or differential power analysis (SPA/DPA) and electromagnetic (EM) side-channel attacks have been demonstrated in the past [46, 47, 77]. These attacks make it possible to extract the encryption key by monitoring current/voltage fluctuations on the power supply line or EM waves radiated by the device. As the attacked bitstream decryption consists of hard cores, these attacks now pose a threat to each and every design using one of the affected devices, as the hard decryption core cannot be altered.

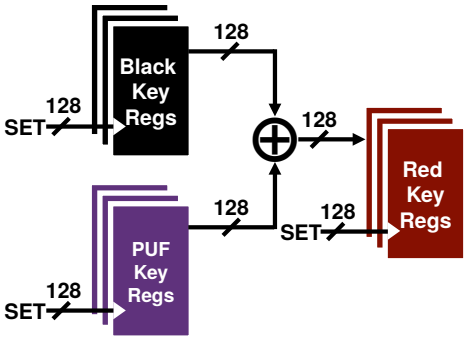


Figure 4.5: Sketch of a 128-bit parallel red key calculation. Reprinted by permission from Springer Nature Customer Service Centre GmbH: Springer Nature, [41], © 2016.

Soft decryption cores, on the other hand, can be replaced using the standard firmware update procedures, as soon as a vulnerability is discovered. Both soft and hard cores have in common that they assume that the temporarily generated red key, as well as the also security-critical PUF response, are safe, as they are temporary, volatile signals inside the silicon die.

4.2.2 *Proof-Of-Concept Key Generation Implementation*

For assessing the vulnerability of these concepts to optical probing, it is necessary to have access to either a hard or soft decryption core. As the experimental FPGA platform (Altera Cyclone IV) available at the time did not contain a hard bitstream decryption core, a soft-core implementation remained the only option for evaluation. However, as it was not possible to gain access to commercial PUF-based soft decryption cores for bitstreams, a proof-of-concept (POC) implementation was developed. For simplicity, this implementation covered only the most critical parts: the reconstruction of the red key and a basic PUF. The implementation followed the concepts of a soft-core bitstream decryption, as described by FPGA manufacturer Xilinx in [56]. The main aspects of this concept have already been explained in Sect. 4.2.1.2. For implementing the red key recreation, two approaches were implemented: a serial as well as a parallel calculation. For the PUF implementation, a ring oscillator (RO) PUF was selected. The following sections briefly explain these aspects of the POC implementation in more detail.

PARALLEL KEY CALCULATION In case of a parallel calculation, as sketched in Fig. 4.5, the 128-bit wide red key will be created by feeding the PUF response (also called the "PUF key") and the black key into a 128-bit wide exclusive or (XOR) function. Therefore, the key will be calculated in a single clock cycle as soon as the PUF and black

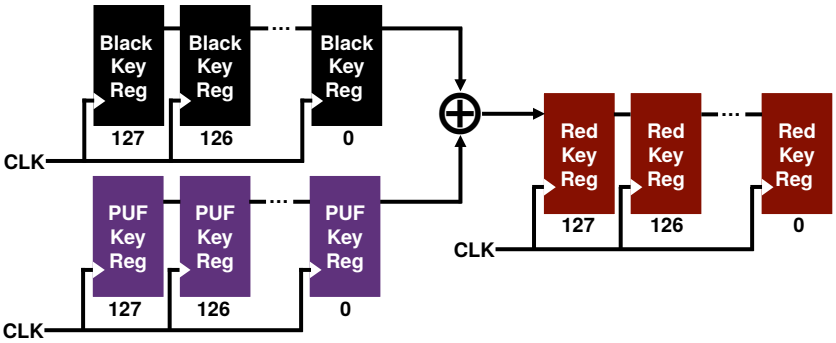


Figure 4.6: Sketch of a 128-bit serial red key calculation. Reprinted by permission from Springer Nature Customer Service Centre GmbH: Springer Nature, [41], © 2016.

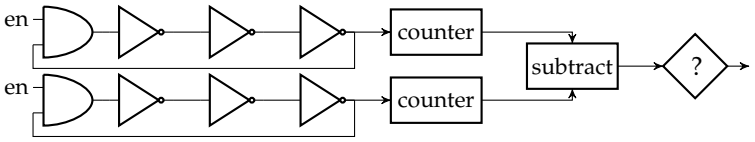


Figure 4.7: Simplified sketch of a ring oscillator pair as used in the proof of concept implementation. Reprinted by permission from Springer Nature Customer Service Centre GmbH: Springer Nature, [41], © 2016.

key registers are loaded and the "set" inputs of the red key registers are activated. The red key registers can then be used as inputs to the decryption core. For simplicity, an 8-bit key width was chosen for the POC implementation.

SERIAL KEY CALCULATION Fig. 4.6 shows a sketch of a 128-bit wide serial red key calculation. Here, the registers for the red, black, and PUF key are connected in series to form three shift registers. The outputs of the black and PUF key shift registers are fed into a single-bit exclusive or (XOR) gate whose output feeds the input of the red key shift register. To calculate the red key, the PUF and black key registers will first be loaded with their respective values. Following that, the shift clock is activated, allowing the individual bit values to get shifted into the XOR, while the resulting red key will get shifted into the red key registers. Therefore, the calculation of the key will take 128 clock cycles. As for the parallel calculation, an 8-bit wide key was chosen for the POC implementation, reducing the needed clock cycles to 8.

RING OSCILLATOR PUF To also evaluate possible vulnerabilities of the PUF itself, a simplified ring oscillator (RO) PUF was realized for the POC implementation, see Fig. 4.7. In this case, two ROs consisting of 11 inverters were programmed into the FPGA. The ROs are connected to two counters which count the RO oscillations for a cer-

tain number of clock cycles. A comparison circuit then uses these counters to determine which of the two ROs was faster and outputs a binary result accordingly. It should be noted that in case of a full RO PUF implementation this circuit would contain additional ROs and multiplexers which would allow for the comparison of multiple RO pairs. This would then deliver a PUF response with a bit width equal to the number of examined RO pairs. However, in the case of the POC implementation, the described simplified version should suffice, as it already contains all relevant elements of an RO PUF.

4.2.3 *Optical Key Extraction Concept*

As explained in Sect. 4.1, in failure analysis (FA) both laser voltage probing (LVP) and laser voltage imaging (LVI) are established techniques. LVP allows probing the waveform present at an electrical node, similar to an oscilloscope, while LVI allows the creation of an activity map of circuits switching with a certain frequency. To understand the standard FA workflow using LVI/LVP better, it might be helpful to consider a simple example: imagine a chain of inverters or buffers carrying some waveform. If this waveform fails to appear at the output, an FA engineer could feed a simple square wave signal with a known frequency into the chain while performing LVI on the device. All nodes of the chain carrying the waveform would then show up in the LVI map, while the first failing element and all following elements will not be detectable. Using layout and schematic the FA engineer would then be able to locate the fault in the vicinity of the last functioning and first failing chain element. If, on the other hand, a waveform appears at the chain output but is malformed, the FA engineer could first use LVI to locate the beam modulating locations of the circuit of interest, and then use LVP to probe the waveforms of the individual chain elements, to see at which point the degradation occurs.

However, for an optical probing attack, it cannot be assumed that the attacker has access to the transistor-level layout. She will furthermore not be able to achieve the same level of control over the input signals as in the FA example of the previous paragraph. Therefore, to assess the feasibility of optical probing attacks against the POC implementation, it is necessary to understand in which way an attacker would most likely approach this challenge. Without transistor-level layout access, the attacker's first action will presumably be to perform LVI in some way, to analyze the circuit's operation. Ideally, this LVI map will already be data dependent and reveal key bits or other secret information to her. If it does not, it must at least reveal to her the transistors carrying secret data, which she might then further analyze using laser voltage probing. The following sections will discuss

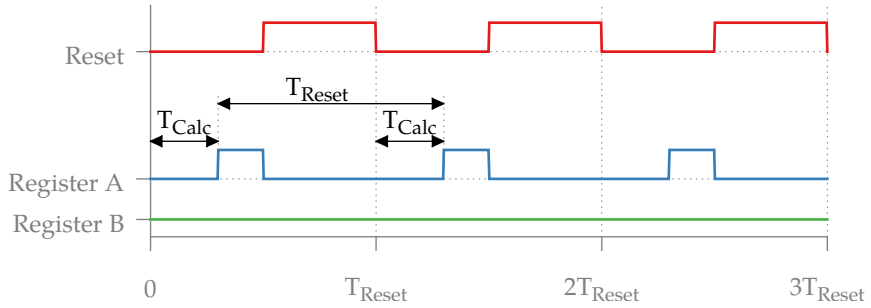


Figure 4.8: Waveforms of registers of the parallel implementation in conjunction with the reset signal. Register A receives a "1" bit during the calculation, while register B receives a "0" bit.

which approach an attacker might pursue, depending on the basic implementation type considered.

It should be noted, that it is assumed that the attacker is able to *approximately* determine the location of the key registers and PUF, which will allow her to narrow her search area. To gain this knowledge, she can analyze the first stage bootloader if it is unencrypted. If it is encrypted, she can try to extract the key, for example by differential power analysis (DPA) or stimulation attacks, see also Chapter 3. If all these approaches fail, she can manually analyze the circuits that are revealed to her by LVI/LVP measurements. Since during the very first stage of booting only the PUF key generation is active, the number of false positives during such an approach should be limited, although the analysis might be tedious.

4.2.3.1 *Parallel Implementation*

This section discusses the attack approach for the parallel implementation. As LVI will be the first step in an attack, and LVI detects frequencies or frequency components, it is necessary for the attacker to analyze the circuit in the frequency domain. An analysis of the parallel implementation introduced in Sect. 4.2.2 reveals that all registers taking part in the red key calculation receive data exactly once for every power-on. Therefore, if the attacker places the device in a reset loop, she can *induce* the reset loop frequency into the calculation circuit. Furthermore, there is a fundamental distinction that can be made for all the registers: each individual register receives either a one or a zero during the calculation. Therefore, the analysis can be simplified to the analysis of just these two types of registers. Fig. 4.8 shows a sketch of the waveforms on these two types of registers when the device is operated in the reset loop scenario.

When the circuit is first powered on, all registers are reset to their default values by the reset circuitry. These default values are assumed to be logic "low" level. As soon as the reset signal goes low, the

circuit will become active and start with the calculations preceding the setting of the register considered. For register "A", receiving a one, it can be seen that as soon as this amount of time, T_{Calc}, has passed, the register will transition from low to high. It will keep its high level until the reset signal is asserted, at which point it will change back to low. Register A will then stay low until the reset signal is de-asserted at which point the calculation cycle will repeat. It can be seen from Fig. 4.8 that this causes the waveform present at register A to repeat after the reset loop period T_{Reset}. This directly implicates that the fundamental frequency of that waveform will be the reset loop frequency. Therefore, if the attacker performs LVI at this frequency, this register should show up in the LVI activity map. For register "B", receiving a zero, the case is much simpler. As this register does not change its state at all, it should not be detectable by LVI. Therefore, if the output values of the LVI spectrum analyzer are color-encoded in the LVI activity map, the attacker can expect registers receiving a one to show up bright, while registers receiving a zero should stay black. Thus, she should be able to directly extract the key bits from the LVI activity map.

4.2.3.2 *Serial Implementation*

In the case of the serial POC implementation, see Sect. 4.2.2, the data cannot be extracted by a simple bright/dark LVI map distinction, as opposed to the parallel case. Because the key values are shifted in serially through the individual registers of the shift registers, the waveform present at each register depends on its position in the shift register. However, since the device is placed in a reset loop, the fundamental frequency of the register waveforms will still be the reset loop frequency. The registers should therefore still show up in an LVI activity map taken at the reset loop frequency. The attacker can then proceed to analyze the individual registers using laser voltage probing until she finds the first, or "shift-in" register, through which the whole key will be shifted. As soon as she has found this register she can extract the key data from the LVP waveform data.

As the LVI signal intensity is dependent on the strength of the frequency components at the boot loop frequency, and these only depend on the register waveforms, the attacker might even calculate the expected LVI intensity beforehand. In Fig. 4.9 the result of such a calculation can be seen. In this case, a simple script for the numerical computational software "Scilab" was written, which calculates the boot loop frequency component of the waveforms of all possible keys and for all registers. The script is documented in appendix A.3. The calculation is performed using a fast Fourier transformation (FFT) and takes about one minute in total on a 2.7 GHz quad core laptop. Register 7 is the shift-in register in this case, compare Sect. 4.2.2.

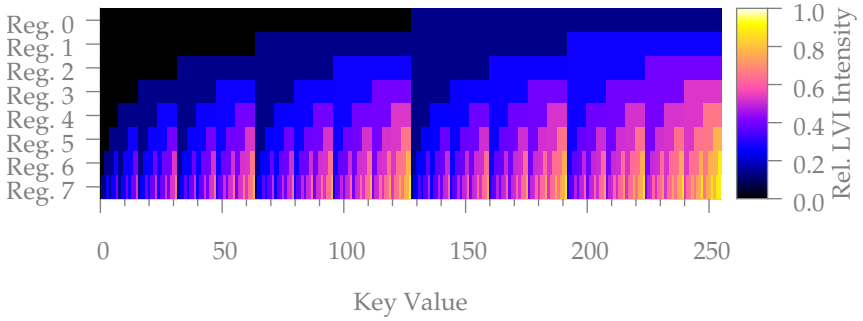


Figure 4.9: Theoretical LVI intensity as a function of register position and key value for a 1 MHz boot loop frequency. Calculated for a serial implementation using an 8-bit key with a key shift duration of 200 ns.

It is evident that every key value has a unique signal intensity distribution along the different registers. This means that even without probing the register waveforms, the attacker might be able to determine the key directly from this distribution in the LVI map. It should be mentioned that this calculation assumes that all registers would create an equally strong signal if they are pulsed with the same waveform. However, in a real-world scenario, this might not be the case, as varying remaining silicon thickness among the electrical nodes might lead to different amounts of signal strength due to interference effects. In this case, an attacker would need to perform an LVI reference measurement first, by pulsing all the nodes with a known waveform and registering their relative signal intensity. She could then use this later to correct her measurements for the signal intensity distribution and identify the correct key using the corrected data. Furthermore, if the system used for LVI does not have a high enough signal to noise ratio (SNR) to distinguish the required nuances in the intensity distribution (one nuance per key bit), it should still be possible to reduce the number of key candidates significantly and perform a brute-force attack using this knowledge.

It is evident that the time required for this calculation scales exponentially with the key length, and therefore calculations for long keys are infeasible using an FFT. However, the observation that registers receiving more "1" bits during the key shift will deliver a stronger LVI signal will remain true for any given key length and a fast simplified analytical calculation could be developed for longer keys. Thus, it is conceivable that key extraction from the serial implementation using *only LVI* is at least theoretically possible.

However, a more straightforward way is available to the attacker, if it is assumed that she is able to also perform LVP. It can be seen in Fig. 4.9 that if the attacker wants to find register 7 for some fixed key, her best guess is always to probe registers with the largest LVI signal first. Therefore, she can take an LVI measurement and systematically

probe the register waveforms, starting with the one with the largest LVI signal intensity.

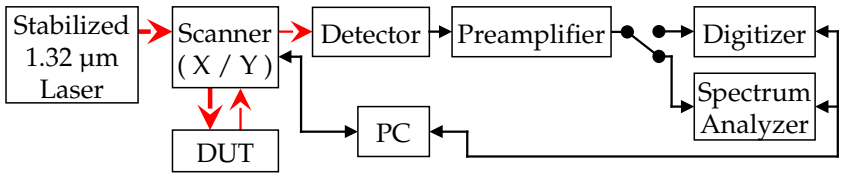
The overall result of the previous discussions is the following: if the attacker has LVP capabilities at her hands, the combination of an LVI measurement and the subsequent LVP probing of the registers with the largest LVI signal is the most simple and straightforward way to extract the secret serial data. If probing is not possible for some reason, she can move on to analyze only the LVI intensity distribution as discussed.

4.2.4 *Ring Oscillator PUF Characterization Concept*

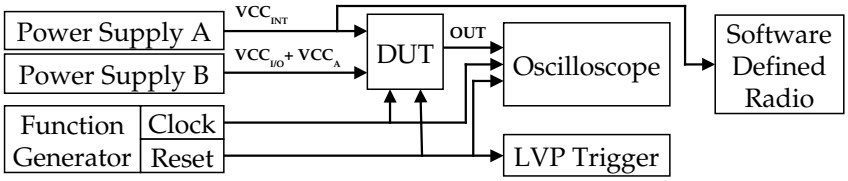
If the attacker is unable to extract the red key directly, there might still be a threat towards the bitstream encryption security. As the black key is present in NVM, it is conceivable that the attacker is able to acquire it. If she can now somehow characterize the used RO PUF, she will be able to derive the PUF response, or "PUF key", and will, in consequence, be able to calculate the red key manually. This section will discuss how she might achieve this RO PUF characterization through a combination of simple power analysis (SPA) and LVI/LVP. It should be mentioned that the same assumptions about knowledge of the *approximate* location of the PUF and its components already discussed in Sect. 4.2.3 apply here too.

As the attacker does not have knowledge of the approximate RO frequencies, she will not be able to directly perform LVI to find the location of the RO nodes. However, active ring oscillators on an FPGA will consume a fair amount of power. This is due to the fact that they constantly have to charge and discharge the parasitic capacitance of the gates and wiring with a high frequency. Furthermore, the current draw of the circuits which make up the RO is increased during switching when both PMOS and NMOS transistors are conducting for a short time. Thus, the ROs will modulate the power supply current with their own operating frequency. In combination with the non-zero power supply output impedance, this should lead to voltage fluctuations, which can then be analyzed using spectrum analysis. Therefore, if the attacker performs spectrum analysis on the power supply line while the ROs are active, this should deliver to her a superposition of all active RO frequencies and thus a first estimate of the RO frequencies. She will however not be able to characterize individual ROs this way.

Nevertheless, she can now use this frequency estimate to perform LVI with a broad filter bandwidth. In this way, she will be able to spatially locate all the ring oscillator nodes on the device under test. She can then move on to characterize each individual ring oscillator by probing its electrical nodes with a modified LVP setup. As in normal LVP, she will hold the laser beam stationary on an electrical



(a) Optical setup



(b) Electrical setup

Figure 4.10: Block diagram of the optical and electrical hardware setup for both key extraction and PUF characterization. Reprinted by permission from Springer Nature Customer Service Centre GmbH: Springer Nature, [41], © 2016.

node of interest. However, the detector signal will then not be fed into an analog-to-digital converter card in a PC for averaging, but instead into a spectrum analyzer. The running ring oscillator will then modulate the reflected light with its own operational frequency, which will then also be present in the electrical detector signal. Using the spectrum analyzer on this signal should then reveal the precise RO frequency as a distinct peak in the spectrum.

If the attacker performs this procedure for all the ROs of the PUF, she will be able to predict the outcome of the frequency comparison of any two ROs and will thus be able to predict the binary PUF response.

4.2.5 *Combined Hardware Setup*

Following the concepts of Sect. 4.2.3 and 4.2.4, a hardware setup was devised which allows for experimental evaluation of the discussed attack scenarios. This section will present and explain said setup briefly.

Fig. 4.10 shows a block diagram of both the optical and electrical hardware setup created for the LVI/LVP experiments. The optical setup consists mainly of a Hamamatsu "Phemos-1000" laser scanning microscope. For laser voltage probing and laser voltage imaging measurements, the Phemos is equipped with a highly stable laser light source (Hamamatsu C12993), as well as an LVI/LVP preamplifier (Hamamatsu C12323). It furthermore incorporates an Agilent Acqiris analog-to-digital converter (ADC) card and an Advantest U3851 spectrum analyzer. The laser source used for this setup emits radiation at 1319 nm wavelength. The laser beam is input into the system, de-

flected by galvanometric scanning mirrors and then focused using the objective lens. The focused laser beam then penetrates into the DUT through the device backside. The reflected portion of the laser light is captured by a detector, whose signal is then fed into the LVP/LVI preamplifier. The output of the preamplifier can then either be routed into the ADC card for averaging and waveform creation (LVP) or can be sent to the spectrum analyzer (LVI). For coarse navigation 5x and 20x objective lenses are available. For detailed measurements with high resolution, a 50x/0.76NA lens with silicon thickness correction can be used. The optical setup is able to deliver a beam of about 50 mW power into the DUT if the laser is set to 100% optical power. A PC with the Phemos control software is used to operate the optical setup and create reflected light images as well as LVI/LVP data.

For the device under test (DUT), Altera Cyclone IV FPGAs with part number EP4CE6E22C8N were chosen, which are manufactured in a 60 nm process. The devices come in a 144-pin TQFP package, which was opened from the backside and thinned to 25 µm using an Ultratec ASAP-1. The samples were then inversely soldered to a custom PCB. Bond wires originally leading to the removed "exposed ground pad" were reconnected using silver conductive paint. For configuration, a JTAG connection was used.

The electrical setup uses two power supplies to provide power to the DUT. For the VCC_{INT} power rail, which powers the internal logic, an Agilent E3645A is used. For the I/O power rail (VCC_{IO}) and the analog/PLL power rail (VCC_A), a Power Designs Inc. model 2005 is used. VCC_{IO} and VCC_A are both set to 2.5 V and VCC_{INT} is set to 1.2 V. All these voltages lie within the range recommended by the DUT manufacturer [2]. A Rigol DG4162 function generator supplies reset as well as clock signals to the DUT. The reset signal is furthermore connected to the Phemos hardware, to be utilized as a trigger for LVP waveform acquisition. The reset and clock signal, as well as an auxiliary DUT output, are connected to a Teledyne LeCroy WaveMaster 8620A oscilloscope to monitor proper DUT operation, as well as to conduct simple tests. To be able to perform the spectral analysis on the power rail mentioned in Sect. 4.2.4, a software-defined radio (SDR) is AC-coupled to VCC_{INT}. The SDR is a $20 USB dongle type which uses a Rafael Micro R820T tuner and a Realtek RTL2832U chipset. As this chipset is well supported by free and open source software, multiple options for control are available. For this setup, the "Gqrx" program [18] and the python script "RTLSDR Scanner" [20] are chosen. If the desired spectral bandwidth is below 2.4 MHz, Gqrx can be used for live viewing of the power supply line spectrum. If a larger bandwidth is needed, RTLSDR Scanner can be used to create a wider spectrum using multiple measurements, at the cost of an increased measurement time.

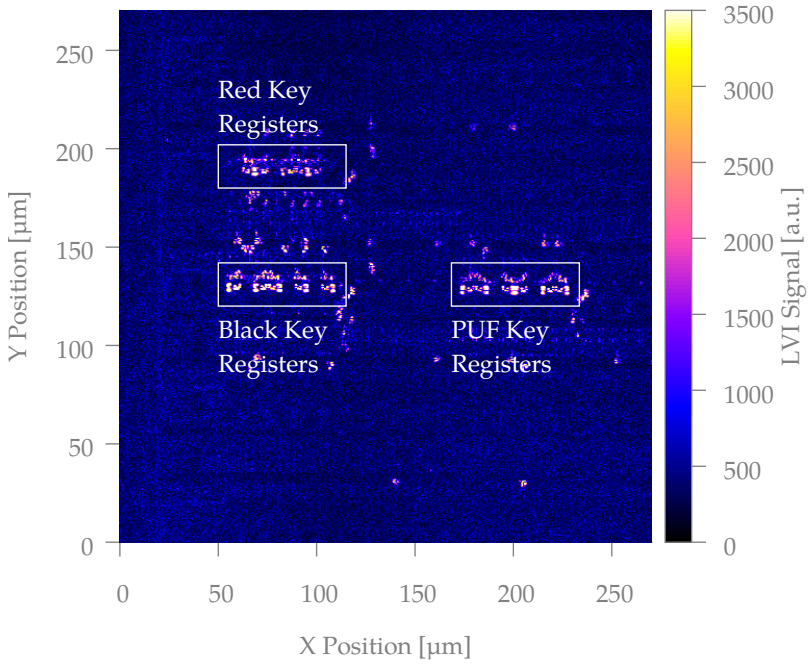


Figure 4.11: LVI overview map of the area containing the parallel POC implementation, taken at the boot loop frequency. The locations of the red key, black key, and PUF key register blocks have been highlighted in white.

4.2.6 *Experimental Results*

This section presents the results achieved by performing the attack approaches of Sect. 4.2.3 and 4.2.4 using the setup of Sect. 4.2.5.

4.2.6.1 *Key Probing*

For achieving the experimental results in this section, the hardware setup and POC implementation were used in a boot loop configuration, as described in the previous sections. For these experiments, the *black key* was set to `0b10101101`, the *PUF key* was `0b11011011` and the resulting *red key* therefore `0b01110110`. All I/O signals were supplied using a 2.4 V high and 0 V low level.

PARALLEL IMPLEMENTATION For the parallel implementation as previously presented in Sect. 4.2.2, a 5 MHz reset loop frequency was used. The clock used to drive the key generation logic was set to 50 MHz. Both clock and reset signal were 50% duty cycle. For LVI, the 50x objective lens was used, along with 10% laser power and a 3.3 ms pixel dwell time. The spectrum analyzer of the LVI setup was set to the reset frequency and 300 Hz bandwidth.

With these settings, an overview LVI map of the area containing the POC implementation was acquired, which can be seen in Fig. 4.11. It

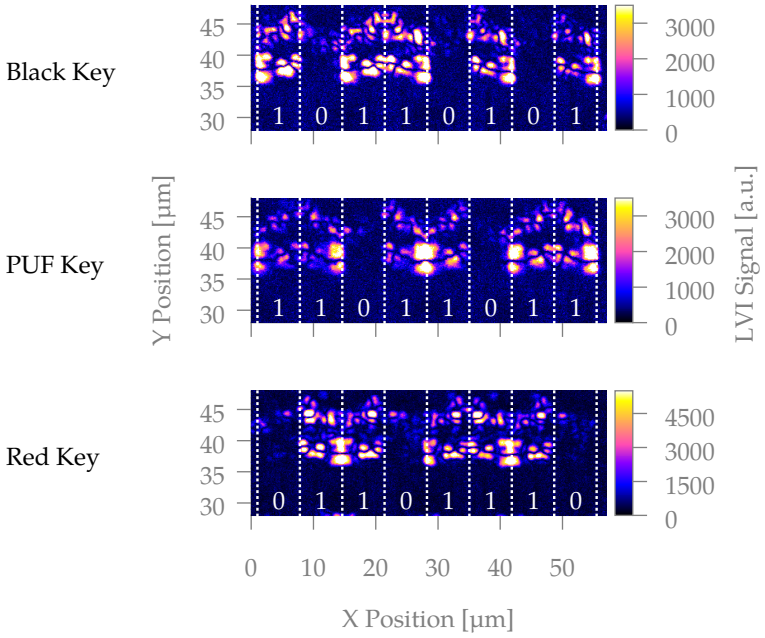


Figure 4.12: Detailed LVI measurements for the red key, black key, and PUF key register blocks with bit states annotated. Logical element (LE) boundaries are indicated by the dashed lines. The key bits can easily be extracted from the LVI activity in each register. Register 7 is on the right-hand side and register 0 is on the left.

is evident that there is some circuitry actively switching at the boot loop frequency. Furthermore, three distinct main areas of activity can be identified which have been highlighted in Fig. 4.11. Comparison with the POC implementation placement shows that these three areas are indeed the register blocks for the black key, the PUF key, and the red key. It should be noted that for an attacker to perform the same unambiguous register block identification, she would need to have partial knowledge of implementation or instead would have to examine all potential key register locations individually, as already discussed in Sect. 4.2.3.

To examine if the key data can actually be extracted, detailed LVI maps utilizing the 50x lens and a 4x scanner zoom were performed on the register block areas. The results of these measurements are depicted in Fig. 4.12. In this figure, the boundaries of the FPGA logical elements (LEs), which contain one register each, are indicated by dashed lines. The bit states of the individual registers are also given in the figure. It should be noted that register 7 is on the right-hand side and register 0 is on the left because of device orientation. These results clearly show that the expected behavior discussed in Sect. 4.2.3 is occurring: registers receiving a “1” bit contribute to the LVI signal while registers receiving a “0” remain inactive. The peculiar activ-

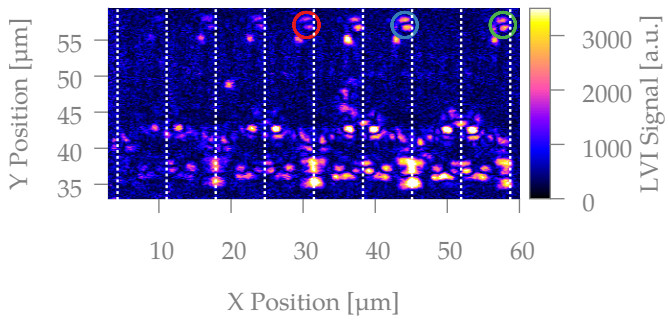


Figure 4.13: LVI map of the red key register block for the serial POC implementation. Locations chosen for probing are indicated by circles. The resulting probing waveforms are depicted in Fig. 4.14. Note that register 7 is on the right-hand side and register 0 is on the left.

ity patterns inside each active LE are caused by the underlying ASIC structure of the LEs and will not be analyzed further. However, it is evident that it is possible to extract the black, PUF and red key bits from these measurements using a simple activity/no activity distinction. Detailed examination of the LVI maps reveals a change in signal intensity and pattern shape among the three different measurements. This can be explained by a slightly different focus position for each measurement due to refocusing. Furthermore, the inverted register order already hints at a possible countermeasure: if the registers are not implemented in this straightforward way but are instead scrambled in their order or even dispersed among other active circuitry, the attacker will have a much more challenging task in figuring out the actual register-to-bit mapping. As a result, the only information about the key that she might gain is the number of "1" bits in the key or even no information at all. Nevertheless, if the registers are implemented naively or if the register-to-bit mapping can otherwise be acquired, the key can be extracted directly.

SERIAL IMPLEMENTATION For the serial implementation, the reset frequency was lowered to 1 MHz, as it needs a larger number of clock cycles to execute the key shift through all registers. Furthermore, the reset signal duty cycle was changed to 58% as a makeshift trigger delay. Also, the laser power was slightly increased to 15% and the pixel dwell time was set to 1 ms. With these settings, an LVI map was acquired for the red key register block using the 50x lens and a 4x scanner zoom. The resulting LVI data can be seen in Fig. 4.13. As expected, no simple activity/no activity distinction for the key registers can be made. Instead, the registers show different levels of activity as already discussed in Sect. 4.2.3. Following the same discussion, an attacker could expect one of the registers on the right-hand side to be the shift-in register, as they have the highest level of LVI activity.

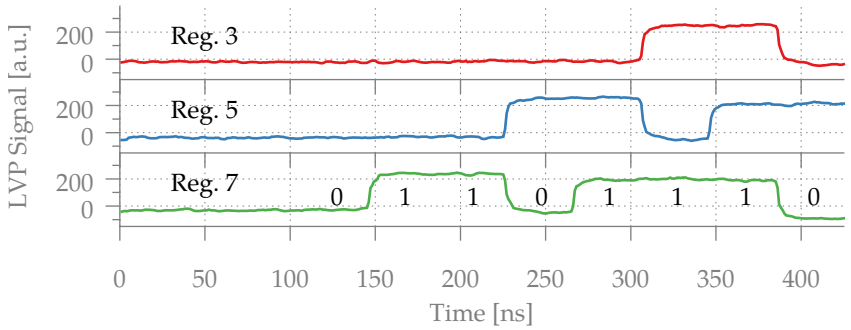


Figure 4.14: LVP waveforms for the red key registers, probed at the locations indicated in Fig. 4.13. The registers can easily be identified by the number of bits shifted through. The bottom signal turns out to be register 7, whose waveform reveals the complete key.

Indeed, probing of the rightmost register reveals it to be register 7, which is the shift-in register. Of course, this could also have been determined by looking at the POC implementation placement in the FPGA, however, as discussed previously, an attacker might not have access to this data.

To illustrate the data extraction, laser voltage probing was performed on register 7, 5, and 3, at the locations indicated in Fig. 4.13. Probing was not performed directly on the main activity area in the lower LE half as the chosen locations at the top were able to deliver a better signal to noise ratio, probably due to their more isolated character. The resulting probing waveforms are depicted in Fig. 4.14. It is evident that all the red key bits can be recovered from the waveform of register 7. For completeness, the waveforms of register 5 and 3, which are further down the shift register, are also given. It is evident that they only receive a part of the key bits during the shift-in operation. For acquiring the probing waveforms shown here, the data of multiple reset loops was integrated. Unfortunately, the Phemos control software does not output the number of integrations, as this value is generated automatically from a number of measurement parameters. However, if all parameters are set to the fastest possible acquisition there is a known lower limit for the number of integrated waveforms in the software. This limit is 100,000 waveforms in the case discussed here. The probing measurements of Fig. 4.14 were repeated using this setting and the signal to noise ratio was still good enough to easily distinguish bit states. It is therefore expected that the data extraction illustrated in this section will still work for a much lower number of integrations if the limit is removed from the software.

Thus, key extraction is also demonstrated for the serial implementation. In conclusion, it can be said that an attacker should be able to extract the key bits using just LVI or a combination of LVI/LVP, depending on which implementation is used.

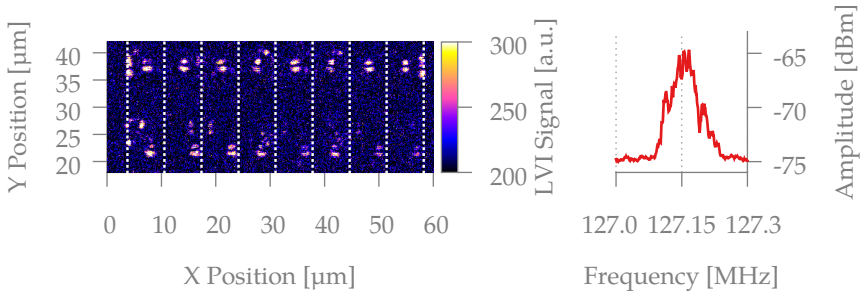


Figure 4.15: LVI map (left) and LVP spectrum (right) of the examined ring oscillator circuit. Logical element (LE) boundaries are indicated by the dashed lines in the LVI map. The spectrum can be acquired at any of the active locations in the LVI map.

4.2.6.2 *PUF Characterization*

To perform the characterization of the ring oscillator (RO) PUF, a first rough frequency estimate of all running ROs was determined using the software-defined radio (SDR) and power analysis in the frequency domain, as discussed in Sect. 4.2.4. Using this estimate with slight variations while setting the spectrum analyzer of the LVI setup to a wide bandwidth (about 100 kHz) led to the discovery of the ROs in the LVI map. One of the ROs was then chosen for further analysis. The LVI map of 8 inverters of said RO is shown on the left side of Fig. 4.15.

This data was acquired using the 50x lens, a bandwidth of 100 kHz and a frequency of 127.3539 MHz for the spectrum analyzer, 60% laser power, and 0.33 ms pixel dwell time. When comparing the bandwidth of this measurement to that of the measurements for the key data extraction from the previous sections, it becomes clear that a much wider filter bandwidth was used (100 kHz vs. 300 Hz). This is due to the fact that ring oscillators, especially with few inverters, are relatively poor clock sources as compared to a conventional clock generator. Their spectrum is more widespread, and therefore the overall power of their signal is distributed over a larger range of frequencies. As a result, the spectrum analyzer bandwidth has to be increased to allow the capture of a decent total amount of signal power. Unfortunately, this also leads to a higher level of noise power being registered by the spectrum analyzer, as the wider bandwidth allows more noise to enter the system. As a consequence, this leads to a worse signal to noise ratio in the LVI map. This effect can be seen clearly when comparing Fig. 4.13 and Fig. 4.15, where the LVI signal has dropped significantly even though the laser power has been raised from 15% to 60%.

However, although ROs seem to be more challenging to analyze using LVI because of the aspects just discussed, the active electrical

nodes of the ring oscillator can still be identified in the LVI map in Fig. 4.15. Therefore, it was possible to hold the laser beam stationary on one of these nodes and feed the resulting detector signal directly into the spectrum analyzer, instead of the waveform acquisition card, as discussed in Sect. 4.2.4. For this measurement, the laser power was further increased to 73% and the spectrum analyzer put into its conventional (non-LVI) frequency sweep mode. The spectrum analyzer resolution bandwidth was then set to 30 kHz and the video bandwidth to 10 Hz. The resulting measurement data can be seen on the right in Fig. 4.15. The spectrum of the RO activity is clearly visible about 10 dBm above the noise floor. The position of the peak value of this spectrum can be determined to lie at 127.168 MHz. The frequency at which this RO operates can thus be determined precisely. It should also be mentioned that the resolution bandwidth of the spectrum analyzer does not constitute the limit for the precision with which the RO frequency can be determined. As the measurement shown here consists of just a single spectrum analyzer frequency sweep, an attacker would be free to average multiple measurements. As she will only be interested in the average frequency of each RO, to determine which one is faster (see Sect. 4.2.4), she can choose to evaluate the peak of an averaged spectrum for each RO. This way she can increase the precision of her measurements comfortably by choosing an appropriate number of averaged measurements. It should be noted that a shift in frequency of about 0.15% was discovered in the probed RO when the laser power was increased from 60% to 73%. However, as all ROs are made up of the same elements, the same frequency shift should occur if a different RO is probed in the same way with the same laser power. Therefore, this should not hinder the attacker in deciding which ROs are faster than others. On the other hand, if the characteristic of the frequency shift with varying laser powers can be determined, an attacker might also measure a single RO with different laser powers and extrapolate to the value at zero power. Nevertheless, the existence of a frequency shift caused by the laser irradiation points at a possible countermeasure, which might use this fact to detect a probing attack while only using the standard FPGA logic fabric. For an example of such a protection scheme see Sect. 6.1.2.

With a technique such as optical probing in mind, it becomes apparent that the exchange of direct NVM key storage for implementations using PUFs does not raise the security level as high as one would have expected in the first place. However, they still offer a significant advantage over conventional key storage schemes, as they raise the effort required for an attack, compare also the key extraction attacks from Chapter 3. Yet, as more and more sophisticated key storage solutions can be thought of to hinder the attacker, it should be kept in mind that optical probing has the potential to circumvent all of them. As LVP has the potential to probe the bitstream data directly after on-

chip decryption, it might make all schemes protecting the key itself irrelevant. As this approach is also a viable option for an attacker, it will be evaluated in the next section.

4.3 PLAINTEXT DATA EXTRACTION

The main aim of this section is to assess the feasibility of directly extracting the plaintext as opposed to the key data. This can under certain circumstances be advantageous for an attacker. For example, key extraction can be hindered by more sophisticated key storage solutions. Examples of such techniques are the PUF-based storage solutions as discussed in the previous sections or other approaches, such as "secret sharing". In the latter case, the secret key data is spread across several shares, requiring an attacker to acquire all of the shares to reconstruct the key, thus raising the required effort. If the plaintext is attacked instead, there is no such protection scheme available. As the plaintext bitstream needs to be used by the FPGA to configure itself, it inevitably needs to be present at some point in the device.

Moreover, this section also aims to address the following questions raised by the results of the previous experiments: for extracting secret data from FPGAs the previous sections used package modifications to access the silicon die. However, more modern devices use flip-chip packages which expose the silicon backside directly. This leads to the question if such packages might be used for *non-invasive* optical probing attacks, eliminating the need for DUT modification. Additionally, previously a proof-of-concept implementation on the FPGA logic fabric was used. It is therefore questionable if similar attacks are possible on real commercial devices which employ application-specific integrated circuits (ASICs) for their decryption cores. Assuming the same technology size, a decryption ASIC can have a much smaller size than a decryption core on the logic fabric, potentially making the features relevant for the attack optically unresolvable. Furthermore, no information is usually released about the layout and implementation of the decryption ASICs, requiring an attacker to reverse engineer these complex circuits.

To evaluate these questions, the feasibility of plaintext extraction will be assessed on an actual commercial device, with only public information about the decryption ASIC available. The scenario is that an attacker has physical possession of such a device and seeks to extract the plaintext bitstream. For this, she has limited access to FA equipment, for example by renting. The main idea is for the attacker to analyze the decryption core, find the location where the plaintext data leaves the core, and then extract it. The setup is mostly the same as in Sect. 4.2, however, the coherent laser source has been exchanged for an incoherent light source, details about this will later be given

in Sect. 4.3.2. For this reason, the optical probing techniques can no longer correctly be referred to as *laser* voltage probing and imaging (LVP/LVI), but will instead be referred to as electro-optical probing and electro-optical frequency mapping (EOP/EOFM). Apart from the difference in coherence, these terms can be seen as interchangeable with regard to the functionality they provide, and EOP/EOFM functions by the same terms that were already explained in Sect. 4.1.

The structure of this section is as follows: first, the concept for the analysis of the decryption core will be described in Sect 4.3.1. Following that, a suitable hardware setup will be developed and an appropriate DUT will be selected in Sect 4.3.2. Finally, the attack concept will be applied to the DUT and the results thereof will be presented and analyzed in Sect 4.3.3.

4.3.1 *Plaintext Data Extraction Concept*

This section will describe how an attacker would most likely proceed to analyze an unknown decryption ASIC in an FPGA and find the locations from which the plaintext can be extracted. The scenario is as follows: it is assumed that an attacker is in possession of a board containing an FPGA in a flip-chip bare-die package. The FPGA loads the encrypted bitstream data from an NVM to configure itself. The attacker now seeks to gain access to the plaintext bitstream to extract either intellectual property (IP) or secrets such as authentication keys contained therein.

Apart from conventional tools such as a soldering iron and a laptop, the only professional equipment the attacker has access to is an optical probing microscope which she rents at a failure analysis lab.

To perform her attack, there are four basic steps that she will need to execute:

- Localize the general configuration logic area on the silicon die
- Localize the decryption core in the configuration logic
- Localize the logic gates carrying plaintext data in the decryption core
- Extract the data from the found plaintext gates

To avoid harming the target device, the attacker will probably perform these steps first on a training device of the same type. This will also enable her to transfer manipulated bitstreams to the device and allow her to set arbitrary security settings and keys and have full control of the device in general. As soon as she has analyzed the training device in this way and found the plaintext gates, she can move on to mount her attack on the actual target.

Her first step will be to acquire reflected light images of the device, to look for general structures in the layout. As it is assumed that she

has access to an optical probing system by renting, this system can be expected to use the common optical probing wavelength of 1.3 µm for illumination. As this wavelength is able to penetrate through hundreds of micrometers of silicon and the device is in a bare-die flip-chip package, she will be able to acquire reflected light images *without any modifications* to the DUT.

In these images, she will directly be able to distinguish the FPGA logic fabric from dedicated functions realized as ASIC blocks. As the FPGA's logic fabric consists of many identical structures, such as look-up tables (LUTs) and SRAM memory, which are arranged in rows and columns, it should have a highly ordered appearance. ASIC blocks, on the other hand, should have a more irregular appearance. As the configuration logic will be implemented as an ASIC, the attacker will focus on the more irregular structures. As she knows that there will be only one configuration ASIC, she can disregard all blocks which are appearing multiple times across the DUT. Additionally, she can consult the datasheets and technical description of the device to acquire additional hints.

She can then start to use optical probing to exclude further candidates. If she is able to estimate some frequency present in the configuration logic, she can perform EOFM at this frequency. In the ideal case, this should then reveal activity in only one of the candidate locations. Alternatively, she might also try to induce a certain frequency into the configuration logic, as was done for the experiments of Sect. 4.2.3.

To then find the decryption core, she can compare the activity for encrypted and unencrypted bitstreams. Areas that are only active for encrypted bitstreams have a high probability of taking part in the decryption process and are therefore strong candidates for the decryption core. When she has narrowed down her search to these areas, she will then have to find a way to distinguish gates carrying the plaintext from all other logic.

This step will not be as straightforward as the previous steps. However, if she recalls some basic properties of bitstream encryption ciphers, she will be able to induce a frequency into the plaintext gates which she can then detect using EOFM. Most modes of operation used for block ciphers for FPGA bitstream encryption have the valuable property of destroying structures and frequencies present in the plaintext during ciphertext creation. This property raises the security of the cipher as it hinders attacks such as frequency analysis. Furthermore, it also means that the encrypted bitstream data is basically indistinguishable from noise in the frequency domain.

However, when the plaintext is regenerated inside the device, it will obviously possess all the structures and frequency components of the original plaintext. Thus, if a certain frequency is inserted into the plaintext data, this frequency will vanish in the ciphertext, and

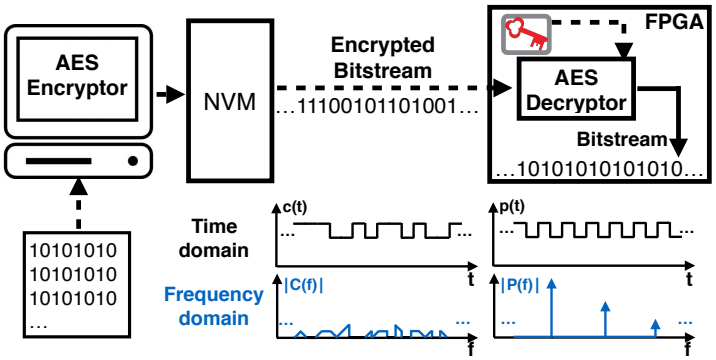


Figure 4.16: Plaintext frequency induction. The spectrum $C(f)$ of the ciphertext function $c(t)$ contains no dominant frequency components, as the cipher destroys all structures which were present in the plaintext. However, when the plaintext $p(t)$ is regenerated as a periodic function inside the device, its spectrum $P(f)$ contains specific harmonics. [81]

only reappear in the plaintext that leaves the decryption core, see Fig. 4.16. If this frequency is then used to perform EOFM, all gates carrying the plaintext should become detectable, while gates carrying the ciphertext should not give rise to a signal.

Mathematically, if a regular repeating "10" bit pattern in the plaintext for frequency generation is assumed, a time-periodic plaintext will be obtained. The periodic voltage of the plaintext signal $p(t)$ can then be written as a square wave [88]:

$$p(t) \quad = \quad 2[H(t/T) - H(t/T-1)] - 1$$

Here, $H(t)$ represents the Heaviside step function and T the bit duration. If this function is written as a Fourier series [88], it can be seen that it only contains specific harmonics:

$$p(t) \quad = \quad \frac{4}{\pi} \sum_{n=1,3,5,\cdots}^{\infty} \frac{1}{n} \sin\left(\frac{n\pi t}{T}\right)$$

In the case of this example, the spectrum will contain only the fundamental frequency $f = \frac{1}{2T}$ and its odd harmonics. This is also illustrated in Fig. 4.16. In other words, every location in the DUT that carries the plaintext signal will be visible in an EOFM measurement carried out at the fundamental frequency or the odd harmonics. It should be noted that the formula also indicates that the higher harmonics will generate a weaker signal.

This now enables the attacker to easily find the plaintext gates: she simply has to manipulate the bitstream to contain a certain fundamental frequency. When this bitstream is then encrypted and loaded from NVM into the FPGA, the fundamental frequency will only be regenerated after the data has left the decryption core, see Fig. 4.16. An EOFM measurement at this frequency will then reveal the plaintext

gates to her. It should be noted, that she needs to choose a frequency which is unlikely to be present in other parts of the device. When she has identified the plaintext gates in this way, she can then proceed to extract the plaintext data using EOP.

Yet, she has to keep in mind that, depending on the actual implementation, the generated frequencies might be altered when the data is processed in the decryption core. However, she can mitigate this by creating a model of the data processing and deducing the generated frequencies from it. She can then either calculate the frequencies she needs to use for EOFM or even use the knowledge to generate a bitstream which generates a specific frequency directly.

As a simple example, a bitstream containing alternating ones and zeroes is considered. If this bitstream is loaded in a serial fashion using a configuration clock (CCLK), and the formula for the fundamental frequency $f = \frac{1}{2T}$ is applied, it can be seen that all gates carrying the data will generate an EOFM signal at the frequency $f_{CCLK}/2$. Yet, if at some point a parallelization occurs, this frequency will be changed. If, for example, the data is loaded onto a 32-bit bus, the ones and zeroes will be aligned in every bus word and the signal on each bus line will be static. Thus, no EOFM signal will be generated at all. Yet, if the attacker takes this parallelization into account she can simply modify the data to contain 32 ones followed by 32 zeroes in a repeating pattern. This will now cause the individual bus lines of the 32-bit bus to toggle for every word. If the input into the device still happens in a serial fashion with f_{CCLK}, she can easily calculate the frequency generated by the bus lines to be $f_{CCLK}/32/2$ or $f_{CCLK}/64$ and therefore use EOFM to detect the bus lines carrying the data.

As the attacker has EOFM available, she can easily derive predictions from her current model and test them using optical probing. Therefore, she will be able to gradually develop a model matching the analyzed implementation and adjust her manipulated bitstream for frequency generation accordingly. This will then allow her to find the plaintext gates using the adjusted bitstream and finally extract the data using EOP. A feasible attack path is thus outlined.

Apart from assessing the feasibility of the attack in general, an estimate of the amount of effort spent on the attack should also be provided. For this, time tracking software will be used during the experiments on the failure analysis equipment. This will then give a measure of how much time the attacker would have needed to rent at an FA lab to develop the full attack. As access to and rent for the optical probing equipment is considered the limiting factor for this attack, the time needed for, for example, soldering, programming or reading datasheets will not be considered.

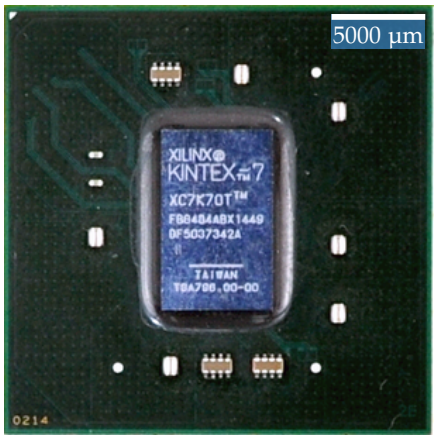


Figure 4.17: Image of a Kintex 7 XC7K70T device in a flip-chip BGA package (FBG484) [51]. In the middle of the package, the exposed silicon backside of the die can be seen.

4.3.2 *Hardware Setup*

This section will present the hardware setup developed to realize the concept for plaintext data extraction presented in Sect. 4.3.1.

4.3.2.1 *Device Under Test*

For the device under test (DUT), a Xilinx Kintex 7 XC7K70T FPGA in a flip-chip BGA package was chosen, see Fig. 4.17. This device is manufactured in 28 nm technology and implements a bitstream encryption scheme of the "simple bitstream encryption" type presented in Sect. 4.2.1. The cipher used in this device is the "advanced encryption standard" (AES) in cipher block chaining (CBC) mode. As the device's bitstream security has already been broken by extracting the key using side-channel analysis [30, 47], it allows for responsible disclosure of all findings during the experiments without causing additional harm to the device's security. It should be stressed that the results published in [30, 47] offer no insight into the gate-level structure of the device, and therefore do not give any advantage for the plaintext extraction attack.

The DUT is mounted on a commercial "Skoll" development board manufactured by Numato Lab [51]. As the board does not have a heatsink mounted, the silicon backside of the DUT is exposed directly. The initial device thickness was measured to be about 700 µm. No die or package modifications were performed.

4.3.2.2 *Optical Setup*

The optical setup is mainly the same as the one already described in Sect. 4.2.5, except for three components which were changed because of defects or general upgrades in the lab. First, the Agilent Acqiris

ADC card for probing waveform acquisition has been exchanged for a LeCroy WavePro 735Zi oscilloscope. Second, the probing light source has been changed from a coherent light source (Hamamatsu C12993) to an incoherent one (Hamamatsu C13193). The new light source has a wavelength of 1330 nm and achieves the following amounts of power on the DUT with the respective lenses: 5x: 63 mW; 20x: 26 mW; 50x: 45 mW. Third, the 50x objective lens has been changed to a 0.71NA type, still with a silicon thickness correction function.

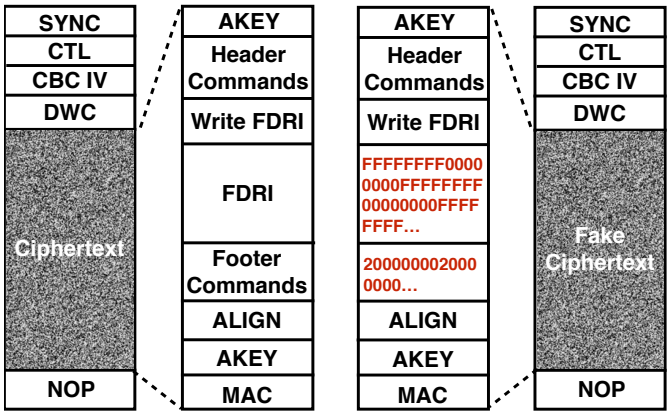
4.3.2.3 *Electrical Setup*

Configuration of the FPGA can either be achieved by using the board's JTAG interface or by loading the bitstream data via a serial peripheral interface (SPI) bus from a 128 MBit on-board flash. The programming of the flash itself is done via an FTDI FT2232H chip over USB. Because of the shorter start-up time, configuration via flash is chosen for the experiments. For this, the "master SPI configuration mode" with standard settings was used, see [91]. In this 1-bit wide serial transfer mode, the FPGA first sends a read instruction to the flash memory via the SPI bus. The flash will then output the bitstream data on its "master in slave out" (MISO) pin, which is connected to the FPGA's "data in" (DIN) pin. The clock for this transfer is called the configuration clock (CCLK) and is generated by the FPGA and fed into the flash via its SCLK pin. The FPGA will then check and decrypt the received data and switch into user mode as soon as configuration has finished without errors.

Power to the board is supplied via the USB connection and then converted to the different required supply voltages. However, during the experiments, the onboard switch mode power supply for the 1.0 V rail was disabled as it caused an increase in noise. Instead, this voltage was provided by an Agilent E3645A power supply. To allow for robust access to the data in (DIN) and configuration clock (CCLK) signals, coaxial cables were soldered to the corresponding printed circuit board traces. An additional cable was added to the PROGRAM_B pin of the FPGA. This connection allows triggering reconfiguration of the device using a low-level pulse. Using a Rigol DG4162 function generator connected to this cable, the configuration can then be triggered repeatedly. Using a manipulated bitstream (see Sect. 4.3.1), this allows for the continuous generation of the switching frequencies needed for EOFM. To trigger the oscilloscope for waveform acquisition in EOP mode, either CCLK, PROGRAM_B, or even the DIN signal can be used.

4.3.2.4 *Manipulated Bitstream Generation*

To easily generate the manipulated bitstreams discussed in Sect. 4.3.1, a Python script with about 400 lines of code was developed. This



(a) Regular bitstream. [84, 91]

(b) Bitstream manipulated for EOFM fundamental frequency generation.

Figure 4.18: Structure of normal and manipulated Kintex 7 bitstreams. [81]

script can also convert bitstream data into a human-readable synopsis and preview the data patterns generated inside the device under the assumption of different bus widths. For encryption and decryption with AES-CBC, the "pycrypto" [38] library is used. To generate a manipulated bitstream, a regular bitstream is first needed, whose structure can be seen in Fig. 4.18a. It should be noted that here only the sections relevant to the generation of the manipulated bitstream will be explained. For information about the other sections, the reader is referred to [84, 91]. To start the generation, the ciphertext portion of a regular bitstream is first extracted and decrypted, see Fig. 4.18a. Size-wise the decrypted portion is mostly made up of the actual configuration data contained in the "FDRI" (frame data register input) block. This block is then replaced with a user-defined repeating pattern. In the case of this example, 32 ones followed by 32 zeroes, see Fig. 4.18b. Additionally, the "footer commands" section, which normally contains the commands to start the FPGA after configuration, is overwritten with "no operation" (NOP, `0x20000000`) commands. This is done to prevent damage to the device which might occur when the device is started with the illegal bitstream data. The data generated thus is then re-encrypted to generate the fake ciphertext, which is then inserted into the regular bitstream structure, see Fig. 4.18b. This manipulated bitstream data can then be loaded into the flash of the FPGA board and be used in combination with the electrical setup to continuously generate the desired frequencies for EOFM. For all the experiments shown here, the key and the initialization vector for the AES-CBC cipher were set to all zeroes.

(a) Reflected light image of the whole die.

(b) Zoomed-in view of (a), showing repeating FPGA logic fabric structures and the more irregular ASIC area.

Figure 4.19: Reflected light overview images of the XC7K70T FPGA. [81]

4.3.3 *Experimental Results*

This section presents the results accomplished using the setup of Sect. 4.3.2 in combination with the approach of Sect. 4.3.1. As opposed to Sect. 4.2, the results will not be visualized based on the EOFM signal level, but are instead thresholded, see e.g. Fig. 4.21. This is due to the fact that for reverse engineering of the decryption core, it is not relevant how strong an EOFM signal is, but just if there is activity in a certain part of the device in general. The threshold is set slightly above the noise floor and locations generating an EOFM signal are shown in green. The used EOFM settings are given in Appendix A.1. To aid in orientation, the EOFM signal maps are overlayed onto reflected light images. To enable readers with a grayscale representation to distinguish between EOFM signal and reflected light image, the reflected light images have been reduced in brightness.

4.3.3.1 *Localization of the Configuration Logic ASIC*

To localize the configuration logic ASIC, reflected light overview images of the die were first acquired, see Fig. 4.19a. To generate this image, multiple 5x lens measurements were combined using stitching software [60]. In this image, the die markings can be seen, compare

(a) Whole configuration ASIC. [81]

(b) Zoomed-in view of the central area.

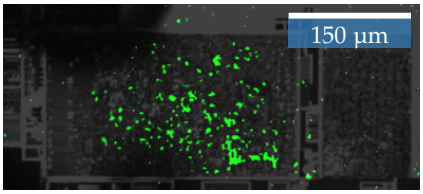
Figure 4.20: Reflected light images of the configuration logic area. The images are 90 degrees tilted with regard to Fig. 4.19.

also Fig. 4.17, as no thinning was performed at all. The markings are mirrored, as the Phemos software automatically flips the image data in backside observation mode. Some differences in the general appearance of the observable structures can be seen. On the one hand, there are strips running from the top to the bottom of the device showing an ordered appearance and covering most of the area. On the other hand, there are "islands" of more irregular structures, for example in the top right corner and in the middle of the die. Fig. 4.19b shows a zoomed-in view highlighting the differences between these areas. Following the reasoning of Sect. 4.3.1, it can be assumed that the ordered structures are the logic fabric, while the other structures are ASIC blocks. Upon investigation of the Xilinx datasheets for this device [92], it becomes clear that the ASIC blocks at the top right corner are GTH/GTX transceivers available in some 7-series devices. The same datasheet shows the configuration logic as a block roughly in the middle of the die, while Fig. 4.19a reveals a strip of somewhat irregular appearance at this location. Therefore, this area is examined more closely.

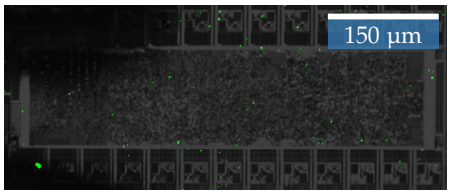
Fig. 4.20a shows a reflected light image of this section which has been tilted with regard to Fig. 4.19, while Fig. 4.20b shows a zoomed-in view of the central portion. These images have been acquired using stitching of multiple 20x lens images. To confirm that this area is actually the configuration logic area, EOFM measurements were performed at the configuration clock (CCLK) frequency, to show circuitry potentially operating on the bitstream data. Indeed, these measurements revealed activity and it was even possible to probe the data entering the configuration logic using EOP. This thus confirms that the area of Fig. 4.20a is indeed the configuration ASIC.

4.3.3.2 *Localization of the AES core*

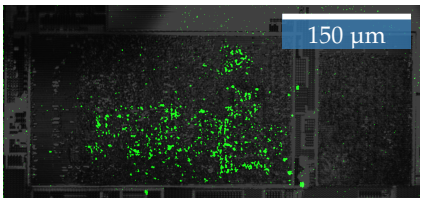
In Fig. 4.20b, two areas can be seen which possess the somewhat random appearance of synthesized logic blocks, one in the left and one



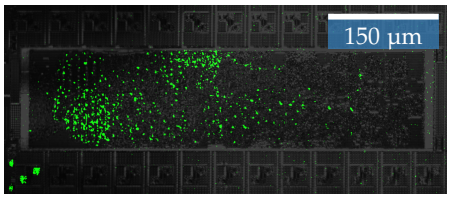
(a) "Always active" region, unencrypted bitstream.



(b) "Encryption only" region, unencrypted bitstream.



(c) "Always active" region, encrypted bitstream.



(d) "Encryption only" region, encrypted bitstream.

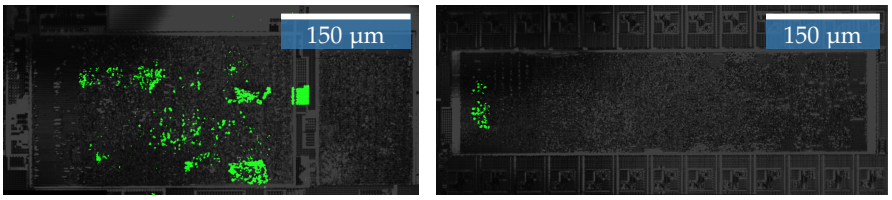
Figure 4.21: Comparison of CCLK activity in the configuration area for different bitstream settings. [81]

in the right half. As in ASIC creation logic synthesis is usually performed to convert a more abstract representation of a circuit's function into a concrete gate level layout, this indicates that these blocks perform some dedicated functions. Therefore, they are candidates for the AES decryption core. As discussed in Sect. 4.3.1, the next step was the comparison of activity for encrypted and unencrypted bitstreams to identify the AES core in these areas. For this, EOFM measurements were performed at the CCLK frequency with different bitstream settings, to see which parts of the ASIC are only active for encrypted bitstreams. These measurements were performed with f_{CCLK} set to the standard "3 MHz" setting in all of the configuration ASIC area. The results revealed an area that was always active, as well as an area that became active only for enabled encryption, see Fig. 4.21.

Based on these results, the region of Fig. 4.21a and 4.21c is assumed to contain the basic configuration logic and is thus named the "main core", while the region of Fig. 4.21b and 4.21d is assumed to be the AES core.

4.3.3.3 *Determination of the Bus Width*

To allow for identification of the gates carrying actual data, as opposed to gates handling the clock signal, additional EOFM measurements were performed. To start with the simplest case, and therefore leave less room for errors, this was first tested with encryption turned off. For these measurements, a bitstream with alternating ones and zeroes was generated and transferred onto the board. As discussed in Sect. 4.3.1, this should lead to the gates handling the data being detectable at $f_{CCLK}/2$ if the bitstream is transferred in a



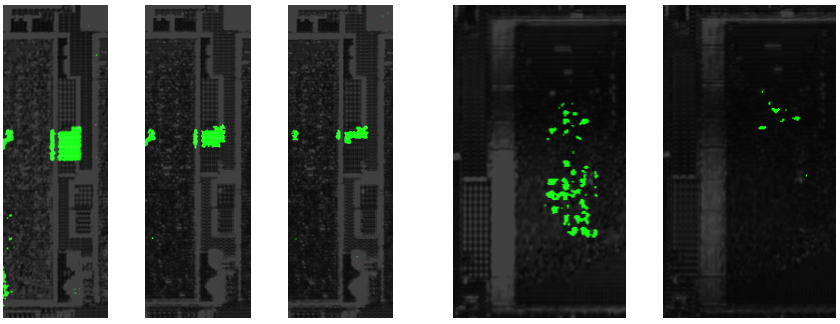
(a) Main core area. (b) AES core area.

Figure 4.22: EOFM measurements at the 32-bit word frequency ($f_{CCLK}/64$) revealing logic gates potentially connected to the 32-bit data bus in the "main" and "AES" areas. Although an unencrypted bitstream is used, the AES input logic is visible in (b). [81]

serial fashion. However, this only revealed minor activity. As the datasheet [91] states that the device uses 32-bit words for its basic bitstream format, a 32-bit bus hypothesis was tested next. For this, a bitstream with a repeating pattern of 32 "1" bits followed by 32 "0" bits was generated, which should lead to the data gates being detectable at $f_{CCLK}/2/32$, see Sect. 4.3.1. Yet, assuming the standard "3 MHz" CCLK, this would require EOFM measurements to be taken at $f_{EOFM} = f_{CCLK}/64 = 3\ MHz/64 = 46.9\ kHz$. As the lower frequency limit of the amplifier is 100 kHz, f_{CCLK} was increased to the "12 MHz" setting. Note that, as discussed in Sect. 4.3.1, the attacker is free to do this since she uses a training device at this stage.

The resulting measurement data for the main core area can be seen in Fig. 4.22a, which reveals a considerable amount of activity. This indicates that the 32-bit bus width hypothesis is correct. Further tests with larger bus widths equally supported the 32-bit assumption. It is noteworthy that there is a rectangular block of EOFM activity at the right edge of the main core structure in Fig. 4.22a. Its ordered appearance and placement suggest that it might be a data input or output port of the main core logic. Astonishingly, although encryption was disabled for this measurement, there was also activity in the AES core, see Fig. 4.22b. This is most likely caused by the very first stages of the AES input logic, such as data buffers for signal recovery. It can be assumed that this first stage is always connected to the data bus regardless of active or inactive encryption. Only at the first gate requiring a clock, such as a latch or a register, the data signal would stop to propagate. Thus, with the AES clock inactive because of the disabled encryption, only the first stage transistors would be visible in the EOFM measurements.

To further verify the bus width, the measurements were repeated with the fundamental frequency of $f_{CCLK}/64$ being generated only on a part of the bus lines. For this, the data for the inactive bus lines was simply set to "always zero" in the bitstream data. The resulting EOFM measurements can be seen in Fig. 4.23. It is evident, that the results also support the 32-bit hypothesis and also indicate that the



(a) "main", 32 bit active (b) "main", 16 bit active (c) "main", 8 bit active (d) "AES", 32 bit active (e) "AES", 8 bit active

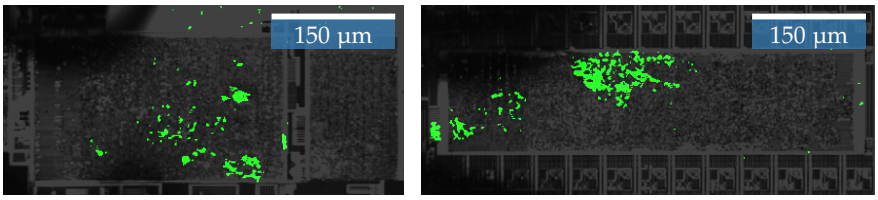
Figure 4.23: EOFM measurements at the 32-bit word frequency ($f_{CCLK}/64$) for a different number of active bits on the data bus in the "main" and "AES" areas. This measurement has been acquired with decryption disabled to show only the input logic in the AES core. [81]

bus lines are laid out in an ordered fashion. Therefore, the main core output and the AES input data buses are assumed to be 32-bit wide.

4.3.3.4 *Localization of the Plaintext Gates*

To allow for detection of the gates carrying the plaintext, the approach of Sect. 4.3.1 was used, see also Fig. 4.16. To generate the fundamental frequencies via the plaintext data, a suitable bitstream was created and encrypted using the python script of Sect. 4.3.2.4 and then transferred to the flash memory of the board. As the fundamental frequencies contained in the plaintext data are only regenerated in the device when the data has been decrypted, this should allow to identify the plaintext data output gates of the AES core, see Sect. 4.3.1. A first measurement using a bitstream generated under the assumption of a 32-bit AES output bus showed no activity. As AES is a 128-bit block cipher, a measurement under the assumption of a 128-bit bus was performed next, resulting in an EOFM frequency of $f_{EOFM} = f_{CCLK}/256$. Similarly to the previous section, for this f_{CCLK} needed to be increased to the "33 MHz" setting because of the preamplifier's lower frequency limit.

Fig. 4.24 shows the resulting EOFM data of these measurements. This time, a considerable amount of activity can be observed in the AES core, see Fig. 4.24b, which indicates that the 128-bit plaintext bus assumption is correct. Additionally, there is a lot of activity in the main core as well, see Fig. 4.24a, which suggests that the data is fed back into it after decryption. This behavior can be explained by the fact that the commands contained in an *unencrypted* bitstream and a *decrypted* bitstream have the same format [91]. Therefore, it makes sense to use the same logic to interpret these commands in both cases.



(a) Main core area. (b) AES core area.

Figure 4.24: EOFM measurements taken with an encrypted bitstream at the plaintext data frequency ($f_{CCLK}/256$) to locate gates potentially carrying the decrypted bitstream data. In the AES area of (b) at the leftmost edge, a block-like structure is visible which might indicate an output port. [81]

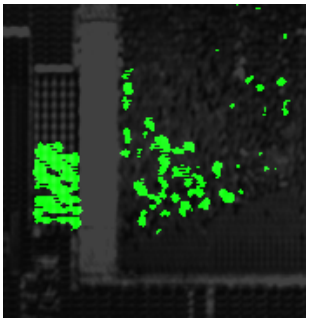
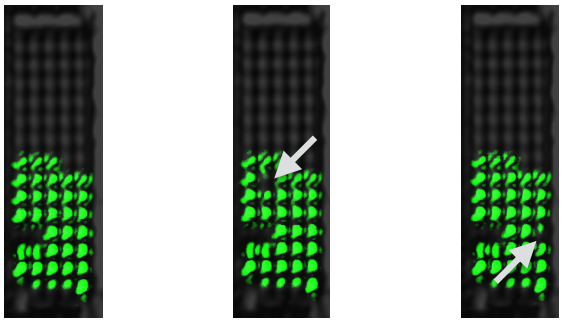


Figure 4.25: Detailed EOFM data of the potential AES output port area. Suspected AES output port (left) and additional plaintext data gates inside the logic mesh (right). [81]

The results would then indicate that this logic is located in the main core, which is why the data is fed back into it.

To perform plaintext data extraction, a suitable candidate for the AES output port needed to be selected. Because of its ordered appearance and location, the block-like activity area at the leftmost edge of the AES in Fig. 4.24b seemed promising. A zoomed-in measurement of this area in Fig. 4.25 reveals similarity to the data output of the main core, compare Fig. 4.23a. This further indicates that this is a promising output port candidate.

The area was thus assumed to be the AES output. For data extraction, it is now important to determine which points of EOFM activity correspond to the individual bus lines, to later allow for reconstruction of the data extracted from each bus line. This process is referred to as logical-to-spatial mapping. For this, a bitstream was generated which only flips the bits on a single bus line, to be able to easily identify its location via EOFM. However, during these measurements, it became apparent that most of the bus lines did not generate any EOFM signal at all. Activity could only be detected for 32 bits of the assumed 128-bit bus. However, these 32 bits showed the expected behavior of enabling or disabling EOFM activity spots in the AES output port, see Fig. 4.26. Further analysis revealed, that these 32 bits



(a) 32 bits active. (b) Bit 25 inactive. (c) Bit 13 inactive.

Figure 4.26: EOFM data for the AES output port with different bits deactivated. [81]

always belonged to the third word of a 128-bit block in the bitstream file.

Thus, the achieved results seem to be conflicting: on the one hand, the used EOFM frequency of $f_{CCLK}/256$ and the previous unsuccessful attempts with $f_{CCLK}/64$ suggests a periodicity matching a 128-bit bus. On the other hand, the EOFM data indicates spatially that there is a 32-bit bus, which for some reason only shows activity for words 128 bit apart.

A straightforward model, which assumes that the decrypted data is output at regular intervals, is not able to resolve this inconsistency. However, a model assuming irregular output can.

According to the datasheet, the FPGA is capable of handling bitstream data much faster than with the settings used in this experiment. This is due to the fact that a simple serial interface was used and that much larger data input bus widths and higher clock speeds are available [91]. It is thus possible that the configuration logic interpreting the bitstream data is always run at a standard speed, which would be the speed needed to process the fastest bitstream data input possible. This would also mean that the data of a 128-bit AES block that has just been decrypted could be immediately processed by the configuration logic as fast as possible. As the basic format of the bitstream commands is 32-bit words, it seems plausible that this is also the amount of data that is processed at once. While the next block is then loaded through the slow serial interface, the configuration logic would have to wait until the next block is ready. This would result in four 32-bit words being transferred in fast succession on the AES output bus, followed by a relatively long pause while the next block is arriving. Such a model would be able to explain the behavior observed during the experiments.

For every bus line, there would be three bits transferred in a short amount of time, whereas the fourth bit would stay on the bus line while the next AES block is decrypted. As soon as the next block

is available, this process would then repeat. As the first three bits are only present at the bus briefly, they will not have a significant influence on the EOFM signal at the fundamental frequency. The fourth bit, however, will. This can explain why only one in four 32-bit words seems to be "active" in the EOFM measurements, as the "active" word probably corresponds to the fourth bits on the individual bus lines. As the next block takes exactly 128 CCLK clock cycles to arrive, the changes on the bus lines caused by the fourth bits will furthermore also generate an EOFM signal at $f_{CCLK}/256$, if toggling of all bits for every 128-bit block is assumed, as was used in the experiments. This model is thus able to explain the observed results well. For verification, EOP measurements were performed at arbitrarily chosen activity spots of Fig. 4.26, using a bitstream whose bits were flipped for every 32-bit word. This indeed revealed three "fast bits" followed by a fourth bit which would stay on the bus until the next AES block was ready.

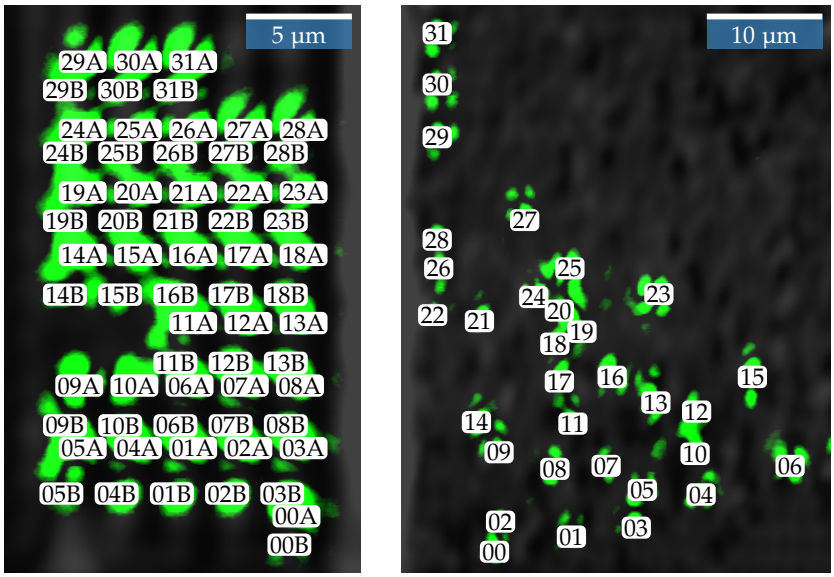
What remains to be explained is why the third 32-bit word in a 128-bit bitstream file block seemed to be "active" and not the fourth as could be expected. This is simply due to the fact that the bitstream data words preceding the ciphertext, see Fig. 4.18a, are not a multiple of four. If the word offset is determined using the actual ciphertext start as a reference, it is indeed every fourth data word that is "active" in LVI.

As this model thus seems to be correct and the AES output operation is now understood, the plaintext data can be extracted by probing the 32 bus lines, as soon as for each bus line the mapping of its physical position to its logical number is determined.

4.3.3.5 *Logical-to-Spatial Mapping*

With the AES output operation understood, the logical-to-spatial mapping of the 32 bus lines could be performed in a straightforward way: by toggling only one bit in the last 32-bit word of every 128-bit AES block and setting all other bits to "always zero". This would cause the data value of the bus line corresponding to that bit to be flipped for every AES block. As a result, an EOFM signal would be generated at $f_{CCLK}/256$ for only the bus line corresponding to that bit. All other bus lines would not generate any signal at all as they always receive "0" bits as data. To map all bus lines, this measurement would simply have to be repeated for all other bus lines.

For this purpose, 32 bitstreams were created, which each only generated an EOFM signal on one of the bus lines. Using these bitstreams, 32 measurements were then performed on the AES output port and the active position for each bus line was recorded. The corresponding EOFM measurements took approximately 2 hours. It should be noted that this time can be shorted by using a divide-and-conquer approach if more than one active bus line is used. For an



(a) Locations in the AES output. (b) Locations in the AES logic mesh.

Figure 4.27: Identified locations of each of the 32 bus lines of the AES plaintext bus. In the AES output, each bus line can be probed at two locations denoted as "A" and "B".

n-bit key, this would require $\log_2(n)$ measurements and possibly one additional reference measurement. However, for simplicity, the more straightforward approach with one active bus line was chosen here. Fig. 4.27a shows the identified positions overlayed onto an EOFM measurement were all bus lines were active.

As each bus line showed two areas of activity, these were denoted by a trailing "A" or "B". It is evident that most spots which appeared as one large spot before are actually composed of two smaller spots, compare also Fig. 4.26. A concern with this might be that the data of the two bus lines of such a spot might mix during probing and this might hinder data extraction. However, it should be noted that there is a large enough gap to the neighboring spots. This means that for data extraction the beam can simply be parked at the edge of such a composed spot, thus only extracting the data of a single bus line. The measurements performed in the next section actually revealed that there is no data mixing if this scheme is employed.

The mapping measurements also revealed that there are further locations for data extraction available inside the AES logic. Fig. 4.27b shows these potential probing locations for each bus line.

Therefore, even if extraction at the output port proved to be problematic, the data could be extracted from inside the logic mesh. Additionally, the measurements furthermore revealed potential probing locations in the main core.

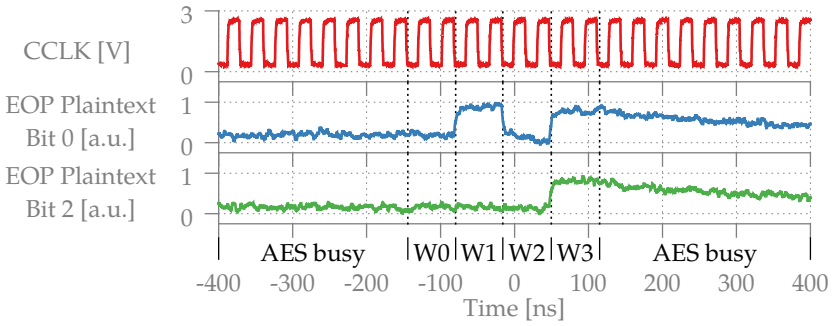


Figure 4.28: Exemplary EOP measurements demonstrating data extraction from two of the 32 plaintext bus lines. The configuration clock signal CCLK is also displayed. W0 to W3 denote the time frames of the four 32-bit words transferred on the bus during this measurement. [81]

4.3.3.6 *Data Extraction*

To prove the feasibility of data extraction, EOP measurements where performed on the plaintext output of the AES. For these measurements the "output port" locations, see Fig. 4.27a, were used. Comparison of the bitstream data to the measurement results showed that all bus lines carried the expected plaintext data. Furthermore, it was observed that the header commands, see Fig. 4.18a, are also transferred on this bus. This indicates that the complete plaintext bitstream is passed through this location and not only the configuration data of the "FDRI" block, compare Fig. 4.18a. Exemplary waveforms probed from two of the 32 bus lines can be seen in Fig. 4.28 along with the CCLK clock signal. This measurement was acquired with 5000 averagings. The expected behavior of three "fast bits" belonging to word W0 to W2 on the plaintext bus, followed by a fourth bit belonging to word W3 can be observed. As expected, the fourth bit stays on the bus line while the AES is busy decrypting the next block. The data on bus line 0 and 2 in this example is 0101 and 0001 respectively. The slight sagging of the waveform after the fourth bit can be explained by the bandpass characteristics of the EOP preamplifier. It should be noted that the externally available CCLK signal is always in phase with the bits extracted from the internal logic. This allows for straightforward synchronization of the EOP acquisition equipment. In this case, an "n-th edge" type trigger was used on the CCLK signal. The trigger was armed by the PROGRAM_B reconfiguration trigger signal. The complete plaintext bitstream could thus be extracted by probing all 32 bus lines in this fashion.

4.3.3.7 *Expenditure of Time*

As the aim of these sections is to assess the feasibility of an optical probing attack on the plaintext, the effort required by the attacker

Milestone	Powered On [h]	Usage Time [h]
Configuration Logic Localization	27.0	19.9
AES Logic Localization	9.1	8.0
Determining Input Bus Width	19.0	14.6
AES Output Localization	7.3	6.6
Successful Plaintext Probing	10.5	9.9
	Sum Powered On:	**Sum Usage:**
	72.9	58.9

Table 4.1: Time spent working on the failure analysis microscope during attack development. Usage time is the time that software was actively used on the PC of the setup. [81]

in terms of time and money is an important aspect. As already discussed in Sect. 4.3.1, the time spent on the failure analysis equipment during attack development was chosen as the metric for an approximation of this effort. This time was furthermore divided down into milestones. Tab. 4.1 shows the resulting times needed to reach each milestone up until the point where the plaintext gates were found and verified using EOP.

In this table, two metrics were used: time "powered on" and "usage time". "Powered on" time is the time the failure analysis microscope was switched on, including periods of standby. "Usage time" is the time that software was actively used on the PC of the setup. This includes actual measurements as well as copying files or looking at datasheets during the experiments. "Powered on" represents the time that an attacker would actually have had to pay for. "Usage time" represents a best case scenario that the attacker could have achieved had she worked as fast as possible. These times also include all overhead of unsuccessful measurements and unfruitful approaches taken during the experiments.

It can be seen that the time for configuration logic localization and input bus width determination required a larger amount of time than the other milestones. For the "configuration logic localization" step this can simply be explained by the fact that it contains the overhead of the initial setup and getting used to the FPGA board and other equipment. The explanation for the longer "input bus width" milestone is that a considerable amount of time was spent on searching for serial data gates. This was done as serial plaintext data would have simplified data extraction, as opposed to the 32-bit wide plaintext bus which was used in the end. For a data extraction attack, the knowledge of the logical to spatial mapping of Fig. 4.27a is also

needed, for which 2 hours of measurement time need to be added to the "powered on" sum of Tab. 4.1. The total rent time for the development of this attack would thus be 74.9 hours. The rate for the equipment used in this work is about $300/hour, including the operator. The total cost of the attack development would thus be $22,500. It should be stressed that this is only required once because as soon as the attacker has found the plaintext gates in one device, she can directly launch the data extraction on any chip of the same type.

4.4 CHAPTER CONCLUSION

This chapter has evaluated likely attack paths using optical probing on advanced key storage solutions using physically unclonable functions (PUFs), as well as direct plaintext extraction, which can circumvent any secure key storage scheme. First, the general principles of optical contactless probing for waveform data as well as for activity maps were briefly introduced.

Then, for the first evaluation, methods for secure key storage using PUFs were discussed and a proof-of-concept (POC) implementation using a PUF key storage concept by Xilinx was presented. Using this POC implementation, likely attack scenarios were discussed, a suitable setup developed, and successful key extraction from a serial as well as a parallel implementation was demonstrated on a 60 nm Altera Cyclone IV device. Furthermore, characterization of the RO PUF used in the POC implementation was performed. Physical interactions possibly useful for countermeasures, such as the frequency shift occurring in ring oscillators during probing, were identified and will later be discussed in more detail in Chapter 6.

In the second part, direct probing of decrypted plaintext data was evaluated as a likely attack path when the key cannot be extracted. For this, an attack scenario was presented in which an attacker can introduce fundamental frequencies into only the plaintext data gates, allowing her to localize them and extract the data. By developing a suitable setup and performing the discussed attack on the commercial decryption ASIC of a 28 nm Xilinx Kintex 7 FPGA in a flip-chip package, it could be shown that the reverse engineering needed for the attack can be performed completely noninvasively in less than 10 working days.

The risk assessment of optical contactless probing in this context has thus shown attacks to be well feasible. It should also be stressed again, that the attack development itself needs to be performed only once for a certain type of chip, as it immediately breaks the security of a whole line of devices. Subsequent data extraction from known locations can be expected to require much less effort. Furthermore, similar weaknesses are expected for other flip-chip type packages and

implementations on other devices. Suitable approaches to mitigate or prevent these kinds of attacks will later be discussed in Chapter 6.

5 RESOLUTION ADVANCEMENTS

This chapter will take a slightly different approach than the previous chapters. Instead of dealing with a specific type of attack, it will discuss a prerequisite for attacks as well as for failure analysis (FA) techniques: a suitable optical resolution. As chip technology sizes continue to shrink, optical resolution can become a hindrance when debugging or attacking the very latest technology devices. However, as will be discussed later, several techniques can be employed to overcome this limitation.

This chapter will assess which techniques can be employed by attackers to extend their reach to smaller technology devices. On the one hand, this will be evaluated by utilizing cutting-edge FA techniques in a prototype setup. Although these techniques constitute current research, in the near future machines utilizing them will be available to FA labs and thus also to attackers. On the other hand, low-cost home-built approaches to these techniques will also be evaluated, which are available to attackers in any case.

For the FA prototype setup, a cooperation project was pursued, where FinFET test devices were supplied by Qualcomm Inc., and Varioscale Inc. provided device preparation. The research on low-cost approaches was conducted in cooperation with the University of Applied Sciences Jena (EAH Jena). Some of the results and figures in this chapter were already published in [11, 39, 43].

5.1 HIGH-RESOLUTION TECHNIQUES

As the trend of shrinking technology size in the semiconductor industry continues, it leads to implications for failure analysis (FA) as well as for IC security. As current chip technologies become more and more challenging for FA analysis [25], these same challenges can to some extent protect security-sensitive ICs. As the pitch of structures shrinks, see Tab. 5.1, it becomes harder and harder to distinguish circuit elements given a specific optical resolution. This hinders analysis in both FA and attack circumstances. In FA this has led to the demand for high-resolution techniques to overcome the current resolution limit.

$$d = 0.51\frac{\lambda}{NA} = 0.51\frac{\lambda}{n\sin(\alpha)} \tag{5.1}$$

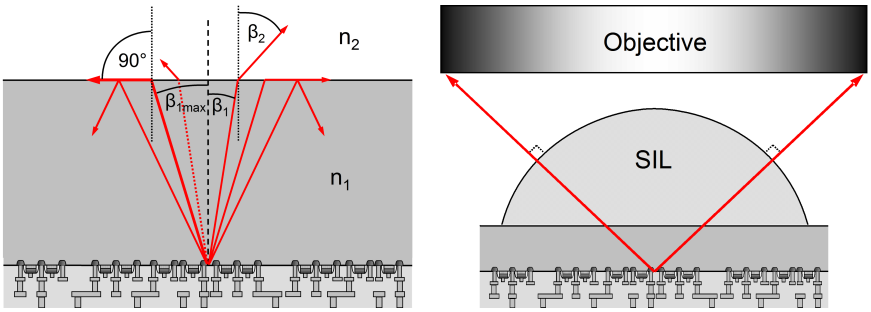
Eq. 5.1 shows an expression for the expected resolution limit of a microscope system, as defined by [17], Eq. 3.57. This expression assumes Sparrow's two-point resolution limit and delivers the same res-

Production Year	Node Label [nm]	Logic Pitch [nm]
2013	"16/14"	80
2015	"10"	64
2017	"7"	50
2019	"5"	40
2021	"3.5"	32

Table 5.1: Logic pitch sizes according to the 2013 international technology roadmap for semiconductors. [31]

olution for standard optical microscopes and confocal scanning optical microscopes, although confocal microscopes will deliver a sharper edge and higher contrast [17]. It should be noted that there are other criteria such as the Rayleigh criterion which will deliver different resolution limits. Furthermore, the resolution limit will also depend on the assumed type of object (e.g. point, line, edge or fluorescent, non-fluorescent, etc.). Depending on the exact theoretical treatment, the derived resolution limit will also differ between conventional and confocal scanning microscopes and also depend on instrument parameters such as the pinhole size. Yet, all these approaches finally arrive at expressions which are proportional to λ/NA, and for *comparison* of achievable resolution, the choice of the resolution limit formula is somewhat arbitrary. Thus, in this chapter, the expression of Eq. 5.1 is selected for resolution limit estimation.

In Eq. 5.1, it can be seen that the resolution depends on the wavelength λ and the objective's numerical aperture NA. The NA itself depends on the surrounding medium's refractive index n and half the lens's angular aperture α. This equation demonstrates that there are two ways for improving resolution: one via λ and the other via NA. Because of the need for transparency in silicon, in conventional FA systems λ is traditionally fixed to infrared (IR) wavelengths, with 1.06 µm and 1.3 µm being the most popular ones. Therefore, resolution increase in this area has focused on increasing the NA. It can be seen that in air ($n \approx 1$) the NA can reach a maximum of 1, even with a lens which captures full 180 degrees of the light leaving the sample. To still be able to increase the NA, liquid immersion lenses have been introduced which fill the space between sample and lens with a high refractive index oil matching the index of the sample. However, for the very high refractive index of silicon (3.5 in IR) no matching liquid is available. To overcome such limitations *solid* immersion lenses (SILs) [45] have been introduced. Fig. 5.1 shows a comparison of a



(a) Without SIL (b) With SIL

Figure 5.1: Comparison of optical imaging in silicon with and without a solid immersion lens (SIL). The sketched SIL is of the hemispheric type and can be seen to increase the effective maximum aperture angle in the silicon. © 2015 Institute of NANO Testing [11]

silicon sample with and without a silicon SIL applied. In Fig. 5.1a it can be seen that without a SIL, refraction occurs at the silicon-air interface. If we trace a ray originating from the area of interest under angle β_1, we can see that it will leave the sample under angle $\beta_2 = \arcsin(n_1/n_2 \cdot \sin(\beta_1))$, according to the law of refraction [35]. As silicon is optically denser than air, it follows that $n_1 > n_2$ and thus $\beta_2 > \beta_1$. Consequently, even if we assume a perfect objective lens which captures all exiting light rays with an aperture half-angle of $\alpha = 90°$, the effective angle in silicon will be β_{1max}, see Fig. 5.1a. All rays under a larger angle will be subject to total internal reflection and not be able to leave the silicon. The same discussion holds for the illuminating rays originating from the microscope lens.

However, if a silicon SIL is placed on the sample, it follows that $n_1 = n_2$ and thus no refraction at the sample-SIL interface can occur, see Fig. 5.1b. If the SIL is placed in such a way that the object is at the center of the SIL sphere, the light rays will leave the SIL perpendicular to its surface, which also leads to no refraction at the SIL-air interface. It follows that with such a setup $\beta_2 = \beta_1$ and consequently $\beta_{1max} = \alpha_{objective}$. As a consequence, the SIL will effectively act like an immersion medium with the refractive index of silicon and thus allow for an NA of up to 3.5 in IR.

The discussed SIL type is called a hemispheric SIL, as it places the object at the center of the SIL sphere. An alternative approach is the aplanatic SIL. This type uses non-perpendicular incidence on the air-SIL interface to transform the angles of the rays incoming from the objective lens to larger angles inside the SIL so that $\beta_{1max} > \alpha_{objective}$. This eases the requirements on the objective lens aperture half-angle α. However, both hemispheric and aplanatic types have a maximum NA limit of n_{SIL}, compare Eq. 5.1.

It should also be noted that a heterogeneous SIL setup is possible where $n_{SIL} \approx n_{Sample}$. In this case, one can derive from the law of refraction that the maximum NA will be limited by the smaller index of refraction. As a consequence, the indices of refraction of the SIL and the sample should not differ too much, so that potential resolution is not wasted.

SIL schemes such as these can already deliver resolutions in the 100 nm to 200 nm range in IR with commercial silicon SILs that are approaching the theoretical limit with an NA of 3.1 to 3.3 [26, 72]. This is already an impressive achievement, yet, comparison with Tab. 5.1 shows that even higher resolutions would be desirable. As can be seen from Eq. 5.1, the only way to increase resolution when the NA cannot be improved further is the wavelength λ. However, this is connected to some challenges. First and foremost, silicon shows increased absorption for shorter wavelengths, which is problematic for backside penetration and transmission through the SIL. Yet, if λ could be moved into the visible light (VIS) spectrum, resolution capabilities would increase by a factor of two or better. In fact, in [7, 8] it was demonstrated that visible light imaging and probing is in principle possible using ultra-thin silicon-on-insulator and bulk silicon devices. These experiments used remaining silicon thicknesses of zero and around one micrometer respectively, which were achieved by specialized device thinning procedures. However, only *liquid* immersion lenses with an NA of 1.4 were applied. It thus remains unclear if a system using high-NA *solid* immersion lenses for visible light is possible. If visible light could be combined with a compatible SIL, this would constitute an improvement for FA techniques and a threat to security ICs at the same time. To evaluate these possibilities, the following sections will analyze the challenges connected to optical imaging and laser voltage probing (LVP) as well as laser voltage imaging (LVI) techniques in the VIS regime in combination with a SIL and assess their feasibility.

5.2 PROTOTYPE SYSTEM FOR VISIBLE LIGHT SOLID IMMERSION LENS PROBING

This section will evaluate the feasibility of optical probing using shorter wavelengths by discussing, designing, building, and testing a prototype system using visible light (VIS) in combination with a solid immersion lens.

5.2.1 *Visible Light Absorption in Silicon*

One reason why visible light has so far not been employed in failure analysis (FA) is the increase of the absorption in silicon when using shorter wavelengths. The intensity of light traveling through silicon

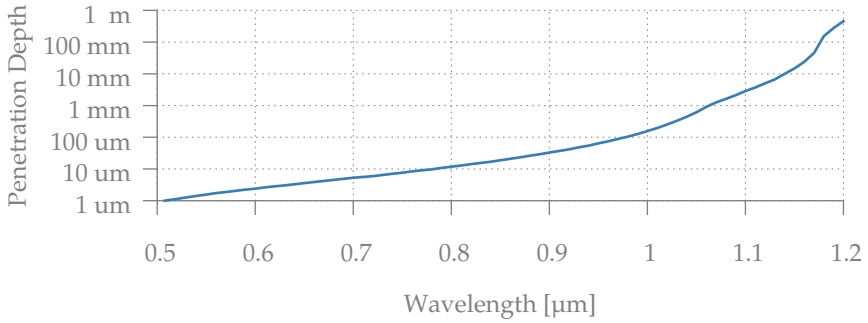


Figure 5.2: Penetration depth in intrinsic silicon at 300 K based on data published in [24].

can be described by the Beer-Lambert law which is shown in Eq. 5.2 [35].

$$I(z) \quad = \quad I_0 \cdot e^{-\alpha_{abs} z} \tag{5.2}$$

Here $I(z)$ is the intensity at depth z, I_0 is the intensity at zero depth (silicon boundary) and α_{abs} is the material-specific absorption coefficient at a given wavelength. From this, it can be seen that the light intensity will decline exponentially, with the speed of the decline being governed by α_{abs}. A parameter derived from α_{abs} is the penetration depth which is simply the inverse of α_{abs}, see Eq. 5.3.

$$\delta \quad = \quad \frac{1}{\alpha_{abs}} \tag{5.3}$$

At depth δ, the light intensity has decayed to $I_0 \cdot e^{-1}$ or about 37% of its original intensity. To give an overview of the absorption changes to be expected when transitioning from infrared into the visible light region, Fig. 5.2 shows a plot of δ versus the wavelength in intrinsic silicon.

It can be seen that while at the common 1.06 µm laser wavelength the penetration depth is 901 µm, at 0.65 µm wavelength it is only 3.6 µm. This means that in order for optical FA techniques to be feasible in the visible light spectrum, the silicon has to be thinned much more aggressively. It should also be kept in mind that for techniques based on reflection, like imaging and optical probing, the light actually has to travel through the remaining silicon thickness twice. If we assume a moderate visible light wavelength of 650 nm and a needed minimum return intensity of 1%, we can calculate using Eq. 5.2 that the remaining silicon thickness has to be less than 8.3 µm. This is a challenging thickness, as it increases the risk of device damage by cracking or overpolishing. Nevertheless, it is still a thickness that is feasible using a suitable parameter window for the polishing machine. If the expected resolution improvement discussed in Sect. 5.1 is

Wavelength [nm]	SIL Material	Refractive Index	Resolution Limit [nm]	Si Penetration Depth [µm] [24]
1064	Si	3.55 [24]	153	901
650	GaP	3.29 [50]	101	3.58
550	GaP	3.44 [50]	82	1.56
440	SiC	2.73 [87]	82	0.326
330	C	2.50 [59]	67	0.008

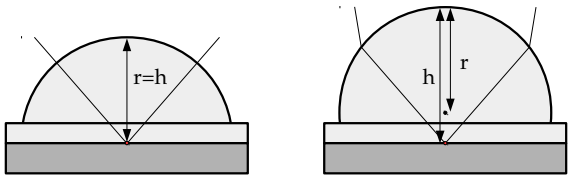
Table 5.2: Suitable SIL materials for different wavelengths, along with their refractive index, expected resolution limit, and light penetration depth in intrinsic silicon at 300 K. Table based on [43], recalculated with more recent material constants.

taken into account, the increased effort for device preparation might be worth it.

5.2.2 *Visible Light Solid Immersion Lens Design*

This section will discuss the design decisions leading to the geometric shape and material selection for a SIL for visible light. The illumination wavelength is an additional parameter which is considered. As could be seen from the previous section, see Fig. 5.2, the absorption of light in a thin remaining silicon layer of the device is already challenging. It would thus be advantageous to select a material for the SIL which does not heavily absorb the employed light wavelength.

Tab. 5.2 gives an overview of suitable wavelength and SIL material combinations which fulfill this requirement for state-of-the-art IR SIL imaging as well as for possible visible light wavelengths. Also given is the refractive index of the SIL material and the expected resolution limit calculated with Eq. 5.1. Furthermore, the penetration depth in the remaining silicon layer of the device is given. From this, it is immediately clear that although the illumination wavelengths of 440 nm and 330 nm will deliver the best resolution, they will require a very challenging remaining silicon thickness in the nanometer range. Apart from this, altering device behavior can also be expected at these small thicknesses even if the preparation is successful [69]. As a consequence, these wavelengths are discarded for first proof-of-concept experiments. This already leaves only one SIL material for consideration: gallium phosphide (GaP). The only remaining design decision would then be the illumination wavelength. However, as a GaP SIL can be used with both 650 nm and 550 nm wavelengths, this decision can be postponed for now. Yet, it should be noted that for first tests



(a) Hemispheric design (b) Aplanatic design

Figure 5.3: Geometry of hemispheric and aplanatic SIL designs for the homogeneous case (silicon-on-silicon).

a 650 nm wavelength is advantageous, as it eases the constraints on device thickness.

With a SIL material selected, attention can now be brought to the shape of the SIL. In general, there are two types of SIL to consider: hemispheric and aplanatic, as already mentioned in Sect. 5.1. Fig. 5.3 shows a comparison of these two designs. It can be seen that while the hemispheric SIL has its center at the focal plane, the aplanatic SIL has its center above it. As discussed in Sect. 5.1, the aplanatic design is able to achieve better resolutions assuming a specific NA of the backing objective. Calculations in [36] show for example that for a 500 µm radius GaP SIL with the same backing objective the hemispheric SIL achieves an effective NA of 0.65, while the aplanatic type achieves an NA of 2.15. However, the aplanatic design is also much harder to manufacture, as it is more sensitive to errors in the SIL height h. Taking the same SIL example from [36], the height tolerance window is 447 µm for the hemispheric SIL and only 0.6 µm for the aplanatic SIL. Thus, the tolerances for an aplanatic design are more than two orders of magnitude tighter and realization of them does not seem feasible for a first proof-of-concept system. Accordingly, the hemispheric SIL concept is selected for the further design steps.

As a hemispheric SIL for homogeneous material has the condition of $r = h$, we can for the heterogeneous case of a GaP SIL on thin silicon samples still assume that $r \approx h$ and correct the exact SIL height using the law of refraction later. This leaves the design choices with one parameter: r. It is evident that this parameter has to be smaller than the working distance of the backing objective lens as well as the available GaP substrate material thickness. For the available VIS objective lenses with high resolution, the working distance is in general smaller than 1 mm. During the design phase, a GaP substrate with a thickness of 300 µm was obtained which was well suited for low working distance lenses. However, if r is too small, this will lead to a limited field of view (FOV). The expected FOV was thus determined using Eq. 12 of [5] using a 300 µm SIL radius which delivered a FOV diameter of 7.2 µm for VIS imaging assuming 650 nm illumination. This FOV should be sufficient for analysis of small technology size devices with high magnifications.

The next design step is the angular aperture of the SIL. For a hemispheric SIL, it is sufficient if the angular aperture of the SIL is the same as the angular aperture of the backing objective lens. As the highest-resolving backing objective available for the setup has an NA of 0.85, the full angular aperture is 116.4 degrees. A full angular aperture of 120 degrees for the SIL should thus suffice, which would result in a design NA of up to 2.9 (assuming 650 nm illumination).

To allow for the remaining thickness of the DUT, the SIL shape has to be truncated at the bottom, see also Fig. 5.3a. In the homogeneous case, the truncation simply equals the DUT thickness. For the heterogeneous case of GaP-on-silicon, this has to be corrected for the refraction at the GaP-silicon interface. Using the law of refraction and assuming a remaining silicon DUT thickness of 10 µm, as well as an objective NA of 0.85, the resulting SIL truncation can be found to be 7 µm. The geometrical design parameters are thus defined and the fabrication can now be considered.

5.2.3 *Solid Immersion Lens Fabrication*

Conventional lens fabrication uses different grinding and polishing steps to achieve the desired lens geometry. However, these were not readily available at Technische Universität Berlin (TUB) and, furthermore, the GaP wafers which were used as the source material were unsuitable for this type of fabrication process. As an alternative, lens fabrication by high-precision turning was selected. In this case, the lens shape is produced with a lathe chisel which removes material from a GaP workpiece rotating about an axis. To achieve a smooth surface when machining the brittle material, a suitable set of process parameters must be selected. This fabrication process was carried out at the department of micro- and precision devices (MFG) at TUB using the design parameters discussed in Sect. 5.2.2. The machine used was a Moore Nanotech FG350. Additional details about the lens fabrication process and cutting parameters are disclosed in [43, 86]. Fig. 5.4 shows a picture of the SIL produced thus.

White light interferometry measurements demonstrated that the SIL surface quality was already satisfactory after the turning process [86] and consequently no additional polishing steps were performed. This results in a relatively low cost for the fabrication of the SIL. An assessment in [43] estimates the material and machine costs to be $3400 per SIL with the current, non-optimized manufacturing process.

To verify the as-manufactured dimensions of the SIL, it was analyzed using a "Dektak" stylus profiler. The results of these measurements are presented in Fig. 5.5. The SIL surface was measured to be 9.5 µm lower than the substrate surface, which is a 2.5 µm deviation from the 7 µm target. As the substrate is 300 µm thick, this would put the truncated SIL height at 290.5 µm. The surface shows good agree-

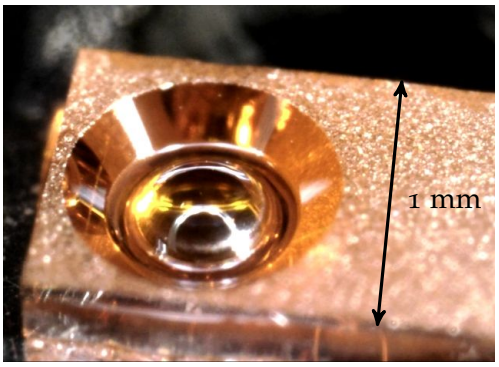
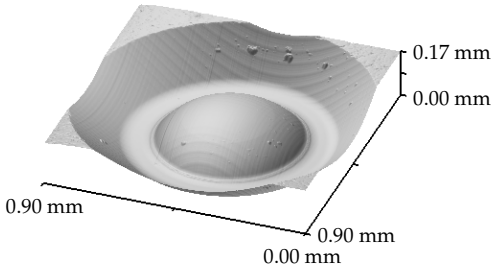
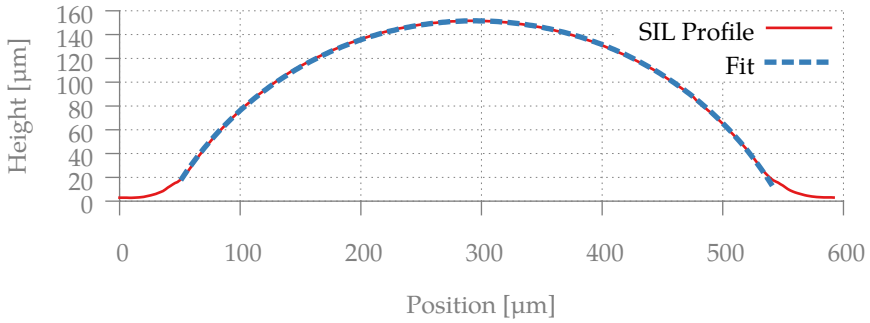


Figure 5.4: Photograph of the gallium phosphide SIL produced by high-precision turning.



(a) Surface plot



(b) Surface line plot and circular curve fit

Figure 5.5: Measurement results of a "Dektak" stylus profiler analysis of the manufactured GaP SIL.

ment with the designed shape in general but also a measured radius of 289 µm versus a 300 µm design value. It should be mentioned that there was drift on the SIL holder during the measurements and the data had to be corrected accordingly. This was done by performing the radius measurements in orientations not affected by the drift direction. Assuming that this correction did not influence the results, the combination of measured radius and truncated SIL height would put the height error of the produced SIL at 1.5 µm minimum, even for a 0 µm thick silicon device. To this minimum error, the effect of the actual DUT thickness needs to be added. According to Eq. 7 in [5], the allowable height error at 650 nm illumination with the measured SIL radius is $\pm$7.1 µm under the assumption of a quarter-wavelength aberration limit. This would result in a maximum DUT thickness of about 5.6 µm for the homogeneous case (GaP SIL on GaP DUT). As the silicon DUT has a different index of refraction, this value needs to be corrected using the law of refraction. Assuming 58.2° incident rays (0.85NA), the correction factor is 1.52 and the allowable silicon DUT thickness thus 3.7 µm. If the DUT thickness exceeds this value, imaging should still be possible, although with decreasing image quality.

Thus, even though the SIL prototype will always produce a minimum error, which will be worsened by thicker DUTs, it is deemed fit for first experiments. However, it should be kept in mind that it will not be able to deliver the best imaging quality theoretically possible and improvement can be expected with an adjusted manufacturing process.

5.2.4 *Prototype System Hardware Setup*

To allow for testing of the visible light SIL described in Sect. 5.2.3, two systems were used. The first system was set up for direct comparison of infrared and visible wavelength *optical imaging*. The second system was designed to evaluate *probing* techniques, namely laser voltage probing (LVP) and laser voltage imaging (LVI). For details regarding the basics of LVP and LVI techniques see Sect. 4.1. Both setups are based on modified Zeiss "LSM 21/31" confocal laser scanning microscopes (LSMs), see Fig. 5.6. Both microscopes contain an objective turret for installation of different objective lenses, X/Y galvanometric scanning mirrors, as well as a confocal telescope and pinhole, and a control PC.

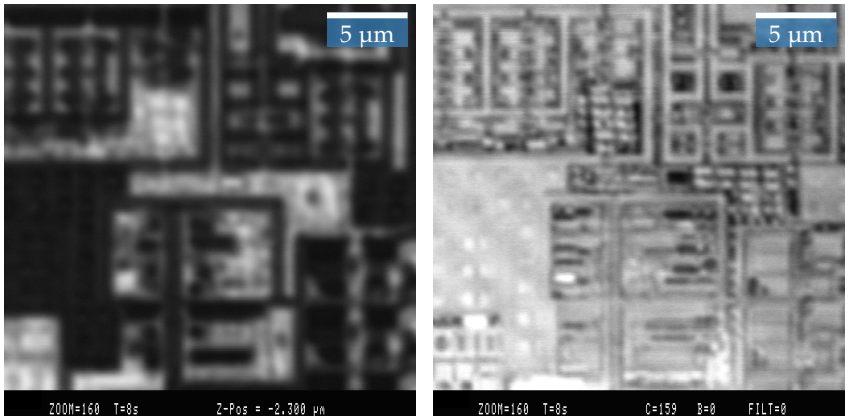
The microscope for optical imaging was already equipped with a 633 nm helium-neon (HeNe) laser with 15 mW power and had a 1 mW/1.15 µm helium-neon laser added. For detection of the reflected light, the microscope already had a photomultiplier tube (PMT) for visible light and a germanium photodiode for IR wavelengths installed. The imaging experiments on this system could thus be conducted with the original Zeiss software of the microscope.

Figure 5.6: Photograph of the laser scanning microscope used as the base for the prototype system. [93]

The system for visible light optical probing had its original HeNe laser source replaced with a 660 nm diode laser. Additionally, an external silicon photodiode was added as a detector. The photodiode is connected to a current-to-voltage preamplifier (Femto DHPCA-100). The output signal of the amplifier can then either be fed into a LeCroy WavePro 735 Zi oscilloscope or a Stanford Research Systems SR844RF lock-in amplifier. If using the oscilloscope, the setup can acquire LVP waveforms by averaging multiple probing measurements while the device is run in a loop and the beam is held stationary. For LVI measurements, the lock-in amplifier is used with a reference signal at the frequency of interest while the beam is scanned. The lock-in will then output the in-phase and quadrature components of signals detected at this frequency. In other words, the lock-in is used as a very narrow frequency filter which additionally delivers phase information about the detected signal. To lower the background noise in the detector signal due to sample vibrations, the setup was additionally fitted with an air-cushioned vibration-insulating table. The resulting full optical probing setup can be seen in Fig. 5.7.

However, as the settling time for LVI measurements is in the order of milliseconds per pixel, the acquisition of such measurements was not possible with the original Zeiss software because of a too high minimum scan speed. To circumvent this, a custom software was developed in the graphical programming system "LabVIEW" by National Instruments. The software enables the slow scan speeds needed for LVI measurements by controlling the laser scan mirrors of the setup manually via the LSM's GPIB bus while sampling the resulting values from the lock-in and assembling them into 2D measurement data. Additionally, the software samples an auxiliary voltage input, which can be used in conjunction with suitable measurement equipment to acquire the values of other physical parameters in dependence of the beam position.

Figure 5.7: Photograph of the visible light probing prototype setup.



(a) IR (1.15 µm) (b) VIS (0.633 µm)

Figure 5.8: Comparison of IR and VIS backside imaging on a 10.2 µm thick, 60 nm technology Altera Cyclone IV FPGA, acquired with a 50x/0.85NA lens. © 2015 Institute of NANO Testing [11]

5.2.5 *Visible Light SIL Imaging Results*

To gain experience with the optical imaging setup, experiments were first performed using readily available commercial ICs without the SIL. Altera Cyclone IV FPGAs, which are manufactured in 60 nm technology, were selected for this task and thinned at Technische Universität Berlin (TUB) to 10.2 µm. Fig. 5.8 shows a test measurement on this sample comparing IR and VIS imaging, which shows the system to function as expected. Even without a SIL, the advantage of VIS imaging is already evident, as there is a drastic increase in image detail. Additionally, a contrast inversion can be observed. This is probably due to layer interference effects behaving differently because of the halved illumination wavelength.

As VIS imaging was thus proven to be feasible in general, the experiments were pursued using the GaP SIL on 16/14 nm FinFET tech-

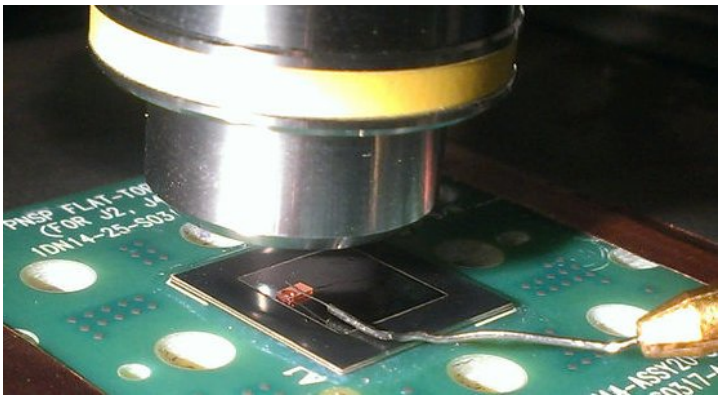


Figure 5.9: Positioning of the GaP solid immersion lens on the DUT using a prober needle and a manipulator.

nology devices. These FinFET devices were provided by Qualcomm Inc. and were thinned to a thickness of about 3 µm by Varioscale Inc. for the following experiments. The SIL was then placed on these samples and manually moved using a prober needle and a manipulator. After positioning, very slight pressure was applied to the SIL using the needle, see Fig. 5.9.

Fig. 5.10 shows a full-frame comparison of the resulting imaging performed in IR and VIS with and without the SIL, while Fig. 5.11 shows detailed views of the same data. A 20x/0.5NA lens was used for this measurement and the changed magnification resulting from SIL use was compensated by adjusting the laser scanner zoom factor. It can be seen that when switching from IR to VIS, some locations again exhibit a contrast inversion as in the previous non-SIL experiments. More importantly, an increase in resolution can also be observed again. This is the case with or without the SIL being used, as can be expected from Eq. 5.1 because of the λ component. The addition of the SIL causes a further increase in resolution, although only a smaller field of view (FOV) can be achieved, see Fig. 5.10. This was expected and already discussed in Sect. 5.2.2. Using the measured SIL radius and the actual setup wavelengths, using [5], Eq. 12 results in an expected usable FOV diameter of 6.89 µm for VIS and 10.2 µm for IR imaging. It can be seen that the *visible* FOV is actually larger in the experiments, however, the approximation in [5] aims at determining a mostly *aberration-free* FOV, and also does so for microscope lenses with an NA close to 1. Yet, as the image quality starts to worsen outside of the calculated diameters, the results seem to be consistent with this FOV approximation. As already discussed in Sect. 5.2.2, the small radius of the SIL which leads to the comparably small FOV is a requirement for the compatibility of the SIL with the small working distances of the high NA lenses of the microscope setup. Also apparent when using the SIL are interference rings originating from the center, compare Fig. 5.11. Later analysis of the SIL revealed that

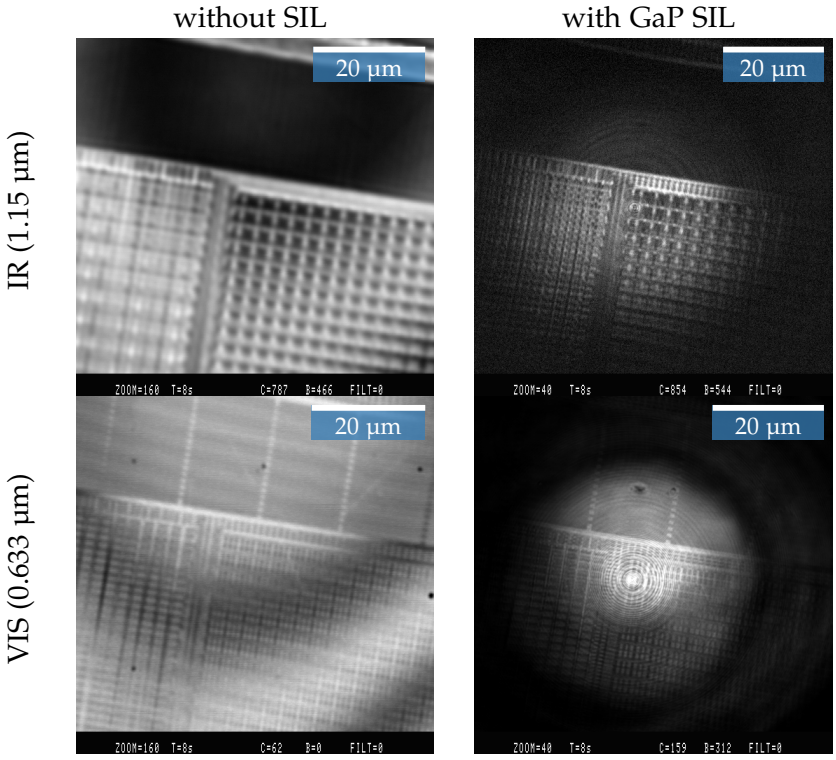


Figure 5.10: Reflected light images with and without the GaP SIL in IR (1150 nm) and VIS (633 nm). Reprinted with permission of ASM International. All rights reserved. [43]

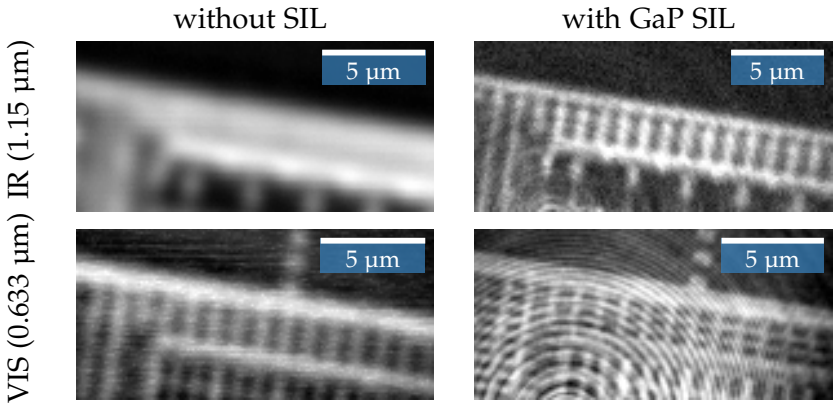


Figure 5.11: Zoomed-in image portions from Fig. 5.10 for optical performance comparison. Reprinted with permission of ASM International. All rights reserved. [43]

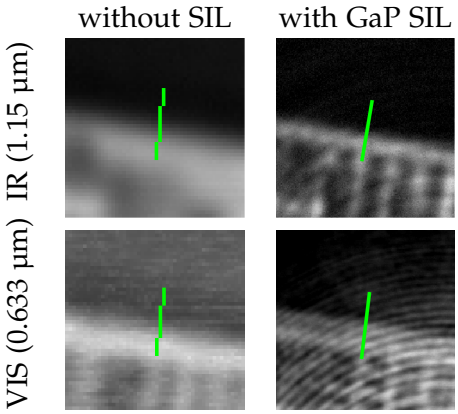


Figure 5.12: Line plot positions for the quantification of resolution improvement. A 20x/0.5NA lens has been used for confocal image acquisition in IR (1150 nm) and VIS (633 nm). The resulting line plots are displayed in Fig. 5.13. Reprinted with permission of ASM International. All rights reserved. [43]

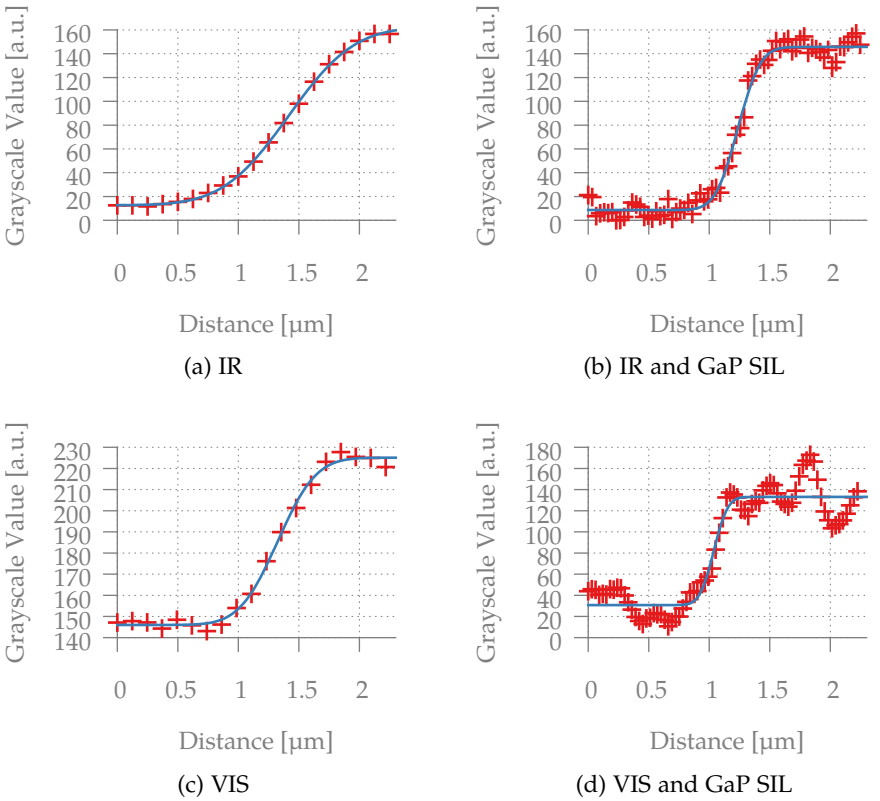
these are caused by a small patch of rough SIL surface directly at the center, which scatters the incoming light. This patch is likely caused by a too low cutting speed at this location during SIL manufacturing. For a smooth SIL surface, a certain process window must be followed for the cutting speed. However, in the middle of the SIL, the cutting radius tends to zero, which in theory would require the rotational frequency of the lathe to tend to infinity. As this is physically impossible, non-ideal cutting parameters result and are a likely cause for the small rough patch.

Apart from these constraints, the SIL delivers satisfactory image quality and a very noticeable increase in resolution, especially with VIS illumination. Unfortunately, the test devices had no dedicated structure for resolution determination. However, layout analysis reveals the bright horizontal lines appearing only in the VIS+SIL images of Fig. 5.11 to be 263 nm in height with a vertical pitch of 676 nm.

To be able to numerically compare the achieved resolution in the individual experiments, line plots were extracted from dark-bright transitions. The locations of these line plots can be seen in Fig. 5.12. To extract parameters related to resolution, an approach based on [32] was followed. For this, the data of said line plots was fitted using Eq. 5.4.

$$I(x) \quad = \quad I_0 + A\frac{erf(s(x - X_0)) + 1}{2} \tag{5.4}$$

This function expresses the line plot grayscale value $I(x)$ with the following parameters: I_0 is the grayscale level of the dark edge, A is the grayscale step height of the transition, X_0 is the position of the edge transition and s is the scaling factor of the function. Here, the parameter s is directly connected to resolution improvement. The



(a) IR

(b) IR and GaP SIL

(c) VIS

(d) VIS and GaP SIL

Figure 5.13: Fits using Eq. 5.4 applied to the dark-bright transitions shown in Fig. 5.12. Reprinted with permission of ASM International. All rights reserved. [43]

extracted line plot data, as well as the fitted function, are presented in Fig. 5.13.

This figure reveals the same resolution improvements discussed before. An increasing steepness of the dark-bright transition edge can be observed when progressively applying VIS imaging as well as the SIL. The fitting function can furthermore be seen to match the data well. When using the SIL in IR and VIS, some discrepancies at the plateaus are visible, most noticeably for VIS+SIL. This can be explained by the interference rings already discussed previously, compare also Fig. 5.12. For performance evaluation of the SIL, the s parameter of different measurements is compared in Tab. 5.3. The detailed numerical values on which this comparison is based can be found in Appendix A.2.

These results indicate that the SIL provides an improvement by a factor of 2.9 when used in IR, which is less than 7% short of the ideal value of 3.1. The improvement factor in VIS is 2.7, which is an 18% derivation from the theoretical limit of 3.3. This is most likely caused by the properties of the GaP SIL prototype, such as the already discussed manufacturing errors, see Sect. 5.2.3. Furthermore, the relatively large footprint of the SIL (entire bottom of the GaP car-

Experimental Improvement			Theoretical Improvement	Experiment vs. Theory
s_1	s_2	s_2/s_1		
IR	IR+SIL	2.9	$3.1 = n_{GaP(IR)}$	93.5%
VIS	VIS+SIL	2.7	$3.3 = n_{GaP(VIS)}$	81.8%
IR+SIL	VIS+SIL	1.6	$1.9 = \frac{n_{GaP(VIS)}\lambda_{IR}}{n_{GaP(IR)}\lambda_{VIS}}$	84.2%

Table 5.3: Comparison of optical improvement as represented by the steepness parameter s for different wavelengths with and without the GaP SIL. The data was extracted from fits of the edge line plots of Fig. 5.13 using Eq. 5.4. The theoretically expected values and to which extent they were achieved is also shown.

rier) requires a large planar DUT surface for effective optical coupling, which is difficult to achieve. Additionally, the very limited pressure that was applied during imaging, see Fig. 5.9, can also lead to the air gap between GaP SIL and Si DUT not being fully eliminated. As the allowable gap is wavelength-dependent [5], the resolution deterioration will be more noticeable in VIS. This can explain the worse performance of the SIL under VIS illumination.

The final comparison to be made between IR+SIL versus VIS+SIL is now a mix of the SIL performance in VIS and IR. Therefore, it is not surprising that it lies in between the two previously discussed cases. To be precise, the experiment is about 16% short of the theoretical limit with an improvement of 1.6 versus 1.9.

Although the prototype system does not reach the theoretical performance limit, it nevertheless improves resolution by 190% and 170% in IR and VIS respectively when compared to the non-SIL case. When comparing VIS and IR illumination for SIL systems, an improvement by 60% is observed. As a side note, it should also be mentioned that the backside resolution of the LSM system has increased by 369% in total. This can be seen when comparing the original system performance in IR without a SIL with the now possible measurements in VIS with the SIL, for details see Tab. A.2 in the appendix.

5.2.6 *Visible Light SIL Probing Results*

As the previous experiments had demonstrated the achieved increase in resolution, further experiments were conducted to demonstrate the visible light optical probing capabilities of the prototype setup. For this, an n-channel FinFET transistor test structure with a size of 1.1 µm by 1.4 µm was used. The structure was provided by Qualcomm Inc. and was thinned by Varioscale Inc. to an average remain-

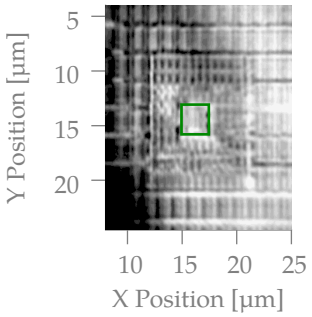


Figure 5.14: Reflected light image of the n-channel FinFET test structure acquired with a 10x/0.3NA objective lens in combination with the GaP SIL. The highlighted area denotes the position of the VIS-LVI measurements shown in Fig. 5.16. Reprinted with permission of ASM International. All rights reserved. [43]

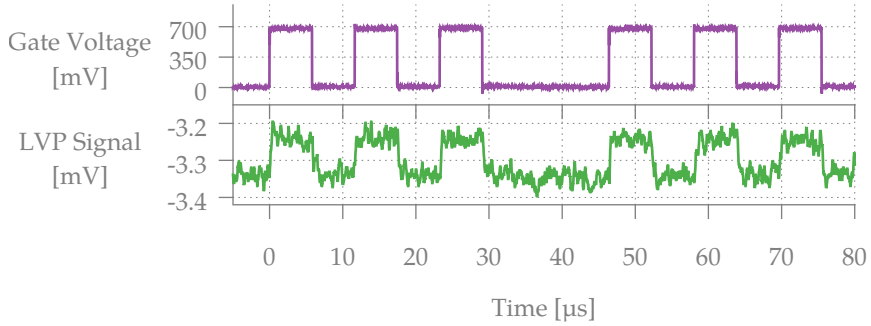


Figure 5.15: VIS-LVP probing results from the center of the n-channel FinFET transistor acquired with a 63x/0.75NA objective lens. Top: applied "missing-pulse" gate signal. Bottom: VIS-LVP signal resulting from an averaging of 10^6 waveforms. Reprinted with permission of ASM International. All rights reserved. [43]

ing silicon thickness of 3.3 µm. An overview image with the test structure area highlighted can be seen in Fig. 5.14. For the operating conditions of the device, values within the nominal range were used. More specifically, the substrate was connected to 0 V and the gate pulsed with a 0 V low, 0.7 V high square 86 kHz waveform. Source and drain were shorted together and pulled to substrate level (0 V) by a Stanford Research Systems "SR570" current preamplifier. This allows for monitoring of the photocurrent injected into source and drain during the experiment via the auxiliary input of the setup, see Sect. 5.2.4. The amplification of the photodiode preamplifier was set to 10^6 V/A and 1 MHz bandwidth AC mode.

As a first test, LVP measurements were performed, which probed the central transistor area. For this, a 63x/0.75NA objective lens was used without applying the GaP SIL. To drive the gate, a "missing pulse" waveform was applied, see Fig. 5.15, top. The resulting LVP waveform, acquired by averaging 10^6 loops, can be seen at the bottom

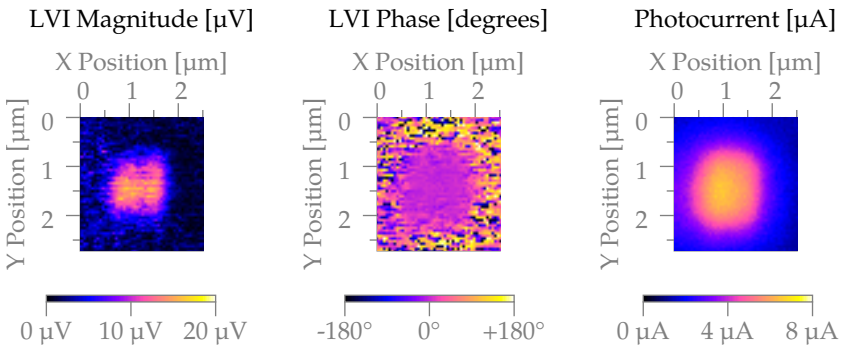


Figure 5.16: VIS-LVI lock-in magnitude/phase and photocurrent (source-and-drain to substrate) maps of the area highlighted in Fig. 5.14 acquired with a 10x/0.3NA objective lens in combination with the GaP SIL. Reprinted with permission of ASM International. All rights reserved. [43]

of Fig. 5.15. It is evident that the missing pulse can be detected and the overall waveform is reproduced.

Following this experiment, LVP measurements were also attempted using the GaP SIL. However, a satisfactory signal-to-noise ratio (SNR) could not be achieved. This seemed to be due to a too low returned optical power at the detector. Most likely imperfections of the SIL and setup already discussed in Sect. 5.2.5 lead to a relatively high loss of light, especially at the SIL-DUT interface. The SIL LVP experiments therefore only demonstrated VIS-LVP in general without SIL application. However, it is expected that a higher laser power or more applied pressure would lead to successful SIL VIS-LVP measurements.

In the case of LVI, on the other hand, it was possible to acquire results with the SIL, although a relatively small magnification objective lens (10x/0.3NA) had to be used to deliver enough optical power. The fact that LVI is possible at all, as opposed to LVP, is caused by the better noise suppression of the lock-in amplifier when compared to the simple averaging used by the oscilloscope. Therefore, a satisfactory SNR could be achieved.

The LVI measurements were performed in the highlighted area of Fig. 5.14 and acquired the lock-in signal magnitude as well as the phase and the injected photocurrent. The resulting data can be seen in Fig. 5.16.

The magnitude plot shows that an LVI signal arises only in the central area of the transistor structure. This is very similar to what one would expect for a gate-pulsed transistor analyzed with infrared LVI [53]. A similar behavior is shown for the phase plot: in the central area, a constant phase angle of zero can be seen, showing the presence of a stable LVI signal without sign inversions or other phase effects. At the edge of the analyzed area, no signal seems to be present, as only phase noise is detected. The photocurrent plot demonstrates

that a significant amount of current is injected into source and drain. As the design current for this transistor is 15 µA, the assumption that visible light probing might influence device operation seems reasonable. The VIS-LVI capabilities of the setup in conjunction with the GaP SIL are thus demonstrated.

To sum up the results section, it can be said that visible light imaging in silicon using a GaP SIL has been proven feasible. Furthermore, visible light laser voltage imaging was also demonstrated using the GaP SIL. Laser voltage probing with visible light could only be demonstrated without the SIL. However, it seems reasonable that it can be performed by applying improvements to the proof-of-concept setup and the SIL manufacturing process.

5.3 LOW-COST VISIBLE LIGHT LSM

During the development of and experiments on the VIS-LVP and VIS-LVI prototype setup of the previous sections, it became evident that the setup and the GaP SIL could be realized with less effort than expected, see Sect. 5.2.3. In a security context, this naturally directly leads to the question if, apart from the results achieved in the academic context of the previous sections, a visible light laser scanning microscope (LSM) setup could actually be realized by hobbyists at home for low cost. If this is the case, the ease of access to such a system would pose a greater security threat than the previously discussed scenarios using professional FA equipment. To put it in another way, this would put attacks from the so-called "lab attack" class into the "shack attack" domain. Consequently, a risk evaluation for low-cost homemade visible light LSMs was desirable.

The most straightforward way to perform such an evaluation would simply be to attempt to build such a system. If this attempt succeeds, the same system could directly be used to assess the attack potential of such an approach. The very first step towards a full system with similar capabilities as the FA systems used in the previous chapters of this work would be the creation of reflected light images, as this is the base of all discussed FA techniques. Even without full FA capabilities, such a system might, apart from the risk assessment, also be desirable for other LSM-based research or activities. Later addition of the GaP SIL and the extension of its capabilities to, for example, fault injection and optical probing is also imaginable.

To construct such a system, the basic components of an LSM need to be reviewed. These are a laser light source, some means of focusing and moving the beam across a device, and a detector to measure the reflected light. An additional piece of hardware, such as a PC, then needs to sample the reflected light detector values while scanning and display them as 2D data. During system design, it should also be taken into account that professional FA LSMs are usually laid out

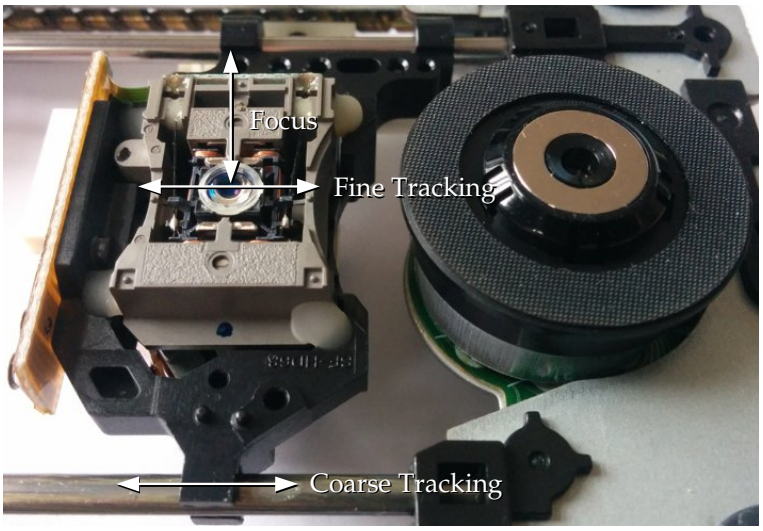


Figure 5.17: Photograph of an optical pickup unit as used in DVD recorders. The translation axes of the pickup lens are indicated. The rotational drive spindle can also be seen on the right.

for work in the context of laboratories and IC fabrication, with many features a low-cost attacker might actually be able to do without, such as fast acquisition. An important question is where an attacker could acquire suitable optical components with ease and for a low price. As lasers and other high-tech optical devices are mass-produced for use in consumer products today, these might be a suitable component source. One class of devices which already have many aspects of an LSM incorporated are optical drives for DVDs and Blu-rays. These incorporate a laser, focusing optics, a light detector, and control of the lens position in two axes (tracking and focus). Usually, these components are integrated together with the necessary driver ICs and control circuits in a compact optical pickup unit. Fig. 5.17 shows an example of such a pickup unit. The whole assembly can be moved for coarse tracking on two rails. More interestingly, the focus and fine tracking axes of the lens can be controlled via electromagnetic coils in the pickup head. Consequently, the laser spot can also be moved along these axes. Therefore, the only thing needed for LSM operation with such a pickup unit would be an additional axis perpendicular to the tracking axis for moving the beam across the DUT. As a consequence, optical pickups are ideal candidates for an approach to low-cost LSM systems.

For this reason, the development of the low-cost LSM was based on optical drives. The development was pursued in a cooperation between TU Berlin (TUB) and the University of Applied Sciences Jena (EAH Jena). The motivation at EAH for such a system was slightly different to the one at TUB. EAH is involved in space electronics research, in particular in the effects referred to as single event upsets (SEUs) or single event latch-ups (SELs). SEU/SEL are disturbances in

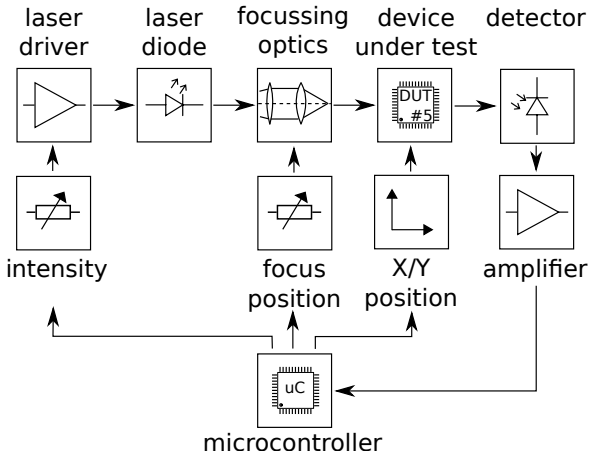


Figure 5.18: Block diagram of the low-cost visible light LSM. © 2017 IEEE. [39]

device operation that are caused by the impact of ionizing particles in space and disrupt device operation or might in the worst case lead to device destruction and catastrophic failures. SEU/SEL can be studied at ground level using actual radiation sources, however, these are expensive and dangerous. Consequently, a common approach is to use light injection to simulate particle impact at ground level without the risks associated with ionizing radiation. For these investigations, a suitable tool was desired at EAH. A system such as the low-cost LSM would be ideal for this, as it would allow for navigation as well as for high-powered light injection into the analyzed ICs. As EAH had already developed a single point laser injection tool from DVD recorder components, this was extended into a working LSM prototype in a first step.

Fig. 5.18 shows a basic block diagram of this system. Apart from a simple translation stage and some circuits to generate control voltages and currents, the system is exclusively composed of optical drive components. A microcontroller is used as the main control component during operation. The microcontroller interfaces with the original laser diode control IC which causes the laser diode to emit 650 nm radiation. This is then focused onto the device under test (DUT) using the lens of the optical pickup. This lens can be moved using the original control coils for focusing as well as for scanning in the fast x-axis. The slower y-axis scan is done by the added translation stage. During the raster scanning process, the signal from the optical pickup's reflected light detector is sampled by the microcontroller and the data is sent to a PC which then assembles it for displaying.

Based on this first version, a second prototype with minor modifications was built at TUB and used for tests of the imaging quality in a security context. An image of this setup can be seen in Fig. 5.19. This second prototype was able to realize the slow y-axis stage by using only optical drive components and thus brought the total hardware

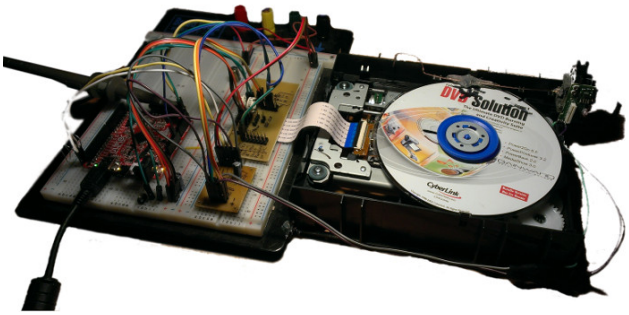


Figure 5.19: Photograph of the low-cost visible light LSM prototype built at TUB.

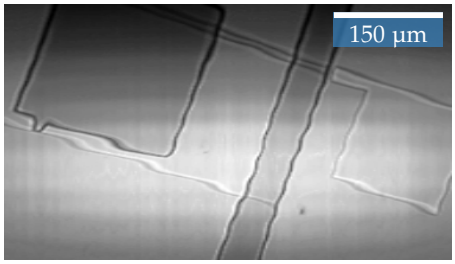


Figure 5.20: Reflected light frontside test scan of a wafer structure using visible light (650 nm) and the low-cost LSM setup. © 2017 IEEE. [39]

cost down to less than $100. In Fig. 5.19, the DUT is mounted on the lower side of the DVD disk. Horizontal scanning is provided by the pickup, which is hidden below the disk and is connected via the white flat ribbon cable. Vertical scanning is provided by rotating the disk via a push rod at its top. The push rod is connected to a modified lens coil from a second pickup, which can be seen on the top right. The control electronics are visible on the left side.

Fig. 5.20 shows a test scan of a wafer structure acquired with this prototype at TUB. Although some distortions are visible along the usually straight structure edges, a satisfactory image quality can be achieved and basic LSM operation is thus demonstrated.

For testing the VIS imaging capabilities through the backside, a 60 nm technology Altera Cyclone IV FPGA was used. The FPGA was thinned to less than 5 µm for this experiment.

The backside scan of this device is shown in Fig. 5.21. It can be seen that fringes are created in this case. This is due to interference effects in the thin silicon layer. Nevertheless, the chip structures can be resolved and backside imaging is thus proven to be feasible.

To acquire an estimate of the achievable resolution, the "land and pit" structure of a CD-ROM was used. The LSM was set to the smallest scan step, which is equivalent to the smallest pixel size and thus

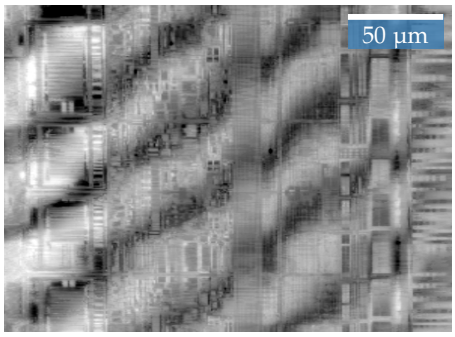
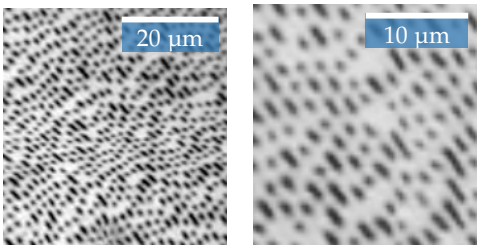


Figure 5.21: Reflected light backside scan of a 60 nm technology Altera Cyclone IV FPGA using visible light (650 nm) and the low-cost LSM setup. © 2017 IEEE. [39]



(a) Full scan (b) Detailed view

Figure 5.22: Reflected light resolution test showing the "land and pit" structure of CD-ROM data tracks. The tracks have a pitch of 1.6 µm and run from the top left to the bottom right. The width of the darker "pits" is 0.6 µm. © 2017 IEEE. [39]

also to the largest magnification. Fig. 5.22 shows the resulting scan images.

This demonstrates that the LSM is able to resolve these data track structures which have a pitch of 1.6 µm, with the darker pits having a width of just 0.6 µm. Consequently, it can be concluded that the LSM is able to achieve resolutions in the sub-micron range. It should be mentioned that the resolution is currently limited electrically by the minimum step size of the digital-to-analog conversion circuit which controls the x- and y-axis. A better resolution might thus be possible with improved control hardware.

In total, the risk evaluation for low-cost homemade visible light LSMs has shown that such a setup is not only possible but that it can achieve sub-micron resolution for less than $100 of hardware cost. In the future, this base system could be expanded to allow for fault injection, which is in fact very similar to the SEU/SEL generation already demonstrated at EAH with the non-imaging predecessor system [96]. Currently, a setup with imaging capabilities for SEU/SEL experiments is being developed at EAH [95]. With minor modifications, such a setup could be used for fault injection attacks with LSM navigation capabilities. This would allow for evaluation of the actual

attack potential of the low-cost approach, apart from basic optical reverse engineering, which is already possible with the setup presented in this section. An additional extension could be the addition of a suitable amplifier/detection circuitry to allow for laser voltage probing and laser voltage imaging, to evaluate this more powerful class of attacks on a low-cost system. Finally, it should also be mentioned that the system is in principle compatible with the GaP SIL presented in Sect. 5.2.3.

5.4 CHAPTER CONCLUSION

This chapter has assessed likely paths for the use of visible light (VIS) for resolution improvement in both failure analysis (FA) and low-cost attack scenarios.

As visible light FA setups will be available for rent in FA labs shortly after their availability on the market, they will also become available to attackers if the method proves to be feasible. Thus, the first part of this chapter has evaluated the possibilities of using visible light in combination with devices thinned down to the micrometer range and a gallium phosphide solid immersion lens (GaP SIL) applied to the backside. For this, a GaP SIL was designed, fabricated, and verified experimentally. It was shown that the GaP SIL can be manufactured for a cost of around $3400 using high-precision turning. Furthermore, it was demonstrated that using the developed hardware and software setup a standard laser scanning microscope (LSM) can be retrofitted for VIS GaP SIL probing. Experimental verification of image acquisition demonstrated resolution improvements by 190% and 170% in IR and VIS respectively when compared to the respective non-SIL resolution. Comparison of VIS and IR illumination when using the SIL demonstrated an improvement of 60%. Furthermore, visible light laser voltage imaging (VIS-LVI) using the GaP lens was demonstrated on 16/14 nm FinFET test devices. Visible light laser voltage probing (VIS-LVP) was also demonstrated on the same device, however, it could not be performed in combination with the Gap SIL. This was due to manufacturing errors in the GaP SIL prototype and shortcomings in the current proof-of-concept setup. Nevertheless, the feasibility of VIS-LVI and VIS-LVP, as well as the fabrication of a working, VIS-compatible SIL, have been demonstrated. In the meantime, VIS-LVI and VIS-LVP systems with corresponding SILs have indeed been released on the market by companies such as e.g. Checkpoint Technologies. Whether they will gain widespread distribution in FA remains to be seen, especially considering the challenging requirements for sample preparation in comparison to the relatively low improvement of optical resolution by a factor of about two over current systems employing SILs in the infrared. Yet, in an attack context, it should be kept in mind that Sect. 5.2 has demon-

strated that a standard LSM can in principle be extended for VIS and SIL probing with a cost in the order of a few ten thousand dollars. It should be stressed that the demonstrated modifications imply an enhancement in system resolution by 369% when the original backside performance in IR without a SIL is compared to the now possible measurements in VIS with the SIL. For skilled and capable attackers, this might thus be an option to extend their reach down to smaller technology nodes.

The second part of this chapter has evaluated if attackers might be able to build a low-cost visible light LSM at home, using components from optical drives. The acquisition of reflected light images using such a setup was evaluated as a first step towards a low-cost visible light attack system. It was demonstrated that LSM image acquisition is indeed possible using DVD recorder components with a total hardware cost of less than $100. Frontside and backside measurements were performed and the achieved resolution was shown to be in the sub-micrometer range. It is planned to extend the setup in the future to allow for evaluation of, for example, fault injection attacks. The possibility of later adding more sophisticated techniques such as LVI, LVP, and a SIL was also discussed briefly.

In total, it can be said that it is currently unclear if VIS techniques for resolution improvement will become widespread in an FA context. However, for attackers with a suitable budget, VIS techniques can indeed allow them to attack smaller technology devices than previously when applying approaches such as demonstrated in the first part of this chapter. On the other hand, the second part has shown that even attackers with significantly constrained resources might be able to perform visible light attacks and do this at a much lower cost than what was previously thought possible.

6 COUNTERMEASURES

This chapter will present countermeasure concepts against the attacks demonstrated in the previous chapters. The first part will deal with countermeasures which were experimentally evaluated. In the second part, selected further potential countermeasures will be discussed.

6.1 IMPLEMENTED COUNTERMEASURES

This section will discuss concepts for countermeasures that were implemented as proof-of-concept (POC) circuits and present the results gained when evaluating them. More specifically, two countermeasures were implemented: one to prevent thermal laser stimulation memory readout and one to detect optical probing and fault injection attacks.

6.1.1 *Thermal Laser Stimulation*

In Chapter 3, it was demonstrated that thermal laser stimulation (TLS) can be used to read out the values of SRAM memory and battery-backed SRAM key storage (BBRAM). It was shown that plotting the current consumption under stimulation as a 2D map allows to reveal the memory contents by analyzing the resulting TLS patterns. As a countermeasure, a circuit was developed which aims at masking the nanoampere TLS currents with a noise current. This approach will be presented and evaluated in this section. The circuit and the results presented here have been published in [42].

The concept was developed to defend primarily against BBRAM TLS attacks on field-programmable gate arrays (FPGAs), although it should also protect against TLS in general. A suitable circuit needs to fulfill the following requirements. First, as the FPGA containing the BBRAM is powered off during the attack, it needs to be supplied by the same battery as the BBRAM memory. As a consequence, it needs to not drain the battery excessively. Second, it should in principle be realizable by standard process technologies, so as not to add production costs. Fig. 6.1 shows a circuit diagram of a POC countermeasure developed with these goals in mind. For testing, the circuit is added externally between the FPGA, which contains the BBRAM, and the current amplifier, which measures the TLS signal. In a real-world scenario, the circuit would be implemented on the FPGA die to prevent tampering.

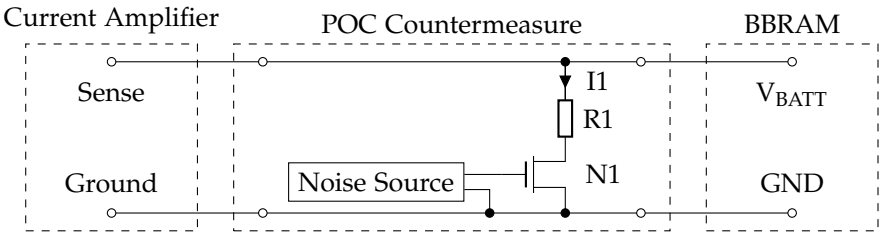


Figure 6.1: Test circuit for the proof-of-concept BBRAM thermal laser stimulation countermeasure. [42]

The circuit injects a noisy current into the V_{BATT} net which supplies the BBRAM in the FPGA. It does so by feeding the gate of a MOSFET transistor (N_1) with a noise signal. This leads to a noisy current I_1 on V_{BATT} which is limited by resistor R_1. I_1 should then mask the data dependent TLS current generated in the BBRAM during the attack.

I_1 can be characterized by its mean or offset current I_{mean} and the peak-to-peak amplitude of the fluctuations around I_{mean}, referred to here as I_{pp}. For protection, the fluctuations characterized by I_{pp} should be as high as possible to achieve sufficient masking. However, the proposed circuit will generate symmetric fluctuations, which results in $I_{pp\ max} = 2I_{mean}$. Thus, a large I_{pp} will increase I_{mean}. This I_{mean} will drain the battery and thus reduce its lifetime if it is too large. As a consequence, with this POC circuit, a suitable trade-off has to be established and set via R_1.

Additionally, it needs to be kept in mind that, depending on the scan speed of the laser and the structural geometry, the frequencies contained in the TLS current can vary widely. Consequently, the noise source needs to deliver a wide-band signal, so as to mask the whole spectral range. When implemented on the FPGA, such a white noise signal can be generated by electronic circuits employing e.g. Zener diodes as a noise source. In the case of this experimental setup, the gate signal of N_1 is supplied by a Keithley 3390 function generator in "noise" mode.

Using the setup shown in Fig. 6.1, the BBRAM key extraction experiments of Sect. 3.4 were repeated. Fig. 6.2 presents the results of these measurements. Only the left block of the BBRAM memory is shown. The first measurement was performed with the countermeasure disabled, 40% laser power, and 72 s scan time, see Fig. 6.2, left. It can be seen that the TLS patterns which are needed for data extraction can be recognized easily. Application of a $2D$ Gaussian smoothing filter further improves the measurement result.

For the next measurement, the countermeasure is enabled with $I_{mean} = 1.3\ \mu A$ and $I_{pp} = 300\ nA$. Even with an increased laser power of 100%, an increased scan time of 120 s, and averaging of five measurements, the clarity of the TLS pattern is drastically reduced,

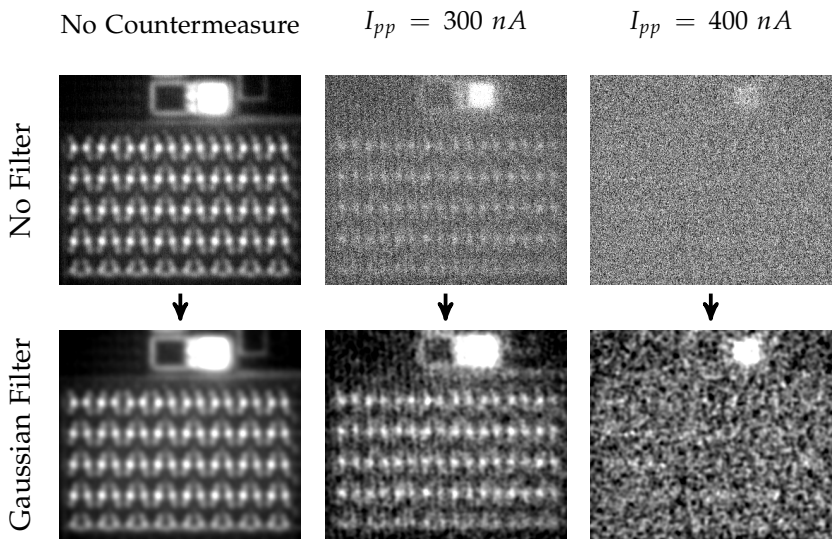


Figure 6.2: Effect of the proof-of-concept countermeasure on TLS results with and without 2D Gaussian filtering and enhanced contrast applied. Left column: countermeasure disabled, 40% laser power, 72 s scan time. Middle column: countermeasure set at $I_{pp} = 300\ nA$. 100% laser power, 5x120 s scan time. Right column: countermeasure set at $I_{pp} = 400\ nA$. 100% laser power, 5x120 s scan time. [42]

see Fig. 6.2, middle. Although a Gaussian filter improves the data, not all bit values can be recognized without error anymore.

Finally, in Fig. 6.2, right, the results of a measurement with $I_{pp} = 400\ nA$ are presented. In this case, not even the general cell locations can be recognized, regardless of if filtering is used or not. This demonstrates that the countermeasure is able to mask the TLS current successfully with the mentioned settings.

Although the expected current consumption of an integrated noise source is unknown at this point, an approximation of the battery lifetime using the I_{mean} of the POC setup can be performed. This results in a lifetime of eleven years that the system can be *permanently* disconnected from *all* power sources. The detailed calculation for this approximation is given in [42]. It should be noted that the FPGA automatically disconnects the backup battery as soon as a power source is available. As a consequence, during normal operation, the battery can be expected to fail from old age before it is drained. Thus, even with this POC setup, the effective battery life is not reduced.

In total, the results discussed in this section demonstrate the potential and feasibility of this countermeasure concept. Additional details, as well as more sophisticated, capacitor-based concepts which allow for higher I_{pp} fluctuations while achieving the same or lower I_{mean}, are discussed in [42]. Finally, it should be noted that this countermeasure can also be applied to SRAM memory readout as demonstrated

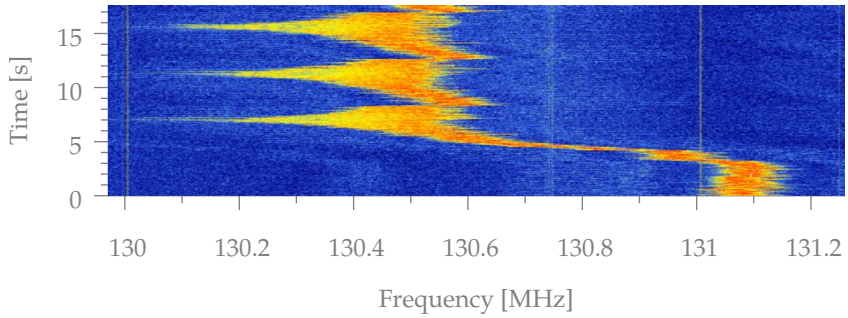


Figure 6.3: Waterfall spectrum plot showing the frequency shift of a single ring oscillator when scanned with a 1.3 µm laser. The scan is started at around three seconds.

in Sect. 3.3. In this case, the requirements for the circuit are more relaxed, as it does not have to operate from a battery.

6.1.2 *Optical Probing and Fault Injection*

This section will discuss a countermeasure that was implemented to defend against optical probing attacks. It is designed primarily for the scenario of an attack on an FPGA such as demonstrated in Chapter 4. As a consequence, it has been designed with the requirement of retroactively adding protection logic to the configuration bitstream of the FPGA. However, in principle, it should also be realizable in an application-specific integrated circuit (ASIC). Furthermore, the results will demonstrate that the structure can also detect the shorter wavelengths used for e.g. fault injection attacks, such as demonstrated in Chapter 2. The countermeasure and the results presented here have been published in [78].

The optical probing attacks of Chapter 4 used a 1.3 µm laser beam which was modulated by the electrical waveforms present in the device and thus allowed for the extraction of key data, plaintext, and for physically unclonable function (PUF) characterization. To implement the PUF in the experiments of Chapter 4, ring oscillators (ROs) were used. These showed a slight shift in frequency when subjected to the laser radiation. Sect. 4.2.6 has already mentioned this shift and discussed why it is not a hindrance for the attack presented there. Yet, in principle, this shift could be used as a potential countermeasure and will consequently be discussed here.

Fig. 6.3 shows a waterfall spectrum plot of the output of a single RO when subjected to 1.3 µm radiation in the setup used for the experiments of Sect. 4.2. It can be seen that at 0 s the RO oscillates with a center frequency of about 131.08 MHz. At about three seconds, the "Phemos" laser scanning microscope (LSM) starts to acquire reflected light images of the device with 100% laser power. The images are ac-

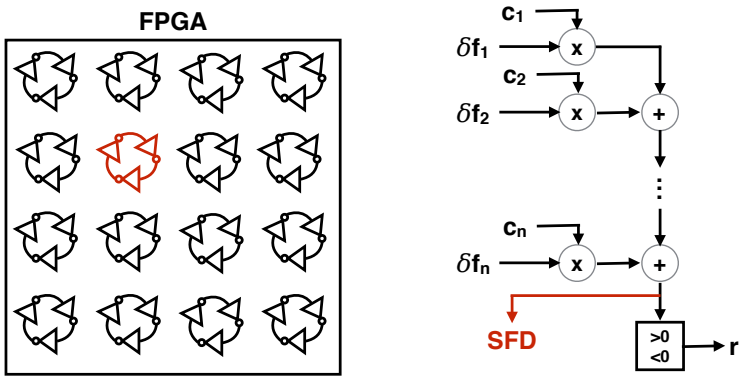
quired in "live mode", which causes the scan to repeat after a frame is acquired. As a consequence, the center frequency of the RO can be seen to shift to about 130.45 MHz, a change of 0.5%. Furthermore, the frequency can be seen to vary periodically with the frame acquisition time, which was set to four seconds.

The general frequency shift can be explained by a global temperature increase of the device caused by the total heat generated by the laser. This alters the delay of the gates which make up the RO [68]. The additional periodic variation is then caused by the scanning beam first moving towards the RO and increasing the local RO temperature, then inducing the maximum shift when hitting the RO directly, and finally moving away from the RO, causing a decrease in the shift. This then repeats for the next frame. This indicates that ROs can, in principle, be used to detect 1.3 µm laser radiation. For shorter wavelength radiation, such as the popular 1.1 µm fault injection wavelength, a similar delay alteration caused by generated photocarriers can be expected [55]. Based on these effects observed during the experiments of Chapter 4, a laser attack detection circuit was developed.

For the design of such a countermeasure, there are multiple requirements. First, a suitable circuit has to cover a large area of the device, so that an attack anywhere will lead to detection (spatial coverage). Second, the circuit should be able to detect an attack even if it is very short, as in the case of fault injection (temporal coverage). Third, the sensor should not be easily deactivated or tampered with.

During the design phase, it became evident that it might be advantageous to combine an RO-based detection circuit with an RO-based PUF to achieve these goals. Such a "two-in-one" solution could share a common set of resources, thus making the overall circuit cheaper to implement. Put in another way, if a circuit requires either a PUF or an attack detection circuit, one would get the other function "for free". The characteristics of PUFs in such a circuit would also have advantages for the detection of tampering performed on the countermeasure structure. A selection from different candidates and concepts, which is detailed in [78], lead to the combination of the known concepts of RO sum PUFs and RO networks into a proof-of-concept circuit for a PUF-based security monitoring scheme. This scheme is referred to here as "PUFMon".

In this scheme, a network of ring oscillators is distributed across the FPGA, see Fig. 6.4a. These ring oscillators form the base of an RO sum PUF, which can be seen in Fig. 6.4b. Here, n pairs of ROs are connected to the PUF. For normal PUF operation, the frequency difference δf for each pair is determined using binary counters connected to the ROs. The counters are simply driven by the ROs, stopped after a certain amount of time, and then subtracted from each other, arriving at n values for δf_1 to δf_n. Using a challenge consisting of n challenge bits c_1 to c_n, these values are then combined into the PUF



(a) ROs distributed inside an FPGA. If an RO is influenced by a laser attack, it will shift its frequency slightly.

(b) RO sum PUF architecture with additional output of the summation of frequency differences (SFD) for attack detection.

Figure 6.4: Basic concept of the "PUFMon" PUF and attack monitoring implementation. © 2017 IEEE. [78]

response r. For this, each δf is multiplied by $+1$ or by -1, depending on its corresponding challenge bit. All these values are then summed to arrive at the summation of the frequency differences (SFD). The sign of the SFD is then evaluated to arrive at the final response r, which will be 0 if the sum is negative or 1 otherwise. This provides PUF functionality under normal operating conditions.

However, the SFD is also output by the PUF and can be used for attack monitoring. If one of the ROs, see Fig. 6.4a, is subjected to laser radiation, it will shift its frequency. This will result in a change of the SFD and can be detected. However, as there will also be changes of the SFD caused by e.g. noise and different operating conditions, this detection is not straightforward. To determine when an attack is likely, the SFD behavior needs to be characterized. A solution would be to exercise the PUF under normal operating conditions while it is still in the trusted field and record the maximum and minimum SFDs for a number of challenges. If challenge-response-pairs (CRPs) are already collected in an enrollment phase of the PUF for later authentication, the SFD characterization can simply be added to the procedure. If the normal limits for the SFDs are known, these can later be used to detect an attack, either by the device itself (offline) or by a remote server (online). Similarly, the limits of the SFDs can also be stored online or offline. For a discussion of the advantages and disadvantages of the respective storage and verification methods see [78].

To determine the feasibility of the general approach of PUFMon, it needs to be demonstrated that an SFD evaluation can indeed detect a laser attack. For this, a proof-of-concept (POC) circuit was implemented on Altera Cyclone IV FPGAs and subjected to laser attacks on

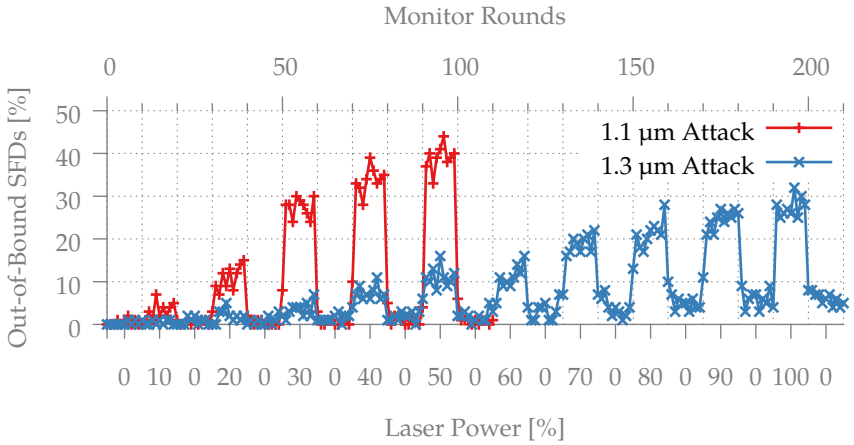


Figure 6.5: Effects of 1.1 µm and 1.3 µm laser attacks on the SFDs of the "PUFMon" implementation. Replotted from data previously published in [78].

the "Phemos" LSM used previously in this work. The device and optical setup are the same as described in Sect. 4.2.5, with the additional option of using the 1.1 µm laser. A 20x lens is used, which delivers about 50 mW of optical power onto the device under test (DUT) with both the 1.1 µm and 1.3 µm sources. The POC contains a 16-bit RO sum PUF with 32 ROs, each consisting of 5 inverters. The PUF can be controlled and evaluated via a UART connection to a PC. Additional details of the setup are disclosed in [78].

During the enrollment phase of the PUF, 100 randomly chosen challenges were each applied 50 times to gather the minimum and maximum SFDs for each challenge. Then, during the attack experiments, the same set of challenges is applied in a loop and the results are compared to the stored minimum and maximum SFDs. The resulting data then allows determining the percentage of out-of-bound SFDs.

For attack detection evaluation, both the 1.1 µm and 1.3 µm lasers were used to scan the device with increasing power levels. For each step of the experiment, the laser was set to a certain power level and 10 rounds of SFD evaluation were performed by applying the set of 100 enrolled challenges. Afterward, this procedure was repeated with the laser switched off before the next power level was set. The results of this experiment are presented in Fig. 6.5.

For the 1.1 µm attack, it is evident that the number of out-of-bound SFDs increases roughly linearly with increasing laser power. It can also be seen that at 20% laser power 10% of the SFDs are out of bound. At 50% laser power, the experiment had to be aborted, as there were concerns about damaging the device.

For the 1.3 µm attack, a similar behavior is observed, although a smaller percentage of SFDs is out of bounds. It can be seen that to reach the same level of 10% of out-of-bound SFDs, 50% laser power has to be applied instead of 20%. Additionally, the effect of residual

heat can be observed in the interval between the attacks. As the device increases its global temperature with time due to the laser radiation, not all SFDs return to their pre-attack level immediately.

These results show that the PUFMon approach can be used to detect attacks, given enough laser power. Furthermore, 1.1 µm attacks will be more easily detected than 1.3 µm attacks, as they seem to influence the RO delays more strongly. The effect of residual heat raises concerns about the reliability of this approach in environments where the temperature changes significantly and often, such as military and industrial applications. Yet, the general detection capability of PUFMon has been demonstrated. In addition to the detection of laser scan activity discussed here, PUFMon was also shown to be able to detect clock and configuration manipulation, for details see [78].

The evaluation of PUFMon has unveiled several advantages and disadvantages. One of its strengths is that it can be implemented retroactively on the FPGA logic fabric. This allows designers to add this security measure even when the FPGA was not originally designed for hardware security monitoring. Furthermore, the user has full control and knowledge of the security circuits, as opposed to using closed-source proprietary solutions from FPGA vendors. In principle, because of its sensitivity to 1.3 µm radiation, PUFMon could also protect against thermal laser stimulation (TLS) as demonstrated in Chapter 3. However, this is limited to TLS attacks on active devices. If a device is in its powered-off state, as in the case of the BBRAM key extraction discussed in Sect. 3.4 and 6.1.1, PUFMon will not be able to operate, as the backup battery will not be able to supply enough energy. Further experiments in [78] also demonstrated that the challenge-response behavior remained stable, despite large changes in the SFD values, which is another advantage. A weakness of PUFMon is that the used RO network has a relatively high power consumption, which makes it unsuitable for low-power applications. This could be mitigated by the use of fewer ROs only in critical regions or of only activating the PUFMon circuit during critical periods. Another disadvantage is its high sensitivity to global voltage and temperature changes. Depending on the type of application, this might be undesired due to the triggering of false alarms. A solution could be a higher threshold at the cost of reduced attack detection probability.

Future implementations of PUFMon could circumvent the high sensitivity problem by exchanging the SFD evaluation with a circuit which determines the average of all ROs and then compares each RO against this average to detect local attacks. This would also eliminate the need for characterization of the SFD behavior and the resulting SFD database. However, such a circuit would need more resources accordingly.

A resource-saving and pragmatic alternative would be to monitor for challenges whose SFDs are particularly insensitive to global temperature variations during the enrollment phase. Of these, only the most stable ones could then be selected for attack detection. It is reasonable to assume that such challenges exist, as for specific challenges the thermal changes in the individual δf values will cancel out if δf values with similar thermal behaviors are fed into the SFD sum with opposing signs.

Whether such circuits are more reliable in a real-world implementation and whether they can be implemented with a reasonable amount of resources is unknown at the moment. This would need to be verified theoretically and experimentally. Consequently, such evaluations could be a future prospect for the improvement of the PUFMon concept.

6.2 FURTHER POTENTIAL COUNTERMEASURES

As a result of the attack evaluations of Chapter 2, 3, and 4, additional countermeasure approaches have been presented and discussed in the connected publications. However, these approaches have not been verified experimentally and are thus speculative in nature. As a consequence, they are only mentioned here briefly, and not all published approaches are discussed. Readers interested in a more in-depth discussion of these approaches are referred to [10, 39–42, 78, 79, 81] and the sources cited within.

6.2.1 *Reconfiguration Attacks*

The reconfiguration attacks of Chapter 2 have demonstrated that it is possible to alter circuits precisely by using scanning profiling of fault-sensitive locations. This technique was used to attack PUF implementations via laser reconfiguration.

In some modern FPGAs, configuration error surveillance circuits are available in hardware. These are actually intended for single event upset (SEU) configuration errors, which are caused by ionizing radiation strikes. However, these circuits can in principle be used to monitor for reconfiguration attacks. Yet, it is unclear if an attacker might be able to disable the error checking circuit first before moving on to her originally intended attack.

A similar approach was developed as a direct response to the attacks of Chapter 2 and published by Sahoo et al. in [66]. The basic idea of [66] is to include additional fault-checking circuits directly into the PUF design to make the fault injection detectable.

Another elegant approach to detection of disturbances induced by fault injection is proposed by He et al. in [27]. Here, similar to Sect. 6.1.2, a ring oscillator is used as the detection element for laser

radiation. However, instead of detecting the frequency change by logic configured on the FPGA, the RO is fed into one of the phase-locked loops (PLLs) available in hardware in many FPGAs. As PLLs are usually used to generate phase synchronous clocks, they contain dedicated circuitry which will detect if the input clock goes out of phase to the generated output clock. As the laser will disturb the input clock generated by the RO, the PLL will assert an "unlocked" signal which can be used to raise an alarm. This approach has the benefit of using hardware specifically designed to detect frequency disturbances, while not using additional resources on the logic fabric of the FPGA.

For both previous countermeasures, it is again unclear if the detection circuits can be manipulated. In any case, they will make the attacks more challenging.

Another approach to detecting configuration errors is to implement redundant circuits and determine their output by majority voting or raising an alarm when outputs differ. Although this might be a solution for conventional circuits, it seems unfeasible to duplicate (i.e. physically clone) multiple PUF instances with the same behavior inside a device.

6.2.2 *Thermal Laser Stimulation Attacks*

Sect 6.1.1 has already demonstrated that data extraction by thermal laser stimulation (TLS) can be hindered by masking the small TLS currents with a noisy current. In a similar approach, a controlled current-sinking circuit could be implemented which actively keeps the overall current consumption of the device constant. In theory, this should prevent current-based laser stimulation attacks. Furthermore, the constant current consumption would only have to be slightly higher than the expected maximum current consumption of an unprotected circuit.

6.2.3 *Optical Probing Attacks*

The attacks of Chapter 4 have demonstrated that an attacker can first locate transistors of interest and then extract data from them. This is done using the light modulation caused by the electrical signal present at the transistors. The technique used to detect points of interest by mapping only transistors switching at a certain frequency is referred to as laser voltage imaging (LVI). Extracting waveforms from the transistors by averaging multiple traces of the reflected light signal is called laser voltage probing (LVP).

Many attacks of Chapter 4 required either the induction of a chosen frequency for LVI or a trigger for LVP waveform acquisition. In the presented attacks, these were provided by, for example, pulsing ex-

ternal reset signals or using externally available clocks for triggering. A countermeasure could thus be to make sure all critical external signals are routed through a circuit that destroys the fixed relationship between internal and external signals. Hardware candidates for such a circuit could be random delay FIFO buffers, the addition of jitter to the external signals, or even a completely asynchronous internal clock. Similar desynchronization efforts could also be performed in software by adding random delays. It should be kept in mind that the internal clock needs to be unstable in addition to the previously mentioned countermeasures. Otherwise, the attacker might be able to synchronize her equipment to the internal clock by other means, such as electromagnetic or current consumption analysis. An unstable internal clock will also possess a wide spectrum which will worsen the signal-to-noise ratio of LVI signals. It should, however, be kept in mind that the design of such countermeasures might be challenging and could also result in a worse circuit performance.

To protect against optical probing attacks where external signals are not required, such as the physically unclonable function characterization of Sect. 4.2.4, a more fundamental countermeasure might be possible at the transistor level. If gates carrying an inverted signal are placed next to the original gates, and the total structure is below the resolution limit, their signals should cancel out at the detector. However, such an approach would require the development and verification of suitable structures and ASIC design tools.

A solution which limits the number of traces an attacker can acquire for a given device is the "configuration counter" detailed in [57], which is available in Xilinx UltraScale and UltraScale+ FPGA devices. This solution erases the decryption key needed for boot from memory when a certain number of reboots is exceeded. Although this can protect against LVP/LVI attempts, Sect. 3.4 has demonstrated that instead the key can be extracted via thermal laser stimulation on these devices.

6.2.4 *General Protection*

From the previous sections, it is obvious that a more general approach would be desirable to protect the data in integrated circuits. Cryptographic solutions such as additional key encryption or key obfuscation have been implemented by hardware manufacturers [57]. Other, more advanced solutions have also been proposed to prevent attacks on critical and key data. Yet, in most cases, the critical data has to be regenerated to be used for its intended function at some point. Furthermore, such approaches will also be problematic if the device considered has to perform operations on sensitive data. To work with the data, the plaintext necessarily has to be present in some part of the device. Thus, such countermeasures can be circumvented by di-

rectly attacking the plaintext gates, as was demonstrated in Sect. 4.3. Another approach would be to distribute the critical gates across the chip and obfuscate their order. Although this will constitute an additional obstacle for an attacker, it does not prevent attacks in principle. Small technology sizes also do not necessarily protect against attacks, as Chapters 3 and 4 have demonstrated that a resolution of about 1 µm can be used to attack 28 nm and 20 nm technology devices. In conjunction with this, the resolution improvements via solid immersion lenses with or without the application of visible light discussed in Chapter 5 should also be mentioned again.

This discussion leads to the conclusion that a more thorough solution is needed. Ideally, such a solution would completely deny the attacker optical access through the silicon backside. The countermeasure circuit discussed in Sect. 6.1.2 has already demonstrated an approach to *detect* optical access. Still, *prevention* of optical access would be more desirable.

Amini et al. present a structure in [4] which uses a reflective backside coating to achieve this. The coating is not only non-transparent but its integrity can also be verified. To perform this task, infrared light is emitted from multiple p-n junctions inside the silicon of the device and reflected when it reaches the backside coating. The reflected light, still inside the silicon, then travels back to the active area where it is detected by another set of p-n junctions. As the emitters and detectors are distributed, each emitter/detector pair sends and receives light under a different angle. As the coating is engineered to have an angular dependent reflectance, such a structure can be used to detect the integrity of this specific backside coating. To keep the implementation overhead and cost low, the protection structure is planned to use the already available p-n junctions of transistors needed for the core functions of the device in a real-world implementation. Yet, the monitoring circuit will need power to function. Thus, in its current form, this structure cannot prevent against attacks such as the ones demonstrated in Sect. 3.4, where keys or secrets are extracted from a switched-off device. A possible solution for this problem might be to combine the protection structure with a cryptographic PUF bootloader. After power-up, such a bootloader first determines the PUF response and then uses it for the decryption of the data that needs to be protected. In this way, when the device is powered off, no secret data, including the PUF response, is present. Since the PUF response can only be generated when power is applied, the backside protection structure will be active and extraction attacks on the PUF response or the decrypted secrets can be prevented.

A similar protection function might be available with more modern devices which use stacked dies. If critical functions are moved to an inner die, the metal layers of the outer dies will prevent optical access. The presence of the outer dies can simply be assured by exercising the

components provided by them. In combination with suitably sensitive characterization circuits, such as physically unclonable functions, it might even be possible to detect attacks which relocate and reconnect the outer dies to restore their function.

6.3 CHAPTER CONCLUSION

This chapter has discussed potential countermeasures against laser-based attacks on integrated circuits. In the first part of the chapter, the results of the evaluation of two implemented countermeasures were presented. The first countermeasure was designed to protect against data extraction by thermal laser stimulation (TLS). A proof-of-concept (POC) circuit of this approach was shown to successfully mask the TLS signal using noisy current injection. The second implemented countermeasure demonstrated a combination of a physically unclonable function (PUF) and an attack detection circuit. This circuit was implemented on a field-programmable gate array (FPGA) and was able to detect both 1.1 µm and 1.3 µm laser radiation while also functioning as a PUF. The second part of the chapter has then discussed additional potential countermeasures for different types of attacks and in general.

Although technique-specific countermeasures might be helpful if a certain type of attack is especially probable, it has become evident that an attacker is always free to change her technique and approach. As a consequence, a protection structure which prevents an attack on a fundamental level is much more desirable. One approach would be to detect any ongoing attacks, however, the complete denial of optical access seems to be a more thorough solution. The actively monitored backside coating presented by Amini et al. in [4] and already discussed in Sect. 6.2.4 might be a candidate for such a fundamental denial of optical access, especially in combination with a PUF bootloader. Possibly, protection against optical access might come at no extra cost in the future through stacked dies and similar 3D packaging technologies. Although the demonstrated and discussed countermeasures can already protect against a wide range of attacks, none of them seem to be a complete solution or "silver bullet". Furthermore, the actual inclusion of countermeasures in a product also always depends on a cost-benefit analysis. Consequently, in the future, there will still be the need for research on and development of dedicated backside protection structures which are ideally cheap and universal.

7

SUMMARY AND CONCLUSION

This work has examined the question of what an attacker could be capable of if she had access to standard failure analysis (FA) equipment, namely FA laser scanning microscopes (LSMs). To answer this question, likely attack paths have been outlined and the corresponding hardware setups were realized for different classes of attacks. A "Phemos" LSM system was used as an example of failure analysis hardware for most attacks. It was assumed that the attacker can gain access to such a system either by renting or by acquisition. Additionally, modified standard laboratory LSMs and low-cost home-built setups were used.

In Chapter 2, a combination of laser fault injection (LFI) with a scanning test approach was evaluated. This approach was inspired by the laser-assisted and tester-based pass/fail testing performed in failure analysis. A suitable setup was developed and it was demonstrated that by using such a pixel-by-pixel test analysis, an attacker can map fault injection locations in a fully automatic way. The setup was first validated on simple logic gates implemented on the look-up tables of a 180 nm technology Altera MAX V complex programmable logic device (CPLD). The knowledge gained when profiling the gates allowed to precisely manipulate the configuration and thus the function of the gates using LFI. The setup was then used to perform attacks on proof-of-concept (POC) implementations of physically unclonable functions (PUFs) running on the CPLD. This allowed demonstrating reconfiguration attack feasibility against XOR arbiter PUFs, ring oscillator (RO) PUFs, and RO true random number generators.

Additionally, a detailed analysis of the fault mechanism in the employed device was performed. This analysis revealed a two-stage process as the most probable cause for the injected faults instead of the simple "single bit-flip" model usually employed. In the developed fault model, primary faults cause short circuits which then cause secondary faults in the look-up table and similar configuration memory. The secondary faults are caused by a combination of a drop in supply voltage due to the short circuit and the influence of localized heating on the configuration memory cells.

Chapter 3 evaluated thermal laser stimulation (TLS) attacks for data extraction. Previous work by [9, 49] was extended to not only analyze single bits but allow for full memory extraction on static random access memory (SRAM). A suitable setup was created and used to analyze the 2D current consumption response to TLS on a Texas Instruments MSP430 180 nm technology microcontroller. The tasks of

reverse-engineering the logical-to-spatial mapping ("descrambling") and creation of a suitable image recognition software for data recovery were handed out as two bachelor's theses. The results of these were then combined into an automatic data extraction software, partly in a cooperation with Technische Universität Darmstadt. Automatic data readout was demonstrated to achieve error rates between 0.4% and 5.5% on the MSP430. TLS data extraction was also evaluated on the battery-backed SRAM (BBRAM) key memory of a Xilinx Kintex UltraScale field-programmable gate array (FPGA) manufactured in 20 nm technology. The evaluation demonstrated a time of 7 hours for attack development, with a single non-invasive key extraction ultimately taking about 15 minutes.

Chapter 4, the final chapter concerning concrete attacks, evaluated likely attack paths using contactless optical probing techniques. Attacks on advanced key storage solutions as well as plaintext extraction were demonstrated. In the first part, bitstream decryption keys were recovered from a proof-of-concept PUF key storage implementation using laser voltage probing and laser voltage imaging (LVP/LVI). The POC implementation followed a concept by Xilinx and used a 60 nm technology Altera Cyclone IV FPGA. Characterization of the ROs of the implemented RO PUF was also demonstrated. The second part demonstrated circumvention of any key storage mechanism by direct optical probing of the plaintext gates. The attack used plaintext frequency induction to quickly locate the plaintext gates using LVI. The non-invasive attack was demonstrated on the application-specific integrated circuit (ASIC) decryption core of a Xilinx Kintex 7 FPGA manufactured in 28 nm technology. Attack development was performed in less than 80 hours of lab time.

In Chapter 5, the use of visible light (VIS) for resolution improvement in both failure analysis and low-cost attack scenarios was evaluated. For the FA scenario, an off-the-shelf laboratory laser scanning microscope was extended to perform both LVI and LVP with visible light. Furthermore, a gallium phosphide solid immersion lens was designed, manufactured, and integrated into the setup. This then allowed to successfully perform LVI/LVP on 16/14 nm FinFET technology devices. For the low-cost scenario, a visible light LSM with sub-micron resolution, a cost of less than $100, and the option of performing laser fault injection was developed in a cooperation with the University of Applied Sciences Jena.

Finally, Chapter 6 evaluated and discussed countermeasures against the demonstrated attacks. The first implemented countermeasure demonstrated the masking of thermal laser stimulation currents using a suitable noise source to prevent TLS BBRAM readout attacks on a powered-off device. The second evaluated countermeasure presented a combination of an RO sum PUF with a detection circuit for 1.1 µm and 1.3 µm laser attacks. The results demonstrated detection

of both wavelengths given enough laser power. Additionally, potential future countermeasures were presented and discussed.

To conclude, this work has shown that using FA techniques as the basis for attack development is a promising approach for attackers and a challenge to defenders. Even without high-resolution solid immersion lenses (SILs), attacks were successful on commercial devices down to the 20 nm node. The use of SILs, either with infrared or visible light, will allow attackers to extend their reach to even smaller technologies in the future. As failure analysis will always need to analyze the devices currently manufactured, it can be expected that current FA techniques will always pose a threat to current devices in one way or another. Low-cost approaches have further demonstrated that attackers might not even need access to expensive equipment. However, a key factor for all attacks presented in this work is the lack of backside protection. Although device- and attack-specific countermeasures were successfully implemented, denying the attacker optical access through the backside seems to be the most thorough approach for protection of future devices. As discussed in Sect. 6.2.4, monitored backside coatings such as proposed by Amini et al. [4], especially in combination with PUFs, or approaches using 3D packaging to prevent optical access might be a remedy. Yet, currently, an unprotected backside constitutes a large security risk, especially on modern flip-chip devices which drastically simplify attack preparation. Research and development of suitable backside protection structures will thus remain an important aspect of securing modern integrated circuits.

A

APPENDIX

A.1 DETAILED EOFM PARAMETERS FOR SECT. 4.3.3

To improve readability, the detailed measurement settings for EOFM acquisition were omitted in the result description of Sect. 4.3.3 and instead summarized in the following table. For each measurement, the table gives the corresponding figure number, the used objective lens, the measured FPGA configuration clock frequency f_{CCLK}, the EOFM frequency f_{EOFM}, the EOFM bandwidth BW_{EOFM}, the pixel dwell time T_{Dwell}, and the laser power P_{Laser} (in percent), as set in the control software of the Phemos microscope system.

Figure	Lens	f_{CCLK} [MHz]	f_{EOFM} [kHz]	BW_{EOFM} [kHz]	T_{Dwell} [ms]	P_{Laser} [%]
4.21a, 4.21b	20x	2.785	2785	10	0.33	100
4.21c, 4.21d	50x	2.780	2780	10	0.33	100
4.22, 4.23	50x	11.1	173.900	1	0.33	100
4.24	20x	31.3	122.265	1	0.33	100
4.25	50x	31.3	122.265	1	0.33	100
4.26	50x	31.3	122.063	10	0.33	100
4.27	50x	31.3	121.500	1	0.33	30

Table A.1: Detailed EOFM measurement parameters for Sect. 4.3.3.

A.2 FIT RESULTS FOR RESOLUTION IMPROVEMENT ESTIMATION

Tab. A.2 documents the detailed function fitting results used for estimation of optical resolution improvement in Sect. 5.2.5. s is the scaling factor of the function, I_0 is the grayscale level of the dark edge, A is the grayscale step height of the transition and X_0 is the position of the edge transition. For details see Eq. 5.4.

Experiment	s [1/µm]	I_0 [a.u.]	A [a.u.]	X_0 [µm]
IR	1.659	12.656	149.857	1.414
IR+SIL	4.812	8.598	137.139	1.237
VIS	2.829	146.004	79.079	1.325
VIS+SIL	7.776	30.78	102.309	1.037

Table A.2: Detailed fit results using Eq. 5.4 on the data presented in Fig. 5.13. This data has been previously published in [43]. Reprinted with permission of ASM International. All rights reserved.

A.3 SCILAB SCRIPT FOR CALCULATION OF EXPECTED LVI INTENSITY

This script calculates the expected laser voltage imaging (LVI) intensity for the key registers of the serial proof-of-concept key generation implementation of Sect. 4.2.2. Results of such a calculation can be seen in Fig. 4.9.

Listing A.1: Key register LVI intensity calculation in Scilab

```
clear;
//////FUNCTIONS//////////////

//function to generate a serial waveform for the key bits
function  y = serialize_key(x,bitwidth, key)
    bits = length(key)
    if bits>0 then
        for n= 1:bits
            y(find(x >= bitwidth*(n-1) & x < bitwidth*n)) = key(n);
        end
    end
    y(find(x >= bitwidth*bits)) = 0;
    y=y';
endfunction

//Periodic serialized key function
function a=periodic_key(x,T,bitwidth,key)
//We'll inspect every value of the vector x
    for i = 1:length(x)
        shift = floor(x(i)/T)*T;
        a(i)= serialize_key(x(i)-shift, bitwidth, key);
    end
endfunction

//Generate binary double vector from key value
function k=binary_vector(key_value,vect_length)
    k=strtod(strsplit(dec2bin(key_value,vect_length)));
    //k=k($:-1:1,:);//flip vector, comment/uncomment for MSB/LSB first
endfunction

function plot_signal_and_spectrum(t,signal,f,spectrum)
    n=size(f,'*')
    clf()
    subplot(211)
    a=gca();
    a.data_bounds =[0 -0.01;total_length 1.01];
    plot(t,signal)
    subplot(212)
    plot(f,abs(spectrum(1:n)),'x')
    a=gca();
    a.data_bounds =[0 -0.1;(0.75*5*8*(1/loop_length)) 400];
endfunction

//////END//OF//FUNCTIONS//////////////

//experiment properties
key_bits = 8;//key width
```

```
key_value_max = (2^key_bits)-1;//max key value to simulate
loop_length=1/1e6;//length of one boot loop, (seconds)
l_keyshift = loop_length/5; //length of the total key shift (seconds)
total_length=loop_length;//total length considered (seconds)

//FFT
sample_rate=1/total_length*2000;//calculate sample rate from number of samples
t = 0:1/sample_rate:total_length;//create time vector
N=size(t,'*'); //calculate actual integer number of samples

bootfreq_result_matrix=[];
for key_value = 0:key_value_max
    disp(msprintf ('Starting calculation for key value %d',key_value));
    current_key_vector = binary_vector(key_value,key_bits);
    bootfreq_component =[];

    //calculate frequency component for boot frequency for different key shifts
    for i = key_bits:-1:1
        //disp(msprintf ('\tGenerating signal and fft for %d bits',i));
        //create time domain serial signal
        signal=serialize_key(t,l_keyshift/key_bits,current_key_vector);
        spectrum=fft(signal);//compute fft of time domain signal
        //save absolute value for boot frequency component
        bootfreq_component(i) = abs(spectrum(2))
        current_key_vector(i) = 0;//set last key bit to zero

        //plot time domain and fft
        //s is real so the fft response is conjugate symmetric
        //and we retain only the first N/2 points
        //f=sample_rate*(0:(N/2))/N; //associated frequency vector
        //plot_signal_and_spectrum(t,signal,f,spectrum);

        //xclick();//wait for click or button press
    end
    //append calculated bootfreq components to result matrix
    bootfreq_result_matrix = cat(2,bootfreq_result_matrix,bootfreq_component);

end

//normalize values for saving
bootfreq_result_matrix = bootfreq_result_matrix/max(bootfreq_result_matrix);
//save results
fprintfMat("./serial_key_bootfreq_components.txt",bootfreq_result_matrix, ..
           "%lf","#This file contains the value of the boot freq. component"+...
           "for different key shifts (rows) and different key values (columns)");

//rescale and plot result matrix for preview in scilab
bootfreq_result_matrix = bootfreq_result_matrix*100
clf();
xset("colormap",jetcolormap(100));
xlabel("key value")
ylabel("number of bits shifted through")
Matplot(bootfreq_result_matrix($:-1:1,:));//we need to flip as the plot inverts the axis
```

BIBLIOGRAPHY

[1] P. Alfke. *FPGA Configuration Guidelines*. XAPP090. Xilinx Corporation. 1997 (cit. on p. 30).

[2] Altera Corporation. *Cyclone IV Device Handbook*. 2016 (cit. on p. 72).

[3] Altera Corporation. *MAX V Device Handbook*. 2017 (cit. on pp. 12, 30).

[4] E. Amini, R. Muydinov, B. Szyszka, and C. Boit. "Backside Protection Structure for Security Sensitive ICs." In: *ISTFA 2017: Proceedings from the 43rd International Symposium for Testing and Failure Analysis*. ASM International, 2017, pp. 279–284 (cit. on pp. 138, 139, 143).

[5] M. Baba, T. Sasaki, M. Yoshita, and H. Akiyama. "Aberrations and Allowances for Errors in a Hemisphere Solid Immersion Lens for Submicron-Resolution Photoluminescence Microscopy." In: *Journal of Applied Physics* 85.9 (1999), pp. 6923–6925 (cit. on pp. 107, 110, 113, 117).

[6] F. Beaudoin, R. Desplats, P. Perdu, and D. Lewis. "Implementing Thermal Laser Stimulation in a Failure Analysis Laboratory." In: *ISTFA 2001: Proceedings from the 27th International Symposium for Testing and Failure Analysis*. ASM International, 2001, pp. 151–160 (cit. on p. 35).

[7] J. Beutler, J. J. Clement, M. A. Miller, J. Stevens, and E. I. Cole Jr. "Visible Light LVP on Ultra-Thinned Substrates." In: *ISTFA 2014: Proceedings from the 40th International Symposium for Testing and Failure Analysis*. ASM International, 2014, p. 110 (cit. on p. 104).

[8] J. Beutler, V. C. Hodges, J. J. Clement, J. Stevens, E. I. Cole Jr, S. Silverman, and R. Chivas. "Visible Light LVP on Bulk Silicon Devices." In: *ISTFA 2015: Proceedings from the 41th International Symposium for Testing and Failure Analysis*. ASM International, 2015, pp. 1–8 (cit. on p. 104).

[9] 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)*. © 2013 IEEE. IEEE, 2013, pp. 17–21. DOI: 10.1109/IPFA.2013.6599119 (cit. on pp. 37–39, 55, 141).

[10] 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)*. © 2016 IEEE. IEEE, 2016, pp. 365–369. DOI: 10.1109/IPFA.2016.7564318 (cit. on p. 135).

[11] C. Boit, H. Lohrke, P. Scholz, A. Beyreuther, U. Kerst, and Y. Iwaki. "Contactless Visible Light Probing for Nanoscale ICs through 10 μm Bulk Silicon." In: *Proceedings of the 35th Annual NANO Testing Symposium (NANOTS)*. Institute of NANO Testing, 2015, pp. 215–221. DOI: 10.14279/depositonce-5949 (cit. on pp. 101, 103, 112).

[12] M. R. Bruce, V. J. Bruce, D. H. Eppes, J. Wilcox, E. I. Cole, P. Tangyunyong, C. F. Hawkins, and R. Ring. "Soft Defect Localization (SDL) in Integrated Circuits Using Laser Scanning Microscopy." In: *The 16th Annual Meeting of the IEEE Lasers and Electro-Optics Society, 2003*. Vol. 2. © 2003 IEEE. IEEE, 2003, pp. 662–663. DOI: 10.1109/LEOS.2003.1252974 (cit. on p. 36).

[13] E. I. J. Cole, P. Tangyunyong, and D. L. Barton. "Backside Localization of Open and Shorted IC Interconnections." In: *1998 IEEE International Reliability Physics Symposium (IRPS)*. © 1998 IEEE. IEEE, 1998, pp. 129–136. DOI: 10.1109/RELPHY.1998.670462 (cit. on p. 36).

[14] E. I. Cole, J. M. Soden, J. L. Rife, D. L. Barton, and C. L. Henderson. "Novel Failure Analysis Techniques Using Photon Probing with a Scanning Optical Microscope." In: *1994 IEEE International Reliability Physics Symposium (IRPS)*. © 1994 IEEE. IEEE, 1994, pp. 388–398. DOI: 10.1109/RELPHY.1994.307808 (cit. on p. 36).

[15] C. Conn and A. R. Barron. "Microparticle Characterization via Confocal Microscopy." In: *Physical Methods in Chemistry and Nano Science*. OpenStax CNX, 2013. URL: http://cnx.org/contents/59efb8e9-1497-4206-a71e-20ce175a3160@4 (cit. on p. 2).

[16] G. Corera. *Iran Shows 'Hacked US Spy Drone' Video Footage*. Feb. 7, 2013. URL: https://www.bbc.co.uk/news/world-middle-east-21373353 (visited on 09/26/2018) (cit. on p. 1).

[17] T. R. Corle and G. S. Kino. *Confocal Scanning Optical Microscopy and Related Imaging Systems*. Academic Press, 1996 (cit. on pp. 101, 102).

[18] A. Csete. *Gqrx SDR*. 2016. URL: http://gqrx.dk (visited on 03/04/2016) (cit. on p. 72).

[19] J. Delvaux, D. Gu, D. Schellekens, and I. Verbauwhede. "Secure Lightweight Entity Authentication with Strong PUFs: Mission Impossible?" In: *Cryptographic Hardware and Embedded Systems – CHES 2014*. Springer, 2014, pp. 451–475. DOI: 10.1007/978-3-662-44709-3_25 (cit. on p. 19).

[20] Ear to Ear Oak. *RTLSDR Scanner*. 2016. URL: http://eartoearoak.com/software/rtlsdr-scanner/ (visited on 03/04/2016) (cit. on p. 72).

[21] F. Ganji, S. Tajik, and J.-P. Seifert. "PAC Learning of Arbiter PUFs." In: *Journal of Cryptographic Engineering* 6 (2015), pp. 249–258 (cit. on p. 21).

[22] B. Gassend, D. Clarke, M. Van Dijk, and S. Devadas. "Silicon Physical Random Functions." In: *Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS)*. ACM, 2002, pp. 148–160. DOI: 10.1145/586110.586132 (cit. on pp. 18, 42, 60).

[23] C. Goupil, W. Seifert, K. Zabrocki, E. Müller, and G. J. Snyder. "Thermodynamics of Thermoelectric Phenomena and Applications." In: *Entropy* 13.8 (2011), pp. 1481–1517. DOI: 10.3390/e13081481 (cit. on p. 38).

[24] M. A. Green. "Self-Consistent Optical Parameters of Intrinsic Silicon at 300K Including Temperature Coefficients." In: *Solar Energy Materials & Solar Cells* 92.11 (2008), pp. 1305–1310. DOI: 10.1016/j.solmat.2008.06.009 (cit. on pp. 58, 105, 106).

[25] 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: Proceedings from the 41th International Symposium for Testing and Failure Analysis*. ASM International, 2015, pp. 47–51 (cit. on p. 101).

[26] Hamamatsu Photonics K.K. *PHEMOS-1000 Emission Microscope C11222-16*. 2017. URL: https://www.hamamatsu.com/resources/pdf/sys/SSMS0003E_PHEMOS1000.pdf (cit. on p. 104).

[27] W. He, J. Breier, S. Bhasin, N. Miura, and M. Nagata. "Ring Oscillator under Laser: Potential of PLL-Based Countermeasure against Laser Fault Injection." In: *2016 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)*. © 2016 IEEE. IEEE, 2016, pp. 102–113. DOI: 10.1109/FDTC.2016.13 (cit. on p. 135).

[28] 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 and Communications Security (CCS)*. ACM, 2013, pp. 733–744. DOI: 10.1145/2508859.2516717 (cit. on p. 2).

[29] D. E. Holcomb, W. P. Burleson, and K. Fu. "Power-Up SRAM State as an Identifying Fingerprint and Source of True Random Numbers." In: *IEEE Transactions on Computers* 58.9 (2009). © 2009 IEEE, pp. 1198–1210. DOI: 10.1109/TC.2008.212 (cit. on p. 42).

[30] Y. Hori, T. Katashita, A. Sasaki, and A. Satoh. "Electromagnetic Side-Channel Attack against 28-nm FPGA Device." In: *Pre-proceedings of WISA*. 2012. URL: https://staff.aist.go.jp/hori.y/articles/hori_wisa2012.pdf (cit. on p. 84).

[31] International Roadmap Committee. *International Technology Roadmap For Semiconductors 2013 - IRC Overview*. 2013. URL: https://www.semiconductors.org/wp-content/uploads/2018/08/2013Overview-1.pdf (cit. on p. 102).

[32] S. B. Ippolito, B. B. Goldberg, and M. S. Ünlü. "High Spatial Resolution Subsurface Microscopy." In: *Applied Physics Letters* 78.26 (2001), pp. 4071–4073. DOI: 10.1063/1.1381574 (cit. on p. 115).

[33] 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." In: *IEEE Transactions on Device and Materials Reliability* 7.1 (2007). © 2007 IEEE, pp. 19–30 (cit. on p. 58).

[34] 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, publication pending (cit. on pp. 35, 40–43).

[35] C. Klingshirn. "Interaction of Light with Matter." In: *Semiconductor Optics*. Springer, 2007, pp. 37–72. DOI: 10.1007/978-3-540-38347-5_3 (cit. on pp. 103, 105).

[36] M. Lang, E. Aspnes, and T. D. Milster. "Geometrical Analysis of Third-Order Aberrations for a Solid Immersion Lens." In: *Optics Express* 16.24 (2008), pp. 20008–20028 (cit. on p. 107).

[37] J. W. Lee, D. Lim, B. Gassend, G. E. Suh, M. van Dijk, and S. Devadas. "A Technique to Build a Secret Key in Integrated Circuits for Identification and Authentication Applications." In: *2004 Symposium on VLSI Circuits. Digest of Technical Papers*. © 2004 IEEE. IEEE, 2004, pp. 176–179. DOI: 10.1109/VLSIC.2004.1346548 (cit. on p. 21).

[38] D. C. Litzenberger. *Python Cryptography Toolkit (pycrypto)*. 2017. URL: https://pypi.org/project/pycrypto/ (visited on 07/19/2018) (cit. on p. 86).

[39] H. Lohrke, H. Zöllner, P. Scholz, S. Tajik, C. Boit, and J. P. Seifert. "Visible light techniques in the FinFET era: Challenges, threats and opportunities." In: *2017 IEEE 24th International Symposium on the Physical and Failure Analysis of Integrated Circuits (IPFA)*. © 2017 IEEE. IEEE, 2017, pp. 1–6. DOI: 10.1109/IPFA.2017.8060058 (cit. on pp. 101, 122–124, 135).

[40] 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: Proceedings from the 42nd International Symposium for Testing and Failure Analysis*. ASM International, 2016, pp. 1–6 (cit. on pp. 5, 135).

[41] 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 (cit. on pp. 57, 58, 61, 62, 64, 65, 71, 135).

[42] 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." In: *IACR Transactions on Cryptographic Hardware and Embedded Systems (TCHES)* 2018 (3 2018). Licensed under Creative Commons License CC-BY 4.0, pp. 573–595. DOI: 10.13154/tches.v2018.i3.573-595 (cit. on pp. 35, 38, 52–55, 127–129, 135).

[43] H. Lohrke, P. Scholz, A. Beyreuther, U. Ganesh, E. Uhlmann, S. Kühne, M. Jagodzinski, Y. Iwaki, R. Chivas, S. Silverman, et al. "Contactless Fault Isolation for FinFET Technologies with Visible Light and GaP SIL." In: *ISTFA 2016: Proceedings from the 42nd International Symposium for Testing and Failure Analysis*. ASM International, 2016, pp. 31–38 (cit. on pp. 101, 106, 108, 114–116, 118, 119, 146).

[44] M. Majzoobi, F. Koushanfar, and S. Devadas. "FPGA PUF Using Programmable Delay Lines." In: *2010 IEEE International Workshop on Information Forensics and Security*. © 2010 IEEE. IEEE, 2010, pp. 1–6. DOI: 10.1109/WIFS.2010.5711471 (cit. on p. 20).

[45] S. M. Mansfield and G. S. Kino. "Solid Immersion Microscope." In: *Applied Physics Letters* 57.24 (1990), p. 2615. DOI: 10.1063/1.103828 (cit. on p. 102).

[46] 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 (CCS)*. ACM, 2011, pp. 111–124. DOI: 10.1145/2046707.2046722 (cit. on p. 63).

[47] A. Moradi and T. Schneider. "Improved Side-Channel Analysis Attacks on Xilinx Bitstream Encryption of 5, 6, and 7 Series." In: *Constructive Side-Channel Analysis and Secure Design*. Springer, 2016, pp. 71–87 (cit. on pp. 63, 84).

[48] National Cybersecurity and Communications Integration Center. *Alert IR-ALERT-H-16-056-01*. Feb. 25, 2016. URL: https://ics-cert.us-cert.gov/alerts/IR-ALERT-H-16-056-01 (visited on 09/26/2018) (cit. on p. 1).

[49] D. Nedospasov, C. Helfmeier, J.-P. Seifert, and C. Boit. "Invasive PUF Analysis." In: *2013 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)*. © 2013 IEEE. IEEE, 2013, pp. 30–38. DOI: 10.1109/FDTC.2013.19 (cit. on pp. 37–39, 55, 141).

[50] D. F. Nelson and E. H. Turner. "Electro-Optic and Piezoelectric Coefficients and Refractive Index of Gallium Phosphide." In: *Journal of Applied Physics* 39.7 (1968), pp. 3337–3343. DOI: 10.1063/1.1656779 (cit. on p. 106).

[51] Numato Lab. *Skoll - Kintex 7 FPGA Development Board*. 2017. URL: https://numato.com/skoll-kintex-7-fpga-development-board/ (visited on 05/19/2017) (cit. on p. 84).

[52] L. Onsager. "Reciprocal Relations in Irreversible Processes. I." In: *Physical Review* 37 (4 1931), pp. 405–426. DOI: 10.1103/PhysRev.37.405 (cit. on p. 38).

[53] C. Pagano. "Electro Optical Frequency Modulation on Silicon Integrated Circuits with 1300nm and 1064nm Laser Sources." PhD Thesis. Technische Universität Berlin, 2015 (cit. on p. 119).

[54] R. Pappu, B. Recht, J. Taylor, and N. Gershenfeld. "Physical One-Way Functions." In: *Science* 297.5589 (2002), pp. 2026–2030 (cit. on pp. 18, 42, 60).

[55] P. Perdu, R. Desplats, K. Sanchez, F. Beaudoin, D. Lewis, V. Pouget, A. Douin, and P. Fouillat. "Identification of Some Key Parameters for Photoelectric Laser Stimulation of IC: An Experimental Approach." In: *Proceedings of the 12th International Symposium on the Physical and Failure Analysis of Integrated Circuits (IPFA)*. © 2005 IEEE. IEEE, 2005, pp. 21–26. DOI: 10.1109/IPFA.2005.1469124 (cit. on pp. 5, 131).

[56] E. Peterson. *Leveraging Asymmetric Authentication to Enhance Security-Critical Applications Using Zynq-7000 All Programmable SoCs*. WP468. Xilinx Corporation. 2015 (cit. on pp. 62, 64).

[57] E. Peterson. *Developing Tamper-Resistant Designs with UltraScale and UltraScale+ FPGAs*. XAPP1098. Xilinx Corporation. 2017 (cit. on pp. 54, 137).

[58] K. D. Pham, E. Horta, and D. Koch. "BITMAN: A Tool and API for FPGA Bitstream Manipulations." In: *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017*. © 2017 IEEE. IEEE, 2017, pp. 894–897. DOI: 10.23919/DATE.2017.7927114 (cit. on p. 9).

[59] H. R. Phillip and E. A. Taft. "Kramers-Kronig Analysis of Reflectance Data for Diamond." In: *Physical Review* 136 (5A 1964), A1445–A1448. DOI: 10.1103/PhysRev.136.A1445 (cit. on p. 106).

[60] S. Preibisch, S. Saalfeld, and P. Tomancak. "Globally Optimal Stitching of Tiled 3D Microscopic Image Acquisitions." In: *Bioinformatics* 25.11 (2009), pp. 1463–1465. DOI: 10.1093/bioinformatics/btp184 (cit. on pp. 48, 51, 87).

[61] H. Qin, Y. Cao, D. Markovic, A. Vladimirescu, and J. Rabaey. "SRAM Leakage Suppression by Minimizing Standby Supply Voltage." In: *International Symposium on Signals, Circuits and Systems. Proceedings, SCS 2003.* © 2004 IEEE. IEEE, 2004, pp. 55–60. DOI: 10.1109/ISQED.2004.1283650 (cit. on p. 32).

[62] C. Roscian, A. Sarafianos, J. M. Dutertre, and A. Tria. "Fault Model Analysis of Laser-Induced Faults in SRAM Memory Cells." In: *2013 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)*. © 2013 IEEE. IEEE, 2013, pp. 89–98. DOI: 10.1109/FDTC.2013.17 (cit. on p. 8).

[63] J. A. Rowlette and T. M. Eiles. "Critical Timing Analysis in Microprocessors Using Near-IR Laser Assisted Device Alteration (LADA)." In: *International Test Conference, 2003. Proceedings. ITC 2003.* Vol. 1. © 2003 IEEE. IEEE, 2003, pp. 264–273. DOI: 10.1109/TEST.2003.1270848 (cit. on p. 36).

[64] U. Rührmair, F. Sehnke, J. Sölter, G. Dror, S. Devadas, and J. Schmidhuber. "Modeling Attacks on Physical Unclonable Functions." In: *Proceedings of the 17th ACM Conference on Computer and Communications Security (CCS)*. ACM, 2010, pp. 237–249. DOI: 10.1145/1866307.1866335 (cit. on p. 21).

[65] U. Rührmair, X. Xu, J. Sölter, A. Mahmoud, M. Majzoobi, F. Koushanfar, and W. Burleson. "Efficient Power and Timing Side Channels for Physical Unclonable Functions." In: *Cryptographic Hardware and Embedded Systems – CHES 2014*. Springer, 2014, pp. 476–492 (cit. on p. 21).

[66] D. P. Sahoo, S. Patranabis, D. Mukhopadhyay, and R. S. Chakraborty. "Fault Tolerant Implementations of Delay-Based Physically Unclonable Functions on FPGA." In: *2016 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)*. © 2016 IEEE. IEEE, 2016, pp. 87–101 (cit. on p. 135).

[67] D. Samyde, S. Skorobogatov, R. Anderson, and J.-J. Quisquater. "On a New Way to Read Data from Memory." In: *First International IEEE Security in Storage Workshop, 2002. Proceedings*. © 2002 IEEE. IEEE, 2002, pp. 65–69 (cit. on p. 37).

[68] K. Sanchez, R. Deplats, F. Beaudoin, P. Perdu, D. Lewis, P. Vedagarbha, and G. Woods. "Delay Variation Mapping Induced by Dynamic Laser Stimulation." In: *2005 IEEE International Reliability Physics Symposium (IRPS)*. © 2005 IEEE. IEEE, 2005, pp. 305–311. DOI: 10.1109/RELPHY.2005.1493103 (cit. on p. 131).

[69] R. Schlangen, R. Leihkauf, U. Kerst, T. Lundquist, P. Egger, and C. Boit. "Physical Analysis, Trimming and Editing of Nanoscale IC Function with Backside FIB Processing." In: *Microelectronics Reliability* 49.9 (2009), pp. 1158–1164. DOI: https://doi.org/10.1016/j.microrel.2009.06.048 (cit. on p. 106).

[70] J.-M. Schmidt and M. Hutter. "Optical and EM Fault-Attacks on CRT-Based RSA: Concrete Results." In: *Austrochip 2007: Proceedings of the 15th Austrian Workshop on Microelectronics*. Verlag der Technischen Universität Graz, 2007, pp. 61–67 (cit. on p. 5).

[71] A. Schönau. "Ermittlung der Beziehung zwischen physikalischer Position und logischer Adressierung eines SRAM-Speichers durch thermische Laserstimulation." Bachelor's Thesis. Technische Universität Berlin, 2016 (cit. on pp. 44, 45, 47, 48).

[72] Semicaps PTE Limited. *Semicaps ARSIL - World First N.A. 3.3 Aplanatic RSIL*. 2014. URL: http://www.semicaps.com/SEMICAPS%20ARSIL.pdf (cit. on p. 104).

[73] S. Skorobogatov. "Optically Enhanced Position-Locked Power Analysis." In: *Cryptographic Hardware and Embedded Systems – CHES 2006*. Springer, 2006, pp. 61–75 (cit. on p. 37).

[74] S. P. Skorobogatov and R. J. Anderson. "Optical Fault Induction Attacks." In: *Cryptographic Hardware and Embedded Systems – CHES 2002*. Springer, 2002, pp. 2–12 (cit. on p. 5).

[75] G. E. Suh and S. Devadas. "Physical Unclonable Functions for Device Authentication and Secret Key Generation." In: *Proceedings of the 44th Annual Design Automation Conference*. ACM, 2007, pp. 9–14. DOI: 10.1145/1278480.1278484 (cit. on pp. 21, 24).

[76] B. Sunar, W. J. Martin, and D. R. Stinson. "A Provably Secure True Random Number Generator with Built-In Tolerance to Active Attacks." In: *IEEE Transactions on Computers* 56.1 (2007). © 2007 IEEE, pp. 109–119. DOI: 10.1109/TC.2007.250627 (cit. on p. 25).

[77] P. Swierczynski, A. Moradi, D. Oswald, and C. Paar. "Physical Security Evaluation of the Bitstream Encryption Mechanism of Altera Stratix II and Stratix III FPGAs." In: *ACM Transactions on Reconfigurable Technology and Systems* 7.4 (2014), 34:1–34:23. DOI: 10.1145/2629462 (cit. on p. 63).

[78] S. Tajik, J. Fietkau, H. Lohrke, J. P. Seifert, and C. Boit. "PUFMon: Security Monitoring of FPGAs Using Physically Unclonable Functions." In: *2017 IEEE 23rd International Symposium on On-Line Testing and Robust System Design (IOLTS)*. © 2017 IEEE. IEEE, 2017, pp. 186–191. DOI: 10.1109/IOLTS.2017.8046216 (cit. on pp. 130–135).

[79] S. Tajik, H. Lohrke, F. Ganji, J. P. Seifert, and C. Boit. "Laser Fault Attack on Physically Unclonable Functions." In: *2015 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)*. © 2015 IEEE. IEEE, 2015, pp. 85–96. DOI: 10.1109/FDTC.2015.19 (cit. on pp. 5, 11–14, 19, 21, 22, 24, 25, 135).

[80] S. Tajik, D. Nedospasov, C. Helfmeier, J. P. Seifert, and C. Boit. "Emission Analysis of Hardware Implementations." In: *2014 17th Euromicro Conference on Digital System Design (DSD)*. © 2014 IEEE. IEEE, 2014, pp. 528–534. DOI: 10.1109/DSD.2014.64 (cit. on pp. 2, 9).

[81] 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. 57, 82, 86–93, 96, 97, 135).

[82] Texas Instruments Incorporated. *MSP430x5xx and MSP430x6xx Family User's Guide*. SLAU208Q. 2018 (cit. on p. 43).

[83] W. R. Tonti. "eFuse Design and Reliability." In: *2008 IEEE International Integrated Reliability Workshop Final Report*. © 2008 IEEE. IEEE, 2008, p. 114 (cit. on p. 62).

[84] S. M. Trimberger and J. J. Moore. "FPGA Security: Motivations, Features, and Applications." In: *Proceedings of the IEEE* 102.8 (2014). © 2014 IEEE, pp. 1248–1265. DOI: 10.1109/JPROC.2014.2331672 (cit. on p. 86).

[85] I. A. Tschinibaew. "Segmentierung und Analyse der Messdaten von thermisch laserstimulierten SRAM-Zellen zur Erkennung der Speicherzustände." Bachelor's Thesis. Technische Universität Berlin, 2016 (cit. on pp. 44, 45, 48).

[86] E. Uhlmann, M. Jagodzinski, J. Polte, S. Kühne, and P. Scholz. "Ultra-Precision Machined Gallium Phosphide Solid Immersion Micro Lenses for Aperture Magnification in Measurement Sys-

tems." In: *Proceedings of the 2016 European Optical Society Bi-Annual Meeting (EOSAM)*. EOS, 2016 (cit. on p. 108).

[87] S. Wang, M. Zhan, G. Wang, H. Xuan, W. Zhang, C. Liu, C. Xu, Y. Liu, Z. Wei, and X. Chen. "4H-SiC: A New Nonlinear Material for Midinfrared Lasers." In: *Laser & Photonics Reviews* 7.5 (2013), pp. 831–838. DOI: 10.1002/lpor.201300068 (cit. on p. 106).

[88] E. W. Weisstein. *Fourier Series. From MathWorld—A Wolfram Web Resource*. 2018. URL: http://mathworld.wolfram.com/FourierSeries.html (visited on 07/19/2018) (cit. on p. 82).

[89] J. G. van Woudenberg, M. F. Witteman, and F. Menarini. "Practical Optical Fault Injection on Secure Microcontrollers." In: *2011 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)*. © 2011 IEEE. IEEE, 2011, pp. 91–99 (cit. on p. 5).

[90] L. Wouters. *Fast, Furious and Insecure: Passive Keyless Entry and Start in Modern Supercars*. Sept. 10, 2018. URL: https://www.esat.kuleuven.be/cosic/fast-furious-and-insecure-passive-keyless-entry-and-start-in-modern-supercars/ (visited on 09/26/2018) (cit. on p. 1).

[91] Xilinx Corporation. *7 Series FPGAs Configuration User Guide*. UG470. 2016 (cit. on pp. 85, 86, 90, 91, 93).

[92] Xilinx Corporation. *7 Series FPGAs GTX/GTH Transceivers*. UG476. 2016 (cit. on p. 88).

[93] ZEISS Archiv, Carl Zeiss AG. *Technological Milestones of Carl Zeiss*. 2018. URL: https://www.zeiss.com/corporate/int/history/technological-milestones/microscopy.html (visited on 08/08/2018) (cit. on p. 111).

[94] K. Zetter. *Inside the Cunning, Unprecedented Hack of Ukraine's Power Grid*. Mar. 3, 2016. URL: https://www.wired.com/2016/03/inside-cunning-unprecedented-hack-ukraines-power-grid/ (visited on 09/26/2018) (cit. on p. 1).

[95] H. Zöllner, T. Brömel, and B. Voß. "Generating SEL and SEU with a Class 1 Laser Setup - Extensions and Further Investigations Relating Applicability." In: *ARCS Workshop 2018; 31th International Conference on Architecture of Computing Systems*. VDE, 2018, pp. 1–3 (cit. on p. 124).

[96] H. Zöllner, M. Hupka, H. Preußer, and B. Voß. "Generating SEL and SEU with a Class 1 Laser Setup." In: *2016 16th European Conference on Radiation and Its Effects on Components and Systems (RADECS)*. © 2016 IEEE. IEEE, 2016, pp. 1–3. DOI: 10.1109/RADECS.2016.8093163 (cit. on p. 124).