Technische Universität Berlin

Master Thesis Defense

Monitoring physical integrity of FPGAs using Physically Unclonable Functions

Julian Fietkau

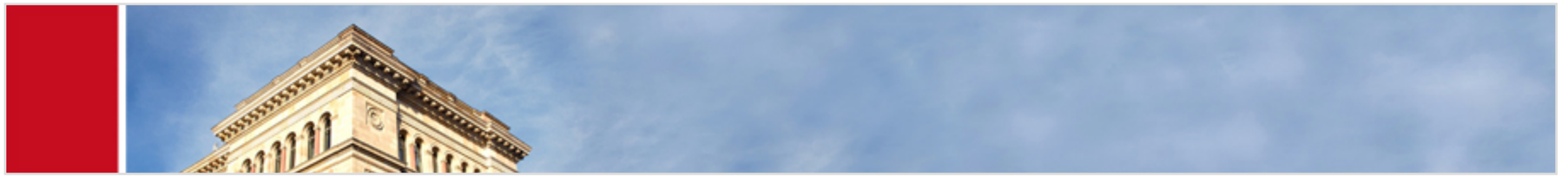
fietkau@campus.tu-berlin.de

Student number: 370062

Supervised by

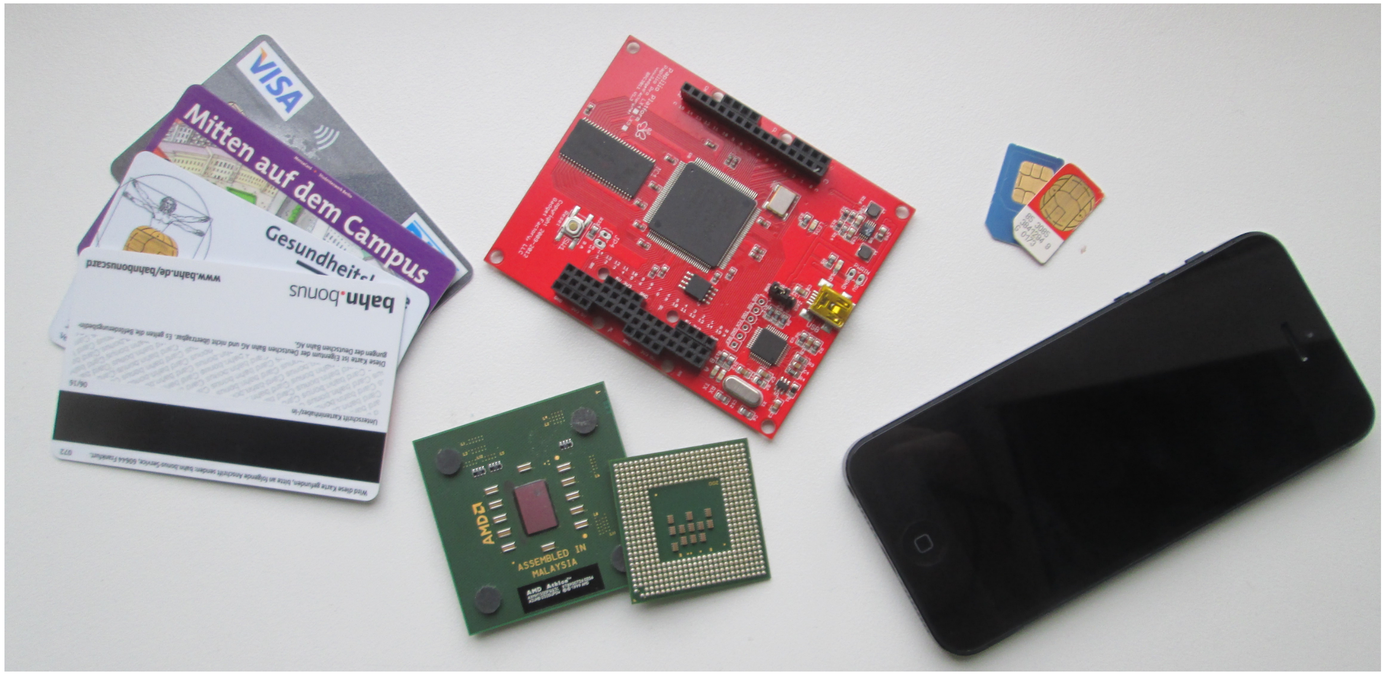
Prof. Dr. Jean-Pierre Seifert

Shahin Tajik



Technische Universität Berlin

Introduction >> Introduction



Scope

- Electronic devices contain **crypto. keys** and **intellectual property** that represents valuable targets for adversaries e.g. sim cards, wireless keys, game consoles smartphones and also FPGAs

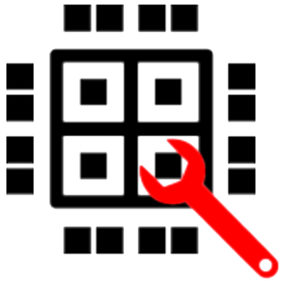
- Several attacks are known to extract them

Technische Universität Berlin

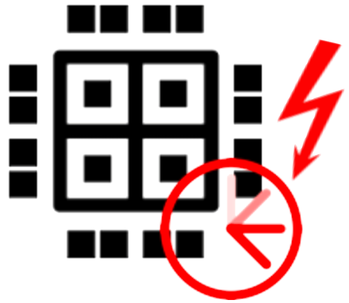
Background >> Threats and Physical Attacks

- **Passive Attacks:** Device operated within specification, secrets are revealed by observing physical properties
  e.g. Side Channel Attacks, Protocol attacks, Laser Scanning, Photonic emission analysis, …

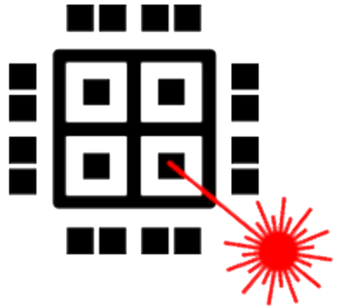
- **Active Attacks:** Device, inputs, or environment is manipulated to make the device behave abnormally
  e.g. Clock & Voltage Glitching, Reconfiguration, Fault Injections, Circuit Edits, Wiretapping, …



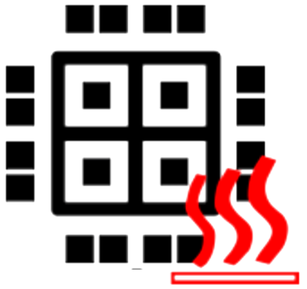
Hardware Trojans
Malicious Logic



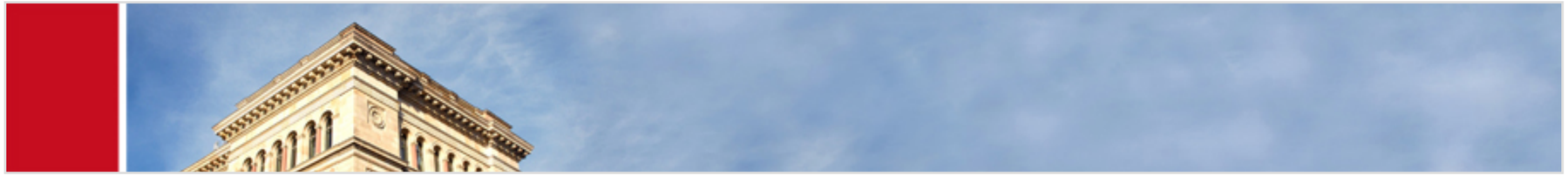
Glitching Attacks
Manipulate execution flow



Laser Scanning
Reversing Design



Temperature Attacks
Fault analysis



Technische Universität Berlin

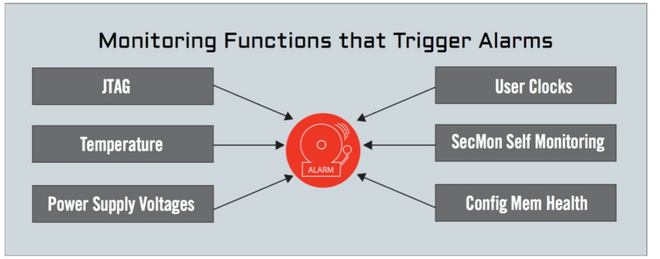
Background >> Vendor Countermeasures

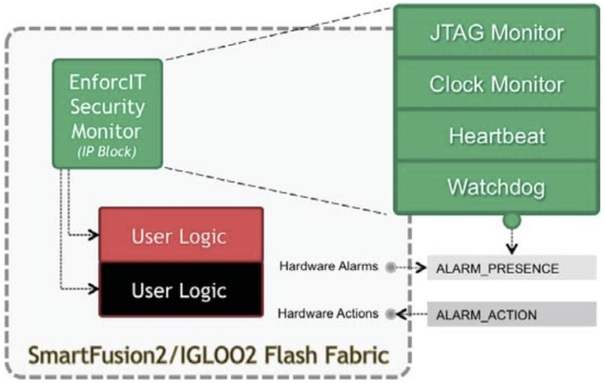
- Create a **Soft-IP core** for the customer (SecMon, Supervisor IP, EnforcIT)

  Fully **placed-and-routed design file** that is added to an existing design

  Utilize **detectors build into the FPGA** to create passive & active security features:

  - **Tamper protections:** Monitor system conditions and trigger alarm
  - **Tamper responses**: React on incidents e.g. erase config., secrets or do lockdown



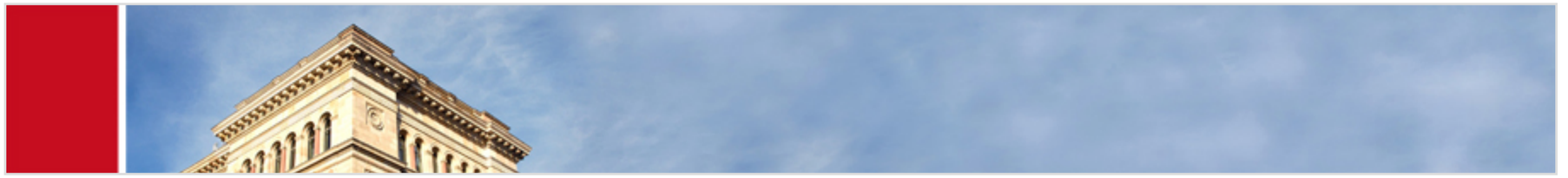


Problems:

! Detectors / mitigation logic can be **spoofed, thwarted, bypassed**

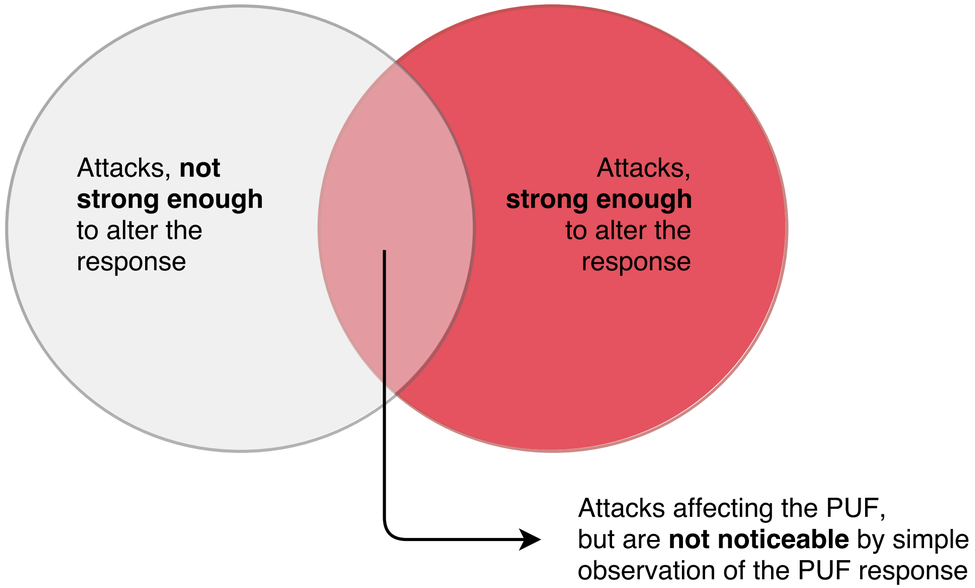
! Do not solve problem at the root: insecure key storage technology

Xilinx	Altera	Microsemi
SecMon	Supervisor IP	EnforcIT Security Monitor
Virtex-5,6,7, Spartan-6, Zynq	Arria V, Stratix V	SmartFusion2, IGLOO2



Technische Universität Berlin

Design >> Concept



Physically Uncloneable Functions can already …

… generate & store secrets within device intrinsics

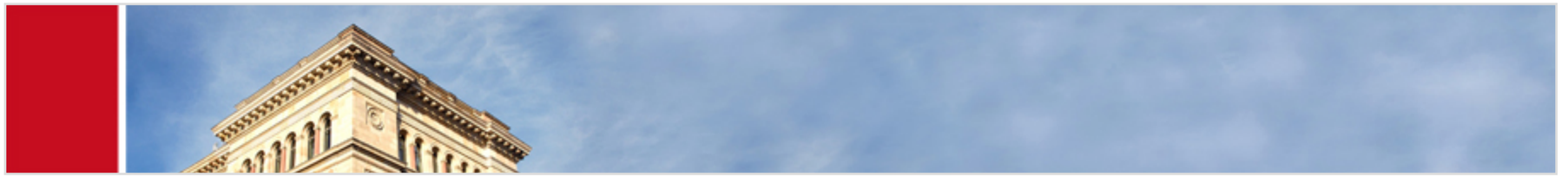
… authenticate a device

“PUFs become suspicious when attacked”

- Can not be spoofed or tampered with… (in theory)
- PUFs have some kind of physical sensitivity

Try to build a monitor based on PUFs

… link detection capabilities to intrinsic device properties

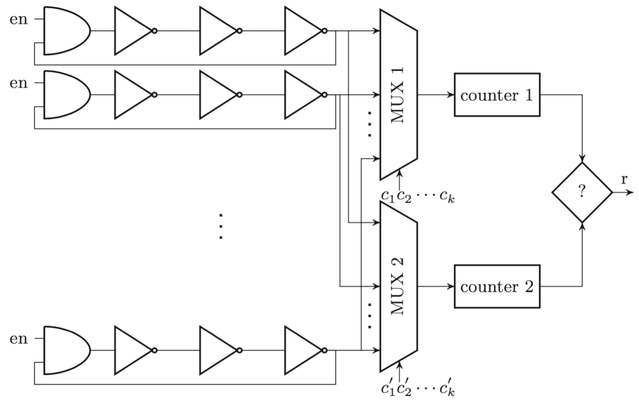


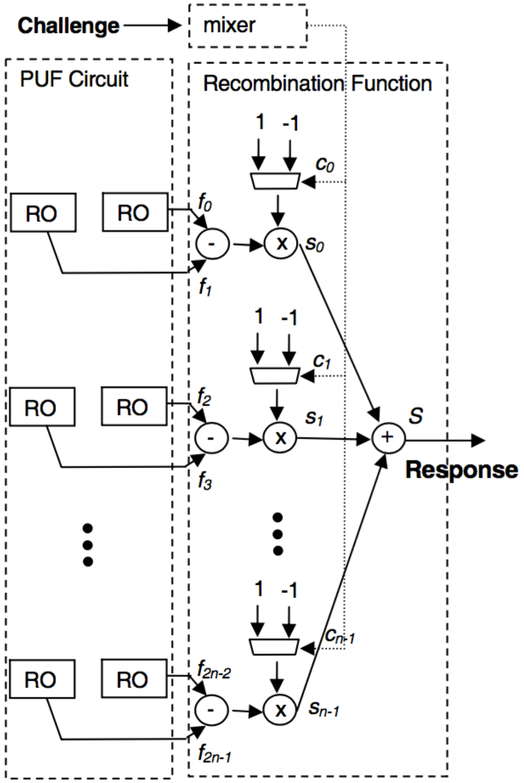
Technische Universität Berlin

Design >> Choose Your Weapons

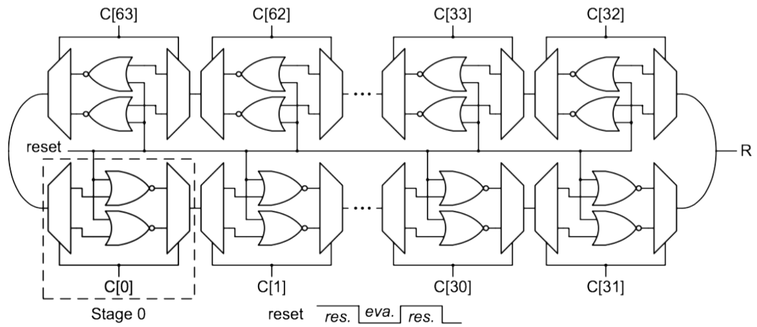
RO Sum PUF

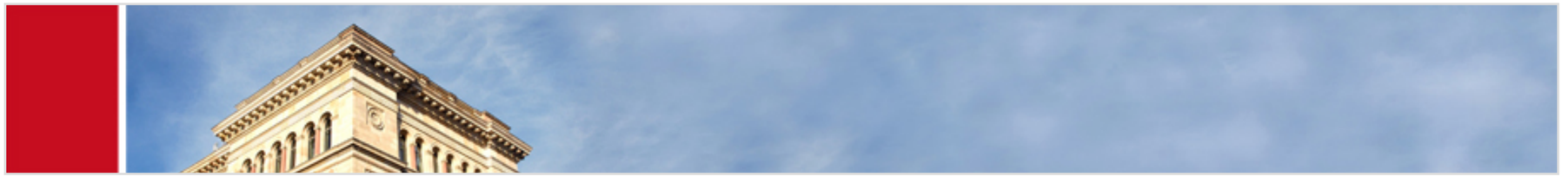
RO PUF





BR-PUF

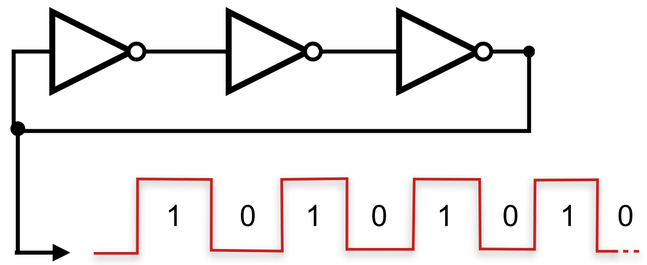




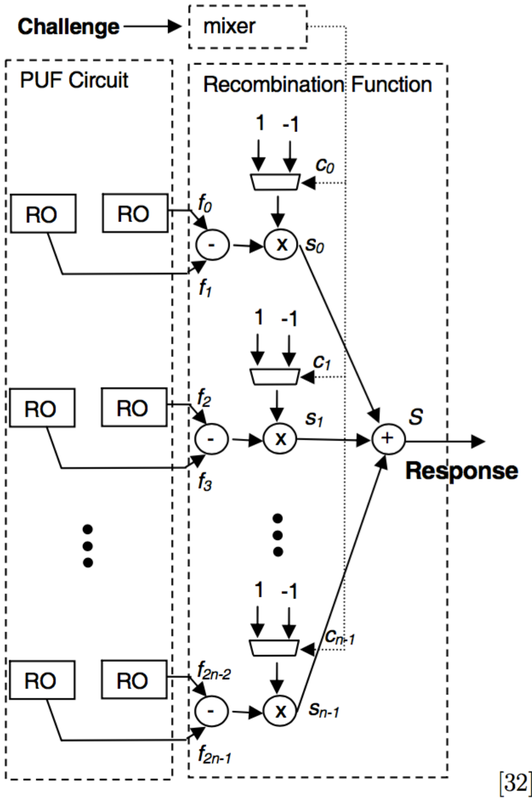
Technische Universität Berlin

Design >> RO Sum PUF

Ring Oscillator

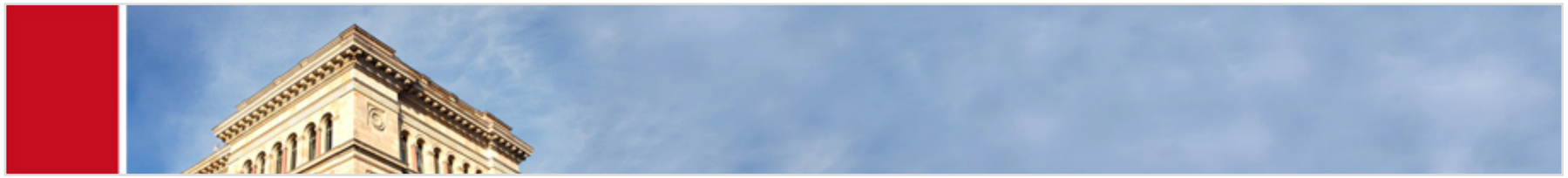


RO Sum PUF



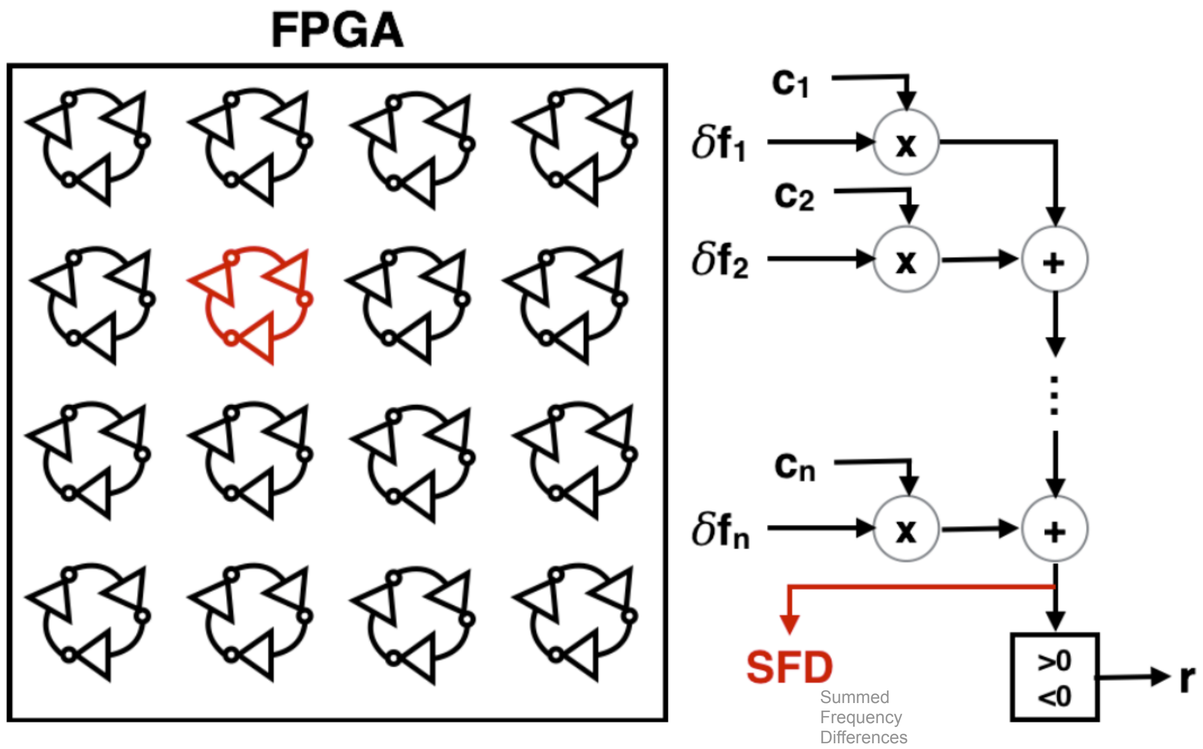
- **64 pairs** of ROs with **counters**
- **Enable** and **sample them** (e.g. 64 clkcyc)
- Compute the **difference** of their counters: **δf**
- **Multiply δf** with 1 or -1 depending on challenge bit (64 bit challenge)
- **Accumulate** all **signed δf's** to generate the **real-values response S**
- **Thresholding**
  - response = 0, when S >= 0
  - response = 1, when S < 0

[32] Meng-Day Yu, Srinivas Devadas. Recombination of Physical Unclonable Functions. *35th Annual GOMACTech Conference*, pages 1–4, March 2010.



Technische Universität Berlin

Design >> The Approach

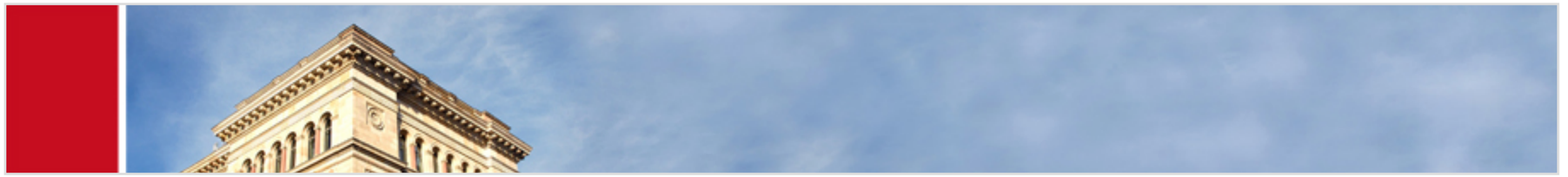


- **Distribute the PUF ROs** on the FPGA

- **Enrolment** (secure environment)
  Take a **set of challenges** (100) and store their real-valued responses

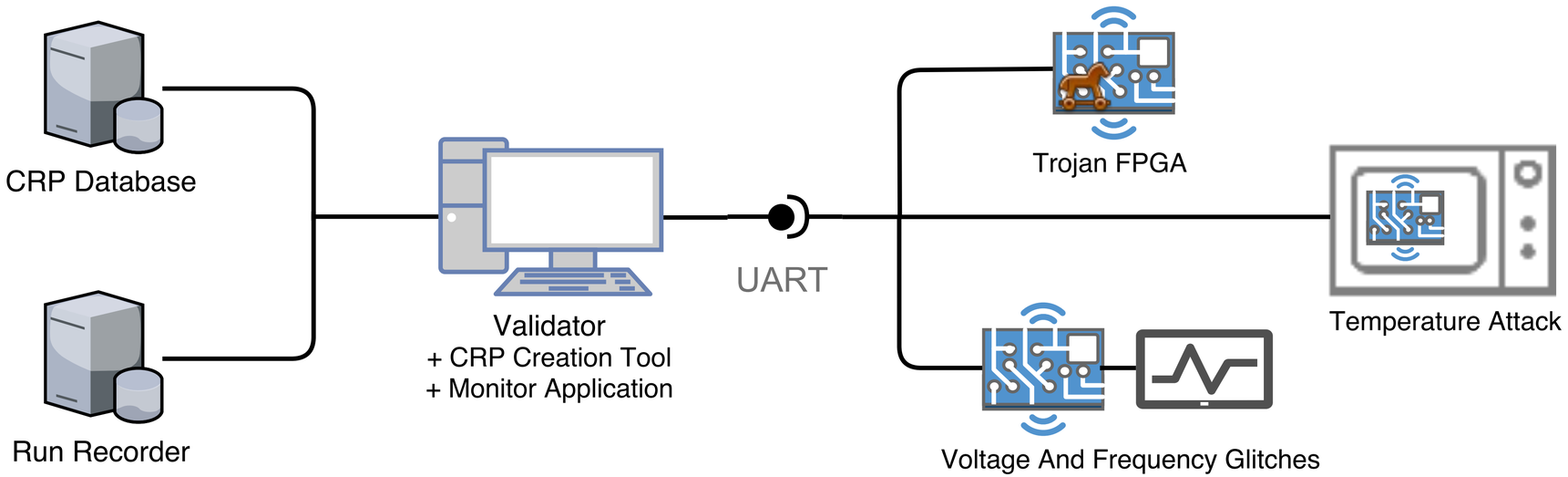
- **Monitoring**
  Reevaluate the **challenges** again and again by using **outliner analysis**:
  + Min/Max or Mean ± {1,2,3} σ

  … and **noise reduction** techniques:
  + Majority Voting (Best of 5)
  + Detection Threshold (20%)



Technische Universität Berlin

Design >> Experimental Architecture

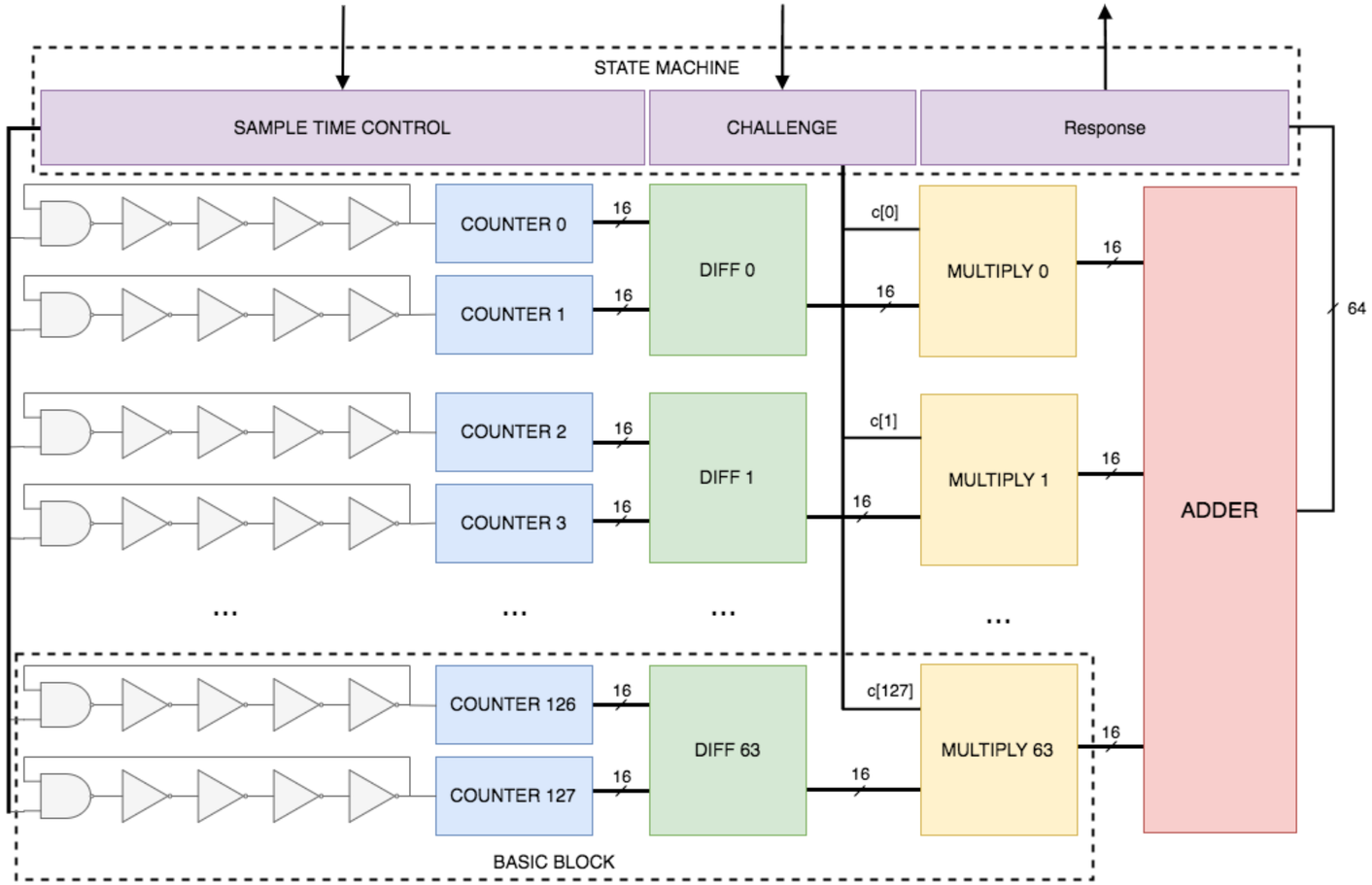


Software Implementation (Python)

Hardware Implementation (Verilog)

Technische Universität Berlin

Implementation >> Hardware



Best Results

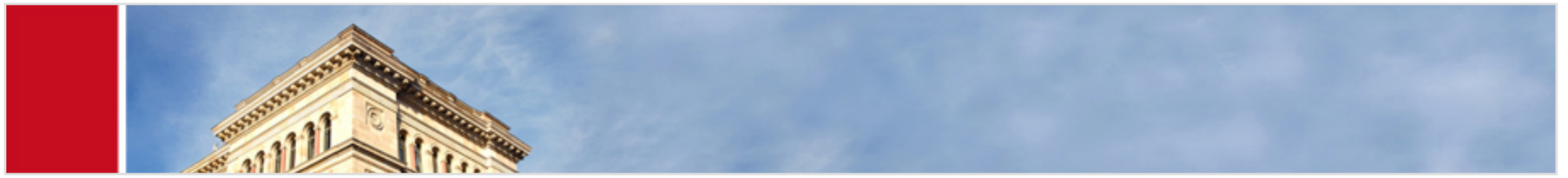
+ ROs with 5 stages
+ 16 bit counter
+ Sample time 64 - 128 clock cycles (50 MHz)
+ Line-by-line placement

Scaleable

+ Useful during experiments

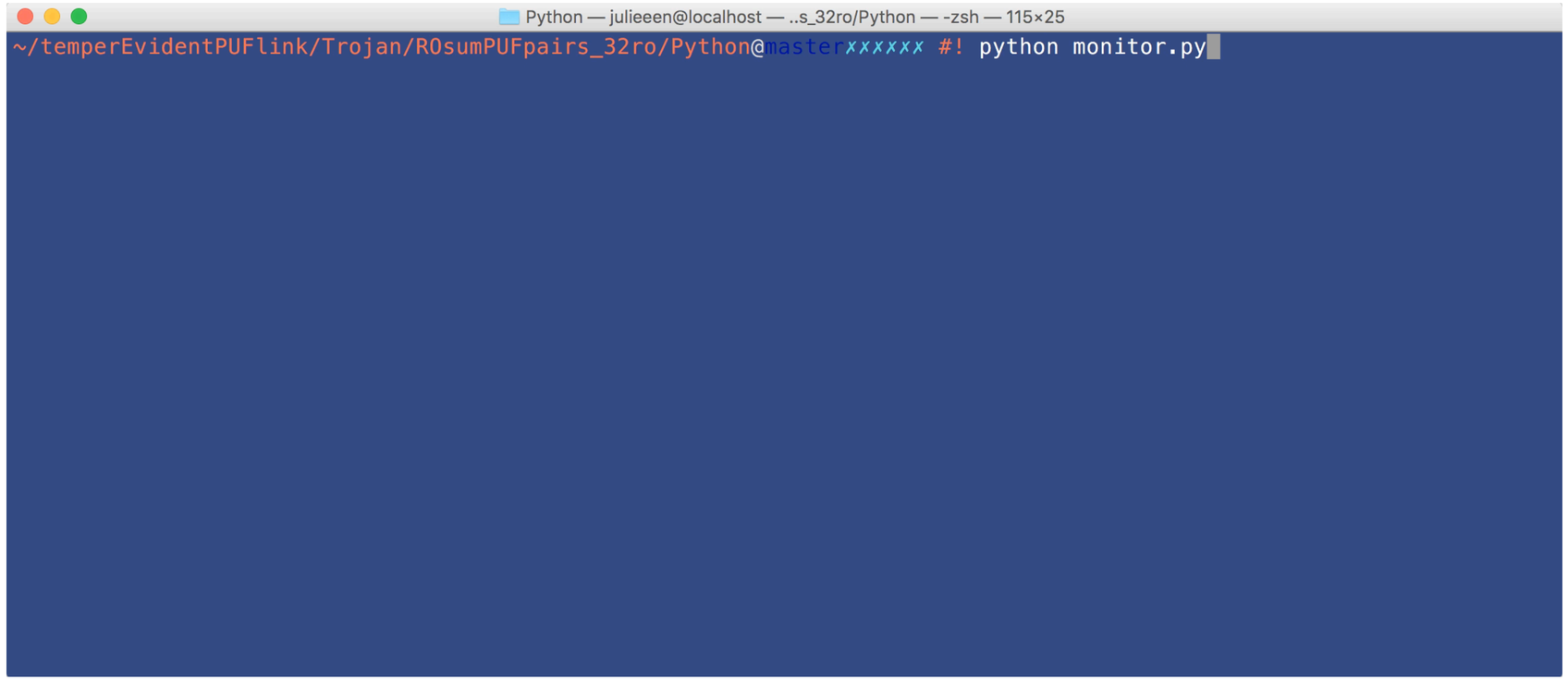
Metric	32 ROs	64 ROs	128 ROs
Logic Cells	2030	3962	7838
Logic Registers	604	1116	2140
LUT-Only LCs	1418	2825	5662
Register-Only LCs	21	49	92
LUT/Register LCs	591	1088	2084

Table 1: Resource utilisation of RO Sum PUF implementation



Technische Universität Berlin

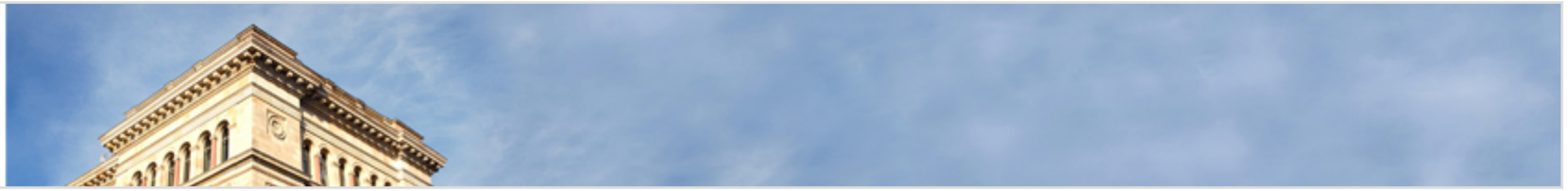
Implementation >> Software



- **training.py**
  + iterate challenges
  + store responses

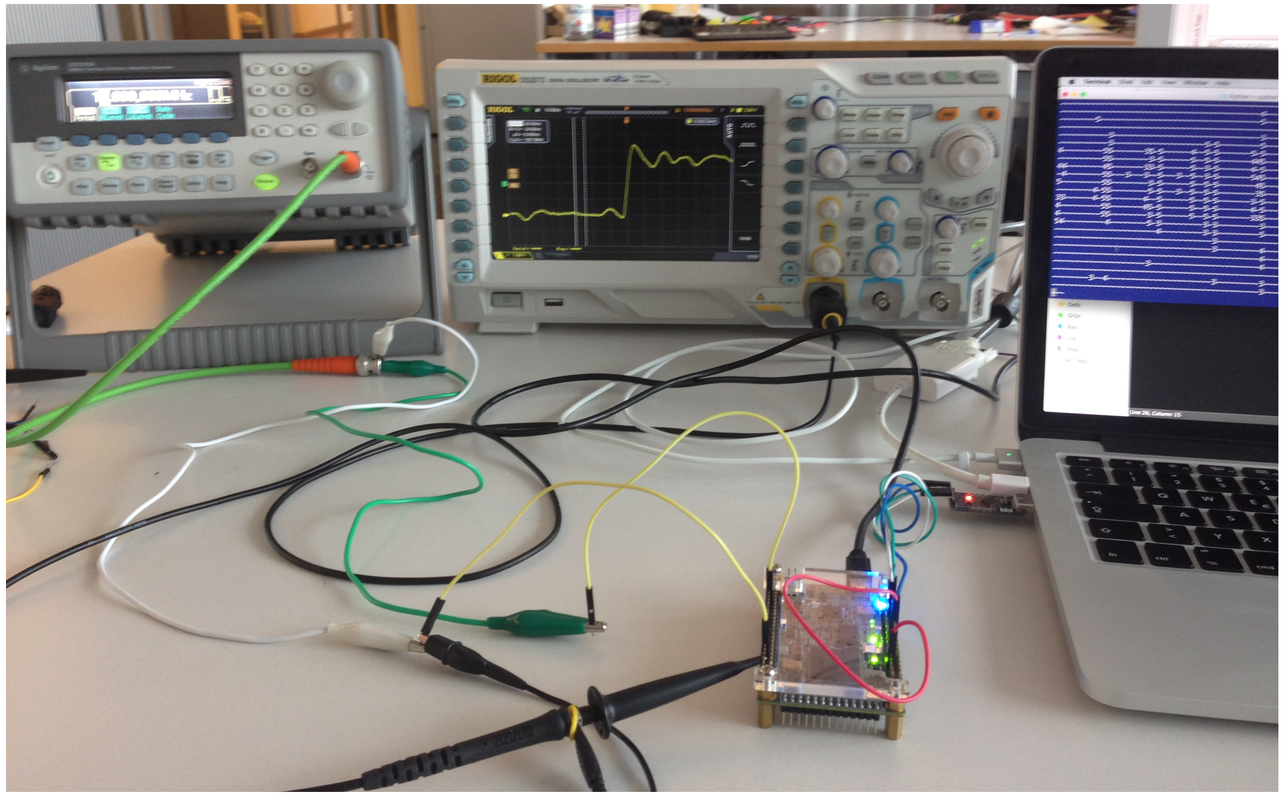
- **monitor.py**
  + iterate challenges
  + evaluate responses
  + graphical feedback

- **replay.py**
  + redo monitoring runs for experimental purposes



Technische Universität Berlin

Evaluation >> Overall Setup



Five experimental cases

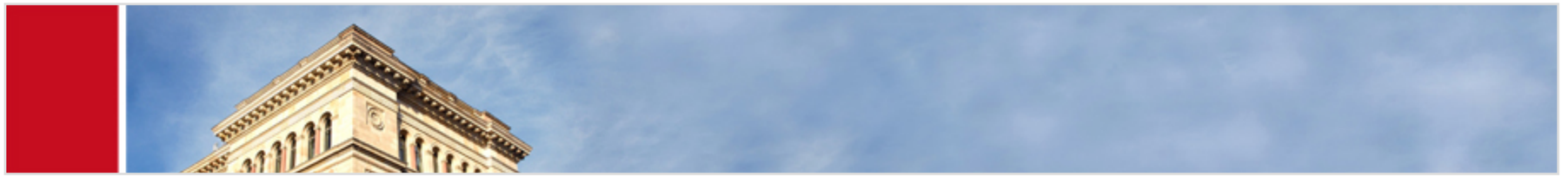
- Hardware Trojan / Reconfiguration
- System clock manipulation
- Voltage manipulation
- Laser scanning
- Temperature manipulation

Procedure

1 - Create unbiased database >> training.py

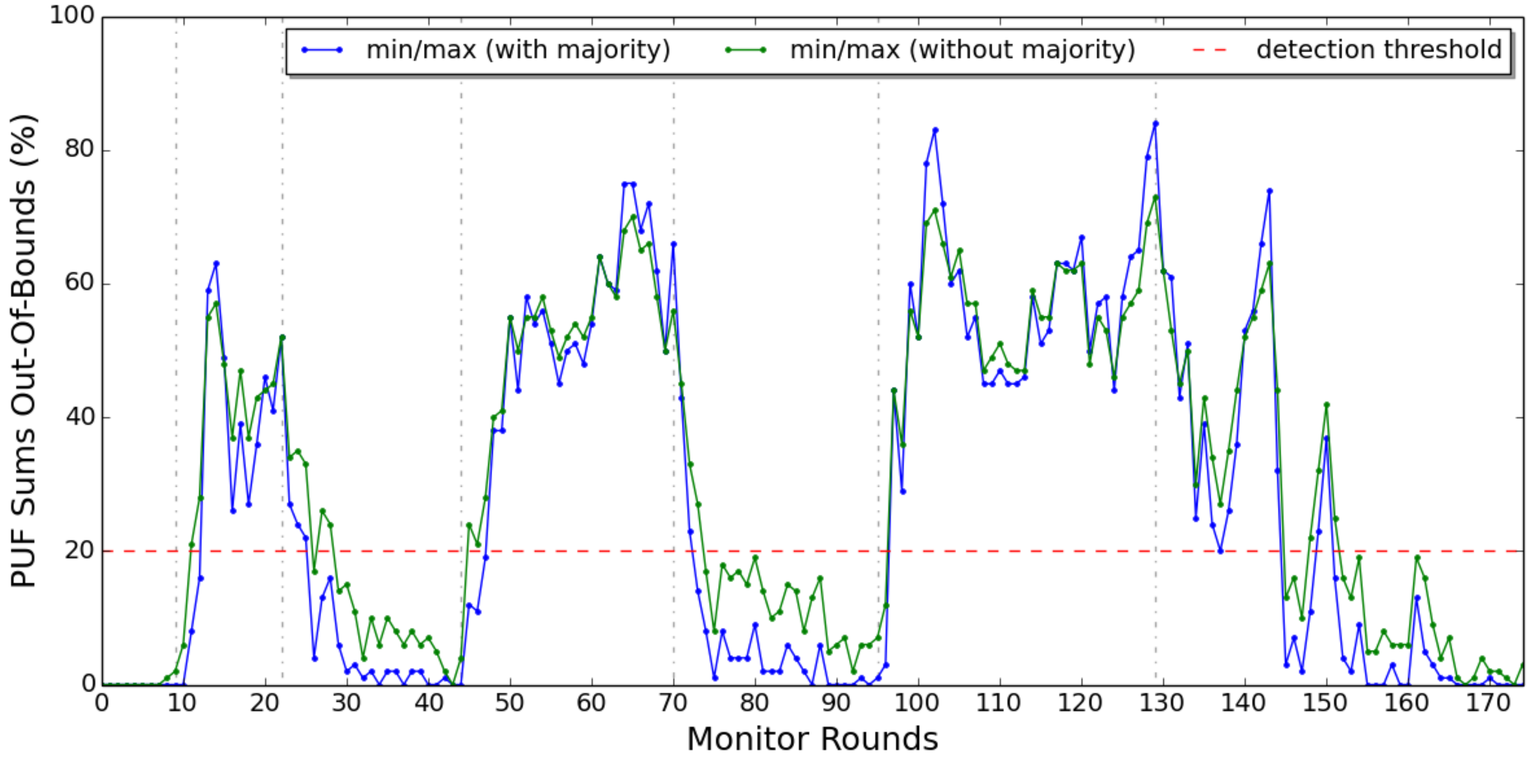
2 - Monitor & attack the device >> monitor.py

3 - Postevaluation >> replay.py



Technische Universität Berlin

Evaluation >> Hardware Trojans / Reconfiguration



Setup

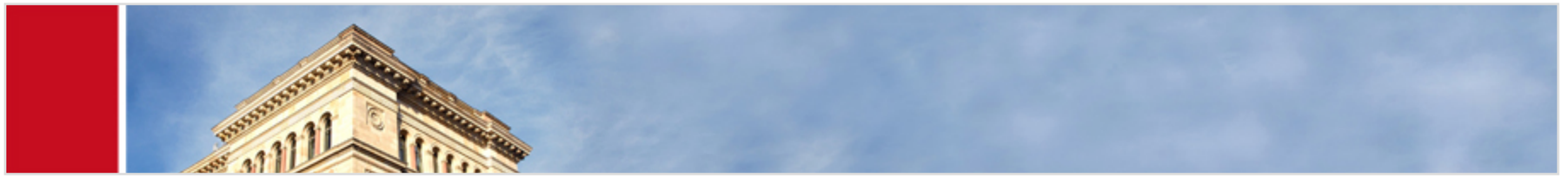
- LFSR Trojan Model
  Clocked Shift-Reg (+feedback)
- Running AES in the background

←

3x Trojan activation

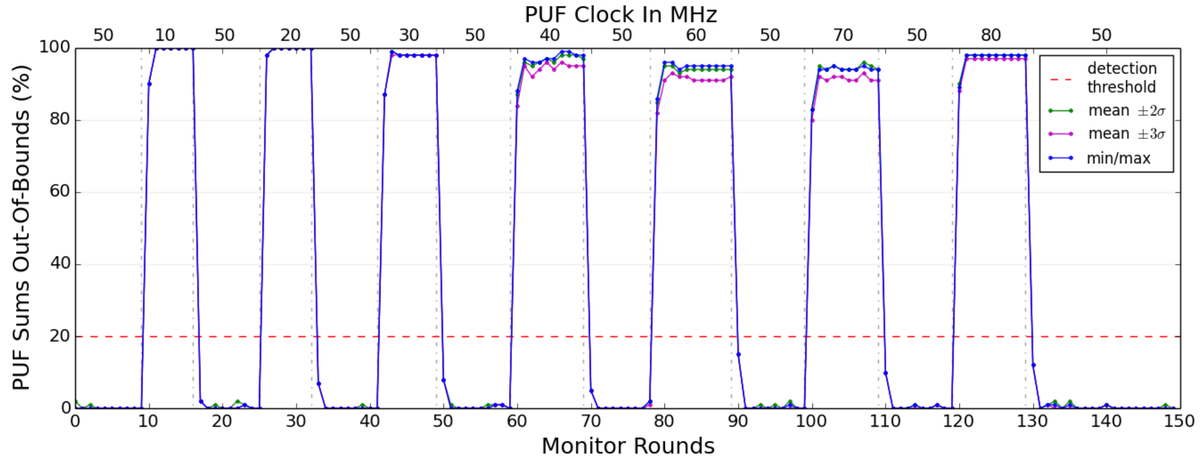
- Require several LABs of malicious logic for detection (> 3 LABs)

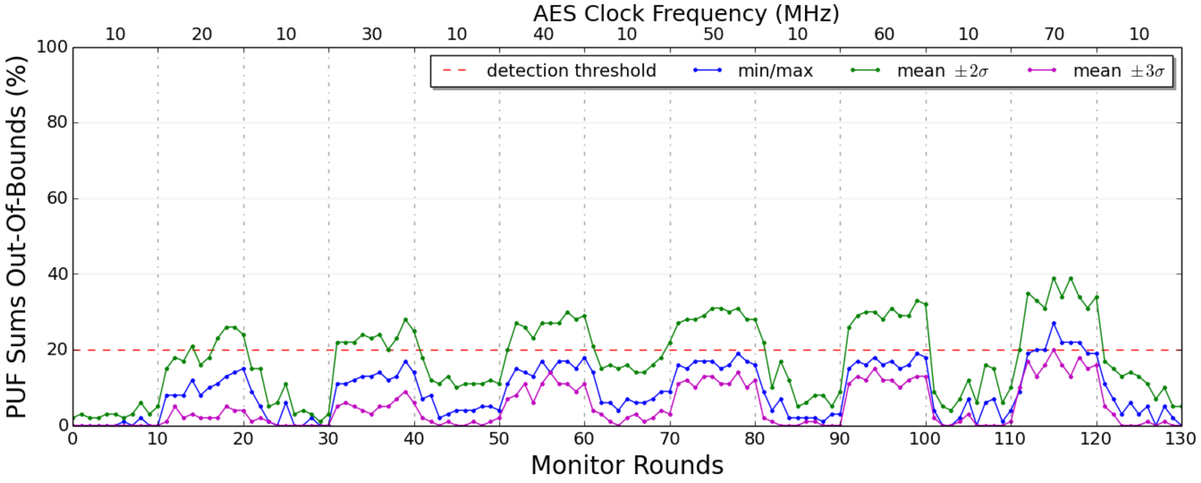
Detectable, when malicious logic has decent size and activity



Technische Universität Berlin

Evaluation >> System Clock Manipulation





Setup

- Detach internal system clock and attach a frequency generator

← **Direct Clock Manipulation** (Unbiased 50 MHz)

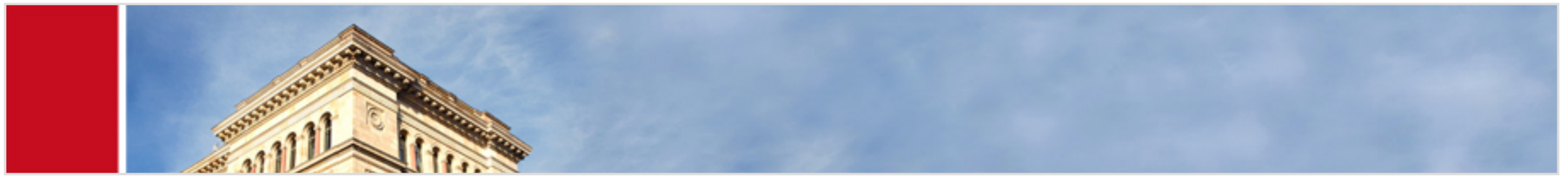
- Apply f = 10, 20, 30, 40, 60, 70, 80 MHz
- Directly affect the PUF sample time unit

← **Indirect Clock Manipulation** (Unbiased 10 MHz)

- Isolated target or multiple clock networks
- Apply 2x, 3x, 4x, 5x, 6x, 7x the basic clock

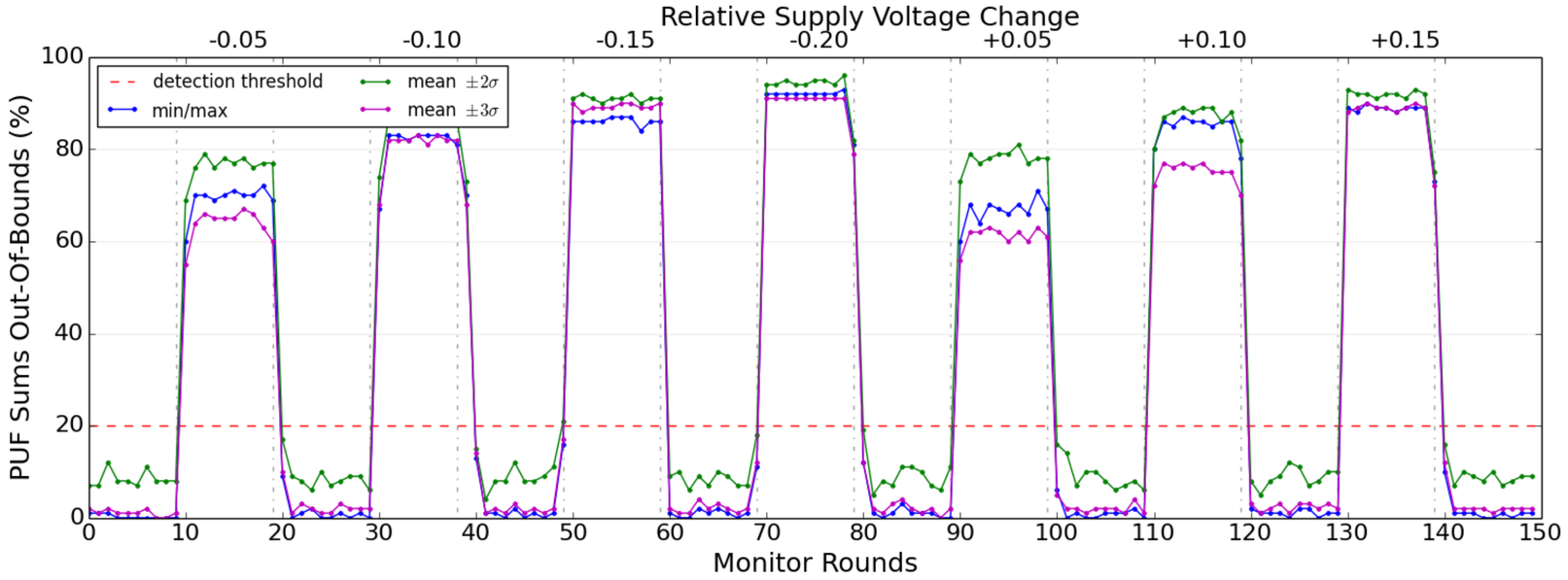
Detectable, as long as PUF is attached to clock

But, Glitch time << Sample time



Technische Universität Berlin

Evaluation >> Supply Voltage Manipulation



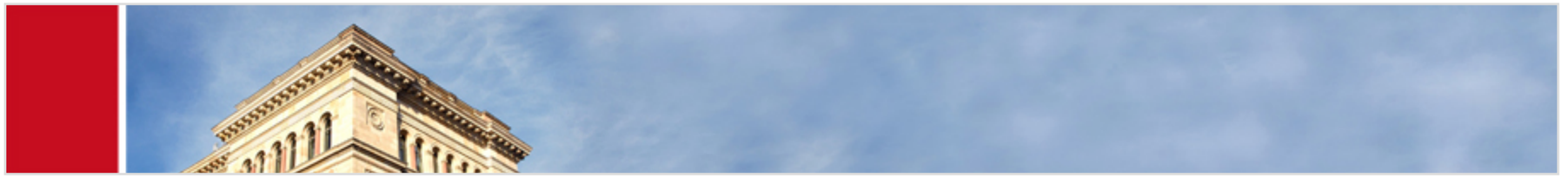
Setup

- Detach device voltage line, attach manual power supply

← **Voltage Manipulation**

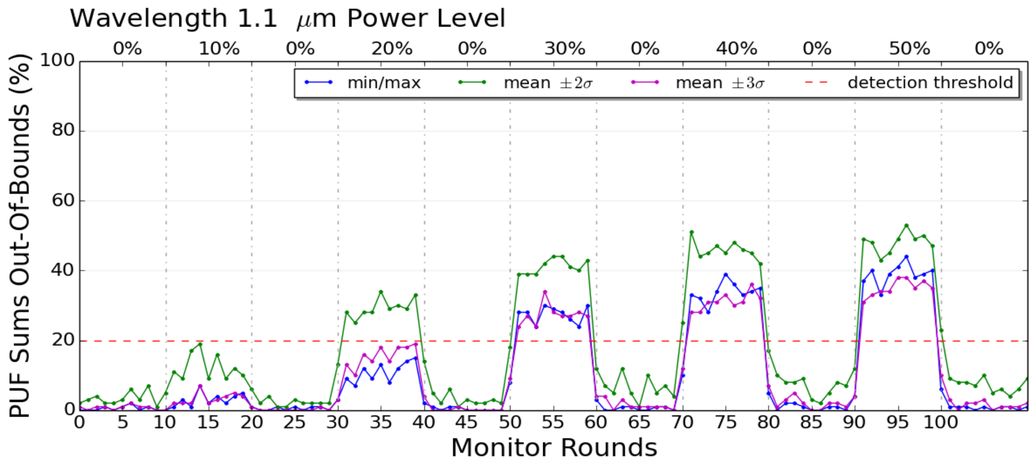
- Normal Condition: 1.2 V
- Apply ± 0.05 V => 60% OOB
- Immediate & strong change

Detectable



Technische Universität Berlin

Evaluation >> Laser Scanning



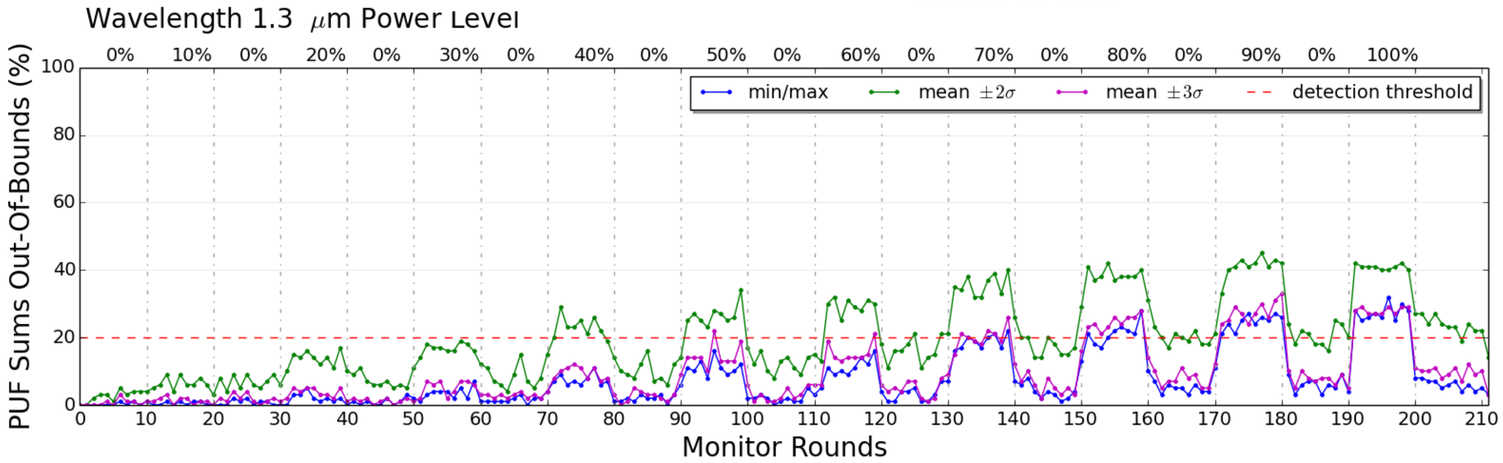
Setup

- PHEMOS-1000 Microscope

←

1.1µm Laser Scanning

- 1.1µm wavelength
- 10% up to 50% Power level

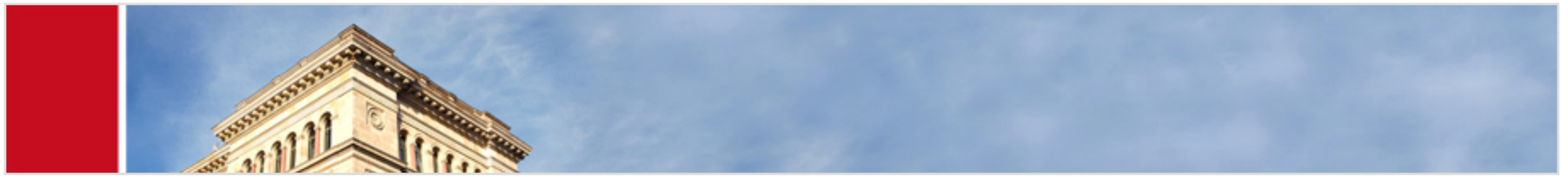


←

1.3µm Laser Scanning

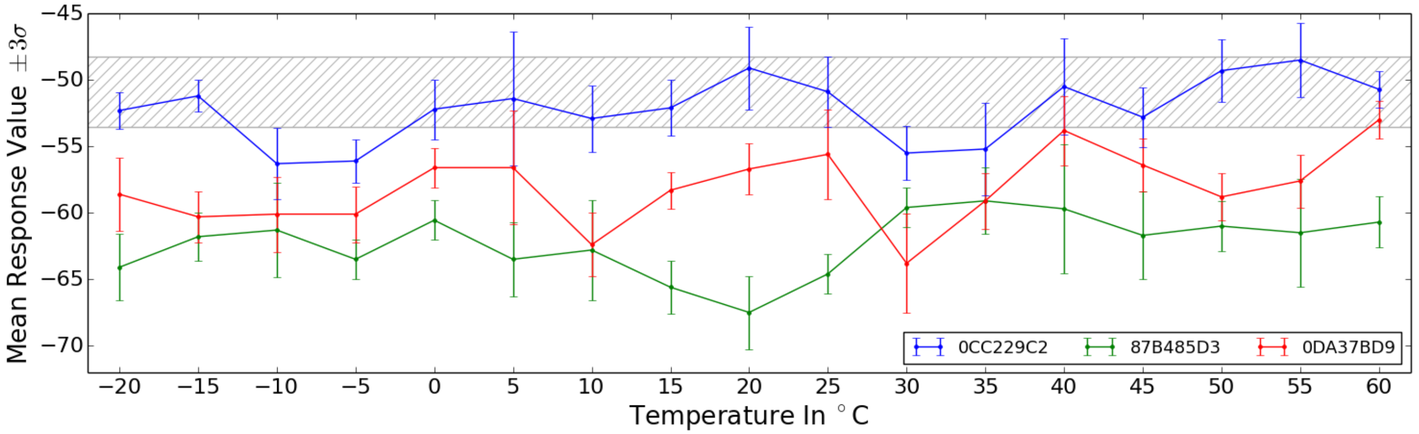
- 1.3µm wavelength
- 10% up to 100% Power level

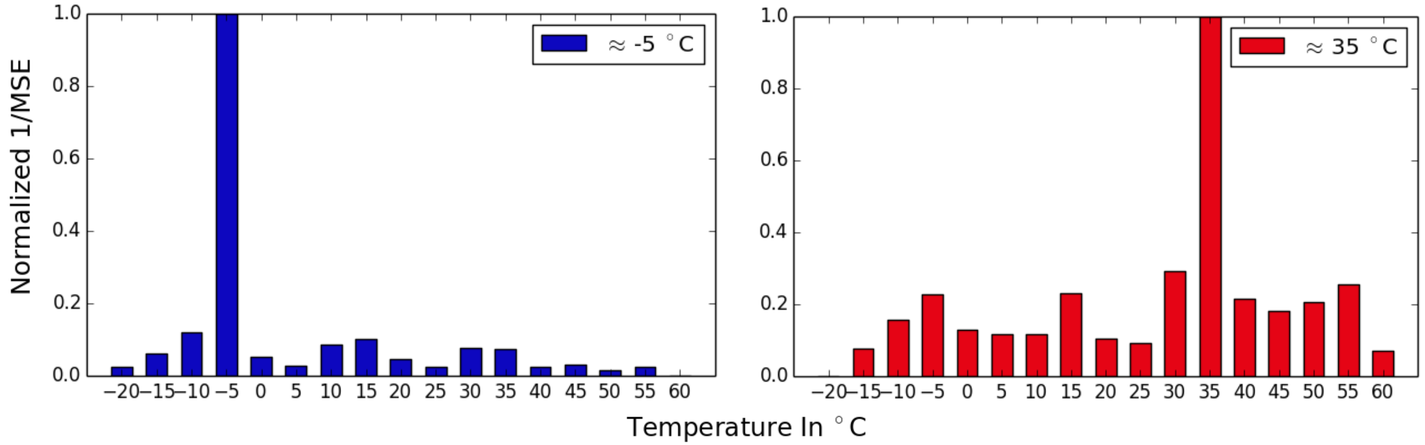
Detectable, but weak when using **low power levels** and **long wavelengths**



Technische Universität Berlin

Evaluation >> Temperatur Manipulation





Setup

- Experimental oven
- -20°C up to 60°C
- 5°C resolution

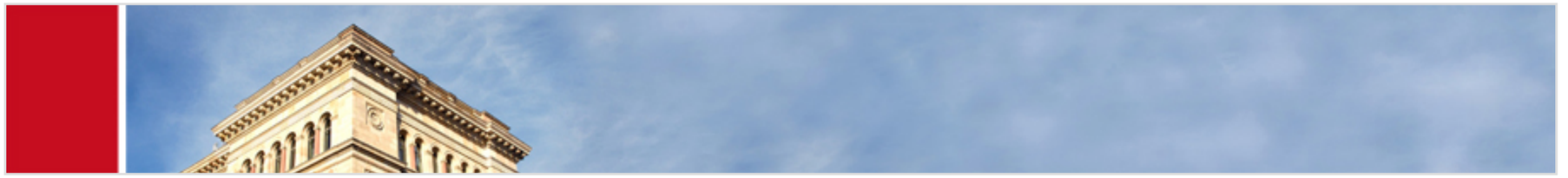
- **δf compensate temperature change**
  => good for the PUF
  => can not determine temp. oob.

Only way so far

← **Correlation with temperature DB**

- Measure current values
- Compare with all temp. entries

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\text{Response}_{n,T_1} - \text{Response}_{n,T_2})$$



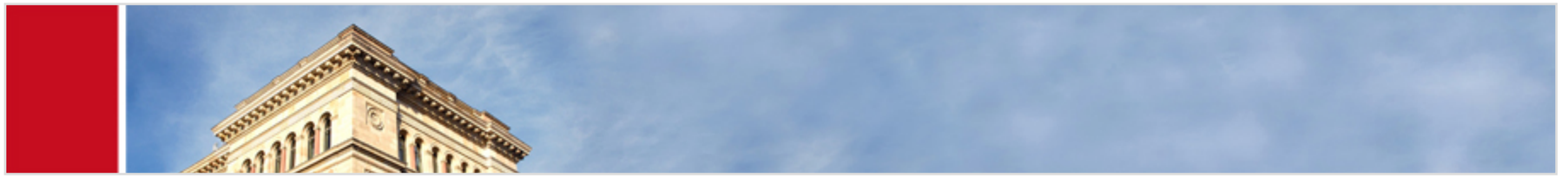
Technische Universität Berlin

Conclusion

- **Concept works!**
  - Allow us to detect attack types not covered by PUFs originally
  - Strong impact: **Supply voltage, system clock** manipulation
  - Meaningful impact: **Trojans** and **laser scanning,** (**temperature**)

- Several **attacks not covered** and **temper responses are not discussed**:
  - Recommendation: Traditional countermeasures + intrinsic security layer
    - Generate a more sound evaluation about the current device integrity
    - Detect fraud and manipulation even when traditional countermeasures are tricked

Advantages	Disadvantages
• Verification of the device integrity (even remotely) • Authentication and secure key storage included • Small development overhead (when a satisfying PUF is already part of the design)	• Power consumption of RO (depends on size and evaluation method) • Low temporal resolution: Glitch time << Sample time (even short attacks are build on try and error)



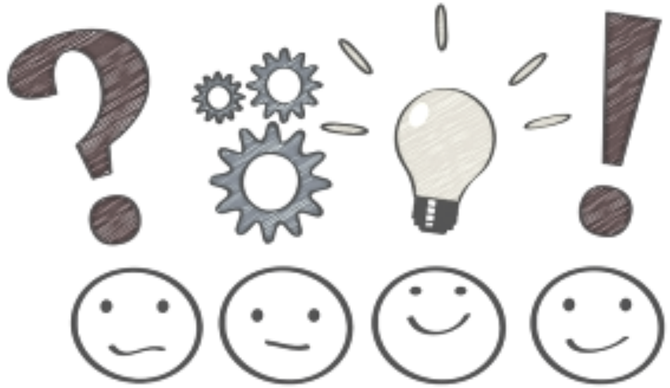
Technische Universität Berlin

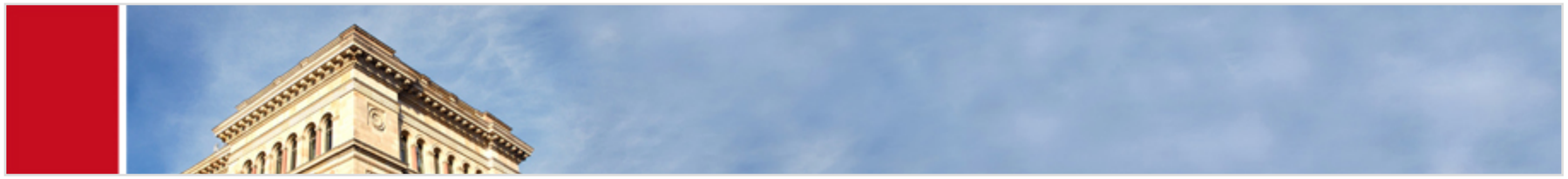
Future Work

- **Architecture**
  - Low temporal resolution: Evaluate multiple challenges in parallel!?
  - Find other/new sophisticated PUF functions or PUFs (BR-PUF, PE-PUF)

- **Evaluation**
  - Supervised learning to evaluate the PUF, like [36]
  - Dynamically select challenges during evaluation?

- **Experiments**
  - ASIC experiments to tackle hardware related issues
  - Further Attacks (resistance against combination of attacks)





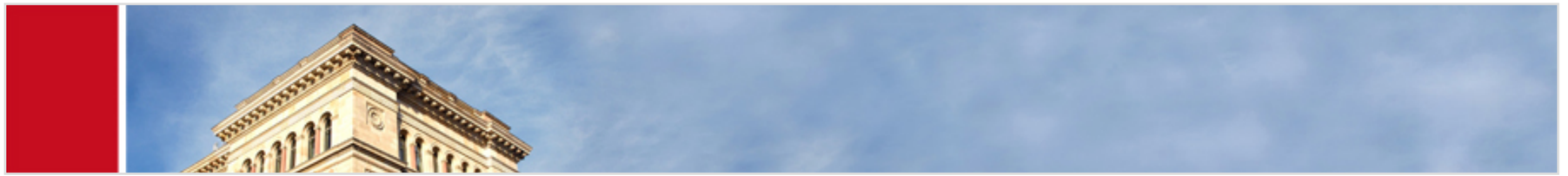
Technische Universität Berlin

Thank you!

Prof. Dr. Jean-Pierre Seifert
Shahin Tajik
Heiko Lohrke

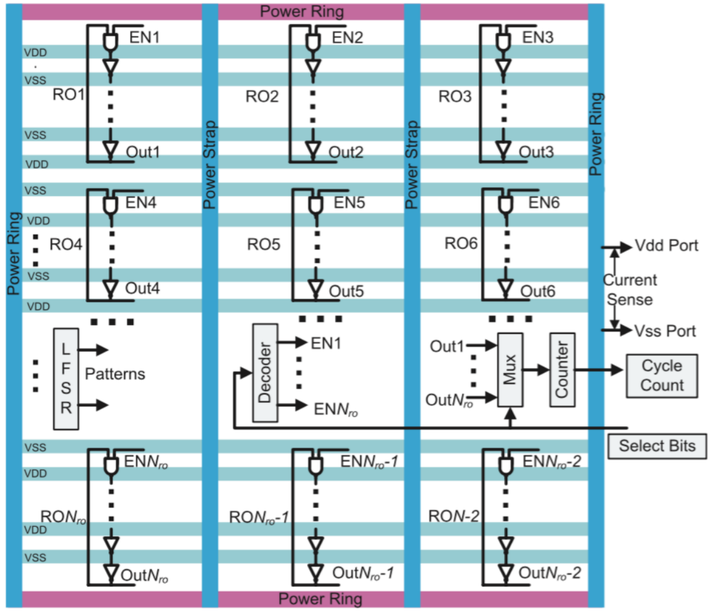
Further Interested People can …

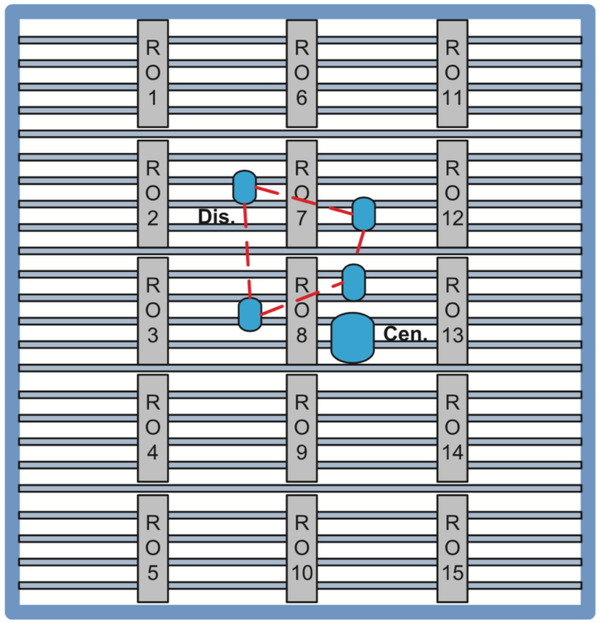
+ Read the thesis https://tubcloud.tu-berlin.de/index.php/s/KHTgCPBAV5K9LBi
+ Check out the GIT https://gitlab.sec.t-labs.tu-berlin.de/jfietkau/TamperEvidentPUF
+ Ask me anything jfietkau@sec.t-labs.tu-berlin.de
+ Request the paper from Shahin stajik@sec.t-labs.tu-berlin.de



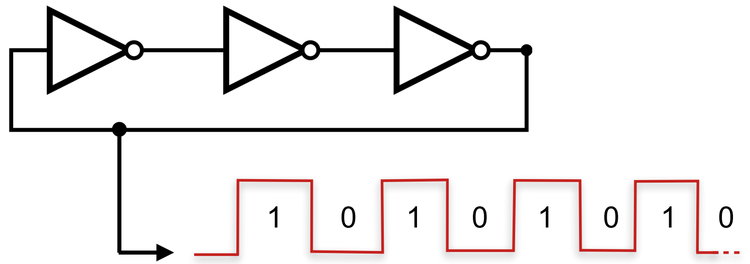
Technische Universität Berlin

Background >> Hardware Trojan Detection





Ring Oscillators



Ring Oscillator Network

+ Distribute ROs on the device
+ Active ROs iterate a counter
+ Compare biased vs. unbiased (golden chip)

=> Detect Trojans with decent size &

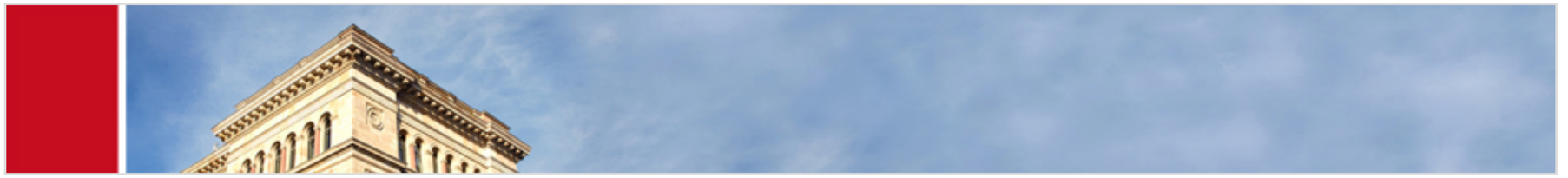
[53] Xuehui Zhang, Andrew Ferraiuolo, Mohammad Tehranipoor. Detection of Trojans Using a Combined Ring Oscillator Network and Off-Chip Transient Power Analysis. *ACM Journal on Emerging Technologies in Computing Systems 9, 3, Article 25*, September 2013.

[36] Nima Karimian, Fatemeh Tehranipoor, et al. Genetic Algorithm for Hardware Trojan Detection with Ring Oscillator Network (RON). *2015 IEEE International Symposium Technologies for Homeland Security (HST)*, 2015.

Technische Universität Berlin

Bonus Slide

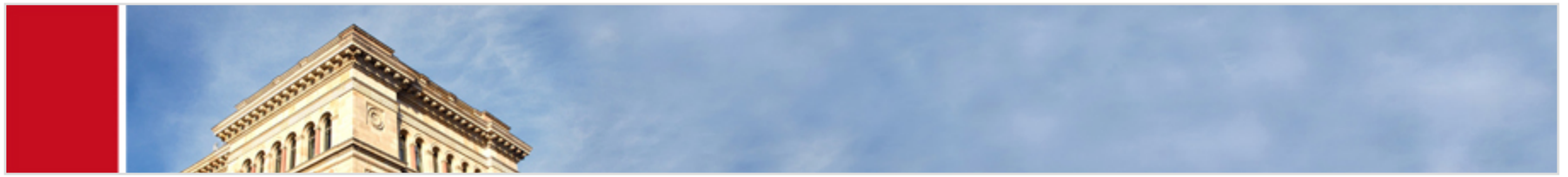
Vendor	Xilinx	Altera	Microsemi
Product	SecMon	Supervisor IP	EnforcIT Security Monitor
Support	Virtex-5,6,7, Spartan-6, Zynq	Arria V, Stratix V	SmartFusion2, IGLOO2
Features	configuration and memory integrity as well as temperature, voltage, user clocks, and JTAG activity, monitoring for partial reconfiguration	monitoring functions for configuration errors, key values and states, temperature and heartbeat signals	JTAG & clock frequency monitor, system heartbeat function. Customizable clocks, timeout monitoring, logic integrity and fault detection, runtime IP version reporting
Special	In future: Multiple SecMon cores shall be connectable, acting as a single unit on board level (device-to-device) and stacked silicon interconnect (SSI) devices.	Build into logic lock region with low latency access to monitors, sensors and the partial reconfiguration control (used for zeroization)	-



Technische Universität Berlin

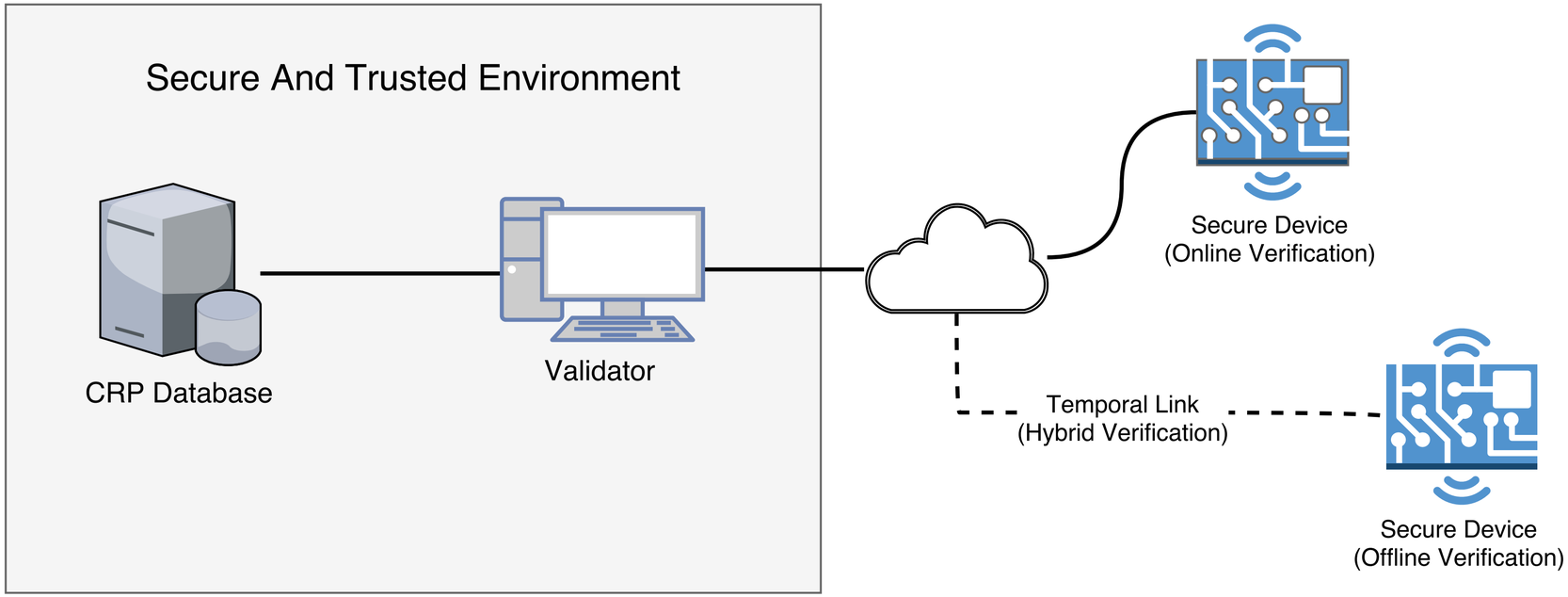
Design >> PUF Requirements

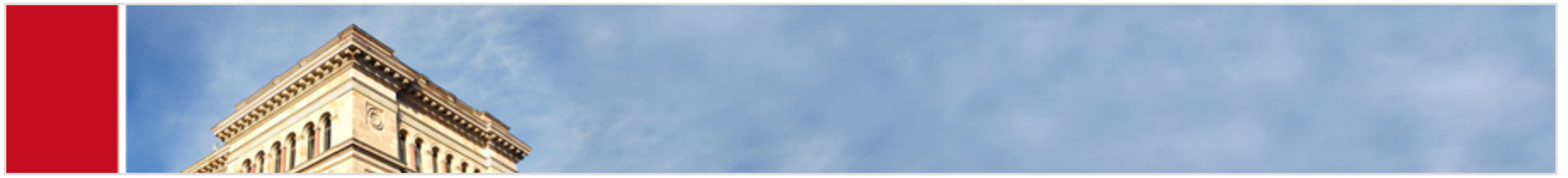
- **Strong PUF**: The architecture should support a large challenge space to prevent behaviour prediction and allow support of origin PUF applications like authentication.
- **Physical Sensitivity**: The architecture must embed components that are sensitive to several physical phenomena which can be measured in an adequate way.
- **Routable**: The architecture must be able to be distributed over the whole IC to allow detection of local effects like introduced by laser and Trojans.
- **Participation**: Each measurement component of the architecture must equally contribute to the PUF response to ensure that local effect will not be out- weighed by stronger components.
- **Time Resolution**: The architecture must allow a permanent observation of the device to detect non-permanent changes. Furthermore, the components should be evaluated simultaneously.



Technische Universität Berlin

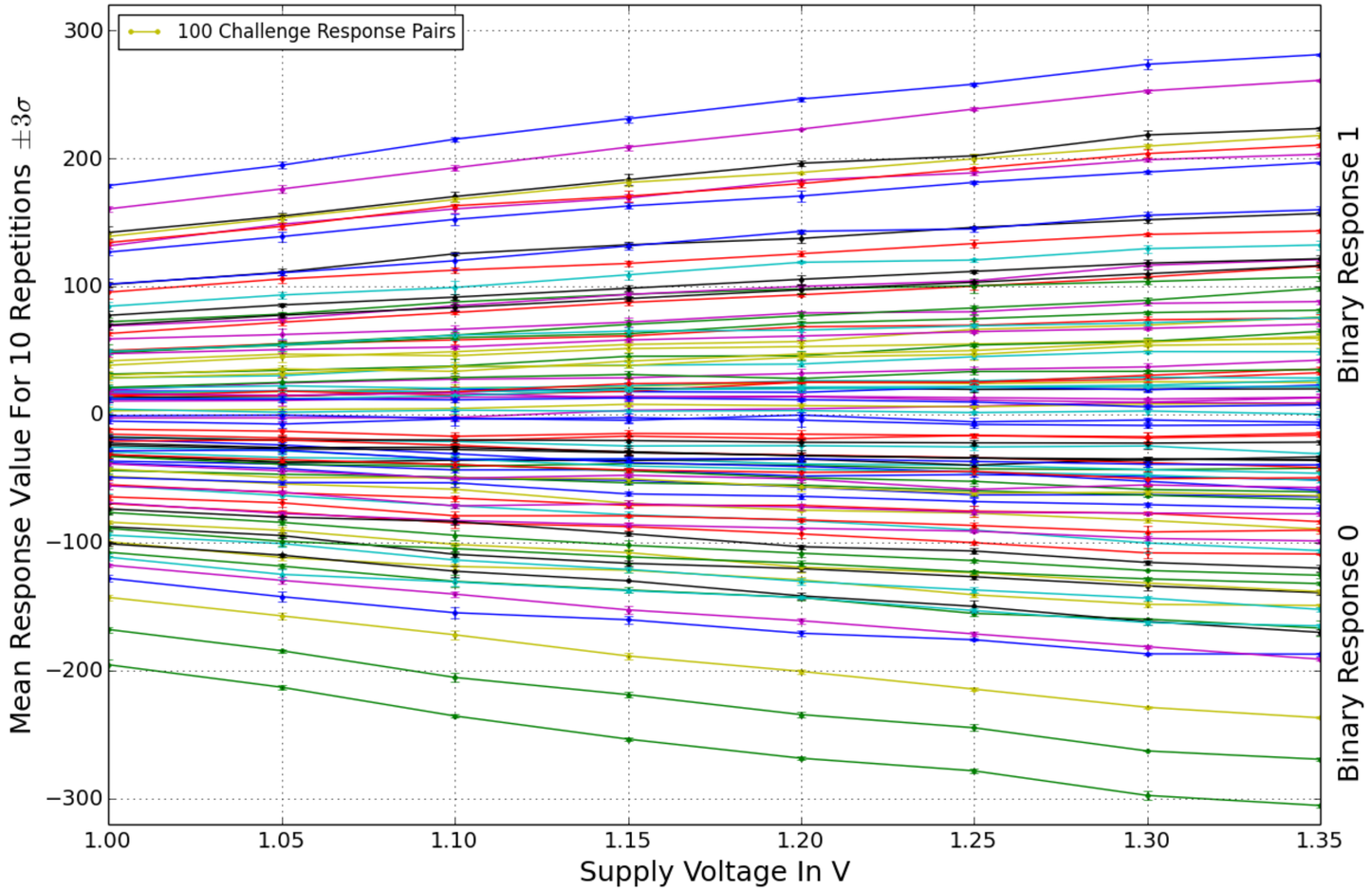
Design >> Architecture





Technische Universität Berlin

Evaluation >> Fault & Glitching Attacks >> Voltage



Voltage Attack

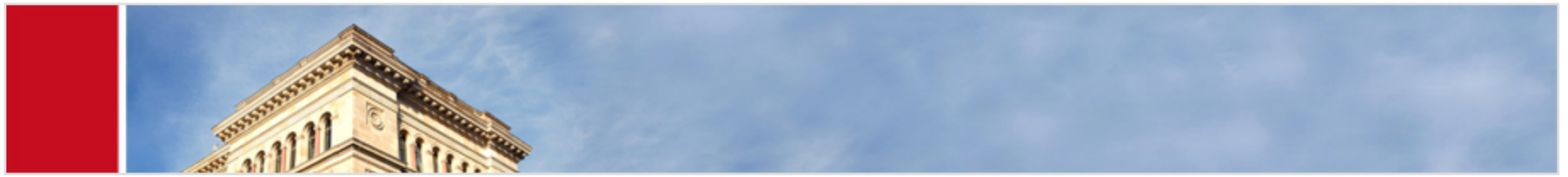
- Normal Condition: 1.2 V
- Apply ± 0.05 V => 60% OOB
- immediate change

Why?

- Voltage accelerate the RO
- Faster RO => higher counter
- Response Value ~ Voltage
- Different slopes!

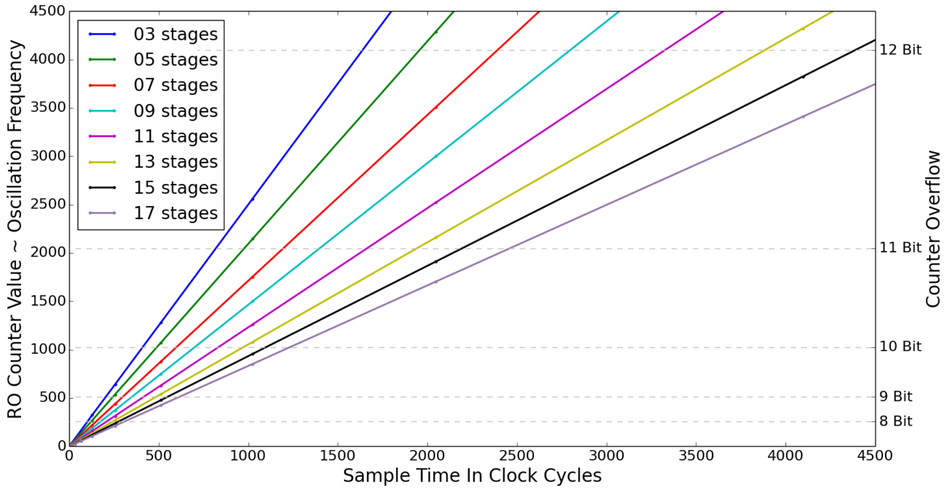
==========

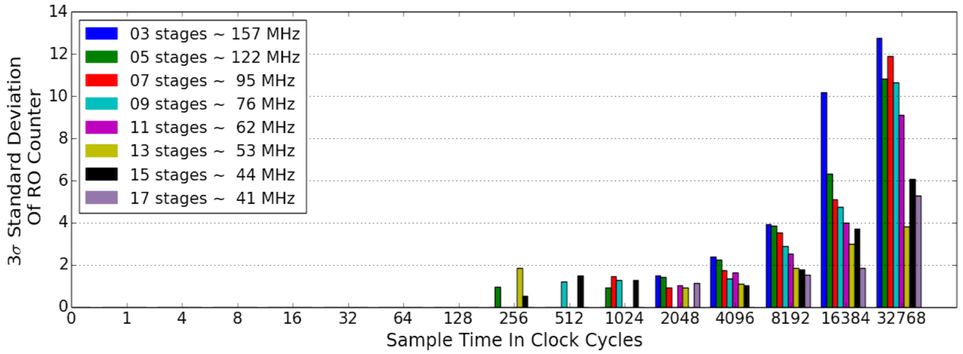
Easy to detect



Technische Universität Berlin

Implementation >> The hardest part … build proper ROs on the FPGA





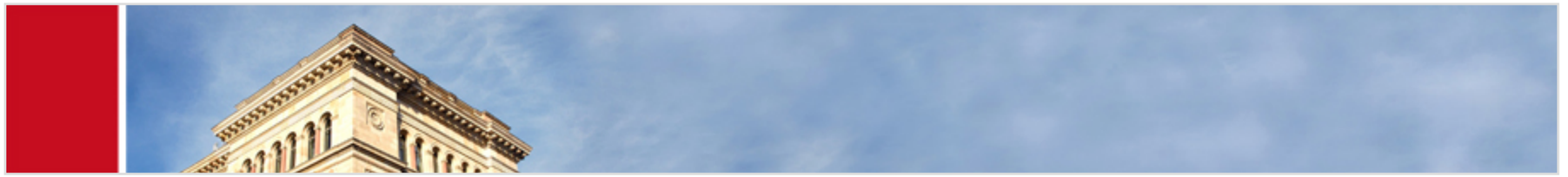
$$e_{measure,max} = \pm\frac{f_{clk}}{2 \cdot n_{clkcyc}} \qquad n_{clkcyc} = \frac{50\text{ MHz}}{2 \cdot 0.1\text{ MHz}} = 250\text{ clkcyc} \qquad (3)$$

(a) Maximum quantisation error based on system clock f_{clk} and sample time n
(b) Solved equation to find minimum clock cycles required for a 0.1 MHz error

$$f = \frac{1}{2 \cdot n \cdot t_d} \qquad t_d = 0.52 \cdot \frac{C_L V_{dd}}{(W/L)k' V_{DSAT}(V_{dd} - V_{th} - V_{DSAT}/2)} \qquad (2)$$

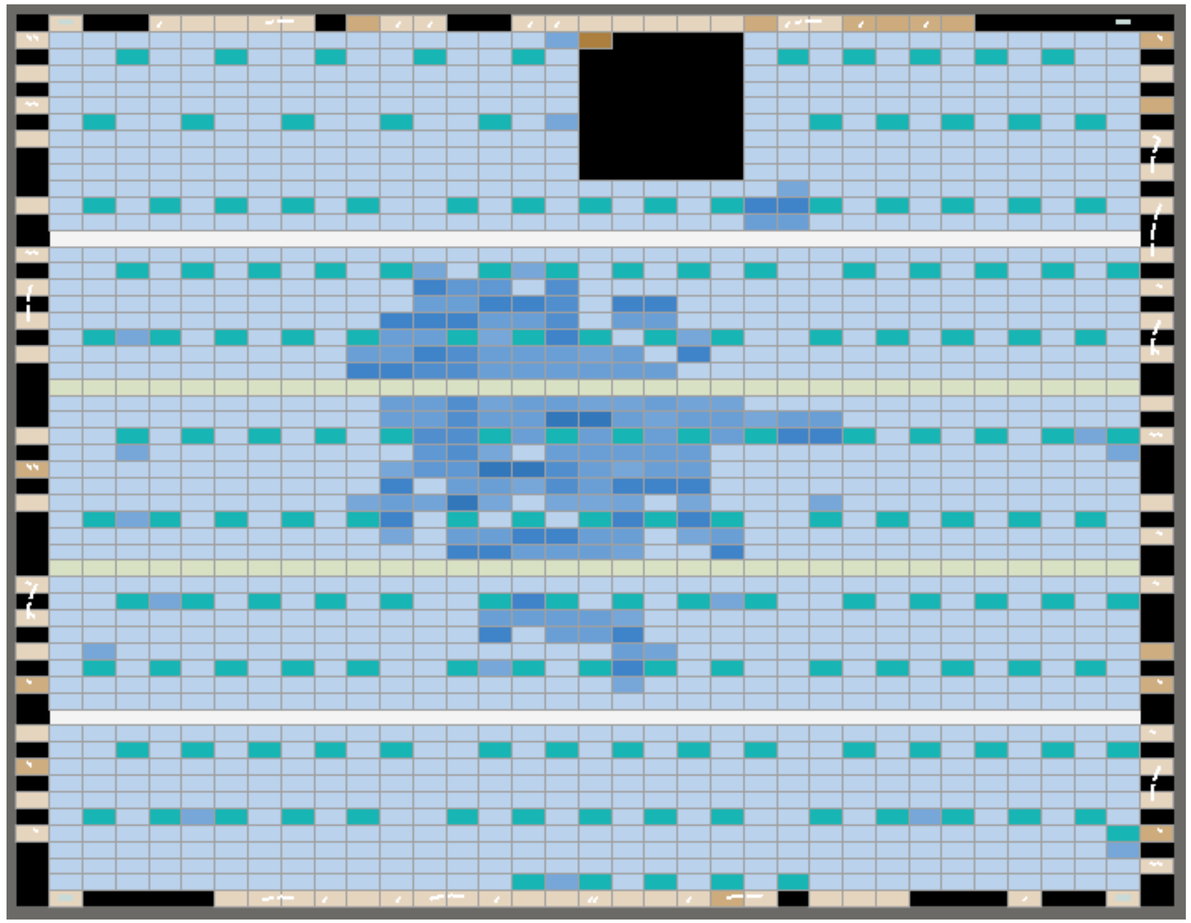
Frequency of a RO with n inverters having an delay of t_d each. With C_L = load capacitance, k' = transconductance, W/L = ratio between transistor width to channel length and saturation, threshold and supply voltage. [53]

[19] Dominik Merli, Frederic Stumpf, Claudia Eckert. Improving the Quality of Ring Oscillator PUFs on FPGAs. In *Proceedings of the 5th Workshop on Embedded Systems Security*, New York, NY, USA, 2010. ACM.



Technische Universität Berlin

Implementation >> Placement

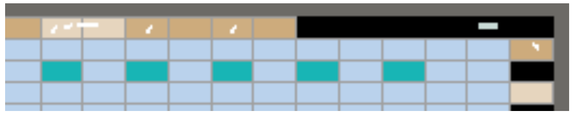


Legend

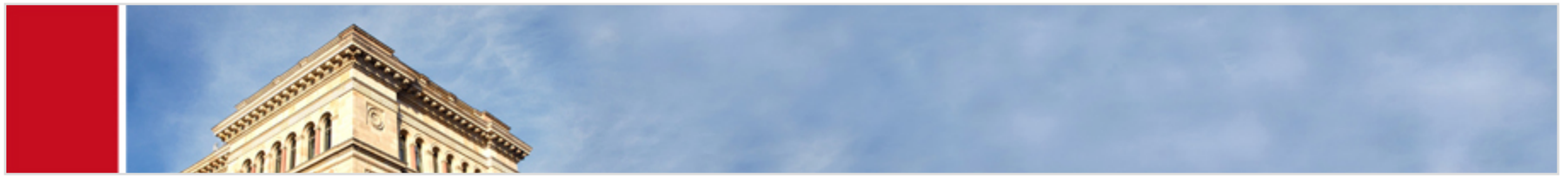
+ 32 ROs with 5 stages each
+ Further logic e.g. adder, UART, state machine
+ Unused configuration space
+ Hardwired logic (ASIC)

Line-by-line Placement Technique

+ First RO

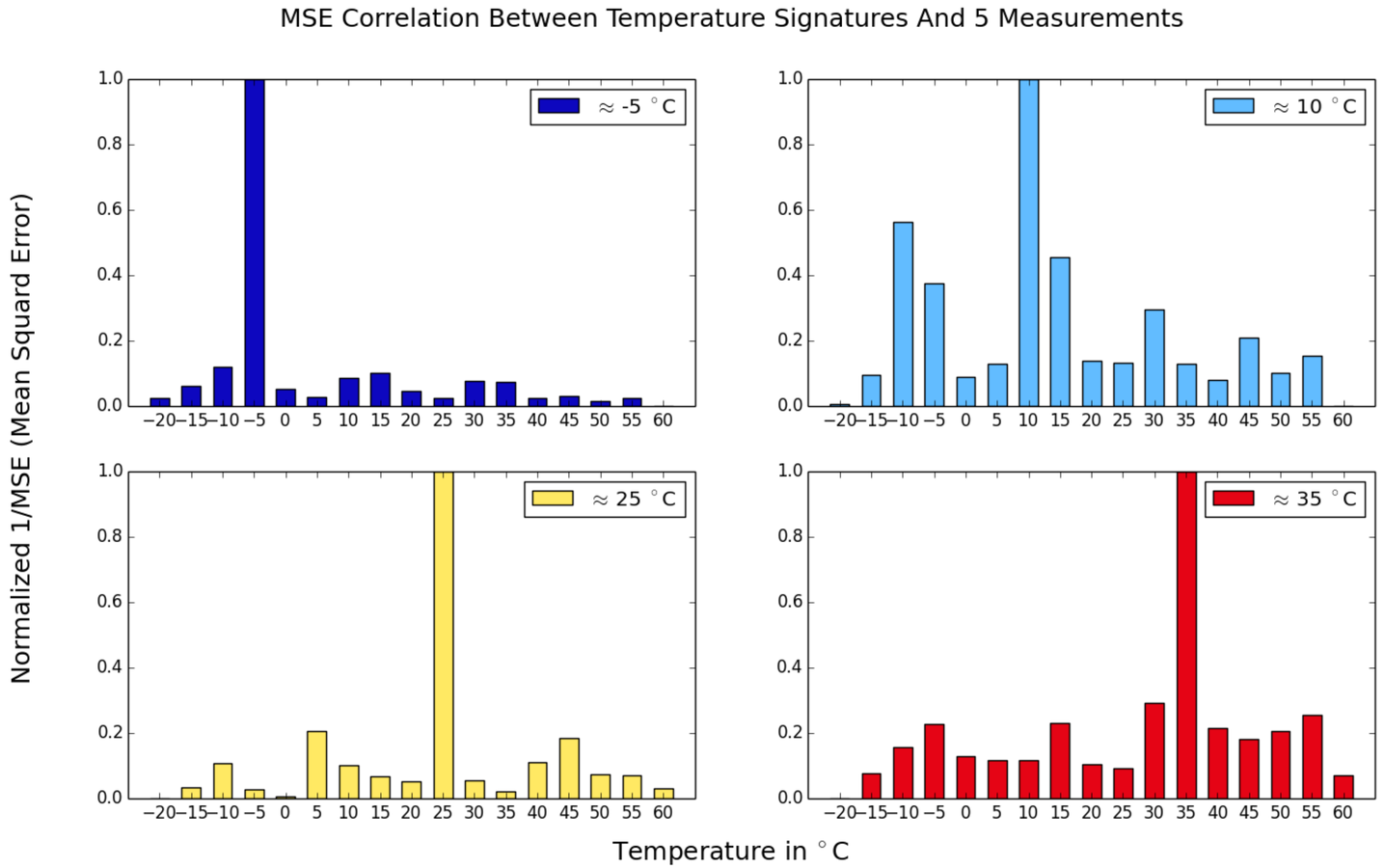


+ Keep intermediate space constant
+ Do not overflow



Technische Universität Berlin

Evaluation >> Temperatur Manipulation



Using DB for measurement

- Measure current values
- Compare with all temp. entries

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\text{Response}_{n,T_1} - \text{Response}_{n,T_2})$$