Hardware Security Evaluation of Embedded Applications against Fault Injection Attacks

ZAHRA KAZEMI

JOURNÉE THÉMATIQUE SUR LES ATTAQUES PAR INJECTION DE FAUTES (JAIF 2021)

24 SEPTEMBER 2021

Cybersecurity Institute
Univ. Grenoble Alpes

Outline

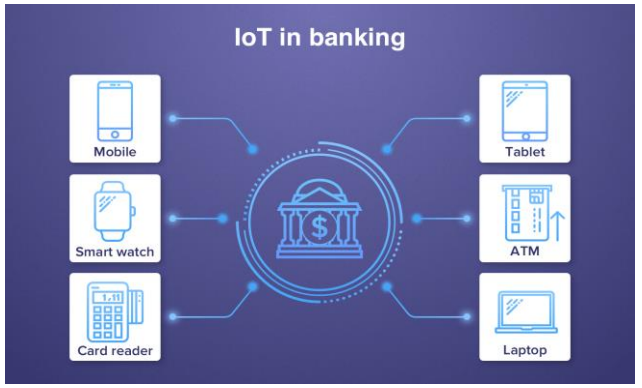
- Introduction
- Our Hardware Security Evaluation Platform
- High-Level Hardware Security Assessment Methods
- Fine Tuned Experimental Evaluation for RISC-V
- Conclusions and Future Works
- References

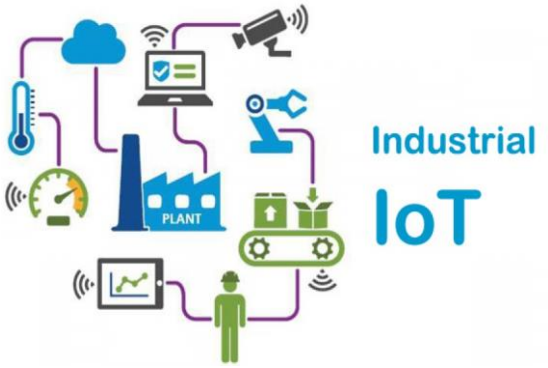
Outline

- Introduction
- Our Hardware Security Evaluation Platform
- High-Level Hardware Security Assessment Methods
- Fine Tuned Experimental Evaluation for RISC-V
- Conclusions and Future Works
- References

Introduction:

Embedded IoT Systems Role in Daily Life





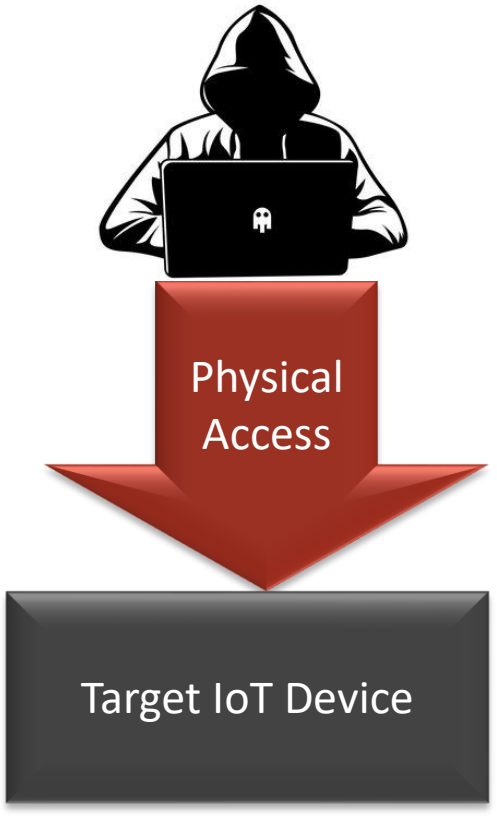
Introduction:

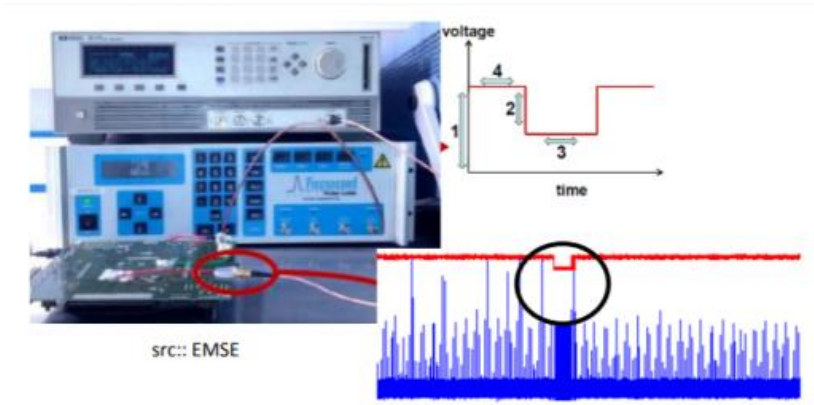
Imbalance Security as a Key Element In IoT Market

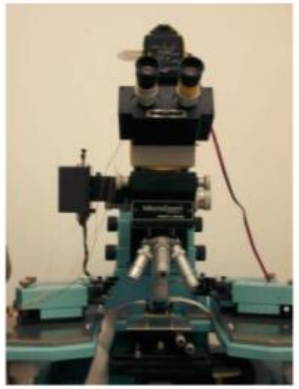
Imbalance Security as a Key Element in IoT Market

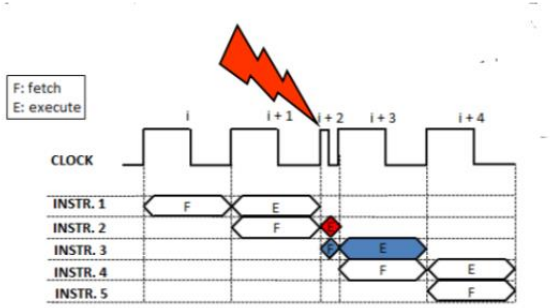
Introduction:

Different Security Attacks Against IoT/Embedded Systems



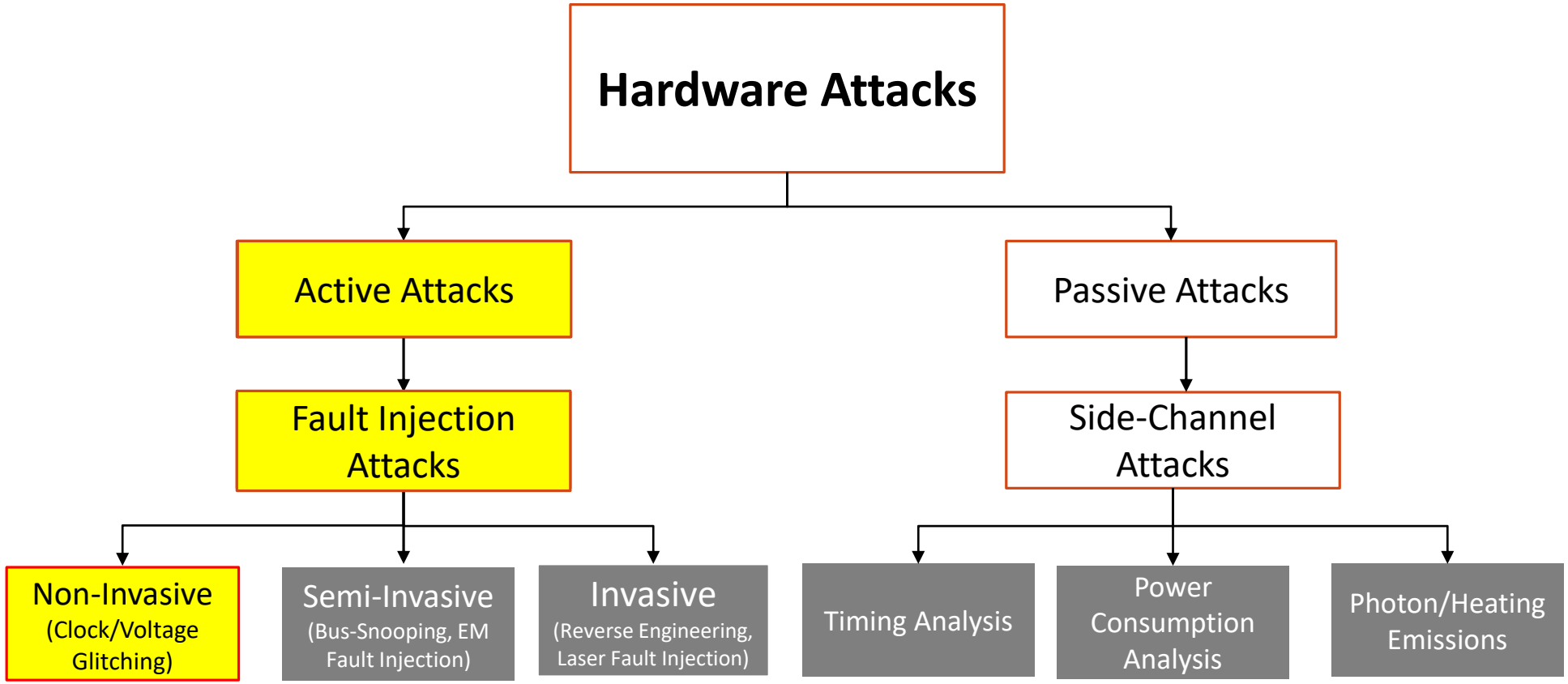






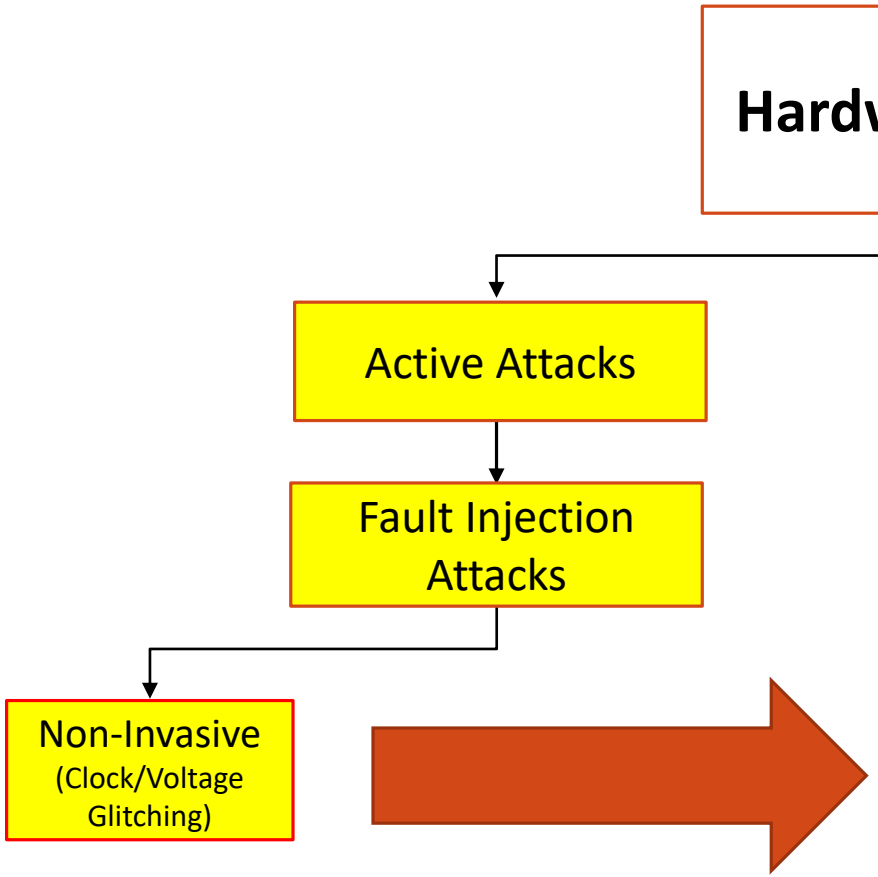
Introduction:

Different Hardware Security Attacks

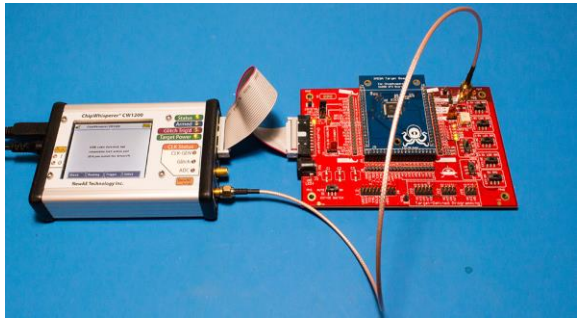


Introduction:

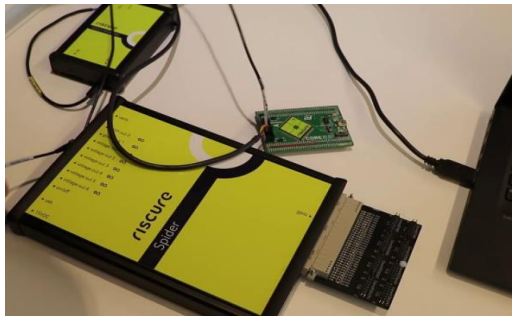
Different Hardware Security Attacks



Examples in Academia and Industry:



ChipWhisperer tool from NewAE©



Spider tool from Riscure©

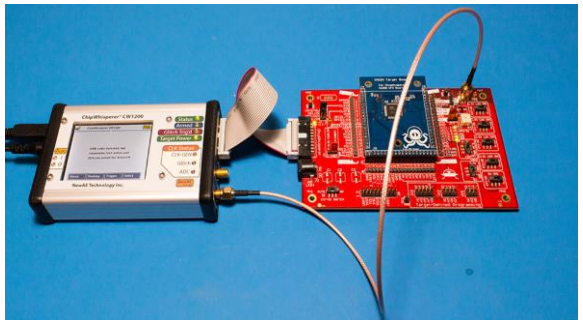
Our focus is on non-Invasive attacks because they can be performed by:

- **low-cost/accessible** tools and equipment so they are practical for low-cost targets
- **mid-level** knowledge attackers

Introduction:

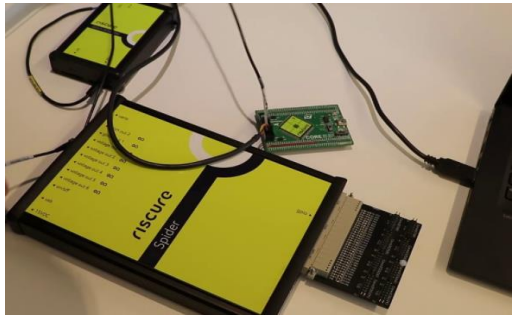
Different Non-Invasive Fault Injectors

Examples in Academia and Industry:



ChipWhisperer tool from NewAE©

+ Proper for academia
- Hardware dependent
- Not fully automated
- Lack of parameter database
- Not proper for all the external targets



Spider tool from Riscure©

+ Proper for industry
+ Automatic testing scenarios
- Hardware dependent
- Too expensive
- Not easy to get acquainted for non-security users

Introduction:
Our Goal

1. Review the state of the art of existing clock glitch generators
2. Develop an evaluation platform for non security experts which is:
   - **Capable of injecting precise faults**
   - **Open source implementation**
   - **Low cost and accessible**
   - **Configurable for various targets**
   - **Automatic scenarios**
   - **Easy to use for novice hardware security evaluators**
3. Introduce high-level evaluation methods for common functions and patterns of embedded IoT applications

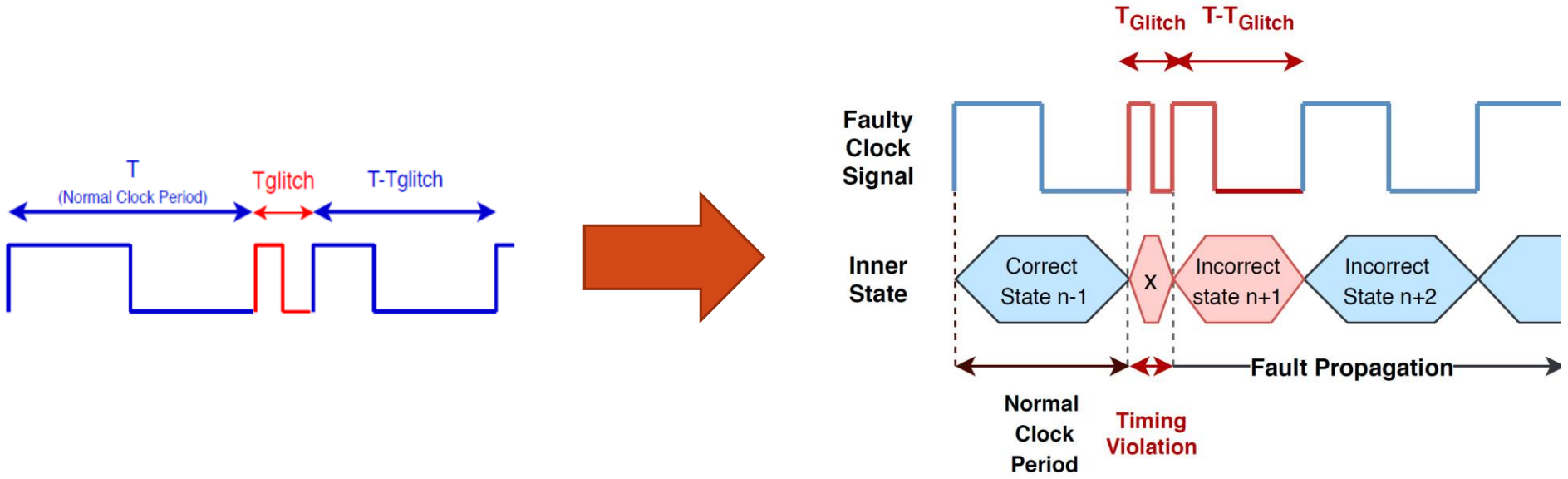
Outline

- Introduction
- Our Hardware Security Evaluation Platform
- High-Level Hardware Security Assessment Methods
- Fine Tuned Experimental Evaluation for RISC-V
- Conclusions and Future Works
- References

Our Hardware Security Evaluation Platform:

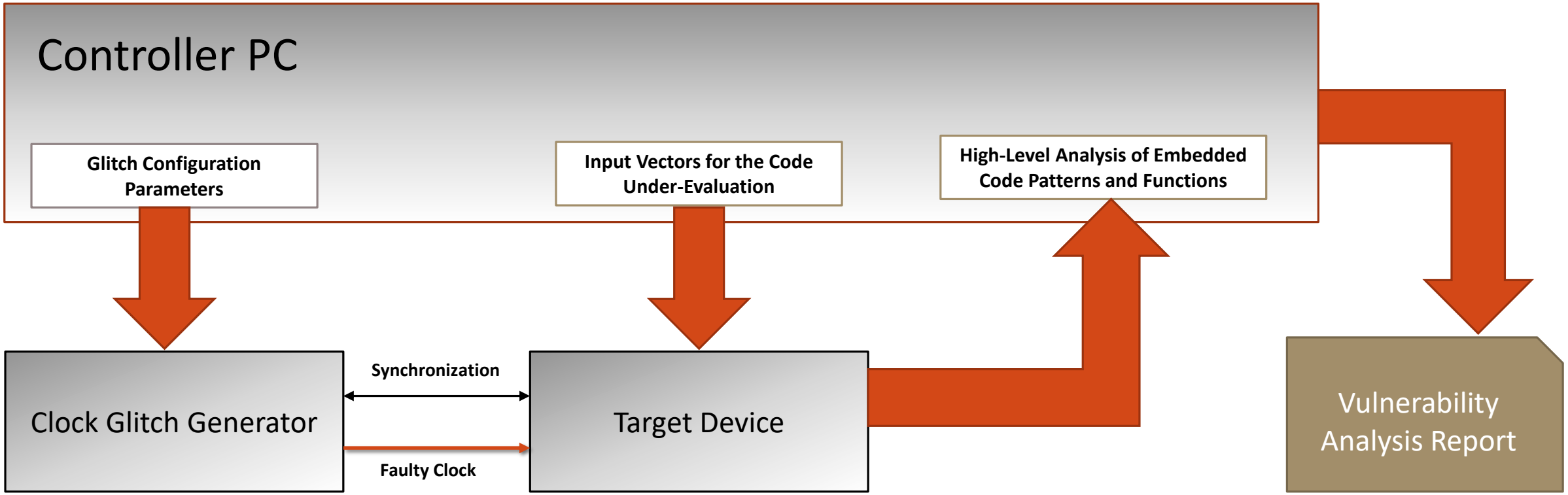
Clock Glitching Attacks

Clock glitching attack violates critical path delay by insertion of additional positive clock edge and leads to incorrect values and states in the target system:



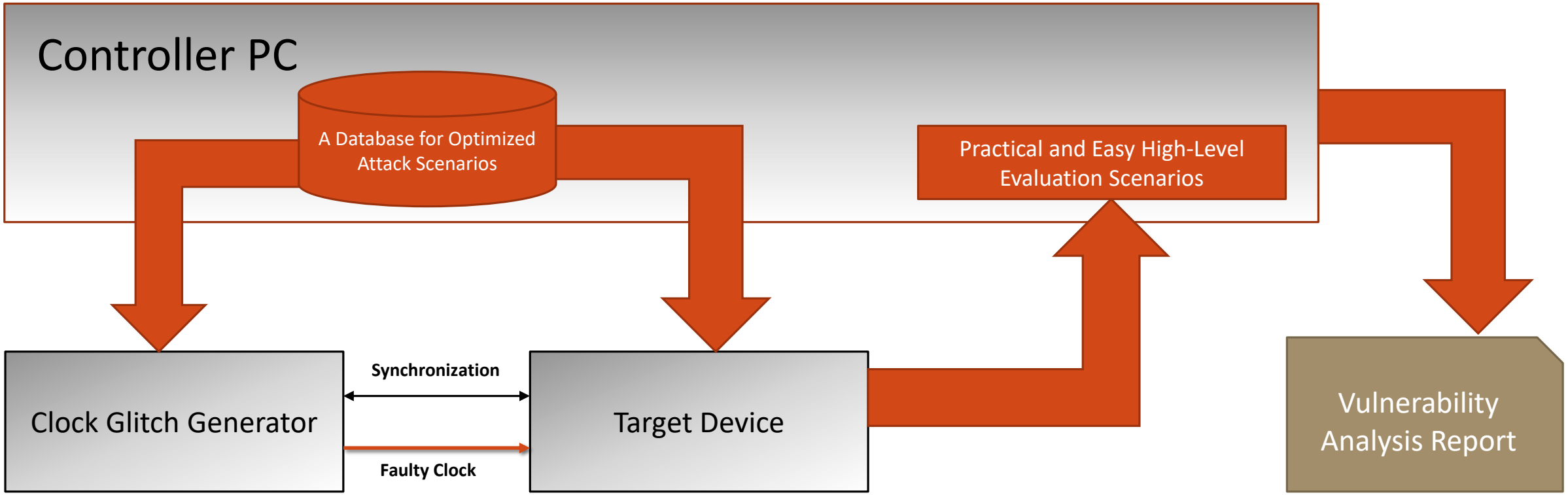
Our Hardware Security Evaluation Platform:

A High-Level Schematic



Our Hardware Security Evaluation Platform:

A High-Level Schematic



Our Hardware Security Evaluation Platform:

Clock Glitch Generation Methods

Clock Glitch Generator

Important Characteristics:

1. Needed Equipment and Cost
2. Complexity
3. Minimum Glitch Width
4. Capability to Inject Glitch into Specific Clock Cycle
5. Capability to Perform Run-Time Configuration
6. Capability to Control Generated Faulty Clock Frequency
5. Reproducibility of Faulty Clock

Table 3. Review of previously proposed clock glitch generators.

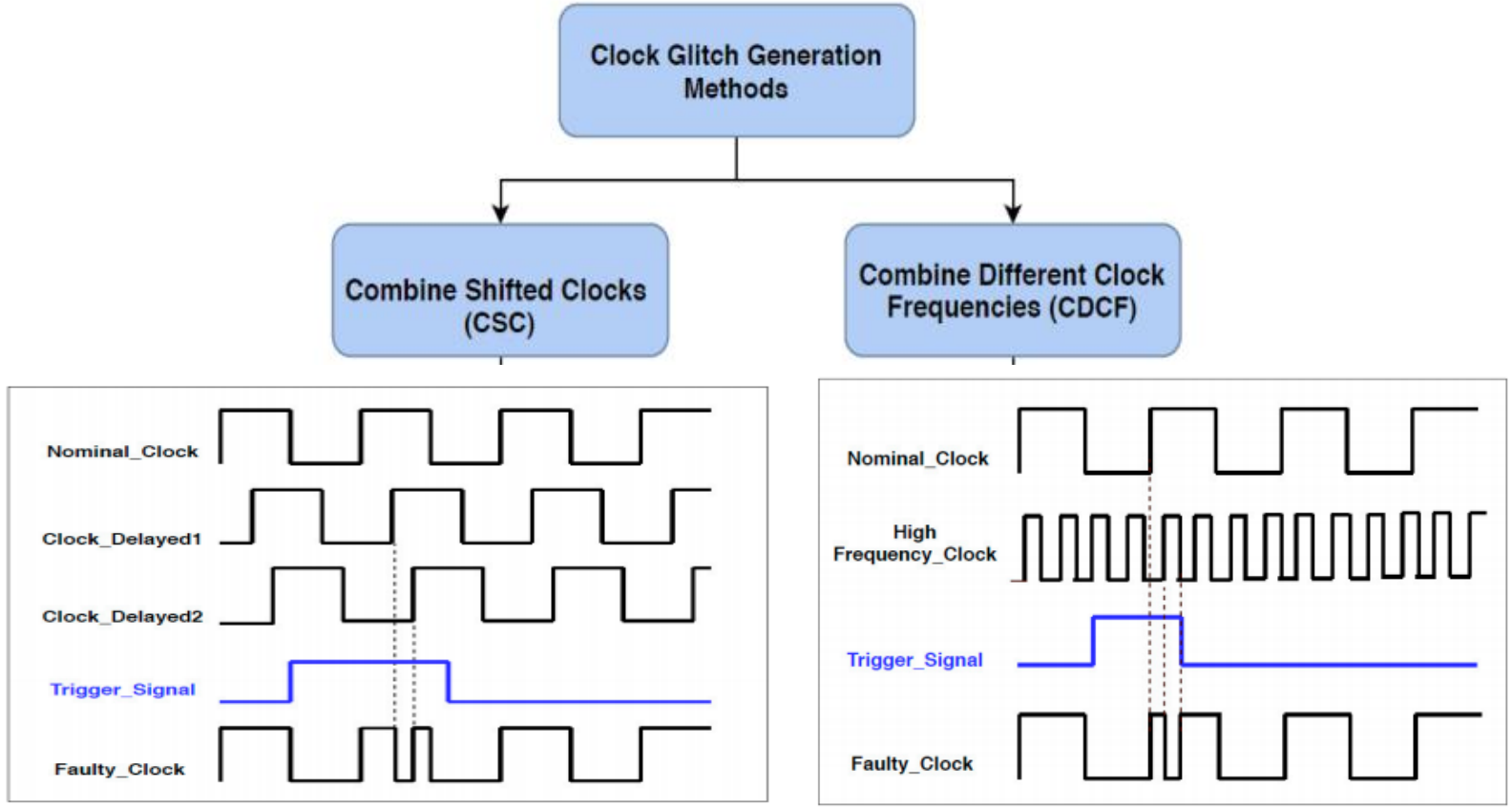
Characteristics \ Methods		Fault Attack By Tampering Clock Input								
		CSC					CDCF			
		Using FPGA Features				Using External Clock Sources	Using External Sources		Using FPGA Features	
		[39]	[19]	[37]	[29]	[40]	[41]	[11]	[38]	[27]
Evaluation Platform	Equipment	Xilinx Virtex5 FPGA	Spartan6 FPGA	Spartan6 FPGA	Spartan6 FPGA	FPGA on SASEBO-G VirtexII-Pro XC2VP30	Agilent 11152 Pulse generator and SASEBO-G	Agilent E4438C 6GHz Waveform generator DE2-115 development board	Ring Oscillators on Spartan-3E FPGA	Virtex-II Pro XC2VP30 FPGA
	System Complexity	Moderate	Moderate	Moderate-High (PCB designing)	Moderate-High (PCB designing)	Moderate-High (evaluation platform)	Moderate-High (evaluation platform)	Moderate	Moderate-High	Moderate
	Cost	Moderate	Moderate-High *	Moderate	Moderate	Moderate-High * (evaluation platform)	Moderate-High * (evaluation platform)	Moderate	Moderate	Moderate

*[1] Please see: Kazemi, Zahra, David Hely, Mahdi Fazeli, and Vincent Beroulle. 2020. "A Review on Evaluation and Configuration of Fault Injection Attack Instruments to Design Attack Resistant MCU-Based IoT Applications" *Electronics* 9, no. 7: 1153. https://doi.org/10.3390/electronics9071153

Our Hardware Security Evaluation Platform:

Clock Glitch Generation Methods

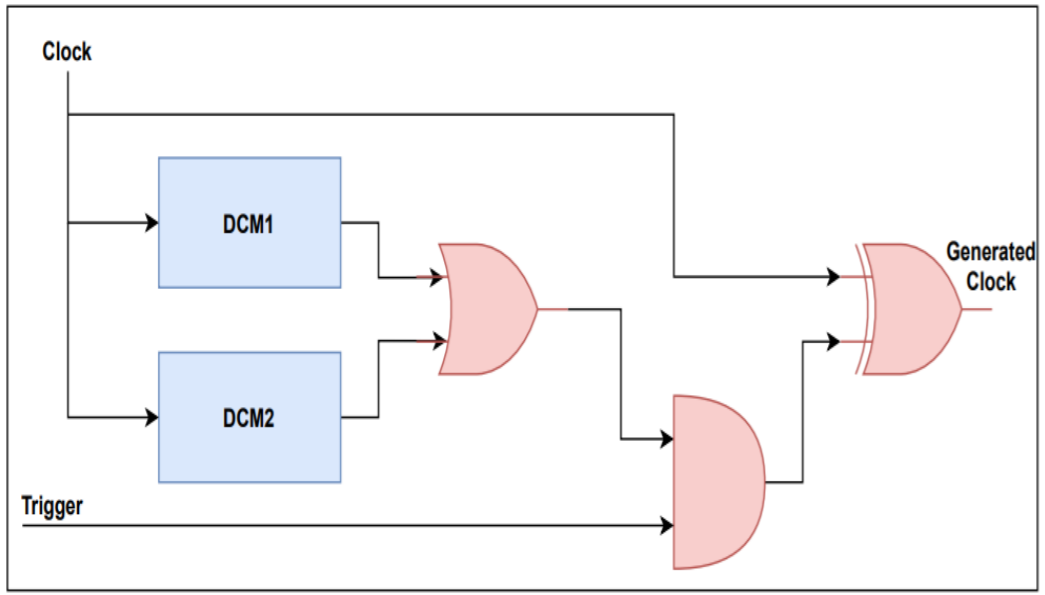
Clock Glitch Generator



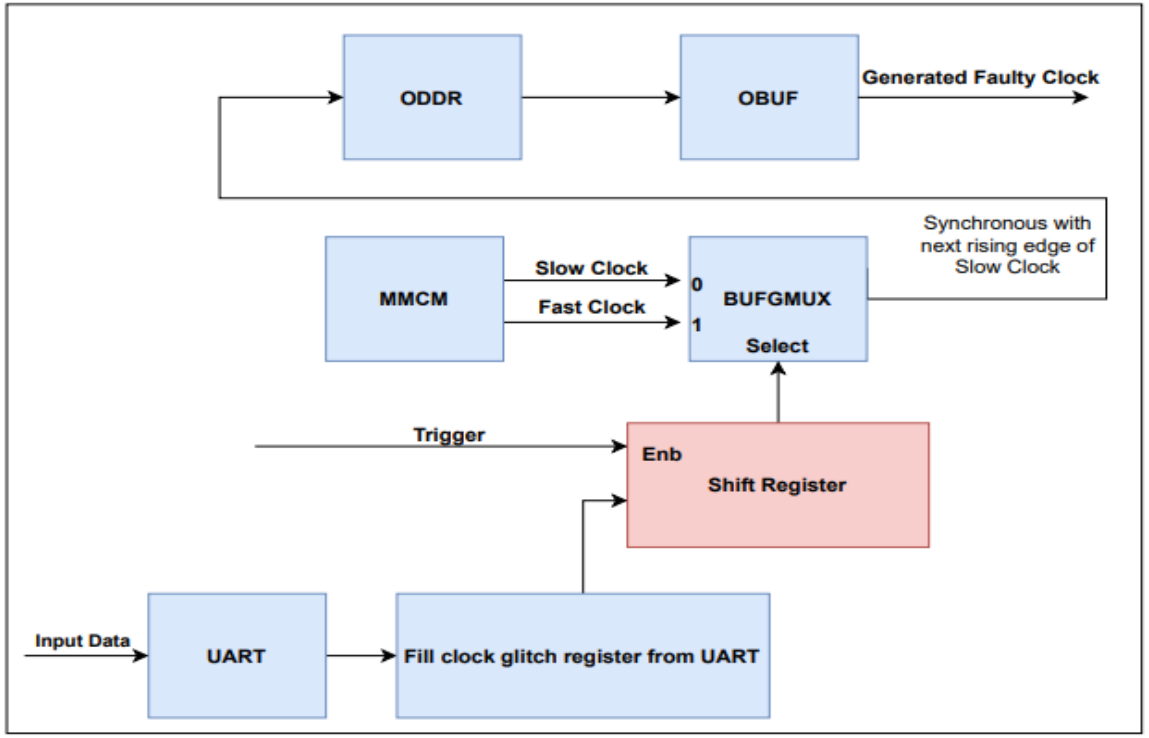
Our Hardware Security Evaluation Platform:

Clock Glitch Generation Methods

Implemented Clock Glitch Generator Based On Combine Shifted Clocks (Xilinx FPGA Arty-S7-50) [2]

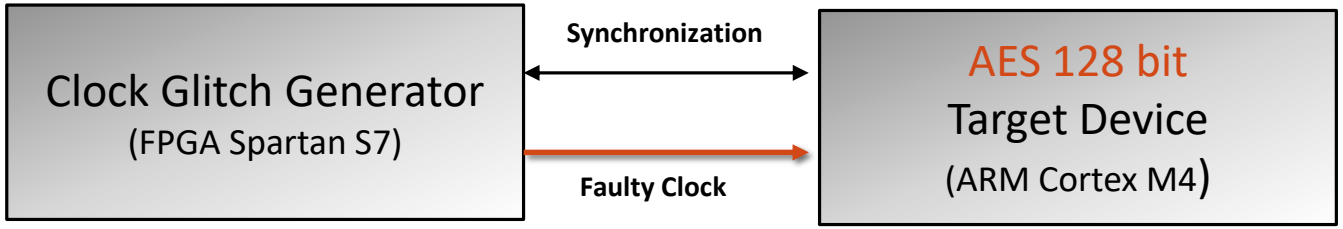


Implemented Clock Glitch Generator Based On Combine Clocks with Different Frequencies (Kintex 7 FPGA Digilent Genesys-2) [2]

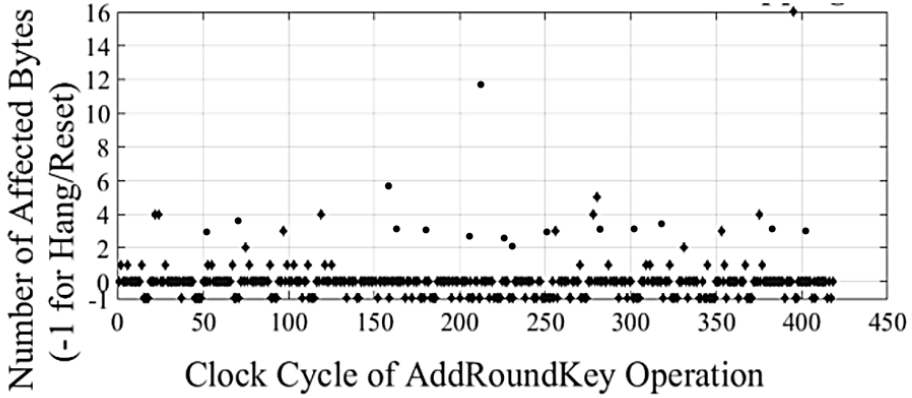


Our Hardware Security Evaluation Platform:

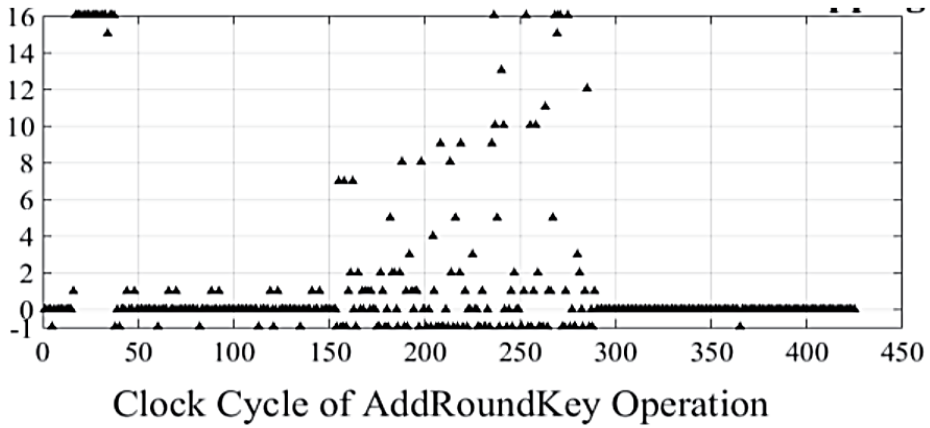
Comparison of Different Clock Glitch Generators



CSC:



CDCF:



Our Hardware Security Evaluation Platform:

1. One can use our open-source glitch generator on various FPGA types with one of the following features:
   - **Digital Clock Management (DCM)** (e.g., Spartan-3 and Virtex-4)
   - **Phase Locked Loop (PLL)** (e.g., Virtex-5 and Spartan-6)
   - **Mixed-Mode Clock Manager (MMCM)** (e.g., Virtex-6 and the seven series FPGAs)

2. To be able to use runtime reconfiguration facture one needs to use Xilinx®7 series, UltraScale., and UltraScale+.

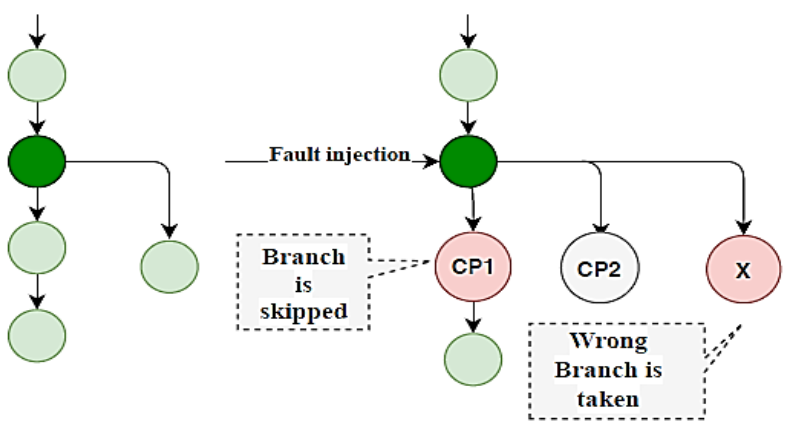
Outline

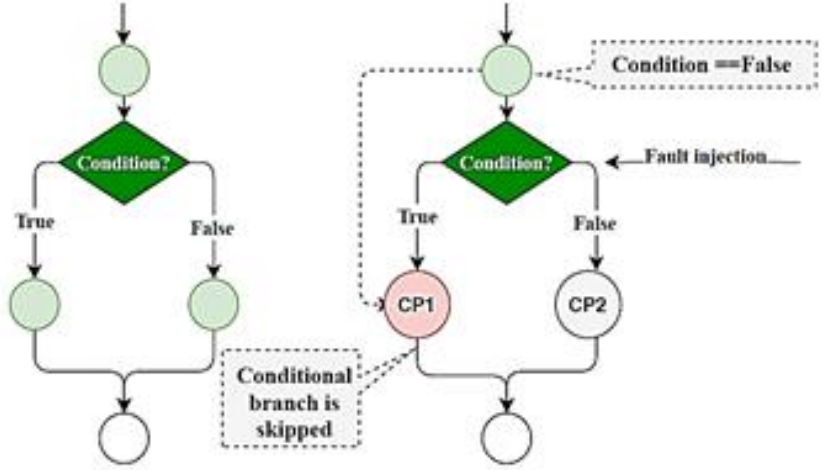
- Introduction
- Our Hardware Security Evaluation Platform
- High-Level Hardware Security Assessment Methods
- Fine Tuned Experimental Evaluation for RISC-V
- Conclusions and Future Works
- References

High-Level Hardware Security Assessment:

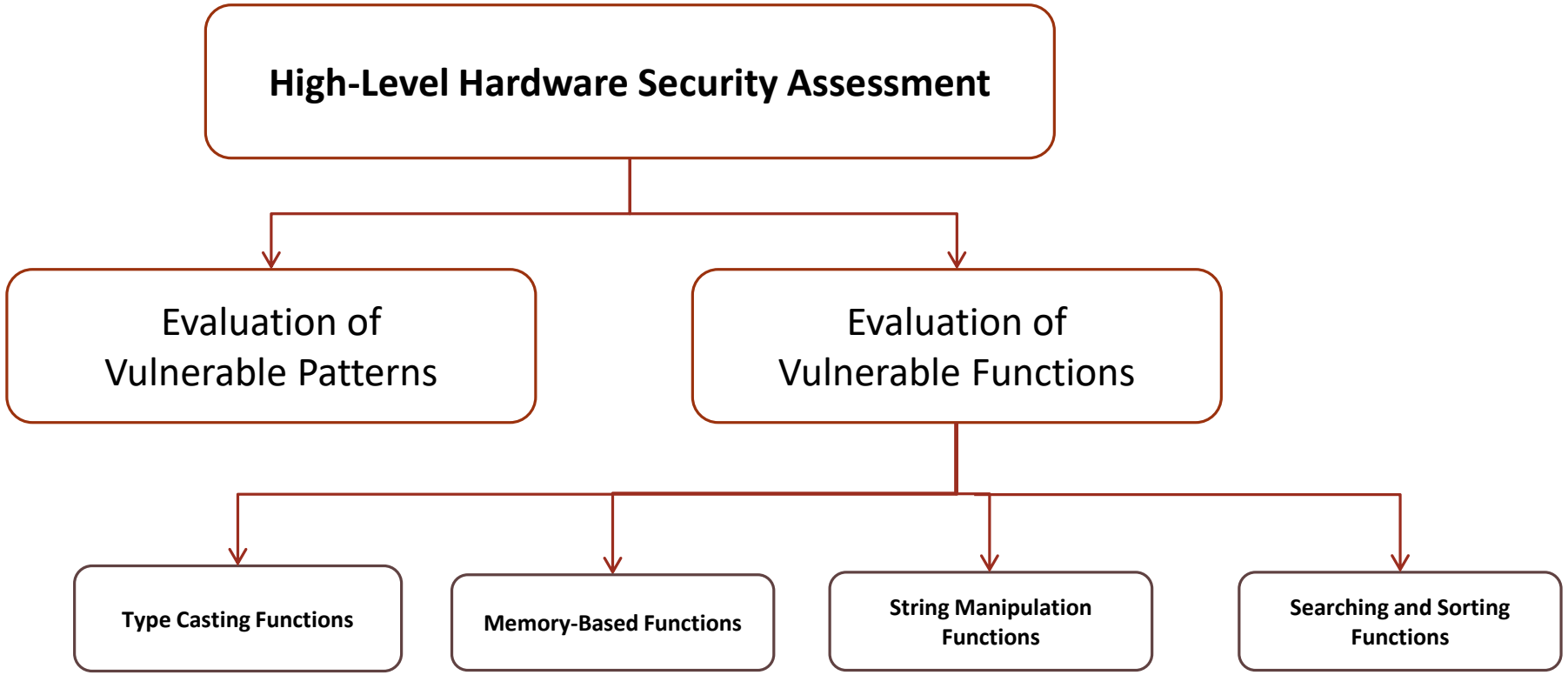
Our Approach

checkpoints Insertion at the important points of control flow for evaluation [3]



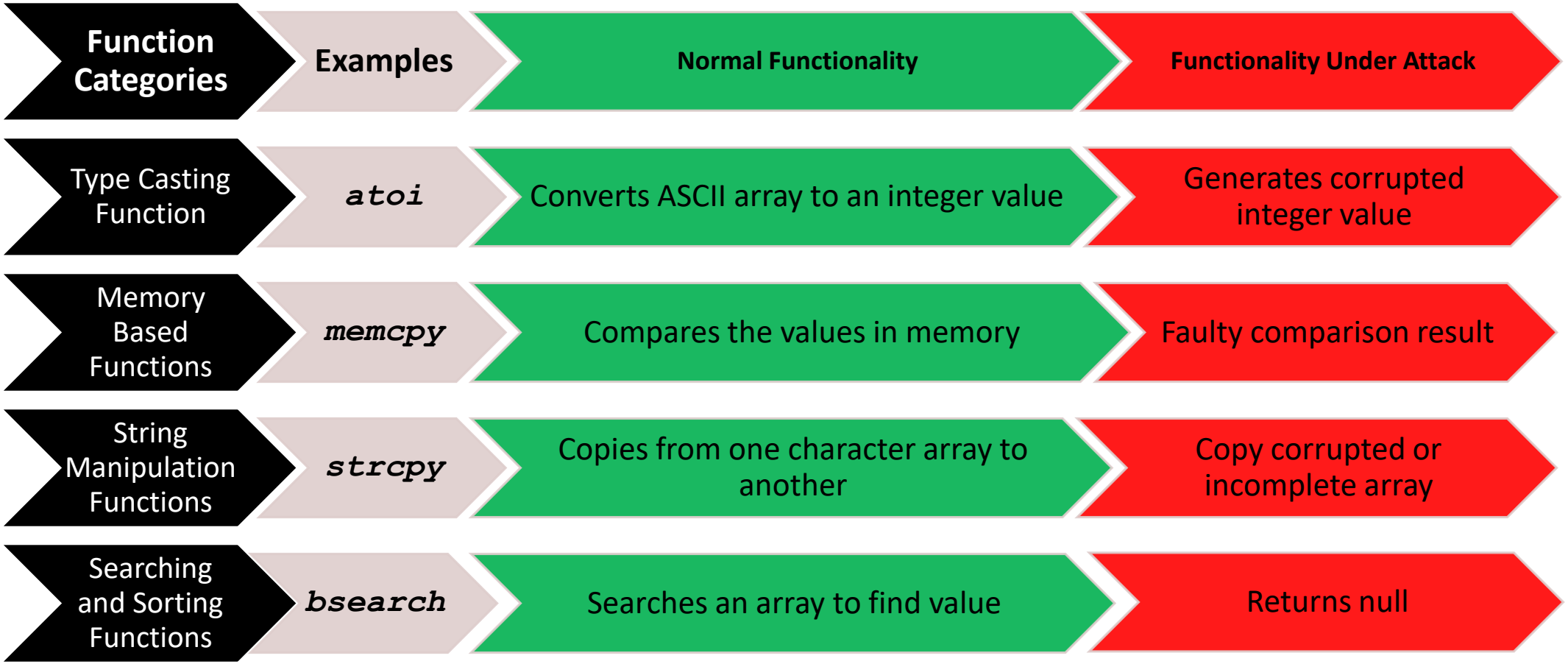


High-Level Hardware Security Assessment:
Our Approach



High-Level Hardware Security Assessment:

Our Approach



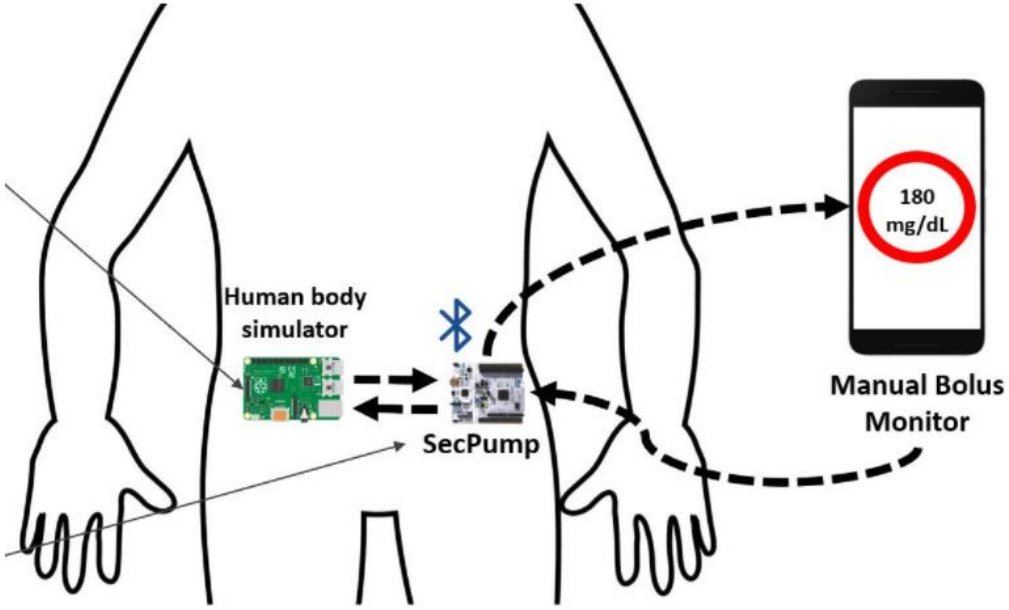
High-Level Hardware Security Assessment:

Our Approach on a Case-Study

Sec-Pump:

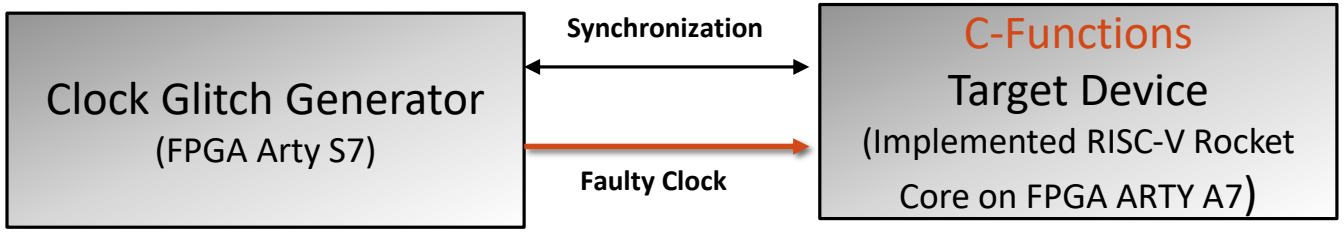
An Open-Source Secured Medical Application (https://github.com/r3glisss/SecPump)

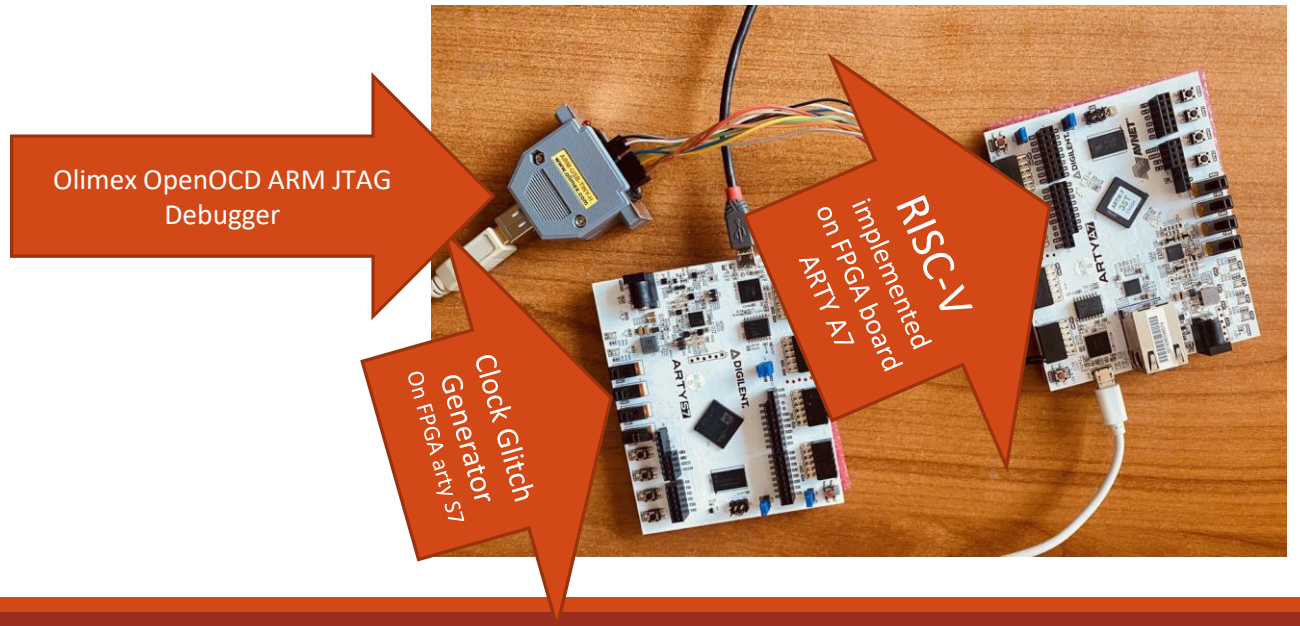
`atoi` Modify Cure Module	**`memset`** Delete Cure Module
`strcpy` Modify Cure Module	**`bsearch`** Library Module



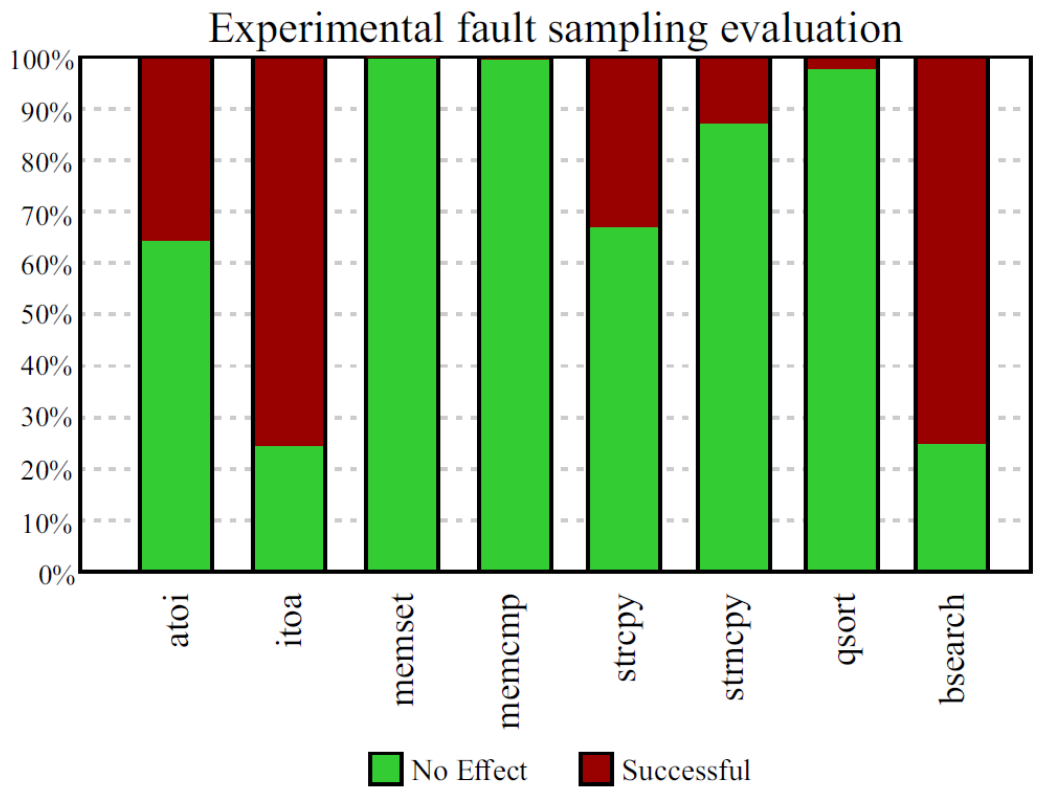
High-Level Hardware Security Assessment:

Our Experimental Platform





High-Level Hardware Security Assessment:
Experimental Results



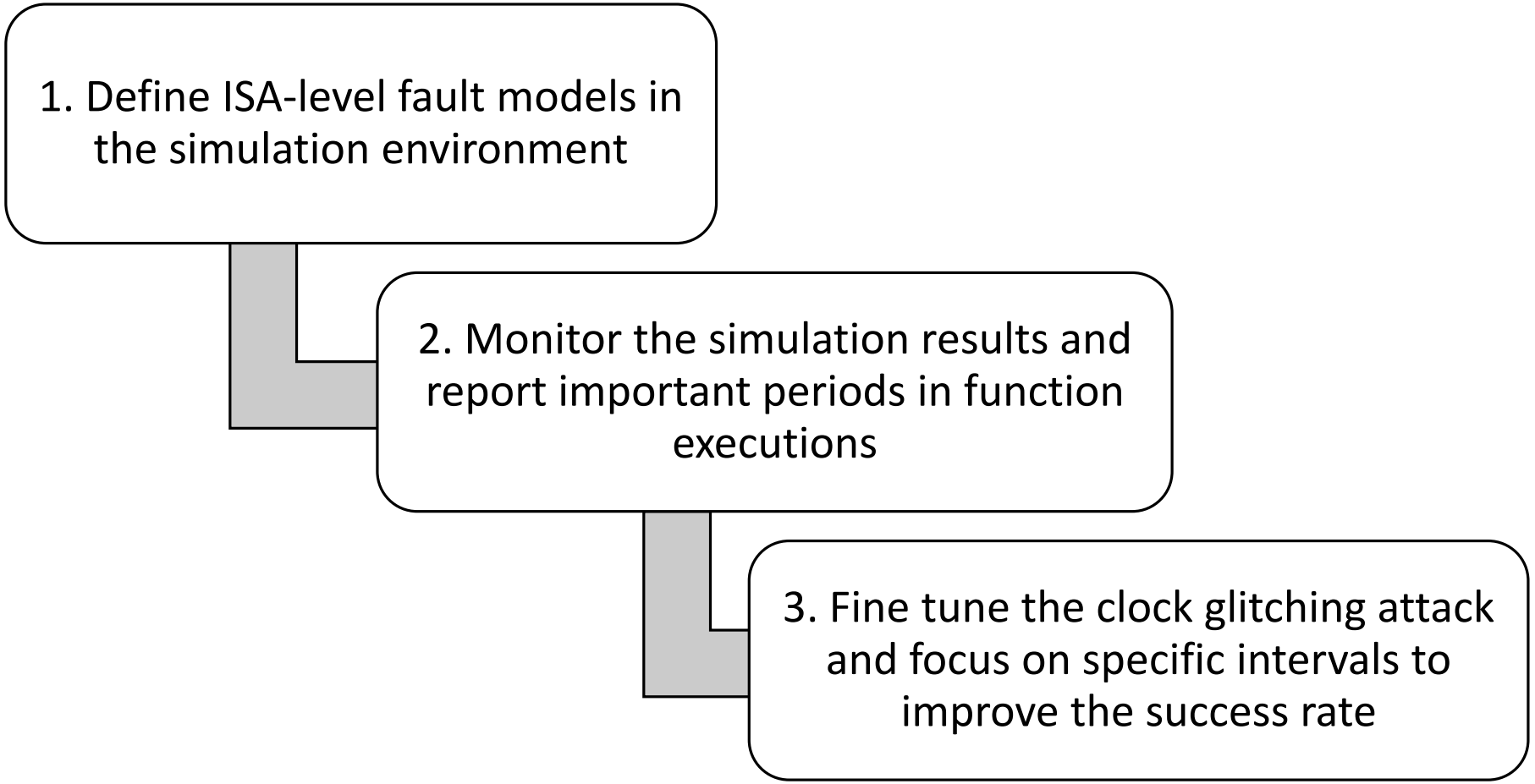
[4]

Outline

- Motivation
- Introduction
- High-Level Hardware Security Assessment
- Fine Tuned Experimental Evaluation for RISC-V Processors
- Conclusions and Future Works
- References

Fine Tuned Experimental Evaluation for RISC-V Processors:

Our Approach

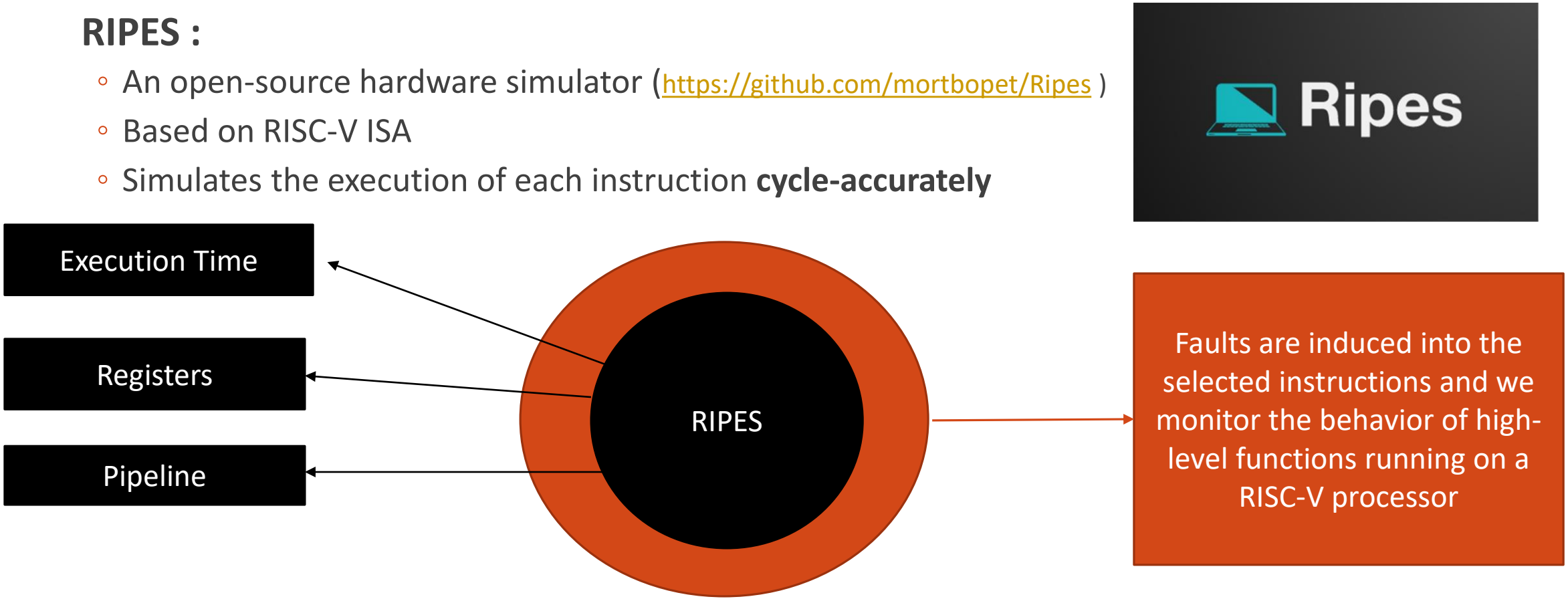


Fine Tuned Experimental Evaluation for RISC-V Processors:

Simulation Environment

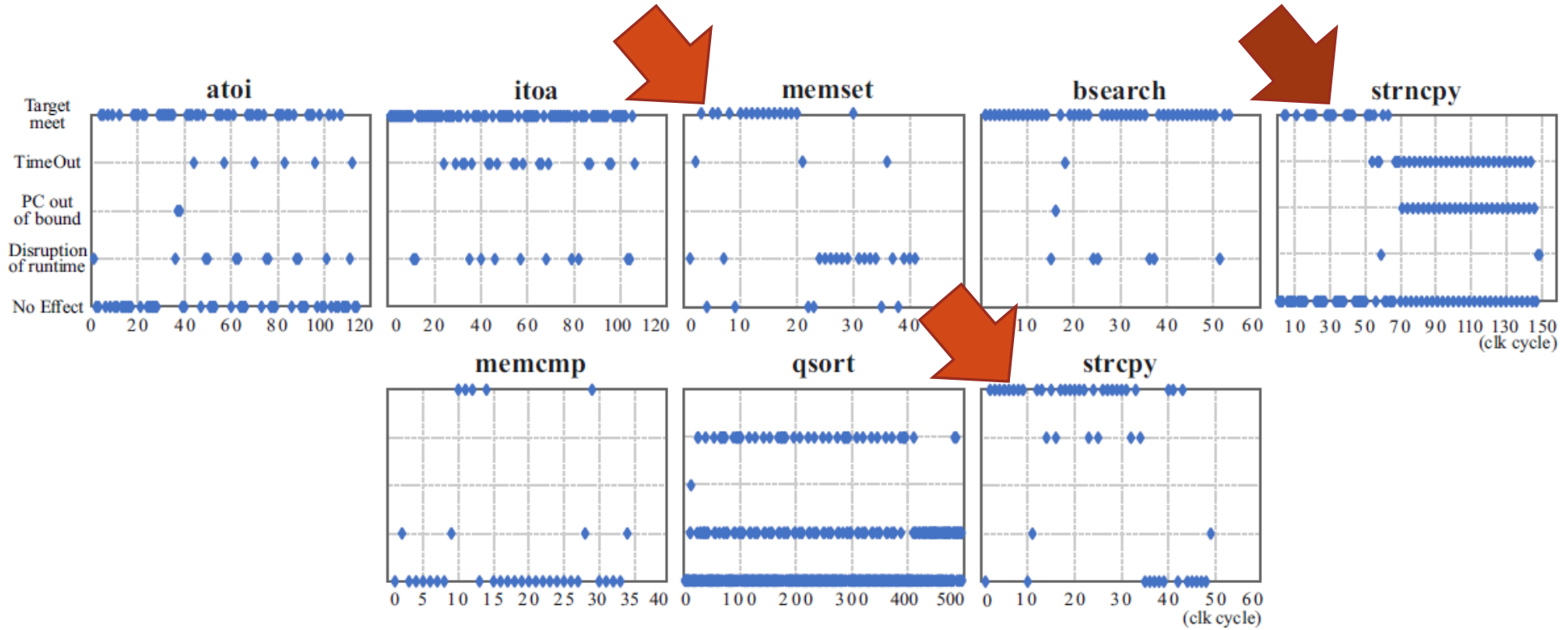
RIPES :

- An open-source hardware simulator (https://github.com/mortbopet/Ripes)
- Based on RISC-V ISA
- Simulates the execution of each instruction **cycle-accurately**



Fine Tuned Experimental Evaluation for RISC-V Processors:

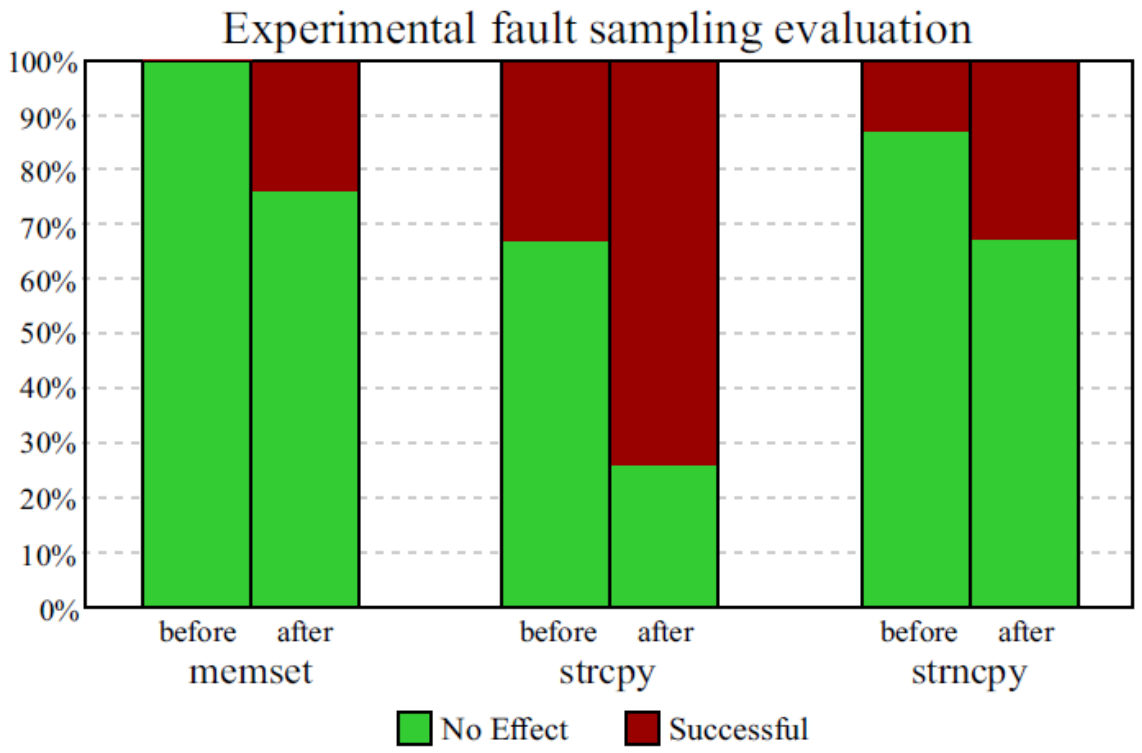
Simulation Results



Functions such as **memset, strncpy, and strcpy** are more vulnerable at their initial execution clock cycles.

Fine Tuned Experimental Evaluation for RISC-V Processors:

Results



Outline

- Motivation
- Introduction
- High-Level Hardware Security Assessment
- Fine Tuned Experimental Evaluation for RISC-V Processors
- Conclusions and Future Works
- References

Conclusions

1. We have reviewed all the existing clock glitching platform and extract their important characteristics
2. We have developed our open-source and practical platform which can help non-security specialists and developers to make their applications robust against low-cost but effective physical attacks
3. We have presented high-level evaluation approaches for common functions and patterns
4. Our experimental results have been improved using a ISA-Level simulation environment

Future Works

1. Demonstrating the potential risks of overlooking such vulnerabilities in different case-studies
2. Building up a database on clock glitch configurations for embedded application assessments
3. Assessing the vulnerability of different patterns and functions depending on their location in an embedded application by using symbolic executions
4. Proposing software level mitigation patterns/countermeasures

References

[1] Kazemi, Zahra, David Hely, Mahdi Fazeli, and Vincent Beroulle. 2020. "A Review on Evaluation and Configuration of Fault Injection Attack Instruments to Design Attack Resistant MCU-Based IoT Applications" *Electronics* 9, no. 7: 1153. https://doi.org/10.3390/electronics9071153

[2] Z. Kazemi, A. Papadimitriou, I. Souvatzoglou, E. Aerabi, M. M. Ahmed,D. Hely, and V. Beroulle, “On a low cost fault injection framework for security assessment of cyber-physical systems: Clock glitch attacks,” in 2019 IEEE 4th International Verification and Security Workshop (IVSW). IEEE, 2019, pp. 7–12.

[3] Z. Kazemi, M. Fazeli, D. Hely, and V. Beroulle, “Hardware security vulnerability assessment to identify the potential risks in a critical embedded application,” in 2020 IEEE 26th International Symposium on On-Line Testing and Robust System Design (IOLTS). IEEE, 2020, pp.1–6.

[4] Z. Kazemi, A. Norollah, A. Kchaou, M. Fazeli, D. Hely, and V. Beroulle, “An in-depth vulnerability analysis of RISC-V micro-architecture against fault injection attack,”International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFTS), 2021.

Thank you for your attention!

Questions?