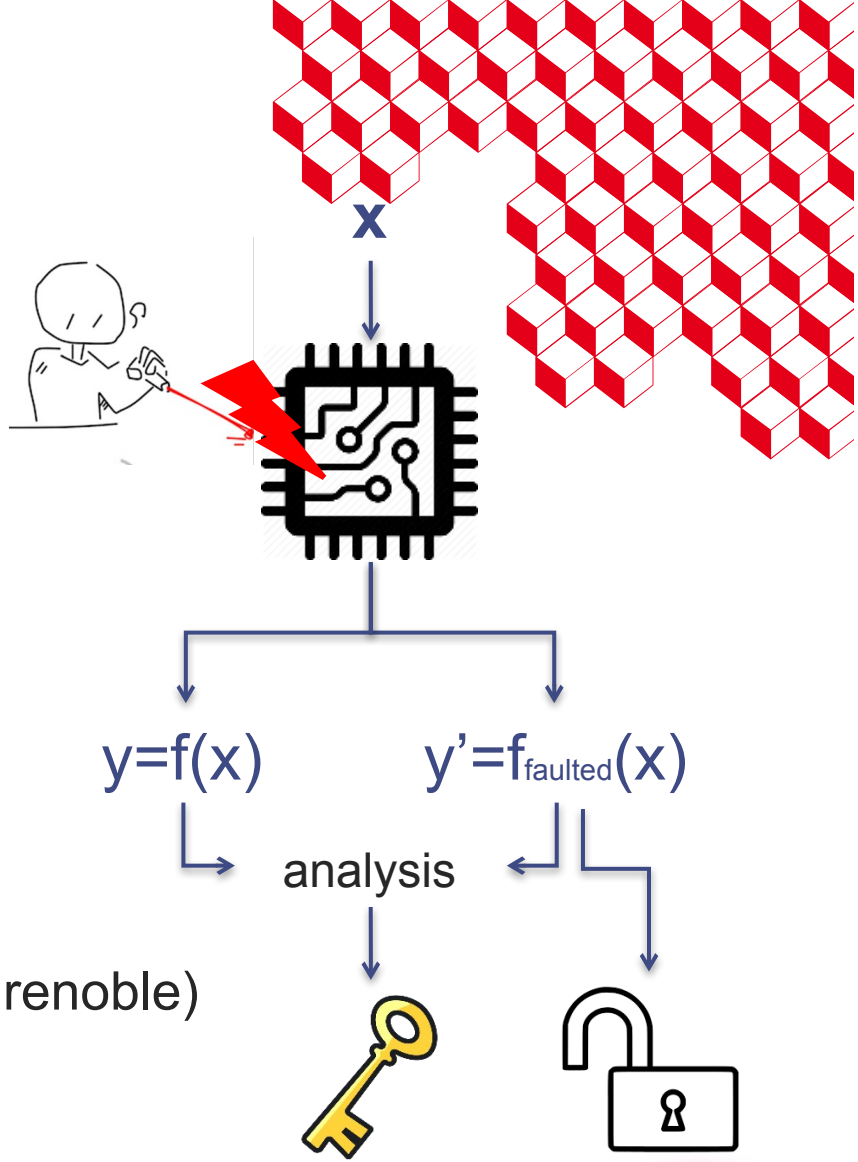
Pre-Silicon Analysis of Microarchitectures Against Fault Injection Attacks



Mathieu Jan (CEA-List / Saclay), Damien Couroussé (CEA-List / Grenoble)

November 20th 2025

Outline

Pre-silicon Security Analysis Methodology

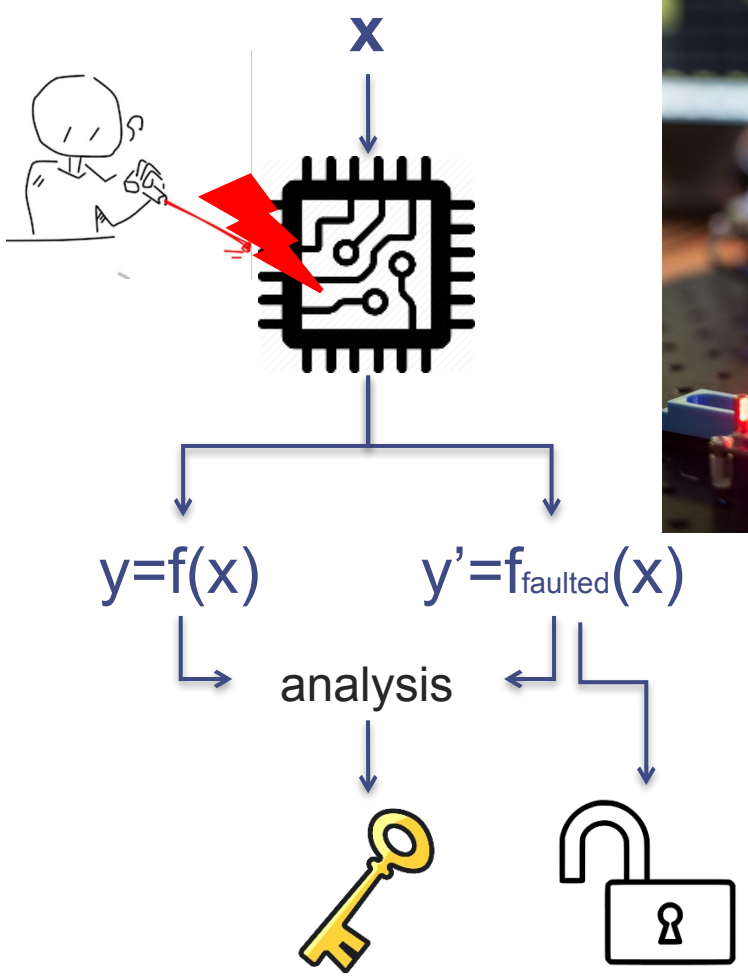
- Motivations
- Workflow overview
- Challenge: scalability vs. state space explosion

k-Fault Resistant Partitioning

- Overview
- OpenTitan case study

Conclusion & Next Steps

Fault Injection Attacks (FIA): motivations



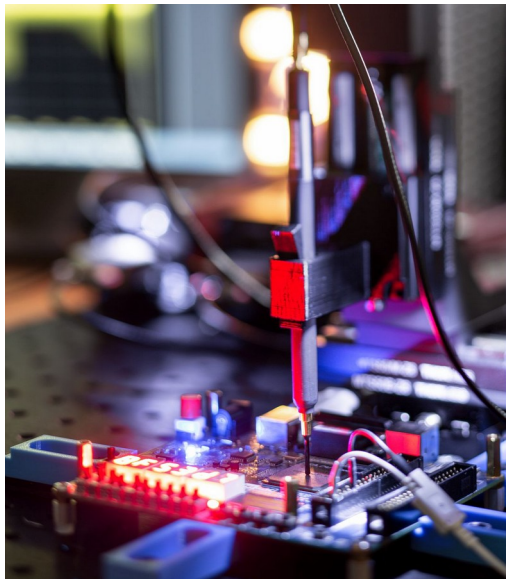


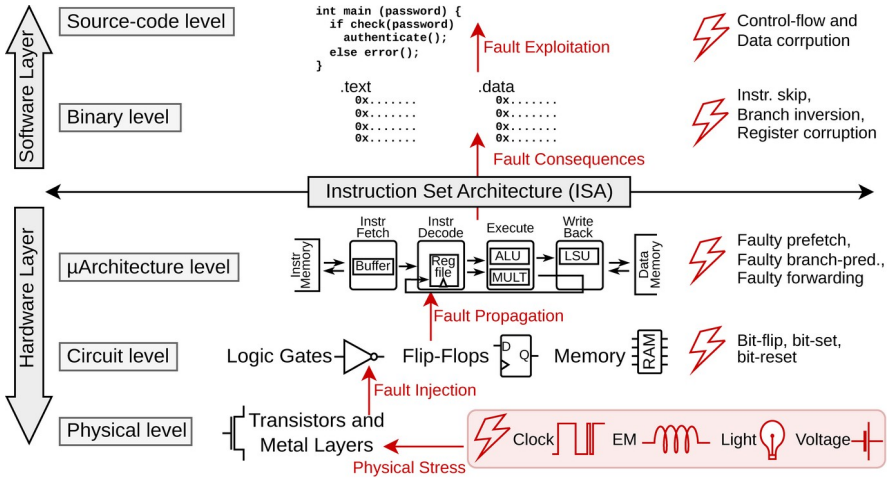
Photo credit:
Christian Morel /
CEA

Security evaluation (real system & real testbench) / **security verification** (pre-silicium analyses)

Turning attention to processor microarchitecture

- FIA on processor pipelines can bypass SW protections [Yuce, 2016]
- Importance of hidden microarchitectural registers and mechanisms [Laurent, 2021][Tollec, 2022]

Automated Joint HW-SW analysis is mandatory!



[Yuce, 2016] Software Fault Resistance is Futile: Effective Single-Glitch Attacks. 10.1109/FDTC.2016.21
[Laurent, 2021] Bridging the Gap between RTL and Software Fault Injection. 10.1145/3446214
[Tollec, 2022] Exploration of fault effects on formal RISC-V microarchitecture models. 10.1109/FDTC57191.2022.00017

µArchiFI+: our fault injection analysis tool

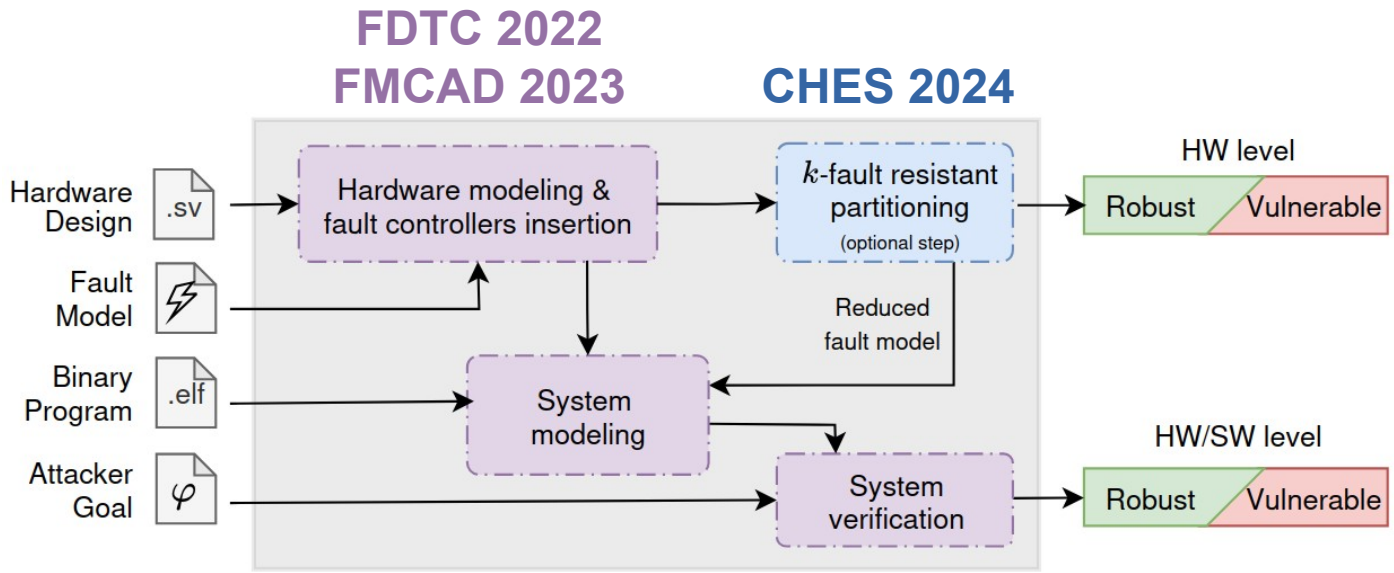
https://list.cea.fr/fr/page/µarchifi/

To do what?

- Identify fault effects at the HW level and check whether it generates vulnerabilities at SW level
- Prove the robustness of (HW/SW) countermeasures to secure a system
- Reduce design costs and delays (avoid HW respin)

How?

- Exhaustively analyze the impact of fault injections over systems (HW/SW) using formal methods
- Configurable fault model: (gate) location, effect, timing and fault order
- Analyses at RTL/netlist levels and agnostic to EDA flows



Open-source flows

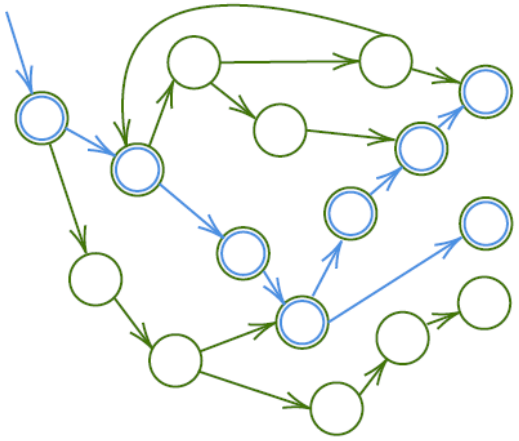
µArchiFI (FDTC 2022 & FMCAD 2023):
https://github.com/CEA-LIST/uArchiFI

k-Fault-Resistant Partitioning (CHES 2024):
https://github.com/CEA-LIST/Fault-Resistant-Partitioning

Supported platforms

- HW: OpenHW Group processors (CV32P, CV32S), OpenTitan secure element (Secure Ibex), Crypto circuits (AES-128, etc.),
- SW level: FISCC benchmark, Tiny AES, SecureBoot

State Space: limitation of a monolithic HW/SW analysis



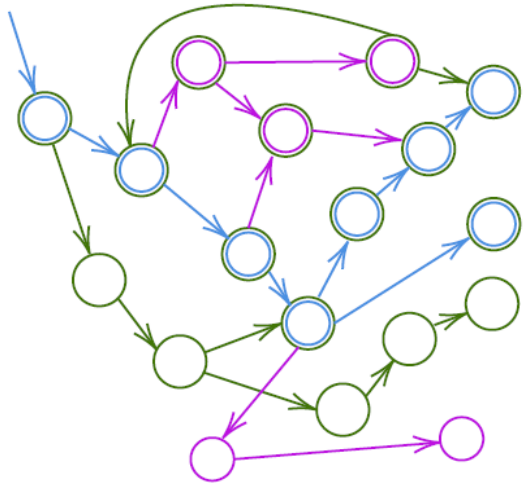
—Hardware
—Software

Factors to state space explosion

- **Large HW designs, large programs (SW)**
  - Focus on specific parts to be analysed
  - Simulation on other parts ...

Use case names		*I - Robust Software*	*II - Robust Hardware*	*III - Cryptographic Software*
Hardware design	name:	**CV32E40P (Riscy)** - RISC-V - 4 stages	**Secure Ibex** - RISC-V - 2 stages - dual core	**Ibex** - RISC-V - 2 stages
	gates:	2842	4422	1983
	FFs:	179	211	114
	size*(GE):	89954	61452	26327
Software program		VerifyPIN_V7 [Dur+16]	VerifyPIN_V1 [Dur+16]	Key Schedule (AES) [kok19]
Attacker Goal φ		Bypass authentication without triggering SW alert	Bypass authentication without triggering HW alert	Set to 0 a byte in the penultimate round key
Fault model $\mathcal{F}$	location:	Sequential logic Control Path	Sequential logic Redundant CPU Core	Combinational logic Execute stage of CPU
	effect:	Symbolic	Symbolic	Reset
	timing:	60:*	*	*
Number of FIs N		1	5	2
BMC depth k		75	46	38
Verification results		φ is reachable	φ is unreachable	φ is unreachable

State Space: limitation of a monolithic HW/SW analysis



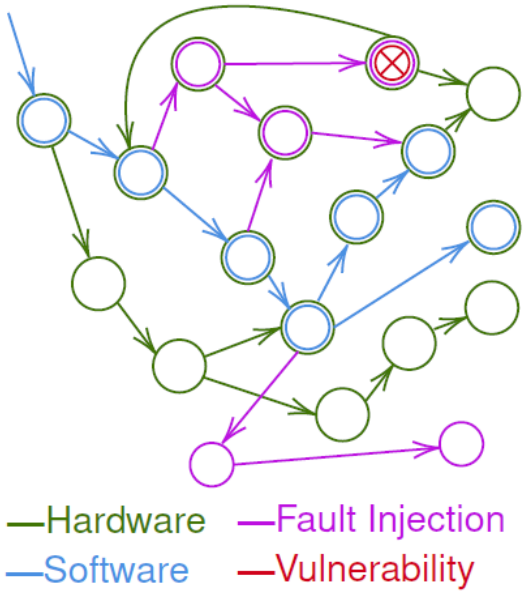
Leading to ~ 20 000 injections points

Factors to state space explosion

- **Large HW designs, large programs (SW)**
  - Focus on specific parts to be analysed
  - Simulation on other parts ...
- **Faults incur extra analysis complexity**
  - In particular, multiple faults: combinatorial explosion
  - Restrict areas to be faulted (combinational / sequential logics)

Use case names		*I - Robust Software*	*II - Robust Hardware*	*III - Cryptographic Software*
Hardware design	name:	**CV32E40P (Riscy)** - RISC-V - 4 stages	**Secure Ibex** - RISC-V - 2 stages - dual core	**Ibex** - RISC-V - 2 stages
	gates:	2842	4422	1983
	FFs:	179	211	114
	size*(GE):	89954	61452	26327
Software program		VerifyPIN_V7 [Dur+16]	VerifyPIN_V1 [Dur+16]	Key Schedule (AES) [kok19]
Attacker Goal φ		Bypass authentication without triggering SW alert	Bypass authentication without triggering HW alert	Set to 0 a byte in the penultimate round key
Fault model $\mathcal{F}$	location:	Sequential logic Control Path	Sequential logic Redundant CPU Core	Combinational logic Execute stage of CPU
	effect:	Symbolic	Symbolic	Reset
	timing:	60:*	*	*
Number of FIs N		1	5	2
BMC depth k		75	46	38
Verification results		φ is reachable	φ is unreachable	φ is unreachable

State Space: limitation of a monolithic HW/SW analysis

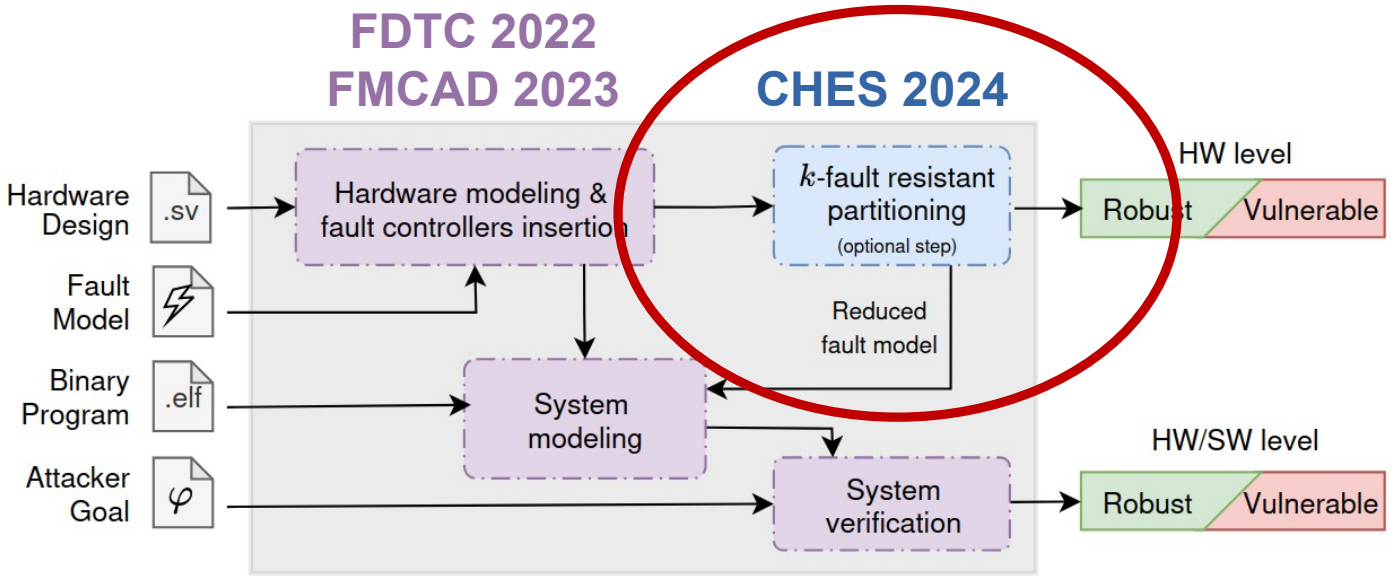


Factors to state space explosion

- **Large HW designs, large programs (SW)**
  - Focus on specific parts to be analysed
  - Simulation on other parts ...
- **Faults incur extra analysis complexity**
  - In particular, multiple faults: combinatorial explosion
  - Restrict areas to be faulted (combinational / sequential logics)
- Verification using Bounded Model Checking (BMC) -> **limited unrolling**

Use case names		*I - Robust Software*	*II - Robust Hardware*	*III - Cryptographic Software*
Hardware design	name:	**CV32E40P (Riscy)** - RISC-V - 4 stages	**Secure Ibex** - RISC-V - 2 stages - dual core	**Ibex** - RISC-V - 2 stages
	gates:	2842	4422	1983
	FFs:	179	211	114
	size*(GE):	89954	61452	26327
Software program		VerifyPIN_V7 [Dur+16]	VerifyPIN_V1 [Dur+16]	Key Schedule (AES) [kok19]
Attacker Goal φ		Bypass authentication without triggering SW alert	Bypass authentication without triggering HW alert	Set to 0 a byte in the penultimate round key
Fault model $\mathcal{F}$	location:	Sequential logic Control Path	Sequential logic Redundant CPU Core	Combinational logic Execute stage of CPU
	effect:	Symbolic	Symbolic	Reset
	timing:	60:*	*	*
Number of FIs N		1	5	2
BMC depth k		75	46	38
Verification results		φ is reachable	φ is unreachable	φ is unreachable

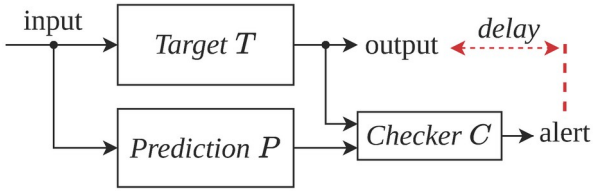
k-Fault Resistant Partitioning: one step further in scalability



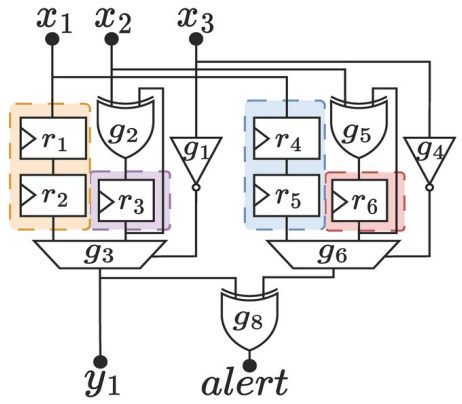
k-Fault Resistant Partitioning [CHES, 2024]

Intuition

Concurrent Error Detection Scheme



Partitioning example with *k = 1*



Theorem

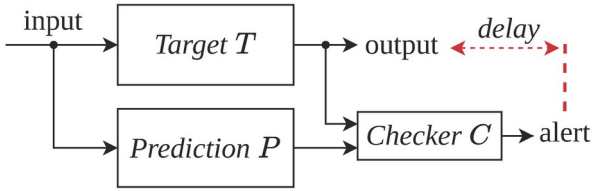
k-fault resistant partitioning ⇒ k-fault security

[CHES, 2024] Fault-Resistant Partitioning of Secure CPUs for System Co-Verification against Faults. 10.46586/tches.v2024.i4.179-204

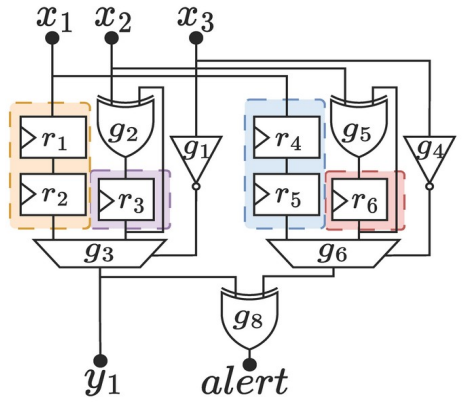
k-Fault Resistant Partitioning [CHES, 2024]

Intuition

Concurrent Error Detection Scheme



Partitioning example with *k = 1*



Theorem

k-fault resistant partitioning ⇒ k-fault security

Validation

Research Objectives

- No CPUs analysis tools or benchmarks available for comparison
- Research Questions
  - Evaluate verification performance
  - Consider multiple-fault attacks
  - Compare with prior work like FIVER [RR21]

Impeccable Circuits [AM19]

- Symmetric Bloc Ciphers (AES, LED, Simon, Skinny …) protected with Error Detection Codes (EDC)
- Designed to detect up to 3 faults (up to 7 faults for AES)

Test case	# faults	FIVER	kFRP (ours)
Skinny	2		10 sec.
Skinny	3		39 sec. **(*)**
AES	2	130 h	4 h
AES	3	∞	55 h **(*)**
(*) we identify exploitable faults in the checker of Skinny and AES			

[AM19] Aghaie, Anita, et al. "Impeccable circuits." *IEEE Transactions on Computers* (2019)
[RR21] Richter-Brockmann, Jan, et al. "Fiver–robust verification of countermeasures against fault injections", *CHES 2021*

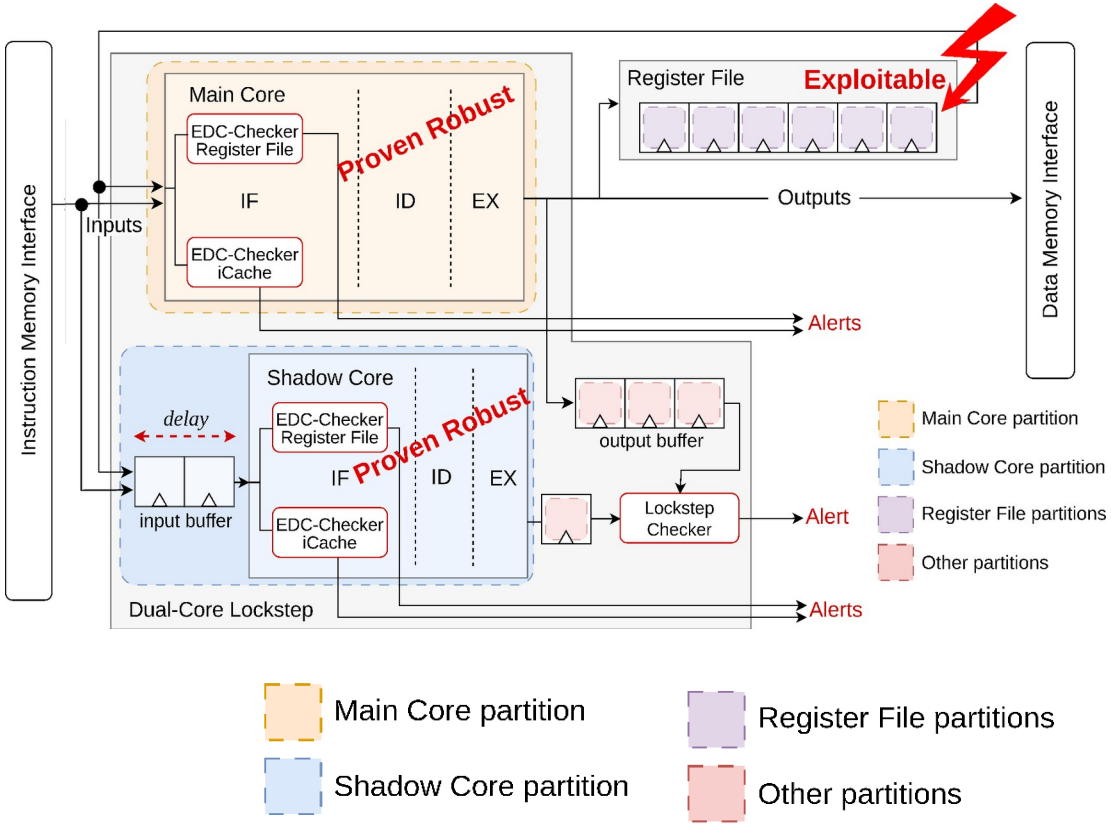
Case study: analysis of a Secure Element

OpenTitan: open-source Root of Trust [JR18]

- Developed by a consortium of leading players in digital systems and cybersecurity
  - TRL 8 according to LowRISC
- Secure-Ibex processor (development version) [Lo18]
  - Embbeds several HW countermeasures: Dual Core Lockstep (DCLS), Error Detection Codes in Register File

Results of applying our pre-silicon methodologies

- **Fault model:** single transient bit-flip everywhere at any time
- **Vulnerability Reported:** 172 exploitable faults — allow reading from an incorrect register location
  - We proposed a security fix and formally prove it using our methodology
  - **Our fix was integrated into the OpenTitan project**
- **Secure Ibex** is now **proven 1-fault secure** unconditionally of the executed software
  - Prove 1-fault security (DCLS + Error detection codes) in 68 hours



[JR18] Johnson, Scott, et al. "Titan: enabling a transparent silicon root of trust for cloud." *Hot Chips: A Symposium on High Performance Chips*. Vol. 194. 2018.
[Lo18] lowRISC. Ibex RISC-V Core github repository. https://github.com/lowRISC/ibex. Accessed: February 22, 2024.

k-FRP's scalability improvements

Evaluation of the Secure Ibex and its modules using k-fault-resistant partitioning

Circuit Characteristics			Faults		Algorithm 1 Performance			Results		
Name	Size (GE)	Regs (#)	Loc. (#)	Order k	*BuildPartitioning*		*CheckIntegrity* Time	Partitions (#)	Exploitable Faults	
					Iter. (#)	Time			$\mathcal{P}'$ (#)	$\mathcal{F}'$ (#)
Register File	12 075	1 326	8 331	1	172	38 s	53 s	1 326	0	172
			1 326[a]	3	1	349 s	344 s	1 326	0	0
Register File with fix	11 913	1 326	8 667	1	1	17 s	73 s	1 326	0	0
			1 326[a]	3	1	135 s	383 s	1 326	0	0
DCLS	117 998	5 918	116 561	1	508	20 h 12	5 h 10	1 108	0	0
				2	11	11 s	—	445	—	—
Secure Ibex (no iCache)	130 194	7 248	125 080	1	1	10 h 45	30 h 50	2 438	0	0 (+172)
				2	48	53 s	—	421	—	—

[a] Restricted fault model targeting the sequential logic only

	kGE (#)	faults (#)	CPU cycles (#)	analysis level
µArchiFI	20-90	1 (* ≤5)	≤100	RTL
kFRP	130	≤3	unlimited	netlist

System Co-Verif. [CHES 2024]: SW Case Studies

- **Register File vulnerability is exploitable on Secure Ibex**

OpenTitan's Secure Boot - 1st stage

- Goal: **Bypass memory signature check**
- # instructions: **2 526**
- # faults: **122 048**
- Performance: **2.5 hours** (8 threads)
- Results: **Secure**

VerifyPIN [DP16]

- Goal1: **Bypass authentication**
- Goal2: **Increase max number of tries**
- # instructions: **187**
- # faults: **7 424**
- Performance: **6 mins** (1 thread)
- Results: **Insecure**

DFA on tiny AES [Ko19]

- SW implementation of AES

- Goal1: **DFA on key schedule**
- # instructions: **221**
- # faults: **5 760**
- Performance: **7 mins** (2 threads)
- Results: **Insecure**

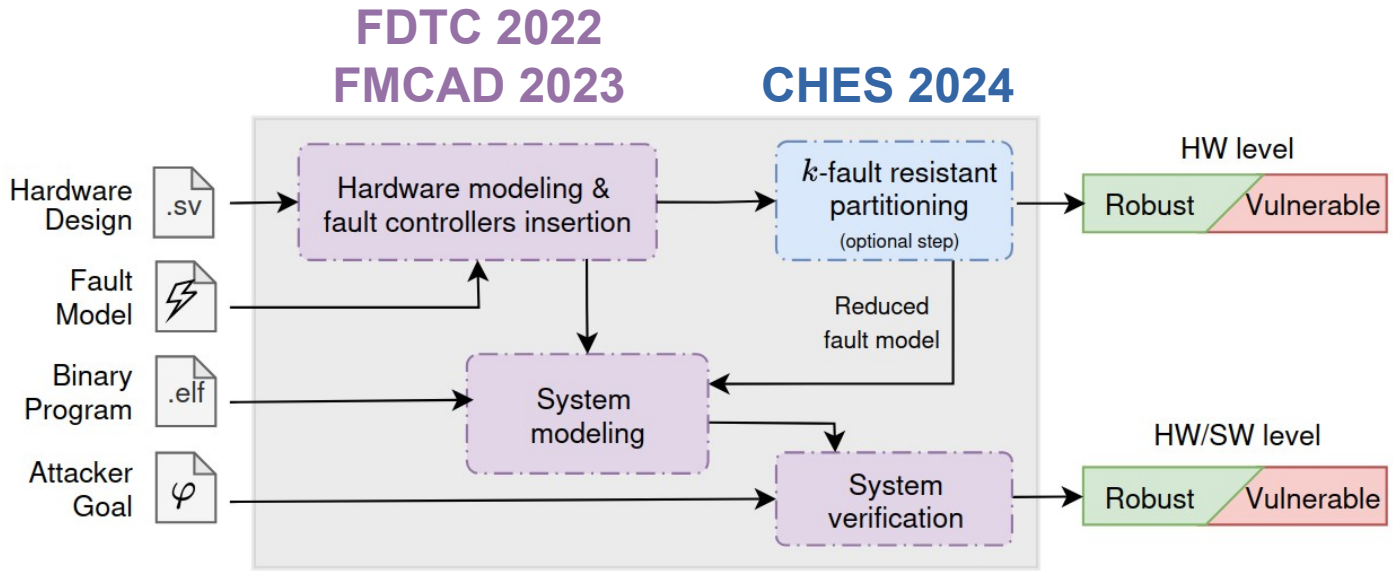
- Goal2: **DFA on 7th AES round**
- # instructions: **1 144**
- # faults: **38 912**
- Performance: **29 mins** (8 threads)
- Results: **Insecure**

Provable system security, despite HW vulnerabilities

→ **Avoids HW respin**

[DP16] Dureuil, Louis, et al. "FISSC: A fault injection and simulation secure collection." *Computer Safety, Reliability, and Security: 35th International Conference, SAFECOMP* 2016.
[Ko19] kokke. Tiny AES, release 1.0. https://github.com/kokke/tiny-AES-c, 2019. Accessed: February 22, 2024.

µArchiFI+: pre-silicium formal methodologies for FIA robustness



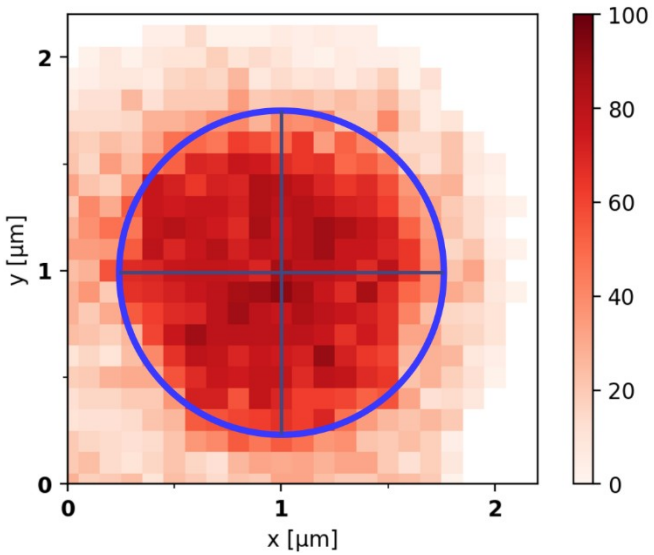
Take-away messages

- Pre-silicon methodologies to identify vulnerabilities / prove robustness of HW/SW systems against FIA
  - Formal & exhaustive approach using a parametric fault model
  - RTL- and netlist-level analyses (leveraging HW countermeasures to speedup analyses)
  - Agnostic to EDA flows
  - Better understanding of subtle fault effects due to HW/SW coupling
  - Reduce design costs/delays (e.g. prove SW countermeasures to avoid HW respin) & increase confidence before certifications
- Supports various processors (secured with HW/SW countermeasures) + cryptographic HW accelerators
  - **Found vulnerability in an industry-grade (TRL 8) secure element**

µArchiFI+: next steps

Leveraging layout information to design new fault models

- A laser spot may fault several neighbour cells:
  → Select signals according to location constraints of laser spot
- Clock glitch fault models:
  → Leverage timing information



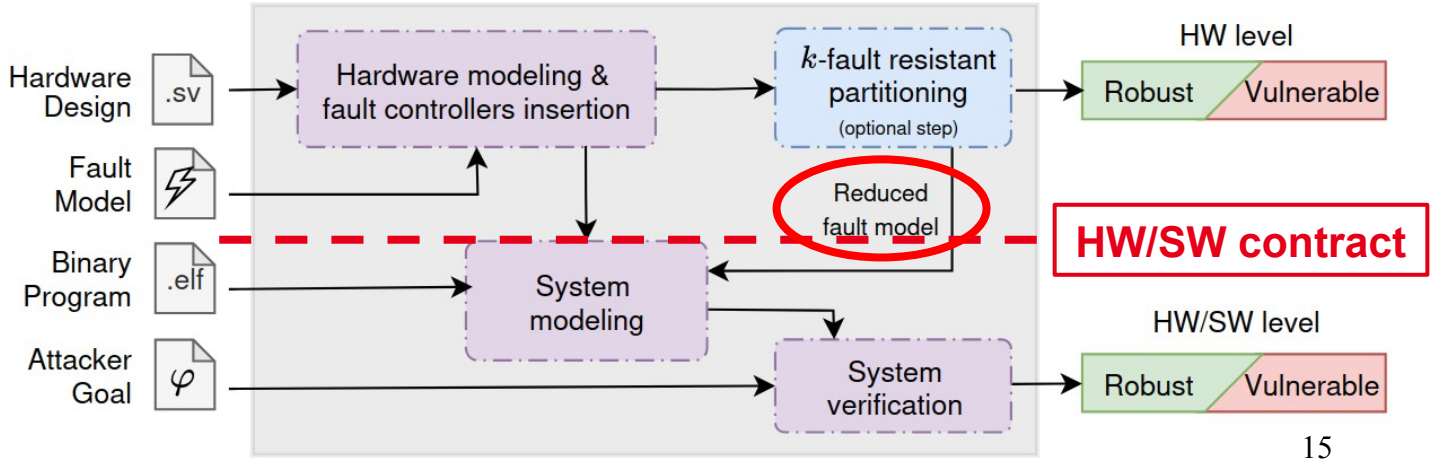
Laser characterization, 28nm bulk
CEA Leti, L. Mangin (2025)

Generation of fault characterization programs
(PhD Jonah Alle Monne – 2024-2027)

- Current practice for designing characterization programs: expert knowledge & educated guesses
- Automate programs generation according to target fault location or fault effect

HW/SW fault injection contracts
(PhD Israël Kafando – 2025-2028)

- Need for a model that supports reasoning at the HW and SW levels, *separately*
  → Contracts: formal security abstraction
  → Integrate fault-models in contracts



Thank you

Questions?

[FDTC 2022] S. Tollec, M. Asavoae, D. Couroussé, K. Heydemann, and M. Jan “Exploration of Fault Effects on Formal RISC-V Microarchitecture Models,” in FDTC, 2022.
[FMCAD 2023] S. Tollec, M. Asavoae, D. Couroussé, K. Heydemann, and M. Jan “ARCHIFI: Formal Modeling and Verification Strategies for Microarchitectural Fault Injections,” in FMCAD, 2023.
[CHES 2024] S. Tollec, V. Hadžic, P. Nasahl, M. Asavoae, R. Bloem, D. Couroussé, K. Heydemann, M. Jan, and S. Mangard “Fault-Resistant Partitioning of Secure CPUs for System Co-Verification against Faults,” IACR Transactions on Cryptographic Hardware and Embedded Systems (TCHES), 2024.