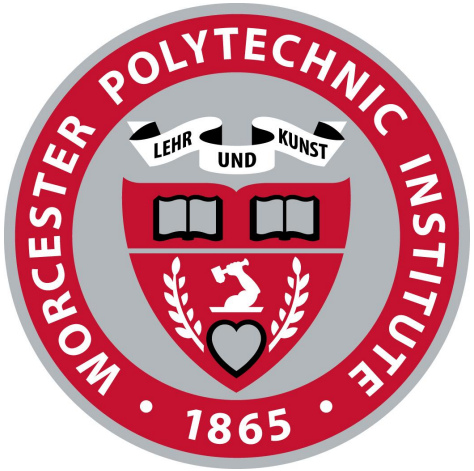
CAPRI 6: A Solution for Fault Root Cause Detection

Dillibabu Shanmugam, Zhenyuan Liu, Patrick Schaumont



IEEE MDTS

21st May 2025

Outline

Motivation

Hardware-Software abstraction layers

Fault root cause detection : CAPRI6

Experiments : Clock glitch & EMFI

Results

Summary

Motivation

Small faults in SoC's can have big consequences

Physical access by an attacker often possible

Tampering is a serious concern for vendors

Hardware bugs reduce reliability

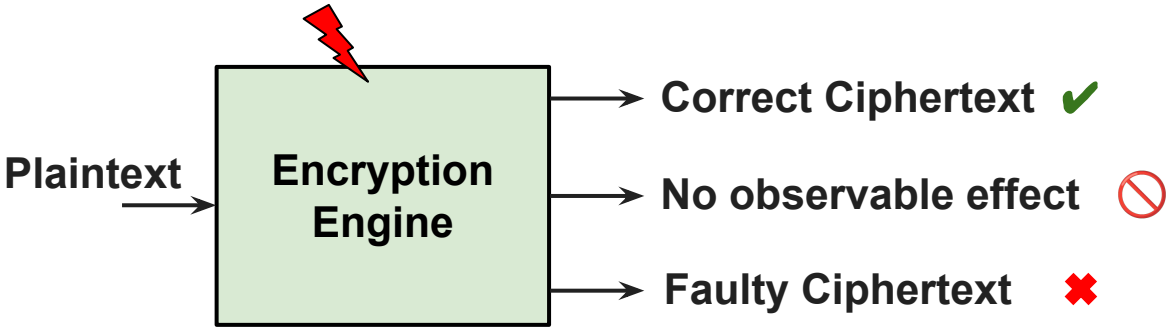
1 in 1000 CPUs experiences silent data corruption [1]

Emphasize the need to understand the impact of faults as well as thorough pre- and post-silicon testing.

[1] https://www.sigarch.org/silent-data-corruption-at-scale/

Why do we care?

Fault attacks on cryptographic engines have three outcomes:



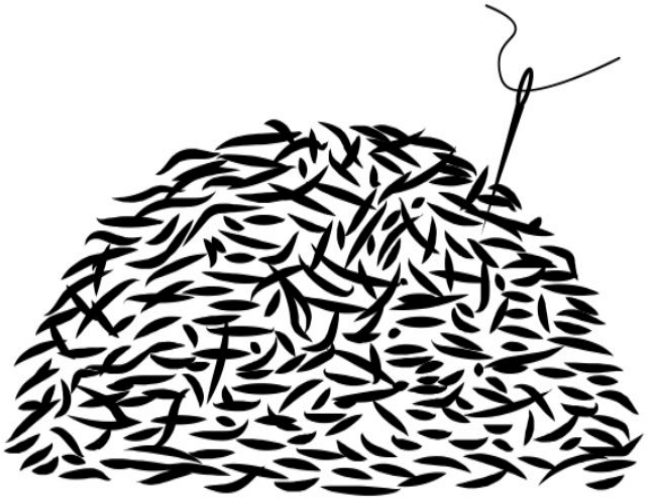
Key Challenges

Limited observability across HW/SW stack

Difficult to understand fault behaviour

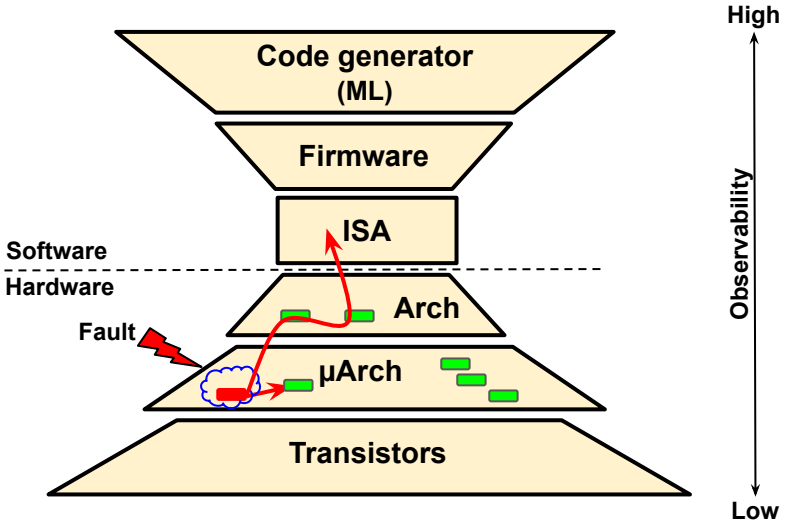
Pinpointing fault origin is challenging

HW/SW abstraction layers



Finding a needle in a haystack

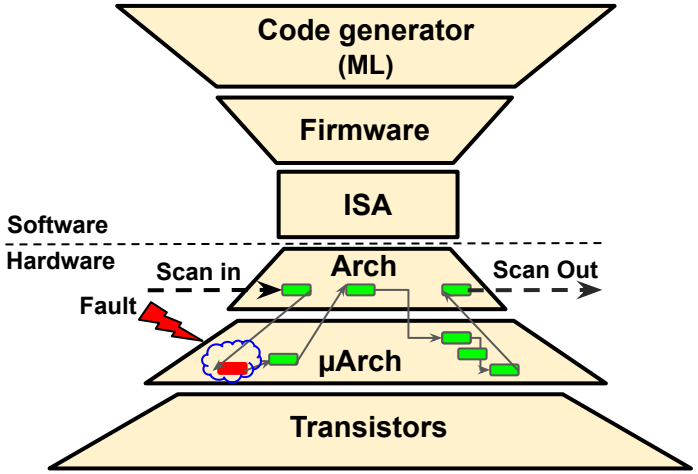
Fault: any unintended bit-flip

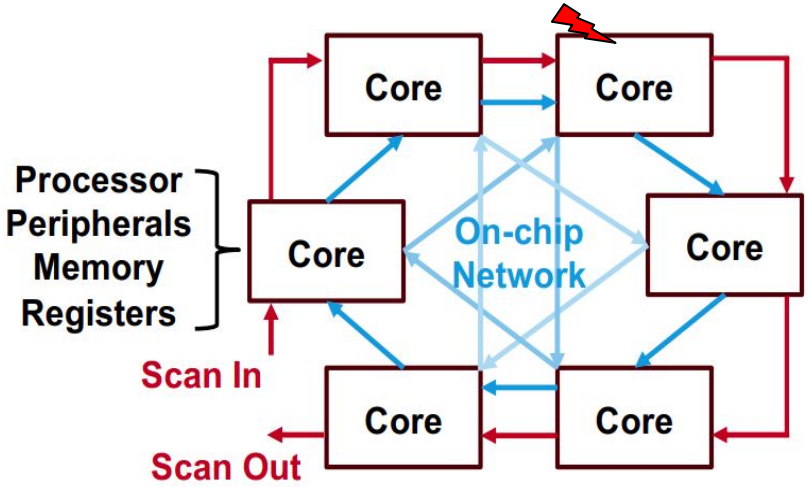


Finding a fault is like finding a needle in a haystack. Because faults can hide deep in the HW/SW stack, we need targeted pre- and post-silicon analysis.

Observability – Improving Insight via Scan Chains and Lockstep

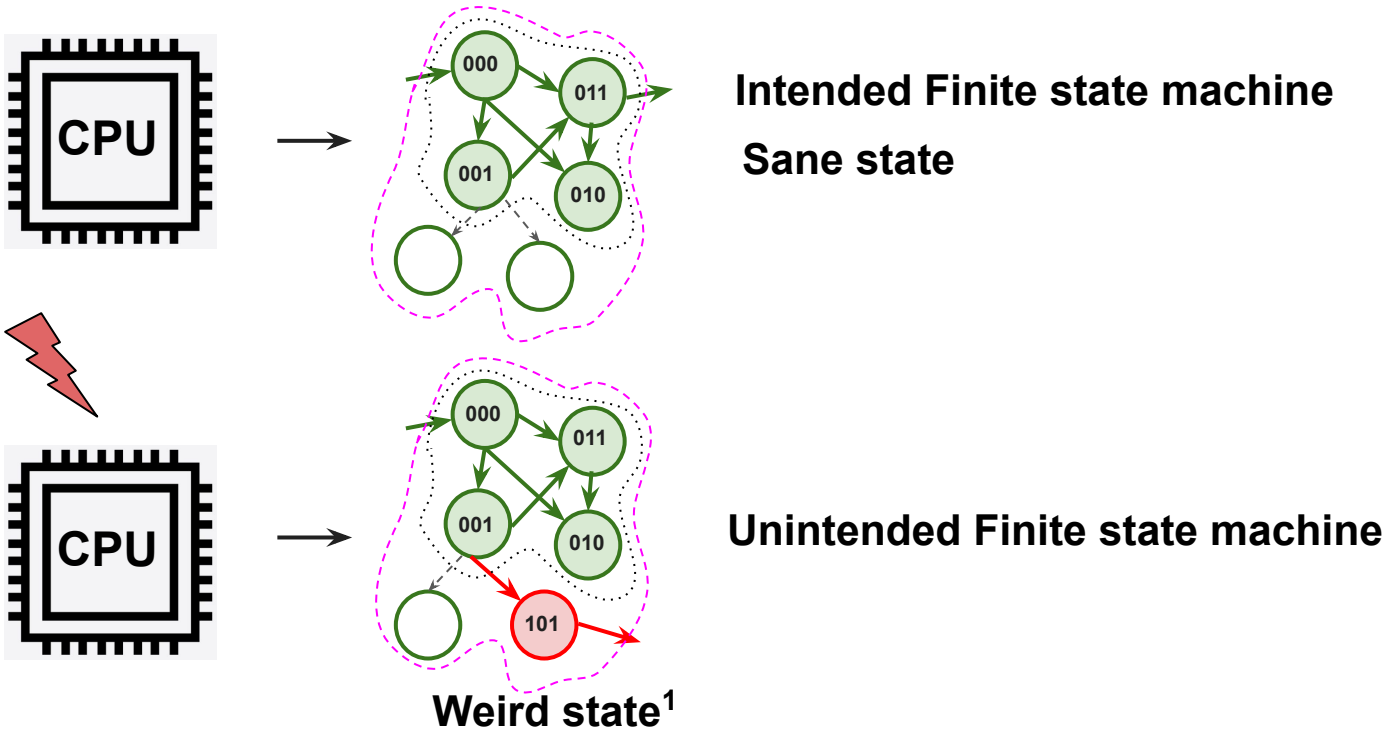
Scan chain and hardware redundancy in lock step





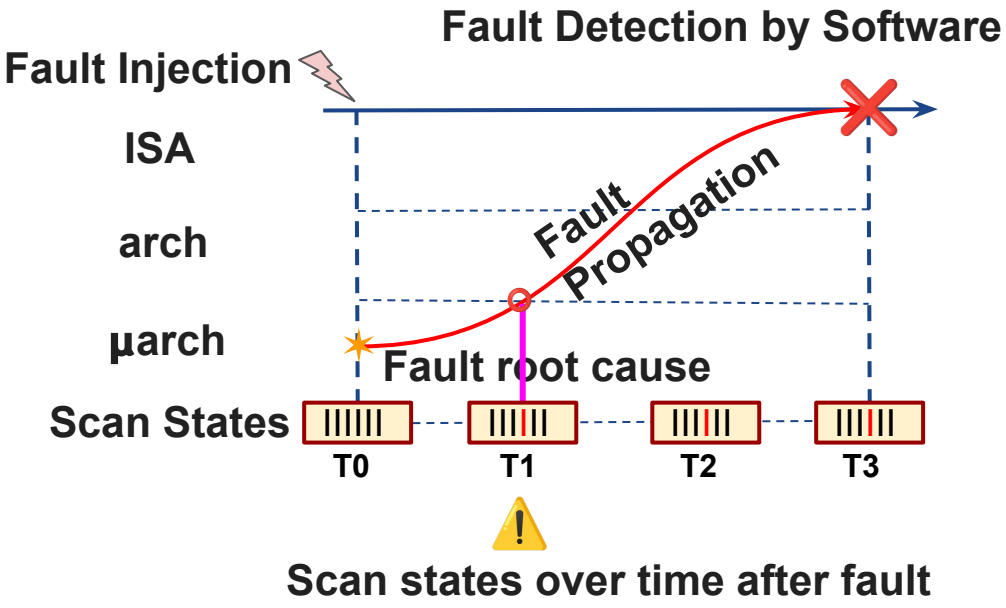
Understand fault behaviour : Sane vs Weird state

State Space Explosion (Difficult to Model All Faults)
The state space for a 1000-bit processor contains 2^{1000} possible state



[1]Concept of 'weird states' from Thomas Dullien, http://www.dullien.net/thomas/weird-machines-exploitability.pdf

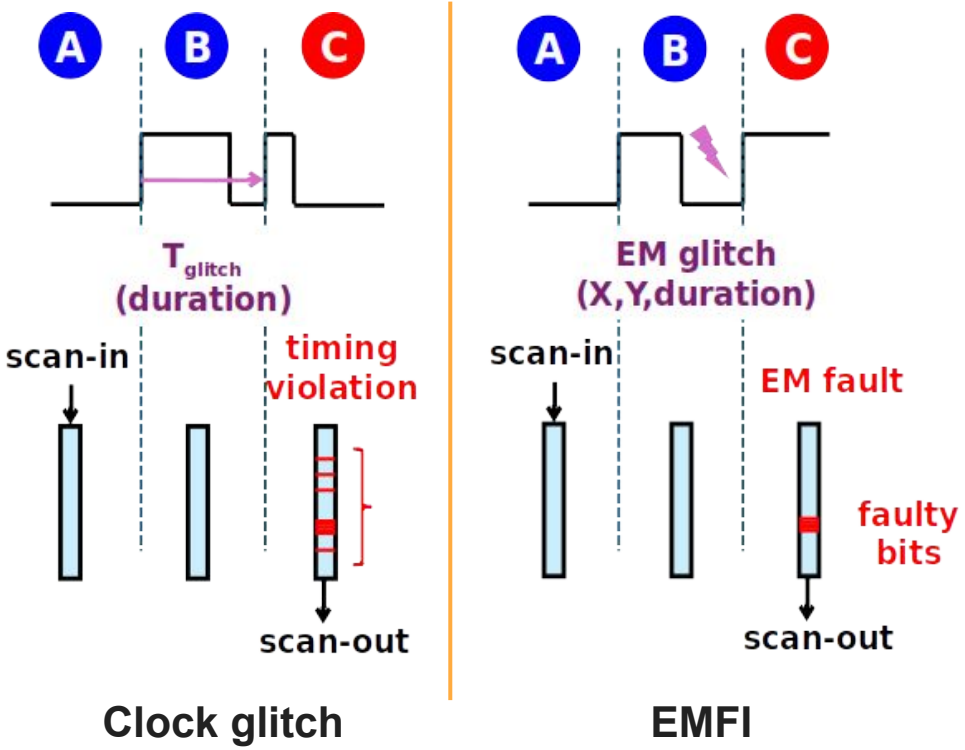
Root-Cause Analysis



1. **Inject Fault** at **T0** into the microarchitectural state.
2. **Fault** propagates upward through **µarch**, **arch**, and ISA layers.
3. **Capture State** periodically via scan-chain snapshots.
4. **Detect Divergence** when one lockstep core's state differs.
5. **Pinpoint Root Cause** as the earliest scan snapshot showing that divergence.

Experimental Methodology

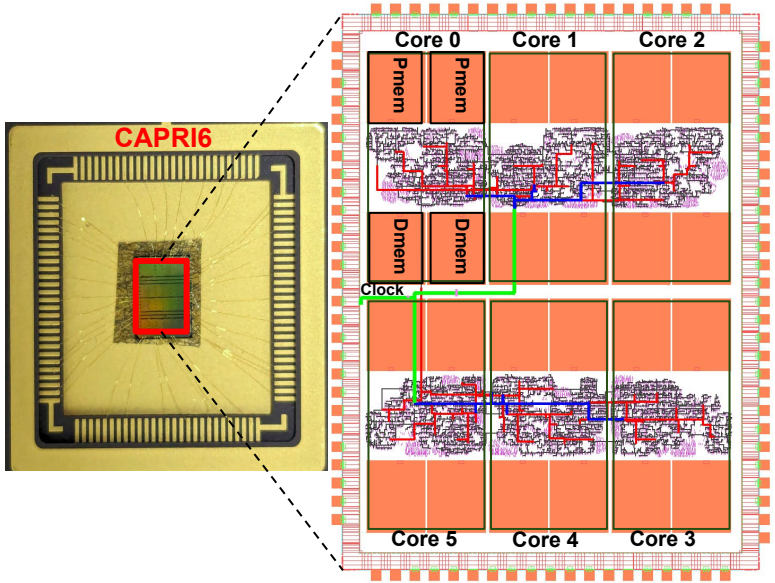
Single clock cycle fault injection

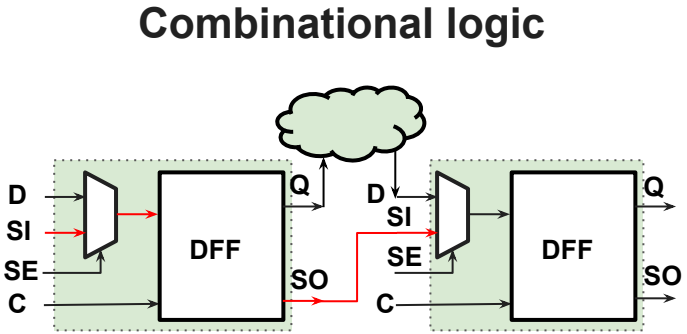


Clock Glitch : One-cycle clock pulse stretching to disturb execution.

EMFI : Directed EM pulse to induce a circuit fault non-invasively.

CAPRI6 : Multi-core drives fault localization





Scan chain: 10212 scan flip flops

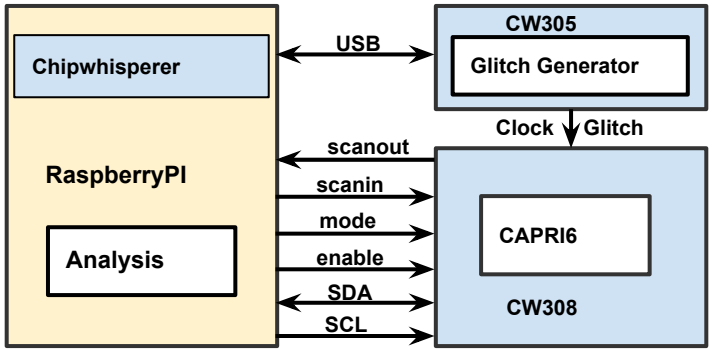
Design features

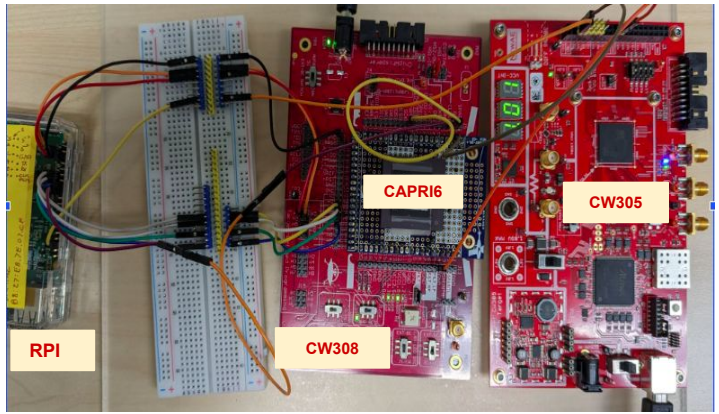
- Realized using TSMC 180nm
- Six OpenMSP430 core @ 40 Mhz
- 2KB IMEM, 2KB DMEM
- HW/SW Lock step
- P & R differs for each core

Instructions(used in crypto operations)

- MOV (turing machine complete)
- ADD
- MULT (peripheral module)

Clock glitch measurement setup





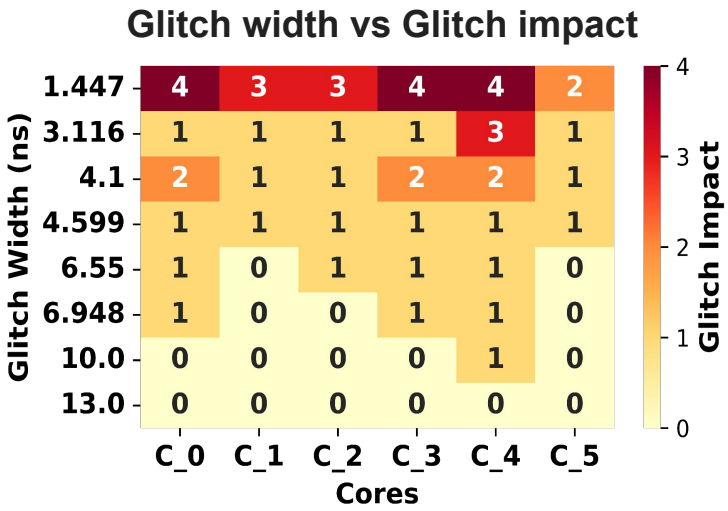
Device Under Test (DUT) : CAPRI6

Glitch Generator : ChipWhisperer → CW305

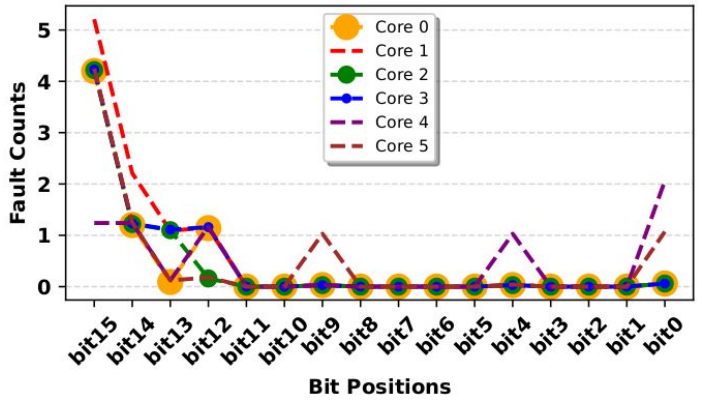
Controller/Analyzer : Raspberry Pi

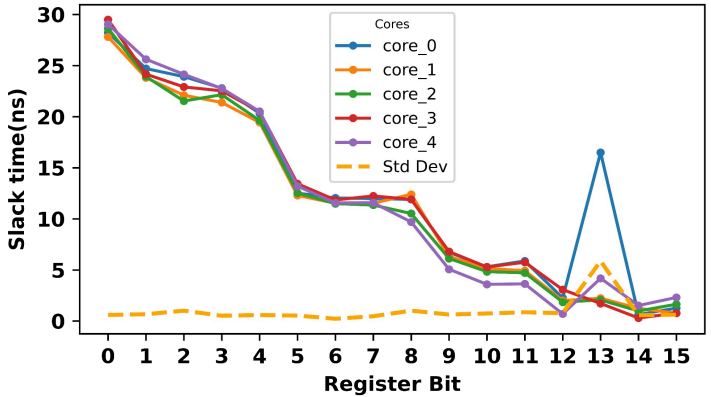
Glitch fault : mov instruction characterisation

MOV R6, R15



Faulty bit(s) of R15





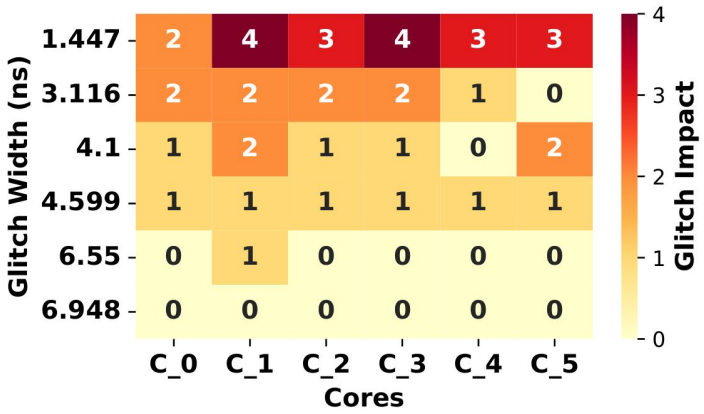
R15 : STA for validation

Root cause analysis

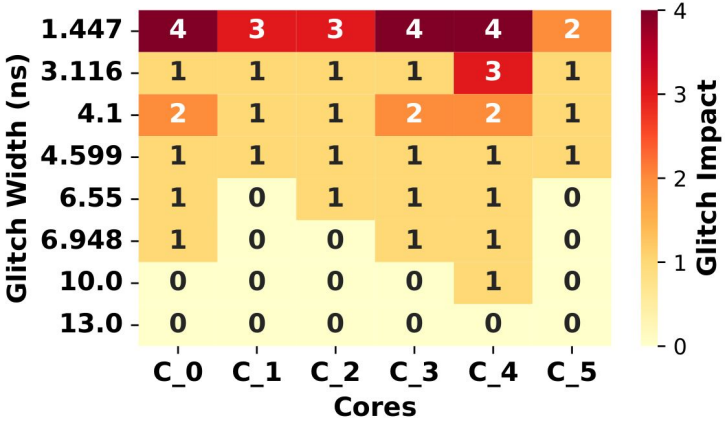
- Susceptible core: core 4
- Susceptible bit : bit 15
- STA validation : bit 15 has least slack time

Glitch fault : ADD, MULT instructions characterisation

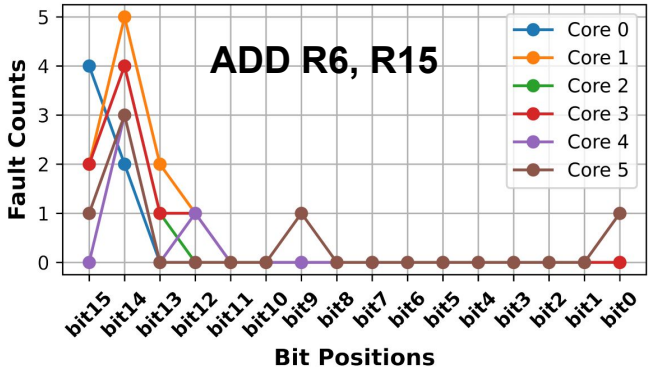
ADD core 1 is more sensitive



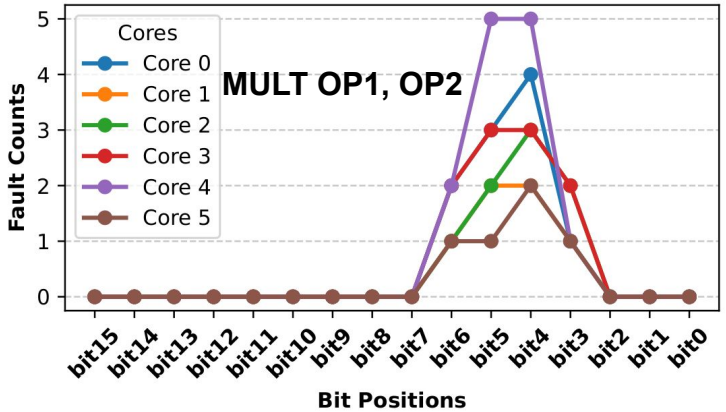
MULT core 4 is more sensitive



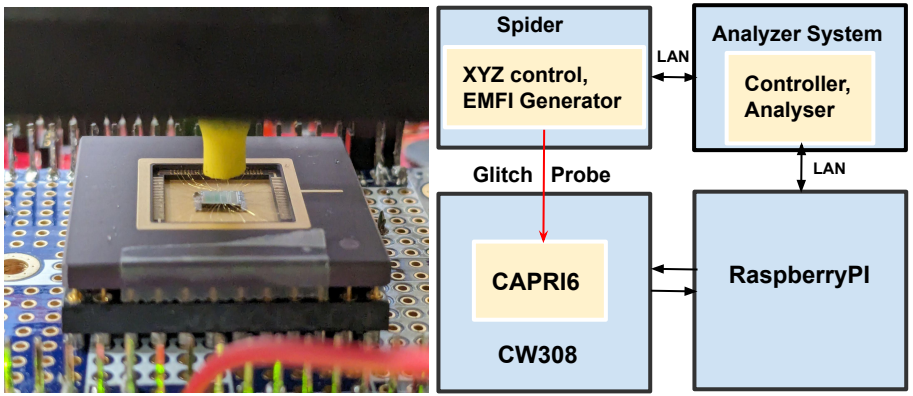
ADD core 1, bit 14 is more sensitive



MULT core 4, bit 4 is more sensitive



EMFI measurement setup

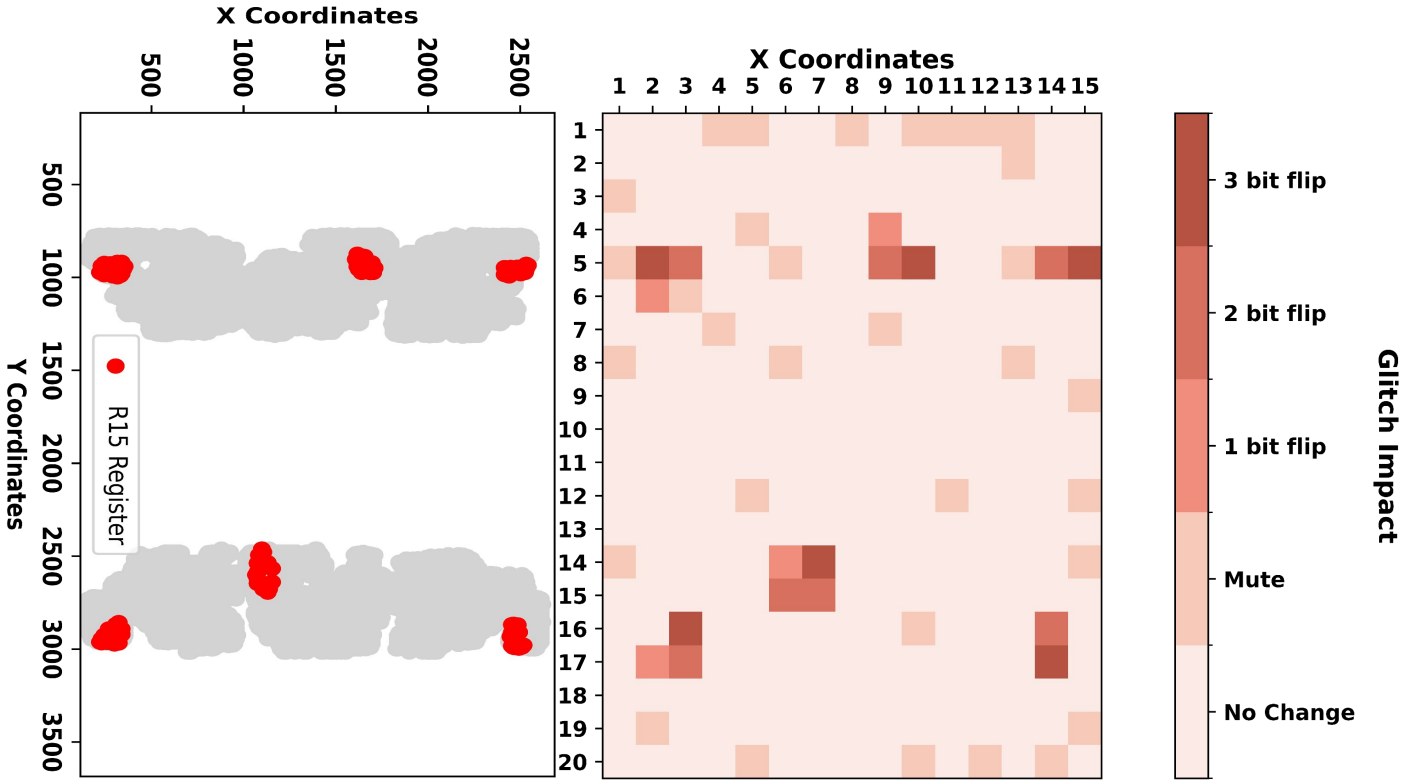


- EMFI identifies spatial (vulnerable) hotspots
- 4–25 ns pulses (±4 V) injected at each spot (15 * 20 spots)
- Scan-chain maps bit flips into a heatmap

EMFI : Mov instruction

MOV R6, R15

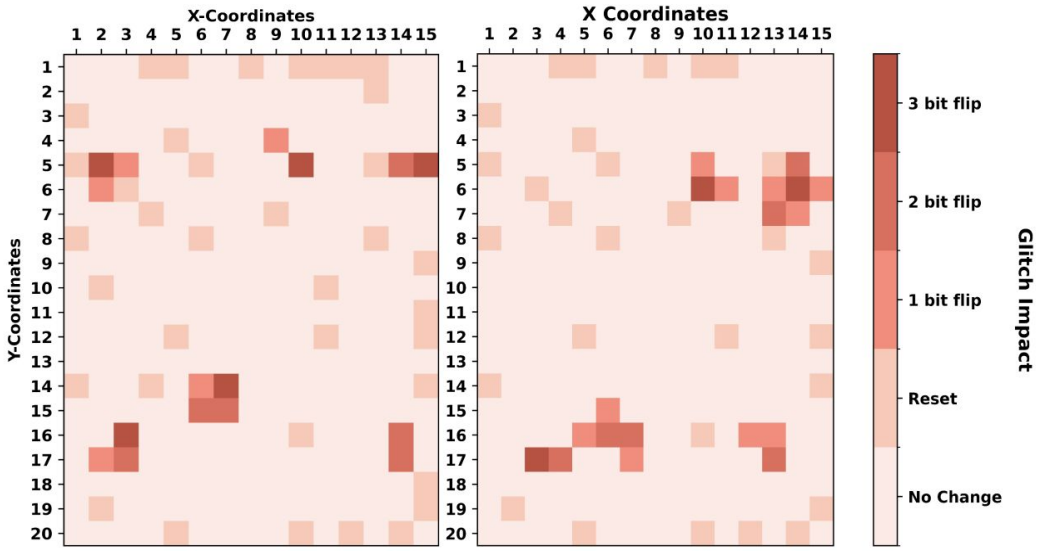
Core 4 more susceptible
R15 fault heatmap correlates with the chip's floorplan
Why is Core 4 vulnerable? Sparse metal increases EMFI exposure



Layout

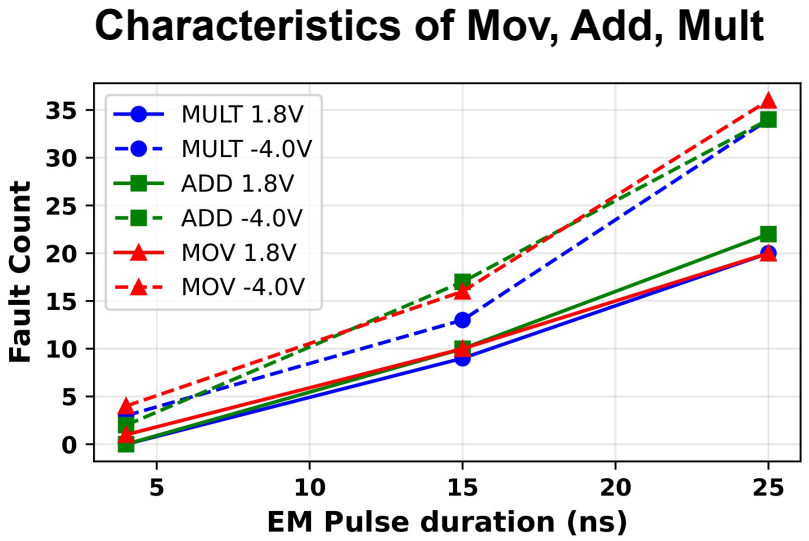
CAPRI6

EMFI : Instructions characterisation



ADD R6, R15 **MULT op1, op2**

Instruction	Faulty Core	Affected Bit(s)	Notes
ADD	Core 4	Bit 12	Hotspot aligns with layout exposure
MULT	Core 3	Bits 3–5	Multiple vulnerable locations



Longer EMFI pulses increase fault counts

−4.0 V pulses consistently induce more faults than +1.8 V pulses.

Summary

- **Bit-Level Localization:** Scan chains capture and pinpoint faults at individual flip-flop resolution.
- **Cross-Layer Diagnosis:** Six-core lockstep enables explainable tracing from hardware up to software.
  - **Glitch-STA Correlation:** Clock-glitch results are validated against static-timing analysis.
  - **EMFI-Layout Mapping:** EM pulse heatmaps overlay the layout to reveal metal-layer vulnerabilities.
- **Silicon-Cycle Validation:** Pre-silicon timing data and post-silicon scans jointly confirm each root cause.

References

- FaultDetective : Explainable to a Fault, from the Design Layout to the Software : TCHES 2024
- “Weird machines, exploitability, and provable unexploitability” Thomas Dullien
- “The Forgotten Threat of Voltage Glitching: A Case Study on Nvidia Tegra X2 SoCs” : FDTC
- “Glitched on Earth by Humans: A Black-Box Security Evaluation of the SpaceX Starlink User Terminal” : Black hat 2022
- 1 in 1000 chips produce Silent Data Corruptions: https://www.sigarch.org/sdcs-a-b-c/

Any Questions?

Thank you
dshanmugam@wpi.edu
zliu12@wpi.edu
pschaumont@wpi.ed

Fault impacts

- **“Glitched on Earth by Humans: A Black-Box Security Evaluation of the SpaceX Starlink User Terminal” : Black hat 2022**
- **“The Forgotten Threat of Voltage Glitching: A Case Study on Nvidia Tegra X2 SoCs” : FDTC**
- **1 in 1000 chips produce Silent Data Corruptions: https://www.sigarch.org/sdcs-a-b-c/**