Comparing Voltage and Clock Glitch Attacks on a RISC-V Implementation on FPGA

Roua Boulifa, Giorgio Di Natale, Paolo Maistri
Univ. Grenoble Alpes, CNRS, Grenoble INP, TIMA, 38000 Grenoble, France
`firstname.lastname@univ-grenoble-alpes.fr`

Abstract

Embedded systems, vulnerable to fault injection attacks, require robust security measures. This work analyzes voltage and clock glitch attacks on a RISC-V processor. We present a comprehensive analysis of the glitch setup and demonstrate that some fault models apply to both injection methods. These models help clarify the impact of glitches on system behavior, aiding vulnerability assessment and the development of cost-effective countermeasures.

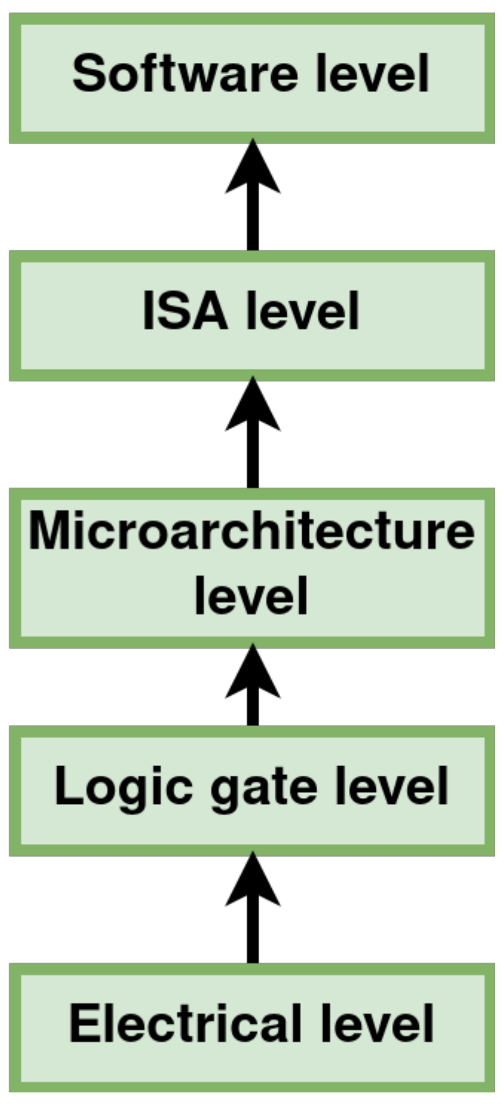
Motivation

The growing complexity of processors makes it challenging to characterize fault effects, limiting our understanding of the faults:

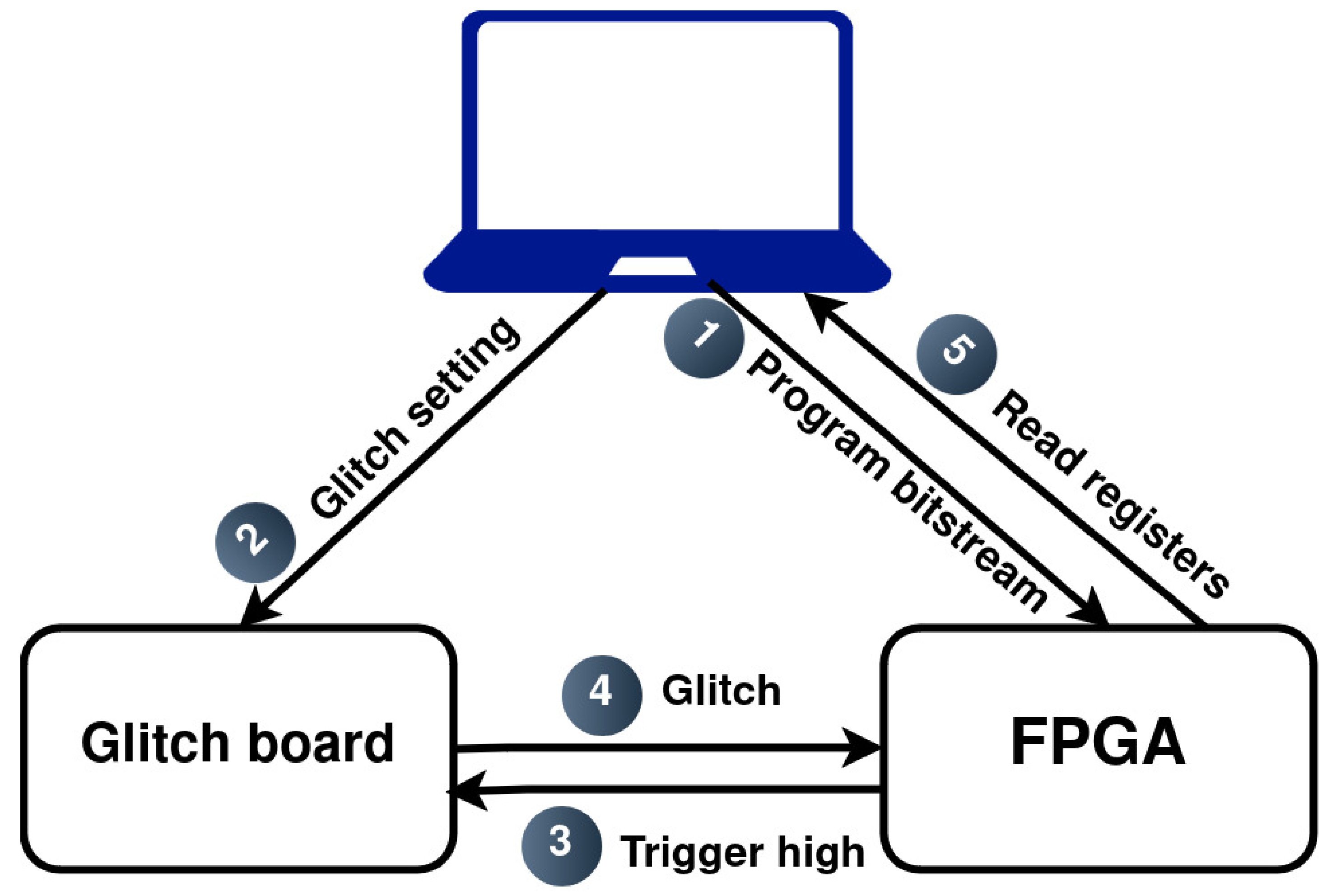
- This can lead to incomplete fault models, resulting in a lack of understanding of what is happening within the microarchitecture, and an inability to develop proper countermeasures:
  - Over-engineered countermeasures, leading to unnecessary costs and potential performance degradation
  - Under-engineered countermeasures, leaving security vulnerabilities unaddressed
- To design effective countermeasures, it is crucial to understand the fault models which are built upon experimental observations

Fault modeling

- Safeguarding digital systems against fault attacks involves a detailed analysis of injection effects to create realistic fault models
- Fault models can be defined at different abstraction levels, each with trade-offs:
  - Higher abstraction levels: less accurate but easier and faster to simulate
  - Lower abstraction levels: more realistic but require greater simulation effort


Software level
ISA level
Microarchitecture level
Logic gate level
Electrical level

Methodology

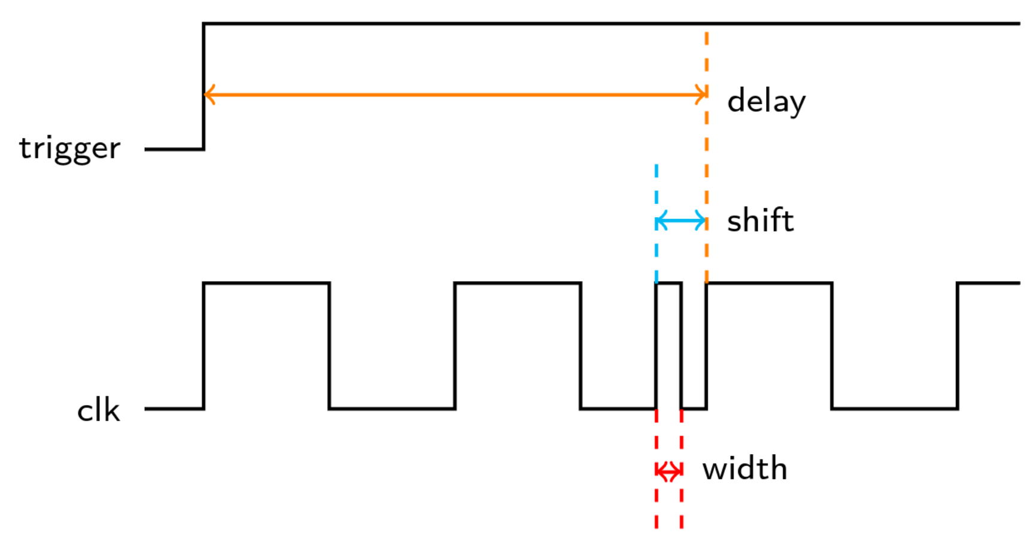


Step 1: The FPGA is programmed with the bitstream
Step 2: The host machine configures the glitch parameters
Step 3: The target CPU execute the target program, which includes a synchronization trigger
Step 4: The glitch is injected at the given delay after the trigger
Step 5: The content of the registers are sent back to the control PC

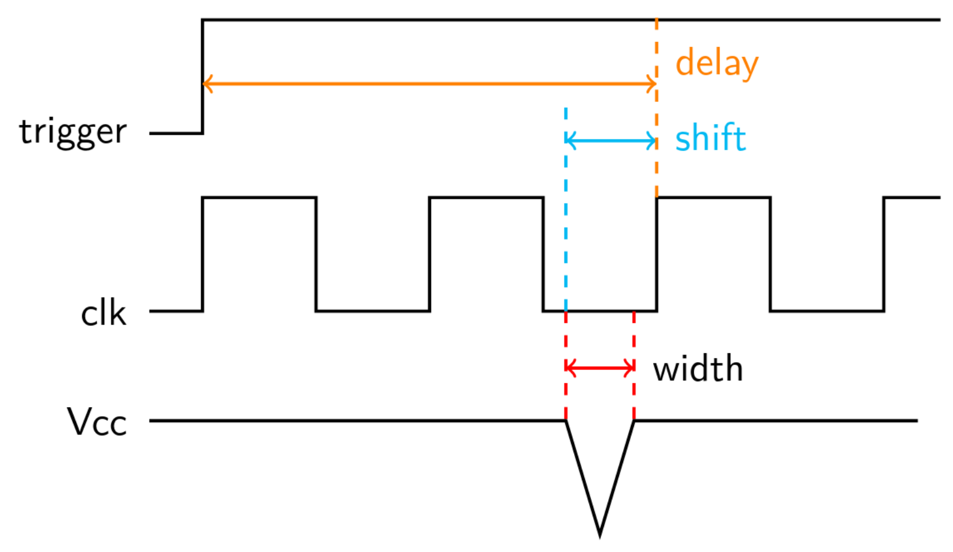
Experimental setup

- Target device: CV32E40P, a 32 bit RISC-V processor with 4 stage pipeline [1] (fetch, decode, execute, and write back)
- Chipwhisperer environment [2] has been used to perform the fault injection campaigns

Clock glitch



Voltage glitch



Results and analysis

Fault model	clock glitch	voltage glitch
Skip	16.94 %	4.3 %
Early Result Capture [3]	48.62 %	33.33 %
Complex	34.44 %	62.37 %

class	clock glitch	voltage glitch
Silent	77.4%	10.5 %
Crash	0.01 %	0.17%
Fault	22.6 %	89.5 %

- We observed instruction skips: the destination register kept its initial value.
- Some faults caused Early Result Capture [3]: an unintended propagation of an instruction's result into the following instruction is caused by a timing overlap introduced by the fault. The fault affects the synchronization of the second instruction within the pipeline, leading it to erroneously read the result of the first due to delayed signals.
- Voltage glitches often led to complex behaviors involving multiple fault effects.

Acknowledgements

This work is supported by the ARSENE project, funded by the France 2030 government investment plan managed by the French National Research Agency (ANR-22-PECY-0004).

References

[1] OpenHW Group, CV32E40P User Manual, Version 1.2.1.
[2] NewAE Technology Inc. "ChipWhisperer Documentation". *https://chipwhisperer. readthedocs.io/en/latest/index.html*.
[3] R. Boulifa, G. Di Natale, P. Maistri. Early Result Capture: Racing Conditions in Pipeline due to Clock Glitches. 30th IEEE European Test Symposium (ETS 2025).