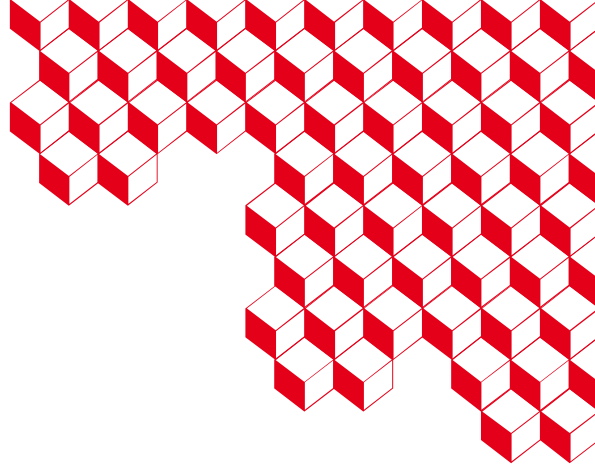


FDTC 2024
Fault Diagnosis and Tolerance in Cryptography

PoP DRAM: A new EMFI approach based on EM-induced glitches on SoC

Clément Fanjas, Driss Aboulkassimi, Simon Pontié, Jessy Clédière

Summary

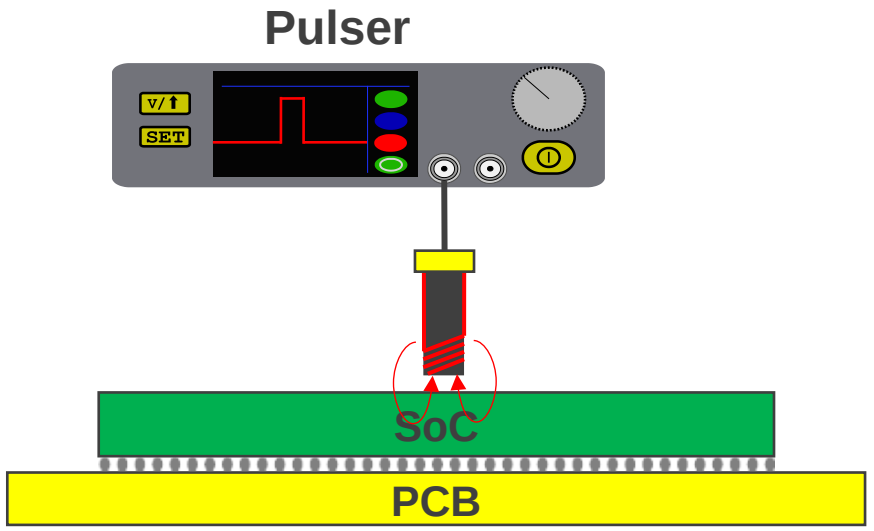
Context : Electromagnetic Fault Injection (EMFI)

How do we inject fault using EMFI:

- Injecting a voltage pulse into an active probe located above the targeted chip.

- Depending on the probe position over the chip, an EM coupling is created between the target and the probe.

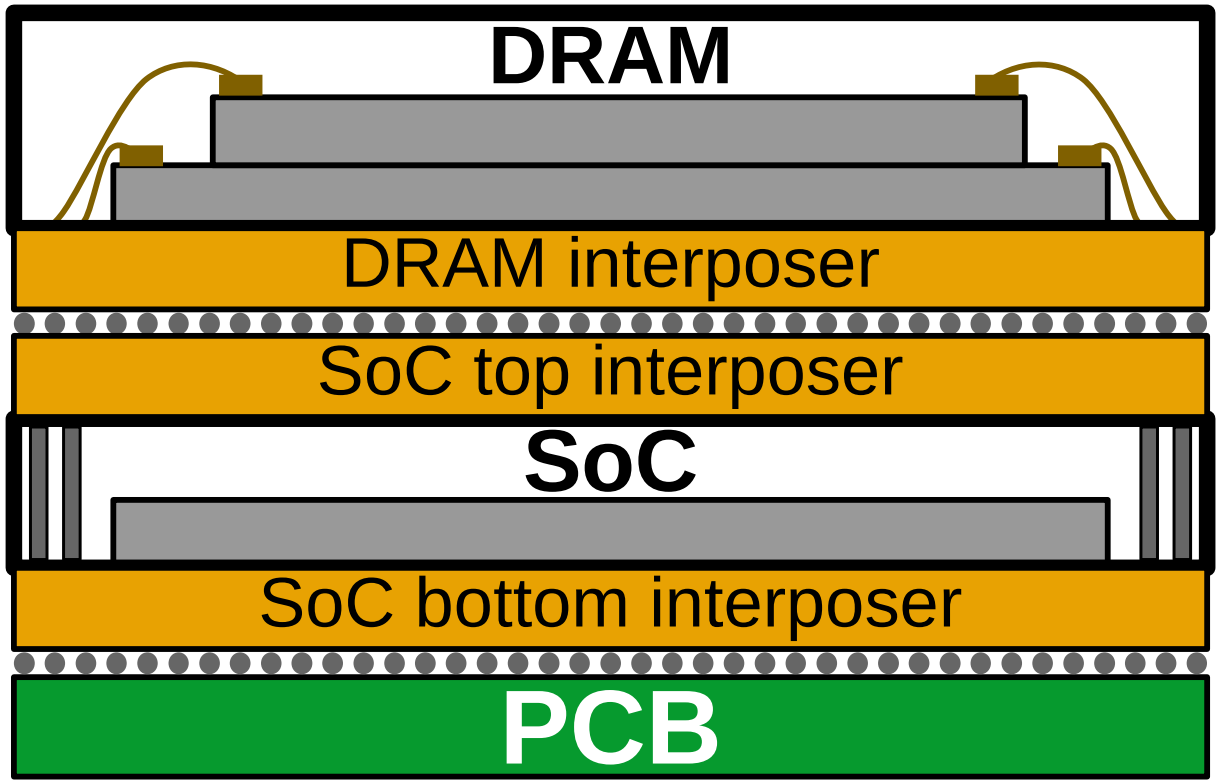
- This coupling induces a transient current inside the chip which can corrupt the normal operation.



Context : Package-on-Package SoC

In modern SoC, an other chip (often a DRAM) is stacked **above** the SoC.

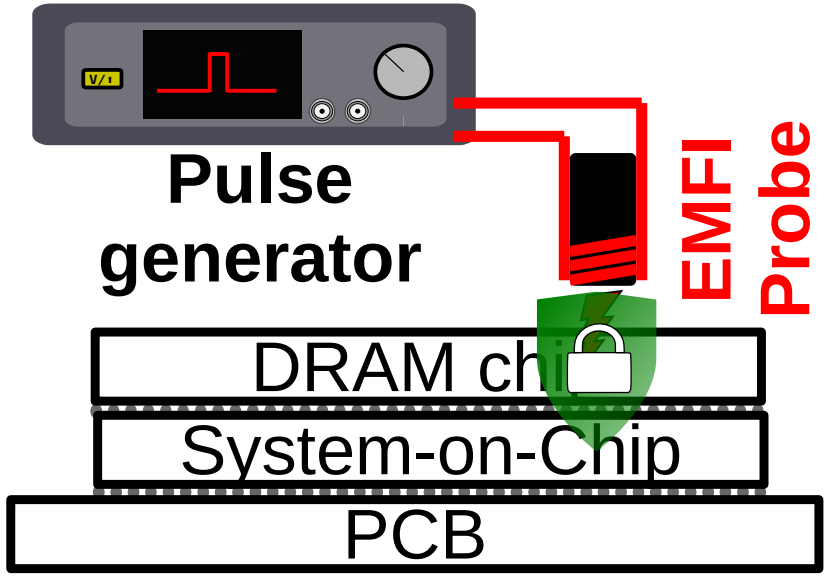
What is the impact of PoP on Fault Injection ?



Related works : Package-on-Package SoC

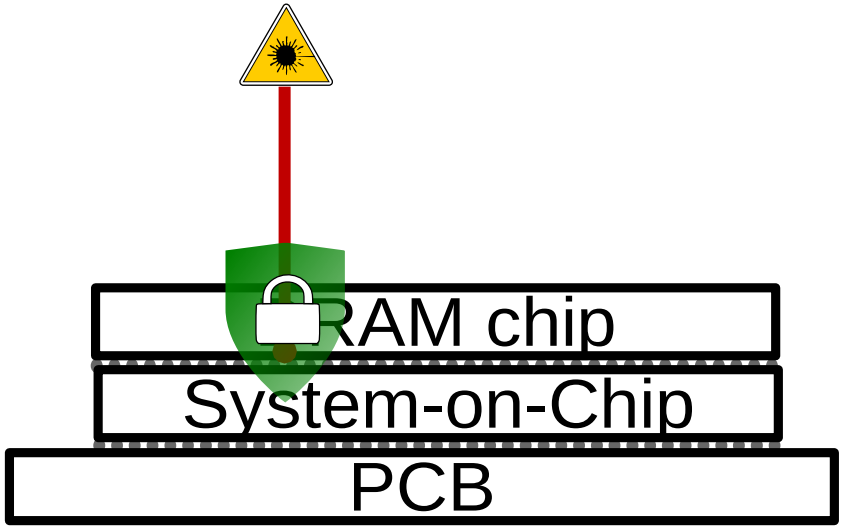
It reduces the efficiency of local Fault Injection method such as EMFI or Optical Fault Injection :

Nourdin Aït El Mehdi [2019]
Analyzing the Resilience of Modern Smartphones Against Fault Injection Attacks



Fault model identified but not relevant in early boot attack scenarios.

Vasselle et al. [2017]
Laser-Induced Fault Injection on Smartphone Bypassing the Secure Boot

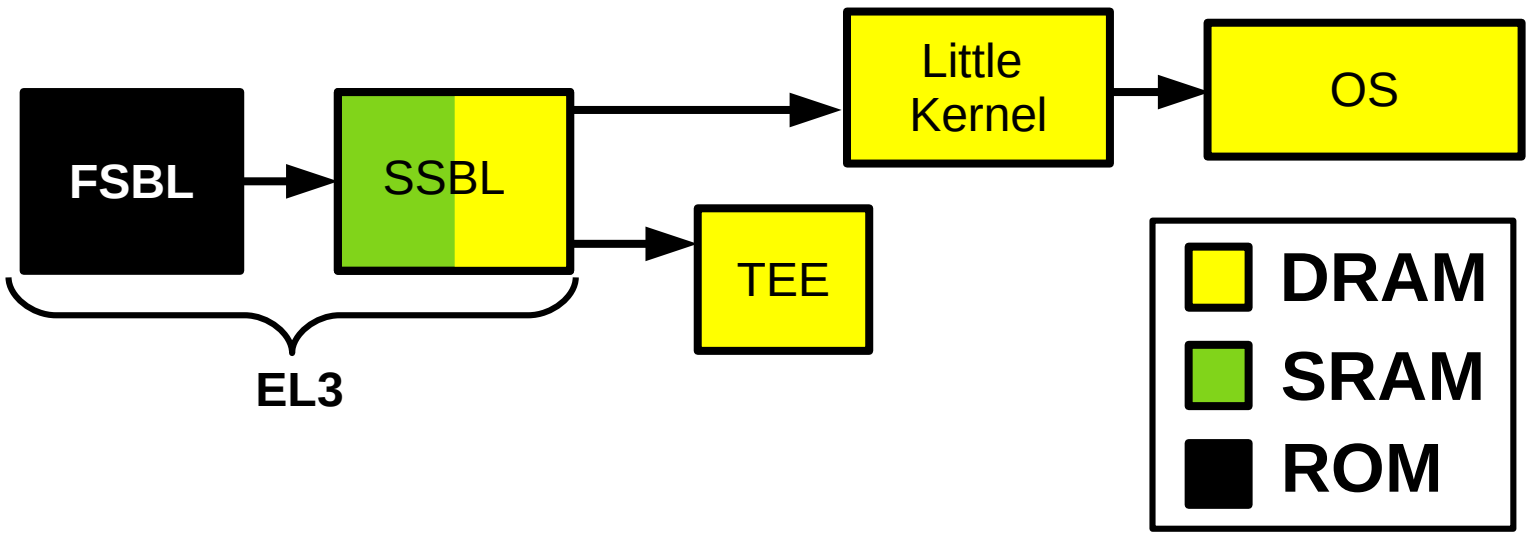


Only early boot attack scenarios available, but needs aggressive modification of the target.

Target

Target: Smartphone System-on-Chip on dev-board:

- 4 custom ARM (aarch64) cores
  ⇒ 2 cores up to 2.15GHz
  ⇒ 2 cores up to 1.59GHz
- Secure-Boot disabled
- Emergency program running in SRAM at the SSBL level



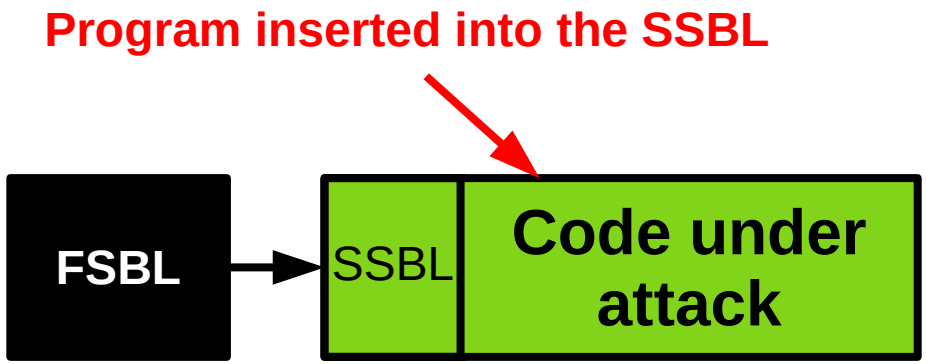
Objective:
Fault the target despite the PoP DRAM.

Target

Target: Smartphone System-on-Chip on dev-board:

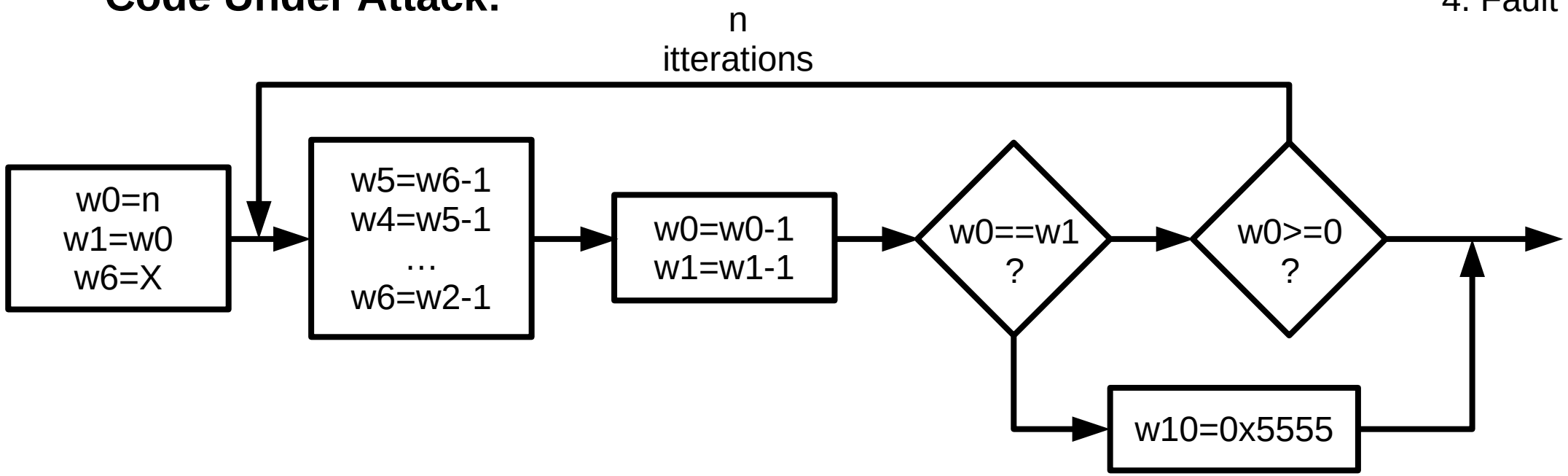
- 4 custom ARM (aarch64) cores
  ⇒ 2 cores up to 2.15GHz
  ⇒ 2 cores up to 1.59GHz
- Secure-Boot disabled
- Emergency program running in SRAM at the SSBL level

Objective:
Fault the target despite the PoP DRAM.



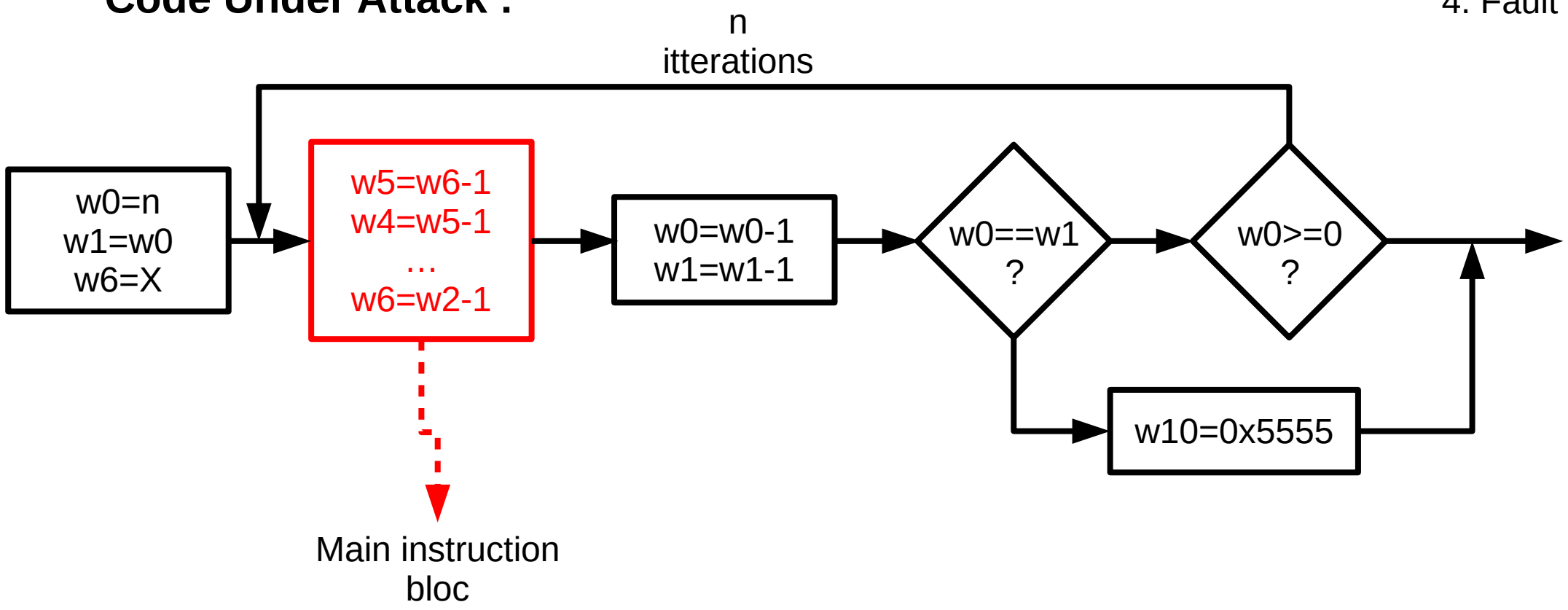
Fault characterization

Code Under Attack:



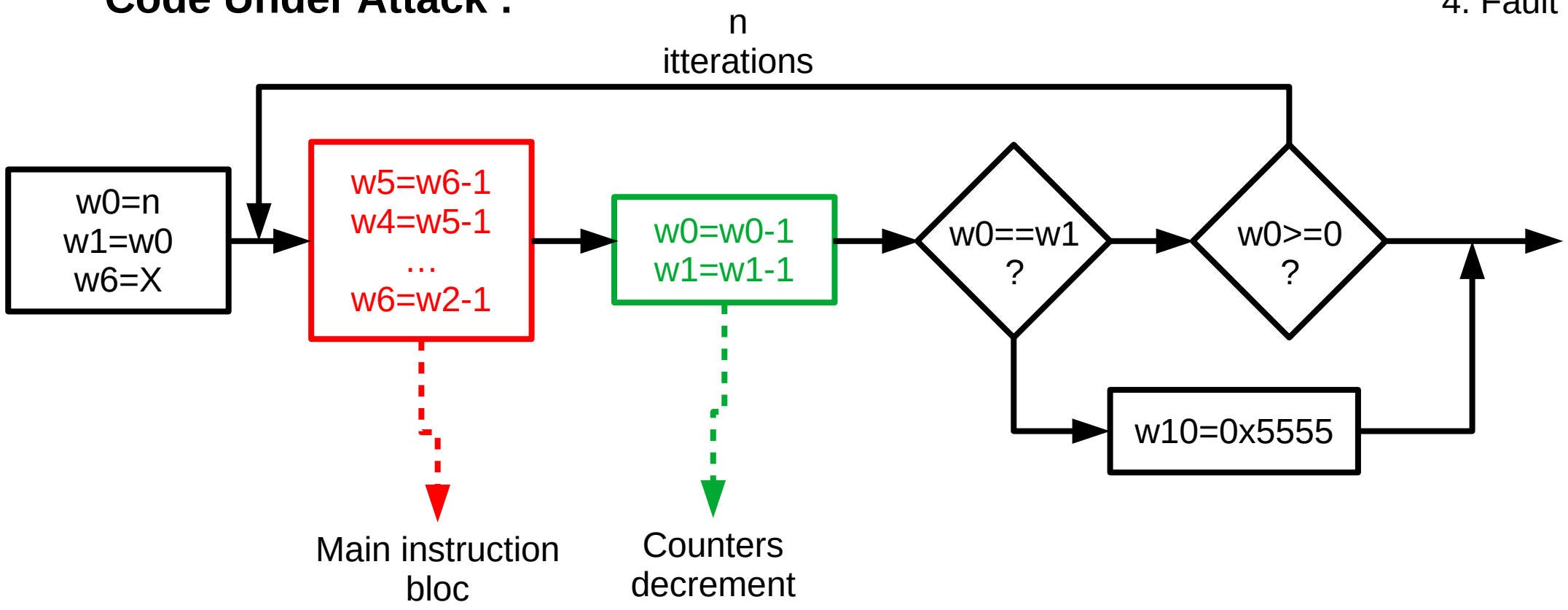
Fault characterization

Code Under Attack :



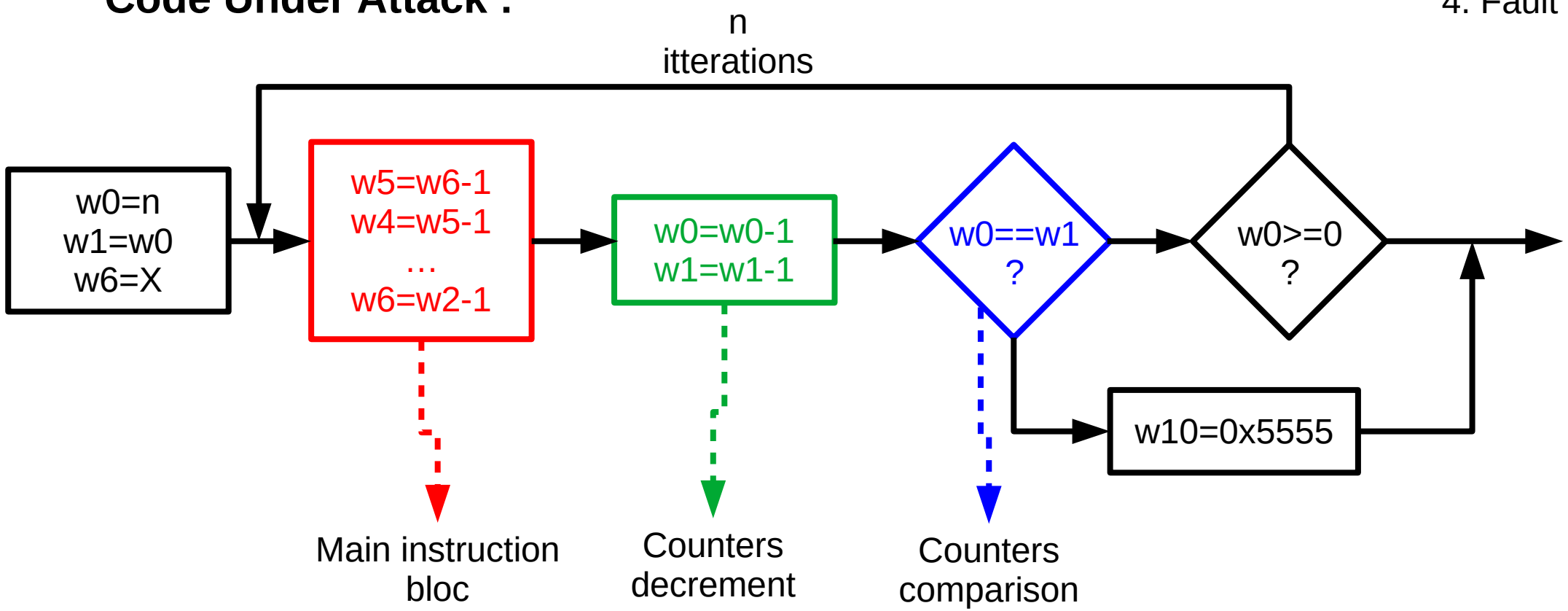
Fault characterization

Code Under Attack :



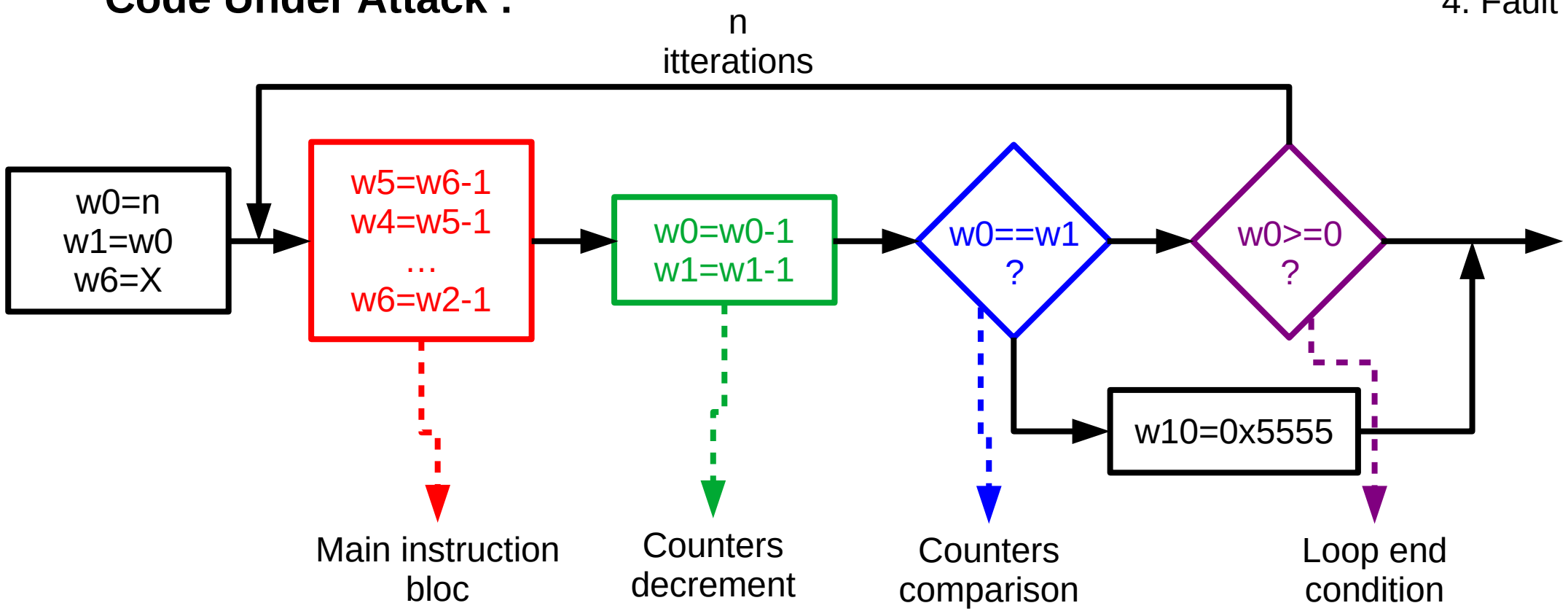
Fault characterization

Code Under Attack :



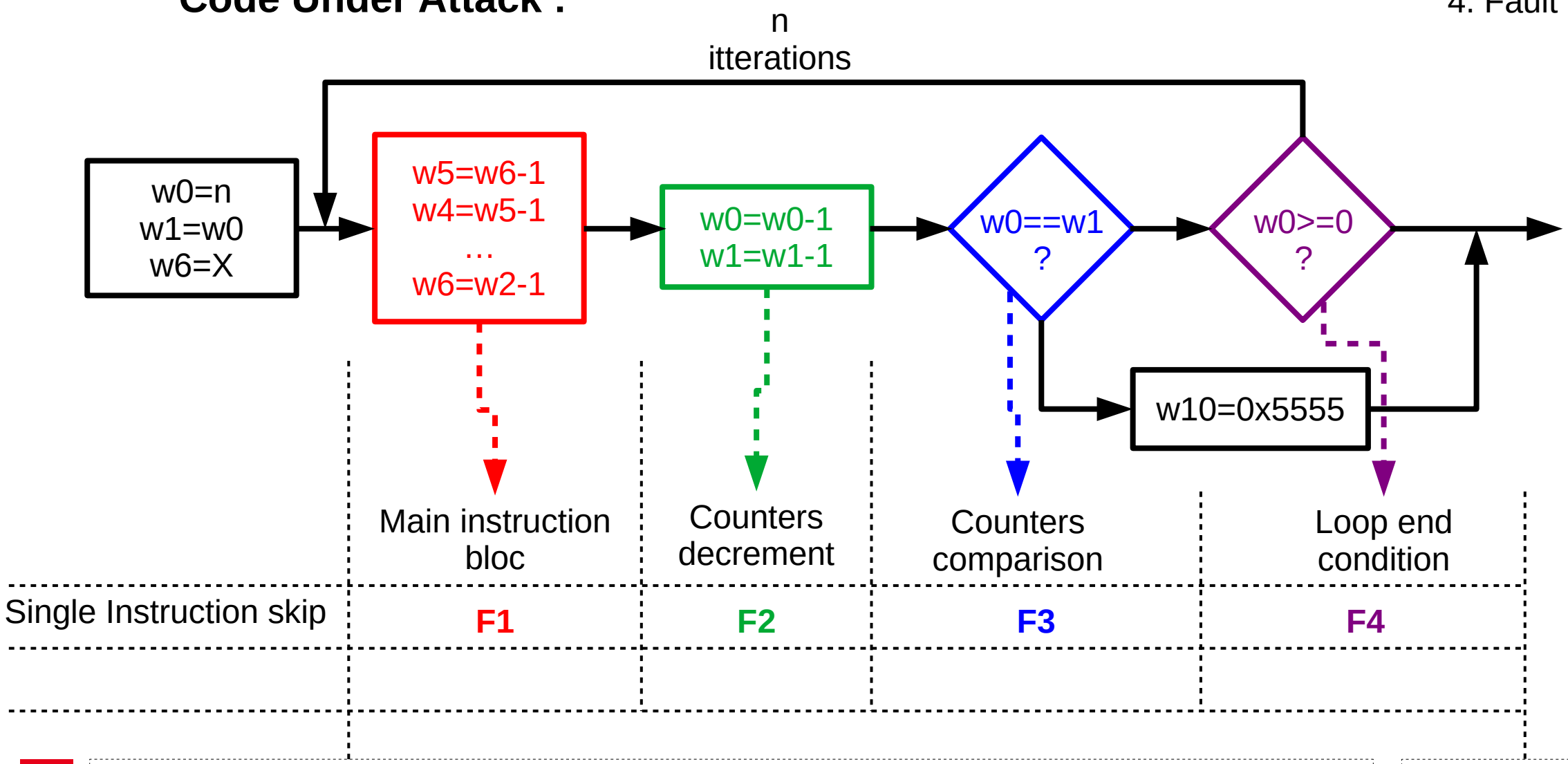
Fault characterization

Code Under Attack :



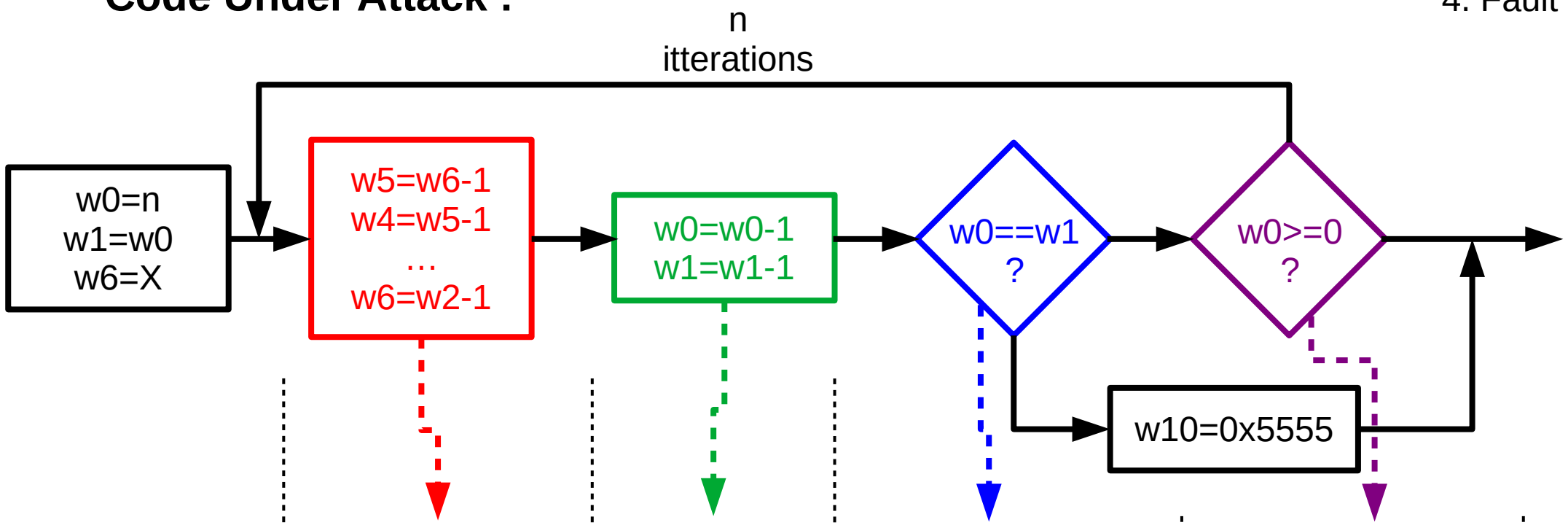
Fault characterization

Code Under Attack :



Fault characterization

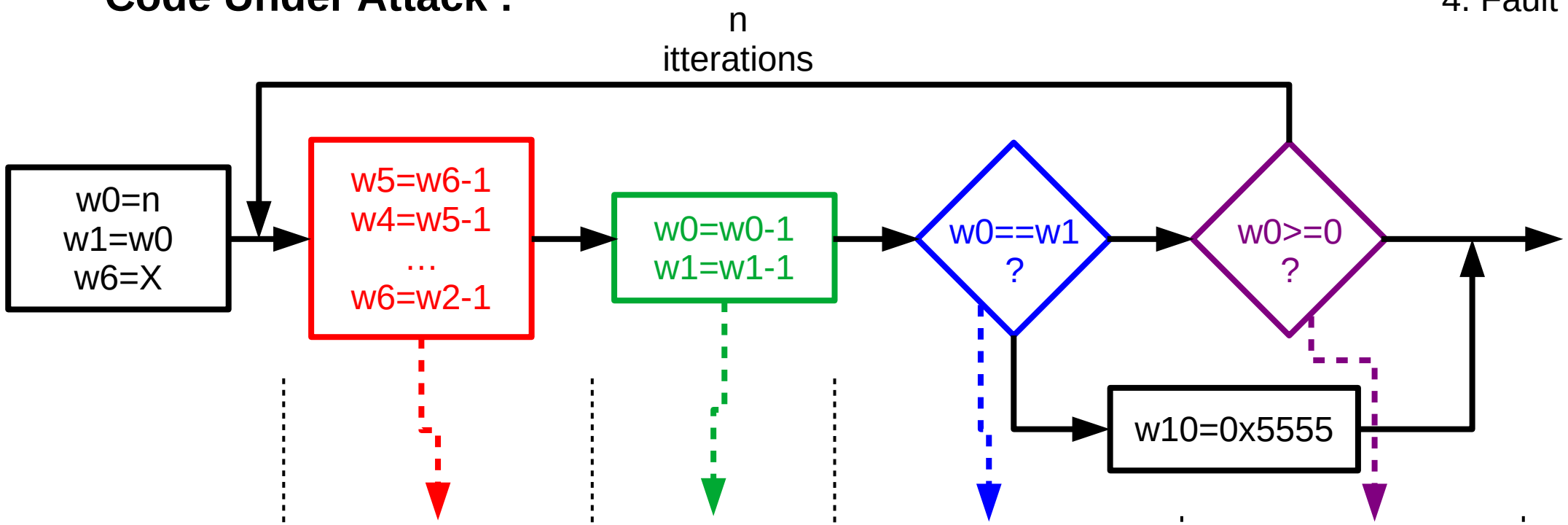
Code Under Attack :



	Main instruction bloc	Counters decrement	Counters comparison	Loop end condition
Single Instruction skip	**F1**	**F2**	**F3**	**F4**
More complex models	**F5**	**F6**	**F7**	**F8**

Fault characterization

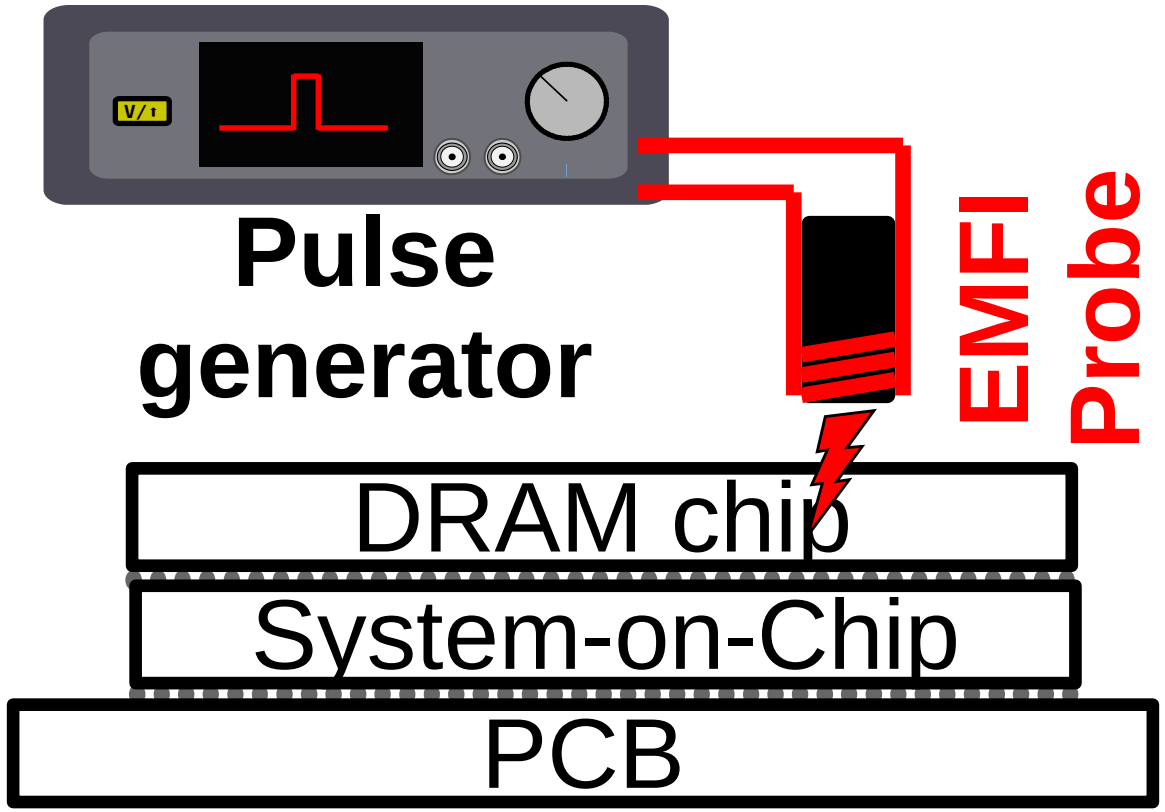
Code Under Attack :



	Main instruction bloc	Counters decrement	Counters comparison	Loop end condition
Single Instruction skip	**F1**	**F2**	**F3**	**F4**
More complex models	**F5**	**F6**	**F7**	**F8**
Others behaviours	**F9 / Crash (mute) / No Fault**			

Fault Injection methods

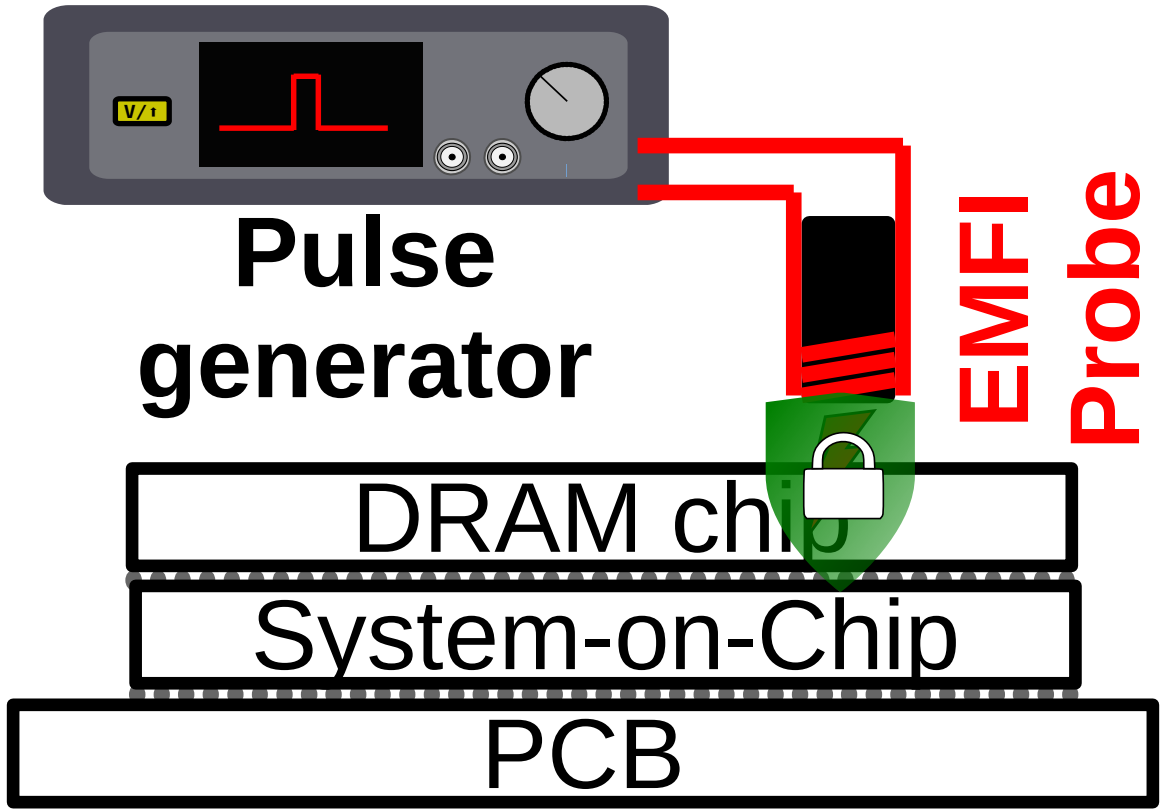
Method 1 : Conventional EMFI above the SoC with the PoP DRAM.
Reducing the target supply voltage might help injecting faults.



Fault Injection methods

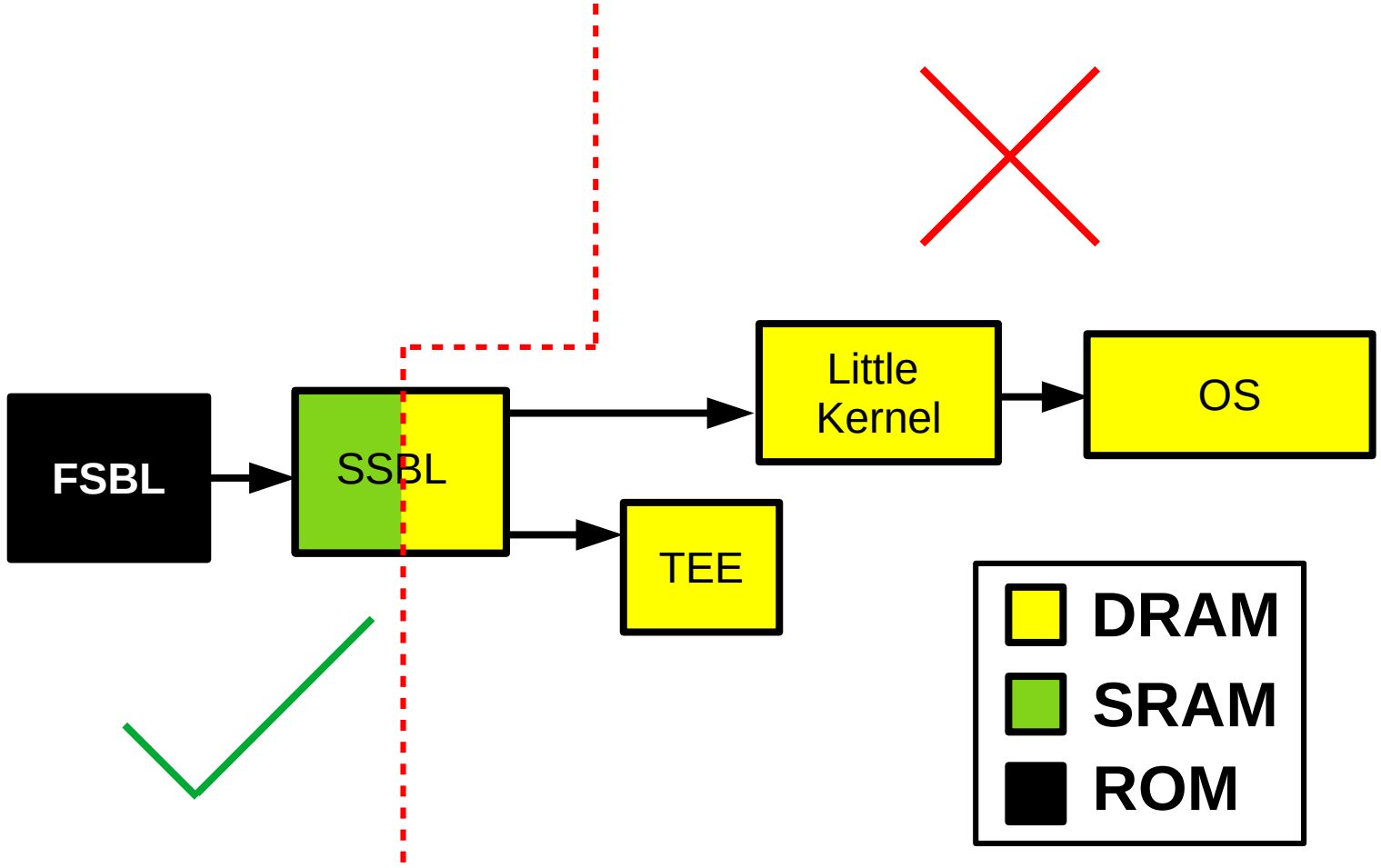
Method 1 : Conventional EMFI above the SoC with the PoP DRAM.

Results of this approach : No Faults



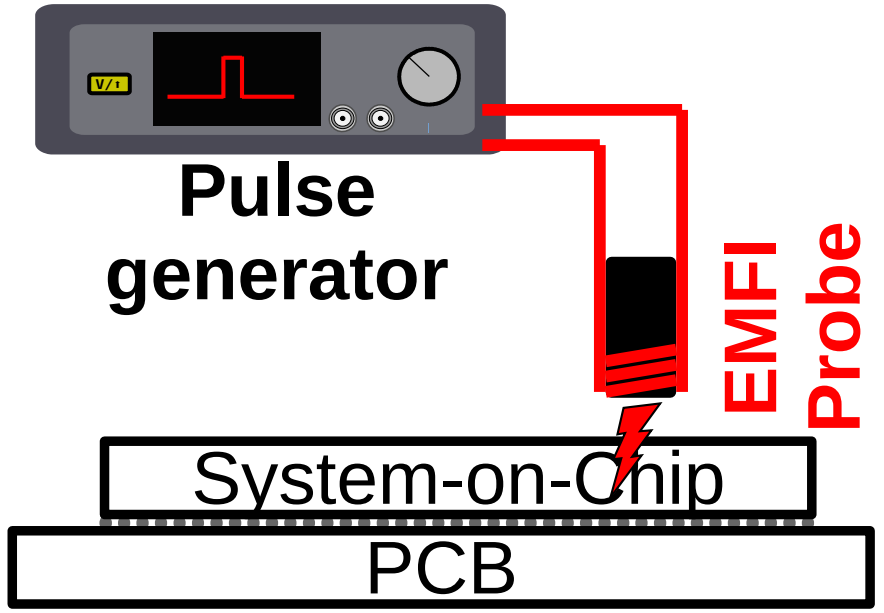
Fault Injection methods

<u>Method 2 : Conventional EMFI above the SoC without the PoP DRAM.</u>
Vasselle et al. (2017) show that the first stages of the boot run correctly even if the DRAM has been removed.



Fault Injection methods

<u>Method 2: Conventional EMFI above the SoC without the PoP DRAM.</u>
Vasselle et al. (2017) show that the first stages of the boot run correctly even if the DRAM has been removed.



Fault Injection methods

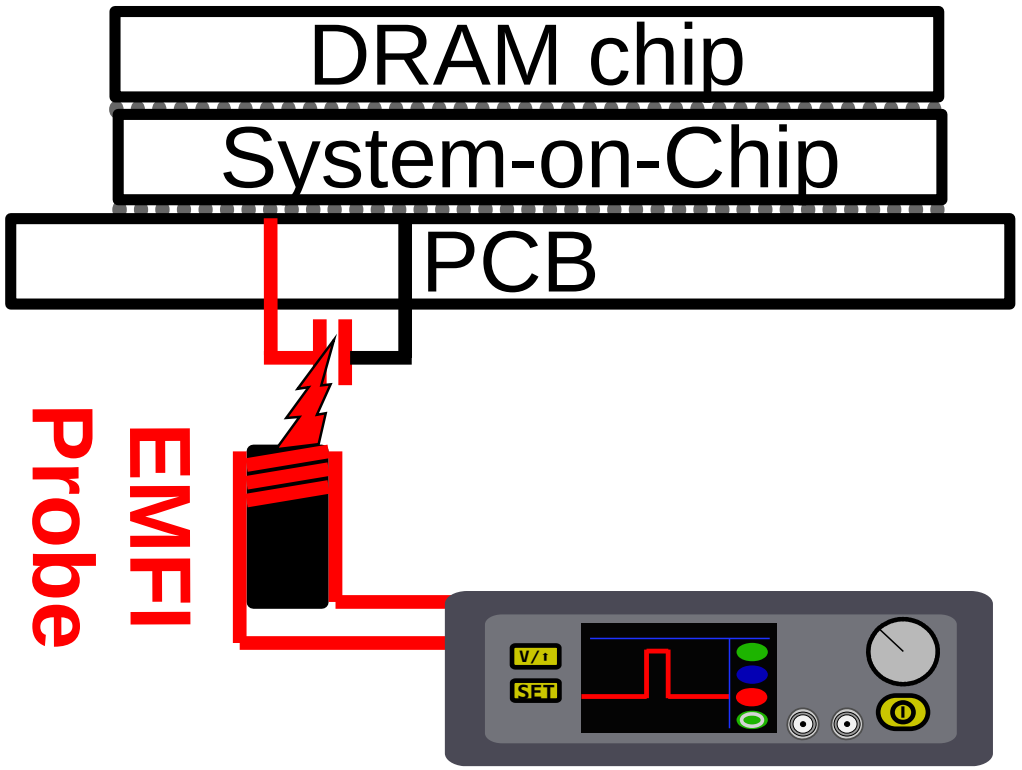
Method 2 : Conventional EMFI above the SoC without the PoP DRAM.

XY Fault Map over the SoC imaging
(100um step, fixed Z axis)

Fault Injection methods

Method 3 : EM-induced glitches :

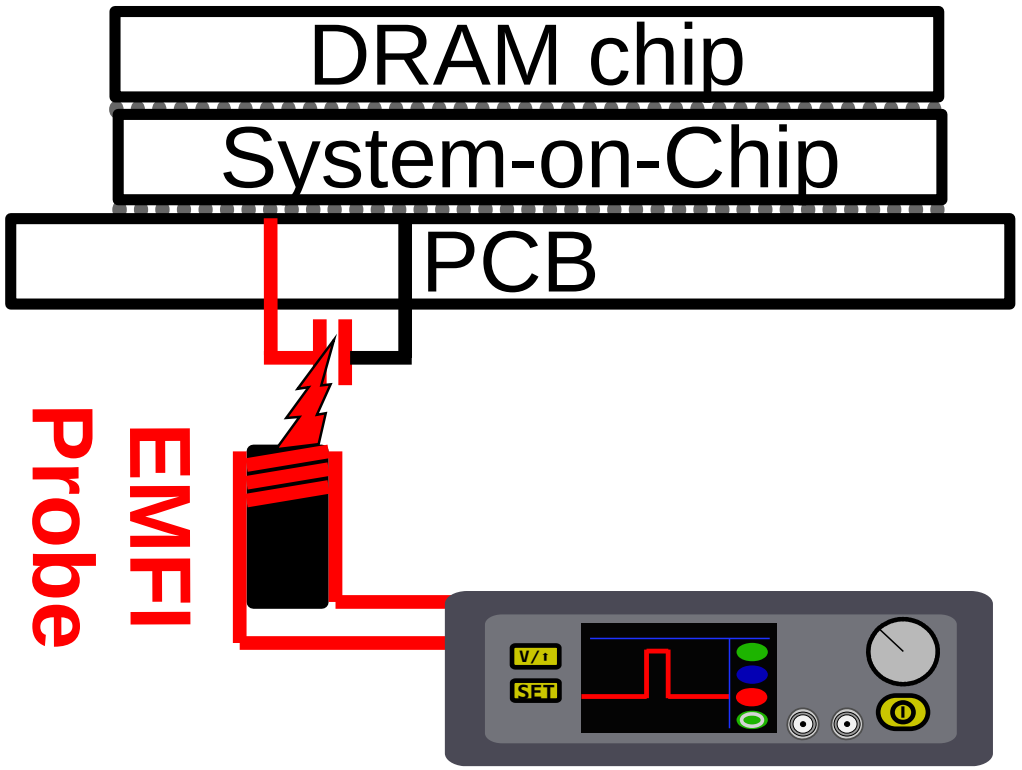
The idea is to induces voltage glitches on the target supply voltage line by using EM pulses.



Fault Injection methods

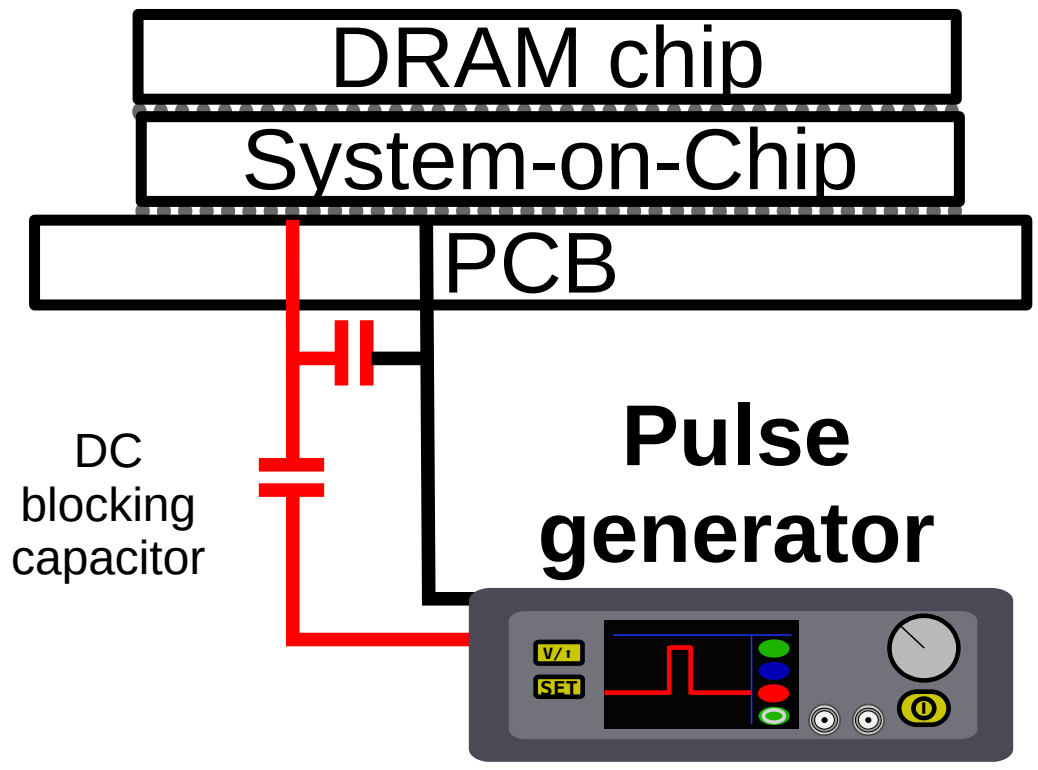
Method 3 : EM-induced glitches :

The idea is to induces voltage glitches on the target supply voltage line by using EM pulses.



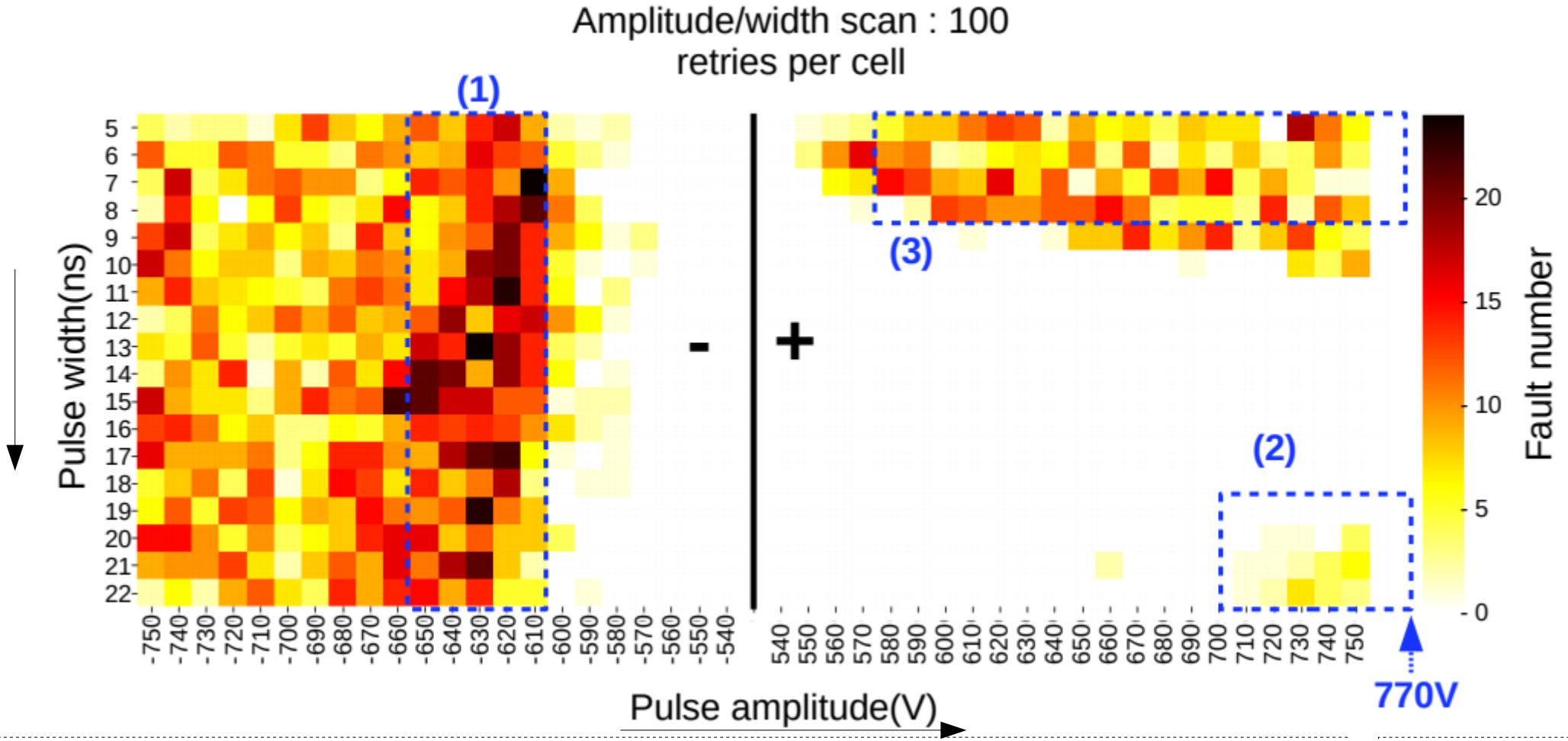
Method 4 : Conventional voltage glitches :

A pulse is injected through a 100nF capacitor soldered on the supply voltage line.



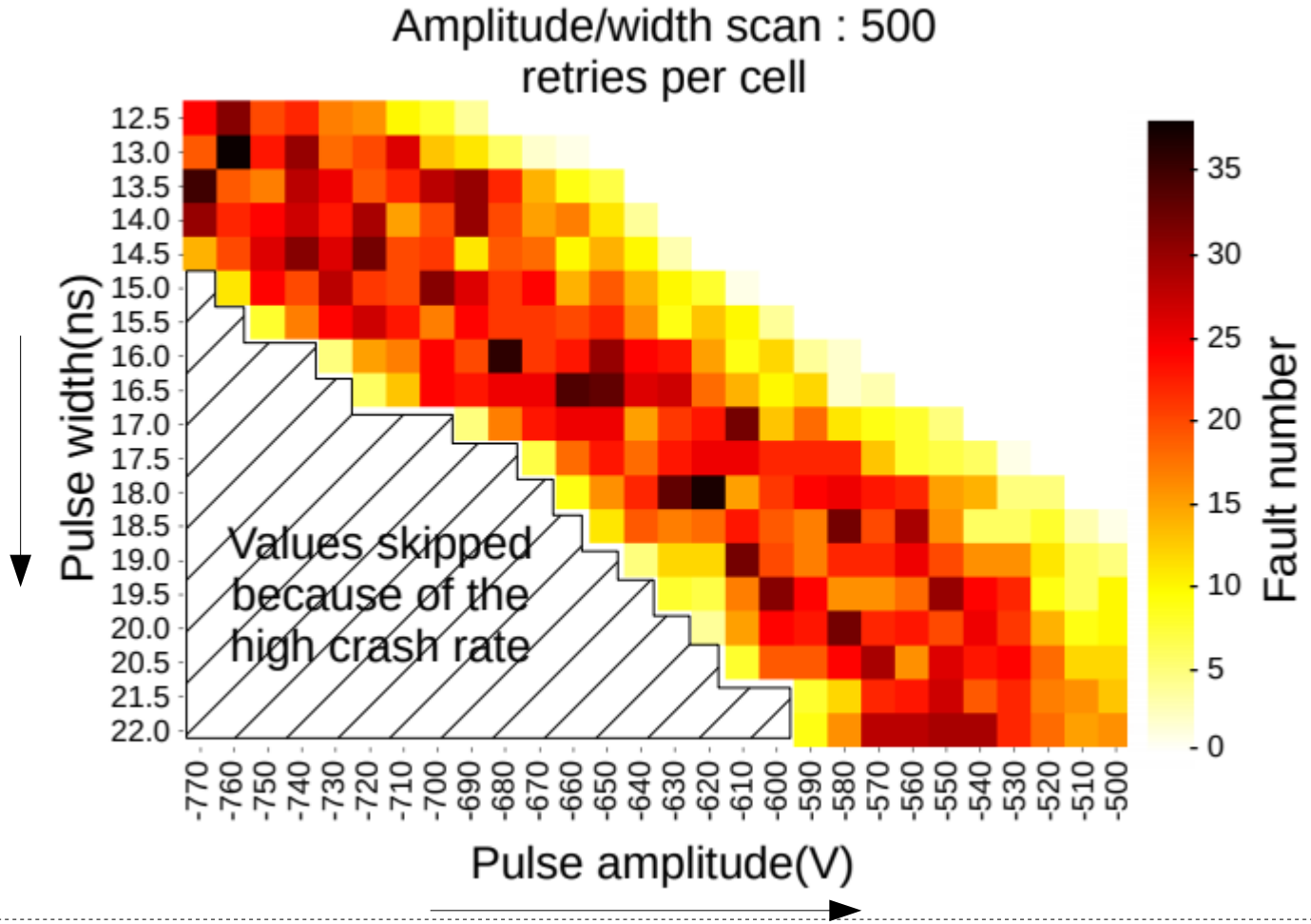
Fault obtained

Method 3 : EM-induced glitches :

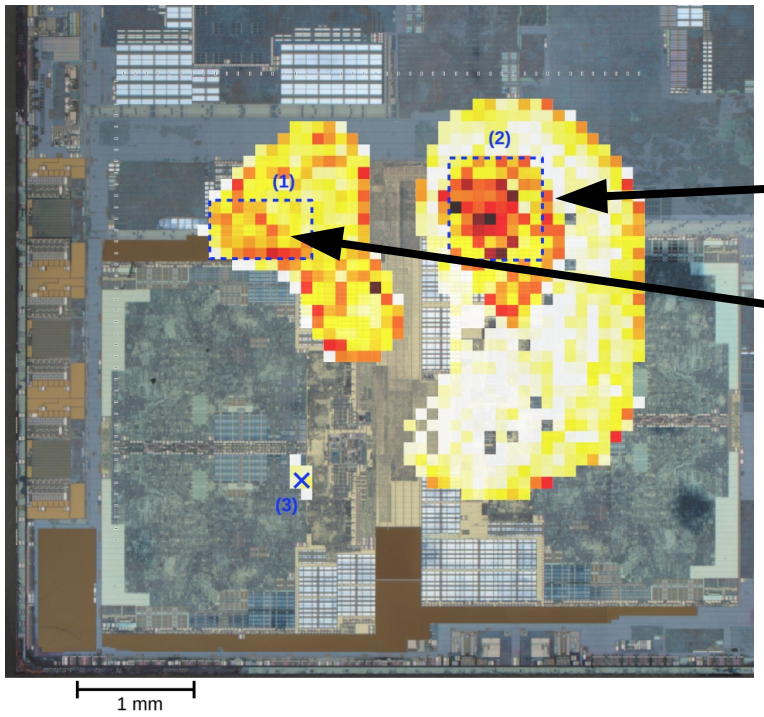


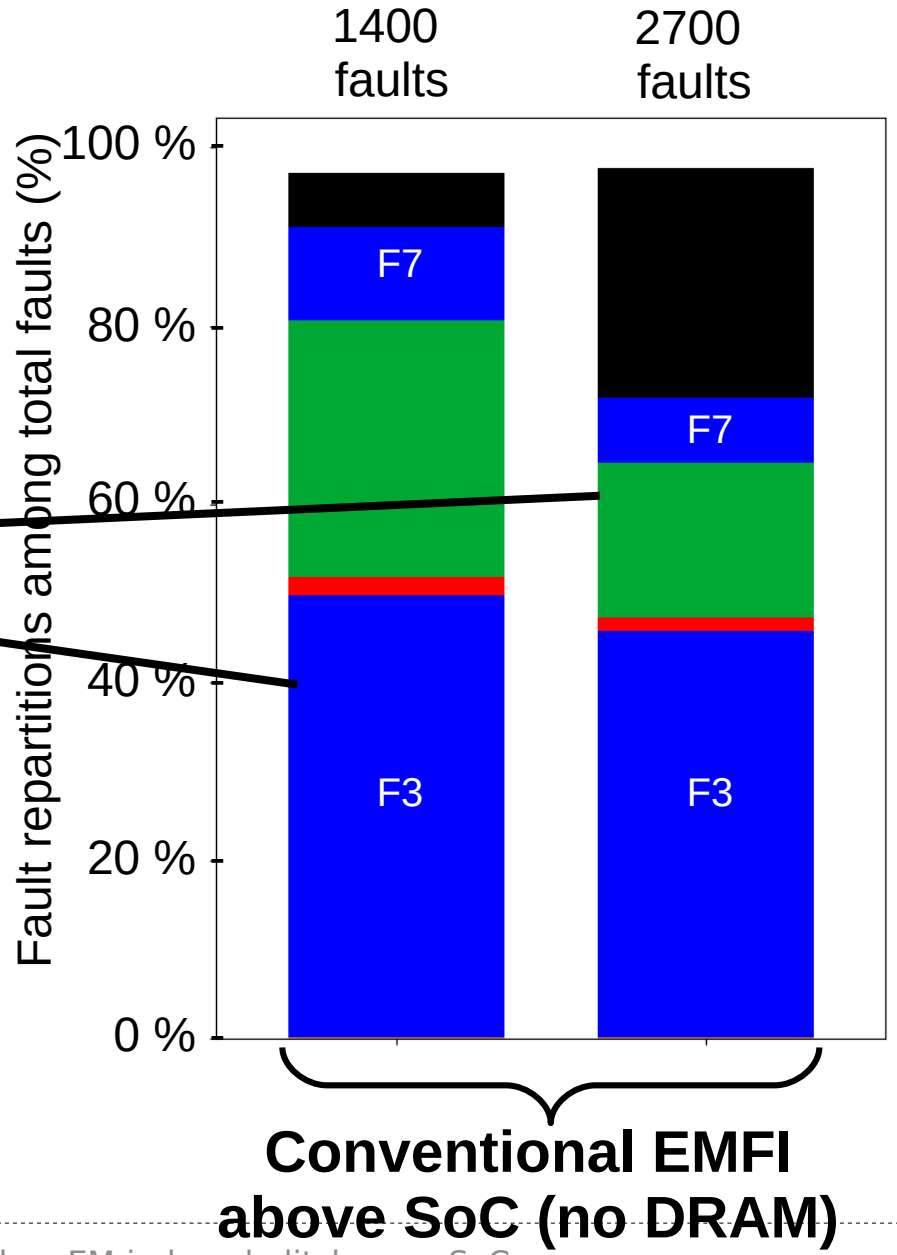
Fault obtained

Method 4 : Conventional voltage glitches :

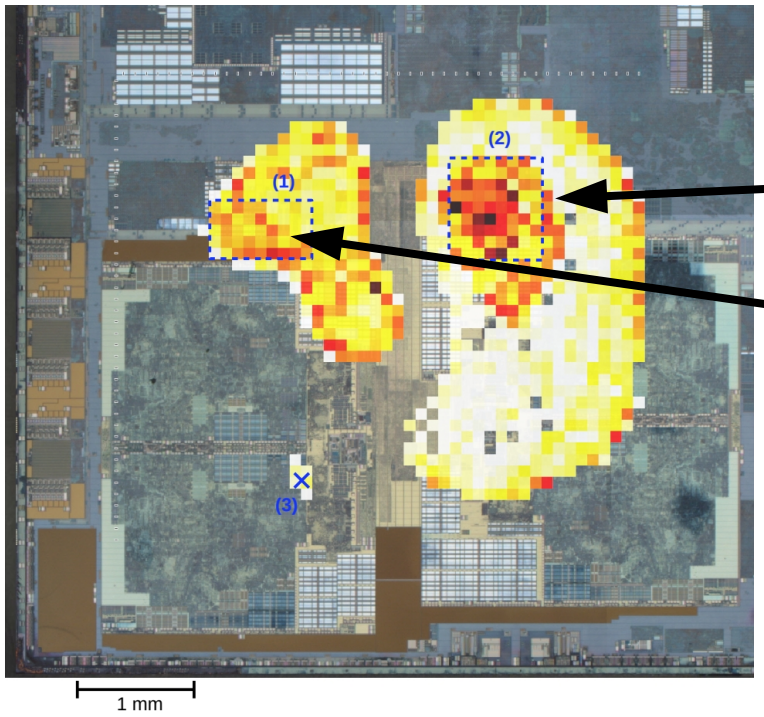


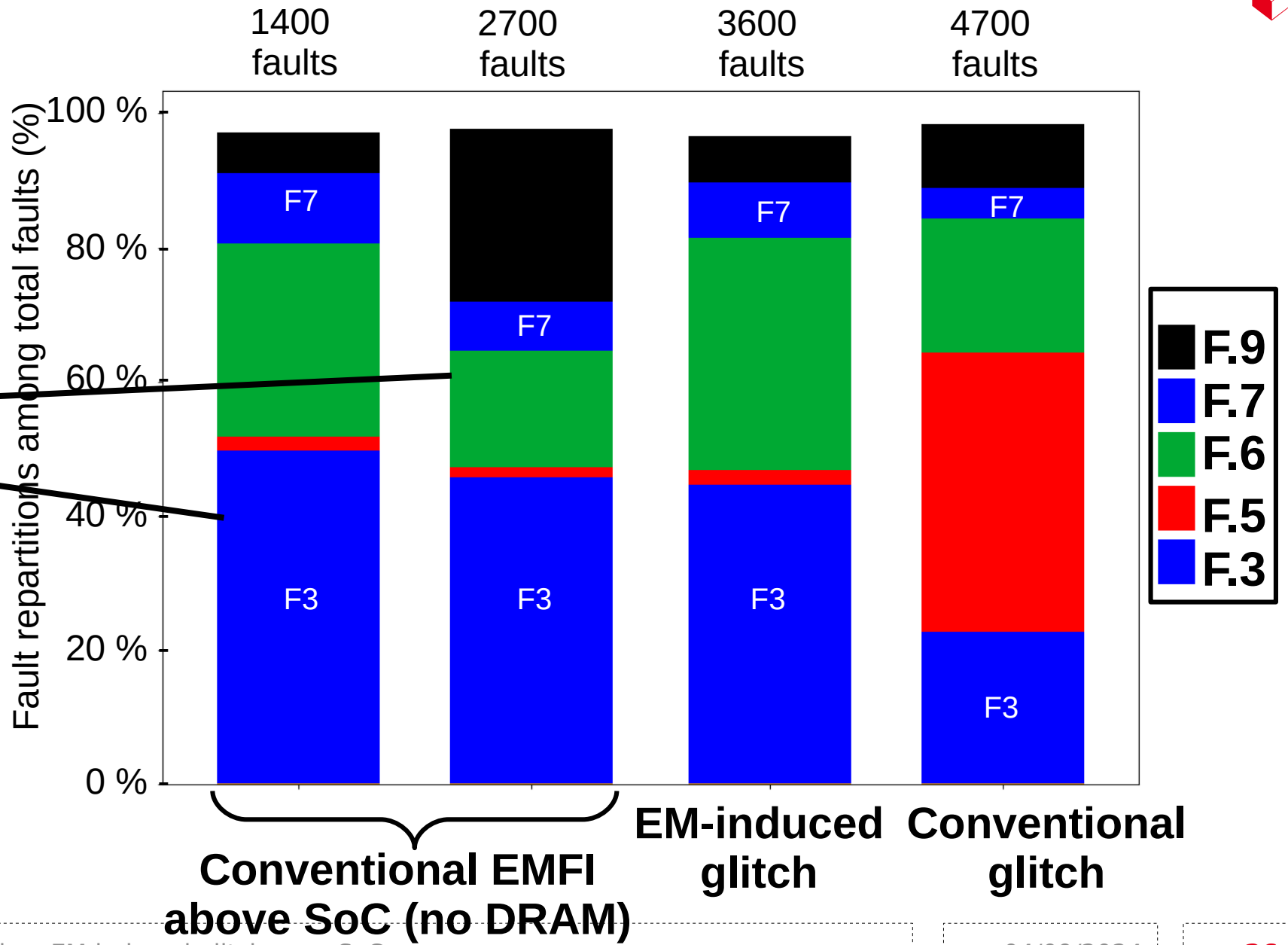
Results


(1)
(2)
(3)
1 mm


1400 faults
2700 faults
Fault repartitions among total faults (%)
100 %
80 %
60 %
40 %
20 %
0 %
F7
F3
F7
F3
F.9
F.7
F.6
F.5
F.3
Conventional EMFI above SoC (no DRAM)

Results


(1)
(2)
(3)
1 mm


1400 faults
2700 faults
3600 faults
4700 faults
Fault repartitions among total faults (%)
100 %
80 %
60 %
40 %
20 %
0 %
F7
F3
F.9
F.7
F.6
F.5
F.3
Conventional EMFI above SoC (no DRAM)
EM-induced glitch
Conventional glitch

Package-on-Package : Conclusion

Evaluation of PoP as a countermeasure against EMFI:

- Conventional EMFI is not possible with our setup on a PoP target

Presentation of 3 methods to inject faults in a SoC despite the PoP:

- EMFI above the SoC, the target DRAM has to be removed (**Vasselle et al. (2017)**).
- **EM-induced glitch** ⇒ injection of EM pulses above the decoupling capacitor to induce glitches on the target power supply rail.
- Conventional voltage glitch on the target power supply rail.

Comparison of the faults induced with the 3 methods:

- EMFI ≈ EM-induced glitch ≠ Voltage glitch

- **Prospects:**
  - Repeat this work to fault a real-life target (PoP SoC implemented in a smartphone)
  - Extend the fault modelization
  - Study EM-induced glitches on others component (bulk capacitors, PMIC).

Thank you for your attention.

Clément Fanjas

clement.fanjas@cea.fr