Charaterization of fault induced by clock-glitch by comparison with faults obtained with laser injection

Ludovic Claudepierre, Edna Rocio Ferrucho Alvarez,
Laurent Le Brizoual, Laurent Pichon

IETR

Glitch vs Localized injection

Voltage and clock glitches

- Easy to implement ✔
- Cheap ✔
- Non-localized with some unexpected effects ✗

EM and Laser injection

- Precise location ✔
- Fault phenomenon more understable physically ✔
- Expensive ✗
- Lot of parameters ✗

Faulting capabilities

Clock glitch

Types of fault:

- Skip
- Skip/Replay
- Instruction corruption

Faults between **which pipeline stages** ?

For classic clock glitch:

- *Alshaer et al.*[1] hypothesis: fault on transfer from flash memory can happen
- Some bits of the word not updated ⇒ instruction corruption

What about TRAITOR glitch ?

Laser injection

- Very precise attack
- *Khuat et al.*[2] : fault pipeline with Laser.

Method proposition:
Compare the fault timing in clock glitch with LFI timing

[1] *Alshaer et al.* - Microarchitectural Insights into Unexplained Behaviors under Clock Glitch Fault Injection - CARDIS 2023.

[2] *Khuat et al.* - Laser fault injection in a 32-bit microcontroller: from the flash interface to the execution pipeline - Workshop on Fault Detection and Tolerance in Cryptography 2021.

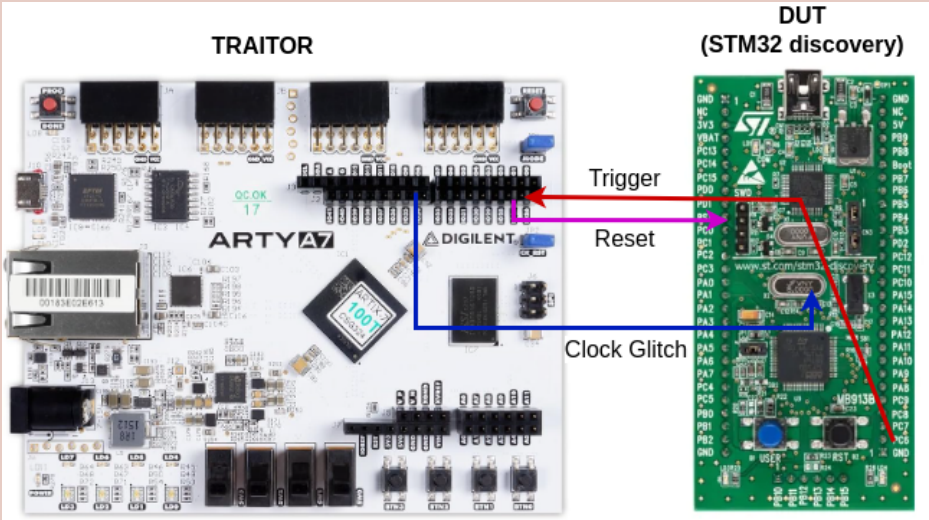
Method overview

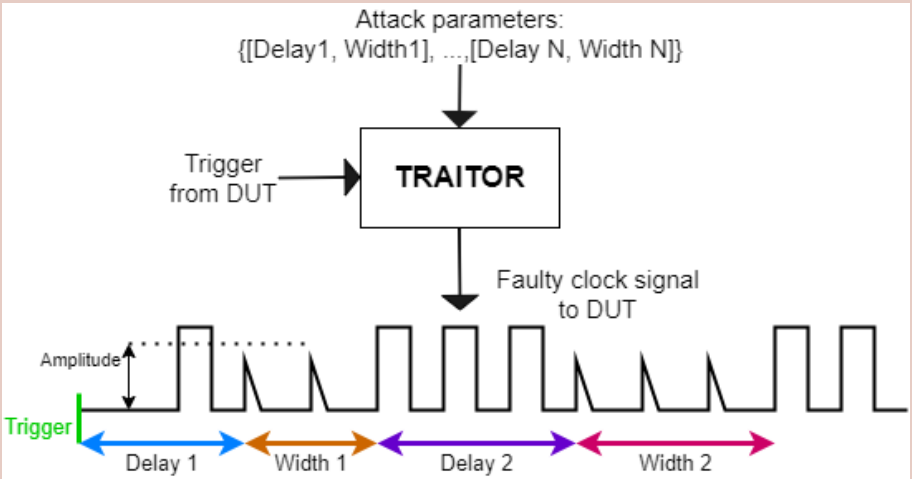
- **Clock glitch platform**: choose a type of fault to analyze
- Choose carefully a **test code** with a unique pattern of instructions words with different execution duration
- **Laser injection**: choose an area to attack that is clearly used in a known pipeline stage.
- For both way of fault injection, **analyze the fault timing** and compare to the execution timing
- **Compare** both results and **conclude**

Clock glitch on STM32F100RB

DUT

- STM32F100RB (Cortex M3)
- Clock 8 MHz
- 32 bits prefetch buffer
- 1 word = 1 instruction 32 bits or 2 instructions 16 bits

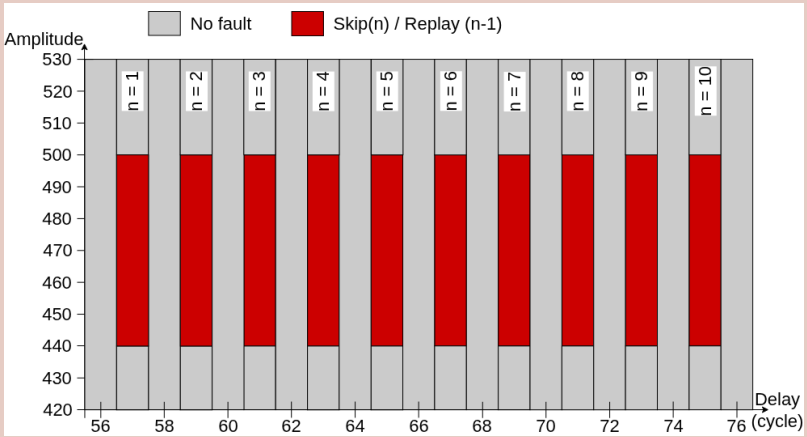




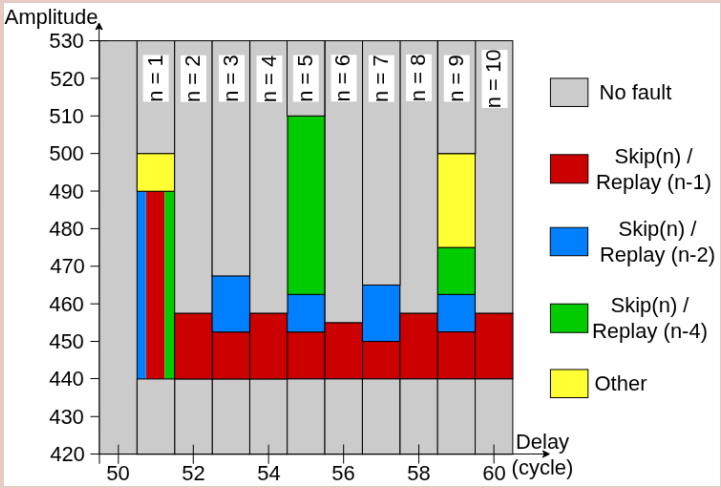
Clock glitch platform: TRAITOR

- Multifault clock glitch
- Generation of glitched clock signal by FPGA
- Parameters: delay (cycle), burst duration (cycle), amplitude

Type of fault to study



a) Faults on 16 bits arithmetic instructions



b) Faults on 32 bits arithmetic instructions

Faults on TRAITOR

Test on simple arithmetic codes

- 2 cases: 16 bits instructions (a) and 32 bits instructions
- Glitch every clock cycle
- Sweep on amplitude value from 420 to 530
- n = index of skipped word

Identified faults effects:

- **Skip$_n$/Replay$_{n-1}$**
- Skip$_n$/Replay$_{n-2}$
- Skip$_n$/Replay$_{n-4}$
- **Hypothesis**: avoid fetching new instructions.
- Fault transfert Flash $\Rightarrow$ Prefetch buffer ?
- Fault Prefetch buffer $\Rightarrow$ Fetch ?

Usercode

Assembly code :	Execution duration
NOPS	2 cycles
...	...
subs R3,R3,#4	2 cycles
adds R0,R0,#11	
subs R4,R4,#5	2 cycles
adds R1,R1,#13	
addw R5,R5,#1	1 cycle
subw R2,R2,#17	1 cycle
addw r6,r6,#19	1 cycle
subs r3,r3,#23	2 cycles
adds r0,r0,#29	
subw R4,R4,#2	1 cycle
addw R1,R1,#31	1 cycle
subs R5,R5,#3	2 cycles
adds R2,R2,#37	
subs r6,r6,#41	2 cycles
adds R3,R3,#5	
subs R0,R0,#43	2 cycles
adds r4,r4,#47	
subw r1,r1,#6	1 cycle
subs R5,R5,#53	2 cycles
adds R6,R6,#61	
...	...
NOPS	2 cycles

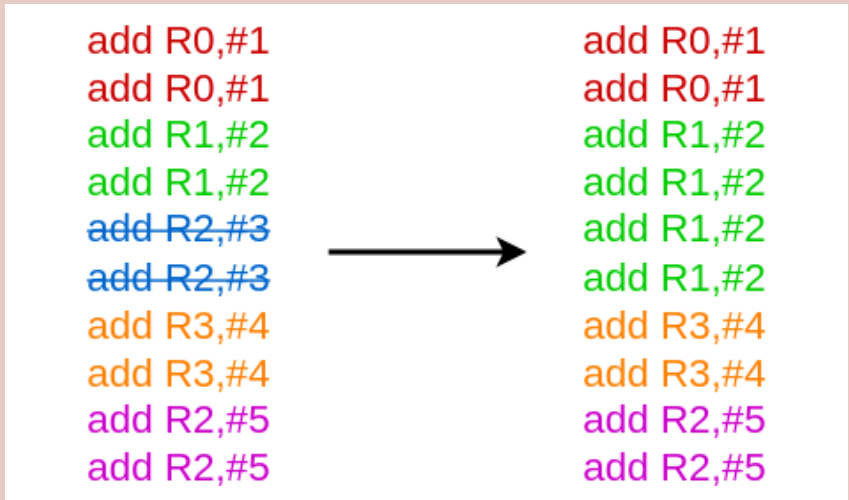
Testcode structure

- Alternate instruction words with different execution time
- 1 arithmetic (SUB or ADD) instruction = 1 cycle
- $\Rightarrow$ alternate 16 bits and 32 bits instructions with no repetitive pattern
- Constraint: keep aligned instructions

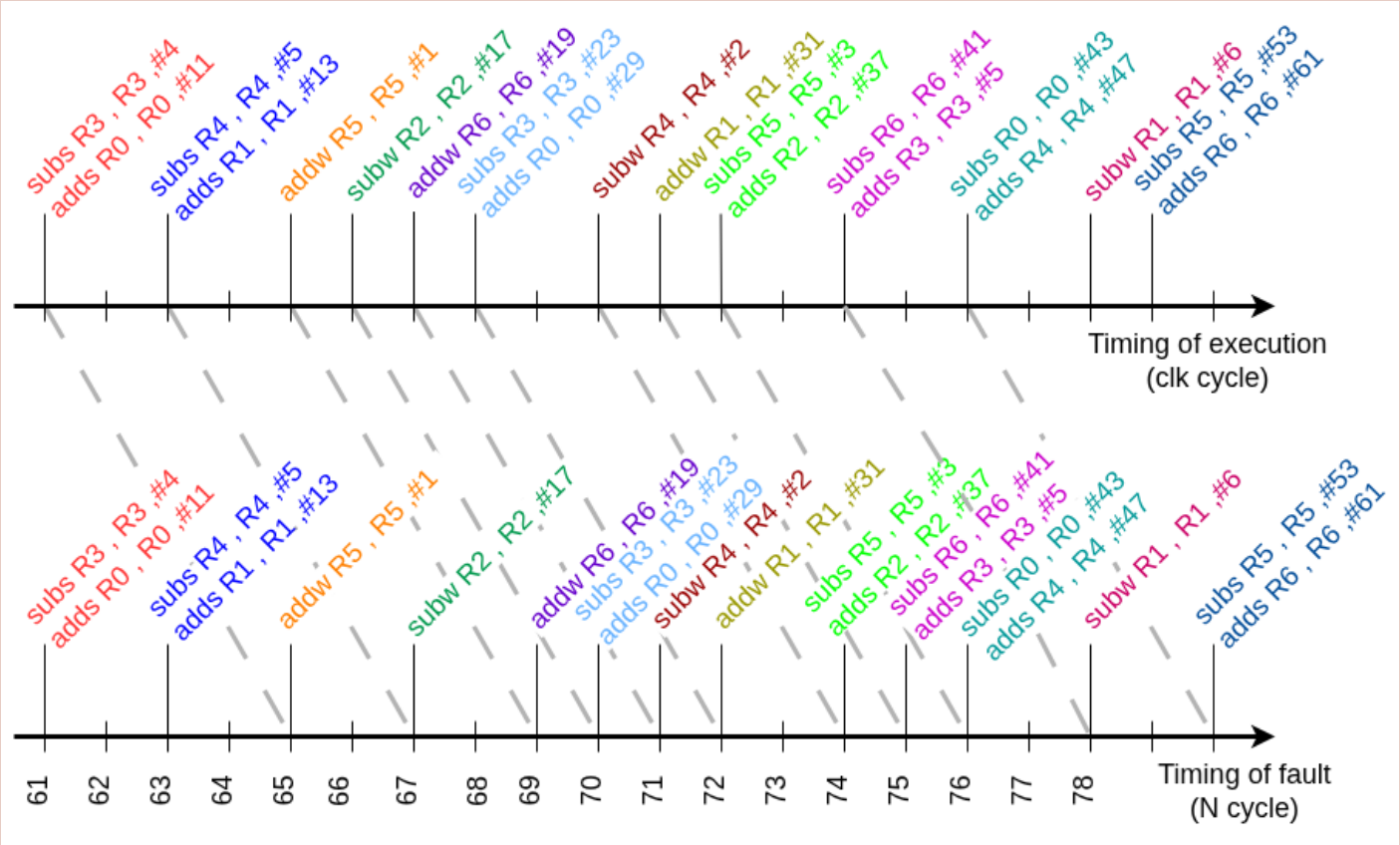
Results with TRAITOR

Parameters of clock glitch

- **Skip**$_n$/**Replay**$_{n-1}$ only fault effect achievable for 16 bits and 32 bits instructions
- Amplitude: adapted to reach that fault model
- Delay values from 61 to 78
- Timing of fault instruction = timing where the instruction is skip



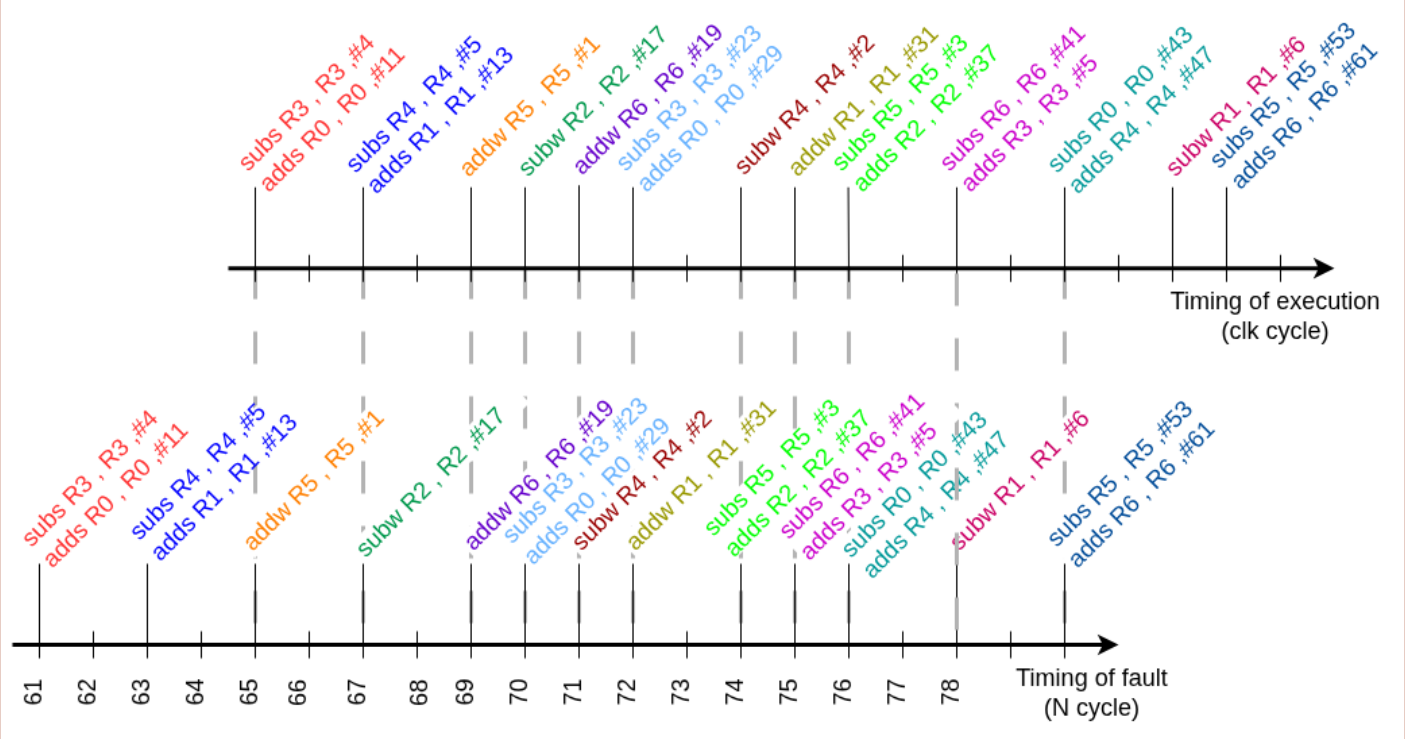
Results with TRAITOR



Analysis

- Fault instruction not during its execution
- Same rhythm: ΔW constant between skip time and execution time
- $\Delta W = 2$ words

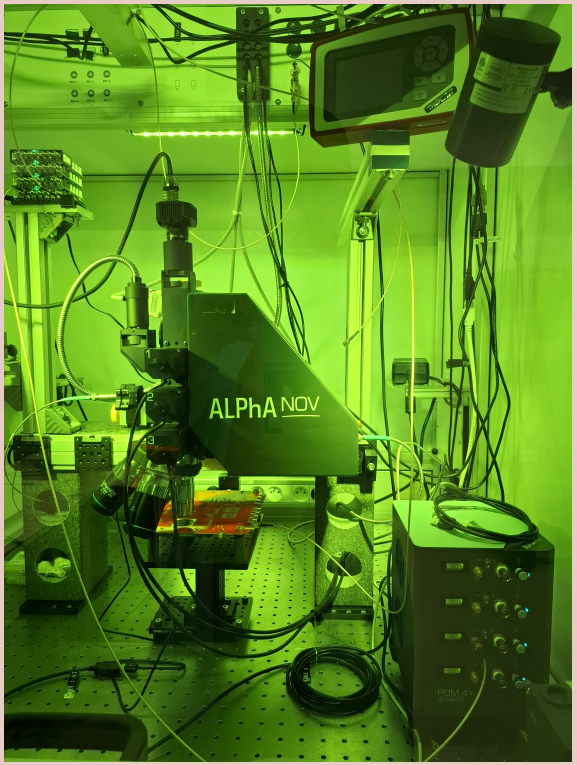
Results with TRAITOR



Analysis

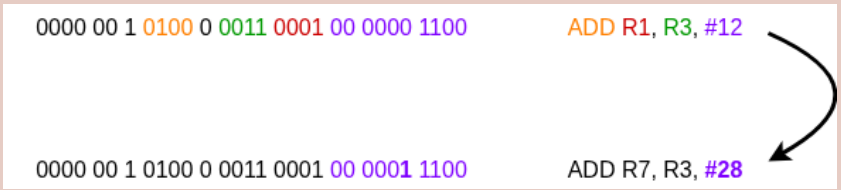
- Fault instruction not during its execution
- Same rhythm: ΔW constant between skip time and execution time
- $\Delta W = 2$ words

Laser Platform

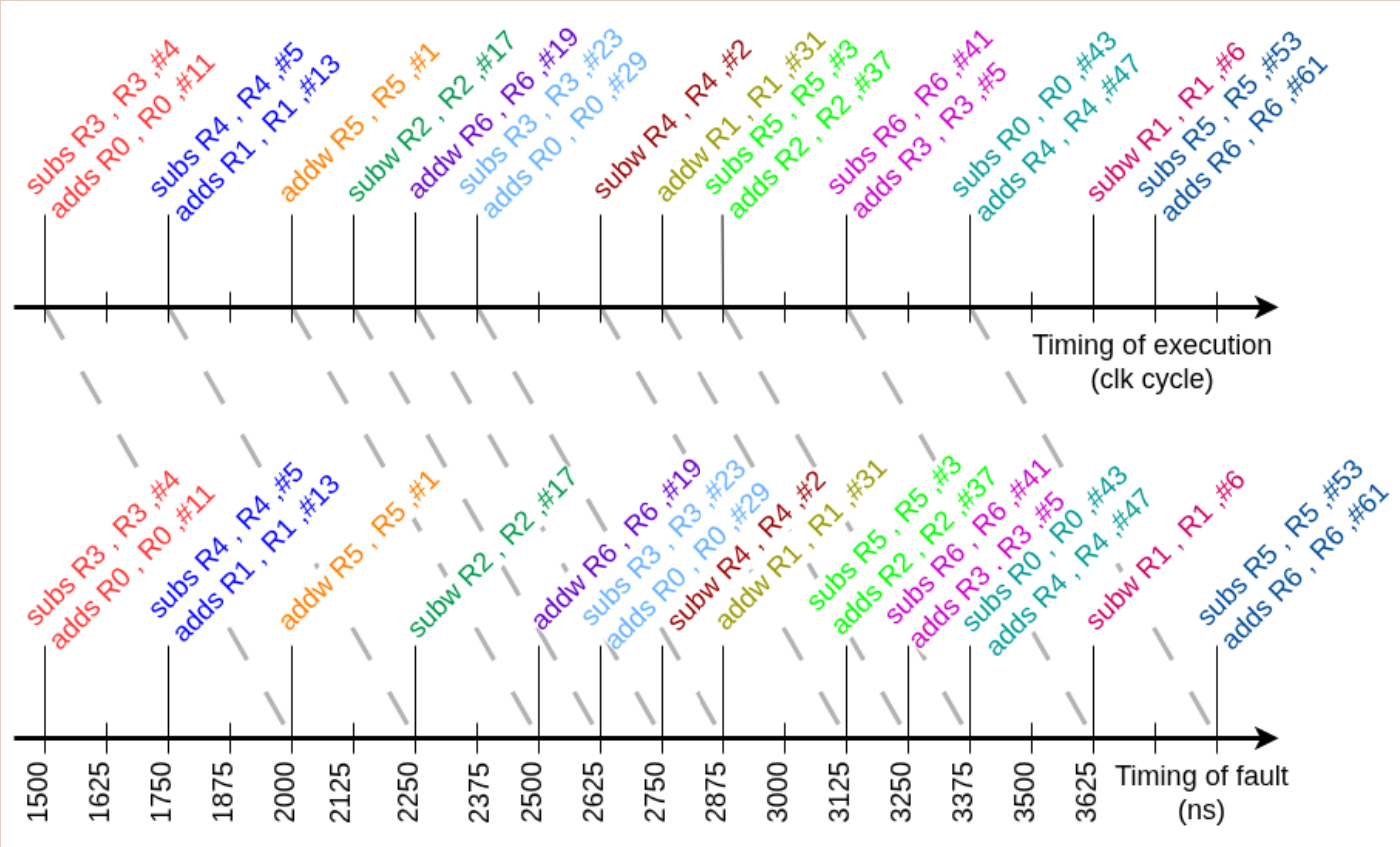


Laser Platform

- Targeted area: red square in flash memory, fault on differents bits.
- Fault model: bit set
- Objectif x5, $\lambda = 1064$ nm
- Parameters: 400 ns pulsewidth, 100 mW power, test delay every 125 ns



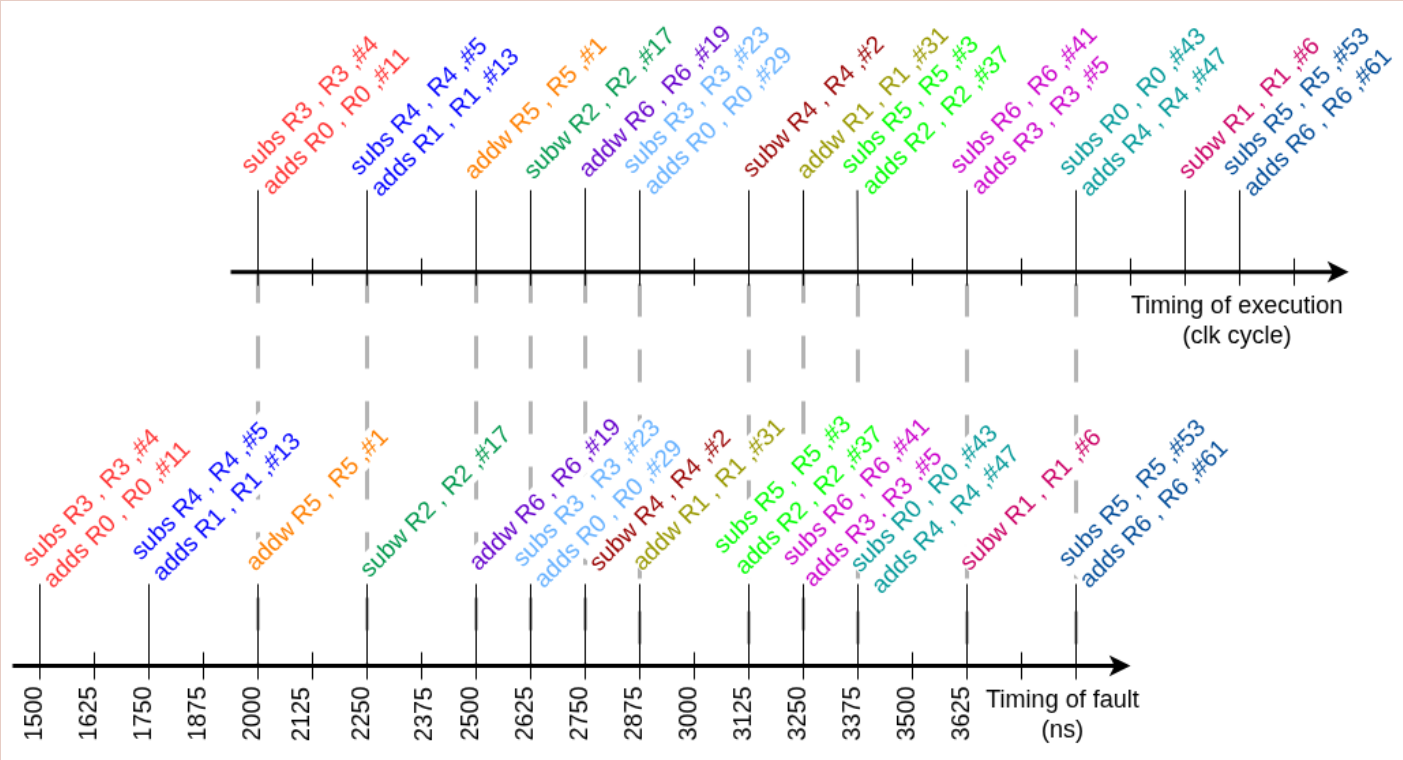
Results with Laser



Analysis

- Fault instruction not during its execution
- Same rhythm: ΔW constant between fault time and execution time
- $\Delta W = 2$ words

Results with Laser



Analysis

- Fault instruction not during its execution
- Same rhythm: ΔW constant between fault time and execution time
- $\Delta W = 2$ words

Comparison and conclusion

Analysis

- In both experiment $\Delta T = 2$ words between fault time and execution time
- So the stage during the fault occurs is the same with TRAITOR and with Laser
- Flash memory is only accessed during the transfer to prefetch buffer
- We can conclude that TRAITOR $\text{Skip}_n/\text{Replay}_{n-1}$ is due to a disturbance on the prefetching stage.

Conclusion and Perspectives

Conclusions

- Clock-glitch non localized fault: where is the weakness ?
- Comparing the execution timing and fault timing: we don't fault at exec
- Comparing with a localized attack by Laser: confirm hypothesis that TRAITOR skip/replay act on the transfer between flash memory and prefetch buffer.

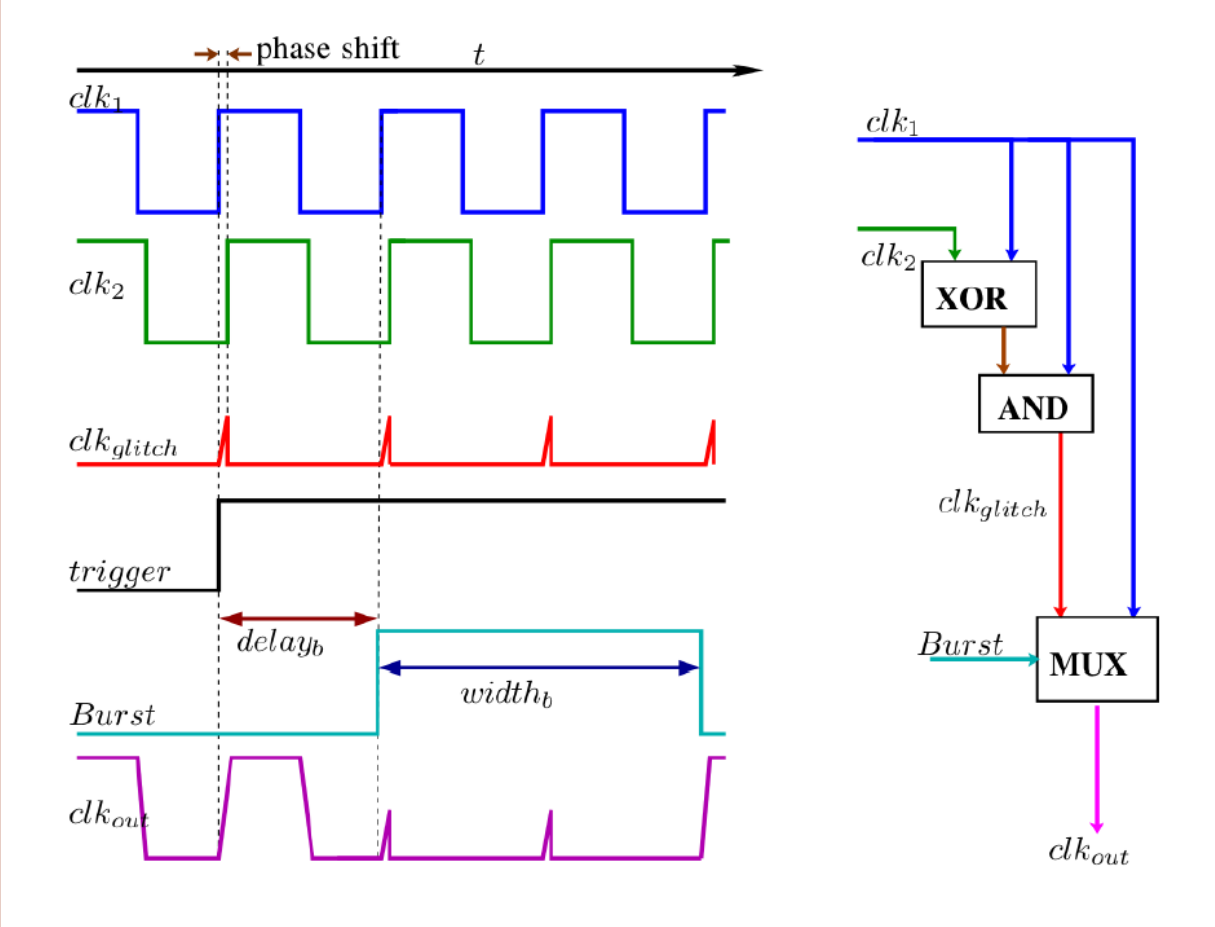
Perspectives

- Other kind of faults exists (around ldr/str instructions or branch for example)
- Could use same method to identify the location of the disturbance and the corresponding pipeline stage

Any question ?

Detail TRAITOR signal

How the TRAITOR output is built[1]



1 ammplitude = 1 step of phase shift (step depending on PLL parameters)

[1] *Claudepierre et al.* - TRAITOR: A Low-Cost Evaluation Platform for Multifault Injection. - ASSS 2021