DE LA RECHERCHE À L'INDUSTRIE

# ELECTROMAGNETIC FAULT INJECTION ON MICROCONTROLLERS

**Nicolas Moro**[1,3], Amine Dehbaoui[2], Karine Heydemann[3], Bruno Robisson[1], Emmanuelle Encrenaz[3]

[1] **CEA**
Commissariat à l'Energie Atomique et aux Energies Alternatives

[2] **ENSM.SE**
Ecole Nationale Supérieure des Mines de Saint-Etienne

[3] **LIP6 - UPMC**
Laboratoire d'Informatique de Paris 6
Université Pierre et Marie Curie

Chip-to-Cloud Security Forum
**Smart Trusted Technologies & Services for the Networked Society**
September 25-27, 2013 – Nice, French Riviera

## MOTIVATIONS

- **Security** of microcontroller-based embedded systems against fault injection attacks

- **Target**: ARM Cortex-M3 microcontroller
- **Fault injection means**: Pulsed electromagnetic fault injection

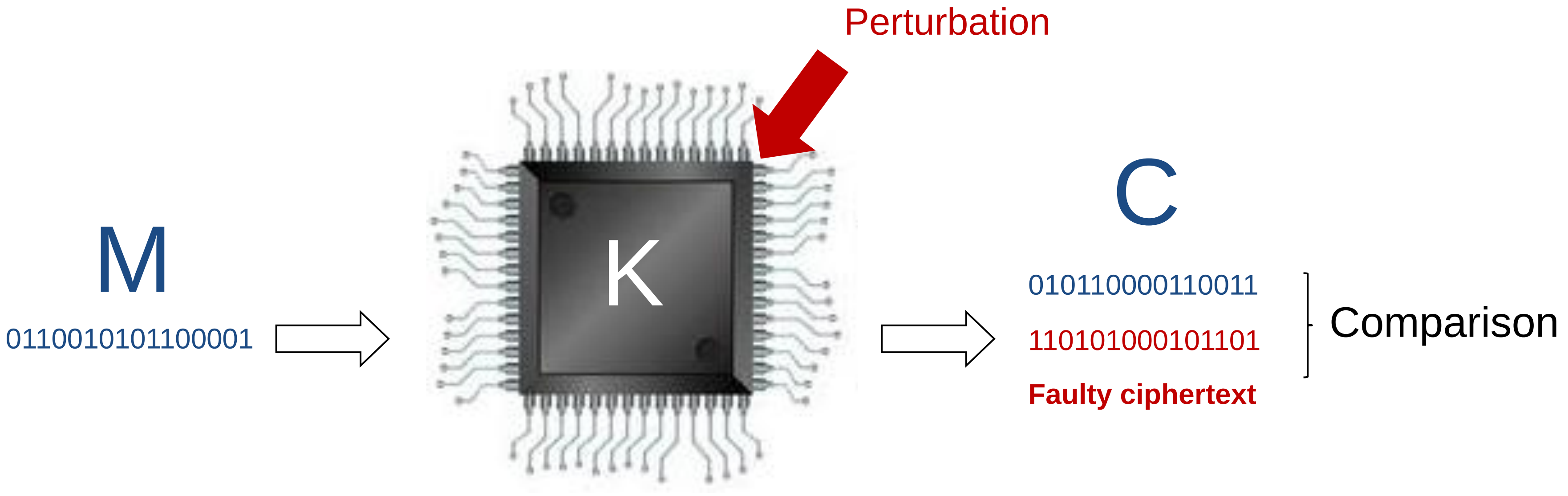
- Theoretical attacks rely on an **attacker's fault model**
- Electromagnetic fault injection is quite recent
- Very few **in-depths studies** of the effects on complex systems

➔ Better understanding of the effects of **EM fault injection**

➔ Detailed fault model at a **register-transfer level**

# OUTLINE

## FAULT INJECTION ATTACKS



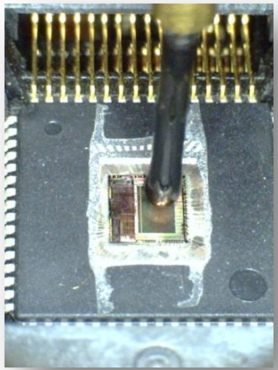
- Several **physical ways** to inject faults into a circuit's computation
- Necessary for an attacker to **know the type of injected faults**

| | |
|---|---|
| **Fault target** | Data, instructions |
| **Fault type** | Bit flip, reset at 0, set at 1, stuck |
| **Granularity** | Bit, byte, word |
| **Determinism** | Deterministic, metastable, random |
| **Temporal aspect** | Single piece of data/instruction, multiple |

## EXPERIMENTAL SETUP

**Pulsed electromagnetic fault injection**

- **Transient and local** effect of the fault injection
- **Standard circuits not protected** against this technique
- **Solenoid** used as an injection antenna
- Up to **200V** sent on the injection antenna, pulses width longer than **10ns**



## Microcontroller based on an ARM Cortex-M3

- Frequency 56 MHz

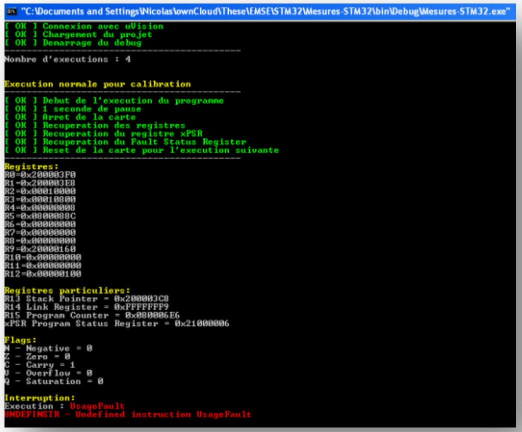
-16/32 bits **Thumb2** RISC instruction set

- ARMv7-M modified Harvard architecture

- SWD link to **debug the microcontroller**

# DETAILED EXPERIMENTAL PROTOCOL

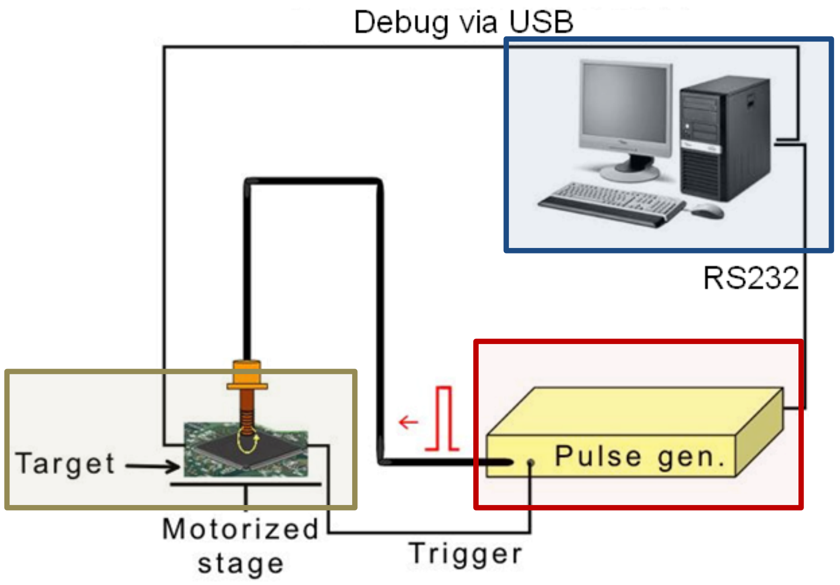
- Experiment **driven by the computer**
- Execution of a **computation on the target device**
- Sending of a **voltage pulse**
- **Stop** of the microcontroller
- **Harvesting** of the microcontroller's internal data
- Analysis of the obtained results



```
"C:\Documents and Settings\Nicolas\ownCloud\These\EMSE\STM32\Mesures-STM32\bin\Debug\Mesures-STM32.exe"
[ OK ] Connexion avec uVision
[ OK ] Chargement du projet
[ OK ] Demarrage du debug
------------------------------------------
Nombre d'executions : 4


Execution normale pour calibration
------------------------------------------
[ OK ] Debut de l'execution du programme
[ OK ] 1 seconde de pause
[ OK ] Arret de la carte
[ OK ] Recuperation des registres
[ OK ] Recuperation du registre xPSR
[ OK ] Recuperation du Fault Status Register
[ OK ] Reset de la carte pour l'execution suivante
------------------------------------------
Registres:
R0=0x200003F0
R1=0x200003E8
R2=0x00010000
R3=0x00010800
R4=0x00000008
R5=0x0800088C
R6=0x00000000
R7=0x00000000
R8=0x00000000
R9=0x20000160
R10=0x00000000
R11=0x00000000
R12=0x00000100

Registres particuliers:
R13 Stack Pointer = 0x200003C8
R14 Link Register = 0xFFFFFFF9
R15 Program Counter = 0x080006E6
xPSR Program Status Register = 0x21000006

Flags:
N - Negative = 0
Z - Zero = 0
C - Carry = 1
V - Overflow = 0
Q - Saturation = 0

Interruption:
Execution : UsageFault
UNDEFINSTR - Undefined instruction UsageFault
```

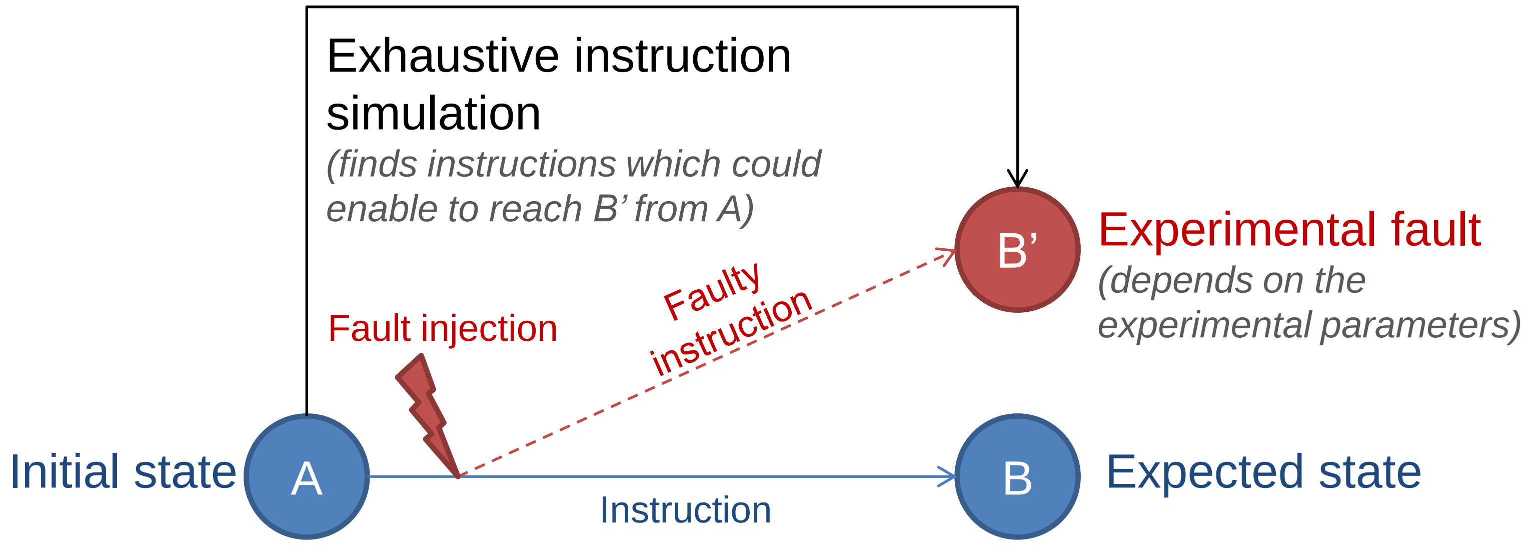
## Main experimental parameters

- ***Position*** of the injection antenna
- ***Electric parameters*** of the pulse
- ***Injection time*** of the pulse
- ***Executed code*** on the microcontroller



# OUTLINE

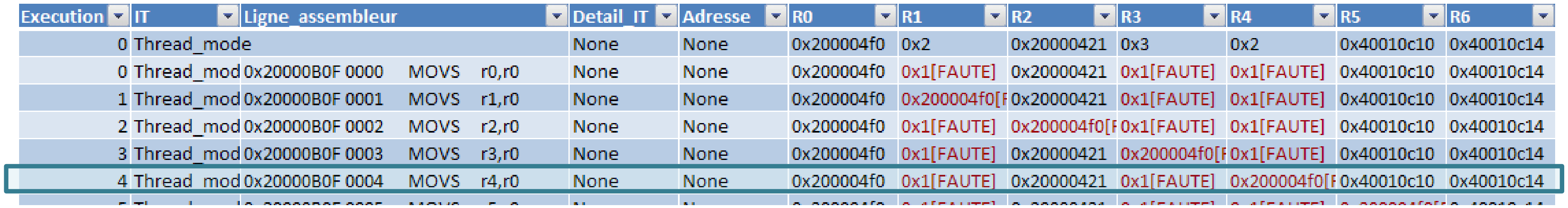
# GENERAL APPROACH


Exhaustive instruction simulation
(finds instructions which could enable to reach B' from A)
Fault injection
Faulty instruction
B'
Experimental fault
(depends on the experimental parameters)
Initial state
A
Instruction
B
Expected state

| Output pieces of data | Detail |
|---|---|
| **R0 to R12** | General-purpose registers |
| **R13 (SP)** | Stack pointer |
| **R14 (LR)** | Link register |
| **R15 (PC)** | Program counter |
| **XPSR** | Program Status Register<br>- Flags<br>- Details about the triggered interruptions<br>- Details about the execution mode |
| **Result** | Memory address that contains the calculation's output |

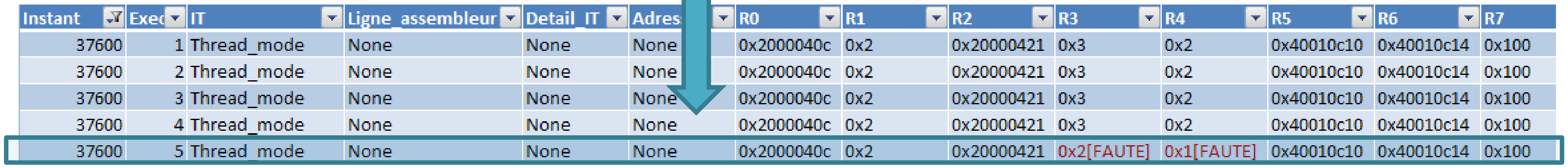
# SIMULATION OF A FAULT MODEL

## Instruction replacement simulation



| Execution | IT | Ligne_assembleur | Detail_IT | Adresse | R0 | R1 | R2 | R3 | R4 | R5 | R6 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Thread_mode | | None | None | 0x200004f0 | 0x2 | 0x20000421 | 0x3 | 0x2 | 0x40010c10 | 0x40010c14 |
| 0 | Thread_mod | 0x20000B0F 0000 MOVS r0,r0 | None | None | 0x200004f0 | 0x1[FAUTE] | 0x20000421 | 0x1[FAUTE] | 0x1[FAUTE] | 0x40010c10 | 0x40010c14 |
| 1 | Thread_mod | 0x20000B0F 0001 MOVS r1,r0 | None | None | 0x200004f0 | 0x200004f0[F | 0x20000421 | 0x1[FAUTE] | 0x1[FAUTE] | 0x40010c10 | 0x40010c14 |
| 2 | Thread_mod | 0x20000B0F 0002 MOVS r2,r0 | None | None | 0x200004f0 | 0x1[FAUTE] | 0x200004f0[F | 0x1[FAUTE] | 0x1[FAUTE] | 0x40010c10 | 0x40010c14 |
| 3 | Thread_mod | 0x20000B0F 0003 MOVS r3,r0 | None | None | 0x200004f0 | 0x1[FAUTE] | 0x20000421 | 0x200004f0[F | 0x1[FAUTE] | 0x40010c10 | 0x40010c14 |
| 4 | Thread_mod | 0x20000B0F 0004 MOVS r4,r0 | None | None | 0x200004f0 | 0x1[FAUTE] | 0x20000421 | 0x1[FAUTE] | 0x200004f0[F | 0x40010c10 | 0x40010c14 |
| 5 | Thread_mod | 0x20000B0F 0005 MOVS r5,r0 | None | None | 0x200004f0 | 0x1[FAUTE] | 0x20000421 | 0x1[FAUTE] | 0x1[FAUTE] | 0x200004f0[F | 0x40010c14 |

## Experimental measurements



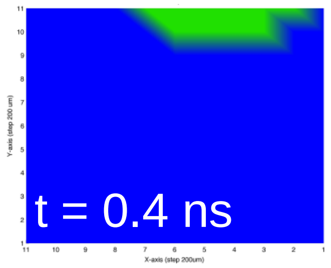
| Instant | Exec | IT | Ligne_assembleur | Detail_IT | Adres | R0 | R1 | R2 | R3 | R4 | R5 | R6 | R7 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 37600 | 1 | Thread_mode | None | None | None | 0x2000040c | 0x2 | 0x20000421 | 0x3 | 0x2 | 0x40010c10 | 0x40010c14 | 0x100 |
| 37600 | 2 | Thread_mode | None | None | None | 0x2000040c | 0x2 | 0x20000421 | 0x3 | 0x2 | 0x40010c10 | 0x40010c14 | 0x100 |
| 37600 | 3 | Thread_mode | None | None | None | 0x2000040c | 0x2 | 0x20000421 | 0x3 | 0x2 | 0x40010c10 | 0x40010c14 | 0x100 |
| 37600 | 4 | Thread_mode | None | None | None | 0x2000040c | 0x2 | 0x20000421 | 0x3 | 0x2 | 0x40010c10 | 0x40010c14 | 0x100 |
| 37600 | 5 | Thread_mode | None | None | None | 0x2000040c | 0x2 | 0x20000421 | 0x2[FAUTE] | 0x1[FAUTE] | 0x40010c10 | 0x40010c14 | 0x100 |

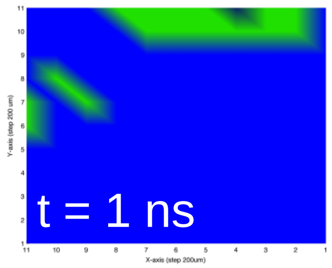
- Two lines are **equal** ⬄ **R0 to R12 + XPSR + result + SP + PC** are equal

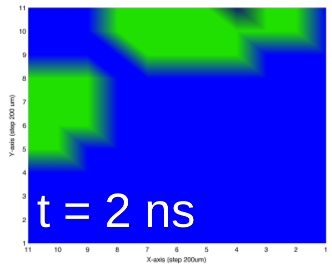
## OUTLINE

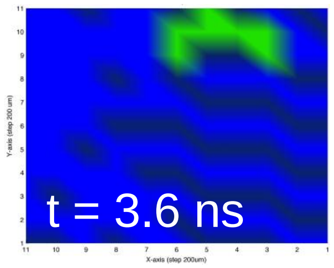
# INFLUENCE OF THE ANTENNA’S POSITION

- **Green : hardware interrupts have been triggered**
- **Red : faults on the output value have been obtained**

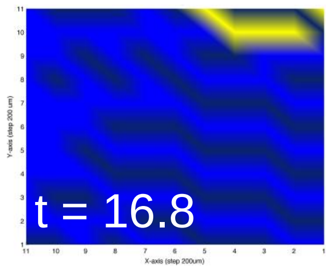


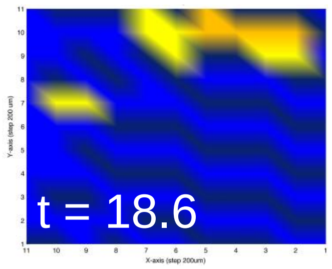


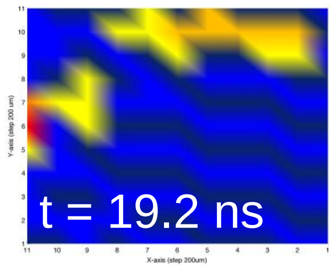


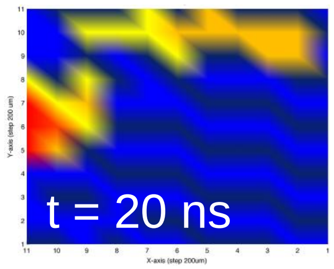


**Frequency 56 MHz – Pulse width 10 ns – Pulse voltage 190V – Period 17ns**





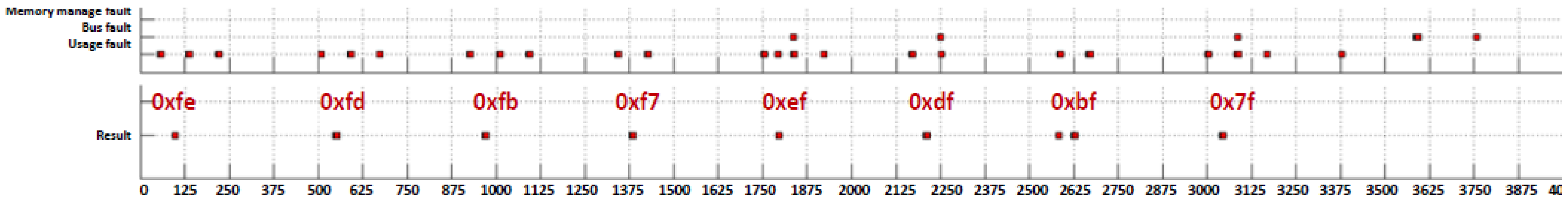




- **Target instruction** : single LOAD instruction that loads 0x12345678 into R8
- **20 ns** time interval, by steps of **200 ps - 3 mm** square, by steps of **200 µm**
- **Variable increase** of the Hamming weight of the loaded piece of data
- **No fault** on other registers than **R8** (except for very few faults on R0)

## INFLUENCE OF THE INJECTION TIME

### Example of temporal cartography on an addition loop



**0xfe 0xfd 0xfb 0xf7 0xef 0xdf 0xbf 0x7f**

**Observations:**

- One power of two has not been added
- BusFault or UsageFault interrupts

➔ Does our fault injection have an effect on **the data flow or the control flow** ?

**Test program:**
loop to sum the elements of an array that contains eight powers of two
3.5 µs, by steps of 200 ps

**Expected result:** 0xFF

```
addition_loop:
ldr r4, [r2,r1, lsl #2] ; r4 = array[i]
ldr r3, [r0,#0]          ; r3 = result
add r3, r4               ; r3 = r3 + r4
str r3, [r0,#0]          ; result = r3
add r1, r1, #1           ; r1 = r1 + 1
cmp r1, #8               ; r1 == 8 ?
blt addition_loop
```

## INFLUENCE OF THE PULSE'S VOLTAGE

**LDR R4, PC#44** with 0x12345678 at the address PC#44

| Pulse voltage | Output value | Occurrence rate |
|---|---|---|
| 172V | 1234 5678 | 100 % |
| 174V | **9**234 5678 | 73 % |
| 176V | **FE**34 5678 | 30 % |
| 178V | **FFF**4 5678 | 53 % |
| 180V | **FFFD** 5678 | 50 % |
| 182V | **FFFF 7F**78 | 46 % |
| 184V | **FFFF FFFB** | 40 % |
| 186V | **FFFF FFFF** | 100 % |

- Simulation : corresponds to **no instruction replacement**
- Looks like a **set at 1** fault model on the **Flash memory data transfers**
- Possible **precharge of the data bus** on this architecture

## OUTLINE

## FAULTS ON THE CONTROL FLOW

- Experiments with a sequence of **NOP** (BF 00)
- **Four kinds of faults**
  - Fault on **R7**
  - The program does not stop
  - **UsageFault** exceptions (Invalid Instruction / No Coprocessor)
  - Fault on **R0**
- Sometimes a modification of the **number of executed cycles**
- Simulation on the ISA: some instructions can explain the results
- **Some faults only equivalent to a STR R0, [R0, #0] instruction**

| | | | |
|---|---|---|---|
| **NOP - BF00** | 1011 1111 0000 0000 | **STR R0, [R0, #0] - 6000** | 0110 0000 0000 0000 |
| **NOP - BF00** | 1011 1111 0000 0000 | **NOP - BF00** | 1011 1111 0000 0000 |

# SEVERAL PHYSICAL POSSIBILITIES

- **Direct coupling on the bus lines**
  → ***Unlikely****: otherwise we could also inject faults on the address bus*

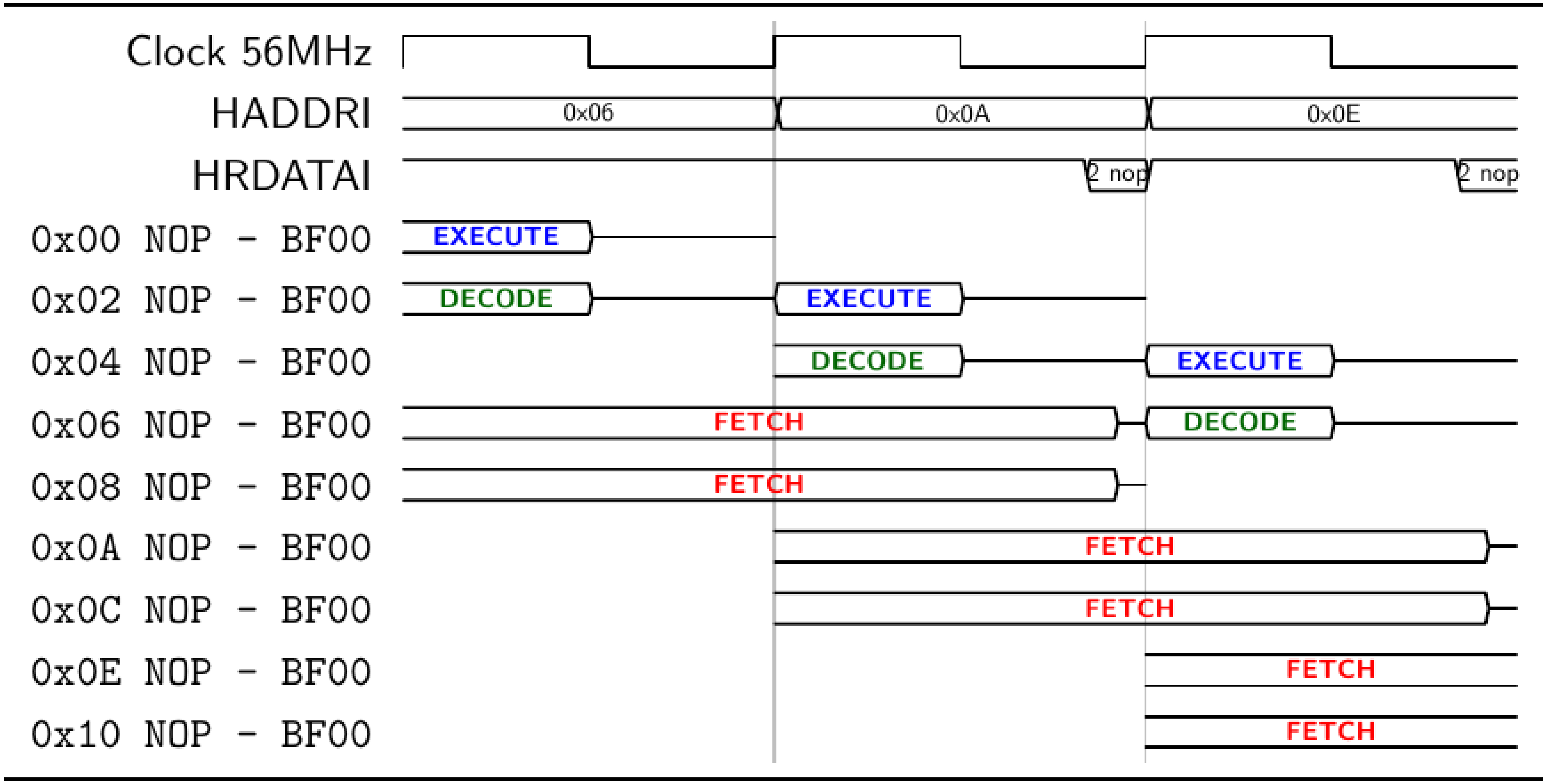
- **Coupling on the power grid or the ground network**
  → ***Likely****: could slow down the transfers on the bus*

- **Coupling on the clock tree**
  → ***Possible****: could provoke a shorter clock cycle*

**Investigation of timing constraints violation as a fault injection means**
**2012 -** Loïc Zussa, Jean-Max Dutertre, Jessy Clédière, Bruno Robisson, Assia Tria
*27th Conference on Design of Circuits and Integrated Systems (DCIS)*. Avignon

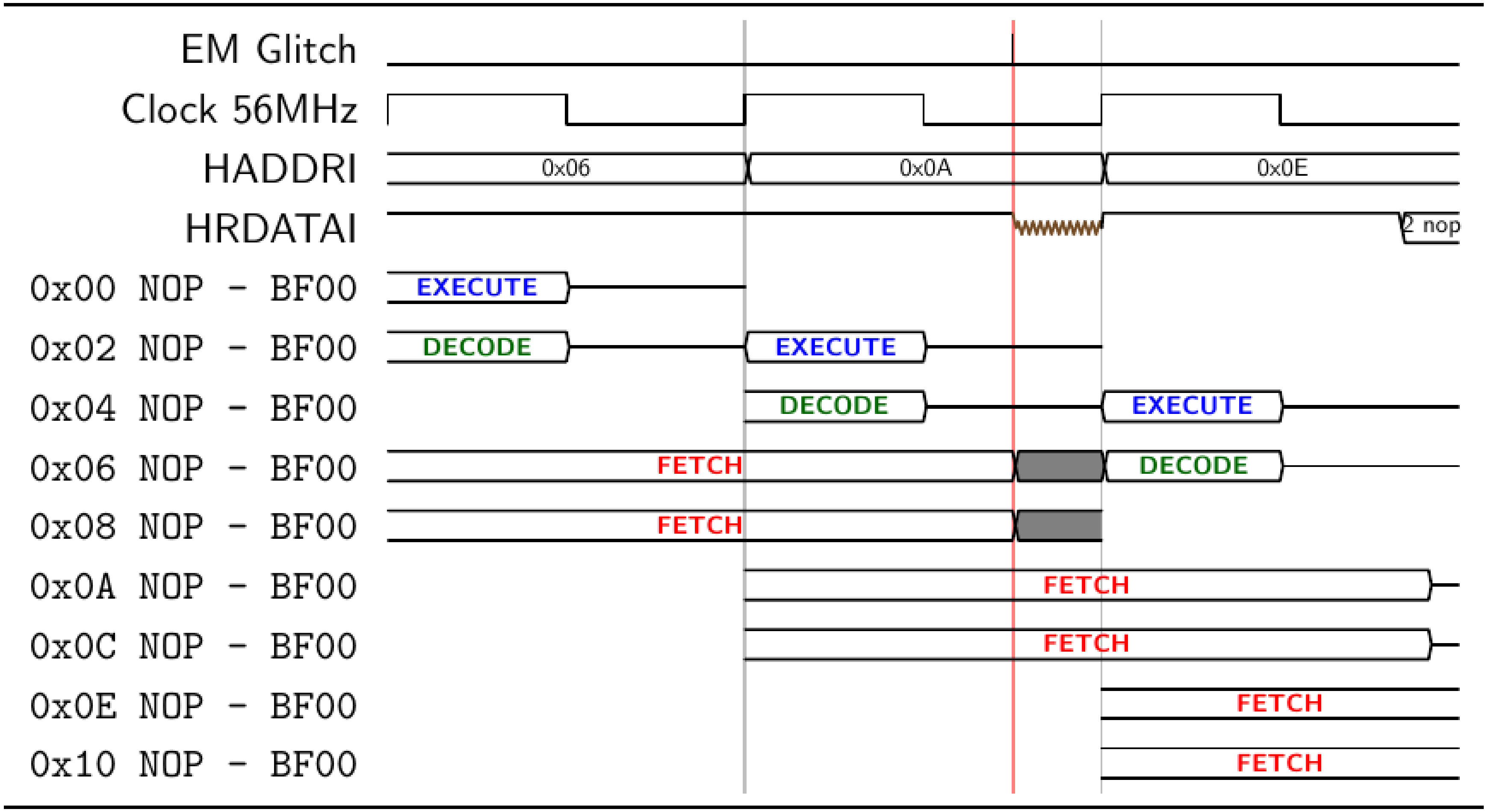
## INSTRUCTION FETCH

### Normal behaviour



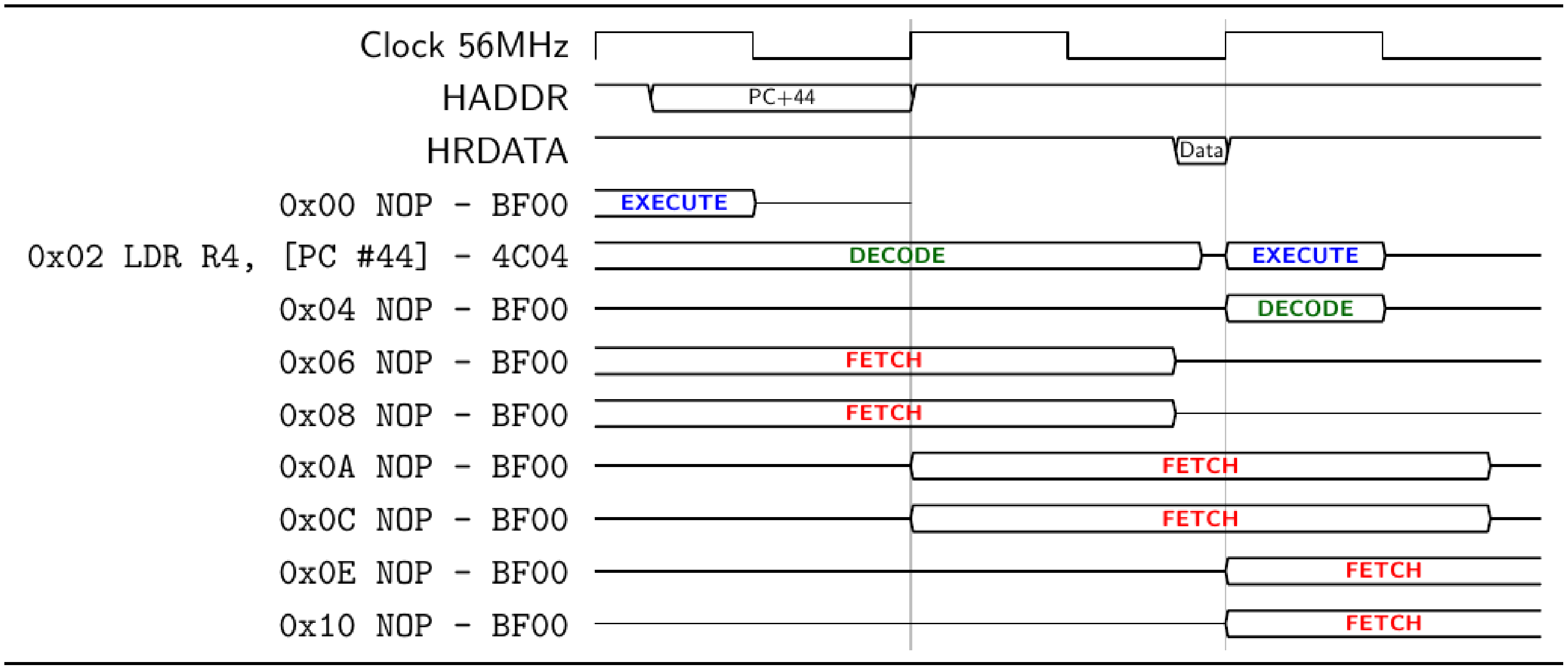
# INSTRUCTION FETCH

## With an electromagnetic fault injection



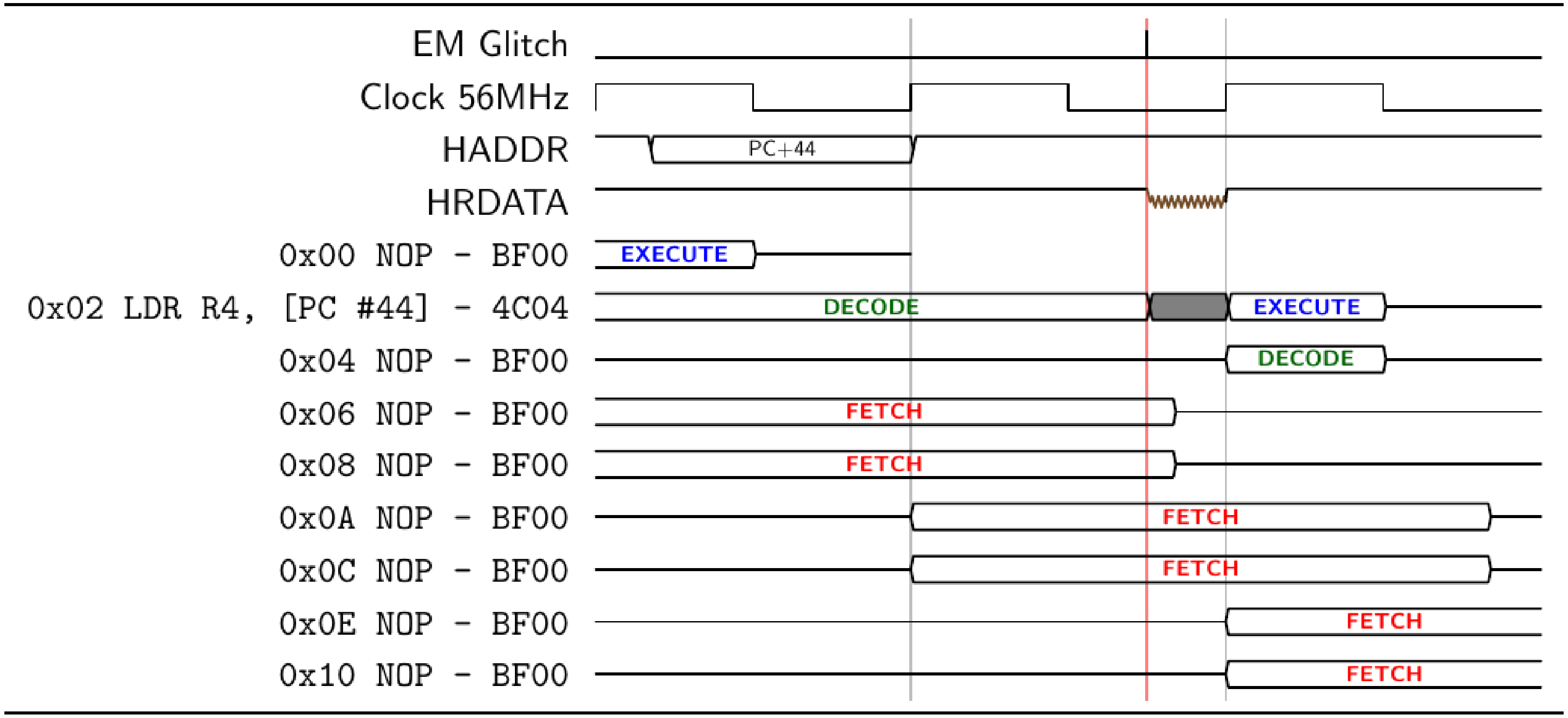
# DATA LOAD FROM THE FLASH MEMORY

## Normal behaviour



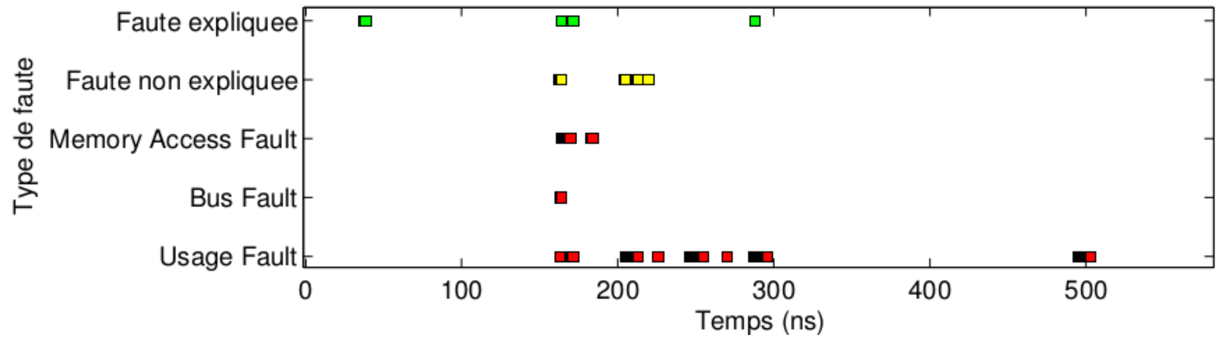
# DATA LOAD FROM THE FLASH MEMORY

## With an electromagnetic fault injection



# INSTRUCTION SKIP EFFECTS

- Many instruction replacements have an effect which is equivalent to **instruction skips**



In our experiments, up to **30%** of successful fault injection lead to an instruction skip effect

Balasch *et al.* (FDTC 2011) **50%** on another platform

**Possible explanations**

- **ARM conditional execution** ?

(unlikely on this architecture with the Thumb2 instruction set)

- **Replacement** by a useless instruction ?

(writing on a dead register, on a useless memory address)

- **Re-execution** of the previous instruction ?

(many instructions are idempotent)

## OVERVIEW OF THE DEFINED FAULT MODEL

➔ **Possible to fault the transfers from the Flash memory**

**Consequences regarding the instruction flow**

- Instructions **replacements**
- Instruction **skips under certain conditions** (~ 20-30% of time)
- Some instructions may be **more sensitive than others**
- Some registers seem to be **more sensitive than others**

**Consequences regarding the data flow**

- Corruption of the `LOAD` instructions from the Flash memory (encryption keys,…)
- Some metastability phenomena, but deterministic under some conditions
- Faulty values with higher Hamming weight (on this architecture)

## OUTLINE

## CONCLUSION AND PERSPECTIVES

- A first attempt of **fault model** for EM fault injection on a 32-bit µC
- Corruption of the **transfers from the Flash memory** on the buses
- The obtained effects seem **very similar** to the ones obtained with **clock glitches or other fault injection means**
- **Similar effects** obtained previously on a very different architecture
  (Atmel AVR ATmega128 8-bit microcontroller)
- Possibility to perform **instruction skips** under some specific conditions

**Perspectives**

- Use more advanced debug techniques to understand better instruction replacements
- Define a higher-level fault model that can be used for theoretical attacks

DE LA RECHERCHE À L'INDUSTRIE

## THANK YOU FOR YOUR ATTENTION

Ecole Nationale Supérieure des Mines SAINT-ETIENNE

UPMC SORBONNE UNIVERSITÉS

**Any questions ?**