

# An In-depth and Black-box characterization of the effects of Clock Glitches on 8-bit MCUs

**Josep Balasch**, Benedikt Gierlichs, and Ingrid Verbauwhede

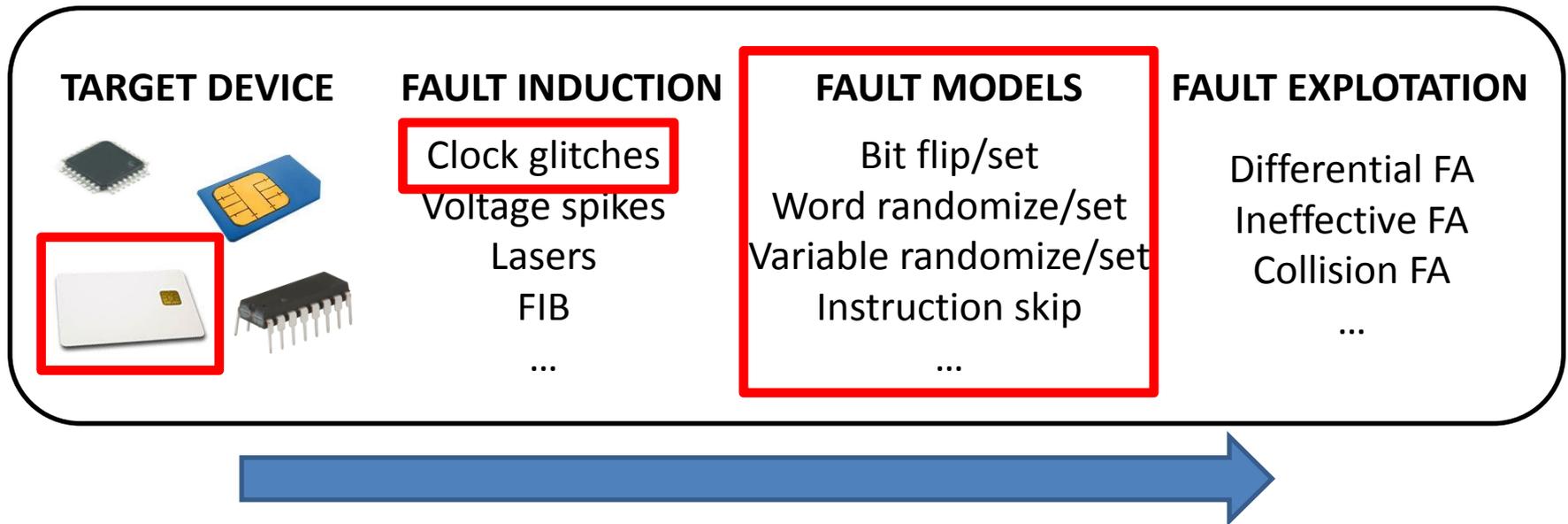
Katholieke Universiteit Leuven ESAT / COSIC

Workshop on Fault Diagnosis and Tolerance in  
Cryptography – FDTC 2011

Nara, Japan, 28 September 2011

# Motivation (I)

## FAULT ANALYSIS



## Our work:

- Fix target device and fault induction mechanism
- Full characterization of fault models

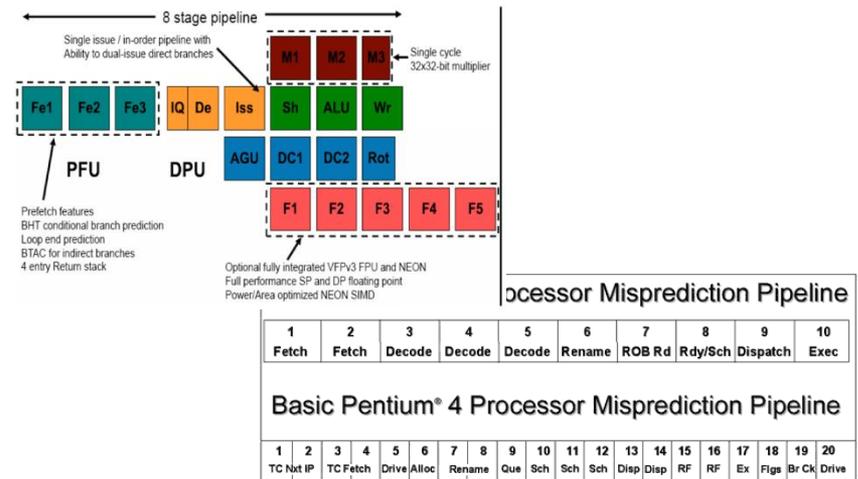
# Motivation (II)

## Fault induction mechanism

- Clock glitches
  - Non-invasive
  - Inexpensive

## Glitch effect vs. Target device

- Pipeline architecture



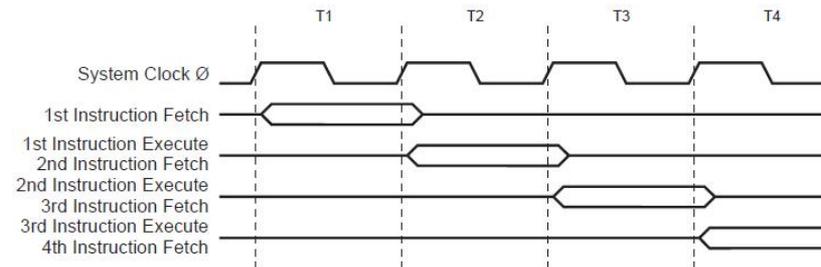
## Fault characterization vs. Experimental Platform

- Reproducibility
- Accuracy
- ...

# Target Device

## Target device

- Atmel ATmega163
  - 8-bit microcontroller
  - Harvard architecture
  - Externally clocked
  - RISC architecture
  - 2-stage pipeline



- Experiments on **five** cards

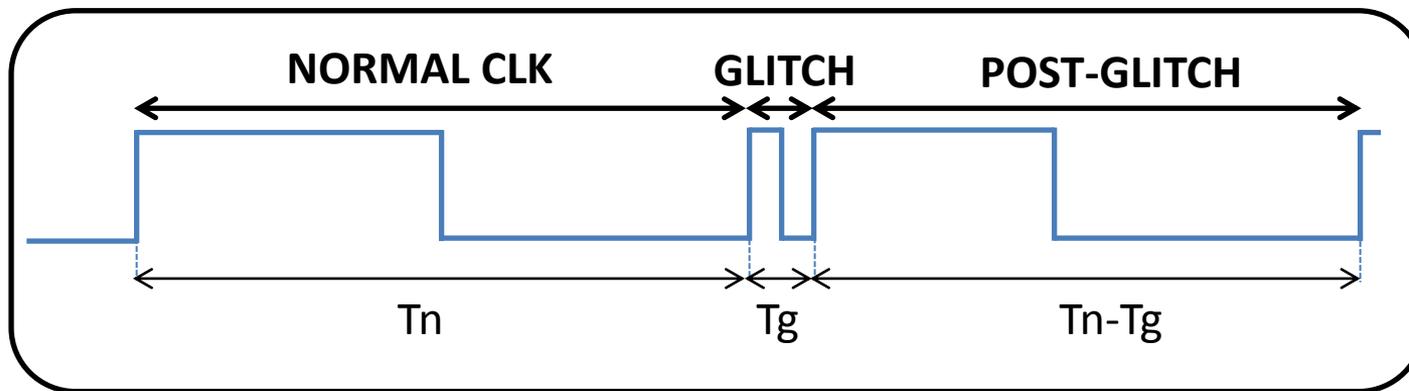
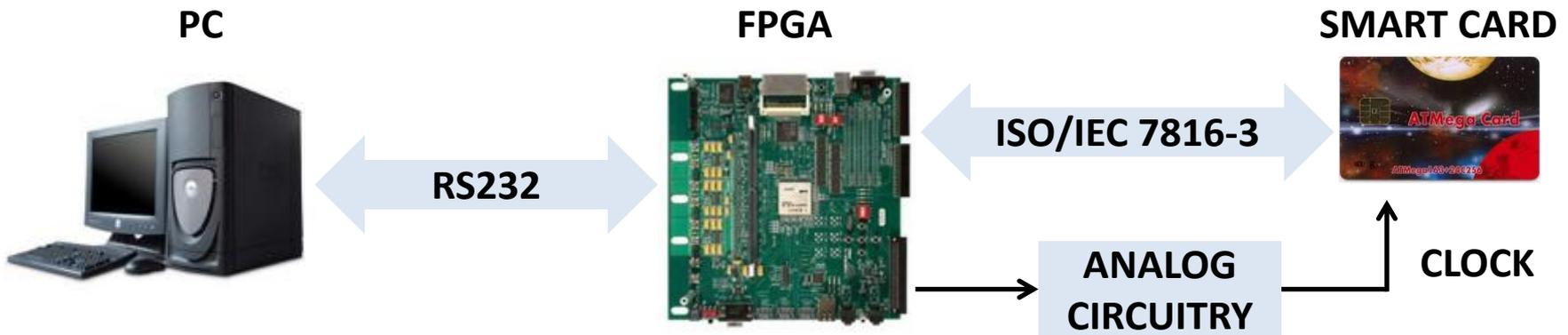


### DISCLAIMER

Legacy smart card product

Not specifically designed for security applications  
[refer to Atmel ATxxSC family]

# Experimental Setup



## DESIGN CRITERIA:

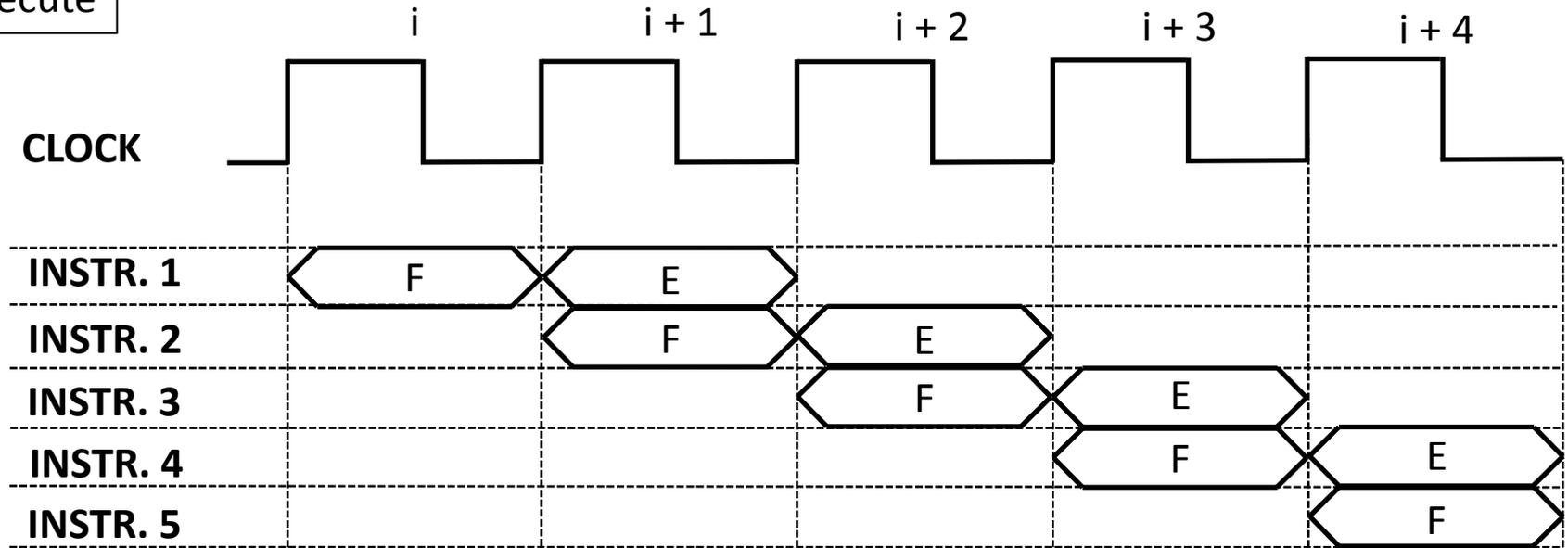
- Reproducibility
- Flexibility
- Automatization

For the rest of this presentation assume one glitch per execution:

- Upper bound:  $T_g = 15 \text{ ns}$  (65 MHz)
- Accuracy: Steps of 1 ns

# Methodology (I)

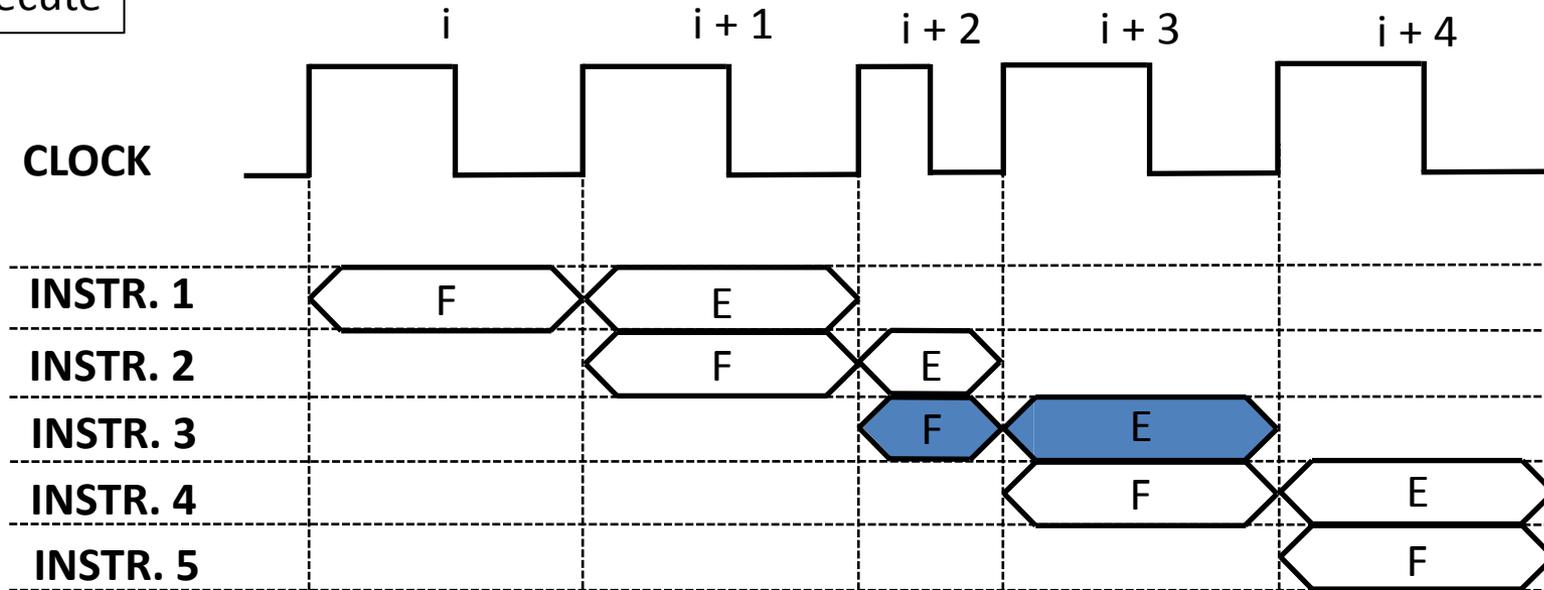
F: fetch  
E: execute



Inject glitch in arbitrary clock cycle

# Methodology (I)

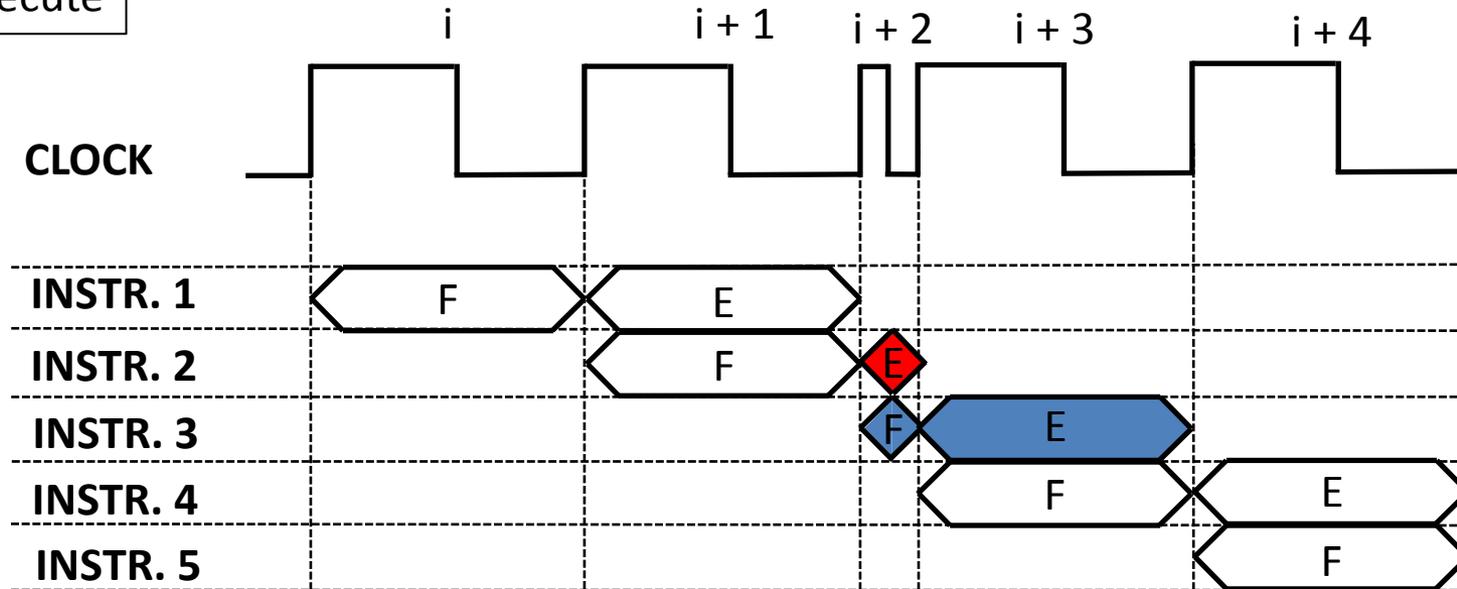
F: fetch  
E: execute



Effects on **Program Flow** (i.e. fetching phase of pipeline)

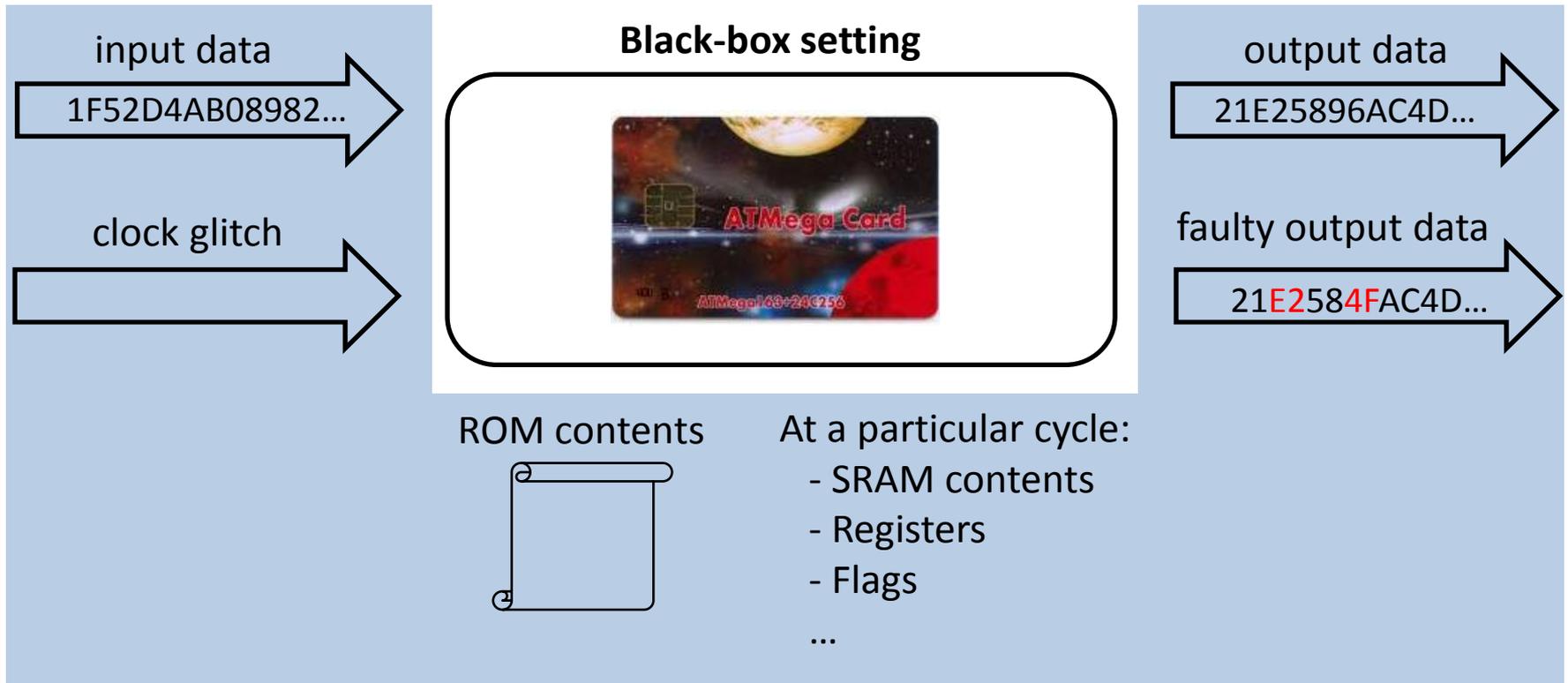
# Methodology (I)

F: fetch  
E: execute



Effects on **Data Flow** (i.e. execution phase of pipeline)

# Methodology (II)



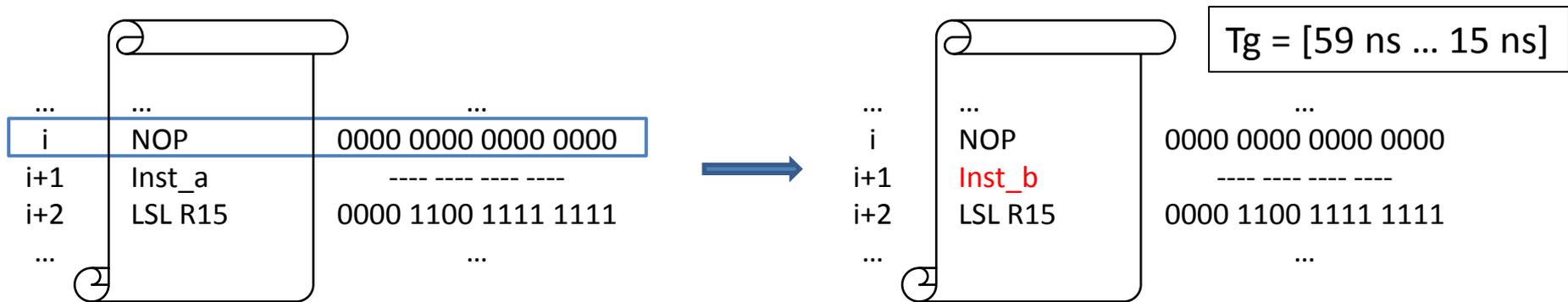
## REVERSE-ENGINEER FAULT EFFECT

Examples to illustrate the observed effects

# Effects on Program Flow (I)

## First experiments

- Commands that do not affect the data flow



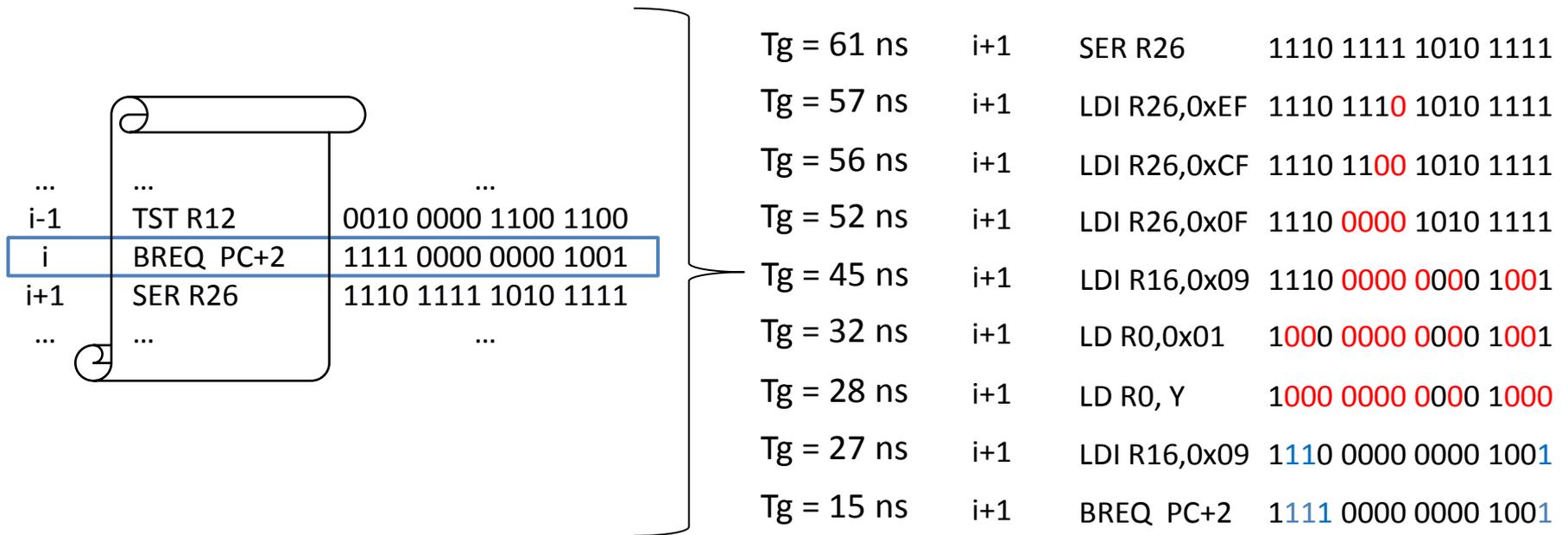
## Observations:

- Fault model: **replacing** instructions
- Program Counter (PC) not affected by fault

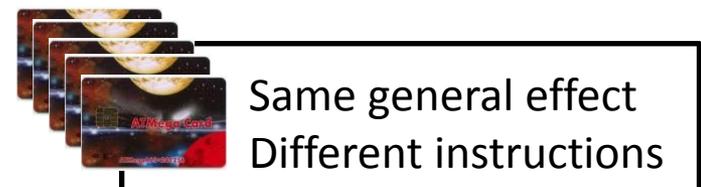


# Effects on Program Flow (II)

## Branching commands: Manipulate Program Counter



- Transition: new opcode → old opcode
- Single fault can affect both program flow **and** data flow



# Effects on Program Flow (III)

## Rest of instruction set

i	SUB R7,R5	0001 1000 0111 0101
i+1	ADD R8, R4	0000 1100 1000 0100
...	...	...
i+k	MOV R27, R6	0010 1101 1011 0110
...	...	...



Glitch widths:  
[57 ns ... 28 ns]

## Observations:

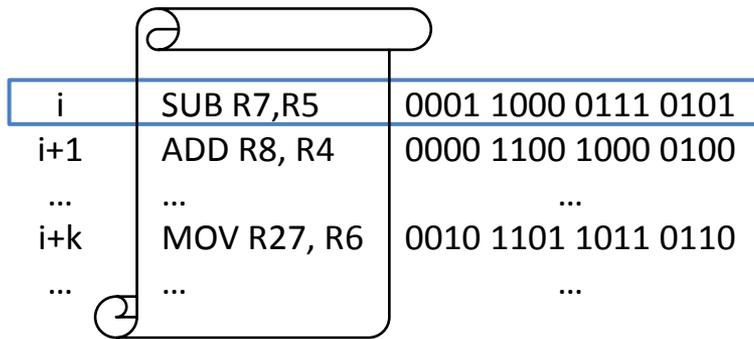
- “Effective” skips ~ 50% of time
- Replaced by commands that do not affect data flow (NOPs, illegal commands, ...)



Same effect

# Effects on Data Flow (I)

## Single-cycle instructions



i	SUB R7,R5	0001 1000 0111 0101
i+1	ADD R8, R4	0000 1100 1000 0100
...	...	...
i+k	MOV R27, R6	0010 1101 1011 0110
...	...	...



Glitch widths: [27 ns ... 15 ns]

Appearance of errors in data flow

Dependant on the specific instruction,  
registers used, and glitch width

## Observations:

- Fault effect: **deterministic** values (not random!)
- Characterization is too complex...

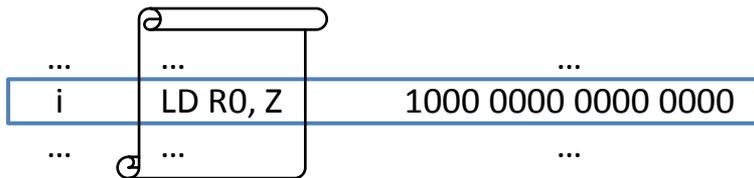


Same general effect  
Different errors!

# Effects on Data Flow (II)

## Multi-cycle instructions

- LD (Load From Data Space)



2-cycle instruction:

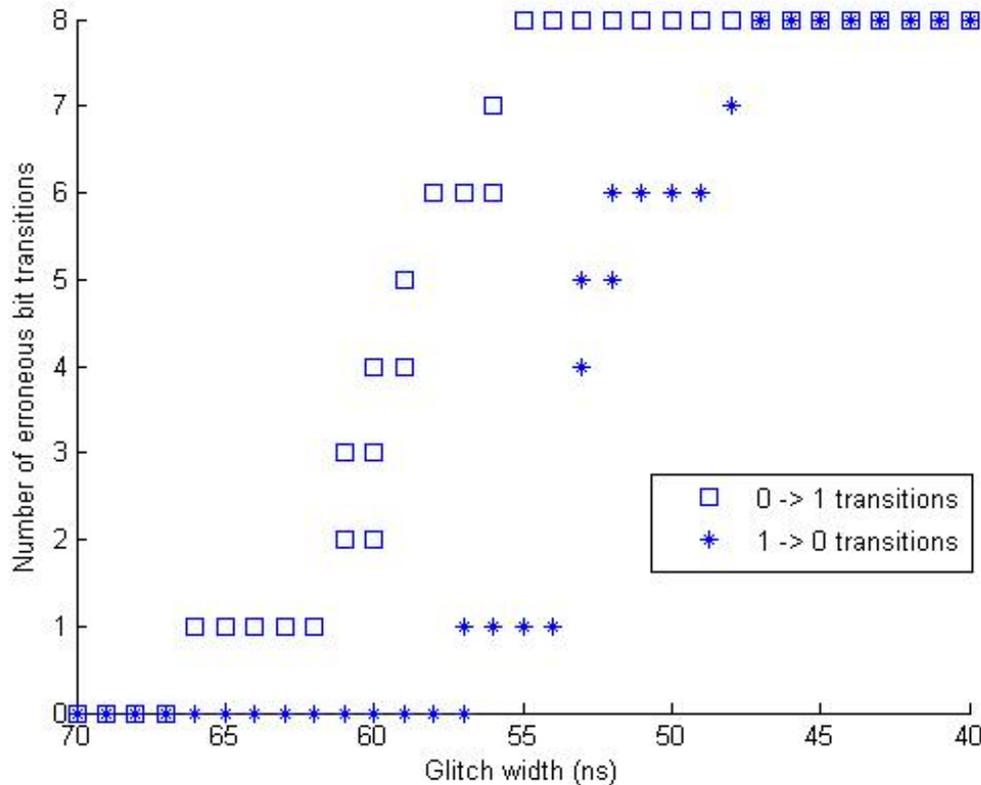
- 1st cycle: load address from Z
- 2nd cycle: load value from SRAM

## Observations

- 1st cycle: value from erroneous SRAM address is loaded
  - Different depending on glitch width
- 2nd cycle: prevent bit transitions in data bus

# Effects on Data Flow (III)

## Erroneous transitions in data bus



No errors until  $T_g = 67$  ns

Prevent 0->1 transitions in 1 bit

- Stable result (always bit 4)
- E.g. 0x00 -> 0xFF, fault 0xEF
- **Stuck-at-zero** model ?

For  $T_g \leq 49$  ns no transitions

- Data bus not updated !
- **Set word** model



Same general effect  
Different values of  $T_g$

# Summary / Conclusions

Effects of clock glitches on 2-stage pipeline MCU

Black-box study

- Limited knowledge of internal MCU hardware

Characterization of fault models

- Instructions can be **replaced**, rather than skipped
- Faults on data flow are **deterministic** and **reproducible**
- Easiest / More stable results for multi-cycle instructions
  - Stuck-at-zero bit and **set word** fault models stable

Possible to implement theoretical fault attacks

# Thanks for your attention

## QUESTIONS ?

