On the Effects of Clock and Power Supply Tampering on Two Microcontroller Platforms

Fault Diagnosis and Tolerance in Cryptography
FDTC 2014

Busan, Korea—Tuesday, September 23, 2014

Thomas Korak
thomas.korak@iaik.tugraz.at

Michael Höfler
michael.hoefler@student.tugraz.at

Outline

- Overview
- Investigated Microcontrollers
- Fault-Injection Setup
- Instruction-Set Attacks
- Conclusion

Overview

- Effects of similar faults on different pipeline architectures
  - Fetch stage
  - Execute stage

Overview

- Effects of similar faults on different pipeline architectures
  - Fetch stage
  - Execute stage
- Effects of fault injections on three different instruction groups

Overview

- Effects of similar faults on different pipeline architectures
  - Fetch stage
  - Execute stage
- Effects of fault injections on three different instruction groups
- Combination of short-time underpowering with clock glitches

Overview

- Effects of similar faults on different pipeline architectures
  - Fetch stage
  - Execute stage
- Effects of fault injections on three different instruction groups
- Combination of short-time underpowering with clock glitches
- Interval for attack parameters to thwart sample distribution

Overview

- Fault injection attacks
  - Actively affecting a device
  - Enforce faulty behavior

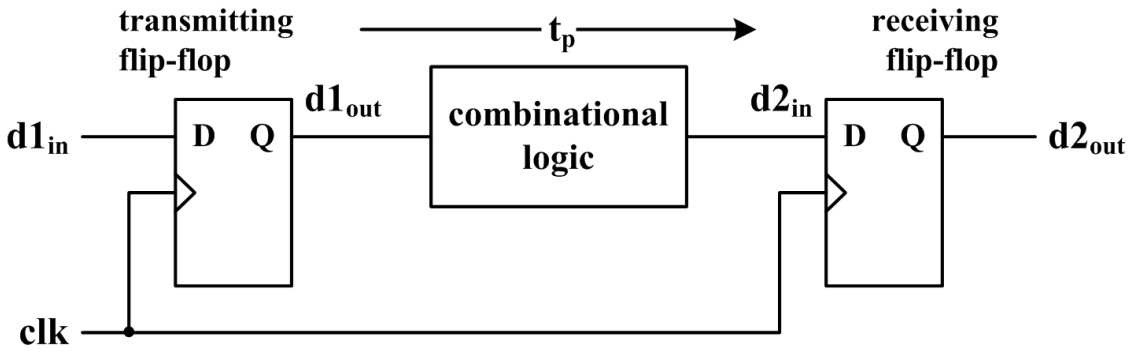
Overview

- Fault injection attacks
  - Actively affecting a device
  - Enforce faulty behavior

- Threat to cryptographic devices
  - RFID applications
  - Wireless sensing platforms
  - Mobile devices
  - Embedded Systems

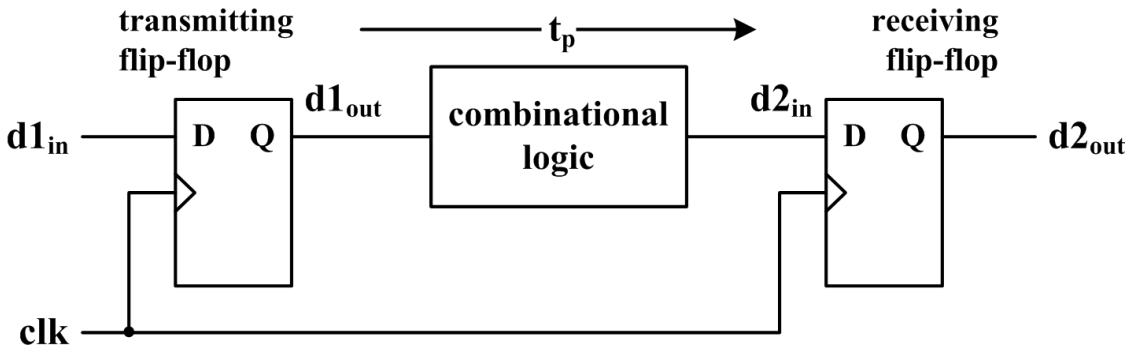
Overview

- Fault Injection Methodology
  - Timing-constraint violations
  - Clock glitches
  - Underpowering



Overview

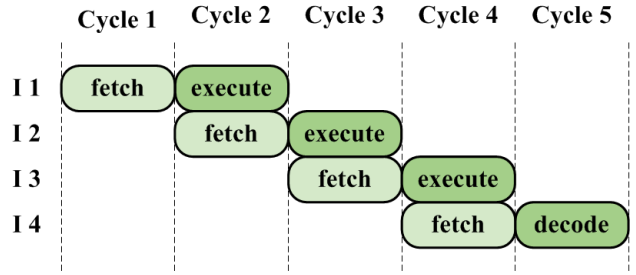
- Fault Injection Methodology
  - Timing-constraint violations
  - Clock glitches
  - Underpowering



- Attacks
  - Instruction execution procedure
  - Arithmetical, branch and memory instructions

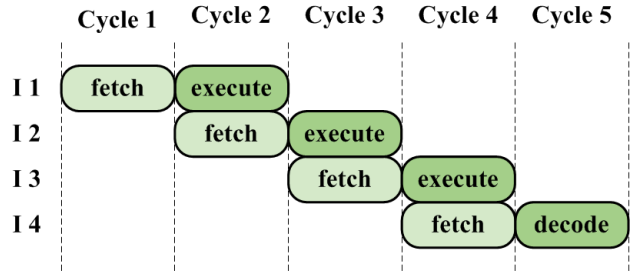
Investigated Microcontrollers

- Atmel ATxmega256
  - 8-bit microcontroller
  - 16-bit instructions (RISC)
  - Harvard architecture
  - Two-stage pipeline
  - f_{max} = 32 MHz (31.25 ns)

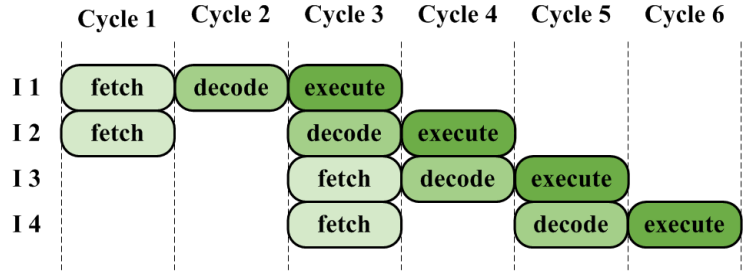


Investigated Microcontrollers

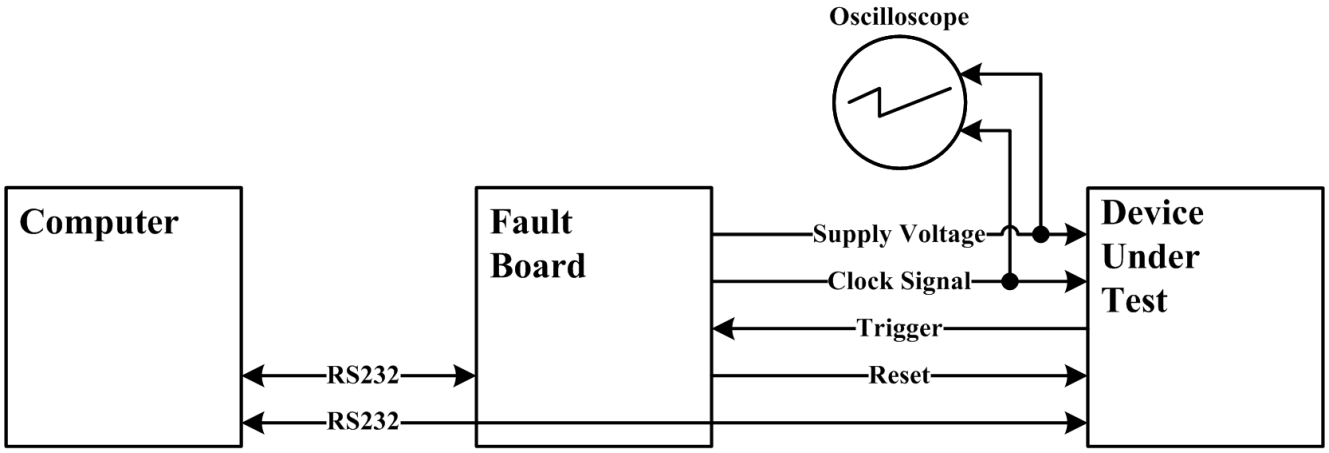
- Atmel ATxmega256
  - 8-bit microcontroller
  - 16-bit instructions (RISC)
  - Harvard architecture
  - Two-stage pipeline
  - f_{max} = 32 MHz (31.25 ns)



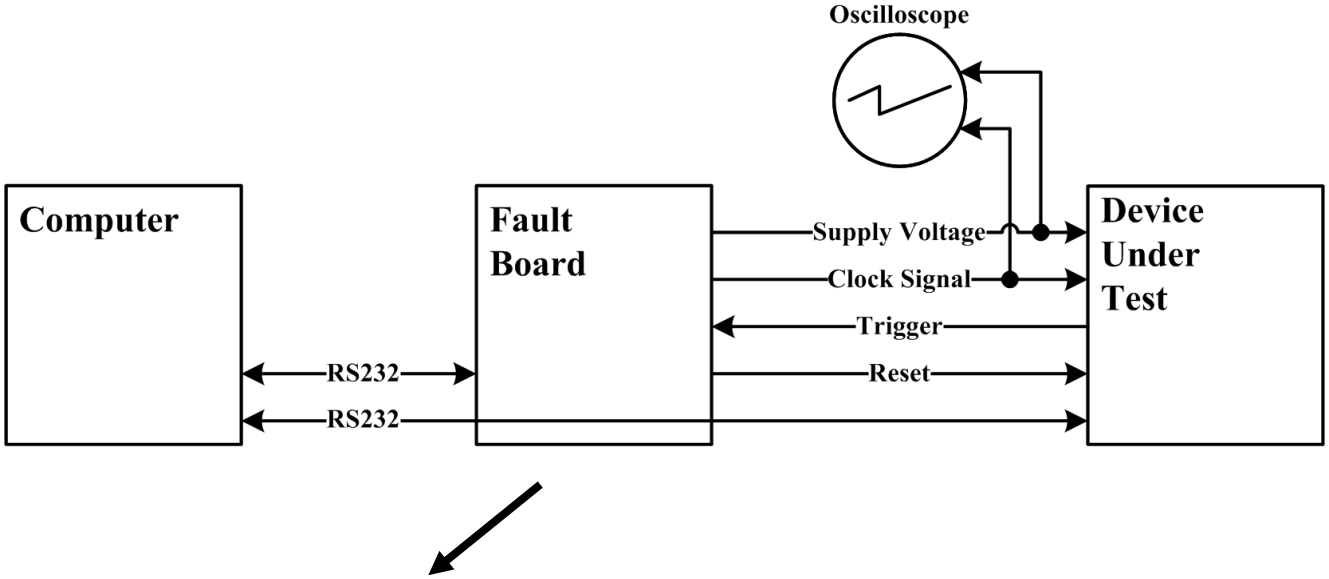
- NXP LPC1114 (Cortex-M0)
  - 32-bit microcontroller
  - 16/32-bit instructions (RISC)
  - Von-Neumann architecture
  - Three-stage pipeline
  - f_{max} = 50 MHz (20 ns)



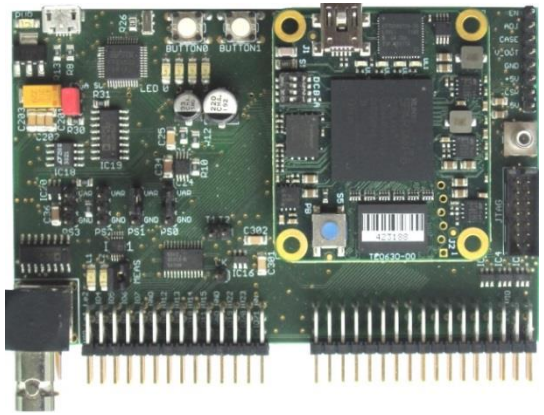
Fault Injection Setup



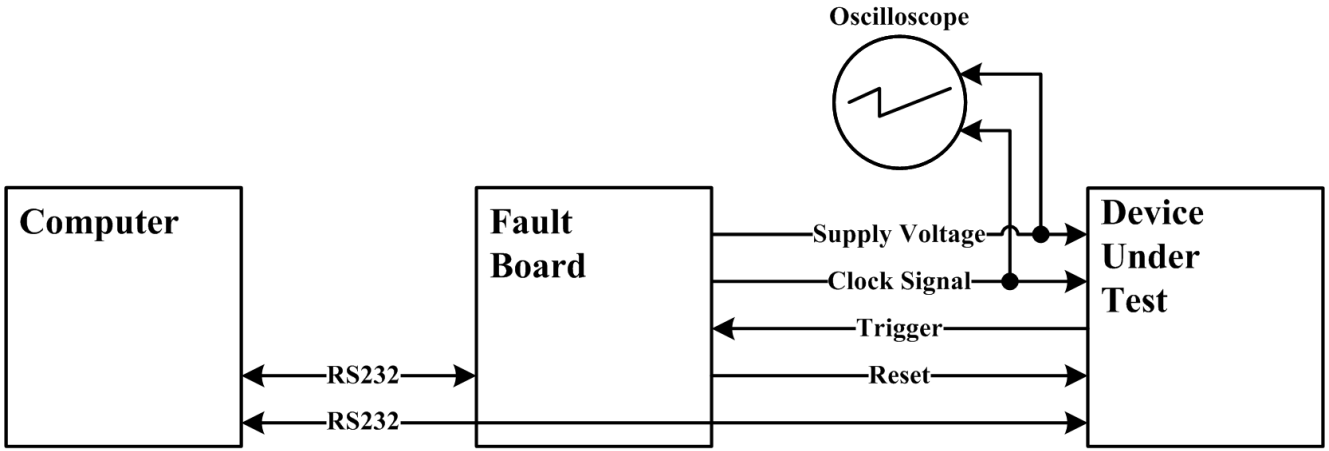
Fault Injection Setup



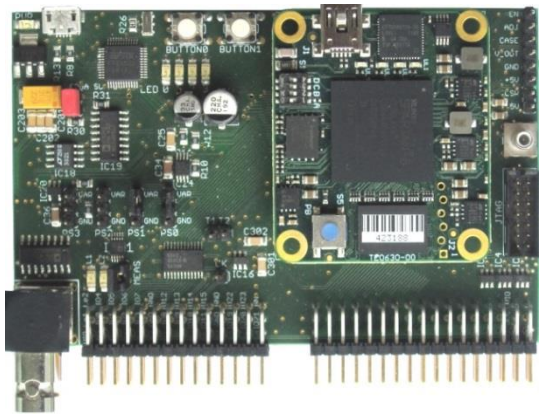
Fault Board



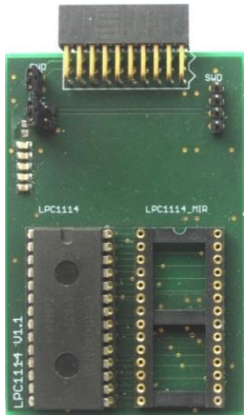
Fault Injection Setup



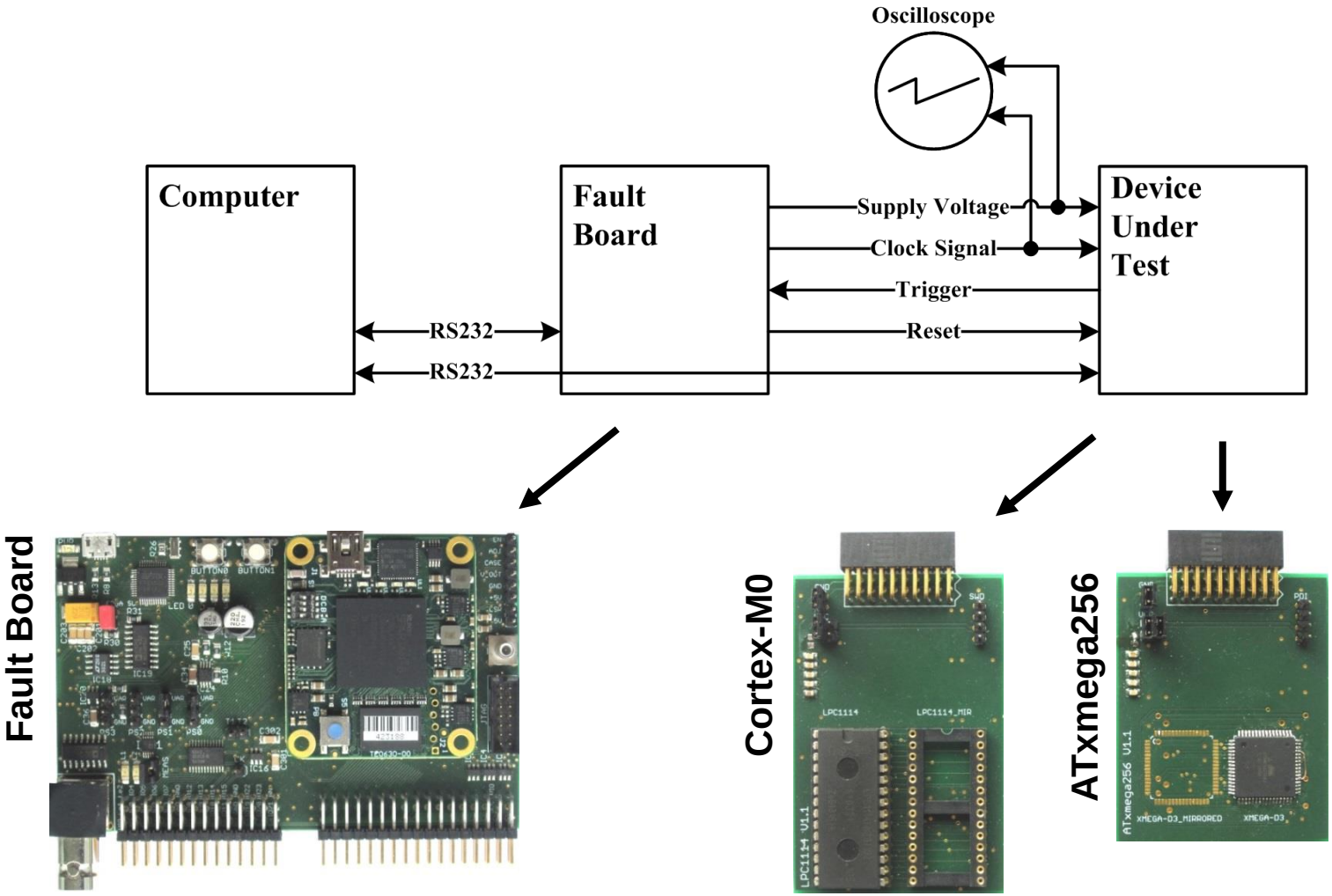
Fault Board



Cortex-M0

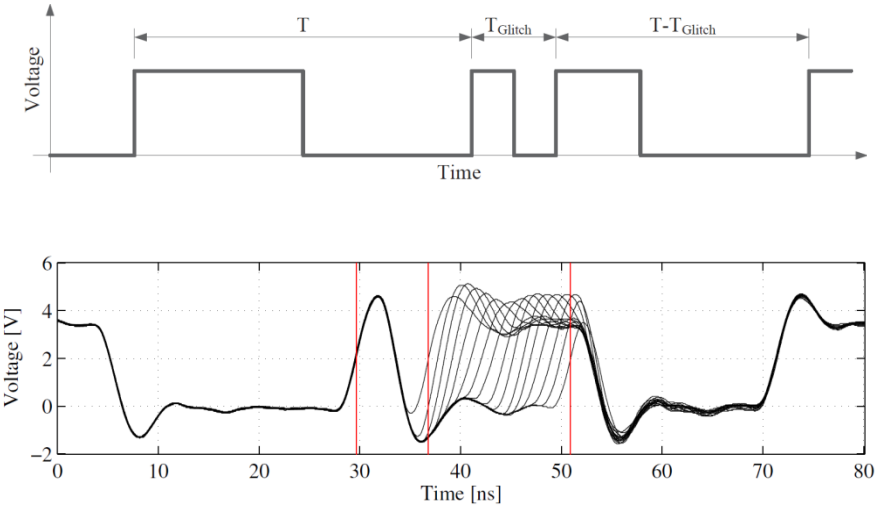


Fault Injection Setup



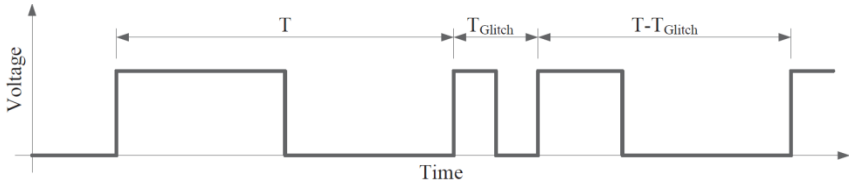
Attack Parameters

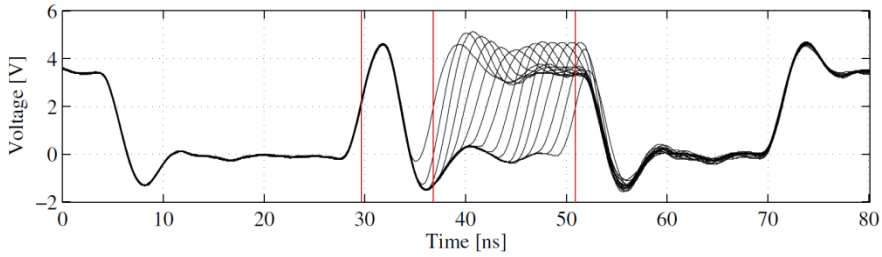
- Clock glitch period T_{Glitch}
  - 24 MHz nominal clock frequency (T ≈ 42 ns)
  - Clock glitch period T_{Glitch} between 5 and 18 ns



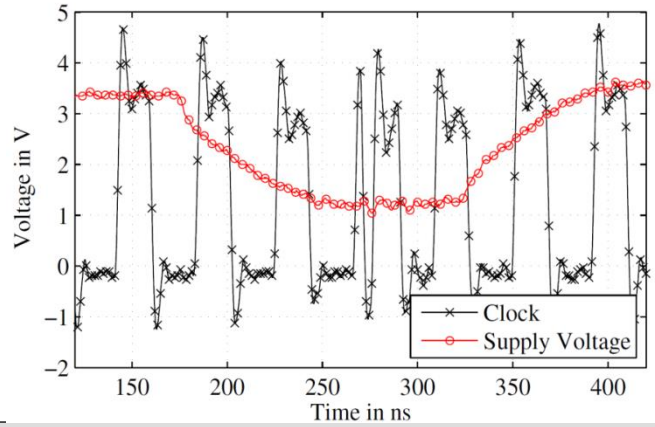
Attack Parameters

- Clock glitch period T_{Glitch}
  - 24 MHz nominal clock frequency (T ≈ 42 ns)
  - Clock glitch period T_{Glitch} between 5 and 18 ns



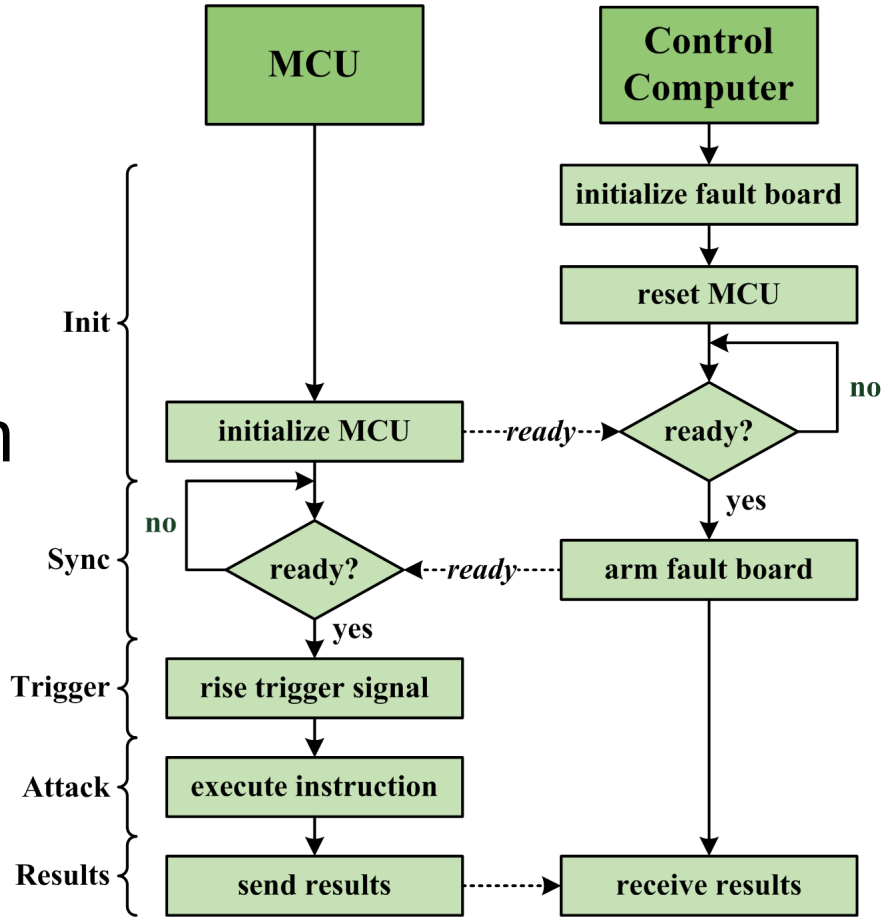


- Underpowering voltage (Cortex-M0)
  - 3.3 V nominal supply voltage
  - Underpowering voltage U_{Glitch} of 1.2 V



Instruction Set Attacks

- Single clock glitch
  - Fetch stage
  - Decode stage
  - Execute stage
- Investigated instruction
  - Inline assembly
  - Surrounded by `nop` instruction

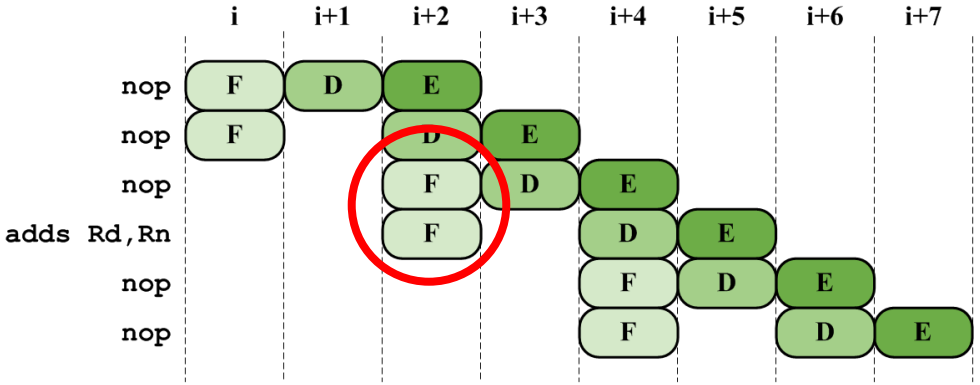


Investigated Instructions

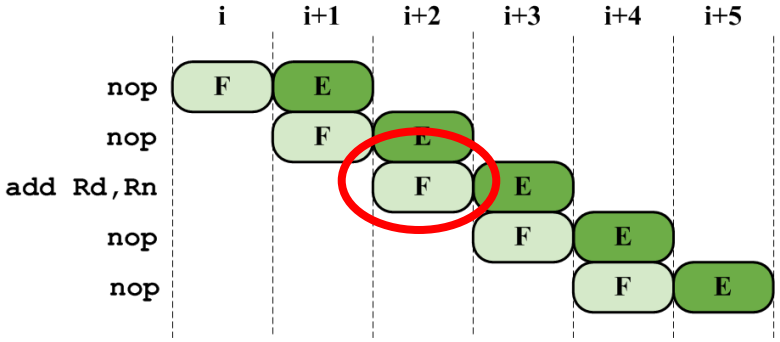
Instruction Class	ATxmega256	Cortex-M0
Arithmetical	`add Rd,Rn`	`adds Rd,Rn`
	`mul Rd,Rn`	`muls Rd,Rn`
		`lsls Rd,#imm`
Memory	`ld Rd,X`	`ldr Rd,[Rn]`
	`st X,Rn`	`str Rd,[Rn]`
Branch	`breq label`	`beq label`

Results

Cortex-M0



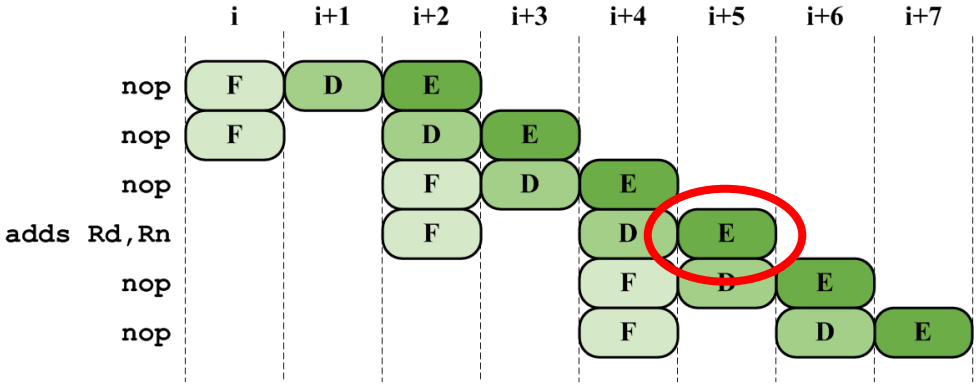
ATxmega256



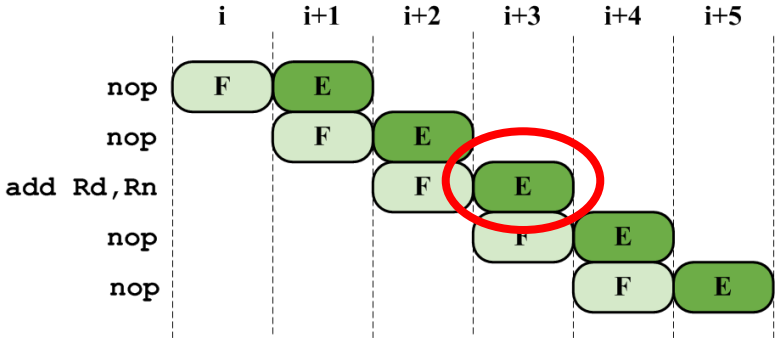
- Fetch stage
  - Fetch buffer not updated
  - Instruction not executed
  - Instructions executed twice
  - Program flow modification

Results

Cortex-M0



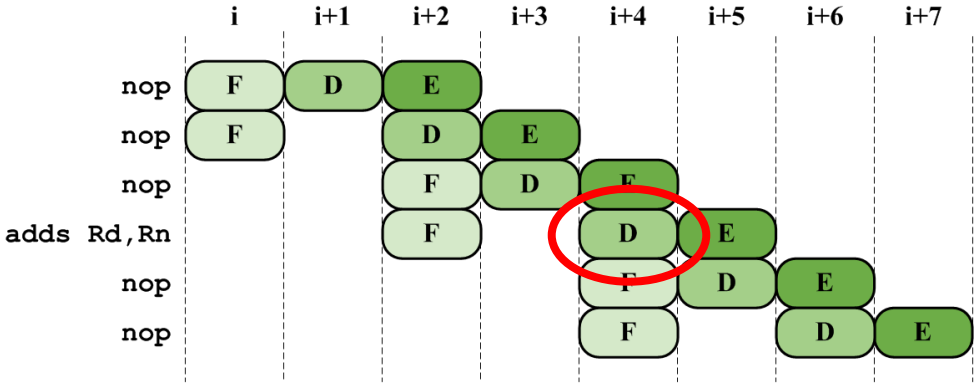
ATxmega256



- Execute stage
  - Wrong results
  - Constant values
  - Varying values (T_{Glitch})
  - Data flow modification

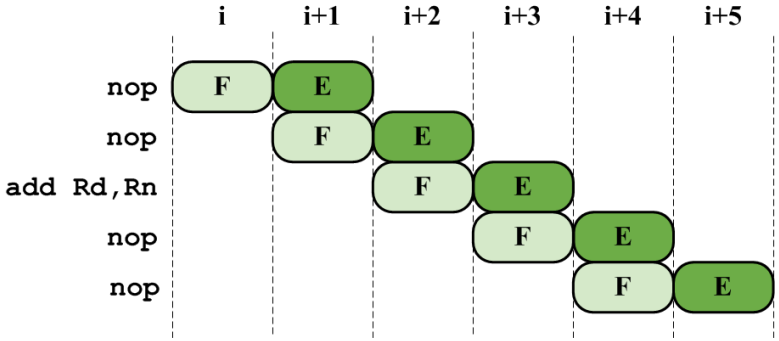
Results

Cortex-M0



- Decode stage not affected

ATxmega256



Results: Arithmetical Instructions

Cortex-M0

`adds, muls, lsls`

- Fetch stage
  - Buffer not updated
- Execute stage
  - Wrong results
    → `adds, muls`
  - Result set to zero
    → `lsls`

ATxmega256

`add, mul`

- Fetch stage
  - Buffer not updated
- Execute stage
  - Wrong results

Results: Memory Instructions

Cortex-M0

`ldr, str`

- Fetch stage
  - `Rd` set to zero → `ldr`
  - Memory set to zero → `str`
- Execute stage
  - Not executed
  - Address in `Rd` → `ldr`
  - Address in Memory → `str`

ATxmega256

`ld, st`

- Fetch stage
  - Buffer not updated
- Execute stage
  - Wrong results in `Rd` → `ld`
  - `Rd` set to zero → `ld`
  - Wrong results in memory → `str`

Results: Branch Instructions

Cortex-M0

`beq`

- Fetch stage
  - Buffer not updated
- Execute stage
  - No effects

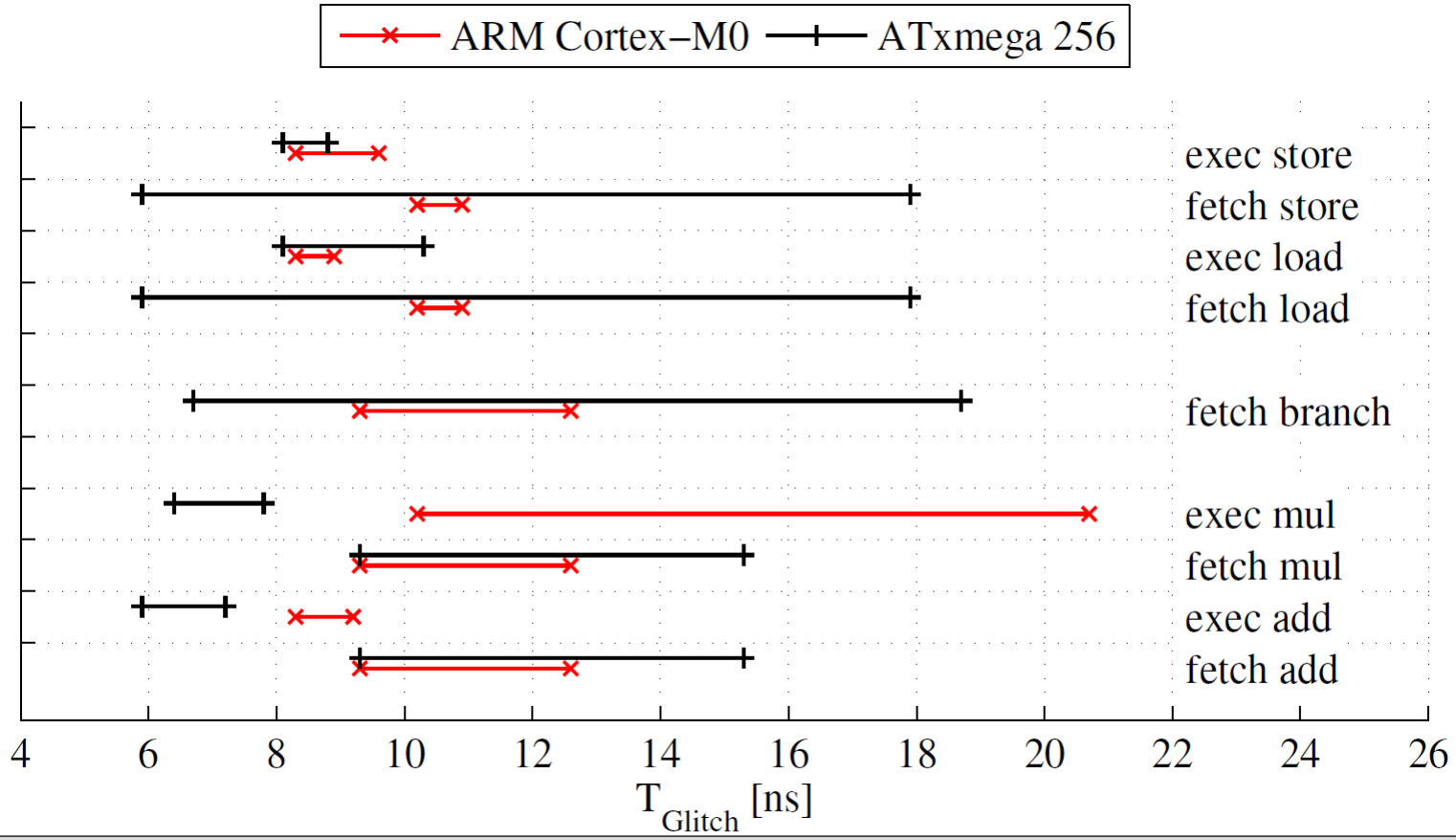
ATxmega256

`breq`

- Fetch stage
  - Buffer not updated
- Execute stage
  - No effects

Results

- Summary



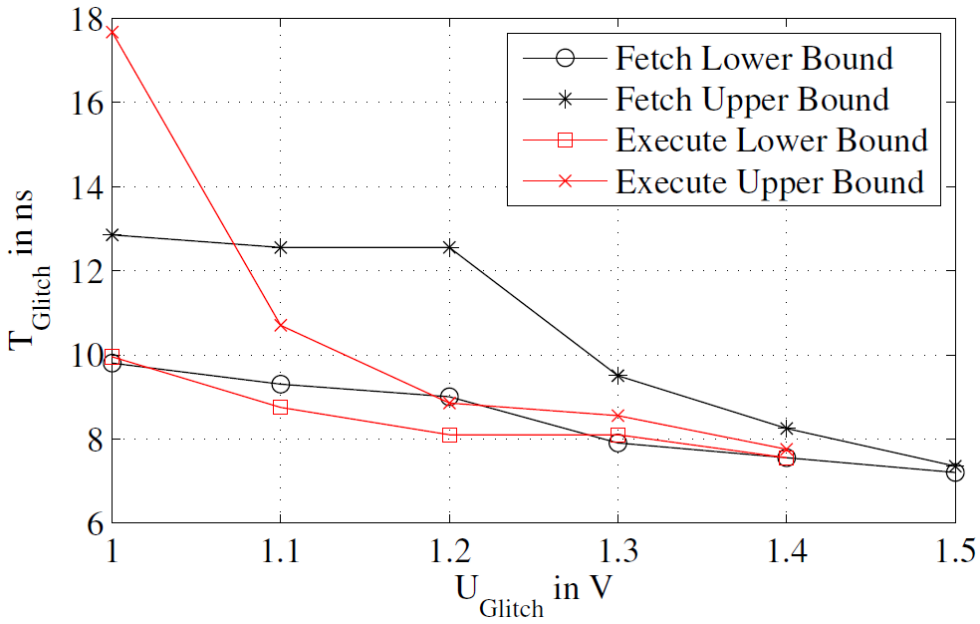
Results

- Reproducible Faults
- Interval of T_{Glitch}: [6.0, 20.0] ns

Results

- Reproducible Faults
- Interval of T_{Glitch}: [6.0, 20.0] ns

- Underpowering (Cortex-M0)
  - Increased sensitivity
  - Separate effects of fetch and execute stage
  - Detection: Brown-out detection



Conclusion

- Reliable and constant fault injection on both microcontrollers possible
  - Fetch Stage
  - Execute Stage
- Instruction dependent effects
- Increase efficiency by combining clock glitches with underpowering
- Basis for developing countermeasures
  - Which instructions are vulnerable
  - How can instructions be modified

Investigating the Vulnerabilities of Two Microcontroller Platforms to Fault Injection Attacks

Fault Diagnosis and Tolerance in Cryptography
FDTC 2014

Busan, Korea—Tuesday, September 23, 2014

Thomas Korak
thomas.korak@iaik.tugraz.at

Michael Höfler
michael.hoefler@student.tugraz.at