Safety ≠ Security

A SECURITY EVALUATION OF STATE OF THE ART AUTOMOTIVE MICROCONTROLLERS

Nils Wiersma

May 2, 2017

Radboud University

Riscure

What do they do?

- Security evaluation, including SCA and FI
- Security training, including SCA and FI
- Make tools for SCA and FI
- Research

Riscure

What did they do for me?

- Initial topic
- Ramiro and Albert
- Equipment and workspace

Topic

- Investigate the security of modern microcontroller units,
  used in the automotive industry,
  by means of fault injection.

Why fault injection?

- Increased security on certain fronts
- Other roads of exploitation

Why the automotive microcontrollers?

- Harsh environment
- Safety critical
- Fault tolerant
- ISO26262
- Safety mechanisms

Why the automotive microcontrollers?

- Safety mechanisms as countermeasures
- Mass production

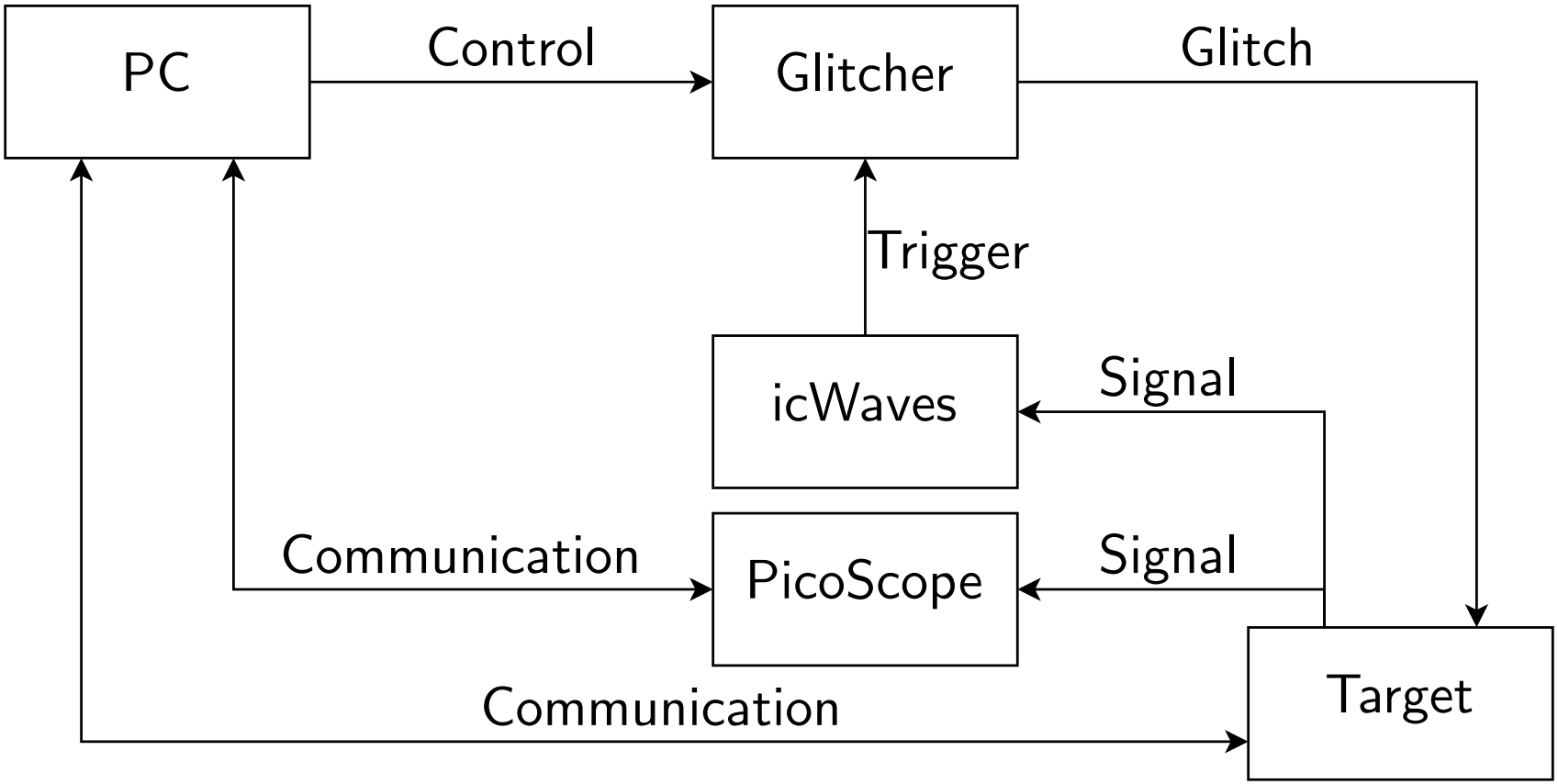
Fault injection

- Inject glitch in the environment
- Produce useful fault

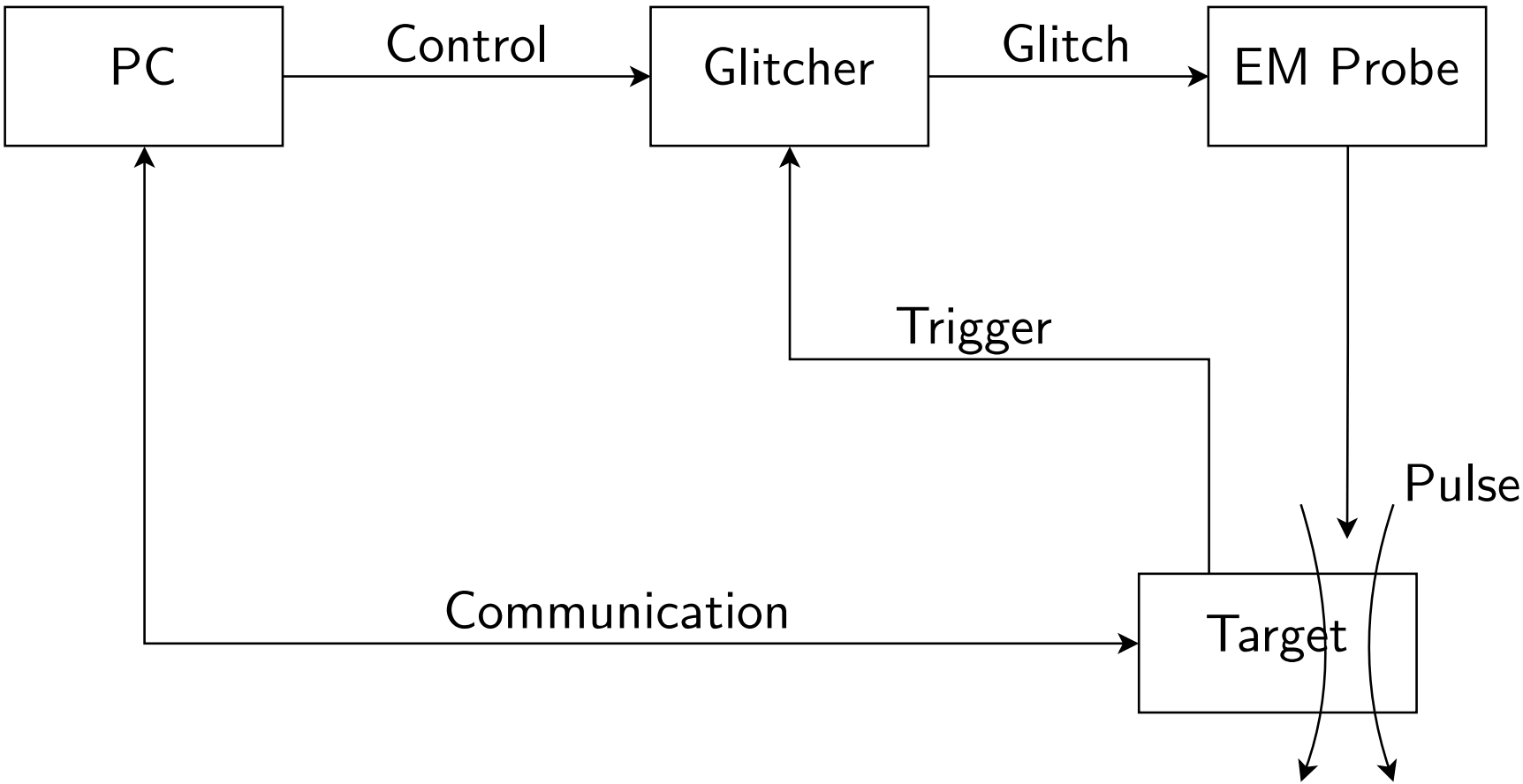
Fault injection

- Clock
- Optical
- Power
- Electromagnetic

Fault injection

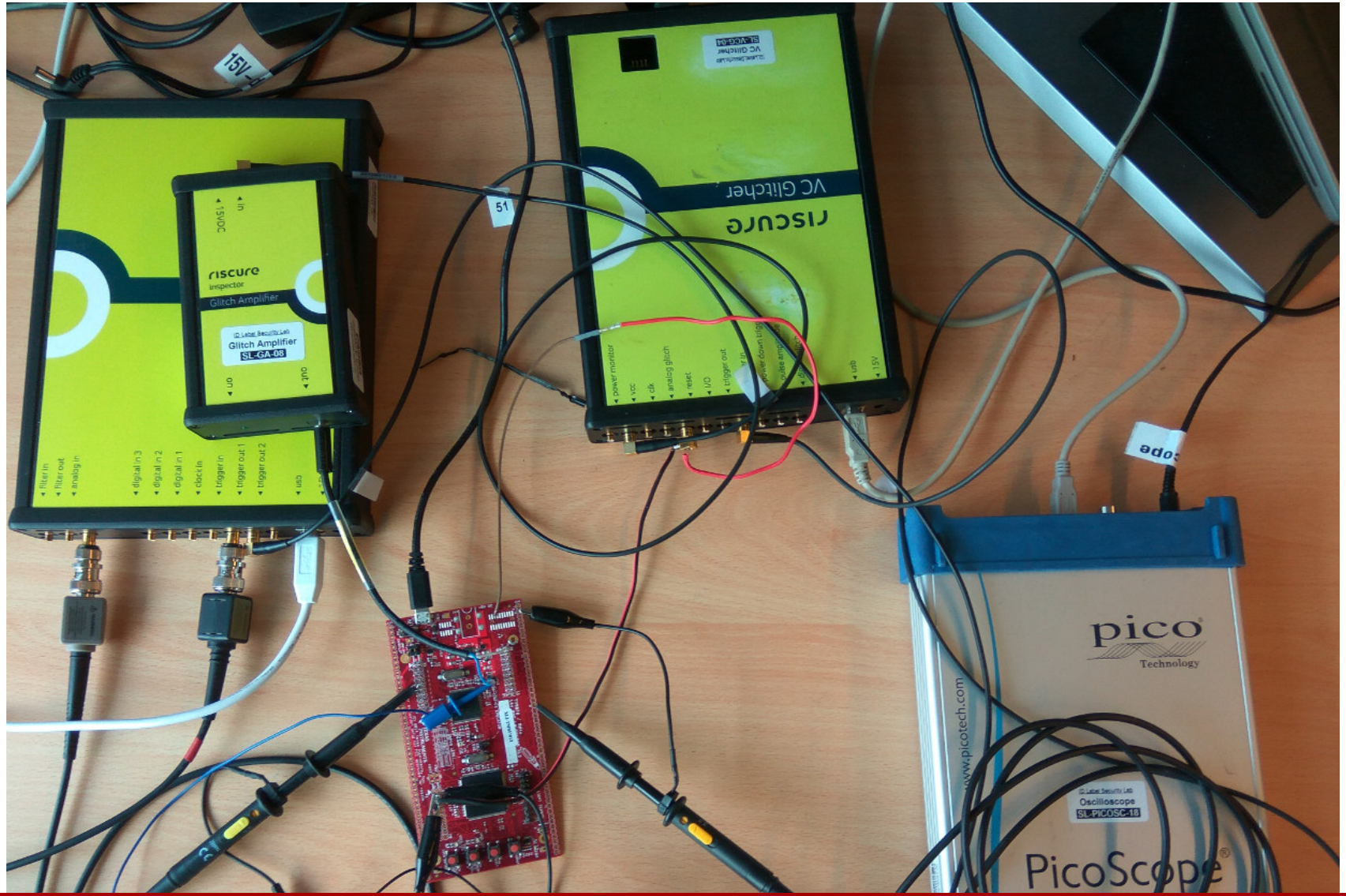


Fault injection

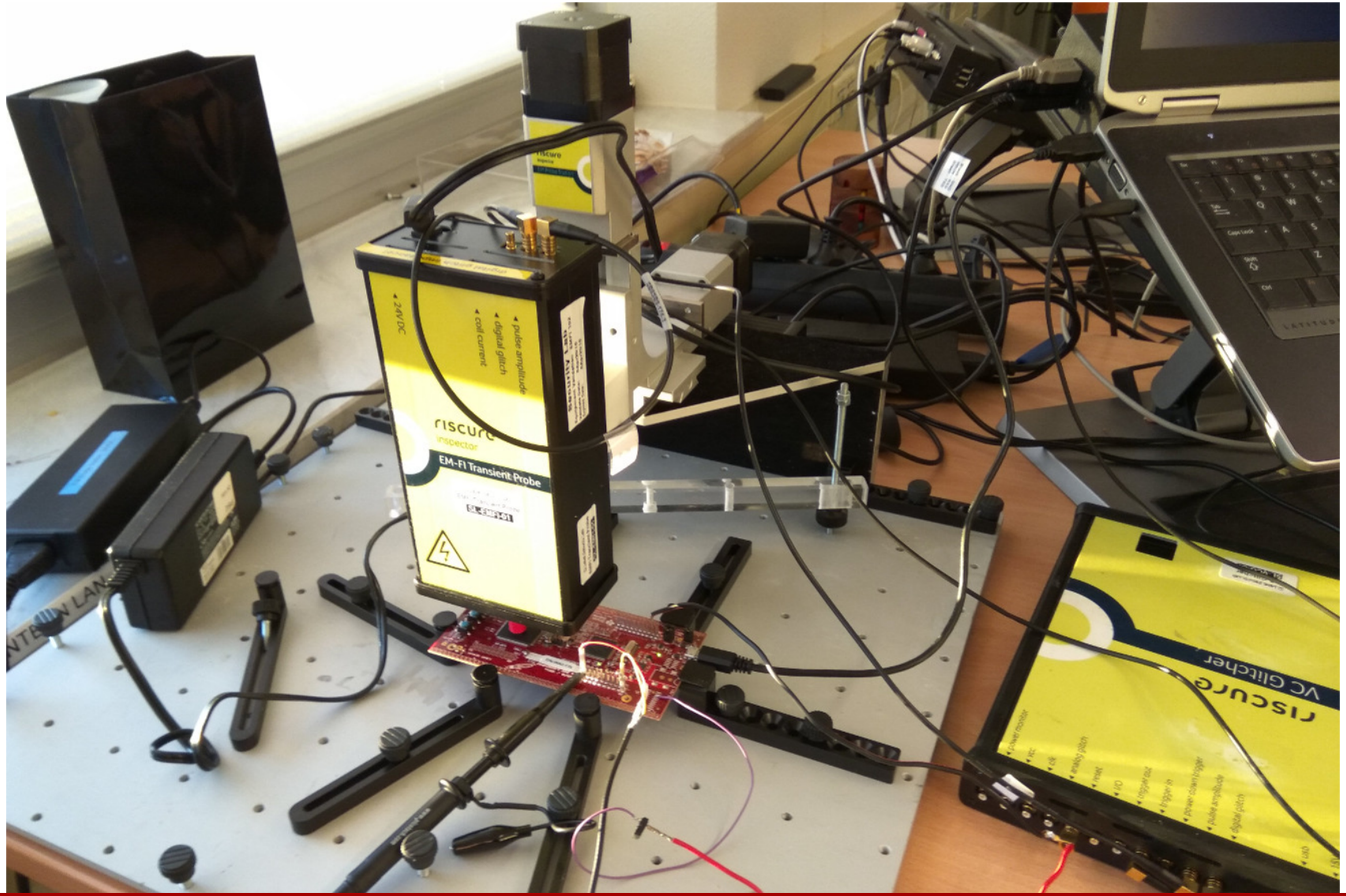


Fault injection

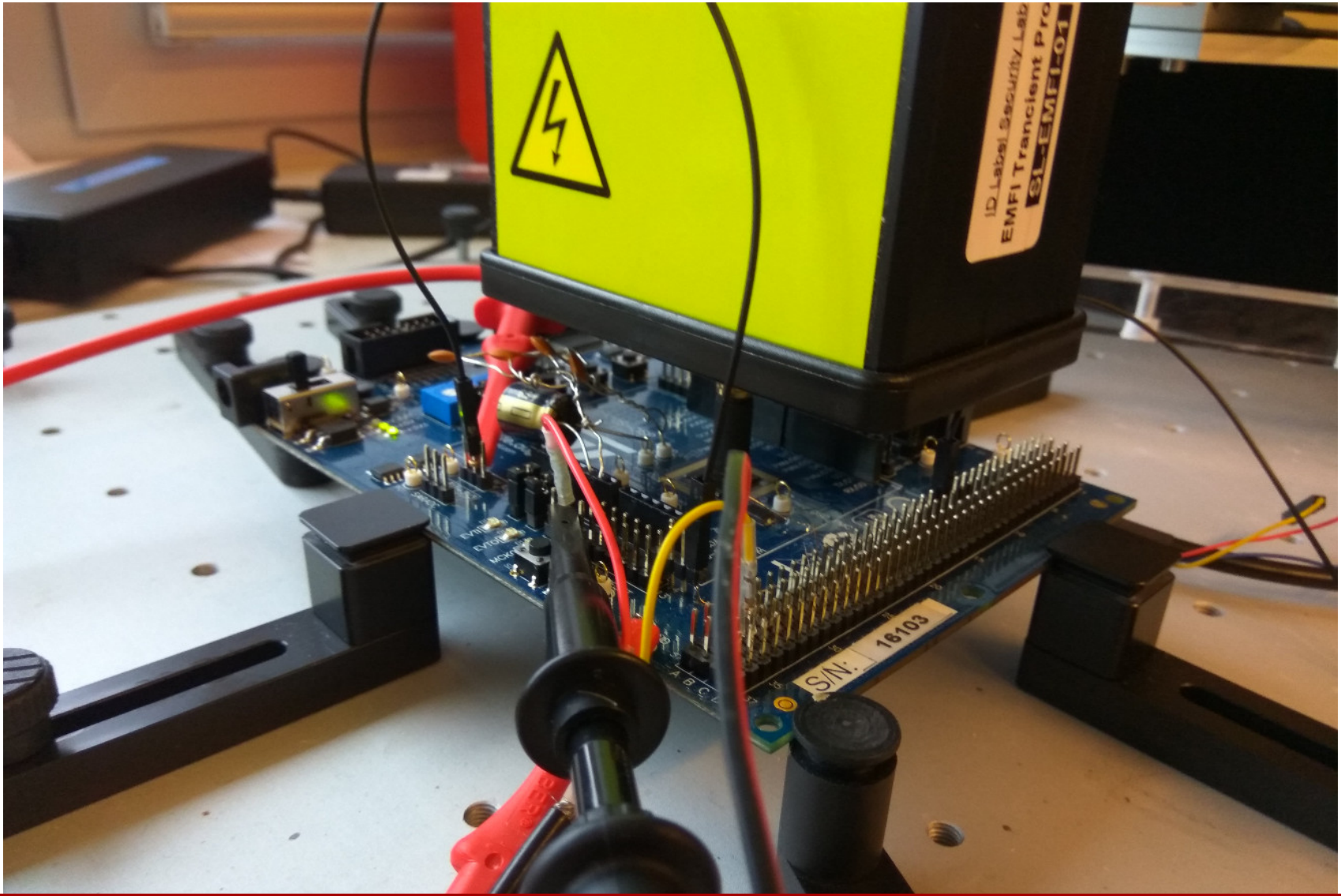
Fault injection



Fault injection



Fault injection



Fault injection

- Flip bits

Fault injection

```
int flag = 1;
if (flag == 0)
 bad();
else
 good();
```

Fault injection

0	...	Load flag
1	...	Compare flag
2	0000 1010 0000 0000 0000 0000 0000 0011	Branch to 5
3	...	bad()
4	...	Branch to 6
5	...	good()
6	...	Continue

Fault injection

0	...	Load flag
1	...	Compare flag
2	0000 10**0**0 0000 0000 0000 0000 0000 0011	Store r0,r1
3	...	bad()
4	...	Branch to 6
5	...	good()
6	...	Continue

Fault injection

```
int flag = 1;
if (flag == 0)
  bad();
else
  good();
```

Fault injection

```
int flag = 1;
if (flag == 0)
 bad();
else
 good();
```

Safety mechanisms / Countermeasures

Software

- Smart coding
- Multiple flag checks
- Code flow integrity checks

Focus on faults

Safety mechanisms / Countermeasures

Hardware

- Power input monitoring and regulating
- Electromagnetic shields
- Optical sensors

Focus on glitches

Safety mechanisms / Countermeasures

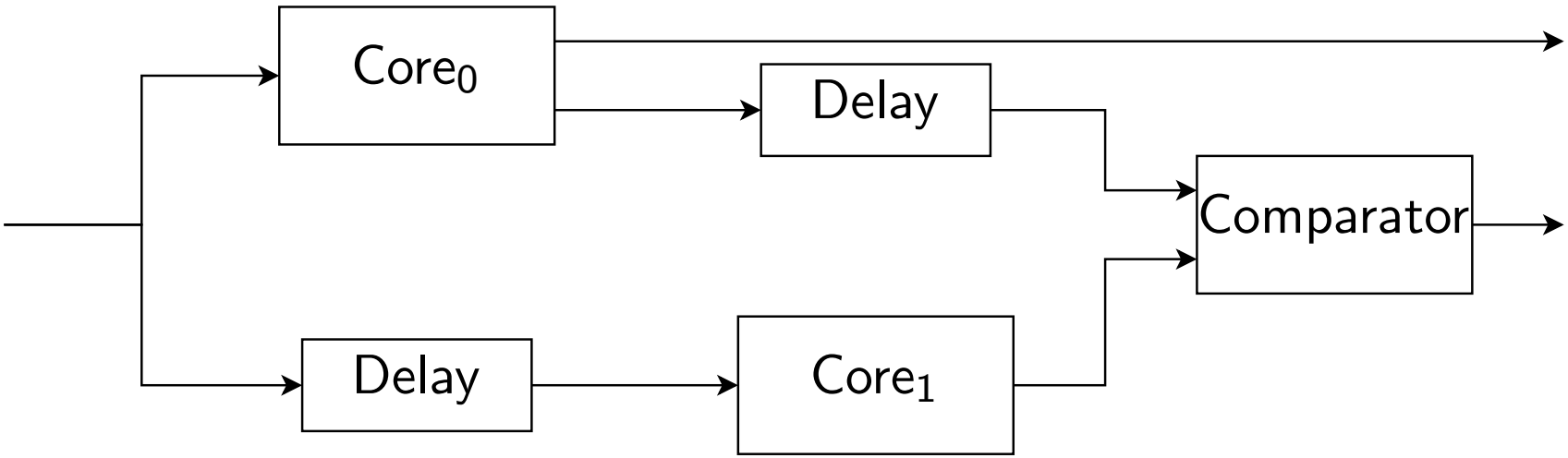
Advanced hardware

- Lockstep
- Error correction and detection codes
- Memory duplication

Focus on faults

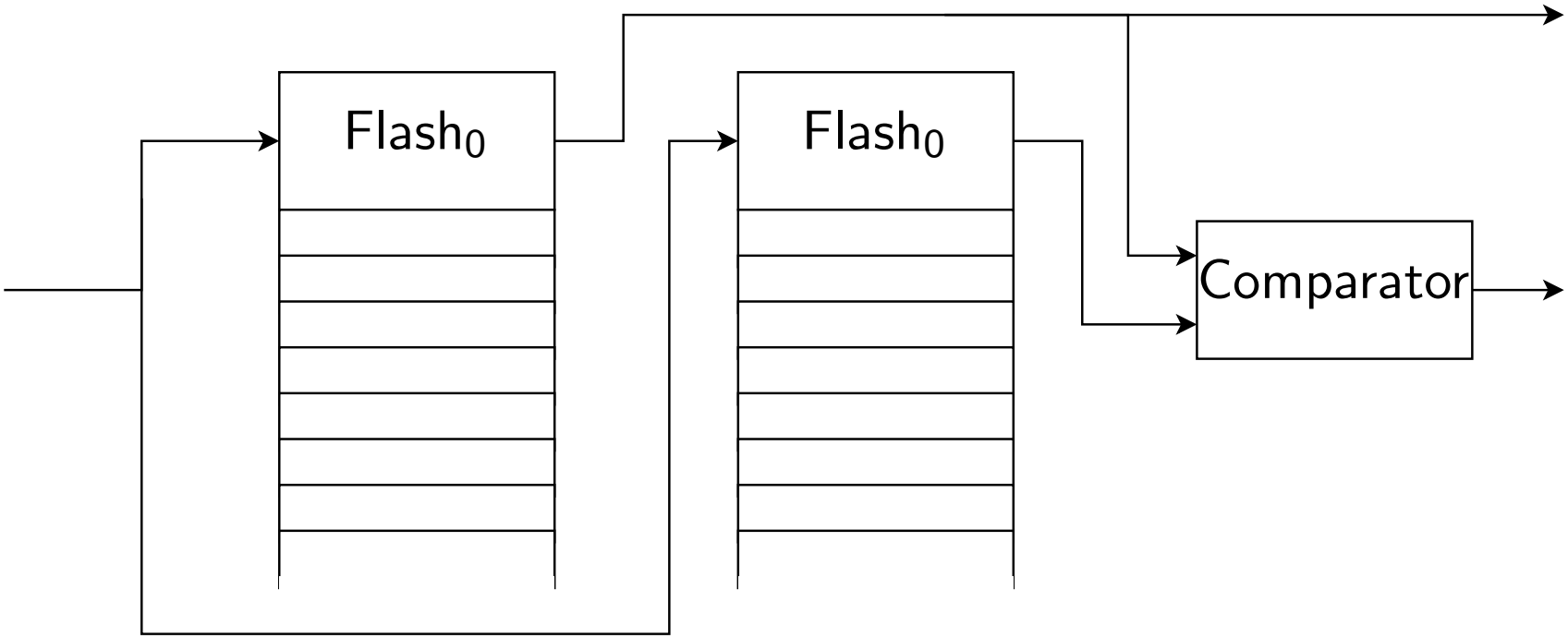
Safety mechanisms / Countermeasures

Duplicate execution and compare (lockstep)



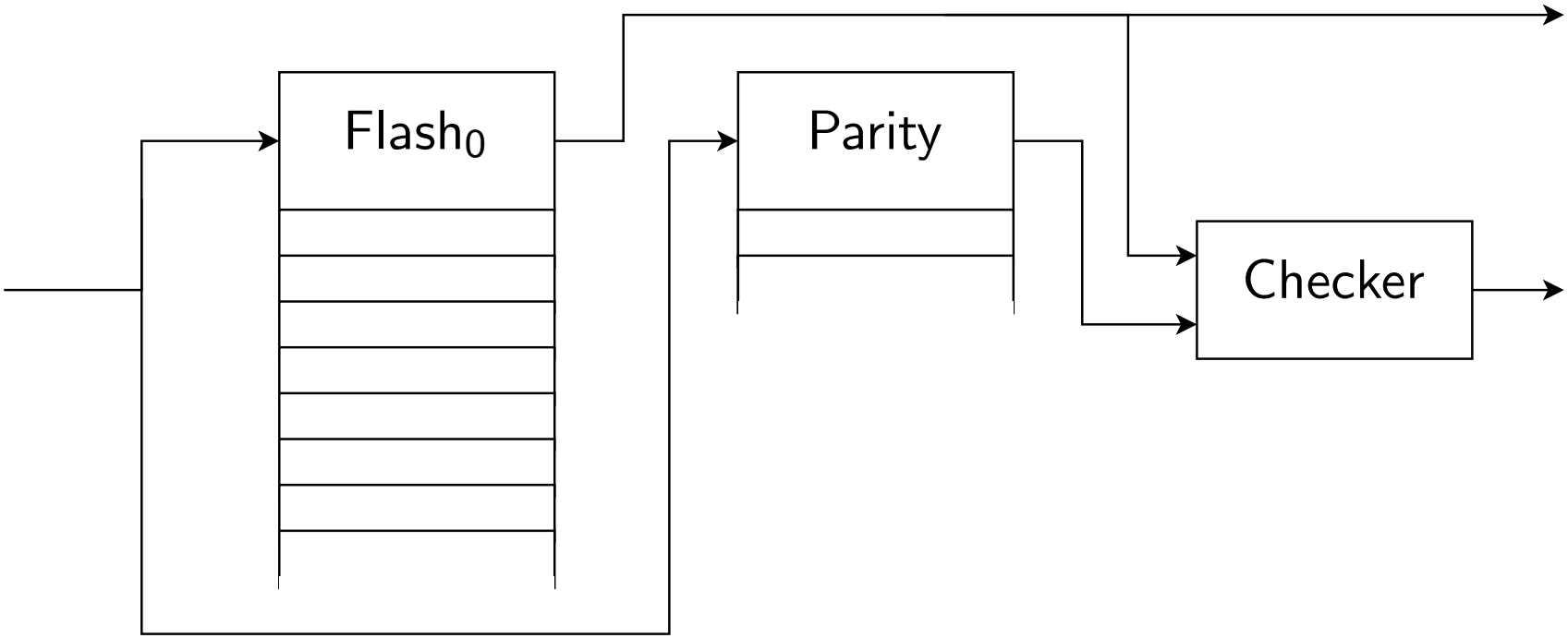
Safety mechanisms / Countermeasures

Memory duplication

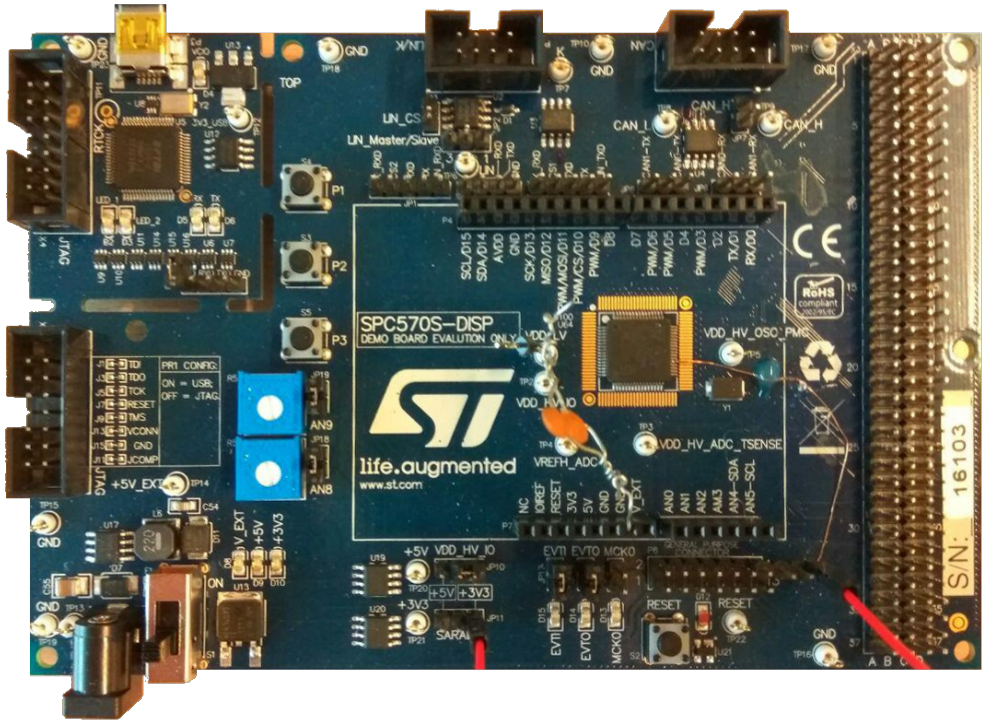


Safety mechanisms / Countermeasures

Error control codes (parity bits, cyclic redundancy codes)



Targets



- e200z0h PowerPC 32bit by STMicroelectronics

Targets

- Cortex-R4F ARM 32bit by Texas Instruments

Experiments

- Characterization of targets
- Unlocking debug interface (JTAG)

Characterize

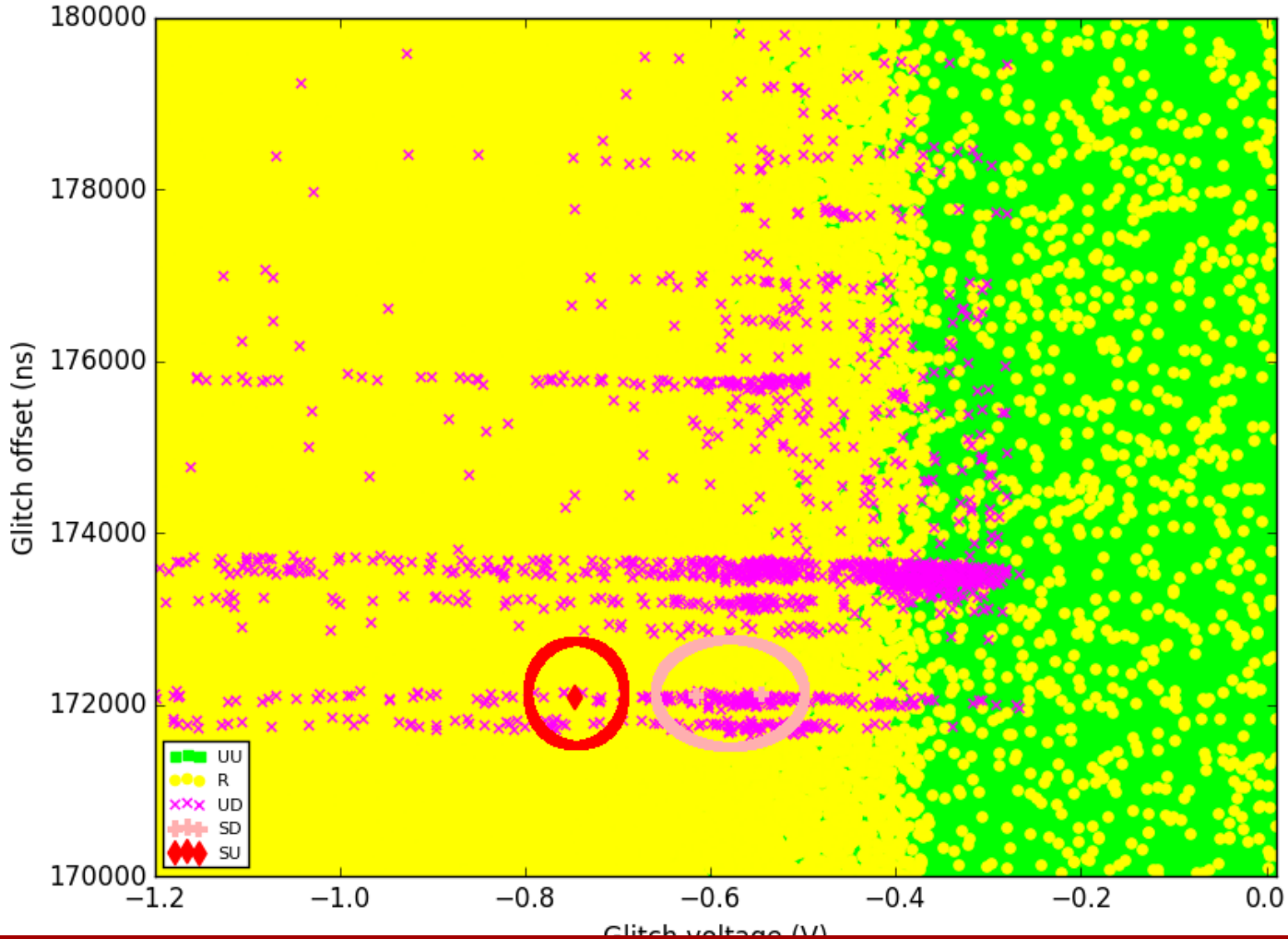
- Target firmware under complete control of the attacker
- Reduce variables to minimum
- Investigate internal register states

Characterize

1. Investigate normal behavior through (side) channels
2. Determine rough ranges of values for different parameters
3. Determine fixed values for parameters that yield good results

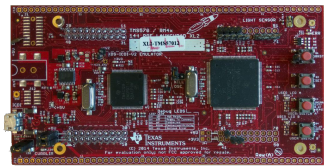
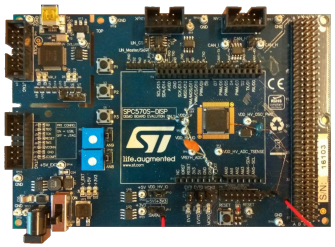
Characterize

3. Determine fixed values for parameters that yield good results



Characterize

Branch experiment results summary

Target	Successful	Detected
 TI (power)	60%	0%
 STM (EM)	58%	0%

Characterize

Comparison of triggered measures in TI

Successful	Detected	Other
0.7%	24%	75%

Characterize

Comparison of triggered measures in TI

Lockstep	RAM parity	Flash address parity
98%	21%	27%

JTAG


TMS570 / RM4x
XL2-TMS57012
XDS-ICDI-V2 EMULATOR
LIGHT SENSOR
Debug LEDS
TEXAS INSTRUMENTS
(C) 2014 Texas Instruments, Inc.
For evaluation only; not FCC approved for resale.
ICDI
CURRLIM
USER LED A
USER LED B

JTAG

- Read / write access to memory
- Find secret values, intellectual property

JTAG

- Locked, unlockable by providing password
- No knowledge of firmware
- No clear trigger signals available
- No register output to categorize

JTAG

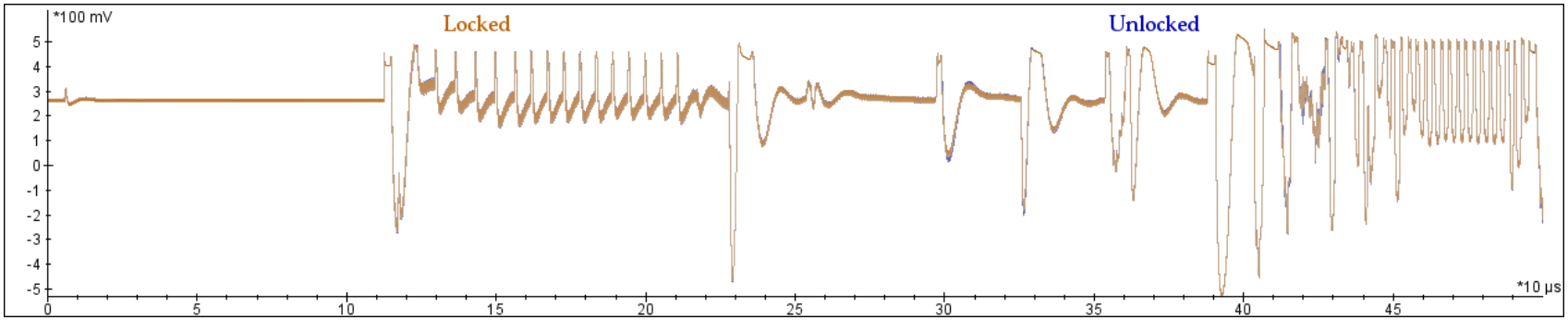
- Inject fault during boot
- Obtain JTAG password

JTAG

1. Investigate normal behavior through (side) channels
2. Determine rough ranges of values for different parameters
3. Determine fixed values for parameters that yield good results

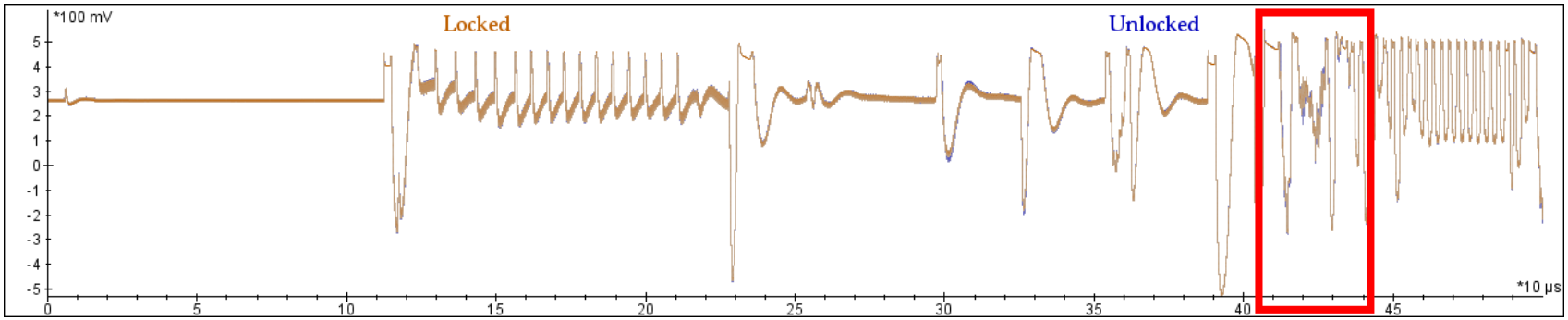
JTAG

1. Investigate normal behavior through (side) channels



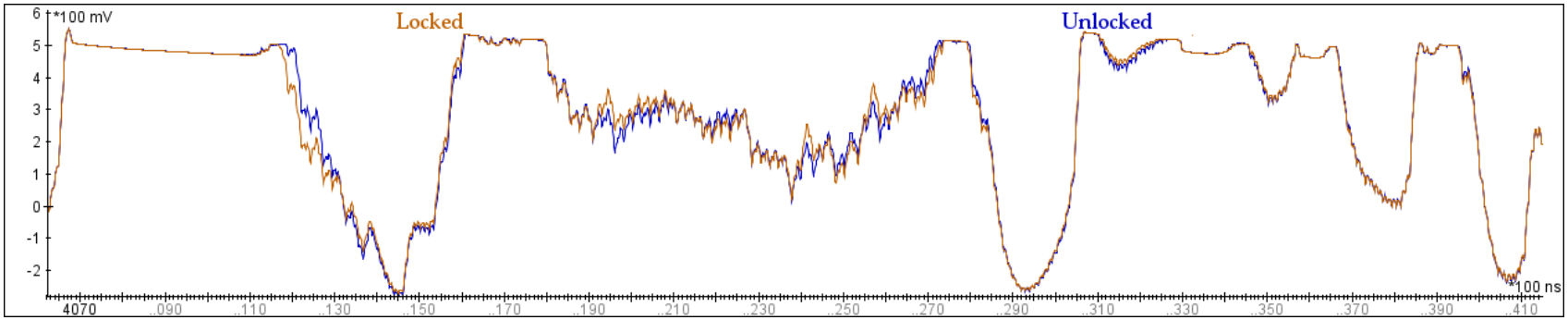
JTAG

1. Investigate normal behavior through (side) channels



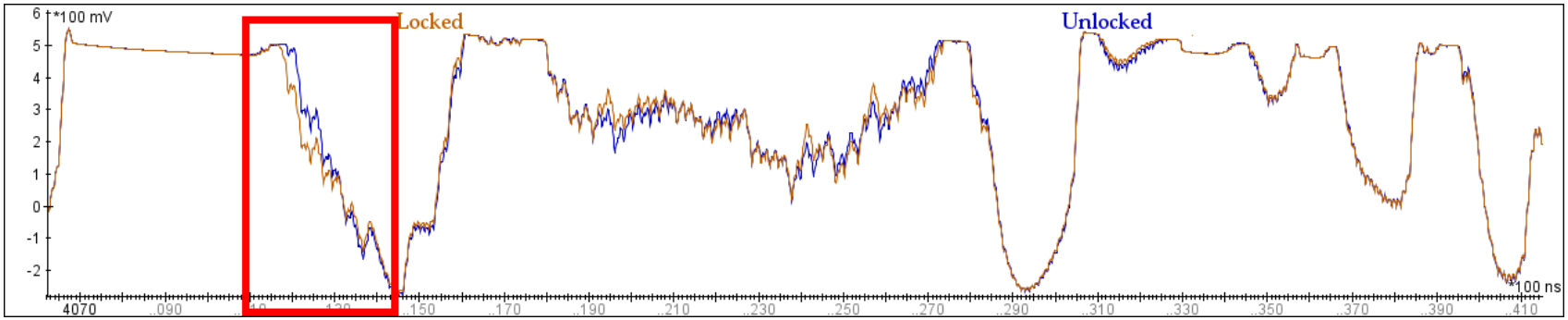
JTAG

1. Investigate normal behavior through (side) channels

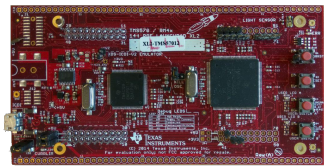


JTAG

1. Investigate normal behavior through (side) channels



JTAG

Target	Successful	Detected
 TI (power)	1.4%	4.5%

Conclusions

- These specific hardware countermeasures not good enough for detection by themselves
- Of the mechanisms, lockstep most effective as countermeasure
- Proper mitigation requires additional measures, either additional in hardware or in software

	Software	Hardware
Latency	higher	**lower**
Cost	**lower**	higher
Fixability	**easier**	harder
Responsibility	implementor	silicon