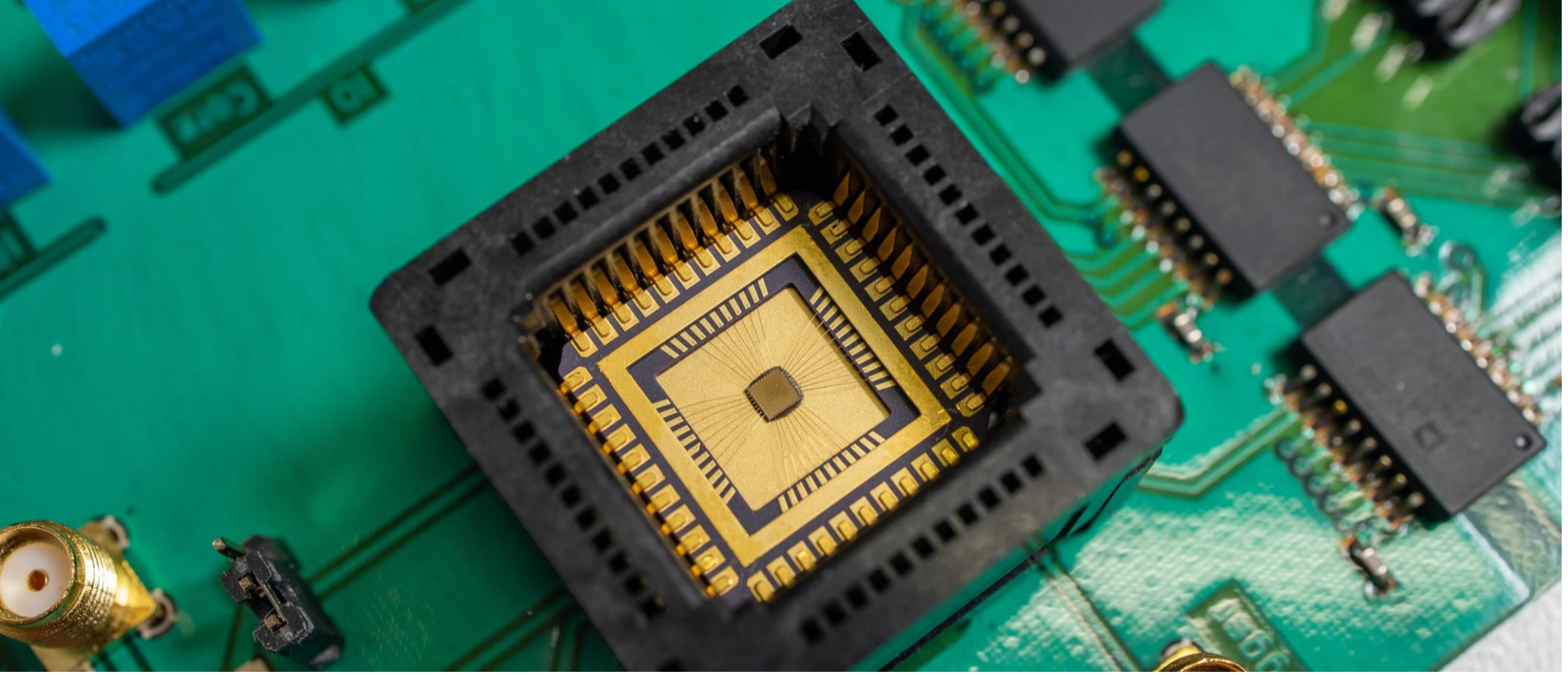


RUB

IMPLEMENTATION SECURITY

RUHR-UNIVERSITÄT BOCHUM

# Energy Consumption of Protected Cryptographic Hardware Cores

Aein Rezaei Shahmirzadi, Thorben Moos, Amir Moradi

CASA
Cyber Security in the Age of Large-Scale Adversaries

DFG

# Motivation

## Low-power designs are crucial for battery-powered devices

- Direct impact on the longevity and overall efficiency
- Smaller batteries

## Battery-powered devices

## Examples in daily life:

- Laptops
- Smartphones
- Tablets
- Wearables
- IoT devices
- Transponders keys

# Related Works

**MIDORI [1] is the only block cipher optimized for low-energy consumption**

**Measuring the energy consumption in most of the publications is based on simulation [2,3]**

**No practical investigation on protected designs:**

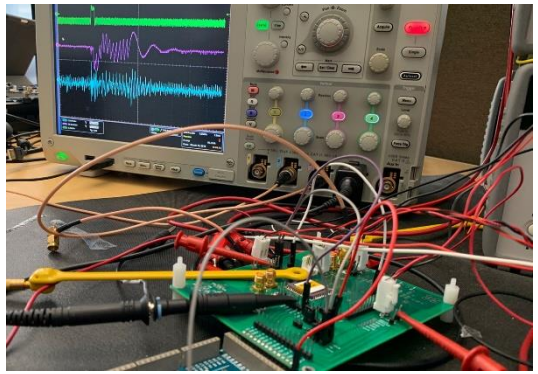
- Masking schemes
- Fault-attacks countermeasures

[1] Banik, Subhadeep, et al. "Midori: A block cipher for low energy." Advances in Cryptology–ASIACRYPT 2015:
[2] Caforio, Andrea, et al. "A deeper look at the energy consumption of lightweight block ciphers." 2021 Design, Automation & Test in Europe Conference & Exhibition (DATE). IEEE, 2021.
[3] Caforio, Andrea, Fatih Balli, and Subhadeep Banik. "Energy analysis of lightweight AEAD circuits." Cryptology and Network Security: 19th International Conference, CANS 2020.

# Masking

## Problem: Side-Channel Analysis

- Timing
- Power Consumption
- Electromagnetic Radiation



## Solution: Masking

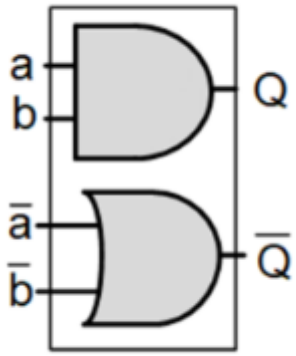
- Randomizing sensitive data during the execution of the cipher
- Split sensitive intermediates into $d + 1$ shares
- Any $d$ wires ➔ No information

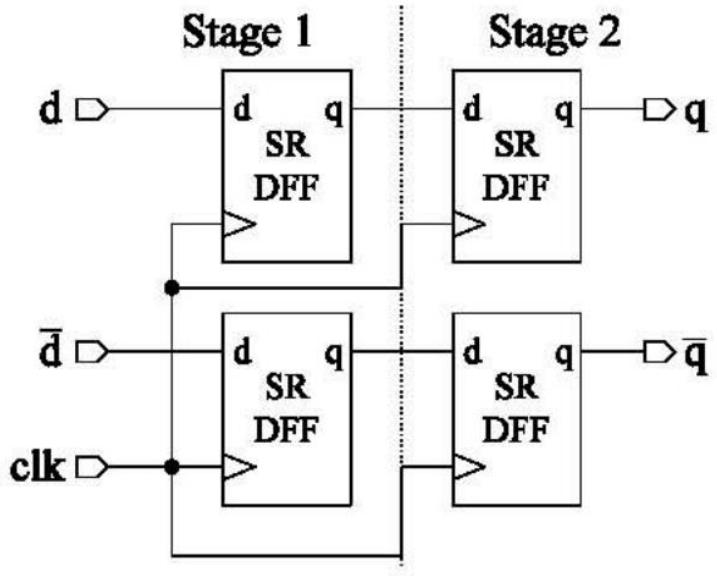
# Wave Dynamic Differential Logic

WDDL has been introduced in [1] to thwart SCA

WDDL can be realized by existing standard cell libraries

A WDDL gate consists of a parallel combination of two positive complementary gates




[1] Tiri, K., Verbauwhede, I.: A logic level design methodology for a secure DPA resistant ASIC or FPGA implementation. In: 2004 Design, Automation and Test in Europe Conference and Exposition (DATE 2004)

# Fault Attacks

**Physical access to the target device**

- Set plaintext and observe ciphertext

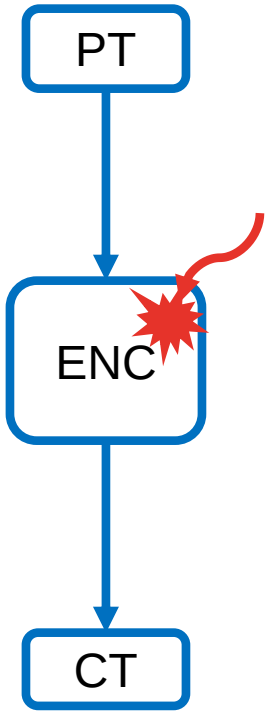
**Inject fault by:**

- Laser
- Electromagnetic
- Clock glitches
- Voltage glitches

**Collect faulty or fault-free ciphertexts**

**Key recovery**

- Sub-key guesses
- Determine correct key

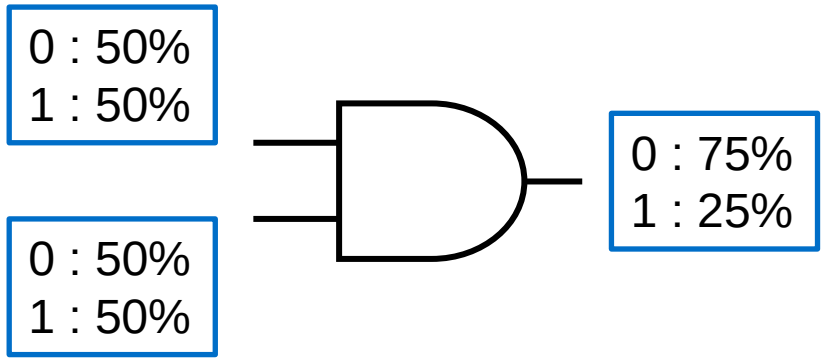


# Statistical Ineffective Fault Attacks

## SIFA requires [1]:

- Only fault-free ciphertexts
- No knowledge about the location or effect of the fault
- Just a single fault

## Intuition:



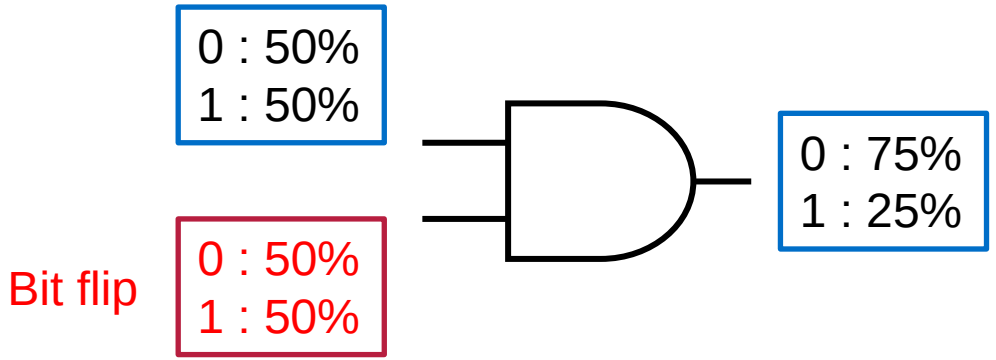

[1] C. Dobraunig, M. Eichlseder, T. Korak, S. Mangard, F. Mendel, and R. Primas, “SIFA: exploiting ineffective fault inductions on symmetric cryptography,” IACR TCHES, vol. 2018, no. 3, pp. 547–572, 2018.

# Statistical Ineffective Fault Attacks

## SIFA requires [1]:

- Only fault-free ciphertexts
- No knowledge about the location or effect of the fault
- Just a single fault

## Intuition:



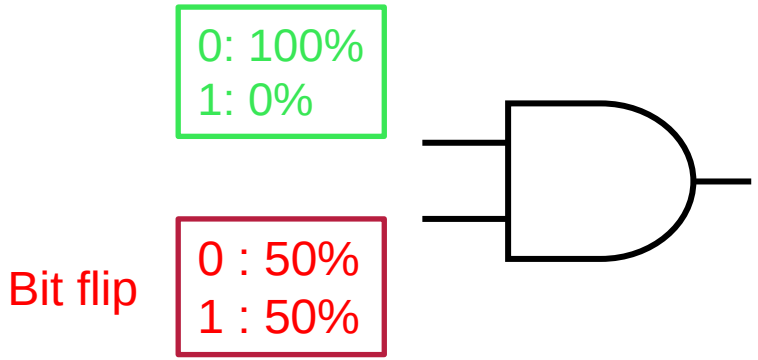

[1] C. Dobraunig, M. Eichlseder, T. Korak, S. Mangard, F. Mendel, and R. Primas, "SIFA: exploiting ineffective fault inductions on symmetric cryptography," IACR TCHES, vol. 2018, no. 3, pp. 547–572, 2018.

# Statistical Ineffective Fault Attacks

## SIFA requires [1]:

- Only fault-free ciphertexts
- No knowledge about the location or effect of the fault
- Just a single fault

## Intuition: Consider fault-free ciphertexts




[1] C. Dobraunig, M. Eichlseder, T. Korak, S. Mangard, F. Mendel, and R. Primas, “SIFA: exploiting ineffective fault inductions on symmetric cryptography,” IACR TCHES, vol. 2018, no. 3, pp. 547–572, 2018.

# Correction Based Countermeasures

**Code-based approach**

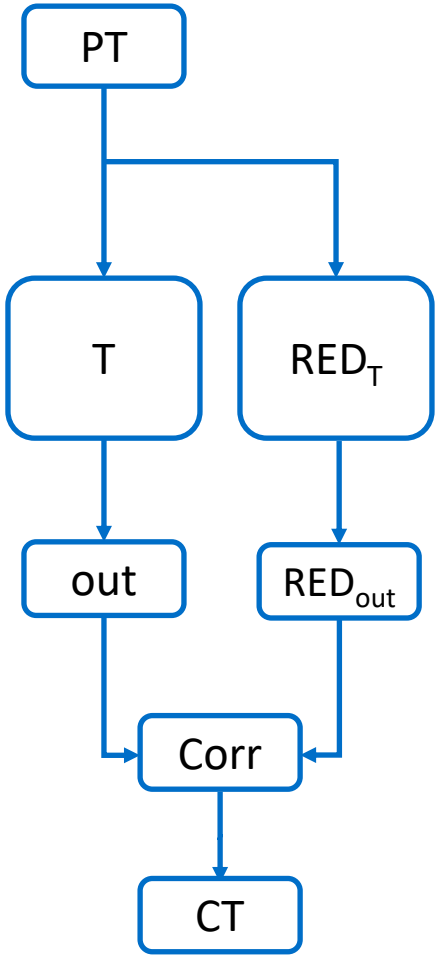
**Employing Error Correction Code**

- Propagation of faults is considered

**Covers every component of the design**

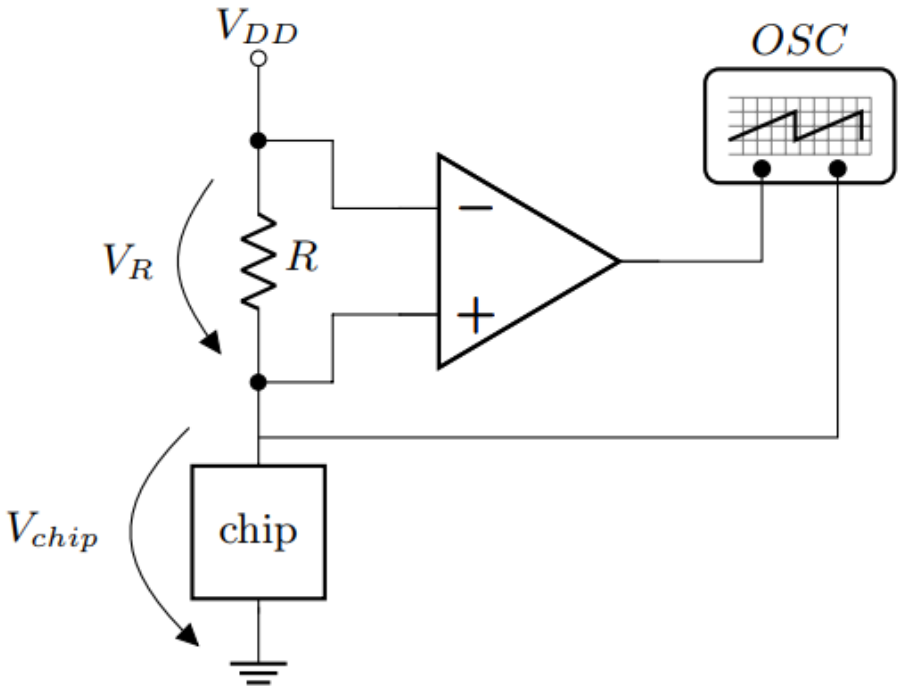
- Data Path
- Control logic
- Consistency check module

**Secure against SIFA**



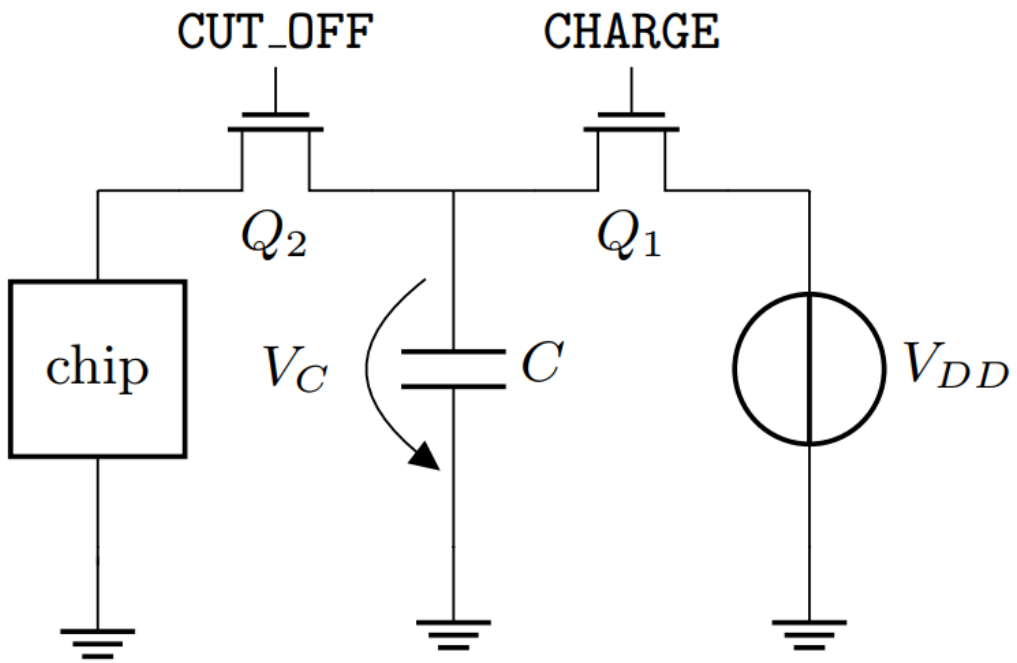
# Measurement Methods

- $P(t) = V(t)I(t)$
- $E = \int_t P(t)dt = \int_t V(t)I(t)dt$



# Measurement Methods

- $E = \frac{1}{2}CV^2$
- $\Delta E = E_{start} - E_{end} = \frac{1}{2}C(V_{start}^2 - V_{end}^2)$

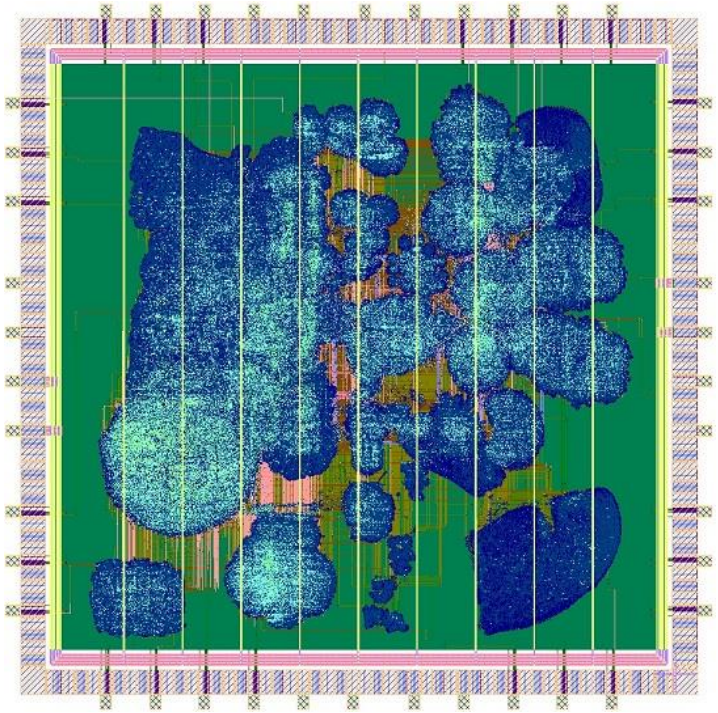


# Our Prototype Chip

## Prototype ASIC in a 65nm technology

## Several cryptographic cores all of which are clock gated

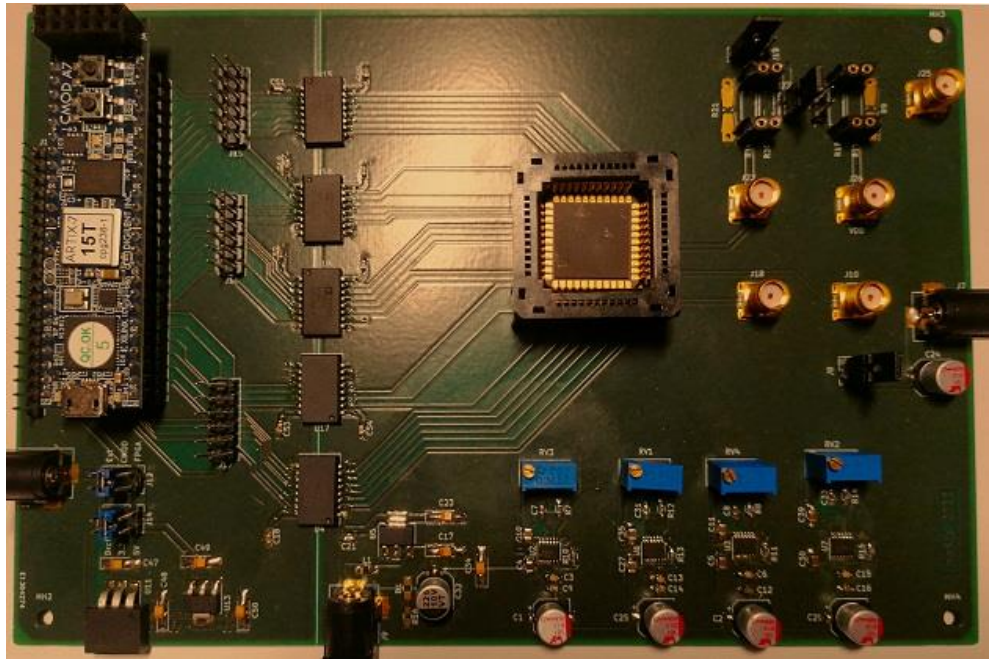
## General control logic



# Measurement Board

## A Xilinx ARTIX-7 FPGA

## The chip is clocked at 12 MHz



# Energy Reference

Energy consumption of the control logic can be considerable

We need to subtract the energy consumed by other components

To have an energy reference we follow exactly the same procedure of measuring the core energy consumption, but we select no core

Average over 10,000 measurements

# Unrolled Implementations

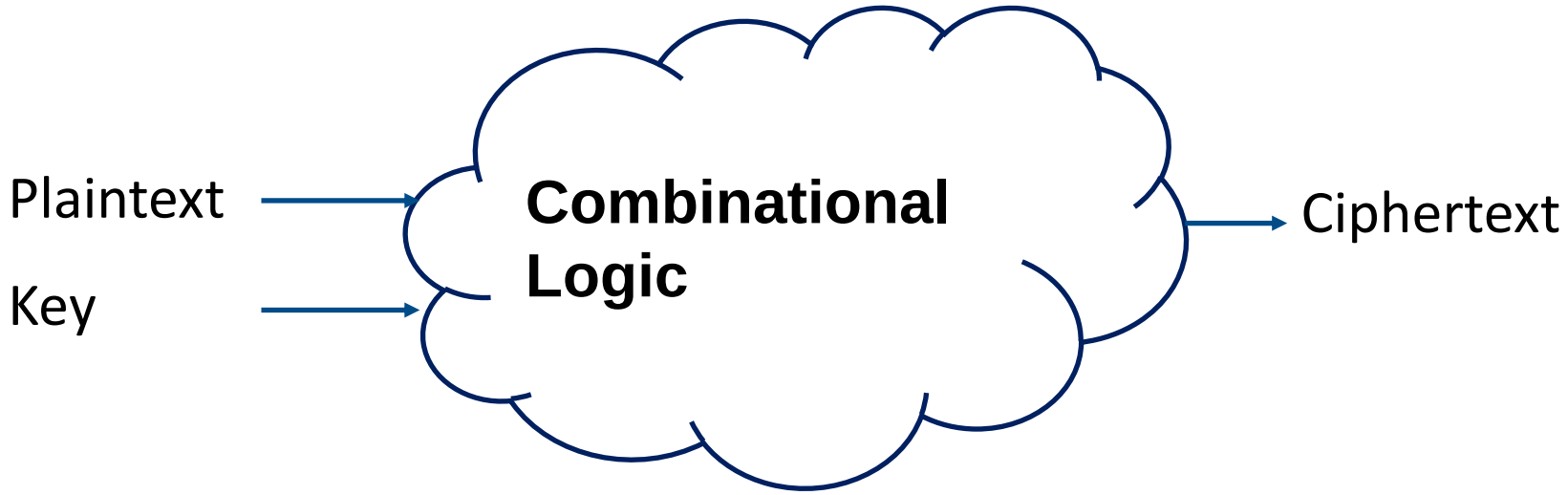
**High area overhead with low latency**

**High power consumption**

- A large number of glitches

**10,000 measurements**

- Random plaintexts
- Fixed Key



# Unrolled Implementations

| Design | Block Size [bit] | Key Size [bit] | Energy/Enc [pJ] | Energy/bit [pJ] |
|---|---|---|---|---|
| PRINCE | 64 | 128 | **361.6** | 5.64 |
| PRINCEv2 | 64 | 128 | 366.4 | 5.72 |
| PRINCE+v2 | 64 | 128 | 373.5 | 5.84 |
| MIDORI | 64 | 128 | 507.5 | 7.93 |
| MANTIS | 64 | 128 | 471.6 | 7.37 |
| QARMA$_7$-64-$\sigma_0$ | 64 | 128 | 572.1 | 8.94 |
| QARMA$_7$-64-$\sigma_1$ | 64 | 128 | 593.6 | 9.27 |
| QARMA$_7$-64-$\sigma_2$ | 64 | 128 | 655.8 | 10.25 |
| Orthros | 128 | 128 | 966.4 | 7.54 |
| SPEEDY-5-192 ENC | 192 | 192 | 598.4 | 3.12 |
| SPEEDY-6-192 ENC | 192 | 192 | 792.0 | 4.12 |
| SPEEDY-7-192 ENC | 192 | 192 | 916.2 | 4.77 |
| SPEEDY-5-192 DEC | 192 | 192 | 1431.6 | 7.46 |
| SPEEDY-6-192 DEC | 192 | 192 | 1910.3 | 9.95 |
| SPEEDY-7-192 DEC | 192 | 192 | 2552.1 | 13.3 |
| GIMLI | 384 | 384 | 811.1 | **2.11** |

# Round-Based Implementations

## AES has better energy efficiency than CRAFT

## The WDDL design consumes roughly 6 times more energy

| Design | Block Size [bit] | Key Size [bit] | Latency [cycles] | Energy/cycle [pJ] | Energy/Enc [pJ] | Energy/bit [pJ] |
|---|---|---|---|---|---|---|
| AES | 128 | 128 | 11 | 3.53 | **38.9** | **0.30** |
| CRAFT | 64 | 128 | 32 | **2.26** | 72.5 | 1.13 |
| CRAFT WDDL | 64 | 128 | 64 | 7.32 | 468.4 | 7.32 |

# First-Order SCA-Secure Implementations

| Design | #Block [bit] | #Key [bit] | Latency [cycles] | #P.[1] | Rand. [bit/cycle] | En./Enc[2] [pJ] | En./P.[3] [pJ] | En./bit[4] [pJ] |
|---|---|---|---|---|---|---|---|---|
| CRAFT TI[5] | 64 | 128 | 64 | 2 | 0 | 867.8 | 433.9 | 6.8 |
| CRAFT CMS | 64 | 128 | 128 | 4 | 0 | 2209.3 | 552.3 | 8.6 |
| CRAFT NF | 64 | 128 | 64 | 2 | 0 | 763.3 | 381.6 | 6.0 |
| CRAFT TI WDDL[5] | 64 | 128 | 128 | 2 | 0 | 4748.1 | 2374.1 | 37.1 |
| CRAFT CMS WDDL | 64 | 128 | 256 | 4 | 0 | 12029.0 | 3007.3 | 47.0 |
| CRAFT NF WDDL | 64 | 128 | 128 | 2 | 0 | 5453.3 | 2727 | 42.6 |
| SKINNY HPC2 | 64 | 64 | 160 | 5 | 64 | 6226.8 | 1245.4 | 19.5 |
| SKINNY HPC3 | 64 | 64 | 96 | 3 | 128 | 2596.6 | 865.5 | 13.5 |
| SKINNY GHPC | 64 | 64 | 96 | 3 | 64 | 4654.6 | 1551.6 | 24.2 |
| SKINNY $\text{GHPC}_{\text{LL}}$ | 64 | 64 | 64 | 2 | 1024 | 1439.7 | 1439.7 | 22.5 |
| SKINNY COMAR | 64 | 64 | 544 | 17 | 6 | 54487.8 | 3205.2 | 50.1 |
| PRESENT TI[5] | 64 | 128 | 64 | 2 | 0 | 1195.7 | 597.9 | 9.3 |
| PRESENT NF | 64 | 128 | 656 | 1 | 0 | 4460.9 | 4460.9 | 69.7 |
| AES | 128 | 128 | 216 | 1 | 8 | 8961.7 | 8961.7 | 70.0 |
| KECCAK NF | 200 | - | 72 | 3 | 0 | 1995.3 | 665.1 | 10.0 |

[1] The number of plaintexts that the design can process due to pipeline architecture.
[2] Energy Consumption per Encryption.
[3] Energy Consumption per Plaintext.
[4] Energy Consumption per bit.
[5] with 3 shares.

# Higher-Order SCA-Secure Implementations

| Design | #d[1] | #Block [bit] | #Key [bit] | Latency [cycles] | #P.[2] | Rand. [bit/cycle] | En./Enc[3] [pJ] | En./P.[4] [pJ] | En./bit[5] [pJ] |
|---|---|---|---|---|---|---|---|---|---|
| SKINNY NF | 2 | 64 | 64 | 128 | 4 | 128 | 4202.2 | 1050.6 | 16.4 |
| SKINNY HPC2 | 2 | 64 | 64 | 160 | 5 | 192 | 10286.8 | 2057.4 | 32.1 |
| SKINNY HPC3 | 2 | 64 | 64 | 96 | 3 | 384 | 4613.3 | 1537.8 | 24.0 |
| SKINNY HPC2 | 3 | 64 | 64 | 160 | 5 | 384 | 15339.3 | 3067.9 | 47.9 |
| SKINNY HPC3 | 3 | 64 | 64 | 96 | 3 | 768 | 6608.8 | 2202.9 | 34.4 |
| PRESENT NF | 2 | 64 | 128 | 666 | 1 | 8 | 10546.6 | 10546.6 | 164.8 |
| KECCAK NF | 2 | 200 | - | 72 | 3 | 0 | 3637.9 | 1212.6 | 18.2 |
| LED | 2 | 64 | 128 | 64 | 2 | 384 | 7502.4 | 3751.2 | 117.2 |

[1] Security order, $d + 1$: number of shares.
[2] The number of plaintexts that the design can process due to pipeline architecture.
[3] Energy Consumption per Encryption.
[4] Energy Consumption per Plaintext.
[5] Energy Consumption per bit.

# Fault Attacks Countermeasures

| Design | #Red./Nib[1] [bit] | #Correction [bit] | #Detection [bit] | Latency [cycles] | En./Enc[2] [pJ] | En./bit[3] [pJ] |
|---|---|---|---|---|---|---|
| CRAFT | 0 | 0 | 0 | 32 | 72.5 | 1.13 |
| CRAFT IC II | 3 | 1 | 0 | 32 | 146.4 | 2.29 |
| CRAFT IC III | 4 | 1 | 2 | 32 | 223.1 | 3.49 |
| CRAFT IC II | 7 | 2 | 0 | 32 | 464.1 | 7.25 |
| CRAFT MV[4] | 8 | 1 | 0 | 32 | 217.5 | 3.39 |

[1] Redundancy size per nibble.
[2] Energy Consumption per Encryption.
[3] Energy Consumption per bit.
[4] Estimated from CRAFT

# Summing up

**We investigated the energy consumption of various implementations on real silicon:**

- Unrolled implementations
- Round-based implementations
- First-order secure implementations
- Higher-order secure implementations
- Protected implementations against fault-injection attacks

**Energy consumption increases by a factor of 6 with WDDL**

**Simply reducing the number of shares in masked implementations does not always result in decreased energy consumption**

**Achieving higher-order security requires a significant increase in energy consumption**

**A bigger area footprint does not necessarily result in higher energy consumption**

# Thanks!

# Any Questions?

Aein.RezaeiShahmirzadi@rub.de