



A Cost-Effective FPGA-based Fault Simulation Environment

Angelika Janning, Johann Heyszl, Frederic Stumpf and Georg Sigl

28th September 2011

1. Motivation
2. FPGA-Based Fault Simulation Environment
3. ECC Case Study

1. Motivation
2. FPGA-Based Fault Simulation Environment
3. ECC Case Study

Fault attacks - Attackers inject faults to recover secrets.

- ▶ Glitches in power or clock supply.
- ▶ Optical attacks (e.g., laser).



Powerful threat for cryptographic implementations.

Cryptographic implementations must be protected.

Cryptographic implementations must be protected.

- ▶ Physical level.
Voltage, light sensors.

Cryptographic implementations must be protected.

- ▶ **Physical level.**
Voltage, light sensors.
- ▶ **General, algorithmic error detection methods.**
Duplication or redundancy.

Cryptographic implementations must be protected.

- ▶ **Physical level.**
Voltage, light sensors.
- ▶ **General, algorithmic error detection methods.**
Duplication or redundancy.
- ▶ **Dedicated, algorithmic detection or prevention.**
E.g. point validity check for ECC.

Cryptographic implementations must be protected.

- ▶ Physical level.
Voltage, light sensors.
- ▶ General, algorithmic error detection methods.
Duplication or redundancy.
- ▶ Dedicated, algorithmic detection or prevention.
E.g. point validity check for ECC.

Countermeasures must be verified.

Laboratory.

- ▶ Late in design flow.
- ▶ Highly complicated.
- ▶ Difficult to reach high coverage.

Laboratory.

- ▶ Late in design flow.
- ▶ Highly complicated.
- ▶ Difficult to reach high coverage.

Simulation.

- ▶ Earlier in design flow.
- ▶ Fault models for abstraction.
- ▶ High performance and coverage possible.

Two choices for hardware simulation.

1. **Software-based simulation.**

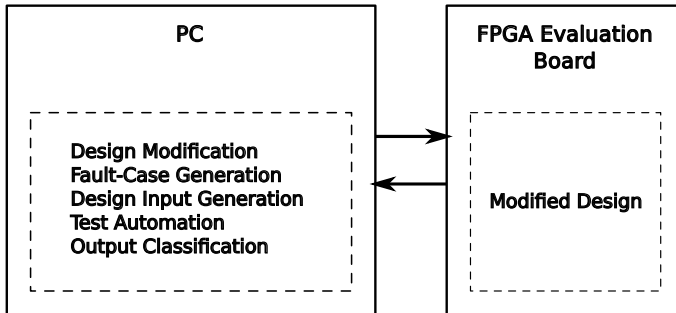
- ▶ Performance low for large designs.

2. **FPGA-based simulation.**

- ▶ Performance high.

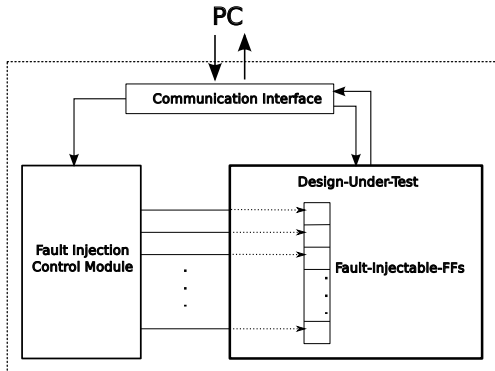
1. Motivation
2. FPGA-Based Fault Simulation Environment
3. ECC Case Study

For digital hardware designs of cryptographic algorithms.

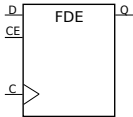


Design-Under-Test (DUT) is modified through script.

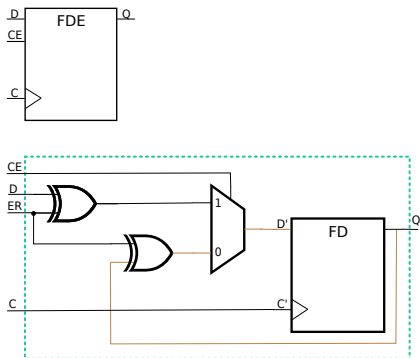
- ▶ Replace flip-flops.
- ▶ Netlist.
- ▶ New top-level.
- ▶ Additional fault injection control and communication module.



Flip-flop cell replacement.



Flip-flop cell replacement.



design source (RTL, VHDL/Verilog)



Synthesis (Xilinx ISE WebPACK).

netlist (VHDL)



Modification for fault simulation (Script).

modified netlist (VHDL)



2nd Synthesis, map, place, route (Xilinx ISE WebPACK).

configuration bit-file

Fault model specific for fault attacks.

Fault model specific for fault attacks.

- ▶ Complete timing control.
- ▶ Complete location control.
All single or multi-bit faults.
- ▶ Bit flip effect.
- ▶ Transient (one cycle) fault duration.

Fault model can be further restricted.

Fault-case generation automated.

- ▶ Requires list of fault-injectable flip-flops.
- ▶ Requires fault configuration.
- ▶ Outputs list of fault test cases.
If space to large, random choice of subset.

Simulation is automated.



1. FPGA configured with modified design.
2. All fault cases are performed. For every case:

Simulation is automated.



1. FPGA configured with modified design.
2. All fault cases are performed. For every case:
 - ▶ Fault injection control module re-configuration.
 - ▶ Start simulation.
 - ▶ Collect output and store testcase outcome.

Available information.

- ▶ DUT output vs. expected output.
- ▶ DUT status indication.
- ▶ Timing compliance.

Available information.

- ▶ DUT output vs. expected output.
- ▶ DUT status indication.
- ▶ Timing compliance.

Fault case simulation outcome.

1. Silent - still correct.
2. Detected - detected fault.
3. Fault - faulty result output.
4. Timeout - No response.

- ▶ **FPGA-based for high performance.**
- ▶ **Supports Verilog and VHDL designs.**
- ▶ **Automated modification on netlist - no design effort.**
- ▶ **Cost-effective Tooling.**

1. Motivation
2. FPGA-Based Fault Simulation Environment
3. ECC Case Study

DUT =

Digital hardware design of **Elliptic Curve Scalar Multiplication**.

- ▶ **Countermeasure** against fault attacks:
point validity check at the end of computation.
→ This countermeasure is evaluated.

Fault model configuration:

- ▶ Derived from fault attacks on ECC.
- ▶ Every clock cycle during ECSM.
 - ▶ **95,000** cycles.
- ▶ Transient (one cycle) faults.
- ▶ Bit-flip fault effect.
- ▶ Single bit (to simplify case study) faults in all registers.
Split simulation in two parts:
 - ▶ **1,600** data-path flip-flops.
 - ▶ **90** control-path flip-flops.

Fault injection **hardware overhead** for DUT $\sim +25\%$.

Simulation performance of *16 ms* per test-case.

- ▶ DUT at *50 MHz*.
- ▶ RS232 transmission more than **70%**.

Simulation in two parts:

- ▶ **Control-path flip-flops.**
 - ▶ ~ 9 million test-cases.
 - ▶ 100 % coverage.
- ▶ **Data-path flip-flops.**
 - ▶ ~ 35 million test-cases.
 - ▶ 22 % coverage.

Simulation in two parts:

- ▶ **Control-path flip-flops.**
 - ▶ ~ 9 million test-cases.
 - ▶ 100 % coverage.
- ▶ **Data-path flip-flops.**
 - ▶ ~ 35 million test-cases.
 - ▶ 22 % coverage.

9 days of simulation ~ 46 years in software simulation.

Control-path.

- ▶ 1 % Timeouts.
- ▶ 38 % Silent faults.
- ▶ 44 % Detected faults.
- ▶ **18 % Faults.**

In most cases program counter was manipulated and intermediate values output. → **Serious problem.** → **Countermeasure required.**

Control-path.

- ▶ 1 % Timeouts.
- ▶ 38 % Silent faults.
- ▶ 44 % Detected faults.
- ▶ **18 % Faults.**

In most cases program counter was manipulated and intermediate values output. → **Serious problem.** → **Countermeasure required.**

Data-path.

- ▶ 0 % Timeouts.
- ▶ 24 % Silent faults.
- ▶ 77 % Detected faults.
- ▶ **0.02 % Faults.**

Most changed output after point check. → **Useless for attacker.**

- ▶ **Speed-up of ~ 2000 compared to software simulation.**
- ▶ **Point validity check countermeasure proved very effective.**
- ▶ **However, weaknesses were successfully exposed.**
 - ▶ **Control path needs protection.**
 - ▶ **Detected faults can be used for safe-error attacks.**