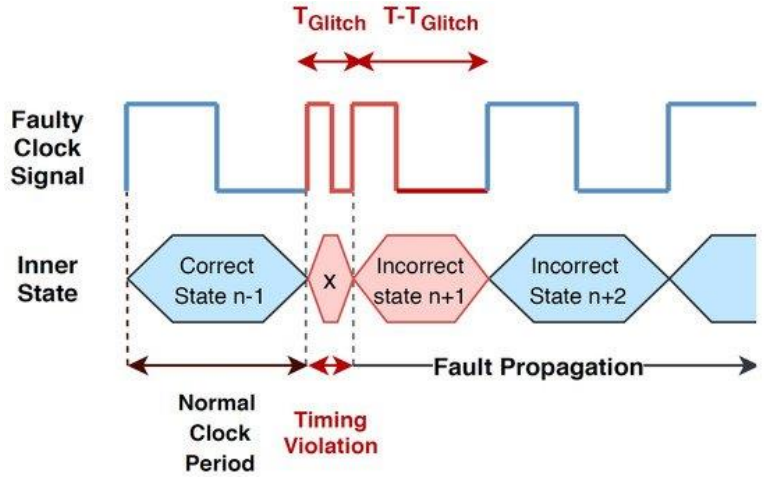
FAME: Fault-Injection Assessment and Mitigation of Microelectronics at Pre-Silicon

Outline

- **Problem Statement**
- **Program Objectives**
- **Accomplishments**
  - Security-property-driven flow to localize fault-Injection vulnerabilities at RTL and gate-level
  - ML-assisted laser fault-injection assessment at layout level
  - ML-driven pre-silicon EM fault-injection evaluation
  - Layout-aware timing fault-injection attack assessment and countermeasures
- **Conclusion, Future Work, and Publications**

Problem Statement

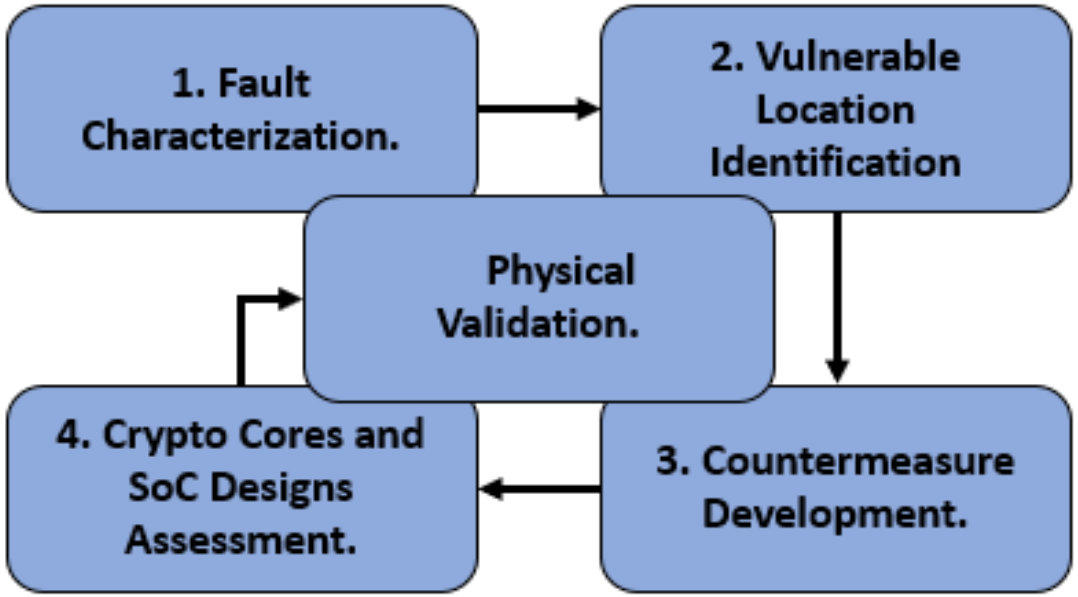
- **The rise of fault-injection attacks**
  - Errors are intentionally injected in a system to compromise the security of the design and facilitate the leakage of assets in the system
  - Taxonomy: non-invasive, semi-invasive, and invasive attacks
- **Security asset:** A resource of value worth protecting from an adversary
- **Examples:**
  - On-device keys
  - IoT device configuration
  - Manufacturer firmware
  - Random number or entropy
  - Application software
  - On-device sensitive data
  - Communication credentials



Technique	Accuracy	Cost	Damage to device
Voltage Glitching	low	low	no
Clock Glitching	low	low	no
EM	low	low	possibly
Light Beam	moderate	moderate	possibly
Laser Beam	high	high	possibly

Program Objectives

- Task 1: Characterize the fault models from different fault injection techniques.
- Task 2: Develop fault injection vulnerability assessment framework.
- Task 3: Create countermeasures using the machine learning and assess the resiliency of them using the developed framework in Task 2.
- Task 4: Apply the proposed assessment framework and countermeasures to crypto cores, processors, and SoCs. Perform physical validations on FPGAs.

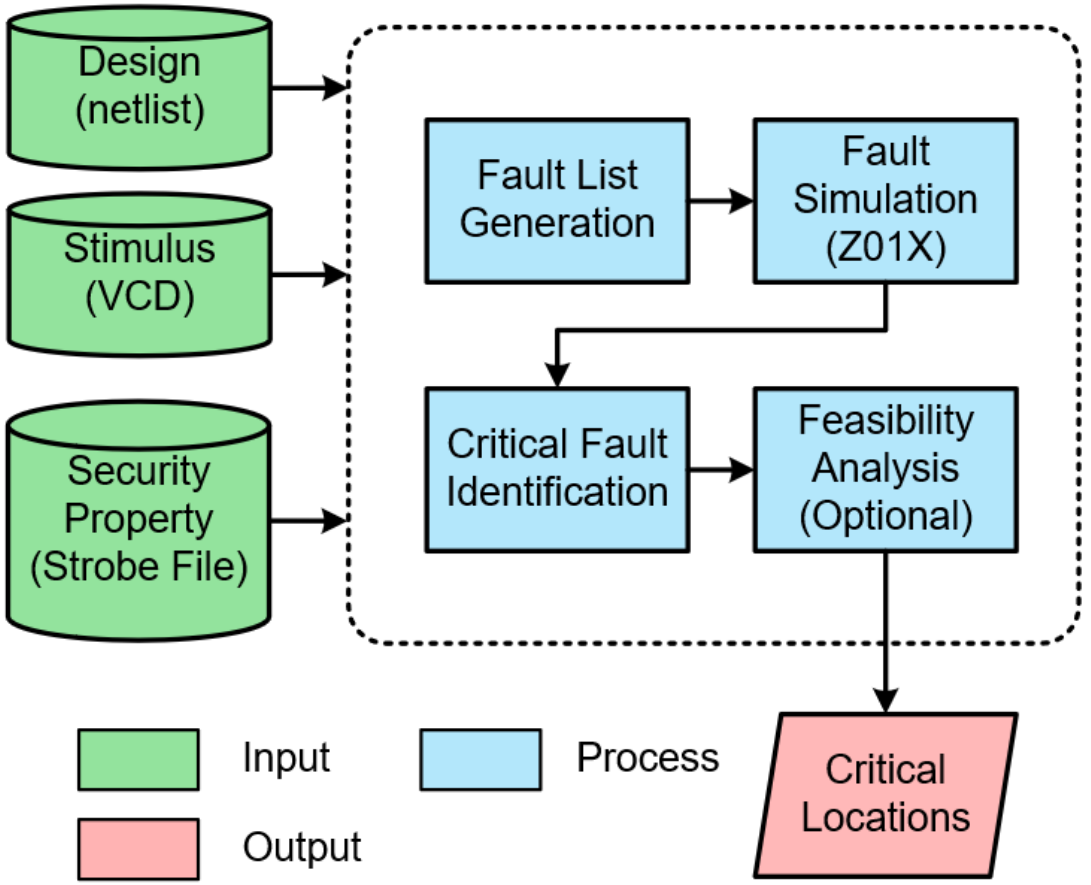


FAME: Fault-injection assessment and mitigation framework

SoFI: Security Property-Driven Vulnerability Assessments of ICs Against Fault-Injection Attacks

SoFI Framework

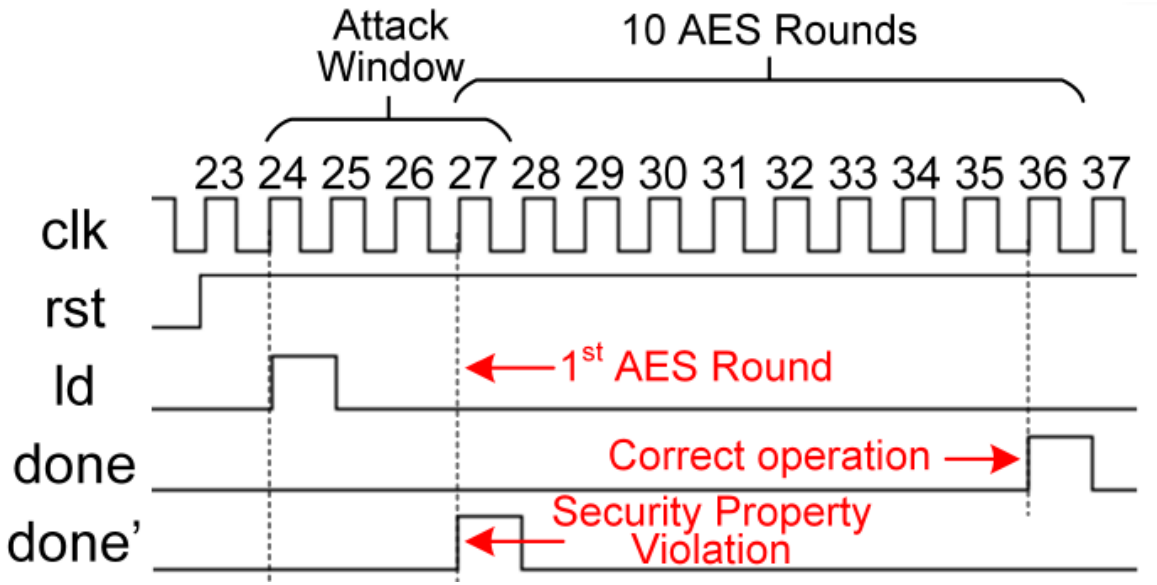
- Assessment tool for ICs against fault injection attack at gate-level
- Security property driven
- Consider the capability of specific fault injection technique
- Critical locations are identified
- Provide opportunity for local countermeasures with low overhead



Security Property

- **Security property**
  - defines behaviors that must be present or must not be present in a design to maintain the integrity, confidentiality, and availability
- **Executable for SoFI**
  - Should be related to or can be violated by one of the fault-injection attacks
  - Should be converted to executable formal presentations with explicit verification metrics

- **One example in AES controller (SP1)**
  - The *done* signal that indicates the completion of 10 AES rounds cannot be raised in the 1st AES round.



Strobe File

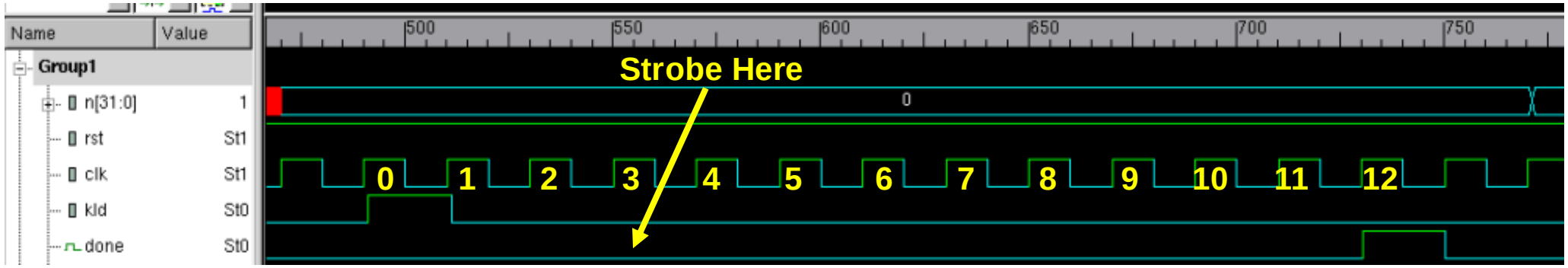
- **Security Property (SP)**
  - Are checked in the fault simulation using strobe file
- **Strobe File**
  - Written in System Verilog
  - **When, Where, and What** to check for security property violation
  - If diff in GM and FM, SP violation

```
initial
  begin
    forever @(posedge done_fanin.ld)
      begin
        @(posedge done_fanin.clk);
        @(posedge done_fanin.clk);
        @(posedge done_fanin.clk);
        #5
        // Compare alu_out signal in GM and FM
        cmp = $fs_compare(done_fanin.done);
        if (1 == cmp)
          begin
            // cmp == 1 indicates diff in GM and FM signals
            $fs_set_status("DD", done_fanin.done);
            break;
          end
        else if (2 == cmp)
          begin
            // cmp == 2 indicates diff in GM and FM signals with X
            $fs_set_status("PD", done_fanin.done);
            break;
          end
      end
  end
```

When: 3 clock cycles after '*load*' *is raised*

Where: at signal '*done*'

What: compare the signals in GM and FM



Fault List Generation

- **Global Faults**
  - Injected by the techniques with no/less control of the fault location, e.g., clock glitching or voltage glitching
  - **Model:** transient bit-flip fault in **FFs** for one clock cycle
- **Local Faults**
  - Injected by the techniques with some/complete control of the fault location, e.g., laser or EM
  - **Model:** transient bit-flip fault in **any cells** for one clock cycle
- **Number of concurrent fault locations**
  - **Small design**: all possible combinations of target cells
  - **Large design**: only consider at most 2 concurrent fault locations
    - The possibility to inject faults at specific multiple locations to violate security property is quite small
    - Limited by the number of beam sources

Evaluation on AES, RSA, and SHA

- **Security Properties in FSMs**
  - SP2.1: In the FSM of AES controller, Initial Round state cannot directly jump to Final Round state without going through Do Round state.
  - SP2.2: In the FSM of RSA controller, Square and Multiply states cannot be bypassed to Result state.
  - SP2.3.1: In the FSM of SHA controller, each time when a block is loaded, the Data Input state should not be bypassed.
  - SP2.3.2: In the FSM of SHA controller, when the last block is loaded, the Block Process and/or Block Next state should not be bypassed.
- **Security Properties against Differential Fault Analysis (DFA)**
  - SP3.1: At the 9th round of AES, any 1-3 bytes of the first word in the round key cannot be faulty and the faulty bytes cannot propagate to the following words in the same round.
  - SP3.2: At the 9th round of AES, 4 bytes of any word in the round key cannot be faulty and the faulty bytes cannot propagate to the following words in the same round.

1. A. Nahiyan, et al, “Security-aware fsm design flow for identifying and mitigating vulnerabilities to fault attacks,” IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 38, no. 6, pp. 1003–1016, 2019
2. C. H. Kim and J.-J. Quisquater, “New differential fault analysis on aes key schedule: Two faults are enough,” in Smart Card Research and Advanced Applications, pp. 48– 60. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008

Fault Simulation Results

Benchmark	Security Property	Stimulus	Fault Category	# of Concurrent Fault Locations	Total Faults	Effective Faults	Critical Faults	Feasible Faults	Critical Locations	% in AES/ RSA/SHA	CPU Run Time (s)
AES FSM 1	2.1	1	Global	1-7	508	19	4	0	0	0.00%	1
			Local	1-4	145,824	7,173	113	NA	7	0.07%	82
AES FSM 2	2.1	1	Global	1-7	508	18	3	1	1	0.01%	1
			Local	1-4	83,412	6,516	63	NA	7	0.07%	44
RSA FSM 1	2.2	1	Global	1-7	381	18	3	0	0	0.00%	1
			Local	1-4	95,790	8,533	13	NA	8	0.01%	141
RSA FSM 2	2.2	1	Global	1-7	381	21	3	1	1	0.002%	1
			Local	1-4	72,471	5,104	14	NA	7	0.01%	137
SHA FSM	2.3.1	1	Global	1-3	42	2	2	0	0	0.00%	1
			Local	1-3	301,098	2,491	160	NA	12	0.28%	401
SHA FSM	2.3.2	1	Global	1-3	42	2	2	1	1	0.02%	1
			Local	1-3	301,098	17,508	85	NA	24	0.55%	411
SHA FSM	2.3.1 or 2.3.2	1	Global	1-3	84	4	4	1	1	0.02%	1
			Local	1-3	602,196	19,999	245	NA	25	0.58%	801
AES KS	3.1	100	Global	1	420	12	12	6	6	0.06%	432
			Local	1	6,921	1,783	1,783	NA	1,783	17.33%	5,855
AES KS	3.2	100	Global	1	420	0	0	0	0	0.00%	414
			Local	1	6,921	1	1	NA	1	0.01%	4,993

- Inappropriate FSM encoding scheme can bring additional vulnerability (yellow)
  - FSM 1 and FSM 2 are using different FSM encoding
- < 0.6% locations are identified as critical locations against fault-injection (green)
  - Except the local faults for SP3.1 (17.33%), since SP3.1 is too easy to violate

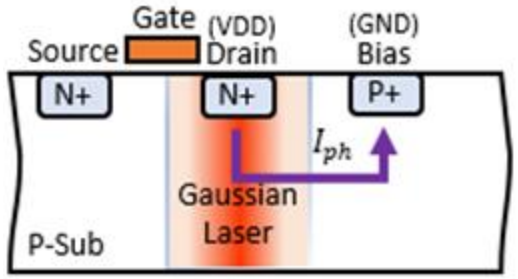
A Summary of FIA-Critical Location Types

- **Observations from FSM-based results on crypto cores (AES, RSA, SHA)**
  - FSM state registers
  - FSM counters
- **Observations from DFA-based results**
  - AES intermediate state registers (especially for the 8^{th}, 9^{th}, and 10^{th} round)
  - AES round key registers (especially for the 8^{th}, 9^{th}, and 10^{th} round)
  - AES key expansion module's combinational logic cells
- **Common critical locations for software execution in processors**
  - Program counter registers
  - Instruction registers
  - Instruction memory cells
  - Multiplexers and their control signals for arithmetic logic units (ALUs)
  - Pipeline registers in float-point units (FPUs)
- **Critical locations for SoCs**
  - Arbitrators
  - AI accelerators' critical components (e.g., FSM signals in activation functions of DNNs)

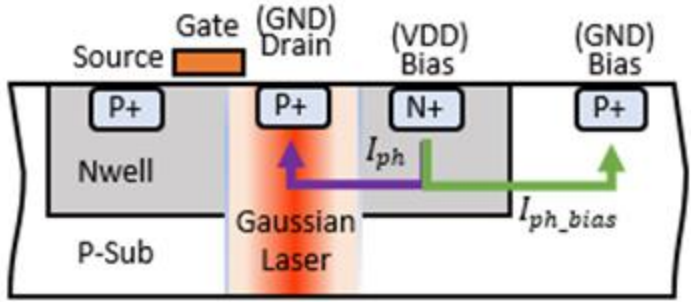
SPILL: Security Properties and Machine-Learning Assisted Pre-Silicon Laser Fault Injection Assessment

Laser Fault Injection (LFI)

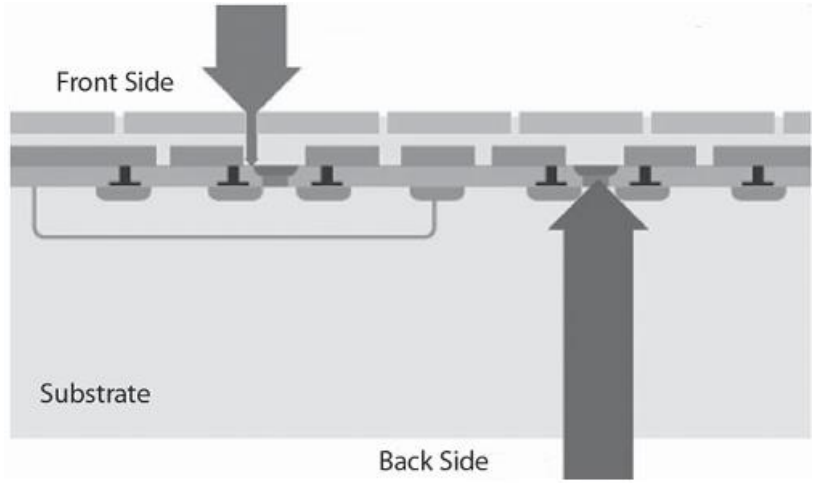
- Infrared laser is used to affect a smaller area. (wavelength: 760 nm - 1100 nm, typically from backside of the active region)
- Laser creates electron-hole pairs at the drain of a NMOS and thus creates a current pulse.
- Current pulses of all impacted cell add up and create IR drop & ground bounce.
- Short time to avoid damage.
- **Objective of LFI / OFI assessment** is to test all cells of an IC for unwanted IR drop and ground bounce, and identify possible vulnerabilities.



(1) NMOS under laser



(2) PMOS under laser



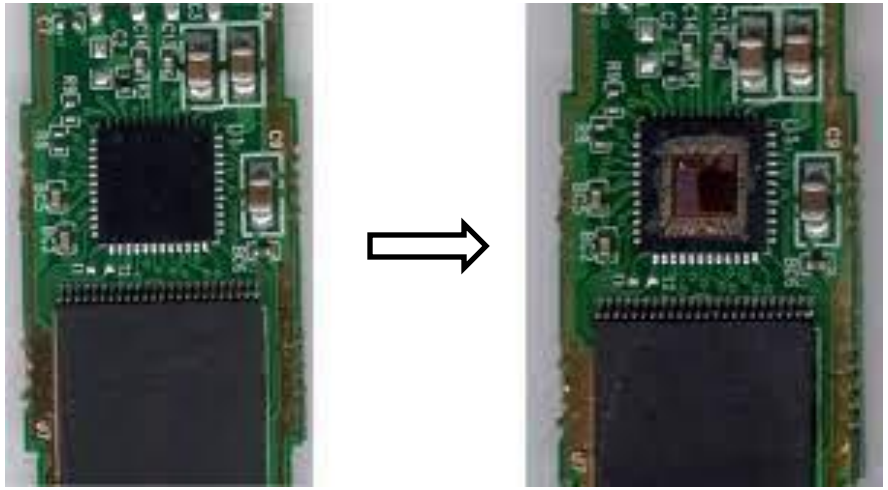
(3) illustration of front side and back side laser illumination

Post-Silicon LFI Verification Steps

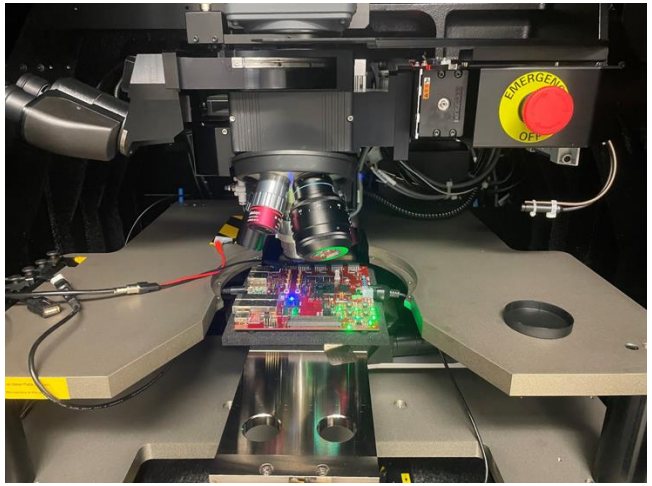
- **Steps to perform laser verification**
  - De-package
  - Thinning
  - Laser Illumination
  - Monitor output

Challenges:

- Improper surface can cause laser scattering.
- Can lead to false negatives – False sense of security.
- A lot of time required to focus the laser and alignment.



(1) Chip depackaged while on board

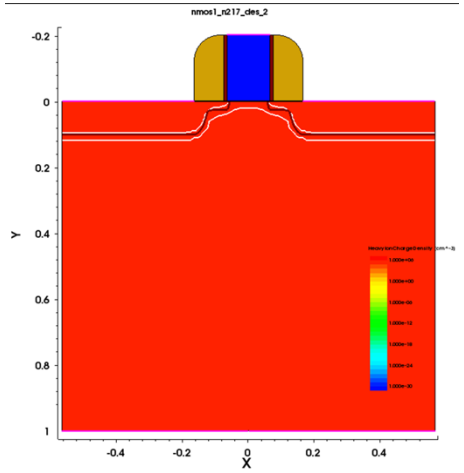


(2) Phemos-1000 with 1060nm laser

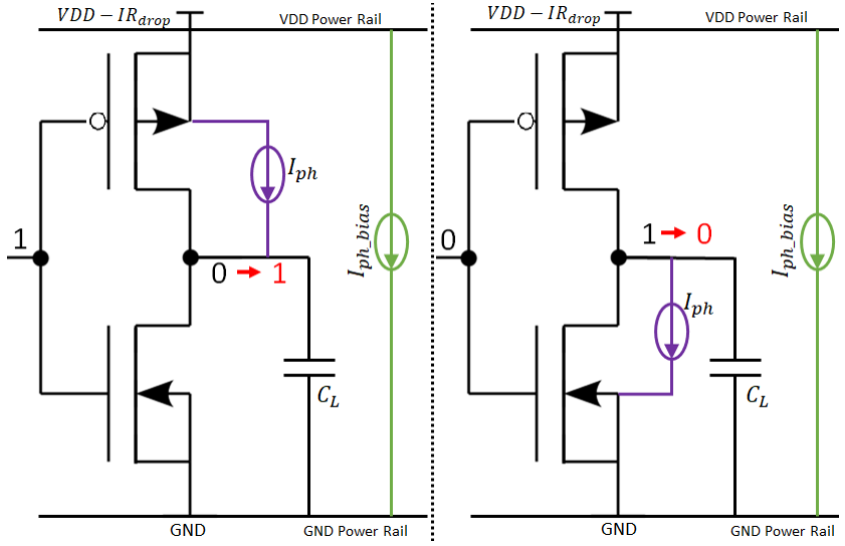
Moundi, P. L. (2019). Cost effective techniques for chip delayering and in-situ depackaging. *Cryptology ePrint Archive*.

Pre-Silicon LFI Modeling Efforts

- **Logical fault simulation** relies on injecting fault during RTL/netlist simulation, thus is fast but cannot account for any physical characteristics of the laser (laser intensity, spot size, wavelength, etc.) nor the layout information.
- **For electrical fault models**, the laser's impact is modeled as a photocurrent (using the current source) induced at the reverse-biased PN junctions during SPICE simulations. However, not scalable for large designs.
- **Device-based (TCAD) fault models** use heavy ions to model the laser impact during device simulation. Not scalable.



(1) Laser modeled as heavy ion on 32nm NMOS



(2) Laser modeled as current source on reverse-biased PN junctions of an inverter

Motivation

- **There is need for a tool that can allow laser verification on the entire chip**

- **Consideration of physical characteristics**
  - Physical information such as cell's spatial position, neighbor cells, power distribution network, switching activity, etc. should be taken into account
  - Can be used to evaluate the effectiveness of possible countermeasures, since such evaluation is difficult to perform in post-silicon stage

- **Flexibility in design changes**
  - Rapid assessment for different designs

- **Variations in threat models**
  - Vary the modeling based on different attacker capabilities (money and equipment)
  - Enable a cost-vs-countermeasure efficacy study depending on different users

Our Goal: A scalable pre-silicon LFI / OFI assessment framework and countermeasure development, to allow a fast and LFI / OFI resistant physical design flow.

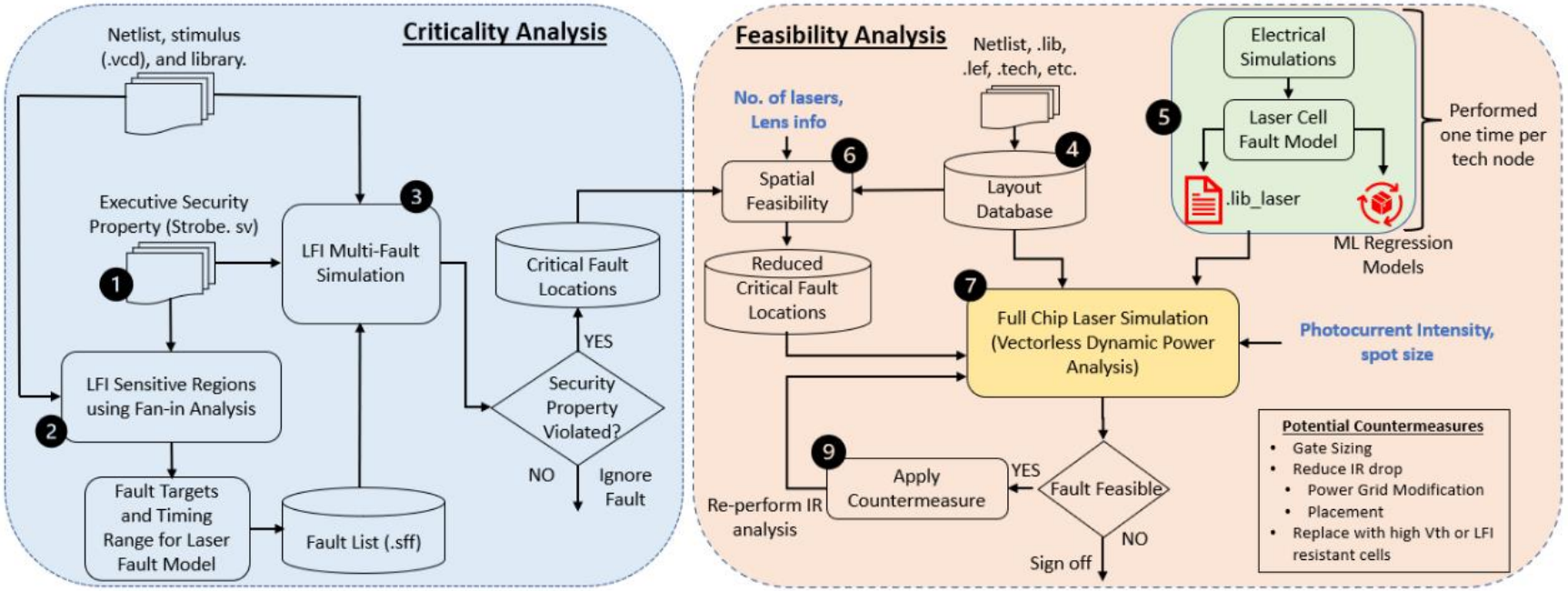
Laser Fault Injection Assessment

Anticipated Results:

To obtain laser fault locations after performing laser fault simulation and to discover feasibility of the fault.

Overall flow for fast and scalable pre-silicon LFI assessment:

- **Criticality Analysis**: security property driven assessment using logical fault simulations at the gate-level netlist to identify the critical locations (gates/flip-flops) in the layout.
- **Feasibility Analysis:** SPICE simulations and machine learning to develop cell-level laser fault models under different laser-induced transient current intensities. This laser cell library is used during full-chip LFI feasibility analysis for the cells inside laser illumination.



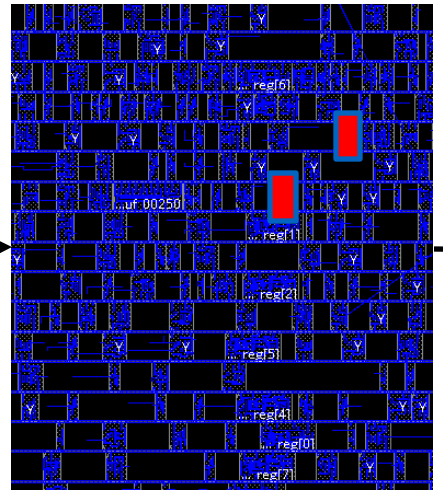
Feasibility Analysis

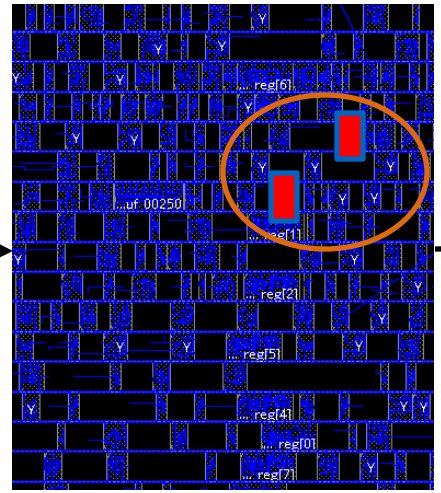
Multi-fault LFI feasibility is assessed in two ways

1. Given the geographical locations of the fault nodes, is it possible for laser spot to cover nodes simultaneously?
2. Given the laser parameters and distance from the spot center, enough energy is injected to cause fault?

1. Cells Location Feasibility

For the given multi-fault identified registers





Can they be covered within the same laser spot?

2. Laser Power Feasibility

Once identified that this particular multi-fault scenario is geographically possible.

Identify the distance of the cell from the spot center.

Model the laser effect and perform full-chip layout simulation.

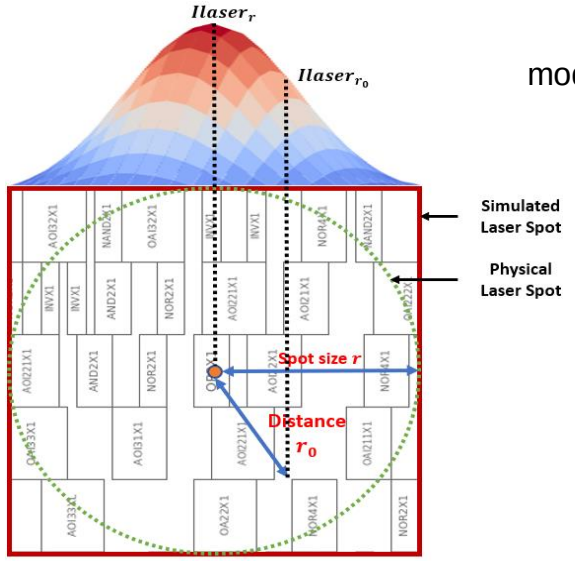
For the given laser power, is it possible to flip the bit?

ML-Assisted Laser Modeling at Layout

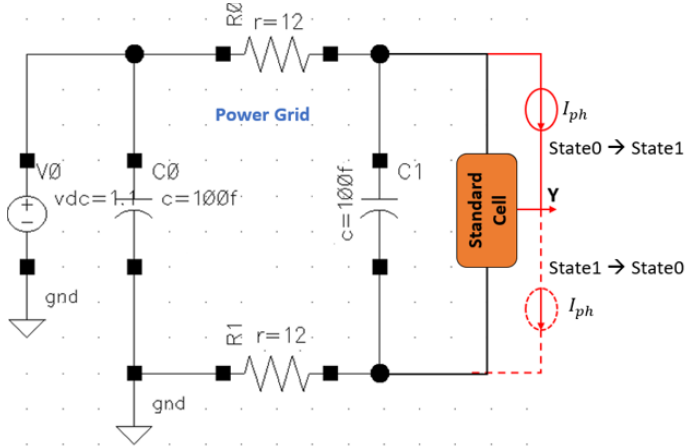
- **Gaussian distribution of laser intensity**
  - All cells inside laser spot are replaced with cells from the cell-level laser library.
  - Depending on the distance from the spot center, we scale up or down the current demand on the cell.

- **Regression model for laser cell library**
  - Current change due to laser is modelled as current sources from output to Vdd/Gnd
  - Regression models are needed to identify the current demand for the given laser transient current.
  - Due to non-linear relationship, KNN model worked the best with more than 98% accuracy on 80-20 train test split.
  - This needs to be done only once for all standard cells in the technology library.

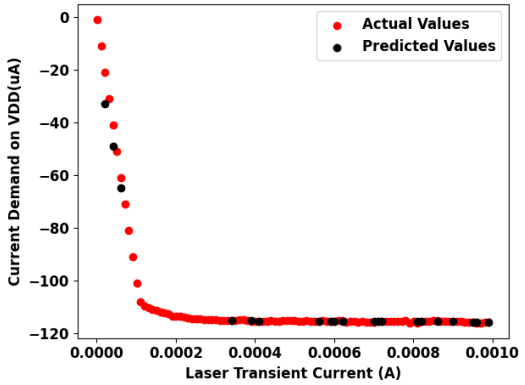
(1) illustration of laser spot and Gaussian distribution of laser intensity.



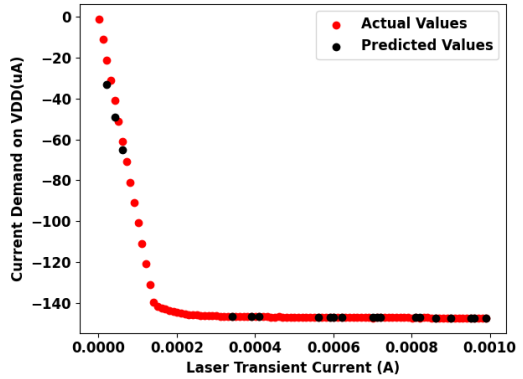
(2) Current demand change due to laser is modelled as current sources in SPICE simulations



(3) Cell current demand prediction



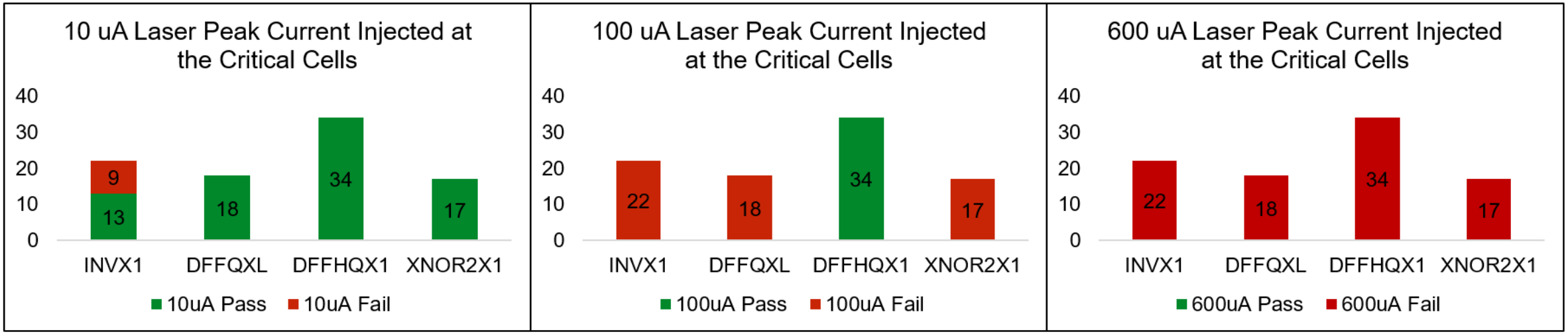
(3.a) Laser State 0, Cell: INVX1, Pin VDD.



(3.b) Laser State 1, Cell: INVX1, Pin VDD.

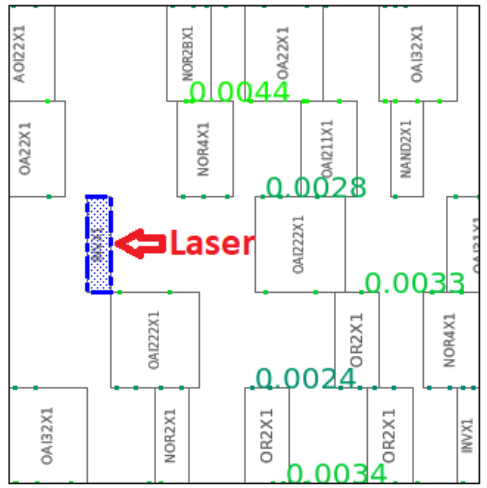
Feasibility Analysis

Number of critical cells failing for different laser current intensities

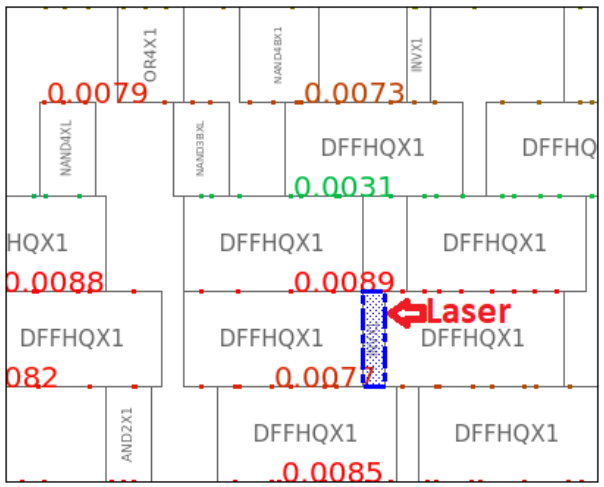


Observations

- Cell Size can impact susceptibility.
- Surrounding cells and their switching activity can impact the susceptibility.
- Distance from PDN can impact the susceptibility.



(1) INVX surrounded by small cells

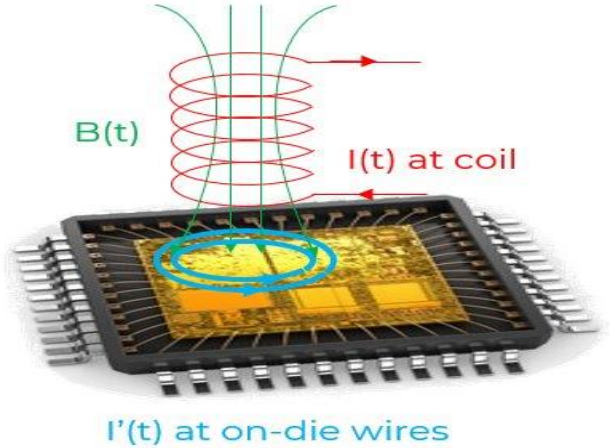


(2) INVX surrounded by bulky cells

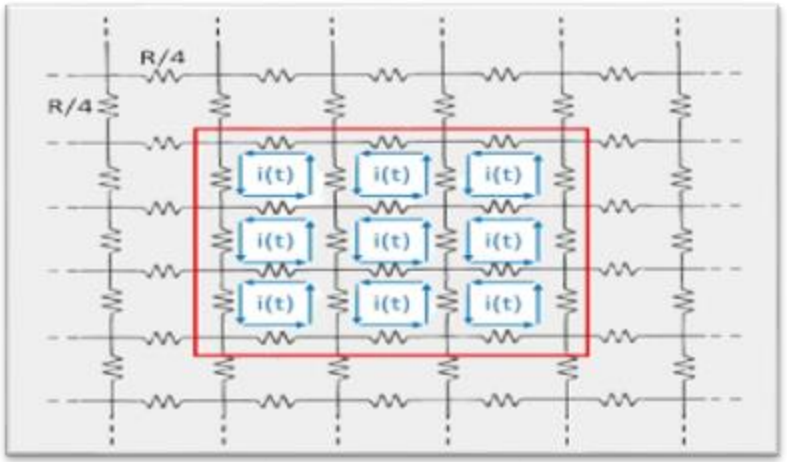
ML_PREMISE: A Machine Learning Driven Pre-Silicon Electromagnetic Fault Injection Security Evaluation for Robust IC design

Introduction to EM Fault Injection (EMFI)

- **EMFI Definition**
  - EMFI introduces faults into electronic ICs
    - Coil generates EM radiation
    - Requires EM Coils, pulse current generator
    - Under $1000 setup cost



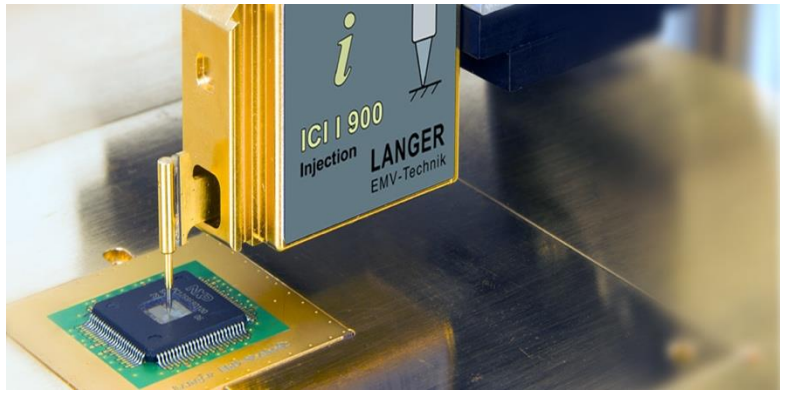
- **EMFI Effects**
  - EM radiation induces Eddy current in PG metals
    Induced current can flow along on-chip grid
  - Propagation of current translate to voltage drop
  - Timing violation may happen on critical path
  - Negative slacks and slacks with values close to zero will be most affected.



Eddy current on the PG grid

Problem Definition and Objective

- **Problem Definition**
  - EMFI can cause transient faults that can
    - corrupt data
    - Lead to erroneous outputs
    - Cause unpredictable behavior, crashes, or system resets
    - Bypass encryption, and reveal sensitive information
  - A designer needs to be able to analyze the robustness of the IC against EMFI in pre-silicon stage



A typical EMFI setup (Langer EMV)

- **Objective**

  This project aims to develop a fast and novel EMFI simulation flow that can accurately predict the induced currents with relatively low cost and reduced design time cycle.

EMFI Simulation Flow

ML based optimization

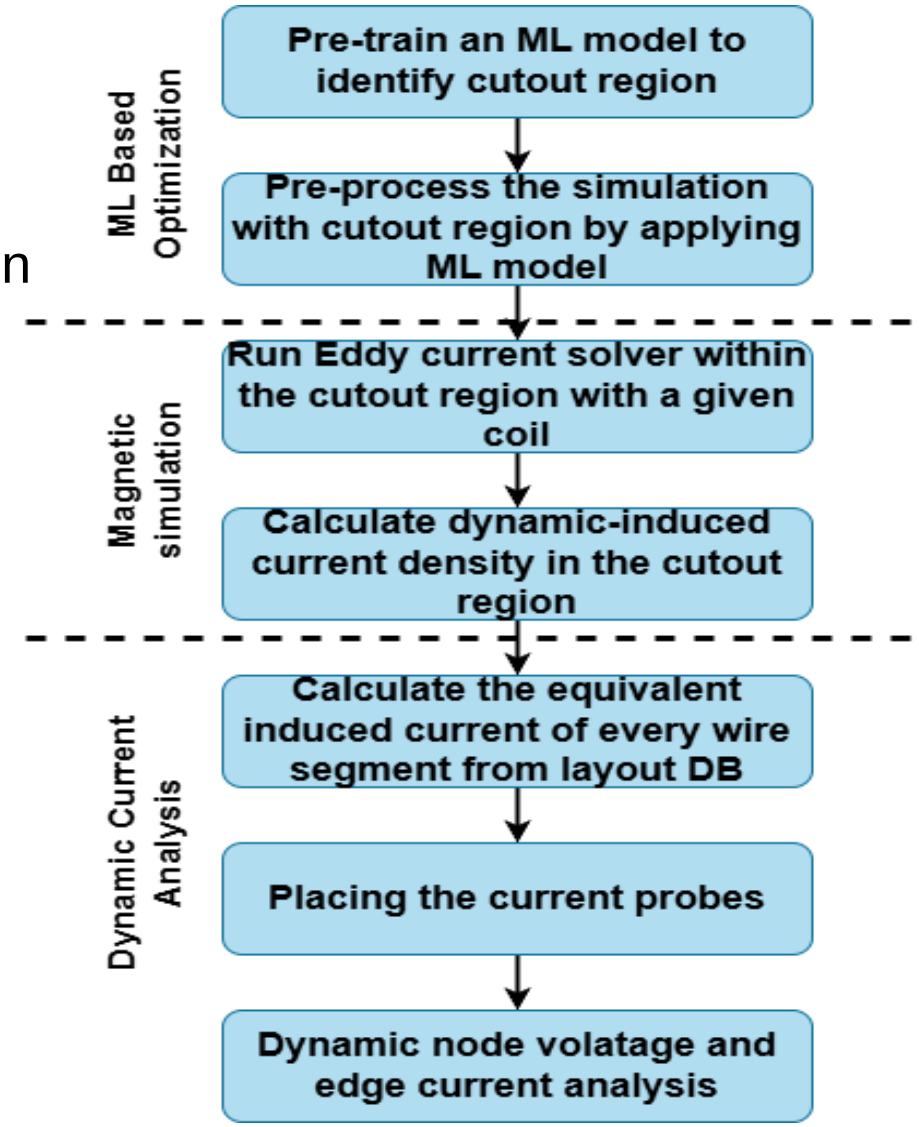
- Pretrain an ML model to identify Cutout region to save runtime
- Preprocess the simulation with cutout region by applying ML model

Magnetic solver simulation

- Run Eddy current solver within the cutout region for a given coil
- Export current density within the cutout region

Layout based simulation

- Calculate the dynamic current of every wire segment
- Place the current probes
- Dynamic edge current and node voltage analysis
- Calculate Figure of Merit



Magnetic Transient Simulation

- Eddy current induction occurs when a time-varying magnetic field, typically produced by an AC coil, induces circulating currents in a nearby conductor.

$$\mathcal{E} = -\frac{d\Phi_B}{dt}$$

Where, Φ_B is the magnetic flux

$$\Phi_B = \mu_0 \mu_r N A I(t)$$

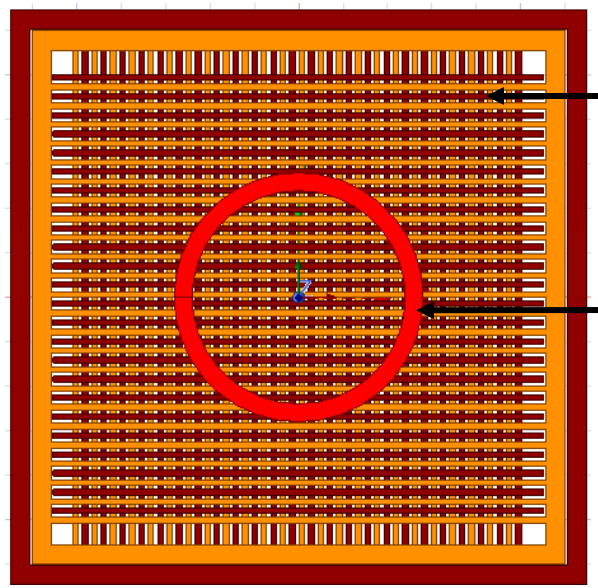
μ0: Permeability of free space,
μr: Relative permeability,
A: Cross-sectional area of coil.

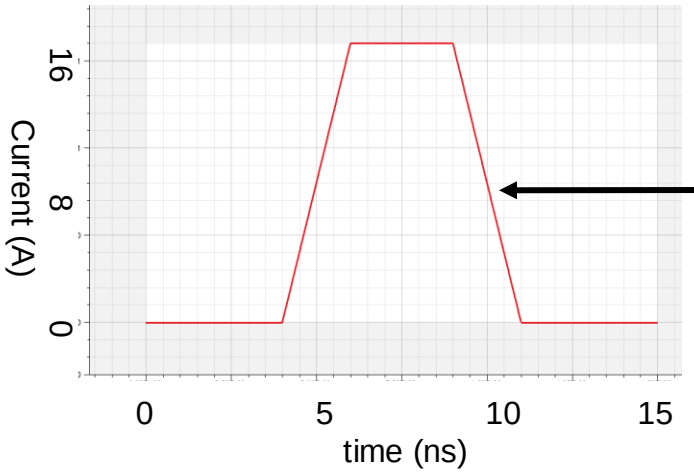
$$\mathbf{J} = \sigma \mathbf{E} = -\sigma \frac{\partial \mathbf{A}}{\partial t}$$

J: Current density,
σ: Conductivity of the material (S/m)
A: Magnetic vector potential (Wb/m)

- A commercial Eddy current solver named Maxwell was used for calculating Eddy current

Benchmark: AES 128





Modeling in Maxwell

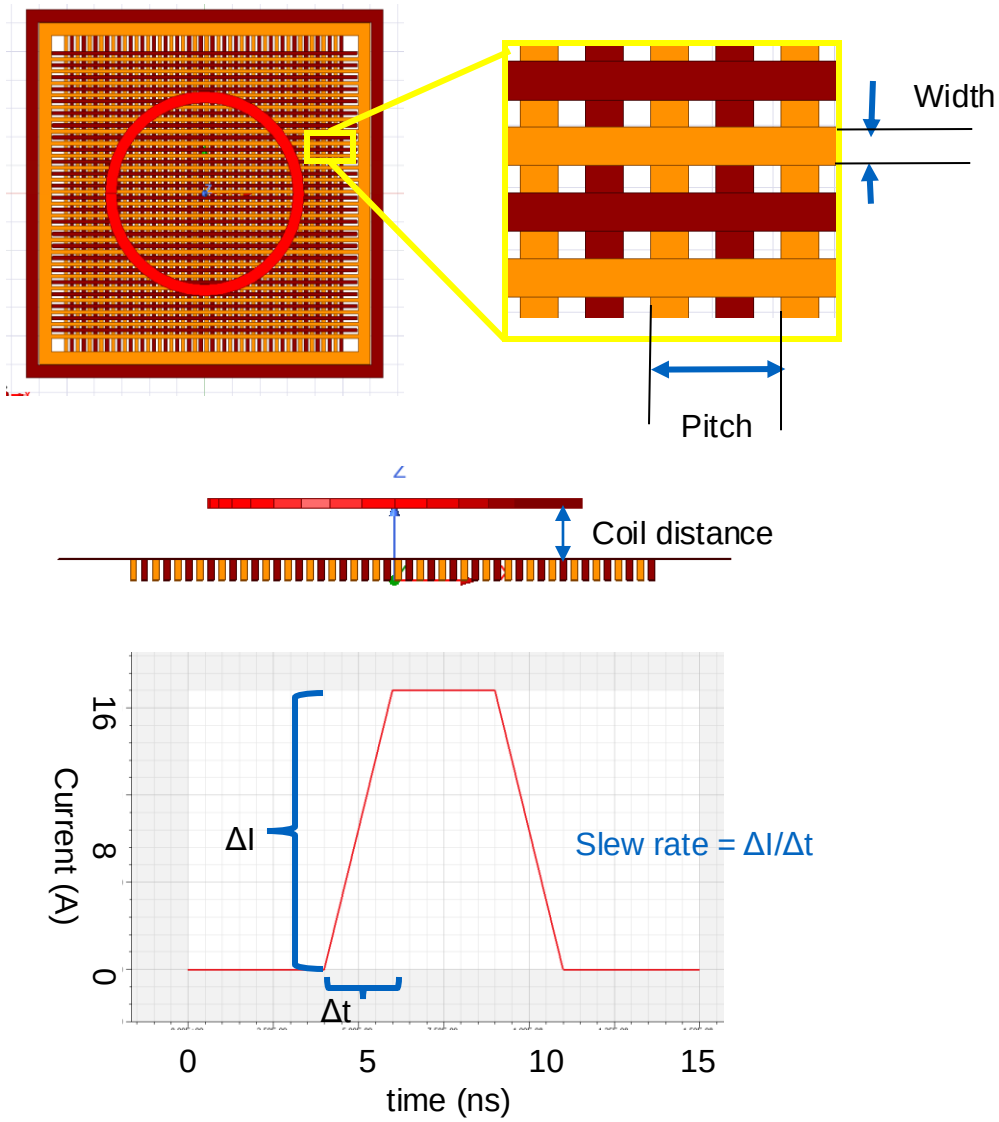
ML Based Cutout Region Identification

Objective

- Reduce magnetic simulation time by identifying a ‘cutout’ region
- Cutout region: most affected area by the EM coil

Generation of Training Data

- Input variables
  - Metal Layers Width
  - Metal Layers Pitch
  - Slew Rate
  - Coil Distance from top metal layers
- Output Response
  - Area (x, y) that has a Mag(J) value > 5% Mag($\mathbf{J}$)$_{max}$

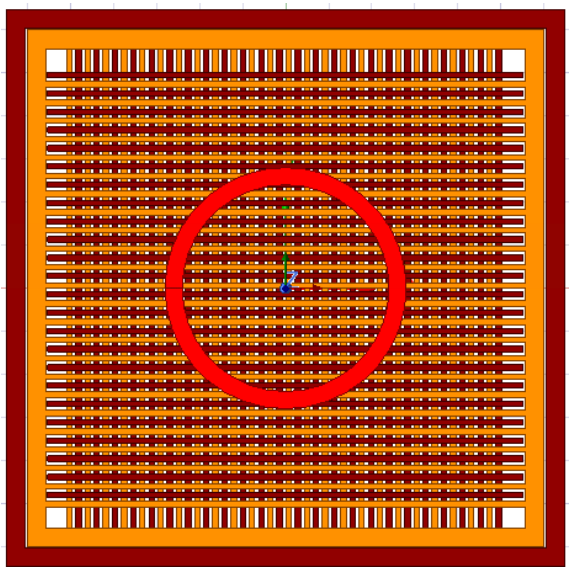


ML Based Cutout Region Identification (Cont'd)

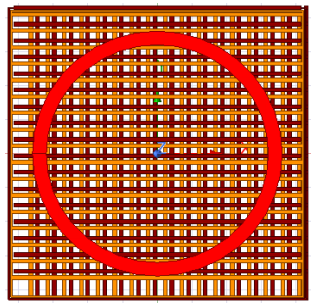
Training dataset

Input Variable	Nominal Values	Sweep Range
Metal Pitch	21.2 um	+/- 20%
Metal Width	6.2 um	+/- 20%
Coil Distance	20 um	10-40 um
Slew Rate	8A/ns	6-10 A/ns

Full Geometry



Cutout Region

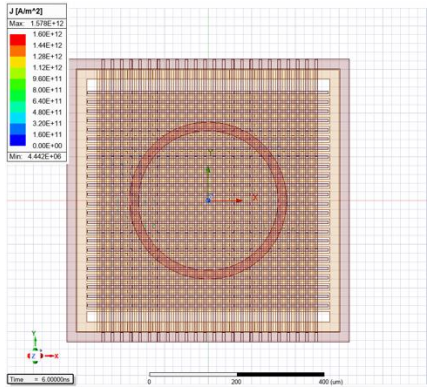


Testcase

Input Variable	Nominal Values
Metal Pitch	20 um
Metal Width	8 um
Coil Distance	30 um
Slew Rate	12A/ns

	MagJ
X	284 um
Y	308 um

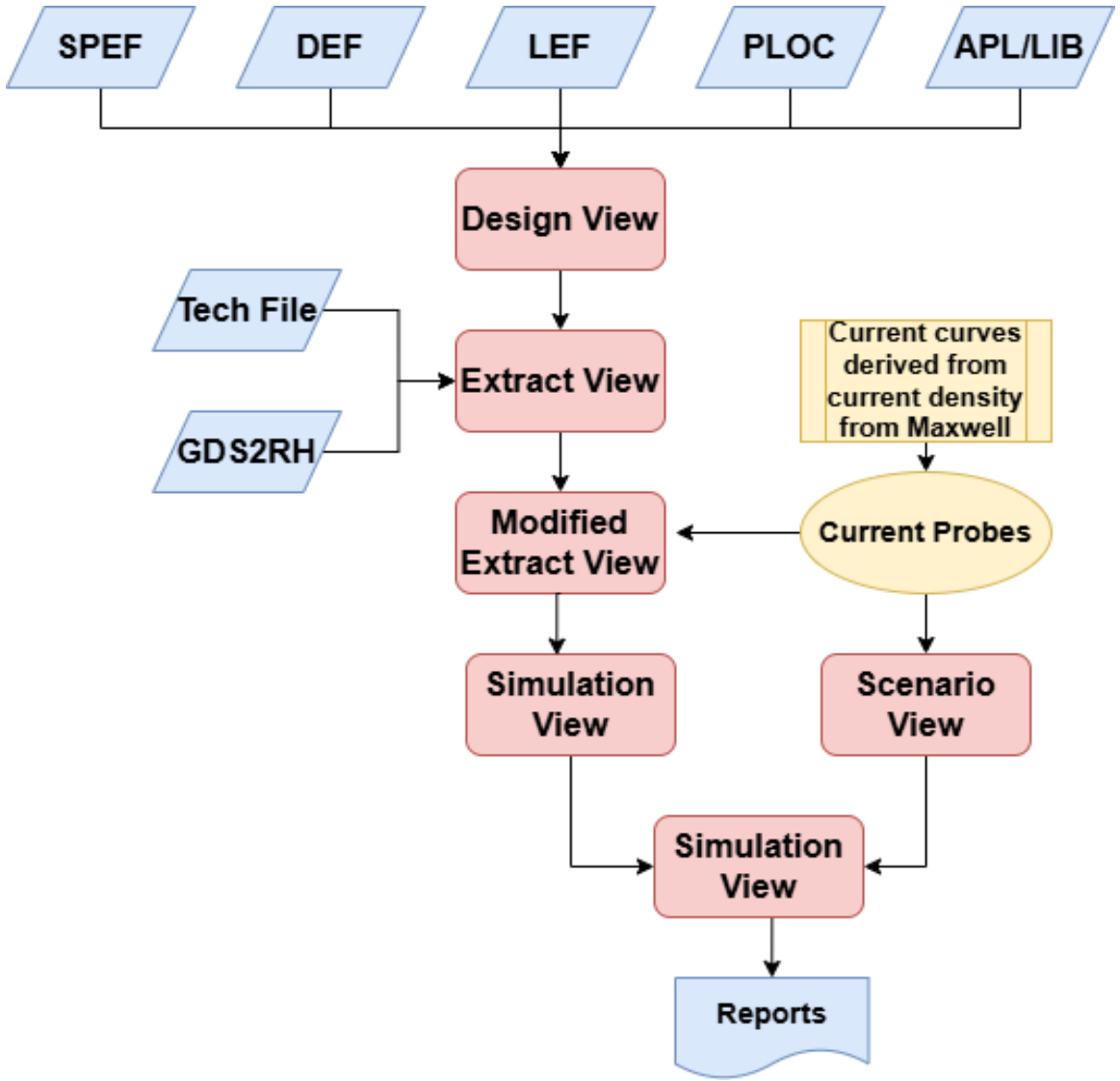
Model	Maxwell Simulation Time (s)
Full Geometry	334
Cutout Region	126

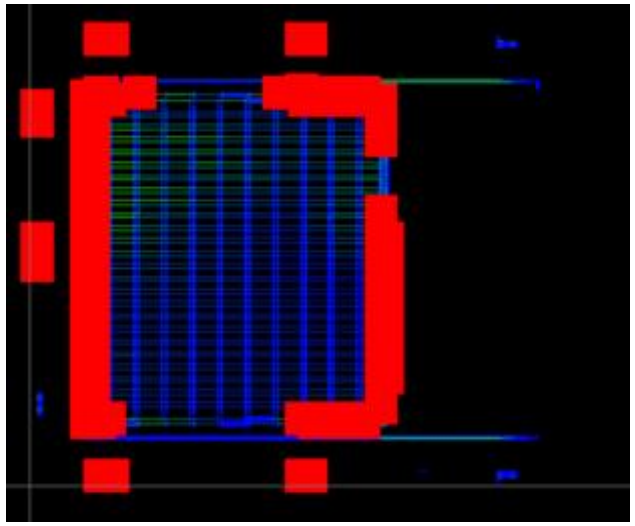


MagJ_Field

max(MAgJ) = 1.5E+12 A/m^2

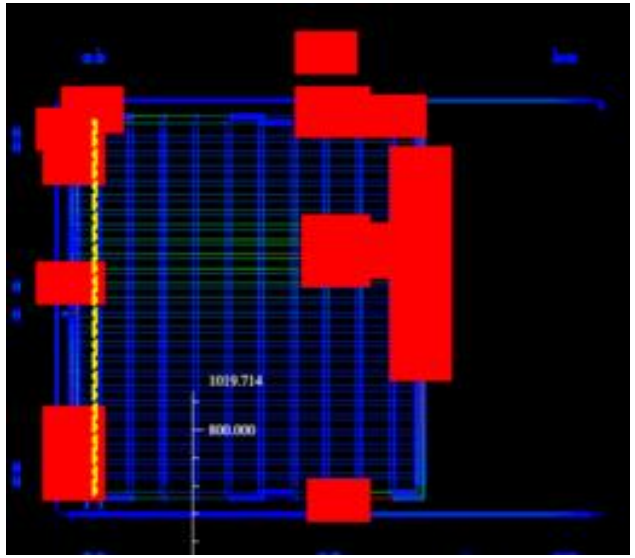
Layout Based Dynamic Current Voltage Analysis





Edge current
M6 : 0.0674 A
M5: 0.0214 A

Analysis view
@ 5ns

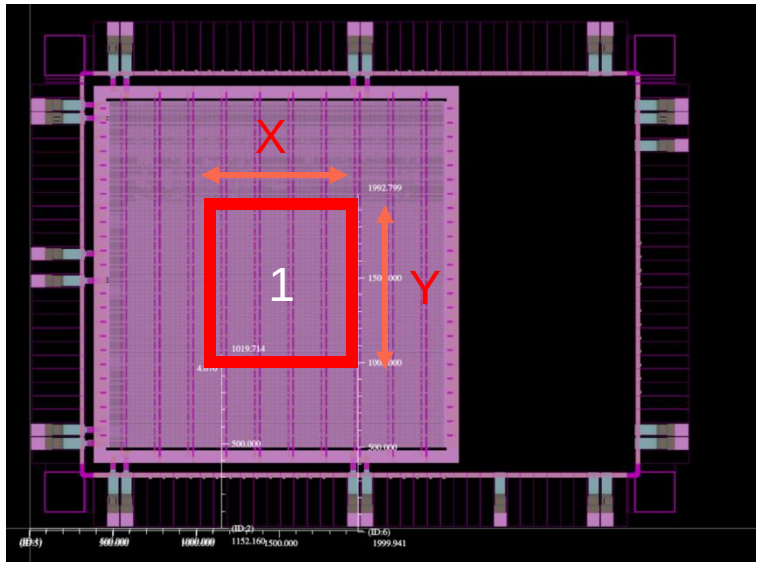


Edge current
M6 : 0.0274 A
M5: 0.0158 A

Analysis view
@ 10ns

Figure of Merit Based Vulnerability Analysis

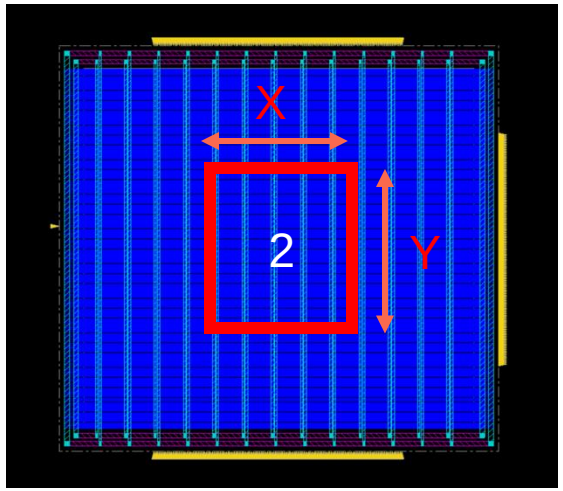
$$\text{FOM} = \frac{\sum_{i=1}^{n} R_i I_i}{\frac{dI_{Coil}}{dt}}$$



Layout 1

FOM_1 = 6.02x10^-8 Ω-s

FOM_2 = 8.39x10^-8 Ω-s



Layout 2

Future Work

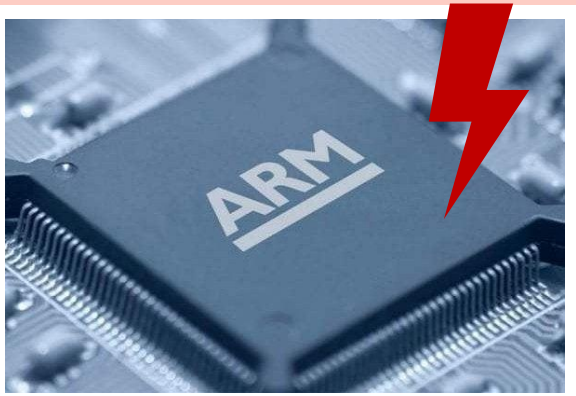
- Conduct real-world measurements using electromagnetic coils
- Validating simulation accuracy

FLAT: Layout-aware Timing Fault-Injection Attack Assessment Against the Violation of SoC Security Properties

Motivation

- Crypto Engines and other SoC components are Prone to FIAs
- Existing Countermeasures have redundancy
  - ➢Power, performance and area overheads
- Existing timing FIA assessment frameworks-
  - ➢ Focus only on RTL and gate level
  - ➢ With no timing (RTL) or incomplete timing (Gate-level)
- Physical design steps changes device timing significantly
- Estimating post-silicon unpredictable factors require consideration of probabilistic model of timing FIA instead of statistical analysis with specific pattern

A Real-world Example of FIA on the Boot ROM of ARM Cortex-A SoC

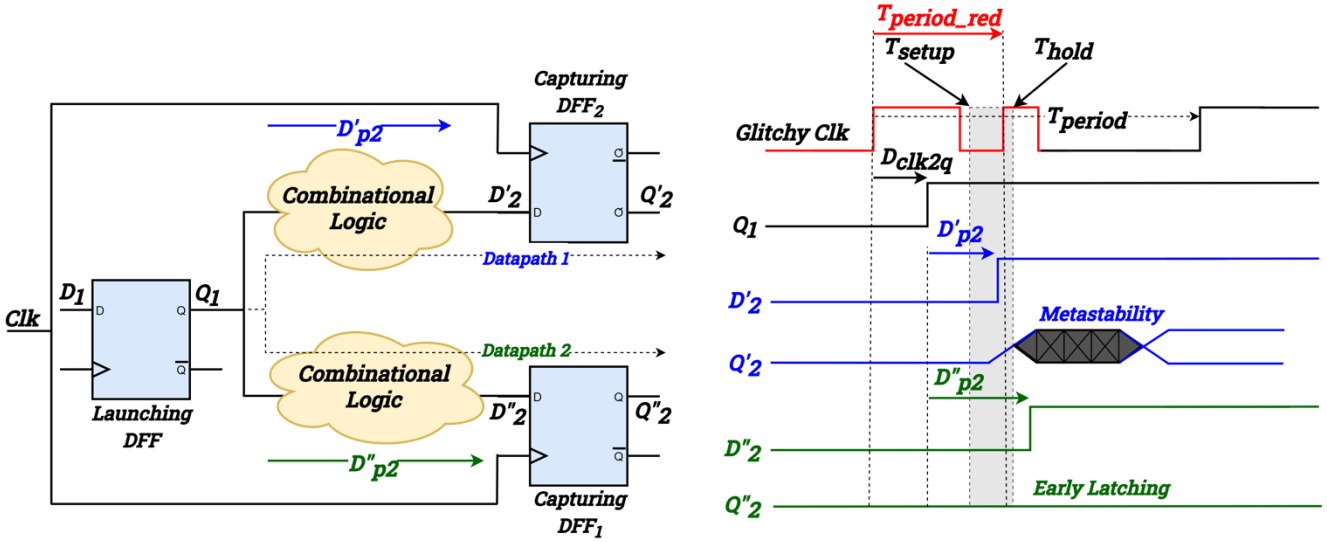


Source: https://research.nccgroup.com/2020/10/15/theres-a-hole-in-your-soc-glitching-the-mediatek-bootrom/

A Real-world Example of DFA on AES

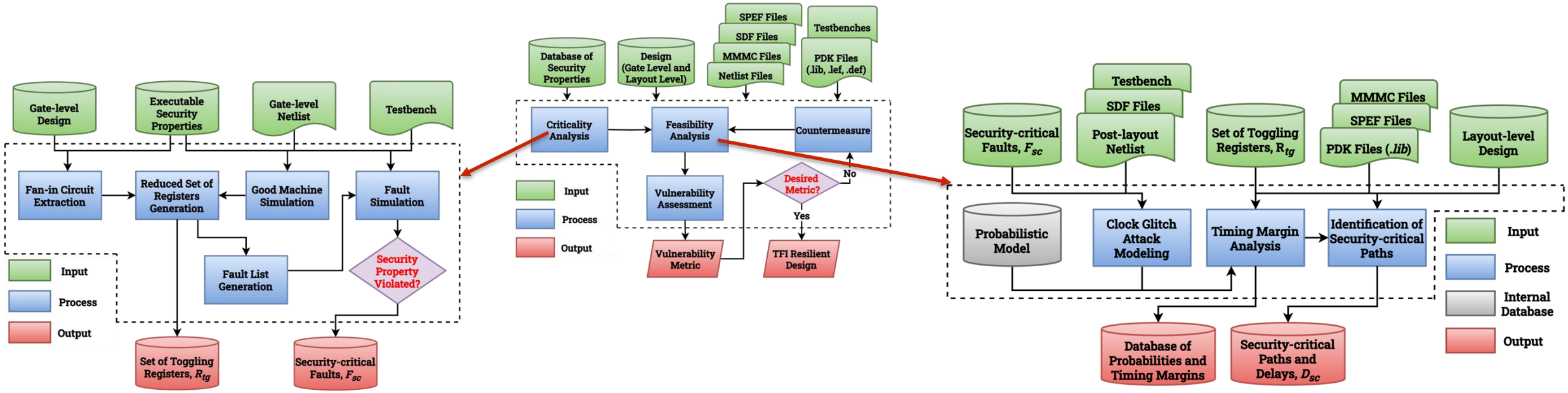
Source: https://semiengineering.com/security-highlight-exploiting-persistent-faults-in-crypto/

Threat Model



- Timing fault injection by clock glitch
  - Setup-time violations or early latching
  - ***Metastability*** induces bit-flip faults
  - Successful fault propagation
- Extraction of the secret key of crypto modules (AES, RSA)
  - Differential fault analysis (DFA), premature encryption by access control
- Malicious modification of the configuration bits of an FPU

Proposed FLAT Framework



Our proposed FLAT Framework is More Scalable, Generalized and Efficient

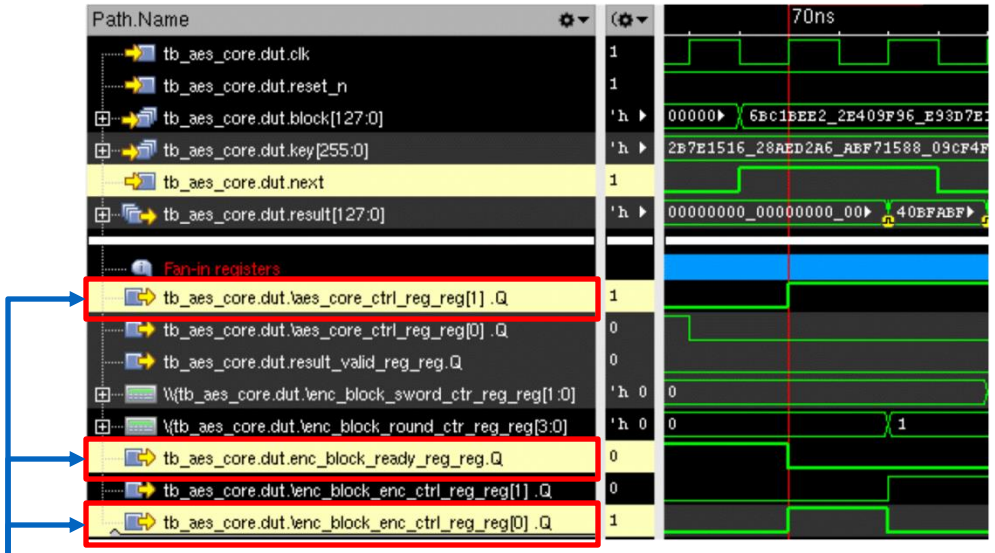
Main Contribution

- Considers several SoC security properties (confidentiality, integrity and access control)
- Utilizes a probabilistic model of timing fault injection during feasibility analysis and vulnerability assessment
- Implements local countermeasures more efficiently
  - Reduced number of path adjustment

Criticality Analysis

Identifies Two Sets of Registers Using Gate-level Fault Simulation

- Security-critical register, R_{sc} violate a security property when experience a bit-flip fault
  - *Important for security*
- Some other registers, R_{nsc} that toggles at the same time of a security-critical register
  - *Important for desired functionality*



Good Machine Simulation (No Fault)

3 Toggling registers, R_{tg}= 2 R_{sc} + 1 R_{nsc}

Fault Simulation

Violation of a property indicates that faults are induced at the security-critical registers

Feasibility Analysis and Vulnerability Assessment

Layout-Aware Fault Simulation With SDF Annotation

- Identifies the sweet-spot for an attacker
  - Bit-flip fault at a security-critical register, R_{sc}
  - No fault at the other toggling registers, R_{nsc}

Vulnerability Metric, V_{TFI} Depends on the **Probability** that the glitches can inject a bit-flip fault at least at a security-critical register, R_{sc} and no fault at any other toggling register, R_{nsc}

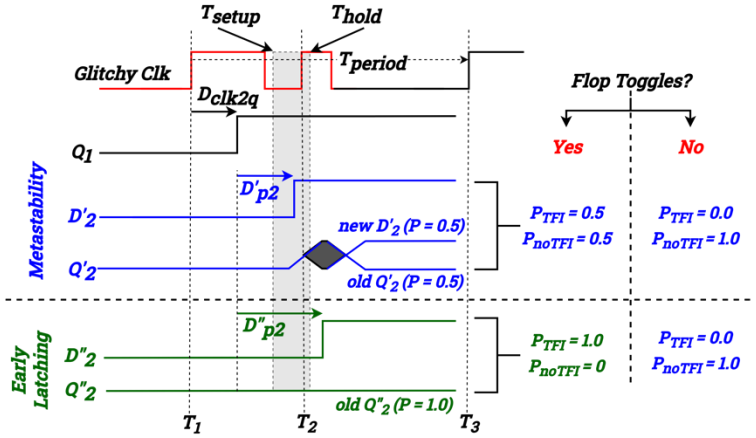
$$P_{\text{vsp}} = \sum_{f=1}^{N} P_{\text{vspf}} = N \times P_{\text{sc}}^{(\text{fault})} \times P_{\text{nsc}}^{(\text{noFault})}$$

$$= N \times \prod_{i=1}^{\text{len}(R_{\text{sc}})} P_{\text{TFI}}(i) \times \prod_{j=1}^{\text{len}(R_{\text{nsc}})} P_{\text{noTFI}}(j).$$

Vulnerability metric for each security property of an SoC **(More Scalability)**

$$V_{\text{TFIp}} = \sum_{t_{\text{gl}}=t_{\text{glMinp}}}^{t_{\text{glMaxp}}} P_{\text{vsp}}(t_{\text{gl}}) \times G_r.$$

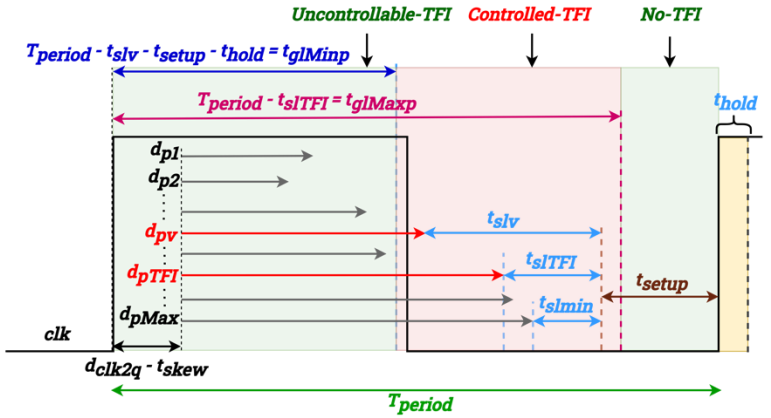
Probabilistic Model of TFI



No-TFI → $P_{vsp} = 0$,

Controlled-TFI → $0 < P_{vsp} \leq 1$

Uncontrollable-TFI → $P_{vsp} = 0$



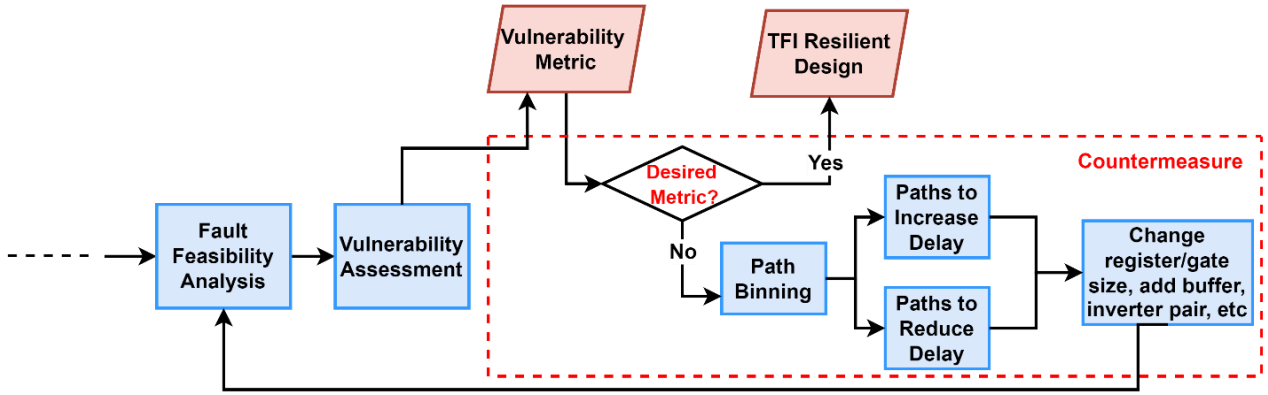
Local Countermeasures

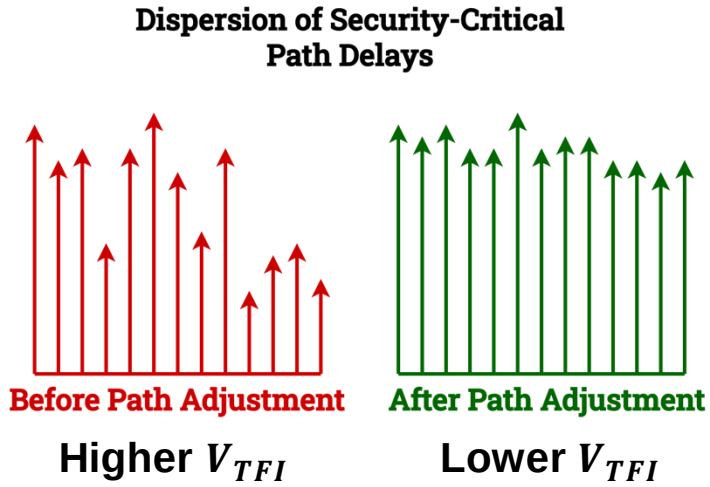
Adjusting physical design parameters and timing (gate resizing, adding buffers/ inverter pairs) of the post-layout design

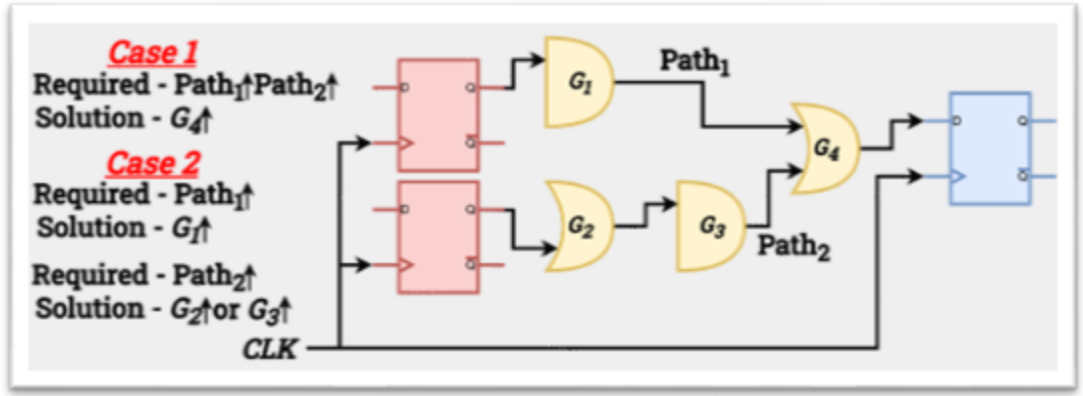
An attacker will get confused by the interference of the impacts of multiple timing faults

- To increase the probability of *uncontrollable TFI*
- To reduce the dispersion of the paths within the margins (t_{glMaxp} and t_{glMinp}) of controlled-TFI
- To reduce the vulnerability metric, V_{TFI}

Countermeasure application flow







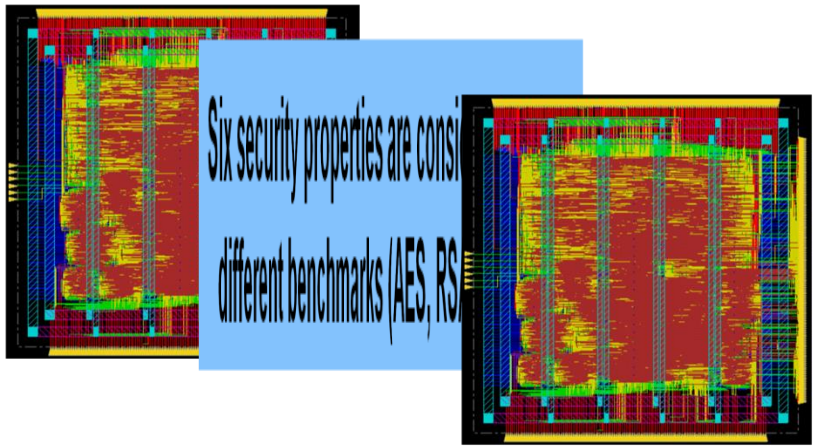
Experiments

Toolchain Used From ***Cadence***

- Gate-level synthesis (Genus Synthesis Solution)
- Layout design (Innovus Implementation System)
- Static timing analysis (Tempus)
- Post-layout SDF simulation (Xcelium Fault Simulator)

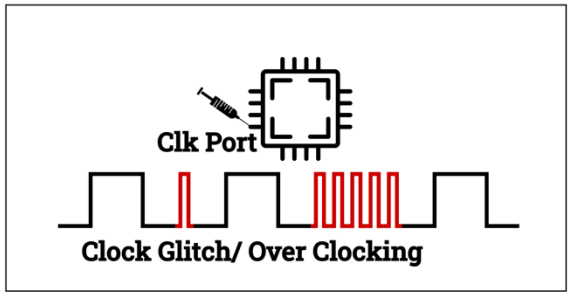
Six security properties are considered for three different benchmarks (AES, RSA, $32bit$ FPU)

Evaluation of the FLAT framework on layouts of three benchmarks (AES-128, RSA-128, $32bit$ FPU)

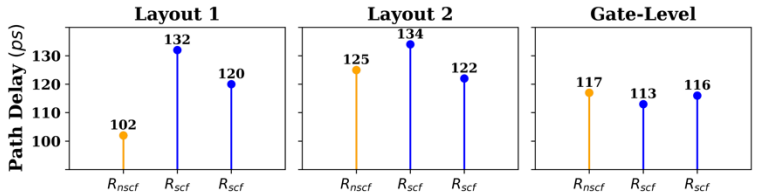


We Use

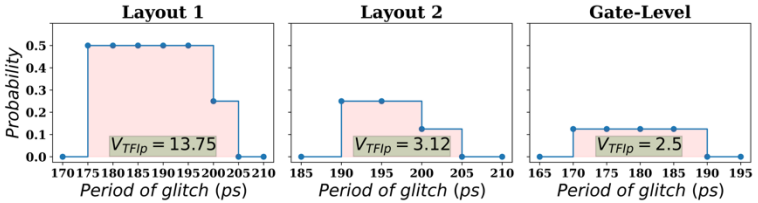
Fault Simulation-based Clock Glitch Injector



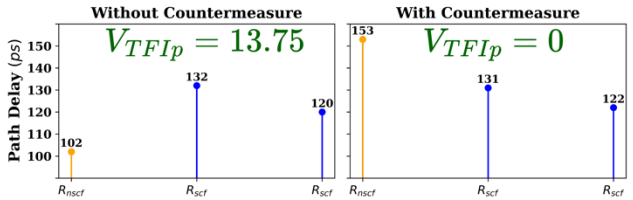
Result Analysis



Distributions of the delays of security-critical paths associated with a property of two different layouts and gate-level design of AES-128



Vulnerability to clock glitch-induced TFI of two different layouts and gate-level design of AES-128 with respect to a security property



Two distributions of the delays of security-critical paths associated with a property of AES-128 before and after path adjustments.

Inaccurate timing scenario at the gate level gives inaccurate vulnerability

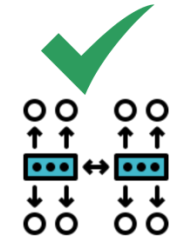
Different timing scenarios of different layouts provide different vulnerabilities

RESULT OF VULNERABILITY ASSESSMENTS

Layout Design	AES-128		RSA-128		32bit-FPU	
Instance count	16319		133411		9431	
Security Property	*SP1.1*	*SP1.2*	*SP2.2*	*SP2.3*	*SP3.1*	*SP3.2*
Simulation time (s)	15.8	10.6	323.7	523	64.9	51.4
Without Countermeasure						
Span of *Controlled-TFI*	30ps	20ps	100ps	1140ps	105ps	85ps
V_{TFIp}	13.75	8.04	31.04	269.35	94.17	76.25
Area (μm^2)	47294		374223		20833	
Power (mW)	5.898		17.521		1.415	
# Security-critical paths	3	68	9	76	4	5
# Paths adjusted	1	1	2	5	1	2
# Gates Added/Resized	29		295		17	
With Countermeasure						
V_{TFIp}	0	0	0	0	0	0
Area (μm^2)	47389		375350		20900	
Area Overhead	0.20%		0.30%		0.30%	
Power (mW)	5.904		17.593		1.419	
Power Overhead	0.10%		0.41%		0.28%	

Reduced number of paths adjusted

Vulnerabilities Reduced to **0** with Local Countermeasure
Negligible **Area** Overheads → < 0.5%
Negligible **Power** Overheads → < 0.5%
No Additional Sequential Cell introduced → No Change of Latency → 0% **Performance Overhead**



ASIC Design Flow

Electronic Design Automation (EDA)

Minimal Power, Performance and Area Overheads

Key Takeaways From Layout-Aware FIA Assessments

- It is recommended to consider post-silicon unpredictable factors affecting the device timing during SDF annotated fault simulation
- Feasibility analysis must be performed at different operating conditions (delay corners) to observe the change in vulnerabilities
- Path delay adjustments must be managed in a way that no timing violations occur in a fault free design
- Commercial timing sign-off EDA tools show their capability in precise estimation of post-silicon device timing under different delay corners
- The workflow of the layout-aware assessments are integral part of pre-silicon security verification → No additional steps in ASIC design and verification flow

Conclusion and Publications

Conclusion and Future Work

- **Conclusion**
  - **FAME aims to provide efficient fault-injection assessment and mitigations across various pre-silicon design stages**
  - **We have developed both assessment tools and countermeasures including:**
    - Security-property-driven flow to localize fault-Injection vulnerabilities at RTL and gate-level
    - ML-assisted laser fault-injection assessment at layout level
    - ML-driven pre-silicon EM fault-injection evaluation
    - Layout-aware timing fault-injection attack assessment and countermeasures
- **Future work**
  - Integrate the RTL/gate-level FIA assessment flow with local countermeasures
  - Expand the layout level laser assessment framework by modeling more attack scenarios and more side-effects of laser impact.
  - Validate both of our laser and timing fault-injection assessment results on FPGA, which also helps us improve the physical models and build more robust and scalable countermeasures.

Publications

- H. Wang, H. Li, F. Rahman, M. Tehranipoor, and F. Farahmandi, “SoFI: Security Property-Driven Vulnerability Assessments of ICs Against Fault-Injection Attacks,” IEEE Transactions on Computer-Aided Design (TCAD), 2021.

- N. Pundir, H. Li, L. Lin, N. Chang, F. Farahmandi, and M. Tehranipoor, “SPILL: Security Properties and Machine Learning Assisted Pre-silicon Laser Fault Injection Assessment,” International Symposium for Testing and Failure Analysis (ISTFA), 2022.

- A. M. Shuvo, N. Pundir, J. Park, F. Farahmandi, and M. Tehranipoor, “LDTFI: Layout-Aware Timing Fault-Injection Attack Assessment Against Differential Fault Analysis,” IEEE Computer Society Annual Symposium on VLSI (ISVLSI), 2022.

- M. R. Muttaki, M. Tehranipoor, and F. Farahmandi, “FTC: A Universal Fault Injection Attack Detection Sensor,” IEEE International Symposium on Hardware-Oritented Security and Trust (HOST), 2022

- H. Li, S. Dey, and F. Farahmandi, “Physically-Aware Laser-Fault Injection Assessment”, GOMACTech, 2023.

- A. M. Shuvo, T. Zhang, F. Farahmandi, and M. Tehranipoor, “FLAT: Layout-Aware and Security Property Assisted Timing Fault-Injection Attack Assessment,” IEEE Transactions on VLSI (TVLSI), 2024.

- M. R. Muttaki, M. H. Rahman, A. Kulkani, M. Tehranipoor, and F. Farahmandi, “FTC: A Universal Framework for Fault-Injection Attack Detection and Prevention,” IEEE Transactions on VLSI (TVLSI), 2024.

Thank You!