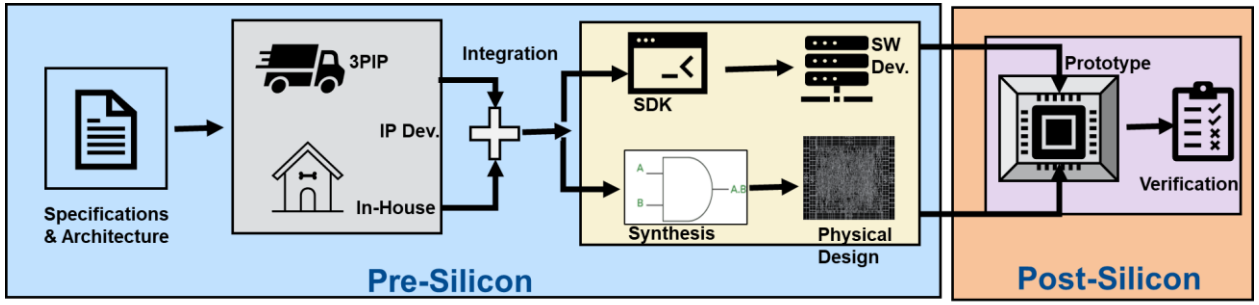
EmFIA: A Novel Emulation-based Fault Injection Vulnerability Assessment Framework at RTL Level

Farimah Farahmandi, PhD
Associate Professor
Department of Electrical and Computer Engineering
University of Florida

Background

- Physical attacks such as fault injection attacks (FIA) can reveal design vulnerabilities in a functionally correct designs.
- Errors are intentionally injected in a system to compromise the security of the design and facilitate the leakage of assets in the system.
- Attack surfaces are ever growing/ evolving rapidly, tools are readily available
- Real World examples: RSA Smart card attacks, Rowhammer Attack. Spectre/Meltdown Attack, Pager Attacks.
- Traditional verification approaches verify each layers (HW or SW) separately.
- HW/SW co-verification relies mostly on protype completion (leaves room for HW induced SW vulnerabilities and vice-versa).
- "**Rule of 10**" applies for the cost of re-design and verification.



SoC Design & Verification Flow

Vulnerability Assessment Against Fault Attacks

- **Fault Injection Attack Types:**
  - Non-Invasive Attacks, e.g., clock glitching, power glitching
  - Semi-Invasive Attacks, e.g., laser-based fault injection
  - Invasive Attacks, e.g., focused ion beam (FIB) based FIA
- Current countermeasures, e.g., hardware/time redundancy or error correction may involve 100% or more area or timing overhead.
- Not every design component is vulnerable, not every design component needs protection.
- Pre-si vulnerability assessment is very important against FIA
- **Pre-Si Security verification against physical fault attacks are typically done at:**
  - Layout-level simulation (e.g., glitch propagation analysis)
  - Gate-level simulation for setup/hold violation modeling
  - FPGA-based emulation, often manual or black-boxed

Existing Limitations and Our Research Objective

Key Limitations:

- Slow and non-scalable: fault simulation is computationally expensive.
- Requires late-stage artifacts (e.g., GDSII, netlist) — limits early-stage design insights.
- Limited fault models
- Poor integration with formal security properties and system behaviors.

Feature	SoFI/FLAT	FPGA-Based FI
Abstraction	Gate/Layout	Netlist/Bitstream
Fault Control	Setup-time	Bitstream modification
SP Integration	Weak	No
Runtime	Hours+	Minutes+
Automation	Low to moderate	Low

- **Objective:**
  - To develop an RTL emulation-based FIA framework.
  - To define realistic fault models (FIB-style stuck-at).
  - To enable property-aware detection using System Verilog Assertions.
  - To achieve significant speed-up over simulation.

Motivation for Emulation-Driven Assessment at RTL

- **Emulation at RTL: A Balanced Tradeoff:**

RTL-level emulation offers a powerful combination of:

- Cycle-accurate signal control
- High observability of design internals
- Execution speed suitable for full-system validation

- **Advantages over Layout- and FPGA-Level Analysis**

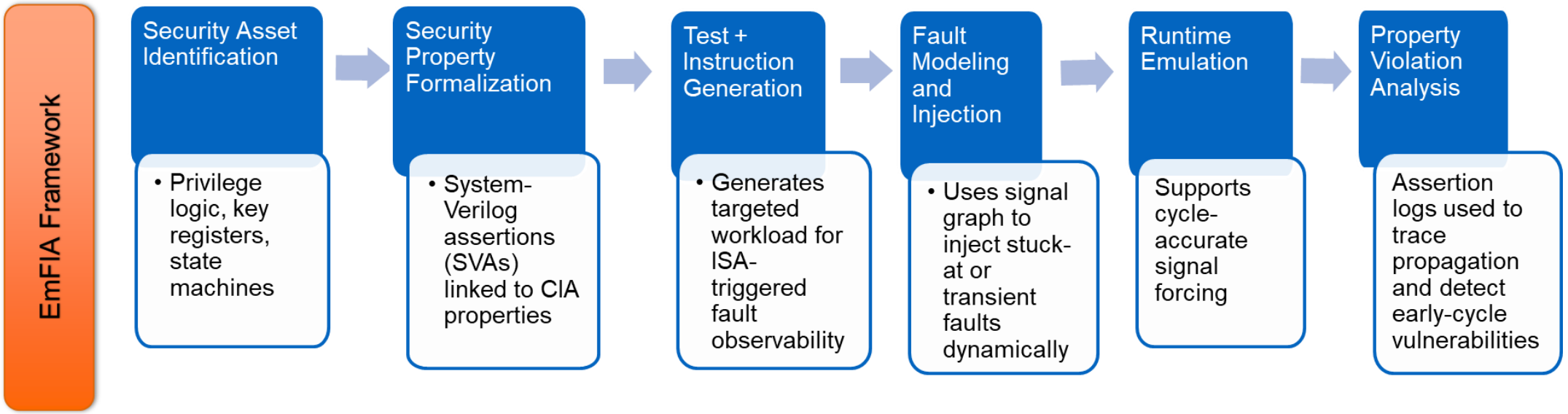
- No long synthesis or place-and-route dependency
- Preserves high-level design intent
- Supports precise spatial and temporal fault injection
- Enables SVA monitoring

Emulation delivers finer control, better automation, and early actionable insights into design-level security flaws

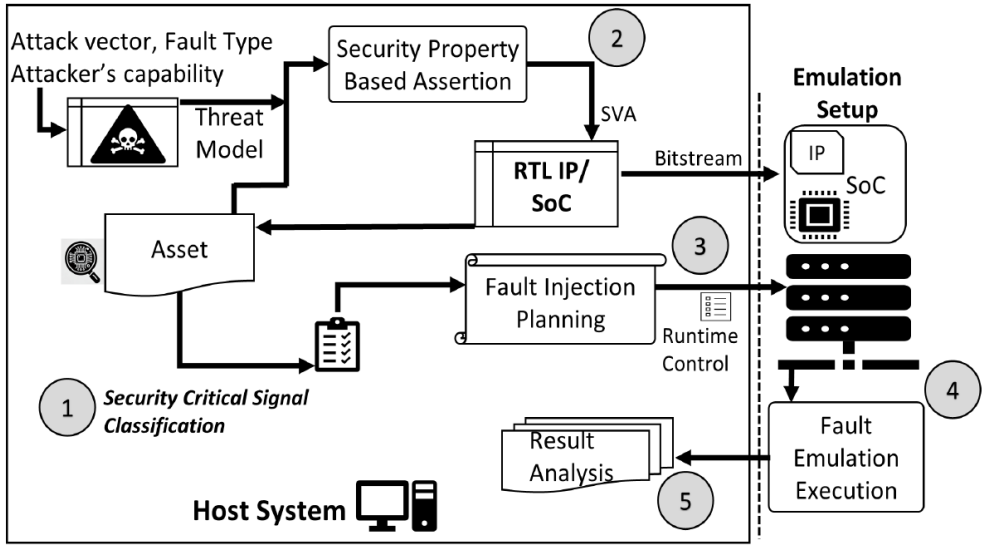
Pre-Requisite: Threat Model & Fault Model

Threat Model	Fault Model
Pre-silicon adversary with **RTL access** (not silicon).	**Stuck-at faults** at RTL (SA-0, SA-1).
Goal: break **Confidentiality, Integrity, Availability (CIA)**.	Models **FIB-style permanent disruptions**.
Attacks via **permanent modifications** (e.g., FIB).	Fault persists once injected, unless released.
Targets: **registers, control paths, interrupts**.	Injected into **security-sensitive signals** only.
Observes RTL signal propagation to find violations.	One fault per experiment (for tractability).

Framework Overview



- **Platform:** Synopsys ZeBu Server
- **Host PC:** Compiles RTL & testbench (VCS + Vivado); sends fault schedules.
- **Emulator**: Runs cycle-accurate RTL bitstream, injects faults at runtime, monitors assertions.
- **Logs**: Assertion violations and timing contexts returned to host for analysis.
- **Benefits:** No re-synthesis needed for each fault. high speed, full observability with RTL granularity.



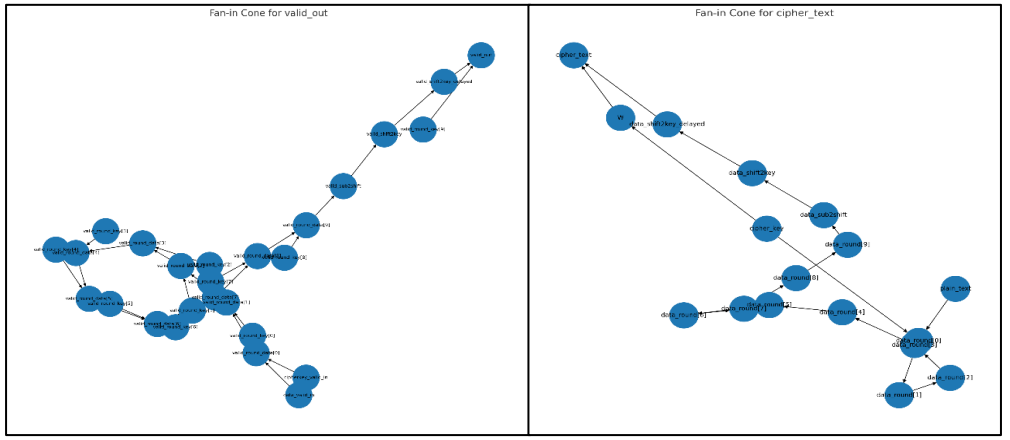
Benchmarks and Security Properties

SP	Design	Security Asset	Security Property Description	Violation Type
SP-1a	AES-128	Plaintext	The output ciphertext must be the valid encrypted form of the input plaintext once the encryption process completes.	Data Integrity
SP-1b	RSA-128	Plaintext	Same as above.	Data Integrity
SP-2	AES-128	Ciphertext	Ciphertext should be available for use when the encryption completes.	Availability
SP-3	RSA-128	Plaintext	Done signal should be asserted at the end of encryption.	Availability, Integrity
SP-4	RSA-128	Operation Validity	Transition from FSM state *Start* → *State_1* must not be immediately followed by a transition from *State_1* → *Done* that bypasses intermediate state *Check_condition*.	Integrity
SP-5	RSA-128	Operation Validity	The FSM must not transition from 1st internal state to *Done* if the checking condition is false.	Integrity
SP-6	RSA-128	Process Completion Integrity	The *Done* state must be reached exclusively from FSM internal state which checks for encryption completion ensuring that only valid transitions lead to completion.	Integrity, Availability
SP-7	NEORV32 SoC	Access Control	A user space application should never have access to the *Machine mode*.	Confidentiality, Integrity
SP-8	NEORV32 SoC	Program Flow Control	The program counter (PC) must remain within the valid instruction memory (IMEM) address range.	Confidentiality, Integrity

Emulation Results Summary and Fan-in Analysis

Design	Security Property	Violation Cycle in Emulation	Observation
AES-128	SP-1a / SP-2	56	Integrity & availability fail in round control logic and key registers.
RSA-128	SP-3, SP-4, SP-6	8 – 139	FSM transition errors during Montgomery steps.
NEORV32 SoC	SP-7, SP-8	153 / 202	Privilege escalation and illegal PC access.

- Runtime: < 6 s per campaign (vs. minutes to hours for simulation)
- **Fan-In Analysis: AES:**
  - SP-2: Ensure ciphertext available after encryption (Availability Property).
  - **Process**: Fan-in cone extraction from selected security critical signals → 41 nets → 23 candidate fault sites.
  - Single-Site Injection → 9 of 11 final-round signals caused SP-2 violation.
  - Faults injected in paths identified in static analysis triggers same violations, revealing complex interdependencies between internal signals and security invariants.



Conclusion and Future Works

- **Key Takeaways:**
- EmFIA introduces a novel RTL-level fault injection assessment framework driven by hardware emulation offering:
  - Cycle-accurate fault injection on emulation platforms.
  - Property-aware detection via SystemVerilog Assertions linked to CIA (Confidentiality, Integrity, Availability).
  - Rapid fault-space exploration with speedups of several orders of magnitude.
  - Shifts verification earlier into the RTL phase, where design fixes are low-cost and practical.Enables quantifiable, repeatable, and scalable security validation
- **Future Directions**:

Transient fault models, CWE Guided Automated Vulnerability Analysis, Cross-Layer Fault Propagation Studies

THANK YOU!!!

Questions??