Hierarchical EMFI Analysis on a RISC-V SoC

Dillibabu Shanmugam Zhenyuan Liu Thei Riley Patrick Schaumont

***Abstract*—Electromagnetic fault injection (EMFI) can induce transient faults in SoCs. Such faults are often hard to understand and analyze due to limited hardware observability. We present a scan-chain-assisted EMFI analysis on a custom RISCV system-on-chip (SoC) called `CAPRI1`. The scan chain offers full visibility of the impact of EMFI on the SoC state. We then perform hierarchical fault analysis and follow the fault propagation from initial bit-flip through the SoC micro-architecture into the high-level software behavior. We evaluate the method with EMFI experiments on two PIN-verification firmware workloads. From scan captures, we classify faults by source component and by system-level outcome. We also relate the initial bit-flip (x,y) coordinates to layout-level features. This framework improves understanding of design behavior under EMFI fault scenarios and offers early vulnerability assessment.**

I. Introduction

Fault injection attacks exploit the fact that integrated circuits are not perfectly robust under all operating conditions. By perturbing supply voltage, clock, temperature, or the electromagnetic environment, an adversary can force a secure embedded system into erroneous states that violate control-flow or data-flow integrity. These errors are then analysed to extract secrets or bypass security checks.

Electromagnetic fault injection (EMFI) is particularly attractive to attackers because it can be performed contactlessly, often without invasive modification of the target device. A short, high-intensity current pulse through a small coil placed close to the die induces eddy currents and transient voltages in the chip, which can lead to timing violations and bit upsets. Prior work [1] has shown that carefully timed EMFI can cause instruction skips or arithmetic faults in microcontrollers.

Most published EMFI studies work in a black-box setting. The internal state of the device is not directly observable; only external behaviour such as program output, error flags, or crashes is recorded. To explain the observations, the analyst postulates a fault model—for instance, a single-bit flip in a register, or the replacement of one instruction by another—and tries to match it to the data. In reality, the effect of a physical disturbance may be more complex: multiple bits may flip simultaneously, or the device may visit states that cannot be reached in normal operation. Without visibility into the internal state, these "weird" behaviours [2] remain opaque, complicating validation of countermeasures.

This work addresses that visibility gap using a system-on-chip derived from CROC [3] called `CAPRI1`. `CAPRI1` integrates a microcontroller core, memories, peripherals, and, crucially, a full scan chain that connects all 12756 flip-flops. The scan infrastructure allows us to halt the device after a fault injection and dump the full internal state. By comparing these scan dumps against a fault-free reference as depicted in Fig. 1, we can pinpoint the first bit-flip in hardware, track its propagation over subsequent cycles, and relate it to the eventual software-level outcome.

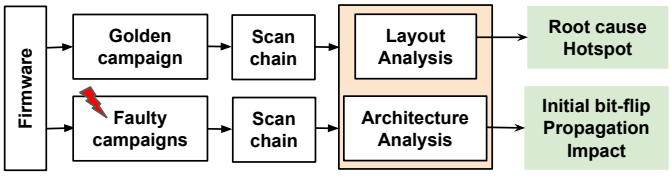

Fig. 1. Scan-assisted EMFI analysis flow on `CAPRI1` The golden campaign and faulty campaigns each produce scan-chain dumps, which are processed by two analysis layers: an architecture-level layer (fault propagation) and a layout-level layer (initial bit flip and physical hotspot localisation).

First, we study how EMFI affects `CAPRI1` as a piece of hardware: magnitudes and locations of bit flips, common patterns, and correlations with physical structures. Second, we analyse how the same hardware faults interact with two firmware workloads, denoted `verifyPIN0` (VP0) and `verifyPIN7` (VP7) firmware [4], leading to either exploitable behaviour or benign no-effect.

The main contributions of this paper are:

- A scan-assisted EMFI framework on `CAPRI1` that provides cycle-resolved, bit-level visibility of the internal state across all 12 756 flip-flops after injection.
- A hierarchical fault analysis that relates physical fault manifestations to architectural effects through a study of the VP0 and VP7 firmware workloads under EMFI.
- Demonstration of the impact of long PDN loops excited by EMFI align the pre-silicon loop model with post-silicon register bit-flip observations.

The remainder of this paper is organized as follows. Section II describes the `CAPRI1` SoC, the EMFI experimental setup and the fault injection methodology. Section III develops a hierarchical fault analysis that relates physical fault manifestations to architectural effects through an application study of the VP0 and VP7 firmware workloads under EMFI. Section IV links pre-silicon and post-silicon findings by showing that long PDN loops in the top metal layers couple strongly to the EMFI probe, inducing transient loop currents that create a local supply voltage drop. This voltage droop reduces noise and timing margins, explaining the first bit flips observed in the scan-chain data. Section V concludes the paper.

II. `CAPRI1` SoC and EMFI Experimental Setup

`CAPRI1` is a 32-bit RISC-V microcontroller SoC based on the Ibex (CV32E20) core, developed as a fault injection test platform. The chip includes two on-chip SRAM blocks using DFF(128 words each, for instruction and data memory) and

TABLE I
GROUPED FLIP-FLOP DISTRIBUTION IN CAPRI1

Category	Flip-flops
Peripherals	2733
GPR	992
μarch: (Fetch: 283, Decode: 14, CSR: 420, LSU: 67)	784
Memory: (IMEM: 4129, DMEM: 4129)	8258

standard peripherals (timer, UART, GPIO, and control/status registers), all accessed via a memory-mapped on-chip bus. The design was fabricated in a 0.18 µm CMOS process, with a core area of about 2 mm × 2 mm and a 1.8 V supply voltage. A JTAG-based debug unit provides program loading, breakpoints, and execution control. CAPRI1 also incorporates full design-for-test (DFT) features (notably a comprehensive scan chain and on-chip debug interfaces), enabling precise control of execution and complete observation of the entire internal state. In the following, we describe the SoC's architecture and scan chain, highlight key physical layout aspects, and detail the EMFI experimental setup and triggering methodology.

A. Architecture Hierarchy and Scan Chain

CAPRI1's processor employs a two-stage pipeline (instruction-fetch and decode/execute) implementing the RV32IMC RISC-V instruction set. The architecturally visible state includes the 32 general-purpose registers (GPRs), the program counter (PC), the control and status registers (CSRs), and memory-mapped I/O registers. In contrast, micro-architectural state (e.g., prefetch buffers, pipeline registers, internal control signals) is not visible to software. All sequential elements of the SoC (flip-flops in the CPU core, SRAMs, and peripheral interfaces) are connected into a single scan chain. This chain comprises 12756 flip-flops in total (Fig. 2 and Table. I). Each scan cell associate with a register instance name and physical (x, y) coordinate in the chip layout. The scan chain supports two modes:

- *Scan-load*: an arbitrary state can be shifted into the chain (or a previously captured state can be looped back) to initialize all flip-flops before executing a test program.
- *Scan-capture*: In this mode, after halting the system clock, the content of all flip-flops is shifted out for analysis.

This ability to initialize the device to a known state and later capture its entire post-execution state is central to our fault injection methodology. Mapping each observed bit from a scan dump to its known die coordinates allows us to correlate fault-induced state changes with specific regions of the chip. This capability greatly aids in diagnosing the effects of EMFI.

B. Physical Design and Layout Components

From a layout perspective (Fig. 3), CAPRI1 includes several features that influence its susceptibility to EMFI.

- **Power Distribution Network (PDN):** Top metal layers (M5/M6) form a grid-style VDD/VSS power network covering the entire core. The PDN uses core rings on M3/M4 and M5/M6 (4.75 μm width, 3 μm spacing, 8 μm offset). Dense M5 horizontal and M6 vertical straps (70 μm pitch) provide global distribution. M2–M4 carry fewer supply lines. Table II lists total wire lengths for supply (VDD, VSS), PDN, reset, and clock networks. VDD and VSS combined exceed 4.5 m of wiring, mainly on M4–M6, whereas M1 contributes minimally. Under EMFI, disturbances in the top metal cause localized supply fluctuations that may upset nearby flip-flops.
- **Clock Distribution:** The clock tree uses buffered trunk and leaf branches. Trunk segments have slews near 0.18 ns and leaves around 0.23 ns. Clock nets couple strongly to EMFI.
- **Reset Network:** Reset Network: A global active-low reset line spans mid- and low-metal layers and is synchronized to release with low skew relative to the clock; each flip-flop uses an asynchronous reset pin. Under EMFI, a local transient at this pin can reset a subset of flip-flops, disrupting state even when the synchronized reset logic remains deasserted.
- **Routing and Cell Placement:** Signals route mainly on Metals 1–4; Metal 3 contributes roughly 50–70% of total length and Metal 4 carries longer runs. High impedance

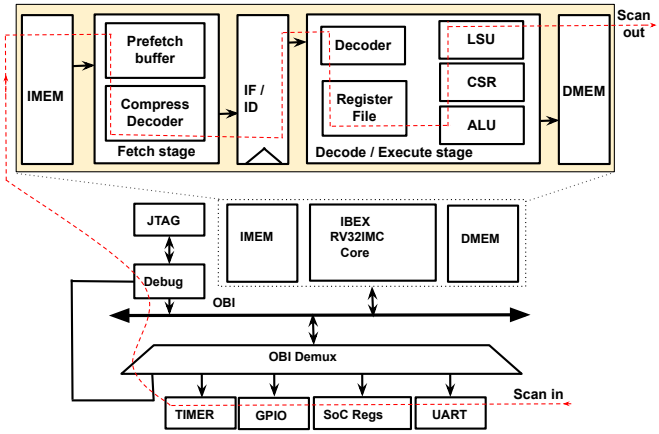

Fig. 2. Block diagram of the CAPRI1 SoC. The Ibex core connects through an OBI demultiplexer to instruction and data SRAMs and to MMIO peripherals. A single scan chain span all sequential elements, enabling full-state initialisation and readout under EMFI. This architectural organisation is the basis for the cycle-resolved, bit-level fault analysis in later sections.

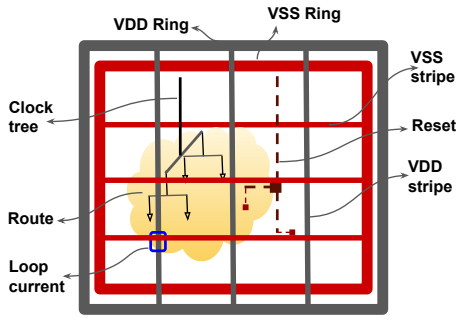

Fig. 3. Key CAPRI1 layout structures—power rings, stripes, and clock/reset routing—form conductive loops that determine the chip's susceptibility.

TABLE II
GLOBAL INTERCONNECT LENGTHS FOR SUPPLY, RESET, CLOCK, AND PDN, WITH PDN LENGTH BY METAL LAYER (ROUNDED UNIT).

Network	Total length	PDN layer	Length
VDD (all layers)	2,315,495	METAL6	934,752
VSS (all layers)	2,192,268	METAL5	1,104,051
RST	18,748	METAL4	982,077
CLK (clk_i)	7,858	METAL3	749,330
PDN*	4,507,763	METAL2	688,456
		METAL1	49,096

*Total PDN length equals VDD+VSS.

limits EMFI effects on signal nets.

C. EMFI Bench Setup and Triggering Methodology

To assess CAPRI1's vulnerability to electromagnetic fault injection (EMFI), we performed experiments on a test bench that allowed precise control of the injection parameters. A small coil probe, Fig. 4 was placed just above the chip's die using a computer-controlled micro-positioning stage. We used a high-speed pulse generator to deliver current pulses (amplitude up to ± 10 V, width 4–10 ns) through this coil. The coil's pulses generated a brief magnetic field that inductively coupled into on-chip structures (such as the power distribution network mesh and the clock/reset network), creating transient disturbances in the chip's voltages and currents.

A Raspberry Pi-based controller orchestrated each EMFI campaign (see Fig. 4). This controller provided the 1 MHz system clock, managed the reset line and JTAG interface, and coordinated the scan operations. Meanwhile, the CAPRI1 firmware toggled a GPIO pin as a trigger once per clock cycle, immediately before executing the instruction under test. In our experiments, the test program ran for N clock cycles. We therefore defined N separate fault-injection campaigns, each targeting a specific clock cycle of the program's execution. For example, campaign 10 targeted the 10th cycle. By aligning the EM pulse immediately after the clock's rising edge (Fig. 5), we ensured that the injection occurred outside the flip-flops' setup/hold window. This timing allowed the EM pulse to directly disturb the flip-flops' stored output value Q.

Once a fault is injected, the controller immediately switches to scan mode. In scan mode, it froze all flip-flops and shifted out their contents, thereby dumping the internal state. It then returned the chip to functional mode to execute the next clock cycle before switching back to scan mode. This sequence of one clock cycle followed by a scan dump continued until the program completed. We later compared the sequence of scan dumps from each injection campaign to reference traces from a fault-free execution of the program. By correlating the locations of flipped bits with the specific cycle targeted by the EM pulse, we identified which regions or circuit blocks of CAPRI1 were most susceptible to EMFI during that cycle. Thus, our systematic campaign-based analysis enabled us to gauge the SoC's fault tolerance and pinpoint specific blocks or interconnects that were particularly vulnerable to EM pulses.

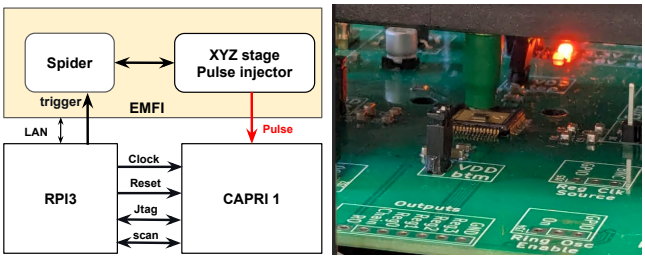

Fig. 4. Experimental EMFI bench. A Raspberry Pi 3 controls the CAPRI1 board, the pulse generator, and a motorised XYZ stage that positions the EM probe above the die. The controller provides clock, reset, JTAG, and trigger signals and collects scan-chain dumps after each injection. This setup enables automated spatial and campaign-based, cycle-resolved EMFI experiments.

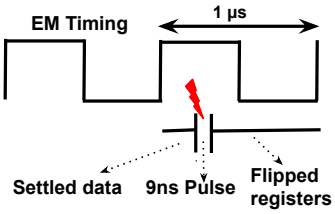

Fig. 5. Alignment of the system clock and EM pulse, illustrating the sampling window and the mechanism of an EM-induced bit flip.

III. HIERARCHICAL FAULT ANALYSIS

The CAPRI test chip integrates an Ibex RISC-V core with full-scan access, which exposes the internal flip-flop state at cycle granularity. This feature enables precise identification of the first corrupted element after a physical disturbance and allows tracking of its propagation through the microarchitecture. Rather than viewing each experiment only through the final program outcome, we reinterpret the EMFI campaigns using a hierarchical, component-centric perspective.

Hierarchical fault analysis studies fault propagation bottom–up. A physical disturbance first perturbs one or more microarchitectural bits (e.g., within the register file or pipeline latches). The resulting upset then travels across hardware components, eventually modifies architectural state, and becomes observable through software interaction with the ISA. This multi-layer view links a concrete physical effect to an architectural manifestation and, finally, to an application-level failure. To capture this chain, we adopt a *component-centric* fault classification instead of a purely output-based taxonomy. Traditional output-based classes (e.g., crash, reset, success, fail) characterise only the externally visible result and thus conflate many distinct root causes. In contrast, a component-centric view attaches each observed outcome to specific hardware blocks, which exposes which structures enable or block the success of a given fault.

A. Component-Centric Fault Classification in CAPRI

Our component-centric hierarchical analysis steps:

1) **Initial component upset:** identify the microarchitectural or architectural component that first devi-

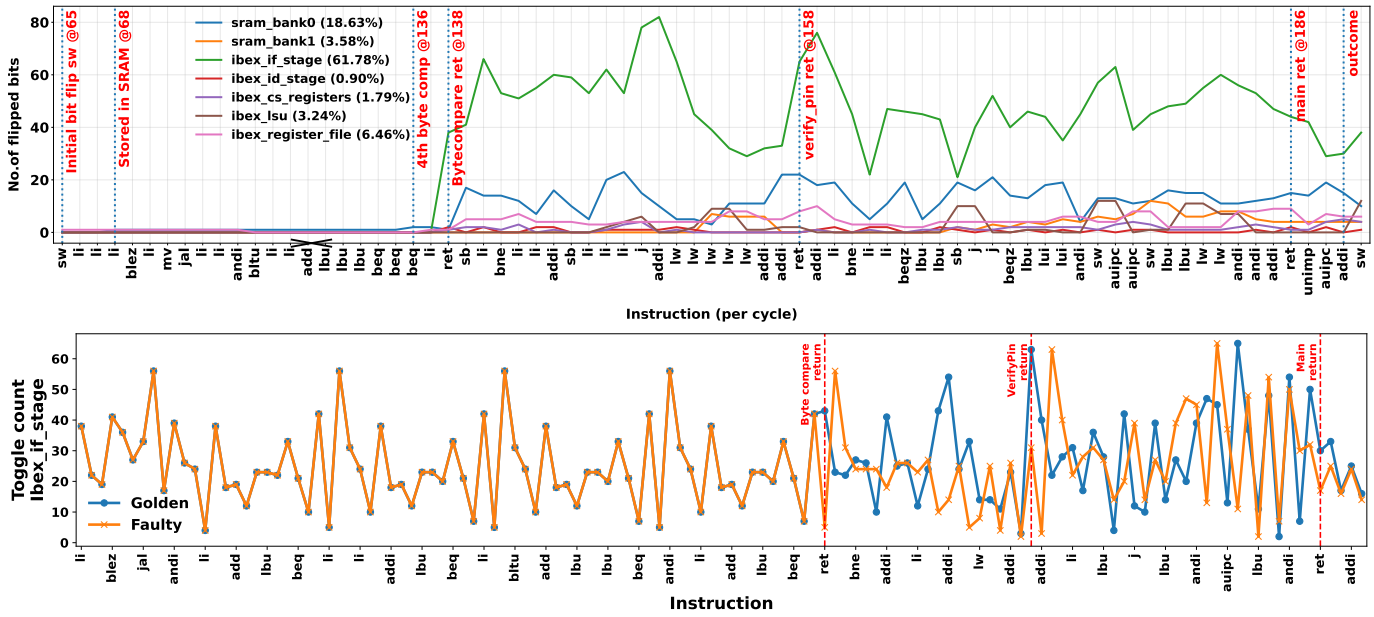

Fig. 6. Timeline of fault propagation for the register-file upset case. Top: number of flipped scan bits per instruction for major Ibex components. After the corrupted instruction is stored in `sram_bank0` and later refetched, the error concentrates in the fetch path: `ibex_if_stage` and `sram_bank0` dominate the flip count, while other blocks remain close to their golden behaviour, illustrating a narrow but effective propagation channel from the initial register-file upset to the final outcome. Bottom: toggle count per instruction for the golden and faulty executions of `Ibex_if_stage` component; both traces overlap until the corrupted instruction is fetched, after which they diverge around the branch event and the subsequent returns.

ates from the golden execution (e.g., a flip in `ibex_register_file`, a corrupted instruction word in `sram_bank0`, or a misaligned PC in the fetch stage).

2) **Propagation path across Ibex blocks:** reconstruct how this initial upset propagates through Ibex blocks and pipeline registers, including control-flow divergence and secondary data corruptions.
3) **Resulting system-level behavior:** characterise the resulting program behavior, such as abnormal termination, preserved control flow with wrong data, or complete bypass of a security check. At the high level the system behavior is identified as fault pattern [5] such as Detour, Tunnel, Spinner, Wormhole and soon.
4) **Component-centric outcome label:** assign a label that captures both the observed effect and the originating component, for example "register-file–originated data corruption" or "IF-stage–originated control-flow hijack".

We apply this hierarchical framework to two PIN-verification kernels executed on CAPRI: VP0, a naïve implementation that runs for 190 cycles, and VP7, a hardened implementation that completes within 140 cycles. A case studies of these firmware are summarized below in a precise, component-focused manner.

B. *Register File Upset:*

In the 65th EMFI campaign of VP0, we observe a register-file upset. We compare the golden and faulty executions and summarize our observations below.

Initial component: At clock cycle 65, a single-bit upset flips one bit in register `x15` of the `ibex_register_file`. The corresponding scan-cell coordinates (302.4, 731.92) identify the physical location of the disturbed flip-flop in the layout and thus the origin of this initial flip.

Propagation: The corrupted value remains architecturally valid until it is used as a store operand, which overwrites the instruction word at `IMEM[96]` in `sram_bank0` at cycles 68 and 135. When the program later fetches this location in the comparison loop, the corrupted instruction writes `0x04` into `x13` (`a3`), so that `a3` equals `a4` (`0x04`) at the final `beq`. Although the branch condition then holds, the faulty execution shows a one-instruction lag in the program counter due to a misaligned PC and invalid prefetch in the `ibex_if_stage`; the branch is not taken and the error moves from the data path into the control-flow path. The resulting divergence spreads through the fetch pipeline and later corrupts additional registers such as `x9`, `x10`, and a word in data memory. Fig. 6 show that more than 60% of flipped bits occur in the `ibex_if_stage`, about 19% in `sram_bank0`, and about 6% in the register file, while other components contribute only small fractions; the fetch stage thus forms the dominant hot spot of error propagation.

System-level behaviour: The core does not trap or reset. It completes the program and returns a status value, yet the result is incorrect and the PIN check is bypassed. The toggle comparison for `ibex_if_stage` in Fig. 6 bottom shows that, once the fault reaches the fetch pipeline, the faulty execution follows a distinct high-activity pattern while remaining bounded, reflecting a stable but wrong control flow that still satisfies the program's termination conditions.

Component-centric label: From the hierarchical viewpoint,

TABLE III
FAULT-INJECTION SUMMARY FOR VP0 AND VP7: (A) CAMPAIGN OUTCOMES AND (B) IDENTIFIED FAULT ORIGINS.

(a) Campaign outcomes

Workload	Campaigns	Reset	Crash	No effect	Success	Unclassified
VP0	191	31	39	118	2	1
VP7	140	23	24	91	0	2
Total	**331**	**54**	**63**	**209**	**2**	**3**

(b) Identified fault origins

Work load	Instruction Fetch	Decode Execute	Load Store	Control & Status Register	ALU	Register file	SRAM
VP0	19	5	6	5	4	1	1
VP7	10	4	4	3	2	1	0
Total	**29**	**9**	**10**	**8**	**6**	**2**	**1**

this case is classified as a register-file–originated multi-bit silent data corruption with fault-induced *success*. A single upset in the architectural register file propagates through instruction memory and the fetch subsystem, changes more than 60 bits of the program's return value, and yields an apparent success for an invalid PIN without any error signal.

C. Fault outcome and origin analysis

Table III summarizes EMFI outcomes for VP0 and VP7. Across 331 injections, most runs are benign (209 Fail), while the predominant observable effects are resets (54) and crashes (63). VP7 (protected) exhibits no fault-induced successes, whereas VP0 (unprotected) shows two successes.

For the Crash and Success cases where a localized root cause could be identified, the earliest deviating component is most often in the instruction-fetch (IF) unit (29/65), followed by the load–store unit (LSU, 10) and decode/execute (DE, 9). Architectural storage elements such as the register file are rarely the first point of divergence (2). This bias is expected because IF logic is exercised every cycle and directly controls the next instruction, so small upsets rapidly become control-flow divergence. By contrast, register upsets often remain latent, get overwritten, or are masked before affecting a security-relevant comparison, making them less likely to appear as the earliest observable deviation.

IV. CONCLUSION

This paper presented a scan-chain-assisted EMFI study on the `CAPRI1` RISC-V SoC. Full-scan visibility enabled cycle-level identification of first-flip events and reconstruction of fault propagation across Ibex microarchitectural blocks. The hierarchical analysis showed that a single-bit upset can traverse components and emerge as distinct software behaviors, including fault-induced success, stalls, and reset-driven failures. Across VP0 and VP7, identified origins concentrated in the instruction-fetch stage, with LSU and DE as secondary sources. To connect these architectural effects to physical coupling, we extracted PDN loops from routed DEF data and modeled rail disturbance at a victim tap as the combined effect

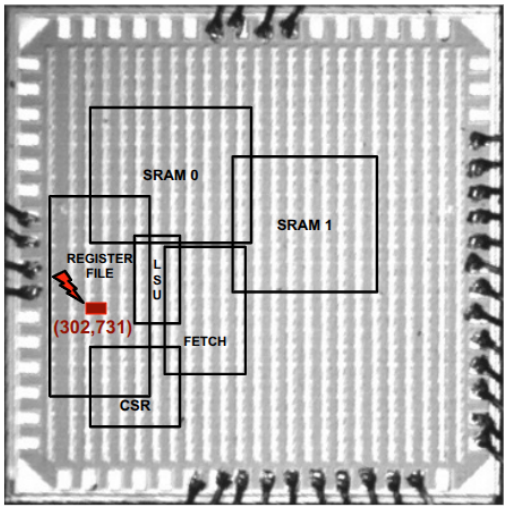

Fig. 7. Die overview with macro blocks and a scan-derived first-flip marker. *Placeholder: replace with the final annotated die image.*

of loop excitation and loop-to-tap delivery. Using a supply-swing bound, we derived a probe-position susceptibility map that aligns with the experimentally sensitive region, providing a scan-to-layout mechanism for hotspot interpretation.

V. LINKING DIRECTION-AWARE PDN LOOPS TO SCAN-DERIVED FIRST-FLIPS

This section links a layout-extracted PDN-loop model (pre-silicon) to scan-chain first-flip locations (post-silicon). Probe placement and pulse settings define the model input; scan data provides an independent response used for validation.

A. Post-silicon first-flip localization

For each injection campaign, the post-injection scan dump is compared against a fault-free reference. The *first flip* is defined as the earliest scan cell (bit order) that differs from the reference. That scan cell is mapped to a placed flip-flop instance, yielding the die coordinate $(x_{\mathrm{ff}}, y_{\mathrm{ff}})$.

B. Pre-silicon PDN loop set and direction-aware longest-loop definition

VDD and VSS routing is extracted from the DEF `SPECIALNETS` and encoded as one multilayer rail graph per supply. Graph edges correspond to wire segments and vias, with layer labels and physical lengths. Candidate PDN loops are sampled as closed cycles $\mathcal{L}_r = \{\ell\}$ for each rail $r \in \{\mathrm{VDD}, \mathrm{VSS}\}$.

Loop selection uses a direction-aware `METAL6` effective length. For probe center $\mathbf{p} = (x_p, y_p)$ and loop traversal direction, define

$$L_{\mathrm{eff}}(\ell; \mathbf{p}) = \left| \sum_{e \in \ell \cap \mathrm{M6}} s_e(\mathbf{p}) \, \mathrm{len}(e) \right|, \qquad s_e(\mathbf{p}) \in \{+1, -1\}. \tag{1}$$

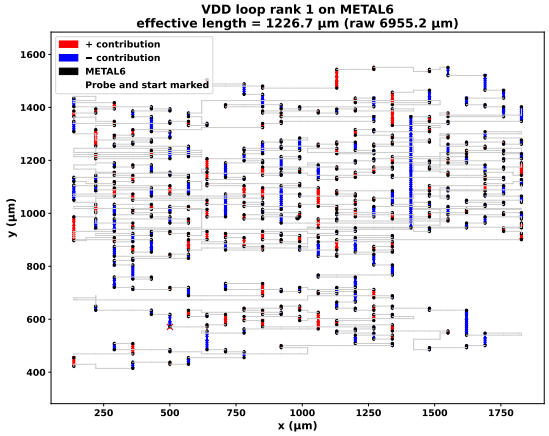

Fig. 8. Example METAL6 loop with sign assignment from (1)–(2). Red/blue segments contribute with opposite signs; the probe center is marked. *Placeholder: optionally mark* $(x_{\mathrm{ff}}, y_{\mathrm{ff}})$ *on the same axes.*

TABLE IV
TOP 5 EFFECTIVE LOOP LENGTHS ON METAL6 FOR THE VDD AND VSS NETS, TRANSPOSED SO RANKS ARE COLUMNS.

Net	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5
VDD	1226.7029	1223.8617	1220.7136	1218.6643	1218.2803
VSS	1455.4731	1454.2374	1440.9946	1439.2908	1438.0180

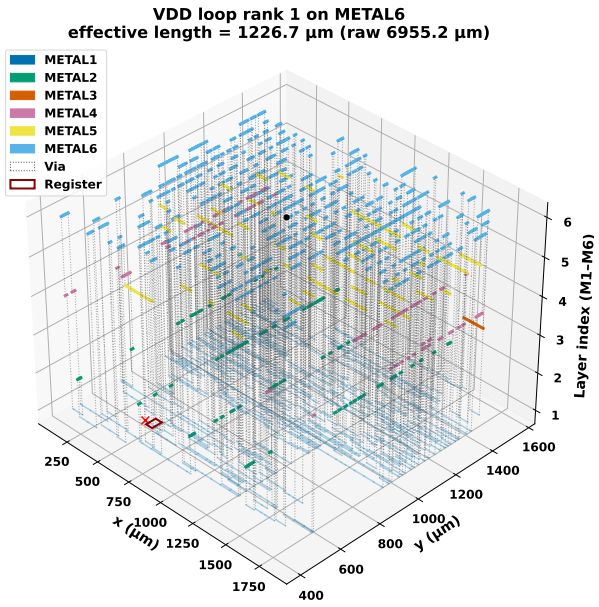

Fig. 9. Top candidate loops ranked by L_{eff} for VDD and VSS under one probe placement. *Optional: remove if space is tight; Fig. 8 is sufficient to define* L_{eff}.

For each METAL6 segment e with midpoint $\mathbf{m}_e = (x_e, y_e)$ and unit direction $\hat{\mathbf{u}}_e$ (given by the loop traversal order), the local tangential direction around $\mathbf{p}$ is approximated as

$$\hat{\mathbf{t}}(\mathbf{m}_e;\mathbf{p}) = \frac{1}{\|\mathbf{m}_e - \mathbf{p}\|}\begin{bmatrix} -(y_e - y_p) \\ (x_e - x_p) \end{bmatrix}, \qquad s_e(\mathbf{p}) = \mathrm{sign}\big(\hat{\mathbf{u}}_e \cdot \hat{\mathbf{t}}(\mathbf{m}_e;\mathbf{p})\big). \tag{2}$$

For a fixed probe placement and rail, the *longest loop* is defined as the candidate loop with maximum L_{eff}.

C. Rail disturbance at the first-flip tap

The first-flip coordinate $(x_{\mathrm{ff}}, y_{\mathrm{ff}})$ defines a victim tap t by snapping to the nearest METAL1 PDN segment. For each selected loop ℓ, the loop current under a pulsed probe is modeled as

$$I_\ell(\omega;\mathbf{p},D) \approx \frac{\kappa_\ell(\mathbf{p},D)\,V_0(\omega)}{Z_\ell(\omega)}, \tag{3}$$

where $Z_\ell(\omega) = R_\ell + j\omega L_\ell$ and $\kappa_\ell(\mathbf{p},D)$ captures coupling dependence on geometry and placement. Loop selection and coupling polarity follow the L_{eff} ordering for the given $\mathbf{p}$.

Delivery from loop ℓ to tap t proceeds through the minimum-impedance path $Z_{\ell\to t,r}(\omega)$ on rail r. The rail disturbance at t is estimated by

$$\Delta V_{t,r}(\omega;\mathbf{p},D,\tau,V_p) \approx \sum_{\ell\in\mathcal{L}_r} I_{\ell,r}(\omega;\mathbf{p},D,\tau,V_p)\cdot Z_{\ell\to t,r}(\omega), \tag{4}$$

and the swing bound is

$$|\Delta S|_{\mathrm{worst}} = |\Delta V_{t,\mathrm{VDD}}| + |\Delta V_{t,\mathrm{VSS}}|. \tag{5}$$

D. Pre- vs. post-silicon agreement and limitations

For each injection setting $(\mathbf{p}, D)$, the pre-silicon model ranks taps by $|\Delta S|_{\mathrm{worst}}$. The post-silicon scan dump provides the first-flip tap t_{ff}. Agreement is summarized by the hit-rate metric

$$\mathrm{hit@}K = \frac{1}{M}\sum_{i=1}^{M} \mathbb{1}\Big[t_{\mathrm{ff}}^{(i)} \in \mathrm{Top}K\big(|\Delta S|_{\mathrm{worst}}(\cdot;\mathbf{p}^{(i)},D^{(i)})\big)\Big], \tag{6}$$

and by rank correlation between predicted swing and observed first-flip frequency across taps (reported in the results section).

The PDN-loop model is first order. It uses sampled cycles, a tangential-field sign rule for L_{eff}, and an RL impedance model. Mutual coupling, capacitive effects, and package/board interaction are omitted, so the model is used for spatial ranking rather than absolute voltage prediction.

VI. Linking Direction-Aware PDN Loops to Scan-Derived First-Flips

This section links a layout-extracted PDN-loop model (representing *pre-silicon* analysis) to scan-derived first-flip locations (the *post-silicon* observations). In other words, the scan chain pinpoints where a fault first occurs in hardware, while the PDN model predicts which locations would experience a large supply disturbance from the EM pulse. The model is driven by the probe placement and pulse parameters, and we validate its predictions against the independent scan data.

A. Post-silicon first-flip localization

After each injection, we compare the captured scan-chain state to a fault-free reference. The earliest bit that differs from

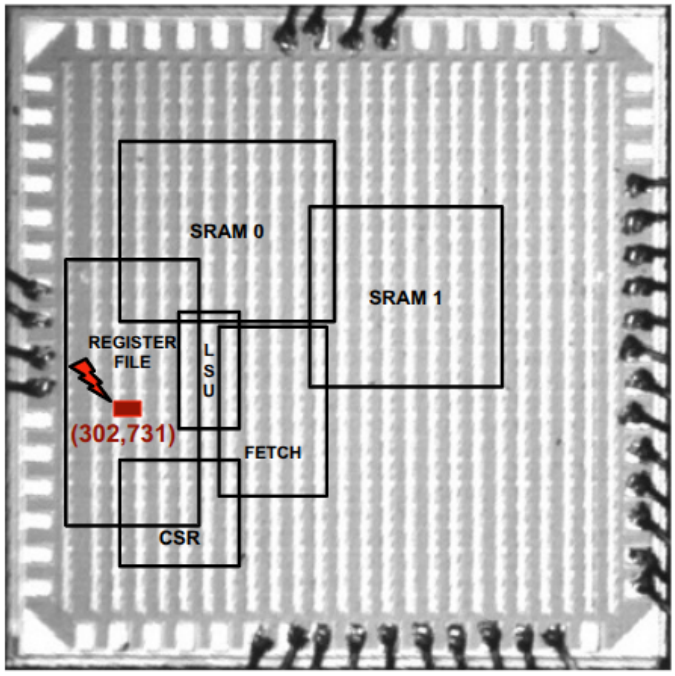

Fig. 10. Chip layout of CAPRI1 with major modules annotated. The star marker denotes the physical location of a representative first-flip event. *This demonstrates how scan data pinpoints a fault's origin on the die.*

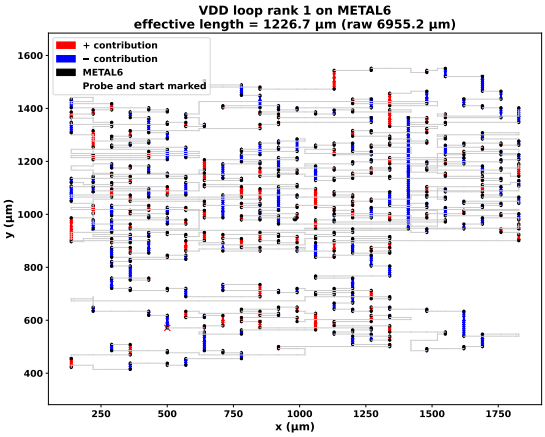

Fig. 11. Example of a top-layer (M6) PDN loop with sign assignments as defined in Eqs. (1) and (2). Red and blue segments contribute with opposite polarity; the probe's center is indicated. *This illustrates that loop orientation relative to the probe affects the effective coupling length.*

TABLE V
TOP FIVE L_{eff} VALUES (IN μM) FOR M6 LOOPS ON VDD AND VSS, WITH RANKS LISTED ACROSS COLUMNS. NOTABLY, THE LONGEST VSS LOOP IS LONGER THAN THE LONGEST VDD LOOP FOR THIS PROBE LOCATION.

Net	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5
VDD	1226.7	1223.9	1220.7	1218.7	1218.3
VSS	1455.5	1454.2	1441.0	1439.3	1438.0

the reference (in scan order) is identified as the *first flip*. We map this scan cell to its physical flip-flop instance, yielding the flip's die coordinate $(x_{\text{ff}}, y_{\text{ff}})$.

B. Pre-silicon PDN loop set and direction-aware longest-loop definition

We extract the VDD and VSS routing from the layout (DEF SPECIALNETS) and construct a multilayer rail graph for each supply. Graph edges correspond to metal segments (with physical lengths and layer information) and vias. We then enumerate closed loops $\mathcal{L}_r = \{\ell\}$ on each rail $r \in \{\text{VDD}, \text{VSS}\}$.

To quantify the loop most susceptible to EM excitation, we define a direction-aware effective length on the top metal layer (M6). For a probe centered at $\mathbf{p} = (x_p, y_p)$, the effective length of a loop ℓ is

$$L_{\text{eff}}(\ell; \mathbf{p}) = \left| \sum_{e \in \ell \cap \text{M6}} s_e(\mathbf{p}) \, \text{len}(e) \right|, \qquad s_e(\mathbf{p}) \in \{+1, -1\}, \tag{7}$$

where each segment e on M6 is assigned a sign $s_e(\mathbf{p})$ based on whether its orientation $\hat{\mathbf{u}}_e$ aligns constructively or destructively with the probe's field. We compute $s_e(\mathbf{p})$ by projecting e's unit vector onto the tangential direction around the probe (see Eq. 2). For a given probe position and rail, the *longest loop* is the candidate ℓ with the maximum L_{eff}.

C. Rail disturbance at the first-flip tap

The first-flip's coordinate $(x_{\text{ff}}, y_{\text{ff}})$ identifies the victim tap t (the nearest METAL1 segment on the PDN). For a given probe pulse, we approximate the induced current through each loop ℓ as:

$$I_\ell(\omega; \mathbf{p}, D) \approx \frac{\kappa_\ell(\mathbf{p}, D)\, V_0(\omega)}{R_\ell + j\omega L_\ell}, \tag{8}$$

where R_ℓ and L_ℓ are the loop's resistance and inductance, and $\kappa_\ell(\mathbf{p}, D)$ is a coupling factor capturing the probe-to-loop geometry. We assume loops with the largest L_{eff} dominate the coupling (with current polarity given by their sign).

This loop current propagates through the PDN to the tap t along the lowest-impedance path. The resulting transient on rail r at the tap is the superposition of contributions from all excited loops:

$$\Delta V_{t,r}(\omega; \mathbf{p}, D) \approx \sum_{\ell \in \mathcal{L}_r} I_{\ell,r}(\omega; \mathbf{p}, D)\, Z_{\ell \to t,r}(\omega), \tag{9}$$

where $Z_{\ell \to t,r}(\omega)$ is the transfer impedance from loop ℓ to tap t on rail r. Finally, assuming VDD is pulled low while VSS is simultaneously pulled high, we define a conservative worst-case supply swing at the tap:

$$|\Delta S|_{\text{worst}} = |\Delta V_{t,\text{VDD}}| + |\Delta V_{t,\text{VSS}}|, \tag{10}$$

which bounds the instantaneous VDD–VSS difference. In our model, this worst-case swing can exceed 1.4 V under a strong EM pulse—on the order of the disturbance needed to flip a 1.8 V storage element.

D. Pre- vs. post-silicon agreement and limitations

For each injection setting $(\mathbf{p}, D)$, the pre-silicon model produces a ranked list of tap locations by their predicted

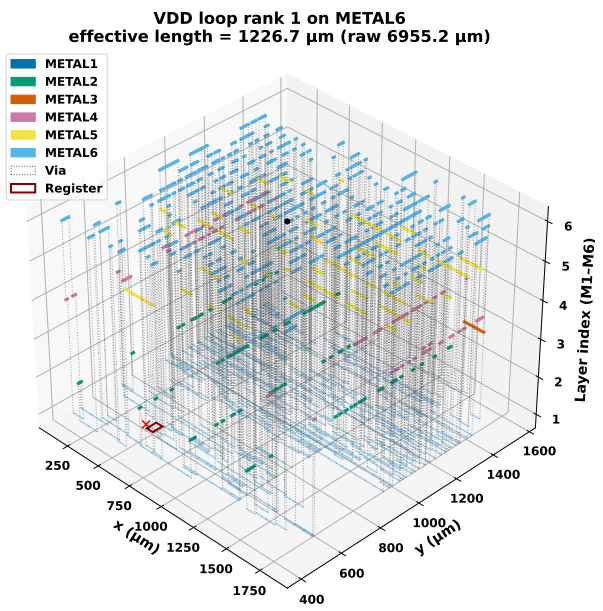

Fig. 12. Top-ranked PDN loops by L_{eff} for VDD and VSS under one probe placement. Here the longest VSS loop is slightly larger than the longest VDD loop, suggesting that the ground network provides a somewhat stronger coupling path in this case.

$|\Delta S|_{\text{worst}}$. The scan outcome provides the actual first-flip location t_{ff}. We evaluate the model's accuracy with a hit-rate metric:

$$\text{hit@}K = \frac{1}{M}\sum_{i=1}^{M} \mathbb{1}\Big[\, t_{\text{ff}}^{(i)} \in \text{Top-}K\big(|\Delta S|_{\text{worst}}; \mathbf{p}^{(i)}, D^{(i)}\big)\big)\Big], \tag{11}$$

which is the fraction of experiments for which the first-flip tap ranks among the top K predicted taps. We also compute the rank correlation between the model's predicted swing magnitudes and the observed first-flip frequencies at each tap (as reported in Section **??**).

Our PDN-loop model is a first-order approximation. It considers only the dominant loops (via L_{eff} ranking) and uses a simple RL representation for each loop with a tangential coupling assumption. It neglects higher-order effects such as mutual inductance between loops, capacitive coupling, and package/PCB interactions. Thus, the model is suited for identifying relative hot spots rather than precise voltage amplitudes. Even with these limitations, we find that locations with the highest predicted $|\Delta S|$ reliably correspond to the scan-observed fault hotspots, supporting this model's utility for early design-stage vulnerability assessment.

References

[1] S. Ordas, L. Guillaume-Sage, and P. Maurine, "Electromagnetic fault injection: The curse of flip-flops," *J. Cryptogr. Eng.*, vol. 7, no. 3, pp. 183–197, 2017.

[2] T. Dullien, "Weird machines, exploitability, and provable unexploitability," *IEEE Transactions on Emerging Topics in Computing*, vol. 8, no. 2, pp. 391–403, 2020.

[3] P. Sauter, T. Benz, P. Scheffler, H. Pochert, L. Wüthrich, M. Povišer, B. Muheim, F. K. Gürkaynak, and L. Benini, "Croc: An end-to-end open-source extensible risc-v mcu platform to democratize silicon," 2025. [Online]. Available: https://arxiv.org/abs/2502.05090

[4] L. Dureuil, G. Petiot, M. Potet, T. Le, A. Crohen, and P. de Choudens, "FISSC: A fault injection and simulation secure collection," in *Computer Safety, Reliability, and Security - 35th International Conference, SAFECOMP 2016, Trondheim, Norway, September 21-23, 2016, Proceedings*, 2016, pp. 3–11.

[5] Z. Liu, D. Shanmugam, and P. Schaumont, "Faultdetective: Explainable to a fault, from the design layout to the software," *IACR Trans. Cryptographic Hardware and Embedded Systems*, no. 4, pp. 610–632, Sep. 2024. [Online]. Available: https://tches.iacr.org/index.php/TCHES/article/view/11804

[6] A. Ghosh, M. Nath, D. Das, S. Ghosh, and S. Sen, "Electromagnetic analysis of integrated on-chip sensing loop for side-channel and fault-injection attack detection," *IEEE Microwave and Wireless Components Letters*, vol. 32, no. 6, pp. 784–787, 2022.

[7] M. Dumont, M. Lisart, and P. Maurine, "Modeling and simulating electromagnetic fault injection," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 40, no. 4, pp. 680–693, 2021.