PowerScout: A Security-Oriented Power Delivery Network Modeling Framework for Cross-Domain Side-Channel Analysis

Huifeng Zhu*, Xiaolong Guo†, Yier Jin‡ and Xuan Zhang*
*Washington University in St. Louis, †Kansas State University, ‡University of Florida
zhuhuifeng@wustl.edu, guoxiaolong@ksu.edu, yier.jin@ece.ufl.edu, xuan.zhang@wustl.edu

***Abstract*—The growing complexity of modern electronic systems often leads to the design of more sophisticated power delivery networks (PDNs). Similar to other system-level shared resources, the on-board PDN unintentionally introduces side channels across design layers and voltage domains, despite the fact that PDNs are not part of the functional design. Recent work have demonstrated that exploitation of the side channel can compromise the system security (i.e. information leakage and fault injection). In this work, we systematically investigate the PDN-based side channel as well as the countermeasures. To facilitate our goal, we develop PowerScout, a security-oriented PDN simulation framework that unifies the modeling of different PDN-based side-channel attacks. PowerScout performs fast nodal analysis of complex PDNs at the system level to quantitatively evaluate the severity of side-channel vulnerabilities.**

With the support of PowerScout, for the first time, we validate PDN side-channel attacks in literature through simulation results. Further, we are able to quantitatively measure the security impact of PDN parameters and configurations. For example, towards information leakage, removing near-chip capacitors can increase intra-chip information leakage by a maximum of 23.23dB at mid-frequency and inter-chip leakage by an average of 31.68dB at mid- and high-frequencies. Similarly, the optimal toggling frequency and duty cycle are derived to achieve fault injection attacks with higher success rate and more precise control.

I. Introduction

The power delivery network (PDN) is an indispensable component central to the correct operation of any electronics, as each functional unit of the system requires delivery of stable supply voltage and sufficient power. To satisfy the exponential demand for computing power, modern electronic systems are becoming increasingly complex. Also growing is the sophistication of the PDNs in these systems, in order to supply multiple voltage domains and satisfy their distinctive requirements, such as supply voltage levels, maximum load currents, and voltage noise margins for supply reliability. For example, IBM's new generation 24-core POWER9 processor has ten different input supply voltages and supports hundreds of voltage domains [1]. Moreover, modern computing platforms often integrate several different modules such as CPUs, GPUs, FPGAs, and DRAMs on the same motherboard, requiring a shared hierarchical PDN to facilitate the distribution of supply voltage among the modules across the chip, package, and PCB levels. A sample PDN is illustrated in Figure 1 where an FPGA board is used as an example.

Nonetheless, as a shared resource, PDNs create many pathways for unintended interactions and expose a system to various side-channel attacks [2], [3], [4]. Further, recent works have shown that many such vulnerabilities can be exploited remotely, making them especially potent security threats to modern electronic devices with ubiquitous connectivity. For example, in information leakage attacks, hackers can implement malicious voltmeters on FPGAs to steal sensitive information without physical access to the target systems [5], [6], [7]. PDN-based side channel can also be utilized to induce supply glitches (e.g., by implementing a power virus) in victim modules for denial-of-service (DoS) attacks [8] or differential fault analysis (DFA) [9] on cloud FPGAs. These ad-hoc experimental approaches, although useful in providing proof-of-concept demonstration of certain PDN vulnerabilities and attack scenarios, do not offer systematic and quantifiable guidance to discover new vulnerabilities or evaluate system resilience. As a consequence, it is urgently needed for a security-oriented modeling framework to accurately capture PDN behaviors that may lead to security vulnerabilities across multiple design layers and/or voltage domains.

978-1-7281-8952-9/20/$31.002020IEEE

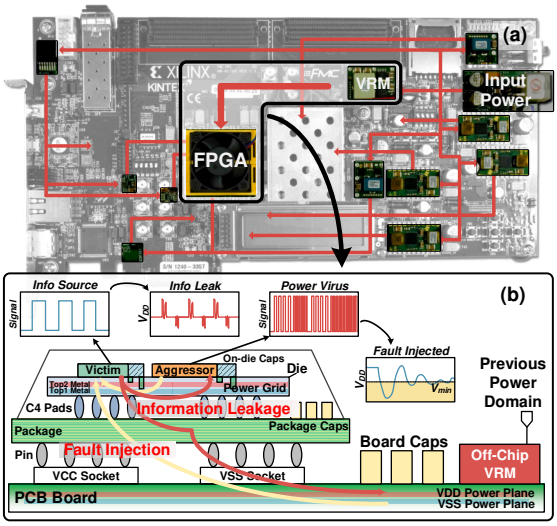

Figure 1: (a) An FPGA (Xilinx Kintex-7) PDN subsystem comprises of hierarchical VRMs (the boxes) and passive networks (the red lines), creating multiple voltage domains. (b) Single-stage PDN schematic and PDN-based side-channel attacks: information leakage and fault injection.

PDN modeling and simulation tools have been widely investigated mainly to estimate PDN characteristics (e.g., PowerSoc [10]), explore cross voltage domain PDN design space (e.g., Ivory [11]), and optimize PDN configurations (e.g., VoltSpot [12]). These existing tools tend to focus on the performance trade-off of PDN designs (i.e., performance vs efficiency vs supply noise) and lack essential capabilities to perform specific side-channel vulnerability analysis. In this work, we propose PowerScout—a unified PDN modeling framework that is able to perform thorough side-channel vulnerability analysis by simulating a complete PDN system across multiple design layers (i.e., chip, package, board) and voltage domains.

Instead of developing new circuit-level models for PDN, our methodology focuses on attaining a balance between security interpretability and simulation accuracy by using frequency-domain analysis for vulnerability exploration and transient simulation for security validation. Thus unlike previous coarse lumped PDN models [13], we are able to build a precise and unified PDN model to perform cross-domain nodal analysis with fast speed, which is capable of characterizing information leakage between arbitrary nodes in the system.We abstract critical components at multiple layers of a PDN, and provide a voltage regulator module (VRM) model to capture bi-directional voltage domain interactions and three load models to serve different PDN-based side-channel attack scenarios and analysis objectives. PowerScout is equipped with a user-friendly interface that enables easy generation of complex PDNs and fast exploration of full attack space under different PDN configurations. The main

contributions of this paper are as follows:

- We present a unified security-oriented PDN modeling framework named PowerScout. It can perform efficient evaluation of PDN side-channel vulnerabilities and systematic attack space exploration to guide secure PDN designs and effective defense strategies.
- PowerScout can correctly predict information leakage strength associated with PDN parameters and configurations. Our information leakage case study reveals $23.23dB$ and $31.68dB$ increase of intra- and inter-chip leakage from the removal of a near-chip capacitor, corroborating previous experimental results.
- We systematically explore the attack space of fault injection to identify effective region with linear sensitivity to toggling frequency and duty cycle.

II. Background

A. Power Delivery Network

The Power Delivery Network (PDN) is an essential subsystem in modern electronic systems. Figure 1 (a) and (b) show a simplified PDN across multiple layers, from the board to the chip. It contains board-level VRMs, interconnects from the VRMs to the pads on the chip, on-chip power grids to distribute power locally on the die, and decoupling capacitors along various stages of the PDN as well. In a system, there are many devices with different voltage supply and power distribution requirements, hence multiple voltage domains are created, each with its own VRMs to drive the local supply voltages [11]. These VRMs form a tree structure where upper nodes have higher voltages. Between the hierarchical VRMs and chips is the board-level passive distribution network containing PCB wire lines, PCB planes, and board-level decoupling capacitors. Via the package-level sockets, pins, and C4 pads, power is supplied to the microelectronic chip, where a multi-layer metal mesh forms the power grid that locally delivers power to each module inside the chip [13], [10]. Decoupling capacitors are implemented on both the package and die to further mitigate supply noise.

B. PDN-Based Side-Channel Attacks

Emerging PDN-based security threats can be categorized into two major classes: information leakage attacks and fault injection attacks. Figure 1 (b) shows the mechanisms of two attacks. Information leakage exploits the deterministic relationship between the switching activities of digital circuits and their dynamic currents. The induced supply voltage fluctuations can further propagate to other modules connected to the same PDN. Recent works suggest implementing malicious on-chip voltmeters, such as ring oscillators (ROs) [5] or time-to-digital converters (TDCs) [7], [6], to perform remote side-channel analysis in multi-tenant FPGAs. Similarly, the PDNs can also be used as a medium for covert channel communications. The attackers may implement dedicated oscillating cells (e.g., LFSR [14]) as transmitters to generate information-modulated currents. The receivers can be modules that are sensitive to supply voltage. Fault injection attacks take advantage of extreme supply fluctuations to violate the timing constraints and thus induce faults. The attackers normally create out-of-tolerance and precise controlled voltage drops by manipulating the power-hungry blocks known as power viruses. For multi-tenant FPGAs, the attackers exclusively implement RO-based power viruses (ROPVs) and use toggling signals to control their activities [9], [15]. As shown in [16], [17], PDN plays an important role in such power side-channel attacks and intrinsically determines the performance of the attacks. A PDN design with side-channel vulnerability will compromise the security of the whole system. However, to the best of our knowledge, there still lacks an effective framework supporting systematic analysis of PDN vulnerability.

III. Security-Oriented PDN Modeling

A. PowerScout Framework

A system diagram of the proposed PowerScout is shown in Figure 2 (a). PowerScout contains three main parts: the parameter panel, the PDN generator, and the vulnerability analyzer. Users input simple Python codes to call modules inside PowerScout with pre-defined PDN templates, which determines the topology of the PDN. While the parameter panel abstracts the PDN parameters from electronic component datasheets and technology libraries. For example, for the chip-level PDN topology, users can define several parameters such as the number of the power grids and C4 pads and the loads' types and locations. Given these inputs, PDN generator then automatically generates and simulates a full-system PDN netlist that specifies the complex hierarchical network. The raw outputs are parsed according to user-defined security analysis objectives and the side-channel vulnerability results are then reported by vulnerability analyzer.

In PowerScout, the induced voltage fluctuation $v(t)$ is computed by invoking the SPICE-level simulator, which performs numerical nodal analysis that can be expressed in a simplified form as:

$$v(t) = \int_0^t [Ce^{A(t-\tau)}B] \times i(\tau)d\tau \tag{1}$$

where $i(\tau)$ is the current consumption of the module; A, B, and C are the state-space matrices of the PDN. The values of the matrices depend on parameters and the topology of the PDN, as well as positions of the current source and observation point. Given the PDN model, time-domain results can be obtained by computing the convolution between the state-space matrices and the current waveform. However, solving such a high-order differential equation is time-consuming. To efficiently explore the vulnerabilities in a large design space, in PowerScout, the PDN is evaluated in the frequency domain, and Equation 1 can be rewritten as:

$$v(t) = \mathcal{F}^{-1}[Z(f) \times I(f)] \tag{2}$$

where $\mathcal{F}^{-1}$ is the inverse Fourier operator, and $Z(f)$ and $I(f)$ are the spectra of the PDN impedance and current consumption, respectively. It is widely accepted that the PDN can be viewed as a linear time-invariant (LTI) system and inverse Fourier transform can be directly applied. Since $\mathcal{F}^{-1}$ is a linear operator, $Z(f)$ can influence the induced voltage fluctuation in a straightforward manner and serve as the quantitative metric for evaluating the vulnerability to information leakage or fault injection. $Z(f)$ can usually be obtained easily since AC analysis is supported by most SPICE engines, which allows frequency analysis for PDN vulnerability evaluation.

B. PDN Model Construction

Passive RLC Network Model: The structure of PDN model is shown in Figure 2 (b). The model aims to fully reflect the power supply paths for critical devices with sufficient details. The board-level supply wireline is modeled as an inductor and a resistor, whose parameters depend on its length, width, and metal material characteristics. The PCB planes use the planar model with lumped capacitor and resistor, since the distributed effect is minimal at this scale. For the board-level capacitors, we model the characteristics of each capacitor. The frequency response of a single real capacitor is a band-pass filter instead of an ideal low-pass filter due to the parasitic effects [18]. The capacitor is thus modeled as the equivalent series inductor (ESL), equivalent series resistor (ESR), and an ideal capacitor. Later in Section IV-C, we will show the necessity of individually modeling each capacitor by demonstrating the role of near-chip capacitors.

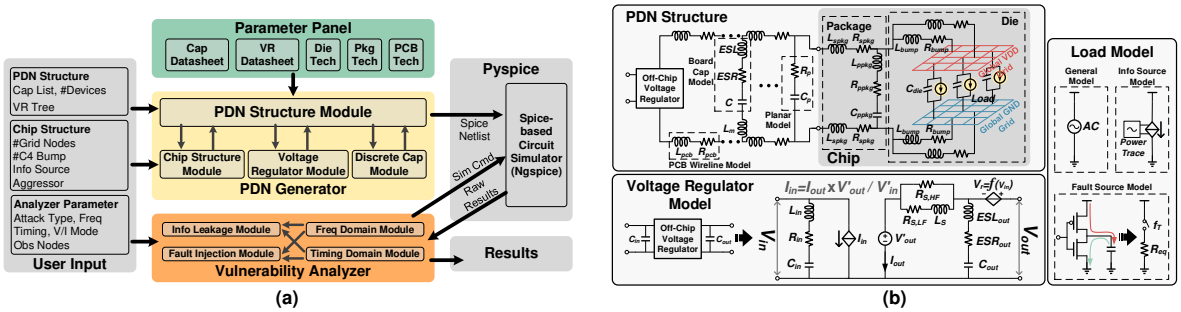

Figure 2: (a) The system diagram of the proposed PowerScout framework. (b) Security-oriented modeling of different PDN components.

For the chip-level PDN, we use the widely accepted package and die models [13]. The package is modeled as an RLC network, and the C4 bumps are modeled as parallel RL pairs that connect the grid to the package. The on-chip grid (i.e., the die model) is represented as an RL network. The on-chip capacitance is evenly distributed between the VDD and GND grids. The PDN structure and parameters have complex influence to the A, B, and C of Equation 1. Overall, higher capacitance leads to a decrease in $Z(f)$ while higher resistance and inductance have the opposite effects.

Active Voltage Regulator Model: In previous works and industrial models, VRMs are typically modeled as a fixed voltage source [13], [11] or a fixed voltage regulator in series connected to the equivalent inductor, capacitor, and resistor [19]. But this kind of model is not suitable for security-oriented PDN modeling since it ignores the interactions between different voltage domains. In PowerScout, we model the bi-directional interactions of different VRM topologies, including low-dropout regulators (LDOs), buck converters, and switched-capacitor converters. The VRM model, shown in Figure 2 (b), contains a voltage source V_{out}, an RLC network. The ESL_{OUT}, ESR_{OUT}, and C_{OUT} are based on the off-chip decoupling capacitor recommended in the datasheet; $R_{S,LF}$ is determined by the load regulation; and $R_{S,HF}$ and L_S are set to match the load transient response of the VRM. Before the output, a dependent voltage source V_{PSRR} is used to model the influence of the input voltage fluctuations on the output. The frequency response of V_{PSRR} matches the reverse of the power supply ripple rejection specified in the datasheet. To capture the reverse influence by the output on the input, we notice that the output side of the VRM indirectly impacts the input side by changing the current through the previous stage of the PDN. Therefore, we use a dependent current source ($I_{IN} = I_{OUT}V_{OUT}/V_{IN}$). For the input side RLC network, the values of ESL_{IN}, ESR_{IN}, and C_{IN} are also taken from the datasheet.

Analysis-dependent Load Models: In PowerScout, we provide three different load models that are suitable for different attack types and analysis objectives. As shown in Figure 2 (b), the general model is an ideal AC current source and is used for generalized side-channel vulnerabilities analysis. Using this model, frequency-domain analysis is performed, which can expose potential attacks, as shown in Equation 2. These attacks may need large amount of time-domain experiments to successfully exploit the vulnerability, if any. For dedicated information leakage evaluation or validation of specific vulnerability, we use a time-varying current source to model the changing power consumption of the information source in a transient simulation, which is based on Equation 1. The current source takes waveform file as input, which is generated by the power traces from other simulators, such as the architectural simulators (e.g., GEM5), or FPGA simulators (e.g., Xilinx ISE).

In fault injection attacks, the behavior of the power-hungry modules can be modeled as switching capacitors, which shares the same idea of classic power consumption model ($P = \alpha_{0\to1}C_L V_{dd}^2 f_{clk}$) of digital CMOS circuit [20]. The capacitor is the sum of the load capacitance of the malicious module. The current sink (i.e., the logic switches) is controlled by the toggling signal. When the time constant ($\tau = R_{PDN}C$) of switching capacitor is much smaller than toggling signal cycle, the fault source model can be further abstracted to a switched resistor, where the switch represents the toggling signal f_T.

C. Simulation Infrastructure

PowerScout is based on the open-sourced SPICE simulator Ngspice [21] and the Python package Pyspice [22]. Pyspice provides an API for Ngspice so that PowerScout can be written in Python. The users first define the model parameters in *parameter()* function. In *component()* function the basic components of PDN are defined, where the PowerScout built-in module library is called to generate the SPICE netlist blocks. Then the topology of PDN is determined in the *structure()* function. Inputting a small Python code block can generate SPICE netlist with a few hundreds of lines and it is easy for users to modify the configuration of PDN and regenerate the netlist, so that the PDN with different topologies can be fast evaluated. The generated SPICE netlist are simulated by Ngspice using vulnerability analyzer. PowerScout uses transient simulation for dedicated information leakage or fault injection attacks, and uses frequency-domain simulation for nodal vulnerability analysis. The raw results are parsed by vulnerability analyzer according to the analysis objective. For nodal vulnerability analysis, vulnerability analyzer calculates the information leakage strength or voltage drops for each node. If performing statistical side-channel analysis attacks, vulnerability analyzer also has built-in scripts to perform correlation power analysis (CPA) or differential power analysis (DPA).

IV. Information Leakage Attack Evaluation

In this section, we show that PowerScout can predict the PDN vulnerability leading to information leakage attacks. Aided by the PowerScout, we further provide insights on how to exploit PDN design parameters to maximize (or minimize) the information leakage. The experiments described here focus on side-channel analysis attacks, but PowerScout is also suitable for other information leakage attacks, including covert channel communications.

A. Attack Primer

The traditional approach of PDN-based side-channel analysis is to measure the voltage of the sampling resistor inserted to the power supply rail of the victim chip. In recent works [7], [6], the authors perform both intra- and inter-chip remote power analysis attacks on the SAKURA-G board without external measurements. The board contains two Spartan-6 FPGAs on the same board. The authors implement a 128-bit advanced encryption standard (AES) module on one of the FPGAs and implement TDCs on either the same

Table I: PDN Model Parameters for the Attack Evaluations.

Parameter	SAKURA-G Board			ML605 Board		
	R	L	C	R	L	C
Z_{pcb}	$0.58m\Omega$	$0.09nH$	—	$58\mu\Omega$	$91.7pH$	—
Z_{spkg}	$3.3m\Omega$	$0.5nH$	—	$0.55m\Omega$	$0.06nH$	—
Z_{ppkg}	$1.8m\Omega$	$28pH$	$270nF$ $172.3nF$	$0.1m\Omega$	$2.8pH$	$52\mu F$
Z_{bump}	$10m\Omega$	$0.32nH$	—	$20m\Omega$	$36pH$	—
Z_{die}	$3m\Omega$	$2.91fH$	$5.3nF$	$25m\Omega$	$2.91fH$	$10nF$

or the other FPGA as malicious on-chip voltmeters to measure the fluctuations of the power supply. The AES module runs at 24MHz and is based on a 32-bit datadepth without side-channel protection. The authors first illustrate the remote intra-chip CPA attacks when the voltmeter is implemented on the same victim FPGA. They also successfully perform CPA attacks when the voltmeter is on the other FPGA, showing the vulnerabilities of inter-chip side-channel analysis.

B. PowerScout Configuration

To analyze information leakage using PowerScout, we build a PDN model configured with parameters extracted from SAKURA-G board, which is suitable for evaluating both intra- and inter-chip information leakage. The structure of PDN is based on the schematic of the SAKURA-G board, including hierarchical VRMs and the two FPGAs. For each capacitor, the model parameters are extracted from the component datasheets [23]. The parameters of the PCB , package, and die model [13] are listed in Table I.

The vulnerabilities of information leakage attacks are systemically evaluated in PowerScout under multiple PDN configurations. To predict the performances of the side-channel analysis attacks, the information source model is implemented on the victim chip. The traces of the information source model are generated by Xilinx ISE power estimation, where we set the simulation interval adequately small to approximately represent the transient power consumption. The noise level of the PDN from the information source to the observation point is simulated, and a voltage source with Gaussian noise is accordingly implemented at the observation point. We record the power traces from both intra- and inter-chip observation points and perform CPA attacks. In the vulnerability analysis, the general model is used. We perform AC analysis and observe the information strength (i.e., $Z(f)$ in Equation 2), on each node of the PDN. The information strength is defined as the amplitude of the voltage fluctuations induced by the unit information source current. A higher information strength means information is more easily leaked at the same noise level. Note that noise is not included in the general analysis since we focus on the worst case for the defender side (i.e., the fewest power traces needed in CPA attacks). Users can still insert noise given the application scenarios.

Validation for the information leakage attack prediction by comparing prediction results with real-world experiments from prior work [7], [6] is presented in Figure 3. The upper panels are intra-chip CPA attack results from PowerScout and corresponding experiment measurements. During CPA attacks, the correlation coefficient is iteratively computed between the power traces and the modeled power consumption. As the number of power traces increases, the correlation coefficient of the correct key guess becomes distinguish from the other guesses. After multiple tests, we find that removing the capacitors near FPGAs can significantly reduce the number of needed traces. This configuration is similar to [7], and the results shown in Figure 3 (a) and (b) are consistent. The bottom panels show comparative inter-chip CPA attack performances. Besides removing the near-chip capacitors, we find the higher information leakage strength is achieved when we short the voltage regulators of the two FPGAs,

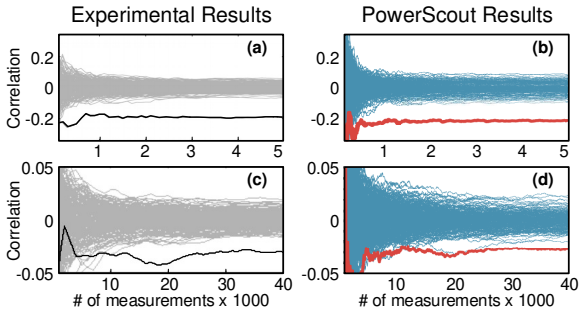

Figure 3: The experimental results for (a) intra-chip and (c) inter-chip CPA attacks [6], [7], and (b) (d) corresponding results from PowerScout.

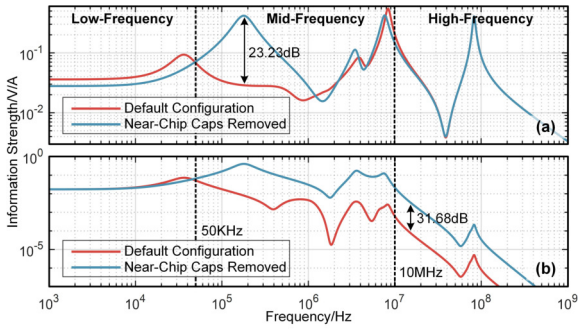

Figure 4: The information strength of (a) intra-chip and (b) inter-chip information leakage under two configurations.

where each FPGA is originally supplied by one voltage regulator. Comparing the results of experiment measurements (Figure 3 (c)) with similar scenario [6], PowerScout can also precisely predict the performances of information leakage attacks across several domains. Although the absolute values of needed power traces of two attacks are different due to the parameters setup, the relative values can sufficiently validate PowerScout methodology.

C. PowerScout Results and Discussion

Near-Chip Capacitors: In paper [6], [7], the authors remove the near-chip capacitors without detailed explanation on how the removal of capacitors will impact the experimental results. Our PowerScout clearly reveals the reason. The values of board-level capacitors cover a wide range and can be split into two groups: distant large capacitors and near-chip small capacitors. As mentioned before, removing the near-chip capacitors can significantly increase the information leakage. Figure 4 enables comparison of the information strengths of the two PDN configurations. The upper part shows that changes in intra-chip information leakage, where the information strength at mid-frequency greatly increases (as much as 23.23dB) when near-chip capacitors are removed. From Equation 2, this removal can increase the induced voltage fluctuation for a given information source, and thus increase the information leakage at this frequency. However, due to the C4 bump parasitic inductance, near-chip capacitors have relatively small effects at high frequency. For inter-chip information leakage, as shown in Figure 4 (b), near-chip capacitors significantly increase information strength at both mid and high frequencies (by an average of 31.68dB). Thus, near-chip capacitors play an important role in information leakage, although they account for only a minor portion of the gross capacitance.

Cross-Domain Leakage: In paper [6], a small bridge shorts the power rails so that the core voltage of the main FPGA is provided by the same power supply as for the auxiliary FPGA. The authors of [6] claimed that this configuration resembles more typical industrial boards and did not provide analysis on how this modification

Table II: The information strength after passing through the voltage regulators between different domains.

Voltage Domain	V_{DD}	$12V$	$5V$	$3.3V$	$1.2V$
PDN Branch	B_{vict}	—	$-15.17dB$	$-11.18dB$	$\mathbf{0dB}$
	B_{agg1}	—	⌊	$-21.26dB$	$-101.85dB$
	B_{agg2}	⌊	$-25.42dB$	$-86.97dB$	$-165.03dB$

would affect attack performance. Again, our PowerScout framework provides the reason of such setting. The information leakage among multiple domains is presented in Table II, where a PDN with three branches (B_{vict}, B_{agg1}, B_{agg2}) and four supply voltage levels (12V, 5V, 3.3V, 1.2V) is built. A voltage regulator is inserted between the adjacent voltage domains of one branch. B_{vict} and B_{agg1} share the same 5V→3.3V voltage regulator, while B_{vict} and B_{agg2} come from one 12V→5V voltage regulator. For other voltage domains, there are no direct connections. The information source is located at $B_{vict,1.2V}$, with an information strength of 0dB. Although the information decays greatly after passing through the voltage regulators, it still can leak through multiple voltage domains. It would be harder to detect the information if the observation point is structurally far from the information source, e.g. $B_{aggr2,1.2V}$ compared to $B_{aggr1,1.2V}$. For better attack performance, attackers need to reduce the distance from the source. This is achieved effectively in inter-chip attacks by directly connecting the power supply of two chips (as shown previously [6]): the leakage increases by 57.93dB when the two FPGA chips share the same voltage regulator.

V. Fault Injection Attack Evaluation

In this section, we evaluate fault injection attacks that use ROPVs. Rather than performing time-consuming experiments to evaluate the fault injection attacks, PowerScout allows comprehensive and efficient attack space exploration via simulation. Our findings not only are consistent with those at previous works [8], [9], [15], but also provide better interpretability.

A. Attack Primer

In current multi-tenant FPGA fault injection attacks, the adversaries exclusively implement ROPVs which are first introduced in [8], where the authors implement 18720 ROPVs (12.4% LUTs used) on an ML605 board and conducted several experiments to investigate the performance of attacks with different ROPVs toggling frequency. Later more precise control of fault injection using ROPVs is investigated. For example, FPGAHammer [9] controls fault injection by changing the toggling frequency and duty cycle. An automated calibration algorithm is developed to iteratively tune these two parameters according to the results for faults. Follow-up work also discusses the precise injection of faults by independently controlling two groups of ROPVs [15]. The ROPVs are first toggled with a period of fast-changing signals. Then the first group is kept active and the second group is disabled. After a specific delay, these two groups switch. In this way, attackers can induce a controllable period of time of stable voltage drop, which is sufficient for fault injection without crashing the system.

B. PowerScout Configuration

To systematically explore the attack space of fault injection attacks, we generate a PDN model using PowerScout and perform extensive experiments with different attack parameters. The structure of the model is based on the ML605 FPGA board schematic. For simplicity, we build only one stage of the supply voltage domain. We use the general load model for vulnerability analysis and use the fault source model to perform timing simulation. Since the oscillation frequency of ROPVs is usually much higher than the frequency of the toggling signals, its current consumption can thus be viewed as constant to increase simulation speed without much accuracy loss. The detailed parameters of the generated PDN model are listed in Table I. We believe that both the toggling frequency and duty cycle can affect the induced voltage drop. So we first simulate the attack in PowerScout while sweeping the toggling frequency from 1kHz to 10MHz and varying the duty cycle ranging from 10% to 90%. For each configuration, the minimum supply voltages when the fluctuations become periodic are recorded. Then two groups of ROPVs with different control timings are evaluated to design the methodology of precise fault injection.

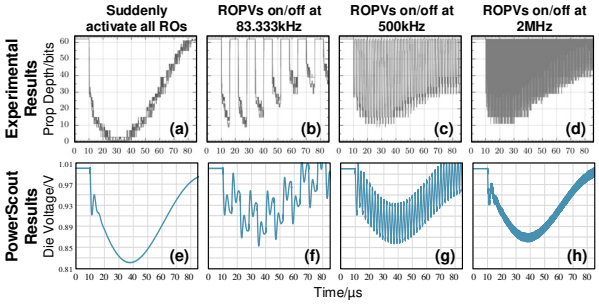

Figure 5: The experimental results of fault injection attacks [8], where the propagating depth is linear proportional to voltage drop (a)-(d), and corresponding results generated by PowerScout (e)-(h).

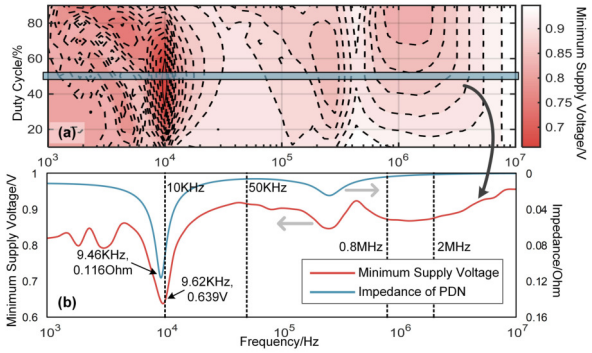

Figure 6: (a) The minimum die voltage under different toggling frequencies and duty cycles, and (b) the PDN impedance compared to the minimum die voltage at 50% duty cycle.

C. PowerScout Simulation Results

The consistency of the fault injection results between PowerScout and experiment measurements [8] is shown in Figure 5. In the upper row, we specifically plot the induced voltage fluctuations versus time when the toggling signals are one pulse, (83.3kHz,50%), (500kHz,50%), and (2MHz,50%). Compared to the experimental results that are also based on the ML605 board, even though the experimental results contain more glitches due to the oscillation of ROPV and measurement noise, it is clear that with the same toggling signals, the supply waveform envelopes match between the experiment and PowerScout simulation.

The attack space exploration for a single group of ROPVs is presented in Figure 6, where the heat map shows the minimum die voltage under different toggling frequencies and duty cycles. Generally, the toggling frequency and duty cycle do not have linear influences on the voltage drop. But when the toggling frequency ranges between 10KHz and 50KHz, the fault injection performance is linear to the toggling frequency. For some regions (e.g. around 0.8MHz and 2MHz), the voltage drop is proportional to the duty cycle. Moreover, there exists a most efficient toggling frequency which can induce the maximum voltage drop. At this frequency, the duty cycle does not have much influence on the fault injection

Table III: Characteristics comparison between PowerScout and related works.

	Ivory[11]	PowerSoC[10]	VoltSpot[12]	[24]	[17]	**PowerScout (This work)**
Security-Oriented	No	No	No	Yes	Yes	**Yes**
Programming Language	C++	C++	C	N/A[1]	N/A[1]	**Python**
Modeling Level	PCB-Level[2]	PCB-Level[2]	Chip-Level	PCB-Level[2]	Chip-Level	**System-Level**
PCB PDN Model Type	Lumped	Lumped	-	Lumped	-	**Distributed**
Chip PDN Model Type	Distributed	Distributed	Distributed	Lumped	Distributed	**Distributed**
Transient Simulation	Yes	Yes	Yes	Yes	Yes	**Yes**
Frequency-Domain Simulation	Yes	No	No	No	No	**Yes**
Analysis Type	N/A	N/A	N/A	Information Leakage	Information Leakage	**Information Leakage Fault Injection Attack Nodal Vulnerability Evaluation**

[1] This work provides a simulation method instead of a framework.
[2] It contains both the PCB-level modeling with one-stage voltage domain and the chip-level modeling.

performance. Figure 6 (b) shows the minimum supply voltage versus toggling frequency when the duty cycle is 50%, and also shows the simulated PDN impedance. The resonant frequency of the PDN impedance is almost the same as the most efficient toggling frequency, so that by using the resonant frequency and corresponding impedance (9.46KHz,0.116Ohm) from vulnerability analysis, we can effectively predict the maximum fault injection performance (9.62KHz,0.639V). Our findings not only are consistent with previous works [8], [9], [15], but go beyond to provide key insight for future more efficient exploration of the attack space.

VI. Related Works and Discussion

Table III lists the characteristics comparison between PowerScout and related works including PDN model and simulation frameworks, and simulation method. Since most frameworks [11], [10], [12] are not security-oriented, the authors in [24] and [17] propose simulation methods aiming to PDN-based side-channel analysis attacks. However, the authors do not provide the framework for further analysis. PowerScout enables full system-level simulation by bi-directional VRM model. Moreover, by considering the effects of both distributed on-board capacitors and the on-chip power grid, PowerScout achieves high accuracy and fidelity in its simulation of the PDN subsystem. Besides transient simulation, PowerScout can perform fast system-level nodal vulnerability analysis frequent-domain simulation. Programmed in Python, PowerScout also has good scalability and extensibility to be combined with other frameworks.

VII. Conclusion

In this paper, we present a security-oriented PDN modeling framework named PowerScout. Focusing on the vulnerability of PDN itself, we enable cross-domain nodal analysis by providing a precise and unified PDN model. Multiple PDN side-channel vulnerability simulations are demonstrated with proposed analysis-dependent load models. Having a user-friendly interface, PowerScout also can easily generate complex PDNs and perform thorough attack space exploration. We show that PowerScout can successfully predict the performances of both information leakage attacks and fault injection attacks. Insights to better exploit PDN side-channel vulnerabilities are further provided by PowerScout. It should be noted that we do not enable evaluating the information leakage from a specific digital module or fault injection within the module. These two topics still are significant problems and they are out of the scope of this work.

Acknowledgments

Portions of this work were funded by Semiconductor Research Corporation (Task No. 2810.003 through UT Dallas' Texas Analog Center of Excellence), DARPA and the Office of Advanced Scientific Computing Research, US Department of Energy. The views expressed in the paper are the opinions of the authors and do not represent official positions of DARPA, Pacific Northwest National Laboratory, the US Department of Energy, nor the US Government. Pacific Northwest National Laboratory is operated by Battelle for US Department of Energy under contract DE-AC05-76RL01830.

References

[1] C. Gonzalez *et al.*, "The 24-core power9 processor with adaptive clocking, 25-gb/s accelerator links, and 16-gb/s pcie gen4," *IEEE Journal of Solid-State Circuits*, 2017.
[2] A. Tang *et al.*, "Clkscrew: exposing the perils of security-oblivious energy management," in *26th USENIX Security Symposium*, 2017.
[3] L. Bossuet *et al.*, "Dvfs as a security failure of trustzone-enabled heterogeneous soc," in *25th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, 2018.
[4] S. K. Khatamifard *et al.*, "Powert channels: A novel class of covert communication exploiting power management vulnerabilities," in *International Symposium on High Performance Computer Architecture*, 2019.
[5] M. Zhao *et al.*, "Fpga-based remote power side-channel attacks," in *IEEE Symposium on Security and Privacy (SP)*, 2018.
[6] F. Schellenberg *et al.*, "Remote inter-chip power analysis side-channel attacks at board-level," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2018.
[7] ——, "An inside job: Remote power analysis attacks on fpgas," in *IEEE Design, Automation & Test in Europe Conference & Exhibition*, 2018.
[8] D. R. Gnad *et al.*, "Voltage drop-based fault attacks on fpgas using valid bitstreams," in *Field Programmable Logic and Applications*, 2017.
[9] J. Krautter *et al.*, "Fpgahammer:remote voltage fault attacks on shared fpgas,suitable for dfa on aes," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2018.
[10] X. Wang *et al.*, "An analytical study of power delivery systems for many-core processors using on-chip and off-chip voltage regulators," *Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 2015.
[11] A. Zou *et al.*, "Ivory: Early-stage design space exploration tool for integrated voltage regulators," in *ACM Proceedings of the 54th Annual Design Automation Conference*, 2017.
[12] R. Zhang *et al.*, "Architecture implications of pads as a scarce resource," in *IEEE International Symposium on Computer Architecture*, 2014.
[13] M. S. Gupta *et al.*, "Understanding voltage variations in chip multiprocessors using a distributed power-delivery network," in *IEEE Proceedings of the conference on Design, automation and test in Europe*, 2007.
[14] S. Kutzner *et al.*, "Trojanus:an ultra-lightweight side-channel leakage generator for fpgas," in *IEEE International Conference on Field-Programmable Technology(FPT)*, 2013.
[15] D. Mahmoud *et al.*, "Timing violation induced faults in multi-tenant fpgas," in *Design,Automation Test in EuropeConferenceExhibition*, 2019.
[16] X. Wang *et al.*, "Role of power grid in side channel attack and power-grid-aware secure design," in *50th Design Automation Conference*, 2013.
[17] J. Yang *et al.*, "Power supply noise aware evaluation framework for side channel attacks and countermeasures," in *IEEE International Conference on Field-Programmable Technology (FPT)*, 2014.
[18] S. Zhao *et al.*, "Frequency-domain power delivery network self-characterization in fpgas for improved system reliability," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 11, pp. 8915–8924, 2018.
[19] M. Kar *et al.*, "Reducing power side-channel information leakage of aes engines using fully integrated inductive voltage regulator," *IEEE Journal of Solid-State Circuits*, vol. 53, no. 8, pp. 2399–2414, 2018.
[20] A. P. Chandrakasan *et al.*, "Minimizing power consumption in digital cmos circuits," *Proceedings of the IEEE*, pp. 498–523, 1995.
[21] H. Vogt *et al.*, "Ngspice users manual version 31," http://ngspice.sourceforge.net/docs/ngspice-31-manual.pdf, 2019.
[22] F. Salvaire, "Pyspice," https://pyspice.fabrice-salvaire.fr, 2019.
[23] Murata, "Simsuring capacitor models," https://ds.murata.co.jp/simsurfing/mlcc.html?lcid=en-us, 2019, accessed July 10, 2019.
[24] A. Tsukioka *et al.*, "A fast side-channel leakage simulation technique based on ic chip power modeling," *IEEE Letters on Electromagnetic Compatibility Practice and Applications*, 2020.