Mitigating Voltage Attacks in Multi-Tenant FPGAs

GEORGE PROVELENGIOS, DANIEL HOLCOMB, and RUSSELL TESSIER, University of Massachusetts Amherst, MA, USA

Recent research has exposed a number of security issues related to the use of FPGAs in embedded system and cloud computing environments. Circuits that deliberately waste power can be carefully crafted by a malicious cloud FPGA user and deployed to cause denial-of-service and fault injection attacks. The main defense strategy used by FPGA cloud services involves checking user-submitted designs for circuit structures that are known to aggressively consume power. Unfortunately, this approach is limited by an attacker's ability to conceive new designs that defeat existing checkers. In this work, our contributions are twofold. We evaluate a variety of circuit power wasting techniques that typically are not flagged by design rule checks imposed by FPGA cloud computing vendors. The efficiencies of five power wasting circuits, including our new design, are evaluated in terms of power consumed per logic resource. We then show that the source of voltage attacks based on power wasters can be identified. Our monitoring approach localizes the attack and suppresses the clock signal for the target region within 21 µs, which is fast enough to stop an attack before it causes a board reset. All experiments are performed using a state-of-the-art Intel Stratix 10 FPGA.

CCS Concepts: • **Security and privacy → Hardware attacks and countermeasures**; • **Hardware → Reconfigurable logic and FPGAs;**

Additional Key Words and Phrases: Embedded FPGAs, cloud FPGAs, voltage attacks

ACM Reference format:
George Provelengios, Daniel Holcomb, and Russell Tessier. 2021. Mitigating Voltage Attacks in Multi-Tenant FPGAs. *ACM Trans. Reconfigurable Technol. Syst.* 14, 2, Article 9 (July 2021), 24 pages.
https://doi.org/10.1145/3451236

1 INTRODUCTION

As FPGAs have grown in logic capacity and performance, their use in a diverse set of applications has dramatically increased. Not only are these devices now used in a variety of embedded systems, but also they are now widely deployed in cloud computing installations [3–5]. It is widely expected that the logic capacity and cost of FPGAs will continue to encourage the simultaneous use inside an FPGA of logic components from multiple untrusted designers. These multi-tenant cases can arise as a result of **intellectual property (IP)** core use for embedded systems and deployment scenarios for cloud FPGAs, both current and forward-looking.

This research was funded by NSF grants CNS-1619558 and CNS-1902532 and a grant from Intel's Corporate Research Council.
Authors' address: G. Provelengios, D. Holcomb, and R. Tessier, Department of Electrical and Computer Engineering, University of Massachusetts Amherst, MA 01003, USA; emails: {gprovelengio, dholcomb, tessier}@umass.edu.

1936-7406/2021/07-ART9 $15.00
https://doi.org/10.1145/3451236

Although the diversity of logic element types found in SRAM-based FPGAs has grown significantly since their commercial introduction 36 years ago, on-chip power distribution approaches have mostly stayed the same. Unlike multi-core microprocessors, all on-FPGA core logic shares the same **power distribution network (PDN)**. Several researchers have shown that on-chip voltage drops induced by attackers can drive an FPGA-based board into reset [14] or induce timing faults in neighboring circuits [1, 20, 24, 27]. Although many circuits that deliberately waste power can be detected via netlist or bitstream scans [12], the spectrum of power wasters is continually evolving, not unlike software viruses that attack personal computers. In this article, we describe and evaluate a power waster circuit that is difficult to detect via scanning since it appears similar to standard design logic. This circuit is easy to implement and consumes significant power by deliberately propagating signal glitches through circuitry with high fanout and minimal logic masking. We show that its deployment on an Intel Stratix 10 device can lead to board reset.

Recent research [12, 27, 38, 40] has highlighted the importance of developing active defense strategies to mitigate voltage attacks in FPGA devices. These approaches include off-chip tracking of FPGA power consumption and on-chip voltage monitoring [27, 40]. Several techniques that use on-chip sensors to diagnose aggressive on-FPGA power consumption behaviors have been examined [12, 27]. In this article, we advance the application of such sensor-based solutions by using voltage values from a low-overhead, integrated on-chip sensing system to enable the real-time detection and mitigation of on-chip voltage attacks. The contributions of our work can be summarized as follows:

- We examine a stealthy power wasting circuit that can be easily implemented in logic in a state-of-the-art FPGA. The design includes standard synchronous logic clocked by a global clock, making it difficult to identify by compile-time scanners looking for malicious design circuitry.
- To better examine the scope of the voltage attack mitigation problem, we examine the voltage response of a Stratix 10 device to the sudden activation of our stealthy power wasting circuit.
- An optimized on-FPGA sensor network is implemented in Stratix 10 circuitry. This network is able to isolate the location of activated voltage attacks. Tradeoffs in accuracy versus numbers of sensors are evaluated.
- Finally, we propose and develop a practical system to mitigate voltage attacks in real time by deactivating clock signals in an FPGA logic region suspected by our sensing system of containing malicious circuitry. We demonstrate that our sensing and mitigation system can successfully defend a Stratix 10 against a board reset attack using power wasters.

A Terasic DE10-Pro [34] development board containing a 14 nm Stratix 10 FPGA device is used to evaluate the sensor network and prototype the mitigation system. All experiments were performed under typical operating conditions and without DE10-Pro board modifications.

The work presented in this article substantially extends our preliminary research on FPGA voltage attacks and their effects [27, 28]. In this article, we characterize the voltage behavior of a 14 nm Intel Stratix 10 device rather than an earlier-generation 28 nm Intel Cyclone V device. A glitch-based power waster, introduced in Provelengios et al. [28], is used for voltage attacks rather than wasters that are typically flagged by cloud computing vendors. Using the Stratix 10 devices, we extend our attack localization techniques and introduce a mechanism to suppress design clocks in regions where malicious power consumption is suspected. We demonstrate that our approach reacts quickly enough to prevent a malicious design on a Stratix 10 FPGA from causing the device to reset its circuit board, necessitating a bitstream reload.

The remainder of the article is organized as follows. Background on our threat model and FPGA voltage attacks, sensors, and monitoring and mitigation technologies are described in Section 2.

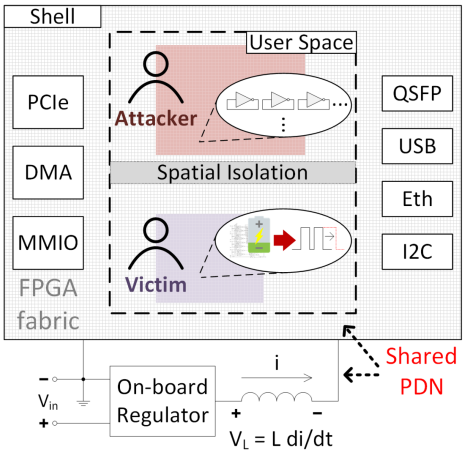


Fig. 1. Schematic of an FPGA multi-tenant scenario. Due to the shared use of the FPGA PDN, current drawn aggressively by a malicious application can cause voltage droops that impact co-located circuits that are spatially isolated.

Section 3 describes our methodology to detect power fluctuations using on-chip voltage sensors, and Section 4 shows how our approach can be used to localize the source of the instability in a Stratix 10 device. Our system for on-chip voltage sensing and real-time attack mitigation is described in Section 5. Section 6 concludes the article and offers directions for future work.

2 BACKGROUND AND RELATED WORK

2.1 Multi-Tenant FPGA Threat Model

The threat model associated with multi-tenant FPGA voltage attacks can be defined as follows. Circuits created by multiple, potentially untrusted designers can simultaneously be instantiated and executed in an FPGA. Design circuitry is spatially isolated and the programming information for each design is protected. It is assumed that interface circuity and supporting software are also secure. An individual user has the freedom to craft a circuit of any type allowed by the platform operator and implement it in the assigned portion of the FPGA. Figure 1 illustrates the nature of the threat.

The threat model for this work follows several commercial use cases, both current and likely in the near future [17]. The large logic capacity of existing FPGA designs often necessitates the use of third-party IP cores that could potentially be malicious. These cores could be in cloud FPGAs or other types of embedded systems, creating the first use case for our threat model. This use case is possible in current practice, although no specific cases of malicious FPGA IP vendors have been reported.

As the extent of FPGA-based cloud computing grows [3–5], the simultaneous shared use of an FPGA device by multiple tenants becomes an appealing business model for cloud providers, forming our second use case. Several integrated approaches have been developed recently [18, 19, 36] that leverage operating system environments to dynamically allocate and simultaneously execute multiple FPGA applications in a cloud setting. Khawaja et al. [18] and Knodel et al. [19] consider circuit swapping on demand to address application acceleration. Yazdanshenas [36] considers the on-chip organization of FPGA resources to support multi-tenant cloud computing. This multi-tenant cloud model of FPGA usage is susceptible to voltage attacks since all cloud FPGAs from major commercial vendors have power distribution networks that are shared across the entire

device. As a result, the multi-tenant approach with multiple independent users is not yet supported by commercial vendors, although it is likely to be in the future [17].

2.2 FPGA Voltage Manipulation Attacks

A sizable number of FPGA voltage attacks [14, 20, 24, 41] that can be exploited in multi-tenant scenarios have been reported. One tenant may try to maliciously induce localized instability in the supply voltage through **lookup table (LUT)**-based shift registers [41] or deliberate short circuits [29], possibly also exploiting resonances of the power grid. Additional on-chip FPGA attacks are often based on **ring oscillators (ROs)**, which are asynchronous loops containing an odd number of inverters. Several recent works [14, 20, 24] show that in some cases, RO-based power wasters that consume significant dynamic power can be used in an FPGA to cause voltage instability. However, these works do not consider remediation approaches.

Prior work has shown that a shared FPGA PDN creates coupling between applications. The coupling has been exploited for side channel attacks [30, 37] in which an encryption key is extracted from an unsuspecting victim crypto circuit. FPGA PDN manipulation can also be used to form covert communication channels between circuits on the same chip. Gnad et al. [11] show that data transfer rates up to 8 Mbit/s are possible on FPGAs if ring oscillators are used to detect voltage changes induced at a distant part of the chip. In Giechaskiel et al. [7], covert communication across multiple dice in the same FPGA package was shown. These side channel communication approaches are orthogonal to our work, which aims to detect fault injection attacks.

2.3 FPGA Voltage Attack Remediation

Voltage attack remediation involves identifying on-FPGA attempts to manipulate voltage and taking steps to suppress them. FPGA logic can be crafted into sensors that can detect the voltage drops caused by power consumption [38–40]. Although recent-generation FPGAs often include one or more voltage sensors per device [15], their sample rates are in the millisecond range, which is too slow to react to most attacks. Gnad et al. [13] examined spatial and temporal voltage effects across an FPGA for a variety of workload characteristics. This work did not consider remediation approaches to stop an attack.

Studies that address voltage attack suppression have generally focused on compile-time bitstream scanning or runtime clock suppression. Bitstream scanning to locate power wasting circuits, such as ROs, was proposed by Krautter et al. [21]. This approach is used by Amazon EC2 F1 compile software [3] as a protection step before FPGA design deployment. A clock edge suppressor was used at runtime by Shen et al. [31] to defeat voltage spikes caused by user designs. The use of voltage information collected across the die to identify attacks was demonstrated by Zick et al. [40]. Hardware circuits to suppress the attacks were proposed but not demonstrated. AWS EC2 F1 provides power monitoring through its runtime management tools using the command *fpga-describe-local-image*, although measurements are only provided to the user at a time granularity of one measurement per minute [9]. The system does support clock deactivation for the FPGA board if power consumption exceeds 85 W.

3 FPGA PDN ATTACKS ON AN INTEL STRATIX 10 FPGA

In this section we examine the PDN response of a Stratix 10 FPGA following the activation of on-chip power wasters. To fully illustrate the effects, we consider five circuits that can be crafted from FPGA logic to deliberately consume excessive power. These circuits can be quickly activated under user control.

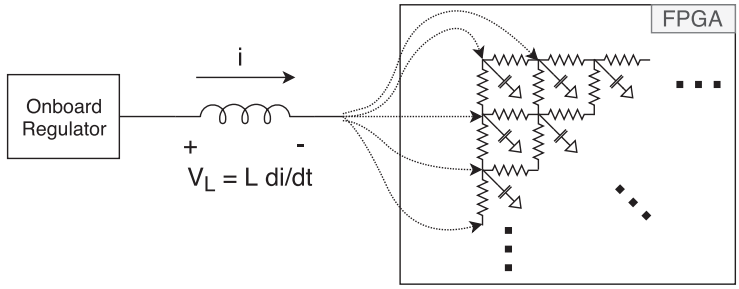


Fig. 2. Schematic of on-chip FPGA power system. A voltage drop occurs across the inductor due to di/dt. A steady-state voltage drop occurs in the PDN due to its resistance.

An illustration of a standard on-FPGA PDN can be seen in Figure 2. Supply voltage is provided to the FPGA by an on-board regulator via an inductor. The output of the inductor drives the FPGA PDN via multiple device input pins. As discussed in Section 3.2.2, the inductor leads to a substantial, instantaneous voltage drop upon power waster activation followed by a sustained on-chip resistive drop.

3.1 Adversarial Power Consumption Circuits

The criteria for evaluating power wasting circuits include power consuming efficiency, size, and similarity to other types of design circuitry, which can make the wasting circuits harder to detect. Dynamic power consumption in FPGAs (Equation (1)) is due to the logic signals' switching capacitance C at frequency f_{sw} between low and high voltage levels (V_{DD}). Circuits that maximize signal toggling and can be packed together for high utilization are ideal candidates for wasting power in FPGAs. In the following subsections, we evaluate five such synchronous and asynchronous power wasters both qualitatively and quantitatively.

$$p_{dyn} = V_{DD}^2 \cdot f_{SW} \cdot C \tag{1}$$

We used an Intel Stratix 10 SX FPGA (1SX280HU2F50E1VG) located on a Terasic DE10-Pro board to evaluate the power-consuming abilities of the waster circuits. Power to the DE10-Pro board is provided from a 12 V DC source. The 0.9 V internal FPGA core voltage (*VCCINT*) is supplied by a Linear Technology LTM4677 step-down regulator switching at 425 kHz. Two Linear Technology LTC2945 power monitor chips were used to track the 12 V board input supply and 0.9 V FPGA core voltages.

3.1.1 RO-Based Power Waster. Ring oscillators are commonly used in FPGAs to consume dynamic power since they typically achieve high switching frequencies and are easy to design and build in FPGAs. Figure 3 shows two single-stage RO instances used as power wasters. Up to 20 ROs can be packed into a single Stratix 10 **logic array block (LAB)**. A script can be used to uniformly place ROs throughout a region of a device (e.g., a clock region). All of these circuits can be enabled nearly simultaneously through the use of an `Enable` signal assigned to a high fanout global network signal.

The power consumed per RO-based power waster instance for an increasing number of wasters is shown in Table 1. These values were read from the power monitor chip attached to the core voltage. As the number of the RO instances grows, a local drop in supply voltage is induced, slowing down oscillation. This effect results in less power being consumed per instance. Activating 30,000 instances at once causes a loss of JTAG communication between the host PC and the board.

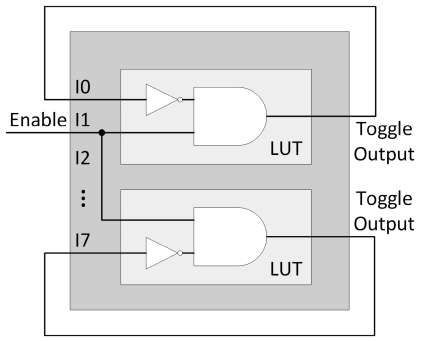


Fig. 3. Two instances of the single-stage RO-based waster.

Table 1. Power Consumed by Each RO-Based Power Waster Instance (Figure 3)

Number of PW Instances	Power / Instance [mW]
10,000	2.58
14,000	2.49
18,000	2.50
22,000	2.48
26,000	2.31
30,000	2.30

†30,000 instances utilize 1.6% of the LUTs available on the 1SX280 FPGA device.

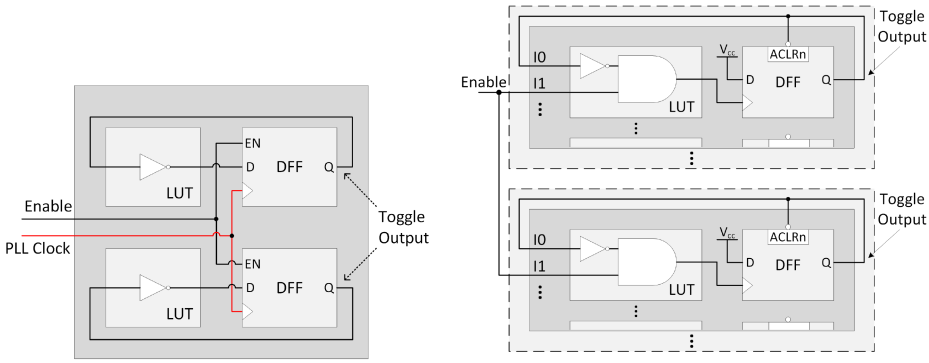
Activating the 30,000 instances incrementally in an attempt to gradually increase power consumption results in the same outcome. The LTC2945 reported that while the wasters were active with 30,000 instances, the total power consumed by the FPGA device reached 98 W. Attempting to activate more than 30,000 instances results in an immediate crash of the board. In the case of either communication loss or an immediate crash, a hard reset is insufficient to make the board accessible again and a manual power cycle is required.

Although ROs are clearly efficient and have legitimate FPGA uses for voltage [27] and temperature [39] sensing, their association with malicious attacks makes them a target for cloud FPGA vendors. For example, the compilation software for Amazon EC2 F1 examines candidate netlists for ROs and flags them without generating an FPGA bitstream [8, 32]. As a result, ROs made strictly from LUTs are not a suitable choice for an attacker.

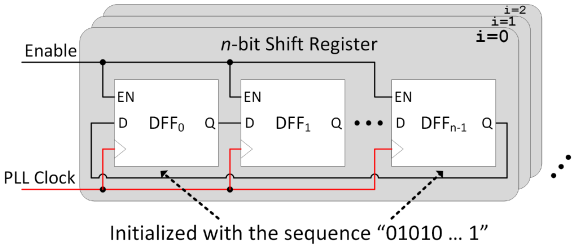
3.1.2 Alternative Power Wasters. Several researchers have determined that RO-style behavior can be obtained from FPGA circuits that also contain at least one flip-flop. These types of circuits evade the combinational loop detector in cloud FPGA compilers (at least for now). Figure 4(a) shows an RO alternative based on a high-speed sequential clock generated from an on-FPGA **phase-locked loop (PLL)** [21]. The subfigure shows two power wasters of this type clocked using an on-chip PLL. These power wasters can be implemented in a Stratix 10 **adaptive logic module (ALM)**. This circuit appears more similar to the standard single-clock sequential circuitry one would typically find in a user design, although it is only effective at wasting power when provided an input clock of hundreds of MHz [21]. The power consumption of this circuit is maximized when the PLL clock frequency approaches the oscillation frequency of a combinational RO.

The need for a high-speed input clock signal generated by a PLL in the power wasting circuitry of Figure 4(a) can be eliminated by rearranging the design input connections to implement a transparent latch or flip-flop triggered by an oscillating data signal [8, 32] (Figure 4(b)). Since flip-flops in Stratix 10 devices cannot be converted to latches, a flip-flop-based design was tested. A `D` flip-flop with an **active-low asynchronous clear control input (`ACLRn`)** and `D` input permanently connected to V_{CC} is used. The `Q` output of the flip-flop loops back to itself and drives its inverted clock and `ACLRn` inputs. Initially, both the clock input and `Q` output are low. When the enable signal is asserted, the clock input transitions from low to high and V_{CC} is clocked to the output `Q` of the flip-flop. Then, the high `Q` output is inverted at the `ACLRn` input of the flip-flop, forcing it to transition to a logic low, completing one oscillation.

A limitation of this approach is the need to utilize routing resources dedicated to driving the control signals of the ALM. Although the Stratix 10 LAB contains 10 ALMs (20 lookup tables), only



(a) RO + flop triggered by a PLL generated clock. (b) RO + flop triggered by oscillating signal.



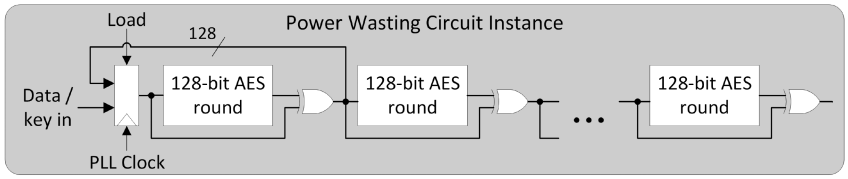
(c) Multiple instances of n-bit shift registers.

Fig. 4. Figures in (a) and (b) show the two alternative RO-based designs used to dissipate dynamic power. Design (c) shows the shift register-based waster.

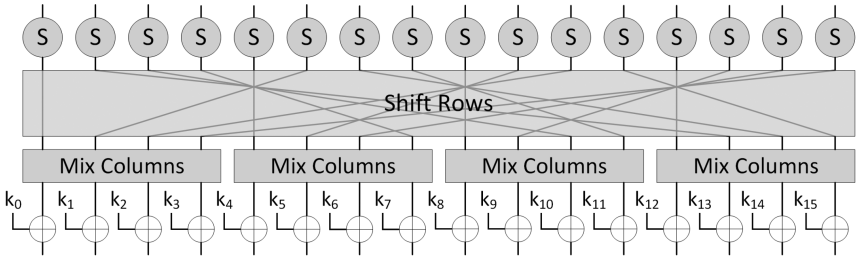
a single clock signal is supported per LAB [16]. Since each waster shown in Figure 4(b) requires a separate clock source, only a single waster can be instantiated in each LAB. In addition, the wasters illustrated in Figures 4(a) and 4(b) can be identified by diagnostic tools searching for short sequential paths [21, 25], although they do currently pass Amazon's **design rule checks (DRCs)**. To verify this claim, we synthesized the circuits using AWS EC2 F1's compilation software and successfully passed them through F1's design rule checking tools.

Design scanning for potential malicious circuits can become challenging when standard circuits are employed for wasting power. Ziener et al. [41] deploy a number of 16-bit shift registers (Figure 4(c)) to shape the power profile of the FPGA and effectively use them for power watermarking an IP core. Although shift registers are less effective in wasting power than the RO-based wasters, they are typically coupled with the functional logic of the IP core, which makes them practically indistinguishable from the rest of the design. Therefore, a malicious user can hide a multitude of shift registers in an IP core to cause voltage instability.

3.1.3 An AES-Based Power Waster. Recently, researchers have explored power waster implementations in FPGAs based on signal glitching that do not use combinational loops or short sequential paths [26, 28]. Glitching is known to consume significant dynamic power in FPGAs [6]. If not properly managed, differences in signal arrival times at the inputs of logic gates due to imbalanced path delays can cause unintentional and unnecessary output transitions. Studies [22, 23] have shown that glitch power can consume up to 19% of total power dissipation in some designs. Glitch-based wasters intentionally amplify signal glitching in FPGA logic and interconnect to consume significant dynamic power. Although these alternative power wasting methods require more



(a) N chained 128-bit AES rounds.



(b) Structure of a single 128-bit AES round.

Fig. 5. Design in (a) shows our unrolled waster based on glitching that uses copies of AES encryption rounds. (b) shows the structure of a standard 128-bit AES round used in our design.

logic than their RO-based counterparts, they can be crafted to avoid detection by netlist and bitstream scanners or via static power analysis.

In this work, we use the glitch-based power waster shown in Figure 5(a), previously evaluated for Cyclone V and Arria 10 FPGAs [28], using Stratix 10 devices. This circuit has the basic structure of a standard 128-bit **advanced encryption standard (AES)** circuit, although it does not perform encryption or any other useful function beyond wasting power. Unlike a standard 128-bit AES circuit that has 10 rounds, in our circuit, rounds are replicated to form a chain of a user-selected number of rounds. The structure of a round (Figure 5(b)) includes S-boxes (effectively 8-bit to 8-bit lookup tables, shown as S is the figure), shift rows (wire shuffling with no logic needed), mix columns, and XOR gates. Between rounds, an additional XOR gate has been added along with feed-forward paths to enhance glitching through timing imbalance. The power wasting efficiency of the AES-based circuit primarily results from the use of XOR gates in the AES rounds and between rounds. These gates, in conjunction with the imbalanced path delays, enhance signal glitching, leading to significant dynamic power consumption.

Our circuit can waste power effectively using a modest clock frequency of $\leq$50 MHz and does not require extensive hand tuning of delay paths to operate. From a structural standpoint, neither high-speed clocks nor combinational loops nor short sequential feedback paths are needed. To avoid being flagged for timing violations, the long combinational paths formed in the chained rounds can be marked as false paths that should be ignored for timing closure. The additional XOR gates inserted between rounds can be embedded in LUTs and masked with other logic. To locate this circuit (or one of its many variants) in a user design, a design rule checker must now consider the logic function of the circuit and not just its topographic structure to identify malicious intent. The checker cannot simply examine and flag designs with large numbers of LUTs implemented as XORs.

3.1.4 Power Waster Comparison. To assess the effectiveness of AES-based power wasters, we contrast the efficiency of the five wasters introduced in this section in Table 2. Ten thousand instances of the RO- and shift-register-based wasters and one AES-based waster containing 95

Table 2. Power Consumed per ALM (mW) for the Five Power Wasting Designs shown in Figures 3, 4, and 5(a)

	Shift Reg. (Figure 4(c))	RO+Flop (Figure 4(a))	AES-Based (Figure 5(a))	RO+Flop (Figure 4(b))	RO (Figure 3)
50 MHz	0.02	0.12	1.58	4.08	5.16
990 MHz	0.44	2.16	1.66		

Table 3. Resources used in AES-Based waster

Chained Rounds	1	20	40	60	80	100	120
LUTs (Avail.: 1,866k)	646 (0.03%)	19,999 (1.07%)	39,847 (2.14%)	59,695 (3.20%)	79,543 (4.26%)	99,391 (5.33%)	119,239 (6.39%)

The percentage logic LUT usage of a Stratix 10 1SX280 device is shown in parentheses.

rounds were used to generate the entries in the table. The unclocked RO circuits (Figures 3 and 4(b)) oscillate at frequencies >990 MHz. The RO + flop (PLL clock, Figure 4(a)), shift registers (Figure 4(c)), and AES-based circuits (Figure 5(a)) are clocked at 50 MHz and 990 MHz to generate comparative results. Results are represented in dynamic power dissipated per ALM. Each ALM includes two LUTs and four flip-flops.

As one might expect, the single-stage RO waster (Figure 3) is much more efficient in wasting power than the other four approaches. The RO + flop design of Figure 4(b) is efficient, but only one such circuit can be implemented in each 20-logic element LAB because of the architectural limitation that Stratix 10 LABs can only use one distinct clock input. The AES-based waster (Figure 5(a)) outperforms its shift-register-based counterpart (Figure 4(c)) and is competitive with the RO + flop design (PLL clock, Figure 4(a)) at high frequency. When all are run at the same 50 MHz clock frequency, the AES-based waster consumes considerably more power. For the AES-based circuit, F_{max} is much less than 50 MHz. Effectively, the circuit is overclocked, allowing frequent and repetitive glitch generation throughout the circuit.

The FPGA resources used by the AES-based power wasters with differing numbers of rounds are shown in Table 3. Clearly, the amount of logic needed for the circuits is more than a single RO (one LUT). However, previous work [20, 27] has shown that, typically, thousands of ring oscillators are needed to perform a voltage attack.

The result of the excessive glitching on the long combinational paths of the design can be seen in Figure 6. Each added round requires approximately 0.03% of the available LUTs and results in an almost linear increase of consumed power for a power waster clock frequency (f_{pw}) of 50 MHz. Similar to the activation of 30,000 RO-based wasters discussed in Section 3.1.1, activating more than 95 rounds results in total power consumption that exceeds 85 W and makes the device inaccessible after the completion of the experiment. With 120 rounds the board crashes right after the waster is turned on and no power measurement data can be extracted. All the experiments with configurations containing more than 91 rounds required a manual power cycle of the board in order to restore operation. Even at an arbitrarily chosen low f_{pw} of 4 MHz, the effects of increased glitching can be seen.

Figures 7(a) and 7(b) illustrate these effects more clearly for a 120-round AES waster clocked at 4 MHz and 50 MHz, respectively. Each triangle represents switching that occurs for a different clock cycle (e.g., in every 250 ns when the waster clocked at 4 MHz). The two sides of a triangle in the figures represent the delay of the slowest and fastest paths of the AES waster through an increasing

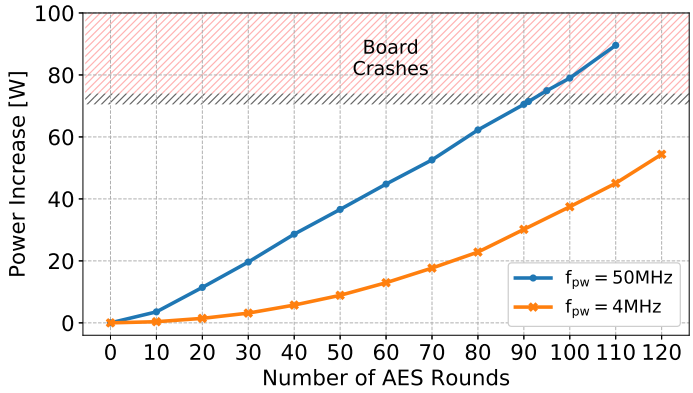
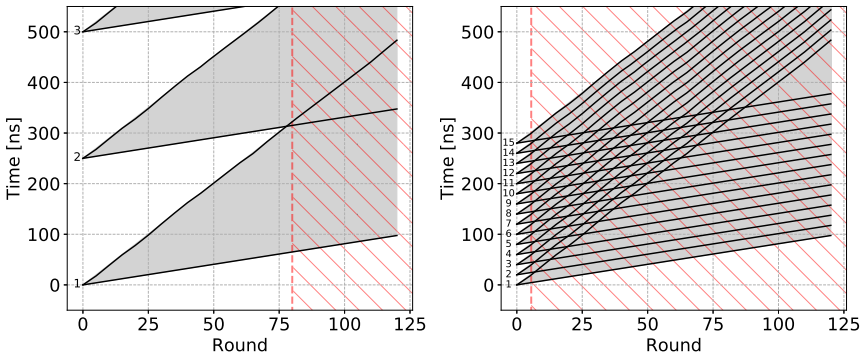


Fig. 6. Increase of power consumption for different numbers of AES chained rounds clocked at 50 MHz and 4 MHz. When the waster is clocked at 50 MHz, turning on 91 to 94 chained rounds sometimes results in a board crash (gray shaded area), and 95 to 110 rounds always causes the board to crash at the end of the experiment (red shaded area), while 120 rounds result in the immediate crash of the board and no power consumption data can be extracted. A manual power cycling of the board is necessary to recover from a crash. When the waster is clocked at 4 MHz, the power consumption for the examined number of rounds is insufficient to cause a board crash.



(a) Signal activity at 4 MHz (b) Signal activity at 50 MHz

Fig. 7. Propagation of signals in the AES waster. The two sides of a triangle represent the times at which the slowest and fastest paths will reach each round of the AES waster for an increasing number of rounds. The start of the clock cycle is labeled with a number on the left. The gray shaded area of a triangle depicts times when signal activity occurs in the rounds of the 120-round waster. The red shaded areas indicate maximum overlap of signal activity.

number of rounds. The gray shaded area of a triangle depicts times at which signal glitching occurs in the circuit. Increased glitching in later rounds occurs due to the imbalanced propagation paths. The power wasting efficiency of a round maximizes when the areas of the triangles overlap because these rounds will not be idle for any portion of the clock cycle, as illustrated by the red shaded areas. The figures demonstrate that increasing the clock frequency enhances signal transition overlap in earlier rounds of the chain.

Figure 8 illustrates the amount of power consumed per round in the AES waster, determined from the slope of the curves shown in Figure 6. Even when clocked at 4 MHz, the amount of power consumed per round in later rounds is substantial due to the overlapped paths shown in Figure 7.

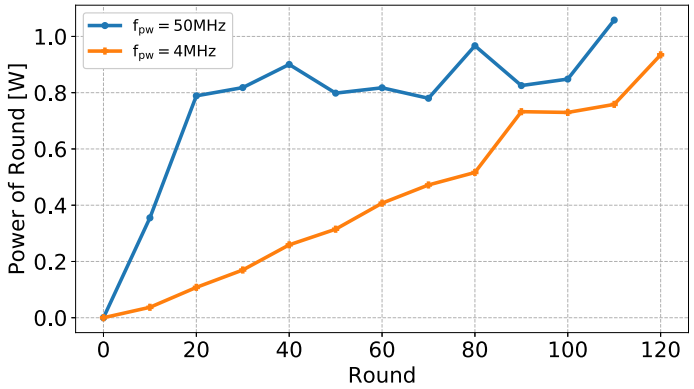


Fig. 8. Per-round power consumption of a 120-round AES waster clocked at 4 and 50 MHz. Later rounds have enhanced glitching leading to increased per-round power consumption.

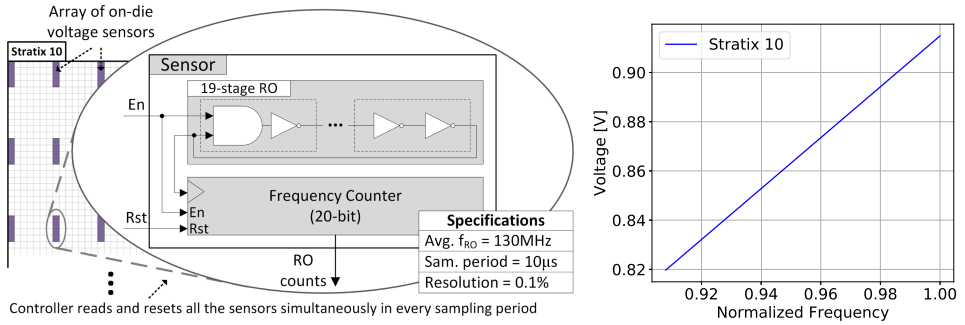
3.2 Stratix 10 PDN Characterization

In the remainder of this section, we use RO- and AES-based power wasters to evaluate Stratix 10 voltage response to instantaneous waster activation. These efforts require the acquisition of voltage information from on-chip voltage sensors fashioned from FPGA logic.

3.2.1 On-Die Voltage Sensors. A voltage sensing system is needed to observe the PDN response to adversarial power consumption during an attack. To determine on-chip voltage, we measure the voltage at selected positions of the PDN using ring oscillator-based voltage sensors. Figure 9(a) illustrates the architecture of the sensor network. The sensors are placed on the die forming a regular rectangular grid, which is sufficient to perform power analysis attacks [37]. Each sensor consists of a 19-stage RO triggering a 20-bit frequency counter. With 19 inverting stages, the oscillation period of the RO exceeds the critical path of the frequency counter, local delay variations are minimized [33], and RO stacking can be used in a single Stratix 10 LAB. Although shorter ROs are possible by inserting open latches in the ring to increase the path delay [39], the lack of built-in latch elements in the Stratix 10 device makes this technique unsuitable. The 19 inverting stages of the RO design shown in Figure 9(a) achieve an average frequency of 130 MHz in the Stratix 10 device. We use a 1 ms measurement period for the Stratix 10 sensor calibration and 10 µs for all experiments in the remainder of the article. The 10 µs period provides the capability to detect 0.1% frequency changes, corresponding to a sub-millivolt resolution in supply voltage measurement.

The frequency of the RO-based sensor decreases in a consistent way in response to voltage drops, and a calibration procedure is required to learn the correspondence between voltage and RO frequency. After calibration, frequency measurements made at each sensor can be translated into the voltages that cause them. The Stratix 10 FPGA is equipped with an on-chip voltage sensor [15] that is used to calibrate the RO sensors. In a series of calibration experiments, we varied the number of RO-based power wasters on the Stratix 10 device (Figure 3) from 8,000 up to 30,000, while monitoring readings from both the on-chip voltage sensor and RO sensors. The resulting calibration curve that relates frequency to voltage is shown in Figure 9(b).

Although RO operation can potentially influence chip temperature, voltage gradients have a much more immediate impact on the measured RO delay than temperature [2, 35]. To minimize heating effects, an idle period of a few seconds between calibration iterations was introduced. The ambient temperature during the calibration and characterization experiments was kept at 24 °C.



(a) Schematic of the monitoring sensor network. (b) Stratix 10 calibration curve.

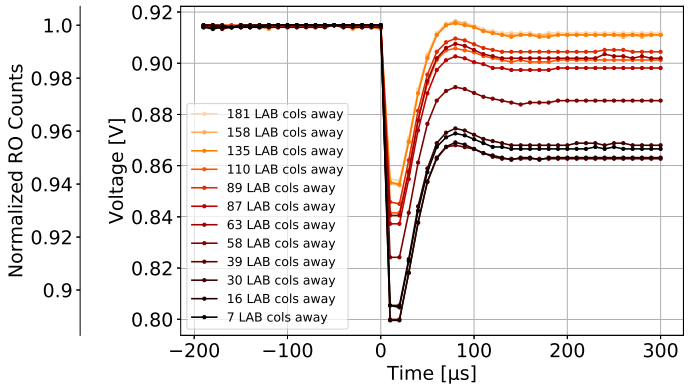
Fig. 9. Figure (a) illustrates the architecture of the sensor network and RO-based voltage sensor. Figure (b) shows the experimentally derived Stratix 10 calibration curve, which relates frequency changes to the supply voltage values that account for them. The frequency of a sensor is inversely proportional to the propagation delay of the oscillating signal.

Neither the on-board nor on-chip temperature sensor of the Stratix 10 device reported temperature fluctuations during the calibration and characterization processes. This result indicates that thermal effects are negligible in our experimentation.

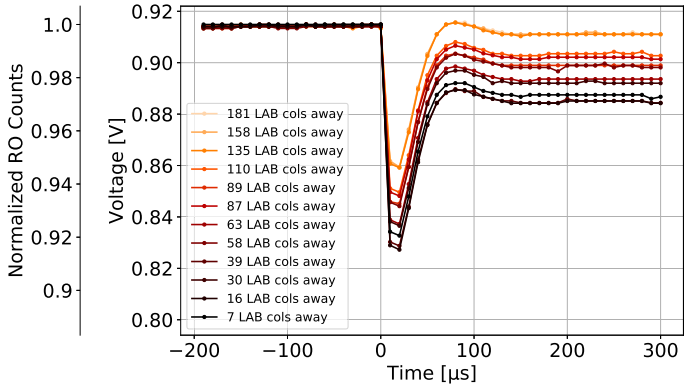
3.2.2 PDN Response to Power Consumption. We evaluate the sensor network using two types of power wasting circuits, one using 30,000 ring oscillators (Section 3.1.1) and the second using a 95-round AES-based waster (Section 3.1.3) clocked at 50 MHz. For experimentation, we allocate an area of 6,656 LABs (104 rows by 64 columns) to RO-based power wasting circuitry. For the AES-based waster, we allocate an area of 16,384 LABs (128 rows by 128 columns), which corresponds to approximately 15% of the total Stratix 10 FPGA area. Each waster is the minimum-sized circuit of the selected type (RO-based or AES-based) needed to cause a board reset. The area difference for the two waster types is related to their relative difference in power consumption (Table 2). The AES-based waster requires more area to consume a similar amount of power.

To evaluate the Stratix 10 1SX280 PDN response to high power consumption, experiments were performed with sensors placed at various distances away from a region with power wasters. The plots shown in Figure 10(a) were generated with 30,000 RO-based power wasters. The plots in Figure 10(b) were generated with a 95-round AES-based waster. At time 0 the power wasters turn on and the frequency of the sensors, or equivalently their supply voltages (Figure 10), drops in response to the attacker's power consumption. The supply voltage measured by each sensor initially drops, undershoots, and then settles back to a steady-state voltage that is lower than the nominal 0.9 V for as long as the power wasters remain active. For the RO-based waster, the supply voltage at the center of the power consumption area drops to a minimum of 800 mV and reaches a steady state of 863 mV. Sensors farther away observe a similar behavior but a smaller magnitude of voltage drop. The supply voltage at the center of the AES-based waster drops to a minimum of 827 mV and reaches a steady state of 884 mV.

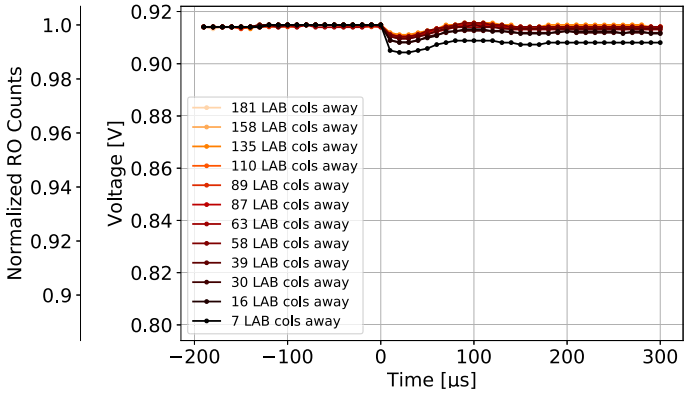
The feed-forward paths, XOR gates, and extra rounds used to implement the 95-round AES-based waster significantly enhance its power wasting capabilities. For comparison, the supply voltage effects of activating a standard 10-round AES circuit are shown in Figure 10(c). The supply voltage at the center of the power consumption area drops by a modest 10 mV to a minimum of 905 mV.



(a) RO-based waster containing 30,000 instances



(b) AES-based waster containing 95 rounds, clocked at 50 MHz



(c) Unrolled standard AES core containing 10 rounds, clocked at 50 MHz

Fig. 10. Normalized RO sensor counts (left axis) and their corresponding voltages (right axis) measured by sensors before and during a power wasting attack that begins at time 0. The legend shows the distance between each sensor and the center of the power wasting region.

Table 4. Peak Voltage Drop per ALM (μV) for the Five Power Wasting Designs Shown in Figures 3, 4, and 5(a)

	Shift Reg. (Figure 4(c))	RO+Flop (Figure 4(a))	AES-Based (Figure 5(a))	RO+Flop (Figure 4(b))	RO (Figure 3)
50 MHz	0.09	0.83	2.11	5.22	6.78
990 MHz	0.63	3.14	2.22		

Table 4 illustrates the peak voltage drop per ALM for each of the five wasters described in Section 3 and analyzed in Table 3. The AES waster clocked at 50 MHz achieves good efficiency without the need for a fast PLL-generated clock or asynchronous loops.

4 LOCALIZING VOLTAGE DROOPS

PDN attacks require power consumption, transiently or in steady state, beyond what the power distribution network can handle. Our results have shown that the power consumption of one adversarial block can cause a measurable and significant difference in the voltage of other blocks. Circuits closest to the power consumption experience the largest voltage drop, and the voltage drop becomes smaller moving farther away (Figure 10). The voltage gradients effectively provide a map pointing toward the center of the attack, which will have the lowest voltage. A spatially distributed network of voltage sensors can enable a resource manager to monitor voltage gradients and identify the source of any attacks that occur. The resource manager can then prevent further instances of the same attack by taking the offending application offline or banning it from co-tenant settings.

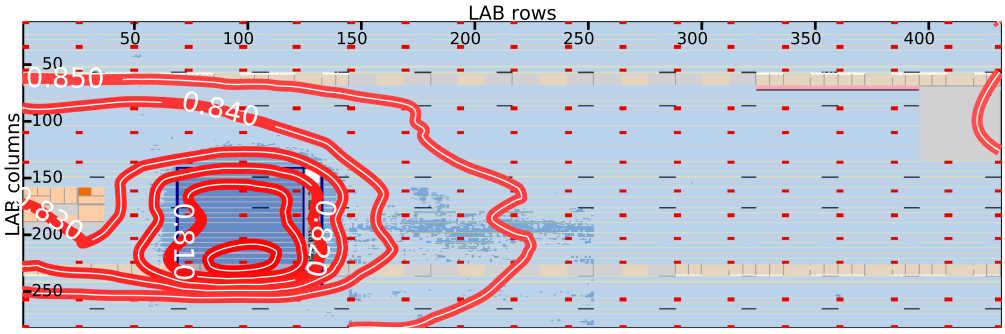
4.1 Sensor Network

A network of 218 sensors uniformly placed across the Stratix 10 FPGA fabric is created to monitor voltage fluctuations and log the data for processing. Each sensor utilizes 39 ALMs and 20 flip-flops. The controller logic that logs the sensor data to memory and the 218 sensors collectively consume less than 1% of the available ALMs and flip-flops.

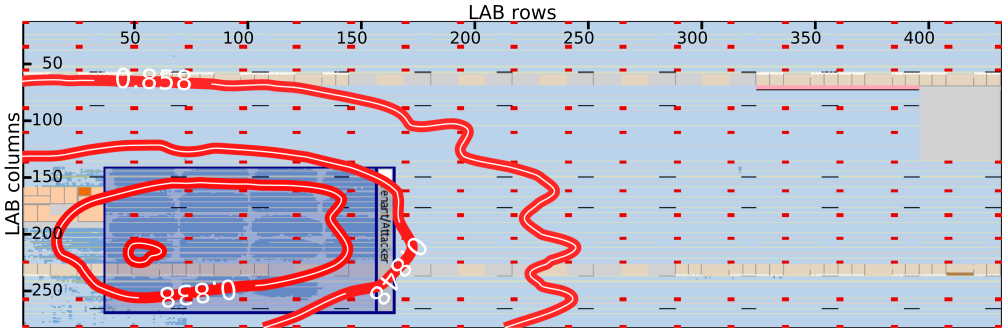
Figure 11 shows the voltage contours of the chip based on sensor data during two different power attacks. The specific data used to generate the plot is the minimum value observed by each sensor in the 800 μs time period that contained the attack. A cubic interpolation algorithm reconstructs the smoothed voltage contours from the samples collected at the discrete sensor locations.

In the first attack (Figure 11(a)), the attacker turns on 30,000 power wasters within an area spanning 6,656 LABs (104 rows by 64 columns) shown in dark blue in the figure. The voltage at the center of the attack drops below 800 mV. In the second attack (Figure 11(b)), the attacker turns on an AES-based waster containing 95 rounds within an area spanning 16,384 LABs (128 rows by 128 columns). The AES-based waster has a somewhat lower impact, dropping the voltage 73 mV below 900 mV to 827 mV at the center of the disruption while the voltage in almost half of the FPGA fabric drops below 858 mV.

Figure 12 shows voltage plotted against distance from the center of the attack on the Stratix 10 device. The black line in the figure was generated by plotting the minimum voltages observed during the attack by the sensors located at the LAB row labeled as 100 in Figure 11(a). The red line in the figure was generated by plotting the minimum voltages the sensors observed at the LAB row labeled as 50 in Figure 11(b). Although the enabling of the two wasters results in similar power consumption, the RO-based waster leads to a steeper gradient because power consumption is denser in the attack area.



(a) RO-based waster containing 30,000 instances.



(b) AES-based waster containing 95 rounds.

Fig. 11. Map of voltage contours on chip during power attacks, reconstructed from sensor data. Dark blue rectangle denotes location of the attacker's power waster circuits. Red rectangles denote the 218 sensors.

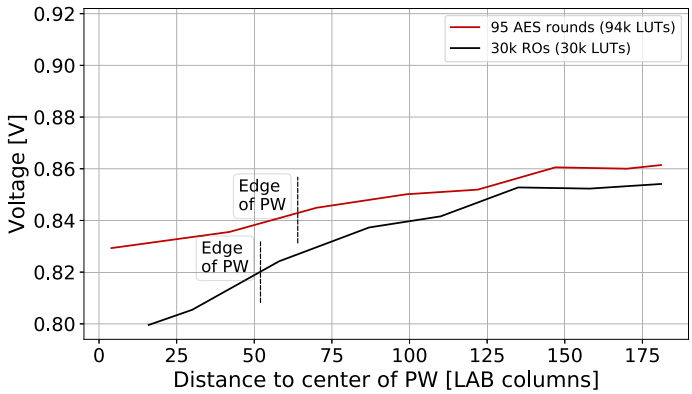


Fig. 12. Voltage change across distance for the RO- and AES-based wasters.

4.2 Minimizing the Number of Sensors

The goal of the sensor network is to identify the source of any attacks that occur. The spatial extent of the voltage drops caused by PDN attacks makes it hard for an attacker to mask his or her identity. Given our intention to monitor and process sensor data in real time, we want to determine the minimum number of sensors required to localize an attacker occupying a certain area of LABs.

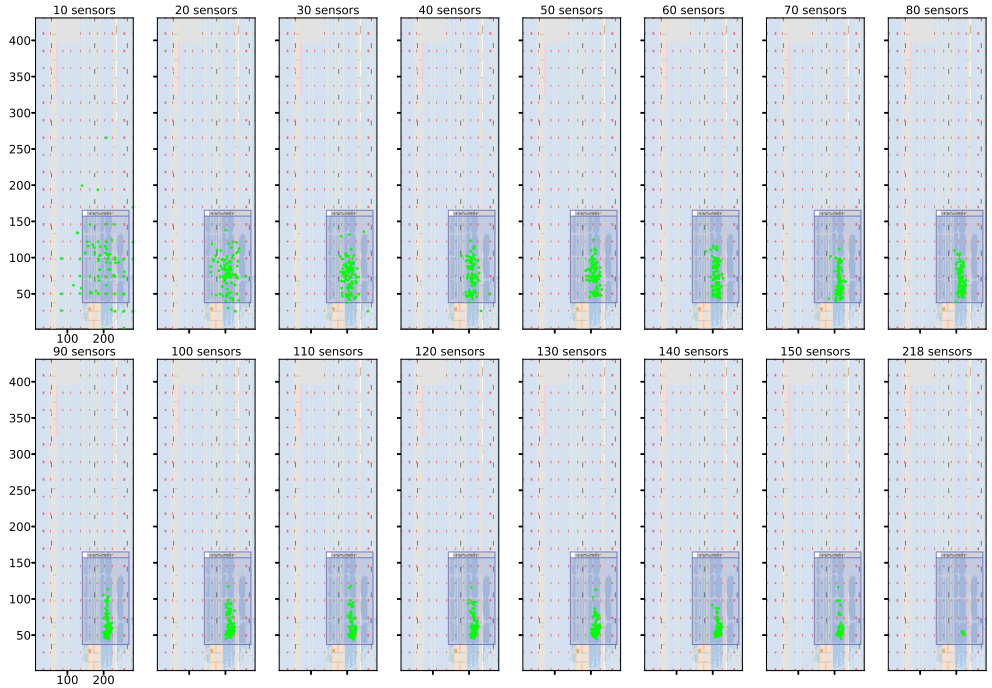


Fig. 13. Locating the attacker with the minimum number of sensors required. Green marks represent the predicted center of attack area based on a randomly selected subset out of the total 218 sensors. Each subplot contains 100 points. Note that although the predictions converge to a specific location when all the 218 sensors are used (lower right corner of figure), the center of the disruption and the center of the attacker area may not coincide.

Considering the AES-based attack scenario of a 16,384 LABs area (128 rows by 128 columns) from the previous subsection as an area a tenant might occupy, we examine how precisely the attacker can be located using different numbers of sensors. For each number of sensors, we randomly select 100 different subsets containing that number of sensors, and from each subset we try to predict the location of the attack circuitry.

The result of this analysis is shown in Figure 13. The green dots on each plot are 100 different predictions of the attacker location. We found that when we utilize data from all 218 sensors of the network, all predictions converge to a specific location inside the attack area (the bottom, rightmost subplot of Figure 13). The epicenter of the voltage disruption and the topological center of the attacker area might not precisely coincide. Using this result as a best case, we evaluated results for a sensor count of 10 and then increased sensor counts in increments of 10. As one might expect, utilizing data collected from more sensors increases the precision of the predictions. Using only 10 sensors leads to predictions that point outside of the attacking circuit. Predictions start to point to locations inside the attack area when 50 or more sensors are used.

To quantify the error of each number of sensors, we use the Euclidean distance between the mean of the 100 location predictions of a given subset of sensors and the mean of the 100 location predictions that utilize all 218 sensors. The distance errors for each configuration are expressed in LABs and shown in Figure 14. The 70-sensor predictions, for example, on average predict the center of the attack to be 14 LABs away from the predicted center when all 218 sensors are used.

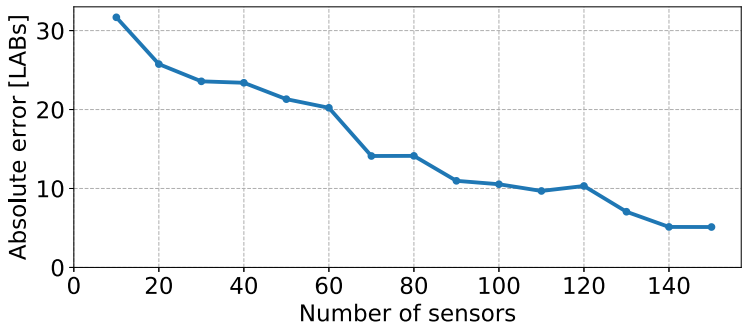


Fig. 14. Absolute error expressed in LABs of the Euclidean distance between the mean of the 100 location predictions of a given sensor subset and the mean of the 100 location predictions obtained when all 218 sensors are used (see the bottom, rightmost subplot of Figure 13).

5 ON-CHIP MONITORING AND ATTACK THROTTLING

In this section, we enhance the voltage sensing network described in Section 4 to include direct remediation to prevent a voltage attack. The network is augmented with processing capabilities and clock throttling circuitry for suspected power wasting regions. Our solution is evaluated using an AES-based waster that attempts to crash the FPGA-based board. We show that our system is able to respond quickly enough to prevent the board from going into reset by throttling the clock attached to the power waster without affecting surrounding user logic.

Our approach is designed to respond quickly to significant drops in on-FPGA voltage at the onset of an attack, as shown in the $L(\mathrm{di/dt})$ voltage drop region at the left in Figure 10(b). These voltage drops exceed what would typically be expected from the activation of standard user circuitry.

5.1 Remediation Overview

Our monitoring and attack throttling approach consists of both hardware and software components. The steps involved in this system include:

(1) Our system periodically collects on-FPGA voltage values from the voltage sensor network. Multiple sensors are assigned to each region in the FPGA. These values are passed to a microprocessor.
(2) The processor compares the values to a predetermined threshold to determine if an attack is potentially in progress.
(3) If the measured voltage in a region is less than an acceptable threshold, the clock buffer to the associated region is deactivated, throttling the attack.

The details and effectiveness of our system are examined in the remainder of this section.

5.2 System Infrastructure

Figure 15 shows an overview of the monitoring system implemented on a Stratix 10 SX (1SX280) FPGA device. The sensor network, an ARM-based **Hard Processor System (HPS)**, and interfacing logic implement the monitoring system. The system isolates each tenant's circuity in a clock region. Each region's clock buffers are controlled by the ARM-HPS. This component periodically collects voltage information in the form of RO counts from the voltage sensors. These values are analyzed to detect incidents of aggressive power consumption. If the ratio of the reported RO counts to expected (nominal) RO counts in a clock region is below a predetermined threshold, the ARM-HPS disables the clock buffers for that region, forcing all sequential circuits to freeze.

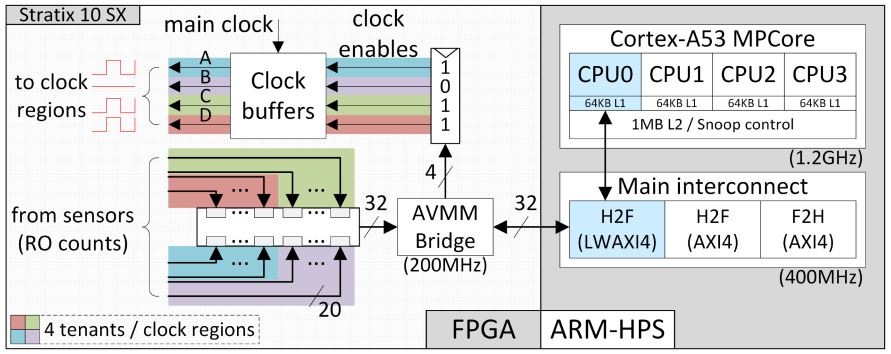


Fig. 15. On-chip monitoring system overview.

The ARM-HPS contains four A53 64-bit CPU cores with floating-point units, two levels of cache memory, and a 256 kB on-chip memory. The sensor network interfaces to the ARM-HPS via one 20-bit register per voltage sensor. The ARM-HPS accesses each register via the **Avalon Memory-Mapped (AVMM)** bridge and the AMBA **Lightweight Advanced eXtensible Interface (LWAXI4)**. The AVMM bridge also connects the ARM-HPS to the clock to enable signals of the clock buffers. The CPU cores are clocked at 1.2 GHz, the LWAXI4 bridge is clocked at 400 MHz, and the FPGA logic of the sensor network is clocked at 200 MHz. The monitoring and clock throttling software run on bare metal on CPU0 (see Figure 15). To load the executable on the CPU, a **first-stage boot loader (FSBL)** program is used, which initializes the HPS to run bare metal applications.

5.2.1 Clock Regions and Sensors. The 1SX280 device contains 117,072 LABs (432 rows by 271 columns), of which 111,300 LABs can be used by the designer. The remaining LABs are reserved for use by support circuitry for the ARM-HPS, FPGA I/O logic, and system peripherals. In our prototype, 65,536 LABs (256 rows by 256 columns), 59% of the available logic, are allocated for use by multi-tenant applications. The 65,536 LABs span four clock regions that contain 16,384 LABs (128 rows by 128 columns) each. These regions can be allocated to four distinct users. Figure 16 depicts the clock regions and their relative locations on the Stratix 10 1SX280 fabric with four rectangles of different colors (i.e., cyan, purple, green, and red).

A total of 36 uniformly placed sensors were used to localize an attacker within one of the four 16,384 LAB regions. Figure 16 shows the 36 sensors in the regions as numbered rhombuses. The sensor indices correspond to the order in which the ARM-HPS reads the sensor values. The sensors are placed so that the boundary between two regions is equidistant from sensors in both regions.

Table 5 shows the FPGA resources required by the monitoring system. The 36 sensors in the four clock regions collectively consume less than 0.1% of the available ALMs and flip-flops in the 1SX280 device. The nine sensors in each clock region occupy 18 out of the 16,384 LABs in each region. The monitoring system interface to the ARM-HPS (774 ALMs) has minimal hardware requirements. The interface is located outside of the four clock regions. A total of 18,363 ALMs (Table 5) are consumed by the FPGA portion of the ARM-HPS that interfaces to memory and bridge interconnects. The dynamic power consumption of the ARM-HPS FPGA portion and hard processor (1.14 W) was measured to be more than the dynamic power consumption of the sensor network (0.09 W) and interfacing logic (>0.01 W). This result indicates a negligible heating effect by the monitoring system.

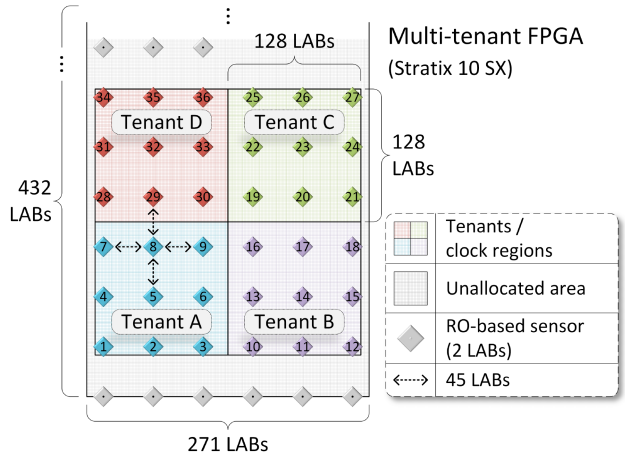


Fig. 16. The figure illustrates a partitioning of an area spanning 65,536 LABs (256 rows by 256 columns) into four clock regions that can be used by multi-tenant applications. Each region contains nine RO-based sensors. Their relative locations on the device are indicated by colored rhombuses. The four regions span roughly 60% of the FPGA logic area.

Table 5. FPGA Resources Used in the On-Chip Monitoring System

Type	Sensor Network (36 sensors)	Interfacing Logic	ARM-HPS (FPGA Portion)
ALMs (Avail.: 933 k)	720 (<0.1%)	774 (<0.1%)	18,363 (2%)
Flip-flops (Avail.: >3.732 M)	720 (<0.1%)	1,519 (<0.1%)	26,046 (0.7%)
Memory (Avail.: 229 Mibit)	–	–	2.25 Mibit (1%)

5.2.2 Threshold Calibration. A decision to throttle the clock of a specific clock region requires two pieces of information, voltage sensor information indicating that an attack is in progress and the likely location of the attack. As described in Section 3.2.1, the voltage at an on-FPGA location is approximated by a sensor that counts RO oscillations over a fixed period of time. A significant drop in RO count below a certain threshold value indicates voltage instability that can potentially reset the FPGA board. In this work, we identify a threshold value via calibration that helps identify a potential attack. If the ratio of measured RO counts in a region to expected (nominal) counts is below a *threshold_ratio*, an attack is considered to be in progress, requiring immediate remediation.

The calibration method used to determine *threshold_ratio* for our experiments included the following steps:

- Using the AES-based waster introduced in Section 3.1.3, we determined the minimum number of AES rounds required to crash the board (see Figure 6). A waster that contains 91 chained rounds was found to crash the Stratix 10 FPGA-based board in <1% of trials. A waster with 95 or more rounds resulted in a board crash in all trials.
- In the 1SX280 device under typical operating conditions, the sensors of the attack region report an average RO count decrease to 92% to 93% of their nominal counts when the waster

ALGORITHM 1: Monitoring sensors and switching off clock regions upon detecting an attack

```
1:  nom_counts[N] ← determine_nominals();          ▷ N is the number of clock regions
2:  start_sensors();
3:  while True do
4:      x ← read_sample_when_is_ready();           ▷ it returns the sums of the N regions
5:      for i ← 0 to N − 1 do
6:          ▷ calculate the reported-to-nominal RO count ratio of region i
7:          y ← x[i] / nom_counts[i];
8:          if y ≤ threshold_ratio then
9:              disable_clock(i);
10:         end if
11:     end for
12: end while
```

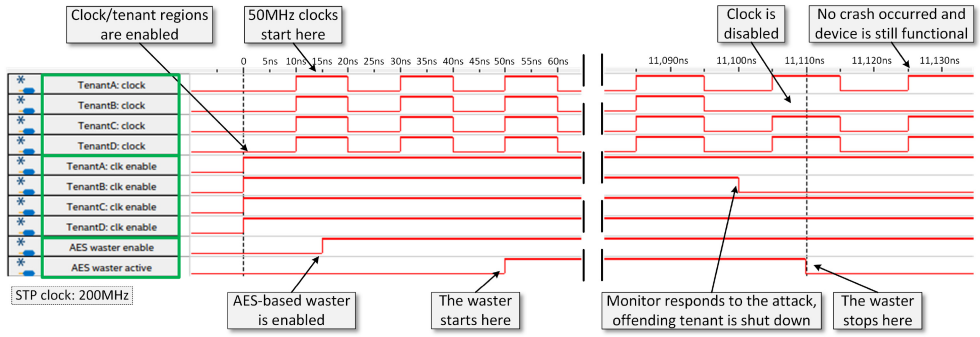
is activated. This RO count decrease to 93% (0.93) was set as the *threshold_ratio* for further experimentation.

Our experimentation showed that the extracted threshold consistently identified an attack and no recalibration was necessary during experimentation. However, the threshold selection process should be adapted when operating conditions change (e.g., temperature). For system operation following a significant on-FPGA temperature change, the threshold value could be dynamically adjusted based on temperature values measured with an on-chip sensor. For example, Stratix 10 devices contain an on-chip temperature sensing diode that allows a user to monitor core die temperature once per millisecond [15]. These per-temperature threshold adjustments would need to be characterized for the device prior to system execution.

5.2.3 Attack Detection and Remediation. Following the collection of voltage sensor RO counts every 5 µs, an algorithm is executed in software to evaluate possible attacks. A sample period of 5 µs provides sufficient time to obtain reliable RO counts from the voltage sensors. This sample period is also brief enough to allow the ARM-HPS to respond to the attack and suppress it. Pseudocode for the algorithm executed on the processor is shown in Algorithm 1. The algorithm proceeds as follows. At the beginning (line 1), a function determines the nominal average RO count sum of each clock region across nine sensors without activated wasters. These sums are determined by collecting and averaging 10,000 samples for each sensor over a time period of 50 ms. The resulting values are used in subsequent steps of the algorithm to calculate the average reported-to-nominal RO count ratio of a region. This step is executed only once, before the monitoring of the clock regions begins.

Next, monitoring begins (line 2) and the execution waits at line 4 for the sensor data of the first sampling period to become available (e.g., after 5 µs). The RO counts from each region i are summed to form a combined count $x[i]$. The algorithm then calculates the ratio of the reported RO count $x[i]$ to nominal RO count of each region i (line 7). The resulting values are compared against the *threshold_ratio* (line 8) determined during calibration. If the reported-to-nominal RO count ratio of a region is less than or equal to the threshold, the algorithm disables its clock buffer (line 9). If it is greater than the threshold, no action is taken. When all regions have been checked, the execution loops back to line 4 where it waits for the next sample.

Floating-point division operations (line 7) and I/O read transactions (line 4) are the most time-consuming operations that take place during the execution of the *while-loop* in Algorithm 1. Executing a single division requires 17.8 ns, on average, while a single read transaction takes 480.0 ns,



(a) Start of the attack (b) End of the attack

Fig. 17. Intel SignalTap waveforms capturing the start and end of a voltage attack attempting to crash the FPGA board using the 95-round AES-based waster and prevented by the on-chip monitoring system.

on average. Without considering line 9, which is executed only when an attack is detected, the loop within a single iteration performs two I/O reads at line 4 and N divisions because of line 7. Neglecting the single-cycle operations of the loop (e.g., index increment, etc.), a single iteration of the loop requires approximately 1.03 µs, on average, when $N = 4$.

5.3 System Evaluation

To evaluate the effectiveness of our remediation system, we performed a series of experiments using the Stratix 10 device. The experimental setup included the system shown in Figures 15 and 16. A 95-stage AES-based waster was placed in the clock region labeled `Tenant B` in Figure 16. After loading the monitoring application on the ARM-HPS, Algorithm 1 is executed. Waveforms obtained in real time from the Intel SignalTap logic analyzer are shown in Figure 17. The clock enable and clock signals of the four clock regions were controlled by the ARM-HPS during the test, as shown in the waveforms in Figure 17.

Figure 17(a) captures the beginning of the attack. At time 0 the monitoring application enables the four clock regions by asserting the enable signals of the clock buffers (`TenantA: clk enable` - `TenantD: clk enable`). At time 10 ns the clock buffers are activated by the clk enable signals and feed the clock signals (`TenantA: clock` - `TenantD: clock`) to the four regions. To initiate the attack, the enable signal of the waster (`AES waster enable`) is asserted at time 15 ns by user logic, which then activates the AES-based waster one 50 MHz clock cycle later at time 50 ns. A reference signal `AES waster active` is generated by the waster for reference only and is shown on the waveforms.

Figure 17(b) captures the end of the attack and the successful prevention of a board crash. The monitoring system responds to the reduced RO counts reported by the sensors in the `Tenant B` region by switching off its clock buffer (`TenantB: clk enable`). The clock signal of `Tenant B` is then disabled and the waster is forced to a halt. The ARM-HPS response is fast enough to prevent a device crash in the 21 µs range. The clock signals of the three remaining tenants continue oscillating. The attack shown in the waveforms was performed 1,000 times. The monitoring system was able to stop the attack for all 1,000 attempts and prevent a board crash. The response time of the system to the attack is 9.95 µs, on average.

Algorithm 1 could be modified to consider multiple consecutive sample periods of below-threshold average RO counts rather than just one sample period. These adjustments would depend

on environmental factors such as expected user circuit behavior and circuit operating temperature. The algorithm could also be adjusted to consider steep drops in voltage across sample periods.

5.4 Limitations

The following points summarize the limitations of our monitoring approach:

(1) **Clock gating-based mitigation.** In the current prototype, clock gating is used to suppress a potential attacker. This approach is only effective if the attack circuitry depends on a clock signal for operation (e.g., the AES-based waster used for this work). This approach would not be effective for wasters without globally distributed clocks (e.g., Figure 4(b)), although our sensors would detect the attack and flag the user, possibly preventing future attacks.
(2) **Local user clocks.** Often, a designer uses clock resources (e.g., clock managers, phase lock loop modules, etc.) to generate custom clock signals for internal use in an allocated region. For a secure cloud-based system, the reference clock driving these clock resources could be provided by global clock-buffers controlled by the monitoring system so it can be disabled when needed. For example, in AWS EC2 F1 instances, user clock signals are generated by a PLL in the shell [10] and could be suppressed in case of attack.
(3) **Response to waster-induced impulses.** Our method requires that a waster be activated for at least 5 µs for an attack to be detected. Repetitive, rapid activation and deactivation of the wasters may not be detected.

6 CONCLUSION AND FUTURE WORK

In this article, we evaluate power wasting circuits that can be implemented in multi-tenant FPGA platforms by a potential attacker. The power efficiencies of the circuits are evaluated. A power waster, based on an expandable collection of AES rounds, is shown to consume significant power even when operating at low frequency. This circuit is used to characterize the on-chip voltage response of an Intel Stratix 10 device. We have also described a new on-FPGA remediation approach that collects voltage values from multiple tenants in real time and throttles the clock to any region suspected of malicious behavior. Our approach can respond to an attack within 21 µs and successfully prevented a series of voltage attacks from causing board reset.

Several extensions of this research can be considered in future work. Although our approach can suppress board reset, it is not fast enough to prevent power-waster-induced timing faults in neighboring circuits [28]. New remediation approaches that locally suppress clocks on the FPGA could be considered. The evaluation of the monitoring system discussed in this manuscript examined attack scenarios where wasters were constrained to a single tenant region and it could be extended to consider scenarios with multiple malicious tenants. It may be possible to profile these attacks and identify them using sensor readings from multiple regions. Our approaches could also be tested on FPGAs from other commercial vendors. Further investigation of the role of on-board power distribution systems during power attacks is also necessary to assist mitigation. Power supply systems, for instance, could be designed to better tolerate brief out-of-specification current peaks and enhance attack detection.

REFERENCES

[1] Md Mahbub Alam, Shahin Tajik, Fatemeh Ganji, Mark Tehranipoor, and Domenic Forte. 2019. RAM-Jam: Remote temperature and voltage fault attack on FPGAs using memory collisions. In *2019 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC'19)*. 48–55.
[2] Abdulazim Amouri, Jochen Hepp, and Mehdi Tahoori. 2015. Built-in self-heating thermal testing of FPGAs. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 35, 9 (2015), 1546–1556.
[3] AWS. 2020. Amazon ECE F1 Instances. https://aws.amazon.com/ec2/instance-types/f1/.

[4] Adrian M. Caulfield, Eric S. Chung, Andrew Putnam, Hari Angepat, Jeremy Fowers, Michael Haselman, Stephen Heil, Matt Humphrey, Puneet Kaur, Joo-Young Kim, Daniel Lo, Todd Massengill, Kalin Ovtcharov, Michael Papamichael, Lisa Woods, Sitaram Lanka, Derek Chiou, and Doug Burger. 2016. A cloud-scale acceleration architecture. In *IEEE/ACM International Symposium on Microarchitecture (MICRO'16)*. 1–13.

[5] Alibaba Cloud. 2018. Deep Dive into Alibaba Cloud F3 FPGA as a Service Instances. https://www.alibabacloud.com/blog/deep-dive-into-alibaba-cloud-f3-fpga-as-a-service-instances_594057.

[6] Naveen Kumar Dumpala, Shivukumar B. Patil, Daniel Holcomb, and Russell Tessier. 2017. Energy efficient loop unrolling for low-cost FPGAs. In *IEEE International Symposium on Field-Programmable Custom Computing Machines (FCCM'17)*. 117–120.

[7] Ilias Giechaskiel, Kasper Rasmussen, and Jakub Szefer. 2019. Reading between the dies: Cross-SLR covert channels on multi-tenant cloud FPGAs. In *IEEE International Conference on Computer Design (ICCD'19)*. 1–10.

[8] Ilias Giechaskiel, Kasper Bonne Rasmussen, and Jakub Szefer. 2019. Measuring long wire leakage with ring oscillators in cloud FPGAs. In *International Conference on Field Programmable Logic and Applications (FPL'19)*. 45–50.

[9] AWS GitHub. 2020. AFI Power. https://github.com/aws/aws-fpga/blob/master/hdk/docs/afi_power.md.

[10] AWS GitHub. 2020. Clocks and Reset. https://github.com/aws/aws-fpga/blob/master/hdk/docs/AWS_Shell_Interface_Specification.md#ClocksNReset.

[11] Dennis R. E. Gnad, Cong Dang Khoa Nguyen, Syed Hashim Gillani, and Mehdi B. Tahoori. 2019. Voltage-based covert channels in multi-tenant FPGAs. *Cryptology ePrint Archive* Report 2019/1394 (2019). https://eprint.iacr.org/2019/1394.

[12] Dennis R. E. Gnad, Fabian Oboril, Saman Kiamehr, and Mehdi B. Tahoori. 2016. Analysis of transient voltage fluctuations in FPGAs. In *2016 International Conference on Field-Programmable Technology (FPT'16)*. 12–19.

[13] Dennis R. E. Gnad, Fabian Oboril, Saman Kiamehr, and Mehdi B. Tahoori. 2019. An experimental evaluation and analysis of transient voltage fluctuations in FPGAs. *IEEE Transactions on VLSI Systems* 26, 10 (2019), 1817–1830.

[14] Dennis R. E. Gnad, Fabian Oboril, and Mehdi B. Tahoori. 2017. Voltage drop-based fault attacks on FPGAs using valid bitstreams. In *International Conference on Field Programmable Logic and Applications (FPL'17)*. 1–7.

[15] Intel Corporation. 2019. *Intel Stratix 10 Analog to Digital Converter User Guide*. Intel Corporation.

[16] Intel Corporation. 2020. *Intel Stratix 10 Logic Array Blocks and Adaptive Logic Modules User Guide*. Intel Corporation. https://www.intel.com/content/www/us/en/programmable/documentation/wtw1441782332101.html.

[17] Chenglu Jin, Vasudev Gohil, Ramesh Karri, and Jeyavijayan Rajendran. 2020. Security of cloud FPGAs: A survey. *arxiv* arXiv:2005.04867 (2020). http://arxiv.org/abs/2005.04867.

[18] Ahmed Khawaja, Joshua Landgraf, Rohith Prakash, Michael Wei, Eric Schkufza, and Christopher J. Rossbach. 2018. Sharing, protection, and compatibility for reconfigurable fabric with AmorphOS. In *USENIX Symposium on Operating Systems Design and Implementation (OSDI'18)*. 107–127.

[19] Oliver Knodel, Patrick Lehmann, and Rainer G. Spallek. 2016. RC3E: reconfigurable accelerators in data centres and their provision by adapted service models. In *IEEE International Conference on Cloud Computing*. 19–26.

[20] Jonas Krautter, Dennis R. E. Gnad, and Mehdi Tahoori. 2018. FPGAhammer: Remote voltage fault attacks on shared FPGAs, suitable for DFA on AES. *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2018, 3 (2018), 44–68.

[21] Jonas Krautter, Dennis R. E. Gnad, and Mehdi B. Tahoori. 2019. Mitigating electrical-level attacks towards secure multi-tenant FPGAs in the cloud. *ACM Transactions on Reconfigurable Technology and Systems (TRETS)* 12, 3 (2019), 1–26.

[22] Fei Li, Deming Chen, Lei He, and Jason Cong. 2003. Architecture evaluation for power-efficient FPGAs. In *ACM/SIGDA International Symposium on Field Programmable Gate Arrays (FPGA'03)*. 175–184.

[23] Fei Li, Yizhou Lin, Lei He, Deming Chen, and Jason Cong. 2005. Power modeling and characteristics of field programmable gate arrays. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 24, 11 (2005), 1712–1724.

[24] Dina Mahmoud and Mirjana Stojilović. 2019. Timing violation induced faults in multi-tenant FPGAs. In *Design, Automation & Test in Europe Conference & Exhibition (DATE'19)*. 1745–1750.

[25] Kaspar Matas, Tuan La, Nikola Grunchevski, Khoa Pham, and Dirk Koch. 2020. Invited tutorial: FPGA hardware security for datacenters and beyond. In *ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (FPGA'20)*. 11–20.

[26] Kaspar Matas, Tuan Minh La, Khoa Dang Pham, and Dirk Koch. 2020. Power-hammering through glitch amplification–Attacks and mitigation. In *IEEE 28th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM'20)*. 65–69.

[27] George Provelengios, Daniel Holcomb, and Russell Tessier. 2019. Characterizing power distribution attacks in multi-user FPGA environments. In *International Conference on Field Programmable Logic and Applications (FPL'19)*. 194–201.

[28] George Provelengios, Daniel Holcomb, and Russell Tessier. 2020. Power wasting circuits for cloud FPGA attacks. In *International Conference on Field Programmable Logic and Applications (FPL'20)*. 231–235.

[29] Daniel Chase Savory. 2012. *Power Side-Channel DAC Implementations for Xilinx FPGAs*. Master's thesis. Dept. of Electrical and Computer Engineering, Brigham Young University.
[30] Falk Schellenberg, Dennis R. E. Gnad, Amir Moradi, and Mehdi B. Tahoori. 2018. An inside job: Remote power analysis attacks on FPGAs. In *Design, Automation & Test in Europe Conference & Exhibition (DATE'18)*. 1111–1116.
[31] Linda L. Shen, Ibrahim Ahmed, and Vaughn Betz. 2019. Fast voltage transients on FPGAs: Impact and mitigation strategies. In *IEEE International Symposium on Field-Programmable Custom Computing Machines (FCCM'19)*. 271–279.
[32] Takeshi Sugawara, Kazuo Sakiyama, Shoei Nashimoto, Daisuke Suzuki, and Tomoyuki Nagatsuka. 2019. Oscillator without a combinatorial loop and its threat to FPGA in data centre. *Electronics Letters* 55, 11 (2019), 640–642.
[33] Tomoyuki Takahashi, Takumi Uezono, Michihiro Shintani, Kazuya Masu, and Takashi Sato. 2009. On-die parameter extraction from path-delay measurements. In *IEEE Asian Solid-State Circuits Conference*. 101–104.
[34] Terasic Technologies. 2019. *DE10-Pro User's Manual*. Terasic Technologies.
[35] Shuang Xie and Wai Tung Ng. 2014. Delay-line temperature sensors and VLSI thermal management demonstrated on a 60nm FPGA. In *IEEE International Symposium on Circuits and Systems (ISCAS'14)*. 2571–2574.
[36] Sadegh Yazdanshenas. 2019. *Datacenter-Optimized FPGAs*. Ph.D. Dissertation. Department of Electrical and Computer Engineering, University of Toronto.
[37] Mark Zhao and G. Edward Suh. 2018. FPGA-based remote power side-channel attacks. In *IEEE Symposium on Security and Privacy (S&P'18)*. 229–244.
[38] Kenneth M Zick and John P Hayes. 2010. On-line sensing for healthier FPGA systems. In *ACM/SIGDA International Symposium on Field Programmable Gate Arrays (FPGA)*. 239–248.
[39] Kenneth M. Zick and John P. Hayes. 2012. Low-cost sensing with ring oscillator arrays for healthier reconfigurable systems. *ACM Transactions on Reconfigurable Technology and Systems* 5, 1 (2012), 1–26.
[40] Kenneth M. Zick, Meeta Srivastav, Wei Zhang, and Matthew French. 2013. Sensing nanosecond-scale voltage attacks and natural transients in FPGAs. In *Proceedings of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays (FPGA'13)*. 101–104.
[41] Daniel Ziener, Florian Bauereger, and Jürgen Teich. 2010. Using the power side channel of FPGAs for communication. In *IEEE International Symposium on Field-Programmable Custom Computing Machines (FCCM'10)*. 237–244.

Received August 2020; revised December 2020; accepted January 2021