Measuring Long Wire Leakage with Ring Oscillators in Cloud FPGAs

Ilias Giechaskiel, Kasper B. Rasmussen, Jakub Szefer

29th International Conference on Field-Programmable Logic and Applications (FPL), Barcelona, Spain, Sep. 2019

Measuring Long Wire Leakage with Ring Oscillators in Cloud FPGAs

Ilias Giechaskiel
University of Oxford
Oxford, United Kingdom
ilias.giechaskiel@cs.ox.ac.uk

Kasper Bonne Rasmussen
University of Oxford
Oxford, United Kingdom
kasper.rasmussen@cs.ox.ac.uk

Jakub Szefer
Yale University
New Haven, CT, USA
jakub.szefer@yale.edu

***Abstract*—Recent investigations into FPGA routing resources have shown that long wires in FPGAs leak information about their state in a way which can be measured using ring oscillators. Although in many cases this leakage does not pose a security threat, the possibility of multi-tenant use of FPGA resources invites potential side- and covert-channel attacks exploiting long wire leakage. However, prior work has ignored the realities of cloud environments, which may pose restrictions on the generated bitstreams, such as disallowing combinatorial loops. In this paper, we first demonstrate that the long wire leakage phenomenon persists even in the high-end Virtex UltraScale+ FPGA family. We then evaluate two ring oscillator designs that overcome combinatorial loop restrictions employed by cloud FPGA providers. We experimentally measure the long wire leakage of Virtex UltraScale+ FPGAs in the lab as well as in the Amazon and Huawei FPGA clouds. We show that the two new ring oscillator designs provide almost-identical estimates for the strength of the leakage as traditional ring oscillators, allowing us to measure femtosecond-scale changes in the delays of the long wires. We finally present a set of defense mechanisms that can prevent the new ring oscillator designs from being instantiated in the cloud and the long wire leakage from being exploited.**

***Index Terms*—long wire leakage, cloud FPGAs, ring oscillators, side channels, covert channels, crosstalk**

I. Introduction

With the availability of FPGAs in public cloud infrastructures rapidly rising, and with FPGA designs becoming more sophisticated, several security concerns arise from the prospect of multi-tenant FPGA usage. IP core integration from multiple sources [4], [5], [13], shared FPGA resources between different users [4], [5], [9], [13], [16], [24], and CPU/FPGA hybrid designs [4], [5] enable previously-unexplored attacks, without the need for physical access to the FPGA boards.

Early work in remote FPGA attacks primarily generated valid bitstreams with the potential to crash the FPGA, for instance by causing voltage over- and under-shoots by switching many programmable interconnect points (PIPs) [26], or by causing voltage drops through toggling many ring oscillators (ROs) [6]. ROs have also been used to cause fault attacks and recover cryptographic keys [10], or as sensors for covert channels [20] and side-channel attacks, even for designs with logical isolation enabled [25]. Recent work has also discovered that so-called *long wires* in Xilinx [4], [5] and Intel [13], [16] FPGAs leak information about their state in a way which can be measured fully on-chip using nearby ROs. However, these works ignore restrictions placed by cloud providers such as Amazon Web Services (AWS), which prohibit combinatorial loops from their designs [3]. In this paper, we address this limitation and further the state of the art as follows:

1) We show that the long wire leakage phenomenon persists in the Virtex UltraScale+ FPGA family found in many Xilinx-based cloud providers [23]. As we explain in Section II, UltraScale+ long wires differ significantly from those investigated in prior work [4], [5].
2) We introduce a novel flip-flop-based RO and also evaluate a latch-based RO, both of which overcome combinatorial loop restrictions. These designs are described along with the rest of the experimental setup in Section III.
3) We experiment with 11 boards locally and on 2 cloud providers (Amazon AWS and Huawei Cloud) and determine that the new RO designs provide almost-identical estimates for the long wire leakage as the traditional RO design. As a result, the proposed ROs do not decrease the quality of measurements, and reveal intra- and inter-process variations across the FPGA boards (Section IV).
4) We finally present a set of defense mechanisms which can reduce the impact of the long wire leakage and prevent the RO designs from being instantiated (Section V).

II. Background & Related Work

This section discusses cloud FPGAs (Section II-A), the Xilinx Virtex UltraScale+ architecture used in our experiments (Section II-B), as well as prior work in long wire leakage (Section II-C) and ring oscillator designs (Section II-D).

A. Cloud FPGAs

In recent years, there has been an emergence of public cloud providers offering FPGAs for customer use in their data centers. Xilinx Virtex UltraScale+ boards are available on Amazon AWS, Huawei Cloud and Alibaba Cloud; Kintex UltraScale boards are used on Baidu Cloud and Tencent Cloud; while Nimbix is equipped with Alveo Accelerator Cards [23]. Similarly, Intel Arria 10 boards can be used on Alibaba Cloud and OVH [8]; Stratix V FPGAs are available at the Texas Advanced Computing Center (TACC) [19]; and Stratix 10 FPGAs are available for AI applications on

This work was supported in part by NSF grants 1716541 and 1901901. We would also like to thank Amazon for donating AWS EC2 research credits.

Microsoft Azure [12]. In this work, we ran experiments on Amazon AWS and Huawei Cloud, as they both make their development kits publicly available [2], [7]. Both providers use Virtex UltraScale+ boards, described in Section II-B.

B. Xilinx Virtex UltraScale+ Architecture

The Xilinx Virtex UltraScale+ architecture organizes logic resources into Configurable Logic Blocks (CLBs), each of which contains 8 lookup tables (LUTs), 16 flip-flops, and other resources (multiplexers, carry chains, etc.). Each CLB connects to a switch matrix containing interconnect resources, including (vertical) "long" wires (simply called *longs* or `VLONG`s), spanning 12 CLBs. Unlike previous FPGA generations, `VLONG`s are unidirectional, do not have intermediate tap points, and are organized in channels of 8. This necessitates that adjacent long wires be driven from the same switch matrix, with shorter "local" wires connecting CLB resources to long wires. Moreover, the Virtex UltraScale+ architecture uses a Stacked Silicon Interconnect (SSI) technology, which connects separate 16 nm FPGA dies (or "Super Logic Regions" (SLRs)) using a silicon interposer. The XCVU9P chips used in our experiments contain approximately 1.3 million LUTs distributed over 90 clock regions in 3 SLRs. As we show in Section IV, these SLRs can have significant process variations and should thus be analyzed as separate chips.

C. Long Wire Leakage

Prior work in characterizing long wire leakage in Xilinx [4], [5] and Intel [13], [16] FPGAs has shown that when a long wire carries a 1, the delays of its adjacent long wires are slightly smaller than their delays when the same wire is carrying a 0. These small differences in delay can be estimated using ring oscillators routed to use long wires. The ring oscillator frequency deviates about $0.001-0.06\%$ in response to the long wire state, and can be used to distinguish between a logic 0 and a logic 1. It is worth highlighting that the long wires in Virtex UltraScale+ devices are significantly different from their counterparts in earlier Xilinx FPGA generations. For example, the `VLONG`s in the 28 nm Artix 7 devices used by Giechaskiel et al. [4], [5] span 18 CLBs, are bi-directional, and have an intermediate tap after 9 CLBs, compared to the unidirectional, tap-less `VLONG`s in 16 nm Virtex UltraScale+ devices, which span only 12 CLBs. Adjacent longs in the Artix 7 family are also not driven from the same CLB, unlike the routing channel of 8 that exists in UltraScale+ devices. In this work we show that despite these differences, Virtex UltraScale+ long wires still leak information about their state.

D. Ring Oscillators

Ring oscillators (ROs) are combinatorial loops whose values oscillate, and are typically implemented by chaining an odd number of `NOT` gates in a ring formation. They have been used as transmitters and receivers, for example to characterize FPGA long wire leakage [4], [5], [13], [16] and conduct voltage-based attacks [6], [10], [25]. Some cloud providers, such as Amazon AWS, detect and prohibit them [3].

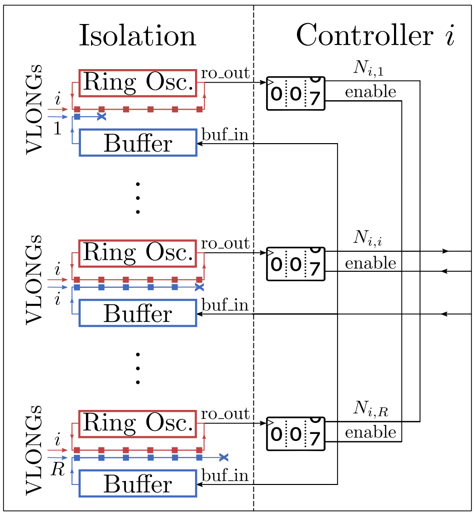


Fig. 1: Controller i has R ring oscillators with i `VLONG`s each. The buffers adjacent to the ROs use 1 to R `VLONG`s.

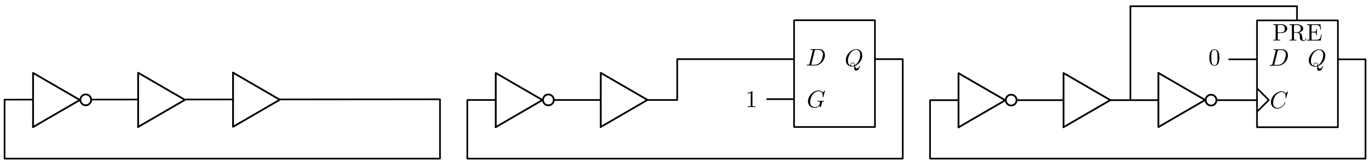
So far, most papers exploring alternative RO designs have not focused on security implications to cloud deployment. ROs replacing one or more stages with an open latch have been used for Physical Unclonable Functions (PUFs) [21], [22] and RO-based temperature sensors [15]. Moreover, flip-flop-based ROs have been proposed to characterize flip-flop delays [14], [17], but these have only been tested in SPICE simulations. Recent proof-of-concept work by Sugawara et al. [18] has also shown that latch-based and flip-flop-based ROs can be instantiated on Amazon AWS, but their performance was not compared to that of traditional ROs. Moreover, all existing flip-flop-based designs differ from our proposal of Section III-B, which is evaluated along with latch-based ROs in Section IV.

III. Experimental Setup

In this section, we first describe the boards and architectural design used in our experiments (Section III-A) and then introduce the three ring oscillator designs (Section III-B). We finally discuss the metric used to estimate the delay difference due to the state of adjacent wires (Section III-C).

A. Architectural Design

For our experiments, we use Xilinx Virtex UltraScale+ boards containing XCVU9P chips. As the compilation and programming time for these boards is significantly higher compared to their Series 7 counterparts, we do not test ring oscillators in isolation nor do we transfer measurements via ChipScope or SignalTap as done in prior work [4], [5], [13], [16]. Instead, we use a hierarchical design of R controllers, each of which contains R identical ring oscillators. All ring oscillators in controller i ($1 \leq i \leq R$) use i longs each. Each controller also contains R buffers. These buffers use between 1 and R longs each, and are adjacent to the long wires of the ring oscillators, as shown in Figure 1. This setup thus contains R^2 combinations of different long wire overlaps. The buffers and ring oscillators of each controller are all



(a) LUT-based RO (b) Latch-based RO (c) Flip-flop-based RO

Fig. 2: The three ring oscillator designs used in the experiments.

placed on separate CLBs spanning two clock regions, and are routed to use adjacent `VLONG`s. No other logic (including the RO frequency counters) is placed in these regions through the `EXCLUDE_PLACEMENT` and `CONTAIN_ROUTING` constraints. Finally, for a given bitstream, all ring oscillators are of the same type (i.e., LUT-based, latch-based, or flip-flop-based), and all logic is placed on a single SLR.

Locally, we use a VCU118 Evaluation Board and communicate the experimental configuration and measurements over the UART. We also simultaneously test 8 FPGAs on an Amazon AWS `f1.16xlarge` instance, and 2 FPGAs on two Huawei Cloud `fp1.2xlarge.11` instances to evaluate inter-device process variations, and transfer data over PCIe in both cases. As the cloud providers reserve some clock regions for their "shell" interface, we reduce the number of controllers and ring oscillators to overcome placement restrictions, while also meeting timing requirements. We do not make any modifications to the boards, nor attempt to improve clock accuracy, for instance via a Mixed-Mode Clock Manager (MMCM) or a Phase-Locked Loop (PLL). We instead use the default clock configuration available in each device tested. These properties are summarized in Table I.

For each setup tested (i.e., for each RO type, on each SLR), we run three tests of $10,000$ measurements each, for a total of $30,000$ data points from each RO per testing configuration. All results are reported at the 99% confidence level. To estimate the frequency of each RO, we count the number of its signal transitions during a 2^n clock-cycle period. Since Giechaskiel et al. [4], [5] have shown that the pattern of transmission does not affect the strength of the wire leakage phenomenon, we toggle the input to the signal buffers after each measurement period. We use $n = 23$ below (28-67 ms, depending on clock speed), but we have verified that the phenomenon persists with $n \in \{13, 15, 17, 19, 21, 25\}$ (but is noisier as n decreases).

B. Ring Oscillator Designs

In this section, we introduce the three types of ring oscillators that are used in our experiments. The first ring oscillator, shown in Figure 2a, is composed of one inverter and two buffer stages, similar to prior work [4], [5], [13], [16]. These stages are implemented using "1-Bit Look-Up Table with General Output" `LUT1` primitives with the `INIT` parameter. The second ring oscillator design, shown in Figure 2b, replaces one of the two buffers with a latch. We used the "Transparent Latch" `LD` primitive, with the gate input `G` tied to 1, and confirmed we could also use the "Transparent Latch with Clock Enable and Asynchronous Clear" `LDCE` primitive, setting gate enable `GE` to 1, as suggested by Sugawara et al. [18].

Property	**Local**	**AWS F1**	**Huawei FP1**
# of Boards Tested	1	8	2
Board	VCU118	Proprietary	Proprietary
XCVU9P Chip	flga2104-2-e	flgb2104-2-i	flgb2104-2-i
Shell Clock Regions	None	X4Y0:X5Y9	X3Y4:X5Y9
Prohibits Comb. Loops	No	Yes	No
Clock Frequency (MHz)	300	125	200
Communication	UART	PCIe	PCIe
Vivado Version	2017.4	2018.2	2017.2
# of RO Combinations	$81 = 9^2$	$64 = 8^2$	$36 = 6^2$

TABLE I: Properties of the boards used in the experiments.

Our flip-flop-based design, shown in Figure 2c, uses the "D Flip-Flop with Clock Enable and Asynchronous Preset" `FDPE` primitive, with its `D` input tied to 0. The output of the RO buffer stage is connected to the preset input `PRE`, while its inverted value is connected to the clock port `C`. When `PRE` is high, the output `Q` is also high. When it is falls low, the flip-flop clock transitions from low to high, thereby mirroring `D` to output a 0, and acting as a buffer stage. This RO design differs both from designs which depend on delay stages between clock and clear inputs [14], [17], and from the flip-flop-based design by Sugawara et al. [18], which uses the "D Flip-Flop with Clock Enable and Asynchronous Clear" `FDCE` primitive, and delays between the output and the clock.

C. Measurement Metric

Giechaskiel et al. introduced a "relative count difference" metric [4], [5] to estimate long wire leakage, defined as:

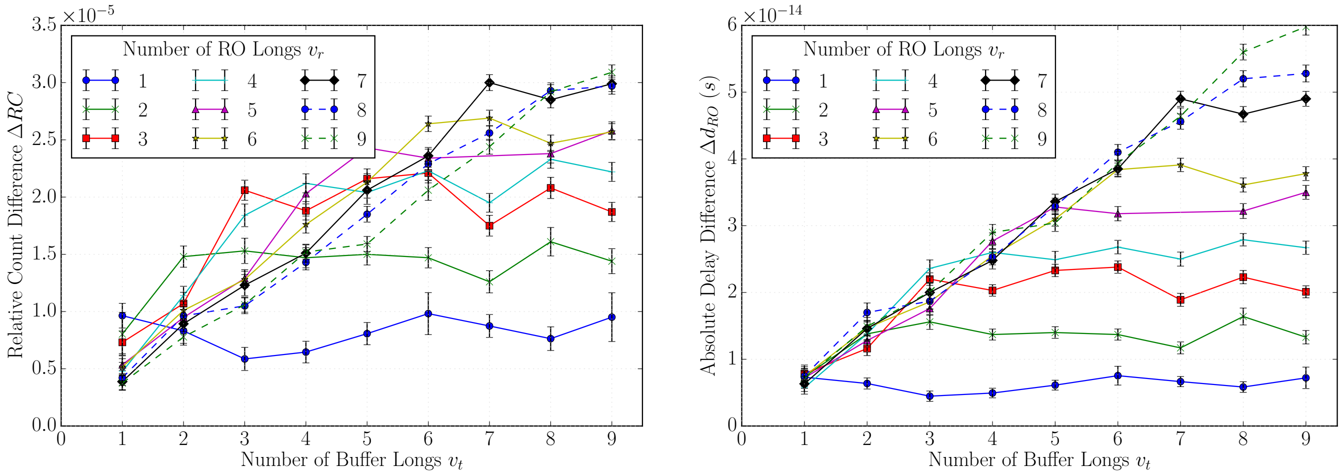
$$\Delta RC = \frac{C^1_{RO} - C^0_{RO}}{C^1_{RO}} \tag{1}$$

where C^1_{RO} and C^0_{RO} denote the ring oscillator counts when the adjacent long wire is carrying logic 1 and 0 respectively. Although this metric is independent of the measurement period and the clock frequency, it is sensitive to the absolute ring oscillator frequency, which is affected by nearby logic [11].

To overcome this drawback we use a metric which is independent of the RO frequency, and can be used to estimate the absolute delay difference of the long wires due to nearby state. The frequency of the ring oscillator f_{RO} is given by:

$$f_{RO} = \frac{1}{2d_{RO}} = f_{CLK} \cdot \frac{C_{RO}}{C_{CLK}} \tag{2}$$

where d_{RO} is the delay of the ring oscillator, f_{CLK} is the frequency of the clock, and C_{RO}, C_{CLK} are the counts



(a) Relative Count Difference

(b) Absolute Delay Difference

Fig. 3: Relative Count (Fig. 3a) and Absolute Delay (Fig. 3b) differences on the VCU118 board for different RO and buffer lengths (number of VLONGs used). The measurements were taken on SLR 0, using LUT-based ring oscillators.

driven by the RO and the clock during a measurement period respectively. As a result, the absolute difference in delay Δd_{RO} can be calculated as follows:[1]

$$\Delta d_{RO} = \frac{1}{2}\left(\frac{1}{f^0_{RO}} - \frac{1}{f^1_{RO}}\right) = \frac{f^1_{RO} - f^0_{RO}}{2 f^0_{RO} f^1_{RO}} \tag{3}$$

Assume the ring oscillator and buffer use n adjacent VLONGs each, and that the overlap of adjacent VLONGs is fixed. Then the delay of a signal travelling through the ring oscillator is $d_{RO} = n \cdot d_L + d_c$, where d_L is the delay of one long wire, and d_c is an RO-specific constant that accounts for local routing, logic delays, and process variations. As a result:

$$\Delta d_{RO} = d^0_{RO} - d^1_{RO} = n\left(d^0_L - d^1_L\right) = n \Delta d_L \tag{4}$$

As Δd_L is in the order of femtoseconds, using $n > 1$ VLONGs allows us to better estimate the per-long wire delay:

$$\Delta d_L = \frac{\Delta d_{RO}}{n} = \frac{1}{n} \cdot \frac{C_{CLK}}{2 f_{CLK}} \cdot \frac{C^1_{RO} - C^0_{RO}}{C^0_{RO} C^1_{RO}} \tag{5}$$

The above formulas apply to all three RO designs, since the leakage affects the delay of long wires, but not the logic of the ring oscillators themselves. As a result, design differences can be incorporated in the RO-specific constant d_c, which does not influence Δd_{RO} and Δd_L as shown in Equations (4) and (5). Section IV shows this experimentally by comparing the per-long wire delay difference across 11 different Virtex UltraScale+ boards and all three RO designs.

IV. Evaluation

In this section, we first show that the resource usage of the communicating channel circuitry is minimal (Section IV-A), and demonstrate that the new delay-based metric is superior to the old relative count difference metric (Section IV-B). We then investigate the long wire leakage across 11 boards and 33 SLRs (Section IV-C), and finally compare the long wire leakage estimates using the three RO designs (Section IV-D).

A. *Resource Utilization*

The buffers in our setup always use 2 LUTs in 2 CLBs. Ring Oscillators also use 2 separate CLBs, and a total of 3 LUTs (LUT-based RO), 2 LUTs and a register acting as a latch (latch-based RO), or 3 LUTs and a register acting as a flip-flop (flip-flop-based RO). Thus, the combined usage of 81 ring oscillators and buffers amounts to at most $405/1,182,240 = 0.034\%$ of LUT resources and $81/2,364,480 = 0.0034\%$ of register resources in the XCVU9P chips. The whole design (including counters, the UART communication interface, etc.) uses less than 0.85% of all resources. Due to differences in local routing and the delays of the logic elements, the LUT-based RO is the fastest, while the flip-flop-based RO is the slowest. Using similar equations to those of Section III-C, we can determine that the delay differences between the 3 setups are in the order of 100s of picoseconds, an estimate which is within the range of the speed models reported by Vivado.

B. *Metric Comparison*

Giechaskiel et al. observed that for a given number of VLONGs v_r used by the ring oscillator, ΔRC increases linearly with the adjacent number of VLONGs v_t used by the buffer as long as $v_t \leq v_r$ and then remains constant [4], [5]. Moreover, for a fixed v_t, among ring oscillators with $v_r \geq v_t$, a smaller v_r results in a larger ΔRC. The opposite is true among ring oscillators with $v_r \leq v_t$: a larger v_r results in a larger ΔRC.

The ΔRC metric emphasizes relative count differences, and is thus sensitive to even small changes in the frequency of the ring oscillator, which is influenced by Process, Voltage, and

[1] A similar formula was derived by Provelengios et al. [13], with an alternative approach focusing on signal transitions. Our earlier derivation has been verified through personal communication with an author of [13].

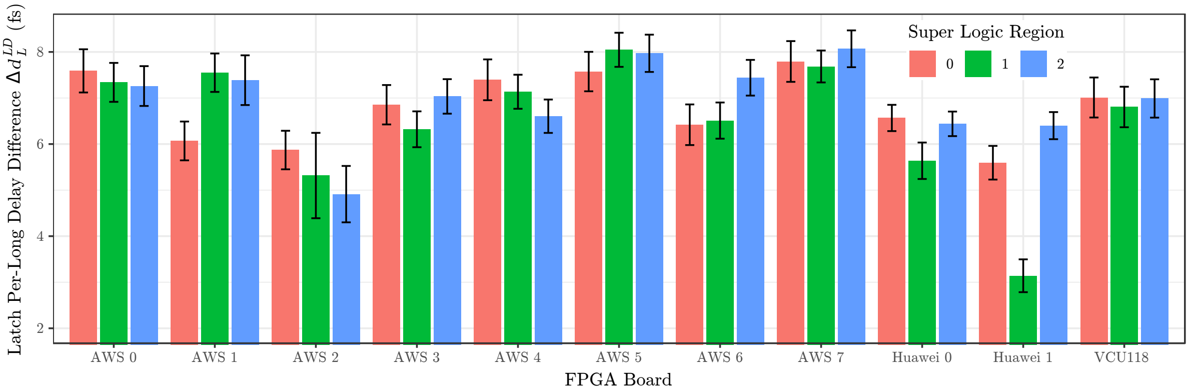


Fig. 4: Per-long delay difference Δd_L^{LD} estimates using latch-based ROs for all SLRs and boards tested.

Temperature (PVT) variations. This can be seen in Figure 3a, where the distinction between different lengths of longs is sometimes unclear. For example, the leakage of $(v_t, v_r) = (9, 5)$ is lower than the leakage of $(v_t, v_r) = (9, 6)$, but Figure 3a suggests the opposite (the pairs $(9, 7)$ and $(9, 8)$ are similarly inverted). Meanwhile, the Δd_{RO} metric can clearly identify the number of longs used, as shown in Figure 3b. Since our metric measures the absolute delay difference due to adjacent state, it is proportional to the size of the VLONG overlap between the buffer and the RO, and is also independent of the clock and RO frequency.

C. Inter- and Intra-Device Variations

We tested device variations on 11 FPGAs (8 on AWS, 2 on Huawei Cloud, and 1 in the lab), and used latch-based ROs to overcome restrictions placed by Amazon. For each bitstream, we find the absolute delay difference Δd_{RO} for each ring oscillator and buffer combination, and estimate Δd_L using Equation (5). We then plot the average over all combinations with 99% confidence intervals in Figure 4.

Figure 4 allows us to draw three main conclusions. First, in all boards and all individual SLRs, VLONGs leak information about their state through a change in their delay, which is in the order of a few femtoseconds. Second, for *most* boards the strength of the leakage is approximately the same for all SLRs, suggesting that SLRs in the same chip might be manufactured together. However, the inter-SLR variation can sometimes be as large as the inter-chip variation between physically distinct boards (e.g., the AWS 1 and Huawei 1 boards). As a result, different SLRs should be treated as distinct chips with respect to process variations. Finally, within a board there is no consistent pattern in how long wire leakage varies between SLRs, despite the heavy logic placed by cloud providers in nearby clock regions (SLRs 0 and 1). This suggests that the strength of the leakage is not influenced by nearby logic, allowing an adversary to measure it in the presence of large circuits not under his/her control.

D. Ring Oscillator Comparison

This section shows that all three ring oscillators give approximately the same estimate for the difference in the delay of the long wires (averaged over all RO and buffer combinations). As shown in Figure 5, there is no consistent pattern for how the estimates using different RO types vary. Even though, on average, latch-based ROs result in larger Δd_L estimates compared to the other ROs, the estimates are very close in absolute terms: for example, for the AWS 0 board (SLR 0), the 99% confidence estimate is 7.59 ± 0.47fs with a latch RO, and 7.51 ± 0.46fs when using a register RO. Hence, all 3 ROs can distinguish nearby state, and estimate femtosecond-scale differences in the delay of VLONGs to within 10% of each other, despite environmental noise and process variations.

V. Defense Mechanisms

This section presents a set of countermeasures to protect against malicious designs exploiting long wire leakage, or abusing the new ring oscillator designs in the cloud.

Routing Restrictions: As the long wire leakage can only be determined though adjacent VLONGs, multi-tenant designs need to enforce physical isolation between users, and explicitly protect sensitive signals. More generally, cloud providers may need to disallow custom user placement and routing, and instead randomize their location, similar to address randomization protections for software binaries.

Design Rule Checks (DRCs): Checks to prevent latches from being used, and to ensure that clocks driving registers are derived from cloud-provided clocks can raise the bar for attackers, until alternative malicious designs emerge. We have also determined that the following DRC warnings appear in our designs locally and on Huawei cloud (AWS suppresses such warnings), and should thus be promoted to errors:

- LUTLP-2 appears when combinatorial loops are allowed, but currently only results in an error on Amazon AWS.
- PLHOLDVIO-2 appears when there are “Non-Optimal connections which could lead to hold violations”, such as when a LUT is driving a clock pin.
- PDRC-153 appears when a combinatorial pin sources a “gated clock net” directly instead of through the CE pin.

Runtime Protections: In order to prevent damage to the physical hardware, runtime monitors for temperature and power usage are necessary if not all self-clocked designs

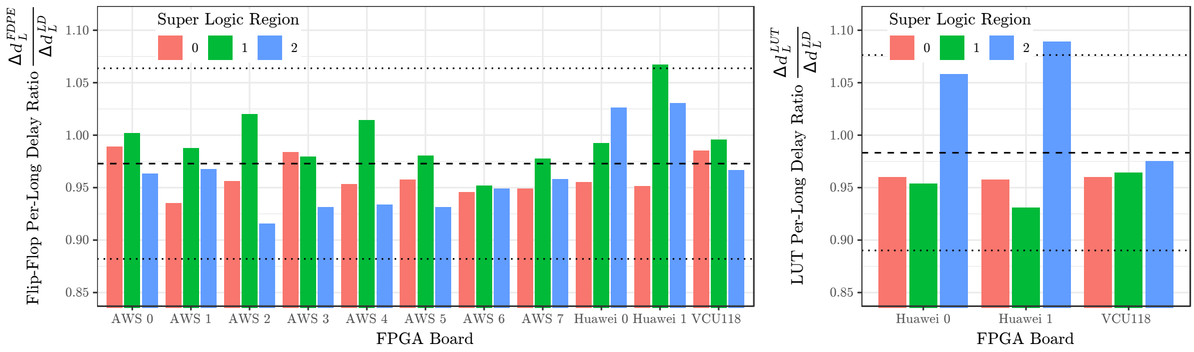


Fig. 5: Ratio of per-long wire delay difference estimates using flip-flop-based ROs Δd_L^{FDPE} (left) and LUT-based ROs Δd_L^{LUT} (right) to estimates using latch-based ROs Δd_L^{LD}. The three RO designs estimate Δd_L within 10% of each other.

can be prevented. Although Amazon gates clocks should a maximum power threshold be reached [1], ROs could still cause damage to the device, necessitating more aggressive protection mechanisms such as clearing the FPGA.

VI. Conclusion

In this paper we showed that VLONGs in Xilinx Virtex UltraScale+ FPGAs leak information about their state to nearby logic. We successfully demonstrated this leakage on 11 boards across two cloud providers (Amazon AWS and Huawei Cloud) through two new ring oscillator designs which overcome combinatorial loop restrictions, and are currently undetected by cloud providers. We used a metric which allows us to measure femtosecond-scale changes in the delay of the long wires, and showed that the two designs provide almost-identical estimates for this delay difference as traditional ring oscillators. We finally proposed some countermeasures in response to remote FPGA attacks without physical access, paving the way towards secure multi-tenant cloud FPGAs.

References

[1] AWS GitHub, “AFI power,” https://github.com/aws/aws-fpga/blob/master/hdk/docs/afi_power.md, Accessed: 2019-05-20.

[2] ——, “AWS EC2 FPGA hardware and software development kit,” https://github.com/aws/aws-fpga, Accessed: 2019-05-20.

[3] ——, “AWS EC2 FPGA HDK+SDK errata,” https://github.com/aws/aws-fpga/blob/master/ERRATA.md, Accessed: 2019-05-20.

[4] I. Giechaskiel, K. Eguro, and K. B. Rasmussen, “Leakier wires: Exploiting FPGA long wires for covert- and side-channel attacks,” *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*, 2019.

[5] I. Giechaskiel, K. B. Rasmussen, and K. Eguro, “Leaky wires: Information leakage and covert communication between FPGA long wires,” in *Asia Conference on Computer and Communications Security (ASIACCS)*, 2018.

[6] D. R. E. Gnad, F. Oboril, and M. B. Tahoori, “Voltage drop-based fault attacks on FPGAs using valid bitstreams,” in *Field Programmable Logic and Applications (FPL)*, 2017.

[7] Huawei Cloud GitHub, “Huawei Cloud FPGA development kit,” https://github.com/huaweicloud/huaweicloud-fpga, Accessed: 2019-05-20.

[8] Intel Corporation, “Accelerating cloud applications with Intel Xeon scalable processors and Intel FPGAs,” https://www.intel.com/content/dam/www/public/us/en/documents/brief/fpga-as-a-service-opportunities.pdf, Accessed: 2019-05-20.

[9] A. Khawaja, J. Landgraf, R. Prakash, M. Wei, E. Schkufza, and C. J. Rossbach, “Sharing, protection, and compatibility for reconfigurable fabric with AMORPHOS,” in *Operating Systems Design and Implementation (OSDI)*, 2018.

[10] J. Krautter, D. R. E. Gnad, and M. B. Tahoori, “FPGAhammer: Remote voltage fault attacks on shared FPGAs, suitable for DFA on AES,” *Transactions on Cryptographic Hardware and Embedded Systems (TCHES)*, vol. 2018, no. 3, pp. 44–68, Sep 2018.

[11] D. Merli, F. Stumpf, and C. Eckert, “Improving the quality of ring oscillator PUFs on FPGAs,” in *Workshop on Embedded Systems Security (WESS)*, 2010.

[12] Microsoft Research Blog, “Microsoft unveils Project Brainwave for real-time AI,” https://www.microsoft.com/en-us/research/blog/microsoft-unveils-project-brainwave/, Accessed: 2019-05-20.

[13] G. Provelengios, C. Ramesh, S. B. Patil, K. Eguro, R. Tessier, and D. Holcomb, “Characterization of long wire data leakage in deep submicron FPGAs,” in *Field-Programmable Gate Arrays (FPGA)*, 2019.

[14] T. Ragheb and A. Marshall, “Calibration of propagation delay of flip-flops,” in *SOC Conference (SOCC)*, 2012.

[15] N. Rahmanikia, A. Amiri, H. Noori, and F. Mehdipour, “Performance evaluation metrics for ring-oscillator-based temperature sensors on FPGAs: A quality factor,” *Integration: The VLSI Journal*, vol. 57, pp. 81–100, Mar 2017.

[16] C. Ramesh, S. B. Patil, S. N. Dhanuskodi, G. Provelengios, S. Pillement, D. Holcomb, and R. Tessier, “FPGA side channel attacks without physical access,” in *Field-Programmable Custom Computing Machines (FCCM)*, 2018.

[17] R. P. Ribas, A. I. Reis, and A. Ivanov, “Performance and functional test of flip-flops using ring oscillator structure,” in *International Design and Test Workshop (IDT)*, 2011.

[18] T. Sugawara, K. Sakiyama, S. Nashimoto, D. Suzuki, and T. Nagatsuka, “Oscillator without a combinatorial loop and its threat to FPGA in data centre,” *Electronics Letters*, Apr 2019.

[19] Texas Advanced Computing Center, “TACC to launch new Catapult system to researchers worldwide,” https://www.tacc.utexas.edu/-/tacc-to-launch-new-catapult-system-to-researchers-worldwide, Accessed: 2019-05-20.

[20] S. Tian and J. Szefer, “Temporal thermal covert channels in cloud FPGAs,” in *Field-Programmable Gate Arrays (FPGA)*, 2019.

[21] A. Wild, G. T. Becker, and T. Güneysu, “On the problems of realizing reliable and efficient ring oscillator PUFs on FPGAs,” in *Hardware Oriented Security and Trust (HOST)*, 2016.

[22] ——, “A fair and comprehensive large-scale analysis of oscillation-based PUFs for FPGAs,” in *Field Programmable Logic and Applications (FPL)*, 2017.

[23] Xilinx, Inc., “Reconfigurable acceleration in the cloud,” https://www.xilinx.com/products/design-tools/cloud-based-acceleration.html, Accessed: 2019-05-20.

[24] S. Yazdanshenas and V. Betz, “Interconnect solutions for virtualized field-programmable gate arrays,” *IEEE Access*, vol. 6, pp. 10 497–10 507, Feb 2018.

[25] M. Zhao and G. E. Suh, “FPGA-based remote power side-channel attacks,” in *IEEE Symposium on Security and Privacy (S&P)*, 2018.

[26] K. M. Zick, M. Srivastav, W. Zhang, and M. French, “Sensing nanosecond-scale voltage attacks and natural transients in FPGAs,” in *Field-Programmable Gate Arrays (FPGA)*, 2013.