Characterization of `IOBUF`-based Ring Oscillators

Julia Burgiel
Yale University
New Haven, CT, USA
julia.burgiel@yale.edu

Daniel Esguerra
Yale University
New Haven, CT, USA
daniel.esguerra@yale.edu

Ilias Giechaskiel
Independent Researcher
London, United Kingdom
ilias@giechaskiel.com

Shanquan Tian
Yale University
New Haven, CT, USA
shanquan.tian@yale.edu

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

***Abstract*—Ring Oscillators (ROs) are fundamental primitives that are used as building blocks in many other types of circuits. This paper presents an in-depth characterization of ring oscillators which leverage the `IOBUF` primitive found in modern Xilinx FPGAs. This work first analyzes the impact of the *drive strength* and *slew rate* attributes of the `IOBUF`s on the ROs, and also characterizes the impacts of external temperature, internal voltage, and external voltage fluctuations on the frequency of the proposed ROs. This work further demonstrates that `IOBUF`-based ROs can detect whether electrical connections to the `IOBUF` pins have changed, including whether the DRAM module has been physically removed. Finally, the proposed ROs can be realized on cloud FPGAs, bypassing the restrictions that some cloud providers impose on combinatorial loops, and thus presenting a new security threat to remote FPGAs.**

I. Introduction

Ring Oscillators (ROs) are a combinatorial loop circuit that contains an odd number of inverter gates, and any number of buffer gates. The inverter and buffer gates are typically implemented in lookup-tables (LUTs), but they can also include latches or flip-flops [1], [7]. ROs can be used for many types of defensive circuits such as True Random Number Generators (TRNGs) and Physical Unclonable Functions (PUFs) [9]. At the same time, ROs can also be used for covert- or side-channel attacks as both receivers (since their frequencies can change in response to temperature or voltage fluctuations) and transmitters (by causing such fluctuations) [5].

The primary contribution of this work is using the `IOBUF` primitive to create an RO design that can be tuned and is not detected by existing Design Rule Checks (DRCs) used by cloud providers such as Amazon Web Services (AWS). This work further discovers that `IOBUF`-based ROs can be used to detect whether electrical connections to the `IOBUF` pins have changed, and demonstrates this by detecting whether the attached DRAM module has been removed. We open source the `IOBUF`-based RO design at https://caslab.csl.yale.edu/code/iobuf-ro/.

II. Background

The ROs presented in this paper are based on the `IOBUF` primitive found in UltraScale+ and earlier Xilinx FPGA families, which are popular both for in-house deployments and in public cloud infrastructures.

A. *Ring Oscillators*

Ring oscillators are typically realized using a combinatorial loop consisting of an odd number of inverter gates. ROs can also include any number of buffer stages, which are usually implemented in lookup tables (LUTs), but can alternatively be realized using latches or flip-flops [1], or other primitives such as multiplexers, carry chains, and Digital Signal Processing (DSP) blocks [4]. Xilinx documentation also mentions the use of `IODELAY` primitives to manipulate the frequencies of ROs [11] as an additional building block that can be applied to different types of ring oscillators. Our ROs instead use the `IOBUF` primitive [8], can be tuned post-routing without RTL changes, and can be deployed in cloud FPGAs, bypassing Design Rule Checks, and hiding their functionality from existing defenses, e.g., [4].

B. `IOBUF` *Primitives*

An `IOBUF` is a Xilinx primitive which connects internal logic to an external bidirectional pin. It is made up of a buffer input, a buffer output, a bidirectional (in/out) port connection to an external device pad, and tri-state control (on/off). In our RO design, we use `IOBUF`s in the low-impedance (non-tri-stated) mode. `IOBUF`s have three additional attributes, which can be set through parameters during their instantiation, or altered in a later stage of the design process through Tcl commands: the I/O standard (`IOSTANDARD`), the drive strength (`DRIVE`), and the slew rate (`SLEW`). Although settable in software, the I/O standard in practice depends on the voltage level and signaling mode of the physical pin and the FPGA I/O bank in which it belongs [12]. On the other hand, the drive strength (in mA) and the slew rate (corresponding to `SLOW` or `FAST` output rise and fall times) can be meaningfully changed, even post-routing, and, as we show, directly impact the frequency of the resulting ring oscillator.

C. `IOBUF`*-based Ring Oscillator Design*

Our `IOBUF`-based RO is shown in Figure 1. It is similar to the traditional LUT-based ROs, but instead connects the output of one of the inverter gates to the input buffer of the `IOBUF` primitive. The output buffer of the primitive then feeds the `AND` gate, while the `T` signal is set low, making it act as a regular (rather than a tri-state) buffer.[1] Because the input and

[1] In Xilinx terminology, the "output" of an `IOBUF` is the external signal coming into the FPGA, while its "input" is the internal signal going out of the FPGA and driving the external I/O pin.

This work was supported in part by NSF grant 1901901.

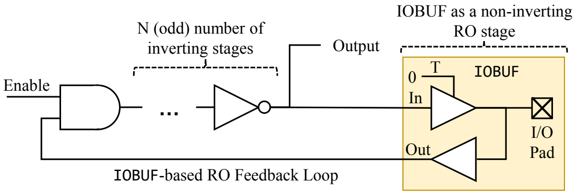

Fig. 1. Diagram of the proposed IOBUF-based RO design.

the output of the IOBUF are connected together inside the IOBUF primitive, they behave as a buffer stage that mirrors its input to its output, while also driving the I/O pad pin itself. By inserting the IOBUF between one of the inverting stages, the combinatorial feedback loop detected by the FPGA tools is eliminated, allowing the IOBUF-based RO to be instantiated even on commercial cloud FPGA deployments. Moreover, by changing the drive strength and slew rate properties of the IOBUF primitive, it is possible to change the frequency of the RO, without changing its structure or routing. Finally, as we show, the connection to the I/O pad makes the IOBUF-based ROs able to detect changes in electrical connections from adjacent IOBUF pins.

III. Experimental Setup

Our tests are conducted on two 7 Series development boards from Xilinx: the KC705 board with a Kintex 7 XC7K325T FPGA, and the AC701 board with an Artix 7 XC7A200T chip. We use IOBUF primitives corresponding to I/O pins from an FPGA Mezzanine Card (FMC) as well as IOBUF primitives corresponding to I/O pins from Dynamic Random Access Memory (DRAM). In addition, we test the IOBUF-based ROs on UltraScale+ FPGAs in AWS, by using DRAM pins, which are the only ones directly accessible to users.

In our tests we instantiate multiple IOBUF-based ROs in an FPGA, and control and measure them independently one-by-one, along with a baseline LUT-based RO that is instantiated in the FPGA as a reference. All ROs use 7 inverters unless otherwise specified, but we have confirmed that results remain similar for other numbers of stages. For all experiments, a Xilinx Design Constraints (XDC) file is used to specify the placement location of the inverters for both the baseline reference LUT-based RO and the IOBUF-based ROs. We further leverage the FPGA floorplan and FPGA board schematics to find which IOBUFs map to FMC pins, DRAM pins, as well as to adjacent IOBUFs on the FPGA.

For experiments on the RO sensitivity to external thermal changes, the FPGA board is placed in a TestEquity 115A Temperature Chamber, which is remotely controlled by a Python script that changes the temperature from 15 °C to 45 °C in steps of 5 °C. To test the RO sensitivity to external voltage fluctuations, the FPGA board is attached to a Keithley 2231A-30-3 power supply, which is remotely controlled by a Python script that varies the input FPGA board voltages from 11.5 V to 12.5 V in increments of 0.1 V. Finally, to test the RO sensitivity to internal voltage fluctuations, a set of five RO-based stressors is instantiated inside the FPGA, alongside the baseline LUT-based RO and the IOBUF-based ROs. Each stressor group consists of 2,000 ROs with two buffer stages each, and can be turned on or off to manipulate the internal FPGA voltage, since the more stressors that are turned on, the higher the internal voltage drops are [2], [6].

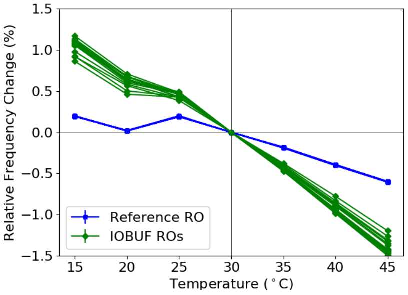

Fig. 2. External temperature sensitivity of 16 different IOBUF-based ROs (green diamonds) and 16 (identical) measurements of the reference RO (blue squares) on the KC705 board. All ROs contain 7 inverters, and their respective frequencies are approximately 83-85 MHz to 107-110 MHz for the IOBUF-based ROs (depending on pin and routing) and 174-176 MHz for the reference LUT-based RO.

IV. Characterization of IOBUF-Based ROs

In this section we characterize the impact of the *drive strength* and *slew rate* attributes of IOBUF primitives on the frequencies of the proposed ROs, and also evaluate the impact of external temperature, internal voltage, and external voltage changes on their behavior.

A. External Temperature Sensitivity

Figure 2 shows how external temperature changes affect the IOBUF-based ROs and the baseline LUT-based RO, as tested in the thermal chamber. The absolute RO frequencies may vary slightly in each run due to noise, thus we repeat the experiments and calculate the averages. The relative change in the frequencies of the IOBUF-based ROs is significantly larger compared to the reference RO, with a slope of approximately 0.06% per degree centigrade, compared to about 0.02% for the reference RO.[2] The proposed ROs are thus more sensitive to external temperature variations, and could potentially serve as better thermal sensors compared to existing ROs used for, e.g., thermal temporal attacks [10].

B. Internal Voltage Sensitivity

Figure 3 shows the relative frequency change in the IOBUF-based ROs as 1, 3, or all 5 stressors are turned on, inducing progressively bigger internal FPGA voltage drops. Here, the two types of ROs exhibit similar trends, although IOBUF-based ROs seem to be impacted less by the stressors (and voltage changes). As a result, the proposed ROs could be possibly used to make more stable RO-based Physical Uncloneable Functions (PUFs) [9].

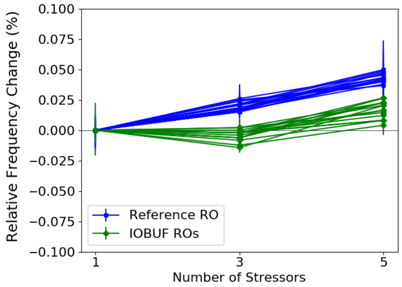

Fig. 3. Comparison of RO frequencies with different number of stressors enabled, for both IOBUF-based ROs (green diamonds) and the reference RO (blue squares). Each stressor has 2,000 ROs.

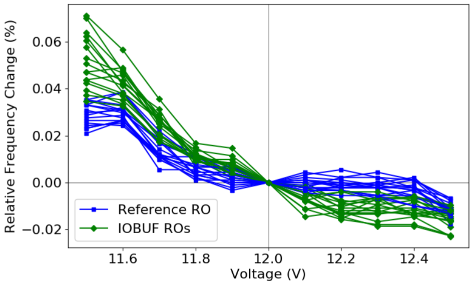

Fig. 4. External voltage sensitivity of 16 different IOBUF-based ROs (green diamonds) and 16 measurements of the reference RO (blue squares) for the same setup as in Figure 2.

C. External Voltage Sensitivity

Figure 4 shows how the FPGA board input voltage (prior to the regulator) affects the IOBUF-based ROs and the reference RO. Here, the two types of ROs exhibit similar sensitivity to external voltage changes. As a result, the proposed ROs could be used as drop-in replacements for traditional RO sensors used for voltage sensing, e.g., in cross-FPGA voltage attacks [2], while bypassing Design Rule Checks.

D. Drive Strength and Slew Rate

Figure 5 shows the impact of different drive strengths and slew rates on the IOBUF-based ROs: in the given setup, it is possible to adjust an RO's frequency by over 30 MHz through different drive strength and slew rate parameters. Keeping the drive strength the same but changing the slew rates from SLOW to FAST generally increases the RO's frequency by approximately 3-10 MHz. The frequency change when increasing the drive strength is as high as 20-33 MHz, and thus has a higher impact compared to the slew rate. Note that these parameters are modified in a post-routing Design Check Point (DCP) file using Tcl commands, with the synthesis, placement, and routing of the entire design remaining identical. This fact allows us to adjust the delay of the IOBUF-based ROs, without the need for dedicated IODELAY elements.

E. Repeatability: Slice Location & Boards

We repeated experiments in several slice locations within the IOBUF's clock region (top/bottom right/left and adjacent to the IOBUFs), with similar results: although the concrete frequencies of the ROs changed (the further the LUTs were placed from the IOBUFs, ths lower the RO frequency), the patterns remained similar.

Figure 6 further shows the results of our experiments on different temperature, voltage, slew rate and drive strength combinations across two KC705 boards and an AC701 board. In the interest of space, we only show a limited number of such combinations. The different boards generally exhibit very similar behavior, highlighting that the IOBUF-based ROs have reliable properties across several setups.

V. Detecting Changes in Electrical Connections

We further investigated whether the IOBUF-based ROs can be abused for new types of vulnerabilities. To that end, we used the floorplan information from Xilinx tools to find triplets of physically adjacent IOBUFs. We chose the middle IOBUF to be used for an IOBUF-based RO. The remaining two adjacent IOBUFs (left and right) were then driven with a fixed signal representing either high or low digital values. We focused on triplets using IOBUFs connected to DRAM address pins: since address pins are outputs from the FPGA and inputs to the DRAM, it is safe to drive these IOBUFs without damaging the DRAM.

For each test, we took 512 measurements from the IOBUF-based RO, while adjacent IOBUFs cycled through the four possible fixed values of 00, 01, 10, and 11 (i.e., in the first measurement both IOBUFs carry value 0, while in the second measurement, the right IOBUF is updated to value 1, etc.) A reference RO was placed elsewhere on the device to monitor for noise and any external voltage or temperature fluctuations.

Figure 7 shows an example set of measurements from an IOBUF triplet on the KC705 board. In this example, the RO frequency is the highest when both adjacent IOBUFs carry the value 1, the lowest when they both carry 0, and in between otherwise. Although this effect may be due to other aspects of the routing and placement logic, e.g., long wires [1] or multiplexers [3], we discovered that for a fixed bitstream, the crosstalk characteristics change when there is a physical or electrical change to the pins and wires.

Specifically, we measured the frequency behavior of 5 randomly selected IOBUF-based RO triplets on the board before and after the DRAM chip removal. When the DRAM module was physically connected to the board, the IOBUF-based ROs were consistently the fastest when the two adjacent IOBUFs carried the value 1. However, when the DRAM was removed (but the bitstream remained the same), this was no longer the case: the 11 pattern did not always lead to the fastest RO frequency. To put it differently, for a random set of 5 IOBUF-based RO triplets, 100% of the RO triplets with adjacent values of 11 were the fastest when the DRAM was present. However, when the DRAM was removed, this percentage dropped to the 40% to 60% range. This unexpected ability to infer when the

[2] Plots of absolute frequency changes are similar: IOBUF-based ROs are more sensitive to temperature fluctuations in both relative and absolute terms.

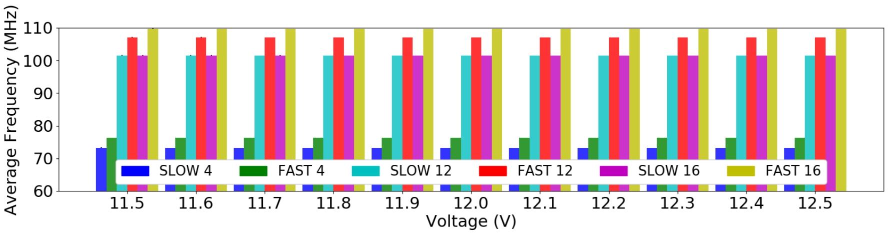

Fig. 5. Frequencies of an `IOBUF`-based RO with different drive strength and slew rates. The routing of the entire design remains identical in all tests, with changes made directly to the post-routing Design Check Point (DCP) file. The data is shown for a sample RO corresponding to FMC pin `FMC_LPC_CLK0_M2C_N` of a KC705 board, but is representative of other pins and boards tested.

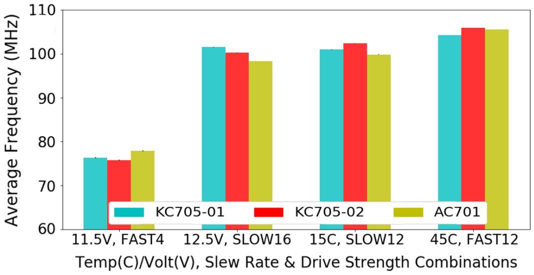

Fig. 6. Comparison of `IOBUF`-based RO frequencies across two KC705 boards and an AC701 board for FMC pins `FMC_LPC_CLK0_M2C_N` and `FMC1_HPC_CLK0_M2C_N` for the two respective boards. Similar results were obtained across different setups and pins tested.

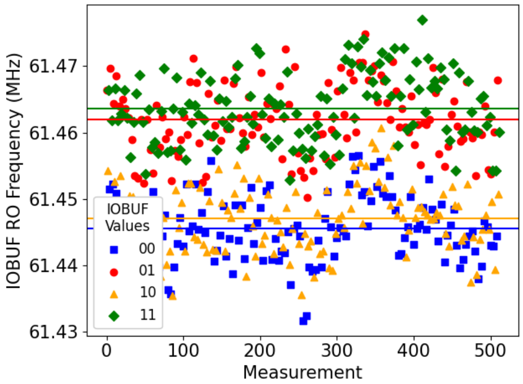

Fig. 7. Measured `IOBUF`-based RO frequencies on a KC705 FPGA when adjacent `IOBUF`s cycle through the values 00, 01, 10, and 11.

DRAM has been removed from the FPGA board hints at a new means of detecting whether an attacker has physically tampered with the memory of an FPGA.

VI. `IOBUF`-Based ROs on Cloud FPGAs

In addition to the above evaluation on AC701 and KC705 boards, we have also confirmed that our `IOBUF`-based ROs can be successfully instantiated on Amazon's F1 instances, which use Virtex UltraScale+ boards. Our experiments were conducted on `f1.2xlarge` instances in the `us-east-1e` region, using AWS FPGA Developer AMI version `v1.7.0`. By using pins from the attached DRAM modules, we were able to realize `IOBUF`-based ROs on F1 instances, despite the current Design Rule Checks (DRCs) which aim to detect and prevent traditional LUT-based ROs.

VII. Conclusion

This paper presented an in-depth characterization of `IOBUF`-based ROs, including measuring the impact of the *drive strength* and *slew rate* attributes, temperature changes, as well as external and internal voltage changes. This work further discovered that `IOBUF`-based ROs can be used to detect whether electrical connections to the `IOBUF` pins have changed, and can be deployed on public cloud infrastructures.

References

[1] I. Giechaskiel, K. B. Rasmussen, and J. Szefer, "Measuring long wire leakage with ring oscillators in cloud FPGAs," in *International Conference on Field Programmable Logic and Applications (FPL)*, 2019.

[2] ——, "C^{3}APSULe: Cross-FPGA covert-channel attacks through power supply unit leakage," in *IEEE Symposium on Security and Privacy (S&P)*, 2020.

[3] I. Giechaskiel and J. Szefer, "Information leakage from FPGA routing and logic elements," in *International Conference on Computer-Aided Design (ICCAD)*, 2020.

[4] T. M. La, K. Matas, N. Grunchevski, K. D. Pham, and D. Koch, "FPGADefender: Malicious self-oscillator scanning for Xilinx UltraScale+ FPGAs," *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*, vol. 13, no. 3, Sep. 2020.

[5] S. S. Mirzargar and M. Stojilović, "Physical side-channel attacks and covert communication on FPGAs: A survey," in *International Conference on Field Programmable Logic and Applications (FPL)*, 2019.

[6] G. Provelengios, D. Holcomb, and R. Tessier, "Characterizing power distribution attacks in multi-user FPGA environments," in *International Conference on Field Programmable Logic and Applications (FPL)*, 2019.

[7] 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*, vol. 15, no. 11, pp. 640–642, May 2019.

[8] F. Z. Tazi, C. Thibeault, Y. Savaria, S. Pichette, and Y. Audet, "On extra delays affecting I/O blocks of an SRAM-based FPGA due to ionizing radiation," *IEEE Transactions on Nuclear Science*, vol. 61, no. 6, pp. 3138–3145, Dec. 2014.

[9] S. Tian, A. Krzywosz, I. Giechaskiel, and J. Szefer, "Cloud FPGA security with RO-based primitives," in *International Conference on Field-Programmable Technology (FPT)*, 2020.

[10] S. Tian and J. Szefer, "Temporal thermal covert channels in cloud FPGAs," in *International Symposium on Field-Programmable Gate Arrays (FPGA)*, 2019.

[11] Xilinx, Inc., "Creating a controllable oscillator using the Virtex-5 FPGA IODELAY primitive (XAPP872)," https://www.xilinx.com/support/documentation/application_notes/xapp872.pdf, 2009, Accessed: 2021-07-01.

[12] ——, "7 Series FPGAs SelectIO resources (UG471)," https://www.xilinx.com/support/documentation/user_guides/ug471_7Series_SelectIO.pdf, 2018, Accessed: 2021-07-01.