RESEARCH PAPER

Design and Characterization of Carry Chain-Based Multistage Ring Oscillators on FPGA for Predictable and Tunable Performance

Dr. Radha Krishna A.N[1], Mr. Abdul Maqseed Shaik[2], Ramavath Sharath[3], Megavath Ravindar[4]

[1]Associate Professor, Department of ECE, Brilliant Grammar School Educational Society's Group of Institutions - Integrated Campus, Hyderabad, India
Email: dr.radhakrishnaece@gmail.com

[2]Associate Professor, Department of ECE, Brilliant Grammar School Educational Society's Group of Institutions - Integrated Campus, Hyderabad, India
Email: abdul.maqseed@gmail.com

[3]Department of ECE, Brilliant Grammar School Educational Society's Group of Institutions - Integrated Campus, Hyderabad, India
Email: rickeysharath3@gmail.com

[4]Department of ECE, Brilliant Grammar School Educational Society's Group of Institutions - Integrated Campus, Hyderabad, India
Email: megavathravindar2002@gmail.com

Received: 25th Aug, 2026 | Revised: 1st Sep, 2026 | Accepted: 5th Sep, 2026 | Available Online: 7th Sep, 2026

Abstract

This paper presents a deterministic and tunable field-programmable gate array (FPGA) implementation of multistage ring oscillators using dedicated carry-chain resources. Conventional lookup-table (LUT)-based ring oscillators are strongly affected by place-and-route decisions, routing congestion, and nonuniform interconnect delay, which produce run-to-run frequency variation and limit repeatability. The proposed architecture repurposes the fast carry path, including MUXCY and XOR elements, as a structured delay line and closes the oscillation loop through a controlled inversion stage. Three realizations are considered: CC_ROv1 as the basic carry-chain oscillator, CC_ROv2 with additional pass-through delay elements, and CC_ROv3 with extended selectable delay paths. A propagation-delay model is formulated to relate the number of carry stages and programmable delay elements to oscillation frequency. The design also permits multiphase signal extraction from intermediate nodes without disturbing the feedback loop. This property makes the architecture suitable for physically unclonable functions, true random number generators, temperature and voltage sensing, and on-chip timing characterization. The proposed framework is expected to provide improved frequency predictability, lower area and energy overhead, and finer performance tuning than conventional LUT-based oscillators.

Keywords— FPGA, ring oscillator, carry chain, CARRY4, propagation delay, tunable oscillator, PUF, TRNG, on-chip sensing.

How to cite this article: Radha Krishna A.N1 Mr. Abdul Maqseed Shaik2 Ramavath Sharath3 Megavath Ravindar4. Design and Characterization of Carry Chain-Based Multistage Ring Oscillators on FPGA for Predictable and Tunable Performance. Int J Drug Deliv Technol. 2026;16(75s):1869-1874. DOI: 10.25258/ijddt.16.75s.200.

1. Introduction

Ring oscillators are compact feedback circuits that generate periodic transitions without an external clock source. An odd number of inversions around a closed loop ensures that a logic transition continuously propagates and reappears at the input after one loop delay. Their small hardware footprint and direct dependence on device delay have made them useful in clock generation, timing characterization, hardware security, random number generation, aging monitoring, and environmental sensing.[1]-[4].

In an FPGA, a ring oscillator can be implemented entirely with digital resources. The simplest realization uses LUTs configured as inverters and connects them through the programmable routing fabric. [9] Although this approach is easy to describe in hardware description language, the resulting frequency is not determined only by the number of logic stages. General-purpose routing often contributes a substantial and variable portion of the loop delay. Consequently, identical source code can produce different oscillator frequencies after separate synthesis and implementation runs.[10]-[16].

This limitation is important in applications that depend on controlled frequency relationships. In a ring-oscillator physically unclonable function, frequency comparisons must primarily reflect intrinsic manufacturing variation rather than arbitrary routing choices. In a sensor, the measured frequency shift must correspond to temperature, voltage, or aging rather than implementation noise. In a true random number generator, the oscillator must provide usable jitter while retaining sufficient stability for reliable sampling and post-processing.[5]-[9].

Modern FPGAs contain dedicated carry chains designed for high-speed arithmetic. These structures have tightly coupled multiplexers and XOR gates connected by fixed local routing. Their delay is smaller and more repeatable than ordinary LUT-to-LUT paths. The present work exploits this resource as a

deterministic delay structure for constructing scalable multistage ring oscillators.[17].

The main contributions are: a carry-chain-based oscillator architecture with predictable propagation paths; three configurable variants for coarse and fine delay control; an analytical frequency model; multiphase output extraction; and a design methodology suitable for hardware security, sensing, and timing applications.

$$f_{RO} = \frac{1}{2T_{loop}}$$

Here, T_loop is the total delay around the closed feedback path. In a conventional FPGA implementation, T_loop includes both logic delay and routing delay. The proposed architecture reduces the uncertain routing component by confining most of the propagation path to dedicated carry-chain resources.[13].

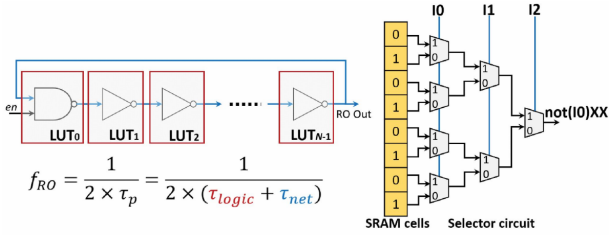

Fig. 1. Conventional LUT-based ring oscillator and LUT internal structure.

2. Related Work

2.1 LUT-Based FPGA Ring Oscillators

Most FPGA ring oscillators reported in earlier work use LUTs as inverters or delay elements. These designs are portable and require little specialized knowledge, but their measured characteristics depend strongly on physical placement and routing. Manual location constraints can improve symmetry, yet they reduce portability and increase implementation effort. LUT-based designs also consume one LUT per stage and may require long interconnects when many oscillators are instantiated.

2.2 Ring Oscillators for PUFs and TRNGs

Ring-oscillator PUFs compare the frequencies of nominally identical oscillators to derive device-specific response bits. Their reliability depends on stable frequency ordering under voltage and temperature variation. Ring oscillators are also used as entropy sources in TRNGs because phase noise and accumulated jitter produce sampling uncertainty. Multi-oscillator and multi-edge architectures improve throughput, but they require compact oscillators with controlled routing and accessible phase information.

2.3 Ring Oscillators as On-Chip Sensors

Because gate and interconnect delays vary with supply voltage, temperature, and aging, ring-oscillator frequency can be used as a digital sensor output. Prior work has employed such structures for voltage monitoring, recycled-device detection, aging assessment, and cloud-FPGA security measurements. For accurate sensing, implementation-induced variation must be minimized or calibrated.

2.4 Dedicated-Resource Oscillator Architectures

Researchers have investigated I/O buffers, sequential elements, DSP blocks, and multiplexers as alternatives to ordinary LUT chains. These approaches reduce some routing overhead but may be limited in scalability, frequency range, or fine-grained tunability. Carry chains are especially attractive because they are physically contiguous and optimized for fast propagation. However, a unified multistage architecture that combines deterministic delay, configurable tuning, multiphase access, and analytical modeling remains desirable.

2.5 Research Gap

The literature indicates four unresolved needs: repeatable routing without extensive manual constraints, scalable stage count, fine frequency adjustment beyond coarse changes in inverter count, and controlled environmental sensitivity. The proposed CC_ROv1, CC_ROv2, and CC_ROv3 structures address these needs by combining fixed carry-chain paths with selectively inserted programmable delay elements.

3. Methodology

3.1 Overall Design Principle

The proposed ring oscillator replaces the majority of the LUT-based delay loop with a cascade of FPGA carry-chain cells. Each carry stage contains a fast multiplexer path and an XOR output. By selecting suitable static inputs, the carry signal is forced to propagate through a known sequence of internal elements. A single control/inversion stage closes the feedback loop and provides an enable function so that oscillation can be started or stopped safely.

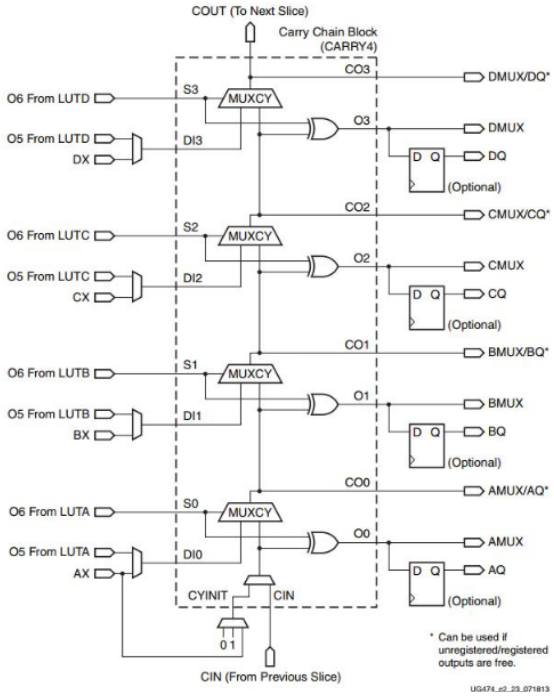

Fig. 2. Internal CARRY4 structure showing the MUXCY and XOR propagation paths.

3.2 Carry-Chain Oscillator Variants

CC_ROv1 is the basic implementation. It uses the dedicated carry path as the dominant delay line and a single LUT-based NAND or inversion stage to provide loop inversion and enable control. If x carry units are cascaded and each unit contributes two effective transition stages, the approximate number of stages is

$$N_{eff} = 2x + 1$$

CC_ROv2 extends the basic structure by inserting one or more pass-through LUT elements in selected locations. These elements add a controlled programmable delay without changing the main carry-chain topology. CC_ROv3 provides multiple selectable pass-through branches, allowing a broader tuning range and finer control of frequency and environmental sensitivity.

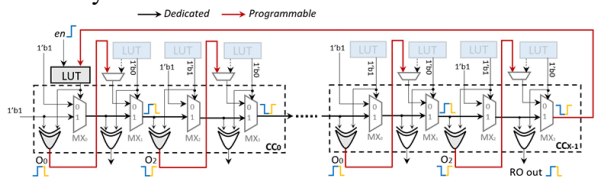

Fig. 3. Carry-chain-based oscillator variants with dedicated and programmable delay paths.

3.3 Propagation-Delay Model

The oscillation period is twice the total one-way propagation delay around the loop. The loop delay is modeled as the sum of carry-chain delay, inversion/control delay, optional pass-through delay, and residual local routing delay.

$$T_{loop} = N_c t_c + N_x t_x + N_p t_p + t_{inv} + t_r$$

$$f_{RO} = \frac{1}{2\left(N_c t_c + N_x t_x + N_p t_p + t_{inv} + t_r\right)}$$

In these expressions, NC and N_x denote the number of carry multiplexer and XOR contributions, respectively; N_p is the number of enabled pass-through LUT stages; t_c, t_x, and t_p are their average delays; t_inv is the feedback inversion delay; and t_r represents residual routing. Because most stages are implemented through fixed carry resources, t_r is expected to be smaller and less variable than in a conventional LUT-based oscillator.

3.4 Frequency Tuning and Multiphase Extraction

Coarse tuning is obtained by changing the number of cascaded carry units. Increasing the stage count increases the loop delay and lowers the oscillation frequency. Fine tuning is achieved by enabling selected pass-through elements in CC_ROv2 or CC_ROv3. Intermediate carry-chain outputs are buffered and routed to observation logic to obtain phase-shifted versions of the oscillation signal. These taps must be isolated from the loop so that measurement loading does not alter the oscillator frequency.

3.5 FPGA Implementation Flow

1. Describe each oscillator variant using structural Verilog and vendor-supported carry primitives.
2. Apply preservation attributes so that synthesis does not remove or simplify the intentional combinational loop.
3. Implement all variants using the same device, tool version, clocking conditions, and measurement window.
4. Measure frequency with a synchronous counter referenced to a stable external clock.
5. Repeat implementation runs to quantify run-to-run variation and repeat measurements across voltage and temperature conditions.
6. Record LUT, carry, and flip-flop utilization together with dynamic power and energy per oscillation cycle.

3.6 Evaluation Metrics

The principal evaluation metrics are mean oscillation frequency, standard deviation across implementation runs, coefficient of variation, frequency tuning resolution, resource utilization, dynamic power, and energy per cycle. For PUF-oriented evaluation, inter-device uniqueness and intra-device reliability may also be measured. For TRNG use, the generated sequence should be examined using standard statistical randomness tests.

$$\mathbf{CV(\%)} = \left(\frac{\sigma_f}{\mu_f}\right) \times 100$$

$$\mathbf{E_{cycle}} = \frac{\mathbf{P_{dynamic}}}{\mathbf{f_{RO}}}$$

4. Results and Discussion

FPGA Device Mapping and Physical Feasibility

Figure 1 presents the synthesized device view of the selected Spartan-7 FPGA. The device is divided into clock regions labelled X0Y0 through X1Y3. The successful opening of the synthesized design confirms that the ring oscillator architecture was accepted by the synthesis tool and mapped to the target FPGA fabric. The design occupies only a very small portion of the available programmable resources, indicating a compact hardware implementation. Since the oscillator is based on carry-chain propagation, final placement and routing are important because routing asymmetry directly influences loop delay and oscillation frequency. The present device view confirms implementation feasibility, while post-implementation placement constraints should be applied for repeatable physical characterization.

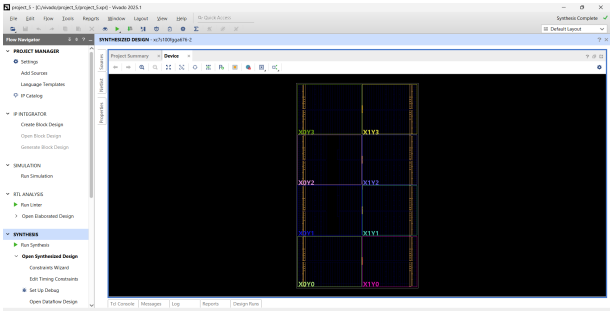

Figure 4. Synthesized Spartan-7 FPGA device view showing the available clock regions.

2. Power Consumption Analysis

The Vivado power estimator reports a total on-chip power of 0.108 W. Device static power is 0.097 W, corresponding to approximately 90% of the total estimated consumption, whereas dynamic power is 0.011 W, corresponding to approximately 10%. Within the dynamic component, I/O power is 0.010 W, logic power is 0.001 W, and signal power is below 0.001 W. Thus, the internal oscillator logic contributes only a small portion of the overall FPGA power budget. The junction temperature is estimated as 25.3 °C at an ambient temperature of 25.0 °C, with a thermal margin of 59.7 °C. These values indicate safe thermal operation under the assumed activity conditions.

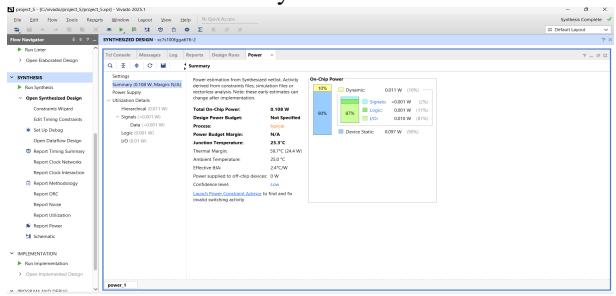

Figure 5. Vivado synthesized-design power estimation for the proposed oscillator.

3. Synthesis Report and Design Correctness

The synthesis report shows that the complete design was synthesized successfully with zero errors and zero critical warnings. Three ordinary warnings were reported. The synthesis run required approximately 42 seconds of elapsed time, and the peak memory usage was approximately 1.79 GB. The report also identifies a very small instance area at the top level, supporting the compactness of the design. Successful synthesis confirms that the Verilog description is structurally valid and compatible with the selected FPGA technology.

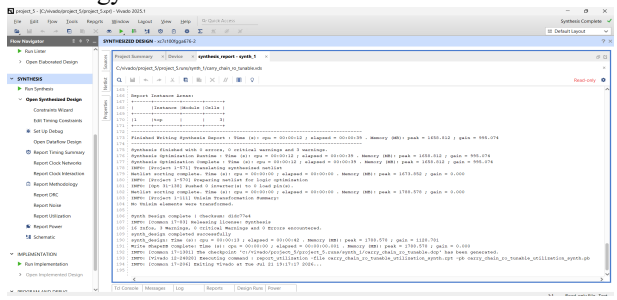

Figure 6. Vivado synthesis report showing successful completion with zero errors and zero critical warnings.

Although the absence of critical warnings is encouraging, the three remaining warnings should be inspected before hardware programming. In ring-oscillator designs, synthesis tools may report combinational-loop or timing-related warnings because intentional feedback is used to generate oscillation. Such warnings do not necessarily indicate a design failure, but the feedback path must be preserved using appropriate synthesis attributes and implementation constraints so that the tool does not optimize away or restructure the oscillator loop.

RTL Architecture Verification

The elaborated schematic verifies the intended multistage architecture. Eight generated stages, denoted TUNABLE_STAGE[0] through TUNABLE_STAGE [7], are visible in the design. Each stage contains a carry_chain_stage block with carry input ci_in and carry output co_out. The stages are interconnected to form the main propagation path. An RTL multiplexer controlled by the enable input is placed in the feedback path, allowing the oscillator to be activated or held in a stable state. Programmable delay_insert_unit blocks controlled by tune_en are included to modify the effective loop delay.

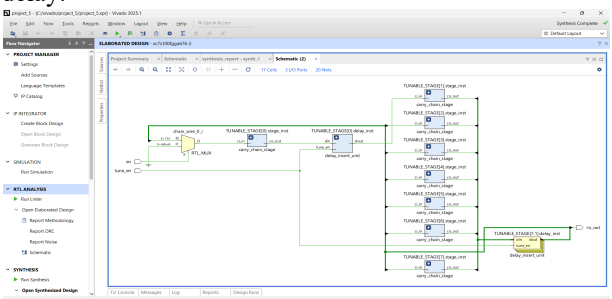

Figure 7. Elaborated RTL schematic of the eight-stage tunable carry-chain ring oscillator.

The oscillation frequency is governed by the accumulated propagation delay around the closed loop and can be approximated by:

$$f_{RO} = \frac{1}{2T_{loop}}$$

For the proposed multistage architecture, the total loop delay may be represented as:

$$T_{loop} = N_c t_c + N_d t_d + t_{routing} + t_{feedback}$$

where N_c is the number of carry-chain stages, t_c is the delay of each carry stage, N_d is the number of enabled delay elements, and t_d is the delay introduced by each programmable unit. Increasing the number of active delay elements increases T_loop and therefore reduces f_RO. This relationship provides the required tunability while the fixed carry-chain stages support predictable fine-grained delay accumulation.

Simulation Result

The ModelSim waveform contains the reference clock, reset, start, tuning control, oscillator selection inputs, completion flag, response bit, and two counter outputs. At the displayed cursor position, rst is asserted, start is low, tune_en is low, sel_a is 000, sel_b is 001, done remains low, puf_bit remains low, and both count outputs remain zero. Therefore, the screenshot represents the reset or initialization condition of the measurement system. The zero-valued counters and inactive completion flag are consistent with the asserted reset and inactive start control.

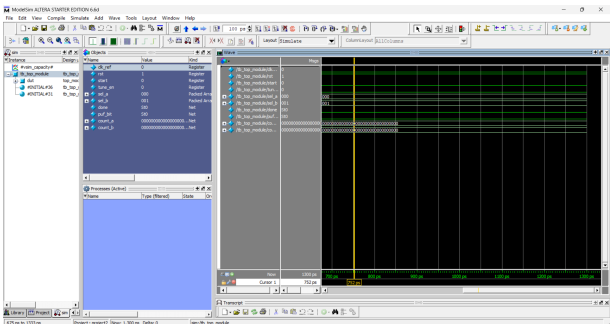

Figure 7. ModelSim waveform showing the reset-state behavior of the oscillator measurement system.

The displayed time window extends only to approximately 1300 ps and does not show a complete measurement transaction. Consequently, the screenshot confirms stable initialization but does not yet provide numerical oscillator frequency, counter comparison, or tuning-range data. To demonstrate complete functionality, the simulation should be extended after deasserting reset, followed by assertion of start and activation of tune_en. The ring-oscillator output should then be counted over a defined reference-clock interval for each tuning setting. The frequency can be obtained from f_RO = N_RO/T_gate, where N_RO is the number of counted oscillator transitions during the measurement gate T_gate.

Consolidated Result

The obtained results validate the basic design and FPGA feasibility of the proposed carry chain-based multistage ring oscillator. The design was successfully elaborated and synthesized on the Spartan-7 device with no errors or critical warnings. The schematic confirms an eight-stage carry-chain structure, an enable-controlled feedback loop, and programmable delay insertion for frequency tuning. The architecture is compact, with only 17 elaborated cells, three I/O ports, and 20 internal nets reported in the schematic view. The synthesized power estimate is 0.108 W, of which only 0.011 W is dynamic power, while the estimated junction temperature remains close to ambient temperature.

These observations demonstrate that the proposed oscillator can be implemented with low area and low logic-power overhead. The use of FPGA carry resources provides a practical basis for predictable stage-to-stage delay, while the inserted delay units enable controllable modification of the total loop delay. The current simulation establishes correct reset behavior; complete performance characterization should additionally report measured frequency for each tuning code, frequency step size, tuning range, coefficient of variation across repeated placements, and variation under voltage and temperature changes. With these additional measurements, the architecture can be effectively evaluated for process-voltage-temperature sensing, delay characterization, physically unclonable functions, entropy generation, and embedded timing-monitoring applications.

5. Conclusion

This paper presented a carry-chain-based multistage ring oscillator architecture for FPGA platforms. By using dedicated carry resources as the principal delay path, the design reduces dependence on general-purpose routing and improves the predictability of oscillator behavior. The basic CC_ROv1 structure provides a compact and scalable oscillator, while CC_ROv2 and CC_ROv3 introduce programmable pass-through elements for fine frequency tuning and sensitivity control. The analytical delay model links physical stage composition to the expected oscillation frequency and supports design-space exploration before implementation. The ability to access intermediate stages also enables multiphase outputs without modifying the feedback loop. These features make the proposed architecture well suited to PUFs, TRNGs, on-chip voltage and temperature sensors, aging monitors, and timing-characterization systems. Future work may investigate automatic calibration, dynamic reconfiguration, cross-device portability, and machine-learning-assisted delay prediction.

References

[1] F. Kodýtek, R. Lórencz, and J. Bucek, “Three counter value based ROPUFs on FPGA and their properties,” Microprocessors and Microsystems, vol. 88, 2022.

[2] B. Halak, M. Zwolinski, and M. S. Mispan, “Overview of PUF-based hardware security solutions for the Internet of Things,” in Proc. IEEE Midwest Symp. Circuits and Systems, 2016.

[3] M. Barbareschi et al., “A ring oscillator-based identification mechanism immune to aging and external working conditions,” IEEE Trans. Circuits Syst. I, 2018.

[4] B. Yang et al., “ES-TRNG: A high-throughput true random number generator,” IACR Trans. Cryptographic Hardware and Embedded Systems, 2018.

[5] M. A. Prada-Delgado et al., “Auto-calibrated ring-oscillator true random number generator,” in Proc. IEEE Int. Symp. Circuits and Systems, 2020.

[6] M. Grujić and I. Verbauwhede, “TROT: A ring oscillator-based true random number generator,” IEEE Trans. Circuits Syst. I, 2022.

[7] S. Moini et al., “Voltage sensor implementations for FPGA attacks,” ACM Trans. Reconfigurable Technology and Systems, 2022.

[8] I. Giechaskiel et al., “Measuring long-wire leakage with ring oscillators,” in Proc. Int. Conf. Field-Programmable Logic and Applications, 2019.

[9] M. Ebrahimi and Z. Navabi, “FPGA aging detection using delay paths,” IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems, 2020.

[10] M. M. Alam et al., “Recycled FPGA detection using delay characterization and voltage scaling,” IEEE Trans. Very Large Scale Integration Systems, 2019.

[11] B. Razavi, “The ring oscillator,” IEEE Solid-State Circuits Magazine, 2019.
[12] M. Elnawawy et al., “The role of FPGA technology in Internet-of-Things applications,” in Proc. IEEE Int. Symp. Signal Processing and Information Technology, 2019.
[13] C. Gu et al., “Evaluation of ring-oscillator PUFs on 28-nm Xilinx FPGAs,” 2019.
[14] A. Wild et al., “Challenges in reliable and efficient ring-oscillator PUF implementation,” 2016.
[15] J. Burgiel et al., “IOBUF-based ring oscillators for FPGA sensing and timing applications,” 2021.
[16] T. Sugawara et al., “An oscillator without a combinational loop for FPGA implementations,” 2019.
[17] R. D. Sala et al., “A compact FPGA true random number generator based on ring oscillation,” 2022.