Final Project 3 Report: Implementing Ring Oscillator Physical Unclonable Function

Student Name: Beichen Su
Department: *ECE department of University of Florida*
Group: *EEE5716/EEE4714: Introduction to Hardware Security and Trust --- Project 3 EDGE UG Group 1*
UFID: 4113-7166
Email: su1998@ufl.edu
Location: Gainesville, FL

***Abstract*—This report documented the approach and evaluation for implementing a 32-bit ring oscillator (RO) based physical unclonable function (PUF) on field programmable gate arrays (FPGAs). As requested, I present and analyze the designed architecture of RO-PUF and evaluation of the quality of the designed PUF using the metrics- reliability, uniqueness, and uniformity /randomness. The measurements are performed on 2 identical the DE10-Lite FPGA Board across normal temperature. There are total 6 different challenge sets are inputted to FPGA board, each challenge set has 32 different 32-bit challenges and a database of challenge response pairs (CRPs) is collected, which will also be presented.**

Keywords— physical unclonable function, hardware security, FPGAs, IEEE.

I. INTRODUCTION

Information or data security have remained an issue of paramount concern for system designers and users alike since the beginning of computers and networks. Consequently, protection of systems and networks against various forms of attacks, targeting corruption/leakage of critical information and unauthorized access, have been widely investigated over the years. emerging trends in electronic hardware production, such as intellectual-property-based (IP-based) system on chip (SoC) design, and a long and distributed supply chain for manufacturing and distribution of electronic components— leading to reduced control of a chip manufacturer on the design and fabrication steps—have given rise to many growing security concerns. [1]

Physical Unclonable Function is a popular function used to protect hardware information nowadays that based on the physical system, easy to evaluate but unpredictable even for an attacker with physical access. PUFs aim at addressing the shortcomings of the digital key storage by relying on the secrets generated by the inherent and unclonable unique mesoscopic characteristics (signatures) of the physical phenomena. [2] [3] The physical properties of each device determine a specific mapping between a set of challenges (inputs) to a set of responses (outputs). Security protocols take advantage of the unique mappings provided by the CRPs to authenticate the device and/or its components [4]. There are multiple different types of PUFs used currently, this report will mainly focus on RO based PUF as requested.

FPGAs provide a generic substrate of interconnected blocks that can be (re)programmed to achieve the desired functionality. The inherent flexibility of FPGAs compared to Application Specific Integrated Circuits (ASICs) together with their lower time-to-market as well as availability of third party IPs, have made them the platform of choice for many applications. [5]

This report specifically records a way of implementing RO PUF on FPGAs that enable different devices to provide unique response pairs in return after issued a challenge. RO PUF is composed of multiple delay loops that oscillate with clock frequency. The routing between different ring oscillator blocks are nearly identical with minor variations in manufacturing which lead to loops with slightly different frequencies. The counters will be driven by loops during certain time period and produce the response bits to a given challenge. Using the measurement data collected from PUFs on two identical FPGA across same temperature environment, this report quantifies the response uniqueness, reliability, and randomness of each group.

The rest of the report is organized as follows. In Section 2, the presentation of my design of RO PUF and the way of implementation using Quartus. In Section 3, I provide the data analysis and experimental evaluation about the data I

collected. In Section 4, concludes the report. In Appendix, all my VHDL codes of implementation and test data are provided.

II. DESIGN

A. *The Problem of the basic RO PUF design*

As mentioned eariler, I will design a 32 bit ring oscillator PUF as preoject requested. I started with a fairly basic design learned from lecture like Figure 1 shown. During the process of studying and researching, I found out some problems of the basic design of RO PUF:

- It includes an enable bit on the ring oscillator design for sake of preventing chip to be overheated. However, there is no time limit among counter, so it will keep counting the times of the assigned signal becoming one all the time and comparing with the other counter. Because of the struture of the design, the basic RO PUF will still consume a lot of power and keep outputing the results of the comparision of the counters.
- Since there is only one inputting challenge inserted to the MUX according to the design, the assigned signals must be same, which, to some degree, limits the range of the choices among signals and the randomness of the experimental data.

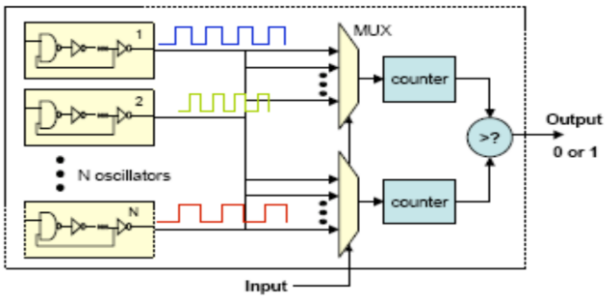

Figure 1. Basic Structure of RO PUF

B. *My Design of RO PUF*

In order to have a strong and secure PUF, Figure 2 shows how I use 65536 (2^16) ring oscillators in combination with some other components to generate the output. Each ring oscillator contains 51 not gates. I reused 65536 ring oscillators connecting with both 65536 to 1 MUX. I will seperate 32 bit challenge input into two groups and connect to separate MUX in order to have more choices of different signal combinations and increase the randomness of the results. I add an enable bit as an output from timer to connect with counters. Through VHDL codes, I can set certain time period like 0.5 us for the timer. When RST is on, enable and ready will be off. When RST is off, enable will become 1 and counter will start couting the assigned signal from challenge. Once the period achieved, ready will be on and counters will submit their data to comparator to generate the output bit. A one is output if the first counter value is larger than the second, and a zero will be output if not. It won't work again until timer is maually reset.

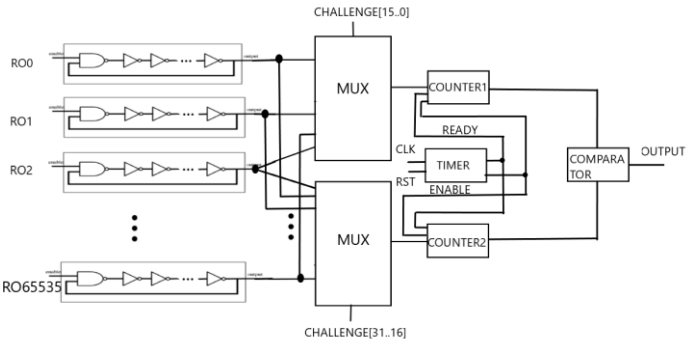

Figure 2. My Design of RO PUF

C. *The Novelty of My Design*

- Increase the combination choices of the signals and increase the randomness
- Save more energy from ring oscillator and make PUF more efficient with a timer.
- It will not keep outputting the response unless you allow it to.
- Absolutely strong and secure PUF with 65536 Ros to ensure randomness.

D. *Routing of the New RO PUF*

Keeping the routing identical between ring oscillator blocks is crucial since an RTL-based design without any routing constraints allows the synthesis and layout tool to provide an optimized structure which typically is not symmetric and identical. In Quartus, simply inputting not gate 51 times will not work since it will automatically simplify the gates to only one not gate. Thus, after consulting and researching, I changed not gates into LCELL buffers which will keep all gates in routing. In total, there are 285 pages structures like Figure 3 to implement all ring oscillators. Figure 4 shows the center of all structure which will be a comparator to ensure the symmetric design. Figure

5 shows a close look of each of the routed RO PUF.

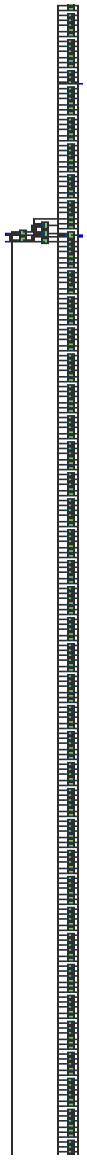

Figure 3. Symmetric Routed Structure of RO PUF

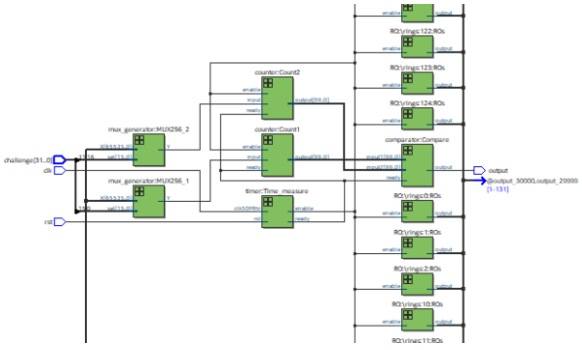

Figure 4. Closer version of RO PUF

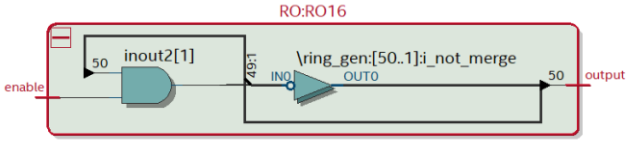

Figure 5. RTL View of the RO Structure

III. EXPERIEMNT EVALUATION

A. *Experiemntal Setup*

The success of PUF need to be viewed by its performance in reliability, uniqueness, and uniformity /randomness. First, for a challenge that is given to PUF, the output response ideally should always be same under all circumstances. Second, the responses from different PUFs to a same challenge should not be same due to the uniqueness of PUF. Finally, for two different challenges, the responses should be different. All perspectives are important to the evaluation of PUF design. In order to fully evaluate the performance of PUF, I design the following tests on the DE10-Lite FPGA Board.

B. *Test*

Due to the limitation of man power and the access to the equipment, I can only perform small amounts of tests. Because of the limitation of switches that I can use for testing, I put switch 9 to switch 5 connecting to the first mux and switch 4 to switch 0 connecting to the second mux to compare with each other. In addition, the other 22 bits of input will be manually connected to ground for each challenge. I will input 6*32 = 192 different 32 bits challenges, and each challenge will be performed twenty-five times under two different devices in normal temperature. Then I will analysis data according to four perspectives below:

- One PUF, One Challenge Test:

  A signal device is given a 32-bit challenge 25 times and this process will be performed 192 times. Every response of a challenge should be exact same, which means the ideal intra-hamming distance should be zero

- Two PUFs, Multiple Challenges Test

  Two different signal devices are given 192 different 32-bit challenges. The average inter-hamming distance between responses is calculated, and the

ideal average inter-hamming distance is 50 percent.

The average inter Hamming Distance (HD) among a group of chips and is defined as

$$\overline{\text{Inter}-d_{\text{HD}}}(k)=\frac{2}{k(k-1)}\sum_{i=1}^{k-1}\sum_{j=i+1}^{k}\frac{\text{HD}\left(R_i,R_j\right)}{n}\times 100\%.$$

This distance should converge to 50% in the case of an ideal PUF.

The average intra HD over samples for the chip is used to ensure the reliability of a PUF and is defined as

$$\overline{\text{Intra}-d_{\text{HD}}}(x,i)=\frac{1}{x}\sum_{y=1}^{x}\frac{\text{HD}\left(R_i,R_{i,y}^{'}\right)}{n}\times 100\%.$$

This distance should be close to 0% to ensure reliable responses from the PUF in a given chip at various operating conditions. [6]

C. *Analysis*

The results are collected from MATLAB and shown in table 1. For One PUF, One Challenge Test, I calculated the average and standard deviation of intra-hamming distance of all sets in both FPGAs. Most of sets in both FPGAs are smaller than 10%. The overall correctness of PUF response is not completely close to 0%, but these are not bad value to prove the high reliability of my PUF design. Second, through the comparation of both FPGAs, I calculated the average inter-hamming distance for each set, which is around 47%. The results are close to 50% and prove the absolute uniqueness and randomness of my PUF.

Experimental Data							
		set1	set2	set3	set4	set5	set6
FPGA1 & FPGA2	inter_hamming_average	0.54	0.44	0.56	0.46	0.38	0.48
	inter_hamming_std	0.40620192	0.40104031	0.32659863	0.35118846	0.29860788	0.33788558
FPGA1	intra_hamming_average	0.09333333	0.06	0.04666667	0.11333333	0.06	0.05333333
	intra_hamming_std	0.08443713	0.08164966	0.07637626	0.0793492	0.08164966	0.0793492
FPGA2	intra_hamming_average	0.08666667	0.05333333	0.04	0.04	0.06666667	0.10666667
	intra_hamming_std	0.08498366	0.0793492	0.07264832	0.07264832	0.08333333	0.08164966

Table 1. Experimental Results

Due to the limitation of equipment, I cannot test the effects of temperature or voltage variation to further prove the reliability of my design.

IV. Conclusion

In conclusion, I was able to successfully design, implement and test the ring oscillator PUF. Through the testing evaluation and data analysis, the functionalities of RO PUF are demonstrated. However, there are still a lot of areas in which additional work could be done like giving more testing. Since the limitation of my challenges and device, the experimental results are highly unconvincing. If I want to demonstrate the fully functionalities, I need more proof and data analysis. Due to the limitation of access experiment equipment, I couldn't demonstrate how voltage variation and temperature variation will affect the responses and how these two will be related. In addition, my design is a strong PUF with good randomness but not perfectly secure since instability and slight asymmetric structure still exist. So, if I had chance in the future, I will keep refining and improving my design to achieve absolute safe and secure.

Through the process of designing, I understood a lot of definition and details that I will never learn from books and lectures. In addition, the experience of finding problems from basic design and designing new structure with novelty is extremely important to undergraduate students like me. I really enjoyed the design process and satisfied about what I am capable of designing right now. There are some regrets in my design, but what I gain I believe are more valuable for my future.

References

[1] Tehranipoor and C. Wang (Eds.), Introduction to Hardware Security and Trust, *Springer*, 2011

[2] R. Pappu, B. Recht, J. Taylor, and N. Gershenfeld. 2002. Physical one-way functions. Science 297 (2002), 2026–2030.

[3] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas. 2002. Silicon physical random functions. In CCS. 148–160.

[4] M. Majzoobi, M. Rostami, F. Koushanfar, D.S. Wallach, and S. Devadas. 2012. Slender PUF Protocol: A Lightweight, Robust, and Secure Authentication by Substring Matching. In IEEE Symposium on Security and Privacy Workshops (SPW). 33 – 44.

[5] M. Majzoobi, A. Kharaya, F. Koushanfar and S. Devadas , "Automated design, implementation, and evaluation of arbiter - based PUF on FPGA using programmable delay lines" , 2014.

[6] Florent Bernard, Viktor Fischer, Crina Costea, and Robert Fouquet, "Implementation of Ring-Oscillators-Based Physical Unclonable Functions with Independent Bits in the Response," International Journal of Reconfigurable Computing, vol. 2012, Article ID 168961, 13 pages, 2012. https://doi.org/10.1155/2012/168961.