A highly reliable butterfly PUF in SRAM-based FPGAs

Xiumin Xu¹, Huaguo Liang¹, Zhengfeng Huang¹ a), Cuiyun Jiang², Yiming Ouyang³, Xiangsheng Fang⁴, Tianming Ni¹, and Maoxiang Yi¹
¹ School of Electronic Science and Applied Physics, Hefei University of Technology, 193 Tunxi Rd., Baohe District, Hefei 230009, P. R. China
² School of Mathematics, Hefei University of Technology, 193 Tunxi Rd., Baohe District, Hefei 230009, P. R. China
³ School of Computer and Information, Hefei University of Technology, 193 Tunxi Rd., Baohe District, Hefei 230009, P. R. China
⁴ Department of Information Project, Anhui Institute of Economic Management, 115 Wangjiang Rd., Baohe District, Hefei 230051, P. R. China
a) huangzhengfeng@139.com

Abstract: This paper presents a butterfly physically unclonable function (PUF) implementation in SRAM-based field programmable gate arrays (FPGAs). To avoid output instability, we propose a delay difference test to identify reliable slices (mapped to which butterfly PUF cells are highly reliable) and then PUF reliability is significantly improved by selective mapping PUF cells to reliable slices, which is validated in experimental results.

Keywords: physically unclonable function (PUF), SRAM-based FPGAs, delay difference test, selective mapping, highly reliable

Classification: integrated circuits

References

[1] S. Tao, et al.: “Ultra-energy-efficient temperature-stable physical unclonable function in 65 nm CMOS,” Electron. Letter 52 (2016) 805 (DOI: 10.1049/el.2016.0292).
[2] C. Herder, et al.: “Trapdoor computational fuzzy extractors and stateless cryptographically-secure physical unclonable functions,” IEEE Trans. Depend. and Secure Comp. 14 (2016) 65 (DOI: 10.1109/TDSC.2016.2536609).
[3] L. Gao, et al.: “Physical unclonable function exploiting sneak paths in resistive cross-point array,” IEEE Trans. Electron Devices 63 (2016) 3109 (DOI: 10.1109/TED.2016.2578720).
[4] Y. Yu, et al.: “Improving RO PUF design using frequency distribution characteristics,” IEICE Electron. Express 12 (2015) 20141043 (DOI: 10.1587/exel.12.20141043).
[5] C. Gu, et al.: “Ultra-compact and robust FPGA-based PUF identification generator,” ISCAS (2015) 934 (DOI: 10.1109/ISCAS.2015.7168788).
[6] M. Uddin, et al.: “Robustness analysis of a memristive crossbar PUF against modeling attacks,” IEEE Trans. Nanotech. 16 (2017) 396 (DOI: 10.1109/TNANO.2017.2677882).
[7] H. Yu, et al.: “An FPGA chip identification generator using configurable ring oscillators,” IEEE Trans. Very Large Scale Integr. Syst. 20 (2011) 2198 (DOI: 10.1109/TVLSI.2011.2173770).
1 Introduction

Physically unclonable function (PUF) [1-3], an innovative hardware security technology, exploits physical randomness derived from uncontrollable integrated circuit (IC) manufacturing process [4] to generate high-security secret keys. Good randomness in PUF outputs comes from highly symmetrical placement and isometric routing [5], i.e., no systematic skews in the circuit implementation, only in this way can the minute and random mismatches induced by manufacturing process dominate the PUF outputs.

Having been implemented with a nominally symmetric layout, the PUF outputs are likely to change from time to time (i.e., poor reliability) [6] when the fluctuation caused by operating conditions and thermal noise is greater than or close to the manufacturing mismatch. The conventional method to improve PUF reliability is based on the use of error correction codes (ECC), while the approach significantly increases the complexity and cost of the design, and worse, the correcting code could reveal some important information. Another widely used method is mismatch selection, for instance, adjusting ring oscillator (RO) configuration to select maximum frequency difference between two ROs with the same layout, and then the authors in [7] greatly improve RO PUF reliability.

While in the case of simple-structure PUFs (e.g., SRAM PUF and butterfly PUF), it is quite difficult to perform an accurate mismatch estimation since propagation delay of each chain is too small to be measured. Based on multiple enrollment tests in different environmental conditions, a methodology to identify unreliable bits is proposed in [8] in which choosing only the reliable ones in the PUF implementation leads to high reliability. However, the approach is not recommended due to high complexity, time consuming, and limited precision (inability to cover all operating conditions). In this paper, we propose a novel mismatch estimation approach in SRAM-based field programmable gate arrays (FPGAs), utilizing the effect of load adjustment (fan-outs increasing/decreasing) on chain delay, to identify reliable slices (large delay difference between two chains) and then selective mapping PUF cells to the reliable slices, effectively improving PUF reliability.

[8] K. Xiao, et al.: “Bit selection algorithm suitable for high-volume production of SRAM-PUF,” HOST (2014) 101 (DOI: 10.1109/HST.2014.6855578).
[9] E. I. Vatajelu, et al.: “Towards a highly reliable SRAM-based PUFs,” DATE (2016) 273 (DOI: 10.3850/9783981537079_0753).
[10] A. Rukhin, et al.: “A statistical test suite for random and pseudorandom number generators for cryptographic applications,” NIST Special Publication 800-22 (2001).
2 PUF Implementation

Our proposed butterfly PUF cell actually is a simple SR NAND latch (can be compactly mapped to single slice) as shown in Fig. 1 in which four D-type flip-flops (DFFs) surround the latch (avoid direct connection to external routing resources) in order for a highly symmetrical layout. And appending one buffer to each chain is to facilitate fan-out insertion for load adjustment. It should be noted that, in the paper, a fan-out insertion (to a chain) is implemented by connecting one node in the chain to an input port of an adjacent look up table (LUT). According to the Post-Place & Route Static Timing, each inserting a fan-out, chain delay is increased by 8 picoseconds, while actually the mean delay increment in three ML605 platforms (equipped with a 40 nm Virtex-6 FPGA) is about 3.4 ps as shown in Fig. 2.

![Fig. 1. The structure of butterfly PUF cell.](image1)

![Fig. 2. Delay increment versus each fan-out insertion with three ML605 platforms and two placements on each platform.](image2)

A LUT has 6 inputs, thus mapping a 2-input NAND to single LUT has 30 choices. Taking four LUTs and four DFFs (one slice) into consideration, and then mapping the PUF cell to one slice has a large number of configuration
cases. With the help of FPGA Editor, PUF is implemented with a highly symmetric case with negligible delay difference between the two chains (the mismatch nominally is just 4 ps).

3 PUF Output

As shown in Fig. 1, firstly, the latch state is initialized when setting CLR to 1, then Y1Y2=11. Then by applying a low-to-high transition on TEST, both outputs, i.e., Y1 and Y2, go towards changing their value from 1 to 0. Since there may be slight propagation delay difference between the two chains, one chain may change its value sooner than the other, which causes the slower one to keep its state. This means that Y1Y2=01 (PUF outputs 0), if chain1-chain2<0, and Y1Y2=10 (PUF outputs 1), otherwise.

To have an in-depth understanding of PUF output, two chains’ delay difference, determined by both manufacturing mismatch (marked in green) and the fluctuation (marked in yellow), is illustrated in Fig. 3 in which manufacturing mismatch follows a similar Gaussian distribution [9]. As shown in Fig. 3a, when manufacturing mismatch mapped to a slice is small, PUF output swings between 0 and 1. This means that the PUF cell cannot produce reliable bit when operating conditions change, in other words, the slice is unsuitable to be mapped since corresponding PUF cell does not show a strong preference to any of the two states (0 and 1). In the complementary case shown in Fig. 3b, PUF output has always been a state due to a large manufacturing mismatch and thus suitable for PUF purpose.

![Fig. 3. PUF output reliability in two cases of manufacturing mismatches: a) unreliable when the mismatch is small; b) reliable when the mismatch is large.](image)

4 Improving PUF Reliability

The basic idea of improving PUF reliability is to selectively place PUF cells to reliable slices, mapped to which the manufacturing mismatch between the two chains is large, in such a way that the bit flipping can be considerably reduced (i.e., high reliability). To identify the reliable slices, in this paper, we exploit the influence of load adjustment (fan-outs increasing/decreasing) on chain delay to determine the manufacturing mismatch.
As illustrated in Fig. 4a, if manufacturing mismatch is small, two PUF outputs in both cases of respectively inserting fan-outs to chain1 and chain2 are different. Fig. 4b shows an opposite case where manufacturing difference is large and thus the PUF outputs in both cases are same. Obviously, based on whether the PUF outputs in both cases are same, we can determine whether the manufacturing mismatch is larger than the setting value, i.e., delay increment of each fan-out insertion multiplied by the number of inserting fan-outs. In other words, we can identify the desirable manufacturing mismatch through configuring corresponding number of fan-outs insertion.

The setting value directly affects the efficiency of the method. Using a low value introduces the risk of under-testing, i.e., some of the slices which cause PUF unreliability remained undetected during enrollment test. If the value is too high, there is the risk of over-testing, i.e., more suitable slices are eliminated then needing more enrolled slices. It should be noted that the delay difference test is easily implemented in SRAM-based FPGAs due to rich routing resources and dynamic partial reconfiguration (DPR) capabilities.

5 Experimental Results

In the experiment, multiple groups of fan-outs insertion on three ML605 platforms are chosen to validate the effectiveness of our proposed reliability improvement methodology. As shown in Table I, firstly, 300 instances of butterfly PUF cells are implemented and each slice is defined as reliable slice if corresponding PUF instance outputs in both cases (respectively inserting fan-outs to chain1 and chain2) are same at enrollment test (enrollment environment is 25 degrees Celsius and 1 V). Then, we implement a 128-bit PUF which mapped to 128 reliable slices and the PUF is tested 630 (7*9*10) times under a wide range of operating conditions (changing the temperature from 20 degrees Celsius to 80 degrees Celsius in 10 degrees Celsius steps and the core voltage from 0.8 V to 1.2 V in 0.05 V steps), the PUF cell remaining unchanged in all measurements is considered as reliable cell.

As shown in Table I, taking chip1 for example, when the number of fan-outs insertion is 1, the number of detected suitable slices is 245. Unsurpris-
ingly, the test coverage is lower than 100%, i.e., not all unreliable slices have been detected, thus the reliability probability is 78.9%. Increasing the number to 4, the number of defined reliable slices is 186 and the test coverage is larger than 100% (PUF reliability is 100%). It is clear that our proposed reliability improvement methodology can effectively improve PUF reliability.

The NIST statistical test suite [10], established for evaluating random number generators, here is used to prove PUF randomness. In this experiment, 99 128-bit PUF bitstreams are implemented in three ML605 platforms. Table II gives detailed test result from which we conclude that our proposed PUF passes the randomness test and acquires good randomness.

| 6 Conclusion |

The fluctuation caused by operating conditions and thermal noise may damage PUF’s ability to reproduce the identical output. In the paper, we propose a novel manufacturing mismatch estimation to identify reliable slices and then selective mapping effectively improves PUF reliability.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (grant nos. 61674048, 61574052, 61474036, and 61371025) and Project Team of Anhui Institute of Economics and Management of China (no. YJK-T1417T01).

### Table I. Reliability test with fan-outs insertion on three ML605 platforms.

| ML605 Platform | The Number of Fan-outs Insertion (%) | The Number of Defined Reliable Slices (%) | The Number of Reliable PUF Cells (%) | Reliability Probability (%) |
|----------------|--------------------------------------|-------------------------------------------|--------------------------------------|-----------------------------|
| chip1          | 0                                    | 264                                       | 91                                   | 71.1                        |
|                | 1                                    | 245                                       | 101                                  | 78.9                        |
|                | 2                                    | 231                                       | 117                                  | 91.4                        |
|                | 3                                    | 212                                       | 122                                  | 95.3                        |
|                | 4                                    | 186                                       | 128                                  | 100                         |
| chip2          | 0                                    | 247                                       | 94                                   | 73.4                        |
|                | 1                                    | 234                                       | 105                                  | 82.0                        |
|                | 2                                    | 212                                       | 118                                  | 92.2                        |
|                | 3                                    | 195                                       | 125                                  | 97.7                        |
|                | 4                                    | 167                                       | 128                                  | 100                         |
| chip3          | 0                                    | 252                                       | 86                                   | 67.2                        |
|                | 1                                    | 241                                       | 99                                   | 77.3                        |
|                | 2                                    | 223                                       | 116                                  | 90.6                        |
|                | 3                                    | 209                                       | 124                                  | 96.9                        |
|                | 4                                    | 194                                       | 128                                  | 100                         |

### Table II. NIST statistical test result

|              | NIST Pub 800-22, Randomness Test | P-value | NIST Pub 800-22, Randomness Test | P-value |
|--------------|----------------------------------|---------|----------------------------------|---------|
| Approximate entropy | 0.619142       Serial (1)       | 0.595473 |
| Linear complexity     | 0.976618       Serial (2)       | 0.738418 |
| Cumulative sum (1)    | 0.534146       Frequency         | 0.739918 |
| Cumulative sum (2)    | 0.759918       Longest runs      | 0.066882 |
| Overlapping template  | 0.572461       Block frequency   | 0.739918 |
| Non overlapping template | 0.978214   Runs                  | 0.739918 |
| Universal             | PASS             FFT                | 0.017912 |
| Random excursions variant | PASS           Matrix rank        | 0.312791 |
| Random excursions     | PASS             |         |                                  |