The UCONN-PUF: ESC 2011 PUF Entry

Nicholas Tuzzio, Xuehui Zhang, and Andrew Ferraiuolo
Advisor: Professor Mohammed Tehranipoor
Dept. of Electrical & Computer Engineering
University of Connecticut
{npt05001},{xhzhang},{andrew.ferraiuolo}@engr.uconn.edu

Abstract

We have designed a PUF that we have dubbed the "UCONN-PUF". This PUF takes a 16-bit challenge and produces an 8-bit response. The randomness and repeatability aspects of the PUF are obtained through the use of relative comparisons between adjacent ring oscillators. The reconfigurability aspect of the PUF is obtained by making the ring oscillators configurable. With an inter-die variation of 49% between PUF instantiations, an intra-die variation of about 33% in the same PUF instantiation, and a measurement noise of about 6% at room temperature, this PUF could be used for identification purposes in FPGA or ASIC designs.

I. Introduction

The goal of this competition was to design a physical unclonable function (PUF) on a Xilinx Spartan-6 field programmable gate array (FPGA) which produced an 8-bit output when presented with a 16-bit input. Two main criteria would be used to judge the quality of this PUF: the intra-Hamming distance, which is the number of output bits that are different when two inputs with a 1-bit difference are presented to the same PUF, and the inter-Hamming distance, which is the number of output bits that are different between two PUFs presented with the same input. Ideally, the intra-Hamming distance should be, on average, 50% of the output bits, which is also what the ideal inter-Hamming distance should be. A third main criteria that we imposed on our PUF was a measurement noise criteria- the number of output bits that are different between two measurements of the same PUF with the same input should be zero. Several other secondary criteria were mentioned in relation to the judging of the PUF; these include power, area, and delay of the PUF.

To achieve these goals, we chose to make novel improvements on an existing type of PUF rather than to develop a new type of PUF. The ring-oscillator PUF (RO-PUF) was first described in detail in 2007 [1]. The RO-PUF produces output bits by comparing the frequencies of two adjacent ring-oscillators. This idea was extended in 2009 [2] to make it so that the ring oscillators could be configured, using multiplexers, similar to how the Arbiter-PUF [3] operates.

In Section 2, we will describe the pre-existing theory and design decisions that influenced our design. In Section 3, we will describe the architecture of our UCONN-PUF. In Section 4, we will describe the performance and quality of the UCONN-PUF. In Section 5, we will compare the UCONN-PUF to other similar PUFs. Finally, in Section 6, we will summarize the results.

II. Theory

Ring oscillators (ROs) have been used to produce physically unclonable bitstreams for use in identification, authentication, and cryptographic key generation. This is done by creating an array of ROs which can be individually selected and measured using multiplexers and a counter. There are many ways to create a bitstream using n ROs; -the simplest way is to compare each RO to another RO that is physically adjacent to it, generating a 0 when the first RO is faster than the second, and generating a 1 when the first RO is slower than the second. We can generate $n/2$ uncorrelated bits by measuring and comparing n ROs. In [2], a method for creating configurable ROs (CROs) was described. This technique inserted multiplexers between each stage of inverters, so that for each stage a particular inverter could be selected. Essentially, a CRO with n stages is the same as 2^n ROs, albeit in a much smaller area. Figure 1 shows the general design of a CRO.

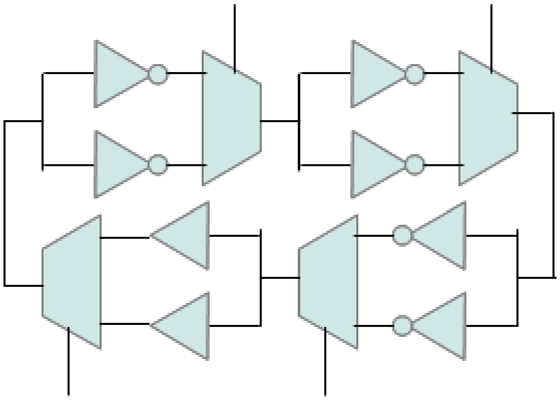

Fig. 1. An 8-stage configurable ring oscillator, designed to fit in 4 Xilinx Spartan-6 FPGA slices.

An RO makes a good foundation for a PUF because the frequency of an RO is dictated by uncontrollable silicon variation present in all ICs. These frequencies are relatively stable, and assuming that the RO is not left to oscillate for extended periods of time, will not change much over the operational lifespan of the IC. A CRO is a good improvement to the RO because it allows us to produce the same result as a normal RO with a large reduction in area. In this paper, we will describe how the useful characteristics of the CRO were used to produce a PUF which fit the criteria layed out for the competition.

III. Architecture

Our PUF uses an array of 16 CROs as its main source of silicon randomness. The outputs of the CROs are sent to a multiplexer array, the output of which is sent to a 32-bit counter. One CRO is selected, enabled, and measured at a time. The counter values of sequential CROs (1 and 2, 3 and 4, up to 15 and 16) are compared to generate 8 output bits. Each CRO has a 4-bit input which configures the CRO. These inputs are tied to the output of a 32-bit linear feedback shift register (LFSR). This LFSR is loaded with the initial 16-bit challenge at the beginning of the measurement process, and is allowed to run for 128 cycles before the measurement process starts. Each 4-bit slice of the 32-bit LFSR output is sent to two of the CROs; the first four bits are sent to CROs 1 and 2, and so on. Each pair of CROs that we are comparing is given the same configuration. The overall process through which output bits are generated is:

(*i*) Load the LFSR with the 16-bit challenge and run for 128 cycles.

(*ii*) Apply the LFSR output to the 16 CROs.

(*iii*) Measure each CRO's frequency and generate the 8 output bits by comparing the CRO frequencies.

CROs are allowed a 100-cycle "warm-up" time before the measurement begins, and they are measured for 50,000 cycles, giving us a high degree of measurement accuracy. Each CRO uses four adjacent FPGA slices; in each slice, two look-up tables (LUTs) and one multiplexer (MUX) are used. The array of 16 CROs uses a 3-wide by 32-high section of slices in the FPGA. Note that the CRO takes up a 3-by-2 section of slices because of the varied types of slices in the Spartan-6 FPGA; not all slices contain the MUX necessary for the CRO.

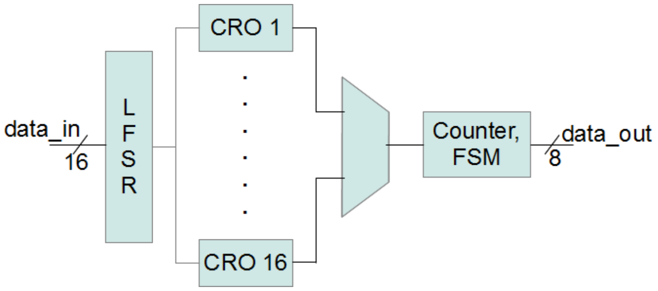

Fig. 2. Architecture of the UCONN-PUF. The LFSR configures the CROs, whose frequencies are compared to generate output bits.

When the UCONN-PUF is enabled, the 16-bit challenge is fed through the LFSR, which configures each pair of CROs. The CROs are measured and compared to produce an 8-bit output. The general architecture of the PUF is shown in Figure 2. In the next section, we will analyze the quality of the outputs of this PUF.

IV. Results

A. Intra-Hamming Analysis

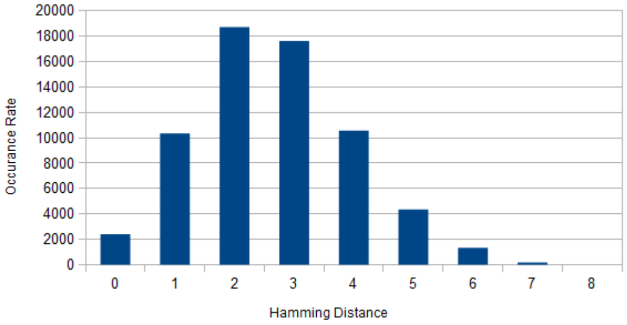

Fig. 3. Hamming distances between UCONN-PUF outputs from inputs with single-bit differences; the average value is 33%.

To perform intra-Hamming analysis of this PUF, we measure the number of output bits in the PUF which change when a single bit of the PUF's input is changed. A sequence of binary numbers where the difference between sequential entries is only a single bit is known as a "Gray code". We instantiated 8 UCONN-PUFs on a single FPGA and measured the 8-bit output from each for 8,192 sequential Gray code entries. The occurrence rate of each of the 9 possible Hamming distances across the 8,192 output changes over all 8 PUF instantiations is shown in Figure 3. The average number of different output bits when the input was changed by a single bit was approximately 2.66 bits out of 8, or about 33%. This falls short of the avalanche criteria set by the competition, which desired a 4 out of 8 or 50% change in the output bits when the input changed by a single bit, but the PUF is still able to vary its output significantly across different inputs.

B. Inter-Hamming Analysis

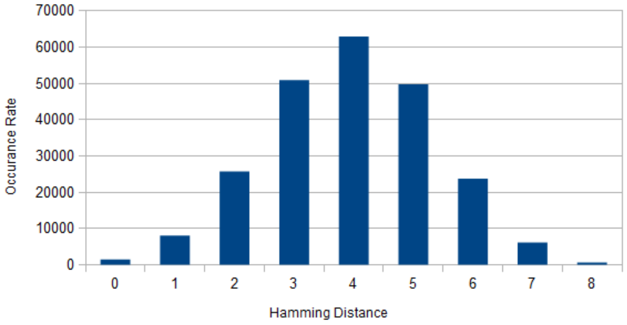

Fig. 4. Hamming distances between UCONN-PUF outputs from 8 different UCONN-PUFs with the same input; the average value is 49%.

To perform inter-Hamming analysis of this PUF, we compare the outputs of different instantiations of the same PUF when presented with the same input. We can use the same data set that we used to produce the intra-Hamming analysis to do this analysis. Each of the eight PUFs was presented with 8,192 different inputs. For each input, we examine the eight different 8-bit outputs that were produced, and calculate the Hamming distance distribution for those eight outputs. The occurrence rates of each of the 9 possible Hamming distances are shown in Figure 4. The average Hamming distance between the eight different outputs, when presented with the same input, was 3.94 bits out of 8, or about 49%. This is quite good- it means that different instantiations of the same PUF produce outputs that are very different from each other even when presented with the same input.

C. Measurement Noise Analysis

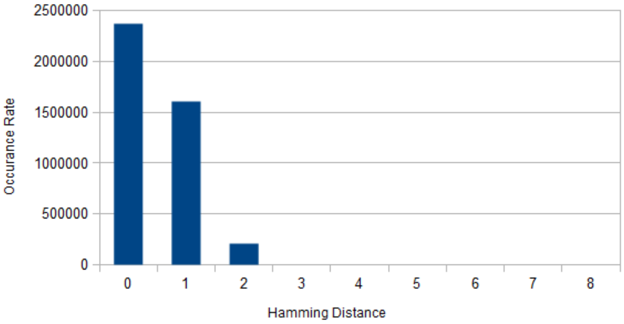

Fig. 5. Hamming distances between UCONN-PUF outputs when the input does not change; the average value is 6%.

The randomness of the UCONN-PUF is not the only important criteria. It is also important that a PUF be able to reproduce the outputs that it creates. To test this, we created eight instances of the same PUF on an FPGA. Each PUF was given the same input, and the output of the PUF with this input was measured 8,192 times. To determine the effect of measurement noise, we calculated the Hamming distance between each of the 1,024 PUF outputs for each of the 8 UCONN-PUFs. The occurrence rates of these Hamming distances are shown in Figure 5. The average Hamming distance between outputs of the same PUF with the same input was 0.48 bits out of 8, or about 6%. This is a relatively low amount of noise that results when the frequencies of the two CROs being measured are too close to each other to produce a reliable output. The fact that the average Hamming distance between measurements was less than 1 bit means that the same output is produced in most measurements, but the occasional 1-or-2-bit difference does occur.

D. Overall PUF Quality Analysis

The average intra-Hamming distance is about 33%, the average inter-Hamming distance is about 50%, and the average measurement noise is about 6%. The overall result of these numbers is the following summary: the difference between outputs of different PUFs is high, the difference between outputs of the same PUF is large but not ideal, and the stability of the PUF outputs is acceptable for purposes such as IC identification. In the next section, we will compare these characteristics to the characteristics of other similar PUFs to show the quality of our PUF.

V. PUF Comparisons

A. RO-PUF

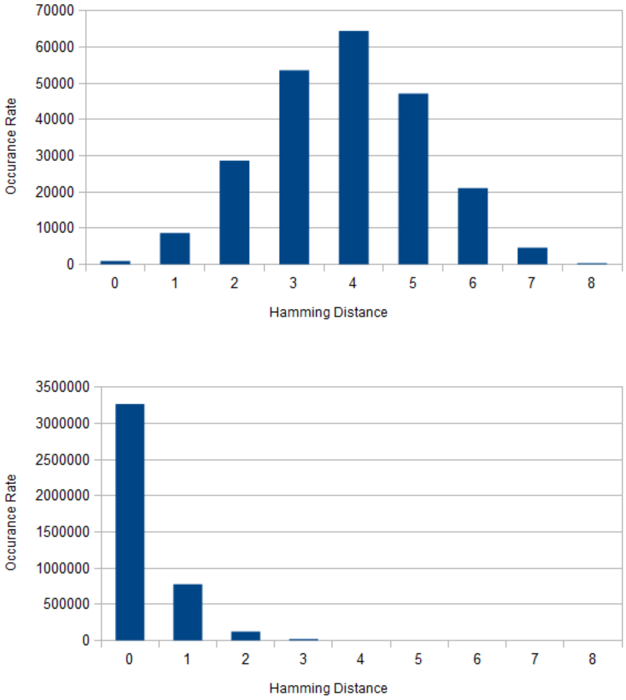

Fig. 6. (top) Hamming distances between outputs from different RO-PUFs; (bottom) Hamming distances between outputs from the same RO-PUF.

The first PUF that we will compare our PUF to is the traditional RO-PUF. The RO-PUF produces an 8-bit output by performing comparisons between 8 pairs of adjacent ring oscillators. Because there is no way to apply a challenge to these 8 pairs of ROs, we cannot analyze the intra-Hamming distance of the outputs of this PUF. However, we can still analyze the inter-Hamming distance and measurement noise, and compare them to the UCONN-PUF. The average inter-Hamming distance of the RO-PUF, as measured across 8 instantiations on the same FPGA, is 3.84 bits out of 8, or about 48%. This is very similar to the inter-Hamming distance of 50% that the UCONN-PUF was able to obtain. The average measurement noise of the RO-PUF, as found by measuring the same 8 RO-PUFs 1024 times each, was 0.26 bits out of 8, or about 3%. This is lower than that of the UCONN-PUF, but not significantly so.

B. CRO-PUF

The second PUF that we will compare our UCONN-PUF to is a more traditional CRO-PUF. The CRO-PUF is similar to the RO-PUF, but each CRO can receive a challenge. The

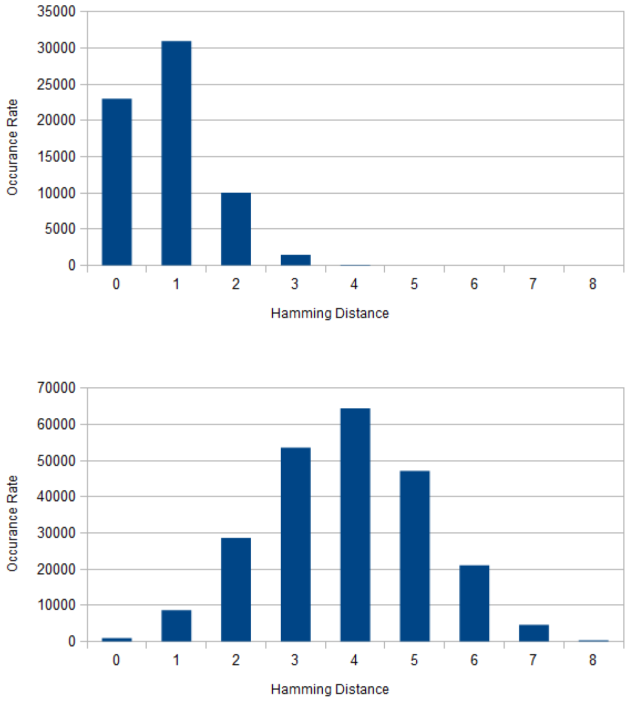

Fig. 7. (top) Hamming distances between outputs from CRO-PUFs with single-bit differences; (bottom) Hamming distances between outputs from CRO-PUFs with the same input.

main difference between the UCONN-PUF and CRO-PUF is the lack of an LFSR to modify the input bits. With the LFSR, a single-bit change at the input of the UCONN-PUF greatly changes the signals that control the CROs. Without the LFSR, a single-bit change at the input of the CRO-PUF only changes the challenge for a few of the CROs. It is obvious, then, that the intra-Hamming distance will be higher for the UCONN-PUF than the CRO-PUF. The numbers from 8,192 different inputs to the CRO-PUF bare this out: the average difference between two outputs with only a 1-bit difference in the input is 0.85 bits out of 8, or about 10.5%, as opposed to the 33% that the UCONN-PUF was able to achieve. The inter-Hamming distance is still very nearly 50%, and the measurement noise is identical because the CROs and measurement apparatus are exactly the same as in the UCONN-PUF.

C. Overall Comparisons

The UCONN-PUF is preferable to either the RO-PUF or CRO-PUF in the way that we are using it. The UCONN-PUF can create a large variety of random responses in the same amount of area that the RO-PUF requires. Additionally, the use of an LFSR to condition the inputs to the PUF greatly increases the intra-Hamming distance as opposed to not having the LFSR. The area differences between these PUFs are generally negligible; each PUF takes around 1600 to 1700 FPGA slices to implement 8 different instantiations on the FPGA. Additionally, each PUF takes the same amount of time to produce an output, and the general power usage- while large, due to the use of ring oscillators- should be relatively constant for each of these PUFs as well.

VI. Conclusion

The UCONN-PUF makes good use of the benefits of the RO-PUF and CRO-PUF while also adding some novelty to their use. By using configurable ring oscillators, we have created a challenge-response mechanism with the reliability of an RO-PUF. By conditioning the input to the CROs with an LFSR, we increase the intra-Hamming distance of the PUF outputs. The output from different PUFs is different in a very ideal way, and the output from each PUF is reliable enough to uniquely identify each PUF. The UCONN-PUF does not fully achieve the ”avalanche criteria” of a single-bit input change resulting in a 50% output change, but 33% may work well enough in some circumstances, especially when considering the other benefits of the UCONN-PUF.

References

[1] Suh, G.E.; Devadas, S.; , ”Physical Unclonable Functions for Device Authentication and Secret Key Generation,” Design Automation Conference, 2007. DAC ’07. 44th ACM/IEEE , vol., no., pp.9-14, 4-8 June 2007

[2] Maiti, A.; Schaumont, P.; , ”Improving the quality of a Physical Unclonable Function using configurable Ring Oscillators,” Field Programmable Logic and Applications, 2009. FPL 2009. International Conference on , vol., no., pp.703-707, Aug. 31 2009-Sept. 2 2009

[3] Gassend, B.; Clarke, D.; van Dijk, M.; Devadas, S.; ”Delay-Based Circuit Authentication and Applications,” Proc. of the 18th Annual ACM Symposium on Applied Computing, March 2003