Asynchronous Physical Unclonable Functions – ASYNCPUF

Julian Murphy

Centre for Secure Information Technologies,
Queens University Belfast,
Belfast, BT3 9DT
United Kingdom
j.p.murphy@qub.ac.uk

Abstract. Physically Unclonable Functions (PUFs) exploit the physical characteristics of silicon and provide an alternative to storing digital encryption keys in non-volatile memory. A PUF maps a unique set of digital inputs to a corresponding set of digital outputs. In this paper, the use of asynchronous logic and design techniques to implement PUFs is advocated for Asynchronous Physically Unclonable Functions (APUFs). A new method of using asynchronous rings to implement PUFs is described called ASYNCPUF which features inherent field programmability. It is both a novel and holistic PUF design compared to the existing state-of-the-art as it naturally addresses the two challenges facing PUFs to-date that prevent wide-spread adoption: robustness and entropy. Results of electrical simulation in a 90 nano-meter lithography process are presented and discussed.

Keywords: Cryptography, Security, Physically Unclonable Functions, PUFs, Asynchronous Physically Unclonable Functions, Clockless Physically Unclonable Functions.

1 Introduction

Many security mechanisms are based upon the concept of a secret. Classic cryptography applications contain a secret key as input to encryption algorithms in order to scramble and decipher data. While they are secure against attack at the algorithm and mathematical level, it is commonly known that digitally-stored secret keys can be attacked or cloned relatively easily. In security tokens, such as smartcards, keys are stored on-chip in non-volatile memory. While field-programmable gate arrays (FPGAs) instead store keys in off-chip memory. This is because FPGA technology cannot easily integrate non-volatile memory, and besides read latency issues, it only acts to further increase vulnerability to attack.

Physical Unclonable Functions (PUFs) offer an efficient alternative to storing digital keys in on or off-chip memory. They exploit the physical lithography manufacturing variations of silicon integrated circuits (ICs) - colloquially referred to as *chip variation* by those skilled in the art. A PUF maps a unique set of digital inputs, known as challenges, to a corresponding set of digital outputs, known as responses, for use in challenge-response security protocols. Almost every year since 2000 there has been a new PUF design proposed as highlighted in **Table 1**.

Table 1. Different types of PUF

Year	PUF Type
2000-2004	Device mismatch [9], One-way function [10], Physical Random Function [11], Arbiter PUF [12]
2005-2008	Coating PUF [13], Ring Oscillator PUF [2], SRAM PUF [14], Butterfly PUF [15]
2009-2011	Power distribution PUF [16], Glitch PUF [17], Mecca PUF [18]
2012	**ASYNCPUF (this paper)**

While a typical challenge-response identity authentication scenario is illustrated in **Fig. 1**. Here, a challenge is given to an IC to authenticate its identity via the on-chip PUF. If the received response is not equal to the known challenge (recorded during manufacturing) it is identified as fake and illegal.

Sadly, the unique benefits of silicon PUFs come with inherent stability design issues. In addition, in their basic configuration PUFs lack enough entropy to prevent modeling attacks [1]. However, it can be observed that PUFs are naturally asynchronous in nature. Insomuch as that they attempt to exploit non-synchronous effects such as metastability, propagation delay or binary signal glitches. Therefore it follows that asynchronous techniques, widely known for robustness and high-entropy (e.g. random number generation), may deliver much better PUF designs or provide an alternative to the existing state-of-the-art. The case and use of Asynchronous Physically Unclonable Functions (APUFs) is advocated and proposed for the first time here.

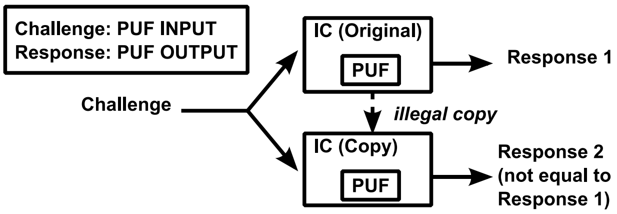

Fig. 1. Challenge-response authentication of chip identity using a PUF

In this paper, we present ASYNCPUF that uses asynchronous rings for robust operation and to replace inverter chain ring oscillators typically used in ring oscillator PUFs [2] (RO-PUFs) - it is both a novel and holistic PUF design compared to the existing state-of-the-art. It is fully digital and features inherent field programmability which naturally addresses the two challenges facing PUFs that prevents wide-spread adoption: robustness and entropy. Results of electrical simulation using a 90 nano-meter UMC lithography are discussed.

1.1 Contributions and Paper Organization

Our research, technical and scientific contributions are as follows:

- We propose Asynchronous Physically Unclonable Functions (APUFs) for the first time
- We advocate the use of asynchronous logic and techniques to implement PUFs.
- We propose ASYNCPUF that is inherently field-programmable to address robustness and entropy challenges. It uses asynchronous rings to replace inverter ring oscillators (IROs) used in ring oscillator PUFs [2] (RO-PUFs).

The remainder of the paper is organized as follows: Section 2 gives an overview of asynchronous logic. Section 3 discusses asynchronous rings. Section 4 describes ASYNCPUF. Section 5 presents results from electrical simulation. Section 6 draws conclusions.

2 Asynchronous Logic

The design of synchronous digital circuitry is based upon the discretization of time, where a synchronous system changes from one state to the next at transitions of a system clock. The state is held in a set of registers and the next state outputs are derived from Boolean logic acting on the old state and present inputs. The next state is copied through the registers on every rising and falling edge of a global clock signal. Hence, the system exhibits deterministic behavior as specified as long as certain timing constraints on the inputs are met.

Asynchronous designs do not follow this regime. In general there is no global clock to govern the timing of state changes. Subsystems and components exchange information at mutually negotiated times. Therefore, naturally, certain parts of a design are always quiescent when they are not in use and hardware runs as faster as: computational dependencies, input rate and the lithography device switching times.

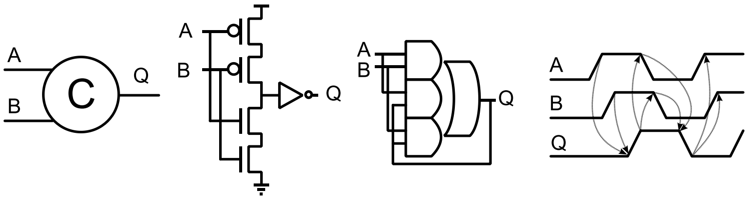

Fig. 2. The Muller C-element

As a field it is historically seen as niche due to the profound understanding of concurrency, hardware, and semiconductors it takes to implement functionally-correct designs. However, interest in the field has grown linearly in recent years in terms of applications as the fringes of Moore's Law have been reached and Cyber security has become main-stream.

A plethora of design paradigms and techniques are known in literature. These range from high performance transistor level pipelines for processor design [3] and application to physical security [4]. The common denominator in all of which is the hysteresis capable Muller-C element [5] shown in **Fig. 2**. Both inputs must be equal to set or reset its output - otherwise it holds its original state.

3 Asynchronous Rings

One of the most widely-used structures that use Muller-C elements are asynchronous rings (ARs) [7], which are purposely used here to implement ASYNCPUF. That is, as an alternative to inverter ring oscillators (IROs) in RO-PUFs for increased PUF stability and entropy.

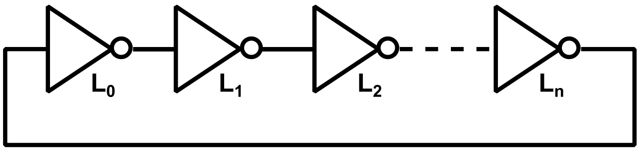

Fig. 3. L stage inverter ring oscillator

To illustrate how ARs operate an IRO structure is shown in **Fig. 3**. Here, L inverter stages are connected to form a ring. The oscillation time is the propagation delay of one logical transition all around the ring.

While an AR structure of L stages is shown in **Fig. 4** and corresponds to the control path of a micro-pipeline [7]. Each stage is composed of a Muller C-element and an inverter, where for stage i: F_i is the forward input, R_i is the reverse input, and C_i is the output. In words, the forward input value is written to the output if the forward and reverse input values are different. Otherwise the previous output is maintained.

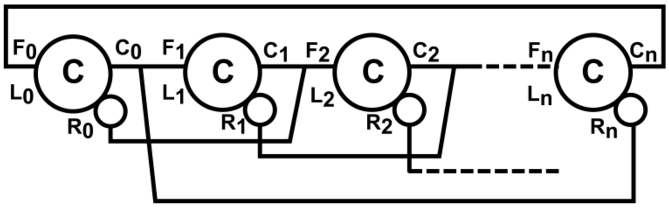

Fig. 4. Asynchronous Ring

3.1 Bubbles and Tokens

With reference to **Fig. 4** the *bubbles and tokens* concept is as follows:

- Stage i contains a bubble if its output C_i is equal to the output of the previous stage: $C_i = C_{i-1}$.
- Stage i contains a token if its output C_i is different than the output of the previous stage C_{i-1}: $C_i \neq C_{i-1}$.

Hence, for a 5 stage AR an initial state could be the token-bubble tuple:

$$\{Bubble_0, Token_0, Token_1, Bubble_1, Bubble_2\} \tag{1}$$

This would correspond to the initial binary state:

$$\{S_0, S_1, S_2, S_3, S_4\} = \{1,0,1,1,1\} \tag{2}$$

Moreover, as each stage i has a value of token or bubble determined by its output C_i and the output of the previous stage C_{i-1} the mapping from (1) to (2) should be intuitive: $Token\ _0 = \{\mathrm{C}_0, \mathrm{C}_1\} = \{1,0\}$, $Token\ _1 = \{\mathrm{C}_1, \mathrm{C}_2\} = \{0,1,$ $Bubble_1 = \{\mathrm{C}_2, \mathrm{C}_3\} = \{1,1\}$ etc.

Since it is possible to configure an AR with respect to bubbles and tokens, as explained above, it can be easily understood that they are naturally field-programmable and will increase the available entropy in an AR based PUF design such as ASYNCPUF presented in this paper.

3.2 Token and Bubble Propagation

Therefore, from the token and bubbles concept, a token propagates from the stage i to the stage $i+1$, if, and only if, the next stage $i+1$ contains a bubble as shown in **Fig. 5**. In the same way, a bubble propagates from the stage i+1 to the previous stage i, if and only if, the previous stage i contains a token. Hence, ARs will have an oscillatory behavior if the following conditions hold:

- $L \geq 3$ and $L = N_t + N_b$.
- $N_b > 1$, where N_b is the number of bubbles.
- N_t is a positive even number of tokens.

The oscillation depends on the stage timing parameters determined by process variability (i.e. higher entropy) and the ratio N_t/N_b. It should be understood, while it is possible to maintain high frequencies in ARs, frequency decreases linearly with the number of stages in IROs. That is: different AR ring configurations (i.e. the number of tokens and bubbles) will result in different frequencies for the same ring lengths.

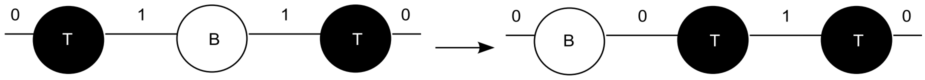

Fig. 5. Token and bubble movement

3.3 Noise

Both ARs and IROs exhibit thermal noise (known as jitter in the time-domain and phase noise in the frequency domain) such that the propagation delay will resemble a Gaussian distribution. **Fig. 6.** Effect of illustrates the effect of jitter on an IRO in a 130nm SPICE transient noise analysis simulation using thermal noise with a bandwidth of 100KHz to 10GHz - a clear 71 pico-second variance is observable.

Where ARs and IROs differ is through how jitter accumulates. An IRO's period is defined by two loops of one token around the ring, and accumulates jitter from the number of crossed stages. But, in an AR, several tokens propagate in the ring simultaneously indicating the period is governed by the time between successive tokens. As such, each token crossing a stage experiences a variation in its propagation delay due to the jitter contribution of that particular stage. This is contrary to the IRO effect of jitter accumulation. This naturally provides improved robustness against noise instabilities caused by jitter in PUF designs, that is, by use of ARs instead of IROs.

In addition to Gaussian jitter, deterministic jitter occurs from non-random variations in propagation delays due to external global influences. The main difference is again in that in an AR several events propagate simultaneously, so deterministic jitter affects each event in the same way rather than the whole structure. This again leads to increased robustness in ARs versus IROs, and a more stable PUF design if ARs are used instead of IROs.

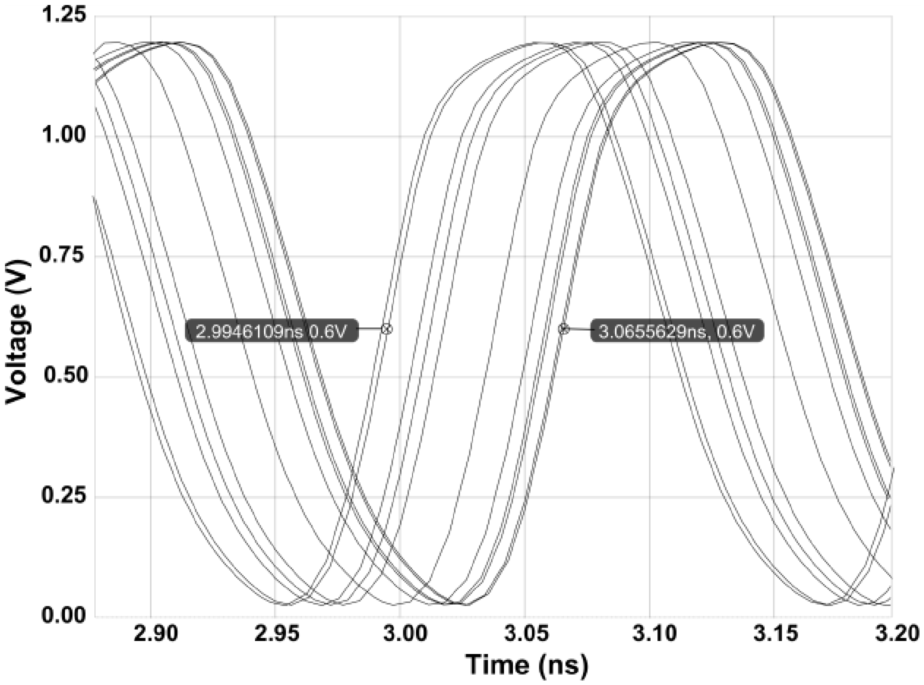

Fig. 6. Effect of jitter

4 AsyncPUF

We present in this section how to build AsyncPUF using asynchronous rings by replacing IROs in RO-PUFs.

A 1-bit RO-PUF is composed of 2 identically laid-out RO's, $R0_1$ and $R0_2$ with frequencies f_1 to f_2. They are selected using a pair of multiplexers that takes a bit of the PUF challenge as the select bit. Due to process variation, f_1 and f_2 will differ generating one response bit, R, of the PUF from comparison of the two frequencies measured by their respective counters. When enabled, R will be 1 if $f_1 > f_2$ otherwise 0, hence producing a single bit of a PUF response signature. The exemplary design in **Fig. 7** produces a single PUF bit - n-bit PUF configurations are built by cascading these 1-bit RO-PUF structures.

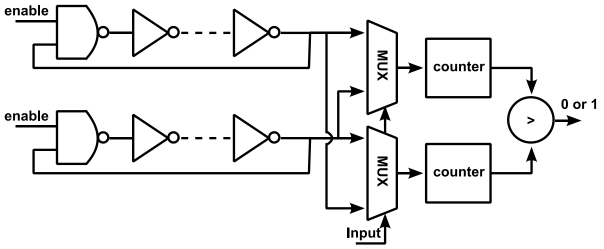

Fig. 7. Ring-oscillator based PUF design

Since IRO frequencies are closely matched, environmental effects can cause the oscillators to switch their outputs, for increasing temperature and/or decreasing voltage resulting in incorrect responses. It can be also observed large arrays of ring oscillators can cause a change in local chip temperature. These temperature stability issues are depicted on the left in **Fig. 8**. The ideal scenario is that the frequency difference should be sufficient to ensure consistent operation over temperature and voltage as shown on the right in **Fig. 8**. The approach to this problem in PUFs to-date has been to use error correcting methods, which are expensive in terms of silicon area and add additional complexity to the challenge-response protocol. The other disadvantage of RO-PUFs is that they can be easily modeled to break the underlying security [1] to enable cloning.

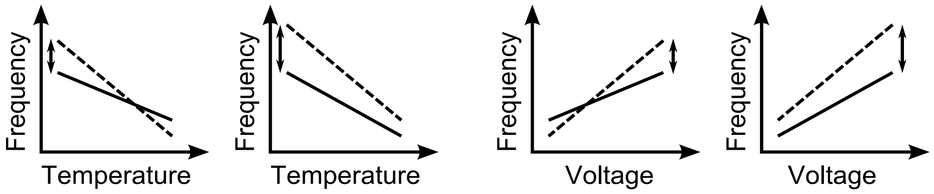

Fig. 8. Temperature and voltage effects on RO-PUFs

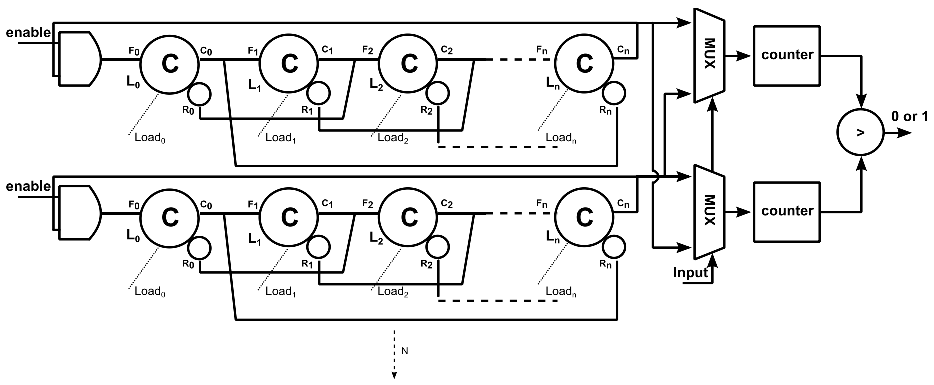

Fig. 9. ASYNCPUF

ASYNCPUF is an AR based PUF, as shown in **Fig. 9**, and gives the opportunity to address the above issues as well as noise. By configuring N_t and N_b, that is, by purposely controlling L, the number of stages, and their initial value by setting or resetting the Muller-C elements through the load inputs. By determining the configuration of the ARs with the maximum frequencies differences maximum reliability can be attained. This inherent configurable permits extremely low error rates by tending towards the ideal scenario. A further opportunity is to calibrate the ASYNCPUF configurability according to different operating conditions. For example, the entire operating range of temperature and voltage could be divided into regions and have different AR load bit patterns.

AR PUFs offer the opportunity to not only increase robustness through tolerance to environmental effects, but the fact they can be re-configured increases entropy to address modeling attacks. As discussed, ARs can be easily configured to change their frequency by controlling N_t and N_b. Thus varying N_t and the load bit patterns in-field will result in whole new PUF designs, therefore thwart modelling as no two PUFs are the same. Another alternative is to allocate different values randomly during manufacture and store in on-chip non-volatile memory.

For correct operation, the AR run-time has to be low enough so that the counters do not overflow. Hence, care has to be taken to ensure the counters are matched to the estimated frequencies. It is worth noting also, other methods are perfectly plausible to convert the varying AR frequencies to a binary bit, rather than using a pure multiplexer approach. How RO-PUFs are cascaded for n-bit PUFs may also differ e.g. AR reuse.

5 Results

Experiments were performed using Monte Carlo SPICE analysis on the highest accuracy setting with a 65nm UMC lithography process and thermal noise with a bandwidth of 100KHz to 10GHz. Firstly, ARs were characterized to quantify how their oscillation frequency is affected by intra-die and inter-die process variation i.e. to understand their response to the lithography effects PUFs exploit. Simulations were conducted for a 6-stage AR using a 20 nano-second window and 1000 iterations for the two types of process variation (die-to-die and within-die). They took approximately 8 hours to complete on a high-end multi-core Linux server under the Cadence Design Framework. **Fig. 10** shows the results from each of the 1000 simulations.

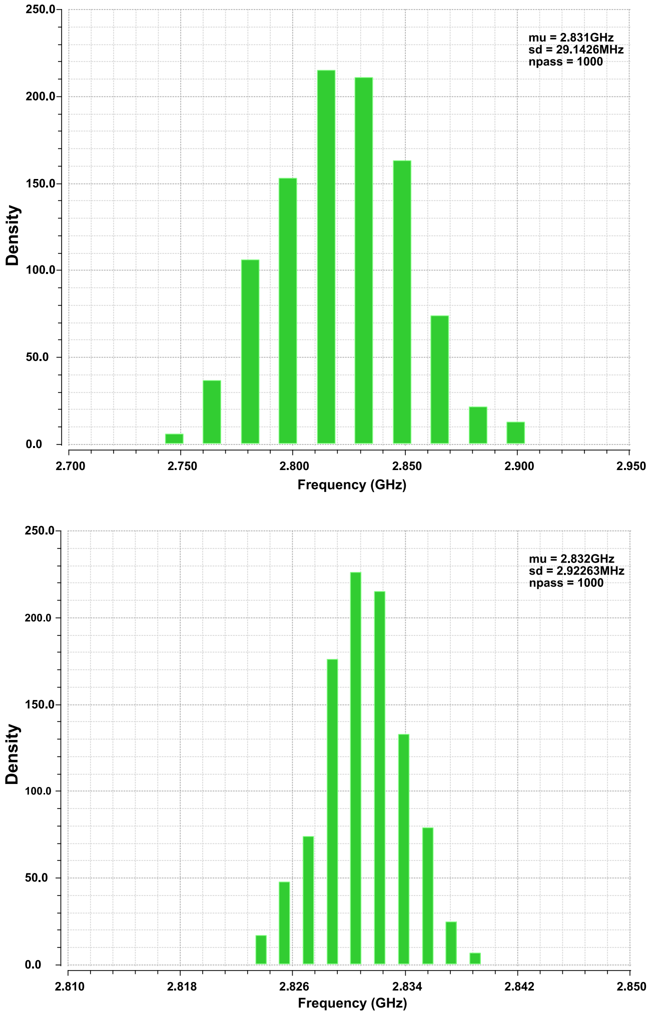

Fig. 11. Die-to-die (top) and within-die (bottom) variation

The ARs exhibit clear frequency deviations confirming their suitability for use as PUFs. For die-to-die variation an average frequency of 2.83GHz is obtained and a standard deviation of 29.14MHz, which indicates a die-to-die variation of 1.03%. And for within-die variation an average frequency of 2.83GHz is obtained and a standard deviation of 2.92MHz, which indicates a within-die variation of 0.10%. Clearly, the variation in AR frequency is greater between silicon wafers than on the same wafer for this particular lithography process; while both results exhibit a bell-curve Gaussian distribution.

Next 20 ASYNCPUFs of length 6, 12 and 18 each able to generate 32-bits of a response (i.e. 64 rings) were constructed, which was found in the setup phase to allow practical SPICE simulation. Note, using four different AR configurations a 128-bit ASYNCPUF output can be generated, which highlights the trade-offs that are possible with ASYNCPUF due to its inherent field-programmability.

This time both die-to-die and within-die process variation SPICE simulation switches were activated together for analogous electrical simulation of 20 ASYNCPUF silicon chips. Matlab was used to parse and process the simulation data obtained and to generate random input challenges. Using two well-known PUF metrics, uniqueness and reliability (defined below), ASYNCPUF was evaluated. Both uniqueness and reliability results were captured at supply voltages between 0.4 V and 1.1 V, and temperatures ranging from -30C to 100C. Note, these result graphs were produced by Matlab rather than exported directly from Cadence as in **Fig. 11**. And to fit within the paper length, the presented results highlight the effect of temperature effects only. This is also because temperature affects PUFs silicon chips more than regulated voltage that can be viewed as a constant variable.

- Uniqueness is a measure of how easily an individual PUF can be differentiated; and quantifies the hamming distance between the responses of different ICs implementing the same PUF design that have been challenged with the same input. It is characterized by the probability density distribution (PDF) of the hamming distances, where PUFs with PDF curves centered at half the number of response bits and tall are more easily identifiable (unique) than PUFs with flatter curves.
- Reliability is a measure of how easily a given PUF can reproduce the same output response for the same input challenge. This is measured by the bits that remain unchanged under varying environmental conditions with the same input challenge. The PDF representing hamming distance of the response characterizes reliability of the same PUF subject to different environmental conditions i.e. changes in temperature and supply voltage. PUFs with PDF curves centered at 0 and tall are more stable than PUFs with flatter curves.

It was observed with increasing length of the ring, uniqueness is consistent, with a slight tendency for a stronger PDF the longer the length shown on the left in **Fig. 11.** This result was consistent across all ASYNCPUF lengths initialized with arbitrary token patterns that satisfy the requirements in Section 2.

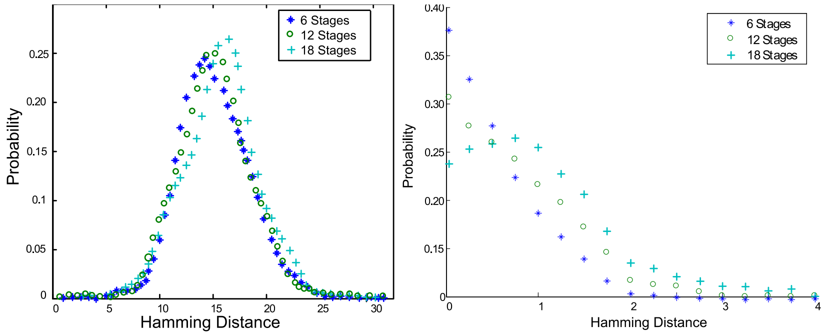

Fig. 11. Uniqueness and reliability of AsyncPUF with respect to temperature

Fig. 11 on the right shows the effect of the stage length for AsyncPUF for reliability. It was observed for AsyncPUF that 6 stages are most stable followed by 12 and 18 stages. Therefore it can be concluded that shorter stages leads to better stability. This can be exploited for area efficient PUF implementations.

6 Conclusions

We have proposed using asynchronous logic to address the inherent issues with physically unclonable functions. We have presented and described a method of using asynchronous rings to implement a novel APUF architecture design, AsyncPUF to enable increased robustness and entropy. We presented Monte Carlo Spice analysis results of uniqueness and reliability. The results represent as close as possible to physical silicon chip results. It is common practice to rely on statistical SPICE transistor level simulation based on foundry process information before actual physical implementation. Due to the requirements of asynchronous circuits to be correct by construction (hysteresis from feedback) FPGAs were not used.

As cryptographic primitives, PUFs have several useful applications in security but are most frequently used for device authentication (i.e. **Fig. 1**). However, a new level of robustness and entropy for APUFs allows increased resistance to modeling attacks and makes it feasible for real-life efficient design. And also enables new applications: secret key generation, Intellectual Property (IP) protection, and to prevent product counterfeiting – or even to use in a software and hardware scenario.

Our future work is to consider application of asynchronous techniques to further PUF technologies and tape-out of a silicon chip. For instance, it would be possible to build PUF designs using elements from asynchronous elastic controllers [4] or eager monotonic logic [5]. Or alternative structures could be used instead of C-elements to implement AsyncPUF ring stages that are widely published in literature e.g. GasP.

References

1. U. R. ührmair, F. Sehnke, J. S ölter and G. Dror “Modeling attacks on physical unclonable functions”, Proceedings of 17th ACM Conference on Computer and Communications Security, pages 237-249, 2010.
2. G. E. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation", In Proceedings of the 44th annual Design Automation Conference, DAC '07, pages 9-14, New York, NY, USA, 2007.
3. I. Sutherland, S. Fairbanks, "GasP: a minimal FIFO control", Seventh International Symposium on Asynchronous Circuits and Systems, Page 46-53, 2001.
4. J. Murphy and A. Yakovlev, "An Alternating Spacer AES Crypto-processor", Proceedings of the 32nd European Solid-State Circuits Conference, September 2006, Pages 126 - 129.
5. D.E. Muller and W.S. Bartky, "A Theory of Asynchronous Circuits", Proc. Int'l Symp. Theory of Switching, Part 1, Harvard Univ. Press, 1959, pp. 204–243.
6. T. E. Williams and M. A. Horowitz, "A Zero-Overhead Self-Timed 160-ns 54-b CMOS Divider", IEEE Journal of Solid-State Circuits, Vol 26 (11), Pages 1651-1661, Nov 1991.
7. I. E. Sutherland, “Micropipelines”, Communications of ACM, Vol. 32, Issue 6, pages 720-738, 1998.
8. J. C. Ebergen, S. Fairbanks and I.E. Sutherland, “Predicting performance of micropipelines using charlie diagrams”, Proceedings of fourth international conference on Asynchronous Circuits and Systems, Pages 238-246, 1998.
9. K. Lofstrom, W. Daasch, and D. Taylor, "Ic identication circuit using device mismatch", Digest of Technical Papers, IEEE International Conference in Solid-State Circuits (ISSCC), pages 372-373, 2000.
10. R. S. Pappu, B. Recht, J. Taylor, and N. Gershenfeld, "Physical one-way functions", Science, Vol. 297, 2026-2030, 2002.
11. B. Gassend, D. Clarke, M. van Dijk, and S. Devadas, "Silicon physical random functions", In Proceedings of the 9th ACM conference on Computer and communications security (CCS), pages 148-160, New York, USA, 2002.
12. D. Lim, J. Lee, B. Gassend, G. Suh, M. van Dijk, and S. Devadas, "Extracting secret keys from integrated circuits", IEEE Transactions on Very Large Scale Integration Systems, Vol. 13, Issue 10, Pages 1200-1205, 2005.
13. P. Tuyls, G.-J. Schrijen, B. Skoric, J. van Geloven, N. Verhaegh, and R. Wolters, "Read-proof hardware from protective coatings", In Cryptographic Hardware and Embedded Systems Workshop (CHES), volume 4249 of LNCS, pages 369-383. Springer, October 2006.
14. J. Guajardo, S. S. Kumar, G.-J. Schrijen, and P. Tuyls, "FPGA intrinsic pufs and their use for ip protection", In Proceedings of the 9th international workshop on Cryptographic Hardware and Embedded Systems (CHES), 2007, pages 63-80, Berlin, Heidelberg, 2007.
15. S. Kumar, J. Guajardo, R. Maes, G.-J. Schrijen, and P. Tuyls, "Extended abstract: The butterfly puf protecting ip on every fpga", IEEE International Workshop on Hardware-Oriented Security and Trust (HOST), pages 67-70, 2008.
16. R. Helinski, D. Acharyya, and J. Plusquellic, "A physical unclonable function defined using power distribution system equivalent resistance variations", In Proceedings of the 46th Annual Design Automation Conference (DAC), pages 676-681, New York, USA, 2009.
17. D. Suzuki and K. Shimizu, "The glitch puf: a new delay-puf architecture exploiting glitch shapes", In Proceedings of the 12th international conference on Cryptographic hardware and embedded systems (CHES), pages 366-382, Berlin, Heidel-berg, 2010.
18. A. R. Krishna, S. Narasimhan, X. Wang, and X. Wang. Mecca, "A robust low-overhead puf using embedded memory array", In Proceedings of the 13th international conference on Cryptographic hardware and embedded systems (CHES), pages 407-420, Berlin, Heidelberg, 2011.