An On-Chip High Resolution Measurement Structure for Measuring Path Delays in an Arbiter PUF

M. Martin and J. Plusquellic
ECE Dept. University of New Mexico

ABSTRACT

A physical unclonable function (PUF) is a integrated circuit hardware primitive that is designed to leverage naturally occurring variations to produce a random bitstring. The arbiter (ARB) PUF is one of the first to be described in the literature. It derives its entropy from variations that occur in the delays of identically configured logic paths. The ARB PUF uses a phase comparator to decide which path of a pair is faster under a given challenge, and generates a 0 or 1 as a response indicator bit. Unfortunately, the ARB PUF is not reliable, requiring error correction in cases where the sequence of response bits (the bitstring) needs to be reproduced. In this paper, we describe a test structure, called a time-to-digital converter (TDC) that is capable of measuring the actual delays of the paths. This type of 'soft' information can be used to improve the reliability of the ARB PUF. Data obtained from a set of chips fabricated in IBM's 90 nm technology, and collected across 9 temperature-voltage corners is used to demonstrate its effectiveness. The bitstrings are evaluated using statistical tests which measure randomness, uniqueness and reliability.

1.INTRODUCTION

Physical Unclonable Functions (PUFs) are poised to represent the next generation of hardware security primitives for integrated circuits. The chip-specific identifiers produced by PUFs can serve several applications including chip ID, authentication, metering and encryption. PUFs measure and digitize the natural variations that occur in path delays, leakage current, SRAM power-up patterns, etc. to produce a long sequence of random bits, i.e., a bitstring. Most of the applications that use these bitstrings require that they be 1) unique among the chip population, 2) random in sequence and 3) reproducible across adverse environmental conditions.

The Arbiter PUF was introduced in [1] and is designed to leverage delay variations that occur in identically configured paths. In order to avoid biases, the paths that are timed are implemented in a specialized test structure which allows the gate-level components that define the paths to be 'swapped'. A digital challenge controls the specific configuration of the swapped and unswapped gate-level components using 2-to-1 multiplexors. A phase detector is inserted at the endpoints of the test structure to determine the relative delay of the two paths-under-test. The relationship is binary, i.e., either the first path is faster than the second or vise versa, and therefore can be represented as a 0 or 1 response bit. The sequences of response bits produced by a sequence of challenges defines the bitstring.

This type of binary response evaluation circuit does not contain any hint as to how different the delays for a given pair of paths are. In this paper, we investigate a supporting test structure for obtaining 'soft' information from the ARB PUF, that is designed to measure the delay of the actual paths. The TDC produces a digital value in the range of 0 to 120 that is proportional to this delay. Additional benefits of the TDC over, for example, ring-oscillator variants of the ARB PUF, include; 1) the ability to self-compensate for variations in ARB PUF delays that are introduced by changes in temperature and voltage, 2) the ability to provide very fast data collection times, e.g., single-shot measurements times are less than 20 ns/sample, and 3) the ability to tune resolution down to sub-gate-delay levels.

The ARB PUF and TDC are evaluated in 20 copies of a custom ASIC fabricated in a 90 nm technology across 9 temperature-voltage (**TV**) corners, i.e., at all combinations of the temperatures -40°C, 25°C and 85°C and voltages 1.08 V, 1.2 V and 1.32 V. (NOTE TO REVIEWER: We will present the results of 60 chips in the final version of this paper, if accepted). Statistical tests are applied to the bitstrings to evaluate their randomness, bias, uniqueness and stability. A thresholding technique is proposed that uses the TDC value to screen path comparisons where the delay difference is small. This technique is shown to allow the ARB PUF to achieve 100% reproducibility of the bitstring without error correction.

2.BACKGROUND

Random bit strings form the basis for encryption, identification, authentication and feature activation in hardware security. The introduction of the PUF as a mechanism to generate random bit strings began in [2], although their use for chip identifiers began a couple years earlier [3]. Since their introduction, there have been many proposed architectures that are promising for PUF implementations, including those that leverage variations in transistor threshold voltages [3], in speckle patterns [2], in delay chains and ROs [4-7], in SRAMs [8], in metal resistance [9][10], and many others.

Previous work on ARB PUFs is summarized as follows. In [11] the authors propose a circuit architecture for a key card authentication scheme by using a delay-based ARB PUF circuit. In [12], a Feed Forward version of the ARB PUF is proposed that is designed to increase the difficulty of model building attacks. In [13], the authors combined multiple Feed Forward ARB circuits as a means of increasing the uniqueness and resilience to model building attacks. The authors of [14] propose an XOR-based ARB PUF that uses multiple individual ARB PUFs in parallel, each with k stages, and configured so that the same challenge can be applied to each copy. In [15], testing techniques for PUFS are described, where the authors propose methods to evaluate PUF predictability, sensitivity to noise, and strength against reverse engineering. They also propose a non-destructive method to reverse engineer a PUF, and an interleaved PUF structure to defeat the proposed reverse engineering technique. In [5], a lightweight secure PUF is introduced which uses a set of ARB PUFs in parallel, but have their individual outputs XORed to produce a multi-bit response. The author in [16] propose a low overhead Arbiter PUF-enabled RFID scheme. In [17], the authors usee an FPGA based ARB PUF circuit to demonstrate a reduction of spatial gradients of the response by using a common centroid layout. The authors of [18] propose a reconfigurability technique for FPGA-based PUF designs that reduce susceptibility to various types of reverse engineering attacks. In [19], the use of an FPGA based ARB PUF is proposed in which different routing techniques are used in the determination of delay variation. A design is proposed in [20] of the ARB PUF that is completely described in VHDL and doesn't rely on the usage of manual routing. The authors of [21] propose a procedure for obtaining PUF performance, and define indicators to quantitatively evaluate the performance of a PUF. In [22], a symmetric switch structure is proposed with an FPGA-based ARB PUF which helps mitigate metastability problems. The authors of [23] propose a method of attack based on numerical modeling that creates an algorithm that has a near identical behavior to that of the original PUF. A 4-XOR Arbiter PUFs is proposed in [24] as well as pattern matching techniques that reveal only a minimal amount of public data thereby discouraging modeling attacks. The authors in [25] propose an ARB circuit that contains an array of PMOS capacitors at both output nodes. In this method, the delay-time difference of the two paths is used to determine the response. In [26], the authors propose a low-power/small area FPGA-based RFID ARB PUF design. The authors of [27] propose a set of performance metrics that are based on delay statistics. The proposed technique exposes a weakness to randomness for the ARB PUF used. The authors in [28] propose a machine learning attack on an ARB PUF. It uses the additive delay model ARB PUF to train the machine learning algorithm which then is able to predict CRP with a fairly high accuracy.

This paper for the first time investigates the use of an on-chip TDC to obtain 'soft' information from the ARB PUF, and uses this information to improve its reliability across industrial temperature and voltage corners.

3.CHIP DESIGN

The architecture of the proposed PUF consists of two basic components; an Arbiter (ARB) PUF, which implements the paths to be tested, and a Time-to-Digital Converter (TDC), which provides high resolution timing measurements of the path delays in the ARB PUF. The following presents the implementation details of these two components.

3.1 ARB PUF

Fig. 1 shows the layout of the ARB PUF and TDC in the 90 nm test chip architecture. The ARB is shown along the top as a sequence of 16 series-connected segments of 8 elements each. Fig. 2 gives a schematic level representation of the elements within the ARB. Each of the 128 elements consist of a flip-flop (FF) and two copies of a 2-to-1 MUX. The FF is scan-connected with the others (not shown) and can be configured with a challenge bit. The challenge bit determines whether two paths (labeled P_A and P_B) propagate signals straight through the 2-to-1 MUXes (when 0) or cross-over with P_A propagating through the bottom MUX and P_B through the top (when 1). The input to the ARB PUF is shown on the left side of Fig. 2. and connects to both of the P_A and P_B paths. A signal transition is introduced into the ARB PUF by asserting or de-asserting this input signal.

A unique feature of the ARB PUF proposed in this paper is the introduction of a set of 'tap points' (several are labeled in Fig. 1). The first tap point is connected directly to the input of the ARB. The remaining tap points are implemented by fanning out at specific points along the paths P_A and P_B to a pair of buffers. For example, the second tap point connects to P_A and P_B at a point that is 32 elements from the input side of the ARB PUF. The remaining 6 tap points connect at points further downstream after sequences of 16 additional elements. The outputs of the buffers at each tap point route to the inputs of the TDC as shown on the left side of Fig. 3.

3.2 Time-to-Digital Converter (TDC)

The TDC is designed to measure the relative delay between two input signals which are provided by a pair of tap points on the ARB PUF. The relative delay is digitized by the TDC using a pulse-shrinking mechanism (described below). The digital code is 'scanned out' of the TDC for post-processing.

The TDC is implemented as two components, labeled Path Select/Pulse Gen Unit and Pulse Shrinking Delay Chain in the layout of Fig. 1 and in the schematic of Fig. 3. Scan FFs in Path Select/Pulse Gen Unit, labeled 'Sel A' and 'Sel B', drive the inputs of two 8-to-1 MUXes, which, in turn, select a specific pairing of tap point inputs, one from group 'A' and one from group 'B'. The outputs of the 8-to-1 MUXes route to the inputs of a XNOR gate, which serves to generate a negative pulse for the Pulse Shrinking Delay Chain on the right (see annotation in Fig. 3). The arrival of an edge on one of the tap points propagates to the XNOR and generates the 1-to-0 transition of this negative pulse, and an edge (arriving later) on the second tap point generates the 0-to-1 transition of the pulse. Specific configurations of tap point pairs that provide sufficient skew between the two edges from paths in the ARB PUF are described in the next section.

The TDC is designed to 'pulse shrink' the negative output pulse from the XNOR as it propagates down a current-starved inverter chain. As the pulse moves down the inverter chain, it activates a corresponding set of set-reset latches to record the passage of the pulse, where activation is defined as storing a '1'. A thermometer code, i.e., a sequence of '1's followed by a sequence of '0's, represents the digitized delay of a path within the ARB PUF.

A call-out of a current-starved inverter used in the delay chain is shown on the far right side of Fig. 3. The NFET transistor with input labeled 'Calx' implements the current-starving mechanism. The Calx inputs are driven by two analog control voltages, labeled 'Cal0' and 'Cal1'. The current-starved inputs of all the even numbered inverters (numbered starting with 0) are connected to Cal0 while the inputs of the odd numbered inverters are connected to Cal1. This type of configuration allows independent control over the propagation speed of the two transitions associated with the negative pulse. For example, increasing the voltage on Cal1 toward the supply voltage allows the odd numbered inverters to switch more quickly when the first transition, i.e., the 1-to-0 *input transition*, propagates to their inputs. Note that the 1-to-0 *input transition* creates 0-to-1 transitions on the inputs of the odd numbered inverters in the chain, which activates the pull-down paths of these inverters. With Cal0 fixed at a specific voltage, larger

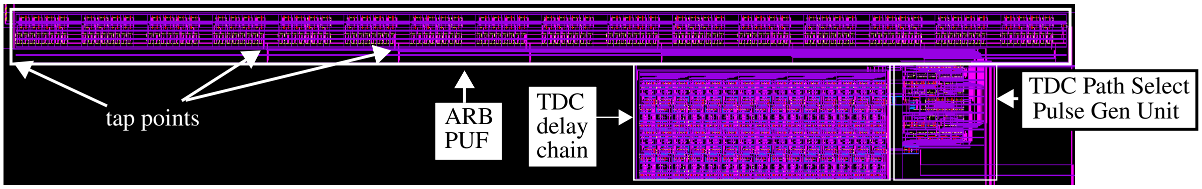


Fig. 1. Layout of ARB PUF and TDC.

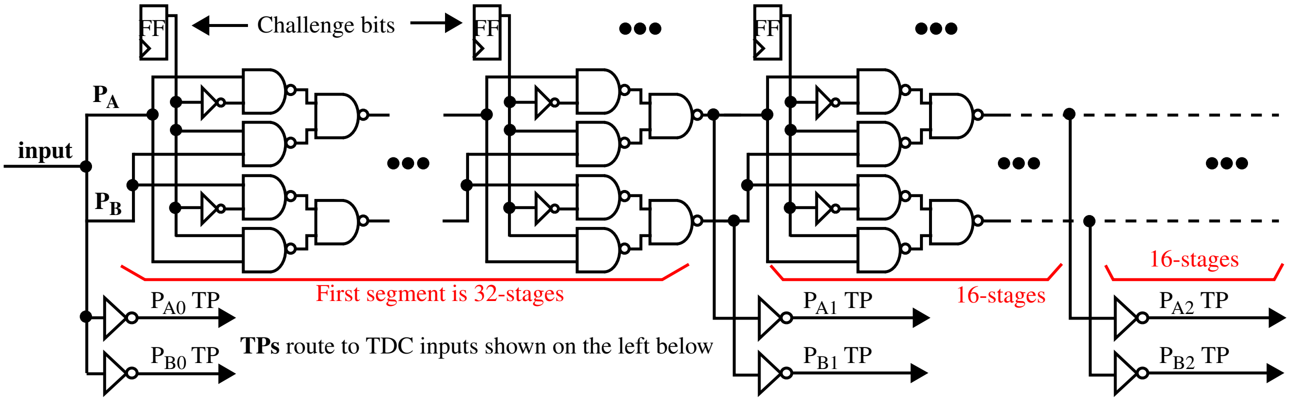


Fig. 2. Arbiter PUF configuration showing Tap Points (TPs)

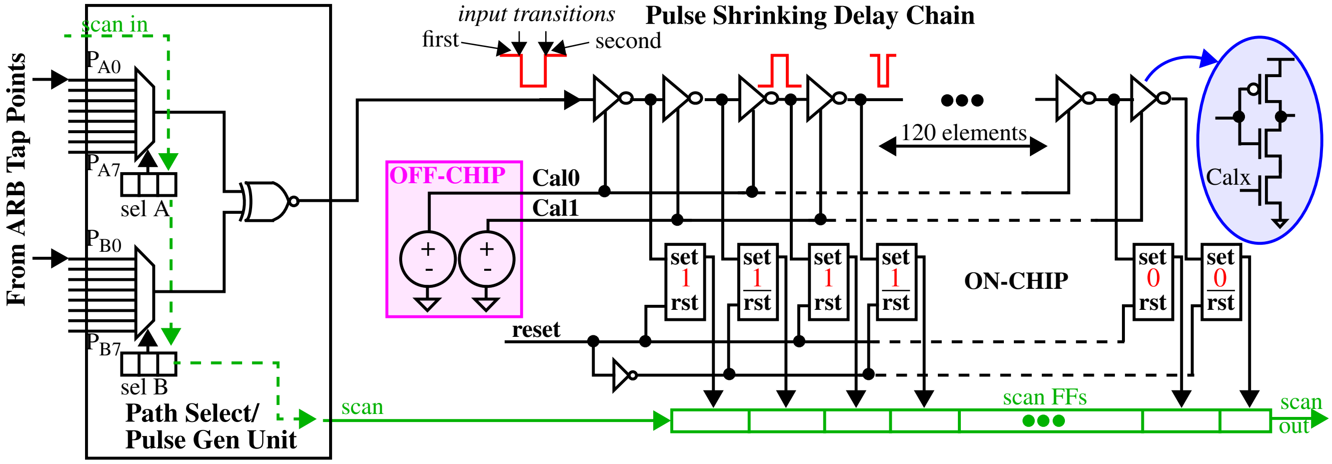


Fig. 3. Time-To-Digital Converter with Path Select/Pulse Generator front-end for interfacing with the Arbiter PUF TP.

assigned Cal1 voltages allows the pulse to ‘survive’ longer in the delay chain because the first edge propagates more quickly. The speed of trailing 0-to-1 input transition does not change with Cal0 fixed, and therefore it takes longer for this edge to catch-up to the leading transition. Eventually it does (assuming Cal0 and Cal1 are set such that the trailing edge is faster) and the pulse disappears. All latches up to the point where the pulse disappears store a ‘1’, while those beyond this point store ‘0’. The state of the latches can then be transferred to the scan FFs shown along the bottom of Fig. 3 for scan-out and analysis.

The pulse-shrinking behavior of the TDC allows very high timing resolution, i.e., 10’s picoseconds, in measurements of the width of the input pulse assuming the Cal0 and Cal1 voltages are fixed and stable. The timing resolution of the TDC is related to how far the pulse propagates along the delay chain, where pulse propagations to points near the end of the delay chain provide the highest resolutions. It is possible, however, for the pulse to propagate off the end of the TDC, a condition referred to as overflow, which obviously must be avoided. By choosing the proper Calx voltages, the overflow condition can be prevented while simultaneously allowing for high timing resolutions.

In our experiments, both of these voltages are controlled using off-chip power supplies. This allowed us to explore the parameters of this new architecture so that a functional and fully integrated version can be implemented properly on the next test chip. The off-chip power supplies will be replaced with an on-chip resistor ladder network, and a controller will be used to select the proper Calx voltages from this resistor ladder network. As discussed in the following sections, the primary function of the controller will be to carry out a calibration process that is designed to prevent overflow. From our experiments, maximizing the timing resolution is of benefit but is not a requirement for the TDC to be effective in

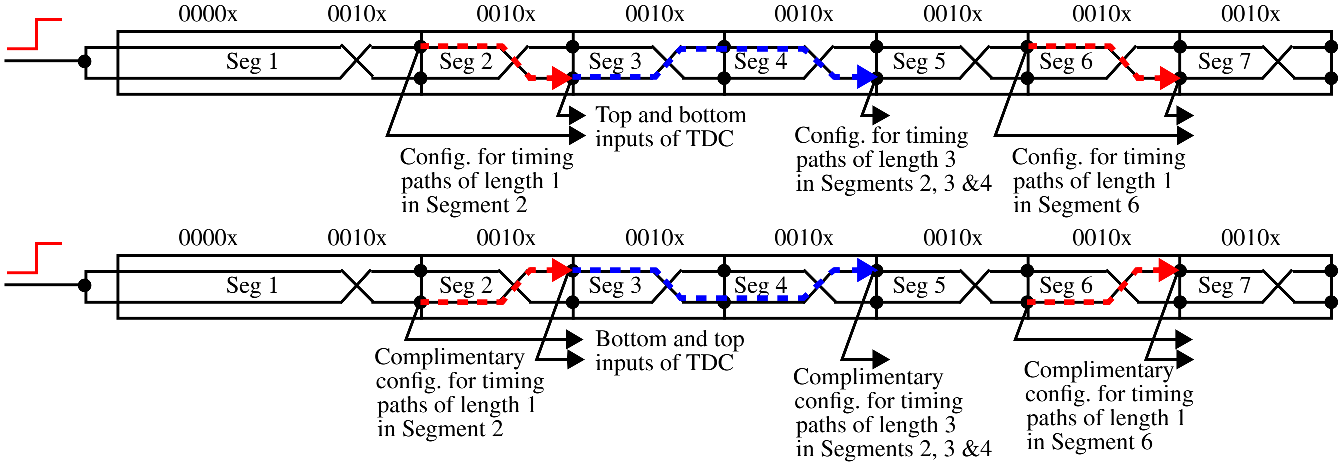


Fig. 4. Examples showing configurations for timing path segments of length 1 and 3 in the ARB PUF.

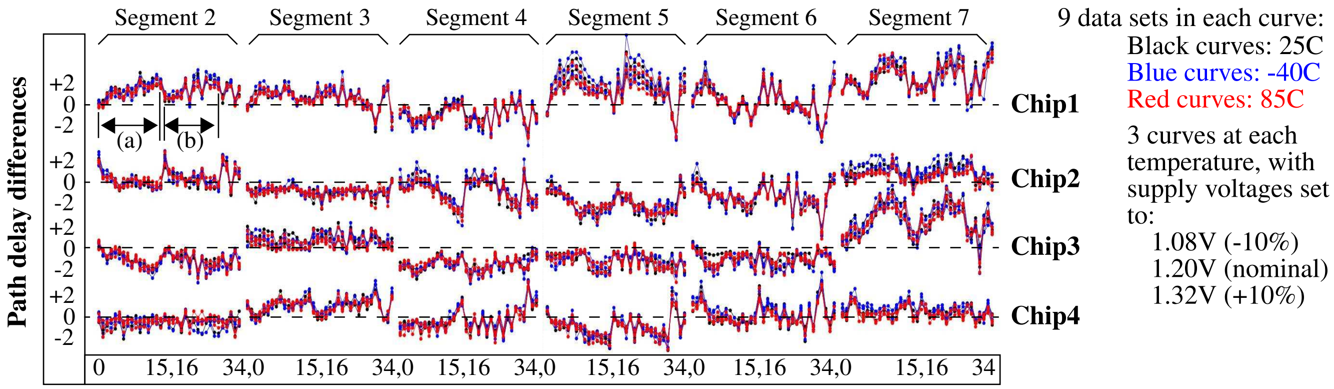


Fig. 5. TDC-timed path delay differences from 4 chips for paths of length 1 across Segments 2 through 7 (in reference to Fig. 4). The 1st 16 points in each segment (labeled 0-15 on the x-axis) are from tests that introduce 1 x-over, points 16 through 29 represent 3 x-overs, point 30 represent 7 x-overs, point 31 represent 15 x-overs, and the remaining points (>32) represent a random number of x-overs.

improving reliability of the ARB PUF.

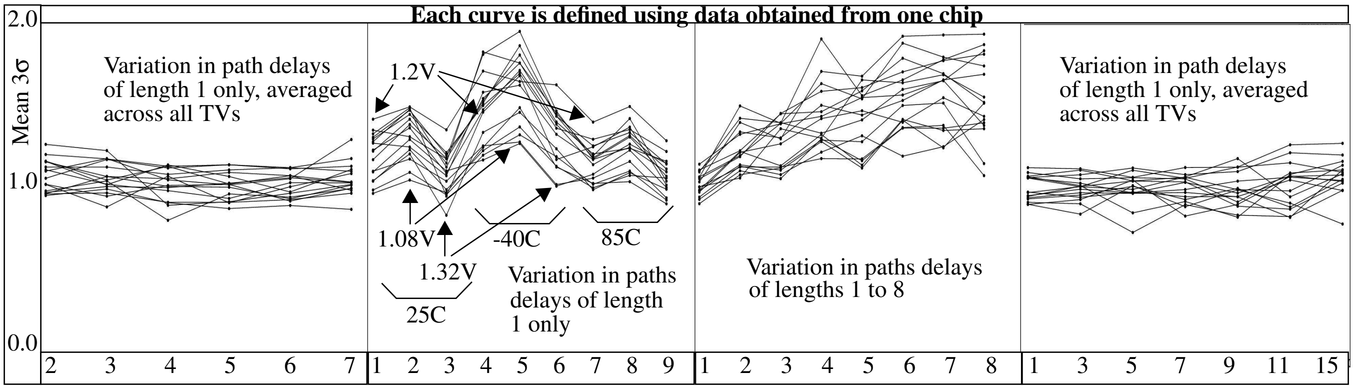
In preliminary experiments, we discovered that it is not necessary to have independent control over the leading and trailing edges of the pulse. The data presented in this paper is obtained by fixing Cal0 to the supply voltage. Therefore, only Cal1 is tuned in our experiments. The Cal1 voltage required to meet the above constraints varied as a function of the ambient temperature and voltage conditions but was largely self-compensating. We provide more details on this issue after we describe how the ARB PUF and TDC are used together to collect delay measurements in the next section.

The overhead of the proposed ARB + TDC combination is as follows. The ARB PUF with 128 elements occupies an area of approx. 525 um x 25 um (13k um^2) while the TDC occupies an area of 176 um x 60 um (10k um^2). As we show in the following sections, the size of the ARB PUF is sufficient to generate several hundred delays, each of which has at least one constituent element in a given ARB delay path that is completely independent of the others. Simple modifications can be made to increase the number of independent delays to a 1000 or more with only a moderate increase in area.

3.3 Usage Scenario of the ARB PUF and TDC

As covered above, the addition of the tap points provides a unique opportunity to measure delays along *segments* of the ARB PUF (traditional approaches do not allow entropy to be extracted from the constituent elements of the ARB's delay chains). The diagram in Fig. 4 illustrates how the tap points can be used to measure delays along path segments. The elongated rectangles represent an abstraction of the ARB PUF in which the 128 elements are partitioned into seven segments labeled 1 to 7. The first segment contains 32 elements while the remaining segments contain 16 elements. The top portion shows two configurations for measuring paths of length 1 (in segments 2 and 7) and one configuration for measuring paths of length 3 (across segments 2, 3 and 4)[1]. In the examples shown, the number of switches configured with a '1' is odd, which ensures that the TDC times a single path. For example, the signal propagating along the top path to the tap point at the beginning of segment 2 crosses-over to the bottom path before reaching the second tap point at the beginning of segment 3. The path that is timed is highlighted in the figure. We use

[1] A path of length 1 is defined as a 16-element segment within the ARB PUF.



(a) Segment # **(b)** TV # **(c)** Path Length **(d)** Number of x-overs

Fig. 6. Measurement noise analysis under different path selection criteria in a set of 16 chips. All graphs plot mean 3 σ along the y-axis. a) Noise as a function of segment number in reference to Fig. 4, b) noise as a function of temperature and voltage (TV), c) noise as a function of path length (also Fig. 4), and d) noise as a function of the number of switch cross-overs (x-overs).

the term x-over to refer to switches that are configured to cause the path to cross-over from top to bottom or bottom to top.

In order to eliminate any bias that exists in the TDC measurement structure, in particular along the paths from the tap points through the 8-to-1-MUXes in the TDC and to the XNOR gate, complementary paths of those shown in the top portion of Fig. 4 are also tested and the two measurements are subtracted. The bottom portion of Fig. 4 shows the complementary configuration of the three tap point combinations given in the top portion.

The waveforms shown in Fig. 5 depict TDC measurement results for paths of length 1 (enumerated along the x-axis) for four chips. The waveforms are offset along the y-axis to facilitate comparisons between the waveforms. The path delays plotted along the y-axis are given in units of TDC bits. As indicated above, the values plotted are actually the difference in the number of TDC ‘1’ bits measured from two complementary paths. The plotted differences are computed using the average number TDC bits from a set of 11 measurements carried out on each path and its complement. There are a total of 211 data points for each chip, which represent all the paths of length 1 that we tested using 40 different challenges. The curves for each of the nine temperature-voltage (TV) corners are superimposed to illustrate the ‘noise’ introduced by environmental variations. From the graphs, it is clear that TV variations are smaller in many cases than the delay variations introduced by process variations.

The first 16 data points in each 16-element segment of the waveform show the results from a set of canonical challenges. The canonical challenges introduce exactly 1 x-over, similar to those shown in Fig. 4 for paths of length 1. These points are labeled as (a) for Chip1 in segment 1 in the upper left portion of the figure. The data points are ordered so that the position of the x-over element in each test is adjacent to x-over elements that were tested under previous (and subsequent) challenges. This arrangement allows the magnitude of delay variations introduced by swapping a single pair of elements to be observed incrementally along each of the waveform segments. The data points labeled (b) are arranged similarly except the consecutive tests introduce 3 x-overs. Although delay variations within these groups are relatively small, variations across groups and especially across segments are much larger. In the experimental results section, we show that good statistical results can be obtained from these TDC measured delays.

As indicated above in reference to Fig. 4, other tap point configurations allow the measurement of delays from paths that traverse multiple segments. However, the statistical averaging effect of delays along longer path segments makes it difficult to measure distinguishing characteristics in them at sufficient resolution, and therefore, their usefulness for PUF bit generation is limited. Therefore, only paths of length 1 are used to generate the bitstrings analyzed in this paper. Plots similar to those shown in Fig. 5 are described in the supplementary material for completeness.

3.4 Measurement and TV Noise Analysis

Measurement noise and noise introduced by varying temperature and voltage conditions work to reduce the reliability of the ARB PUF. Reliability is defined here as the ability of the ARB PUF to exactly reproduce the same bitstring during ‘regeneration’ experiments. The bitstrings produced at 25^{o}C and at 1.20 V (nominal supply voltage) are referred to as the reference (or **enrollment**) bitstrings[1], while bitstrings produced at the remaining 8 temperature-voltage corners are referred to as **regeneration** bitstrings. As indicated in the Introduction, the chips used in our experiments are tested at all combinations of temperatures -40^{o}C, 25^{o}C and 85^{o}C and voltages 1.08 V, 1.2 V and 1.32 V. In this section, we evaluate these noise levels independently.

The plots in Fig. 6 depict noise levels as ‘average 3 σ values’ on the y-axis[2]. Each of the plots labeled Fig. 6(a) through 6(b) depict sixteen different waveforms, one for each of the sixteen chips considered in this analysis. The waveforms in Fig. 6(a) give the average 3 σ’s of all measurements within each of the six ARB PUF segments (in reference to Fig. 4 for paths of length 1). As indicated earlier, we collect eleven TDC samples for each tested path. The average 3 σ’s in this plot represent the measurement noise in these repeated samples. From the figure, the average ‘hovers’

[1] Enrollment defines the bitstring generation process that is carried out initially.

[2] Three σ is a statistical measure that bounds 99.73% of the population.

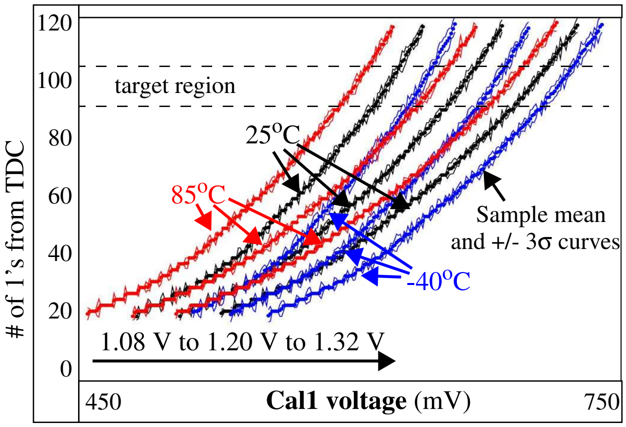


Fig. 7. TDC Cal1 vs. thermometer code curves across 9 temperature/voltage corners on one chip.

around 1 for all chips and is independent of the segment number given along the x-axis. Therefore, the timing uncertainty remains constant across the six segments.

In contrast, noise introduced by temperature-voltage variations, as shown in Fig. 6(b), is not constant. Again, only paths of length 1 are included in the analysis. The y-axis in this case plots the average 3 σ's of all path measurements at each of the nine TV corners, as indicated by the labels in the figure. In general, noise levels are larger at -40°C (center portion of plot) than at 25°C and 85°C. Also, noise increases as supply voltage is lowered, as shown by the y-magnitudes of the points within each temperature group.

Fig 6(c) also shows that noise increases as the length of the path increases, which is expected given that longer signal paths have larger amounts of jitter and are exposed over longer periods to power supply noise variations. Similar to Fig. 6(a), noise levels as a function of the number of x-overs appears to remain relatively constant as shown in Fig. 6(d).

3.5 TDC Calibration

As indicated earlier, Cal1 needs to be 'tuned' to compensate for changes in the TDC behavior introduced by TV variations. The curves in Fig. 7 illustrate the behavior of the TDC in one of the chips at the nine TV corners investigated in this work. The x-axis sweeps the Cal1 voltage over a range of 450 to 750 mV. The y-axis plots the number of 1's read from the TDC under each of these Cal1 settings. The individual curves are labeled to indicate the TV corner under which the data was collected. The mean values as well as the 3 σ upper and lower limits are superimposed. Although the shapes of the curves change to some degree, the main effect of TV variations is reflected as the shift in the curves along the x-axis.

In order to ensure that the TDC is able to produce values in the region labeled 'target region' at each of these TV corners, it is necessary to 'tune' the Cal1 voltage. Note that shifts due to voltage variations will be automatically calibrated for by an on-chip resistor ladder network. This is true because the resistor ladder network will be connected to the power grid on the chip and will track changes in the power supply voltage automatically. Therefore, the primary issue is dealing with shifts introduced by temperature variations.

A calibration procedure is proposed to tune Cal1 so that overflow does not occur and the TDC produces values in the target region under temperature variations. The objective of the calibration process is to select a voltage produced by the on-chip resistor ladder network and apply this voltage to the Cal1 signal of the TDC. This can be accomplished by choosing a tap point combination and iteratively testing that path and adjusting the voltage until the number of 1's produced is in the target region. The process can be implemented by an on-chip state machine and using a binary search process (to make it fast). In our experiments, we emulate the binary search process in LABVIEW software and use an external power supply to emulate the on-chip resistor ladder network.

4.BitString Generation

Bitstrings are generated by comparing the 211 TDC data points obtained from each chip in all combinations, which yields bitstrings of length 21,155 bits. As noted below, a thresholding technique is used to discard those comparions which are vulnerable to producing 'bit flips' under TV variations.

4.1 Thresholding Technique

The 'soft' information provided by the TDC can be used to avoid those path delay pairings whose difference is likely to result in a bit flip during regeneration. A thresholding technique is proposed that accomplishes this goal. During enrollment, comparisons of delay differences which are smaller than the threshold are discarded. The comparisons that are discarded are recorded in public data so that they are avoided during the regeneration process. Based on our preliminary analysis, we found that a threshold of approx. 5, in units of the number of 1's produced by the TDC, eliminates all bit flips that occur in the bitstring generation of our chips.

5.EXPERIMENTAL RESULTS

In this section, we evaluate the several important statistical properties of the bitstrings including randomness, uniqueness and probability of bit flips, e.g., failures to regenerate the bitstring under different environmental conditions.

The size of the bitstring after thresholding is 1,955 bits on average. The inter-chip HD requires that the bitstrings for all chips are the same size. We accommodate this requirement by finding the chip with the shortest bitstring and reducing the size of the other bitstrings to this length. The smallest bitstring is 1,503 bits. The HDs from the bitstrings of the 20 chips are computed under all combinations. Fig. 8 gives the inter-chip hamming distance (HD) distribution along with superimposed Guassian curve bit to illustrate the level of conformance of the distribution to a normal distribution. The average inter-chip HD is given as 50.1%, which is very close to the ideal of 50.0%. With thresholding, all bit flips are avoided and therefore the intra-chip HD is 0%. The true intra-chip HD is given as 11.1% to illustrate the fraction of the population that is unstable. Only nine of the NIST tests are applicable to bitstrings of size 1,503. The bitstrings pass all of the tests except several Non-overlapping Template tests and the ApproxmiateEntropy Test. However, the tests that are failed only fail by 2 chips beyond the required 18 for a population of this size.

6.CONCLUSIONS

7.Supplementary Material

(NOTE TO REVIEWERS: We will add supplementary material here as promised above in the final version of the paper if accepted).

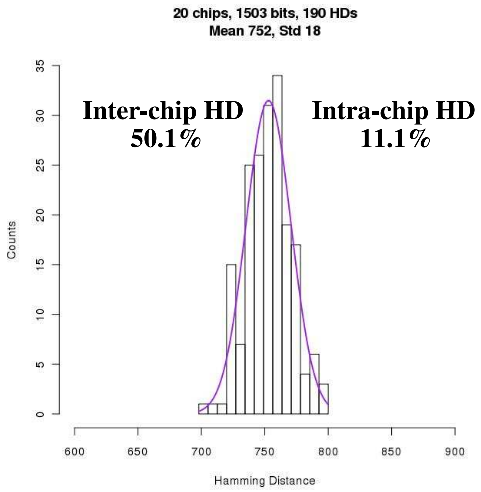


Fig. 8. Inter-Chip Hamming Distance Distribution for 20 Chips.

8.REFERENCES

[1] B. Gassend and D. E. Clarke and M. van Dijk, S. Devadas, “Silicon Physical Unknown Functions,” *Proc. of Conference on Computer and Communications Security*, 2002, 148-160.

[2] R. S. Pappu, *et al.*, “Physical One-Way Functions,” *Science*, 297(6), 2002, pp. 2026-2030.

[3] K. Lofstrom, *et al.*, “IC Identification Circuits using Device Mismatch,” *SSCC*, 2000, pp. 372-373.

[4] B. Gassend, *et al.*, “Controlled Physical Random Functions,” *Conference on Computer Security Applications*, 2002.

[5] M. Majzoobi, *et al.*, “Lightweight Secure PUFs”, *ICCAD*, 2008.

[6] G. Qu and C. Yin, “Temperature-Aware Cooperative Ring Oscillator PUF”, *HOST,* 2009, pp. 36-42.

[7] A. Maiti and P.Schaumont, “Improving the Quality of a Physical Unclonable Function using Configurable Ring Oscillators”, *FPLA*, 2009. pp. 703-707.

[8] J. Guajardo, *et al.*, “Physical Unclonable Functions and Public Key Crypto for FPGA IP Protection,” *FPLA*, 2007, 189-195.

[9] R. Helinski, *et al.*, “Physical Unclonable Function Defined Using Power Distribution System Equivalent Resistance Variations”, *DAC*, 2009, pp. 676-681.

[10] J. Ju, R. Chakraborty, R. Rad, J. Plusquellic, “Bit String Analysis of Physical Unclonable Functions based on Resistance Variations in Metals and Transistors”, *HOST*, 2012, pp. 13-20.

[11] Gassend, Blaise, et al. "Delay-based Circuit Authentication and Applications." Proceedings of the 2003 ACM symposium on Applied computing. ACM, 2003.

[12] Gassend, Blaise, et al. "Identification and Authentication of Integrated Circuits." Concurrency and Computation: Practice and Experience 16.11 (2004): 1077-1098.

[13] Daihyun Lim, Lee, J.W., Gassend, B., Suh, G.E., van Dijk, M., Devadas, S., , "Extracting Secret Keys from Integrated Circuits," Very Large Scale Integration (VLSI) Systems, IEEE Transactions on , vol.13, no.10, pp.1200-1205, Oct. 2005.

[14] G.E. Suh and S. Devadas, “Physical Unclonable Functions for Device Authentication and Secret Key Generation”, Proceedings of the 44th annual Design Automation Conference, page 14, 2007.

[15] M. Majzoobi, F. Koushanfar, and M. Potkonjak, “Testing Techniques for Hardware Security,” in IEEE Intl Test Conf., 2008, pp. 1-10.

[16] Devadas, S.; Suh, E.; Paral, S.; Sowell, R.; Ziola, T.; Khandelwal, V.; , "Design and Implementation of PUF-Based ‘Unclonable’ RFID ICs for Anti-Counterfeiting and Security Applications," RFID, 2008 IEEE International Conference on , vol., no., pp.58-64, 16-17 April 2008.

[17] [8] H. Yu, P. Leong, H. Kinkelmann, L. Moller, and M. Glesner,”Towards a Unique FPGA-based Identification Circuit using Process Variations,” in IEEE Intl Conf. on Field Programmable Logic and Applications, 2009, pp. 397-402.

[18] [9] M. Majzoobi, F. Koushanfar, and M. Potkonjak, “Techniquesfor Design and Implementation of Secure Reconfigurable PUFs,” ACM Trans. on Reconfigurable Technology and Systems, vol. 2, no. 1, pp. 1-33, 2009.

[19] [10] Morozov, Sergey, Abhranil Maiti, and Patrick Schaumont. "An Analysis of Delay Based PUF Implementations on FPGA," Reconfigurable Computing: Architectures, Tools and Applications (2010): 382-387.

[20] [11] Jason H. Anderson, "A PUF Design for Secure FPGA-based Embedded Systems," Proc. of the 2010 Asia and South Pacific Design Automation Conference, IEEE Press, 2010.

[21] [12] Hori, Yohei, et al. "Quantitative and Statistical Performance Evaluation of Arbiter Physical Unclonable Functions on FPGAs," Reconfigurable Computing and FPGAs (ReConFig), 2010.

[22] [13] Majzoobi, Mehrdad, Farinaz Koushanfar, and Srinivas Devadas. "FPGA PUF using Programmable Delay Lines." Information Forensics and Security (WIFS), 2010.

[23] [14] U. Ruhrmair, F. Sehnke, J. Solter, G. Dror, S. Devadas and J. Schmidhuber, "Modeling Attacks on Physical Unclonable Functions," Proc. ACM CCS, Oct. 2010.

[24] [15] Z. Paral, S. Devadas, "Reliable and Efficient PUF-based Key Generation using Pattern Matching," Hardware-Oriented Security and Trust (HOST), pp.128-133, 2011.

[25] [16] Fruhashi, Kota, et al., "The Arbiter-PUF with High Uniqueness Utilizing Novel Arbiter Circuit with Delay-Time Measurement," Circuits and Systems (ISCAS), 2011.

[26] [17] Soybali, Mehmet, Berna Ors, and Gokay Saldamli, "Implementation of a PUF Circuit on a FPGA," New Technologies, Mobility and Security (NTMS), 2011.

[27] [18] Jouini, Zouha CHERIF, J. Danger, and Lilian Bossuet. "Performance Evaluation of Physically Unclonable Function by Delay Statistics." New Circuits and Systems Conference (NEWCAS), 2011.

[28] [19] Hospodar, Gabriel, Roel Maes, and Ingrid Verbauwhede. "Machine Learning Attacks on 65nm Arbiter PUFs: Accurate Modeling Poses Strict Bounds on Usability," Information Forensics and Security (WIFS), 2012.