Stability Analysis of a Physical Unclonable Function based on Metal Resistance Variations

J. Ju, C. Lamech* and J. Plusquellic
ECE Dept., University of New Mexico
*Intel Corp.

Abstract -- ***Keying material for encryption is stored as digital bitstrings in non-volatile memory on FPGAs and ASICs in current technologies. However, secrets stored this way are not secure against a determined adversary, who can use probing attacks to steal the secret. Physical unclonable functions (PUFs) have emerged as an alternative. PUFs leverage random manufacturing variations as the source of entropy for generating random bitstrings, and incorporate an on-chip infrastructure for measuring and digitizing the corresponding variations in key electrical parameters, such as delay or voltage. PUFs are designed to reproduce a bitstring on demand and therefore eliminate the need for on-chip storage. In this paper, we evaluate the randomness, uniqueness and stability characteristics of a PUF based on metal wire resistance variations in a set of 62 chips fabricated in a 90 nm technology. The stability of the PUF and an on-chip voltage-to-digital converter are evaluated at 9 temperature-voltage corners.***

Keywords - Physical Unclonable Function, power grid, metal resistance variations

1 Introduction

Physical unclonable functions (PUFs) are promising components in the next generation of integrated circuit (IC) security. PUFs derive random but reproducible bitstrings that can be used in security applications such as encryption, authentication, feature activation, metering, etc. The bitstrings are generated on-the-fly using dedicated hardware primitives and processing engines, and thereby avoid the need for storage in on-chip non-volatile memories. This feature not only improves their resilience to invasive attacks designed to steal the secret keying material, but it also reduces the cost of manufacturing the IC. The latter is true because, in many cases, PUFs are designed using components that can be fabricated using standard CMOS processing steps, and therefore, the cost of integrating non-standard components, such as non-volatile memories, is eliminated. Another important characteristic of the PUF as a next generation security mechanism is its potential for generating large numbers of repeatable random bits. This feature offers new opportunities for software processes to strengthen security mechanisms, for example, by allowing frequent re-keying in encrypted communication channels and by allowing a large, changing set of shared keys to be utilized among multiple communicating entities.

PUFs are designed to be sensitive to variations in the printed and implanted features of wires and transistors on the IC. Precise control over the fabrication of IC components is becoming more difficult in advanced technology generations, resulting in a wider range of electrical variations among and within the replicated copies of the chip. Signal variations that occur within the IC are the source of entropy for the PUF.

Several statistical criteria have emerged as important metrics for judging the quality of a PUF. **Interchip hamming distance (HD)** is used to determine the *uniqueness* of the bistrings among the population of chips. Similarly, the NIST statistical test suite can be used to evaluate the *randomness* of the bistrings produced by each chip [1]. And **intra-chip HD** can be used to evaluate *stability* of the bitstrings, i.e., the ability of each chip to reproduce the same bitstring time-after-time, under varying temperature and voltage conditions.

In this paper, we focus on determining the temperature and voltage (**TV**) stability of a PUF that is based on resistance variations which occur in the metal wires of the chip's power grid. A significant benefit of using metal structures is that "noise-related" variations, such as those introduced by TV variations, result in linear changes to the measured voltages. This linear scaling characteristic allows the relative magnitude of two voltages to remain consistent across changes in temperature and voltage, which, in turn, improves the stability of the PUF to bit-flips[1], when compared, for example to PUFs which leverage transistor-based variations.

In our experiments, we evaluate the power grid (PG) PUF at 9 TV corners, i.e., over all combinations of 3 temperatures; -40°C, 25°C and 85°C, and 3 voltages; nominal and +/- 10% of nominal. The evaluation is carried out on a set of chips fabricated in IBM's 90 nm, 9 metal layer bulk silicon process. The stability of the bitstrings is measured using intra-chip HD and 'probability of failure' techniques. Randomness and uniqueness are also evaluated using the NIST test suite and inter-chip HD methods. A bit-flip avoidance scheme is proposed and evaluated that reduces the probability of a failure to reproduce the bitstring to less than 1E-9. We also investigate an on-chip voltage-to-digital converter (VDC) and its stability across the 9 TV corners.

2 Background

Random bitstrings form the basis for encryption, identification, authentication and feature activation in hardware security. The introduction of the PUF as a mechanism to generate random bitstrings began in [2] and [3], although their use as chip identifiers began a couple years earlier [4]. Since their introduction, there have been many proposed architectures that are promising for PUF implementations, including those that leverage variations in transistor threshold voltages [4], in speckle patterns [2], in delay chains and ROs [2-3][5-9+many others], in thin-film transistors [10], in SRAMs [11-12], in leakage current [13], in metal resis-

1. A bit flip are defined as '0-to-1' and '1-to-0' change in generated bitstring as temperature and voltage are varied.

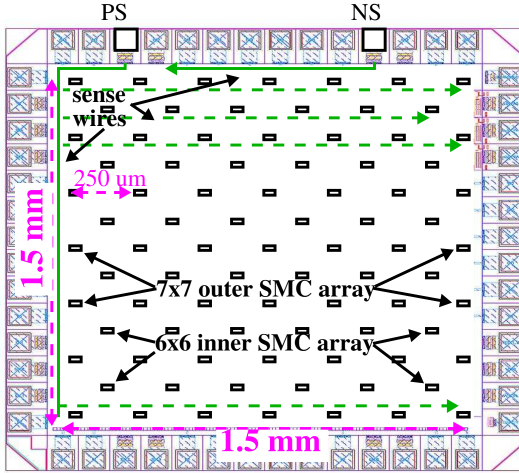


Fig. 1. a) Block diagram of 90 nm chips, with voltage sense pads along top and two arrays of SMCs, a 7x7 outer array and a 6x6 inner array.

tance [14-15], in optics and phase change [16], in sensors [17], in switching variations [18], in sub-threshold design [19], in ROMs [20], in buskeepers [21], in microprocessors [22], using lithography effects [23], and aging [24].

The main contributions of this paper are 1) the evaluation of the TV stability of the PG-PUF, and 2) the evaluation of an on-chip voltage-to-digital converter and its corresponding TV stability.

3 Experiment Setup

3.1 Test Chip Design

Fig. 1(a) gives a block diagram of the 90 nm test chip architecture. The chip padframe consists of 56 I/Os, and surrounds a chip area of approx. 1.5 mm x 1.5 mm. Two PADs labeled PS and NS along the top of the figure refer to *voltage sense* connections, the 'P' version for sensing voltages near V_{DD} and the 'N' version for voltages near GND. These terminals wire onto the chip and connect to 85 copies of a *Stimulus/Measure circuit* (SMC). The SMCs are distributed across the entire chip (see small rectangles) as two arrays, a 7x7 outer array and a 6x6 inner array. Although not shown, a scan chain connects serially to each of the SMCs to allow each of them to be controlled.

The schematic diagram of the SMC is shown in Fig. 2. A pair of large 'shorting transistors', capable of sinking approx. 10 mA of current through the power grid when enabled, are shown along the bottom of the figure[1]. A set of 16 'pseudo' transmission gates (TGs), labeled 1_a through 1_h, serve as *voltage sense* devices. Eight of the TGs connect to 8 (of the 9) metal layers that define the V_{DD} stack-up of the power grid, as shown on the left side of Fig. 2, while the other 8 connect to the GND stack-up. Scan FFs and 3-to-8 decoders allow exactly one of the TGs to be enabled in each of the stack-ups.

An additional TG connects to the drains of the 8 stack-

1. The resulting voltage drop/rise on the V_{DD} and GND grid, resp. is less than 10 mV.

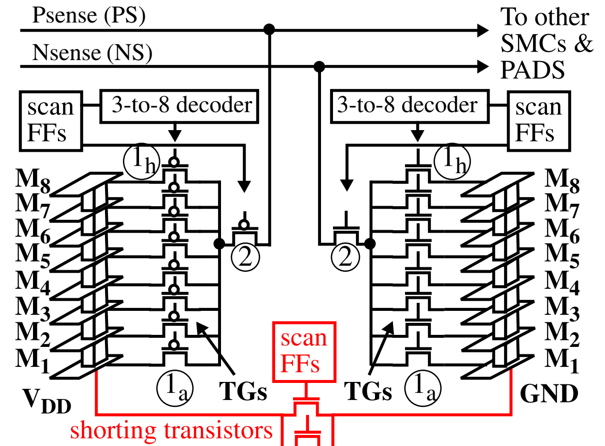


Fig. 2. SMC schematic in 90 nm chips.

up TGs, labeled as '2' in Fig. 2, one for V_{DD} and one for GND. Separate scan FFs control their connection to the chip-wide wires that route to the PS and NS pins of Fig. 1. This configuration and control mechanism allows any V_{DD} and GND voltage to be measured using off-chip voltmeters.

3.2 PGV Experiments and Challenge Scenarios

A 'challenge' in our experiments is applied by configuring the scan chain to 1) enable the shorting transistors within an SMC, and 2) enable two TGs in that same SMC, in particular, the TG labeled 2 in Fig. 2 and one from the group 1_a through 1_h. Once enabled, the voltage drop/rise is measured on the NS and PS pads using voltmeters.

In order to reduce bias effects and correlations that exist in the V_{DD} and GND stack-ups, we compute **inter-layer** voltage drops/rises by subtracting pair-wise, the voltages measured from consecutive metal layers, i.e., V_{M1} - V_{M2}, V_{M2} - V_{M3}, etc. These voltage *differences*, called power grid voltage differences (**PGVDs**), also allow the PUF to leverage the independent resistance variations that occur in each of the metal layers of the power grid. The 8 TGs in the V_{DD} and GND stacks as shown in Fig. 2 indicate that 7 PGVDs can be computed per stack. However, the structure of the power grid on the chips reduces the voltage drops on the upper layers of the power grid. Therefore, we restrict our analysis to PGVDs generated using the lower 4 metal layers, which allows 3 PGVDs to be computed. Therefore, each chip generates 85 SMCs * 3 metal layer pairings = 255 PGVDs for each of the V_{DD} and GND stacks.

Each of the PGVDs can be compared with other PGVDs in various combinations to produce a bitstring. We focus our analysis on bitstrings generated by comparing each PGVD with all others generated using the same metal layer pairing. Therefore, the total number of bits per chip is 85*84/2 per metal layer pairing * 3 metal layer pairings * 2 grids = 3,570 * 6 = 21,420 bits.

As is customary, we randomize the order in which the comparisons are made. In an on-chip implementation, this can be accomplished using an LFSR and a seed. The process is modeled in our experiments using the functions *srand(seed)* and *rand()* from the C programming library.

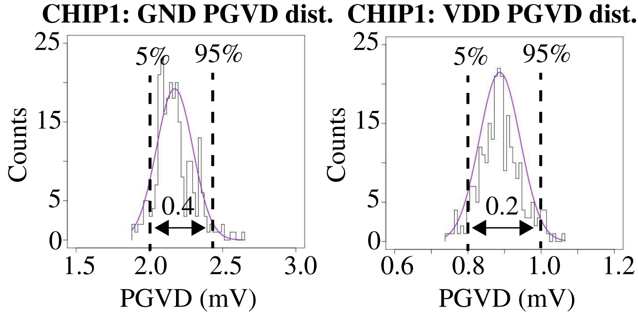


Fig. 3. $CHIP_1$ GND and VDD PGVD distributions with Gaussian curve fits and 5% and 95% thresholds.

4 Experimental Results

4.1 Bit Stability

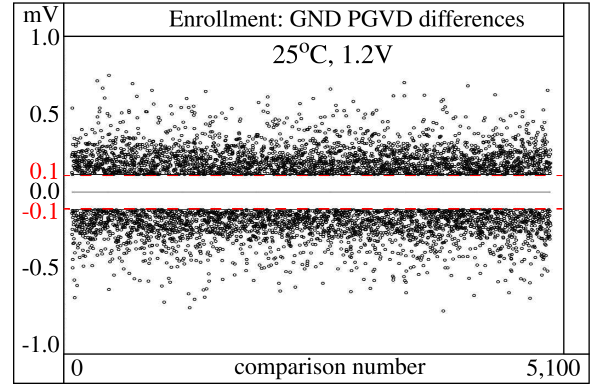
In our experiments, we found that *unstable* bits, defined as bits that are susceptible to 'flipping' because their PGVDs are very similar, actually reduce several quality metrics associated with the overall bitstring, including inter-chip HD and NIST statistical test scores. Moreover, including unstable bits in the bitstring requires the inclusion of error correction [3] and Helper Data schemes [25], that weaken security and increase overhead.

We propose an alternative scheme that identifies and discards unstable bits, which we call **thresholding**. Thresholding is carried out by first computing a threshold from the distribution characteristics of the PGVDs. This is illustrated using the GND and V_{DD} PGVD distributions for a sample chip, $CHIP_1$, in Fig. 3. Each distribution contains 255 PGVD values, derived as described in Section 3.2. The distance between the 5% and 95% points in the distributions is used to derive the thresholds for the threshing algorithm, which is approx. 0.4 mV for GND PGVDs and 0.2 mV for the V_{DD} PGVDs for this chip. The limits at 5% and 95% are used to avoid distortions caused by potential outliers in the PGVD values for each chip.

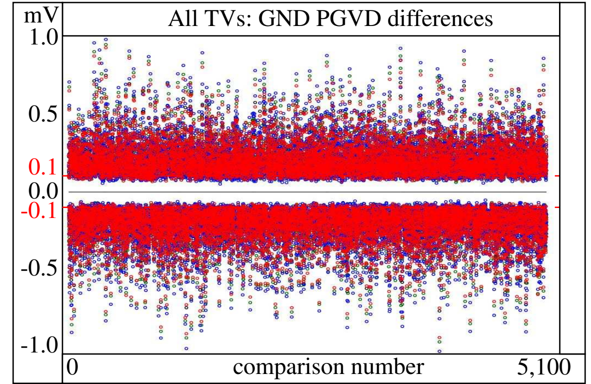
The thresholds are then scaled by a constant to produce the actual threshold used during bit generation. Fig. 4 provides an illustration of the bit generation process using the GND PGVDs for $CHIP_1$. Both graphs plot the bit comparison number along the x-axis against the value of the difference between the two PGVDs being compared. Only the bits that survive the thresholding, called **strong bits**, are included in the plots, i.e., the x-axis shows only about half of the 10,710 comparisons. Points that appear in the upper portion of the figures generate a '1' bit while points in the lower portion generate a '0' bit.

Fig. 4(a) shows only the points obtained from enrollment, which is carried out at 25°C, 1.2 V. The thresholds are depicted using two horizontal lines at 0.1 and -0.1 mV. This value is obtained by scaling the 0.4 mV obtained from the distribution by a constant 0.25[1]. In contrast, Fig. 4(b) adds in the data points from the remaining 8 TV (regeneration)

1. The scalars 0.25 and 0.50 for the GND and V_{DD} thresholds, resp. were determined to be sufficient to prevent bit-flips in all chips.



(a)



(b)

Fig. 4. $CHIP_1$ GND PGVD differences computed for bit generation during enrollment (a) and regeneration (b) at 9 TVs. Points in upper portion of plots generate '1's, points in lower portion generate '0's.

corners, color-coded to indicate the temperature; green for 25°C, blue for -40°C and red for 85°C. Close inspection reveals that some of the data points from regeneration appear within the threshold band of width 0.2 mV, centered around 0.0. Noise that occurs during regeneration causes points to move vertically, but as long as none move across the 0.0 line, no bit-flips occur.

The usage scenario that enables this process to be applied in situations where exact regeneration of a bitstring is required works as follows. During the initial bitstring generation, thresholding is used to identify the unstable bits. For each unstable bit, its numbered position in the sequence of challenges applied to generate the bitstring is recorded in public storage. Later, during regeneration, thresholding is disabled and public memory is consulted to determine which challenges to apply during bit generation.

4.2 Statistical Characterization of the Bitstrings

The results of applying the thresholding technique to 62 chips tested under 9 TV corners are described in this section. An important concern regarding the thresholding technique deals with the fraction of bits that survive it. In our experiments, we found this fraction to be different for the GND and V_{DD} stacks. On average, approx. 50% of the comparisons using the GND PGVDs survive the thresholding, while only 20% of the comparisons survive using the V_{DD} PGVDs. The lower value for the V_{DD} PGVD analysis

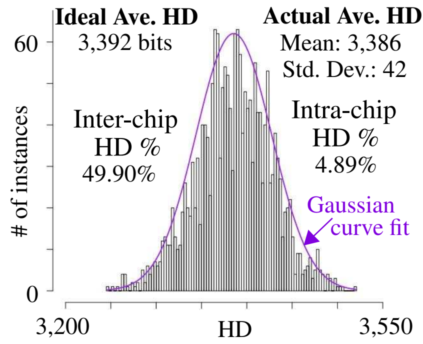


Fig. 5. Distribution of HDs using stable bitstrings from 62 chips. Number of HDs is 1,891 using bitstrings of length 6,784 bits.

occurs because of the increased noise levels on the V_{DD} grid, relative to the GND grid. As a consequence, the average bitstring length reduces to approx. 7,715 bits from the original size of 21,420 bits. These bitstrings are, however, reproducible at all of the 9 TV corners.

The true average intra-chip HD, which is a measure of the underlying bit stability across the TV corners, is computed as 4.89%. This value is obtained by analyzing the full length, i.e., 21,420-bit, bitstrings with thresholding disabled and counting the number of times a bit-flip occurs in each bit position across all 9 TV corners. Any value less than approx. 5% is considered high quality according to the published literature on PUFs.

Inter-chip HD, as indicated earlier, measures the uniqueness of the bitstrings, where the best possible result is 50%, i.e., on average, half of the bits in the bitstrings of any two arbitrary chips are different. Fig. 5 plots the distribution of inter-chip HDs. The 1,891 HDs included in the distribution are obtained by pairing the stable bitstrings from all chips under all combinations. The chip with the shortest stable bitstring is used to set the size of the bitstrings used in each HD calculation, requiring all bitstrings to be truncated to 6,784 bits. The average HD is 3,386 (49.90%), which is very close to the ideal HD of 3,392 (50.00%).

We also evaluated randomness using the NIST statistical tests at the default significance level of 0.01 [1]. Given the relatively short length of the stable bitstrings, only 11 of the 15 tests are applicable[1]. The bar graph shown in Fig. 6 gives the number of passing chips on the z-axis for each of the 11 tests on the x-axis, and for each of 10 different seeds on the y-axis. The number of passing chips is in reference to passing the *null hypothesis*. The null hypothesis is specified as the condition in which the bitstring-under-test is random. Therefore, a good result is obtained when the number of chips that pass the null hypothesis is large.

With 62 chips, NIST requires that at least 59 chips produce a *p value* that is larger than the significance level, otherwise the whole test is considered ‘failed’. Overall, of the 11*10 = 110 bars, 45 are full height indicating that all 62 chips passed the test, 34 bars have height 61, 23 have height 60 and 6 have height 59. Therefore, 108 bars of the 110 are

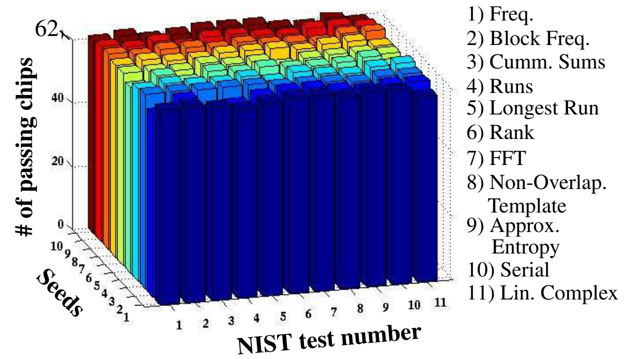


Fig. 6. Number of passing chips from NIST tests using 11 of the 15 applicable tests.

equal to or larger than the required value to pass the test, and only 2 bars are below the threshold; one at 58 and the second at 57. Moreover, **all but 9 of Pvalue-of-the-Pvalues tests passed**, indicating the P-values are uniformly distributed between 0.0 and 1.0. The fails in this category occurred in the Rank and Non-Overlapping Template tests, both which NIST recommends testing with much larger bitstrings than those used here. Overall, these are very good results and indicate the bitstrings are cryptographic quality.

4.3 Bit-Flip Probability Analysis and Triple-Module-Redundancy (TMR)

The large size of the bitstrings produced by the PUF can be used to further enhance their reliability over that provided by thresholding alone. This can be accomplished by creating 3 copies of a fixed-length bitstring from the sequence of strong bits produced by the PUF. The 3 copies can then be compared as a means of detecting and correcting bit flips, in the spirit of a popular scheme used in fault tolerance called triple-module-redundancy or TMR. TMR is based on a ‘majority voting’ scheme in which the final bit for a given bit position is obtained by taking the majority across all 3 copies of the bitstrings.

We investigate this technique using fixed-length bitstrings of 256-bits. A TMR-based bitstring is created during enrollment by copying the first 256 strong bits into the ‘1st copy’ of the fixed-length bitstring as shown in Fig. 7. The second two copies are created by parsing the remaining strong bits, searching for matches to the 1st copy[2]. As described above for thresholding, the positions of the matching bits are indicated by writing a ‘1’ in the public storage bitstring (not shown), while the positions of the skipped bits (and the weak bits encountered under thresholding) are indicated by writing a ‘0’. Later, during regeneration, the public storage bitstring is consulted to determine which challenges are to be used to re-construct the 3 copies of the bitstring. Once created, the final bitstring is obtained by majority vote on each column as shown in Fig. 7. This allows the correct bitstring to be generated despite any single bit-flips that may occur in a column, such as the one shown in the last column of ‘Redundant BS_1’.

In order to illustrate the improvement provided by

1. See [1] for details concerning NIST tests.

2. The TMR-based bitstring of length *n* requires approx. 5**n* strong bits to construct.

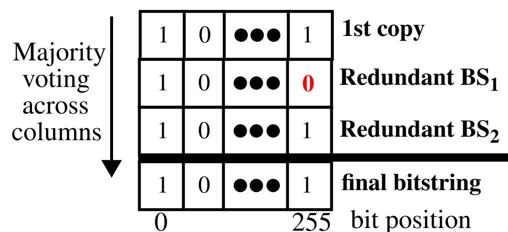


Fig. 7. TMR process for bitstring regeneration.

TMR over voltage thresholding alone, we iteratively decreased the GND threshold scalar given above as 0.25, in 0.01 steps down to 0.0. As the threshold is decreased, bit flips begin to occur in the thresholding-only bitstrings. A thresholding-only 'probability of failure' curve can be constructed by counting the number of bit flips that occur in the bitstrings from all 62 chips and dividing it by the total number of bits. A similar curve can be constructed using TMR, but in this case, a bit flip is not counted unless it occurs in 2 or more of the 3 bits of a column as shown in Fig. 7. Moreover, the total number of bits used in the denominator for the TMR-based curve is reduced by a factor of 3 to account for the actual number used in the final TMR-based bitstring.

Fig. 8 plots to the data points for these two curves as well as two 'exponential-curve' fits to them. The GND threshold scaling constant is plotted along the x-axis against the *probability of failure* on the y-axis. The exponential curve fits allow the probability of failure to be predicted for thresholds beyond (to the right) of the last recorded bit flip in our small population of chips. For example, the probability of failure using voltage thresholding alone at the 0.25 threshold is 2E-6. This improves by over three orders of magnitude to 1E-9 using the TMR-based scheme. Of course, the TMR-based scheme can be expanded to further improve bit-flip resilience by generating 5 (or more copies) of the bitstring, at the expense of increased usage of bits and public storage size.

5 VDC Experiments

The analysis presented in the previous sections is carried out on digitized voltages obtained from an off-chip voltmeter. In this section, we analyze the bitstrings generated by the PUF after digitizing the voltages using an on-chip voltage-to-digital converter (VDC) that is subjected to the same TV corners as the PUF itself. The VDC is similar in design to that described in [26] but is used in our application in a unique way.

The architecture of the VDC is shown in Fig. 9. The VDC is designed to 'pulse shrink' a negative input pulse as it propagates down an inverter chain. As the pulse moves down the inverter chain, it activates a corresponding set of latches to record the passage of the pulse, where activation is defined as storing a '1'. A thermometer code (**TC**), i.e., a sequence of '1's followed by a sequence of '0's, represents the digitized voltage.

The VDC works by introducing a fixed-width (constant) input pulse, which is generated by the pulse generator shown on the left side of the Fig. 9. Two analog voltages, labeled Cal0 and Cal1 connect to a set of series-inserted NFET transistors in the inverter chain, with Cal0 connecting to NFETs in even numbered inverters and Cal1 to the

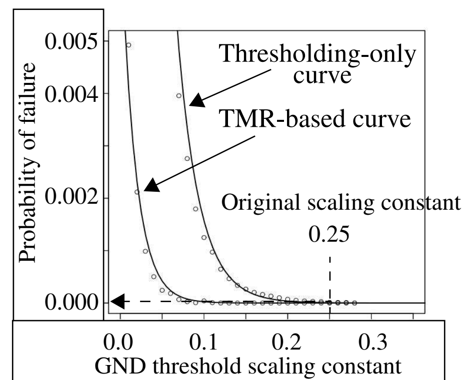


Fig. 8. GND threshold scaling constant vs. probability of failure (y-axis).

NFETs in odd numbered inverters (see call-out on right side of Fig. 9). The propagation speed of the two edges associated with the pulse are controlled separately by these voltages. The pulse will eventually die out at some point along the inverter chain when the trailing edge of the pulse 'catches up' to the leading edge. This is ensured by fixing Cal0 at a voltage higher than Cal1. The digital representation of the applied Cal0/1 voltages can then be obtained by counting the number of '1's in the latches.

As described earlier, PGVDs are created by subtracting the voltages measured on consecutive metal layers in the power grid. Instead of digitizing these PGVs one-at-a-time with the VDC and then subtracting them, we carry out the difference operation in the analog domain by applying the two voltages from consecutive metal layers to the Cal0 and Cal1 inputs. The larger PGV from the lower metal layer, M*n*, of the pair is applied to Cal0 while the PGV from the adjacent, higher metal level layer, M*n+1*, is applied to Cal1 (this convention is reversed for the V_{DD} grid voltages).

Fig. 9 shows how this is accomplished. The PG array is configured to enable the PGV on M*n* to drive the NS (or PS) pin and an off-chip voltmeter (VM) is then used to digitize the value (same process as described for the original experiments). The PGV is then multiplied by 15 and added to an offset, and the voltage sum is used program an off-chip power supply which drives Cal0. The exact same process is carried out for the PGV produced on metal layer M*n+1* except the final value is used to program a second off-chip power supply which drives Cal1. The multiplication and offset operations are necessary because the VDC requires the Cal0/1 voltages to be set between 500 mV and 800 mV for proper operation.

Note that unlike the PGV experiments in Section 4, the on-chip VDC is subjected to the same TV variations as the PUF (as would be the case in an actual implementation), and therefore its characteristics will vary as well. We developed a calibration process that 'tunes' the offset voltage to compensate for some of the changes in VDC behavior but since the measurements are differential, the VDC is able to self-calibrate and cancel out most of the adverse effects of TV variations by itself.

We carried out the same set of experiments and followed the same process as described in Sections 3 and 4 on 6 chips (note to reviewers: the results from all 62 chips will

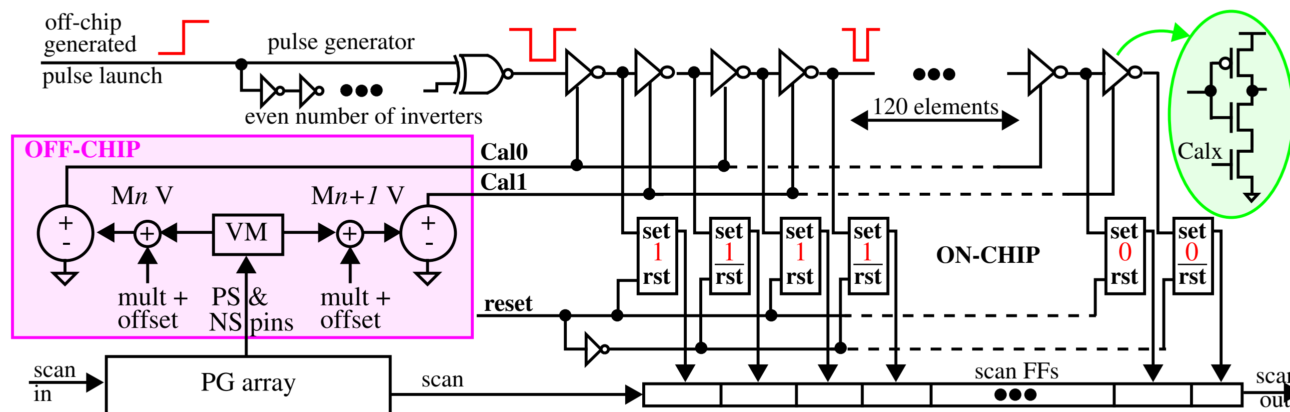


Fig. 9. Voltage-to-Digital Converter (VDC). On the left side is off-chip instrumentation that measures two voltages from the PG array, adds an offset and programs the Cal0/Cal1 inputs of the VDC on the right.

be included if accepted). The results are as follows. The inter-chip HD is 50.02% and all of the 11 NIST statistical tests are passed except for 2 Non-Overlapping Template tests. The average length of the bitstrings is 2,200 bits, which indicates that approx. 10% survive the thresholding process (down from approx. 35% in PGV experiments). Therefore, the VDC digitization process adds noise and reduces the size of the bitstrings. The main sources of the noise are in the off-chip processing of the signals (we expect this to be smaller in a complete on-chip implementation currently underway), and in power supply noise coupling to the VDC.

6 Conclusions

We analyze the statistical quality of bitstrings produced by a PUF that leverages resistance variations in the power grid wires of an IC. Experimental results are reported for chips fabricated in a 90 nm technology, and which are tested under 9 different temperature-voltage corners given by the industrial standard specifications. Voltage thresholding and TMR-based techniques are investigated as a means of improving the bit-flip resilience of the regenerated bitstrings. An on-chip voltage-to-digital converter is also investigated. The statistical results indicate that the power grid PUF is able to generate cryptographic quality bitstrings of significant length.

References

[1] NIST: Computer Security Division, Statistical Tests, http://csrc.nist.gov/groups/ST/toolkit/rng/stats_tests.html

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

[3] B. Gassend, *et al.*, "Controlled Physical Random Functions," *Conference on Computer Security Applications*, 2002.

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

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

[6] G. Qu and C. Yin, "Temperature-Aware Cooperative Ring Oscillator PUF", *Workshop on 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] Y. Meng-Day, *et al.*, "Performance Metrics and Empirical Results of a PUF Cryptographic Key Generation ASIC," *HOST*, 2012, pp. 108-115.

[9] S. S. Mansouri and E. Dubrova, "Ring Oscillator Physical Unclonable Function with Multi Level Supply Voltages", *ICCD*, 2012, pp. 520-521.

[10] S. Maeda, *et al.*, "An Artificial Fingerprint Device (AFD): a Study of Identification Number Applications Utilizing Characteristics Variation of Polycrystalline Silicon TFTs," *Trans. on Electron Devices*, number 50, issue 6, June, 2003, pp.1451- 1458.

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

[12] M. Bhargava, *et al.*, "Reliability Enhancement of Bi-Stable PUFs in 65nm Bulk CMOS, " *HOST*, 2012, 79-83.

[13] Y. Alkabani, *et al.*, "Trusted Integrated Circuits: A Nondestructive Hidden Characteristics Extraction Approach," *Information Hiding*, 2008.

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

[15] J. Ju, *et al.*, "Bit String Analysis of Physical Unclonable Functions based on Resistance Variations in Metals and Transistors", *HOST*, 2012, pp. 13-20.

[16] K. Kursawe, *et al.*, "Reconfigurable Physical Unclonable Functions - Enabling Technology for Tamper-Resistant Storage", *HOST*, 2009, pp.22-29.

[17] K. Rosenfeld, *et al.*, "Sensor Physical Unclonable Functions", *HOST*, 2010, pp. 112-117.

[18] W. Xiaoxiao and M. Tehranipoor, "Novel Physical Unclonable Function with Process and Environmental Variations", *DATE*, 2010, pp. 1065-1070.

[19] L. Lin, *et al.*, "Low-Power Sub-Threshold Design of Secure Physical Unclonable Functions", *LPED*, 2010, pp. 43-48.

[20] U. Ruhrmair, *et al.*, "Applications of High-Capacity Crossbar Memories in Cryptography", *Trans. on Nanotechnology*, Volume: 10 , Issue: 3, 2011, pp. 489-498.

[21] P. Simons, *et al.*, "Buskeeper PUFs, a Promising Alternative to D Flip-Flop PUFs", *HOST*, 2012, pp. 7-12.

[22] A. Maiti and P. Schaumont, "A Novel Microprocessor-Intrinsic Physical Unclonable Function," *FPLA*, 2012, pp. 380-387.

[23] A. Sreedhar and S. Kundu, "Physically Unclonable Functions for Embeded Security based on Lithographic Variation", *DATE*, 2011, pp. 1-6.

[24] S. Meguerdichian and M. Potkonjak, "Device Aging-Based Physically Unclonable Functions", *DAC*, 2011, pp. 288-289

[25] Y. Dodis, *et al.*, "Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data", SIAM J. Comput., 38(1):97-139, 2008.

[26] L. Guansheng, Y.M. Tousi, A. Hassibi and A. Afshari, "Delay-Line-Based Analog-to-Digital Converters,", Trans. on *CAS II*, Volume: 56 , Issue: 6, 2009, pp. 464-468.