IP-Level Implementation of a Resistance-Based Physical Unclonable Function

Dylan Ismari and Jim Plusquellic, University of New Mexico
dismari@unm.edu, jimp@ece.unm.edu

Abstract -- *A complete on-chip implementation of a bit generation engine using a physical unclonable function is presented in this paper. The bit generation engine, called the JellyFishPUF (JFP), provides keying material for encryption, authentication bitstrings for anti-counterfeiting and true random number generation. JFP utilizes a Physical Unclonable Function that is based on resistance variations in metals and transistors. JFP is fully implemented as a layout in an area of 0.125 mm^2 using a 65 nm technology, which includes a 2KB SRAM for public data storage. The bitstrings produced from Monte Carlo SPICE-level simulations of the entropy source in combination with logic simulations of the digital engine are evaluated with respect to randomness, uniqueness and stability metrics across a wide range of temperature and voltage corners.*

Keywords -- Physical Unclonable Function

1 Introduction

Physical unclonable functions (PUFs) are hardware security primitives designed to produce random but reproducible bitstrings from variations in the printed and implanted features of wires and transistors on an integrated circuit (IC). Each IC is uniquely characterized by random manufacturing variations, and therefore, the bitstrings are unique from one chip to the next. PUFs can serve several important security applications including authentication and cryptography, which in turn can be used for secure communications, anti-counterfeiting, detecting malicious system alterations in the field, feature activation, hardware metering, etc.

In this paper, we present a full on-chip implementation of a PUF system that we call the JellyFishPUF (**JFP**). The entropy source leveraged in the PUF is based on within-die variations in transistors and conductors, e.g., polysilicon, metal wires, vias and contacts, and is similar in structure to a living jellyfish. An array of 2,048 identically designed cells, called stimulus/measure circuits or SMCs, defines the entropy source. The voltages produced by the SMCs are routed using pass gates to an on-chip voltage-to-digital-converter or VDC. The VDC converts the SMC voltages into 8-bit digital values, which reflect their relative magnitudes. A digital controller accepts inputs from user applications and carries out the specified function, e.g., enrollment for secret key generation. The output of the JFP engine is a 256-bit (or larger) bitstring plus public data to handle functions that require regeneration.

The contributions of this work are as follows:

- A complete PUF engine, implemented in a 65 nm technology, with area 0.125 mm^2 and 256-bit bitstring generation time as small as 2 mS.
- The implementation of bitstring functions including enrollment, regeneration, authentication and random number generation, as well as a new function call 'band enrollment' which provides up to 2^n unique 256-bit bitstrings/chip.
- An implementation of a process called **normalization** that eliminates transistor-based variations from the entropy source but preserves resistance variations in the conductors, i.e., polysilicon, metal wires, contacts and vias. Our analysis shows this process improves reliability of the bitstring regeneration process.
- The implementation of a differential power analysis resistant VDC and a process called **calibration** that allows the VDC to digitize voltages from the SMC across temperature variations from -40^{o}C to 125^{o}C and supply voltage variations of +/- 10% of nominal.
- An entropy source that leverages single vias, minimum width metal and polysilicon wires over 5 of the metal layers available in the IP block.

References [11-12] describe preliminary results from a test chip which uses a metal-based entropy source, a 'pulse-shrinking' version of the VDC as well as processes related to calibration, thresholding and XMR. However, this paper integrates all of these components in a unified system architecture and investigates each of the novel concepts described above.

2 Related Work

Random bitstrings form the basis for encryption, identification, authentication and feature activation in hardware security. The introduction of the silicon PUF as a mechanism to generate random bitstrings began in [1], although their use as chip identifiers began a couple years earlier [2]. Since their introduction, there have been many proposed architectures that are promising for PUF implementations, including those that leverage variations in transistor threshold voltages [2-3], in delay chains and ROs [1][4-7+many others], in SRAMs [8-9], in leakage current [10], in metal and transistor resistance [11-12], in clock networks [13], in scan chains [14] and transmission lines [15]

3 System Implementation

A block level diagram of the JellyFishPUF (JFP) PUF system architecture is shown in Fig. 1 and its layout in Fig. 2. The PUF Engine is a digital controller that coordinates a series of operations described in the following sections. The SRAM shown on the left in Fig. 2 is used by the PUF Engine for Voltage Distribution analysis and for storing helper data during bit generation. The edge generator (Edge Gen.), voltage-to-digital converter (VDC) and SMC array define the entropy source and conversion components.

The SMC (stimulus-measure-circuit) array (Fig. 1(a)) is defined as a set of 128 4x4 SMC blocks, arranged in 16 rows and 8 columns. Each SMC element within a 4x4 block (Fig.

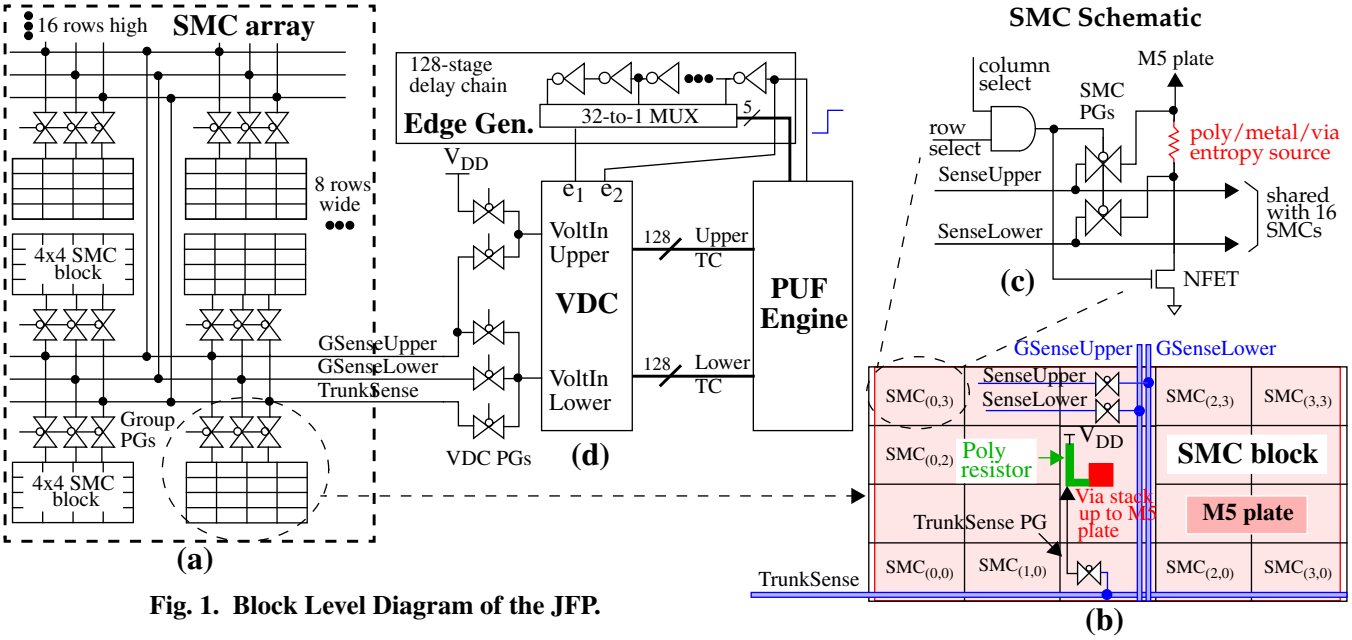

Fig. 1. Block Level Diagram of the JFP.

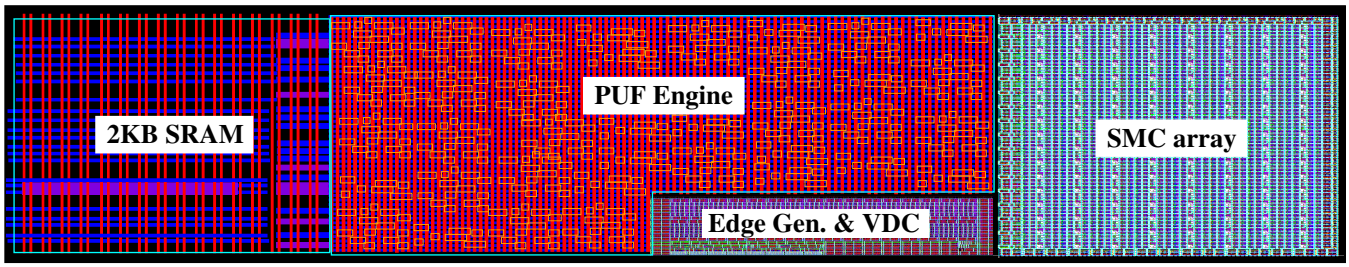

Fig. 2. JFP layout in a 65 nm technology with dimensions of 830 µm x 150 µm.

1(b)) is able to provide a single component of entropy. Therefore, the entire array defines an entropy source with 2,048 components.

Fig. 1(c) shows a schematic of an SMC element. It is composed of an AND gate, which serves to enable the SMC, and two pass gates (PGs) connected across the entropy source. The entropy source is a 1 µm silicided polysilicon wire and a single via stack from poly up to M5. An NFET connects to the polysilicon wire and provides the stimulus of approx. 500 µA when the SMC is enabled. The M4-M5 via on the upper end of the entropy source connects to an M5 metal plate that covers the 4x4 SMC block as shown in Fig. 1(b). The plate is connected to the V_{DD} supply grid through a controlled-resistance silicided Poly resistor of approx. 400 Ohms. Therefore, when an SMC is enabled, the NFET current creates a voltage drop across the entropy source which can be sensed by the two PGs. The M5 plate and Poly resistor provide a common node connected to V_{DD} for all SMCs in the 4x4 block. This common node in combination with the TrunkSense PG shown along the bottom of Fig. 1(b) allow voltage variations introduced by the different NFET currents within the SMCs of the block to be eliminated. The process, called *normalization*, is described below.

The SMC PGs connect to two wires labeled SenseUpper and SenseLower, which are shared among all SMCs within the block. Two additional PGs at the block level connect these wires to two globally routed GSenseUpper and GSenseLower wires shown in Fig. 1(a) and (b), which connect across the 128 SMC blocks (the same is true of the TrunkSense wires). These wires route out of the SMC array to three VDC pass gates shown in Fig. 1(d). The PUF Engine provides a sequence of control signals which allows each of these sense voltages to be digitized by the VDC.

The inputs of the VDC are two voltages labeled VoltInUpper and VoltInLower and two wires e_1 and e_2 that are connected to the Edge Gen. (Fig. 1(d)). The VDC outputs two 128-bit thermometer codes (TCs) that reflect the magnitude of the sense voltage inputs. A TC is defined as a string of '0's (or '1's) followed by a string of '1's (or '0's).

Fig. 3 gives a schematic of the VDC to better illustrate the digitization process. The VDC is composed of two 256-stage delay chains. The VoltInLower input connects to 128 NFETs, inserted in series with the odd-numbered inverters in the delay chain. VoltInUpper connects in a similar fashion to the upper delay chain. The PUF Engine starts the digitization process by driving a rising edge into the EdgeGen. as shown in Fig. 1(d). The Edge Gen. passes e_1 to the corresponding VDC input but delays e_2 by a Δt (determined by 32-to-1 select MUX). The two edges then 'race' down the two inverter chains at speeds relative to the magnitude of the VoltInUpper/Lower inputs.

Under the condition that VoltInUpper > VoltInLower,

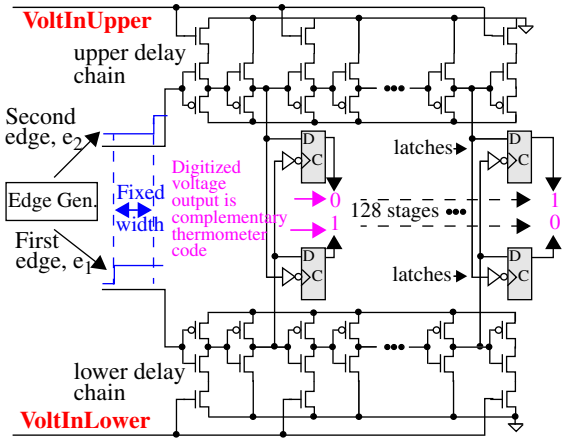

Fig. 3. Voltage-to-Digital Converter (VDC).

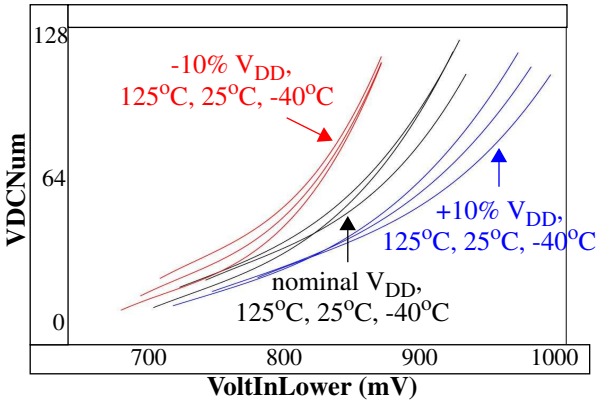

Fig. 4. VDC calibrated transfer curves at 9 temperature/voltage corners under the TT Nominal process corner.

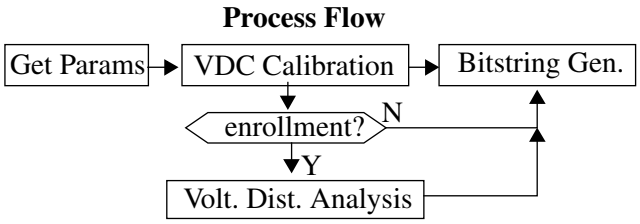

Fig. 5. Overall flow of bit generation process.

the edge propagating along the top delay chain eventually passes the edge on the bottom delay chain. The outputs of the even inverters along both delay chains connect to a set of latches that record the point at which this occurs. As shown in Fig. 3, the TC produced by the latches on the upper chain is a sequence of '0's followed by '1's, while a complementary pattern appears on the latch outputs of the lower chain. A value proportional to the magnitude of the voltage difference between VoltInUpper and VoltInLower can be obtained by counting the number of '1' in either of these TCs. We refer to the number of '1's as a **VDCNum**.

The transfer curves in Fig. 4 illustrate the behavior of the VDC across 9 temperature/voltage (TV) corners. The data for the curves is generated by SPICE-level simulations of an RC layout-extracted model of the Edge Gen. and VDC components of Fig. 2 under typical transistor (TT) and nominal wire resistance and capacitance conditions (Nominal). The data for the curves is generated by fixing VoltInUpper at V_{DD} and sweeping the voltage on VoltInLower from 650 mV to 1.0V, in 20 mV steps. The Δt between the edges e_1 and e_2 is fixed for any one curve but is 'tuned' for each TV corner. A **calibration process** implemented within the PUF Engine determines the appropriate Δt by monitoring the VDCNums produced with the VoltInLower set to the largest value on the curve. The control inputs to the 32-to-1 MUX are set such that the VDCNum produced under this condition is less than the overflow value of 128. Calibration involves successively reducing the Δt by reducing the digital select inputs of the 32-to-1 MUX.

The transfer curves illustrate that the mapping from voltages to VDCNum is non-linear. In particular, the VDC has higher sensitivity to changes in VoltInLower at the high end of the voltage range, than at the lower end. Although this 'distorts' the Gaussian nature of the voltage variations produced by the entropy source, our proposed voltage comparison technique (described below) is able to provide un-biased random values from the distribution.

4 PUF Engine Processing Techniques

The overall flow of the bit generation process is given in Fig. 5. User parameters, such as function and the size of the bitstring to generate, are stored in memory-mapped registers while the PUF Engine remains in idle mode. A 'bit gen start' input to the PUF Engine starts the process by first performing Calibration. The voltages used during calibration are a user-selectable subset of those produced by the SMC array itself, which ensures that overflow does not occur during subsequent processes.

Voltage distribution analysis is the second process to be performed when the function is enrollment or band enrollment (described below). This process constructs a histogram of digitized voltages from a subset of SMC elements and derives a value that reflects its width, which is distinct for each chip because of global variations in process parameters. Our proposed **thresholding technique**, which decides which voltage comparisons are 'stable' enough to generate a bit, makes use of this width information to improve reliability. The bitstring generation process follows the calibration and voltage distribution analysis processes.

4.1 Sample Analysis

User input parameters can be used to specify the number of samples that are averaged for each of the three processes, as a means of reducing measurement noise. As indicated, the VDC produces a value between 0 and 128 (8-bits). In order to fully leverage the benefits of averaging, the VDCNums are scaled to **11-bits**, with the 3 low-order bits representing 3 binary digits of precision. For example, the average value produced when 3 samples of 75 and 5 samples of 76 are generated by the VDC is 605 which is 75.625 in fixed point.

4.2 Digitization Options

The current sourced by the NFET within the SMC creates a voltage drop across the entropy source. The GSense-Upper and GSenseLower sense wires transfer this voltage to the VDC inputs through PGs, as discussed above. We use voltage drops as the entropy source because they eliminate

bias effects that can occur for SMCs in different regions of the array. Voltage drops are defined as ($V_{GSenseUpper}$ - $V_{GSenseLower}$). The control inputs to the VDC PGs shown in Fig. 1(d) provide two options for digitizing the voltage drops, which are referred to as the **digital** and **analog** methods. For the digital method, each of $V_{GSenseUpper}$ and $V_{GSenseLower}$ are digitized separately using the VoltInLower input of the VDC and the voltage drop is computed digitally from the 11-bit VDCNums. The VoltInUpper input in this case is set to V_{DD}. In contrast, the analog method places GSenseUpper on the VoltInUpper input and GSenseLower on the VoltInLower input. In this case, the single VDCNum produced reflects the difference in the analog voltages directly.

4.3 Normalization

The primary reason for partitioning of the SMC array into a set of 128 blocks is to support a special process called **normalization**. The objective of normalization is to eliminate transistor current variations as a component of the measured voltage drops across the entropy stack. Previous work suggests that the current-induced variations contribute significantly to TV noise, which, in turn, acts to reduce the probability of correctly regenerating the bitstring [9].

We think of normalization as a process that 'normalizes' the voltage drops for all SMCs within the block to a reference current. Normalization is derived from the basic circuit theory equation R = V/I given by Eq. 1 which states that the resistance of the entropy source can be obtained from the sense voltage measurements by dividing through by the NFET current. Unfortunately, measuring currents on-chip is

$$R = \frac{(V_{GSenseUpper} - V_{GSenseLower})}{I_{\mathrm{NFET}}} \qquad \textbf{Eq. 1.}$$

$$R_{norm} = \frac{(DV_{GSenseUpper} - DV_{GSenseLower}) \times 256}{(129 - DV_{TrunkSense})} \qquad \textbf{Eq. 2.}$$

challenging and impractical. Eq. 2 provides an alternative in cases where it is only necessary to determine a value that is 'proportional' to resistance. Here, $DV_{TrunkSense}$ is the digitized voltage (a value between 0 and 128) produced at the point where the Poly resistor connects to the M5 plate, as shown in Fig. 1(b). Current from any of the enabled SMCs in the block must flow across the Poly resistor and past this point on the M5 plate. Therefore, the voltage drop defined by (129 - $DV_{TrunkSense}$) is proportional to the NFET current because the Poly resistor is a shared connection to V_{DD} for all elements within the SMC block.

Normalization is a user specified option, that when enabled, instructs the PUF Engine to additionally digitize $V_{TrunkSense}$ and to carry out the operations defined by Eq. 2. The multiplication factor of 256 scales the digitized voltage drop and allows the result, R_{norm}, to be expressed and manipulated as an integer in the bit generation process.

4.4 Thresholding

Thresholding is used to improve the reliability of the bit generation process for applications that require exact regeneration of the same bitstring under different TV conditions. Bit generation options called **enrollment** and a new one called **band enrollment** use thresholding to accomplish this goal. Thresholding requires the difference between a pair of VDCNums obtained from two distinct SMCs to exceed a threshold. The threshold is derived by multiplying a user-specified value between 0.0 and 1.0 with the range of the Voltage Distribution computed earlier.

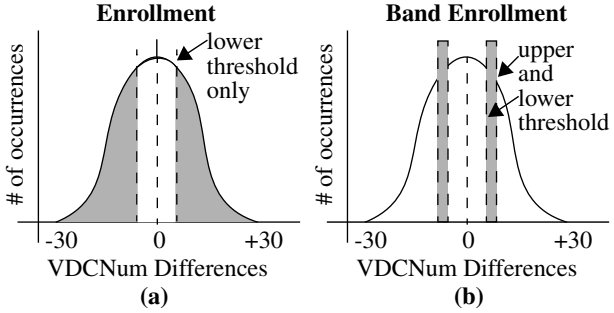

Fig. 6. Thresholding under (a) Enrollment and (b) Band Enrollment.

4.5 Bit Generation Options

The bit generation process compares the digitized voltage **drops** from a sequence of (SMC_x, SMC_y) pairings. The sequence is determined by two user-specified seeds and corresponding linear feedback shift registers (LFSRs). The LFSR sequencing with normalization enabled restricts comparisons to within each SMC block. A '1' bit is generated if the VDCNum of SMC_x in the pairing is greater than the VDCNum of SMC_y, otherwise a '0' is generated.

The PUF Engine allows the user to specify one of five different bit generation options, including enrollment, band enrollment, regeneration, authentication and true-random-number generation (TRNG). The difference in the process of generating a bitstring under each of these options is related to how two threshold parameters are used.

The distributions shown in Fig. 6 are used to illustrate the bit generation options. These distributions, unlike the distribution used in Voltage Distribution analysis, are defined using VDCNum **differences**, i.e., ($VDCNum_{SMCa}$ - $VDCNum_{SMCb}$) obtained from a pairing of SMCs *a* and *b*. The differences are plotted along the x-axis against their frequency of occurrence on the y-axis in the plots.

The thresholding criteria used for **enrollment** is shown in Fig. 6(a). The shaded regions on both sides of the distribution, delineated by 'lower threshold only', represents comparisons whose differences exceed the threshold, and therefore are permitted to generate a bit.

Band enrollment is similar and is illustrated in Fig. 6(b). A second objective of band enrollment is to provide a unique bitstring each time enrollment is carried out when the same seed is used for the LFSRs. This occurs to some degree for the enrollment function because VDCNum differences that are close to the threshold can be on either side of it during any given enrollment process. This is true because the statistical nature of measurement noise introduces uncertainty in the VDCNum differences. Band enrollment simply increases the probability closer to 50% that any VDCNum difference can be inside or outside the band during an enrollment process. It accomplishes this by using a second user-specified 'upper' threshold to create two narrow bands as shown in the figure.

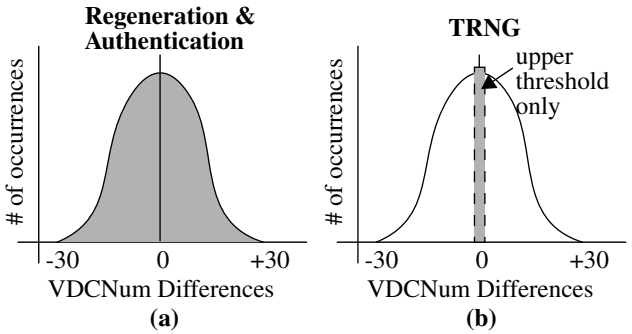

Fig. 7. Thresholding under (a) Regen./Authentication and (b) TRNG.

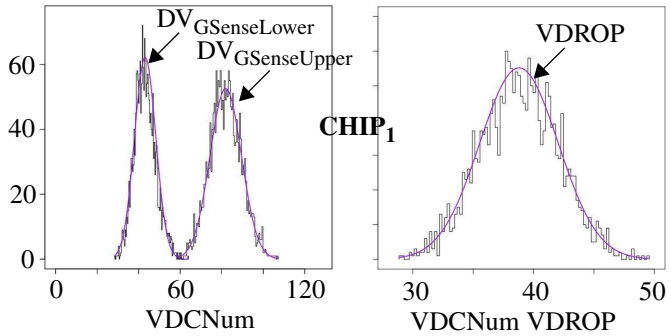

Fig. 8. (a) Digitized GSenseUpper/Lower distributions from TT Nominal, nominal V_{DD}, 25°C MC simulations, (b) and corresponding VDCNum differences.

The **regeneration** and **authentication** functions do not use thresholding and therefore all comparisons are valid, as illustrated by the shaded region in Fig. 7(a). However, the actual comparisons used during regeneration would be chosen in most cases from the shaded regions as shown in Fig. 6(a). This is true because the valid comparisons used in regeneration are determined from a 'helper data' bitstring, which is stored in the SRAM shown in Fig. 2 during enrollment. Enrollment records a '1' when a comparison is valid and a '0' when it is not. Regeneration reads this bitstring to ensure the same sequence of comparisons are carried out.

The valid band for true-random-number-generation (**TRNG**) is narrow and centered around the mean of the distribution as shown in Fig. 7(b). In this region, noise sources readily change the sign of the VDC difference and therefore the bit value varies randomly even for the same sequence of comparisons. The lower threshold is set to 0 and only the upper threshold is used to accomplished this. Note that band enrollment (without regeneration) can also be used as a TRNG.

4.6 Reliability Enhancing Techniques

In addition to the Normalization technique described above, the PUF Engine also implements a method called **XMR** for increasing the probability of correctly regenerating the bitstring [12]. XMR creates an odd number of 'copies' of the bitstring during enrollment and regeneration. A bit-wise majority voting technique is then carried out column-wise for each bit across the *n* copies during regeneration as a means of preventing bit flips in the final bitstring. For example, if the user specifies that 3 copies of the bitstring are to be generated (called 3MR), the majority voting scheme can 'correct' single bit flips that occur in any column. For 5MR, two bit flips per column can occur while preserving the ability to correctly regenerate the bitstring, and so on. We investigate these reliability enhancing schemes in the simulation experiments described below.

5 Simulation Results

Over 100K SPICE-level Monte Carlo simultaneous were run on an RC layout-extracted model of the SMC block shown in Fig. 1(b) under each of the 9 TV corners shown for the VDC transfer curves in Fig. 4. This allowed us to model 50 instances of the SMC array, i.e., 50 instances * 2048 SMCs. In addition to the transistor level variations introduced by the foundry models, we modified the RC netlist to enable CADENCE spectre to introduce within-die variations in the poly, metal wire, contacts and vias of the SMC block. The Monte Carlo sample limits for within-die variations of these components were set to approx. 30% of the chip-to-chip variations specified in the design manual.

The VDCNums corresponding to the $V_{SenseUpper}$ and $V_{SenseLower}$ values obtained from the MC simulations are derived using the transfer curves given in Fig. 4. Fig. 8(a) gives the distributions of the digitized voltages for $CHIP_1$, labeled $DV_{GSenseLower}$ and $DV_{GSenseUpper}$. Although not shown, the distribution using the actual voltage, $V_{GSenseLower}$, is wider than the distribution for $V_{GSenseUpper}$, and is opposite to the behavior shown in Fig. 8(a) where $DV_{GSenseUpper}$ is wider. The changing slope of the transfer curve changes the spread of the two distributions.

Fig. 8(b) shows the distribution of the VDCNum VDROPs, which is derived from a pair-wise subtraction of the DV_{GSense} values from Fig. 8(a). Although the distribution appears to be normal, an analysis of all 50 chips shows that the percentage of values above and below the mean varies by up to 2%, i.e., the worst chip distribution has 48% of the values below the mean and 52% above. This is also an artifact this is introduced by the non-linear transfer curve. However, our comparison methodology selects pairs of values randomly from this distribution to create VDCNum differences during bit generation, and as a consequence, **it is robust to non-Gaussian shapes in the underlying distributions**.

5.1 Randomness and Uniqueness Analysis

Inter-chip hamming distance (HD) [16] and the NIST statistical tests [17] are used to evaluate the statistical quality of bitstrings of size 512 bits generated under three XMR scenarios, including 0MR, 3MR and 5MR, each with (N) and without (NN) normalization enabled. The user-specified threshold in this analysis is set such that no bit flips occurred (intra-chip hamming distance is 0) across any of the 50 chips. The Inter-chip HDs given in Table 1 indicate that the uniqueness of the bitstrings are close to the ideal of 50%.

	0MR N	0MR NN	3MR N	3MR NN	5MR N	5MR NN
Inter HD	50.02%	50.13%	50.14%	50.04%	49.98%	50.05%

Table 1: Inter-Chip HDs under 6 usage scenarios.

Fig. 9 depicts the results of the NIST tests at a significance level of 0.01. The pass criteria for each test is that 47

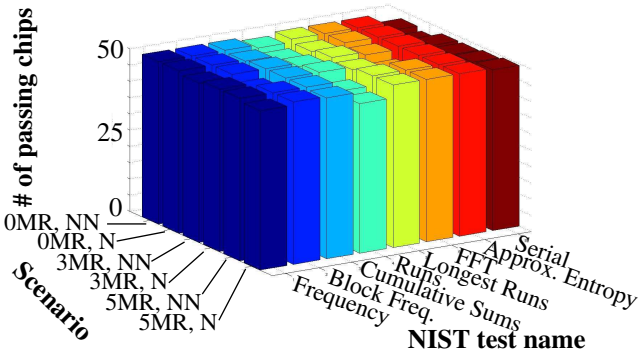

Fig. 9. NIST Test Results

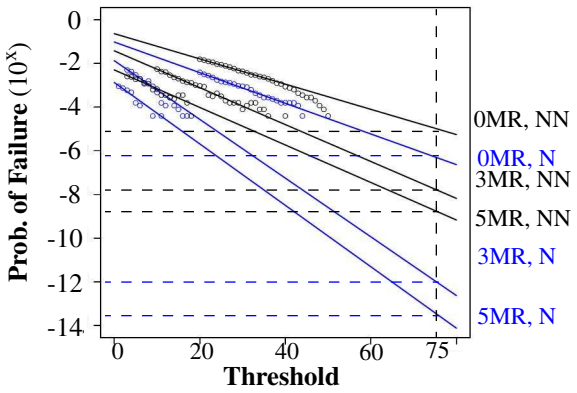

Fig. 10. Probability of Failure results for bitstrings

or more of the chips pass the test individually. All tests are passed except two of the Serial tests, which had 45 and 46 chips passing, and a Runs test with 46 passing. These results indicate that the bitstrings possess a high degree of randomness.

5.2 Intra-chip HD

Without normalization, the average intra-chip HD with thresholding disabled is approx. 8% but drops to 4% with normalization, which demonstrates the reliability enhancement capability of normalization.

5.3 Reliability Analysis

The reliability enhancing techniques are evaluated by computing a **probability of failure** (POF) metric. A failure is counted when a bit flip occurs in one (or more) of the 8 regenerated bitstrings. A POF is computed by dividing the total number of fails that occur across all chips by the total number of bits in the enrollment bitstrings, which is 50 chips * 512-bits = 25,600. Enrollment is carried out at nominal V_{DD}, 25°C and regenerations at the 8 remaining TV corners.

The user-specified threshold parameter is a value between 0 (no threshold) and 256 (large threshold). We found that a threshold of 75 prevents bit flips in all chips under each of the six scenarios. In order to predict the POF at this threshold, a sequence of bitstring generation experiments are carried out using successively smaller thresholds. The set of POFs from these experiments defines a curve that is exponential in shape. The POF curves and exponential line fits for each of the six scenarios are plotted on a $\log_{10}$ scale in Fig. 10. Horizontal dotted lines provide estimates of the POFs at threshold 75, which vary from $1x10^{-5}$ to $4x10^{-14}$.

The progression of the line fits to steeper negative slopes clearly indicates that both XMR and normalization improve reliability. The main trade-off for higher reliability in either case is bitstring generation time. For example, enrollment bitstring generation time for 0MR, N is approx. 4 ms for a bitstring of size 512 and a sample size of 1, and 1 ms for regeneration (using the analog difference described above cuts these times in half). In contrast, this time increases to approx. 180 ms for enrollment and 45 ms for regeneration under 5MR, N with 8 sample averaging.

6 Conclusions

A complete IP-level implementation of a resistance-based PUF engine is used to demonstrate several novel PUF features including the entropy source, band enrollment, normalization and calibration, and to analyze important implementation metrics including bit generation time, power consumption and area overhead. Data from SPICE-level simulation experiments is used to show a high level of statistical quality in the generated bitstring with respect to intra- and inter- chip hamming distances, NIST statistical tests and probability of failure metrics.

References

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

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

[3] M. Kalyanaraman and M. Orshansky, "Novel Strong PUF Based on Nonlinearity of MOSFET Subthreshold Operation", *HOST*, 2013, pp. 13-18.

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

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

[6] Y. Meng-Day, *et al.*, "Performance Metrics and Empirical Results of a PUF Cryptographic Key Generation ASIC," *HOST*, 2012, pp. 108-115.

[7] C. Yin, G. Qu, and Q. Zhou, "Design and Implementation of a Group-Based RO PUF", *DATE*, 2013, pp. 416-421.

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

[9] S. Okumura, S. Yoshimoto, H. Kawaguchi,and M. Yoshimoto, "A Physical Unclonable Function Chip Exploiting Load Transistors' Variation in SRAM Bitcells", *ASP-DAC*, 2013, pp. 22-25.

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

[11] J. Ju, R. Chakraborty, C. Lamech and J. Plusquellic, "Stability Analysis of a Physical Unclonable Function based on Metal Resistance Variations", *HOST*, 2013, pp. 143-150.

[12] R. Chakraborty, C. Lamech, D. Acharyya and J. Plusquellic, "A Transmission Gate Physical Unclonable Function and On-Chip Voltage-to-Digital Conversion Technique", *DAC*, 2013, pp. 1-10.

[13] Y. Yao, M. Kim, J. Li, I. Markov, and F. Koushanfar, "ClockPUF: Physical Unclonable Functions Based on Clock Networks", *DATE*, 2013, pp. 18-22.

[14] Z. Yu, A. Krishna, and S. Bhunia, "ScanPUF: Robust Ultralow-Overhead PUF Using Scan Chain", *ASP-DAC*, 2013, pp. 626-631.

[15] K. Cho, K. Lee, S. Kim, S. Lee, and Y. You, "Implementation of a Physical Unclonable Function (PUF) with Transmission Line Crosstalk in a Chip", *ASQED*, 2013, pp. 240-244.

[16] http://en.wikipedia.org/wiki/Hamming_distance

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