EDFAAO (2022) 4:12-21 1537-0755/$19.00 ©ASM International®

MEMOMETER: MEMORY PUF-BASED HARDWARE METERING METHODOLOGY FOR FPGAs

Anvesh Perumalla and John M. Emmert
Department of Electrical and Computer Engineering, University of Cincinnati, Ohio
john.emmert@uc.edu

INTRODUCTION

Security, assurance, and trust (SA&T) within the integrated circuit (IC) supply chain are of crucial importance to the government and the commercial sector. As the semiconductor business has shifted toward a horizontal ("fabless") model, there is an increasing need to protect against counterfeiting, cloning, overbuilding, and intellectual property (IP) theft and piracy. The U.S. Department of Commerce defined a counterfeit electronic part as: 1) an unauthorized copy, 2) does not conform to original component manufacturer (OCM) design, model, and/or performance standards, 3) is not produced by OCM or is produced by unauthorized contractors, 4) an off-specification, defective, or used OCM product sold as "new" or working, or 5) has incorrect or false markings and/or documentation.[1] A recent 2020 report by the Semiconductor Industry Association (SIA) states that the "United States today now only accounts for 12.5% of total installed semiconductor manufacturing, with more than 80% of production now happening in Asia."[2] This SIA report also shows that state-of-the-art 7-nm and below IC production is happening almost exclusively outside of the United States. This creates an opportunity for untrusted agents and entities to counterfeit and overproduce ICs and place them in the supply chain. This article describes the memometer, a hardware metering technique, which addresses the supply chain integrity of field-programmable gate arrays (FPGAs).

Currently, FPGAs pervade most of the semiconductor ecosystem due to their faster prototyping and time-to-market capabilities when compared to traditional application-specific integrated circuits (ASICs). In the last three decades, the FPGA logic capacity has grown 10,000x and processing speed has grown 100x, and at the same time, the FPGA cost and energy consumption per unit function have reduced over 1000x.[3] As FPGAs have become the predominant choice of circuit realization, SA&T of these

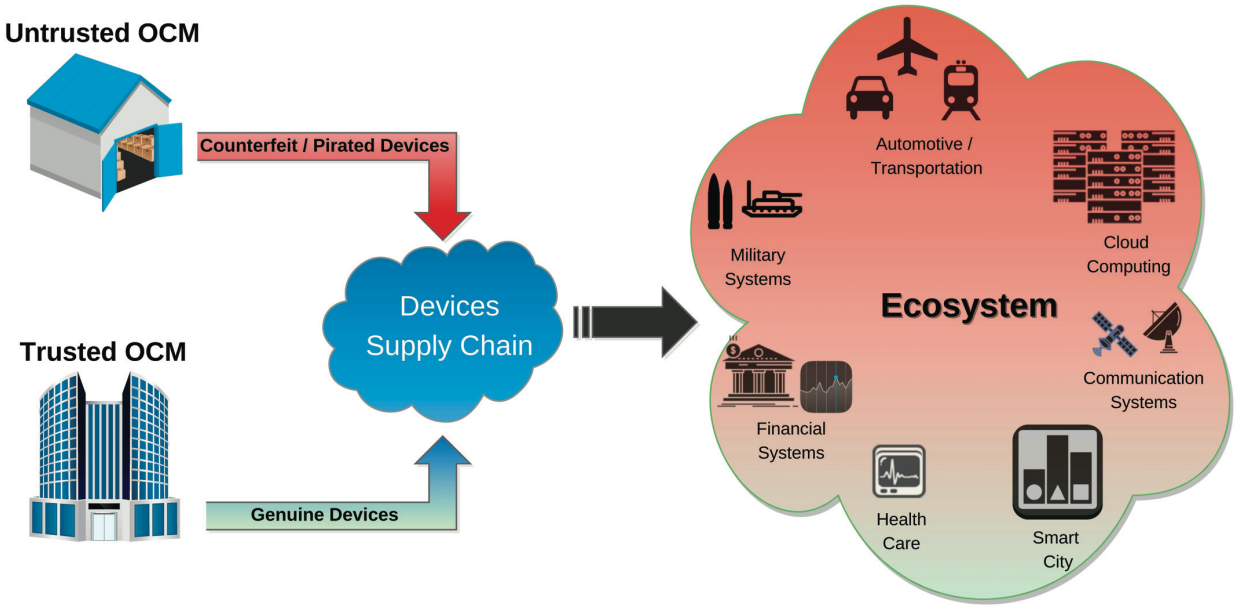

Fig. 1 **An ecosystem affected by untrusted FPGAs.**

devices have become a major concern. Figure 1 exemplifies how counterfeit or illegal FPGAs within the supply chain can end up undetected in our computing systems.

A memometer is suggested to overcome this problem. The memometer is a low-overhead, inexpensive, adaptable hardware metering (fingerprinting) methodology leveraging memory physically unclonable functions (PUFs). Historically, memory PUFs have not been applied to contemporary FPGAs because most of them come with manufacturing memory preset startup values. The authors have overcome this issue by inventing a new memory PUF using cross-coupled look-up tables (LUTs) that imitate the SRAM PUF behavior, thus providing unique start-up values (SUVs) used for fingerprinting each FPGA. These fingerprints are further used in identification and authentication throughout the supply chain.

HARDWARE METERING

Hardware metering helps in identifying authorship of an IC or intellectual property (IP) after fabrication by uniquely locking/tagging each IC that is manufactured under the same mask.[4] Hardware metering is further classified as passive and active metering.[4] Passive metering is used to tag each IC with an unclonable unique identifier. This identifier is further used in recognizing genuine ICs from the overbuilt/counterfeit ICs. Whereas in active metering, in addition to tracking passively, it can also help with enabling/disabling IC functionality and controlling/preventing the ICs from further infiltrating the supply chain.[4] This passive metering methodology can be used to create unique unclonable fingerprints and use them to interrogate ICs within the supply chain. The authors are also leveraging the methodology to actively meter, which is briefly described later.

PHYSICALLY UNCLONABLE FUNCTIONS

Identification and authentication are critical to secure any electronic system. Embedding a unique key can only help identify an IC, but in order to authenticate, a secret key must be embedded onto the IC itself.[5] These secret keys are either stored in nonvolatile memory (NVM) or battery-backed external volatile memory. Both methods not only add additional overhead but are also extremely vulnerable to attackers. A simple side-channel attack[6] can reveal a lot about the IC and allow for the secret key to be stolen, which can be further used in creating clones of those ICs. To overcome this issue, a new authentication mechanism—physically unclonable functions (PUFs)—was invented. PUFs are extremely hard-to-forge, unique to every IC ever manufactured, non-programmed, and low-overhead.[5,7] The basic idea behind a PUF is that each IC exhibits a unique process variation characteristic profile that can be leveraged to create unclonable fingerprints. Even when two ICs are functionally same, the underlying microscopic process variation characteristics are slightly different. When a challenge (input) C_i is applied to a section of an IC, the underlying unique process variation profile in that section exhibits a unique response (output) R_{Ci}.[7]

Uniqueness and reproducibility are the two metrics used to analyze the quality of PUF fingerprints. Uniqueness is measured using inter-chip hamming distance (HD), and reliability or reproducibility is measured using intra-chip HD.[8] Inter-chip HD is the average HD measured between the responses when the same challenge is applied to two different ICs. Ideally, it should be 50%, which means half of the bits from these two fingerprints must be different. This measurement can also be used to analyze two fingerprints obtained from different sections of the same IC. Intra-chip HD is the average HD measured between the responses when the same challenge is applied at different times. Ideally, this should be 0%, which means that each fingerprint must be reproducible or repeatable over time. An example of this illustration is shown in Fig. 2. A programming file (**.bit*) is used as a challenge, applied on different FPGAs, and the response fingerprints are recorded. These responses are used to analyze the uniqueness of these fingerprints. Similarly, a challenge is applied to the same FPGA multiple different times and the responses are used to investigate the repeatability measure of a given fingerprint.

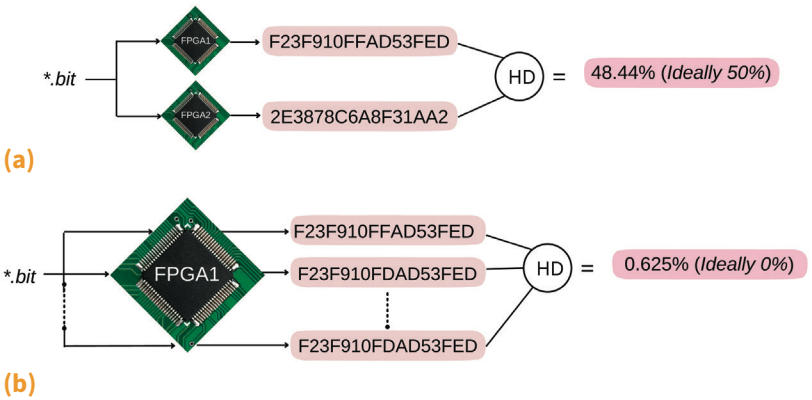

Fig. 2 (a) Uniqueness (inter-chip HD), (b) repeatability/reproducibility (intra-chip HD) of PUF challenge-response pairs.

MEMORY PUFS

Memory components (such as SRAMs and D-FFs) are essential elements in electronic systems. Memory based PUFs use these basic elements to create fingerprints. When a cross-coupled memory structure is manufactured for minimum size, as shown in Fig. 3, the relative drive strength and doping levels are usually balanced. When these devices are powered on, before programming any value, the metastability property of balanced memory elements leads to random start-up values.[9-11] Instability within these cross-coupled components is due to several technological and non-technological parameters, such as probabilistic geometry of transistors, inexact threshold voltages, or channel length modulation.[12] Because of these varying parametric values, some memory cells always power on to a specific state, whether logic "1" or logic "0;" that is, 100% of the time these cells are powered on to the same logic value. However, other cells fluctuate between logic "0" or logic "1" for each power cycle. A fingerprint is created by estimating the most likely power-up state of each memory cell SUV.[10]

MEMORY PUFS FOR CONTEMPORARY FPGAS

One major disadvantage of applying memory PUFs to contemporary FPGAs is that many newer FPGA families come with memory preset. In other words, as soon as the FPGA is powered on, the memory elements within in the FPGA are preset to either logic "1" or logic "0" by default. This makes memory PUFs impractical. One notable effort to overcome this particular problem was the invention of the butterfly PUF (BPUF).[13] The BPUF emulates SRAM behavior at power-up. The BPUF uses built-in FFs configured as cross-coupled latches to emulate memory PUFs, as shown in Fig. 4a. The preset (PRE) signal sets the output of a latch high, and the clear (CLR) signal sets the output low. The BPUF operation starts when the excite signal—connected to the PRE of one latch and CLR of another latch—is set to high for a few clock cycles and brought to low. The BPUF will settle to either logic "0" or logic "1" at the output. The output SUVs are based on the intrinsic characteristics of the FPGA.

A key challenge with this implementation is that the quality of SUVs will purely depend on the symmetric construction of the BPUF cell. As shown in Fig. 4a, the red and green routing paths must be routed identically. In other words, the delay difference between the signals should be equal down to picoseconds. Unlike ASICs, FPGA routing paths are not easily accessible to the designer. Even though FFs are readily available on almost all FPGAs, the symmetric construction of a BPUF makes it challenging to implement for different FPGA architectures. A similar research study concluded that a BPUF is not an ideal candidate for an FPGA.[14] This has led to the invention of the memometer PUF, which is a much simpler implementation of emulating SRAM start-up behavior using LUTs. Table 1 shows the major differences between the BPUF and the memometer PUF.

MEMOMETER

The memometer PUF is implemented by mapping cross-coupled NAND gates to cross-coupled LUTs, as

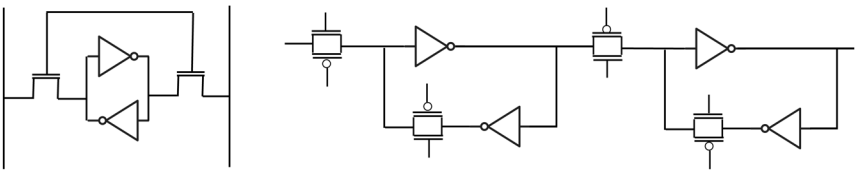

Fig. 3 Common cross-coupled memory structures: SRAM and D-FF.

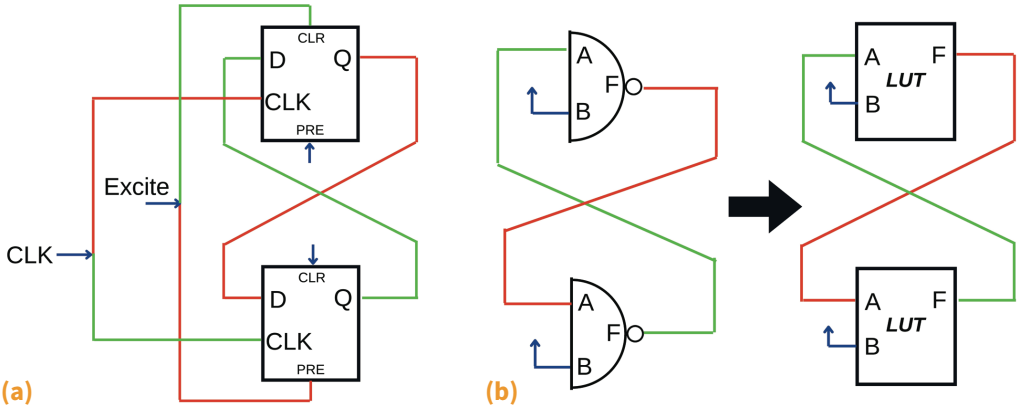

Fig. 4 PUFs for contemporary FPGAs that imitate SRAM PUF.

Table 1 Butterfly PUF compared to memometer PUF

Butterfly PUF	Memometer PUF
Uses cross-coupled latches	Uses cross-coupled LUTs
Complex implementation and routing	Simple implementation and routing
Difficult to balance feedback path delays	Easy to balance feedback path delays
Requires three sets of paths to be symmetric for ideal SUVs Path 1: Global clock to clock pin (clock skew) Path 2: Excite signal to CLR/PRE Path 3: Feedback path delay (latch 1 output Q -> latch 2 input D)	Requires only one path to be balanced Feedback path delay (LUT 1 output F -> LUT 2 input A)
Requires external signal to settle into an unstable state for start-up behavior	Does not require any external signals for start-up behavior
One challenge-response pair	Hundreds of challenge-response pairs

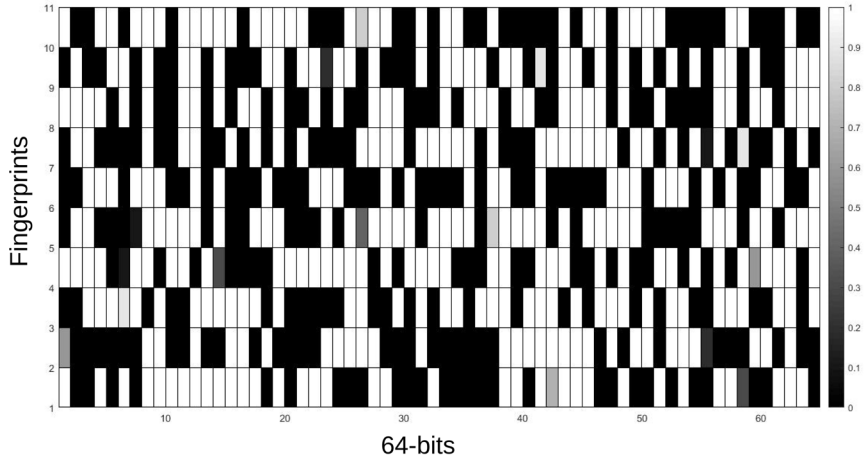

Fig. 5 The probability analysis of a 64-bit memory signature powering up to 1 on ten FPGAs.

shown in Fig. 4b. One of the challenges of this design is balancing feedback path delays. If the feedback routing paths are not matched, then the memory element produces known values instead of random values. In previous works, "implementing a cross-couple element using combinational logic on an FPGA was not straight forward due to inability to create combinational loops;"[13] however, the authors have been successful implementing cross-coupled combinational elements using FPGA LUTs and balancing the feedback path delays. Memory PUF research shows that 64 bits are enough to differentiate between all existing ICs (2^{64} unique signatures).[10] To demonstrate the approach, the authors programmed 64 of these memometer PUF elements on ten Xilinx Artix-7 FPGAs. The same programming (**.bit*) file was used to program all ten FPGAs, and each FPGA gave a unique 64-bit memory signature. Figure 5 shows the probability analysis for fingerprint bits powering up to logic "1" values, where each row corresponds to a unique 64-bit fingerprint from a different FPGA.

Figure 6 shows the probability distribution of the inter-chip and intra-chip HD of these 64-bit fingerprints for ten power cycles on all ten FPGAs. The x-axis represents the percentage of PUF output bits that are different from one FPGA to another for inter-chip HD, and PUF output bits that are changing over time for the intra-chip HD. An average inter-chip HD of 49.7% (vs. an ideal HD of 50%) and intra-chip HD of 0.88% (vs. an ideal HD of 0%) was achieved. These values demonstrate fingerprint uniqueness and reproducibility.

JOURNEY TOWARD A STRONGER PUF

PUFs are generally categorized as weak or strong.[7] A weak PUF contains a limited number of challenge-response ($C_i R_{Ci}$) pairs, whereas a strong PUF contains a large number.[15] Both SRAM and butterfly PUFs are categorized as intrinsic weak PUFs[15] due to their fixed number of ($C_i R_{Ci}$) pairs—most cases typically have one challenge at power-up. Weak PUFs are mainly used in cryptographic systems where a secret key is derived from the PUF response with the help of error-correction codes. On the other hand, a strong PUF not only contains many ($C_i R_{Ci}$) pairs, but also makes it difficult for an adversary to predict the next response.[16] The approach of creating a large number of ($C_i R_{Ci}$) pairs is similar to a reflective PUF or optical PUF.[7] For example, reflective PUFs are used in identifying missiles: a light scattering particle is sprayed onto the missile and an inspector records the images of this particle by illuminating it at different angles. Each angle of incidence gives a unique response, which is recorded and stored in a secure database. For authentication, a random angle

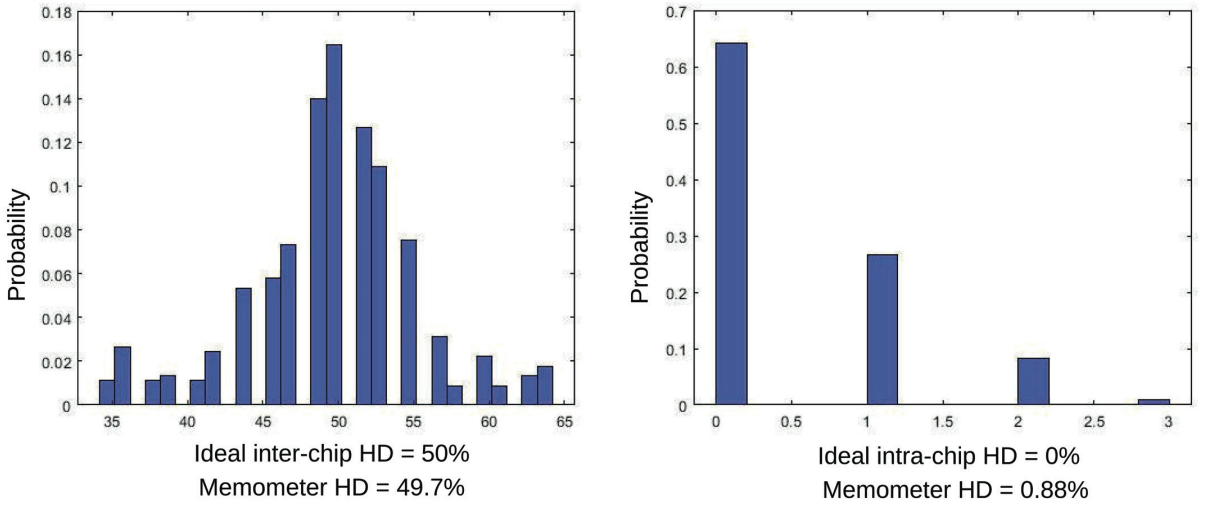

Fig. 6 **Inter-chip and intra-chip HD of a 64-bit memory signature on ten Xilinx FPGAs.**

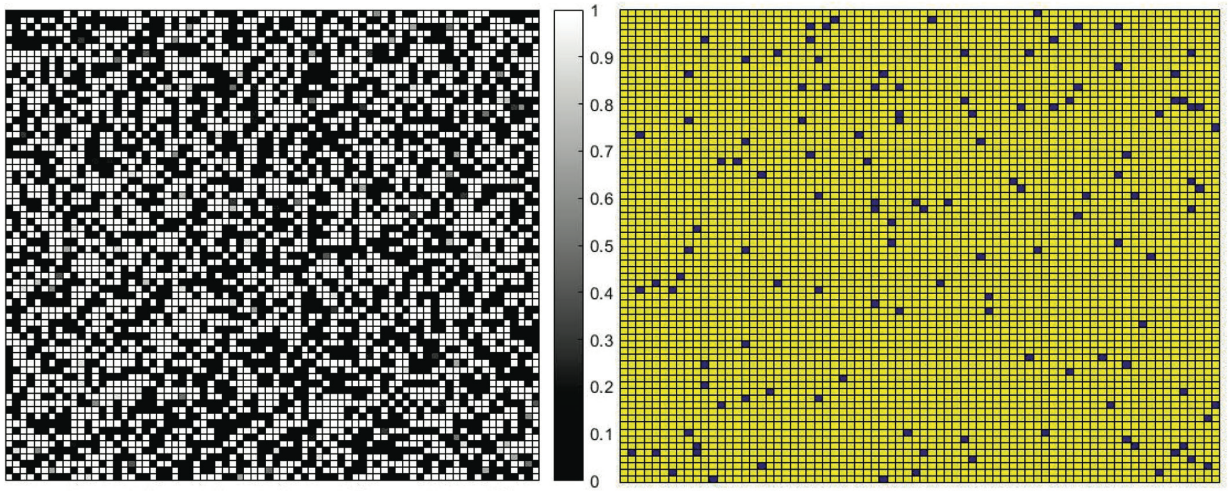

Fig. 7 **The probabilistic analysis of memometer PUF start-up values powering up to 1 on a Xilinx FPGA; stable values (yellow) and unstable values (dark blue).**

of incidence interference pattern is applied, and the response is compared against the known database of images. Using this technique, an optical PUF or reflective PUF creates a large database of (C_i R_{Ci}) pairs. The same concept is applied to the memometer PUF. The authors mapped 5180 memometer PUF cells on ten Xilinx FPGAs. On average, 97.08% of the bits were stable. Figure 7 shows the probabilistic analysis of these memory SUVs powering up to logic "1" for ten power cycles on an FPGA. It also shows the stable values in yellow and unstable values in dark blue. For this particular FPGA, out of 5180 SUVs, there are 5056 stable values (usable in fingerprints). These stable values are consistently logic "1" and logic "0" for each power cycle.

Different permutations of LUTs are used to increase the number of (C_i R_{Ci}) pairs. FPGAs give the flexibility to map different combinations of LUTs to form a single cross-coupled pair memory cell, and thus increase the total number of unique challenge-response pairs by an order of magnitude. To validate the hypothesis, a million different permutations of 5180 different SUVs in the LUT based memory values as (C_i R_{Ci}) pairs were programmed.

Table 2 shows the average inter-chip and intra-chip HD of different 64-bit signatures taken using the million tested combinations of different challenge-response pairs. An average of 48.99% inter-chip HD from the million pairs was achieved. In Table 2, an average inter-chip HD value for each fingerprint is between 42.94% and 53%. An average of 25.65 bits out of 64 bits are different with an upper and lower bound of 27.75 and 22.48 bits respectively. Similarly, an average of 1.05% intra-chip HD was achieved. An average of 0.55 bits out of 64-bits are changing over time with an upper and lower bound of 1.39 and 0.049 bits.

Figure 8a shows the average HD percentage and Fig. 8b shows the average HD values for the million (C_i R_{Ci}) pairs simulated from ten different FPGAs for ten power

Table 2 Average inter-chip and intra-chip HD of different 64-bit signatures permutated using a million challenge-response pairs on 10 FPGAs

Million 64-bit challenge-response pairs	Inter-chip HD (Ideal 50%)	Average bits different	Intra-chip HD (Ideal 0%)	Average bits changing
Average	48.9979	25.6571	1.0514	0.5505
Max	53	27.7527	2.6667	1.3964
Min	42.94	22.4873	0.0938	0.0491

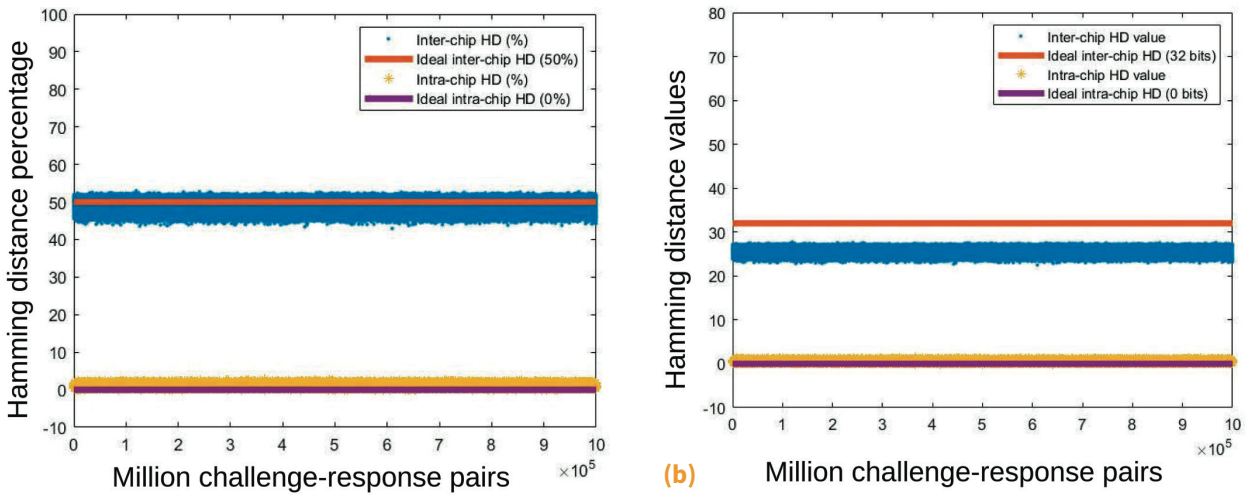

Fig. 8 (a) Average HD percentage of a million challenge-response pairs from ten FPGAs. (b) Average HD values of a million 64-bit challenge-response pairs from ten FPGAs.

cycles. In both Fig. 8a and b, ideal inter-chip and intra-chip HD is shown using straight lines. It should be noted that all million challenge-response values are close to the ideal HD values.

Merely increasing the number of challenge-response pairs by different permutations doesn't necessarily mean this method leads to a strong PUF. To increase the strength of the PUF, the authors are currently researching and extending this method to use different LUT routing strategies. A six input LUT has 64 memory cells and this method only uses two inputs out of those six to map the memory PUF. The hypothesis is that different combinations of these input configuration should give different SUVs because of the underlying FPGA fabric routing changes. Key to this strategy is making sure the feedback path delays are balanced, otherwise it will result in an unbiased fingerprint. This strategy should not only exponentially increase the challenge-response pairs but also make it difficult for an adversary to predict the next pair.

ARTIFICIAL AGING EXPERIMENT

The aging experiment used five Xilinx Zedboards that have a Zynq 7000 processor with Artix-7 FPGA fabric. The processor's operating temperature was between 0 to 85°C.[17,18] In order to artificially age the circuit boards, they were placed in a temperature-controlled chamber at a desired temperature and voltage, using the age acceleration factor:[19,20]

$$AF = \left(\frac{V_{stress}}{V_{nominal}}\right)^{\frac{\alpha}{n}} exp\left(\left(\frac{E_{aa}}{k}\right).\left(\frac{1}{T_{stress}} - \frac{1}{T_{nominal}}\right).\frac{1}{n}\right)$$

The parameters used for the test were based on a similar study:[19] gate voltage exponent, α = 3.5; time exponent, n = 0.25; activation energy, E_{aa} = -0.02 eV; Boltzmann's constant, k = 8.62 x 10^{-5} eV/K; nominal voltage, $V_{nominal}$ = 1.8V; nominal temperature, $T_{nominal}$ = 23°C; higher stress voltage, V_{stress} = 2.5V; and higher stress temperature, T_{stress} = 80°C. After applying these parameters, we calculated an aging factor, AF = 163.99. This means one hour of accelerated aging gave 163.99 hours of estimated aging, which was approximately one week. Using AF = 163.99 in a temperature-controlled chamber for 255 hours resulted in approximately five years of artificial aging. For a more realistic analysis, these FPGA LUTs were programmed during the aging process. These circuit boards were taken out of the temperature-controlled chamber every 1, 2, 4, 8, 16, 32, 64, and 128 hours to measure the fingerprints at nominal conditions. For each aging cycle, we acquired

(continued on page 20)

MEMOMETER: MEMORY PUF-BASED HARDWARE METERING METHODOLOGY *(continued from page 17)*

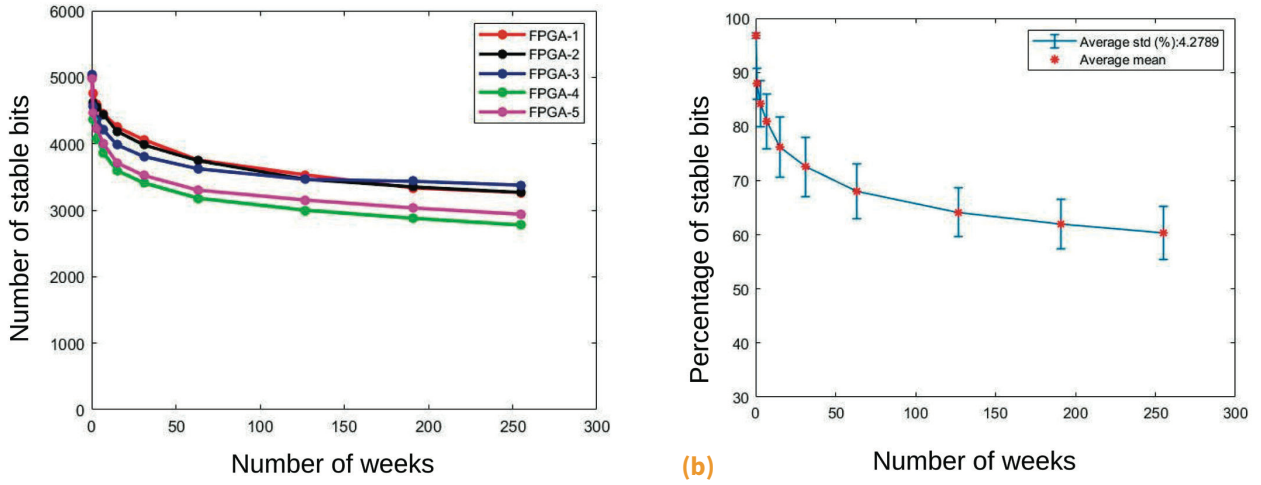

Fig. 9 (a) Number of stable SUVs as the FPGA ages (b) average percentage of stable bits as the FPGA ages.

Table 3 Artificial aging analysis of memometer PUF on five FPGAs for five years

Artificial age (# of weeks)	Number of stable bits to a total of 5180				
	FPGA1	FPGA2	FPGA3	FPGA4	FPGA5
0	5032	5038	5040	4990	4984
1	4755	4626	4557	4371	4465
3	4596	4549	4373	4072	4231
7	4457	4439	4207	3863	4005
15	4251	4189	3984	3596	3711
31	4061	3982	3808	3411	3523
63	3757	3747	3624	3182	3303
127	3532	3466	3463	3000	3153
191	3336	3351	3435	2879	3034
255	3265	3269	3377	2777	2939

a set of ten SUVs. We set the temperature-controlled chamber to 88°C ± 11°C.

Figure 9a shows the degradation plot of the number of stable bits as the IC was aged, and Table 3 shows the stable bit values as the IC was artificially aged. As the IC aged, all SUVs for the five FPGAs were consistently degrading. Figure 9b shows the average percentage of the stable bits as the IC aged. The average number of stable bits for five FPGAs at week-0 was 96.84%, or 5016.31 stable bits. After five years of artificial aging, which was 255 weeks, the average number of stable bits dropped to 60.336%, or 3125.405 stable bits. The average standard deviation of these stable values for all five years of artificial aging was 4.279%. It should be noted that there were enough stable bits available in each IC after artificial aging to create fingerprints.

FUTURE WORK

The authors are exploring different LUT configurations to make the memometer PUF a stronger PUF. This includes performing different temperature measurements to analyze the PUF start-up behavior at various temperatures, and adding error correction to the fingerprints for more robustness over time and under harsh environmental conditions.

The current methodology can create unique unclonable fingerprints for every FPGA (legacy or contemporary). These fingerprints are further used for authentication within the supply chain. The methodology is being extended to not only passively track these FPGAs but to actively control the ICs by preventing them from further entering the supply chain. Methods that can destroy these ICs when proven untrusted, erase IP and prevent access to the IP are also being explored.

CONCLUSION

The memometer is a practical hardware metering fingerprint methodology for both legacy and contemporary FPGAs. The authors have developed a new PUF based on cross-coupled LUTs that can overcome manufacturing memory power-on preset. The fingerprints are not only unique but also reliable with average inter-chip and intra-chip HDs close to the ideal 50% and 0%. Instead of having one fingerprint per device, this methodology makes provision for hundreds of fingerprints. There are plans to extend the methodology to make it a stronger PUF, thus making it difficult for an adversary to reverse engineer or clone the fingerprints.

REFERENCES

1. U.S. Department of Commerce, "Defense Industrial Base Assessment: Counterfeit Electronics," January 2010, https://www.bis.doc.gov/index.php/documents/technology-evaluation/37-defense-industrial-base-assessment-of-counterfeit-electronics-2010/file, p. 2-5.
2. Semiconductor Industry Association, "State of the U.S. Semiconductor Industry," July 2020, https://www.semiconductors.org/wp-content/uploads/2020/07/2020-SIA-State-of-the-Industry-Report-FINAL-1.pdf, p. 9-15.
3. S.M.S. Trimberger: "Three Ages of FPGAs: A Retrospective on the First Thirty Years of FPGA Technology: This Paper Reflects on How Moore's Law Has Driven the Design of FPGAs Through Three Epochs: The Age of Invention, the Age of Expansion, and the Age of Accumulation," *IEEE Solid-State Circuits Magazine,* 2018, Vol. 10, No. 2, p. 16-29.
4. F. Koushanfar: "Hardware Metering: A Survey," ed. by M. Tehranipoor and C. Wang, *Springer,* 2012.
5. B. Gassend, et al.: "Silicon Physical Random Functions," *Proceedings of the 9th ACM Conference on Computer and Communications Security,* 2002, p. 148-160.
6. P. Kocher, J. Jaffe, and B. Jun: "Differential Power Analysis," *Advances in Cryptology — CRYPTO' 99,* ed. by Michael Wiener, *Springer Berlin Heidelberg,* 1999, p. 388–397.
7. U. Rührmair, S. Devadas, and F. Koushanfar: "Security Based on Physical Unclonability and Disorder," ed. by Mohammad Tehranipoor and Cliff Wang, *Springer,* 2012.
8. M. Rostami, F. Koushanfar, and R. Karri: "A Primer on Hardware Security: Models, Methods, and Metrics," *Proceedings of the IEEE,* 2014, p. 1283–1295.
9. J. Guajardo, et al.: "FPGA Intrinsic PUFs and Their Use for IP Protection," *2007 CHES Cryptographic Hardware and Embedded Systems,* Springer Berlin Heidelberg, p. 63–80.
10. D.E. Holcomb, W.P. Burleson, and K. Fu: "Power-Up SRAM State as and Identifying Fingerprint and Source of True Random Numbers," *IEEE Transactions on Computers,* 2009, Vol. 58, No. 9, p. 1198-1210.
11. V. Leest, et al.: "Hardware Intrinsic Security from D Flip-flops," *Proceedings of the Fifth ACM Workshop on Scalable Trusted Computing,* 2010, Association for Computing Machinery, New York, p. 53–62.
12. M. Cortez, et al.: "Modeling SRAM Start-up Behavior for Physical Unclonable Functions," *IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT),* 2012, Austin, TX, p. 1-6.
13. S.S. Kumar, et al.: "The Butterfly PUF Protecting IP on every FPGA," *IEEE International Workshop on Hardware-Oriented Security and Trust,* Anaheim, CA, 2008, p. 67-70.
14. S. Morozov, A. Maiti, and P. Schaumont: "An Analysis of Delay Based PUF Implementations on FPGA," *Proceedings of the 6th International Conference on Reconfigurable Computing: Architectures, Tools and Applications (ARC'10),* 2010, Springer-Verlag, Berlin, Heidelberg, p. 382-387.
15. U. Rührmair and D.E. Holcomb: "PUFs at a Glance," *Design, Automation & Test in Europe Conference & Exhibition (DATE),* 2014, p. 1-6.
16. U. Rührmair, H. Busch, S. Katzenbeisser: "Strong PUFs: Models, Constructions, and Security Proofs," eds. A.R. Sadeghi and D. Naccache, *Towards Hardware-Intrinsic Security, Information Security and Cryptography,* 2010, Springer, Berlin, Heidelberg.
17. Xilinx, "XA Artix-7 FPGAs Data Sheet: Overview (DS197)," 2017.
18. Xilinx, "Xilinx Zynq-7000 SoC Data Sheet: Overview (DS190)," 2018.
19. R. Maes and V. Leest: "Countering the Effects of Silicon Aging on SRAM PUFs," *IEEE International Symposium on Hardware-Oriented Security and Trust (HOST),* 2014, p. 148–153.
20. JEDEC Solid State Technology Association: "Failure Mechanisms and Models for Semiconductor Devices," *JEP122E,* p. 16-18.

ABOUT THE AUTHORS

John (Marty) Emmert received a bachelor of science degree in electrical engineering from the University of Kentucky, his master's degree in electrical engineering from the Air Force Institute of Technology, and his Ph.D. in computer science and engineering from the University of Cincinnati. He is currently a professor with the Department of Electrical and Computer Engineering, University of Cincinnati, and the Director of the NSF Center for Hardware and Embedded Systems Security and Trust (CHEST) I/UCRC. He is also a graduate of the Air War College and a retired Colonel from the U.S. Air Force Reserves.

Anvesh Perumalla received his master's degree in electrical engineering from Wright State University and a Ph.D. in computer engineering from the University of Cincinnati. He is currently a post-doctoral researcher with the Department of Electrical and Computer Engineering, University of Cincinnati. His current research focus is on hardware security topics, such as physically unclonable functions, counterfeit IC detection, FPGA reverse engineering, and asynchronous circuit design methodologies.