Enabling SRAM-PUFs on Xilinx FPGAs

Alexander Wild
Horst Görtz Institute for IT Security
Ruhr University Bochum
Germany
Email: alexander.wild@rub.de

Tim Güneysu
Horst Görtz Institute for IT Security
Ruhr University Bochum
Germany
Email: tim.gueneysu@rub.de

Abstract—**Physically Unclonable Functions (PUFs) based on the evaluation of uninitialized SRAM are one of the most promising PUF candidates to date. However, transferring their concept to Xilinx FPGAs is not straightforward since all SRAM-based block memories in these FPGAs are automatically cleared on power-up, destroying the desired initial bits of information. In this work we therefore propose a novel strategy to convert block memories of 28nm Xilinx FPGAs into SRAM-PUFs by exploiting their recently introduced feature of power-gating and partial reconfiguration.**

I. Introduction

Physical unclonable functions (PUFs) gained a lot of attention the last years. They provide a device-specific response $r = \mathrm{PUF}(c)$ [1] to a given challenge c, e.g., for the use in secure challenge-and-response protocols and secure key generation and storage. A large number of PUF variants have been proposed, for example, PUFs based on signal propagation delays [2], capacitive coatings [3], butterfly circuits [4], [5], sense-amplifiers [6], write collisions in block memories [7], or initial bit configurations in asynchronous memories [8], [9]. All PUFs are required to be reliable, implying that the returned response can be considered deterministic and consistent across environmental conditions (e.g., temperature and voltage) and other influences (e.g., time and aging effects).

Despite the wealth of PUF constructions, an open research question is which of them provides the best properties for use in real-world applications. The straightforward design of Static Random-Access Memory (SRAM)-based PUFs which simply evaluate initial states of memory cells have been identified as promising and have already been emerged to IP cores for integration into ASICs. However, transferring this concept to Field Programmable Gate Arrays (FPGAs) – at least for those of the market leader Xilinx – is not possible since all SRAM-based block memories (BRAM) in Xilinx FPGAs are automatically cleared on power-up [10], destroying the desired initial information bits. Due to this startup procedure, it is therefore widely assumed that it is not possible to implement SRAM-PUFs in Xilinx FPGAs.

Our Contribution: In this work, we present a novel way to implement SRAM-PUFs on Xilinx FPGAs that overcomes the initial clearing of all SRAM-based memory blocks during system startup. In the latest devices manufactured at 28nm technology, Xilinx has added the ability to shut off unused SRAM memory blocks to reduce the power consumption to a minimum. We exploit the availability of these power gates to switch off a block memory while in operation and re-enabling it afterwards *without* explicit initialization. The deactivation and subsequent activation procedure can be done by partial reconfiguration of corresponding parts of the device. Since only small parts need to be replaced, this can be easily done internally by using the integrated ICAP or PCAP interface. Regaining a portion of unitialized SRAM block in Xilinx FPGAs, we demonstrate that the straightforward construction of SRAM-PUFs is indeed possible with the latest generation of Xilinx FPGA.

Related Work: A significant amount of related work has been published on physically unclonable functions in the last decade. In addition to publications on a range of constructions [1], [2], [4]–[9], attacks [11]–[13] and PUF-based protocols have been proposed [8], [14]. All individual studies done in research papers are complemented by text books [15], [16] that also introduce the concept and implementation of SRAM-PUFs which are of high relevance for this work.

Outline: In Section II we describe our strategy to bypass the autonomously triggered resets and how to include corresponding changes into the bitstream. Section III presents our measurement setup that is used to evaluate the quality of the PUF construction. Finally, the paper is concluded in Section IV.

II. Concept of an SRAM-PUF on Xilinx FPGAs

A Static Random-Access Memory (SRAM) cell is a common volatile memory technology and loses its content shortly after power-down. The SRAM cell achieves the memory effect by the two cross-coupled inverters. The inverter circuit has two stable states (bistable). The state represents the value stored in the SRAM cell. The state after power-up is determined by the switching speed of the transistors of the inverters. In theory, the transistors of both inverters are equally build and very balanced so that the initial state is unpredictable. But in practice the SRAM cells show a bias due to process variations. So the unpredictable initial state of biased SRAM cells is used as PUF value.

The fundamental obstacle Xilinx FPGAs have – from the SRAM-PUF point of view – is the global power-up reset and the autonomously triggered initialization of BRAM cells. To enable an SRAM-PUF an efficient work-around for the reset has to be found. For the work-around we make use of the power-gating feature in Xilinx FPGAs that enables to disconnect uninstantiated Block Random-Access Memory (BRAM) from the power supply network. This feature become available with the 7 series [17]. In combination with partial reconfiguration, this feature is exploited to disconnect and connect BRAM block to the power network and reveal the initial state of their SRAM cells.

An evaluation setup to bypass memory initialization can be described in five steps.

1) The FPGA is powered and the power-up reset is triggered.
2) A bitstream, running a design to read BRAM content, is loaded into the FPGA.
3) A partial bitstream with BRAM instances and initialization values is loaded into the FPGA. This process preloads the BRAMs with initialization values of our choice. The effort of this step can also be integrated into the former and is just separated to reset the BRAMs in a repetitive measurement setting.
4) A second partial bitstream is loaded into the FPGA which disable BRAMs and disconnect them from the power network.
5) A manipulated third partial bitstream is loaded into the FPGA with instantiated BRAMs but without initial values to skip the initialization. Loading this bitstream reconnect the BRAMs to the power network but does not overwrite any content currently contained in the memory. At this point in time the PUF responses are available in the BRAMs.

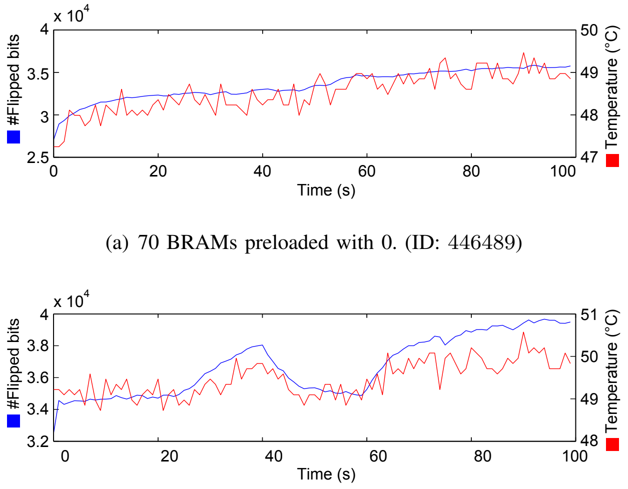
After power-up of the FPGA, the steps 3-5 could potentially be used to create an SRAM-PUF and also repeated without any intermediate hardware reset of the device. We refer to steps 3-5 in the remainder of this work as reconfiguration loop.

Next we briefly describe how we modified configuration bitstreams to prevent the initialization of SRAM block memories. All Xilinx bitstreams are organized in frames. Each frame belongs to a special register to which it is written during the FPGA configuration. Relevant registers for this work are Frame Data Register, Input Register (FDRI) that contains the configuration frames and the Frame Address Register (FAR) to store the frame addresses mapping configuration data to a specific place on the FPGA. Addresses are typically not explicitly included in the frame format and therefore implicitly incremented. As explained in [18], initial BRAM values are placed in separate frames at the end of an FDRI block. To remove these initial values from a configuration bitstream, we simply cut out the corresponding frames with the initialization data from the FDRI data block.

III. Analysis

To verify our concept in practice, we used measurement setup based on a ZedBoard. We connected it to a RS232 interface to receive PUF responses from the Zynq 7020 FPGA and configured the ARM processor via Joint Test Action Group (JTAG). All partial bitstreams are loaded via JTAG into the Double Data Rate (DDR) memory of the ARM processor so that the reconfiguration of the FPGA can be performed rapidly using its internal Processor Configuration Access Port (PCAP) interface.

To evaluate the quality of a PUF class, both *reliability* and *uniqueness* of the responses are essential quality criteria. Further relevant PUF metrics are *unpredictability*, *physical* and *mathematical Unclonability*. We now provide results for the proposed SRAM-based PUF on Xilinx Zynq FPGAs.



(a) 70 BRAMs preloaded with 0. (ID: 446489)

(b) 70 BRAMs preloaded with 1. (ID: 446489)

Fig. 1. The amount of SRAM cells changing their state, based on the time the BRAM was disconnected from the power network.

A. Memory Decay and Flip Direction

As depicted in Section II, the BRAMs under test are always initialized with a given value in the first place. Skorobogatov has shown in his work [19] that shortly after a power-down the SRAM content is still available. This effect is known as *remanescence* of memory cells. To evaluate the influence of this effect on the BRAM we performed the following test. BRAMs are initialized either with 0 or 1 in as described above. A complete reconfiguration loop consists of steps 3-5 to produce an output for measurement. For subsequent measurements all SRAM cells are reset to their initial state. After extracting the output via RS232, all flipped SRAM cells are identified. Figure 1 shows the result for 70× 32K-BRAM on a Zynq 7020 (revision 0). We repeated our experiments with increasing waiting times between reconfiguration cycles as given in Figure 1. Interestingly, we did not see any impact of the remanescence effect. Rather more, it shows that the number of SRAM cells changing their state is correlated to the temperature of the device. We finally conclude that the SRAM cells are more affected by the environmental temperature than the time the BRAM was turned on and off.

Another aspect that is clear to see in Figure 1 is that only about 1.4% of the SRAM cells changed their content while other devices of the same revision shows other switching probabilities. The switching probabilities for each tested device can be found in Table I. This value strongly differ between tested Zynqs of the same and different revisions. Devices of later revisions (1 and 2) come up with very little activity even with long turn off times of the BRAM, so that the remainder of the analysis focuses on devices of revision 0. The reduction of content losing SRAM cells of later revisions is not unusual. A common reason for the changed behavior are improved manufacturing processes.

Comparing the results of Figure 1(a) and 1(b) and taking

TABLE I. FLIP PROBABILITY FOR THE DEVICES UNDER TEST

ZedBoard ID	Revision	% $\text{Flips}_{\text{Init=0}}$	% $\text{Flips}_{\text{Init=1}}$
446435	0	8.1287%	$7,9526\%$
446489	0	1.6688%	1.3839%
447633	0	1.4867%	1.5487%
471018	1	0.0078%	0.0040%
492968	2	0.0005%	0.0019%

the high correlation between temperature and the number of flipped cells into count, both initial values shows approximately the same amount of SRAM cells that changed their content. In a more detailed analysis the addresses of affected SRAM cells of both initial states are extracted and compared to each other. The address comparison shows just a very little overlap. This concludes that the SRAM cells keep or switch their content base on their initial state. Hence, the gathered entropy per BRAM can be increased by evaluating the SRAM cells for both initial values.

B. Unpredictability of SRAM cells

To shows the unpredictability of the SRAM-based Xilinx PUFs the addresses of switching SRAM cells must be independent from other cells of the same and different devices.

To verify the independence of SRAM cells on the same device, the BRAMs where a bit toggle occur on the same address are counted. The result for transitions from 0 to 1 is listed in Histogram 2(a). In case of a total independence an almost horizontal line for the bar edges is expected. Figure 2(a) shows that with exception of the 16 highest and 16 lowest addresses, the switches are almost evenly distributed. With a switching probability p for each bit and x representing the counted flips for one address of 70 BRAM blocks, the given *Binomial Distribution* $f_{bino,intra}$ is expected. Figure 2(b) shows the measured distribution for the bit addresses and the Binomial Distribution given by the red line.

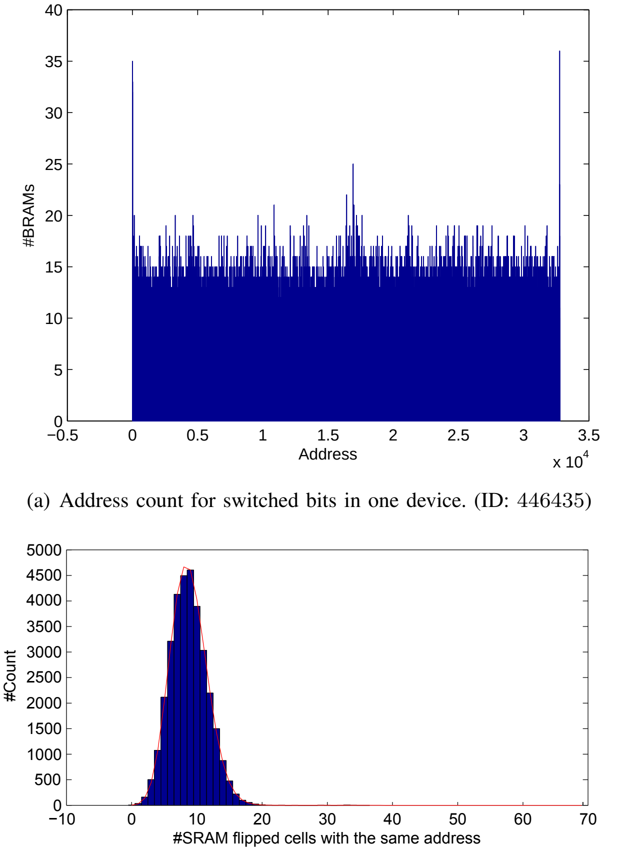
$$f_{bino,intra}(x,70,p) = \binom{70}{x} \cdot p^x \cdot (1-p)^{70-x} \cdot 32 \cdot 1024_{\text{for } 1 \leq x \leq 70}$$

The almost equal distribution of flipped bits over the address space in each BRAM and the high similarity to the expected Binomial Distribution concludes that the SRAM cells flip independently of each other in a device.

The exceptions are the 16 highest and 16 lowest addresses. The transitions from 0 to 1 occur on some of these addresses with a probability of about 50% which seems to be ideal. Further experiments have shown, however, that the inter-distance of these bits is still unbalanced making them for some portions of the RAM predictable. Therefore, such SRAM cells have to be handled carefully or better completely removed from the PUF response.

To show the independence of SRAM cells on different devices, the switching characteristics for each BRAM block and address on 3 devices are compared. More precisely, for each address and BRAM the flips of 3 devices are counted. Ideally, for a switching probability p of an address, the given *Binomial Distribution* $f_{bino,inter}$ is expected, if the switches for each address in each BRAM is counted separately.

$$f_{bino,inter}(x,3,p) = \binom{3}{x} \cdot p^x \cdot (1-p)^{3-x}_{\text{for } 0 \leq x \leq 3}$$



(a) Address count for switched bits in one device. (ID: 446435)

(b) Binomial Distribution for 70× 32K-BRAM in one device. (ID: 446435)

Fig. 2. The distribution of the SRAM cells changing their state (1V, 23°C).

The analyzed measurement result fits exactly into the statistical distribution for the switching behavior of 3 devices. This concludes that SRAM cells on different devices show an independent switching characteristic and hence the BRAM-PUF acts device independently.

C. Reliability

A common metric to evaluate the reliability of SRAM-PUFs is the intra-distance. The intra-distance is the distance between two responses of the same challenge and PUF instance. The distance can be any well-defined distance metric. We use the fractional *Hamming distance* in this paper. The ideal intra-distance for a PUF class is zero. The behavior of a lot of PUF classes is influenced by environmental conditions. Therefore, the intra-distance tests are performed under different environmental conditions. More precisely, the temperature and supply voltage under which the PUF operates are changed. To get a more reliable result, the PUF responses are generated 100 times for each environmental aspect and set in relation to a reference value under nominal conditions. Figure 3 shows the result of the intra-distance test. Beside the mean value, the standard derivation is given for each condition. It is clear to see that under nominal and stable environmental conditions, the behavior of the SRAM cells shows an acceptable reliability. The temperature as well as the supply voltage highly influence the cells which results in noisy PUF responses.

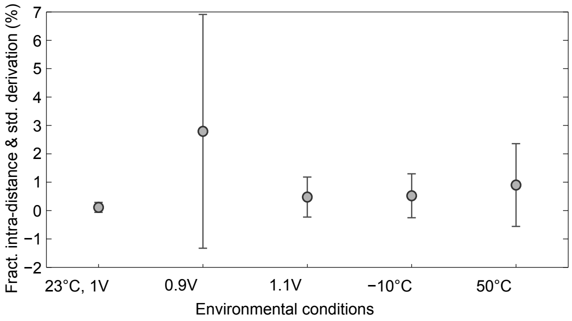


Fig. 3. The mean fractional intra-distance as well as the standard derivation for different environmental conditions. Note that the device under test (ID: 446489) shows a little flip-rate of about 1.5%.

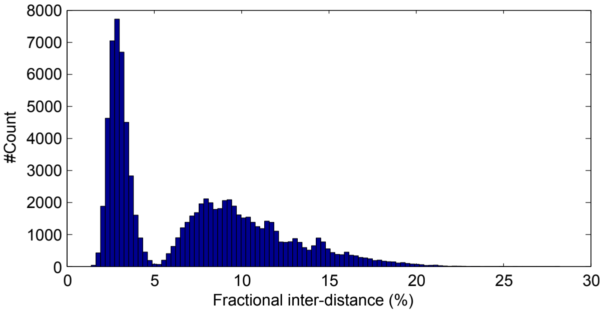


Fig. 4. The inter-distance between 3 devices of revision 0 and 70 BRAMs.

D. Uniqueness

Beside reliability, the uniqueness is also one of the most important properties of a PUF class. To evaluate the uniqueness, the inter-distance is the usual metric. The inter-distance is defined as the distance between two responses for the same challenge produced by different PUF instances. As distance metric we also use the fractional *Hamming distance*. The histogram for the inter-distance is given in Figure 4. Ideally, the inter-distance is distributed around 50%. As stated before, a lot of SRAM cells in the BRAMs keep their initial state. So, many SRAM cells of compared BRAM blocks come up with the same value which lowers the inter-distance between responses. In general a Binomial Distribution is expected for the inter-distance. Unlike this expectation, Figure 4 shows two binomial like distributions. This is due to the strong variation of the flip probability (Table I) and the less amount of devices under test.

IV. Conclusion and Future Work

In this paper we have shown that the availability of the power-gate feature in the BRAMs of latest Xilinx devices can be used to generate SRAM-PUFs even on devices typically running memory initialization on power-up. In this work it turned out that most of the in BRAM located SRAM cells keep their initial value which drastically decreases the gathered entropy of PUF responses. Additionally, just little environmental variations have a strong impact on the PUF response. Both characteristics need to be further investigated in future work. However, the evaluation of the presented SRAM-PUFs are preliminary and uses only three devices. Future tests with a larger set of devices and a inclusion of other tests like aging would evaluate the PUF quality more comprehensively. We are confident that our results can also be applied to other device families besides Zynq. The significant deviations of results obtained from different chip revisions of the same type of FPGA is a further aspect that needs to be addressed in future works.

References

[1] R. Pappu, B. Recht, J. Taylor, and N. Gershenfeld, "Physical One-Way Functions," *Science*, vol. 297, no. 5589, pp. 2026–2030, 2002.

[2] B. Gassend, D. Clarke, M. Van Dijk, and S. Devadas, "Silicon Physical Random Functions," in *Proceedings of the 9th ACM conference on Computer and communications security*. ACM New York, NY, USA, 2002, pp. 148–160.

[3] P. Tuyls, G. Schrijen, B. Skoric, J. van Geloven, N. Verhaegh, and R. Wolters, "Read-Proof Hardware from Protective Coatings," vol. 4249. Springer, 2006, p. 369.

[4] R. Maes, P. Tuyls, and I. Verbauwhede, "Intrinsic PUFs from Flip-flops on Reconfigurable Devices," in *3rd Benelux Workshop on Information and System Security (WISSec 2008)*, 2008.

[5] S. Kumar, J. Guajardo, R. Maes, G. Schrijen, and P. Tuyls, "Extended abstract: The butterfly PUF protecting IP on every FPGA," in *IEEE International Workshop on Hardware-Oriented Security and Trust (HOST 2008)*, 2008, pp. 67–70.

[6] M. Bhargava, C. Cakir, and K. Mai, "Attack resistant sense amplifier based PUFs (SA-PUF) with deterministic and controllable reliability of PUF responses," in *Hardware-Oriented Security and Trust (HOST), 2010 IEEE International Symposium on*, june 2010, pp. 106 –111.

[7] T. Güneysu, "Using data contention in dual-ported memories for security applications," *Signal Processing Systems*, vol. 67, no. 1, pp. 15–29, 2012.

[8] J. Guajardo, S. Kumar, G. Schrijen, and P. Tuyls, "FPGA intrinsic PUFs and their use for IP protection," vol. 4727. Springer, 2007, p. 63.

[9] ——, "Physical Unclonable Functions and public-key crypto for FPGA IP Protection," in *Field Programmable Logic and Applications, 2007. FPL 2007. International Conference on*, 2007, pp. 189–195.

[10] O. Sander, B. Glas, L. Braun, K. D. Müller-Glaser, and J. Becker, "Exploration of Uninitialized Configuration Memory Space for Intrinsic Identification of Xilinx Virtex-5 FPGA Devices," *International Journal of Reconfigurable Computing*, vol. 2012, p. 10, 2012.

[11] U. Rührmair, F. Sehnke, J. Sölter, G. Dror, S. Devadas, and J. Schmidhuber, "Modeling attacks on physical unclonable functions," in *ACM Conference on Computer and Communications Security*, E. Al-Shaer, A. D. Keromytis, and V. Shmatikov, Eds. ACM, 2010, pp. 237–249.

[12] U. Rührmair, X. Xu, J. Sölter, A. Mahmoud, F. Koushanfar, and W. Burleson, "Power and Timing Side Channels for PUFs and their Efficient Exploitation," *IACR Cryptology ePrint Archive*, vol. 2013, p. 851, 2013.

[13] C. Helfmeier, C. Boit, D. Nedospasov, and J.-P. Seifert, "Cloning physically unclonable functions," in *HOST*. IEEE, 2013, pp. 1–6.

[14] E. Simpson and P. Schaumont, "Offline hardware/software authentication for reconfigurable platforms." in *Cryptographic Hardware and Embedded Systems – CHES 2006*, vol. 4249, 2006, pp. 311–323.

[15] A.-R. Sadeghi and D. Naccache, Eds., *Towards Hardware-Intrinsic Security - Foundations and Practice*, ser. Information Security and Cryptography. Springer, 2010.

[16] C. Böhm and M. Hofer, *Physical Unclonable Functions in Theory and Practice*. Springer Publishing Company, Incorporated, 2012.

[17] Xilinx, "7 Series FPGAs Memory Resources: User Guide," *Product Documentation*, October 2013.

[18] ——, "7 Series FPGAs Configuration: User Guide," *Product Documentation*, October 2013.

[19] S. Skorobogatov, "Low temperature data remanence in static RAM," *University of Cambridge Computer Laboratory Technical Report*, vol. 536, p. 11, 2002.