Dynamic Memory-Based Physically Unclonable Function for the Generation of Unique Identifiers and True Random Numbers

Christoph Keller, Frank Gürkaynak,
Hubert Kaeslin, and Norbert Felber

Integrated Systems Laboratory (IIS), ETH Zurich
Gloriastrasse 35, 8092 Zurich, Switzerland
Email: {chrikell, kgf, kaeslin, felber}@iis.ee.ethz.ch

***Abstract*—We introduce a test setup for the extraction of unique identifiers and random bits using DRAM. By utilizing standard 512 MByte DDR3 modules, 400 bit/s of random bits can be generated and unique identifiers are extracted. We show that DRAM can be used as a physically unclonable function. In contrast to SRAM-based PUF, DRAM PUF allow for repeated operations without disconnecting supply voltage while still remaining fully functional for normal storage operations.**

***Keywords*—Physically unclonable Function, Dynamic Memory, True Random Number Generation, Unique Identifier Generation**

I. Introduction

During fabrication of integrated circuits (IC), variations in the size and shape of resistors, capacitors, or transistors and variations in the doping of transistors are inevitable. These variations have a direct influence on the resistance, capacitance, and switching time of these devices. Physically unclonable functions (PUF) exploit these variations to generate unique identifiers (UID). The extracted identifier is a digest of the actual physical peculiarities of this particular specimen of an IC.

Cloning one particular specimen with the intention to also copy the identifier generated by the PUF will not be successful as the processes and their parameters, which cause those variations, cannot be controlled. For this reason, these functions are called "physically unclonable".

In terms of security, PUFs show better resilience against tampering compared to other solutions. When an IC hosting such a PUF is altered, e.g., by etching and directly measuring on the die, the response of the PUF is altered as well. This feature could be used in smart cards where a secret key is usually stored in an embedded EEPROM. With a high technical effort, this EEPROM can be read out and copied. Using a PUF instead would prevent attackers from unauthorized copying.

Dynamic memory is almost ubiquitous in today's electrical devices. Most systems use Dynamic RAM (DRAM) for the storage of temporary data for their higher densities and lower cost per bit. DRAM is used as main memory in servers, workstations, and smartphones. The storage capacity per DIMM (dual in-line memory module) of DDR3 (double data rate dynamic memory type 3) can reach up to 8 GByte, enabling total capacities of up to 1 TByte in large servers.

In this work, we propose a new possibility of building a PUF by using dynamic memory cells. We demonstrate practical feasibility using DDR3 modules. Further, we examine operating conditions which influence the behavior of the PUF and point out limitations of the proposed scheme.

II. Related Work

Physically unclonable functions have been a subject of research for several years. During this time, various approaches have been published for both ASIC and FPGA.

Exploring the delay of signal paths using arbiter-based PUFs have been proposed by Lim et al. [1] in 2005. Derived from those arbiter PUF, ring oscillator (RO) PUF were introduced by Suh and Devadas [2]. The frequency of ring oscillators directly depends on the switching speed and path delay of the used inverters. The authors compare several oscillators and use several counters to determine the speed of each such oscillator.

Arbiter PUFs and RO PUFs have one significant drawback. The PUF circuitry can in most cases only be used for the PUF functionality itself. Therefore, the possibility of dual use structures was investigated.

In 2007, Holcomb et al. [3] proposed using the power-up state of an SRAM as a source for random numbers and UIDs. As this power-up state showed to have only few bits with random characteristics, they performed an entropy estimation and then used a hashing function to extract the random bit pattern from the power-up state. To identify the various SRAM chips, they measured all devices and stored their power-up state in a database. For later identification, they compared the newly extracted power-up state with their database. The combination with the lowest Hamming distance was chosen as the most likely match. The same idea may be applied to latches or flip-flops. The advantage of this kind of PUF is that after identifier generation, the SRAM, latch, or flip-flop is available for regular data storage operation.

The extracted power-up state of SRAM showed to contain only a small amount of random bits. Thus, this power-up state can be seen as a weak random source. NIST published recommendations for random bit generators [4] where they

978-1-4799-3432-4/14/$31.00 ©2014 IEEE

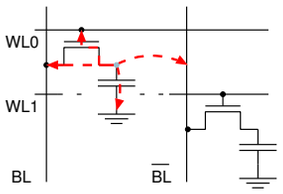

Fig. 1.1 Possible leakage paths of a DRAM cell

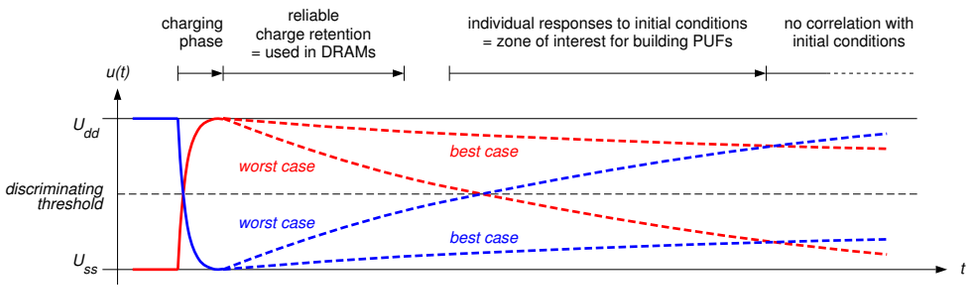

Fig. 1.2 Charge decay of an initially charged capacitor over time.

describe how to extract random bits from such weak random sources.

There exist two patents by Hitachi Ltd [5] and Philips N.V. [6] that mention our proposed features. However, to the best of our knowledge, no functional devices implementing DRAM-based PUF or test results thereof have been published yet.

III. DRAM used as Physically Unclonable Function

SRAM and registers store information as one of two stable states in a positive feedback loop. As opposed to this, in DRAM, information is stored as a charge on a capacitor. The DRAM cell is built from one transistor and one capacitor (see figure 1.1). The main disadvantage of DRAM is the loss of data over time. Due to leakage, the capacitor will eventually discharge and the stored information will be lost. To prevent this, a refresh operation has to be carried out periodically. During this refreshing operation, every cell is read out and the stored data is written back. According to the DDR3 specification, the typical cycle time is 64 ms or 32 ms at high temperatures [7].

Several effects contribute to the loss of charge on the storage node. Subthreshold leakage through the non-conducting access transistor, gate leakage of the access transistor, dielectric leakage in the storing capacitor itself or in the interconnect do influence charge retention (figure 1.1). Each of these effects is directly depending on the actual peculiarities of the transistor, capacitor, or isolation on the chip. Due to the earlier mentioned variations, every storage cell has its own physical trait. Therefore, the leakage effects on the storage nodes will vary as well. In figure 1.2, the possible charge decay over time can be seen.

The goal of each PUF is to generate both UIDs and random numbers. To do so with DRAM, first, some data, the input stream, is stored on the memory array. After a certain time span, the data is read back (output stream) and compared to the originally stored data. Some of the data bits will have flipped between write and read. After several repetitions of this procedure, two types of cells can be distinguished. Most cells tend to have a very predictable behavior. They never flip at all or predictably charge or discharge. The remaining cells do not have a predictable behavior. They behave rather randomly. Depending on the time between write and read, the number and location of these cells may change.

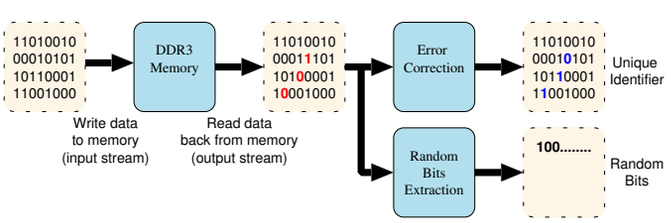

Fig. 2. Operation to generate random bits and UIDs using DDR3 memory

To generate a UID from this output stream, the bits that have a random characteristic have to be filtered out. This can be done by applying an error correction code (ECC) [8]. For the very first identifier calculation, the code word of the ECC is calculated and stored in a code bit storage. For every identifier generation which is carried out later on, the stored code words can be applied and the original identifier can be restored.

Producing random numbers is done by extracting the bits with random characteristic. Using a hashing function and generating a digest from the whole output stream is a well known method to do so [3], [4]. This whole operation is depicted in figure 2.

IV. Experimental DRAM PUF Testbed

DRAM controllers are an integral part of every computer system. Today they are integrated in CPUs and SoCs and do not support such things as disabling refresh operation. However, refreshing the stored data would counteract the desired effect required for PUFs.

To demonstrate the possibility of using dynamic memory for PUFs, we have set up a test system using a Xilinx ML605 board. The ML605 board hosts a Xilinx Virtex 6 FPGA, a DDR3 SO-DIMM slot, and several communication interfaces such as JTAG, UART and Ethernet [9].

On the FPGA, we have implemented our own version of a memory controller with permanently disabled refreshing operation. To facilitate the development, the memory controller is running at the lowest possible speed of 125 MHz [7].

Using this memory controller, we have designed two circuits. The first design, the characterization circuit, was used to investigate the charge decay depending on various parameters such as waiting time or temperature. In the second design, we implemented a fully functional DRAM PUF circuit.

A. Charge Decay Characterization Circuit

A data generator can produce four different input streams: all-zero, all-ones, a pseudo random pattern using an LFSR, and a custom pattern.

In the data comparison unit, the output stream, read from the DRAM, is compared with the initially stored data. A timer is used for the timing of the waiting operation. Various waiting intervals from 1 second to 8192 seconds can be set.

On every DDR3 module, a serial present detect (SPD) EEPROM chip is located to store settings of the used memory chips [10]. Very often, this SPD chips also includes a temperature sensor to measure the current temperature of the module. We used an I^2C controller from opencores [11] to read out the temperature of the module.

Chip scope, a logic analyzer core from Xilinx, which can be accessed through a JTAG connection, has been used to retrieve the data from the FPGA and store in on a computer.

Since the internal storage of the FPGA and therefore the storage of the chip scope instance is limited, only a small portion of the memory could be tested at a time. Therefore, the address range of the address generator was reduced to blocks of 512 kbit.

B. Experimental DRAM PUF Circuit

The results obtained with the evaluator lead to the design of the DRAM PUF circuit (figure 3). We implemented a rather simple (31,26) Hamming Encoder and Decoder [12]. It is able to recover one bit error within a block of 26 bit. Our measurements indicated this to be sufficient for typical room temperature variations. In encoding mode, the calculated code bits are stored in an SRAM internal to the FPGA. In decoding mode, the output stream from the external SO-DIMM is corrected using the code bits from the internal SRAM. To have enough characteristic bits, one 512 kbit block of the output stream is used for the identifier. A 512 kbit string would be a rather large identifier. To reduce its size, these 512 kbit are hashed into a 256-bit string using a SHA-256 core [13]. They represent the actual identifier.

In the output stream, only a small percentage of the bits show random behavior. These bits are widely spread over the whole stream. I.e., the data stream which is read from the DRAM can be regarded as a weak random source. These random bits are extracted using again a SHA-256 core.

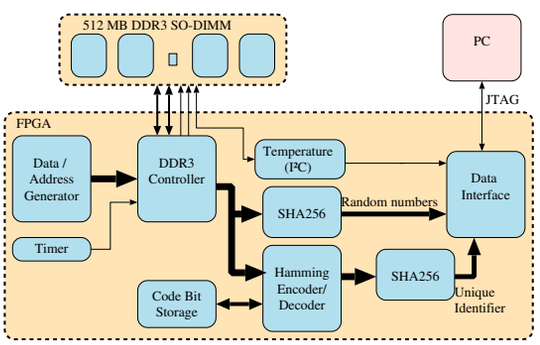

Fig. 3. Block diagram of the experimental DRAM PUF circuit.

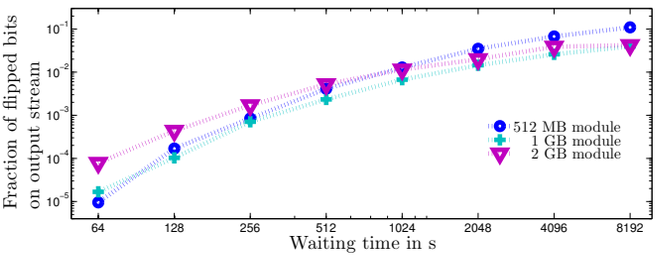

Fig. 4. Bit stability at increasing waiting time

According to the NIST recommendations [4] and after the entropy estimation, the amount of bits from the output stream to be hashed into one 256-bit digest was set to be 8 Mbit. Using 512 MByte modules, this leads to a total of 128 kbit of extracted random numbers per run. The waiting time was set to 320 s. Therefore, in average, 409 random bit/s are produced.

V. Results

To be able to implement a working DRAM PUF circuit, various parameters had to be investigated. Two of these parameters, waiting time and stored data, can be directly influenced by the controller. The temperature of the memory modules can be influenced by directly heating the modules or by increasing the read/write activity. Most of the measurements have been done using 512 MByte DIMMs. For figure 4, two additional modules with 1 GByte and 2 GByte capacity respectively have been utilized.

A. Charge Decay over Time

The system was tested with different waiting time settings. The waiting time was doubled for each new setting. Tests have been conducted with 1 s to 8192 s of waiting time. As can be seen in figure 4, the amount of bits that have changed their value during waiting operation greatly correlates with the actual waiting time. The first flipped bits were detected after 64 s. Our measurements were conducted at room temperature. SO-DIMM temperature was between 31.5°C and 34.4°C.

DDR3 DRAM showed to be very stable in terms of data loss. The measured stability was more than 100 times longer than specified. However, this can be expected, as this RAM was designed for a clock of 533 MHz and temperatures of up to 70°C.

B. Entropy Estimation for Random Number Generation

Entropy is a measure of random bits that are produced by a random source. To generate good random bits, we need to know, how much entropy the output stream contains. To do so, 1228 measurements have been recorded. We used a setup of 256 s waiting time and 512 kbit of pseudo random input stream.

The recorded output bitstreams were then compared with each other and the number of differing bits was calculated. In average, the difference was calculated to be 82.8 bit per 512-kbit block. For the required 512 bit of entropy, blocks of 4 Mbit should be hashed into one 256 bit digest. For the DRAM PUF circuit, we added a safety margin and used blocks of 8 Mbit.

Additionally, we compared the output bitstreams of three identical SO-DIMM from the same manufacturer to find a

TABLE I. HEATING DDR3 RAM

Module temperature in °C	34.4	43.4	53.6	68.5
Number of flipped bits	619	2567	10150	33800
percentage	0.12	0.49	1.9	6.4
Estimated entropy in bit	89	226	671	1960

possible manufacturer-related influence on the location of the flipped bits. A maximum of 1.6% of the flipped bits were found to be at the same location in two modules. Therefore, for the used SO-DIMMs and at the measured temperature, no significant manufacturer-related influence on the output bitstream and therefore on the random numbers or the UID was detected.

C. Impact of Temperature

A further test set has been conducted using a waiting time setting of 256 s. Again, 512 kbit of pseudo random data was stored on the memory. We attached a heating source to the memory module to be able to see the temperature dependency. More than 100 measurements have been conducted for each temperature setting. Table I shows the influence of temperature on the stability of the stored data. Each value is the average over all measurements. Further, the entropy has been estimated. Temperature has a significant influence on the stability. An increase of 9°C multiplied the number of flipped bits by factor of 4. The calculated entropy rose by almost the same factor.

Temperature effects can not only be seen by heating the device. Even small temperature changes of a few °C have a significant impact. For figure 5, the measurements at room temperature have been sorted by their actual measured temperature. Additionally, the average number of flipped bits has been calculated for each temperature. In the temperature range between 33.5°C and 35.75°C, the average number of flipped bits rose by factor 1.4.

The increased entropy at higher temperatures can be used to improve the output rate of the random number generator. For the UID generation, even small temperature variations can result in differing identifiers. To avoid this, either the identifier generation should be stopped if the module temperature leaves a certain range, or a more powerful error correction could be implemented.

D. Analysis of the Experimental DRAM PUF Circuit

The final design was tested for a time period of 12 days. It produced 128 kbit of random data every 320 s resulting in 409 bit/s of random data. The randomness of these bits was tested using the NIST test suite. The 419 Mbit of random data passed all statistical tests provided by the test suite.

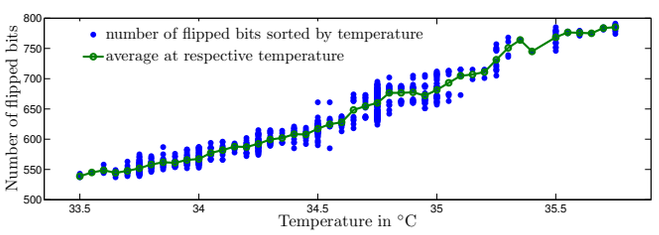

Fig. 5. Small temperature variations and their influence on the charge decay

Further, every 320 s, an identifier was generated. All 3356 extracted identifiers were identical. The measured temperature range of the SO-DIMM was between 31.2°C and 32.8°C, with an average temperature of 31.92°C.

VI. CONCLUSIONS AND FUTURE WORK

In this work, we have set up an experimental DRAM PUF circuit utilizing commercial off-the-shelf DDR3 memory modules. With a first set of measurements, we have shown the influence of temperature and time on the charge decay. Derived from this, we have demonstrated successful UID extraction and random number generation.

In future work, more modules with different storage capacities, clock frequencies and manufacturers will be characterized. Further, the quality of identifiers at higher temperatures will be investigated.

VII. ACKNOWLEDGMENT

This work is part of the QCrypt project, evaluated by the Swiss National Science Foundation and financed by the Swiss Confederation with funding via Nano-Tera.ch.

The authors would like to thank Benjamin Koeppel and Etienne Geiser for their contribution in the design of digital circuit parts.

REFERENCES

[1] D. Lim, J. Lee, B. Gassend, G. Suh, M. van Dijk, and S. Devadas, "Extracting secret keys from integrated circuits," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 13, no. 10, pp. 1200–1205, 2005.

[2] G. E. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," in *Proceedings of the 44th annual Design Automation Conference*, ser. DAC '07. New York, NY, USA: ACM, 2007, pp. 9–14.

[3] D. Holcomb, W. Burleson, and K. Fu, "Power-up sram state as an identifying fingerprint and source of true random numbers for rfid tags," *Proceedings of the Conference on RFID Security*, jul 2007.

[4] *NIST DRAFT Special Publication 800-90C Recommendation for Random Bit Generator (RBG) Constructions*, NIST, aug 2012.

[5] M. MURANAKA, "Method for identifying semiconductor integrated circuit device, method for manufacturing semiconductor integrated circuit device, semiconductor integrated circuit device and semiconductor chip," Japan Patent EP1 341 214 A1, 09 06, 2001.

[6] M. P. Heiligers, A. C. Kruseman, R. H. W. Salters, G. J. Shrijen, P. T. Skoric, Boris an Tuyls, and V. R. S. Van, "Semiconductor device identifier generation method and semiconductor device," Netherlands Patent WO 2 007 119 190 A3, 04 04, 2007.

[7] JEDEC, *DDR3 SDRAM Standard, JESD79-3F*, JEDEC SOLID STATE TECHNOLOGY ASSOCIATION, jul 2010.

[8] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," in *Advances in Cryptology - EUROCRYPT 2004*, ser. Lecture Notes in Computer Science, C. Cachin and J. Camenisch, Eds. Springer Berlin Heidelberg, 2004, vol. 3027, pp. 523–540.

[9] *ML605 Hardware User Guide*, XILINX, oct 2012.

[10] JEDEC, *PC3-6400/PC3-8500/PC3-10600/PC3-12800/PC3-14900/PC3-17000 DDR3 Unbuffered SO-DIMM Reference Design Specification*, JEDEC SOLID STATE TECHNOLOGY ASSOCIATION, dec 2010.

[11] R. Herveille. (2013, aug) I2c controller core :: Overview@ONLINE. [Online]. Available: http://opencores.org/project,i2c

[12] T. K. Moon, *Error Correction Coding: Mathematical Methods and Algorithms*. Wiley-Interscience, 2005.

[13] *Descriptions of SHA-256, SHA-384, and SHA-512*, NIST, 2001.