Hindawi Publishing Corporation
International Journal of Reconfigurable Computing
Volume 2012, Article ID 219717, 10 pages
doi:10.1155/2012/219717

Research Article

Exploration of Uninitialized Configuration Memory Space for Intrinsic Identification of Xilinx Virtex-5 FPGA Devices

Oliver Sander, Benjamin Glas, Lars Braun, Klaus D. Müller-Glaser, and Jürgen Becker

Institute for Information Processing Technology (ITIV), Karlsruhe Institute of Technology (KIT), Engesserstr. 5, 76131 Karlsruhe, Germany

Correspondence should be addressed to Oliver Sander, sander@kit.edu

Received 8 June 2011; Revised 25 October 2011; Accepted 25 October 2011

Academic Editor: Claudia Feregrino

SRAM-based fingerprinting uses deviations in power-up behaviour caused by the CMOS fabrication process to identify distinct devices. This method is a promising technique for unique identification of physical devices. In the case of SRAM-based hardware reconfigurable devices such as FPGAs, the integrated SRAM cells are often initialized automatically at power-up, sweeping potential identification data. We demonstrate an approach to utilize unused parts of configuration memory space for device identification. Based on a total of over 200,000 measurements on nine Xilinx Virtex-5 FPGAs, we show that the retrieved values have promising properties with respect to consistency on one device, variety between different devices, and stability considering temperature variation and aging.

1. Introduction

Identification of devices is a primitive that plays a crucial role for a number of applications, including authentication of devices and protection against cloning of devices (cocalled *product piracy*) or intellectual property. IDs which are stored in nonvolatile memory can often be easily cloned or modified. Hence approaches have been published to overcome the aforementioned drawbacks. They are usually based on unique physical properties of the single chip. For example, such properties are caused by manufacturing process variations. The two main approaches in this context are physical fingerprinting and the use of physical uncloneable functions (PUFs). Former strives to identify a given circuit directly by physical characteristics latter use physical characteristics to perform a challenge-response authentication.

A promising technique used for both approaches is to observe the state of uninitialized SRAM cells. When voltage above a certain threshold is applied to an SRAM cell its initial unstable state will change to one of two possible stable states "0" or "1". The probability for each stable state is heavily dependant on small variations originated during the CMOS fabrication process causing slight deviation in threshold voltage inside the cells. The probability varies between different cells even inside a single chip thus representing a characteristic initial memory content on power-up for each device. Depending on the probability distribution the major part of the memory content is stable for most of the power-ups. Other bits having a probability around 50% show a power-up behaviour similar to random noise. Assuming a high rate of stable data the memory content can be used to provide high quality identification data that is very hard to reproduce deliberately.

Considering SRAM-based field-programmable gate arrays (FPGAs), configuration memory or BRAM cells can potentially be used to retrieve fingerprints from uninitialized SRAM cells. Nevertheless this technique depends on the availability of SRAM that is not automatically initialized to a designated fixed value at device power-up. As a random configuration might well lead to short circuits and therefore damage the device physically, many vendors enforce the clearing of configuration memory by an unavoidable initialization phase on power-up. This is the case, for example, for the Xilinx Virtex-5 series FPGAs considered in this contribution. Certainly not all parts of the configuration are so critical. Initializing SRAM cells on power-up to a fixed value might consume additional area on the chip. Due to area efficiency, there is some chance for certain regions

in configuration memory without this kind of reset-on-powerup procedure. The challenge is to find possibilities for secure hardware identification without having to implement and configure complex additional logic on the device.

In this work we present a method to retrieve identification data from the configuration memory space using readouts from presumably unused and therefore uninitialized hidden address ranges. Besides description of the method and the used tools we give statistical data from our measurements on a population of Virtex-5 devices indicating the potential to use the memory region for identification purposes. Moreover we demonstrate a straightforward methodology to generate reference keys that allow for robust identification of devices.

The remainder of this paper is structured as follows. Section 2 gives an overview of some related publications. The situation and identification approach and the data measurement basics are given in Section 3. The methods for data examination are presented in Section 4. Section 5 presents the analysis results from the measurements that are interpreted in Section 6. Also open points and potential applications are discussed in Section 6. In Section 7 we give some examples of possible security applications. Specific properties and benefits of the used memory region are considered in Section 8. The paper is concluded in Section 9.

2. Related Work

As previously mentioned there exist two main approaches for physical identification of CMOS devices which have received considerable attention in recent years. An introduction to PUFs can be found, for example, in [1–4], and a collection of related publications is given in [5]. For FPGAs also the use of flip flops is presented in [6, 7]. PUFs implementation as proposed in [6] is possible but it requires a specific configuration of the device.

In the following we will focus on physical fingerprinting based on SRAM cells. The use of fabrication process-related variations of CMOS gates for identification has been widely examined. Excellent identification properties show dedicated circuits as in [8]. The use of SRAM cells (see, e.g., [9], patent [10]) comes at the cost of more noisy data but with the benefit of not requiring extra space on the chip. To extract IDs and cryptographic keys from this noisy data, various mechanisms like fuzzy extraction [11] have been proposed. Finally a preliminary version of this work was previously published in [12].

3. FPGA Configuration Memory and Proposed Approach

On many current FPGA devices the used configuration memory is initialized to a defined value at startup, rendering the SRAM cells useless for device identification by physical fingerprinting. We look at the Xilinx Virtex-5 series [13] as representatives of this kind of devices. The challenge is to find areas in the configuration memory which are not initialized due to not being critical for the chip integrity.

Table 1: Frame address register description (Xilinx Virtex-5).

Address type	Bit index	description
Block type	[23:21]	Used in Virtex-5: 000 up to 011.
Top_B bit	20	Selects between top and bottom
Row address	[19:15]	Selects the current row
Column address	[14:7]	Selects a major column
Minor address	[6:0]	Selects a frame

In first experiments we found, that the SRAM configuration cells are reliably set to zero on device power up and can therefore not be used for identification of a single device. Even the BRAM blocks where random content poses no direct threat to the physical device are reliably zeroed out.

As a solution approach we looked at parts of the configuration memory address space that are not used for configuration and are therefore possibly not included in the initialization process. It turned out that readout of address ranges reserved for additional future block types other than configurable logic, BRAMs, special frames and non-configuration frames yielded device-specific data that can serve for identification. We therefore looked at the address regions officially not used, that is, the addresses starting with a "1" in bit 23 (see Table 1).

3.1. Structure of Configuration Memory. Configuration memory in Xilinx Virtex-5 devices is organized in frames as smallest configurable units. All frames have an identical, fixed length of 1312 bits, split into 41 words of 32 bits [14]. Each frame can be addressed individually using a 24-bit address written to the Frame Address Register (FAR) of the device. The FAR is divided into five subfields as described in Table 1. Over several interfaces like JTAG and SelectMAP, frames can be written for configuration and the content can also be read out for verification.

As can be seen, the three most significant bits [23:21] are designated for identifying the block type. The block types used in the Virtex-5 series [14] are Interconnect and block configuration (000), Block RAM content (001), Interconnect and Block Special Frames (010), and Block RAM Non-Configuration Frames (011). Not (officially) used is the complete address range starting with MSB 1 ($1xx$).

3.2. Tooling. For readouting the configuration memory we used the 1149.1 JTAG [15] configuration interface and the readout procedure given by Xilinx [14]. A PC was connected via Digilent USB programming cable to the FPGA board. Based on the Digilent Port Communications Utility (DPCUTIL) [16] Library and API we created a tool for directly reading and writing configuration registers and data frames based on C#. Besides simple JTAG access the tool allows multiple readbacks and generation of some additional data thus providing the basis of our statistical analysis. For writing complete bitstreams to program the devices we used

the standard Xilinx Suite v10.1, the associated programming cables, and the iMPACT tool.

4. Examination

In this section we give some figures about the collected data. Examination results based on this data are given in Section 5.

We looked at a total of nine Xilinx XC5VLX110T devices [17], integrated in Digilent XUPV5-LX110T Evaluation Platforms [18]. In the following we denote the devices with letters A, ..., I. Since the substructure of the address space outside the area used for configuration is not public, we used the autoincrementation of the FAR to determine valid addresses. It turned out that at least 96 kbit of nontrivial data could be read out from an unconfigured device within a certain memory region. A block of ten consecutive frames (13,120 bit) was chosen for closer investigation. In this paper, we refer to this data stream consisting of 10 frames of data. Each time we perform a readout of the device, we read these 10 frames.

For statistical examination for each device $X \in (\mathrm{A}, \ldots, \mathrm{H})$ two measurement series of 10,000 readouts $\Theta_X = (\vartheta_X^1, \ldots, \vartheta_X^{10,000})$ each were collected, one series from the unconfigured device (Θ_X^i) and one from the programmed device $(\overline{\Theta}_X^i)$. So a total of 160,000 data streams (each consists of 10 frames) could be used for statistics and creating master identification keys. In addition, test data $T_X = (\tau_X^1, \ldots, \tau_X^{100})$, resp., $\overline{T}_X = (\overline{\tau}_X^1, \ldots, \overline{\tau}_X^{100})$ of 100 readouts was collected from every board setup, that is, a total of 1,600 test streams. Eventually for the exploration of temperature stability we made another 40000 measurements for three selected boards D, H, and I which were exposed to a wide temperature range.

For ideal identification of devices it would be optimal to have a bijective mapping from an ID to a physical device and vice versa. The examination of the measurement data was therefore done looking in two directions. First the correlation and consistency of the different measurements on one board were investigated, to get a unique mapping from a device to an ID. From each measurement series a reference data stream as master identification key was created that serves as a candidate for device representation. In a second step the results from different boards were compared aiming for an injective mapping from one ID to one specific device. Validation of the identification process was performed using the test data sets and the reference IDs. The results are given in the following paragraphs.

5. Results

5.1. Similarity and Reference Keys. First we look at the conformity of different readouts from the very same device X to map each device to an ID. Therefore we used 10,000 readouts to create a frequency distribution of zeros and ones of every bit in the data stream for each FPGA. Figures 1 and 2 show the results for two single devices. The other devices showed similar figures.

The measurement reveals a distinct accumulation of bits showing constantly the same value. As can be seen in Table 2

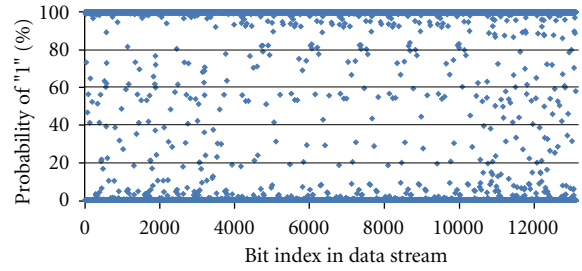

Figure 1: Probability of bit value "1" over 10 k measurements on device D.

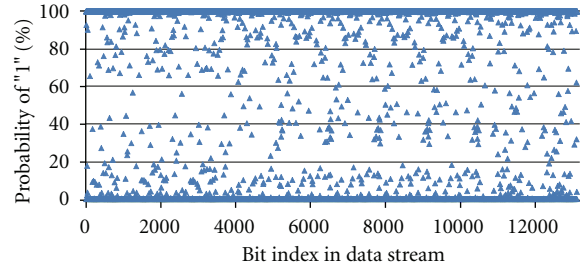

Figure 2: Probability of bit value "1" over 10 k measurements on device F.

the number of ones dominates the number of zeros by a factor of 1.7. The portion of flipping bits is below ten percent ranging from 7.2% to 9.7% with a mean value of 8.4% over the devices under consideration (see Figure 3). The distribution of zeros and ones is pretty similar for different devices (Figure 3). For the very same device—configured or unconfigured—the numbers are not identical but relatively close to each other. These differing results for one device are more probably caused by temperature variances than by the fact of configuration dependencies.

Figure 4 depicts the total amount of constant bits observed over a variable number of compared readouts for different devices. The measurements show that more than 90% of all bits are constant over all compared readouts. We therefore notice a distinctive coherence of measurements from one device.

To quantify this coherence we compared the single data streams with a reference stream ρ to get a measurable deviation value. To achieve this we used the probability distribution to generate a reference data stream ρ of the measurement, setting each bit to the value with the higher probablity of occurrence according to the measured data. ρ therefore represents the bitwise rounded mean of all measured streams.

We then determined the Hamming Distance (HD) of every readout to this reference key. Figure 5 shows the respective distribution for one device. The readouts from one board show a close cross-correlation. For device D the Hamming distances show an expectancy value of 99, equivalent to only 0.75% of the total data, and a standard deviation of 7.4. Device F gives an expectancy value of 137 (1.0%) with a standard deviation of 9.1. All HD values

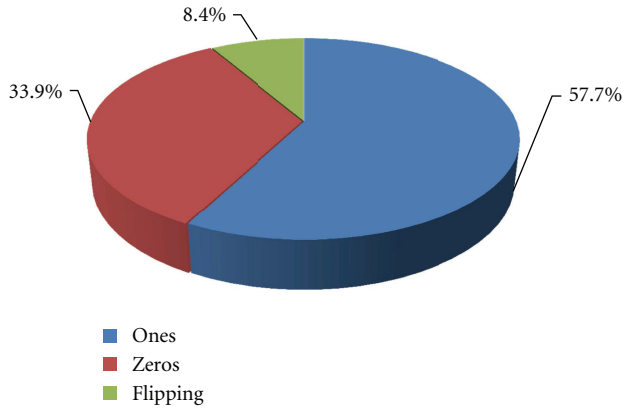

Figure 3: Mean occurrences of constant ones and zeros as well as flipping bits of Table 2.

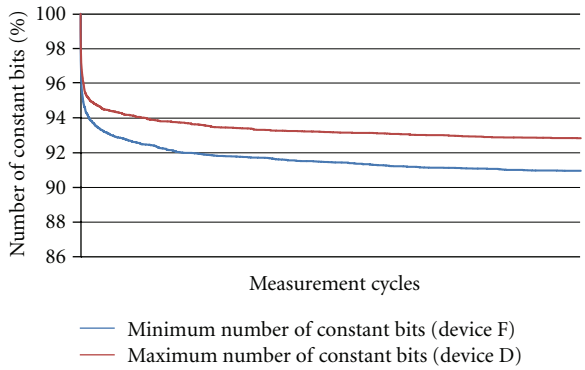

Figure 4: Percentage of constant bits over 10,000 measurements for two devices, that represent the maximum (device D) and minimum (device F) over all devices. All other values reside in the area between the two curves.

computed from all devices resided in an interval of [53–208]. The analyses of the measurement series from the other devices are in between the two extremal values given.

So far, the compared values were originated from unconfigured devices and directly measured after startup. For identification of the device, the key data should be independent from the content of the address ranges of configuration memory that are used for programming the device. Therefore, also the programmed device was examined. Figure 6 depicts the HD values of readouts of a programmed board in relation to the reference key determined on the same board in unconfigured state.

The values read out from the configured board show a slightly higher deviation from the reference but are still within a mean deviation of well below 1% of the total data stream. This is also the case for the other examined devices (see also Table 3). So all data streams from one board show a great mutual similarity. To verify the usability of the proposed approach, we compare in a next step data from different devices.

5.2. Distinction and Identification. For unique identification, the mapping of a given ID value to a device is necessary.

Table 2: Number of constant zeros, ones, and flipping bits.

Device	Ones		Zeros		Flipping	
A	7798	59,4%	4238	32,3%	1084	8,3%
$\overline{A}$	7744	59,0%	4190	31,9%	1186	9,0%
B	7698	58,7%	4435	33,8%	987	7,5%
$\overline{B}$	7661	58,4%	4414	33,6%	1045	8,0%
C	7388	56,3%	4676	35,6%	1056	8,0%
$\overline{C}$	7325	55,8%	4642	35,4%	1153	8,8%
D	7702	58,7%	4479	34,1%	939	7,2%
$\overline{D}$	7638	58,2%	4432	33,8%	1050	8,0%
E	7626	58,1%	4350	33,2%	1144	8,7%
$\overline{E}$	7566	57,7%	4357	33,2%	1197	9,1%
F	7517	57,3%	4418	33,7%	1185	9,0%
$\overline{F}$	7491	57,1%	4352	33,2%	1277	9,7%
G	7498	57,1%	4577	34,9%	1045	8,0%
$\overline{G}$	7418	56,5%	4543	34,6%	1159	8,8%
H	7537	57,4%	4543	34,6%	1040	7,9%
$\overline{H}$	7460	56,9%	4552	34,7%	1108	8,4%

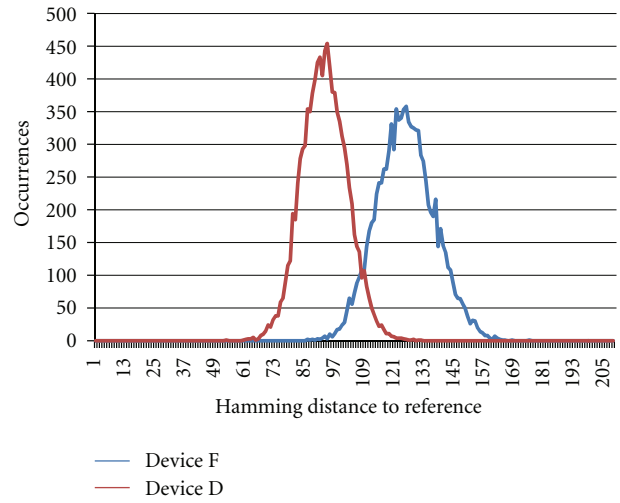

Figure 5: Distribution of Hamming distances to the respective reference streams for unconfigured devices D and F and 10,000 measurements each.

Here the difference between readouts of different boards is the crucial property. A first indicator is the difference between the computed reference keys. Table 3 shows the mutual Hamming distances of all reference keys.

The table shows a very close correlation between the measurements of the same device, meaning that only a slight dependency on the configuration state can be determined. In contrast, HD values between different devices are all near 6000, meaning a 45% deviation and therefore near to the expected value of 50% for pure random streams. A closer look at the streams revealed some bits that are constantly zero over all measurements of all devices, which in combination with the slight nonuniform distribution of ones and zeroes (see Table 2) may account for the lower absolute deviation.

Table 3: Mutual Hamming distances of reference streams from unconfigured devices A . . . G and configured devices $\overline{A}$ . . . $\overline{G}$.

Device	A	$\overline{A}$	B	$\overline{B}$	C	$\overline{C}$	D	$\overline{D}$	E	$\overline{E}$	F	$\overline{F}$	G	$\overline{G}$	H	$\overline{H}$
A	0	90	5903	5918	6161	6151	6020	6024	6089	6118	6201	6201	5981	5973	6073	6100
$\overline{A}$	90	0	5897	5912	6157	6147	6008	6012	6101	6130	6199	6199	5975	5967	6047	6072
B	5903	5897	0	87	6268	6276	5967	5963	5990	6011	6050	6048	6118	6128	5952	5969
$\overline{B}$	5918	5912	87	0	6283	6291	5972	5968	5965	5980	6077	6075	6103	6113	5949	5964
C	6161	6157	6268	6283	0	74	6339	6343	6054	6053	6170	6176	6266	6250	6012	6027
$\overline{C}$	6151	6147	6276	6291	74	0	6315	6319	6032	6031	6146	6152	6262	6246	6014	6029
D	6020	6008	5967	5972	6339	6315	0	78	5991	6010	6165	6159	5989	5991	5979	6000
$\overline{D}$	6024	6012	5963	5968	6343	6319	78	0	5997	6016	6135	6129	5977	5979	5993	6014
E	6089	6101	5990	5965	6054	6032	5991	5997	0	113	6088	6092	6032	6024	6042	6079
$\overline{E}$	6118	6130	6011	5980	6053	6031	6010	6016	113	0	6101	6105	6041	6033	6039	6080
F	6201	6199	6050	6077	6170	6146	6165	6135	6088	6101	0	30	6194	6198	6108	6135
$\overline{F}$	6201	6199	6048	6075	6176	6152	6159	6129	6092	6105	30	0	6188	6192	6098	6125
G	5981	5975	6118	6103	6266	6262	5989	5977	6032	6041	6194	6188	0	58	6080	6077
$\overline{G}$	5973	5967	6128	6113	6250	6246	5991	5979	6024	6033	6198	6192	58	0	6086	6085
H	6073	6047	5952	5949	6012	6014	5979	5993	6042	6039	6108	6098	6080	6086	0	119
$\overline{H}$	6100	6072	5969	5964	6027	6029	6000	6014	6079	6080	6135	6125	6077	6085	119	0

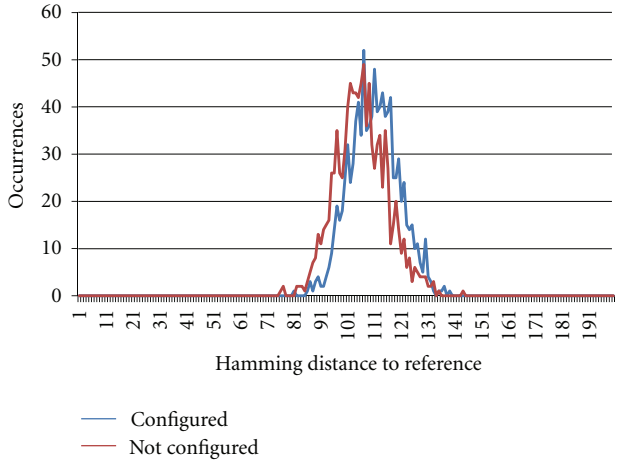

Figure 6: Distribution of Hamming distances to the device D reference stream ρ_D (generated from the unconfigured device data Θ_D) for 1000 test data streams from the unconfigured ($T_X = (\tau_X^1, \ldots, \tau_X^{1000})$) and configured ($\overline{T}_X = (\overline{\tau}_X^1, \ldots, \overline{\tau}_X^{1000})$) device each.

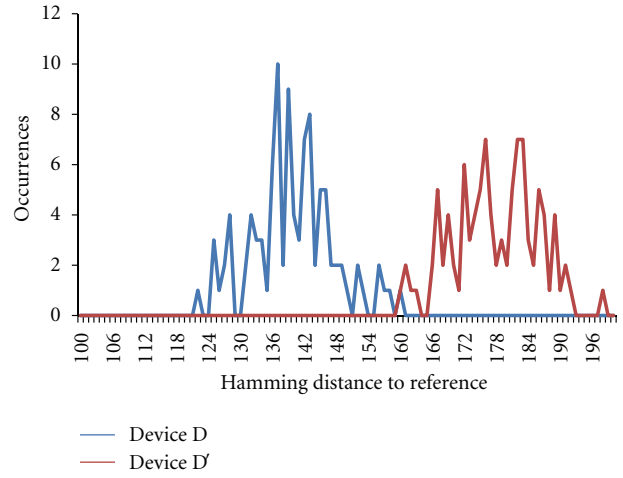

Figure 7: Enlarged presentation of the Hamming distance distribution of T_D relative to ρ_D and $\overline{\rho}_D$.

Figure 8 shows the comparison of the test value set from device D with all reference strings. Two distinctive peak clusters are visible in the chart. Two peaks with relatively small average Hamming weights of about 160 originate from the comparison with the two reference sets ρ_D (unconfigured) and $\overline{\rho}_D$ (configured) from the same board D (see Figure 7 for an enlarged view). The remaining 14 peaks belong to reference values of the other devices and are clustered in a narrow interval around an HD of 6000. Figures 7 and 9 detail the two interesting intervals rescaled for better visibility.

The charts show a clear separation of matching and nonmatching devices with differences in the Hamming distance of more than an order of magnitude. The results for the other devices are very similar and also provide recognition rates of 100% using very simple threshold algorithms.

5.3. Temperature Stability. So far all measurements have been made at standard laboratory conditions including a regular temperature of app. 25°C. In diverse applications the temperature of the FPGA may vary in some range around this temperature. In order to ensure applicability of our approach some independability of temperatures must be proven. For this we selected three different boards D, F, and I and exposed these to different temperatures ranging from −30°C to +80°C in a climate chamber.

The first experiment conducted was to determine the variation of the fingerprint at different temperatures. This was done by reading back the device 1000 times and building a master bitstream out of this measurement. Then each

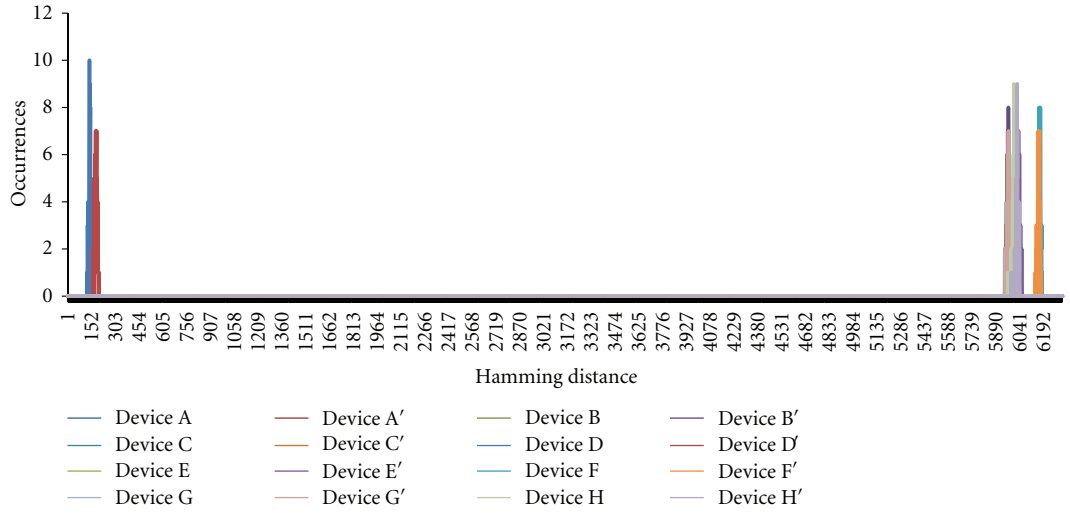

Figure 8: Distribution of Hamming distances of test set $T_D = (\tau_D^1, \ldots, \tau_D^{100})$ to all reference values ρ_X and $\overline{\rho}_X$ for $X \in \{A, \ldots, G\}$.

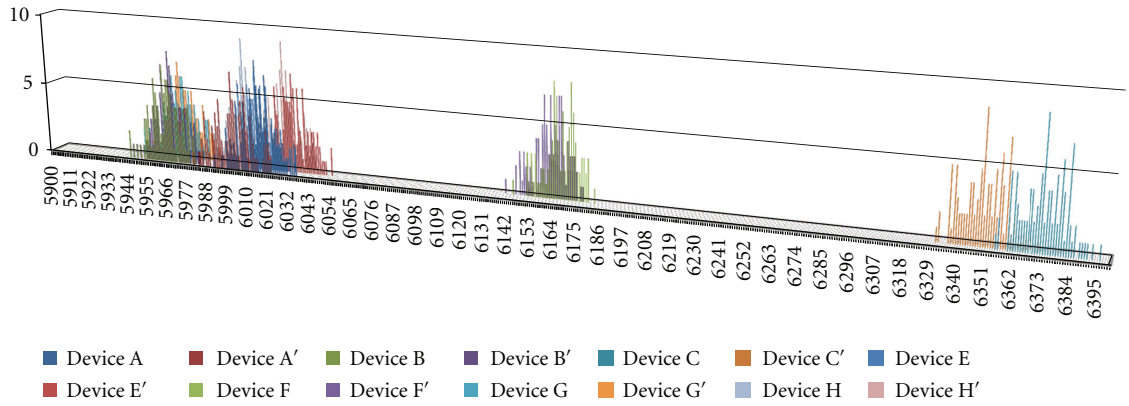

Figure 9: Enlarged presentation of the Hamming distance distribution of T_D relative to ρ_X and $\overline{\rho}_X$ for $X \neq D$.

bitstream is compared to the master. Representative results are depicted in Figures 10, 11, and 12 for the mean value of 20°C and both extrema −30°C and +80°C. The mean hamming distance varies for about 40 bits depending on the temperature, what is quite a moderate value, having in mind a nonmatching HD of about 6000. Moreover for different temperatures the curves look quite similar. Interestingly there is no common temperature-dependent behavior. While in Figure 11 the standard temperature has the smallest variation and the lowest temperature the highest one, the situation changes for the second board, where the lowest temperature has the lowest variation and the highest temperature the highest one. Actually this is the behavior we expected before our experiments. However the situation reverses for board D in Figure 12. So we can only conclude that there is some slight variation with different temperatures.

In our second experiment we compared the measurements taken at different temperatures to the reference bitstream generated from the readbacks at 20°C. The results for Board H are depicted in Figure 13. Results for other boards look similar. Obviously the comparison shows optimum results for 20°C. For all other measurements we get a rising hamming distance with greater temperature differences for both measurements. Again the form of the curves remains similar which again means the variance of the hamming distances are almost equal. The maximum hamming distance goes up to around 300 for the highest and lowest temperatures which is very moderate compared to non matching hamming distances.

The third experiment is a temperature sweep, starting at 85°C and cooling down to −30°C. This process took about 40 min, and 4800 measurements were taken with 500 ms between each measurement. Out of all measurements a master bitstream was built and the hamming distance for each readback calculated. The results are shown in Figure 14. At the beginning the hamming distance is around 350, goes down to 120 at approximately 0°C, and then goes up again with even more decreasing temperature. The blue line shows the chip temperature which is around 5 to 10° higher than the surrounding temperature for an unconfigured Virtex-5 device.

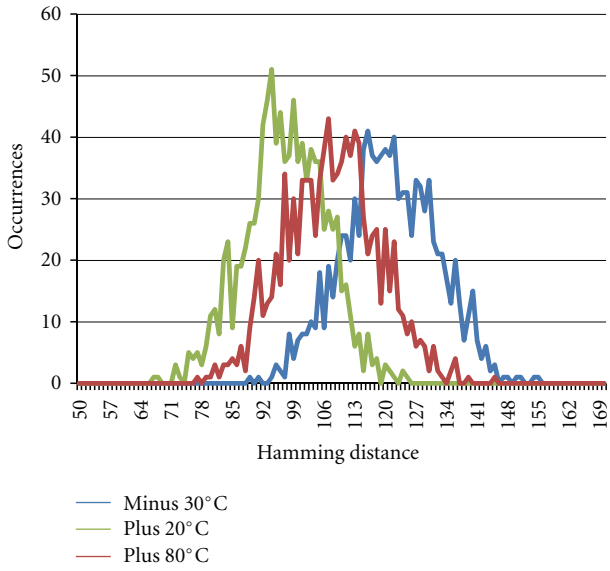

Figure 10: Hamming distance of single readback and master bitstream of board H at different temperatures.

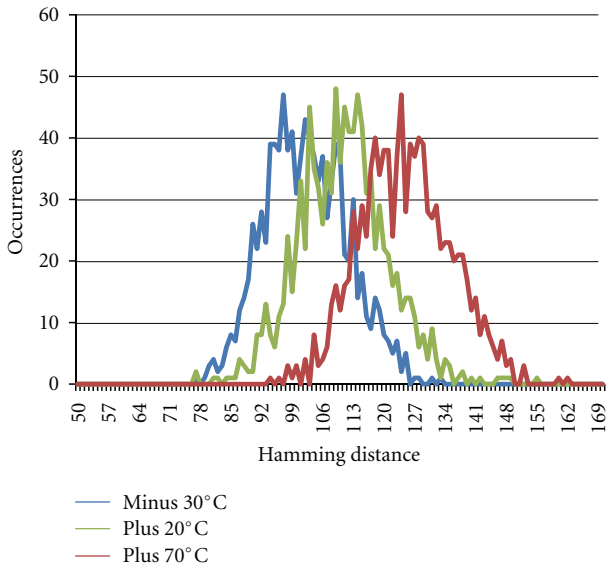

Figure 11: Hamming distance of single readback and masterbitstream of board I at different temperatures.

One can conclude that our device identification mechanism shows slight temperature dependency, which does not influence the identification process. In a final experiment we used the 20°C master bitstreams for non matching devices at different temperatures. For such non matching devices the temperature has almost no influence, resulting in a deviation of about 20 in the hamming distance. Related to an absolute hamming distance of around 6000 this is neglectable. So finally device recognition is feasible for a wide temperature range.

5.4. Aging. Another important aspect is device aging. Again, we have to point out we do not know what information exactly is read from the device by our methodology. However

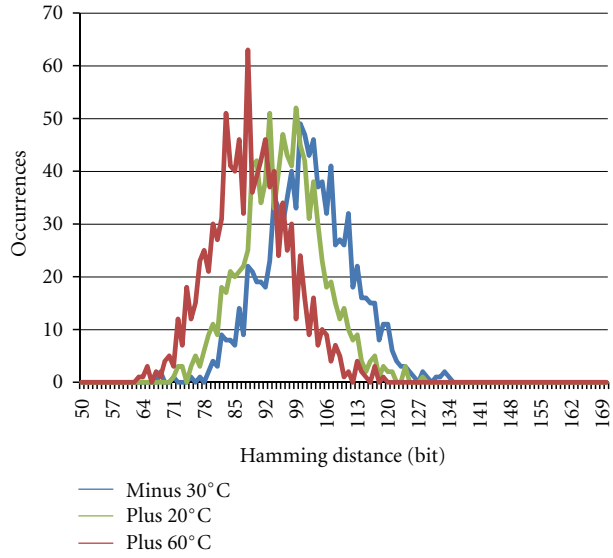

Figure 12: Hamming distance of single readback and masterbitstream of board D at different temperatures.

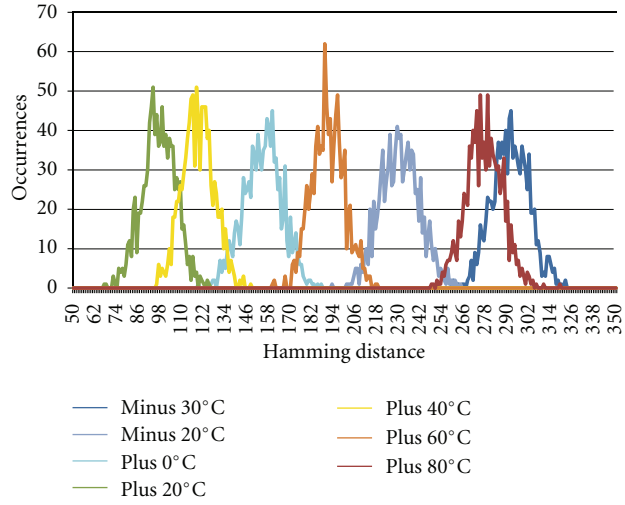

Figure 13: Matching device identification with master bitstream generated at 20°C and identification readback temperature range from −30°C to +80°C (1 k measurements at each temperature step).

we did a readback for different devices with a time difference of one year. The master bitstream generated one year ago was applied to the readback. Figure 15 shows the result for device H. One can see a slight deviation between both devices. The difference might be based on some aging of the device, but also slight deviations in the environment cannot be completely ruled out. However this effect is relatively small and does not affect the identification mechanism presented in this paper.

6. Discussion

The examination shows a very high correlation within data streams from a single device. Test data revealed deviations from the reference master around 1% of the total data

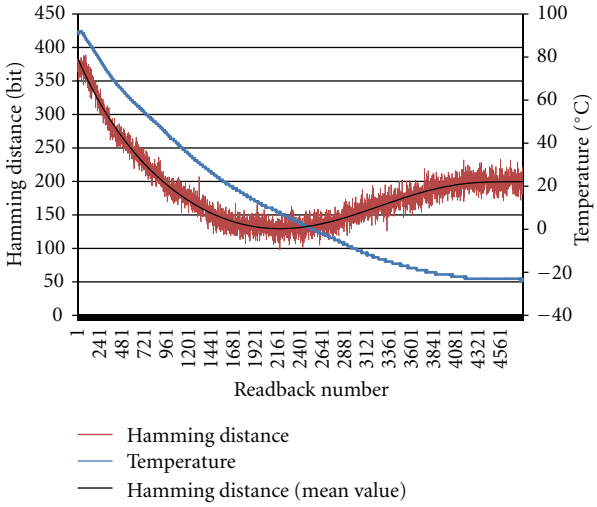

Figure 14: 4800 hamming distance measurements taken during temperature sweep from 85°C to −30°C environmental temperature. The blue curve shows the chip temperature.

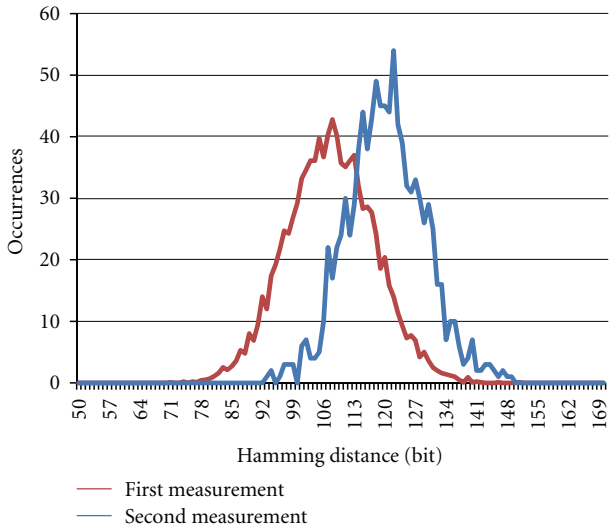

Figure 15: Two device identification measurements taken with one year between 1st and 2nd measurement.

stream. On the other hand comparison to reference keys from other devices resulted in HD values corresponding to a deviation in more than 40% of the bits enabling a reliable identification of the device being read out. In addition the identification technique needs no configuration on the device and therefore no area on the configurable fabric or other hardware resources. We therefore believe that the method can be used to securely and reliably identify physical devices.

An open issue is the question where the data originates from in the first place. Since the address space is not used officially on the devices it could as well be omitted. So it is possible that the readback retrieves data from memory locations that are in fact used for some purpose. On the other side there are the distinct differences between different instances of the same device model and the Hamming

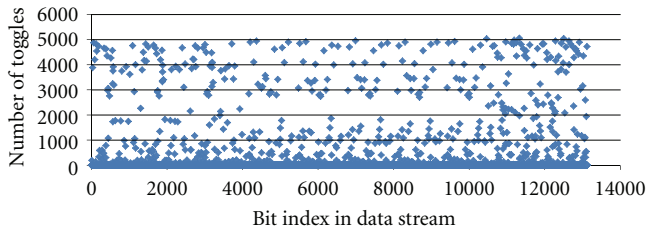

Figure 16: Number of single bit value changes for the different bit indices over 10,000 measurements on device D.

distances near the expectancy values for random strings. This seems to be unlikely for deliberately configured data. Nevertheless we are so far not able to determine the precise origin of the readout data. In addition the data shows slight repetitive patterns as can be seen in Figures 1 and 2 that have to be analysed to get valid statements about the number of usable identification bits.

Moreover the question arises whether the data could also be used to generate random numbers. Here not the constant but the variable bits are crucial for the quality. Figures 1 and 2 already show that the data have different distribution depending on the bit index. Figure 16 shows in addition the number of value changes over the measured set of data streams for the different bit indices.

The variety of behaviours of the different bits could be seen as an indication that an extraction of random numbers could be possible, but further analysis and postprocessing of the data is needed to get reliable results.

7. Security Applications

There are several possible applications for the described SRAM-based PUFs. Since the PUF consists of memory values, it is easy to evaluate also for an attacker. Hence it should be possible to create the memory response without the specific device but only a readout of the memory. A security application therefore has to make sure that the PUF is actually evaluated and not replayed values are used. One possibility for that is guaranteed direct physical access to the device. This is utilized in two applications examplarily looked at in the following: product piracy protection and IP protection.

For the first use case, the initial memory content is read out at production of the device and an adequate error correcting coding mechanism is applied to create a stable fingerprint. This fingerprint is then used to create a manufacturer certificate for the device, for example, by signing a hash value of the fingerprint with a manufacturer-specific secret key. This certificate is then delivered together with the device to the customer who is then able to verify whether he acquired a genuine device since a manufacturer of bogus devices is neither able to create valid certificates for his device nor to clone the PUF-fingerprints for the cloned units. Since the fingerprinting does not rely on memory regions used for configuration the verification can be performed even during operation without interruption and without need to restart the device.

Possible applications for the considered identification mechanism are not restricted to external evaluation of a device but could as well be used inside a configuration bitstream itself. This is considered in the second use case for IP protection. Since the configuration memory is accessible from the fabric through the common ICAP interface, a configured bitstream could easily use the available information to identify the device it is configured on and perhaps react accordingly. A straightforward application would be to bind a configuration to a specific device. To show feasibility we implemented the basic functionality.

Using an ICAP interface connected to a MicroBlaze System we were able to read out the identifying data from within the device, getting similar results as when using the external JTAG access. By comparing the data with a reference string stored in the configuration bitstream, the design can verify whether it is run on a predefined device. If a mismatch is detected, an internal enforcement mechanism can, for example, disable the design or reduce functionality, building a reliable copy protection mechanism for hardware configuration IP.

8. Specific Properties of the Memory Region

When the identification is used for policy enforcement and copy protection, circumventing the identification could be of interest for possible attackers. In the classical use case using memory cells for identification it has to be guaranteed that the cells are read out before any influence on the contents is possible, since the memory cells can easily be written with arbitrary values. In contrast to that we observed that writing on the considered addresses was not possible over the examined interfaces—the readout values were not changeable by write attempts. This is a major benefit in contrast to general memory cells.

In addition, since the used cells are outside the memory space used for configuration, we assume that they are not used for programming a design to the chip. The identification should therefore work also for configured devices during runtime and for partial bitstreams having no control and no information about the contents of the surrounding chip area or even about their own placing on the area as long as they have access to the ICAP interface. So no area has to be reserved for the identification and no constraints for placement and routing are imposed by the approach.

9. Conclusion and Further Work

For the challenge of identifying physical devices of Xilinx Virtex-5 FPGA family we examined configuration memory readouts from address ranges that are assumedly not used in the series. We chose an address range reserved for future block types to read out data and compared the readouts mutually from one device and between devices. Results showed a strong coherence of different streams from the same device and strong deviations between devices. This holds true also for a wide temperature range. Moreover we showed device identification is feasible after one year of aging. We therefore assume the data suitable for identification of physical devices. The method opens identification possibilities for FPGA series where other SRAM-based approaches fail because of enforced initialization to defined values at startup.

To proof reliable identification some future work is necessary. So far it is not clear how long the potential key sequences are and in what ratio they include real identification information and noise. It could also be investigated, whether it is also applicable to devices of other series.

Acknowledgments

The authors acknowledge support by Deutsche Forschungsgemeinschaft and Open Access Publishing Fund of Karlsruhe Institute of Technology.

References

[1] 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 (CCS '02)*, pp. 148–160, ACM, New York, NY, USA, November 2002.

[2] B. Gassend, D. Lim, D. Clarke, M. van Dijk, and S. Devadas, "Identification and authentication of integrated circuits," *Concurrency and Computation: Practice and Experience*, vol. 16, no. 11, pp. 1077–1098, 2004.

[3] J. Guajardo, S. S. Kumar, G.-J. Schrijen, and P. Tuyls, "Fpga intrinsic pufs and their use for ip protection," in *Proceedings of the 9th International Workshop on Cryptographic Hardware and Embedded Systems (CHES '07)*, pp. 63–80, Springer, Berlin, Germany, 2007.

[4] P. Tuyls and B. Skoric, "Strong authentication with physical unclonable functions," in *Security, Privacy, and Trust in Modern Data Management*, M. Petkovic and W. Jonker, Eds., Data-Centric Systems and Applications, pp. 133–148, Springer, Berlin, Germany, 2007.

[5] R. Maes, "PUF Bibliography," 2010, http://www.rmaes.ulyssis.be/pufbib.php/.

[6] R. Maes, P. Tuyls, and I. Verbauwhede, "Intrinsic pufs from flip-flops on reconfigurable devices," in *Proceedings of the 3rd Benelux Workshop on Information and System Security (WISSec '08)*, p. 17, Eindhoven, NL, USA, 2008.

[7] S. S. Kumar, J. Guajardo, R. Maes, G. J. Schrijen, and P. Tuyls, "The butterfly PUF protecting IP on every FPGA," in *Proceedings of the IEEE International Workshop on Hardware-Oriented Security and Trust (HOST '08)*, pp. 67–70, June 2008.

[8] Y. Su, J. Holleman, and B. Otis, "A1.6pJ/blt 96% stable chip-ID generating circuit using process variations," in *Proceedings of the 54th IEEE International Solid-State Circuits Conference (ISSCC '07)*, pp. 406–611, San Francisco, Calif, USA, February 2007.

[9] D. E. Holcomb, W. P. Burleson, and K. Fu, "Power-up SRAM state as an identifying fingerprint and source of true random numbers," *IEEE Transactions on Computers*, vol. 58, no. 9, pp. 1198–1210, 2009.

[10] S. Chaudhry, P. A. Layman, J. G. Norman, and J. R. Thomson, "Electronic fingerprinting of semiconductor integrated circuits," US patent 6,738,294, Agere Systems Inc., 2002.

[11] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith, "Fuzzy extractors: how to generate strong keys from biometrics and

other noisy data," *SIAM Journal on Computing*, vol. 38, no. 1, pp. 97–139, 2008.

[12] O. Sander, B. Glas, L. Braun, K. Müller-Glaser, and J. Becker, "Intrinsic identification of xilinx virtex-5 fpga devices using uninitialized parts of configuration memory space," in *International Conference on Reconfigurable Computing and FPGAs (ReConFig '10)*, pp. 13–18, December 2010.

[13] Xilinx Inc., *UG190: Virtex-5 FPGA User Guide*, 2009, v5.2, November 2009.

[14] Xilinx Inc., *UG191: Virtex-5 FPGA Configuration User Guide*, 2009, v3.8, 14.08.2009.

[15] IEEE, "1149.1: IEEE Standard Test Access Port and Boundary-Scan Architecture," IEEE-SA Standards Board, IEEE Standard 1149.1-2001 (R2008), 2006.

[16] Digilent Inc., *DPCUTIL Programmer's Reference Manual*, 2007, doc 576-000, August 2007, http://www.digilentinc.com/Data/Software/Adept/DPCUTIL_Programmers_RM.pdf.

[17] Xilinx Inc., *DS100: Virtex-5 Family Overview. Product Specification, 2009, v5.0*, February 2009.

[18] Xilinx Inc., *UG347: ML505/ML506/ML507 Evaluation Platform: User Guide, 2009, v3.1.1*, October 2009.