INCREASING STABILITY AND DISTINGUISHABILITY OF THE DIGITAL FINGERPRINT IN FPGAS THROUGH INPUT WORD ANALYSIS

Hiren Patel *

Air Force Research Laboratory
United States Air Force
1940 Allbrook Drive
Wright-Patterson AFB, OH, 45433-5309
email: hiren.patel@us.af.mil

Yong Kim, J. Todd McDonald, LaVern Starman

Air Force Institute of Technology
2950 Hobson Way
WPAFB, OH, 45433-7765
email: {ykim, jmcdonald, lstarman}@afit.edu

ABSTRACT

Field Programmable Gate Arrays (FPGAs) have become increasingly popular in circuit development due to their rapid development times and low costs. With their increased use, the need to protect their Intellectual Property (IP) becomes more urgent. The digital fingerprint accomplishes this by creating a unique ID for each FPGA. In this research, we propose methods to dramatically increase the stability and robustness of the digital fingerprint ID by the proper choice of input sequences. We also show that by properly choosing the input word, we can significantly increase the DF resistance to operating temperature changes.

1. INTRODUCTION

FPGAs are becoming more widely used in commercial applications. They can be easily programmed with the user's circuit and therefore vastly increase development times over traditional ASIC development. However, FPGA flexibility comes at the cost of certain vulnerabilities. IPs programmed on the FGPA are only as secure as the method used to protect them. By attaining the FPGA bitstream, an attacker can create unlimited copies of a proprietary circuit. Sensitve IPs programmed on the FPGA are in this way vulnerable. As a result, the need to protect FPGAs has become very important.

FPGAs can be authenticated by the use of a unique, unforgeable ID. This ID can be integrated into the systems such that the sensitive IPs will only function if the correct ID can be verified. If the FPGA bitstream is obtained, the unforgeable ID would not be copied and thus render the bitstream useless on cloned FPGAs.

The digital fingerprint methodology was developed in [1]. This method uses the transitional glitches generated by a combinational circuit to create a circuit ID. These IDs are unforgeable and cannot be hacked by traditional software reverse engineering as the ID is not stored in memory. In [2], we generated 60 circuits on FPGAs and verified that the IDs geenrated for each of them were unique.

This research develops the concept of the digital fingerprint by determining how to improve stability of the digital fingerprint through the use of proper input sequences.

The remainder of this paper is divided into the following categories: (a) background into the digital fingerprint and similar work, (b) theory and methodology of increasing the robustness and stability of the digital fingerprint, (c) results of testing, and (c) conclusions from this research.

2. BACKGROUND

Transistors that are designed to be identical will in reality vary slightly in their dimensions. Transistor variations are a product of the unique and random motions of atoms. Although considerable effort is expended in developing advanced masking and doping techniques, the diffusion of atoms required to create transistors can never be fully controlled. The exact depth and shapes of the different layers of each transistor will vary slightly from each other. The exact shape of each transistor is therefore unique and random. These variations will manifest into slight differences in performance characteristics of each transistor, such as differences in threshold or saturation voltage. These characteristics can be used to our advantage in generating a unique ID for each FPGA.

Recently, work done in Physically Uncloneable Functions (PUF) has used transistor variations to generate unique IDs [3][4][5]. Transistor variations between identically designed circuits will lead to slight differences in their operation. For example, in the Ring Oscillator (RO) PUF, transistor variations in identical ROs will lead to differences in their frequencies. By comparing the different frequencies, the PUF generates a 0 or 1 value based on which RO is faster.

*The views expressed in this article are those of the authors and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U.S. Government.

The digital fingerprint offers an alternative approach by using differences in the transient glitches produced by combinational circuits to produce an ID. In [1], glitches were generated at the outputs of a 64 bit combinational multiplier on an FPGA. These glitches were counted with the use of a 16-bit one-hot state shift register, where the first bit is set to 1 and remaining bits set to 0. These *glitch counts* were then used to create the digital fingerprint ID. It was found that for the sample set of 60 circuits, 100% were uniquely identifiable. An important advantage of the digital fingerprint methodology is that each output has the potential to record upto 16 glitches. This translats to a 4 bit number per output, as a glitch count of 16 = 2^4. Thus the 64 output lines of the multiplier creates upto a 256 bit ID.

The ID generated by the digital fingerprint is not stored in any memory on the FPGA. Instead, it is generated by the natural operation of the FPGA circuits. As a result, it is resistant to traditional software hacking attacks. In addition, the digital fingerprint ID is a function of several factors. Different input combinations can generate different IDs. The ID is also dependent on the exact combination of all the transistors in the circuit. Also, different circuits can be used to generate glitches. Here we have used a combinational multiplier. Finally the recording method is essential to generating the ID. Here we have used the one-hot state shift register to produce the ID. However, another scheme such as two-hot state shift register can give a biased ID. Thus the digital fingerprint is dependent on several factors and represented by the following equation:

$$\text{Digital Fingerprint ID} = (i, t, c, r) \tag{1}$$

Here, *i* is the input combination, *t* is the exact transistors used, *c* is the type of combinational circuit used, and *r* is the recording method. Thus, in order for an attacker to successfully copy the digital fingerprint, he would need all of these factors. The attacker would need to exactly copy each transistor used, which is not possible because the transistor variations are smaller than the transistor technology. In addition he would need to know the type of combinational circuit used for glitch generation. The correct input word sequence and the output recording method would also be necessary. For all practical purposes obtaining this combination of data about the digital fingerprint is impossible and hence makes the ID generated by the digtial fingerprint highly resistant to being cloned.

The digital fingerprint ID is made up of two important characteristics: distinguishability and stability. Distinguishability is a measure of how robust the ID is. It determines how capable the ID is to uniquely identify a large number of FPGAs. Stability determines how likely the ID of the same FPGA will remain the same over multiple polls. By analyzing the structure of the combinational multiplier used in the digital fingerprint, we found that the choice of the input words used had a significant effect on the digital fingerprint performance. We propose a novel way to improve both distinguishability and stability of the digital fingerprint through properly chosen input values.

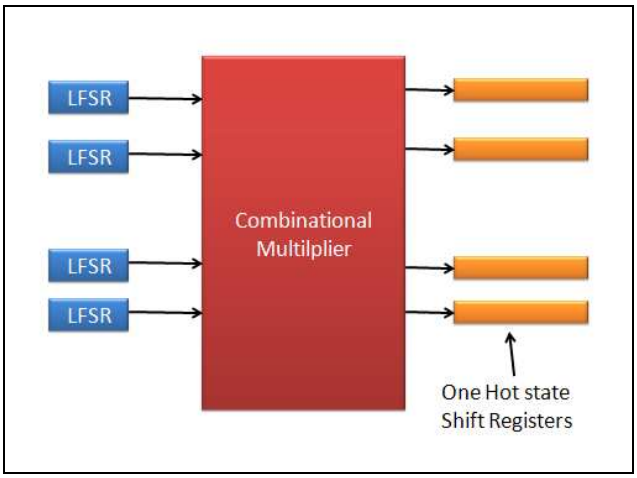


Fig. 1. Block diagram of the basic digital fingerprint circuit. Figure shows the LFSR input generation stage, the combinational multiplier that produces the glitches and the one-hot shift registers to record the glitches.

3. METHODOLOGY

All experiments were conducted using the Xilinx Virtex 2 Pro FPGA.

The digital fingerprint consists of three main parts for ID generation as shown in figure 1: (1) input generation, (2) glitch generation, and (3) glitch recording. Input generation is achieved through the means of a Linear Feedback Shift Register (LFSR). The LFSR is a sequential circuit that provided all 64 inputs to the next stage at the start of the common FPGA clock cylce. This allowed all inputs to be provided to the next stage at once. In this way, race conditions in the input generation process between different input bits were eliminated. Glitch generation was accomplished with a 64 bit combinational multiplier. The multiplier took in inputs from the LFSR and generated glitches at each of its 64 outputs. These outputs were tied to the clocks of 16-bit shift registers that were initiated to the one-hot state. Thus, a glitch would act like a clock pulse, shifting the contents of the shift register one space per glitch. The number of glitches produced could be recorded by noting the final position of the one in the shift register.

In addition, the digital fingerprint pollng and recording was automated using the Xilinx PowerPC. The input loads of the LFSR were set using the C code provided by the PowerPC. This allowed us to quickly change input values without having to resynthesize our entire design, which could

take as long as one hour for each input change.

As this is an arithmetic circuit, certain input words will lead to higher switching than others. This can then lead to a better ID. In the following sections, we will highlight how we determined which input combinations can give the best digital fingerprint ID.

3.1. Distinguishability

Distinguishability is a measure of how well the ID is able to distinguish a large population of FPGAs. In order to determine distinguishability, it is necessary to know the likelihood of *ID collision*. ID collision is the condition where the IDs of two or more distinct chips are the same. If the ID is sufficiently large, we can dramatically decrease the probability of ID collision. Work done in [6] by Su et al details the following likelihood of ID collision. If there are X bits in the ID, then there are 2^X distinct ID possibilities. Thus the probability of collision between two circuits is $2^{1/X}$. Then for the n^{th} circuit to be different from the remaining n-1 chips, the probability is given by

$$1 - \frac{n-1}{2^X} \tag{2}$$

Then if there are Y total chips in the population, the total probability that any of Y chips won't collide with any others is

$$P_{collision} = 1 - \prod_{n=1}^{Y} \left(1 - \frac{n-1}{2^X}\right) \tag{3}$$

For the 64 bit multiplier, there are 16 possible values represented by 4 bits (2^4 = 16), for each of the 64 outputs, giving a possible (64x4)=256 bits. If we set the number of bits X = 256, and the total number of chips in the population Y = 1,000,000 chips, then the possibility of ID collision is $4.318\text{x}10^{-66}$. Thus, creating a sufficiently large ID reduces the probability of ID collision to virtually zero.

The digital fingerprint counts the number of glitches produced at the outputs of a combinational multiplier to produce an ID. Thus if there are a larger number of glitches produced at the outputs, then a more robust ID is generated leading to a higher distinguishability. In this section, we propose methods to increase distinguishability of the digital fingerprint by simply choosing the appropriate input word.

In work done in [7], the most significant bits or MSBs of a multiplier input word were termed as *sign bits*. These sign bits were found to have the maximum effect on the power usage of the FPGA. Intuitively, this makes sense as the input words with high sign bits have the greatest magnitude. Multiplying large numbers together will give a larger product. In order to create a larger product, more parts of the multiplier will be used. This leads to higher switching in the multiplier and hence a larger power usage. We propose that there is a relation between higher switching and larger numbers of glitches produced by the multiplier. Thus by including more active sign bits in the input word, there should be higher numbers of glitches.

In order to determine which MSBs of the input words were sign bits, we set all bits of the inputs to high or 111.... Then we shifted 0's into the input starting at the MSB. Thus we got the input sets: 1111→0111→0011, etc. For each shift the maximum glitch counts generated by the digital fingerprint were recorded. As the input word magnitude decreased with each 0 shifted in, the number of glitches were expected to go down. However, for the sign bits, the rate of decrease in the maximum glitch count will be higher than for the rest of the system. Thus we can determine the sign bits of the input word. By keeping these sign bits high in the system, we are able to increase the number of glitches produced by the digital fingerprint and have a more robust ID.

3.2. Stability

The digital fingerprint was found to successfully generate 60 unique IDs for each of 60 different circuits. However, as with similar methods [6][5], there was instability in the system. We determine stability by counting the number of unstable lines in the system. As shown in similar work [6][5], an ID generated by transistor variations will have certain outputs that are unstable. The ID value for these unstable outputs will fluctuate between different samples. This is due to the fact that glitches generated by the digital fingerprint are of varying durations. If a glitch is of sufficient duration, it will properly meet the timing constraints of the shift register and be correctly recorded. If the glitch is too short, then it will not be recorded. In Figure 2, a combinational multiplier output is attached to a shift register and simulated in Eldo Spice. Of the 9 glitches produced, only

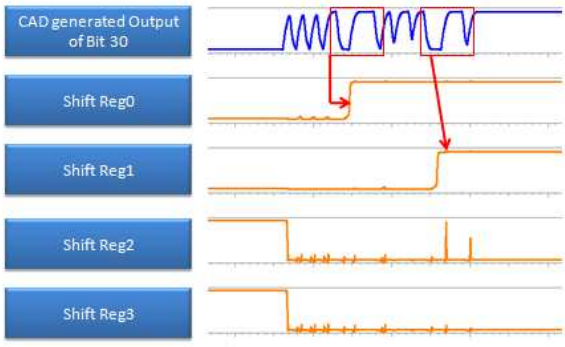


Fig. 2. Simulation of a multiplier output attached to a shift register. Of the 9 glitches produced, only two met the timing constraints of the shift register and were recorded.

two were long enough to meet the shift register timing constraints. In the third case, if the glitch barely meets the timing constraints, then it may or may not be properly recorded. Outputs that create glitches of this third type give unstable readings. These outputs are termed as unstable outputs.

In order to determine the unstable outputs of the digital fingerprint, each bit was sampled 1000 times and total glitch count was summed. If the average glitch count at an output was 3, then the ideal sum would be 3000 when that line was sampled 1000 times. If the measured glitch count was 3003, then it could be concluded that the line gave a different value 3 times out of 1000 samples. We conservatively categorized outputs that were unstable more than 1% over 1000 samples as unstable.

We determined inputs that increased stability by studying the structure of the combinational multiplier. The analysis reveals that it is made up of n rows of n-bit ripple carry adders, where n is the number of bits in the input word. Figure 3 displays an 8 bit multiplier with a value of 1001 given to the B input. As the value of B1 is set to 0, the result of the AND(B1,x) = 0, where x$\in${A0,A1,A2,A3}. As a result, there is lower activity in this row, which allows the glitches produced in the B0 row (where B0 = 1) to be fully propogated down to the next level. Thus, the glitches generated by the B0 row are not intersected by glitches from the B1 stage as the B1 stage has lower glitching. In the following, 0 value inputs are referred to as *inactive* inputs and 1 value inputs are referred to as *active* inputs. Thus, by including more inactive inputs between active inputs, we see that glitches can be fully developed. Figure 4 shows the Spice simulation results of the combinational multiplier for an input with smaller number of inactive inputs (1001... input pattern) and greater number of inactive inputs (1000001... input pattern). The 1001... input pattern will be termed the *baseline inputs* and the 1000001... input pattern will be termed the *improved inputs*. The glitches generated by the baseline inputs in Figure 4(a) shows a large number of short duration glitches. These would normally be either too short to be

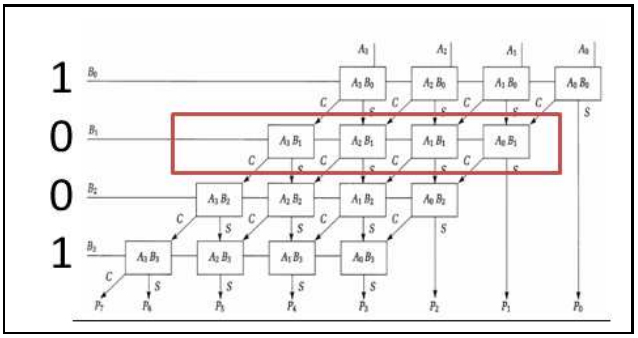
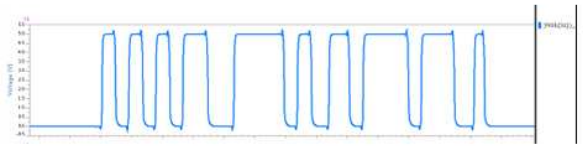
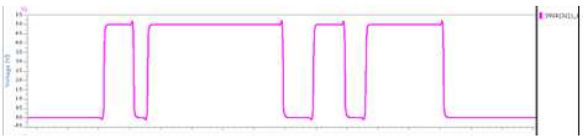


Fig. 3. Simulation of a multiplier output attached to a shift register. Of the 9 glitches produced, only two met the timing constraints of the shift register and were recorded.



(a)



(b)

Fig. 4. (a): Glitch pattern of output bit 31 for the 101... input, (b): Glitch pattern of output bit 31 for the 1000001... input

recorded or on the border of meeting shift register timing constraints. As a result, this output line would have increased instability. Using the improved inputs, the glitches produced are shown in Figure 4(b). Here we see that the glitches generated are longer in duration and thus well defined. These longer glitches have a high probablility of meeting the shift register timing constraints and therefore have a higher probability of getting adequetely recorded. This higher probability with the improved inputs translates to a much more stable output. Thus using improved inputs, stability of the digital fingerprint outputs increases.

4. RESULTS

The following sections describes the results gained through the input variation tests in distinguishability and stability.

4.1. Distinguishability Results

The input word was set to all high bits, i.e. xFFFFFFFF for a 32 bit input word. Then 0's were shifted in from the MSB. For each shift, the maximum glitch count produced at the output lines of the digital fingerprint circuit were recorded. Readings were taken over 60 circuits and averaged to reduce the effect of circuit anomalies.

As expected, the maximum glitch count decreased as the number of 0's shifted in from the MSB of the input word increased, as seen in figure 5. However, for the first four MSBs, the rate of decrease was 311% faster than for the rest of the input word. Therefore, it was concluded that the first 4 MSBs of the input word of the digital fingeprint were the sign bits. By removing these first four MSBs from the input word, we saw the greatest drop in the number of glitches produced by the digital fingerprint. Conversely, including

these sign bits as active will give higher glitch counts at the outputs and hence higher distinguishability. Using sign bits in the input word can lead to significant increases in robustness of the ID. By including these outputs as high in the input word of the digital fingerprint, there were was an average 7.35% higher total numbers of glitches produced than without them.

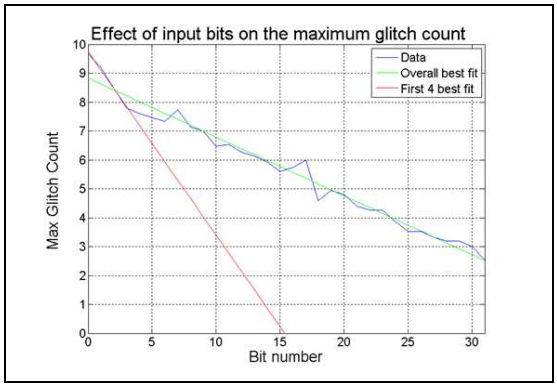
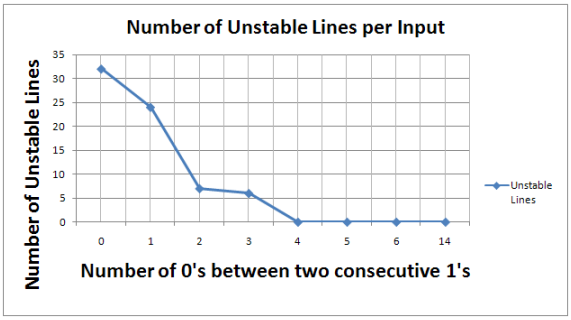
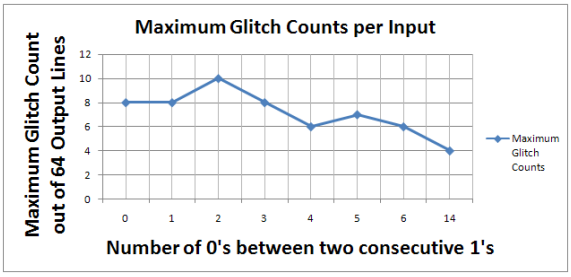


Fig. 5. The average maximum glitch count produced when the input starts at xFFFFFFFF and 0's are shifted in from the MSB.

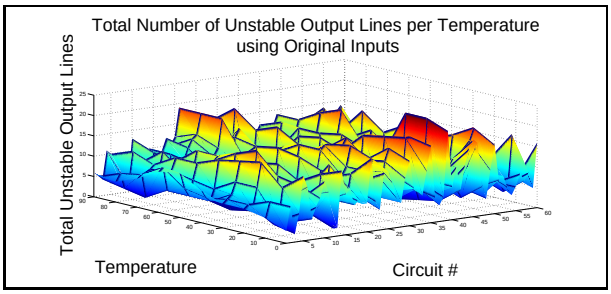


(a)

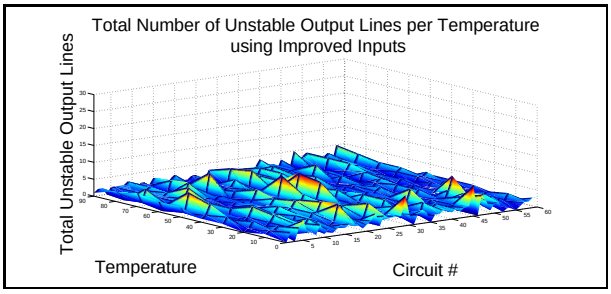


(b)

Fig. 6. (a):The number of unstable output lines in a 64 bit multiplier as the N_{zero} increased. (b) The maximum glitch count in a 64 bit multiplier as the N_{zero} increased



(a)



(b)

Fig. 7. (a): The number of unstable output lines and their change as temperature increases. Original inputs are used. (b) Stability plot with improved inputs. For both cases, a stable line was defined as one that changed its value less than 1% of the time

4.2. Stability Results

For brevity, the number of inactive 0 inputs between consecutive active 1 inputs will be termed N_{zero}. It was hypothesized that a larger N_{zero} in the input word would lead to greater stability. Initially, inputs with all 1's (i.e. xFFFFFFFF for the 32 bit input) was chosen and the number of unstable lines counted. Then, the number of 0's between consecutive 1's in the inputs were increased giving input combinations of 101..., 1001..., 10001..., etc. Results are showin in figure 6. Figure 6(a) shows that stability increased as N_{zero} increased. However, there was a saturation point beyond which increasing N_{zero} did not improve stability. Figure 6(b) shows that as N_{zero} increases, the maximum glitch count in general also goes down. Thus, distinguishability of the digital fingerprint decreases with N_{zero}. A N_{zero} value of 5 was chosen to give the highest stability with lowest drop in distinguishability. This translates to an input word with the 1000001... pattern.

Figure 7 shows the number of unstable outputs for the 60 circuits tested over the full range of operating temepratures from 0°C to 90°C for the Xilinx Virtex 2 Pro FPGA [8]. Figure 7(a) shows the results using the baseline inputs. Readings showed an average of 8.74 unstable outputs per circuit over all temperatures out a total of 64 outputs. Figure 7(a) shows the results using the improved inputs. There

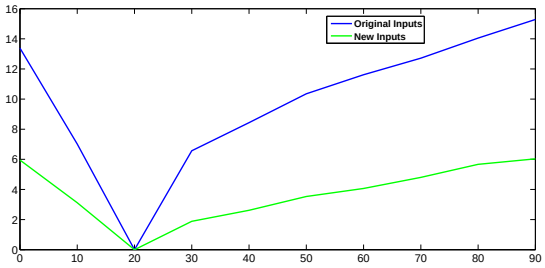


Fig. 8. Average ΔL per temperature change with original and improved inputs

was an average 464% increase in stability of the system over the full range of temperatures. In addition, the average went down to 1.8833 unstable outputs for a circuit over all temperatures.

Figure 8 shows the total number of outputs that changed their values (ΔL) when the operating temperature was changed. The blue line shows the results using the baseline inputs. Results show that as the temperature deviated from room temperature (20°C), the number of outputs that changed their values increased. Using the baseline inputs, we see that at 90°C, as many as 15 outputs changed their values. The green line represents the results using improved inputs. We see that the effect of temperature on the ID is dramatically decreased when using improved inputs. At the highest temperature, ΔL went down from 15 with baseline inputs to 6 with improved inputs. Thus, the use of improved inputs in the digital fingerprint increased the resistance of the digital fingerprint ID to operating temperature changes.

5. ADVANTAGE OF THE UNKNOWN INPUT SEQUENCE

The input value determines the ID generated by the digital fingerprint. Thus if somehow the digital fingerprint ID was compromised, the FGPA could still be authenticated through the use of a different input sequence. The two 32 bit input values used by the digital fingerprint thus offer 2^64 possible digital fingerprint IDs. This number is reduced if we use inputs that will increase stability and distinguishability. However, if we keep the criteria of having 4 sign bits high and N_{zero}¿5, we still have 148225 possible input combinations, each of which offers a unique ID. Thus the digital fingerprint has the capability of producing a high number of different IDs simply by changing the input values. This allows for highly secure and adaptable ID generation with minimal effort.

6. CONCLUSION

The digital fingerprint successfully creates unforgeable unique IDs for each FGPA. By integrating the digital fingerprint in FPGA IP design, proprietary IPs can be protected from being cloned and used illegally. In this paper, we showed methods to dramatically increase the distinguishability and stability of the digital fingerprint. By the simple method of choosing high sign bits, we increased distinguishability 7.35%. In addition, by choosing the improved inputs for stability, we improved stability by 464%. In addition, we increased the digital fingerprint ID's resistance to changes in operating temperature through the use of improved inputs. In this way we provide the user with a simple technique to improving the generated ID without the need for additional large error correction circuitry.

7. REFERENCES

[1] J. Crouch and H. Patel, "Creating unique identifiers on field programmable gate arrays using natural processing variations," *FPL*, 2008.

[2] H. Patel, J. Crouch, Y. Kim, and T. Kim, "Creating a unique digital fingerprint using existing combinational logic," *Submitted to ISCAS*, 2009.

[3] G. E. Suh, "Physical unclonable functions for device authentication and secret key generation," *DAC '07: Proceedings of the 44th annual conference on Design automation ; ACM*, -06 2007, doi: pmid:.

[4] R. G. Bolotnyy L., "Physically unclonable function - based security and privacy in rfid systems," *2007 IEEE International Conference on Pervasive Computing and Communications*, p. 8 pp., 2007, doi: pmid:.

[5] S. S. Kumar, J. Guajardo, R. Maes, G.-J. Schrijen, and P. Tuyls, "The butterfly puf protecting ip on every fpga," Piscataway, NJ 08855-1331, United States, 2008, pp. 67 – 70.

[6] Y. Su, J. Holleman, and B. Otis, "A 1.6pj/bit 96circuit using process variations," *Solid-State Circuits Conference, 2007. ISSCC 2007. Digest of Technical Papers. IEEE International*, pp. 406–611, Feb. 2007.

[7] J. A. Clarke, A. A. Gaffar, G. A. Constantinides, and P. Y. K. Cheung, "Fast word-level power models for synthesis of fpga-based arithmetic," in *Proceedings - IEEE International Symposium on Circuits and Systems*, Piscataway, NJ 08855-1331, United States, 2006, pp. 1299 – 1302.

[8] *Virtex-II Pro and Virtex-II Pro X Platform FPGAs: Complete Data Sheet*, 4th ed., Xilinx, November 2007.