Reconfigurable Architectures for Silicon Physical Unclonable Functions

Yingjie Lao and Keshab K. Parhi
Department of Electrical and Computer Engineering,
University of Minnesota, Twin Cities
{laoxx025, parhi}@umn.edu

ABSTRACT

Physical Unclonable Functions (PUFs) are novel circuit primitives which store secret keys in silicon circuits by exploiting uncontrollable randomness due to manufacturing process variations. Previous work has mainly focused on static challenge-response behaviors. However, it has already been shown that a reconfigurable architecture of PUF will not only enable PUFs to meet practical application needs, but also can improve the reliability and security of PUF-based authentication or identification systems. In this paper, we propose several novel structures for non-FPGA reconfigurable silicon PUFs, which do not need any special fabrication methods and can overcome the limitations and drawbacks of FPGA-based techniques. Their performances are quantified by the inter-chip variation, intra-chip variation and reconfigurability tests.

Keywords: Physical Unclonable Function, Reconfigurable Architecture, Hardware Security, Counterfeit IC Chip Prevention

1. INTRODUCTION

1.1 Physical Unclonable Function

In today's world, as electronic devices become increasingly interconnected and pervasive in people's lives, security, trustworthy computing, and privacy protection have emerged over the past decade as hardware design objectives of great significance. Traditionally, secret keys, which are used as unique identifiers, are embedded into integrated circuits (ICs) in a ROM immediately after manufacturing. Unfortunately, digital keys stored in a non-volatile memory are vulnerable to physical attacks. Several invasive and semi-invasive physical tampering methods have been developed, these include techniques such as micro-probing (access to the silicon to manipulate the internals of system), power analysis (predict the secret keys from power consumption analysis) and so forth. These approaches have made it possible to learn the ROM-based keys through attacks and compromise systems by using counterfeit copies of the secret information.

The described problem has become more intense recently, and this motivated the idea of using intrinsic random features of physical objects for identification and authentication. The concept of physical unclonable function (PUF) proposed in [1–3] has successfully addressed the problems faced by traditional techniques. PUF has been defined as a function that exploits the unique intrinsic uncontrollable physical features by process variations during manufacturing. Physical Unclonable Functions enable significantly higher secure authentication by extracting secrets from complex properties of a physical material rather than storing them in non-volatile memory. Due to the uncontrollable random components, PUFs are easy to measure but almost impossible to clone, predict, or reproduce. Furthermore, it is infeasible for an adversary to mount an attack to counterfeit the secret information without changing the physical randomness. Taking coating PUF as an example, which is a function built in the top layer of an IC by filling the space between and above the comb structure with an opaque material and randomly doping with dielectric particles, any physical attack on a coating PUF would damage the protective coating and destroy the cryptographic key.

1.2 Related Work and Our Contribution

The first PUF in the literature is the optical PUF [2], which utilizes the randomness in the placement of the light scattering particles and the complexity of the interaction between the laser and the particles. After that, several PUF hardware structures have been proposed [1, 3–6]. Most PUFs use conventional silicon techniques so that they do not require any special fabrication and can be easily integrated into IC chips, except a few types such as coating PUF and magnetic PUF. Among these PUFs, silicon PUFs are of great interest, as these exploit manufacturing variability of wire delay to generate a unique challenge-response mapping for each IC. These unique properties of each IC are easy to measure through the circuits but hard to copy without changing the challenge-response pairs (CRPs).

The delay-based silicon PUFs in previous work have always considered a static challenge-response behavior. In those protocols, the PUF should always generate the same or error tolerated response. Unfortunately, recent analysis has demonstrated that those PUF structures are vulnerable to several attack methods including emulation, replay (man-in-the-middle attack), and reverse engineering [7]. Moreover, updatable cryptographic keys are very attractive in some applications [8]. Therefore, a dynamic PUF that can alter the CRPs every time the data is modified to prevent the hidden information leaked out is very desirable.

Our work builds on the prior work of the PUF community. In this paper, we mainly focus on the design of reconfigurable silicon PUFs. We propose several novel reconfigurable PUFs and analyze their performance. We also examine the security of different PUF structures. The key idea in our approach is that we try to make CRPs updatable. By doing this, the challenge-response behavior of a PUF can be altered to generate highly secure hardware system. Furthermore, we discuss the techniques to improve the reliability of silicon PUFs.

1.3 Paper Organization

The rest of the paper is organized as follows. In Section 2, we introduce the background of silicon PUFs, and then present a brief overview of previous works on reconfigurable PUFs and discuss their disadvantages and limitations. In Section 3, we describe our manufacturing process variation model for silicon PUFs and the experimental methods. Section 4 discusses several novel reconfigurable PUF designs and analyzes their efficiency. Section 5 demonstrates the performance of the discussed reconfigurable structures by providing experimental results on SPICE simulation. Finally, Section 6 concludes the paper.

2. BACKGROUND

2.1 Silicon Physical Unclonable Function

Silicon PUFs exploit the delay variations of CMOS logic components to generate a unique response for each IC. There are two main types of delay-based silicon PUFs: Ring Oscillator (RO) PUF [1] and Multiplexor (MUX) PUF [9]. However, the MUX PUF is more secure than the RO PUF, as the frequencies of the ring oscillators can be relatively easily evaluated by attackers; moreover, a MUX PUF is more suitable for resource-constrained applications. Instead of duplicating the hardware N times as in a RO PUF, we can use N different challenges to obtain a N-bit long response in a MUX PUF, as illustrated in Figure 1. This kind of silicon PUF consists of N stages MUXs and one arbiter which connects the final stage of the two paths. MUXs in each stage act as a switch to either cross or straight propagate the rising edge signals, based on the corresponding challenge bit. Each MUX should be designed equivalently, while the variations will be introduced only during manufacturing process. Finally, the arbiter (always simply a D flip-flop) translates the analog timing difference into a digital value. For transistors, manufacturing randomness exists due to variations in transistor length, width, gate oxide thickness, doping concentration density, metal width, metal thickness, and ILD (inter-level dielectric) thickness, etc [10]. These manufacturing variations show a significant amount of variability, which are sufficient to generate unique challenge-response pairs for each IC by comparing the delays of two paths.

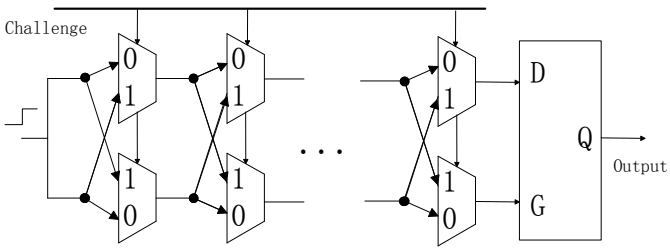

Figure 1: Silicon MUX Physical Unclonable Function.

2.2 Feed-Forward Structure

A feed-forward structure of silicon PUF has been proposed in [11] to prevent attacks by linear modeling. Figure 2 shows one basic structure of feed-forward MUX PUF, which uses the racing result of an intermediate stage as the select signal for a block of MUXs in a later stage. This structure provides nonlinearity to the original PUF, which increases the complexity for numerical modeling attacks. However, the reliability of the PUF has been degraded in this feed-forward structure since an error in the output of an internal feed-forward arbiter caused by environmental variation can increase the noise probability in the final response.

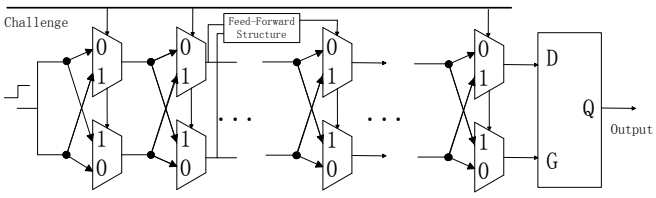

Figure 2: Feed-Forward Silicon MUX PUF Structure.

2.3 Reconfigurable PUF

A reconfigurable PUF is a PUF with a mechanism to update its challenge-response pairs. It should preserve the properties of original PUF but have unpredictably different challenge-response behaviors after every reconfiguration. The first reconfigurable PUF [9] was presented in 2004, which is an arbiter-based PUF built with floating gate transistors. It was also shown that the reconfigurability for a PUF is a very desirable characteristic.

Recently, two types of reconfigurable PUFs have been proposed in [12]: reconfigurable optical PUF and phase change memory based reconfigurable PUF. These two types of PUF can update their CRP mappings while inheriting the properties of the original PUF. However, these PUFs have certain limitations and constraints for practical use; for instance, the reconfigurable optical PUF needs a special material, which is not widely used in industry.

Several implementations of reconfigurable silicon PUFs have also been published [8, 13–15]. However, all of these efforts are based on the reconfigurability of FPGA, and most of them are built on the Ring Oscillator PUF. In these existing FPGA-based solutions, many assumptions are required as significant amount of information regarding the underlying VLSI layout of the reconfigurable fabric is lost. Moreover, many PUF designs require a symmetrical routing that is difficult to implement on an FPGA platform as a designer can only manipulate the higher level design blocks such as the LUTs, the memory blocks, and the connection matrices. Additionally, it is possible to reverse the reconfigurations of contemporary FPGAs, which makes the FPGA-based reconfigurable PUF more vulnerable to attacks.

Therefore, non-FPGA based architectures for reconfigurable silicon PUF are very desirable. Our major contribution is to present several novel structures of silicon PUFs which can be implemented on non-reconfigurable hardware, that is which can be designed at transistor level and fabricated and integrated into chips. Moreover, we also target on the security and the reliability perspectives of the reconfigurable PUFs.

3. METHODOLOGY

3.1 PUF Model

As shown in Figure 1, a MUX PUF consists of a sequence of N-stage MUXs and an arbiter. The rising edge signal will excite the two parallel paths simultaneously. The actual propagated paths will be determined by the external applied challenge bits. After the last stage, the arbiter will generate the output bit by comparing the arrival time of the two different paths. It has become standard to model the MUX PUF via an additive linear delay model. According to the efforts in the field of Statistical Static Timing Analysis (SSTA) [10], the manufacturing process parameter variations for transistors can be modeled by a Gaussian distribution. As a result, the variations of delay will also be approximately Gaussian.

Manufacturing process variations can be classified as the following two categories: inter-die variations and intra-die variations. Inter-die variations refer to parameter variations that affect all devices equivalently across a single die, while intra-die variations have different effects on the devices within the same chip. It is also imperative to consider the correlation of these variations to increase the accuracy of the model. A very widely used model for delay spatial correlation of process variations is the Grid model [10], which assume high correlations among the devices in nearby grids and low correlations in faraway grids, as manufacturing process variations are more likely to have similar effects on closer devices. Additionally, experimental results have already shown that the inter-chip variation across the wafer is similar to that within a single wafer [9]. Moreover, the output of the arbiter in silicon PUF is only based on the difference of two selected paths. Therefore, the inter-chip variations are the primary factors that contribute to the randomness of response for each IC, while these die-to-die and wafer-to-wafer manufacturing variations will have minimum effect on the output response.

For simplicity, as every MUX is designed equivalently in a MUX PUF, we can model the delay of each single MUX as i.i.d. random variable D_i, which follows $N(\mu, \sigma^2)$; therefore, the total delay of

the N stages will be $N(N\mu, N\sigma^2)$. Since the output of arbiter will only depend on the delay difference between the two paths, the time difference will also follow a Gaussian distribution $\Delta \sim N(0, 2N\sigma^2)$.

We denote the delay in the top path of the i-th stage as Dt_i, the delay in the bottom path of the i-th stage as Db_i, and the challenge bit for each stage as C_i. Thus the delay difference of the i-th stage will be:

$$Dt_i - Db_i \sim N(0, 2\sigma^2)$$

Then if the challenge is 0, then the delay difference added into the whole paths will be $Dt_i - Db_i$; otherwise, if the challenge bit for the i-th stage is 1, the additive delay difference will be $Db_i - Dt_i$. It can be expressed as:

$$\Delta_i = (-1)^{C_i}(Dt_i - Db_i) \sim N(0, 2\sigma^2)$$

As a result, the arrival time difference between the two inputs of the arbiter is:

$$\Delta_z = \sum_{i=1}^{N}(-1)^{C_i}(Dt_i - Db_i) \sim N(0, 2N\sigma^2)$$

Thus, the final response is:

$$r = sign(\Delta_z)$$

where we use the convention that $r = sign(a) = 0$ when $a < 0$, and $r = sign(a) = 1$ when $a \geq 0$.

3.2 Simulation Model

In our experiment, we use simulation method to test and analyze the performances of PUFs instead of fabrication method. There are several advantages of using simulation method: First of all, fabrication is relatively expensive. Second, a good simulation method can be used as a pre-fabrication test, which can predict the efficiency of a new PUF design before fabrication. Moreover, we can analyze all the possible properties and characteristics of the PUFs under different environmental conditions. Additionally, it is also convenient to follow the shrinking of technology scale.

In our simulation, we apply the Gaussian model which has already been described in Section 3.1 for manufacturing process variations. We set up the process parameters and their max percentages of deviations based on the predictions from [16, 17]. For spatial correlation, we assume perfect spatial correlations within one single MUX. The process variations will have the same effect on the PMOS and NMOS devices in each MUX, while the parameter variations among different MUXs have no correlation.

In our simulation result, the total delay deviation of 100 stages is $\leq \pm 0.4\%$. Since

$$\sigma_z/\mu_z = \sqrt{(1/N)}(\sigma/\mu)$$

and μ_z increases linearly with N, we can conclude that our result conforms with other published results of 65nm technology, based on the experimental results in [8] that $3\sigma/\mu \approx 5\%$ for a single stage of MUXs. Furthermore, our simulation result of inter-chip variation leads to a Hamming distance range from 22 to 59 bits for a total of 100 stages, while the intra-chip variation is 5.8 bits on average, with a maximum value 13 bits. These results are also in agreement with published results for fabricated chips. Thus, we believe that our simulation delay model is consistent with the industrial manufacturing process variations.

4. NOVEL RECONFIGURABLE PUFS

In order to add reconfigurable property into general MUX based silicon PUFs, we must make the challenge-response pairs (CRPs) reconfigurable, which can be used to update the database for an authentication system. The methods can be classified into two categories:

(a) Make the challenge-response pairs reconfigurable directly, by adding some extra circuits into the structure, but without configuring the main PUF circuit. This can be achieved by utilizing some techniques to pre-process the challenge before applying to PUF or pre-process the response before using it for authentication.

(b) Make the PUF circuit reconfigurable, therefore the challenge-response pairs will be reconfigurable as well.

We propose several novel non-FPGA reconfigurable PUFs implementations for the above two categories, which would be more suitable for practical use than FPGA-based techniques. Furthermore, we address the reliability and the security of the PUF performance, as some information of the hidden secrets that an adversary can take advantage of may leak out during reconfigurations.

4.1 Reconfigurable Challenge and/or Response Structures

The reconfigurable structures of PUF are built on the prior work in Physical Unclonable Function, which can also be applied to various types of silicon PUFs as well as other challenge-response based PUFs. Our goal is to develop reconfigurable PUF which is a PUF with a mechanism to transform it into a new PUF with an unpredictable and uncontrollable challenge-response behavior, even if the challenge-response behavior of the original PUF is already known. Additionally, the new PUF inherits all the security properties of the original one.

An early reconfigurable design PUF [9] in the literature treated some challenge bits as the *configure data*. As an example, the last 10 bits of a 100-bit challenge can be fixed as the *configure data*, leaving only 90 bits for actual challenge. A user can update the CRPs by applying another 10-bit stream to the last 10 stages of the PUF. However, it is very clear that the reconfigured PUF will have high correlation between different configurations and will be vulnerable to attacks, as this method is similar to adding a certain time difference between the two paths or introducing an interval between the two rising edge signals. Even worse, the performance of the PUF will be greatly degraded, if the cumulative variations in the last 10 stages are relatively large. Due to these disadvantages, this architecture of reconfigurable PUFs cannot generate unpredictable challenge-response behaviors.

Intuitively, adding reconfigurable elements before the challenges applied to the PUF can definitely make the PUF reconfigurable. At the same time, the performance of the original PUF will be preserved. The main structure of this type of reconfigurable PUF is shown in Figure 3.

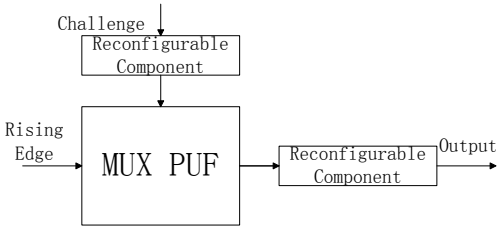

Figure 3: Reconfigurable Challenge and Response PUF Structure.

4.1.1 PUF with LFSR

We can adopt the linear feedback shift register (LFSR) as the reconfigurable component. Such a structure is shown in Figure 4. LFSR is an important part of sequence cipher and can be used to generate pseudo-random key stream. We can apply different seeds to the IC to generate various random patterns. Furthermore, we can also alter the characteristic polynomial by utilizing the properties

of reconfigurable linear feedback shift register [18, 19]. Such capability makes it extremely difficult for adversaries to obtain PUF signature. It is important to point out that we can improve the security of the PUF system, by benefiting from the property of the LFSR in cryptography.

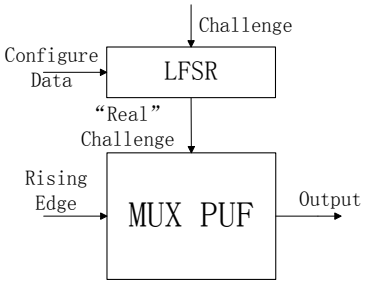

Figure 4: PUF Structure of Using LFSR to Configure the Challenge.

4.1.2 PUF with Hash Function

Hash function is a kind of "one-way" function, which means it is easy to compute the hash value for a given message, but hard to find a message with a given hash. Due to the random property of hash function, we can employ a hash function as the reconfigurable element to generate a reconfigurable PUF. This structure can be reconfigured very easily, such as by adding several different lengths of 0's at the end of every challenge. Additionally, the security of PUF can be increased, due to the "one-way" property of hash function. Many hash algorithms have been investigated and developed in the last years. Currently, the SHA-1 algorithm is the National institute of Standards and technology (NIST) secure hash standard. Several reconfigurable hash function unit architectures have been published in past years [20].

In fact, this structure has already been named as Controlled Physical Unclonable Function in [21], which was described as adding control logic to a PUF structure to prevent an adversary from accessing the PUF directly. Instead of doing a simple hash before the challenges applied to the PUF, we can consider adding another control logic, which would make the CRPs updatable. We propose several reconfiguration methods:

(a) Adding different bit streams into the challenges, $e.g.$, adding different numbers of 0's at the end of the challenges.

(b) Reordering the challenge stream by certain rules.

(c) Reconfiguring the hash function, by using the reconfigurability of these reconfigurable hash function implementations.

Due to the property of hash function, it is extremely hard for an adversary to model the PUF, even after we configure it several times, since the output of hash function is unpredictable.

4.1.3 PUF with Output Recombination

Another idea is to add an extra reconfigurable component to preprocess the output of the arbiter before using it as an authentication key. One simple example is to use two parallel MUX PUFs to update the CRPs, as shown in Figure 5. In this case, the signal (rising edge) will propagate through 4 paths which are selected by challenges. Then we can select two of the four paths using the *configure data* and forward to the arbiter to generate the response. We will have a total of 12 possible combinations if we use a 2 parallel MUX PUFs. Therefore, we can reconfigure this architecture 12 times. However, there will be very high correlations among these 12 different combinations. For example, if we know that path 1 is faster than path 2, and path 2 is faster than path 3, then we can conclude that path 1 will be faster than path 3. Therefore, there should be some constraints for the pre-processing, which will decrease the total number of reconfigurations. In fact, there are $N!$ possible cases for ordering N paths based on their arrival time. Therefore, $log_2(N!)$ independent bits can be produced by N paths. We can increase the number of parallel PUFs to obtain more possible combinations to meet the practical application needs. If we want to achieve the entropy limit as $log_2(N!)$, we need to choose the output comparison pairs adaptively, which would increase the design complexity and fabrication area significantly.

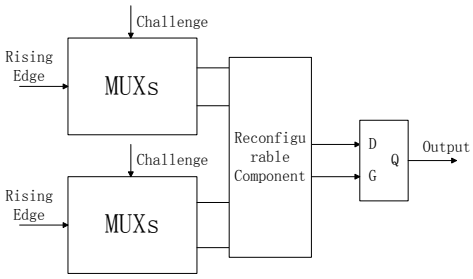

Figure 5: Two Parallel MUX PUF Structure.

However, there will be a problem by employing this structure, since the pre-processing component after the last stage also has variations, which will affect the performance of the PUF. To solve this problem, we can add pre-processing components after the arbiters, as in structure of Figure 6. If we use N parallel MUX-based PUF, we will need 2N-1 arbiters, where we only compare the neighbor paths. This is a concept borrowed from ring oscillator PUF which could ensure there will be no correlation between the output bits of the arbiters, as the comparison pairs are non-cyclic. Therefore, this structure can update its challenge-response behavior in an unpredictable manner.

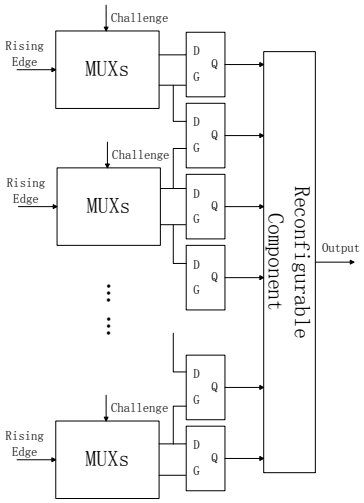

Figure 6: MUX PUF Structure of Output Recombination.

4.2 Reconfigurable Circuit Structures

Instead of only making the CRPs reconfigurable by processing the challenge and response directly, we can alter the main circuit to update the challenge-response behavior. This kind of reconfigurable PUFs will have better performance from the security perspective, since it leads to a different PUF circuit after reconfiguration, while the previous method only changes the CRP mapping.

The most important thing in these structures is to ensure the extra circuit will not affect the PUF performance, or more generally,

the extra circuit will have identical effect on the delay of different paths statistically. Otherwise, the behavior of the PUF can be easily predicted which leads to an insecure system.

4.2.1 Reconfigurable Feed-Forward PUF

It has been shown that the security of the MUX PUF in Figure 2 can be improved by adding feed-forward arbiters to it. However, in previous literature, how to choose the feed-forward stages and how many stages are chosen for feed-forward purpose have not been clearly presented. One constraint is trivial: the signal produced by the feed-forward arbiter should arrive earlier than the two signals propagating through the MUX paths. Therefore, we should ensure that there are at least 5-8 stages between stages connected to the input and the output of a feed-forward arbiter. We denote the stages from the input of a feed-forward arbiter to the output of the feed-forward arbiter as a feed-forward component. We consider the following three feed-forward structures:

(a) Feed-forward overlap: This structure has at least one stage overlap between two feed-forward components.

(b) Feed-forward cascade: In this structure, the last stage of a feed-forward component will be the first stage of another feed-forward component.

(c) Feed-forward separate: Here the different feed-forward components will be separated. Thus, there is no stage overlap between any two feed-forward components.

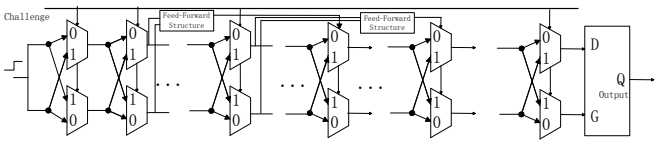

Figure 7: Feed-Forward Silicon MUX PUF Overlap Structure.

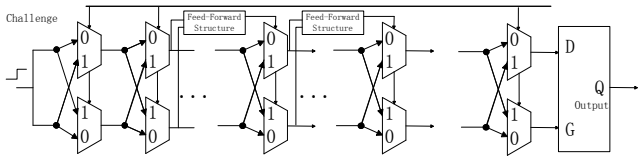

Figure 8: Feed-Forward Silicon MUX PUF Cascade Structure.

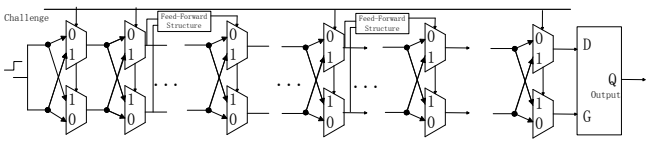

Figure 9: Feed-Forward Silicon MUX PUF Separate Structure.

In our experimental results, the intra-chip variations are increased by adding non-linearity to the circuits. Among the 3 different structures, the feed-forward cascade structure has the largest intra-chip variation, with 10.7 bit Hamming distance on average with response length of 100 bits, compared to 5.8 bits for non-feed-forward structure.

We can also examine the nature of these different structures theoretically. In the feed-forward structure, some of the challenge bits will be the intermediate stage arbiter outputs instead of the external bits. For example, if there is only one feed-forward component in a MUX PUF, which is from the a-th stage to the b-th stage, the time difference of the b-th stage could be expressed as:

$$\Delta_b = (-1)^{sign(\sum_{i=1}^{a}(-1)^{C_i}(Dt_i - Db_i))}(Dt_b - Db_b)$$

An error occurred in the output of the feed-forward arbiter will also affect the time difference in the b-th stage. Therefore, the error probability of the final response is increased by adding nonlinearity.

In the feed-forward cascade structure, the noise from an earlier feed-forward component will directly affect the output of the next feed-forward arbiter. Therefore, this structure will have the least reliability. From above, we know the time difference of the last stage of the feed-forward arbiter is Δ_b. We assume the c-th stage is the first stage of the next feed-forward component. Then the output of the second arbiter is:

$$\begin{aligned} C_c &= sign(\sum_{i=1}^{c-1}(-1)^{C_i}(Dt_i - Db_i) + \Delta_b) \\ &= sign(\sum_{i=1}^{c-1}(-1)^{C_i}(Dt_i - Db_i) \\ &\quad + (-1)^{sign(\sum_{i=1}^{a}(-1)^{C_i}(Dt_i - Db_i))}(Dt_b - Db_b)) \end{aligned}$$

.

It can be seen that the intra-chip variations are increased in this cascade structure, since the noise in earlier stages is more likely to cause the outputs of the later arbiters to flip.

For structure (a), the second feed-forward arbiter output of this overlap structure is:

$$C_c = sign(\sum_{i=1}^{c}(-1)^{C_i}(Dt_i - Db_i))$$

As each output of feed-forward structures will not be affected by the noise from feed-forward arbiters, we expect this structure will have the best reliability.

For structure (c), since there are several stages between two feed-forward components, the noise effect from feed-forward bit will combine with the path difference before the beginning point of the next feed-forward component. If the process variations of these stages are more significant than the noise effect from the feed-forward arbiter, the output of next feed-forward arbiter will also not be affected. The second feed-forward arbiter output of this separate structure is:

$$\begin{aligned} C_c = sign(&\sum_{i=1}^{b-1}(-1)^{C_i}(Dt_i - Db_i) + \Delta_b \\ &+ \sum_{i=b+1}^{c}(-1)^{C_i}(Dt_i - Db_i)) \end{aligned}$$

From above, it can be seen that the mathematical models for the 3 feed-forward structures are different. We find that feed-forward arbiters also enable us to reconfigure the circuit as well as to improve the performance from a security perspective. A basic reconfigurable feed-forward structure that combines overlap and separate approaches is shown in Figure 10.

The structure can be configured among the 3 different structures ((a) overlap, (b) cascade, (c) separate), which will increase the complexity of PUF model. By configuring the PUF, the mathematical model for the PUF will be altered. This makes it infeasible for attackers to break the PUF by only using one single uniform linear model. The delay of MUXs connected after the feed-forward structure (normally just an arbiter) may also affect the delay difference of the two paths. However, this time difference could add into the

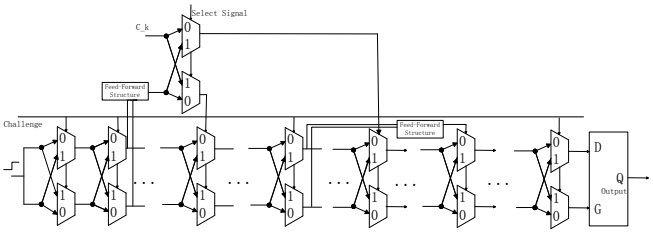

Figure 10: Proposed Highly Secure Reconfigurable Feed-Forward MUX PUF Structure.

total path delay difference both positively and negatively, depending on the select signal. Therefore, the effect of these MUXs would be statistically equivalent to the two paths of original MUX PUF, even if the delay of the added two MUXs vary quite significantly. From above, we conclude that the MUX based PUF will be more secure when feed-forward arbiters are reconfigurable.

4.2.2 MUX and DeMUX PUF

The function of MUX is multiplexing; it selects one of many input signals and forwards the selected signal into a single line. The DeMUX is a device that takes a single input signal that carries many channels and distributes them over multiple output signals. Using DeMUX enables us to select the direction of the propagating signal, and makes the PUF reconfigurable. A basic reconfigurable structure is shown in Figure 11. Instead of propagating the rising edge signal successively, we can choose to skip some stages by adding DeMUX components, which could make the challenge-response behavior reconfigurable and hard to predict. This structure will be harder for attackers to model than the silicon PUF only based on MUX.

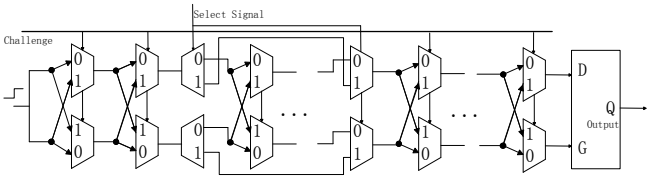

Figure 11: MUX and DeMUX PUF Structure.

5. EXPERIMENTAL RESULTS

All of our experiments have been carried out using SPICE simulations on a 65-nm technology process. We use Monte Carlo method to simulate the effect of process variations and environmental variations. In our simulation, we set up the transistor parameters and process variations based on a major industrial standard model. Each proposed structure has been simulated over at least 20 Monte Carlo runs in SPICE. We simulated 100 MUXs stages for each structure of these silicon PUFs. Accordingly, we need to apply a 100-bit challenge to the PUF to produce a 1-bit response; as a result, 100 different challenges were required to generate the final 100-bit digital signature for each IC.

Simulated PUF Structures: In our experiment, we added 10 feed-forward arbiters into each feed-forward structure of MUX PUF. For instance, the feed-forward arbiters were from stage 1 to stage 11, from stage 11 to stage 21 ... from stage 91 to stage 100 in a feed-forward cascade structure. The feed-forward arbiters were from stage 1 to stage 7, from stage 11 to stage 17 ... from stage 91 to stage 97 in feed-forward separate structure. In a feed-forward overlap structure, the feed-forward arbiters were from stage 1 to stage 51, from stage 6 to stage 56 ... from stage 46 to stage 96. For the reconfigurable feed-forward structure, we also added 10 such arbiters and MUXs structures (as discussed in Section 4.2.1) into the original PUF circuit, which can switch among the 3 different feed-forward structures. Moreover, we also simulated 10 DeMUX components in the MUX and DeMUX PUF. The inputs and the outputs of the DeMUXs were from stage 3 to stage 8, from stage 13 to stage 18 ... and from stage 93 to stage 98. Finally, we simulated an output recombination structure with 20 parallel MUX PUFs. We derived the digital signature by comparing adjacent paths among the total 40 paths. Therefore, except for the first and the last paths, each path was compared to two other paths.

Inter-chip Variations: The inter-chip variations were evaluated by the Hamming distance between two digital signatures which were generated by a same challenge and *configure data* from different chips. Since we simulated 20 chip instances, we had 20*19/2, i.e., 190 possible digital signature comparisons. We provide the maximum and the minimum of these numbers for the inter-chip variations.

Intra-chip Variations: The intra-chip variations were determined by comparing the digital signatures of the same IC under different environmental conditions; in our case, we consider the temperature as the primary environmental factor. It has been shown in our experiment that the intra-chip variations introduced by different temperatures from 0^oC to 100^oC were more significant then the intra-chip variations caused by voltage varying from $1V$ to $1.2V$. The digital signatures of the PUF at 0^oC, 20^oC, 40^oC, 80^oC, 100^oC were obtained; however, we only present the comparisons between 0^oC and 100^oC, as those exhibited the largest variations. We applied 10 different challenges for each IC, and simulated 20 different IC instances. Therefore we had 200 comparisons in total. We provide the maximum and the average of these values of Hamming distance for the intra-chip variations.

Reconfigurability: The reconfigurability was determined by the variations of digital signatures generated by different *configure data* in a same IC. We fixed the challenge for a reconfigurability test, while we fixed the *configure data* of the different structures when examining the inter-chip variations and the intra-chip variations. In fact, the challenge-bit lengths were decreased by adding reconfigurable components in the feed-forward structures; therefore, we need to adjust the challenge bits when simulating these reconfigurable structures. We also applied 10 different *configure data* for each IC, and simulated total 20 different IC instances, which are similar to intra-chip variation test. All the simulations were done under the environmental condition of 25^oC and $1.1V$.

Table 1 presents the inter-chip variations and intra-chip variations for different MUX Physical Unclonable Function structures. First, it can be observed that the minimum inter-chip variations are larger than the maximum intra-chip variations for all of the simulated structures. Thus, we can conclude that the variations caused by the randomness in manufacturing process are more significant than the variations under different environmental conditions. Therefore, these PUFs can be used as reliable secret keys with some error correcting techniques. Second, it can also be observed that by adding feed-forward arbiters into the MUX PUF circuit, the inter-chip variations and intra-chip variations are both increased, since the noise can have influence on the select signals of some intermediate stages. By comparing the inter-chip variations and the intra-chip variations, we can say the feed-forward separate structure is the most reliable structure while the feed-forward cascade is the least reliable one among the 3 feed-forward structures. The reconfigurable feed-forward structure has very close performance to the 3 types of feed-forward structures, since its functionality is switching among the 3. Moreover, the reconfigurable MUX and DeMUX PUF has similar inter-chip variation as the non-feed-forward structure, but the intra-chip variation is increased, as the number of stages may be reduced by configurations. Therefore, the reliability of this structure is decreased.

Table 1: Simulation Results: Variations

Structures	Inter-chip Variation		Intra-chip Variation	
	Max	Min	Max	Avg
Non-feed-forward	59	22	13	5.8
Feed-forward Overlap	66	27	15	8.7
Feed-forward Cascade	64	25	20	10.7
Feed-forward Separate	65	26	17	9.9
Reconfigurable Feed-forward	65	25	19	10.3
MUX and DeMUX	57	23	16	7.1

Table 2 shows the reconfigurability of each reconfigurable structure. It can be seen that the output recombination structure has the best reconfigurability, i.e., by fixing the challenge bits and only changing the *configure data*, the digital signature of this structure has the most significant variations. In our simulation results, the average variation is 38.7 bits. The MUX and DeMUX PUF exhibits the least reconfigurability. This is because the function of the DeMUX is only to determine whether to skip some stages or not. When the process variations of other stages are relatively large, the difference of digital signatures with two different *configure data* may only vary a little bit. For the output recombination structure, it is similar to comparing different paths with different configurations. Therefore, its performance is close to the inter-chip variation of the non-feed-forward MUX PUF. It also can be observed that although the challenge hash structure and the challenge LFSR structure are both pre-processing the challenge before applying to the circuit, their reconfigurability still has some difference. As the challenge LFSR appears to have better reconfigurability, we can conclude that the number generated by the LFSR in our case may have better randomness than that of the hash function. Finally, the proposed reconfigurable feed-forward MUX PUF has the average Hamming distance 32.4 bits by different configurations, which will be sufficient to be used as a secure and reliable secret key storage method, considering its complex and nonlinear functionality.

Table 2: Simulation Results: Reconfigurability

Structures	Variation		
	Max	Avg	Min
Challenge LFSR	44	34.6	28
Challenge Hash	42	28.3	19
Output Recombination	57	38.9	25
Reconfigurable Feed-forward	47	32.4	22
MUX and DeMUX	33	24.7	13

Overall, all the proposed reconfigurable structures have considerable reconfigurability, and can be used for reliable authentication and identification within certain error tolerance, as the minimum of the inter-chip variations is larger than the maximum of intra-chip variations. The output recombination structure has the best reconfigurability; however, the reconfigurable feed-forward MUX PUF has the best performance due to its security, as it is extremely hard to be modeled by linear modeling methods.

6. CONCLUSION

We have presented several reconfigurable silicon MUX Physical Unclonable Functions based on two major approaches and demonstrated their effectiveness by experimental results via inter-chip variation and intra-chip variation. We also have discussed the reliability perspective of PUFs and proposed several methods to increase the security. Ongoing work includes novel highly secure and reliable reconfigurable PUF designs and their mathematical analysis. Furthermore, we are also interested in developing an authentication scheme for reconfigurable PUFs, which will use several pairs of CRPs as a set for authentication by utilizing the reconfigurable property of reconfigurable PUFs.

7. REFERENCES

[1] B. Gassend, D. Clarke, M. V. Dijk, and S. Devadas, "Silicon physical unclonable functions," *the 9th ACM Conference on Computer and Communications Security*, p. 160, 2002.

[2] R. Pappu, B. Recht, J. Taylor, and N. Gershenfeld, "Physical one-way functions." *Science*, vol. 297(5589), p. 2026, 2002.

[3] B. Gassend, D. Clarke, M. V. Dijk, and S. Devadas, "Controlled physical unclonable functions," in *Computer Security Application Conference*, 2002, pp. 149–160.

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

[5] R. Maes, P. Tuyls, and I. Verbauwhede, "Intrinsic PUFs from flip-flops on reconfigurable devices," in *Benelux Workshop Information and System Security (WISSec 08)*, 2008.

[6] D. E. Holcomb, W. P. Burleson, and K. Fue, "Initial SRAM state as a fingerprint and source of true random numbers," in *Conference on RFID Security*, 2007.

[7] U. Ruhrmair, F. Sehnke, J. Solter, G. Dror, S. Devadas, and J. Schmidhuber, "Modeling attacks on physical unclonable functions," in *Conference on RFID Security*, 2010.

[8] M. Majzoobi, F. Koushanfar, and M. Potkonjak, "Techniques for design and implementation of secure reconfigurable PUFs," *ACM Transactions on Reconfigurable Technology and Systems*, vol. 2, no. 1, pp. 1–33, 2009.

[9] D. Lim, J. W. Lee, B. Gassend, G. E. Suh, M. V. Dijk, and S. Devadas, "Extracting secret keys from integrated circuits," *IEEE Transaction on Very Large Scale Integration Systems*, vol. 13, no. 10, p. 1200, 2005.

[10] H. Chang and S. Sapatnekar, "Statistical timing analysis considering spatial correlation in a pert-like traversal," in *IEEE International Conference Computer-Aided Design Integrated Circuits and Systems*, 2003, pp. 621–625.

[11] J.-W. Lee, D. Lim, B. Gassend, G. E. Suh, M. van Dijk, and S. Devadas, "A technique to build a secret key in integrated circuits with identification and authentication applications," in *IEEE International Conference Computer-Aided Design Integrated Circuits and Systems*, 2003, pp. 621–625.

[12] K. Kursawe, A. Sadeghi, D. S. B. Skoric, and P. Tuyls, "Reconfigurable physical unclonable functions – enabling technology for tamper-resistant storage," in *2nd IEEE International Workshop on Hardware-Oriented Security and Trust(HOST)*, 2009, pp. 22–29.

[13] A. M. S. Morozov and P. Schaumont, "An analysis of delay based PUF implementations on FPGA," *Springer*, pp. 382–387, 2010.

[14] D. Merli, F. Stumpf, and C. Eckert, "Improving the quality of ring oscillator PUFs on FPGAs," in *WESS '10 Proceedings of the 5th Workshop on Embedded Systems Security*, 2010.

[15] J. Guajardo, S. S. Kumar, G.-J. Schrijen, and P. Tuyls, "FPGA intrinsic PUFs and their use for IP protection," *Cryptographic Hardware and Embedded Systems*, 2007.

[16] J. Cong, "Challenges and opportunities for design innovations in nanometer technologies," *SRC Design Science Concept Paper*, 1997.

[17] S. Nassif, "Delay variability: Sources, impact and trends," in *Solid-State Circuits Conference*, 2000, pp. 368–369.

[18] L. Alaus, D. Noguet, and J. Palicot, "A reconfigurable linear feedback shift register operator for software defined radio terminal," *IEEE International Symposium on Wireless Pervasive Computing*, 2008.

[19] P. Kitsos, N. Sklavos, N. Zervas, and O. Koufopavlou, "A reconfigurable linear feedback shift register (LFSR) for the bluetooth system," in *IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, 2001.

[20] M. Zeghida, B. Bouallegue, A. Baganne, and M. Machhout, "A reconfigurable implementation of the new secure hash algorithm," *Second International Conference on Availability, Reliability and Security (ARES)*, pp. 281–285, 2007.

[21] B. Gassend, D. Clarke, M. V. Dijk, and S. Devadas, "Silicon physical random functions," in *ACM Conference on Computer and Communications Security*, 2002, pp. 148–160.