B. TECH. PROJECT REPORT

On

Low Power and Low Area PUF Design for Portable and Health Monitoring Applications

BY

L SUDHARANI

DISCIPLINE OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY INDORE

December 2019

Low Power and Low Area PUF Design for Portable and Health Monitoring Applications

A PROJECT REPORT

Submitted in partial fulfillment of the requirements for the award of the degrees

of
BACHELOR OF TECHNOLOGY
in
ELECTRICAL ENGINEERING

Submitted by:
L SUDHARANI

Guided by:
Dr. Santosh Kumar Vishvakarma,
Associate Professor,
Electrical Engineering,
Indian Institute of Technology Indore

INDIAN INSTITUTE OF TECHNOLOGY INDORE
December 2019

CANDIDATE'S DECLARATION

I hereby declare that the project entitled **"Low Power and Low Area PUF Design for Portable and Health Monitoring Applications"** submitted in partial fulfillment for the award of the degree of Bachelor of Technology in **Electrical Engineering** completed under the supervision of **Dr. Santosh Kumar Vishvakarma, Associate Professor, Electrical Engineering, IIT Indore** is an authentic work.

Further, I declare that I have not submitted this work for the award of any other degree elsewhere.

L SUDHARANI
160002027
Discipline of Electrical Engineering
Indian Institute of Technology Indore

CERTIFICATE by BTP Guide

It is certified that the above statement made by the student is correct to the best of my knowledge and belief.

Dr. Santosh Kumar Vishvakarma,
Associate Professor,
Discipline of Electrical Engineering,
Indian Institute of Technology Indore

Preface

This report on "**Low Power and Low Area PUF Design for Portable and Health Monitoring Applications**" is prepared under the guidance of Dr. Santosh Kumar Vishvakarma, Associate Professor, Electrical Engineering, IIT Indore

Throughout this report, detailed description of the technologies that have been used to design and implement the low power and low area D flip-flop PUF (Physically Unclonable Function) module is provided. The implemented low power and low area D flip-flop module is tested for its different inputs and results are presented in a clear and concise manner. I have tried to the best of my ability and knowledge to explain the content in a lucid manner. I have also added figures to make it more illustrative.

Acknowledgements

I would like to thank my B.Tech Project supervisor **Dr. Santosh Kumar Vishvakarma** for his constant support in structuring the project and for his valuable feedback throughout the course of this project. He gave me an opportunity to discover and work in such an interesting domain. His guidance proved really valuable in all the difficulties I faced in the course of this project.

I am really grateful to **Mr. Sajid Khan** who also provided valuable guidance and helped with the problems while working on various technologies. He provided initial pathway for starting the project in the right manner and provided useful directions to proceed along whenever necessary.

I am also thankful to all my family members, friends and colleagues who were a constant source of motivation. I am really grateful to Dept. of Electrical Engineering, IIT Indore for providing with the necessary hardware utilities to complete the project. I offer sincere thanks to everyone who else knowingly or unknowingly helped me to complete this project.

LSUDHARANI

160002027

Discipline of Electrical Engineering

Indian Institute of Technology Indore

Abstract

A Physically Unclonable Function, or PUF, is a physically-defined "digital fingerprint" that serves as a unique identity for a semiconductor device such as a microprocessor. They are based on unique physical variations which occur naturally during semiconductor manufacturing. Because of these variations, PUFs are unique and effectively unclonable. A PUF is a physical entity embodied in a physical structure. Today, PUFs are usually implemented in integrated circuits and are typically used in applications with high security requirements, more specifically cryptography.

Over 40 types of PUF have been suggested till date. Some of the variants include delay-based Arbiter PUF, RO PUF and memory-based SRAM PUF, Latch PUF, Buskeeper PUF, Butterfly PUF, and D flip-flop PUF. In delay PUFs, random variations in delays of wires and gates on silicon are exploited by which output bit is generated. In memory-based PUF implementations, a bi-stable structure of two cross-coupled inverters is used to generate the output bit. In memory-based PUF, PUF circuits are the part of the design itself; hence, the effective hardware cost is zero. This makes them the most preferred architectures in applications where the silicon area is a prime concern.

Several memory-based PUF architectures such as butterfly PUF, latch PUF, and buskeeper PUF require dedicated latches that cannot be shared or reused by the system and thus results in additional hardware overhead which is not the case in SRAM PUF and D flip-flop PUF. But an SRAM PUF does not provide superior security advantage against invasive attacks than a D flip-flop PUF. Thus, a D flip-flop design that can also be used as a PUF is proposed.

Table of Contents

List of Figures

List of Tables

Chapter 1

Overview

1.1 Background

The increasing large number of deployed nodes for monitoring environment, people's health, goods, and shared resources, makes security an underlying challenge in IoT. Presently, severe security challenges are associated with data authenticity, integrity, and confidentiality. Thus, in this deployment, the use of cryptography is the solution to make the transmitted data unreadable from the unintended receiver. Traditionally, the cryptographic algorithms require a nonvolatile key for encrypting and decrypting of the confidential data. Storing the key into the ROM or flash memory, makes the key prone to the invasive attacks by de-capsulating the chip and studying the circuit to observe the stored key. To avoid this scenario and thus to provide security and privacy, physically unclonable function (PUF) has emerged as a required solution. The PUF utilizes the manufacturing and process-induced variations, to generate the key. Basically, the key is not stored but rather extracted on demand, therefore it provides a high level of security against any sort of the invasive attacks. It is evident from the literature that the process and manufacturing variations provide each device an unique and random innate properties such as threshold-voltage (V_{th}), channel length (L_{eff}), which can be used to generate stable, repeatable and device-dependent signature [1]. These signatures are used in applications like, authentication [2], IP protection [3], key generation [4], software binding [5], and many more. Additional important aspect of this approach is, as the generated responses are based on random manufacturing and process-induced variations, therefore, for both, the designer and manufacturer, it is nearly impossible to clone any implemented PUF.

1.2 Motivation

Security challenges in IoT push sensitive information about people's health, environment, goods and shared resources at stake and make them vulnerable to invasive attacks. Data authenticity, integrity and confidentiality are severe security challenges that need to be addressed. Cryptography proves to be a solution which makes the transmitted data

unreadable by unintended receiver. The cryptographic algorithms need a nonvolatile key to encrypt and decrypt confidential data. This key needs to be more secured from attackers who try to extract the key from the architecture. PUFs prove to be a preferable choice in this aspect as they utilize the manufacturing and process-induced variations to generate the key and the key is not stored rather extracted on demand. Therefore, for both designer and manufacturer, it is nearly impossible to clone any implemented PUF. Intrinsic PUFs (SRAM PUFs, D flip-flop PUFs), which are readily available in almost all of the designs, being an integral part of the design, they can be used with compromised limited uniqueness. To increase the uniqueness of D flip-flop PUF, a symmetric tri-state inverter D flip-flop based lightweight PUF is proposed.

Chapter 2

Introduction

2.1 Physically Unclonable Function

A physically unclonable function, or PUF, is a physically-defined "digital fingerprint" that serves as a unique identity for a semiconductor device such as a microprocessor. They are based on unique physical variations which occur naturally during semiconductor manufacturing. Because of these variations, PUFs are unique and effectively unclonable. A PUF is a physical entity embodied in a physical structure. Today, PUFs are usually implemented in integrated circuits and are typically used in applications with high security requirements, more specifically cryptography.

The idea of using intrinsic random physical features to identify objects, systems and people is not new. Fingerprint identification of humans dates at least back to the nineteenth century [6] and led to the field of biometrics. In the eighties and nineties of the twentieth century, random patterns in paper and optical tokens were used for unique identification of currency notes and strategic arms. A formalization of this concept was introduced in the very beginning of the twenty first century, first as physical one-way functions [reference], physical random functions and finally as physical(ly) unclonable functions or PUFs. In the years following this introduction, an increasing number of new types of PUFs were proposed, with a tendency towards more integrated constructions. The practical relevance of PUFs for security applications was recognized from the start, with a special focus on the promising properties of physical unclonability and tamper evidence.

Over the last couple of years, the interest in PUFs has risen substantially, making them a hot topic in the field of hardware security and leading to an expansion of published results.

2.2 Advantages of PUFs

Here are some advantages of physically unclonable functions:

- Physically unclonable functions act as Silicon Biometric.
- They generate keys from a complex physical system.

- They generate responses based on manufacturing and process-induced variations. So the responses are repeatable but unpredictable.
- Keys are not stored but rather extracted on demand. Therefore, PUFs provide high level of security against any sort of invasive attacks.
- No need to program the secret.
- They can generate multiple keys.

2.3 Delay-based Intrinsic PUFs

A delay PUF exploits the random variations in delays of wires and gates on silicon. Given an input challenge, a race condition is set up in the circuit, and two transitions that propagate along different paths are compared to see which comes first. This random delay difference between the two paths determines the output bit. Many circuits realizations are possible and at least two have been fabricated. When a circuit with the same layout mask is fabricated on different chips, the logic function implemented by the circuit is different for each chip due to the random variations of delays.

Arbiter PUF [7] and RO PUF [8] are the two most popular approaches for the delay based PUFs.

2.3.1 Arbiter PUF

The basic idea is to introduce a digital race condition on two paths on a chip and to have a so-called arbiter circuit decide which of the two paths won the race. If the two paths are designed symmetrically, i.e. with the same intended delay, then the outcome of the race is not fixed beforehand. During production of the chip, manufacturing variations will have an effect on the physical parameters determining the exact delay of each path, and causing a small random offset between the two delays. This leads to a random and possibly device-specific outcome of the arbiter and hence explains the PUF behavior of such a construction.

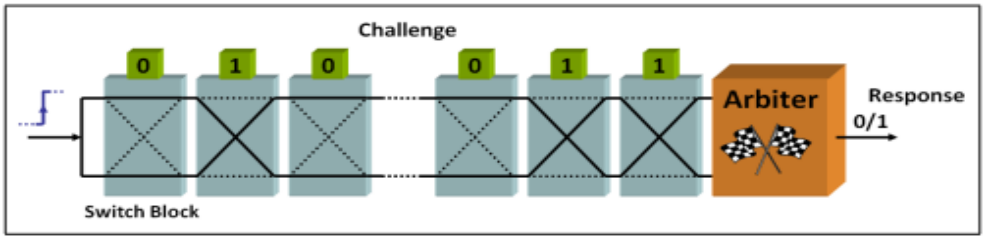

Figure 1: Basic Operation of an Arbiter PUF

2.3.2 RO PUF

Ring oscillator PUFs use a different approach towards measuring small random delay deviations caused by manufacturing variability. The output of a digital delay line is inverted and fed back to its input, creating an asynchronously oscillating loop, also called a ring oscillator. It is evident that the frequency of this oscillator is precisely determined by the exact delay of the delay line. Measuring the frequency is hence equivalent to measuring the delay, and due to random manufacturing variations on the delay, the exact frequency will also be partially random and device dependent. Frequency measurements can be done relatively easy using digital components: an edge detector detects rising edges in the periodical oscillation and a digital counter counts the number of edges over a period of time. The counter value contains all the details of the desired measure and is considered the PUF response.

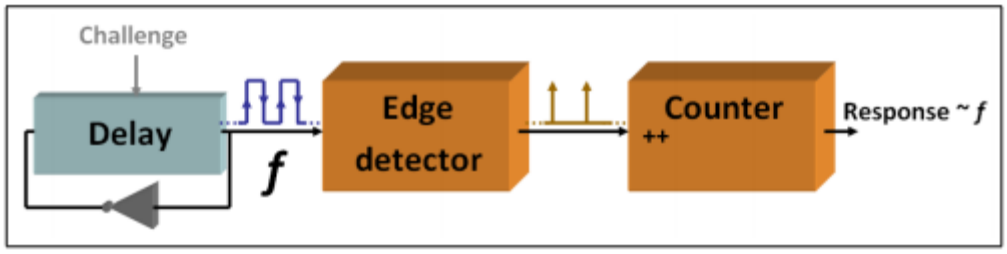

Figure 2: Basic Operation of a ring oscillator PUF

2.4 Memory-based Intrinsic PUFs

In the memory-based PUF implementations, a bi-stable structure of two cross-coupled inverters is used to generate the output bit. The asymmetry due to random variation causes the cross-coupled inverters to resolve to a proffered state at power-up. In the memory-based PUF, PUF circuits are the part of the design itself; hence, the effective hardware cost is zero. This makes memory-based PUF the best candidate for lightweight and resource-constrained applications, where the silicon area is a prime concern.

Presently, various memory-based PUF variants are in use, such as SRAM based PUF [9], latch PUF [10], buskeeper PUF [11], butterfly PUF [12], and D flip-flop PUF [13].

2.4.1 SRAM PUFs

SRAM or static random-access memory is a type of digital memory consisting of cells each capable of storing one binary digit. An SRAM cell is logically constructed as two cross-coupled inverters, hence leading to two stable states. In regular CMOS technology, this circuit is implemented with 4 MOSFETs, and an additional 2 MOSFETs are used for read/write access. For performance reasons, the physical mismatch between the two symmetrical halves of the circuit (each implementing one inverter) is kept as small as possible. It is not clear from the logical description of the cell at what state it will be right after power-up of the memory, i.e. what happens when the supply voltage comes up? It is observed that some cells preferably power-up storing a zero, others preferably power-up storing a one, and some cells have no real preference, but the distribution of these three types of cells over the complete memory is random. As it turns out, the random physical mismatch in the cell, caused by manufacturing variability, determines the power-up behavior.

2.4.2 Butterfly PUFs

In [5], SRAM PUFs were tested on FPGAs. However, it turns out that in general this is not possible, since on the most common FPGAs, all SRAM cells are hard-reseted to zero directly after power-up and hence all randomness is lost. Another inconvenience of SRAM PUFs is that a device power-up is required to enable the response generation, which might not always be possible. To counter these two drawbacks, butterfly PUFs were introduced in [12]. The behavior of an SRAM cell is mimicked in the FPGA reconfigurable logic by cross-coupling two transparent data latches. Again, such a circuit allows two logically stable states. However, using the clear/preset functionality of the latches, an unstable state can be introduced after which the circuit converges back to one of the two stable states. This is comparable to the convergence for SRAM cells after power-up, but without the need for an actual device power-up. Again, the preferred stabilizing state of such a butterfly PUF cell is determined by the physical mismatch between the latches and the cross-coupling interconnect.

2.4.3 Flip-Flop PUFs

Flip-flop PUFs [13] are based on the power-up characteristic of (uninitialized) D flip-flops. Due to uncontrolled process variations, each flip-flop will have the tendency to switch its output to either the zero state or the one state when the IC is powered up. The main security advantage of a flip-flop PUF, compared to an SRAM PUF, is the fact that flip-flops are easily spread over an IC and hence are very difficult to locate by an attacker trying to reverse-engineer the chip and to probe each individual start-up bit.

Chapter 3

Design of Proposed D Flip-Flop

3.1 Selection of PUF Architecture

The use of SRAM PUF is limited in FPGA based implementation as in most of the FPGAs, the SRAM memories are initialized to a known state upon power-up. To solve this issue in FPGA, the butterfly PUF [12] replaces the SRAM cross-coupled inverters by cross-coupled latches on FPGA. The cross-coupled arrangement of latches can be forcibly brought to an unstable condition by resetting one of the latches and presetting the other one. After removing the preset and reset, the butterfly cell settles back to one of the two stable states. It should be noted that the settling state depends upon the manufacturing mismatch between the two latches. However, the butterfly cell needs extra attention in placement and routing, because the preferred stable state is also a function of mismatch present in signal routing. Latch PUF [10] is also a cross-coupled arrangement of NOR gates, which can be brought to an invalid state by forcing `reset0 to logic `10. When `reset0 is forced to logic `00, the latch settles back to the valid state. Similar to the SRAM PUF, D flip-flop PUF [13] uses the power-on values of D flip-flops, D flip-flops are also initialized to a known state upon power-on and suffers from poor uniqueness. Also, in FPGA and ASIC both, after power-up, SRAM and D flip-flop PUF can be used only once until a write operation has not occurred. Once a write operation has been performed, the PUF output is overwritten. Thus, to regenerate the previous response, the cell needs a reboot. Hence if PUF responses are required very often, the part of SRAM or D flip-flop used in PUF, cannot be shared.

Among all of the considered architectures, butterfly PUF, latch PUF, and buskeeper PUF require dedicated latches that cannot be shared or reused by the system and thus results in additional hardware overhead.

To address the hardware overhead issue, use of D flip-flop PUF can be preferred choice, as they offer a higher level of security advantage against invasive attacks when compared with the SRAM PUF. Additionally, it is also possible that the D flip-flops can be spread randomly across a design to make it much harder for an attacker to locate them and their signal lines since D flip-flop is one of the major components in finite-state-machine(FSM).

To keep this advantage, in this paper, we have proposed a D flip-flop design that can also be used as a PUF.

3.2 Various D Flip-Flop Architectures

The state-of-the-art D flip-flop architectures including the conventional D flip-flop is shown in Figure 3.

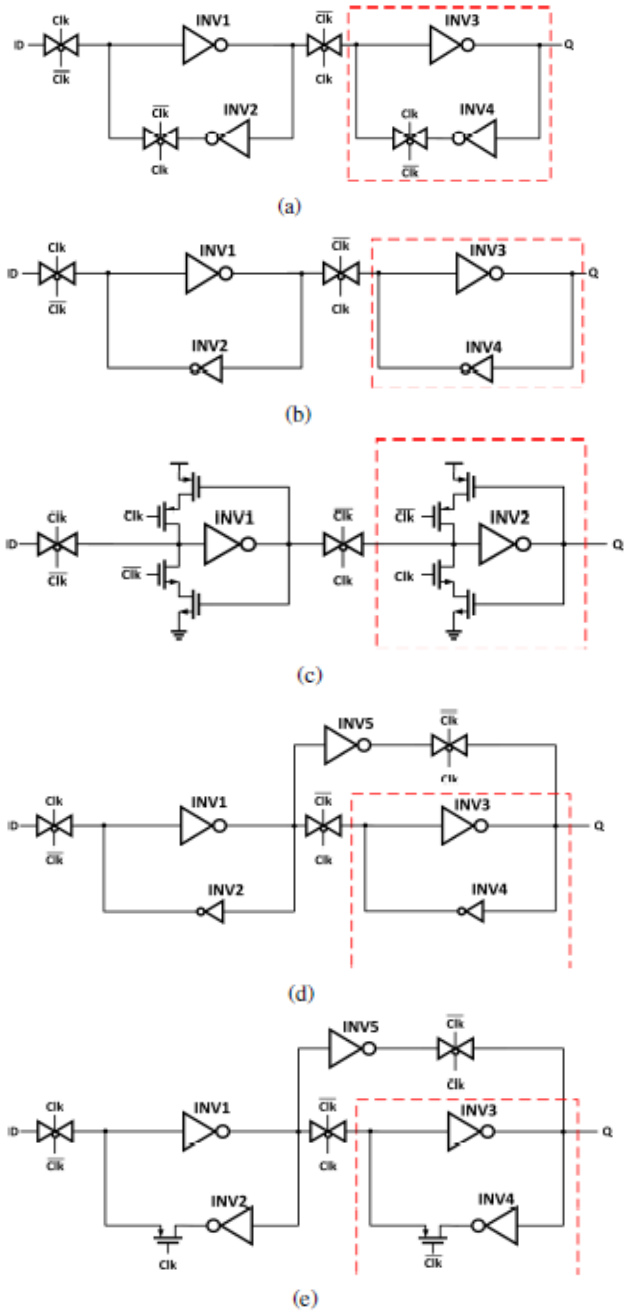

Figure 3: D flip-flop architectures: (a) Conventional [14], (b) Low area [15], (c) Low power [16], (d)Push-pull [15], (e) Push-pull isolation [15].

3.3 Proposed D Flip-Flop with Integrated PUF

PUF architecture with symmetric cross-coupled inverter shows a high value of uniqueness. Thus, the proposed D flip-flop has 2 tri-state inverters in the second latch in which TRI-STATE INV2 makes the proposed architecture symmetric. To regenerate the PUF responses even after a (p)reset or write operation, M8 is added to bring TRI-STATE INV 2 and TRI-STATE INV 3 back in an unstable state.

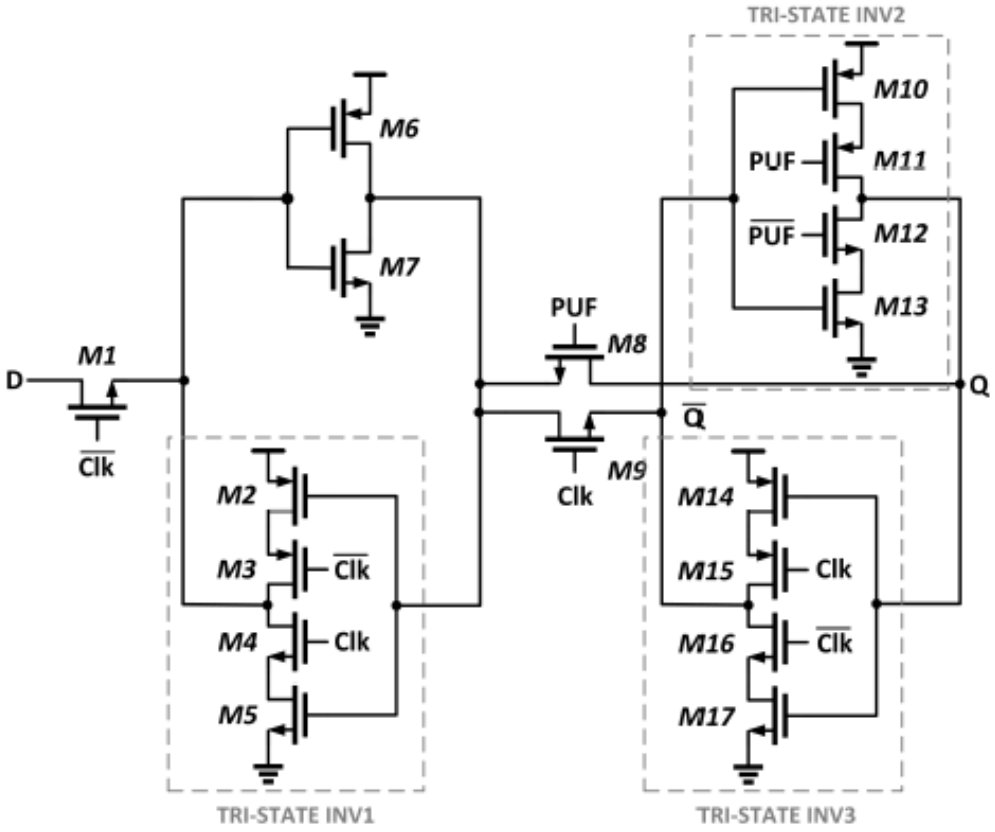

Figure 4: Schematic of the proposed D Flip-Flop

When both Clk and PUF are set to '0', the equivalent circuit of the proposed D flip-flop is shown in figure below, where M11; M12; M15 and M16 are replaced by their ON resistance.

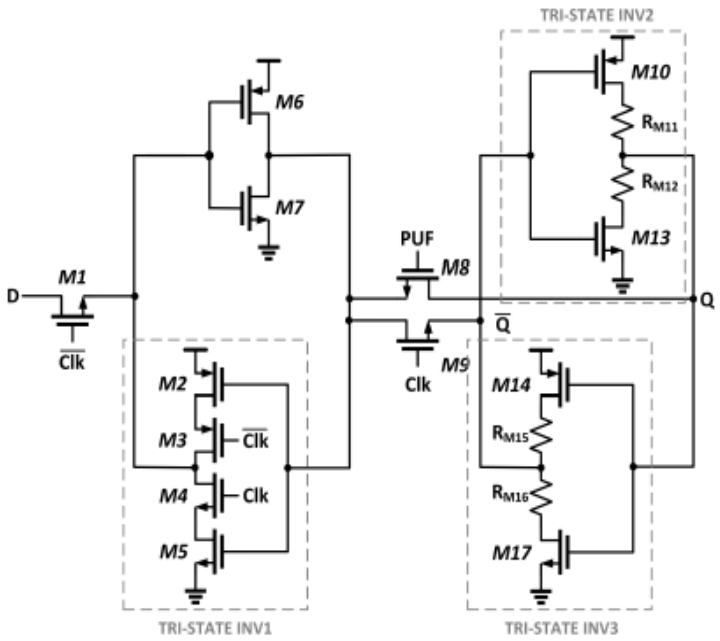

Figure 5: Equivalent circuit when PUF and Clk = '0'

Input			Output		Operation
Clk	D	PUF	Q_{N+1}	$\overline{Q_{N+1}}$	
1	X	0	Q_N	$\overline{Q_N}$	Flip-flop
0	X	0	Q_N	$\overline{Q_N}$	Flip-flop
↑	1	0	1	0	Flip-flop
↑	0	0	0	1	Flip-flop
↓	X	0	Q_N	$\overline{Q_N}$	Flip-flop
1	1	1	1	1	PUF
1	0	1	0	0	PUF
0	X	0	PUF_OUT	$\overline{PUF\_OUT}$	PUF

Table 1: Truth Table of the proposed D Flip-Flop

Chapter 4

Results and Discussion

4.1 ASIC Implementation

ASIC stands for Application Specific Integrated Circuit. As the name implies, ASICs are application specific. They are designed for one sole purpose and they function the same their whole operating life. For example, the CPU inside your phone is an ASIC. It meant to function as a CPU for its whole life. Its logic function cannot be changed to anything else because its digital circuitry is made up of permanently connected gates and flip-flops in silicon. The logic function of ASIC is specified in a similar way as in the case of FPGAs, using hardware description languages such as Verilog or VHDL. The difference in case of ASIC is the resultant circuit is permanently drawn into silicon whereas in FPGAs the circuit is made by connecting a number of configurable blocks. For a comparison, think of creating a castle using Lego blocks versus creating a castle using concrete. The former is analogous to FPGAs, whereas the latter is analogous to ASICs. You can reuse Lego blocks to create a different design, but the concrete castle is permanent.

The proposed D flip-flop along with the conventional, low area based, low power based, push-pull based, push-pull isolation based D flip-flop architectures have been implemented on 40nm industry-standard foundry. Fig. 6 shows the layout of the conventional, low power, and proposed D flip-flop, where it can be seen that compared to the conventional and low power flip-flop, the proposed flop-flop has 1.171_ and 1.323_ larger width respectively and 0.848_and 0.904_ smaller length, respectively. The conventional, low power and proposed architectures have an area of 3.728_m2, 3.099 _m2 and 3.708 _m2, respectively. Although the proposed D flip-flop has more number of transistors than the conventional architecture, after using the drain sharing technique, the overall area of the proposed architecture is the same as the conventional D flip-flop architecture.

All the simulations were performed using Cadence Virtuoso at VDD = 1:1 V and considering operation temperature 27 _C, unless specified. Since the primary function of the proposed circuit is to be used as a flip-flop, hence, apart from PUF all the flip-flop parameters are also extracted.

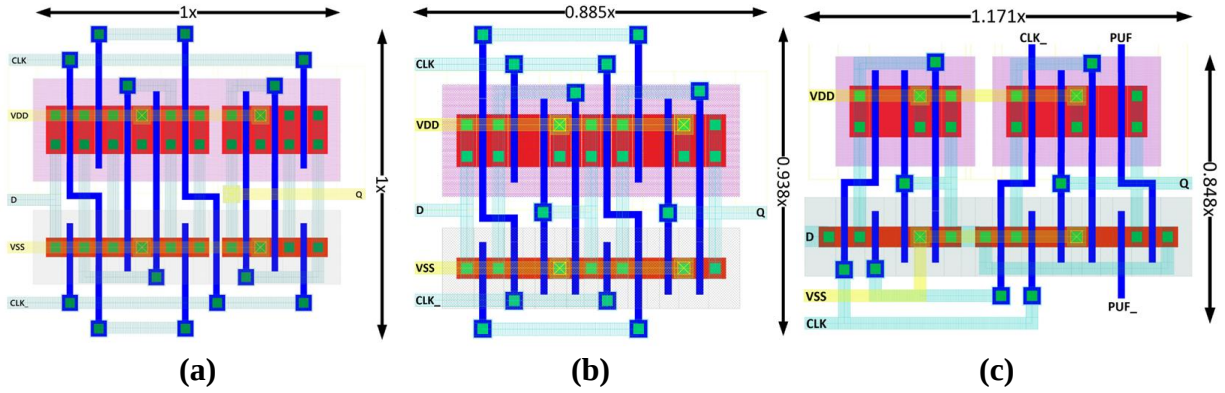

Figure 6: Layout of: (a) Conventional D Flip-Flop (b) Low Power D Flip-Flop (c) Proposed D Flip-Flop

4.1.1 Performance of Proposed D flip-flop

Performance as a Flip-Flop

Power and delay are the important parameters for the analysis of a flip-flop. The flip-flop performance of the proposed architecture along with existing architectures is analyzed by varying supply voltages and observing the C-to-Q delay, dynamic power, and leakage power.

Delay:

The proposed architecture has delay comparable to the conventional and low power D flip-flop as shown in fig. 7. The negligible delay difference is due to the use of an additional tristate inverter in the second latch. The proposed D flip-flop has 1.013x and 1.005x more delay at the supply voltage of 1.1V. At the supply voltage of 0.6V, the proposed D flip-flop has 1.077x and 1.038x more delay as compared to the conventional and low power D flip-flop, respectively. Results show that the delay difference reduces with the increasing supply voltage.

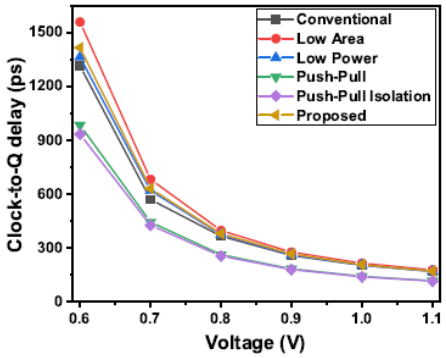

Figure 7: Clk-to-Q delay of various D flip-flop architectures at different supply voltages.

Dynamic Power:

The proposed architecture consumes less dynamic power as compared to push-pull, low area, push-pull isolation because the proposed architecture has less number of transistors compared to push-pull isolation and no short circuit power dissipation as in case of low area and push-pull architectures and is higher as compared to the conventional and low power D flip-flop because the proposed architecture has more number of transistors as shown in Fig. 8.

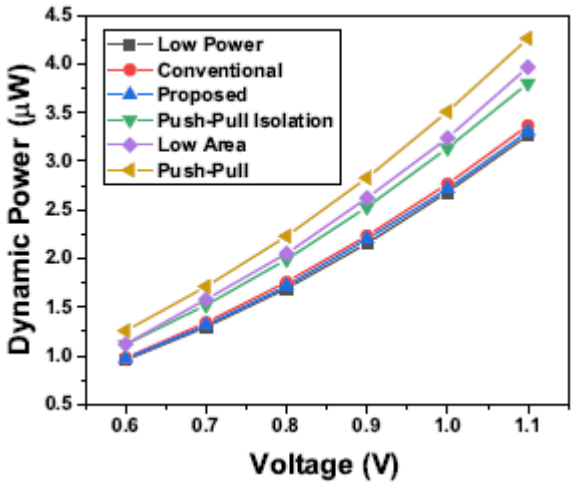

Figure 8: Dynamic Power Consumption

Leakage Power:

The proposed architecture consumes less leakage power compared to push-pull isolation and is higher when power compared to the low area, low power, push-pull and conventional D flip-flop because the proposed architecture has more number of transistors as shown in Fig. 9

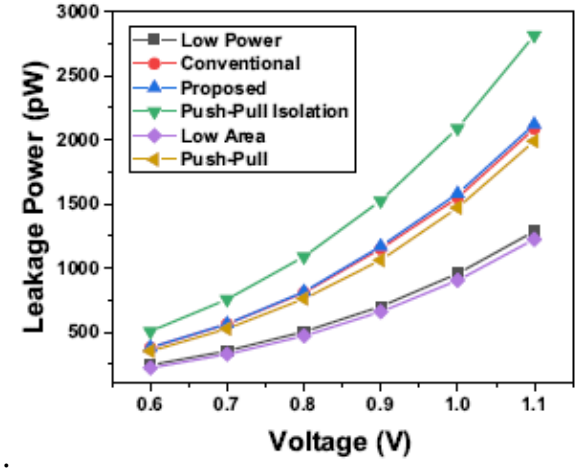

Figure 9: Leakage Power Consumption

4.2 Monte Carlo Simulation

Monte-Carlo simulations are an advanced Computational Physics tool. It is used in a wide variety of areas to predict the functioning, failure of designs and to simulate them for analysis.

Monte Carlo is a mismatch simulation which you can perform in Cadence tool. This is basically used to check if the circuit you have made works across all process, voltage, temperature and mismatch corners. It a must to perform this simulation in order to verify complete working of the circuit

In VLSI circuit design during simulation, we run the design through various PVT (Process, Voltage and Temperature) corners with an aim that the circuit should be able to reliably operate at all extreme conditions. These PVT variations can be generalized as,

1. Temperature from as low as -40^{o} to s high as 125^{o}C,
2. Voltage ±10% variation from its nominal value
3. Process – This is generally two letter convention where first letter is behaviour of NMOS and second letter is of PMOS.TT, SS, FF, SF and FS are corners generally used. Letter T stands for Typical (Nominal V_t), F for Fast (Low V_t) and S fir Slow (High V_t).

We have performed 10k sets of Monte Carlo simulation with _3_ deviation to generate 64-bit challenge-response pairs(CRPs).

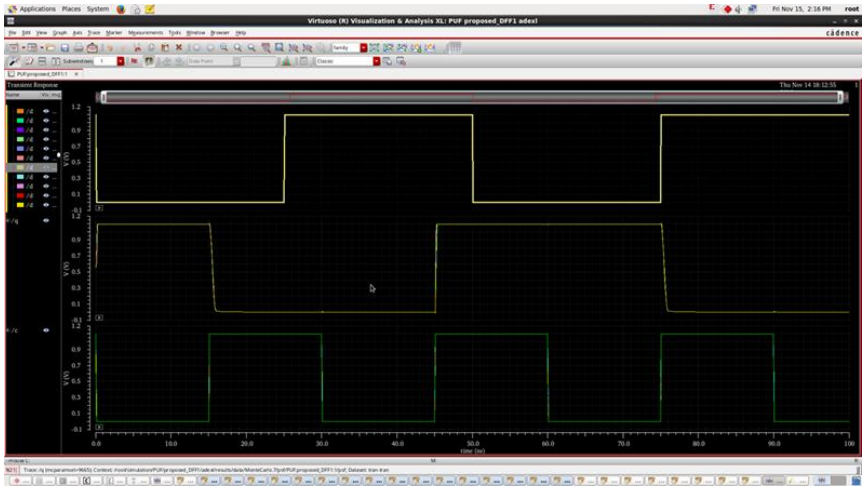

Figure 10: Monte Carlo Simulation.

4.3 FPGA Implementation

FPGA stands for Field Programmable Gate Array. It is an integrated circuit which can be "field" programmed to work as per the intended design. It means it can work as a microprocessor, or as an encryption unit, or graphics card, or even all these three at once. The designs running on FPGAs are generally created using hardware description languages such as VHDL and Verilog. FPGA is made up of thousands of Configurable Logic Blocks (CLBs) embedded in an ocean of programmable interconnects. The CLBs are primarily made of Look-Up Tables (LUTs), Multiplexers and Flip-Flops. They can implement complex logic functions. Apart from CLBs, and routing interconnects, many FPGAs also contain dedicated hard-silicon blocks for various functions such as Block RAM, DSP Blocks, External Memory Controllers, PLLs, Multi-Gigabit Transceivers etc. A recent trend is providing a hard-silicon processor core (such as ARM Cortex A9 in case of Xilinx Zynq) inside the same FPGA die itself so that the processor can take care of mundane, non-critical tasks whereas FPGA can take care of high-speed acceleration which cannot be done using processors. These dedicated hardware blocks are critical in competing with ASICs.

- The functional testing of the proposed tri-state flip-flop based PUF is done on five Xilinx Vertex II XC2VP30 devices using Xilinx ISE 10.1.
- On each FPGA we have instantiated the proposed flip-flop at 1000 times and hence collected 1000-bit response.
- The distribution is symmetric about its center value 32 and has a uniqueness value of 0.4899.

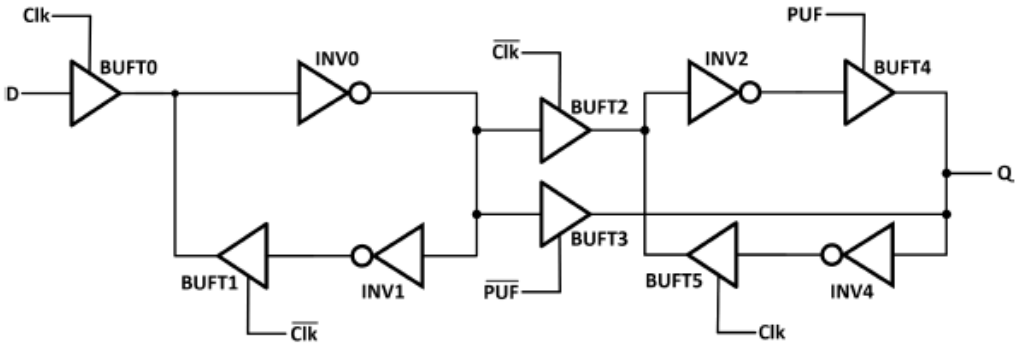

Figure 11: FPGA Implemented architecture of the proposed D flip-flop.

4.3.1 Performance as a PUF

To generate a 64-bit response 64 instances of proposed PUF along with the conventional, low area, low power, push-pull, push-pull isolation are implemented. The performance as a PUF has been analyzed considering Uniqueness, and Reliability. Fig. 14 shows the uniqueness value for the various considered PUF architectures. The result shows that among all the considered architectures, the proposed architecture has the highest uniqueness, which is also very closed to the ideal value 0.5. The higher uniqueness value for the proposed PUF is because of the symmetric feedback and main path. Apart from proposed architecture, the low area and push-pull architectures have almost same and second-highest value. As in both of the architectures, there is no transmission gate or pass transistor present in the path of the cross-coupled inverters. However, the uniqueness value of the low area and push-pull architecture is not good due to presence of mismatch in the width of both inverters.

Uniqueness:

The uniqueness is a standard parameter that differentiates the responses obtained from two PUF instances. It has an **ideal value** of **50%,** which means when the same challenge is applied to any two PUF instances, then half of the PUF response should be different to each other.

Let us consider that corresponds to a challenge C, Rp and Rq are respectively two n-bit responses from randomly selected chip p and q out of m number of available chips. The uniqueness(U) from m chips can be expressed as:

$$U = \frac{2}{m(m-1)} \sum_{p=1}^{m-1} \sum_{q=p+1}^{m} \frac{HD(R_p, R_q)}{n} \times 100\%$$

where, HD (Rp, Rq) is the hamming distance between Rp and Rq

The most important metric for estimating PUF performance is the Hamming distance (HD), which is the number of different bits between two outputs. The intra-HD represents the reproducibility of each chip, and the inter-HD represents the uniqueness between different chips. A sharp distribution at around HD=0 and a sufficient gap between the

intra-HD and the inter-HD indicate the ideal HD distribution. The ideal inter-HD is distributed around half of the ID length.

Fig. 12 shows the distribution of inter-chip Hamming distance for the FPGA implementation. The distribution is symmetric about its center value 32 and has a uniqueness value of 0.4899. Number of ones and zeros in each of the total sample for the proposed flip-flop is shown in Fig. 13. Results demonstrate that the proposed flip-flop has a good distribution of zeros and ones in each sample.

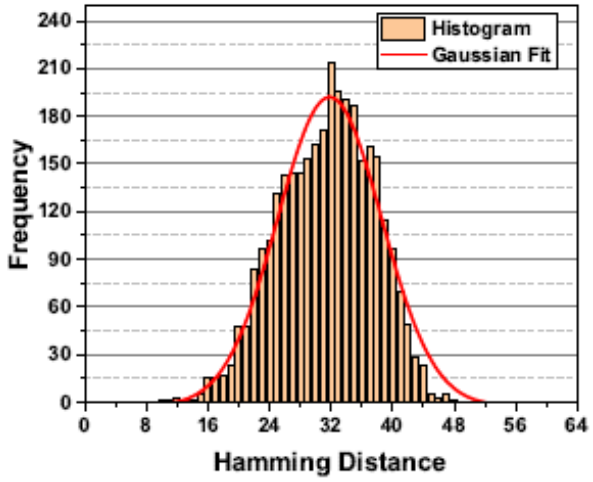

Figure 12: FPGA implemented distribution of Inter-Chip Hamming Distance for the proposed flip-flop.

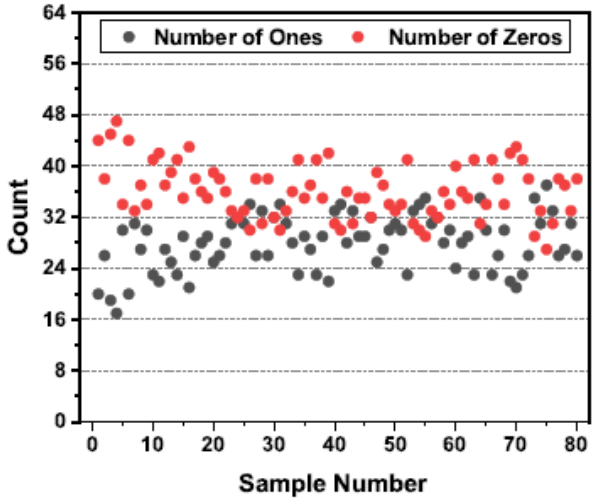

Figure 13: FPGA implemented distribution of one and zero for the proposed D flip-flop.

- Among all the considered architectures, the **proposed architecture has the highest uniqueness**, which is very closed to the ideal value of 0.5.
- Conventional PUF Uniqueness = **0.112**.
- Proposed PUF Uniqueness = **0.4901**.

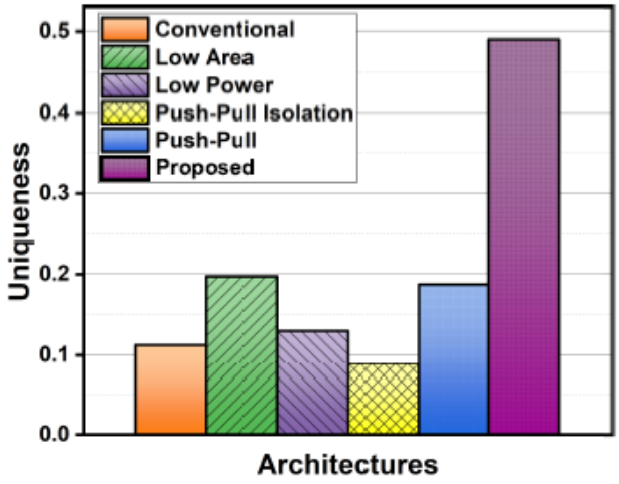

Figure 14: Simulated Uniqueness of Various Architecture.

Reliability:

Reliability is a measure of the PUF stability under various environmental conditions. Ideally, the PUF response under varying environmental conditions should be the same, however, temperature variations and supply voltage fluctuations are the two major factors which affect the performance of a circuit in practice. Reliability can be measured by comparing the two responses of the same chip taken at different temperatures and/or supply voltages. The reliability R of a chip can be measured by:

$$R = 1 - \frac{1}{k}\sum_{m=1}^{k}\frac{HD(R_a, R_a')}{n} \times 100\%$$

where k is the number of samples, n is the number of generated bits, R_a and R_a' are the responses taken at nominal and varying operating conditions, respectively. and HD (R_a, R_a') is the Hamming distance between R_a and R_a'.

- Measured the reliability by varying supply voltage and operating temperature.

- Reliability is a measure of the PUF stability under various environmental conditions.
- The proposed PUF has the **lowest supply voltage reliability** among all the architectures.
- The proposed one has **0.991 and 0.993** times less reliability, respectively, at a supply voltage of **1V** and **0.989 and 0.994** times less reliability at a supply voltage of **1.2V** as compared to the conventional and low power D flip-flop, respectively.
- The proposed PUF has the **lowest thermal reliability** among all the architectures.
- The proposed one has **0.988** and **0.979** times less thermal reliability, respectively, at **100oC** and **0.983** and **0.973** times less thermal reliability at **125oC** as compared to the conventional and low power D flip-flop, respectively.

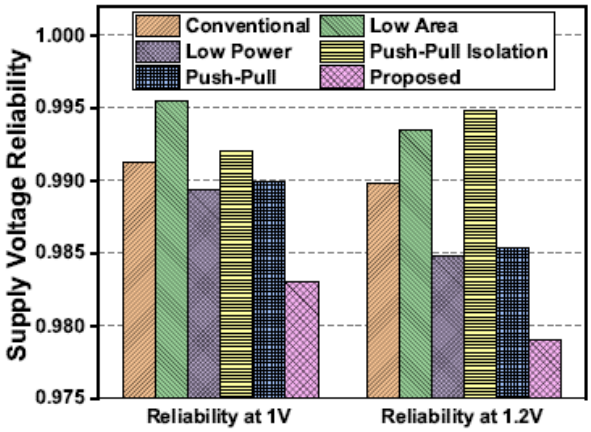

Figure 15: Supply voltage reliability of considered and proposed architecture at 1V and 1.2V.

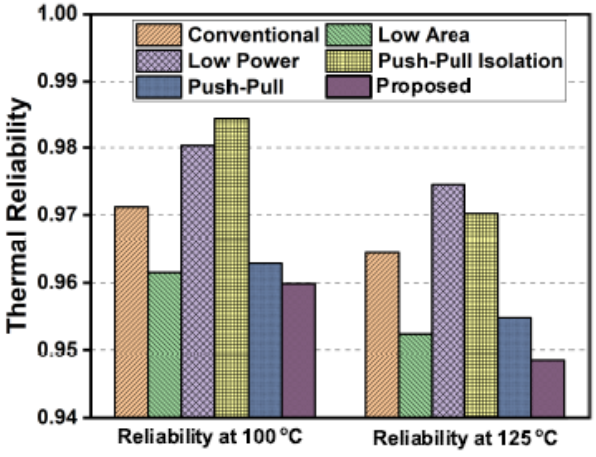

Figure 16: Thermal reliability of considered and proposed architecture at 100°C and 125°C

Chapter 5

Conclusion and Future Scope

In this project, we have presented a low power, low area tristate D flip-flop design which shows improved uniqueness. The proposed design shows a better uniqueness as compared to the other existing architectures without using any post-processing schemes. In ASIC implementation, the proposed architecture consumes 0.982 times less dynamic power and 0.994 times less silicon area when compared with the conventional D flip-flop. Further, it saves a large amount of area since it does not require any post processing schemes. From the above discussion, we can conclude that the proposed architecture has better PUF performance compared to the other existing architectures, which makes it suitable for PUF implementation in miniaturized IoT ASIC.

Reliability has been an issue with the current design which is not desirable in the long run. With more improved technology and the manufacturing processes of the hardware of integrated circuits used in IoT applications, reliability can be increased.

List of Publications:

Sajid Khan, Ambika Prasad Shah, Shailesh Singh Chouhan, Sudha Rani, Neha Gupta, Jai Gopal Pandey and Santosh Kumar Vishvakarma, “Utilizing Manufacturing Variations to Design A Tri-State Flip-Flop PUF for IoT Security Applications,” in communication with Analog Integrated Circuits and Signal Processing, 20199.

References

[1] N. Liu, S. Hanson, D. Sylvester, and D. Blaauw, "Oxid: On-chip one time random id generation using oxide breakdown," in 2010 Symposium on VLSI Circuits. IEEE, 2010, pp. 231–232.

[2] M. Akgun and M. U. Caglayan, "Puf based scalable private rfid authentication," in 2011 Sixth International Conference on Availability, Reliability and Security. IEEE, 2011, pp. 473–478.

[3] J. Guajardo, S. S. Kumar, G.-J. Schrijen, and P. Tuyls, "FPGA intrinsic PUFs and their use for IP protection," in International workshop on cryptographic hardware and embedded systems. Springer, 2007, pp. 63–80.

[4] Z. Paral and S. Devadas, "Reliable and efficient puf-based key generation using pattern matching," in 2011 IEEE International Symposium on Hardware-Oriented Security and Trust. IEEE, 2011, pp. 128–133.

[5] J. Zhang, Y. Lin, Y. Lyu, and G. Qu, "A puf-fsm binding scheme for fpga ip protection and pay-per-device licensing," IEEE Transactions on Information Forensics and Security, vol. 10, no. 6, pp. 1137–1150, 2015.

[6] Herschel, Sir William J.: The origin of finger-printing. Oxford University Press (1916)

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

[8] G. E. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," in Proceedings of the 44th annual design automation conference. ACM, 2007, pp. 9–14.

[9] Guajardo, J., Kumar, S.S., Schrijen, G.J., Tuyls, P.: FPGA intrinsic PUFs and their use for IP protection. In: Cryptographic Hardware and Embedded Systems Workshop, LNCS, vol. 4727, pp. 63–80 (2007)

[10] Y. Su, J. Holleman, and B. Otis, “A 1.6 pj/bit 96% stable chip-id generating circuit using process variations,” in 2007 IEEE International Solid-State Circuits Conference. Digest of Technical Papers. IEEE, 2007, pp. 406–611.

[11] P. Simons, E. van der Sluis, and V. van der Leest, “Buskeeper pufs, a promising alternative to d flip-flop pufs,” in 2012 IEEE International Symposium on Hardware-Oriented Security and Trust. IEEE, 2012, pp. 7–12.

[12] S. S. Kumar, J. Guajardo, R. Maes, G.-J. Schrijen, and P. Tuyls, “The butterfly PUF protecting IP on every FPGA,” in 2008 IEEE International Workshop on Hardware-Oriented Security and Trust. IEEE, 2008, pp. 67–70.

[13] R. Maes, P. Tuyls, and I. Verbauwhede, “Intrinsic PUFs from flip-flops on reconfigurable devices,” in 3rd Benelux workshop on information and system security (WISSec 2008), vol. 17, 2008, p. 2008.

[14] L. P. Ching and O. G. Ling, “Low-power and low-voltage Dlatch,” Electronics Letters, vol. 34, no. 7, pp. 641–642, 1998.

[15] U. Ko and P. T. Balsara, “High-performance energy-efficient D-flip-flop circuits,” IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 8, no. 1, pp. 94–98, 2000.

[16] G. Gerosa, S. Gary, C. Dietz, D. Pham, K. Hoover, J. Alvarez, H. Sanchez, P. Ippolito, T. Ngo, S. Litch et al., “A 2.2 w, 80 mhz superscalar risc microprocessor,” IEEE Journal of Solid-State Circuits, vol. 29, no. 12, pp. 1440–1454, 1994.