Study of Physical Unclonable Functions at Low Voltage on FPGA

Kanu Priya

Thesis submitted to the faculty of the Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
In
Computer Engineering

Leyla Nazhandali, Chair
Patrick Robert Schaumont
Joseph G. Tront

July 22, 2011
Blacksburg, Virginia

Keywords: Physical Unclonable Functions, Ring Oscillator, Process Variation,
Uniqueness, Stability, Low Power

Study of Physical Unclonable Functions at Low Voltage on FPGA

Kanu Priya

ABSTRACT

Physical Unclonable Functions (PUFs) provide a secure, power efficient and non-volatile means of chip identification. These are analogous to one-way functions that are easy to create but impossible to duplicate. They offer solutions to many of the FPGA (Field Programmable Gate Array) issues like intellectual property, chip authentication, cryptographic key generation and trusted computing. Moreover, FPGA evolving as an important platform for flexible logic circuit, present an attractive medium for PUF implementation to ensure its security.

In this thesis, we explore the behavior of RO-PUF (Ring Oscillator Physical Unclonable Functions) on FPGA when subjected to low voltages. We investigate its stability by applying environmental variations, such as temperature changes to characterize its effectiveness. It is shown with the help of experiment results that the spread of frequencies of ROs widens with lowering of voltage and stability is expected. However, due to inherent circuit challenges of FPGA at low voltage, RO-PUF fails to generate a stable response. It is observed that more number of RO frequency crossover and counter value fluctuation at low voltage, lead to instability in PUF. We also explore different architectural components of FPGA to explain the unstable nature of RO-PUF. It is reasoned out that FPGA does not sustain data at low voltage giving out unreliable data. Thus a low voltage FPGA is required to verify the stability of RO-PUF. To emphasize our case, we look into the low power applications research being done on FPGA. We conclude that FPGA, though flexible, being power inefficient, requires optimization on architectural and circuit level to generate stable responses at low voltages.

Acknowledgements

I would like to thank my family for their continued encouragement and support.

I would like to thank Dr. Leyla Nazhandali for her continued support and for providing me with interesting research topics.

Thanks to Dr. Patrick Schaumont and Dr. Joseph Tront for their participation in my thesis committee.

I would like to thank my manager Robert Schlub at Apple for his support and encouragement. I thanks my colleagues Dean Darnell, Nanbo Jin, Nirali Makhija and Yuehui Ouyang for their help.

I would also thank my friends Priyanka Kulkarni and Tina Nandy for their continued support.

Finally, thanks to my friends Michael Henry, Abhranil Maiti, Dinesh Ganta, Meeta Srivastav, Vignesh Vivekraja and Sinan Huang for their assistance through the course of my masters.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction to PUF

"A Physical Unclonable Function (PUF) is a physical object that can take inputs and generate unpredictable outputs; it is unclonable in that the input/output behavior of a physical copy of one PUF will differ from that of the original one due to some uncontrollable randomness in the copying process."[1]

PUF generates a set of responses, as defined by the complex properties of the physical material, such as the manufacturing variability of CMOS devices, when stimulated by a set of challenges [2]**.** The challenge-response protocol is employed for authentication purposes. The complex properties of the material, forming its biometric information, can also be used to extract a key (or electronic fingerprint) [3]. The keys, thus obtained, are generated dynamically whenever a stimulus is provided. So the key is not stored in any memory device, unlike the conventional devices. Thus PUF provides an unclonable, secure, tamper proof and low investment option for maintaining security of a device.

There have been many implementations of PUF till date. Based on their fabrication, these are classified into two kinds, silicon and non-silicon PUFs. Silicon PUFs are fabricated using existing ASIC (Application-Specific Integrated Circuits) process. They are based on uncontrollable random process variations and thus generate unique signatures. Non-silicon PUFs derive their key from variations occurring in the physical device rather than from integrated circuit. Some of the non-silicon PUFs includes optical PUFs and coating PUFs. Ring Oscillator PUF, Switch Based/Arbiter PUF, Tristate Buffer PUF are examples for Delay based PUF which utilize the variations in propagation delay of identical circuits to derive a secret response [4].

PUF can be employed in many fields like anti-counterfeiting of valued products, access to valued services (banking, defense, government facilities etc), authentication of different devices (smart car, SIM card, computers, etc) to provide solutions, to generate private key, etc.

1.2 Motivation

With scaling of technology, low voltage operation or subthreshold operation has become an important field of research for low power applications. Subthreshold circuits minimize power consumed by applying low voltage. Same technique can be used for FPGAs (Field Programmable Gate Array) to reduce their power consumption.

FPGAs are integrated circuits that can be programmed using a Hardware Description Language (HDL) even after its fabrication, unlike ASICs (Application Specific Integrated Circuits). Owing to their flexibility, FPGAs have become one of the key digital circuit implementation media over the last decade. Their ability to support circuit design update after shipping and low non-recurring engineering costs are few of their advantages over ASIC. However, because of their reconfigurable nature, they consume a large amount of power.

Further, it has been shown in [5] that stability of PUF, implemented on ASIC, increases in subthreshold region. From [6], we know that for a circuit variations are enhanced in subthreshold region. As PUF's concept is based on variation, we intend to emulate ASIC's results on a more popular platform, FPGA and thus investigate its stability. Therefore in this work, we study the behavior of Physical Unclonable Functions (PUF) at low voltage on Field-Programmable Gate Arrays (FPGAs).

1.3 Contributions

In first part of the thesis, we describe a Ring Oscillator PUF (RO-PUF). A write-up is provided considering operation of a ring oscillator and how a RO-PUF generates digital signature. The two key metrics which measure the effectiveness of PUF are defined.

In subsequent part, a review of literature is done to understand the basic architecture of a FPGA. A few of the commercial FPGA structures are also looked into to get an insight into modern FPGA structures.

After this, we have discussed about the operating of FPGA at low voltage. This is usually done to reduce power consumption of a FPGA. However, we see that there are many challenges in running a nominal voltage FPGA at low voltage.

In later part, we provide the details for the low voltage experiment conducted on FPGA to establish reliability of PUF. It is established during the experiments that the variation increases when a FPGA is subjected to low voltage. Experiment's motive is to build on the variations obtained and study the stability of PUF over the range of temperature.

1.4 Thesis Organization

This work is organized as per the following:

1. Physical Unclonable Functions: This chapter gives a background of PUF. The classification of PUFs is discussed. In later sections, ring oscillator PUF is discussed, explaining terms such as uniqueness and stability. The related work section lists out different works dealing with PUF implementation on ASIC at low voltage.

2. FPGA Architecture: In this chapter, an overview of recent FPGA architecture is discussed (Logic Blocks + Interconnect + I/Os), with a brief note on its development history. This chapter also presents a compilation of study of few commercial FPGAs. Conclusion has a short note on various challenges faced in FPGA design process with the advancement in technology.

3. Low Voltage FPGA for Low Power Applications: This chapter starts with subthreshold operation explanation and presents the challenges faced in subthreshold FPGA design. Related works mention research done in this area including topics like interconnect architecture at low voltage, novel architecture or switches, etc.

4. Study of PUF on FPGA at low voltage: This chapter first gives a background of PUF emphasizing that RO-PUF is suitable candidate for implementation of FPGA. After this, the methodology and implementation of RO-PUF on FPGA is described. Result and analysis section compares the obtained results with previous works and brings out the reasons for instability in RO-PUF on FPGA. The analysis section has more experiments results to explain instability in 128 ROs configuration PUF. The RO frequencies are later characterized with the help of frequency binning and normalized difference in frequency distribution.

5. Conclusion: This chapter provides a summary for the experiment. It concludes explaining the results with the help of analysis done in previous chapters.

Chapter 2

Physical Unclonable Functions

2.1 Introduction

Physical Unclonable Functions (PUFs) derive secret from complex physical characteristics of the system. These functions can only be evaluated with the physical system and is unique for each physical system. The derived secret is volatile that exists in only in digital form when a system is powered on. Thus PUFs provide security to the physical systems by extracting the secret instead of storing it.

In following sections, we discuss different kinds of PUFs and different terminologies related with PUFs. We discuss Ring Oscillator (RO) PUFs and its key metrics in later sections.

2.2 Classification of PUFs

Based on their fabrication PUFs are of two kinds, silicon and non-silicon ones. Silicon PUFs are fabricated using existing ASIC process. They are based on uncontrollable random process variations and thus generate unique signatures. Non-silicon PUFs derive their key from variations occurring in the physical device rather than from integrated circuit. Some of the non-silicon PUFs includes optical PUFs and coating PUFs. A detailed description of silicon PUFs is as follows:

2.2.1 Delay based PUF (RO PUF, Switch Based/Arbiter PUF, Tristate Buffer PUF)

Delay-based PUFs utilize the variations in propagation delay of identical circuits to derive a secret response.

Ring Oscillator (RO) PUFs are based on frequency variations. It has N identical ROs. Though the characteristic frequency of each one should be identical, due to process variations, their frequencies vary slightly. An N-1 bits signature is generated comparing these frequencies. RO-PUFs can be used for challenge/response authentication where the ROs are chosen as per challenge applied to the MUX. Comparison is made between the

selected pairs and response is generated. Each RO-PUF gives a different response for same challenge.

Switch based/Arbiter PUFs are based on variation in the propagation of identical delay lines. It comprises of k switching elements each with two inputs, two outputs and a control bit. Depending on the control bit, a path is selected from input to output of each switching element. The last switching element is connected to clock and D input a flip flop (arbiter). A challenge is applied to the control bits and a path is selected for each element. Out of the two paths, one is faster due to variations and reaches the flip flop first to give one bit response.
Switch based PUF needs a larger challenge/response space than RO-PUF. Further, there are meta-stability issues with the flip flop. This is why RO-PUF is preferred over switch based PUFs.

Tristate Buffer PUF is same as Arbiter PUF with their switching elements replaced with a tristate buffer. The enable input of the buffer form the challenge. Unlike the Arbiter PUF, here the two paths to D and clock of the flip flop are completely independent of each other. Each buffer has slightly different propagation delay because of process variation. Thus one bit response is generated depending on the faster path. These PUFs are claimed to be more power (18%) and area (23%) efficient than the arbiter PUF.

2.2.2 Memory Based (SRAM and Butterfly PUFs)

Memory-based PUFs depend upon the unpredictable startup state of feedback based CMOS memory structures like flip flop, SRAM, etc to produce a secret response. On power up, these structures settle to one of their stable states and provide a signature when an array of these is used. The response is limited to the internal logic and is not accessible to outer logic for high security. These are more susceptible to environmental noise.

A 6-T SRAM cell consists of two cross coupled inverters and two access transistors. For PUF response, the process characteristics of the two load inverters are taken into account.

Butterfly PUF are designed for FPGAs. It consists of two latches with output of one connected to other. To obtain a response, both of the latches are forced into unstable state. After sometime the unstable condition is removed and the latches are let to settle down, thereby generating a one-bit response.

2.3 Sources of Variations

As silicon PUFs tap into random variations, we look at different sources of variations in IC. Variations in a circuit affect parameters such as threshold voltage, leakage current, delay, etc. This results in change in timing and logical behavior of a circuit. Various reasons for variations are:

- Manufacturing process: These include variations in process parameters such as oxide thickness, doping concentration, diffusion depth, etc. They arise due to non-uniform conditions in silicon wafer and fluctuations in diffusion of dopants. Limited resolution of photo-lithographic manufacturing process leads to variations in dimensions of transistors too.

- Operating conditions: The output of a circuit vary based on environmental factors such as temperature, external noise coupling, operating voltage fluctuation, etc. It is challenging to maintain a stable output in presence of environmental variations

- Aging: Aging leads to permanent degradation in characteristics of transistors with prolonged usage. Aging results in slower operation of circuits, irregular-timing characteristics, increase in power consumption and functional failures.

2.4 Ring Oscillator PUF

Ring Oscillator (RO) PUF was first proposed by Suh and Devdas [7]. RO-PUF is a delay based PUF, deriving its secret response from variations in propagation delay of identical ROs.

A RO-PUF (Fig 2.1) consists of identical ring oscillators, each of which oscillates with their characteristic frequency. Though identical, ROs have different frequencies because of process variation and environmental conditions.

A RO consists of a chain of odd number of inverter stages with the output of last inverter as input to first. The inverters are implemented using MOSFETs, for which the gate capacitance have to be charged for source-drain current to flow. This introduces a delay in the output of inverter, changing it after a finite amount of time the input changes. Each inverter adds to the delay resulting into a square wave, if a small amount of noise is reduced.

RO is sensitive to process and environmental variations. This is why ROs have been widely used for creating sensors to measure voltage and temperature effects on different platforms [8] [9]. PUF takes advantage of variation sensitivity of RO to generate key required for authentication or a secret response to a challenge.

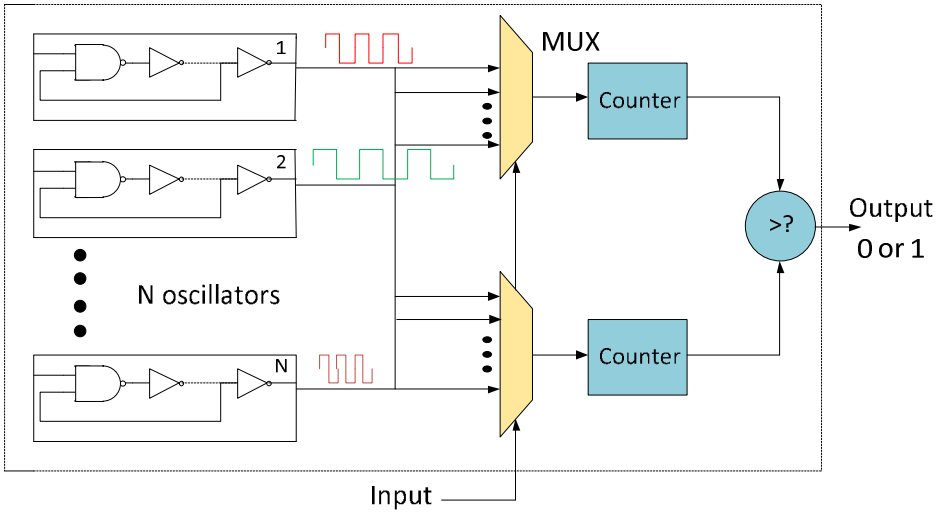

Figure 2.1: Ring Oscillator PUF [7]

2.5 Signature generation

To derive an N-1 bit digital signature of PUF, consisting of N ROs, comparison is made between the frequencies of oscillators. The comparison output bit is assigned 0 or 1 depending on which of the compared oscillators is faster. The selection for comparison is controlled by MUXes based on the input challenge to the circuit.

2.6 Key metrics

In general for all kind of PUFs, there are two metrics used to determine their quality, namely uniqueness and stability. These two quality factors help in establishing the effectiveness of various techniques on the characteristics of PUFs.

2.6.1 Uniqueness

Uniqueness (or Variability or Inter-chip comparison) is the measure of how easily PUFs can be differentiated from one another for same challenge. It can be expressed in terms of

hamming distance between the responses of two PUF for same input. Probability density distribution (PDF) of hamming distances characterizes uniqueness. If the PUF response is truly random in nature, i.e., there is equal probability of 0 or 1, the uniqueness should converge to 50%. So, PUFs with taller PDF centered on half the value of bits in digital signature are more unique than PUFs with flatter distribution curve.

2.6.2 Reliability

Reliability (or Stability or Intra-chip comparison) is the ability of a PUF to produce same output response for same challenge at different times in presence of environmental variations. It can be measured by number of unchanged bits while changing environment variables, but keeping challenge same. It can be quantified as PDF curves representing hamming distance of digital signatures of same PUF subjected to different environmental conditions. A PUF with PDF curve centered on 0 hamming distance is more stable than a PUF with flat PDF curve.

2.7 Related work

A variety of study and investigations are available for PUF in literature. They deal with novel design of PUF, building secure and robust PUF, etc. Here, we have made an effort to study the affect of environmental variations on PUF for FPGA when subjected to low voltage. Our experiment draws upon its idea from RO-PUF implemented on ASIC. Few of the related work which forms the basis for our experiment are as follows:

- [7] proposes RO-PUF to extract secrets from physical characteristics of ICs. It is shown that RO-PUFs provide low-cost authentication of ICs and generate volatile keys for cryptographic operations. This work establishes the guidelines for RO-PUF implementation of FPGA and its evaluating parameters.

- In [5], RO-PUF is shown to be more stable and unique in subthreshold region. The experiments are done using Monte Carlo analysis to obtain inter and intra chip variations with SPICE simulations for 90nm technology node. Effects of supply voltage and body bias, few of the conventional methods to stabilize a subthreshold circuit, are also investigated. It is found that the stability and uniqueness of the PUF increase by 7% and 18% respectively when the transistors are forward-biased.

- In [4], author mentions that subthreshold PUF is more stable than superthreshold PUF because the difference in characteristics of transistors gets amplified in subthreshold

region and it becomes difficult for noise signals to get over this barrier to cause errors. Various advantages of subthreshold PUF like, use of less hardware and energy efficient option have been enumerated. However, it is also observed that subthreshold PUF is ten times slower than superthreshold PUF. The fact that subthreshold PUF in ASIC is more stable led us to conduct same experiment on FPGA.

- [10] analyzes RO-PUF on FPGA. It is shown that systematic variations adversely affect stability of PUF. They proposed a configurable ring oscillator (CRO) technique to counter variability. Their compensation method improves the uniqueness of PUF by 18%. They report nearly 100% for CRO-PUF. As this was the experiment which varied temperature and voltage separately, our motivation was to observe the combined effect of both on RO-PUF response.

- [2] analyses different delay based PUFs on FPGA. While authors agree that RO-PUF works fine, other PUF show inconsistent results. They closely examine the delay based architecture as mapped into the logic blocks and conclude that Arbiter and Butterfly PUFs are ill-suited for FPGA. They observe that there is delay skew due to routing asymmetry which masks the effect of random process variations. This led us to choose to implement RO-PUF for our experiment.

- [6] investigates the spatial variation in digital circuits for a chip in 90nm technology. Their data indicates that both die-to-die and within die variations increase at low voltage. Both the variations are uncorrelated in low voltage domains. As PUF is built on variations, we take this concept to perform experiments on FPGA.

2.8 Summary

In this chapter, PUF and its classification are discussed. We take a look at different sources of variations, which can affect the response of PUF. Later, we discuss about RO-PUF in detail. We discussed what constitutes a RO-PUF and how it can be used to generate digital signatures. Further, the two key metrics which characterize the usefulness of PUF were also explained. We describe different previous works which form the basis for this thesis.

Chapter 3

FPGA Architecture

3.1 Introduction

FPGAs consist of array of programmable logic block which can be programmably connected through interconnects. The array is surrounded by programmable I/O blocks, which interface FPGA with different devices (Fig 3.1).

Since the beginning of its development, FPGA building blocks have undergone various changes. Power, speed and area are the key optimization factors which have made designers choose one design over other. In the following sections, a detailed overview of different types of FPGA and their architecture have been discussed.

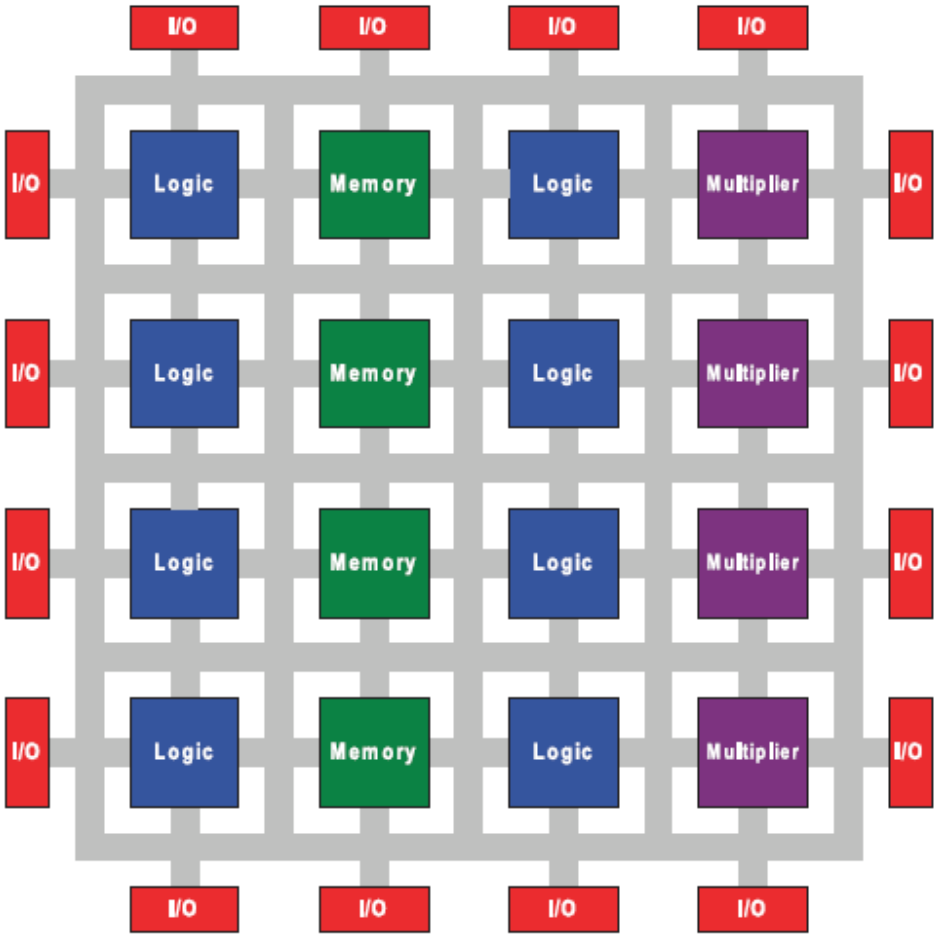

Figure 3.1: Basic FPGA structure

3.2 Architecture: Development and History

3.2.1 History

The origin of FPGA started as early as 1960s, with the development of integrated circuit. These devices provided architectural regularity and functional flexibility. By mid 1960s, field-programmability, the ability to change logic function of a chip after the fabrication process was achieved. Connections between the elements were fixed, but functionality was determined using programmable fuses which could be programmed using currents. In 1970s, read-only memory (ROM) based programmable devices provided ways to implement N-input logic functions with the help of N address inputs. However, area became an issue, which led to the development of programmable logic arrays (PLAs). PLAs have two-level programmable AND-OR logic planes which can be used to

implement logic functions. Here again, datapath and multi-level circuits made area prohibitive.

The first static memory-based FPGA was proposed by Wahlstrom in 1967. This device made both logic and interconnections to be programmable using a stream of configuration bits. First modern-era FPGA was introduced by Xilinx in 1984 which contained Configurable Logic Blocks.

3.2.2 Different Types of FPGA

FPGAs have an underlying programming technology that is used to control their programmable switches. Though a number of programming technologies have been used in past, flash/EEPROM, static and anti-fuse are used for modern FPGAs.

Static Memory (SRAM) based approach use SRAM cells to set the select line of multiplexers for steering interconnect signals and to store data as look-up tables (LUTs) for implementing logic functions (Fig 3.2). This technology has become dominant for FPGAs because of its re-programmability and use of standard CMOS process. However, use of 5 to 6 transistors per SRAM cell, its volatility on power down, security of configuration information and pass transistor switches are its drawbacks.

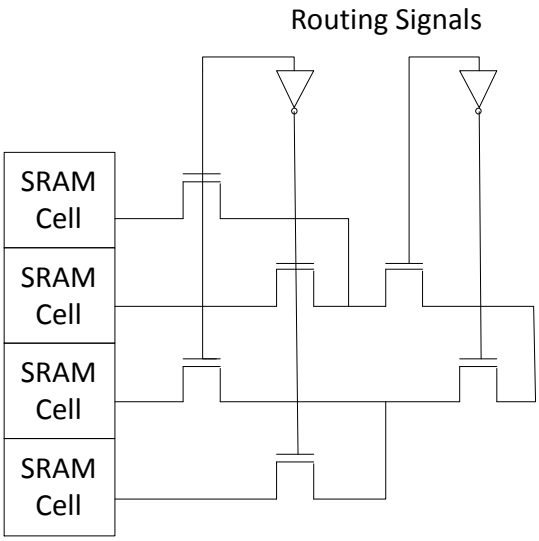

Figure 3.2: Static Memory Cells and Look Up tables

Flash technology uses a floating gate which injects charge on the gate that “floats” above the transistor. They form non-volatile cells which do not lose information when device is powered off, unlike SRAM approach. So an external resource is not required to store and load configuration data. However, these devices cannot be programmed infinite number of times. Other disadvantages are use of non-standard CMOS process and relatively high resistance and capacitance for such devices.

Anti-fuse technology is based on high resistance structures which can be “blown” to create low resistance permanent link to program FPGA. Since connections are made using metal-to-metal anti-fuses, the area overhead decreases. Further, resistance and parasitic capacitance is lower too, which makes it possible to include more switches on the device. Non-volatility is another advantage. However, use of non-standard CMOS process, inability to reprogram and inability of manufacturing tests to detect all possible faults makes it a lesser used approach.

3.2.3 Architecture Overview

Logic Block - Logic blocks implement logic functions. They form the basic computation and storage element of digital logic on FPGA. Over the years, there have been many developments in logic block and architects have chosen from coarse-grain to fine-grain architectures. A fine-grain logic block uses a transistor as its basic logic element, whereas a coarse grain one could consist of entire processor. Thus these structures could be made up of transistors, NAND gates, interconnection of multiplexers, lookup tables or programmable logic array (PAL/PAL) -style wide input gates.

Usually, the choice depends on three key metrics: area, speed and power. To understand the trade-offs of area-efficiency, speed and power for different FPGA architectures, application circuits are synthesized into different architectures through CAD flow, where again different parameters can be varied to study their affects.

Fundamentally, considering the area trade-off, as functionality of logic block increases, its size increases whereas less logic blocks are required to implement a design. Considering speed trade-off, as functionality of logic block increases, fewer logic blocks are on critical path and hence overall speed increases; whereas internal delay of the logic block increases which may offset the increase in speed. For power trade-off, dynamic and

static powers are considered. It has been shown in [12] that logic blocks having less area have less capacitance and thus consume less power. The design of a logic block depends on these trade-offs.

Though commercially, the logic blocks have developed into more complex blocks, we will discuss about K-input lookup table and PLA style block, a few of the basic ones. We will provide an example for other kinds of logic blocks in section 3.3 of this chapter.

PLA (Fig 3.3) have two-level AND-OR logic planes that can be programmed for logic functions [13]. LUT is a one-bit wide array with memory address lines as its input. A K-input LUT corresponds to 2K X 1-bit memory and user can realize K-input logic function's truth table [14]. Most of the SRAM based FPGAs use LUT in their logic blocks. LUTs along with flip-flop, MUX and few other logic elements, such as carry circuitry, particular to an FPGA, form logic block (Fig 3.4).

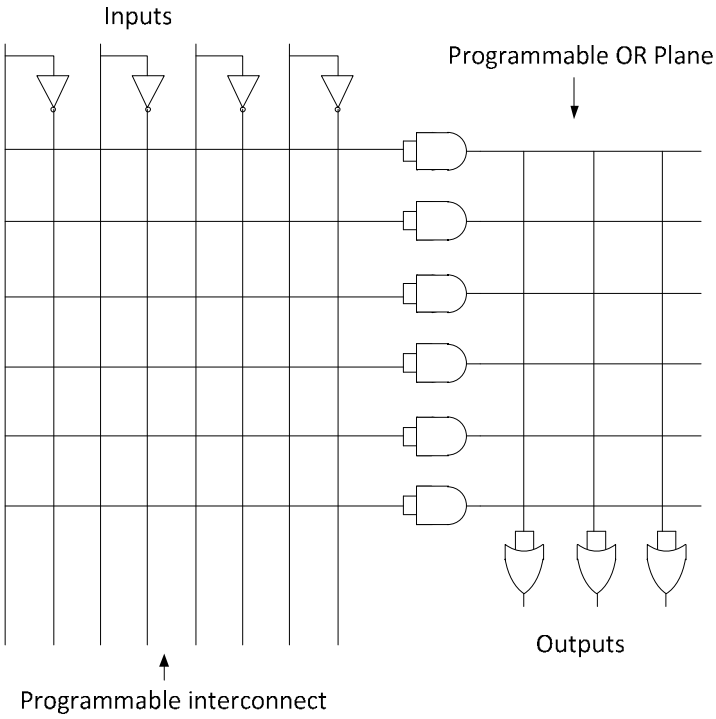

Figure 3.3: PLA architecture

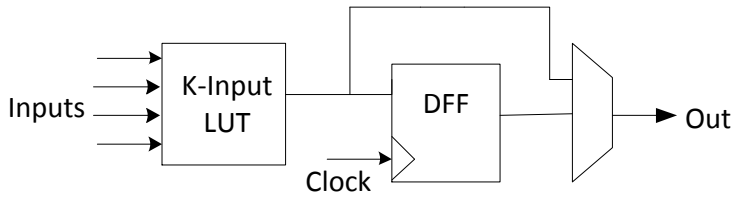

Figure 3.4: Basic Logic Element (BLE)

Interconnect structures - These constitute wires and programmable switches which are used to form desired connection between logic blocks and I/O blocks. The routing structures are organized as global routing (or macroscopic routing) and detailed routing (or microscopic routing). Global routing defines the relative position of routing channels considering positioning of logic blocks, how each channel connects to other and number of wires per channel. The microscopic structures deal with details such as lengths of wires, specific switching quantity and patterns between and among the wires and logic block pins. The global routing can be classified into hierarchical routing and island-style routing.

Hierarchical routing

Hierarchical routing separates FPGA logic block into distinct groups. Connection between logic blocks in a group is made using wire segments at lowest level of routing hierarchy. Connection between logic blocks in separate groups require traversal of hierarchy of routing segments. Fig 3.5 shows hierarchical routing. Recent commercial FPGAs do not use this type of routing for several reasons. Though this style of routing has more predictable inter-logic block delay, design mapping can be an issue. If logic blocks belong to different hierarchy, even if they are placed closely, the routing can incur a significant delay penalty. There can be wide variations in inter-block delay because of capacitance and resistance of interconnect.

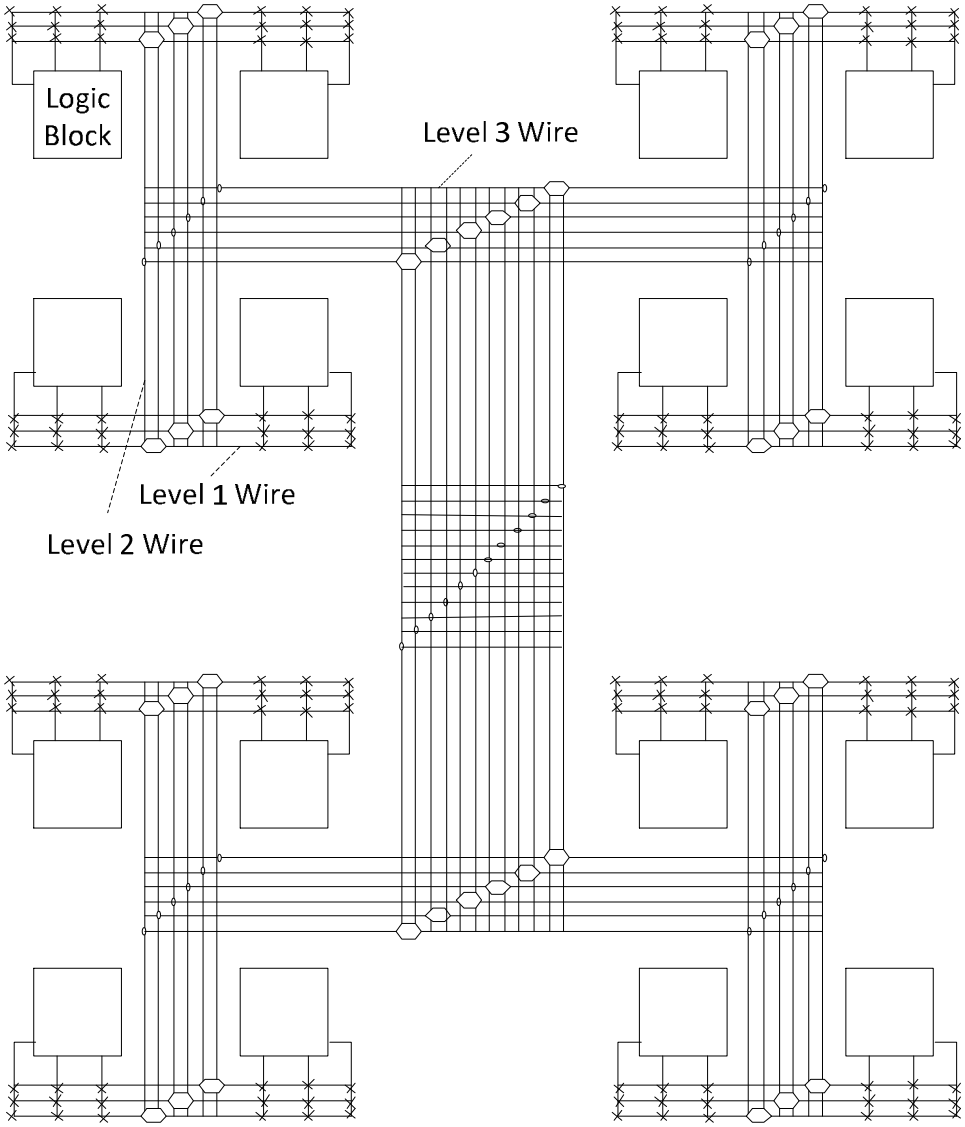

Figure 3.5: Example of Hierarchical FPGA

Island style routing

In island style routing, shown in fig. 3.6, FPGA logic blocks are arranged in two dimensional mesh with routing resources evenly distributed. It has routing channels, consisting of fixed number of wires, on all four sides of the logic blocks. Wire segments in each channel are of different lengths to provide most appropriate length for given connection. Most commercial SRAM based FPGAs use island style architectures. This type of routing provides efficient connections for a variety of designs.

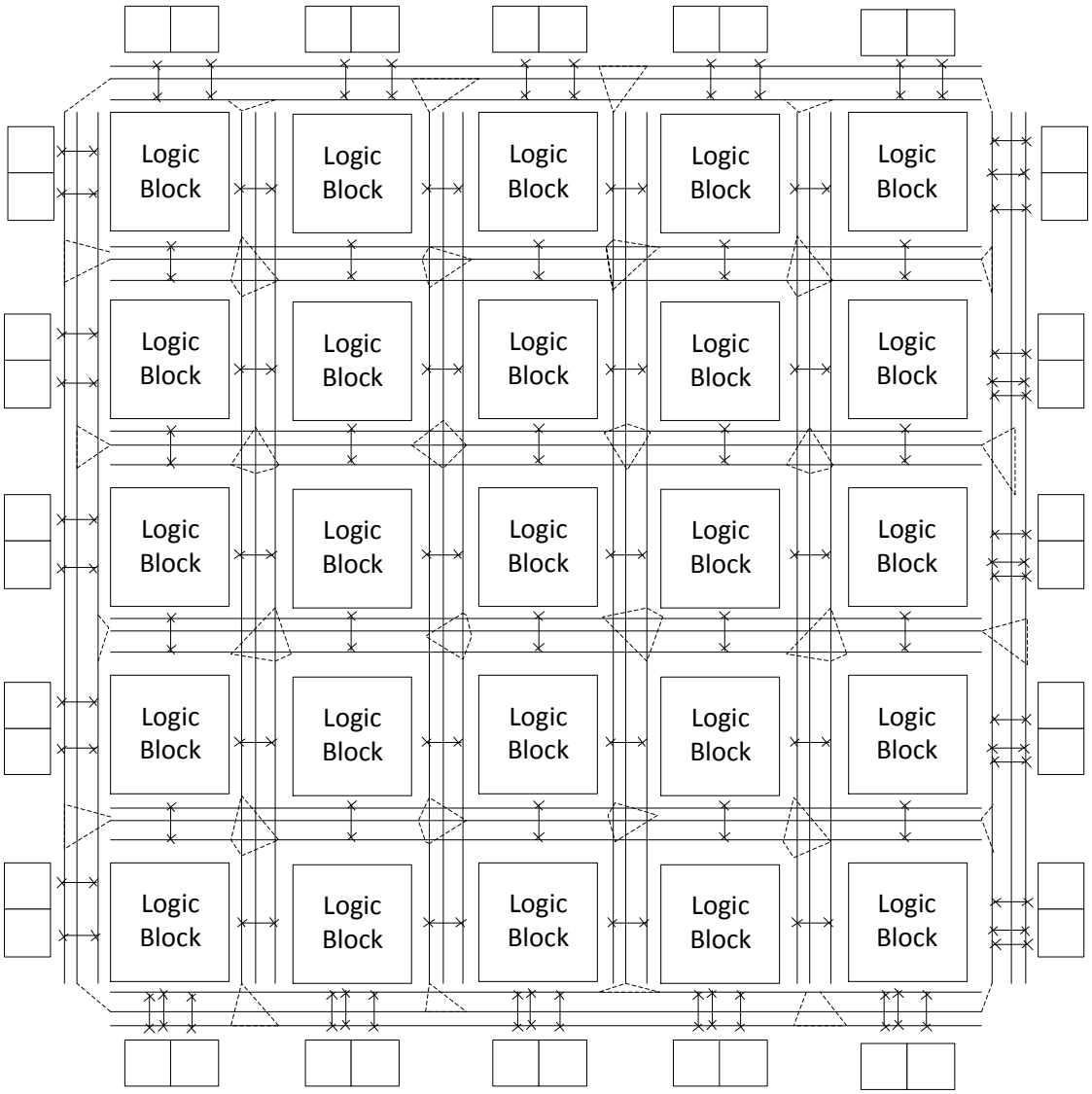

Figure 3.6: Island Style FPGA

The detailed routing of island style, shown in fig 3.7, defines logical structure of interconnection between wire segments in routing channels and between logic block I/O and routing channel wire segments. A logic block input pin connects to channel wire segment through switches in input connection. Similarly, its output pin connects via output connection block. A switch block, with the help of set of switch in it, forms connections between wire segments at intersection of horizontal and vertical channel.

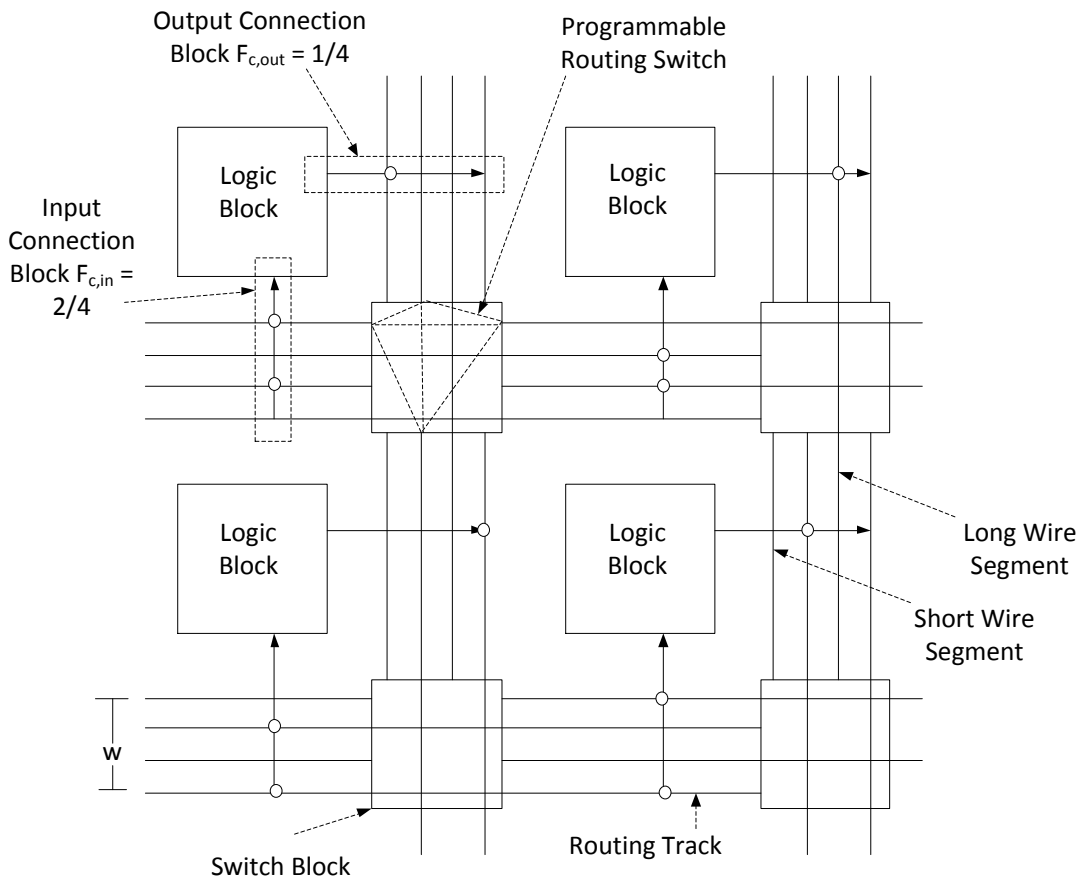

Figure 3.7: Detailed routing of island style FPGA

In addition to connection pattern and quantification parameters, detailed routing performance is decided by types of switches, size of transistors used to build programmable switches and metal width and spacing of FPGA wires [17]. Many FPGAs use pass transistors and tri-state buffers as routing switches (Fig 3.8). It was shown in [17] that routing architectures using both kinds of switches equally are fastest.

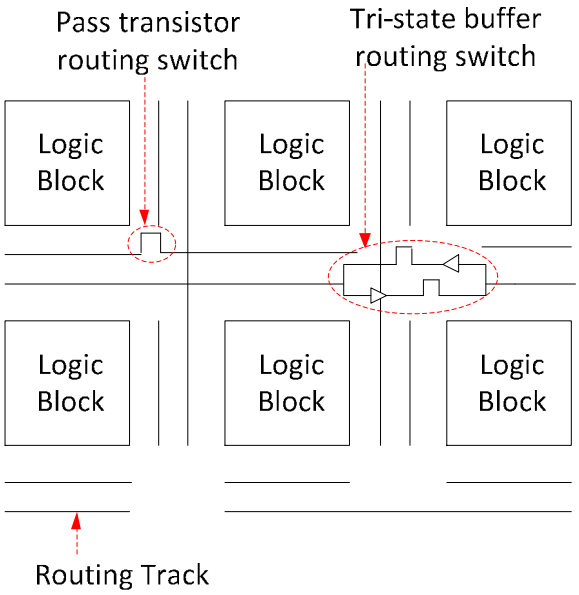

Figure 3.8: Routing switches in island style architecture

Input/Output architecture - The logic and routing structures interface with many different speeds and voltages with wide range of external components, that may be connect to FPGA, through I/O pads and cells on FPGA. An I/O pad along with supporting logic and circuitry is called I/O cell.

Major factor in deciding I/O architecture design is the great diversity in input/output standards. Thus the selection of standards to be supported by FPGA has to be made beforehand. An I/O cell can usually implement only the standards chosen for it. Supporting greater number of standards, increases silicon area of I/O cell and pin capacitance. However, usefulness of FPGA depends on it ability to support different signaling standards. So the standards are selected taking this trade off into consideration.

Once I/O standards to be supported are decided, I/O pins are required to be assigned with standards. Again, every pin can be given every standard, thereby increasing its capacitance, or different standards can be limited to different groups of I/O pins, which limit the flexibility of the printed circuit board. Since many of the signaling standards have conflicting requirements, most modern FPGAs have I/O banks in which I/O cells are grouped. Each bank shares supply and reference voltages. Number of pins to be present in each bank is again a debatable issue.

3.3 Few commercial FPGAs: Architecture Examples

Xilinx FPGAs - Xilinx FPGAs have array-based structure, with two-dimensional array of logic blocks interconnected by horizontal and vertical routing channels on a chip. Xilinx offers many generations of FPGAs, out of which we will take a look at XC4000 series features. XC4000 have configurable blocks (CLBs) based on look-up tables, used to implement a wide range of logic functions (Fig 3.9). Each CLB also contains two flip-flops. Further, users can configure LUTs as read/write RAM cells too. The chip includes a wide AND planes around periphery of CLB array to facilitate implementation of circuit blocks. Routing structures consist of horizontal and vertical channels. Each channel consists of different lengths of wire segments, to which CLBs are connected with programmable switches. [14]

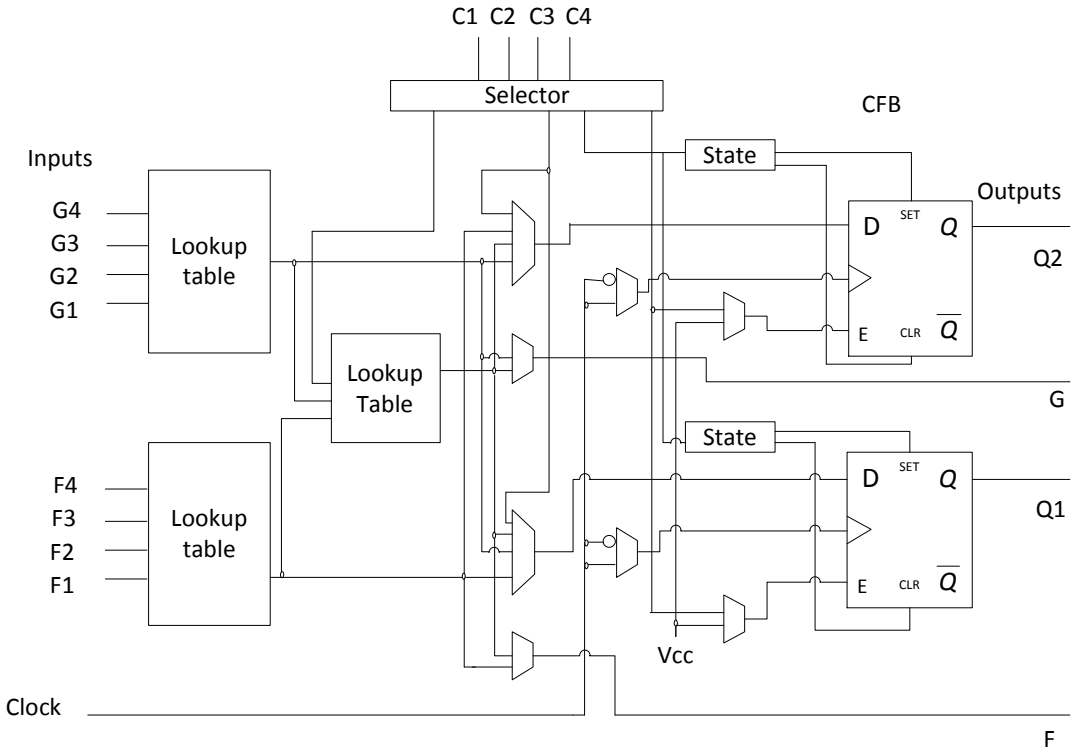

Figure 3.9: Xilinx XC4000 CLB [14]

Altera Flex 8000 and Flex 10K - Altera's SRAM based Flex 8000 series consists of three-level hierarchy. The lowest level consists of a set of LUTs as its basic logic block. The basic logic block, called logic elements, contains a four-input LUT, a flip-flop, special purpose carry circuitry and a cascade circuitry for implementation of wide AND functions. Eight logic elements group into one logic array block. Each logic array block

contains local interconnections, which can connect to any logic element. The local interconnects connect to FastTrack global interconnect. FastTrack consists of only long lines, making interconnect delays predictable. [14]

Flex 10K family offers all Flex 8000 features, along with variable-size blocks of SRAM called embedded array blocks. Embedded array blocks can be configured to serve as SRAM block with variable aspect ratio. [14]

Actel FPGAs - Actel offers three main FPGA families: Act 1, Act 2 and Act 3. Actel's devices use anti-fuse technology and a structure similar to gate arrays. The logic blocks are arranged in rows with horizontal routing channels between adjacent rows. The logic blocks are based on multiplexers and are small compared to those based on LUTs. These blocks consist of a AND and an OR gate connected to multiplexer-based circuit block (Fig 3.10). The multiplexer along with the two gates enables logic block to implement logic functions. The horizontal routing channels consist of various length wire segments with anti-fuses to connect logic blocks or wires.

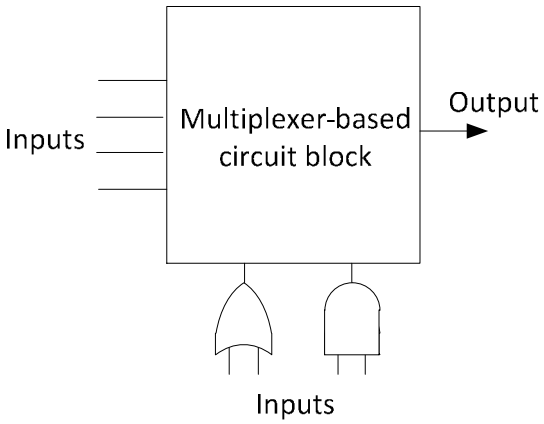

Figure 3.10: Actel Act3 logic module [14]

Quicklogic pASIC - Quicklogic has two anti-fuse based FPGA families, pASIC and pASIC2. Quicklogic FPGAs have array based structure like Xilinx FPGAs. Its logic blocks use multiplexers like Actel FPGAs and it's interconnect consists only of long lines. Its anti-fuse offers less resistance and low parasitic capacitance, when compared to Actel's anti-fuse. Its logic block is more complex than Actel's logic module, with more inputs and wide AND gates on multiplexer select lines. Each logic blocks contain a flip-flop.

3.4 Conclusion

In this chapter, we discussed about different types of FPGAs and its components, logic blocks, interconnects and I/O structures. Few basic kinds of logic blocks and interconnect structures were discussed too. To provide examples for FPGA architecture, we looked into few of the commercially available FPGAs too. Though the modern FPGAs developed into more complex structures, depending on the choice of designer considering various trade-off, these basic information give us a good understanding of FPGA.

There is a room for significant amount of innovation and improvement in all areas of FPGA. However, there are many challenges faced by designers with the advancement of technology. Few of the issues are soft errors, process variability and manufacturing defects. This has led to the development of much alternative architecture such as coarse-grained FPGAs, asynchronous FPGAs, etc.

Chapter 4

Low Voltage FPGA for Low Power Applications

4.1 Introduction

Low power circuits have stringent power requirements. For successful operation of these circuits over a period of time, energy consumed should be minimal. An approach to reduce energy consumed for a circuit is to operate it at low voltage. This has made subthreshold operation a preferred choice for low power applications.

As low power circuits tend to be very specific in their operation, ASIC implementation is an energy efficient solution for such circuits. However, inability to change the hardware makes it impossible to be reused for other application. FPGA offers hardware based operation with the flexibility to reconfigure that hardware.

FPGA, though flexible, consume power, creating a trade-off between efficiency and flexibility. This is why subthreshold FPGA, where power consumption is reduced by decreasing the supply voltage, is a good option for low power applications.

4.2 Subthreshold operation: What does it mean?

Conduction in subthreshold region means operating MOSFET with its gate-to-source voltage less than the threshold voltage, V_{th} (i.e., $V_{GS} < V_{th}$) [16] (Fig 4.1). Ideally, the transistor should turn off. However, some of the energetic electrons at source enter the MOSFET channel and flow to the drain. This resulting current is called subthreshold leakage current,

$$I_D \approx I_{D0}\, e \,\hat{}\, ((V_{GS} - V_{th})/n\, V_T)$$

where I_{D0} = current at $V_{GS} = V_{th}$, the thermal voltage $V_T = kT / q$ and the slope factor n is given by $n = 1 + C_D / C_{OX}$, with C_D = capacitance of the depletion layer and C_{OX} = capacitance of the oxide layer.

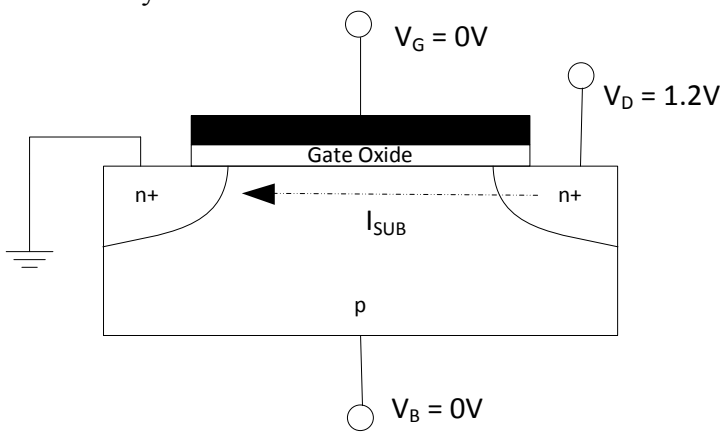

Figure 4.1: Subthreshold leakage of NMOS

The power consumed in a digital circuit as it switches voltage stored on total capacitance, C_{eff}, is given by

$$P_{switching} = f C_{eff} V_{DD}^2$$

where f is operating frequency, V_{DD} is supply voltage and C_{eff} is total stored capacitance. As power consumed varies quadratically with supply voltage, most effective method to lower power consumed is to lower supply voltage V_{DD}. Hence in subthreshold circuits, supply voltage is reduced below the threshold voltage of transistors to decrease consumed power.

4.3 Disadvantages of subthreshold operation

The main disadvantages of subthreshold operation are reduced on current to off-current ratio (I_{on}/I_{off}), slower speeds and increased sensitivity to variations.

The on current I_{on} of transistor reduces by many order of magnitude compared to strong inversion operation due to lower V_{GS}. This results in lower ratio of I_{on} to I_{off} which is likely to lead to circuit failure earlier than conventional CMOS circuits. The lower on current also reduces the speed of subthreshold circuits.

For conventional circuits, threshold voltage of a transistor exhibit normal distribution and the standard deviation of this distribution increases with process scaling to smaller technologies. Threshold voltage variation affects all circuits. However, subthreshold operation increases sensitivity of circuits to variations in threshold voltage. Further, current is an exponential function of V_{DD} - V_{th}, leading to exponential changes in current from threshold voltage variation. This results in further degradation of already reduced I_{on}/I_{off}.

4.4 Challenges for Subthreshold FPGA Design

In [20], authors taped-out a subthreshold FPGA chip in 90nm technology, using low-swing dual-VDD global interconnect. Their chip is 2.7 times smaller, 14 times faster and 4.7 times power efficient than conventional FPGA.

Subthreshold FPGA design faces a combination of subthreshold circuit challenges and problems inherent to FPGA architectures. There are three major challenges for the subthreshold FPGA design namely, variation, interconnect and clock networks, and memory [21].

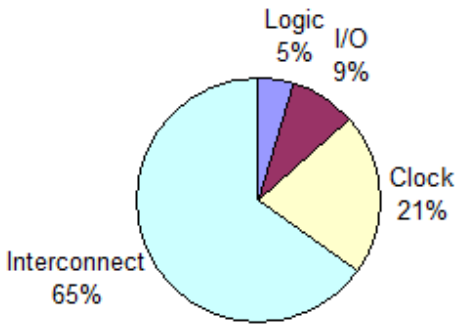

Figure 4.2: Energy breakdown of XC4003A

- Variations tend to disrupt a subthreshold design. Process variations spread variations in various circuit level parameters like delay, I_{on}/I_{off}, V_{th}, etc and make interconnect structures design difficult on FPGA. These cause different part of chips to behave in different manner. This is why a special care needs to be taken with place and route tool to avoid critical paths of synthesized design to be placed in slow area.

- As explained in previous chapter, interconnect network form an important part of FPGA architecture. A typical FPGA dissipates 60%-70% of their power in the interconnect network, 10%-20% in the clock network and 5%-20% in logic (Fig. 4.2) [21][11]. It is can be seen that distributed network of clock and interconnects consume significant amount of power.

  A single large clock network extends across the entire FPGA to drive all the registers in the design. Like ASICs in subthreshold region, variations lead to differences in driving strength of different buffers, which drive large capacitive loads of clock network of FPGA, thus leading to clock skew.

  Variations affect the interconnect network in similar manner. In subthreshold region, transistor drive current decreases whereas wire-dominated capacitive load remain the same. These capacitors, along with variability in interconnect drivers, cause large variations in delay. Further, the leakage on interconnect path from off devices in switch boxes and connection boxes fight against the weakened on-current of the drivers.

  To overcome the variability affect, repeaters can be deployed in switch boxes [21]. However, these still consume largest amount of power in the design. Thus designing a reliable, low power and fast interconnect is a key factor in subthreshold FPGA design.

- Memory can be seen in various contexts on a FPGA. First, CLBs contain registers that load clock net and lie on critical path. Important metrics of these registers do not differ from others in subthreshold context. Second, the LUTs consists of SRAM bit cells that map inputs to arbitrary functions and provide programmability function to FPGA. These require only write access during programming while continuous read access is required during FPGA operation. The read access to these lie on the critical path through the CLB. Thus LUT memory requires a design with high-speed, low-power reads. Further, there is a large volume of SRAM bit cells that provide configuration bits to store programmed function of the device. These bits drive multiplexers and switch boxes to configure the hardware for proper function. As these bits are concerned with read access most of the time, data retention and low leakage design is important for them. Thus high V_{th} transistors are required to reduce leakage for these bit cells.

4.5 Other Related works on Low Power Designs

In order to leverage the flexibility of FPGA, there has been a significant amount of research in the area of energy efficient FPGA architecture. Few of the energy related articles are as follows:

- In [23], authors design an energy efficient FPGA architecture. Reduction in energy consumption is achieved by employing both circuit design and architecture optimization techniques. As speed and energy performance is dominated by interconnect, for architecture optimization they have used hybrid architecture for interconnect incorporating mesh, hierarchical, etc type of schemes. For circuit optimization, the energy of interconnect is reduced by using low-swing circuit techniques. To optimize CLB, a cluster of 3-input LUTs is used as per prior studies. For clock network, dual edge triggered flip-flops are used to reduce activity by a factor of two.

- [24] proposes a new design for low-power LUT which can operate in two different modes, namely, high-performance and low power, as selected by SRAM cell, shared by a number of LUTs. The design uses a variation of well-known technique for leakage reduction, whereby, sleep transistor is included in header (P-network) and footer (N-network) networks. [25], proposed by same group as [24], presents a new programmable FPGA routing switch using sleep transistors networks.

- [26] targets low-power FPGA architectural designs. A FPGA architectural evaluation framework is proposed for LUT based architecture, taking into consideration the effect of architecture parameters like, LUT size, LUT input numbers, cluster size of LUTs, channel width, wire segmentation length, etc on power dissipation. The framework reports a detailed power distribution among different FPGA components and helps FPGA designers to optimize power consumption. Further, the evaluation is performed for 0.10um technology which is helpful to guide FPGA design for future technology generations.

4.6 Conclusion

In this chapter, we discussed about low voltage FPGAs, as an emerging technology for low power applications. Present population of FPGA fails to work at low voltages because of variations introduced in the circuit, when operating at low voltage. We looked

into various challenges faced in low voltage FPGA design. Finally, we briefly looked into few of the research works, which have optimized various architectural components of FPGA for low energy consumption.

Chapter 5

Study of PUF on FPGA at low voltage

5.1 Introduction

SRAM FPGAs, being volatile, need the configuration bits to be loaded on power up. There is a possibility of the configuration bits to be intercepted or stolen [10] [11]. PUF based authentication provides a solution to this problem. PUF solves issues like intellectual property (IP) protection, chip authentication and cryptographic key generation for FPGAs.

There are few difficulties in implementing PUF on FPGA. Unlike ASIC, it is not possible to exploit layout design techniques and one doesn't have access to gate-level structure of FPGA. Programming is done only through logic blocks and interconnects. This factor loses out on variation information because of the averaging effect of individual component-level variation information over larger composite structures of logic blocks.
There have been many PUF implementations in literature. However, it is not possible to implement all of them on FPGA. Many of PUF designs require careful routing (such as Arbiter PUF and Butterfly PUF [2]) to be implemented on FPGA.

RO-PUFs, sensitive to process variations, make an ideal candidate for FPGA. Further, it is easier to implement identical ROs using hard macro technique [7]. However, the process variations and environmental noise also degrade their uniqueness and reliability of RO PUF.

In this work, an experiment was conducted to study the variability and stability of RO-PUF in FPGA at low voltage. The following sections describe implementation and its analysis in details.

5.2 Implementation

This section describes implementation of RO-PUF on Spartan 3-E Starter board. The experimental set up shown in fig 5.1 includes Spartan 3-E Starter FPGA board, an oven,

external supply and USB to RS232 Serial DB9 Adapter Cable. The hardware is set up on the Spartan board and the data is collected on putty terminal on PC using RS232 cable. A population of three FPGA boards is used for data collection.

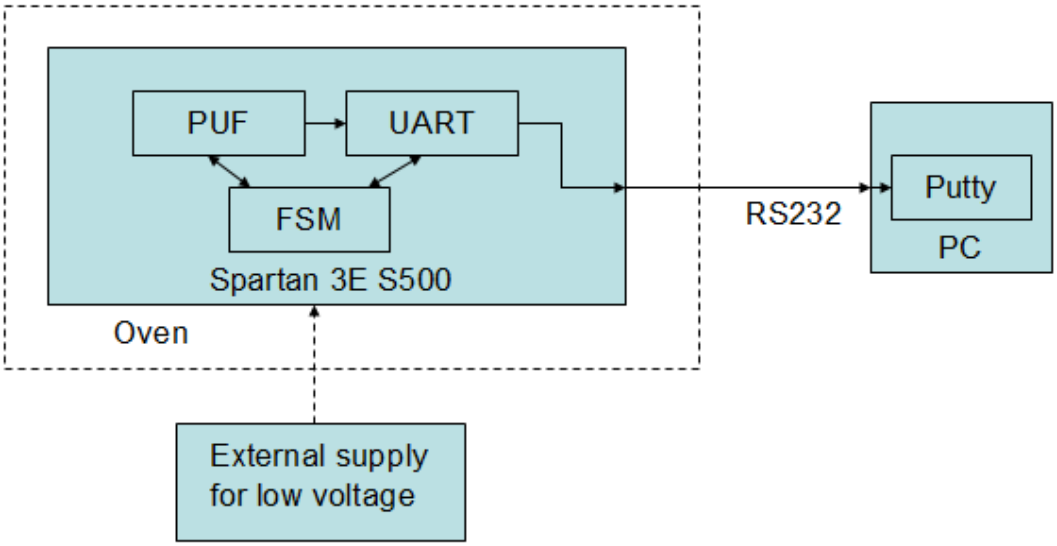

Figure 5.1: Complete experiment set up

5.2.1 Hardware

The hardware is designed using Xilinx ISE Design Suite 12.2 and validated using Chipscope. An array of 32, 64 and 128 5-stage ROs is implemented on Spartan 3E S500 FPGA. The 5-stage RO (Fig 5.2) uses four NOT gate and a NAND gate with enable input as one of its signal. The NAND gate's enable input helps to choose a RO for oscillation.

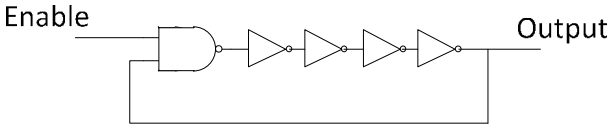

Figure 5.2: Five-stage RO

The array of ROs is laid out on the board, taking following points into account:

- Each of NOT and NAND gates of RO is implemented using look up table on Spartan Board (Fig 5.3). To program a N-input LUT, 2^N values are initialized in the INIT string as per truth table, with right bit representing address zero. So for 2-input NAND gate,

LUT2 (2-input LUT) is chosen and INIT string is 4'bwxyz, where w = output when input is 2'b11, x = output when input is 2'b10, y = output when input is 2'b01 and z = output when input is 2'b00. Similarly for NOT gates, 1-input LUT is programmed with INIT string as 01, which means that for address 0 output is 1 and for address 1 output is 0.

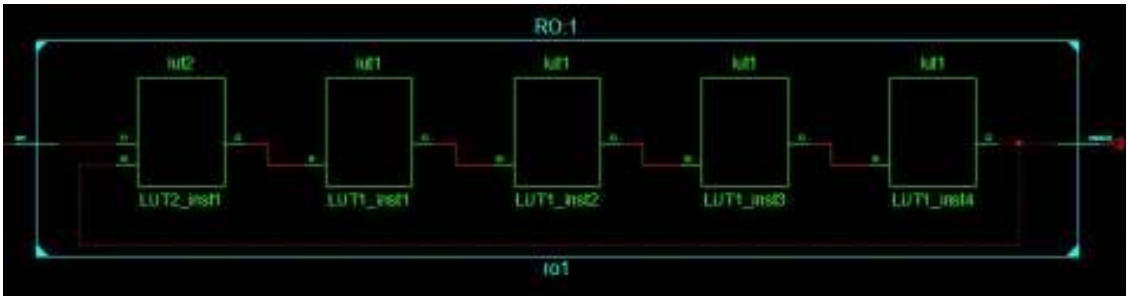

Figure 5.3: Five-stage RO with each gate implemented in a LUT

- Each RO is placed in one CLB on the board. On Spartan 3E 500 FPGA board, a CLB is made up of four slices, which contain look up tables, muxes, etc (Fig 5.4, Fig 5.5). LOC constraint is used to specify slice range for each RO. The range of slices is chosen such that one RO is confined to one CLB [27]. This ensures that only detailed routing is used for the ROs and they are not spread all over the chip using global routing. LOC constraint prevents the automatic placement by the place and route tool, which may lead to non-identical ROs. Using same constraint, ROs are also placed adjacent to each other. The slice numbers for LOC constraint are decided according to the number of ROs and by calculating the co-ordinates of slices as per Figure 5.4.

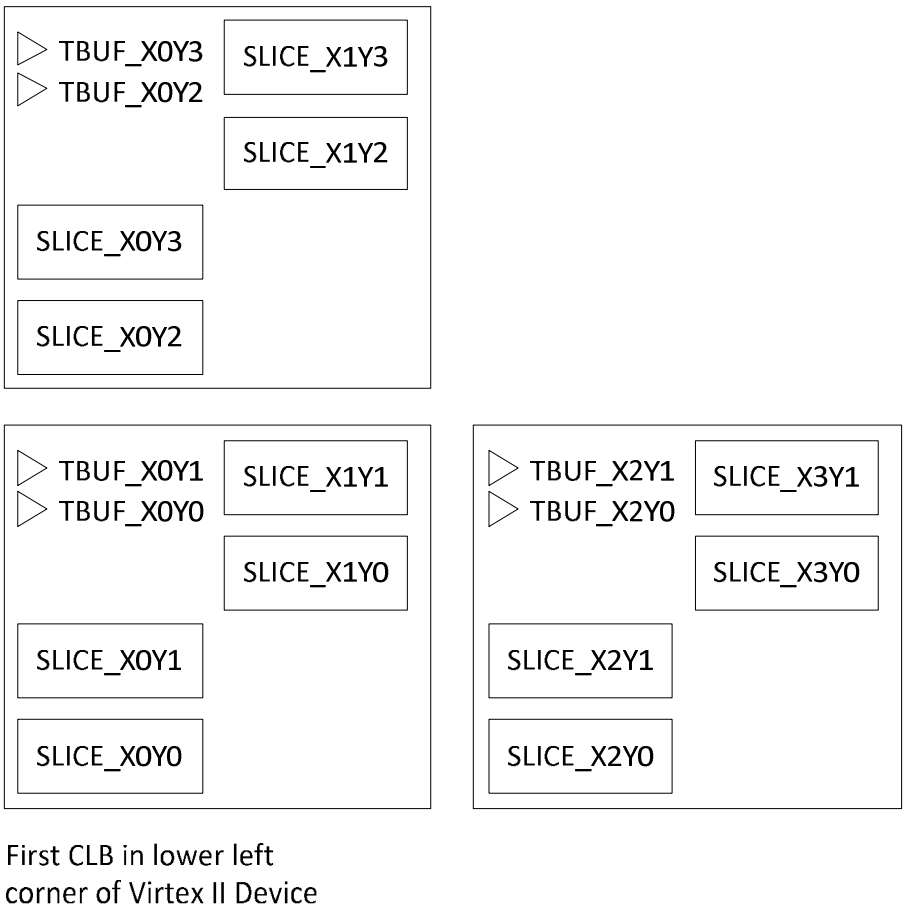

Figure 5.4: Slice and TBUF numbering in Spartan-3E and Virtex-II [27]

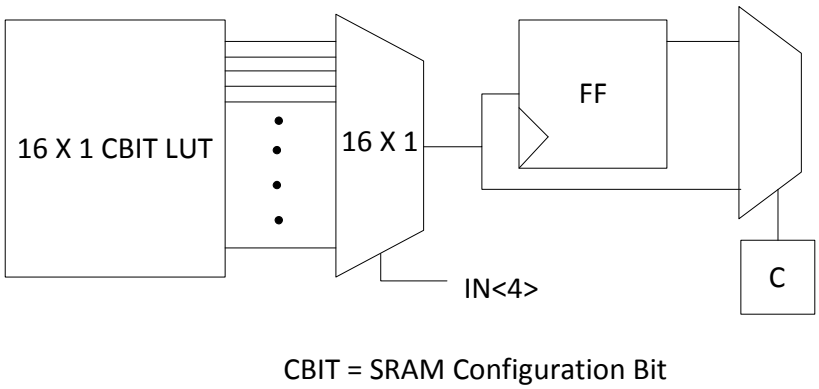

Figure 5.5: Basic Logic Element of FPGA

- Moreover, RO are placed as hard macro in the CLBs [28] (Fig 5.6). Though hard macros are an object of last resort, we choose hard macro method as it becomes easy to

duplicate ROs and ensure that they are identical [7]. Hard macros are good for creating configurations which don't get created by MAP, which in our case means identical ROs.

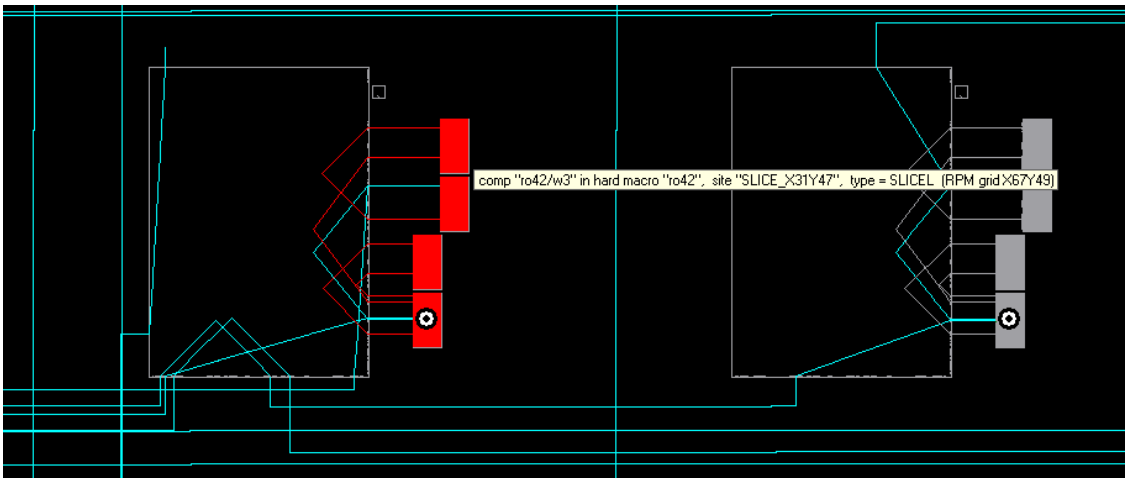

Figure 5.6: Five-stage RO implemented in one CLB as hard macro

- Further, RO, being a combinatorial loop, makes ISE tools to generate warnings as a combinatorial loop is considered a bad design. To avoid these warnings and stop optimization by synthesis tools, KEEP attribute is used for ROs [29].

The outputs of all ROs are fed to a MUX, which in turn is fed to a 32-bit counter. The counter's output is fed to a UART, which communicates with RS232 cable to send results to PC. The UART module, attached to the FPGA is obtained from Xilinx site [30]. ROs, MUX and UART are controlled using finite state machine (FSM). MUX select lines are used to select a RO. The selected RO runs the counter for fixed number of counts depending on the on-board 50 MHz crystal oscillator. Each RO is run 50 times to generate 50 counter values.

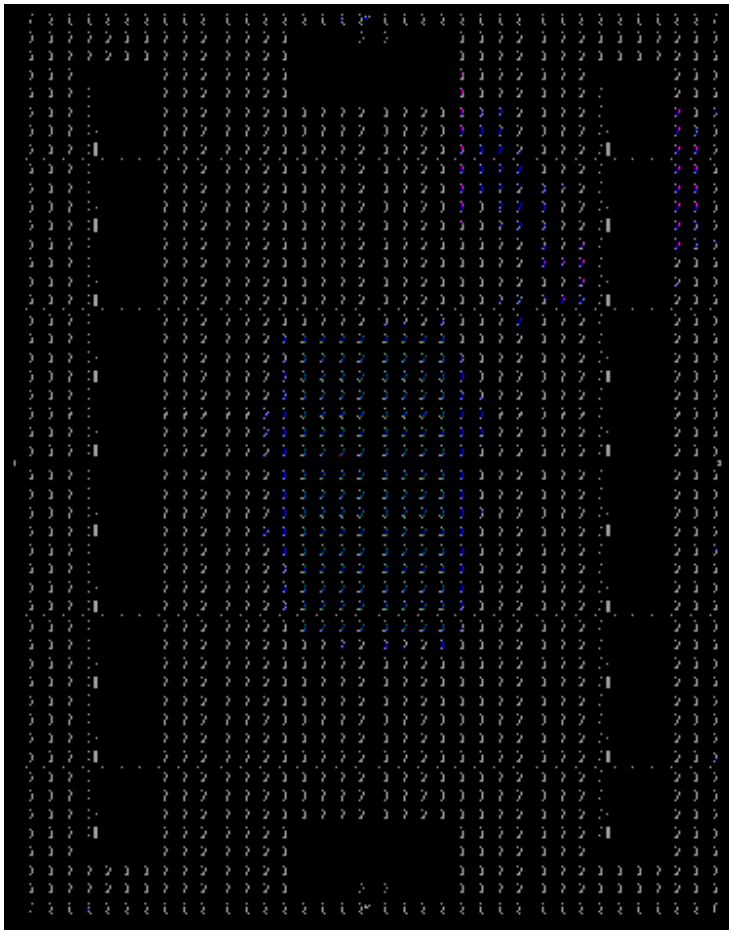

Figure 5.7: Complete hardware for 128 RO-PUF as seen in FPGA Editor

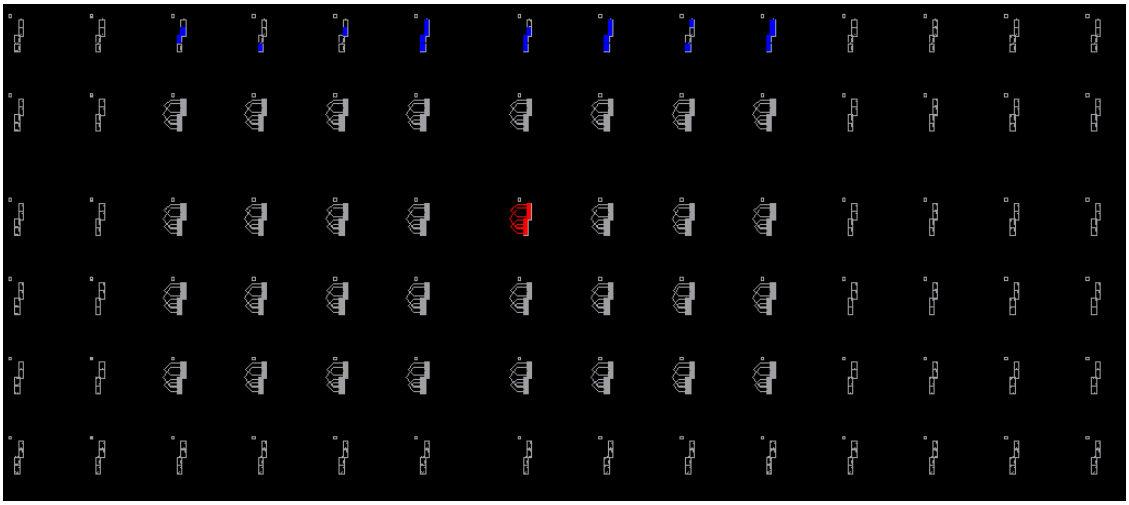

Figure 5.8: Array of 32 ROs placed as hard macro for 32 RO-PUF configuration as seen in FPGA Editor

This experimental set up was used to obtain counter values over a range of temperature at low voltage and at nominal voltage. The FPGA chip is powered with a supply of 1.2V through a voltage regulator. To vary the supply voltage of the chip, external supply is connected to the jumper JP7 pins [31].

5.2.2 Data collection

The counter values are obtained as hex using a putty terminal on PC. These counter values are then processed using MATLAB. The counter values of n ROs are compared to generate n-1 bit signature. 50 signatures are generated using the data obtained.

5.3 Methodology

The RO-PUF was chosen to be implemented on FPGA. The basic aim of the experiment is to measure the stability and uniqueness of RO-PUF on FPGA over a range of temperatures at low voltage. Thus the following methodology was used:

- Build RO-PUF on Spartan Board.
- Obtain the signature at nominal voltage and room temperature.
- Obtain PUF signature after toasting the board.
- Measure the stability of PUF.
- Decrease the supply voltage by 0.2V first, and then by 0.1V to find minimum working voltage for FPGA.
- Take a voltage above minimum voltage.
- Obtain the signature at this minimum voltage. Toast the board and measure stability.

Test Functioning of Spartan Board at low voltage: The operation of the FPGA at low voltage, as required by the experiment, is not recommended by its datasheet. Hence, first it was verified that ring oscillator and counter are working properly at low voltages.

Ring oscillator output signal was observed on an oscilloscope through I/O pin of FPGA. To validate the frequency of RO, its frequency was calculated as ((Counter value of RO * 50 MHz)/ Counter value of 50MHz clock of FPGA) [32]. The observed and calculated frequencies were matched to validate RO.

To verify the counter, it was fed with a RO signal. The frequency of RO was obtained from oscilloscope beforehand. The counter was made to run for a predetermined time according counter value set as per 50MHz clock of FPGA. Its value was obtained through RS232 cable on putty terminal. This value was matched with counter value obtained through hand calculation. Counter value was calculated as (Frequency of RO * (Counter value of 50MHz clock/ 50 MHz)).

Extracting Digital Signature : N-1 bit digital signature, for each FPGA chip, was generated comparing the frequencies (or counter values) of N adjacent ROs one by one in pairs, i.e., first oscillator's frequency (or counter value) was compared to second, second oscillator's frequency (or counter value) was compared to third and so on.

Measuring uniqueness: Uniqueness was measured by comparing digital signatures of PUF on different FPGAs. For the population of 3 FPGAs, a total of 3*2/2 comparisons were made to obtain a probability distribution graph.

Measuring stability: Stability at a voltage was measured by comparing the digital signatures of a PUF on same FPGA chip by varying its temperature. The chip was subjected to a range of temperature from room temperature to 70C at intervals of 10C. The stability was determined by observing the number of unstable bits across the 50 signatures at particular voltage and temperature.

Coefficient of variation: Coefficient of Variation (CoV) of a group of N ROs is the ratio of the standard deviation of its characteristic frequency, to the average characteristic frequency of all ROs of the group [4]. A higher value of CoV indicates that the frequency of ROs indicates that frequencies of ROs on different chips are more spread apart [4].

Scope of the experiment: The minimum voltage for FPGA chip to work properly was found to be 0.7V. So, data was collected from PUF by subjecting FPGA chip to supply voltages ranging from 0.7V to nominal voltage (1.2V), increasing supply voltage by 0.1V each time. To measure stability, the chip was subjected to a range of temperature from room temperature to 70C at intervals of 10C for each voltage.

5.4 Results and Analysis

5.4.1 Uniqueness

To characterize uniqueness at each voltage, we take a reference signature for each FPGA chip at room temperature. Each of these reference signatures is compared with that of other, making a total of three comparisons. The hamming distance is recoded for each comparison. A probability distribution curve gives the uniqueness for a population of 3 FPGA chips for 31 and 127 bit signatures respectively at 0.7V and 1.2V. X-axis represents the hamming distance and y-axis gives the probability.

It is observed that uniqueness for RO-PUF at 0.7V is almost equal to or better than that at 1.2V (Table 5.1). However, we will need a larger population of FPGA to conclude rightly on uniqueness of RO-PUF. This is why we do not present the uniqueness graphs obtained in this work.

Voltage (V)	Signature bit count	Hamming Distance
0.7 V	31	10, 7, 11
1.2 V	31	11, 9, 12
0.8 V	127	52, 57, 49
1.2 V	127	49, 48, 53

Table 5.1: Hamming distance between 31 and 127-bit signatures of each board

5.4.2 Low Voltage Operation Validation

Validation of operation of ROs – As specified in section 5.3, both the frequencies are measured for few ROs. The results are found to match closely. Few examples are provided in 5.4.6 section.

Validation of operation of counter – Counter values are verified as specified in section 5.3. Few examples with error percentage between observed and calculated counter value plots are provided in section 5.4.6.

5.4.3 Stability

To obtain graph for stability, we process the counter values obtained from Spartan board in MATLAB. As each RO is made to run 50 times, we extract 50 N-1 bit signatures for each voltage and temperature as explained in section 5.3. We take one of the signatures, at room temperature for particular voltage, out of 50, as reference signature. All the other

signatures, for that voltage, are compared to the reference signature. Bits which change across the 50 signatures, as the temperature is varied, are counted as unstable bits.

Fig 5.9 and 5.10 show number of unstable bits vs temperature plot for each FPGA for 31-bit signature at nominal and 0.7V respectively. Fig 5.11 and 5.12 show the same for 127-bit signature. It is observed from the plots that as the voltage decrease, number of unstable bits increase.

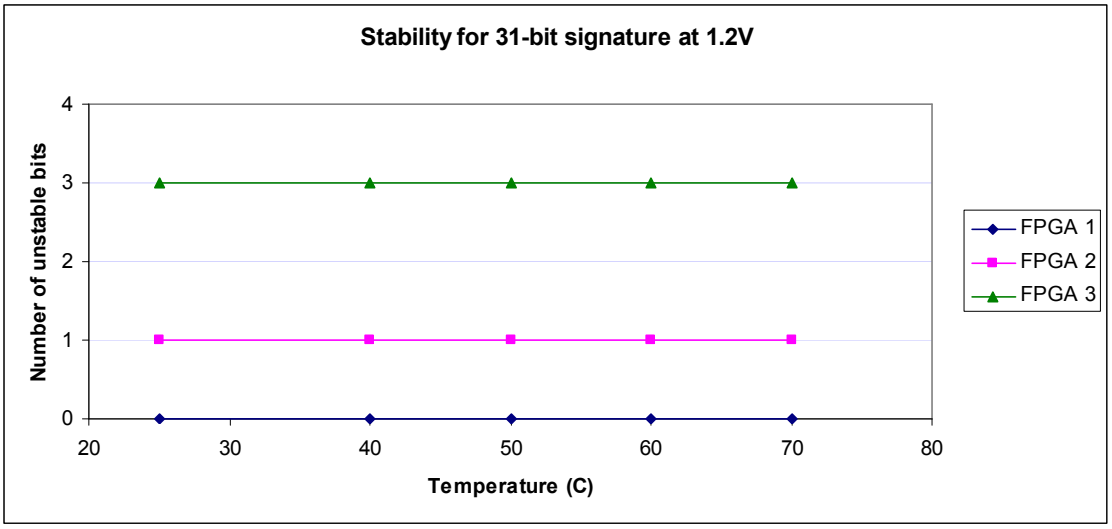

Figure 5.9: Unstable bits Vs Temperature for PUF with 32 ROs at nominal voltage

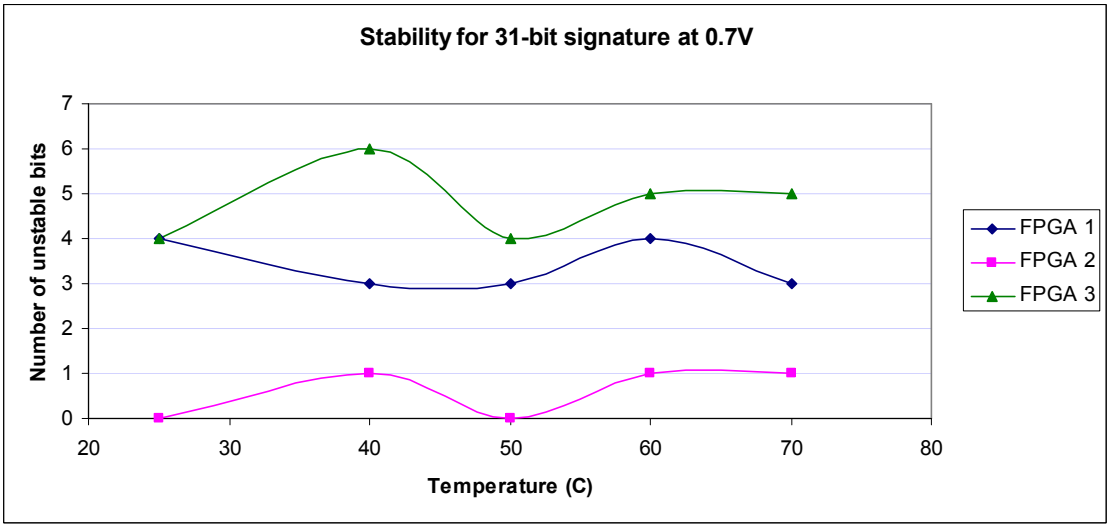

Figure 5.10: Unstable bits Vs Temperature for PUF with 32 ROs at 0.7V

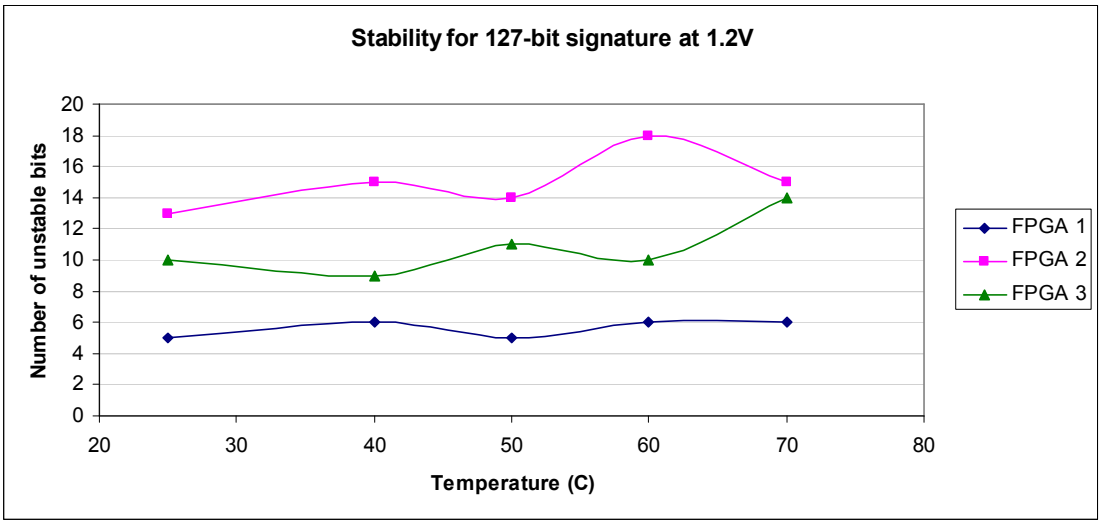

Figure 5.11: Unstable bits Vs Temperature for PUF with128 ROs at nominal voltage

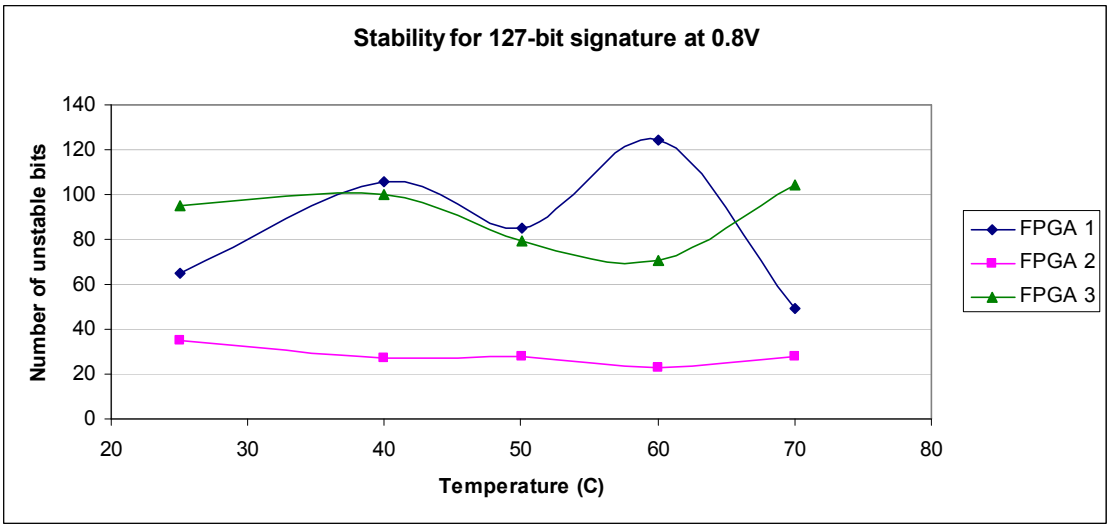

Figure 5.12: Unstable bits Vs Temperature for PUF with128 ROs at 0.8V

5.4.4 Coefficient of Variation

To further, analyze the data, we obtain coefficient of variation for different voltages across the temperature range. Table 5.2 gives the coefficient of variation for PUF with 32 ROs. It can be seen that CoV scales with decrease in voltage.

FPGAs	Coefficient of Variation at Voltages	
	1.2V	0.7V
FPGA 1	7.00E-03	8.94E-03
FPGA 2	7.78E-03	1.13E-02
FPGA 3	7.07E-03	1.06E-02

Table 5.2: Coefficient of Variation for PUF with 32 RO at 1.2V and 0.7V

5.4.5 Comparison with ASIC results

As we built our experiment based on ASIC results, we compare our experiment results with that of ASIC's. The coefficient of variation results confirm the data presented in [4]. [4] reports that CoV increases at a slow pace as the voltage is reduced from nominal voltage. We observe the same pattern for PUF on FPGA.

5.4.6 Analysis

We can see from the results (Table 5.2) the variations in frequency increase and frequencies are more spread apart with lowering of voltage on FPGA. Our idea was to build on these variations and investigate the stability of PUF on FPGA in the similar manner as it is done for RO-PUF at subthreshold voltage for ASIC.

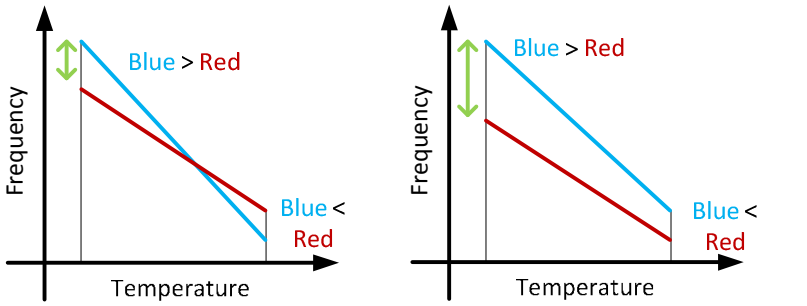

Figure 5.13: Effect of environmental variations on frequencies of ROs showing flipping of closer frequencies [7]

Further, the basic idea is that frequencies are far apart with increase in spread of frequency. Environmental variations effect frequencies of all ROs equally. Spread apart frequencies of ROs are less likely to flip (Fig. 5.13) under the effect of environmental variations [7]. Hence with the increase in coefficient of variation, stability is expected at low voltage.

However, since the experiment makes it clear that PUF is unstable at low voltage (Fig 5.9, 5.10, 5.11, 5.12) on FPGA, we have tried to bring out possible explanations for the same.

At first, we suspected that low voltage may not be enough for the I/O of FPGA, thereby counter data is not transmitted properly through RS232. This possibility was soon ruled out after referring to schematics of Spartan Board [31]. We had suspected that I/O may not be getting enough supply at low voltage; but according to schematic low voltage supply connection to FPGA chip should not affect RS232.

The Spartan Board's datasheet [33] mentions the fact that CMOS configuration latches require 1.0V to retain their data (Fig. 5.14). Spartan-3E FPGAs are programmed by loading configuration data into robust, reprogrammable, static CMOS configuration latches (CCLs) that collectively control all functional elements and routing resources. The FPGA's configuration data is stored externally in PROM or some other non-volatile medium, either on or off the board. After applying power, the configuration data is written to the FPGA [33]. As the FPGA is subjected to a voltage as low as 0.7V, we suspect that CCL fails to retain data.

Table 76: **Supply Voltage Levels Necessary for Preserving RAM Contents**

Symbol	Description	Min	Units
V_{DRINT}	V_{CCINT} level required to retain RAM data	1.0	V
V_{DRAUX}	V_{CCAUX} level required to retain RAM data	2.0	V

Notes:
1. RAM contents include configuration data.

Figure 5.14: Supply voltage necessary to retain CCL contents and RAM data [33]

To further explain the reason of failure, we observe that at 0.7V few of the counter values are garbled for 128-RO configurations. We know that Spartan 3E Starter's Kit FPGA is made up of several SRAM cells which are responsible for storing the configuration data. As seen from Fig 5.14, at low voltage it becomes difficult to distinguish between "0" and "1". Thus garbled data is obtained.

One more question that needs to be answered is why 128 RO configuration less reliable than 32 RO one (Fig 5.10, Fig 5.12) at low voltage. It is observed that resource utilization on FPGA increases by 3-4 times for 128 RO-PUF design (Fig 5.15, Fig 5.16). It is realized that more logic at low voltage, makes the design power hungry. Thus 128 RO circuit is more likely to fail at low voltage.

Device Utilization Summary				[-]
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Flip Flops	164	9,312	1%	
Number of 4 input LUTs	439	9,312	4%	
Number of occupied Slices	312	4,656	6%	
Number of Slices containing only related logic	312	312	100%	
Number of Slices containing unrelated logic	0	312	0%	
Total Number of 4 input LUTs	510	9,312	5%	
Number used as logic	430			
Number used as a route-thru	71			
Number used as Shift registers	9			
Number of bonded IOBs	6	232	2%	
Number of BUFGMUXs	2	24	8%	
Number of hard macros	32			
Average Fanout of Non-Clock Nets	2.74			

Figure 5.15: Resources consumed on FPGA for 32 RO configuration PUF

Device Utilization Summary				[-]
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Flip Flops	262	9,312	2%	
Number of 4 input LUTs	1,148	9,312	12%	
Number of occupied Slices	817	4,656	17%	
Number of Slices containing only related logic	817	817	100%	
Number of Slices containing unrelated logic	0	817	0%	
Total Number of 4 input LUTs	1,223	9,312	13%	
Number used as logic	1,139			
Number used as a route-thru	75			
Number used as Shift registers	9			
Number of bonded IOBs	6	232	2%	
Number of BUFGMUXs	2	24	8%	
Number of hard macros	128			
Average Fanout of Non-Clock Nets	2.41			

Figure 5.16: Resources consumed on FPGA for 128 RO configuration PUF

To add-on in this discussion, we analyze the 128 RO configuration again. An effort is made to optimize power usage in the design. Previously, all the ROs were enabled in the

PUF design, irrespective of the RO selected by MUX. This design is now modified to enable only the MUX selected RO, thereby saving power.

After this, the ROs are validated for the new design. Fig 5.17, Fig 5.18, Fig 5.19 and their related tables provide few examples of RO validation. It is seen that ROs oscillations distort at low voltage. Many counter values are zero at low voltage. This shows that ROs fail to oscillate at low voltage. Table 5.6 shows number of oscillators which fail at different voltages and temperatures for the three FPGAs. The Table 5.6 also shows that more number of ROs fail at room temperature than at high temperature. To explain this trend, we look at delay in a circuit, given as

$$\text{Delay} = (C_l * V_{dd})/I_{dsat}$$

where C_l is capacitance, V_{dd} is supply voltage, I_{dsat} is saturation current [34]. At high temperature, I_{dsat} increases because of decrease in threshold voltage [35], [36]. So delay decreases, making circuit faster (For example, the frequency of RO also increases at high temperature). Thus less number of oscillators fails at high temperature.

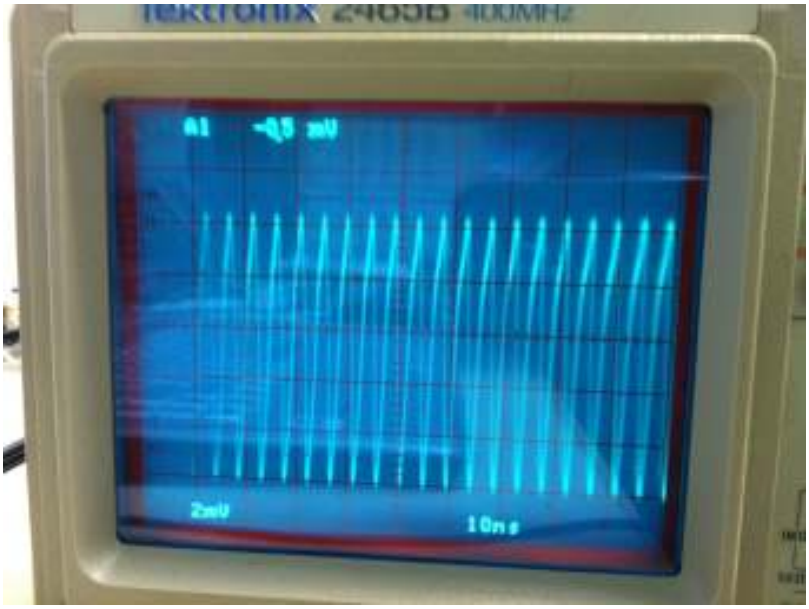

Figure 5.17: Validation of RO for PUF with 128 ROs at 1.2V at room temperature on FPGA 1

Oscillator Frequency Observed (MHz)	Observed Counter Value in Hex	Counter Value in Decimal	Oscillator Frequency Calculated (MHz)	Error %
166.67	00028761	165729	165.73	-0.56458

Table 5.3: Data for validation of RO for Fig 5.17

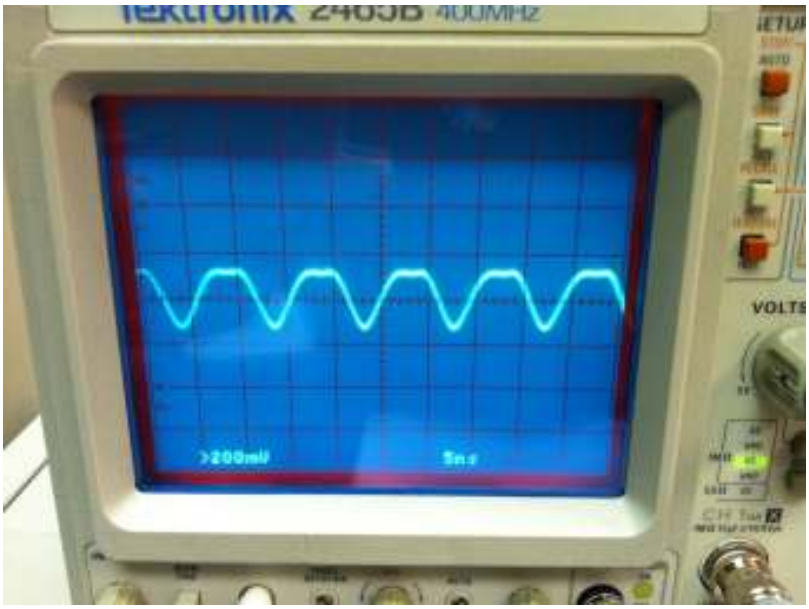

Figure 5.18: Validation of RO for PUF with 128 ROs at 0.9V at 50C on FPGA 3

Oscillator Frequency Observed (MHz)	Observed Counter Value in Hex	Counter Value in Decimal	Oscillator Frequency Calculated (MHz)	Error %
111.11	0001B4A7	111783	111.78	0.60570

Table 5.4: Data for validation of RO for Fig 5.18

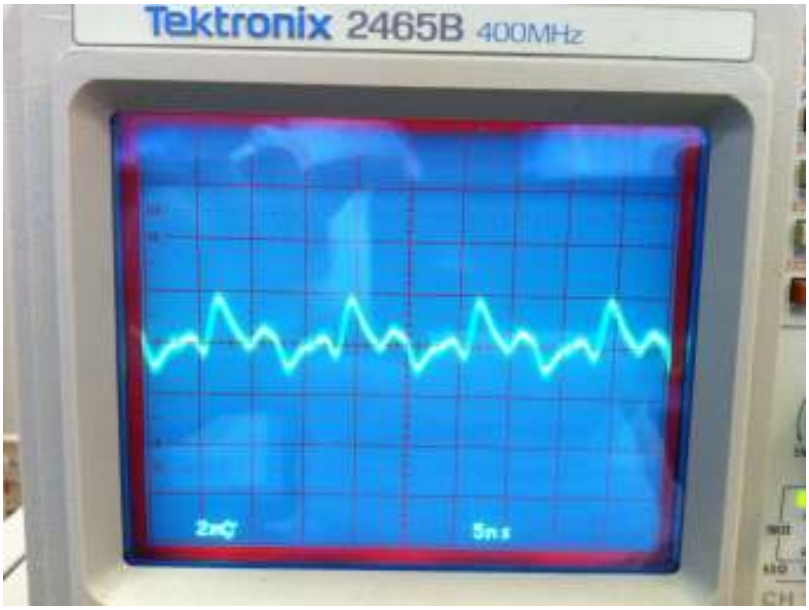

Figure 5.19: Validation of RO for PUF with 128 ROs at 0.7V at 70C on FPGA 2

Oscillator Frequency Observed (MHz)	Observed Counter Value in Hex	Counter Value in Decimal	Oscillator Frequency Calculated (MHz)	Error %
68.97	0001091E	67870	67.87	1.59489

Table 5.5: Data for validation of RO for Fig 5.19

Voltage	Temperature	Number of failed ROs on FPGAs		
		FPGA 1	FPGA 2	FPGA 3
1.2	25	0	0	0
	50	0	0	0
	70	0	0	0
1.1	25	0	0	0
	50	0	0	0
	70	0	0	0
1	25	0	0	0
	50	0	0	0
	70	0	0	0
0.9	25	0	0	0
	50	0	0	0
	70	0	0	0
0.8	25	0	0	0
	50	0	0	0
	70	0	0	0
0.7	25	0	2	90
	50	0	0	10
	70	0	0	3

Table 5.6: Number of ROs that fail to oscillate on FPGA boards at different temperatures and voltages

The counter is validated as explained in section 5.3. RO is run for 1ms and counter values are obtained. The counter values are also calculated with the help of frequency of RO observed on oscilloscope. Few of the examples of this validation can be seen in table 5.7 and 5.8. Plots of error percentage between two counter values at different voltages across three temperatures are shown in Fig 5.20, Fig 5.21 and Fig 5.22.

While validating counter and ROs, error between observed and calculated values were observed higher for more number of ROs at low voltage (Table 5.5, Table 5.9). It was suspected that error percentage in counter value increases at low voltage. To validate our analysis, each of the 128 ROs were observed separately on oscilloscope. The difference between ROs' observed and calculated frequency values were recorded as error percentage. For few cases the error percentage is higher than 0.5% at room temperature (Table 5.3 and Table 5.9). The possible explanations for this behavior are the variation of 50MHz on-board crystal clock frequency and observation error on oscilloscope. Spartan board user guide [31] says that 50MHz on-board crystal clock is accurate to ±2500Hz. Table 5.10 shows the number of ROs which showed more than 0.5% error at different voltages for two boards. It can also be seen from Table 5.11 that there is a discrepancy

between the signatures generated from the actual frequencies of ROs and from counter values.

Though it seemed that counter was at fault, we needed to find out the exact reason for instability in PUF responses under different conditions. To find out if the bit flips are caused by counter error or by RO frequency crossover, actual frequencies were recorded for 128 ROs from oscilloscope at 0.7V at different temperatures.

It was seen that the ring oscillators frequencies also flip at low voltage, making the PUF response unstable (Table 5.12). Thus a cumulative effect of RO frequency crossover and counter error is visible in Table 5.13. However, the fact that counter introduces bit flips and also nullifies bit flips caused by RO frequency crossover, counter error may not contribute as much as RO frequency crossover to the unstable PUF response (Table 5.13). Thus it can be said that RO frequency crossovers are the major reason for instability.

Temperature (C)	Observed Oscillator Frequency (MHz)	Observed Counter Value in Hex	Observed Counter Value in Decimal	Calculated Counter value	Error %
25	178.5714286	0002BE11	179729	178571.4286	-0.64406
50	218.34	0035AC7	219847	218340	-0.68548
70	212.7659574	00033F07	212743	212765.9574	0.010791

Table 5.7: Example for validation of counter for 127-bit signature at 1.2 V on FPGA 1

Temperature (C)	Observed Oscillator Frequency (MHz)	Observed Counter Value in Hex	Observed Counter Value in Decimal	Calculated Counter value	Error %
25	51.28205128	000C6F8	50936	51282.05128	0.679384
50	71.94	00018F5	71925	71940	0.02086
70	64.51612903	000FC6F	64623	64516.12903	-0.16538

Table 5.8: Example for validation of counter at 127-bit signature at 0.7V on FPGA 2

Voltage	Temperature	Error % in counter values on FPGAs		
		FPGA 1	FPGA 2	FPGA 3
1.2	25	0.644064914	0.740057	0.787036596
	50	0.685476718	0.108145	0.033853763
	70	0.010791164	0.104649	0.721043207
1.1	25	0.95911196	0.095671	0.053428531
	50	0.547827488	0.34302	0.089878319
	70	0.084598371	0.407731	0.276334278
1	25	0.541880519	0.443962	0.184558751
	50	0.739420309	0.11681	0.073475386
	70	0.405727149	0.296319	0.310870581
0.9	25	0.622401897	0.078678	0.538633033
	50	0.055920303	0.08246	0.181986746
	70	0.030449269	0.512688	0.459003443
0.8	25	0.845234332	0.358319	0.02578335
	50	0.042219032	0.352692	0.188105187
	70	0.462887388	1.177043	0.741953895
0.7	25	1.131642857	0.679384	0.293337003
	50	0.157436163	0.020855	0.142376391
	70	0.408098676	0.165376	0.376451845

Table 5.9: Error percentage in counter value validation for all three FPGAs at different temperature and voltages

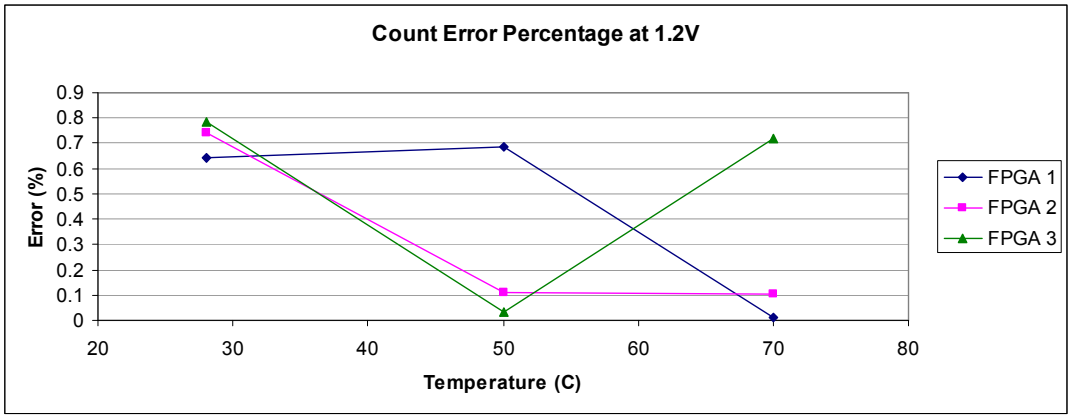

Figure 5.20: Error percentage in counter validation at 1.2V across three temperatures

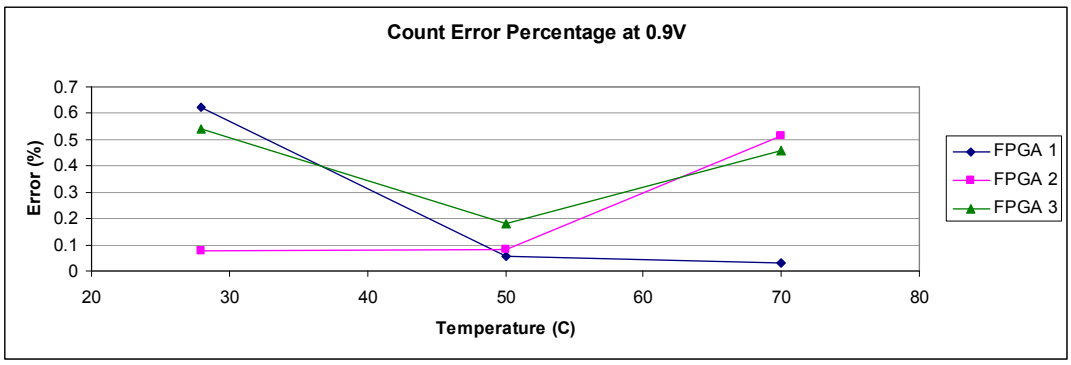

Figure 5.21: Error percentage in counter validation at 0.9V across three temperatures

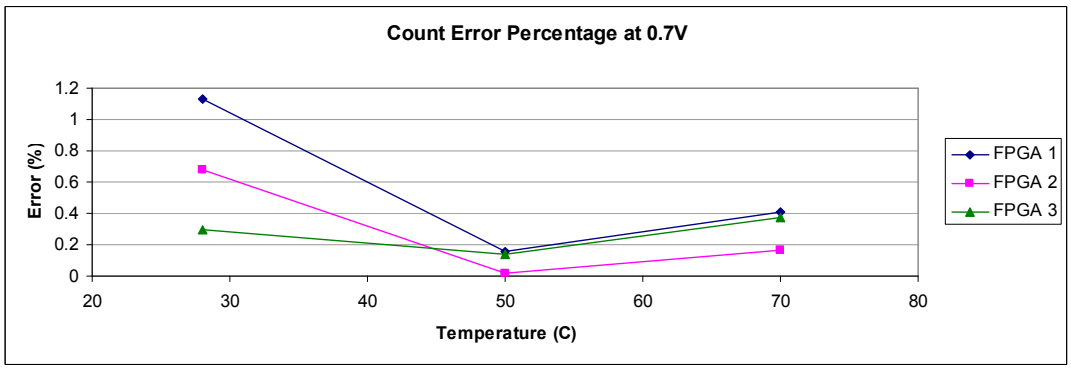

Figure 5.22: Error percentage in counter validation at 0.7V across three temperatures

FPGA	Voltage	Number of Ros where error% > 0.5 ?
FPGA 1	1.2V	0
	0.7V	25
FPGA 2	1.2V	0
	1.0V	14
	0.7V	60

Table 5.10: Error percentage in counter value validation for two FPGAs at different voltages

RO number	Observed Frequency (MHz)	Observed Count	Count in decimal	Calculated Frequency (MHz)	Error %	Signature from RO frequency	Signature from counter
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
32	94.87	00172E2	94946	94.946	0.08011	1	1
33	93.45	00016EE9	93929	93.929	0.512574	0	1
34	93.45	00016C5F	93279	93.279	-0.18299	0	0
35	93.8	0001700C	94220	94.22	0.447761	1	1
36	93.63	00016B3C	92988	92.988	-0.68568	1	0
37	93.45	00016D8D	93581	93.581	0.140182	0	1
38	93.45	16D16	93462	93.462	0.012841	1	1
39	93.28	16A3F	92735	92.735	-0.58426	0	0
40	93.8	00016F83	94083	94.083	0.301706	0	1
41	93.8	00016ECE	93902	93.902	0.108742	0	1
42	94.16	00016E92	93842	93.842	-0.33772	1	0
43	93.45	00016ED2	93906	93.906	0.487961	1	1
44	92.93	00016B15	92949	92.949	0.020445	0	0
45	94.34	0017018	94232	94.232	-0.11448	1	1
46	94.16	0016FB1	94129	94.129	-0.03292	1	1
47	93.8	00016CE0	93408	93.408	-0.41791	0	0
48	95.23	000173F7	95223	95.223	-0.00735	1	0
49	94.69	00017479	95353	95.353	0.70018	0	1
50	94.69	0001735A	95066	95.066	0.397085	0	0
51	95.69	000178A4	96420	96.42	0.76288	1	1
52	94.87	00173A8	95144	95.144	0.288816	0	1
53	94.87	000172A1	94881	94.881	0.011595	0	0
54	95.23	0017566	95590	95.59	0.378032	1	1
55	94.87	00171A6	94630	94.63	-0.25298	1	1
56	94.16	00016FD0	94160	94.16	0	0	0
57	95.23	00017542	95554	95.554	0.340229	1	1
58	94.69	000170B4	94388	94.388	-0.31894	0	0
59	95.42	00017620	95776	95.776	0.373087	1	1
60	94.16	001705B	94299	94.299	0.147621	0	0
61	95.23	00171C1	94657	94.657	-0.6017	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.

Table 5.11: RO frequencies, counter values and signatures (from 32nd bit to 61st bit) for 128 RO PUF on FPGA 1 at 0.7 V and room temperature

RO number	Counter values in hex			Counter values in decimal			Signatures		
	25C	50C	70C	25C	50C	70C	25C	50C	70C
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
18	0000DE45	000119EE	00010C3B	56901	72174	68667	0	0	0
19	0000DF17	00019401	00010CA3	57111	103425	68771	1	1	1
20	0000D1FC	00018042	000109B2	53756	98370	68018	0	0	0
21	0000EC02	00019207	00010A49	60418	102919	68169	1	1	1
22	0000C98F	00018602	00010763	51599	99842	67427	0	1	0
23	0000E9F0	00011A10	00010897	59888	72208	67735	0	1	0
24	0000ED2C	000115B6	00010BAB	60716	71094	68523	1	0	1
25	0000EC18	000118B1	00010A74	60440	71857	68212	1	1	0
26	0000D80D	0001188E	00010ADA	55309	71822	68314	0	0	1
27	0000EA33	000182BB	0001088F	59955	99003	67727	0	0	0
28	0000EC85	000192E7	00010ABE	60549	103143	68286	1	1	1
29	0000EC22	00011A5B	00010A75	60450	72283	68213	1	1	1
30	0000EB6E	000111DF	000109F4	60270	70111	68084	0	0	1
31	0000EB78	000116C6	00010957	60280	71366	67927	0	0	0
32	0000ED92	00011735	00010C28	60818	71477	68648	1	1	1
33	0000D283	00011588	000109B1	53891	71048	68017	1	0	1
34	0000D245	00011BC8	0001093F	53829	72648	67903	0	1	0
35	0000D5F5	00011B66	00010A71	54773	72550	68209	0	1	0
36	0000DF32	00011803	00010C90	57138	71683	68752	0	1	1
37	0000ED67	00011550	00010BB3	60775	70992	68531	1	0	1
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.

Table 5.12: Signatures (from 18th bit to 37th bit) for 128 RO PUF on FPGA 2 at 0.7 V for different temperatures. Red numbers show bit flip.

RO number	RO frequencies (MHz)			Signatures from RO frequencies		
	25C	50C	70C	25C	50C	70C
.	.	.	.	.	.	.
.	.	.	.	.	.	.
7	61.92	76.923	75.75	1	1	1
8	60.976	75.758	74.62	0	0	0
9	62.305	76.923	75.75	0	1	0
10	63.091	75.75	76.62	1	0	1
11	61.728	76.336	75.18	0	0	0
12	62.112	76.923	75.75	1	0	0
13	61.728	77.22	76.04	0	1	1
14	61.92	76.62	75.47	1	1	1
15	61.538	76.08	75.18	0	1	0
16	61.538	76.046	75.18	0	0	0
17	62.112	76.923	75.75	1	1	1
18	60.79	75.472	74.62	0	0	0
19	61.35	76.336	75.18	0	0	0
20	63.091	78.125	76.92	1	1	1
21	62.305	77.22	76.33	1	1	1
22	61.35	76.336	75.18	0	0	0
23	61.728	76.336	75.47	1	1	1
24	61.162	76.046	74.9	0	0	0
25	61.162	76.336	74.9	0	0	0
26	62.305	77.22	76.33	1	1	1
27	61.538	75.75	74.9	1	1	1
.	.	.	.	.	.	.
.	.	.	.	.	.	.

Table 5.13:Signatures (from 7th bit to 27th bit) obtained from RO frequencies as observed on oscilloscope for 128 RO PUF on FPGA 2 at 0.7 V for different temperatures. Red numbers show bit flip.

RO number	Counter Values			Signatures from counter values		
	25C	50C	70C	25C	50C	70C
.	.	.	.	.	.	.
.	.	.	.	.	.	.
7	61681	76845	75462	1	1	1
8	60822	75631	74434	0	0	0
9	62474	76931	75746	0	1	0
10	62899	75657	76402	1	0	1
11	60853	76347	75157	0	0	0
12	62150	76941	75689	0	0	0
13	62233	77140	75823	1	1	1
14	61741	76451	75379	1	1	1
15	61066	76142	74986	0	0	0
16	61122	76206	75014	0	0	0
17	61565	76834	75711	1	1	1
18	60677	75454	74455	0	0	0
19	61706	76283	75278	0	0	0
20	62390	78232	76870	0	1	1
21	62937	77404	76300	1	1	1
22	61561	76372	75330	0	0	1
23	61778	76381	75296	1	1	1
24	60544	75988	74787	0	0	0
25	61588	76070	74818	0	0	0
26	62595	77424	76229	1	1	1
27	61326	75849	74694	0	1	0
.	.	.	.	.	.	.
.	.	.	.	.	.	.

Table 5.14: Signatures (from 7th bit to 27th bit) obtained from counter values for 128 RO PUF on FPGA 2 at 0.7 V for different temperatures. Red numbers show bit flip.

Once the counter and ROs are validated for the new design, counter values are collected to analyze the stability of RO-PUF. The counter values are processed to obtain stability plot at different voltages (Fig 5.23, Fig 5.24 and Fig 5.25). As compared to Fig 5.12, it is observed that number of unstable bits is significantly less (Fig 5.25) for the new design. Further, it is observed that counter values obtained for 128 RO design are garbled at 0.7V, unlike 32 RO PUF configuration (Fig 5.10). Thus it can be said that one of the factors for unreliability of 128 RO configuration is high power consumption.

It was further tried to characterize the ROs frequencies at both high and low voltages with the help of frequency binning (Figure 5.27 and Figure 5.28). It can be seen that RO frequencies at both voltages show almost a similar trend.

Finally, normalized difference in frequency distribution of ROs at 1.2V and 0.7V were generated and plotted in Figure 5.29. Unlike the sharp difference between nominal and subthreshold graphs of similar data in ASIC, there is not much difference between the two graphs at 1.2V and 0.7V in FPGA.

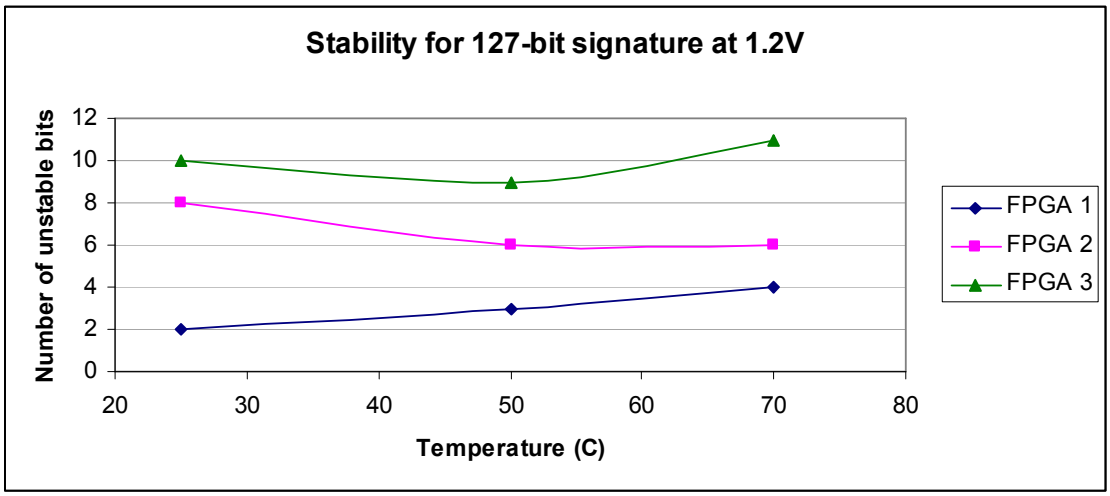

Figure 5.23: Unstable bits Vs Temperature for PUF with 128 ROs at 1.2V

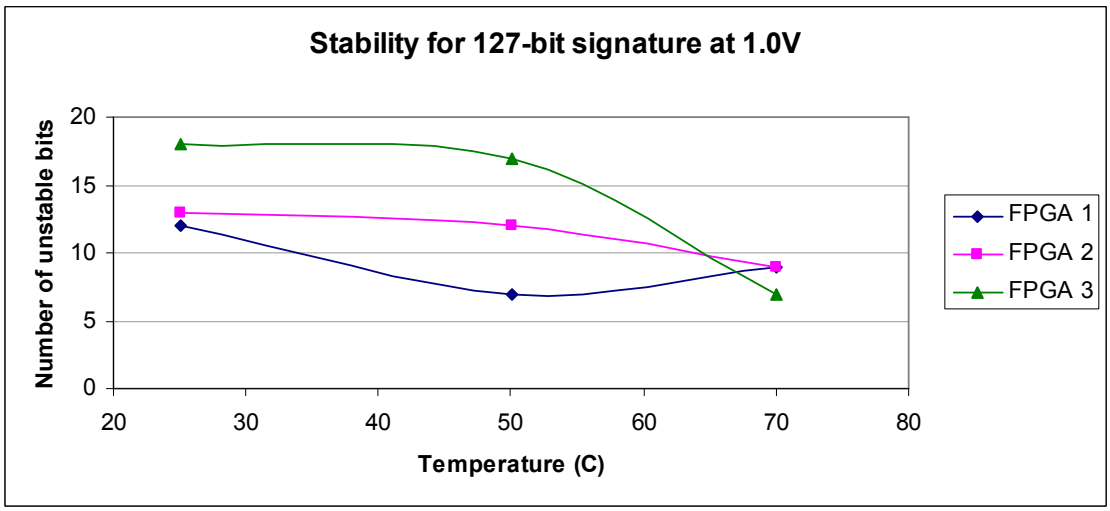

Figure 5.24: Unstable bits Vs Temperature for PUF with 128 ROs at 1.0V

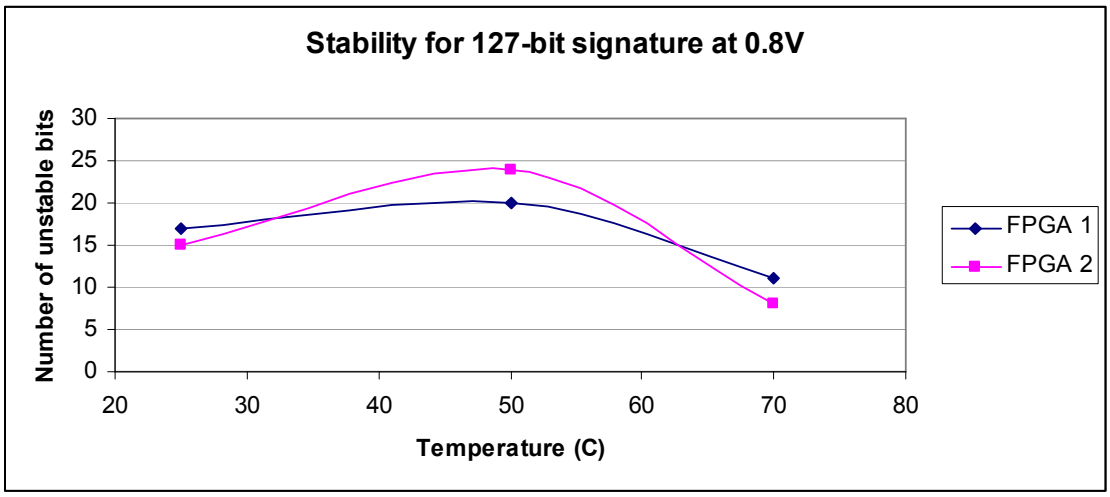

Figure 5.25: Unstable bits Vs Temperature for PUF with 128 ROs at 0.8V

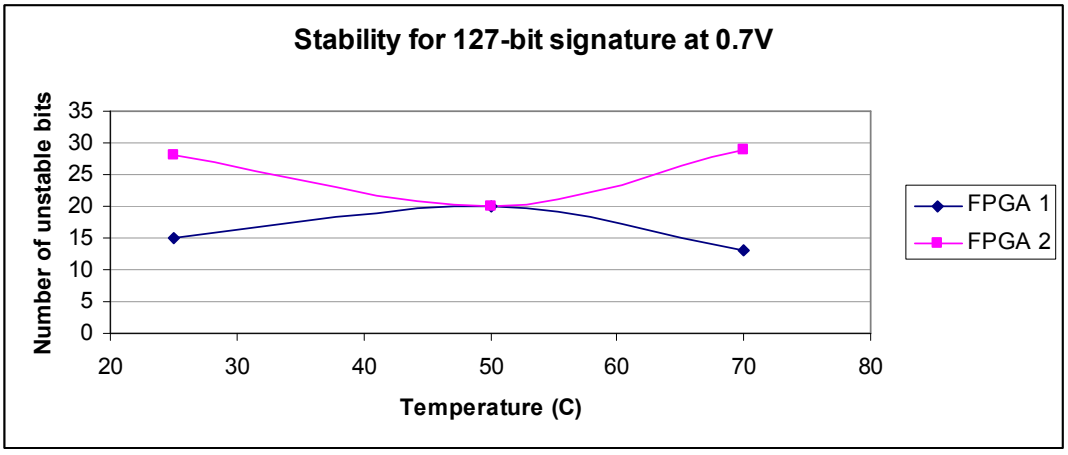

Figure 5.26: Unstable bits Vs Temperature for PUF with 128 ROs at 0.7V

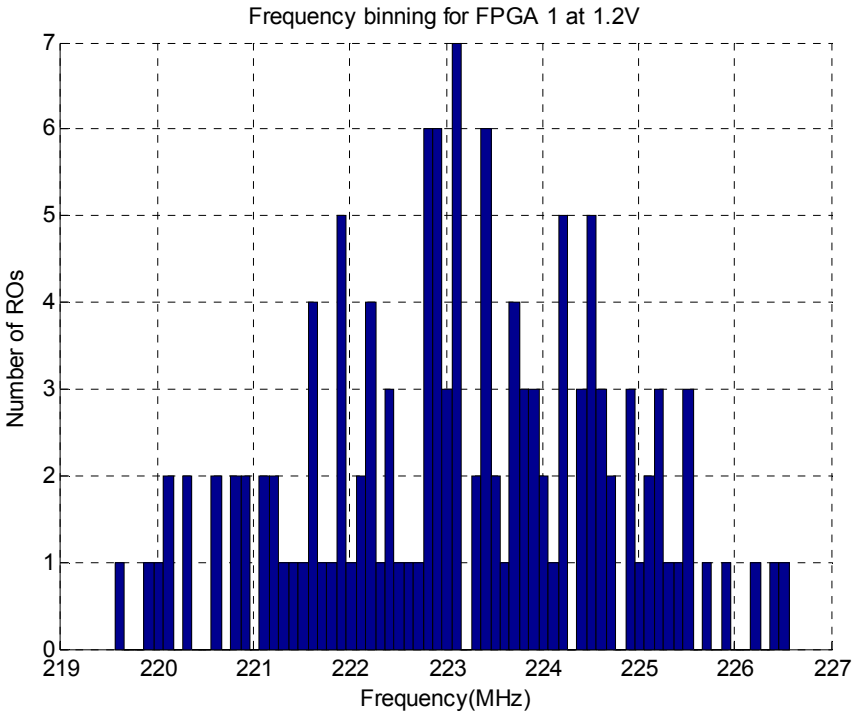

Figure 5.27: Frequency binning for FPGA 1 at 1.2V and room temperature

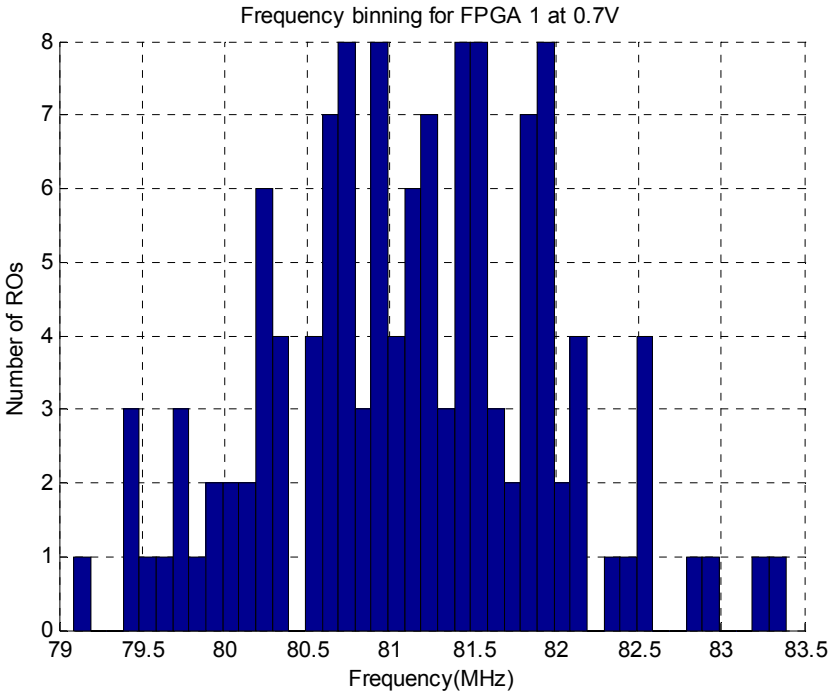

Figure 5.28: Frequency binning for FPGA 1 at 0.7V and room temperature

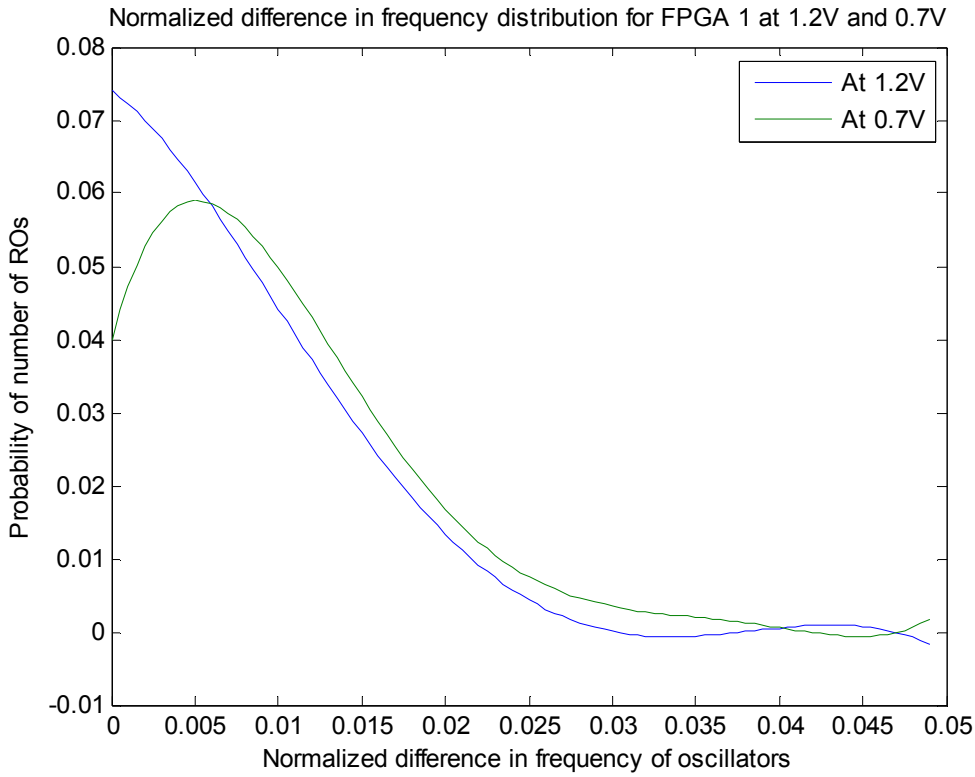

Figure 5.29: Normalized frequency difference distribution on FPGA 1 at 1.2V and 0.7V at room temperature

5.5 Conclusion

This chapter explores the effectiveness of RO-PUF on FPGA when subjected to variations such as, temperature, with FPGA chip subjected to low voltage. Results indicate that the spread of frequencies of ROs increases at low voltage. However, it is difficult to tap into this advantage as experiment results show that FPGA output becomes unstable for low voltages because of fluctuation in counter values and crossover of RO frequencies. As voltage is lowered, the error between the count of ring oscillator and counter values increase. Further more RO frequencies crossover at low voltage. It is observed that RO frequencies are the major reason for instability. We can say that the present commercial FPGA do not generate a stable PUF response at low voltage. Thus a future explanation to this project for further experiments, a low voltage FGPA is required for proper functioning of counter and ROs, thereby establishing the stability of PUF.

6 Conclusion

RO-PUFs are chip identifiers, which generate unique response, when stimulated by a challenge. The response is characteristic to the device and is based on process variations. This thesis investigated the implementation of RO-PUF on FPGA at low voltages. As ASIC implementation of PUF at subthreshold voltage formed the motivation for this work, we have explored the same aspect for FPGA.

At first, we have given a background of RO-PUF, focusing on the key metrics related to PUF evaluation. In next chapter, an architectural detail of FPGA is studied to explain the experiment results with better insight. In the experiment results, it is shown that the variability of RO-PUF at low voltage increases as compared to nominal voltage. However, unlike RO-PUF on ASIC, it is apparent from the results, that RO-PUF is more stable at nominal voltage on a FPGA. Main causes of instability of RO-PUF are greater number of RO frequency crossovers and fluctuation in counter values at low voltage. It is seen that the low and nominal voltage operation show almost the same trend when the RO frequencies are characterized at the two voltages.

Thus we can conclude from the experiment results and previous chapters that at low voltage, though the variations are increased as expected, the FPGA becomes unstable to provide consistent response. Various FPGA architectural components are affected which results in inconsistent responses. This is why researches are being conducted, where designers have tried to optimize FPGA architecture for low voltage operation. Further, low voltage operation, being a conventional method to reduce power consumption, could make FPGA important for low power applications.

References

[1] N. Beckmann and M. Potkonjak, "Hardware-Based Public-Key Cryptography with Public Physically Unclonable Functions," *Lecture Notes in Computer Science*, vol. 5806/2009, 2009, pp. 206-220.

[2] S. Morozov, A. Maiti, and P. Schaumont, "An Analysis of Delay Based PUF Implementations on FPGA," *Lecture Notes in Computer Science*, vol. 5992, 2010, pp. 382-387.

[3] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith, "Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data," *SIAM Journal on Computing*, vol. 38, 2008, p. 97.

[4] V. Vivekraja, "Low-Power , Stable and Secure On-Chip Identifiers Design," 2010.

[5] V. Vivekraja and L. Nazhandali, "Circuit-level techniques for reliable Physically Uncloneable Functions," *2009 IEEE International Workshop on Hardware-Oriented Security and Trust*, 2009, pp. 30-35.

[6] N. Drego, A. Chandrakasan, and D. Boning, "All-Digital Circuits for Measurement of Spatial Variation in Digital Circuits," *IEEE Journal of Solid-State Circuits*, vol. 45, Mar. 2010, pp. 640-651.

[7] G.E. Suh and S. Devadas, "Physical Unclonable Functions for Device Authentication and Secret Key Generation," *2007 44th ACM/IEEE Design Automation Conference*, Jun. 2007, pp. 9-14.

[8] E. Boemo and S. López-buedo, "Thermal Monitoring on FPGAs Using Ring-Oscillators," *Lecture Notes in Computer Science*, vol. 97, 1997, pp. 69-78.

[9] Wikipedia, "Ring Oscillator."

[10] A. Maiti and P. Schaumont, "Improved Ring Oscillator PUF: An FPGA-friendly Secure Primitive," *Journal of Cryptology*, vol. 24, Oct. 2010, pp. 375-397.

[11] I. Kuon, R. Tessier, and J. Rose, "FPGA Architecture: Survey and Challenges," *Foundations and Trends® in Electronic Design Automation*, vol. 2, 2007, pp. 135-253.

[12] Y. Lin and J. Cong, "Power modeling and characteristics of field programmable gate arrays," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, Nov. 2005, pp. 1712-1724.

[13] J. Cong, H. Huang, and X. Yuan, “Technology mapping and architecture evalution for k/m -macrocell-based FPGAs,” *ACM Transactions on Design Automation of Electronic Systems*, vol. 10, Jan. 2005, pp. 3-23.

[14] S. Brown and J. Rose, “FPGA and CPLD architectures: a tutorial,” *IEEE Design & Test of Computers*, vol. 13, 1996, pp. 42-57.

[15] W. Tsu, K. Macy, A. Joshi, R. Huang, N. Walker, T. Tung, O. Rowhani, V. George, J. Wawrzynek, and A. DeHon, “HSRA : High-Speed , Hierarchical Synchronous Reconfigurable Array,” *Proceedings: ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, 1999, p. 125–134.

[16] V. Betz, J. Rose, and A. Marquardt, *Architecture and CAD for Deep- Submicron FPGAs*, Springer, 1999.

[17] V. Betz and J. Rose, “FPGA Routing Architecture : Segmentation and Buffering to Optimize Speed and Density,” *Proceeding: ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, 1999, p. 140–149.

[18] Wikipedia, “MOSFET.”

[19] Wikipedia, “Subthreshold Conduction.”

[20] J.F. Ryan and B.H. Calhoun, “A Sub-Threshold FPGA with Low-Swing Dual- V DD Interconnect in 90nm CMOS,” *Custom Integrated Circuits Conference (CICC)*, 2010, pp. 2-5.

[21] B.H. Calhoun, J.F. Ryan, S. Khanna, M. Putic, and J. Lach, “Flexible Circuits and Architectures for Ultralow Power,” *Proceedings of the IEEE*, vol. 98, Feb. 2010, pp. 267-282.

[22] E. Kusse and J. Rabaey, “Low-energy embedded FPGA structures,” *Proceedings. 1998 International Symposium on Low Power Electronics and Design (IEEE Cat. No.98TH8379)*, pp. 155-160.

[23] V. George and J. Rabaey, “The design of a low energy FPGA,” *Proceedings. 1999 International Symposium on Low Power Electronics and Design (Cat. No.99TH8477)*, pp. 188-193.

[24] N. Azizi and F.N. Najm, “Look-up table leakage reduction for FPGAs,” *Proceedings of the IEEE 2005 Custom Integrated Circuits Conference, 2005.*, pp. 186-189.

[25] J.H. Anderson and F.N. Najm, “A novel low-power FPGA routing switch,” *Proceedings of the IEEE 2004 Custom Integrated Circuits Conference (IEEE Cat. No.04CH37571)*, vol. 2, pp. 719-722.

[26] F. Li, D. Chen, L. He, and J. Cong, “Architecture evaluation for power-efficient FPGAs,” *Proceedings of the 2003 ACM/SIGDA eleventh international symposium on Field programmable gate arrays - FPGA ’03*, 2003, p. 175.

[27] Xilinx Ltd, “Xilinx Constraints Guide,” 2002.

[28] Xilinx Ltd, “FPGA Editor - How do I create a hard macro?”

[29] K. (Xilinx L. Chapman, “Frequency Counter for Spartan-3E Starter Kit (with test oscillators),” 2006, pp. 1-11.

[30] K. (Xilinx L. Chapman, “UART Transmitter and Receiver Macros,” 2003, pp. 1-13.

[31] Xilinx Ltd, “Spartan-3E Starter Kit Board User Guide,” vol. 230, 2006.

[32] A. Maiti, J. Casarona, L. Mchale, and P. Schaumont, “A Large Scale Characterization of RO-PUF,” *IEEE International Symposium on Hardware-Oriented Security and Trust (HOST 2010)*, 2010, pp. 66-71.

[33] X. Ltd, “Spartan-3E FPGA Family : Datasheet,” vol. 312, 2009, pp. 1-233.

[34] N.H.E. Weste and D. Harris, *CMOS VLSI Design A Circuits and Systems Perspective*, Addison Wesley, 2010.

[35] R.J. Baker, H.W. Li, and D.E. Boyce, *CMOS Circuit Design, Layout and Simulation*, Wiley IEEE - Press, 2010.

[36] K. Shakeri and J.D. Meindl, “Temperature variable supply voltage for power reduction,” *Proceedings IEEE Computer Society Annual Symposium on VLSI. New Paradigms for VLSI Systems Design. ISVLSI 2002*, 2011, pp. 71-74.