Study of Monitoring Circuitry for Ageing in FPGAs

PENGXIANG CHENG

MASTER´S THESIS

DEPARTMENT OF ELECTRICAL AND INFORMATION TECHNOLOGY
FACULTY OF ENGINEERING | LTH | LUND UNIVERSITY

Study of Monitoring Circuitry for Ageing in FPGAs

Pengxiang Cheng
pe8244ch-s@student.lu.se

Department of Electrical and Information Technology
Lund University

Supervisor: Erik Larsson, LTH
erik.larsson@eit.lth.se

Supervisor: Farrokh Ghani Zadegan, Ericsson

Examiner: Pietro Andreani, LTH
pietro.andreani@eit.lth.se

October 20, 2021

© 2021
Printed in Sweden
Tryckeriet i E-huset, Lund

Abstract

Along with the down-scaling of CMOS technology, ageing has become one of the most important reliability challenges in CMOS devices. Ageing is defined as degradation in certain device characteristics such as delay, which can result in failure. Field Programmable Gate Arrays (FPGAs) are typically the first among the CMOS devices to adopt the latest technology. It is, therefore, crucial to tackle ageing in FPGAs. As design-time-only techniques might prove insufficient in providing enough margins for future CMOS technology nodes, it becomes important to also monitor for the ageing degree to ensure the correct functionality.

This thesis aims to review previous work to understand ageing and find effective ageing monitoring methods for FPGAs, in terms of the ability to detect degradation of the fabric resources with high accuracy and high precision. A comprehensive survey of previous methods is conducted, reporting a comparison of monitors in detail and comparing their pros and cons. Based on the survey, we perceived the process/performance variation mapping method by use of ring-oscillators (the so-called PV mapping) to have great potential and enhanced the existing PV mapping method by (1) introducing sensors based on new ring-oscillator types that cover significantly more hardware resources, and thus enhance the monitoring coverage, (2) pushing the number of uniform sensors (each sensor being comprised of a ring oscillator and a frequency counter) by the use of carefully developed placement constraints to almost 80% of the maximum theoretically possible number of sensors of the suggested type, and (3) designing extra ring-oscillator types and circuitry for gaining more insights into the precision and coverage of the proposed performance variation (PV) mapping method. The PV mapping method was applied to 20 Digilent Nexys4 boards, featuring a 28nm XILINX ARTIX 7 XC7A100T FPGAs to validate that the proposed method is capable of detecting delay differences among uniformly shaped sensors on the same device, and for each sensor among multiple boards. In addition, the precision and accuracy of the proposed method are reported.

Acknowledgments

Firstly, I would like to appreciate my supervisors Farrokh Ghani Zadegan at Ericsson Lund and Erik Larsson at Lund University, for their patient and tireless help. Especially, Farrokh Ghani Zadegan has guided me to solve techniques problems and given me continuous help during our thesis work. Erik Larsson, I would like to thank him for providing constructive advice. Moreover, thank you for proofreading all my drafts.

Secondly, I also want to thank my line manager, Mikael Lostedt at Ericsson Lund for providing me this opportunity and for his supports.

Thirdly, I am grateful to my friends, Zilin Zhang and Yu Zhu for their practical advice in my debugging time and their cheering through the tough times. Additionally, a great thanks to my friend, Marcus Sandberg for his contribution to our IC project when I am busy doing the thesis.

Finally, I would like to express my gratitude to my family for their understanding and support.

Popular Science Summary

We use electronic systems everyday as they are present in computers, phones, home appliances, cars, etc. Electronic systems are made of small components called chips, and the chips are made of up to billions of nano-scale components called transistors, which are connected with nano-scale wires. The transistors and wires inside a chip are subject to a phenomenon called ageing, which results in the degradation of their functionality. For example, an aged chip cannot perform as fast as when it was new, resulting in malfunctioning. Up until now, to avoid such degradation due to ageing, the systems designers added a lot of performance margin so that even if the chip is affected by ageing, it could still continue working with the intended performance for the intended life-cycle. However, since in the future, due to the need for more performance and lower power consumption, more and smaller transistors should be used. Designing with such extra margins for future technology is going to be too costly. One solution, would be to monitor the chips for ageing and take action, such as replacement, before the chip malfunctions. For monitoring, typically, designers need to integrate monitoring sensors with the circuits inside the chips, which takes some design effort and also requires resources in terms of transistors and wires.

This thesis work looks into a class of chips called FPGAs. FPGAs are programmable, meaning that after they are manufactured, one can program their internal circuitry many times to carry out different functions. In this thesis work, we take advantage of this programmability to perform monitoring for ageing. The advantage with this method is that the designers can save both design efforts and programmable resources by avoiding integrating the monitoring sensors directly into their designs, and instead design one program just for monitoring.

We place a certain number of sensors on the FPGA board, and the frequency reported by the sensor is used as the indicator for ageing. If the sensor frequency is lower than the minimum acceptable value, the chip is judged to have aged too much. We will periodically measure and collect the results of the sensors to monitor aging by observing whether they are close to the minimum acceptable value. We have performed extensive experiments with our monitoring method, and have shown that with our method it is possible to detect subtle frequency differences both between sensors inside the same FPGA, as well as between the same sensor location on different FPGAs. Based on this, we expect that this method will also be capable of detecting ageing (indicated by gradual changes in the frequencies reported by the sensors) and predicting malfunctions when sensor data is collected periodically throughout the lifetime of an FPGA.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ALU	Arithmetic Logic Unit
BTI	Bias Temperature Instability
CLB	Configurable Logic Block
CMOS	Complementary Metal Oxide Semiconductor
CP	Critical Path
DCM	Digital Clock Manager
DSP	Digital Signal Processing
EM	Electro-Migration
FF	Flip-Flop
FPGA	Field-Programmable Gate Array
HCI	Hot Carrier Injection
IOB	I/O block
LE	Logic Element
LFSR	Linear-Feedback Shift-Register
NBTI	Negative Bias Temperature Instability
PBTI	Positive Bias Temperature Instability
PLL	Phase-Locked Loop
PV	Performance Variation
PV	Process Variation
RCP	Representative Critical Path
RNS	Residue Number System
RO	Ring Oscillator

SM	Stress Migration
SR	Shadow-Register
SRL	Shift Register LUT
TCL	Tool Command Language
TDDB	Time Dependent Dielectric Breakdown
TP	Transition Probability
XADC	On-chip Analog-to-Digital Converter

Chapter 1
Introduction

1.1 Motivation

As scaling of Complementary Metal Oxide Semiconductor (CMOS) devices continues, billions of transistors can be integrated on a single die. Such a high level of integration is the enabler for meeting the requirements for performance. Unfortunately, such higher integration comes at the expense of several manufacturing and reliability issues such as process variation, soft errors, thermal challenges, and ageing. One of the most important reliability challenges at nano-scale CMOS technology is transistor ageing [5]. Transistor ageing refers to phenomena such as Bias Temperature Instability (BTI), Hot Carrier Injection (HCI), and Time Dependent Dielectric Breakdown (TDDB). Besides, interconnect ageing cannot be ignored, which includes Stress Migration (SM) and Electro-Migration (EM).

An important family of CMOS devices is Field-Programmable Gate Arrays (FPGAs), which are semiconductor devices with a matrix of configurable blocks connected via programmable interconnects. FPGAs can be configured after manufacturing to meet functionality requirements or desired applications. FPGAs fabricated with the most advanced CMOS technologies and the highest level of integration are broadly used today because of the low power consumption and high-performance demands. FPGAs are widely deployed to accelerate computation in various applications from embedded applications to data center applications.

Due to the effect of degradation mechanisms on transistor threshold voltage, HCI and BTI seriously limit the performance of CMOS devices by increasing the transistor switching and thus the path delay. In addition, the scaling rate of transistor size is larger than that of the supply voltage. As a result, once the delay of critical paths (CPs) exceeds the clock period, timing failures occur and the correct functionality of the circuit will be affected [7]. This trend increases temperature and current density, thus accelerating device degradation in future CMOS devices. Since lifetime decreases due to an increase in path delays, it is important to address the ageing problem. So far, design-time methods such as specific design rules have been used to ensure that ageing does not affect the desired useful life of a product. However, given the increased variation (that leads to reduced design margins) and use of new materials in future process nodes, it becomes necessary to also monitor for ageing.

Ageing monitoring is achieved by implementing circuitry that is able to detect and measure the degree of degradation. In previous works, there are several mainstream methods to monitor ageing, such as an approach based on ring oscillators and frequency counters, and a technique by monitoring the correct functionality of critical paths.

As FPGAs use the most advanced CMOS technologies [8], it is crucial to address nano-scale reliability concerns for FPGAs. To ensure performance, it is important to propose an effective ageing monitoring system to address these problems.

Given the regular architecture of Field-Programmable Gate Arrays (FPGAs) and the ability to program each resource of its structure at a very low level, many works have been devoted to ageing effects and ageing monitoring on this type of device. In this thesis work, we review these works and devise a method suitable for in-field monitoring of FPGA devices, as detailed in the next section.

1.2 Goal and Objectives

This thesis work investigates the design of effective monitoring circuitry for ageing in FPGAs. It conducts a comprehensive survey of prior work on monitoring of FPGA degradation, reports the findings as well as presents a comparison of monitors with respect to resource consumption, the accuracy of measurements, and targeted degradation mechanisms. The main objective of this work is to find several effective ageing monitoring methods from previous works, in terms of the ability to detect degradation of the fabric resources with high accuracy and high precision. In particular, we find the most suitable for in-field monitoring of ageing in a wide range of products and experiment with this method. With the experiments, we aim to get more insights into the practical aspects of the implementation, as well as knowledge about the resource utilization, precision, and accuracy of the chosen measurement method.

1.3 Thesis Organization

The rest of this thesis is organized as follows. The overview of general FPGA structure and background about degradation mechanisms are presented in Chapter 2. Chapter 3 presents a survey of related work on aging monitoring. In Chapter 4, we present our implementation of the method we chose as the most suitable for monitoring ageing in FPGAs. The experimental setup and results are discussed in Chapter 5. In the end, the thesis is concluded in Chapter 6.

Chapter 2

Background

In order to monitor degradation mechanisms in FPGAs, it is crucial to have a basic understanding of the FPGA architecture and its main components, as well as degradation mechanisms in CMOS technology. Currently, SRAM-based FPGAs are the most common commercial type owing to the use of standard CMOS processing technology and its re-programmability. In this thesis, the main focus is the FPGA from Xilinx. Thus, the FPGAs from other vendors are not presented. FPGA architecture, and its main components and degradation mechanisms are described in this chapter.

2.1 FPGA Overview

A Xilinx FPGA comprises of the Configurable Logic Blocks (CLBs) implementing logic functions, which are connected together via switch matrices and interconnect wires, and surrounded by I/O blocks (IOBs). The I/Os are the bridge between the logic blocks and peripheral components. In addition, some specific applications might be integrated into modern FPGAs, such as Digital Clock Managers (DCMs), Digital Signal Processing blocks (DSPs), and On-chip analog-to-digital converters (XADC). Figure 2.1 depicts the FPGA architecture. These components are introduced in the following sections.

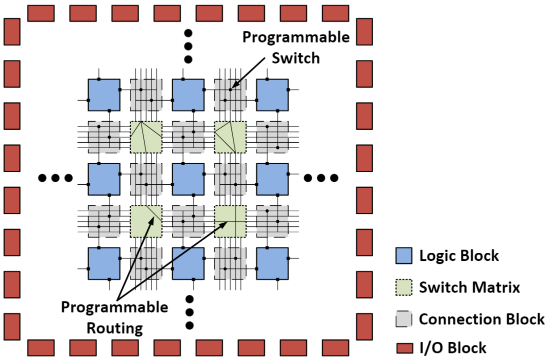


Figure 2.1: An abstracted FPGA architecture (island-style) [1], Copyright © 2013, IEEE

2.1.1 CLBs

As the main logic component of an FPGA, CLBs allow the user to implement any logic function within the chip. Each CLB contains a pair of slices[1] as shown in Figure 2.2. Each CLB consists of a set of Basic Logic Elements (BLEs), containing a Look-Up Table (LUT), a storage element and a multiplexer. Figures 2.3, and 2.4 show the basic structure of the CLB.

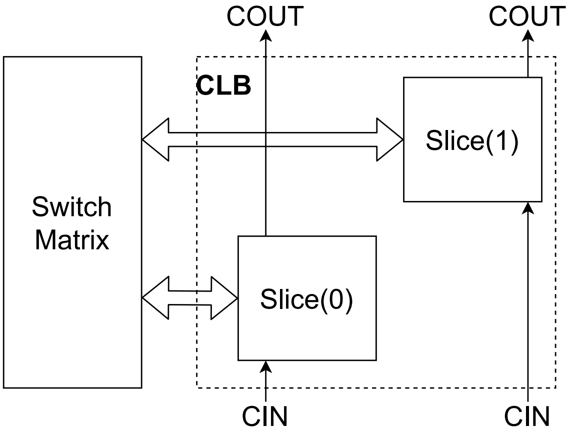
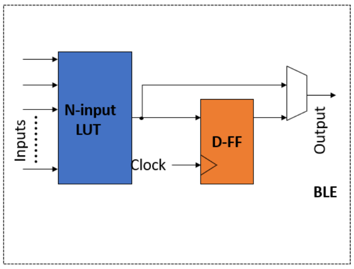


Figure 2.2: Arrangement of Slices within the CLBs [2]



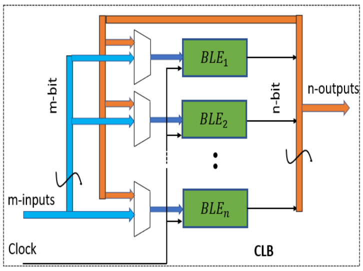


Figure 2.3: The BLE structure

Figure 2.4: The CLB structure

In order to improve the performance, several additional components are integrated into CLBs, which are carry chain and shift register. For example, in Xilinx 7 Series FPGAs, where each SLICE consists of 4 LUTs, 8 flip-flops (FFs), wide-function multiplexers, and one carry logic. There are two types of slices: SLICEM and SLICEL, where 'M' signifies a memory slice and 'L' signifies a logic slice. In addition, LUTs in SLICEMs can be implemented as distributed RAM or shift registers.

[1] This is relevant to the 7 Series, as in newer Xilinx FPGA families, the number of Slices per CLB is different.

Look-up Tables (LUTs)

The Lookup Table (LUT) is the basic logic element in CLB, which can implement Boolean functions. The truth table value of an n-input Boolean function is stored in 2^n-bit SRAM cells. An n-input LUT can calculate an arbitrary n-input Boolean function. The input pins control the multiplexers which can be implemented with pass transistors or transmission gates to pass the value to the output. For example, a general n-input LUT and a two-input LUT are shown in Figures 2.5 and 2.6. The function values are stored in SRAM bits C_0 to C_{2n-1}.

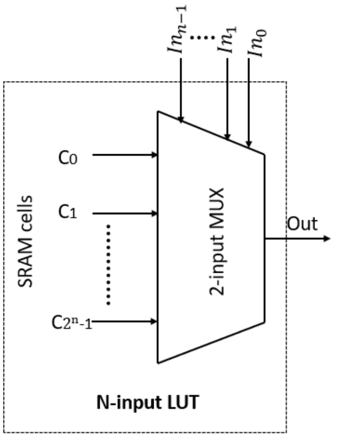


Figure 2.5: N-inputs LUT

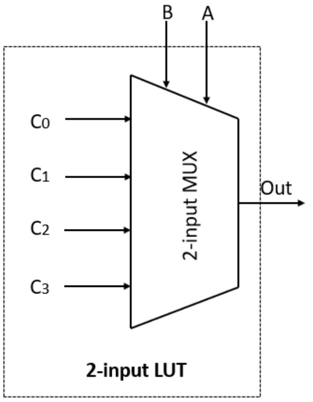


Figure 2.6: Two-input LUT

Storage Elements

Storage elements are used in sequential circuits in the FPGAs, to store the output values of LUTs. These can be configured as a level-sensitive latch or edge-triggered D-type flip-flops.

Carry Chain

The dedicated fast look-ahead carry is used to perform fast arithmetic including subtraction and addition with using fewer resources. The Carry Chain contains four MUXs and four two-input XOR gates, and its structure is shown in Figure 2.7.

Shift Registers

Shift Registers, such as 32-bit shift register (SRL32) and 16-bit shift register (SRL16), can be configured via LUTs in the SLICEM without using flip-flops available in the SLICE. The serial data can be delayed from 1 to 32 clock cycles using SRL32 that can be built to larger or smaller shift registers to satisfy the variable requirements. For example, a 29-bit shift register can be created by setting the address to the 29th bit.

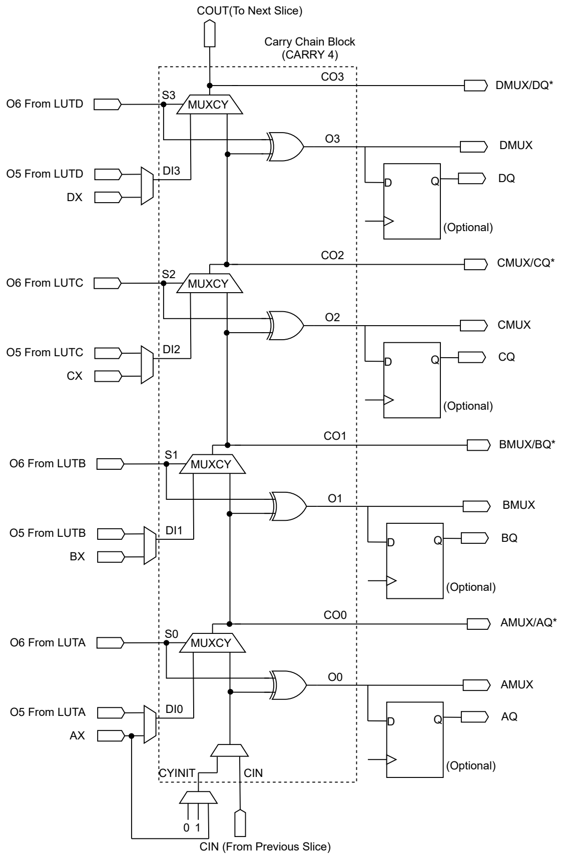


Figure 2.7: Schematic of the Carry Chain [2]

2.1.2 Switch Matrix

According to the requirements of the circuit mapped to the FPGA, the switch matrix uses a set of programmable switches to activate/deactivate the connections as needed to provide the required routing to achieve the connection between the CLBs, to and from I/O blocks.

The number of possible connections between wires in each matrix is usually too large. In order to reach a better balance among the routing flexibility, area and delay, modern FPGAs provide different-length interconnect wires.

2.1.3 Other Components

Digital Clock Managers (DCMs)

The digital clock manager (DCM) can generate multiple clock signals with different duty cycles or a wide range of frequencies. In addition, it can also be used as a phase shift synthesizer, a dynamic frequency synthesizer, and a jitter filter for the input clock. These functions can be easily implemented via a clocking wizard.

Digital Signal Processing (DSP) Applications

As an arithmetic logic unit (ALU) embedded into the fabric of the FPGA, DSP applications consist of many binary multipliers and accumulators, which are designed to perform "multiply and add" operations. It has higher power efficiency and operates at far higher frequencies than the equivalent circuits in a soft implementation.

Xilinx Analog Mixed Signal Module

Xilinx analog mixed signal module, referred to as the XADC, is a new flexible analog interface. When combined with the programmable logic function of the Xilinx Artix-7 FPGAs, it can handle various data acquisition and monitoring requirements.

2.2 Degradation Mechanisms

As mentioned earlier, degradation is one of the most important reliability challenges in CMOS technology. Ageing happens over a long time, which leads to timing failures and adversely affected the performance of CMOS devices. Ageing (also known as wear-out) effects can be observed in the transistors (i.e. BTI, HCI, TDDB) as well as the interconnects (i.e. EM, SM) inside the chip [1]. Each of these mechanisms is briefly explained in the following sections.

2.2.1 Transistor Degradation Mechanisms

Transistor degradation mechanisms cause an increase of threshold voltage, as a result, drain current, mobility, and trans-conductance of the transistor degrade. Consequently, the switching speed of the transistor decreases, and thus, the delay of the circuit's functional paths might exceed the timing requirements. Correspondingly, failures start to happen in the circuits and the operational lifetime of FPGA chips will decrease. There are three transistor degradation mechanisms named TDDB, BTI and HCI [1], which are demonstrated in Figure 2.8.

Time Dependent Dielectric Breakdown (TDDB)

TDDB is a breakdown in the gate oxide layer of a transistor. The breakdown is a result of a conductive path through the gate dielectric which is formed by trapped charges or defects that gradually accumulate over time. TDDB inflicts a rise in leakage current and power consumption and a slowdown of switching activities, and eventually, the dielectric layer will lose its insulating properties, which leads to a permanent failure [3] [1].

Bias Temperature Instability (BTI)

There are two phenomena: NBTI affecting PMOS and PBTI affecting NMOS. Compared with NBTI, the PBTI effect was negligible and ignored mostly in previous technology. However, the PBTI effect should be considered since the introduction of high-κ/ metal gates transistors in sub 45 nm technology. NBTI is a static mechanism driven by gate potential, which is caused by hydrogen ions diffusing away in the interface region. BTI has a stress phase and a recovery phase. The magnitude of threshold voltage increases in the stress phase, and on the contrary, it decreases back toward its initial value in the recovery phase. Nonetheless, this recovery cannot be complete [3] [1]. BTI is a static mechanism and is triggered when a transistor is on [7], which is mostly independent from signal duty cycle.

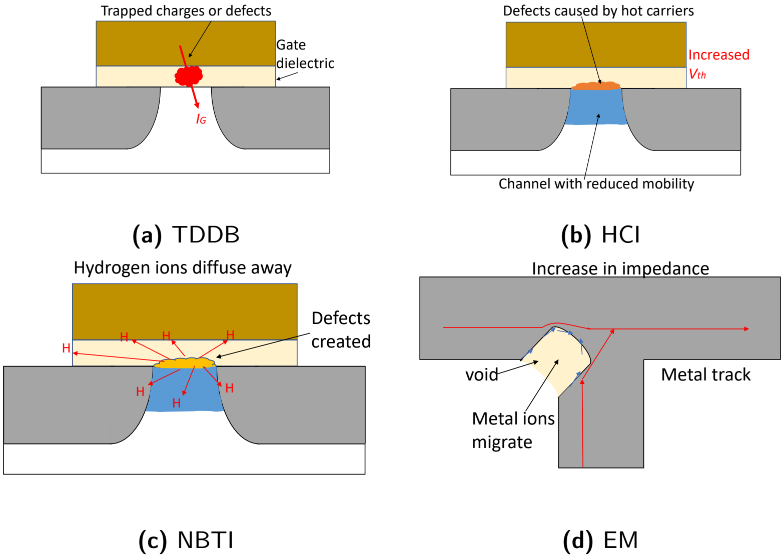
Hot Carrier Injection (HCI)

HCI, also called hot-electron effect or hot-carrier, is based on defect accumulation in the interface region between the gate dielectric and the channel. Hot carriers (electrons) in the channel cause interface defects, and some of them overcome the barrier of the gate dielectric due to obtain enough energy and are accelerated by the gate field. HCI leads to a rise of threshold voltage and a slowdown of carrier mobility, and consequently, which leads to a decrease of switching activities. HCI is primarily dynamic since the procedure is driven by a high-energy carrier flowing into the channel, and it happens when transistors switch. However, unlike the BTI, there is no recovery phase , and thus, the effect, threshold voltage increasing is permanent [3] [1].

HCI is a dynamic mechanism, which is dependent directly on switching activities (speed). HCI degradation mechanism has greater effects on the fall delay than the rise delay. TDDB is also a static mechanism.

2.2.2 Interconnects Degradation Mechanisms

There are two degradation mechanisms, Electro-Migration (EM) and the Stress Migration (SM) affecting interconnects. EM causes atoms to migrate from one side of the wire to the other side in the direction of the electrons due to current flows in wires. In SM, the atoms migrate from the high-stressed side to less-stressed areas due to the thermal differences in the wires without applying current. Consequently, EM and SM leads to voids where the atoms migrated from and creates hillocks where the atoms migrate to, and which in turn leads to short circuit failures [3]. The procedures of main degradation mechanisms are demonstrated in Figure 2.8. The features of these degradation mechanisms are presented in Table 2.1.


Trapped charges or defects
Gate
dielectric
I_G
Defects caused by hot carriers
Increased
V_{th}
Channel with reduced mobility
Hydrogen ions diffuse away
Defects
created
H
Increase in impedance
void
Metal track
Metal ions
migrate

(a) TDDB

(b) HCI

(c) NBTI

(d) EM

Figure 2.8: Degradation Mechanisms [3]

Ageing mechanisms	NBTI	HCI	TDDB	EM
Defects	Vt increase	Vt increase	Leakage increase	Impedance increase
Causes	Hydrogen ions diffuse away	Hot carriers	Defects (traps) overlap	Metal ions movement
Accelerators in lab	Elevated voltage and temperature	Elevated voltage and temperature	Elevated voltage and temperature	High DC current and high temperature
Recovery	Partial recovery	Permanent failure	Permanent failure	Permanent failure
Sensitivity	Duty cycle	Switching activities		Current in wire
Symptoms	Slower switching	Slower switching	Failure	Slower switching

Table 2.1: Comparison of degradation mechanisms

Chapter 3

Literature Survey

In this chapter, we review approaches for detecting ageing in FPGAs, and we will discuss pros and cons of each approach.

Ageing in FPGAs can be monitored by measuring the reduction of the maximum operating frequency, which indicates the worst case path delay. There are three kinds of sensors to monitor ageing. The most common ageing sensor is based on arrays of ring oscillators (ROs), which will be referred to as RO sensor in the rest of this thesis report. The sencond kind is based on the Razor approach using shadow registers (SR) placed at the end of combinational paths [4], which is referred to as the SR sensor in this work. The third kind is based on detecting changes in transition probability (TP) [6], referred to as TP sensor. The RO sensor monitors ageing by comparing the RO's frequencies measured at different times. SR and TP sensors monitor the ageing by detecting timing failures. These three types of sensors are described in the following sections.

3.1 RO Sensor

RO sensor is composed of an RO and a frequency counter, an traditional RO sensors structure is depicted in Figure 3.1. The frequency counter is used to capture the RO's frequency that reflects the FPGA's performance and capture the frequency change due to aging degradation.

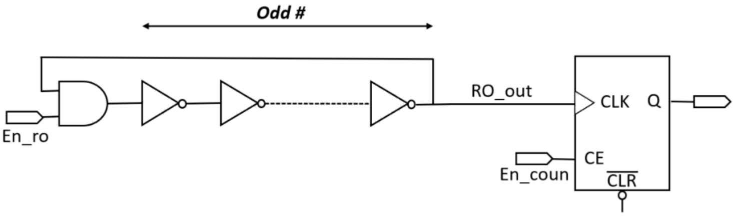


Figure 3.1: Traditional Ring Oscillator Structure

In an RO, traditionally an odd number of inverter gates are connected in a loop to form a ring. The RO output toggles between zero and one since the number of inverters is odd. Consequently, a square wave signal is generated at the output.The frequency of RO is given by Equation 3.1:

$$f = \frac{1}{2 \times n \times t} \tag{3.1}$$

where, f represents the frequency of oscillation, t is the time delay for a single inverter, and n is the number of inverters in the RO. In general, in order to control the RO's oscillations, control logic (e.g., And gate) is employed to build RO. As illustrated in Figure 3.1, the RO is controlled by control logic (e.g., AND gate), which could reduce the effect of self-heating or counter overflow [9]. In [10], the authors set T_s to be $50\mu s$ in order to avoid self-heating phenomena due to longer measurement time.

The frequency counter is used to capture the RO's oscillations for a predefined period of time, called sample period. The oscillation period, T_d of an RO is given by Equation 3.2:

$$T_d = \frac{T_s}{N} \tag{3.2}$$

where T_s is the sample period, N represents the number of cycles obtained by frequency counter.

ROs have been extensively studied in the past, not only to analyze process variations (Process variation is the natural variation in transistor properties (length, width, oxide thickness) when manufacturing integrated circuits.), but also to analyze ageing effects. In addition, there are a large number of ROs with different structures designed to satisfy different study purposes. ROs with different structures mean that sensors have different LUT configurations, different number of LUTs in each RO and different types of and frequency counter.

In next section, process variation analysis based on RO sensors is presented.

3.1.1 Performance Variation Analysis

Performance variation is a measure of both process variation and the degradation induced by ageing. In [10], [11] and [12], ROs were implemented to characterize process variations, which is decomposed into stochastic and systematic intra-die process variability, in 16nm Zynq XCZU7EV FPGAs, 90nm Cyclone II, and 65nm Virtex-5, respectively. In addition, counters were used to measure the frequency of ring oscillators. Moreover, in order to guarantee the resolution, an array of ROs were placed on the die. In [10], performance variation in logic and interconnect resources was characterized and the impact of diverse temperatures and voltages is analyzed. The experimental results showed that, under certain operating conditions, the intra-die and inter-die performance variation can reach up to 9.9 % and 12%, respectively. In addition, the authors concluded that there is a low correlation between logic and interconnect variations. However, in [11] and [12], the authors conducted the experiments without temperature control and did not study the effect of ambient temperature. In [12], all CLBs are used to map 6,480 ROs, which are split into multiple configuration bit-streams to measure. Thus, this test has higher resolution and coverage than the experiments in [10] and [11]. It can also be noticed that the measurement system's impact on measurement accuracy is not reported in these works.

In [13], the authors employed ROs to measure intra-die variation in 65nm Virtex-5 FPGAs. A novel and highly-efficient counter, residue number system (RNS) ring counter using SRLs was utilized, which is more resource-efficient compared with other types of counters such as binary counter and linear feedback shift register (LFSR) using SRLs counter. These three counters will be described in detail in Section 3.1.4. In [13], RO required just eight LUTs, composed of seven inverters and one AND gate to control the oscillation. In addition, four physical parameters were analyzed, namely, variation in delay, static and dynamic power and temperature.

In [14], an array of different sizes of ROs are implemented on 90nm Spartan FPGA. In order to analyze LE characterization and derive the differences in LE delay, the authors built two sizes of ROs, namely an eight-stage RO using all LEs in a CLB and a seven-stage RO utilizing seven LEs. In addition, the authors create a calibration RO utilizing two LEs and a pair of interconnects (e.g., Direct connection, Double, and Hex wires) to analyze interconnect delay characterization. The authors accurately characterize process variation of logic elements (LEs) and interconnects, but the impact of the counter is not analyzed.

In [15], a matrix of ROs was implemented to perform process characterization in 90 nm Spartan-3 FPGA. A different measurement method based on Electro-Magnetic Analysis (EMA) was implemented to measure the frequency from the ROs. It should be noted that the experiments were conducted with complete control, temperature and voltage relatively at constant by using a thermal chamber and a core voltage control. Thus, the differences between measured frequencies is only due to process variation. Compared with the approaches in [10], [11], [12] and [13], this approach has higher accuracy. However, this method makes the experiment more complicated since the lab equipment is used (eg., EM probe and oscilloscope) making this method impractical for in-field monitoring.

3.1.2 Ageing Effect Analysis

Previous work has employed ROs to characterize the effects of degradation on FPGAs in 45nm Spartan 6 [8], 65nm Altera Cyclone III [16], 65nm Altera Cyclone III [17], 65nm Altera Cyclone III [18] and 28nm ARTIX-7 FPGAs [19], respectively. In these papers, different stress conditions were tested to accelerate the ageing process and identify the mechanisms.

In [8], the authors employed a group of controlled ROs with adjustable switching activity and different frequencies. The ageing process was emulated by stressing the FPGA with elevated voltages and temperatures. The experiment was conducted in three steps including capturing the oscillating frequencies in nominal conditions, accelerated ageing in elevated temperature and voltages, and measuring. Like [15], sensors' frequencies were captured using Electro-Magnetic (EM) analysis rather than frequency counter, and the experiment setup is also performed with complete control of voltage and temperature. The EM method is utilized to ensure that only variations caused by ageing are captured in order to monitor the changes in the sensors' performance before and after stress. The frequencies are measured with the help of Electromagnetic Sensing Probe. In [8], the authors concluded that, compared with HCI, BTI ageing is the principal factor in ageing of the used 45 nm technology node.

In [17] and [18], the authors analyzed FPGA LUT delay degradation due to NBTI and HCI under different types of electrical stress signals. The approach has two phases of operation, consisting of measurement mode and stress mode controlled by a Mode input. The sensor contains nine buffers and one inverter, its frequency is measured by a K-bit counter. In [17], a low-frequency signal with varied duty cycles was stressed into LUT. However, in [18], three kinds of frequencies signal with different duty cycles were applied as stress signals. The authors identified that NBTI and HCI degradation mechanisms depend on the duty cycles and frequency of input signals, respectively. NBTI affects the fall delay more than the rise delay, while, HCI has the opposite effect.

In [19], like approaches in [8], [16], [17] and [**?**], the author also adopts two phases of operation to analyze ageing effects. The difference is that ROs with varied LUT configurations and different sizes were employed. In addition, the impact of variable temperatures was analyzed. It shows that the NBTI ageing effect is the dominant factor of timing degradation.

3.1.3 Temperature Monitoring

In [9], to monitor the thermal distribution of FPGAs, an array of RO-based temperature sensors are implemented on Virtex-5 LX50T FPGA. It should be noted that the propagation delay of an inverter is related to its temperature, and the interconnect resistance has a linear relationship with its temperature, an increase of temperature will lead to an increase of inverter path delay and interconnect delay. Thus, the frequency of an RO is decreased as its temperature increases. In order to find the best design, the authors study various types of sensors for variant RO lengths and counters (e.g., Binary counter and RNS ring counter) with variant widths. The authors also introduce four useful criteria to evaluate the relative performance. Comparing these designs, one sensor with the highest efficiency is found to monitor the temperature distribution of FPGAs.

3.1.4 Counter

In previous work, there are three types of counters implemented. A standard binary counter counts by using a binary representation, and its results can be read without special decoding. However, it costs a relatively high number of hardware resources. An alternative design is a linear feedback shift register (LFSR) counter. Another counter is the residue number system (RNS) ring counter [13]. LFSR and RNS counters can implemented in a compact way using Xilinx's shift register LUTs (SRLs). Compared with binary counter and LFSR counter, an RNS ring counter is far more compact and needs fewer resources.

In [13], assuming the requirement for 2^{13} count cycles, the authors compared three types of counters, which is shown in Table 3.1. A binary counter can be implemented with 13 LUTs. An LFSR counter requires an SRL, a flip-flop and a LUT configured as an XOR, two LUTs and hundreds of bytes of memory [13]. However, an RNS counter can be built from 2 SRLs, on Virtex-5/6 FPGA, one acts as a 32-bit SRL, and the other acts as two 16 bits SRLs, using moduli of 33, 17, and 16.

Counter type	Logic (LUTs)	Memory (Bytes)
Binary	13	0
LFSR using SRLs	2	Hundreds
RNS ring using SRLs	2	0

Table 3.1: Comparison of counter implementations assuming a maximum period of 2^{13}

3.2 SR Sensor

In this approach, a register called shadow register[1] is placed at the end of combinational paths, which is independent of the main circuit. One type of SR sensor is shown in Figure 3.2. SR is triggered by an auxiliary clock with a negative skew from the main clock that used to trigger the destination register. Due to the existence of clock skew, the combinational outputs are latched by destination register and shadow register at different times, respectively. During the test, two clocks have the same frequency and the clock skew is changed gradually until two registers capture two different values [4]. With the help of clock skew and frequency values, the combinational path delay can be calculated. Based on the above method, this approach not only measures the path delay but also monitors the ageing phenomena.

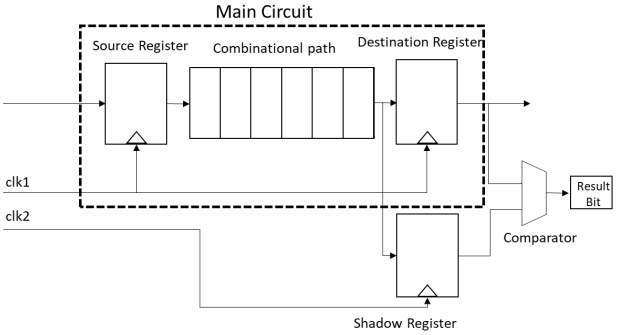


Figure 3.2: Basic principle of delay characterization circuits with shadow register [4]

3.2.1 Performance Variation Analysis

In [4], the authors analyzed the path delay by placing additional shadow registers alongside main register on 130nm Virtex-II FPGAs, respectively. Three different combinational

[1] In this work, it is referred to as shadow register.

logic paths of five 32-bit floating-point adders placed in five different locations are analyzed. In order to measure the path delay, the clock skew was increased step by step until an incorrect value was captured in the shadow register. According to the [4], addition of shadow registers have no adverse impact on the monitored paths. However, the sensor inclusion increase the path delay due to additional loads introduced.

In [20], a method named built-in self-test to accurately measure the path delay is implemented on 90 nm Virtex-4 FPGA. The maximum frequency of path can be captured by increasing the clock frequency until detecting a timing failure. In this experiment, the timing resolution can reach 1ps or lower. However, the authors just tested simple combinatorial circuits. In addition, in this method, another FPGA was required for clock generation, which makes the test setup a bit complicated. Fortunately, this method could be implemented in modern FPGAs without an additional FPGA for clock generation. This method has a limitation to only work for combinational paths.

In [21], the authors focus on the analysis of clock tree variability, a part of performance variations, by using the method [20] on 65nm Virtex-5 FPGAs. In addition, to analyze the variance of clock skew, the authors create a measurement system for detecting clock skew and measure 336 differential path circuits. The approach can be used to alleviate the variation due to clock skew from different components in the clock networks.

3.2.2 Ageing Monitoring

In [5], an ageing sensor based on shadow register was mapped on 40nm Virtex-6 FPGAs. This approach is to capture transitions happening after the rising edge of the clock cycle, and the reason behind using two flip-flops with different clock polarities. Unlike the approaches in [4], [21], and [20], the critical path output is fed to the clock inputs of two registers and the system clock is connected to the inputs of two registers. The schematic is shown in Figure 3.3. Due to the adjustable sensitivity of the ageing sensor, it not only generates a warning signal before ageing, but also detects the timing failure caused by ageing phenomena and sends an error signal. Once the warning signal is detected, effective actions can be taken to ease ageing.

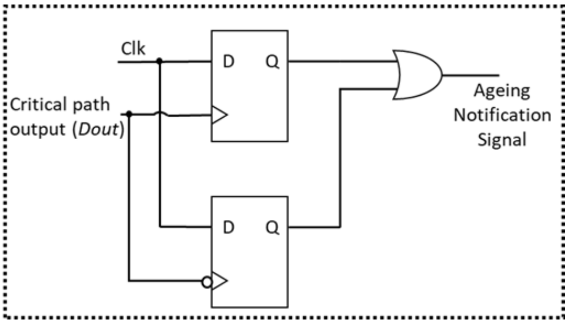


Figure 3.3: Schematic diagram of the aging sensor [5]

In [22] and [23], the authors employed a programmable ageing sensor to monitor ageing along with CPs on 90nm Spartan-3 FPGA. Compared with the method in [20], this approach sets an observation interval (T_g) as a constant clock offset between the main clock and auxiliary clock, it does not need to increase the clock offset step by step. As a result, it costs fewer resources. In addition, the sensor could be activated at some intervals, which means that it does not need to work continuously. Ageing is detected when signal transitions occur during T_g. For this kind of sensor, multiple clock generators are required for inserting various sensors since they need different T_g. It is mentioned that The sensor inclusion leads to an increase in path delay due to additional loads introduced.

Compared with the ageing sensors in [5], [22] and [23], the authors in [7], [24] and [25] , proposed a different ageing sensor for critical path delay monitoring on 40nm Virtex-6, 28nm Artix-7, and 28nm Artix-7 FPGAs, respectively. This sensor has two advantages. First, it does not need multiple clock generators for inserting multiple sensors, which could reduce clock routing complexity. Moreover, the sensor could be mapped into one slice rather than two slices, thus, it costs less area and power overheads. The authors insert ageing sensors alongside the Representative Critical Paths (RCPs) to monitor ageing. RCPs, which are paths with higher ageing rates than Critical Paths (CPs), are selected with the help of the RCP selection algorithm in [24], which can avoid unnecessary sensor insertion. Therefore, the sensors with higher accuracy placed along RCPs are able to detect ageing earlier than the ageing sensors [5] [23].

3.3 TP Sensor

Like the SR sensor, the TP sensor also measures the path delay and monitors ageing effects by detecting circuit failures. Normally, a steady TP is generated at the output node if the input is fed a source with stationary TP. TP of output node depends on input and clock frequency. It changes once the input clock frequency ramps beyond a certain threshold (maximum operating frequency), where the circuit starts to fail. Correspondingly, the path propagation delay can be measured. TP will change due to the ageing effect accumulate. Based on the above method, a sensor based on TP can be employed to monitor ageing phenomena.

Previous work has employed this circuits on 65nm Altera Cyclone III FPGAs [3] [6] and [26]. Figure 3.4 shows the general structure of the TP sensor.

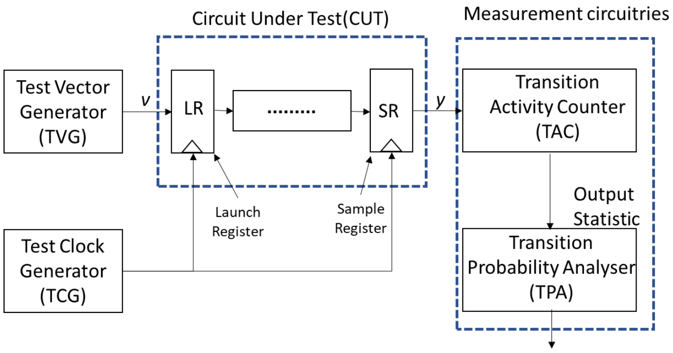

Circuit Under Test(CUT)
Measurement circuitries
Test Vector Generator (TVG)
v
LR
....
SR
y
Transition Activity Counter (TAC)
Output Statistic
Transition Probability Analyser (TPA)
Launch Register
Sample Register
Test Clock Generator (TCG)

Figure 3.4: Basic principle of the TP sensor [6]

This technique can be implemented to measure the path delay and monitor the ageing phenomena of arbitrary circuits on FPGAs. In [6], three typical circuits were tested, which are an adder carry chain, a linear-feedback shift-register (LFSR), and an embedded multiplier. In [3], 684 test circuits were tested, which consist of three different types of resources: LUTs, interconnect, and registers. In [26], the authors proposed three degradation mitigation strategies to improve longevity by replacing the aged with spare resources.

3.4 Comparison of the Three Types of Sensors

In the previous sections, we surveyed the three main sensor types used in ageing detection in FPGAs. Table 3.2 presents a summary of the reviewed methods. In this section, we will present a comparison of these three sensor types in the context of in-field ageing monitoring.

RO sensors can be utilized to monitor the development of ageing (that is, the current degree of degradation). However, our observation from the existing literature is that the RO sensors have been only used for studying the process variation and ageing effects in the lab environment, and not for in-field monitoring of ageing. Compared with the RO sensor, without sweeping the clock frequency, the TP sensor can only be used for diagnostics (that is, for detecting if malfunctioning in a circuit is due to ageing), and not for monitoring the development of ageing. The reason is that on the one hand, without sweeping the frequency, the change in the transition probability of the output node can only be observed when ageing has already increased the delay of the critical path beyond the period of the circuit's clock signal. On the other hand, sweeping the frequency when the device is in the functional mode will interrupt the operation. Moreover, the TP and SR sensors should be incorporated into the circuit and, therefore, the sensor insertion should become part of the design process. This way, the sensor insertion (which involves choosing suitable locations with regard to CPs and NCPs) might slow down the overall design process (unless the sensor insertion process is fully automated and integrated in the design flow). On the other hand, the overall resource consumption might be a limiting factor in the number of sensors inserted. In this regard, we should note that the FPGA design might be updated several times throughout the lifetime of a product (the ability to upgrade and update is one of the reasons that FPGAs are used instead of ASICs), and therefore each time any update is applied, the CP and RCPs should be re-evaluated for the SR sensor, and the TPs should be calculated for the TP sensor. Moreover, the TP sensor has the additional disadvantage that calculating the threshold is not straightforward, and even for the same exact circuit the threshold might change depending on the load and usage conditions of that product (the so called, mission profile). Additionally, adding extra registers to CP (for the TP and SR methods) might add additional capacitive load to the CP and can thus negatively impact the timing margins [23]. On other hand, the TP method has the advantage that its application is not limited to the combinational circuits, and can be used even for detecting ageing DSP blocks and larger sequential modules.

The different variants of the SR method surveyed in this chapter have been shown to have reasonably low resource overhead and performance impact. In this regard, this type of sensor is sufficiently well studied and different variants can be used depending on available resources [5]. Therefore, if the SR sensor insertion is automatically performed as part of the design process and implementation flow, the SR method can be seen as an attractive ageing monitoring solution for FPGAs.

None of the surveyed work has discussed the matter of ageing in the monitoring circuitry itself, which is something we would like to investigate more.

For this thesis work, we decided to take a different approach compared with the existing work, in the sense that instead of mixing the ageing monitors with the product RTL implemented in the FPGA as a product bit-file, we propose using a separate FPGA bit file dedicated solely to monitoring of ageing. This approach, addresses some of the abovementioned shortcomings by being non-intrusive to the design flow and independent of any circuitry that is to be implemented in the design. It should be noted that ageing is a relatively slow process and the ageing monitoring (as was also noted in some of the surveyed articles, such as [5]) does not need to be performed for every clock cycle. It is, therefore, possible to program the FPGAs in products with a special monitoring bit file every now and then (for example, as part of the power-on self-test process) and collect information on the status of ageing, and then reprogram the FPGA with the product bit file to operate in its normal operational mode. In this type of ageing monitoring, a mix of different methods can be used in order to cover as many FPGA resources as possible. For example, the DSP blocks can be tested by using the TP sensors while LUTs are tested via RO sensors. For this thesis work, we will implement one such monitoring bit file for 28nm Artix-7 FPGAs from Xilinx, and we will limit the scope of monitoring to CLBs and switch boxes, for which we will use RO sensors to create a performance variation map (PV map) of an FPGA. The main idea is to observe the trend in the frequency measured by each sensor and alert when the recorded frequency is getting close to the slowest acceptable value (which is calculated by simulating the design for the slow process corner). The knowledge from the maps can also be used in the place and route flow (a.k.a. implementation or fitting in the FPGA nomenclature) to avoid using the much-aged elements or regions of the FPGA fabric. This method, has the additional benefit that the collected data can help give more insight into the the used fabrication process in terms of process variation and ageing (which can be shared with the FPGA vendors). In the next chapter, we will detail our implementation of the proposed PV map.

Table 3.2: A summary of the reviewed methods

Sensor Type	Previous Articles	Research Target	Target Boards	Target Resources	Measurement parameter and subsystem	Pros and Cons from Authors
RO	[10]	Performance variation (Systematic and stochastic)	16nm Zynq XCZU7EV	CLBs and Interconnects	Measuring frequency by Binary counter	Pros: Avoiding potential voltage effects and RO self-heating phenomena
	[11]	Process variation (Systematic and stochastic)	90nm Altera Cyclone II	CLBs and Interconnects		Pros: Avoiding RO self-heating phenomena and accurately characterizing
	[12]	Process variation (Systematic and stochastic)	65nm Virtex-5	CLBs and Interconnects		Pros: Higher resolution and coverage
	[13]	Process variation			Measuring frequency by RNS ring counter	Pros: Improving temperature sensitivity
	[15]	Process variation	90nm Spartan-3		Measuring frequency by EMA (EM probe)	Pros: Higher accuracy and feasibility Cons: Higher complexity
	[14]	Process variation	90nm Spartan-3	CLBs and Interconnects	Measuring frequency by Binary counter	Cons: No explicitly characterizing the delay of a single wire
	[8]	Ageing effects (BTI vs HCI)	45nm Spartan-6	LUTs	Measuring frequency by EMA (EM probe)	Pros: Higher accuracy and feasibility Cons: Higher complexity
	[17]	Ageing effects (NBTI)	65nm Altera Cyclone III		Measuring frequency by Binary counter	Pros: Predicting the degradation due to NBTI
	[18]	Ageing effects (HCI)	65nm Altera Cyclone III			Cons: Not able to compute transistors ageing parameters values
	[19]	Ageing effects (NBTI vs HCI)	28nm Artix-7			
	[9]	Temperature distribution	65nm Virtex-5	LUTs	Measuring frequency by RNS ring counter	Pros: Higher sensitivity and fewer resources
TP	[3]	Monitoring degradation (NBTI)	65nm Altera Cyclone III	LUTs and Interconnects	Detecting changes in the Transition probability by Sweeping frequency	Cons: Difference between the experiment results and results under natural ageing
	[6]	Delay measurement				Pros: Measuring path delay of arbitrary circuits and highly efficient Cons: Completing the theoretical framework
	[26]	Degradation analysis (NBTI) and degradation mitigation strategies				Pros: Presenting degradation mitigation strategies
SR	[4]	Performance variation	130nm Virtex-II	Combination path	Measuring delay by sweeping clock skew	Pros: Measuring the delay of all paths and not increasing the complexity of the main circuit Cons: Resolution 160ps
	[21]	Process variation	65nm Virtex-5	Clock networks	Measuring frequency by sweeping frequency	Pros: Isolating the effects on clock skew from different components in the clock network.

Table 3.2 continued from previous page

Sensor Type	Previous Articles	Research Target	Target Boards	Target Resources	Measurement parameter and subsystem	Pros and Cons from Authors
SR	[20]	Performance variation	90nm Altera Cyclone-II	CLBs	Measuring frequency by sweeping frequency	Pros: Resolution 1ps or lower Cons: External clock generation require and limitation to only work for combinational paths.
	[5]	Aging monitoring	65nm Virtex-5	CLBs	Monitoring late transition	Pros: Detecting transistor aging and erroneous glitches due to intermittent and transient faults
	[22]		90nm Spartan-3	CLBs	Monitoring late transition	Pros: Not affected by the power supply voltage variations and requiring fewer resources
	[23]		28nm Virtex7	CLBs	Monitoring late transition	Pros: Lower overhead Cons: Improving the robustness of the sensor and increasing path delay due to the sensor inclusion
	[24]		40nm Virtex6	CLBs		Pros: Presenting a sensor insertion algorithm to avoid sensors inaccuracy
	[25]		28nm Virtex7	CLBs	Monitoring late transition	Pros: Fewer resources
	[7]	Process variation and ageing monitoring	28nm Virtex7	CLBs	Frequencies (Counter) and Time	Pros: Avoiding unnecessary sensor insertion and performance loss due to sensor insertion

Chapter 4

The Proposed PV Mapping Method

In this chapter, the implementation details of PV map are described. We limit the scope of monitoring for ageing to testing CLBs and switch boxes, and we utilize RO method to monitor the ageing based on the following advantages. Firstly, ROs do not need input data (as compared with using a circuit such a multiplier or full-adder, which is needed for the SR and TP methods). Moreover, unlike the TP and SR methods, every element included in the RO is included in the delay measurement and its degradation becomes the part of monitoring, whereas for the other methods, we should find CPs and NCPs and provide the input data so that those CPs and NCPs are exercised. In addition, ROs have a regular structure and can be systematically placed so that we know exactly which resources (or part of resources are covered by each type of RO). Furthermore, a PV map created of many RO sensors can also find other applications (besides monitoring the ageing process), such as more fine grained binning for a given FPGA speed grade, as well as observing the extent of process variation across the fabric.

The more fabric resources the PV mapping method monitors, the more accurate the analysis of the extent of ageing will be. It was, therefore, important for us to know how much of the fabric resources we could monitor with the RO-based PV map method, but prior work missed the important details. Firstly, none of the previous works reported the total number of sensors. We needed to know how close we could get to the theoretical maximum number of RO sensors before getting blocked by routing congestion or similar practical hurdles. Secondly, in previous work, limited set of LUTs are used for the ROs, where from each LUT mostly only one input is used.This results in only part of the LUT being monitored for ageing. Besides, it is not mentioned if the authors have put constraints on which LUT inputs should be used in the ROs. In the absence of such constraints, the synthesis tools might be inclined to use the inputs that control the muxes closest to the output (to reduce delay), resulting in even less hardware resources being included in the RO circuit. Therefore, it was important for us to investigate how much of the LUT internal resources we could incorporate into each RO sensor, and what (if any) trade-offs we would face in doing so. Thirdly, it was not clear if the differences in routing delays are accounted for in the obtained map, which becomes important when comparing neighboring sensors, as well as comparing different internal paths of a LUT, as discussed in Section 4.2.1.

We decided, therefore, to implement and experiment with a PV mapping design where we could compare different RO sensors (both the RO and counters) with the following objectives:

- How much of the fabric resources can be covered (in terms of number of sensors and type of included elements)?

- How densely can we pack the sensors?
- How uniform can the sensors across the fabric become (in terms of routing delays and internal LUT paths)?
- What is the precision and accuracy of each sensor type is?
- How much variation can we measure across different devices by using the implemented RO (just to validate that the proposed mapping method is capable of detecting delay differences, irrespective of the source being ageing or process variation)?

4.1 Architecture of the Proposed Method

The high-level architecture of the proposed technique is shown in Figure 4.1. The *Sensors* block is an array of RO sensors mapped into FPGA's fabric, where one sensor uses a reference clock instead of an RO to help establish the correctness of the frequency calculations, and each of the remaining sensors is composed of a ring oscillator and a frequency counter.

The *Global Controller* reads the measurement results from the sensors one at a time and stores the results into the memory. Reading from a sensor involves several steps, which will be detailed in Section 4.2.3. The memory part, composed of *Memory Controller* and *BRAMs*, receives and stores the sensor data sent by *Global Controller* from via *AXI Interconnect*. External computer connected to the *JTAG* port can read out the data stored in the BRAMs via the *AXI Interconnect* and *JTAG to AXI* modules, and perform data analysis. The design details of our method are presented in the following sections.

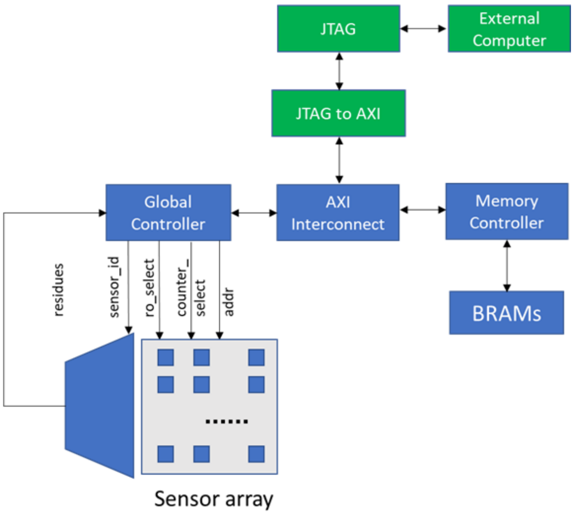


Figure 4.1: Overview of proposed technique

4.2 RO Sensor Design and Deployment

As mentioned in Section 3.1, the RO sensor consists of a ring oscillator and frequency counter. In order to monitor the ageing of logic blocks more accurately, four types of ROs are designed. Considering that the RNS ring counter has higher compactness than the binary counter and the linear shift register counter (LFSR counter), it was chosen as the frequency counter. The downside with using the RNS ring-counter (when aiming for the most compact implementation) is that it can only be placed in SLICEM fabric resources.

4.2.1 Ring Oscillator

It is important to make sure that each RO retains its structure (and thus frequency) between synthesis rounds. The reason is that if a new synthesis round is performed for a product that has been being monitored for a while, the previous measurement results for each RO should still be relevant when being compared to the measurement results after the new synthesis round. In addition, it is desirable to minimize the frequency differences due to the different placement of each RO's elements between ROs in the same design. This way, reasoning on process variation and/or ageing rate in different areas of the same die becomes easier. Finally, it is important to include as many resources in a LUT as possible in the sensor. Therefore, the LUT's inputs should be chosen such the longest path inside each LUT is included in the RO.

The achieve the above three goals, different physical constraints are used to fix the location of each RO, including the placement of each RO element. Besides, the exact inputs to be used for each RO element are specified in the RTL through instantiation and explicit wiring of LUTs and carry chains.

Resource utilization of the four types of ROs with different structures is shown in Table 4.1. In order to get the longest path in the carry chain, 8 LUTs are configured as inverters, and the 'O6' outputs of the other LUTs are configured as buffers to feed the 'S' input of carry chain. Therefore, digit '8' in 'ro_8_cc2_8_cc2' and 'ro_16_8_cc2' names denotes that 8 LUT5s in one CLB are used as inverters. 'ro_low' and 'ro_high' are only developed for experimental comparison of internal LUT delays caused by ageing and process variation, whereas the other two are designed as ageing sensors for the proposed PV mapping technique.

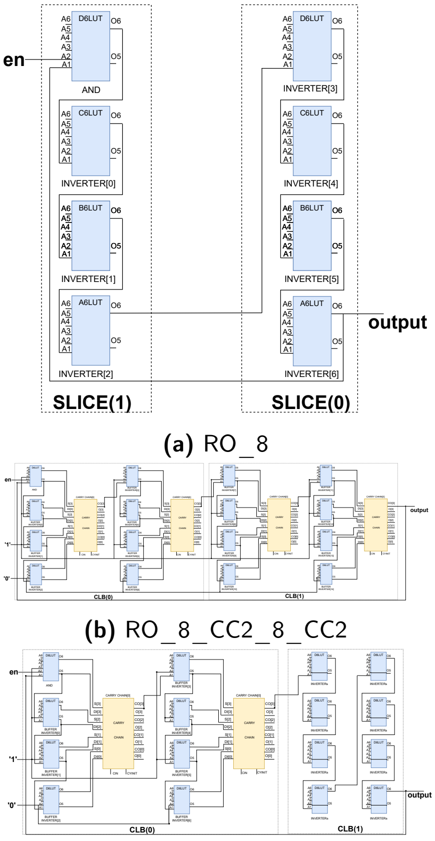
Table 4.1: Resources utilization of four ROs

RO	CLB	LUT6(total)	LUT5(total)	Carry Chain
ro_high	1	8	0	0
ro_low	1	8	0	0
ro_8_cc2_8_cc2	2	0	32	4
ro_16_8_cc2	2	0	24	2

Each RO is implemented by utilizing a certain number of LUT5s or LUT6s and carry chains. The numbers in Table 4.1 represent the number of elements required to form ROs. One LUT5/LUT6 is configured as a control logic (logic And Gate), which activates or deactivates the ring oscillator. Self-heating can be mitigated due to AND gate introduction by activating each RO only during its measurement turn. The remaining LUT5s/LUT6s are configured as inverters. The carry chains are used as delay elements.

Figure 4.2 illustrates the schematics of our RO designs. 'ro_high' and 'ro_low' utilize the same resources to form RO shown in Figure 4.2a, where the input of the inverter is connected to 'A1', the inputs of control logic are connected to 'A1' and 'A2' connected to the signal 'en', and the remaining inputs are connected to '1' in 'ro_high' and '0' in 'ro_low'.

Schematics of 'ro_8_cc2_8_cc2' and 'ro_16_8_cc2' are shown in Figure 4.2b and Figure 4.2c, respectively, where, for the unconnected pins, 'A6' of each LUT is connected to '1' to ensure that the MUX selects the output result of one LUT5 to O6, and the output result of the other LUT5 to O5, and the others are connected to the ground. In order to form a RO, O6s are used to configure the carry chain, and O5s are part of RO. As a result, Carry Chains are configured as buffers achieved by fixing 'S' at "1110", and the longest path in Carry Chain is used.



(a) RO_8

(b) RO_8_CC2_8_CC2

(c) RO_16_8_CC2

Figure 4.2: Schematics of RO designs

4.2.2 Frequency Counter

As mentioned in Section 3.1.4, we can conclude that the RNS ring counter is far more compact and more efficient among the three counters. Therefore, in our experiments, the RNS ring counter is chosen as the frequency counter. Three moduli of 29, 31, and 32 are set in the frequency counter, as shown in Figure 4.3. The highest count for our frequency counter is 28,768 (29 × 31 × 32). The measurement length should be chosen so that for a given RO frequency range, the number of counted clock periods (ticks) does not result in an overflow. For example, assuming that the RO frequency might be as high as 100MHz, the measurement length should not exceed $\frac{28768}{100e6}$ = 287.68us. In general, especially when an RO type with high-frequency is chosen, the RNS counters can even show ageing in their structure as we will get outlying frequencies that might be extremely high or low if any of the three SRLs are affected by ageing. For example, if three residues are 11, 9, and 8 under the condition of no ageing effect, the number of counter ticks is 16,937. Assuming that the first residues change to be 10 due to the first SRL32 affected by ageing, the corresponding counter number is 41.

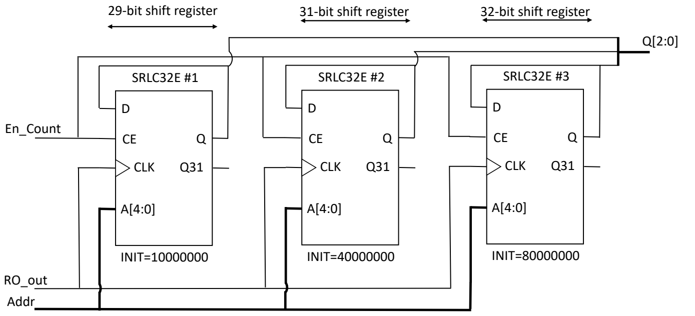


Figure 4.3: Schematic of RNS ring counter

4.2.3 Sensor Deployment

Ideally, we want to include all fabric elements in our sensors, but this is not practical as some of the elements are needed for the data collection infrastructure and the counters. Besides, even those elements included in our sensors could only be partially used (consider that for each LUT6 we only use maximum two internal paths.). The theoretical maximum number of sensors is determined by what is needed for a sensor and what is available in the used FPGA.

A sensor that is only composed of one RNS ring counter is used as a reference to measure the clock signal with a fixed frequency of 20MHz, and the remaining sensors are RO sensors. Sensor locations are fixed using a physical constraint file via a TCL script. The deployment of 1400 sensors is shown in Figure 4.4, where the right image shows the complete fabric and the left image is the magnification of the bottom-left part of the fabric. In Figure 4.4, the sensors are placed into physical blocks (Pblocks) marked in pink consisting of two rectangles next to each other, where the RO is located into the larger rectangle and the counter located into the smaller one. The reference sensor only includes the RNS ring counter (Count_0), which is used to measure the clock signal with a fixed frequency of 20MHz. An ageing sensor (sensor 14) is composed of one RO (RO_14) and one RNS ring counter (Count_14).

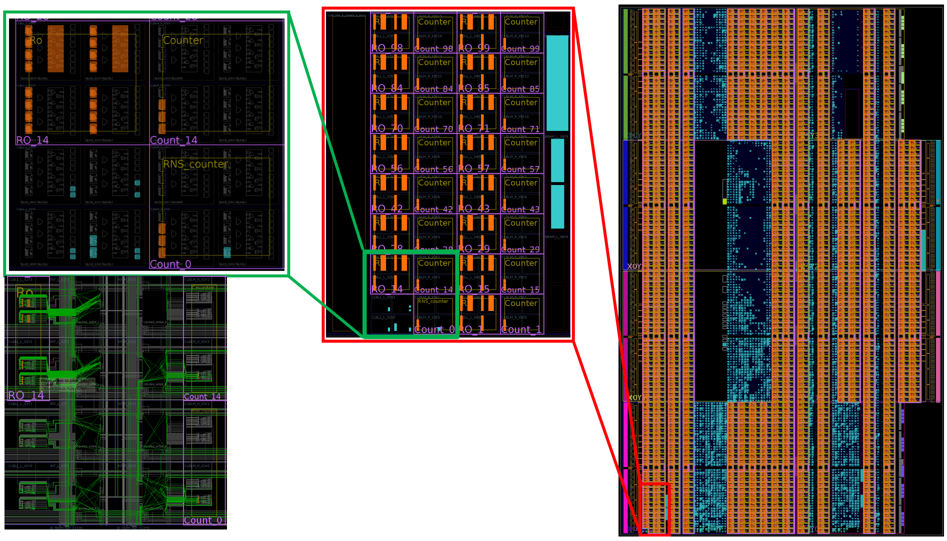


Figure 4.4: The deployment of 1400 sensors

4.3 Design and Operation of the Proposed Method

In this section, the design and operation of the proposed PV mapping technique are presented. The overall architecture is already presented in the introduction of this chapter, and here we focus in more detail on the modules developed in this thesis work and their operation. The main design is shown in Figure 4.5, where N sensors are placed in the sensor array, and a controller is a state machine that operates the sensors and stores the results in memory via the AXI interface.

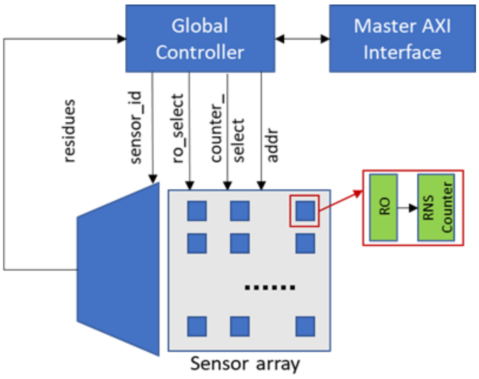


Figure 4.5: The Principal Design of our method

The operation flow of our method is illustrated in Figure 4.6, where 'N_S' and 'N_R' represent the number of sensors and the number of repetitions. The repetition of measurements is a feature that we only need for precision measurement experiments, and is not meant to be always included in the PV mapping circuitry. The main reason for this is that for repeating measurements for a sensor, there is a need to reset the SRL32 blocks (as SRL32s lack a reset signal and are normally meant to be initialized once during the programming of the FPGAs). The added reset circuitry takes resources from the fabric that could otherwise be used for including more sensors into the PV mapping design.

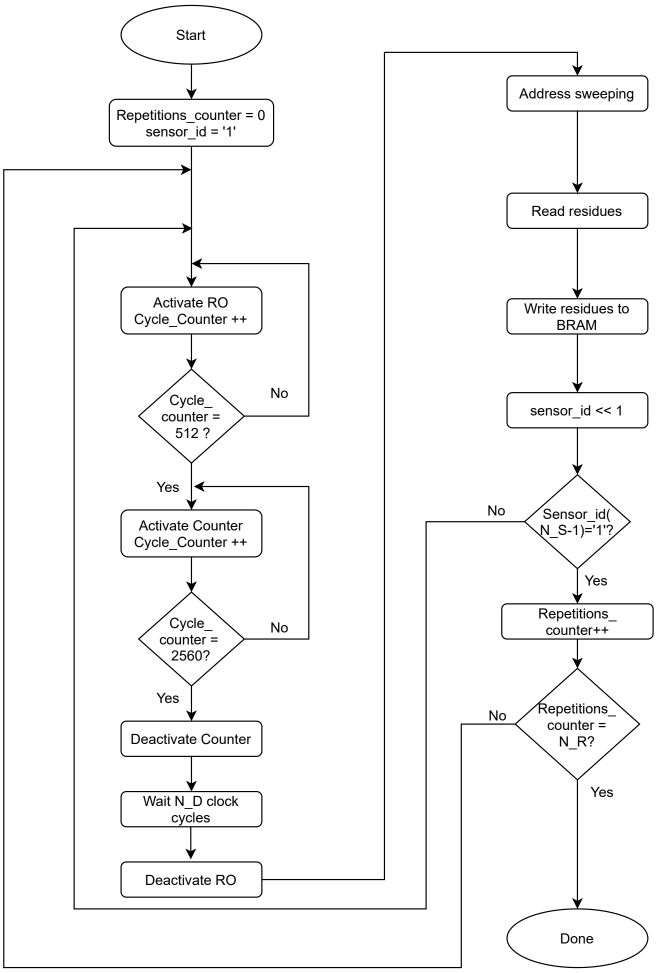


Figure 4.6: The operation flow of our method

As shown in Figure 4.6, the operation is described as follows. First, the signal $sensor\_id[N-1:0]$ is specified to select among the residues outputs of the N sensors. Then, the signal $ro\_select[N-1:0]$ and $counter\_select[N-1:0]$ are used to activate the RO and the counter of corresponding sensor, respectively. In addition, the signal $residues[2:0]$ and $addr[15:0]$ are utilized to find the residues and store the three residues.

In order to avoid metastability in clock domain crossing and remove glitches (by glitches, we mean the width of the clock enable signals being less than what is allowed by the specification), $counter\_select$ needs synchronization, which can be achieved by introducing extra FFs shown as in Figure 4.7.

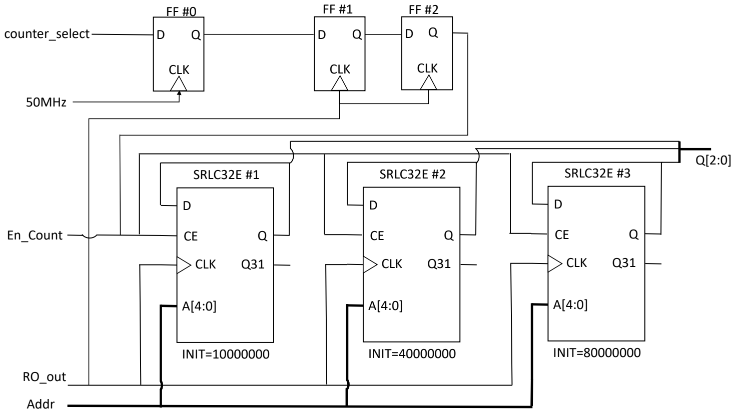


Figure 4.7: The schematic of introducing extra Flip-Flops

To activate and read from each sensor, the select signal (i.e. $sensor\_id[N-1:0]$) is set to choose the residues output of the sensor that is being activated. After the sensor's output is chosen, the enable signal of RO (i.e. $ro\_select = 1$) is activated such that RO starts oscillating. To let RO reach a steady-state with a stable and constant frequency, the controller waits for 10.24 μs, and next, the enable signal of the counter is activated to count (i.e. $counter\_select = 1$) for 40.96 μs. These time durations are chosen to abide by the total $50\mu s$ maximum activation of the RO mentioned in [10] to avoid damages due to extra heating. In Section 4.4, we will discuss the impact of the choice on measurement length of the accuracy of measurements. After that, enabling signals of RO and counter are disabled in separate steps to allow the *counter_select* signal propagate to the counter before we disable the RO (i.e. $ro\_select = counter\_select = 0$). When the controller has a higher frequency than the ROs, we need to wait for at least $2 \times \frac{F_C}{F_{RO}}$ clock cycles (F_C being the frequency of the controller, and F_{RO} being the frequency of the RO) to let the deactivation signal go through the synchronization flip-flops before the RO is disabled. Then, the 15-bit address (5-bit address bus for each SRL32) is swept, and simultaneously the 3-bits output from the RNS ring counters (each bit for one SRL32 output) is monitored for a value of '1' to obtain the residue values of the counter. Since each residue bit comes from a different SRL, the three residue bits do not become '1' at the same time, and therefore, as soon as a residue bit becomes '1', its corresponding 5-bit address is stored. The value of the counter's 15-bit address bus is stored as the concatenation of the three separately stored 5-bit addresses. Next, the obtained residue values are stored in the BRAMs via the AXI interface. When the residues are recorded, it means that one measurement of one sensor is finished. Then, immediately, the next sensor is chosen and the above steps are repeated until one measurement of all sensors is done.

The memory requirements for a design with N sensors and M repetitions is calculated as:

$$M_{size} = \frac{16(bits) * N(sensors) * M(times)}{1024 * 8bits}(Kbytes) \tag{4.1}$$

where M_{size} is the size of memory, and 16-bits is used to store the three residues.

All measured residue values are transferred to an external computer via a JTAG port, which is achieved by using a TCL script.

4.4 Maximum Measurement Error

We calculate the frequency of the ring oscillator RO by counting the number of rising edges (denoted by n) during the measurement period (denoted by T_m) as:

$$f_m = n/T_m \tag{4.2}$$

where f_m denotes the measured frequency. An immediate observation here is that f_m is always a multiple of $1/T_m$. We will refer to $1/T_m$ as the measurement resolution, as the measurement cannot detect frequency changes below $1/T_m$. In this section, we would like to show that the measurement error is not larger than $1/T_m$. We define the measurement error as

$$e = \left| f_m - f_{RO} \right| \tag{4.3}$$

where f_{RO} denotes the (true) frequency of the RO. From Equation 4.2, Equation 4.3 can be rewritten as:

$$e = \left| \frac{n}{T_m} - f_{RO} \right| = \left| \frac{n}{T_m} - \frac{1}{P_{RO}} \right| \tag{4.4}$$

where P_{RO} denotes the period of RO's signal. The use of period instead of frequency makes it slightly easier to reason about the maximum error.

In our solution, the counter_select signal is sampled by the RO's clock. Therefore, the length of measurement period as seen by the frequency counters (T'_m [1]) is not the same as T_m (that is used by the frequency calculation algorithm), and depends also on the frequency and phase of the RO. Figure 4.8 shows how the phase of the RO's output affects T'_m, where for Case 1, two rising edges have coincided with counter_select activation length (i.e., T_m) and for Case 2 and Case 3, three rising edges happen within T_m. Therefore, for Case 1 we have n = 2, and for Case 2 and Case 3 we have n = 3.

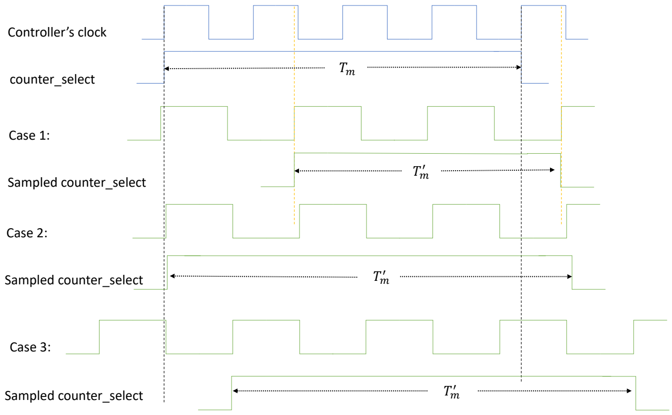


Figure 4.8: Cases of the phase of the RO's output affecting T'_m

[1] It should be noted that $T'_m = nP_{RO}$.

The dependence of the counted number of edges (n) on T_m and P_{RO} can be captured by the following relation:

$$\left\lfloor \frac{T_m}{P_{RO}} \right\rfloor \leq n \leq \left\lceil \frac{T_m}{P_{RO}} \right\rceil \tag{4.5}$$

In Equation 4.5, the floor and ceiling functions are used to represent that the fractional remainder of the T_m/P_{RO} division, depending on the phase, can result in different number of rising edges that coincide with the measurement period. For example, in the Figure 4.8, $T_m/P_{RO} = 16/6 = 2.67$. For Case 1, n = 2 can be captured by the floor function in Equation 4.5, whereas for the other cases, n = 3 will be represented by the ceiling function. Given the dependence on n on T_m and P_{RO}, we should consider the relation between T_m and P_{RO} to be able to reason about the maximum error calculated by Equation 4.4.

We consider three scenarios:
Scenario 1: $T_m < P_{RO}$
In this case, from Equation 4.5 we can only have n = 0 or n = 1. By plugging these values in Equation 4.4 we will have:
$n = 0 \Rightarrow e = \frac{1}{P_{RO}}$
$n = 1 \Rightarrow e = \left|\frac{1}{T_m} - \frac{1}{P_{RO}}\right|$
Between the two error values, $e = 1/P_{RO}$ is the largest.

Scenario 2: $T_m = P_{RO}$
In this case, from Equation 4.5 we have $n = 1$, thus
$n = 0 \Rightarrow e = 0$

Scenario 3: $T_m > P_{RO}$
In this case, we should consider both upper and lower limits for n presented by Equation 4.5. We start by the lower limit:

$$\lfloor \frac{T_m}{P_{RO}} \rfloor \leq n \Rightarrow \frac{T_m}{P_{RO}} - 1 \leq n \leq \frac{T_m}{P_{RO}} \tag{4.6}$$

Noting that we always have $n \geq 0$, we use the bounds for n from Equation 4.6 to reason about e:

$$n = \frac{T_m}{P_{RO}} - 1 \Rightarrow e = \left|\frac{\frac{T_m}{P_{RO}} - 1}{T_m} - \frac{1}{P_{RO}}\right| = \frac{1}{T_m} \tag{4.7}$$

$$n = \frac{T_m}{P_{RO}} \Rightarrow e = \left|\frac{\frac{T_m}{P_{RO}}}{T_m} - \frac{1}{P_{RO}}\right| = 0 \tag{4.8}$$

Now, from Equation 4.5, we consider the upper limit for n:

$$n \leq \lceil \frac{T_m}{P_{RO}} \rceil \Rightarrow \frac{T_m}{P_{RO}} \leq n \leq \frac{T_m}{P_{RO}} + 1 \tag{4.9}$$

Similar to what we did for Equation 4.6, we use the bounds presented by Equation 4.9 to reason about e. We have already considered $n = T_m/P_{RO}$ in Equation 4.8. Therefore, we only take the upper bound in Equation 4.9:

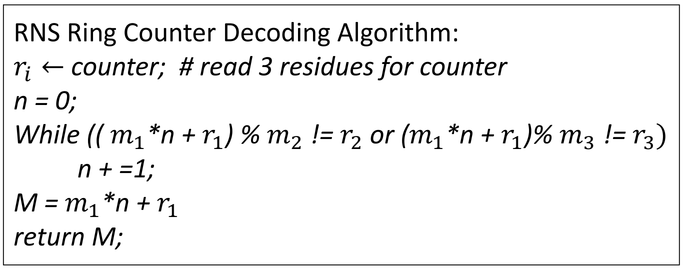
$$n = \frac{T_m}{P_{RO}} + 1 \Rightarrow e = \left| \frac{\frac{T_m}{P_{RO}} + 1}{T_m} - \frac{1}{P_{RO}} \right| = \frac{1}{T_m} \tag{4.10}$$

From Equation 4.7, 4.8 and 4.10, we see that the largest error is $1/T_m$.

In Scenario 1, the measurement error can be as large as $1/P_{RO}$, which from the assumption of $T_m < P_{RO}$, also results in $e < 1/T_m$. Therefore, considering all three scenarios together, we can conclude that the largest measurement error is $1/T_m$.

4.5 Data Retrieval and Analysis

Finally, the data from BRAM content is analyzed and a 3-D PV map of frequency for N sensors is constructed, which will be described in detail in Chapter 5. Data analysis can be performed after decoding the residue values by using the decoding algorithm shown in Figure 4.9.



```
RNS Ring Counter Decoding Algorithm:
$r_i$ ← counter;  # read 3 residues for counter
n = 0;
While (( $m_1$*n + $r_1$) % $m_2$ != $r_2$ or ($m_1$*n + $r_1$)% $m_3$ != $r_3$)
        n + =1;
M = $m_1$*n + $r_1$
return M;
```

Figure 4.9: Algorithm to calculate the counter ticks from the residue values

Where, i is from 1 to 3, r_i is residue values from SRL32, m_i is the moduli value (i.e. m_1 = 29, m_2 = 31, m_3 = 32) and M is the number of counter ticks during the measurement period. To calculate the frequency, M should be divided by the measurement length.

4.6 Conclusion

In this chapter, we presented out a proposed method for ageing monitoring by using a PV mapping technique. In the next chapter, we will present experiments performed using the proposed architecture.

Chapter 5

Experimental Setup and Results

In Chapter 4, we presented our PV Mapping method for monitoring ageing in FPGAs. In this chapter, we present different measurements and experiments we have carried out to validate the proposed method, as well as to gain insight into some of its features such as the precision of the measurements, etc. To perform the measurements, we have used Digilent Nexys4 boards, featuring a 28nm Xilinx ARTIX 7 FPGA, namely XC7A100T. Due to the destructive nature of accelerated ageing, we chose not to expose the boards to accelerated ageing, but instead carry out the measurements on 20 of these boards that have been used in different labs at the Department of Electrical and Information Technology in the past years. Therefore, the measured RO frequencies are affected by both the initial process variation, as well as the different degrees of ageing for each board.

At first, single measurements with the highest number of sensors are performed. Then, 'ro_high' and 'ro_low' are compared to investigate how large the differences between the delay of different internal paths of LUTs are. Subsequently, since it is impossible to cover the whole fabric with RO sensors, two alternatives are presented in the following section. In the last section, multiple consecutive measurements with 140 sensors are conducted to dig into how stable the RO frequencies are.

5.1 PV Mapping Using the Two Large Sensors

To get a high-resolution PV map, each sensor should have the largest possible set of elements, and the largest number of sensors should be placed on the fabric. 'ro_8_cc2_8_cc2' and 'ro_16_8_cc2' are chosen to construct PV mapping since 'ro_8_cc2_8_cc2' and 'ro_16_8_cc2' have the larger set of elements as mentioned in Section 4.2.1.

The relative locations of RO and counter in two sensors are illustrated in Figure 5.1, where the blue cell is SLICEL, the orange cell is SLICEM, the purple cell is RAM, and green cell is DSP.

We have 7925 CLBs on Artix-7 100T. Ideally, we would like to monitor all those CLBs for ageing. However, we cannot monitor all these 7925 CLBs with ROs because we need to use part of these CLBs as frequency counters and frequency counters can only be placed in SLICEM resources. Given the layout of the resources in the fabric, using SLICELs for ROs and SLICEMs for counters limits the total possible of our sensors to 1774, equivalent to $1774 \times 4 = 7096$ CLBs. However, due to routing limitations, the highest number of sensors for which the tool could finish the implementation was 1400. These sensors are deployed on the die in a 100×14 arrangement, as shown in Figure 4.4

Two types of RO sensors were implemented. Their corresponding results with regard to simulation results, measurement results, and PV maps are described in this section.

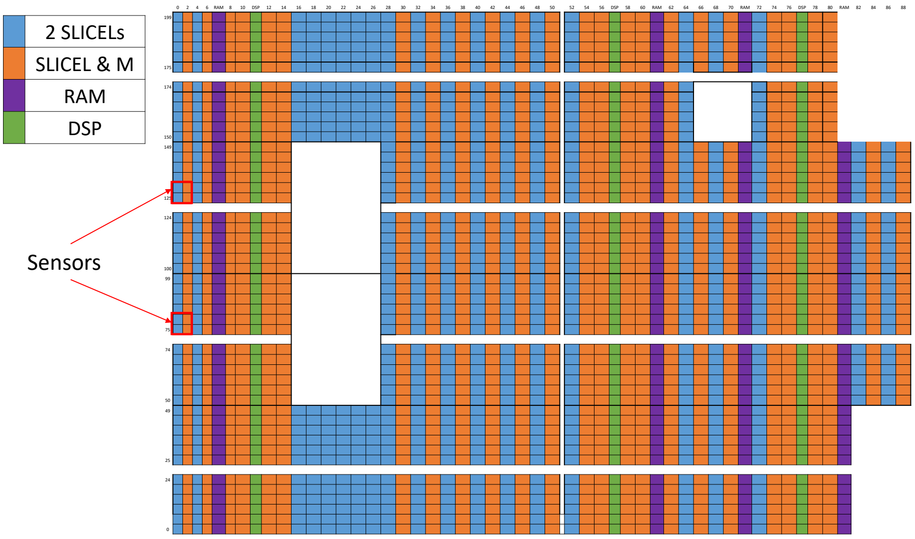

2 SLICELs
SLICEL & M
RAM
DSP
Sensors
0 2 4 6 RAM 8 10 DSP 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50 52 54 56 DSP 58 60 RAM 62 64 66 68 70 RAM 72 74 76 DSP 78 80 RAM 82 84 86 88
199
175
174
150
149
124
100
99
75
74
50
49
25
24
0

Figure 5.1: FPGA resources layout

5.1.1 Simulation Results

We have performed Post-Implementation Timing simulations for the designs with the 'ro_8_cc2_8_cc2' and 'ro_16_8_cc2' sensors, and the results are shown in Table 5.1, where Accuracy is calculated by Eq. 5.2.

$$error\ rate = \left|\frac{measured\ frequency - actual\ frequency}{actual\ frequency}\right| \times 100\% \tag{5.1}$$

$$accuracy = 100\% - error\ rate \tag{5.2}$$

The 'ro_16_8_cc2' has the same exact frequency, 17.895508MHz for all 1400 sensors. The 'ro_8_cc2_8_cc2' kept changing between 24.365234MHz and 24.389648MHz, which are 24.414KHz away (explained in Section 4.4). As shown in Table 5.1, the simulation results are (almost) the same for all sensors, therefore, we can conclude that we have achieved this uniformity. It should be noted that in Table 5.1, the measured frequency reported in Column "Measured" is what our sensor reports in the context of simulation. The slight difference observed for 'ro_8_cc2_8_cc2' might be rather due to the main controller's clock arriving at different sensors with different phases, causing the low-frequency ROs in the 'ro_8_cc2_8_cc2' design to capture the counter enable/disable signal differently.

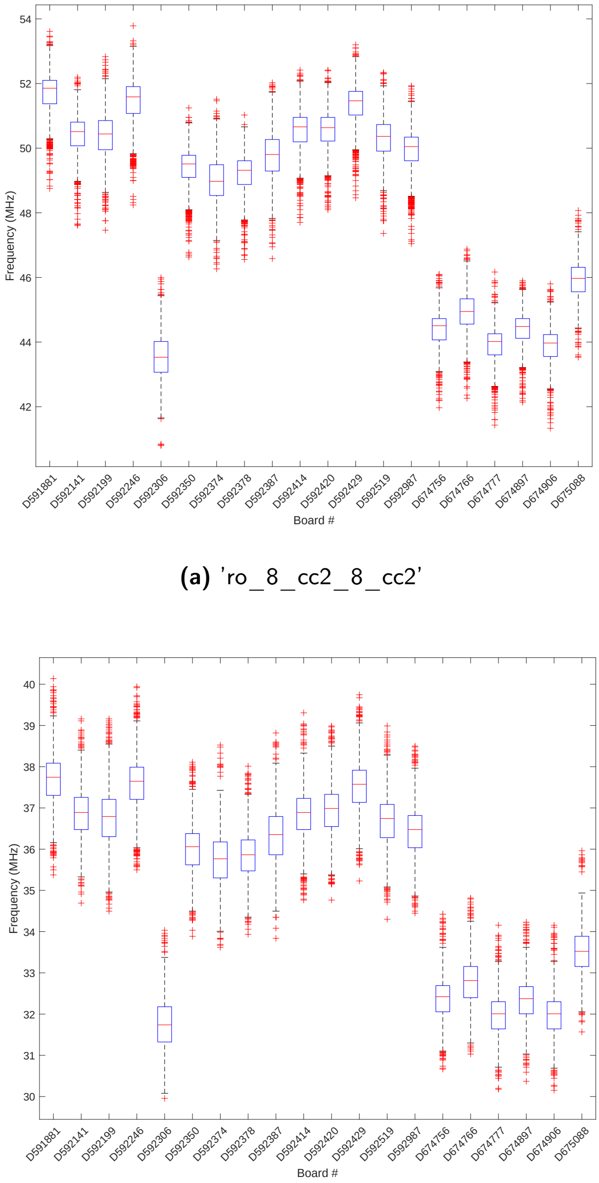
Table 5.1: Simulation and actual results, and accuracy for 'ro_8_cc2_8_cc2' and 'ro_16_8cc2'

Design Name	Ref frequency (MHz)			ROs (MHz)		
Results	Actual	Measured	Accuracy	Actual	Measured	Accuracy
ro_8_cc2_8_cc2	20	19.995117	99.97559%	24.38073	24.365234	99.93644%
				24.39738	24.389648	99.96829%
ro_16_8_cc2	20	19.995117	99.97559%	17.93014	17.895508	99.80683%

The higher accuracy in the case of the 20MHz reference clock can be explained by the fact that we have not used synchronization FFs on the counter_select signal going to the counter for this reference clock, thus having the actual measurement period closer to the one we use for the frequency calculations. The reason for not using synchronization FFs, in this case, is that the 20MHz reference clock and the 50MHz clock for the main controller are generated by the same PLL from a 100MHz reference clock. These simulations are performed in the slow corner. Knowing that typically the same slow corner is used in closing the timing for digital designs, we can consider the simulated frequencies as the lowest acceptable frequencies for each RO, below which the device has aged beyond an acceptable range. In other words, when monitoring for ageing throughout the life-cycle of a product, the difference between the measured RO frequency for each sensor and the simulated one can be used as an ageing indicator; the less the difference, the higher the degradation. If the accuracy is not deemed enough, especially when the measurements show that the observed frequencies are getting closer to the simulated ones, one can increase the accuracy by increasing the duration of the measurement for each sensor (while paying attention to the maximum ticks allowed by the RNS counter before it overflows, and thermal issues that might arise by keeping each RO running longer).

5.1.2 Measurement Results

The two designs 'ro_8_cc2_8_cc2' and 'ro_16_8_cc2' are tested on 20 boards, and the corresponding visualize summary statistics is shown in Figure 5.2. It is clearly shown that the proposed method is capable of detecting the performance differences among these boards. The observed inter- and intra-board frequency differences might be due to a combination of process variation and ageing, as well as small voltage and temperature differences between boards and measurements. In this regard, it should be noted that the purpose of these measurements is to validate the ability of the mapping method to detect frequency differences against simulated values, irrespective of the cause of the difference. For future work, it could be interesting to take into account the effect of factors such as voltage and temperature on the measured frequencies. Two boards, namely D591881 and D592306, stand out as showing the highest and lowest maximum frequency, respectively, among the measured boards.



(a) 'ro_8_cc2_8_cc2'

(b) 'ro_16_8_cc2'

Figure 5.2: Visualize summary statistics with box plot for 'ro_8_cc2_8_cc2' and 'ro_16_8_cc2' on 20 Boards

The corresponding PV-maps of Board 'D591881' and 'D592306' are shown in Figure 5.3, 5.4, 5.5, and 5.6. X_Loc and Y_loc axes represent the location of RO sensors since

XY representation determines the coordinates of FPGA resources. In addition, since no programmable resources exist and therefore no RO sensors are placed in some coordinates, there are some white areas in the charts. As shown in Figure 5.3 and 5.4 , for board 'D591881', the frequency of ROs in two designs ranges from 48 MHz to 53 MHz, and from 35 MHz to 40 MHz. Figure 5.5 and 5.6 show the frequency of PV-ROs with two designs on board 'D592306' ranges from 40 MHz to 45 MHz, and from 29 MHz to 34 MHz. The frequency indicates the delay of FPGA resources varies due to performance variation.

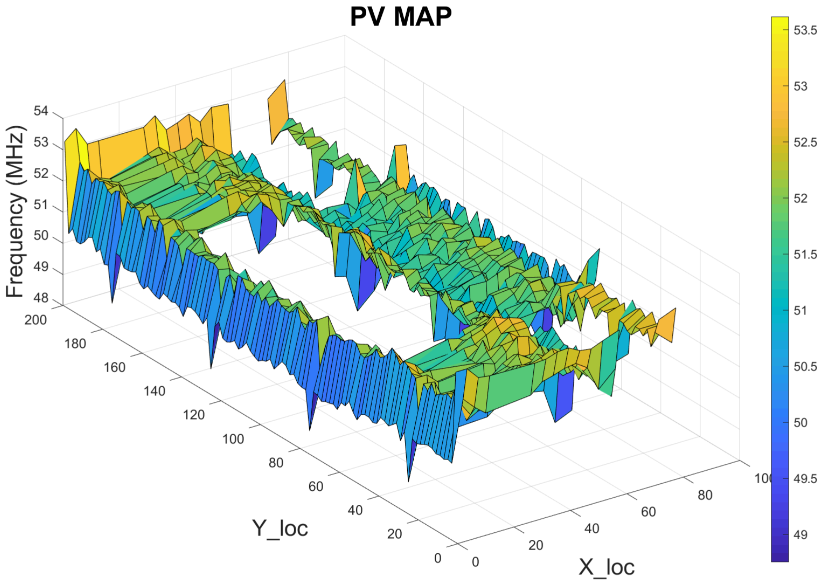


Figure 5.3: PV map for 'ro_8_cc2_8_cc2' of Board D591881

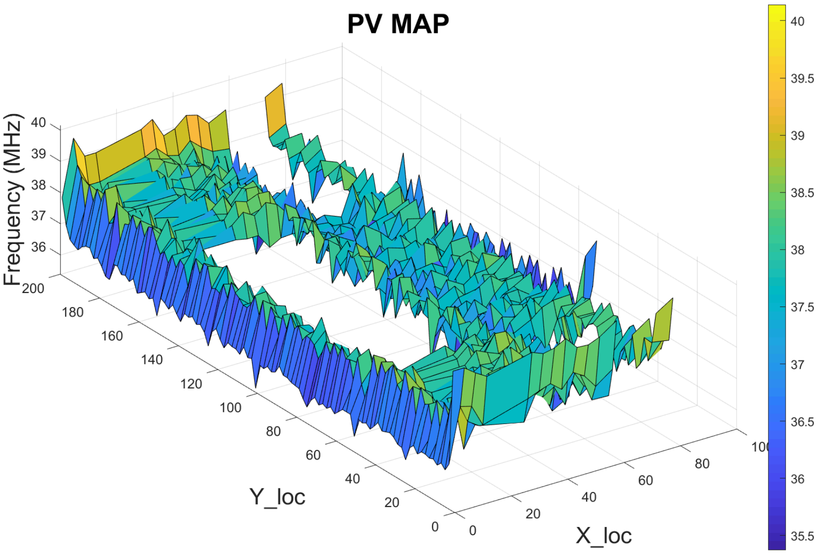


Figure 5.4: PV map for 'ro_16_8_cc2' of Board D592306

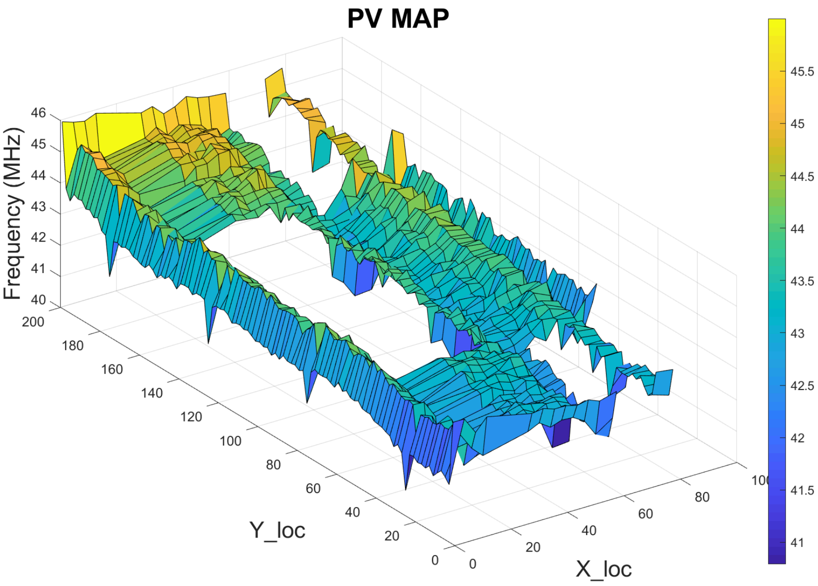


Figure 5.5: PV map for 'ro_8_cc2_8_cc2' of Board D591881

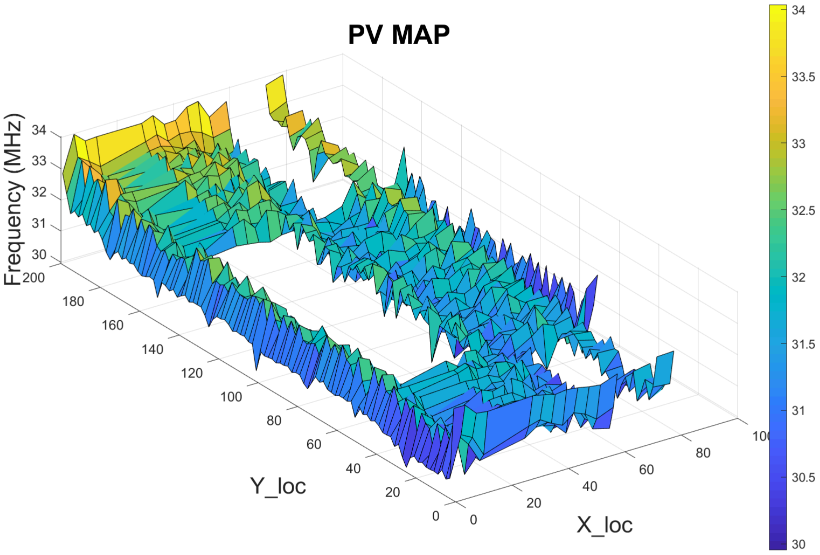


Figure 5.6: PV map for 'ro_16_8_cc2' of Board D592306

5.1.3 Discussion

As mentioned above, the measurement results show that the proposed PV mapping method is capable of detecting differences in delays (that could be due to a mix of process variation and ageing) among different sensors on the same device as well as for the same sensor among different devices. One can, therefore, trust that when continuously monitoring the same FPGA with this method over its life-time, the method is capable of detecting the degradation that gradually happens to the elements covered by each RO. There are, however, several issues that we need to investigate about the proposed method. Firstly, we only test one path in a LUT, while it is necessary to look into how different the other paths might be. Secondly, while it is worthy to investigate if we can extrapolate the ageing degree on resources not used in the PV map. Thirdly, the precision of the sensors should also be studied. These three aspects will be reported in the following sections.

5.2 RO_high vs RO_low

In order to investigate how much error if we only measure one path (out of 64 possible paths) inside the LUT. We chose the two high and low paths so that they have the least number of shared elements (transistors and interconnect). Figure 5.7 shows the LUT6's inner structure, where the red line and blue line represent the path of 'ro_low' and 'ro_high', respectively.

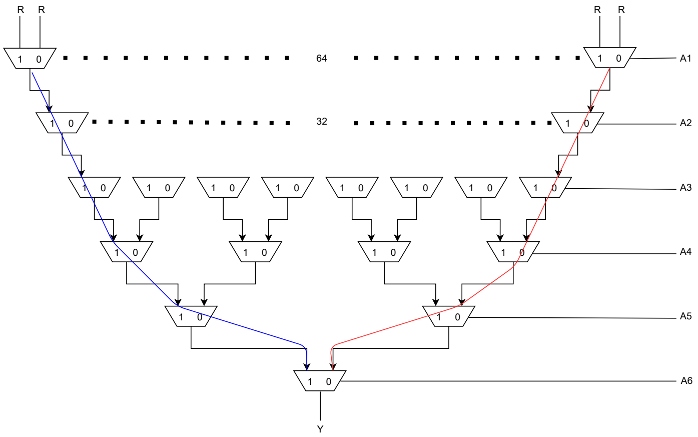


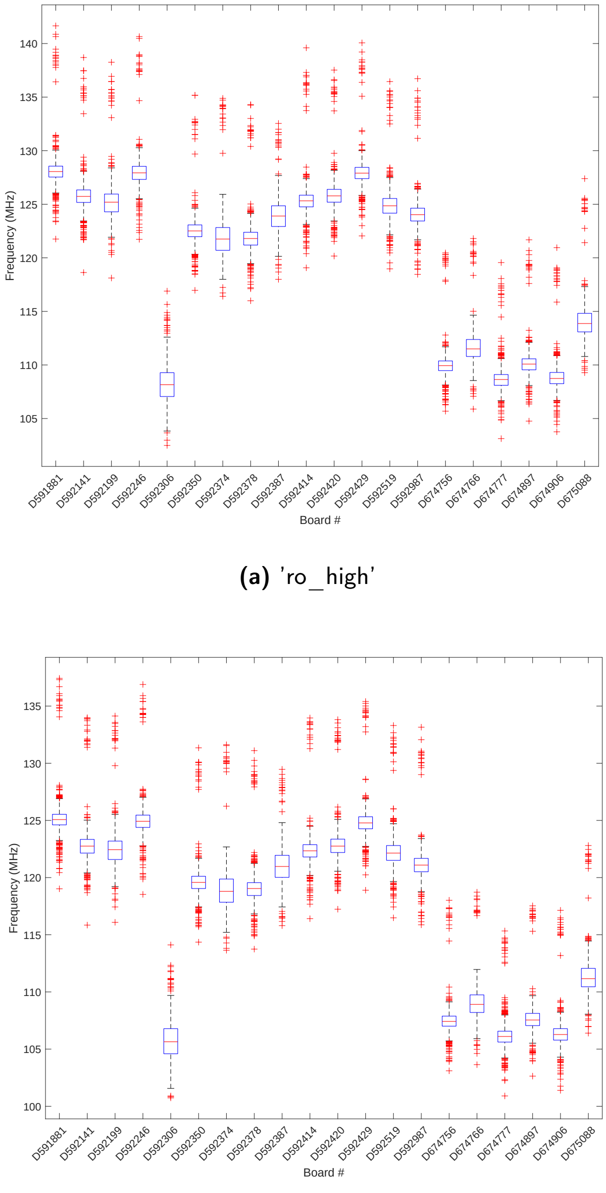
Figure 5.7: LUT6's inner structure

The 'ro_high' utilizes six multiplexers on the left. On the contrary, 'ro_low' uses six multiplexers on the right. We should note that the two paths share the last multiplexer. However, as illustrated in Figure 5.7, both designs have the same length path. This can be confirmed by looking at the simulation results shown in Table 5.2. We are, of course, expecting differences in board measurements due to process variation. However, as the boards have been in use for several years, the observed differences are a mix of both ageing and process variation.

Table 5.2: Simulation and actual results, and accuracy for 'ro_low' and 'ro_high'

Design Name	Ref frequency (MHz)			ROs (MHz)		
Results	Actual	Measured	Accuracy	Actual	Measured	Accuracy
ro_low	20	19.995117	99.97559%	56.657224	56.640625	99.97071%
ro_high	20	19.995117	99.97559%	56.657224	56.640625	99.97071%

Both designs have the same simulation results, which is 56.640625Mhz for all 1400 sensors across the fabric. However, as shown in Figure 5.8, the measurement results of 'ro_high' are a bit higher than 'ro_low's, which is hard to justify without knowing the actual physical layout of the LUTs, as well as without any information on how much the ageing has contributed to the difference (due to the devices not being new). However, considering the difference between the simulation results (for the slow corner) and the measurement frequency, we can conclude that the difference between 'ro_high' and 'ro_low' are not that significant. Therefore, it might not make a big difference if the remaining inputs of LUTs are specified at '0' or '1'. On the other hand, if the observed difference is not acceptable, one could consider such difference as uncertainty margin when reasoning about the degree of degradation.



(a) 'ro_high'

(b) 'ro_low'

Figure 5.8: Visualize summary statistics with box plot for 'ro_high' and 'ro_low' on 20 Boards

In order to investigate the differences at the sensor level, we choose the measurement results from Board 'D591881'. The histogram of differences between the frequencies of each of the 1400 corresponding pairs of sensors in 'ro_high' and 'ro_low' is illustrated in Figure 5.9.

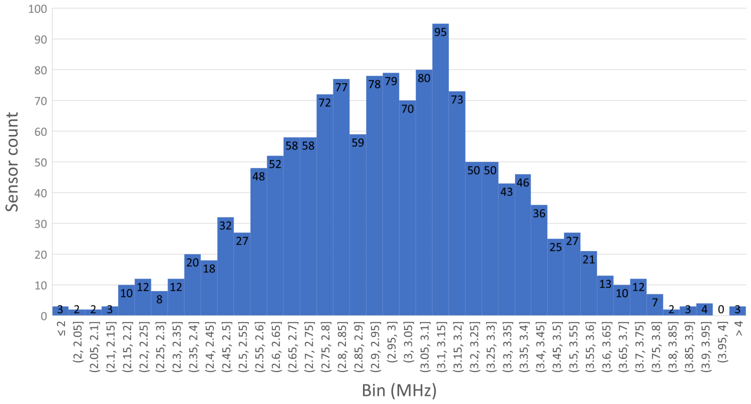


Figure 5.9: Histogram of differences between 'ro_high' and 'ro_low' on 'D591881'

In addition, the maximum, minimum, and average differences between 'ro_high' and 'ro_low' are shown in Table 5.3.

Table 5.3: Maximum, minimum and average difference between 'ro_high' and 'ro_low'

Difference	Avearge	Maximum	Minimum
MHz	2.978044	4.956055	0.341797
%	2.32	3.8	0.25

In this experiment, we chose two of the paths that have the least common number of multiplexers. It might be reasonable to assume that part of the difference that is due to process variation might be similar or even less for the other paths. As for ageing, however, there is a need for more investigation.

5.3 Densification

Ideally, we would like to cover the whole fabric with RO sensors, which would be hugely helpful to get accurate results and investigate ageing effectively. Unfortunately, that would be impractical due to the need for the control circuitry (which takes resources both for logic and routing). Therefore, we needed to know how large the error would be if we extrapolate the ageing of the resources not used in the PV map by using the data from neighboring sensors.

5.3.1 Simulation and Measurement

In this experiment, 'ro_low' is tested on board 'D674906'. One 8×7 sensor patch was created, where the sensors are of type 'ro_low', and are placed next to each other with no unused CLB, DSP blocks, and RAM blocks in between. The locations of sensors on the die are shown as Figure 5.10, where the red box marks one sensor. The simulation results for all 56 sensors are 57.373047MHz. This is slightly different from the simulation results for 1400 'ro_low' sensors (namely, 56.640625MHz), which might be due to longer routes in the denser patch. The measurement results are shown in Table 5.4.

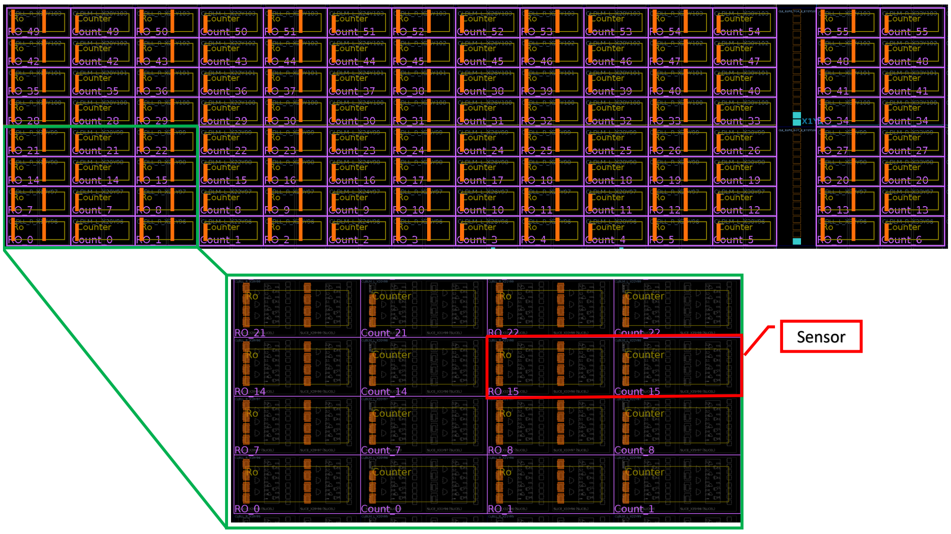


Figure 5.10: An 8 × 7 dense patch

We will use the measurement results to study the approximation error, that is, the difference between the estimated frequencies and the measured frequencies.

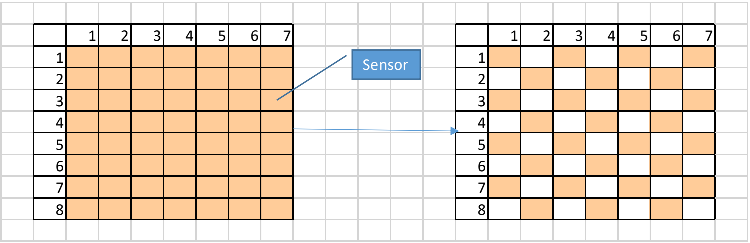


Figure 5.11: One patch covering half resources

In Figure 5.11, the cells marked in yellow denote sensor placement. The values of the resources not used in the PV map can be calculated by linear interpolation using the data from neighboring sensors. It is noted that the value calculated is the mean of neighboring sensors. In this experiment, the calculation results and the corresponding errors are shown in Table 5.5 and 5.6.

Table 5.4: Frequencies (MHz) of 8x7 sensors in a dense patch

	1	2	3	4	5	6	7
1	107.4951	107.7148	107.6172	107.666	107.2998	107.4219	106.4453
2	107.1777	107.5928	108.0322	107.8125	107.666	107.7637	106.6162
3	107.7637	107.8125	107.7637	108.4473	107.959	107.666	106.1523
4	108.2275	107.4951	107.5195	108.3008	108.1055	107.1533	106.7139
5	105.957	106.8848	107.2021	106.3965	105.6641	106.4697	105.2002
6	109.3506	108.1055	107.4951	109.668	109.1064	106.958	106.0791
7	107.4463	108.1787	108.1055	107.5684	107.5928	107.8613	105.9082
8	107.1289	108.1299	107.666	108.0811	107.9102	107.3242	106.4209

Table 5.5: Frequency inferred by linear interpolation

	1	2	3	4	5	6	7
1	107.4951	107.5562	107.6172	107.4585	107.2998	106.8726	106.4453
2	107.5928	107.5928	107.7026	107.8125	107.7881	107.7637	107.7637
3	107.7637	107.7637	107.7637	107.8613	107.959	107.0557	106.1523
4	107.4951	107.4951	107.8979	108.3008	107.7271	107.1533	107.1533
5	105.957	106.5796	107.2021	106.4331	105.6641	105.4321	105.2002
6	108.1055	108.1055	108.8867	109.668	108.313	106.958	106.958
7	107.4463	107.7759	108.1055	107.8491	107.5928	106.7505	105.9082
8	108.1299	108.1299	108.1055	108.0811	107.7026	107.3242	107.3242

Table 5.6: Error in MHz between the inferred frequency and the measured frequency

	1	2	3	4	5	6	7
1	0	-0.15869	0	-0.20752	0	-0.54932	0
2	0.415039	0	-0.32959	0	0.12207	0	1.147461
3	0	-0.04883	0	-0.58594	0	-0.61035	0
4	-0.73242	0	0.378418	0	-0.37842	0	0.439453
5	0	-0.30518	0	0.036621	0	-1.0376	0
6	-1.24512	0	1.391602	0	-0.79346	0	0.878906
7	0	-0.40283	0	0.280762	0	-1.11084	0
8	1.000977	0	0.439453	0	-0.20752	0	0.903321

5.3.2 Discussion

As shown in Table 5.6, the maximum error is 1.3916002MHz (1.2946%), which might or might not be acceptable depending on the application and the degradation degree (that is, how close the measured frequencies are to the simulation results for the slow corner). When such error is acceptable, we can employ linear interpolation to analyze the ageing of the resources not used in the PV map. In case the error between the inferred frequency and the measured frequency is unacceptable for the intended application of this PV mapping method, we can combine two designs shown in Figure 5.12 and use two bit files to produce the whole results.

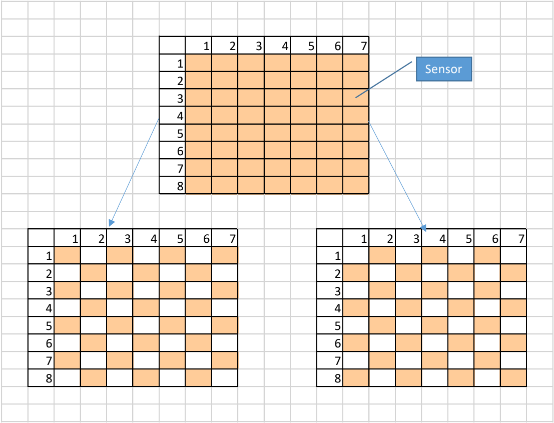


Figure 5.12: Two patches covering half resources

Using multiple placements would then require additional storage of the bit-files, additional FPGA programming time, and additional measurement time.

5.4 Multiple Consecutive Measurements

We define that the precision of this method is how close repeated measurements are to each other, and calculate the precision by using standard deviation for a sample of the population, where N-1 is used in the denominator instead of N. The precision is not only affected by the stability of the RO but also the activation and deactivation of the counters via synchronized FFs. In order to evaluate the precision of this method, we performed an experiment in which the Main Controller repeated the measurements multiple times consecutively. We were also wondering if we would see a decreasing trend in the frequency of an RO when repeated measurements to the same RO or the neighboring ROs cause some local hot spot. In this measurement, 140 sensors ('ro_low') are placed on Board 'D674756' and measured 100 (N_R in Figure 4.6) times consecutively.

In practice, we didn't even need 140 sensors since only one sensor would be sufficient for this experiment. We just wanted to increase the chances of observing anomalies by using more sensors. Also, we cannot use 1400 sensors for this experiment because this experiment requires the implementation of a reinitialization mechanism for the SRLs (that is, resetting the INIT values), which consumes a significant amount of fabric resources. Finally, the extra memory needed to store the data for 100 repetitions would also contribute to more routing congestion, making it impossible to implement 1400 sensors.

5.4.1 Architectural Changes

In this design, 140 sensors are placed by 20×7 on the FPGA. Measurement of 140 sensors is repeated 100 times. The corresponding memory size is also modified, which is shown as Equation 5.3:

$$M_{size} = 16bits * 140sensors * 100times \cong 27.3Kbytes \tag{5.3}$$

Thus, the memory length was set to 32K bytes.

5.4.2 Measurement Results

Table 5.7 shows the difference between the maximum and minimum frequency among 100 measurements for each of the 140 sensors, presented in a 20x7 arrangement.

Table 5.7: Difference between maximum and minimum frequency (MHz) measured for each of the 140 sensors

	1	2	3	4	5	6	7
1	0.097657	0.097656	0.146484	0.146484	0.12207	0.122071	0.170899
2	0.12207	0.12207	0.12207	0.122071	0.12207	0.146484	0.146485
3	0.12207	0.12207	0.122071	0.12207	0.12207	0.12207	0.146485
4	0.146484	0.12207	0.146484	0.146485	0.122071	0.146484	0.146484
5	0.097657	0.097656	0.122071	0.12207	0.146485	0.12207	0.146484
6	0.097657	0.146485	0.097656	0.146484	0.170898	0.170899	0.170898
7	0.12207	0.170898	0.146485	0.146485	0.146484	0.146485	0.146484
8	0.146484	0.097656	0.097656	0.122071	0.146484	0.12207	0.146485
9	0.12207	0.146484	0.12207	0.12207	0.146484	0.146485	0.146484
10	0.122071	0.12207	0.12207	0.146484	0.146484	0.170899	0.146484
11	0.170899	0.12207	0.12207	0.146485	0.097657	0.12207	0.170899
12	0.170899	0.146485	0.097656	0.122071	0.122071	0.097657	0.195312
13	0.12207	0.097656	0.122071	0.097656	0.146484	0.146484	0.146485
14	0.12207	0.146484	0.097657	0.122071	0.122071	0.146484	0.146484
15	0.12207	0.146484	0.12207	0.146484	0.146485	0.146485	0.195312
16	0.097656	0.122071	0.12207	0.122071	0.097657	0.146484	0.146485
17	0.12207	0.097656	0.097656	0.12207	0.122071	0.12207	0.12207
18	0.146484	0.122071	0.12207	0.146485	0.12207	0.12207	0.122071
19	0.12207	0.146484	0.146484	0.122071	0.122071	0.097656	0.122071
20	0.024414	0.195312	0.146484	0.170899	0.170898	0.146484	0.146484

The histogram of standard deviation among 140 sensors in the multiple consecutive measurements is illustrated in Figure 5.13.

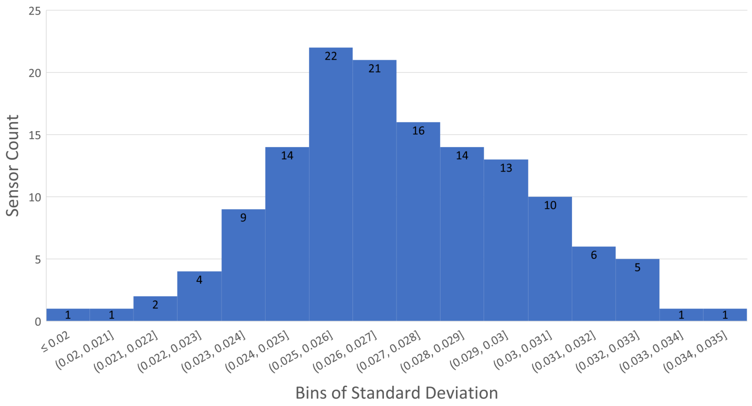
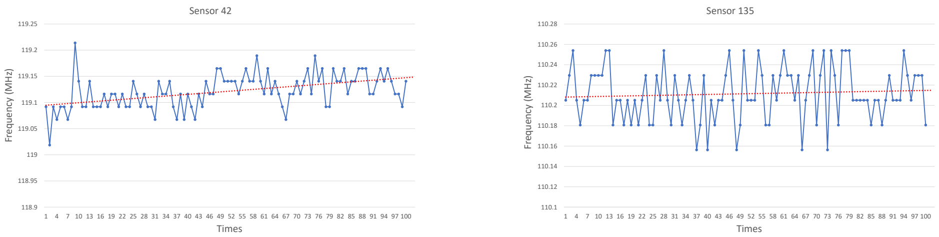


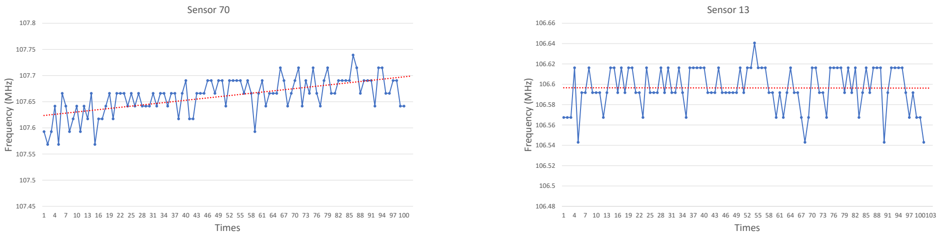
Figure 5.13: The histogram of standard deviation among 140 sensors

In order to see if the chosen 50us measurement time could be too long resulting in overheating, it is necessary to look at the trends on each sensor in the course of multiple measurements. A visual inspection of the trend lines clearly shows that the trends can be decreasing, increasing, or non-increasing. Firstly, we choose sensors (42 and 135) with a maximum difference and minimum difference to investigate. The corresponding line charts are shown in Figure 5.14, where we observe an increasing trend. Secondly, the lines chart of sensors (70 and 13) with maximum standard deviation and minimum standard deviation are shown in Figure 5.15, where we can see an increasing trend on sensor 70 and a slightly decreasing trend on sensor 13. Additionally, we also choose sensor 100 to see the trend randomly. Line chart of frequency on sensor 100 and sensor 0 (reference sensor) are shown in Figure 5.16. We can see an apparent increasing trend on sensor 100 and a slightly increasing trend on sensor 0.



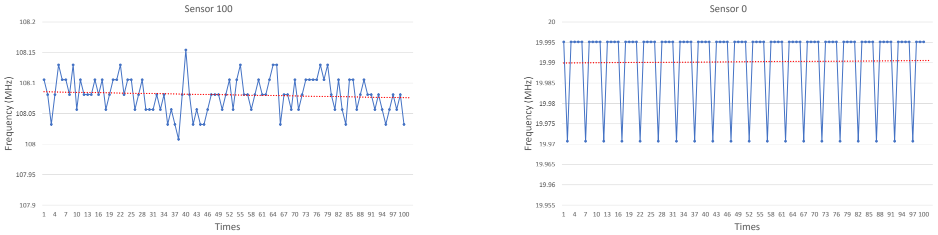
(a) **(b)**

Figure 5.14: Frequency trends of sensor 42 and 135



(a) **(b)**

Figure 5.15: Frequency trends of sensor 70 and 13



(a) **(b)**

Figure 5.16: Frequency trends of sensor 100 and 0

5.4.3 Discussion

Table 5.7 shows that maximum and minimum differences are 195.312KHz (8 ticks) and 97.657KHz (4 ticks) on sensor 42 and sensor 135, corresponding standard deviations are 0.032552MHz and 0.026568MHz respectively. From Figure 5.13, we can see that majority of the measured standard deviations among 140 sensors are between 0.023MHz and 0.033MHz. Given that the frequency of the chosen RO is in the range of hundreds of MHz, we can conclude that our method might have high precision. As shown in Figures 5.14, 5.15 and 5.16, we can see increasing and decreasing trend lines on the six sensors, but we could not conclude that there are particular increasing/decreasing trends among 140 sensors. As we know that, the presence of a decreasing trend would imply that self-heating from the ring oscillator affected the performance. However, the measurement time might not cause a self-heating phenomenon.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

This thesis work surveys several effective ageing monitoring methods presented in previous work. The most suitable method, RO method, is selected and utilized to create a PV map to monitor the ageing, with the limitation of testing only CLBs and switch boxes. Four sensor types and architecture for sensor data collection have been implemented. To analyze the degree of degradation more accurately, this method monitors more fabric resources in each ring oscillator, as compared to similar RO-based approaches.

In this thesis work, we have achieved the deployment of 1400 sensors, which should be compared to the maximum possible number of 1774 sensors for the suggested sensor types. It is very challenging to place more than 1400 sensors due to routing congestion. In addition, LUT inputs have been specified in the ROs, resulting in more LUT-internal resources included in the RO circuit and ensure sensors uniformity, compared to LUT inputs not being not specified in previous work. We have performed multiple experiments to validate the operation of the method, as well as gain insights into some aspects such as the precision of the measurements. All experiments are performed on Digilent Nexys4 boards featuring 28nm Xilinx Artix7 FPGAs. In the first experiment, we used the proposed method to perform measurements on 20 Nexys4 boards (which have been in use since several years back) to observe the frequency variations across the same fabric, as well as for the same sensor across different boards. The results confirm that the method is capable of detecting delay differences (manifested as frequency differences in the measurements) with relatively high accuracy and precision. We note that since we did not employ accelerated ageing, in the measured differences we cannot distinguish between the contributions of process variation and ageing.

To investigate the error in approximating the ageing of the resources not used in the PV map by using the data from neighboring sensors, an experiment with a dense patch of 56 sensors were carried out, in which we extrapolated the frequency of the missing sensors by using the data from neighboring sensors. The results show that the maximum error is 1.29%. Besides, to measure the precision of the method, multiple consecutive measurements with 140 sensors were conducted and the results show that the maximum difference is 0.164% among different measurements for the same sensor.

6.2 Future Work

This work is based on the study of previous works. As such, there are many limitations and potential for future work. Firstly, it can be extended to test the ageing degree of the DSP blocks and memories. The SR and TP methods can be additionally included implemented. For example, the TP method can be used to test the DSP blocks. Secondly, in our experiments, we activate the sensors one by one in a row, which might induce heat on the neighboring sensor. We can modify the activation order of sensors to spread the heat. In addition, we can find other interpolation methods to obtain more accurate results for the unused resources. Finally, we could consider the effect of voltage level and ambient temperature when observing ageing trends in a series of far-apart measurements done for the same product. Notwithstanding, this work provides a very useful direction for the researchers who are planning to investigate ageing of FPGAs.

References

[1] A. Amouri, “Degradation in fpgas: Monitoring, modeling and mitigation,” Ph.D. dissertation, 2015.

[2] “7 series fpgas configurable logic block user guide,” September 27, 2016. [Online]. Available: https://www.xilinx.com/support/documentation/user_guides/ug474_7Series_CLB.pdf

[3] E. A. Stott, J. S. Wong, P. Sedcole, and P. Y. Cheung, “Degradation in fpgas: Measurement and modelling,” ser. FPGA ’10. New York, NY, USA: Association for Computing Machinery, 2010, pp. 229–238. [Online]. Available: https://doi.org/10.1145/1723112.1723152

[4] J. Li and J. Lach, “Negative-skewed shadow registers for at-speed delay variation characterization,” in *2007 25th International Conference on Computer Design*, 2007, pp. 354–359.

[5] A. Amouri and M. Tahoori, “A low-cost sensor for aging and late transitions detection in modern fpgas,” in *2011 21st International Conference on Field Programmable Logic and Applications*, 2011, pp. 329–335.

[6] J. S. J. Wong, P. Sedcole, and P. Y. K. Cheung, “A transition probability based delay measurement method for arbitrary circuits on fpgas,” in *2008 International Conference on Field-Programmable Technology*, 2008, pp. 105–112.

[7] M. Ebrahimi and Z. Navabi, “Selecting representative critical paths for sensor placement provides early fpga aging information,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 10, pp. 2976–2989, 2020.

[8] A. Amouri, F. Bruguier, S. Kiamehr, P. Benoit, L. Torres, and M. Tahoori, “Aging effects in fpgas: an experimental analysis,” in *2014 24th International Conference on Field Programmable Logic and Applications (FPL)*, 2014, pp. 1–4.

[9] N. Rahmanikia, A. Amiri, H. Noori, and F. Mehdipour, “Performance evaluation metrics for ring-oscillator-based temperature sensors on fpgas: A quality factor,” *Integration*, vol. 57, pp. 81–100, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167926016301900

[10] K. Maragos, E. Taka, G. Lentaris, I. Stratakos, and D. Soudris, "Analysis of performance variation in 16nm finfet fpga devices," in *2019 29th International Conference on Field Programmable Logic and Applications (FPL)*, 2019, pp. 38–44.

[11] P. Sedcole and P. Y. K. Cheung, "Within-die delay variability in 90nm fpgas and beyond," in *2006 IEEE International Conference on Field Programmable Technology*, 2006, pp. 97–104.

[12] T. Tuan, A. Lesea, C. Kingsley, and S. Trimberger, "Analysis of within-die process variation in 65nm fpgas," in *2011 12th International Symposium on Quality Electronic Design*, 2011, pp. 1–5.

[13] K. M. Zick and J. P. Hayes, "On-line sensing for healthier fpga systems," in *Proceedings of the 18th Annual ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, ser. FPGA '10. New York, NY, USA: Association for Computing Machinery, 2010, p. 239–248.

[14] H. Yu, Q. Xu, and P. H. Leong, "Fine-grained characterization of process variation in fpgas," in *2010 International Conference on Field-Programmable Technology*, 2010, pp. 138–145.

[15] F. Bruguier, P. Benoit, P. Maurine, and L. Torres, "A new process characterization method for fpgas based on electromagnetic analysis," in *2011 21st International Conference on Field Programmable Logic and Applications*, 2011, pp. 20–23.

[16] T. Nylund, "Degradation of integrated circuits due to scaling in fpgas," 2015.

[17] M. Naouss and F. Marc, "Modelling delay degradation due to nbti in fpga look-up tables," in *2016 26th International Conference on Field Programmable Logic and Applications (FPL)*, 2016, pp. 1–4.

[18] Naouss, Mohammad and Marc, François , "Fpga lut delay degradation due to hci: Experiment and simulation results," vol. 64. Elsevier, 2016, pp. 31–35.

[19] M. Slimani, K. Benkalaia, and L. Naviner, "Analysis of ageing effects on artix7 xilinx fpga," *Microelectronics Reliability*, vol. 76-77, pp. 168–173, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0026271417302901

[20] J. S. J. Wong, P. Sedcole, and P. Y. K. Cheung, "Self-characterization of combinatorial circuit delays in fpgas," in *2007 International Conference on Field-Programmable Technology*, 2007, pp. 17–23.

[21] P. Sedcole, J. S. Wong, and P. Y. K. Cheung, "Characterisation of fpga clock variability," in *2008 IEEE Computer Society Annual Symposium on VLSI*, 2008, pp. 322–328.

[22] M. Valdés, J. Freijedo, M. J. Moure, J. J. Rodríguez-Andina, J. Semião, F. Vargas, I. C. Teixeira, and J. P. Teixeira, "Programmable sensor for on-line checking of signal integrity in fpga-based systems subject to aging effects," in *2011 12th Latin American Test Workshop (LATW)*, 2011, pp. 1–7.

[23] M. D. Valdes-Peña, J. Fernández Freijedo, M. J. Moure Rodríguez, J. J. Rodríguez-Andina, J. Semião, I. M. C. Teixeira, J. P. C. Teixeira, and F. Vargas, "Design and validation of configurable online aging sensors in nanometer-scale fpgas," *IEEE Transactions on Nanotechnology*, vol. 12, no. 4, pp. 508–517, 2013.

[24] M. Ebrahimi, Z. Ghaderi, E. Bozorgzadeh, and Z. Navabi, “Path selection and sensor insertion flow for age monitoring in fpgas,” in *2016 Design, Automation Test in Europe Conference Exhibition (DATE)*, 2016, pp. 792–797.

[25] Z. Ghaderi, M. Ebrahimi, Z. Navabi, E. Bozorgzadeh, and N. Bagherzadeh, “Sensible: A highly scalable sensor design for path-based age monitoring in fpgas,” *IEEE Transactions on Computers*, vol. 66, no. 5, pp. 919–926, 2017.

[26] E. Stott, J. S. J. Wong, and P. Y. K. Cheung, “Degradation analysis and mitigation in fpgas,” in *2010 International Conference on Field Programmable Logic and Applications*, 2010, pp. 428–433.

Appendix A

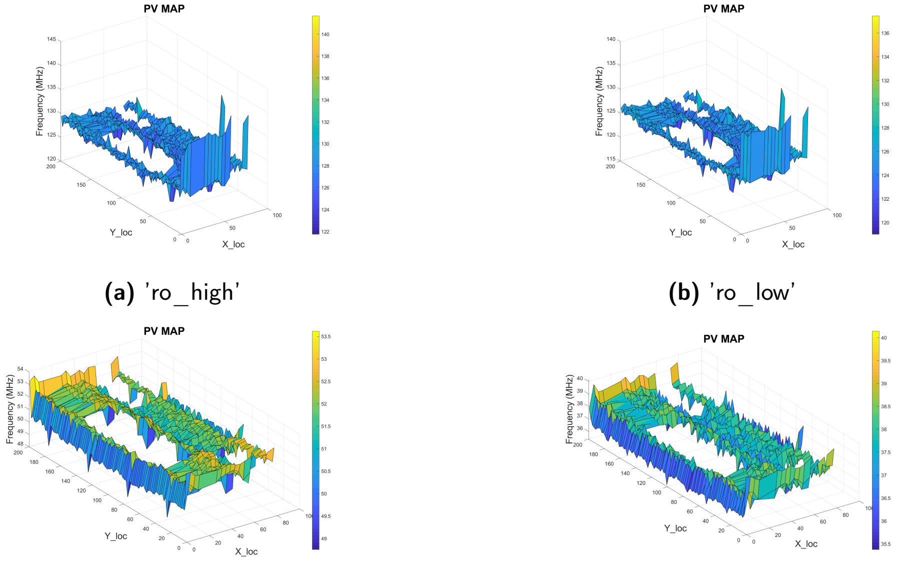
FPGA Board Information

Table A.1: FPGA board information

Serial Number	D591881	D592141	D592199	D592246	D592306
Week	ACX1349	ACX1325	ACX1349	ACX1325	ACX1345
	D4656626A	D4582794A	D4656626A	D4582794A	D4635423A
Serial Number	D592350	D592374	D592378	D592387	D292414
Week	ACX1349	ACX1401	ACX1349	ACX1345	ACX1329
	D4656626A	D4677144A	D4656626A	D4635423A	D4592923A
Serial Number	D592420	D592429	D592519	D592987	D674756
Week	ACX1345	ACX1345	ACX1401	ACX1325	ABX1437
	D4635423A	D4635423A	D4677144A	D4582794A	D5026107A
Serial Number	D674766	D6747777	D674897	D674906	D675088
Week	ABX1437	ABX1437	ABX1437	ABX1437	ABX1433
	D5026107A	D5016107A	D5024810A	D5026107A	D5022061A

Appendix B

PV Maps



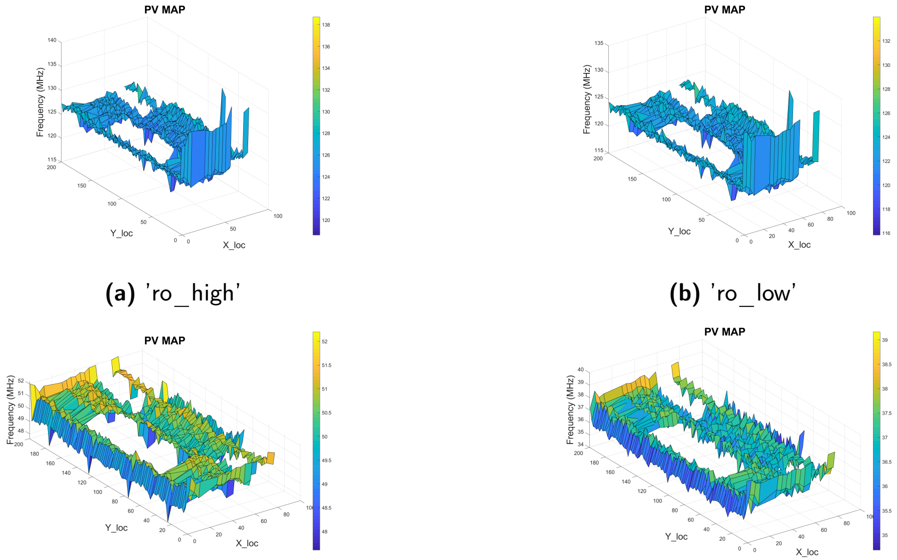
(a) 'ro_high'

(b) 'ro_low'

(c) 'ro_8_cc2_8_cc2'

(d) 'ro_16_8_cc2'

Figure B.1: PV Maps of Board D591881



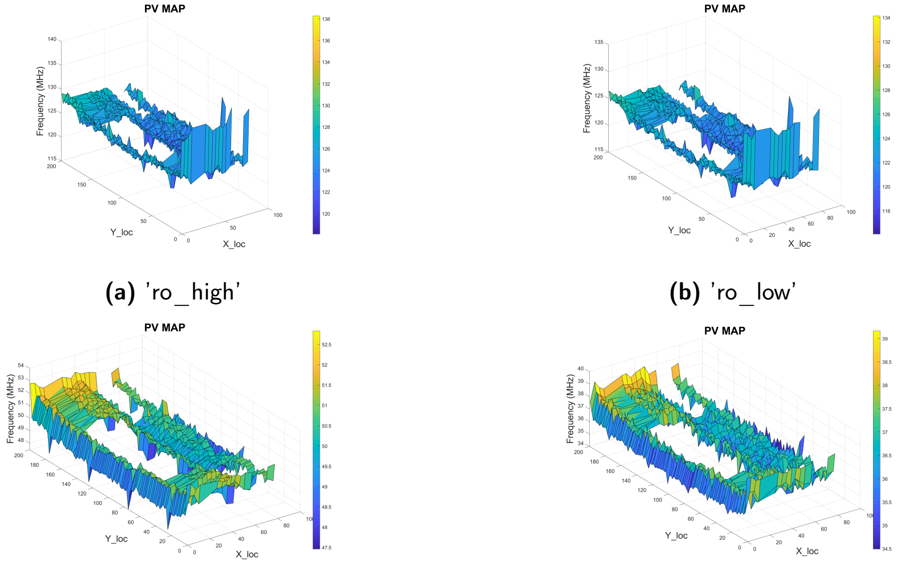
(a) 'ro_high'

(b) 'ro_low'

(c) 'ro_8_cc2_8_cc2'

(d) 'ro_16_8_cc2'

Figure B.2: PV Maps of Board D592141



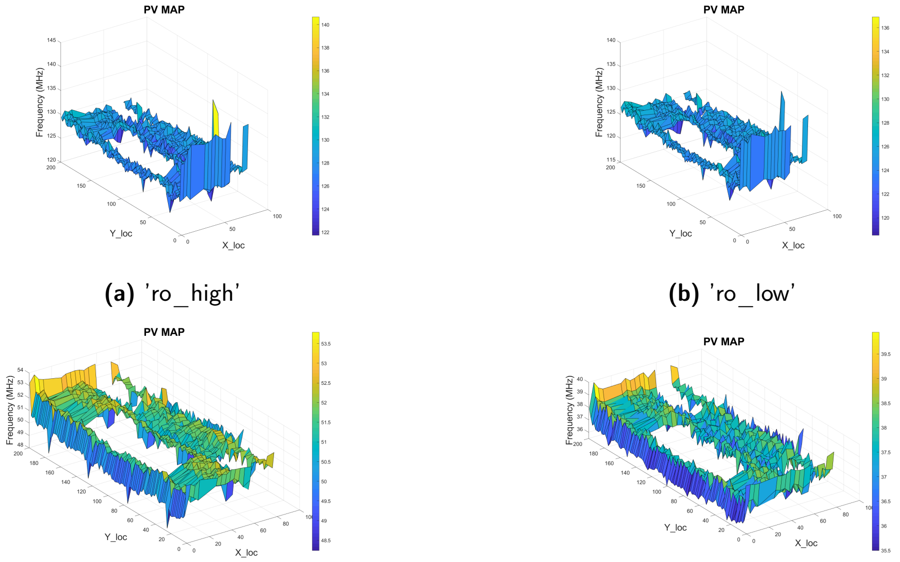
(a) 'ro_high'

(b) 'ro_low'

(c) 'ro_8_cc2_8_cc2'

(d) 'ro_16_8_cc2'

Figure B.3: PV Maps of Board D592199



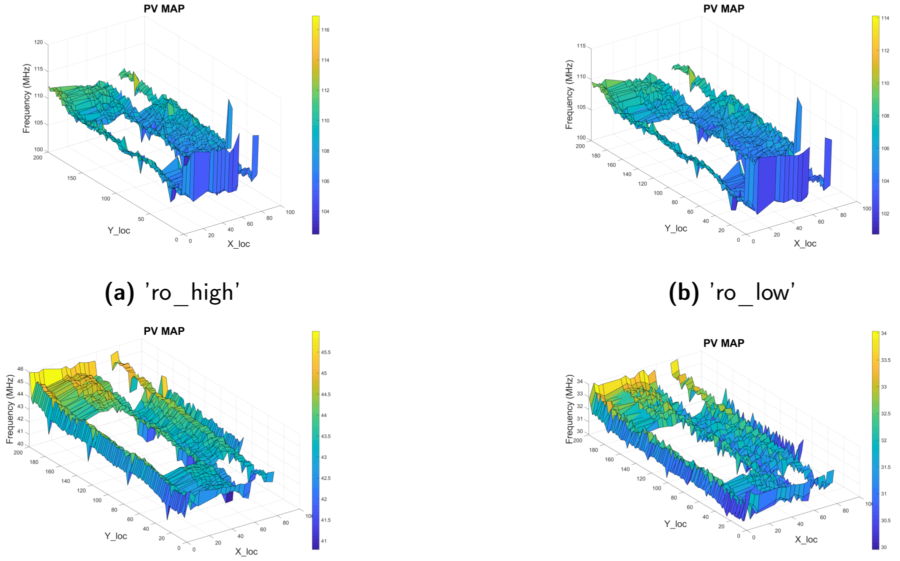
(a) 'ro_high'

(b) 'ro_low'

(c) 'ro_8_cc2_8_cc2'

(d) 'ro_16_8_cc2'

Figure B.4: PV Maps of Board D592246



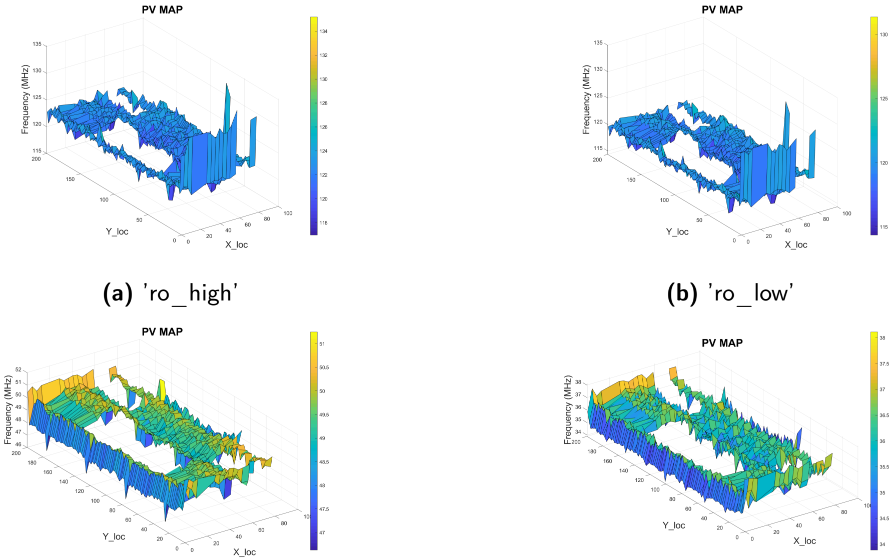
(a) 'ro_high'

(b) 'ro_low'

(c) 'ro_8_cc2_8_cc2'

(d) 'ro_16_8_cc2'

Figure B.5: PV Maps of Board D5923066



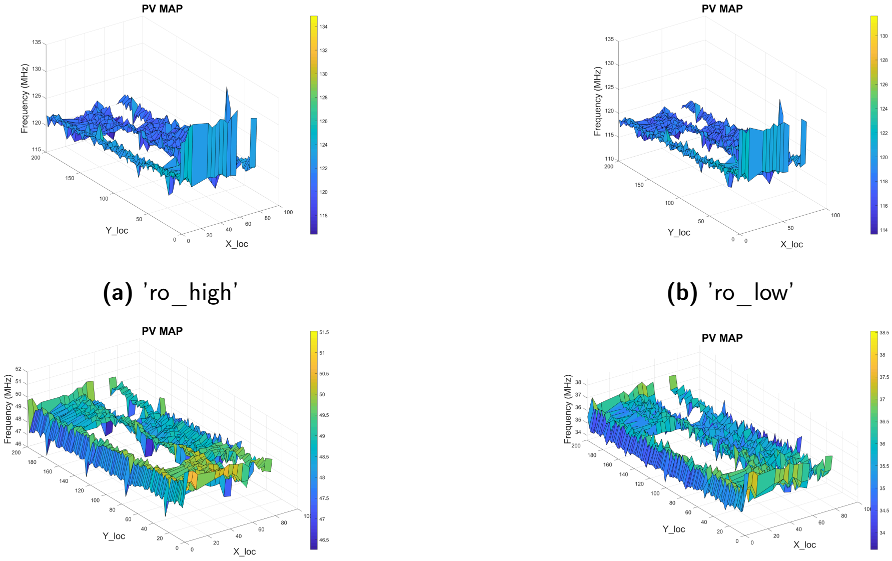
(a) 'ro_high'

(b) 'ro_low'

(c) 'ro_8_cc2_8_cc2'

(d) 'ro_16_8_cc2'

Figure B.6: PV Maps of Board D592350



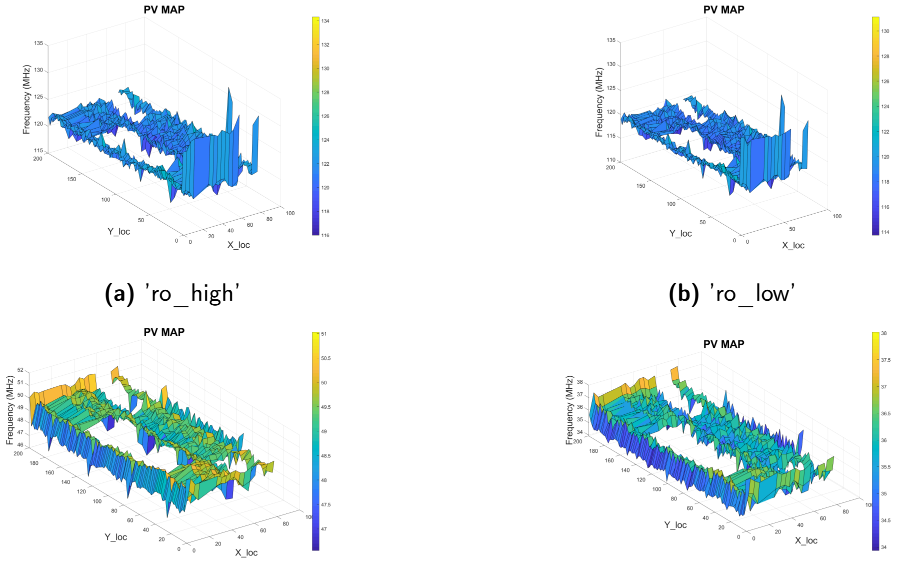
(a) 'ro_high'

(b) 'ro_low'

(c) 'ro_8_cc2_8_cc2'

(d) 'ro_16_8_cc2'

Figure B.7: PV Maps of Board D592374



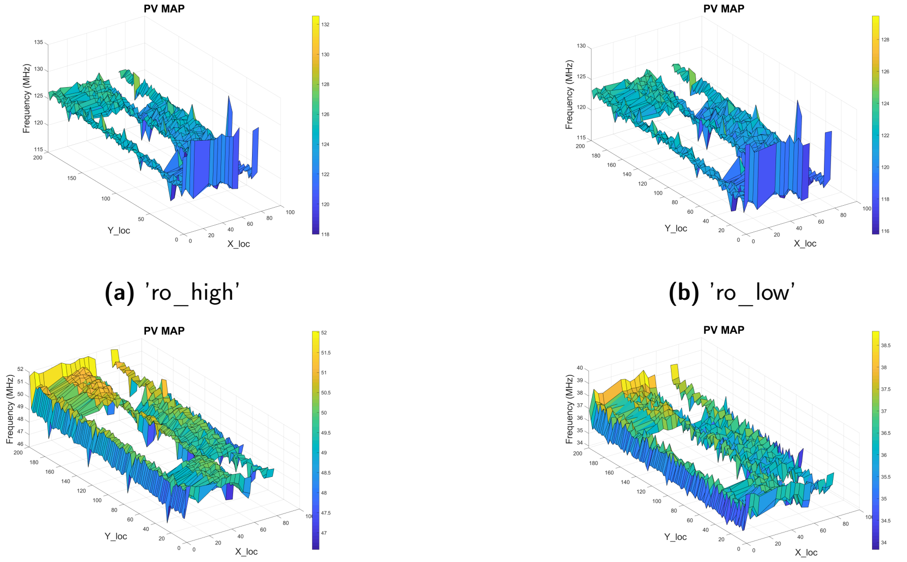
(a) 'ro_high'

(b) 'ro_low'

(c) 'ro_8_cc2_8_cc2'

(d) 'ro_16_8_cc2'

Figure B.8: PV Maps of Board D592378



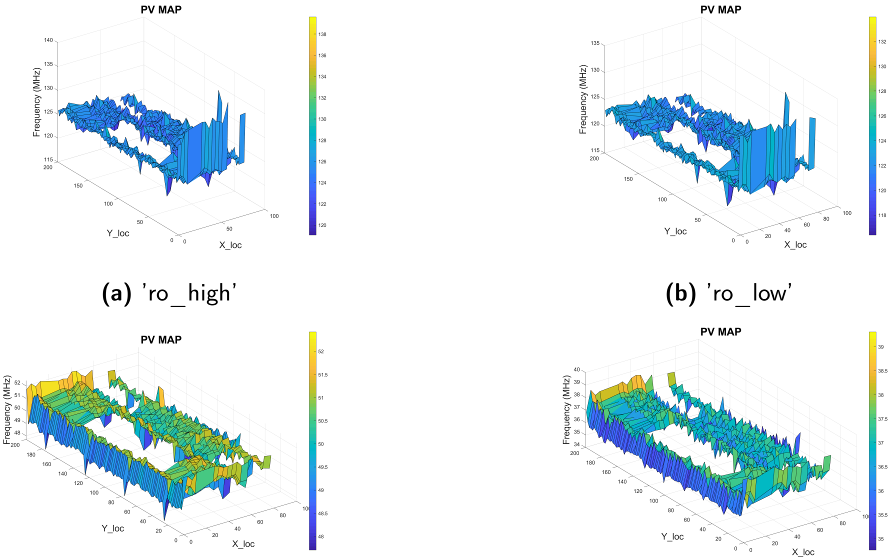
(a) 'ro_high'

(b) 'ro_low'

(c) 'ro_8_cc2_8_cc2'

(d) 'ro_16_8_cc2'

Figure B.9: PV Maps of Board D592387



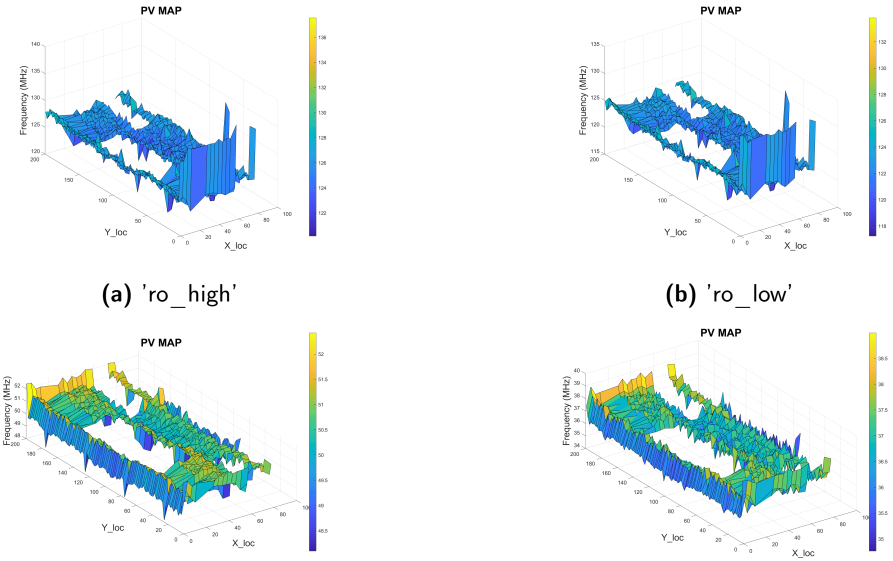
(a) 'ro_high'

(b) 'ro_low'

(c) 'ro_8_cc2_8_cc2'

(d) 'ro_16_8_cc2'

Figure B.10: PV Maps of Board D592414



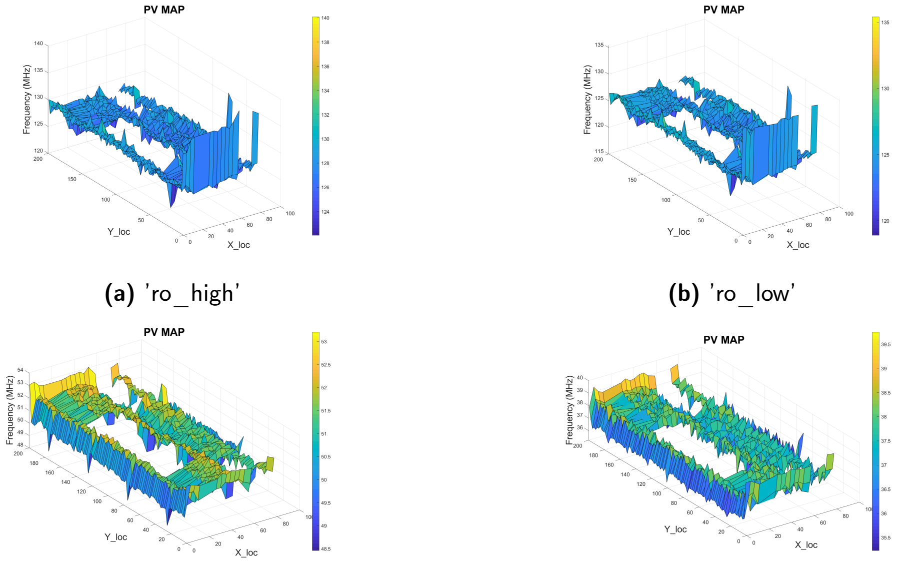
(a) 'ro_high'

(b) 'ro_low'

(c) 'ro_8_cc2_8_cc2'

(d) 'ro_16_8_cc2'

Figure B.11: PV Maps of Board D592420



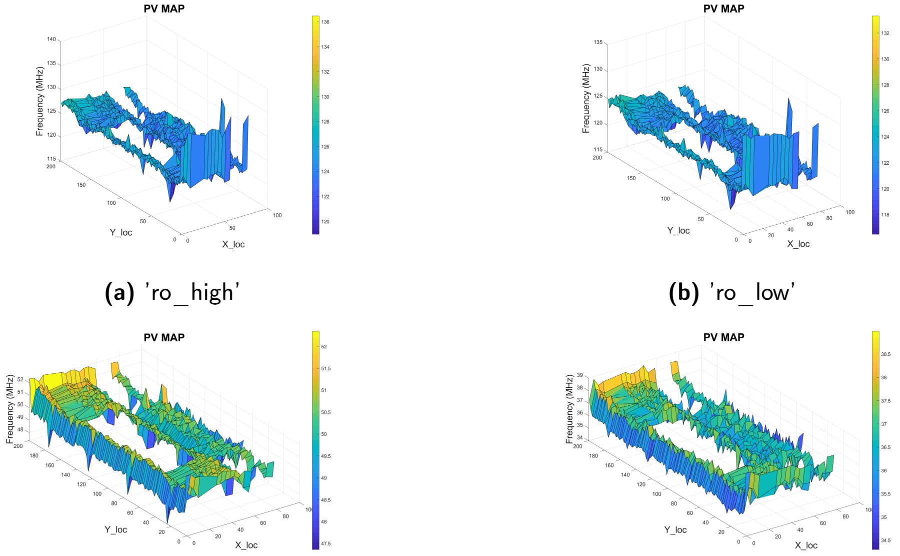
(a) 'ro_high'

(b) 'ro_low'

(c) 'ro_8_cc2_8_cc2'

(d) 'ro_16_8_cc2'

Figure B.12: PV Maps of Board D592429



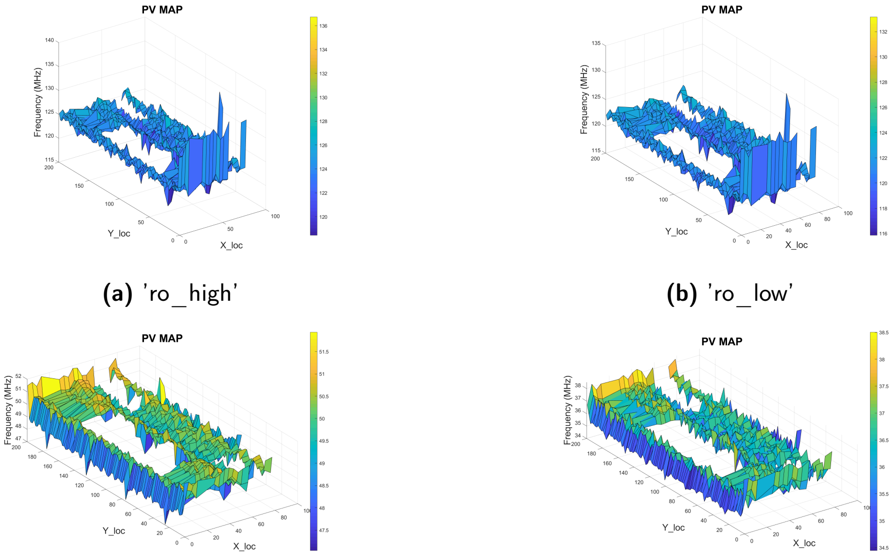
(a) 'ro_high'

(b) 'ro_low'

(c) 'ro_8_cc2_8_cc2'

(d) 'ro_16_8_cc2'

Figure B.13: PV Maps of Board D592519



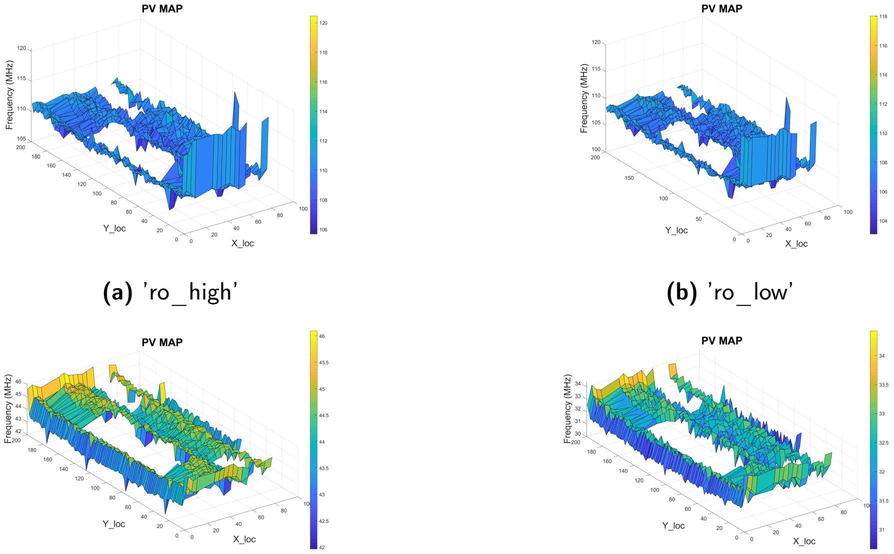
(a) 'ro_high'

(b) 'ro_low'

(c) 'ro_8_cc2_8_cc2'

(d) 'ro_16_8_cc2'

Figure B.14: PV Maps of Board D592987



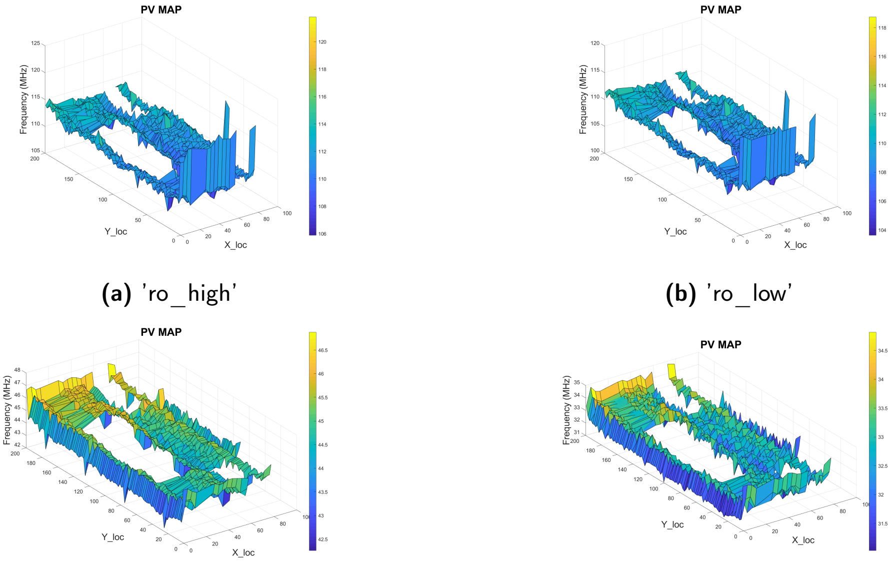
(a) 'ro_high'

(b) 'ro_low'

(c) 'ro_8_cc2_8_cc2'

(d) 'ro_16_8_cc2'

Figure B.15: PV Maps of Board D674756



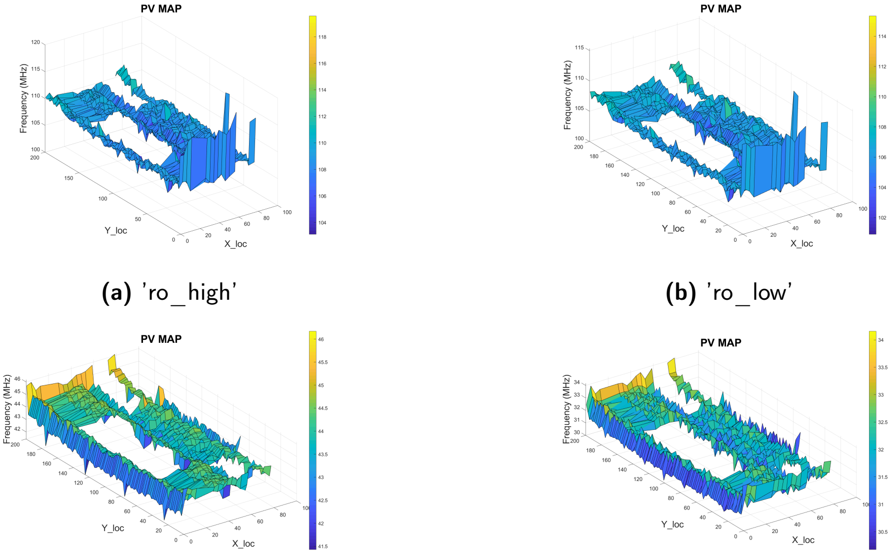
(a) 'ro_high'

(b) 'ro_low'

(c) 'ro_8_cc2_8_cc2'

(d) 'ro_16_8_cc2'

Figure B.16: PV Maps of Board D674766



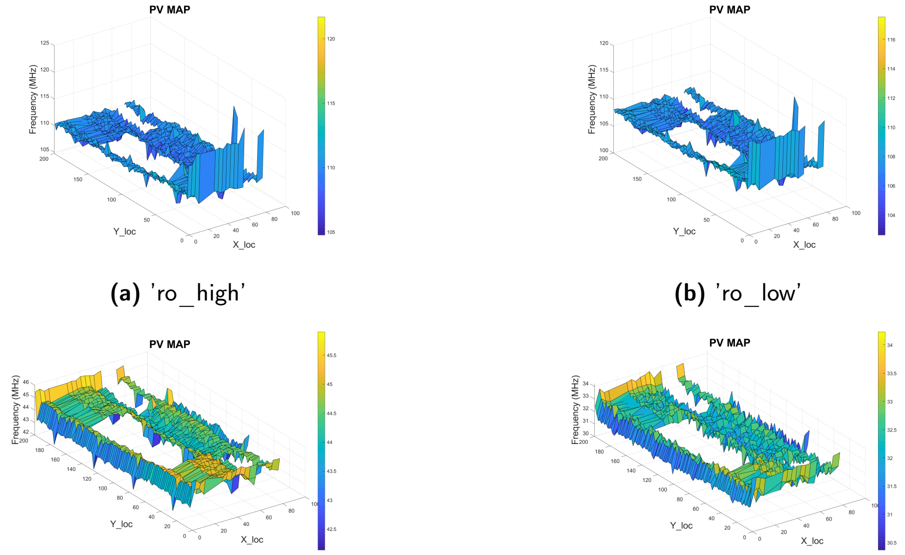
(a) 'ro_high'

(b) 'ro_low'

(c) 'ro_8_cc2_8_cc2'

(d) 'ro_16_8_cc2'

Figure B.17: PV Maps of Board D674777



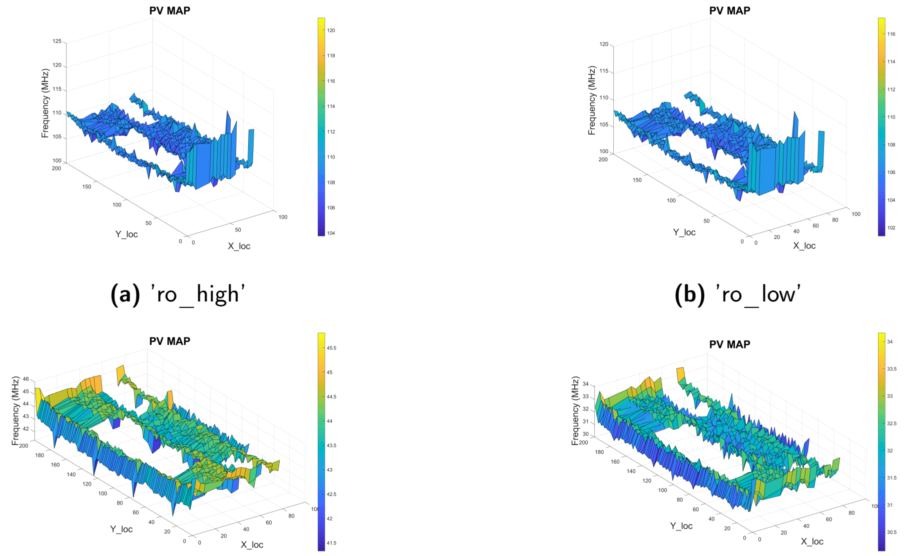
(a) 'ro_high'

(b) 'ro_low'

(c) 'ro_8_cc2_8_cc2'

(d) 'ro_16_8_cc2'

Figure B.18: PV Maps of Board D674897



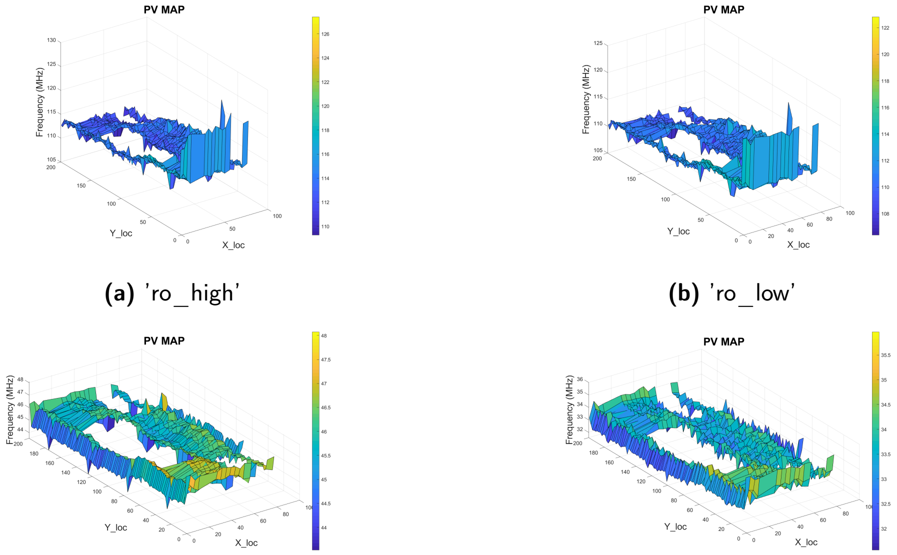
(a) 'ro_high'

(b) 'ro_low'

(c) 'ro_8_cc2_8_cc2'

(d) 'ro_16_8_cc2'

Figure B.19: PV Maps of Board D674906



(a) 'ro_high'

(b) 'ro_low'

(c) 'ro_8_cc2_8_cc2'

(d) 'ro_16_8_cc2'

Figure B.20: PV Maps of Board D675088

Printed by Tryckeriet i E-huset, Lund 2021

LUND UNIVERSITY

Series of Master's theses
Department of Electrical and Information Technology
LU/LTH-EIT 2021-851
http://www.eit.lth.se