ORTA DOĞU TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ MÜDÜRLÜĞÜ

TEZ ŞABLONU ONAY FORMU
THESIS TEMPLATE CONFIRMATION FORM

1. Şablonda verilen yerleşim ve boşluklar değiştirilmemelidir.
2. **Jüri tarihi** Başlık Sayfası, İmza Sayfası, Abstract ve Öz'de ilgili yerlere yazılmalıdır.
3. İmza sayfasında jüri üyelerinin unvanları doğru olarak yazılmalıdır. Tüm imzalar **mavi pilot kalemle** atılmalıdır.
4. **Disiplinlerarası** programlarda görevlendirilen öğretim üyeleri için jüri üyeleri kısmında tam zamanlı olarak çalıştıkları anabilim dalı başkanlığının ismi yazılmalıdır. Örneğin: bir öğretim üyesi Biyoteknoloji programında görev yapıyor ve biyoloji bölümünde tam zamanlı çalışıyorsa, İmza sayfasına biyoloji bölümü yazılmalıdır. İstisnai olarak, disiplinler arası program başkanı ve tez danışmanı için disiplinlerarası program adı yazılmalıdır.
5. Tezin **son sayfasının sayfa** numarası Abstract ve Öz'de ilgili yerlere yazılmalıdır.
6. Bütün chapterlar, referanslar, ekler ve CV sağ sayfada başlamalıdır. Bunun için **kesmeler** kullanılmıştır. **Kesmelerin kayması** fazladan boş sayfaların oluşmasına sebep olabilir. Bu gibi durumlarda paragraf (¶) işaretine tıklayarak kesmeleri görünür hale getirin ve yerlerini **kontrol edin**.
7. Figürler ve tablolar kenar boşluklarına taşmamalıdır.
8. Şablonda yorum olarak eklenen uyarılar dikkatle okunmalı ve uygulanmalıdır.
9. Tez yazdırılmadan önce PDF olarak kaydedilmelidir. Şablonda yorum olarak eklenen uyarılar PDF dokümanında yer almamalıdır.
10. Tez taslaklarının kontrol işlemleri tamamlandığında, bu durum öğrencilere METU uzantılı öğrenci e-posta adresleri aracılığıyla duyurulacaktır.
11. Tez yazım süreci ile ilgili herhangi bir sıkıntı yaşarsanız, Sıkça Sorulan Sorular (SSS) sayfamızı ziyaret ederek yaşadığınız sıkıntıyla ilgili bir çözüm bulabilirsiniz.

1. Do not change the spacing and placement in the template.
2. Write **defense date** to the related places given on Title page, Approval page, Abstract and Öz.
3. Write the titles of the examining committee members correctly on Approval Page. **Blue ink** must be used for all signatures.
4. For faculty members working in **interdisciplinary programs**, the name of the department that they work full-time should be written on the Approval page. For example, if a faculty member staffs in the biotechnology program and works full-time in the biology department, the department of biology should be written on the approval page. Exceptionally, for the interdisciplinary program chair and your thesis supervisor, the interdisciplinary program name should be written.
5. Write **the page number of the last page** in the related places given on Abstract and Öz pages.
6. All chapters, references, appendices and CV must be started on the right page. **Section Breaks** were used for this. **Change in the placement** of section breaks can result in extra blank pages. In such cases, make the section breaks visible by clicking paragraph (¶) mark and **check their position.**
7. All figures and tables must be given inside the page. Nothing must appear in the margins.
8. All the warnings given on the comments section through the thesis template must be read and applied.
9. Save your thesis as pdf and Disable all the comments before taking the printout.
10. This will be announced to the students via their METU students e-mail addresses when the control of the thesis drafts has been completed.
11. If you have any problems with the thesis writing process, you may visit our Frequently Asked Questions (FAQ) page and find a solution to your problem.

☒ **Yukarıda bulunan tüm maddeleri okudum, anladım ve kabul ediyorum. / I have read, understand and accept all of the items above.**

Name : Şener
Surname : Yılmaz
E-Mail : sener.yilmaz@metu.edu.tr
Date : 19.08.2022
Signature : ________________________

THERMAL IMAGING BASED ON MECHANICAL VIBRATIONS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ŞENER YILMAZ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
MECHANICAL ENGINEERING

AUGUST 2022

Approval of the thesis:

THERMAL IMAGING BASED ON MECHANICAL VIBRATIONS

submitted by **ŞENER YILMAZ** in partial fulfillment of the requirements for the degree of **Master of Science** in **Mechanical Engineering, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. M. A. Sahir Arıkan
Head of the Department, **Mechanical Engineering**

Assoc. Prof. Dr. Kıvanç Azgın
Supervisor, **Mechanical Eng. Dept., METU**

Examining Committee Members:

Assoc. Prof. Dr. Mehmet Bülent Özer
Mechanical Eng. Dept., METU

Assoc. Prof. Dr. Kıvanç Azgın
Mechanical Eng. Dept., METU

Assoc. Prof. Dr. Ender Yıldırım
Mechanical Eng. Dept., METU

Assist. Prof. Dr. Hakan Çalışkan
Mechanical Eng. Dept., METU

Assist. Prof. Dr. Emir Kutluay
Mechanical Eng. Dept., Hacettepe University

Date: 19.08.2022

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name Last name : Şener Yılmaz

Signature :

ABSTRACT

THERMAL IMAGING BASED ON MECHANICAL VIBRATIONS

Yılmaz, Şener
Master of Science, Mechanical Engineering
Supervisor: Assoc. Prof. Dr. Kıvanç Azgın

August 2022, 102 sayfa

The thesis proposes a digital, resonance readout method based on a lock-in based digital phase locked loop (DPLL) mechanism, which is designed, simulated, implemented and tested using a Xilinx made Field Programmable Gate Array (FPGA). Implementation is performed using a hardware descriptive language (VHDL) on low level. Certain digital signal processing algorithms such as lock-in detection, DPLL, DDS and CORDIC are implemented, simulated and tested. Moreover, the design is shown to be capable of resonating both single and multiple resonators simultaneously in real time. As a proof of concept, a low-cost RLC based, low quality factor resonant position sensor with sub micron resolution is tested along with high quality factor quartz crystals for demonstration of simultaneous multi device operation capability. Finally, it is shown that the proposed method can be used for detection of temperature using capacitive MEMS resonant microbolometer arrays, as well as any kind of resonance based sensor.

Keywords: Resonant MEMS Temperature Sensing, Lock-in Amplifier, CORDIC, Direct Digital Synthesis, Digital Signal Processing.

ÖZ

MEKANİK TİTREŞİM TABANLI TERMAL GÖRÜNTÜLEME

Yılmaz, Şener
Yüksek Lisans, Makina Mühendisliği
Tez Yöneticisi: Doç. Dr. Kıvanç Azgın

Ağustos 2022, 102 pages

Bu tez, kenetlemeli yükselteç (lock-in) tabanlı dijital faza kenetlenme döngüsü (DPLL) kullanmak suretiyle dijital bir rezonans/tınlaşım okuma mekanizması öne sürmektedir. Bu dijital mekanizmanın tasarımı, simülasyonu, gerçeklendirilmesi ve test süreci, Xilinx yapımı bir kullanıcı programlamalı kapı dizisi (FPGA) üzerinden tamamlanmıştır. Tasarımın gerçeklendirilmesi donanım tanımlamalı bir dil olan VHDL kullanılarak çip seviyesinde yapılmıştır. Lock-in tespiti, DPLL, direkt dijital sentezleme (DDS) ve CORDIC, FPGA içinde gerçeklendirilmesi, simülasyonu ve gerçek zamanlı testi yapılan başlıca dijital sinyal işleme algoritmalarındandır. Bunlara ek olarak, tasarımın, tek bir tınlayıcı haricinde, birden çok tınlayıcıyı da eş zamanlı olarak gerçek zamanlı tınlaşım döngüsü içinde tutabilmeye muktedir olduğu gösterilmiştir. Öne sürülen konseptin kanıtlanması açısından, öncelikle düşük maliyetli ve kalite faktörlü, RLC devresi tabanlı bir pozisyon ölçüm aygıtı önerilmiştir. Mikrometre altında çözünürlüğe sahip olduğu kanıtlanan bu pozisyon sensöründen sonra ise, yüksek kalite faktörüne sahip oldukları bilinen quartz kristalleri, tasarımın birden çok cihazı eş zamanlı olarak tınlama döngüsünde tutabilme kabiliyetini göstermek için kulanılmıştır. Son olarak, öne sürülen metot ile, kapasitif ve rezonans tabanlı MEMS mikrobolometre dizininin sürülebileceği, ve bu yöntemin sıcaklık tespitinde kullanılabileceği gösterilmiştir. Esasen, bu tasarım vasıtasıyla tınlaşım tabanlı herhangi bir sensörün veya sensör dizininin sürülmesi mümkün olmaktadır.

Anahtar Kelimeler: MEMS Rezonans Tabanlı Isıl Ölçüm, Kenetlemeli Yükselteç, CORDIC, Direkt Dijital Sentezleme, Dijital Sinyal İşleme.

To

My lovely mother, Nergiz and my little sister, Selin

My significant other Tutku

My grandparents Günseli & Mehmet

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my supervisor Assoc. Prof. Dr. Kıvanç Azgın for his guidance, advice, criticism, patience, encouragements and insight throughout the research. Relaying his profound understanding of natural and applied sciences, along with their essence, he has paved the way for the success of this thesis.

My colleague and friend Halim Polat, with whom I have experienced this Master’s journey, deserves a special thanks.

TABLE OF CONTENTS

ABSTRACT v

ÖZ vi

ACKNOWLEDGMENTS ix

TABLE OF CONTENTS x

LIST OF FIGURES xii

LIST OF ABBREVIATIONS xiv

CHAPTERS

1 INTRODUCTION 1

1.1 Overview of the Resonance Based MEMS Sensors 2

1.2 Overview of the Thermal Detectors 4

1.3 Field Programmable Gate Array Technology 6

1.4 Digital PLL (DPLL) for Resonant MEMS Sensor Systems 8

1.5 Coherent Sampling 9

1.6 Research Objectives and Thesis Organization 12

2 DESIGN 15

2.1 Theory and Implementation 15

2.1.1 Heterodyning 16

2.1.2 Phase Locked Loop (PLL) 18

2.1.3 Lock-In Detection 23

2.1.4 Direct Digital Synthesis (DDS) 29

2.1.5 Serial Peripheral Interface (SPI) 35

2.1.6 BRAM Based Moving Average Low Pass Filter 37

2.1.7 CIC Filter Based Low Pass Filter 44

2.1.8 COordinate Rotation Digital Computer (CORDIC).................................48

2.1.9 Closed Loop Controller 55

3 CHARACTERIZATION AND TESTS 63

3.1 A Simple RLC Circuit Based Position Sensor as a Proof of Concept 63

3.1.1 System Setup and Circuit Design 65

3.1.2 Position Measurement 66

3.2 Characterization of a Resonator and Implementation of DPLL for Multiple DUTs 77

3.2.1 Frequency Response of a Quartz Crystal 77

3.2.2 Resonating Three Quartz Crystals with Distinct Frequencies...................80

4 CONCLUSION AND FUTURE WORKS 87

REFERENCES 91

APPENDICES

A. LUT Generator for DDS 97

B. Frequency Response of the Moving Average Low Pass Filter........................99

C. LUT for CORDIC Algorithm 100

LIST OF FIGURES

FIGURES

Figure 1.1. Building Blocks of an FPGA [20] 7
Figure 1.2. Resonant MEMS Gyroscope Digital Readout Scheme [21] 9
Figure 1.3. Spectral Leakage [22] 10
Figure 1.4. Coherent Sampling 11
Figure 2.1 Heterodyning's Effect on the Frequency Spectrum [25] 16
Figure 2.2. PLL Basic Block Diagram 18
Figure 2.3. Analog Version of the Proposed PLL 19
Figure 2.4. Digital PLL Implementation Inside the FPGA 20
Figure 2.5. Digital PLL Implementation for Multi DUT 22
Figure 2.6. Lock-In Amplifier Block Diagram 24
Figure 2.7. Phasor and Time domain representation of the signals 26
Figure 2.8. Quarter LUT DDS Flowchart 30
Figure 2.9. Sine and Cosine Wave Phase Difference with Phase Tuning Word 32
Figure 2.10. DDS Blocks in VIVADO 33
Figure 2.11. DDS Block Configuration for a Multi Resonator System 34
Figure 2.12. RLC Model of MEMS resonators [24] 34
Figure 2.13. SPI Interface Testbench 36
Figure 2.14. SPI Blocks Implemented for ADC and DAC 37
Figure 2.15. Frequency Response of the Filter 40
Figure 2.16. BRAM Read Only Mode Timing Diagram [34] 42
Figure 2.17. Filtering Blocks in Vivado 43
Figure 2.18. Single stage, single differential delay CIC filter [37] 44
Figure 2.19. CIC Filter Frequency Response 47
Figure 2.20. CIC IP Blocks in Vivado 48
Figure 2.21. CORDIC Initial Rotation 50
Figure 2.22. CORDIC Flowchart 52
Figure 2.23. CORDIC Blocks in Vivado 53

Figure 2.24. CORDIC Block Simulation Result in Vivado 54
Figure 2.25. Frequency vs Phase Angle and Frequency vs Phase Error 56
Figure 2.26. NCO Integral Controller IP in Vivado 59
Figure 2.27. Closed loop design for a single DUT 60
Figure 2.28. Multi-DUT resonating design in Vivado 61
Figure 2.29. Resource Utilization Report 62
Figure 3.1. Simple RLC Circuit 64
Figure 3.2. Circuit Diagram with RLC as DUT 65
Figure 3.3. Frequency Response of RLC DUT 67
Figure 3.4. System Response and User Interface 69
Figure 3.5. Output Fluctuations at a Fixed Position, raw CORDIC output [40] 70
Figure 3.6. Oscilloscope Readings 71
Figure 3.7. System Response for a Closed Loop RLC System 73
Figure 3.8. DDS Response with the Raw Data from the FPGA 74
Figure 3.9. Oscilloscope Reading for the Closed Loop RLC System 75
Figure 3.10. Step Response of the Closed Loop RLC System 76
Figure 3.11. Circuit Diagram with Quartz Crystal as DUT 78
Figure 3.12. Bode Plot of the Quartz Crystal 79
Figure 3.13. Quality factor of a Quartz crystal 80
Figure 3.14. Top view of the sytem setup 82
Figure 3.15. Overall system setup 83
Figure 3.16. Oscilloscope reading 84
Figure 3.17. Allan Variance of the 60.000 kHz and 65.536 kHz Resonators 85
Figure 3.18. Allan Variance of the 77.500 kHz Quartz Resonator 86

LIST OF ABBREVIATIONS

ABBREVIATIONS

ADC	Analog to digital converter
AR	Artificial reality
ARM	Advanced RISC machine
AW	Amplitude width
BRAM	Block random access memory
CIC	Cascaded Integrator-Comb
CL	Combinational logic
CLB	Configurable logic block
CORDIC	Coordinate rotation digital computer
CPHA	Clock phase
CPOL	Clock polarity
CPU	Central processing unit
D/A	Digital to analog
DAC	Digital to analog converter
dB	Decibels
DDS	Direct digital synthesis
DETF	Dual ended tuning fork
DLP	Digital light processing
DPLL	Digital phase locked loop
DSP	Digital signal processing
DUT	Design under test
FBAR	Film bulk acoustic resonator
FF	Flip flop
FFT	Fast Fourier Transform
FIR	Finite impulse response
FPGA	Field Programmable Gate Array

FPGA LUT	FPGA Look-up table
ftw	Frequency tuning word
GND	Ground
GSM	Global system for mobile communication
I/O	Input/Output
IF	Intermediate frequency
IIR	Infinite impulse response
IP	Intellectual property
I-Q	In-phase & Quadrature Phase
IR	Infrared radiation
IW	Input width
LF	Loop filter
LIA	Lock-in amplifier
LiDAR	Light detection and ranging
LO	Local oscillator
LP	Low pass
LPF	Low pass filter
LSB	Least significant bit
LUT	Look-up table
LVDT	Linear variable differential transformer
MA	Moving averager
MEMS	Micro Electro Mechanical Systems
MSB	Most significant bit
MSPS	Million samples per second
NCO	Numerically controlled oscillator
NEMS	Nano Electro Mechanical Systems
NEP	Net equivalent power
NF	Non filtered
OCXO	Oven controlled crystal oscillator

PC	Personal computer
PCB	Printed circuit board
PD	Phase detector
PLL	Phase locked loop
PROM	Programmable read only memory
PW	Phase width
Q	Quality factor
QAM	Quadrature amplitude modulation
QPSK	Quadrature phase shift keying
RAM	Random access memory
RF	Radio frequency
ROM	Read only memory
RTL	Register transfer level
SCLK	Serial clock
SL	Sequential logic
SNR	Signal to noise Ratio
SPI	Serial peripheral interface
TCRF	Thermal coefficient of resonance frequency
TED	Thermoelastic dissipation
THz	Terahertz
UART	Universal asynchronous receiver-transmitter
VCO	Voltage controlled oscillator
VHDL	Very High-Speed Integrated Circuit Hardware Description Language
VR	Virtual reality
WW	Word width

CHAPTER 1

INTRODUCTION

Resonance, a phenomenon describing the behavior of a system under the influence of periodic excitations, has first been observed by Galileo in 1638, then expressed mathematically by Euler in 1739 [1]. Since then, it has mainly found its primary use case in pendulum clocks for time keeping purposes, until the beginning of the 20th century. Afterwards, resonance based devices have started to find more use cases as a sensing element, where the parameter of interest is correlated with the change in resonance frequency of the device. Sensors for measuring liquid level, density and viscosity of fluids and gases, mass and force measurements, as well as detection of temperature have been devised. Combining recently developed digital electronic components of the time with the mechanical structures, based on single crystal materials, such as quartz and silicon; highly accurate, repeatable and low power consuming devices have been manufactured for various applications. Basically, electromagnetically or piezoelectrically driven mechanical transducers are kept at resonance with a maintaining closed loop. As a matter of fact, mechanical structures have shown to have a greater Quality Factor (Q) than the electrical circuits, where the energy stored in the transducer is much greater than the dissipated energy from the system per cycle, hence low power consumption and independence from the electrical characteristics of the driving circuit is possible thanks to the high mechanical Q [2].

Advancements in miniaturization and the idea of using silicon as a mechanical material for micromachining [3] have given birth to a new field of study called Micro-Electro-Mechanical-Systems (MEMS), where this technology have allowed portable, low-cost, reliable micro mechanical transducers to be manufactured. This novel area has found utility in both civil and military industries as pressure sensors, accelerometers, gyroscopes, inkjet printer heads and Digital Light Processing (DLP) units have begun to be manufactured in masses [4]. Although further miniaturization to the nano scale is possible (NEMS), some limitations as a result of scaling down the mechanical structures take place. Device impedance increases rapidly with down scaling into nano meter size, limiting their practicality [5]. Nowadays, MEMS technology has extended into a much wider area. These are, LiDAR technology for automotive industry, Artificial Reality (AR) and Virtual Reality (VR) applications, wearable medical technologies, 5G Connectivity with RF MEMS and microbolometers based thermal imaging technologies. The industry forecasts a total revenue of $18.2B by the year 2026, a %7.2 compound annual growth rate starting from the 2020 revenue of $12.1B [6].

This chapter initially gives an overview of the resonance based transducers. Afterwards, an overview of the thermal detectors is presented, where state of the arts in resonance based detectors are covered. Furthermore, Field Programmable Gate Array (FPGA) is introduced, for it is utilized in this thesis for the realization of the digital readout of resonating systems. Before the conclusion, some of the digital PLL designs for resonant sensor systems in the recent works have been covered. Finally, research objectives and thesis organization conclude the introduction chapter.

1.1 Overview of the Resonance Based MEMS Sensors

The micro mechanical resonant devices find applications in measuring parametric quantities, accurate time keeping technologies, energy harvesting devices and wireless communication systems as band pass filters and signal mixers [7]. They are even recently suggested to be used as logic devices as replacements for the transistors [8]. Operational range of these devices is known to span kHz-MHz region as well as capable of reaching GHz frequencies with high quality factors [5] in time keeping applications, where they are even seen as a replacement for atomic clocks in wide variety of sophisticated systems. In fact, Q depends on multiple mechanisms such as the environment pressure, thermoelastic dissipation (TED), anchor losses due to design choice and surface losses [9]. Most of the resonance-based systems find themselves being operated at a specific mode shape, where it is designed to be one of the flexural, bulk, shear, torsional and coupled modes. These modes often are selected for the purpose of amplifying the effect of a certain parameter among other parameters affecting resonance. Typically, an electrical signal produced by a DAC gets converted into an external mechanical excitation, causing vibrations in the structure. These vibrations result in electrons to move in an alternating manner before being read by an ADC. Being a bridge between mechanical and electrical domains, MEMS sensors are often simplified into a basic RLC circuitry with a feedthrough and parasitic capacitance affecting the output response. In fact, in the field tests of this thesis, the performance of the proposed readout design is tested first on an RLC circuit before working on the actual system. The method in which the electrical signal (voltage or current) gets transformed into mechanical excitations vary according to the transducer type being implemented for the specific design. These are capacitive, piezoelectric, piezo resistive, thermal and electromagnetic. Capacitive transducers have been the first ones utilized in MEMS for the transduction mechanism and still continues to be used in the recent publications including this thesis, where capacitive sensing and actuation for a resonating plate is used. Other than that, piezoelectric transducers are used for their capability of converting mechanical strain, applied either onto the piezoelectric material or a secondary media, into electrical polarization. Long before MEMS technology, the most known piezoelectric material Quartz have been used in various applications. Another piezoelectric transducer-based technology is Film Bulk Acoustic Resonators (FBAR) where their applications range from flow detectors into telecommunication. Additionally, thermal actuation and piezoresistive sensing are used as a transducer for certain resonant sensors. In this case, resistors are subjected to an alternating current, causing a large mechanical force as a result of alternating temperature distribution in the structure. Making use of piezoresistive elements at the sensing side of these resonators, where the resistance of these elements depends on the applied stress, the mechanical resonance is maintained [10].

The measured quantity is designed to affect either the mass or stiffness of the structure in MEMS resonators. High quality factor systems reflect these changes at the output, hence as long as the device works in a linear region for most of the cases. As the former mentioned method of changing the resonance frequency, various ways of increasing the effective mass of the system exist. Generally, the device is designed to absorb either certain chemical particles or biological molecules, effectively changing the resonance frequency of the system. Forcing particles to flow in microchannels has also shown to affect the effective mass of the system. Although the operational mode of the device varies for achieving higher Q values, the applications remain the same. The latter method, where resonant sensor's stiffness is manipulated by the measured quantity offers different possibilities. The stiffness either change as a function of the applied stress or geometric alterations caused by thermal gradients. Temperature is also known to affect the Young's Modulus of the structure, hence the stiffness. MEMS magnetometers, where the resultant Lorentz Force causes the stiffness of Dual Ended Tuning Fork (DETF) structure to change. Following the resonance frequency, a highly sensitive magnetometers is devised [11]. Similarly, resonance-based accelerometers and gyroscopes fall under this category, where the displacement of proof mass result in stiffness change. Resonance based micro mechanical structures, being actually band pass filters in nature, are extensively used in RF MEMS as well. For example, Wine Glass Disk Array Composite Oscillators with a Q of 100,000 have been demonstrated for GSM applications. Moreover, disk resonators are implemented both for intermediate frequency filtering and signal mixing purposes [5].

1.2 Overview of the Thermal Detectors

There has been an increase in the last century in the usage of thermal detectors working in the infrared and even the Terahertz band of the electromagnetic spectrum.

Although utilization of infrared as a detector had been demonstrated back in the 19th century, there has been a considerable advancement in IR detection technology since 1930 [12]. Since then, two main detection approaches which are photon and thermal based have been adopted. Although thermal based approaches offered uncooled operation unlike photon-based ones, where they have to be cooled down to cryogenic temperatures in order to operate properly, former one had long been considered as having a lower dynamic range, having a slower response and not as sensitive as photon-based arrays until 1990s [12]. Nevertheless, thermal based IR detectors, especially microbolometers, have shown to be a formidable competitor to bulky, expensive and power-hungry photon-based IR detectors [13]. In fact, there have been various and promising approaches in thermal detectors such as photoacoustic (Golay-cell), thermoresistive (microbolometer), pyroelectric, thermoelectric (thermocouple), thermomechanical (micro cantilever) and resonator based [12], [14]. Essentially, they all make use of irradiated material's mechanical or electrical property change upon radiation, and they relate it to temperature.

Among thermal detectors, resonator-based IR imaging has recently been more under the radar. Resonance based designs have the potential of competing with microbolometers, where the majority of commercial and military market uses them as thermal based detectors [12]. It has been known that specifically cut quartz resonators can measure the changes in temperature in the order of μK [15]. According to another study, for better thermal sensing, materials with large TCRF (thermal coefficient of resonance frequency) value over a wide range of temperature, high and consistent Q (quality factor) are required [16]. Exploiting this characteristic, SiN drum resonator operating between $5\,\mu m - 20\mu m$ of the electromagnetic spectrum have been demonstrated. Furthermore, sensitivities as low as 16 $fW/\sqrt{Hz}$ have been achieved by tracking mechanical resonance of SiN resonator's detuning after being exposed to IR [14], [17]. Similarly, resonance shift detection in resonators operating at torsional mode resulted in manufacturing of IR detection with a sensitivity as low as 10 $pW/\sqrt{Hz}$ [18]. For the record, NEP (net equivalent power) values of Golay cells are in the order of 100 $pW/\sqrt{Hz}$ [14], and they are considered to be the most sensitive uncooled thermal sensors; nevertheless, micro manufacturing problems have had a negative effect on their wide adoption as a detector [12]. Furthermore, resonance-based sensors are not only limited to the infrared range, but also useful in THz (Terahertz) region, which is considered to be a relatively unexplored area of the spectrum. Tracking resonance change of GaAs MEMS resonators based on temperature change proved to be a faster and sensitive method than conventional uncooled bolometers [19].

1.3 Field Programmable Gate Array Technology

Change in resonance frequency detection mechanisms in the beforementioned sensor technologies are essentially the same. There are two methods for seeing the effect of changing quantities on resonance. The first method is constantly performing frequency sweep operations approximately around the resonance peak. However, for real time applications, constantly taking the frequency response is not practical as the integration time passed at each individual frequency causes system wide delays and makes it difficult to follow the resonance peak. The second method is to drive the sensor with a known frequency while monitoring the frequency information embedded inside both the input and output signals. Then a closed loop controller is used to lock onto the resonance peak based on phase difference between the output and input of the device.

In order to perform the above-written operations in real-time, an FPGA is utilized due to their flexible design possibilities, hardware defined high speed operations and ability of parallel process execution. FPGAs are defined as reconfigurable hardware, where the reconfiguration is performed through Hardware Descriptive Languages (VHDL or Verilog). A file called bitstream configures the FPGA. Additionally, they are truly parallel in nature and the performance of the processes does not get affected by the addition of other logic blocks, unlike the processors. In fact, many processes can be executed at the rising edge of the same/different clock, hence the parallelism.

The written code is directly synthesized and implemented on the silicon hardware without requiring any kind of operating system or software. An FPGA consists of the building blocks shown in Figure 1.1. Programmable interconnects are used to connect the configurable logic blocks (flip-flops and LUTs reside inside these) of the FPGA, while I/O Blocks are responsible from accessing to the outside world [20]. Other than these, FPGA vendors such as Xilinx, Intel and Altera often add DSP units for multiplication and embedded Block RAMs inside the FPGA. There are even System on Chip devices including an external ARM CPU to help offload simple tasks, such as user interface and UART communication, from the FPGA. In this configuration, the architecture is said to be heterogenous [20].

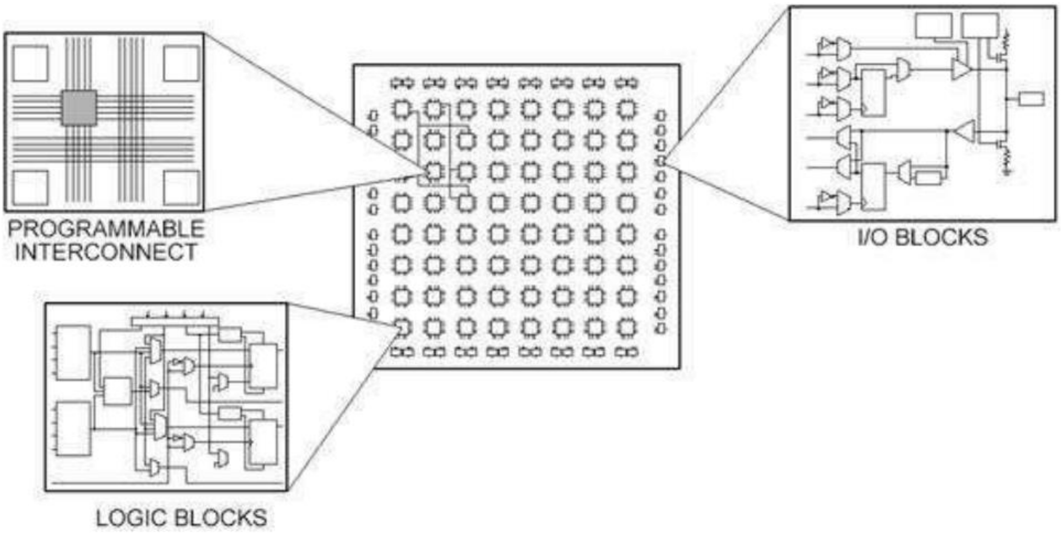


Figure 1.1. Building Blocks of an FPGA [20]

Essentially, configurable logic blocks (CLBs) are used to implement the digital circuitry otherwise would have only been possible with the use of NAND and NOR gates. With the FPGAs, these circuits are easily created by identifying them in a hardware descriptive language such as VHDL or Verilog. For this thesis, VHDL is used as it is a strongly typed language unlike Verilog. Meaning that everything in the code is to be explicitly defined, checking for synthesizable FPGA codes while writing.

1.4 Digital PLL (DPLL) for Resonant MEMS Sensor Systems

Digital based methods are inherently more advantageous than using an analog circuitry as performance of analog PLLs greatly depend on the quality of electronic components and do not have the flexibility of digital methods. For example, resonant MEMS gyroscopes are operated using FPGA based RTL (register transfer level) designs. In fact, a digital closed loop circuit for signal demodulation has been shown to resonate MEMS gyroscopes in literature using FPGA [21]. However, in this thesis, instead of using CORDIC algorithm for signal generation, it is used for angle calculation between the drive and sense sides of the design under test. An example of the proposed digital readout design in the literature is shown in Figure 1.2. Internally created logic-based quadrature signals are used for controlling both the drive and detection side of the gyroscope. Hence, two separate DPLLs are implemented in this kind of configuration. That is because for stable operation, closed loop controller both on the driving and detection axes of the gyro should be controlled with a closed loop controller. Drive voltage for the drive axis is generated by drive loop controller while balance voltage is applied to the detection axis based on the detection loop controller.

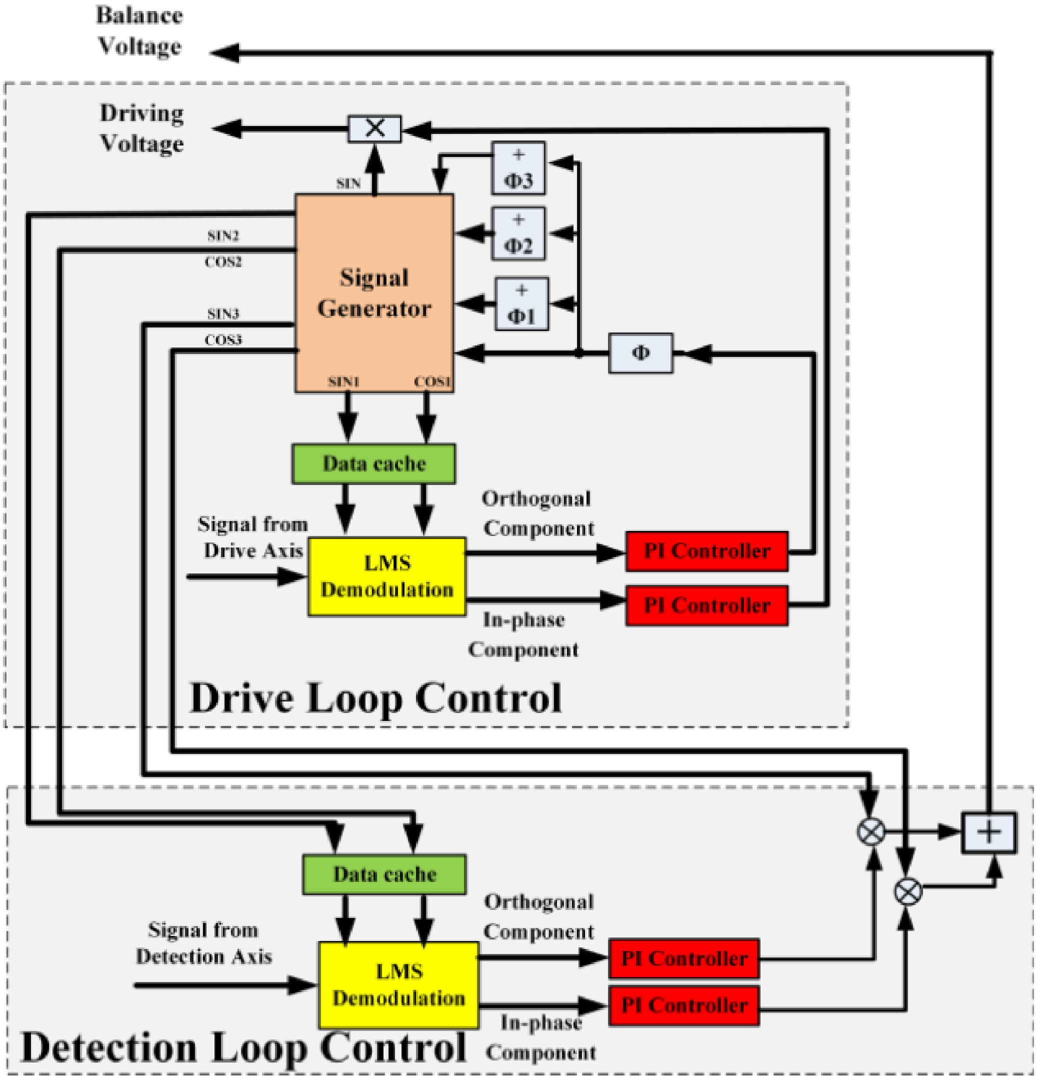


Figure 1.2. Resonant MEMS Gyroscope Digital Readout Scheme [21]

1.5 Coherent Sampling

Sampling a time domain signal with an ADC is equal to multiplying the same signal with a unit pulse train having a period of $\frac{1}{f_{sampling}}$ in time domain. It is critical to understand the effect of sampling an analog signal using digital instruments. In fact, understanding its nature makes digital phase locked loops useful for resonance sensing. According to the Fourier, multiplication in time domain is equal to convolution in frequency domain. However, with an improper ratio between the sampling frequency, number of samples, sampling time and the sampled signal frequency, a phenomenon called spectral leakage shown in Figure 1.3 occurs, resulting in unwanted frequencies in the frequency domain due to non-periodic signals.

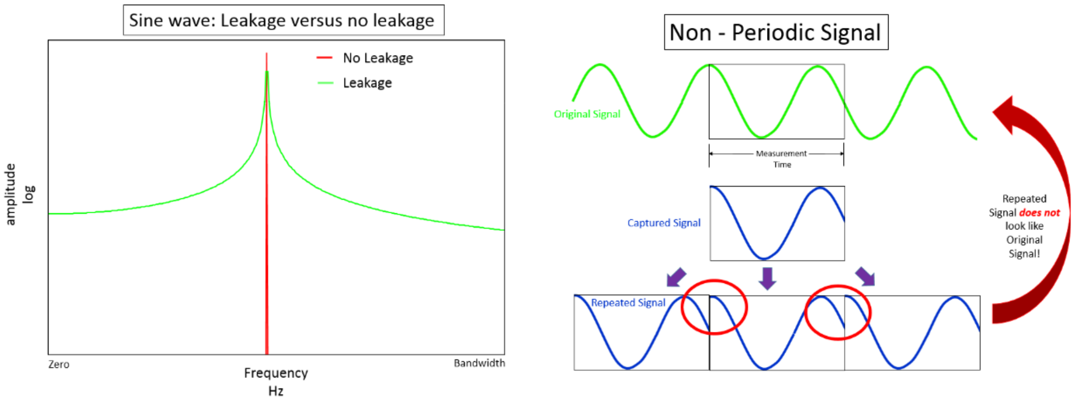
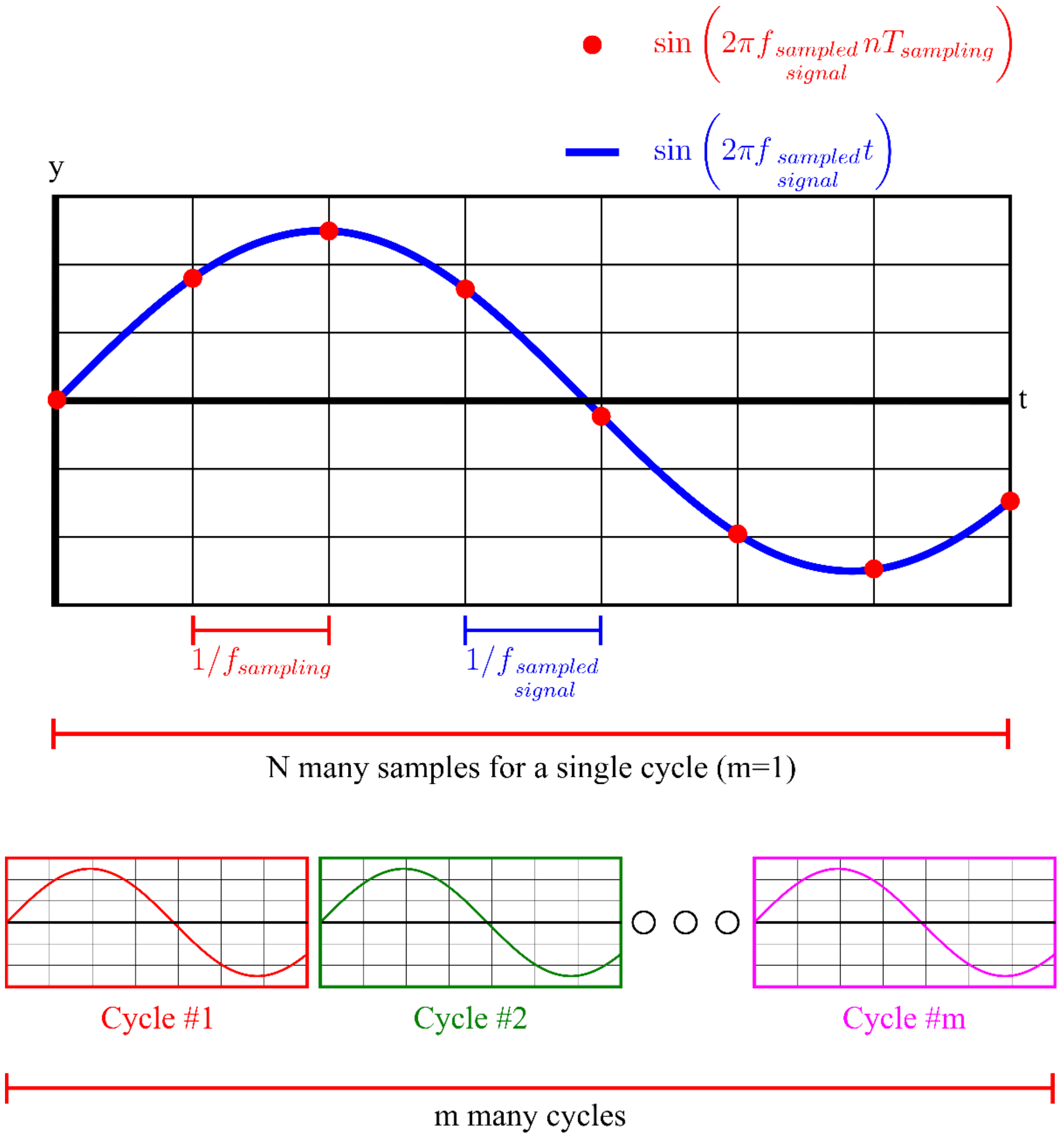


Figure 1.3. Spectral Leakage [22]

This leakage can be mitigated completely if an integer number of cycles can be fitted throughout the sampling time. Although windowing functions are utilized as well, coherent sampling resolves the issue in a more effective way if applied properly. [23] The governing equation is derived in the following way, and the time domain representation of a coherent sampling is given in Figure 1.4.



● $\sin\left(2\pi f_{\substack{sampled\\signal}} nT_{sampling}\right)$
$\sin\left(2\pi f_{\substack{sampled\\signal}} t\right)$
y
t
$1/f_{sampling}$
$1/f_{\substack{sampled\\signal}}$
N many samples for a single cycle (m=1)
Cycle #1
Cycle #2
○ ○ ○
Cycle #m
m many cycles

Figure 1.4. Coherent Sampling

N being the number of total samples taken, where each sample is taken at a period of $\Delta t_{sampling} = \dfrac{1}{f_{sampling}}$, there should be m many of whole cycles of the sampled signal where each cycle has a period of $\dfrac{1}{f_{\substack{sampled\\signal}}}$. Resulting in the following relation,

$$N \frac{1}{f_{sampling}} = m \frac{1}{f_{\substack{sampled \\ signal}}} \tag{1.1}$$

1.6 Research Objectives and Thesis Organization

This thesis proposes a real time digital readout algorithm design written on FPGA. Although the digital design written in VHDL on Xilinx's Vivado Software is able to resonate any kind of resonant transducer, a MEMS resonant microbolometer is taken as the design under test for temperature sensing purposes. The sensor is made out of a single crystalline silicon and operated in its flexural modes, where the drive and sense mechanisms are based on capacitive transduction principles. The research objectives are as follows,

1. Development of a technique for the readout of resonant based sensors using FPGA and developing a digital phase locked loop (DPLL)
2. Derivation of the mathematical operations necessary for resonance detection using digital signal processing algorithms and realization of their real time applications.
3. Performance tests of lock-in detection based DPLL both for a single resonator and system of resonators as proof of concept, using simple RLC circuits and quartz resonators. Performance evaluation of the DPLL along with the characterization of the proof-of-concept sensing elements.
4. Showing that the digital design is useful for resonating an array of MEMS bolometers for temperature sensing

The organization scheme of the thesis is as follows,

In Chapter 2, the digital theory behind the resonance detection is demonstrated. Afterwards, the mathematical operations necessary for this purpose are derived, where the signal processing components for each step of these algebraic expressions are explained under different sections. Finally, the closed loop controller for maintaining the resonance condition is introduced. Note that each of these sections show a portion of the software implementations as well besides the introductory explanations and proofs. The building blocks for the proposed design is arranged both for single resonator and multi resonator systems.

In Chapter 3, certain tests of the implemented design are demonstrated. Since a sensor can be approximated as an RLC circuit with parasitic and feedthrough capacitance [24], it is a good starting point for proving the implemented design is actually working. Afterwards, using the same building blocks (IPs), the design is altered for taking the frequency response of a quartz crystal resonating around 60 kHz. Since the quality factor of quartz crystals are much higher than an RLC circuit built with electronic components, this section is chosen to put the digital design under more stress. Finally, after making sure the system successfully operates on a conceptual level, multiple quartz crystals are resonated through a single driving and sense signal. As a result, the digital read-out technique is shown to be effective for MEMS bolometer arrays whose change in resonance is related to temperature change.

In the concluding Chapter 4, comments about the obtained results are made, and future recommendations about the work in this thesis are given.

CHAPTER 2

DESIGN

In this chapter, mathematical operations performed in digital domain for resonance matching are explained, and the implemented designs are shown. DSP algorithms are brought together in a neat way for an efficient and fast way of resonance sensing and matching within the FPGA. After giving an overall view of the concept, building blocks useful in achieving the design are inspected separately. It is shown that the proposed design is actually scalable and can be implemented for resonating multiple devices rather than a single one. The methods used in this chapter are known to be used in areas ranging from network analyzers to high speed wireless communication and multi channel software defined radios.

2.1 Theory and Implementation

Resonance based sensors work by maintaining the resonance condition through both actuation and sensing. Although the commonly accepted indication for resonance is for phase difference between the input and output to be $\pi/2$, for real time applications the frequency corresponding to that phase difference may not necessarily be equal to the maximum Signal to Noise Ratio (SNR) point. That is because of the feedthrough and parasitic capacitances present in the system. Hence, one might consider operating the sensor at a slightly different reference phase, corresponding to a different frequency. With the methods proposed in this thesis, the sensors can be operated at any phase for maximum SNR. Additionally, capability of resonating multiple DUTs driven through a single channel is demonstrated. As a result, all elements inside are kept at resonance or at the maximum SNR point. In order to realize that, some concepts are borrowed from the radio and radar technologies, which are covered in the aforementioned sections of this chapter. For these purposes, an FPGA card is used, and the following DSP algorithms are written from the ground up using VHDL. The objective is to lock all the resonators to their respective resonance/maximum SNR frequencies simultaneously. This required both a profound knowledge in signal processing and digital programming languages.

2.1.1 Heterodyning

Fundamentally, the proposed operations date back to heterodyning principle that has been used in radio communications for 100 years. Initially used in superheterodyne receivers for radio communication, converting a high frequency signal into a lower, processable intermediate frequency [25], its effects on the frequency domain components of the incoming signal build the foundation of the upcoming sections. Mixing, i.e., multiplying, two separate signals, the output spectrum would look like in Figure 2.1

Figure 2.1 Heterodyning’s Effect on the Frequency Spectrum.

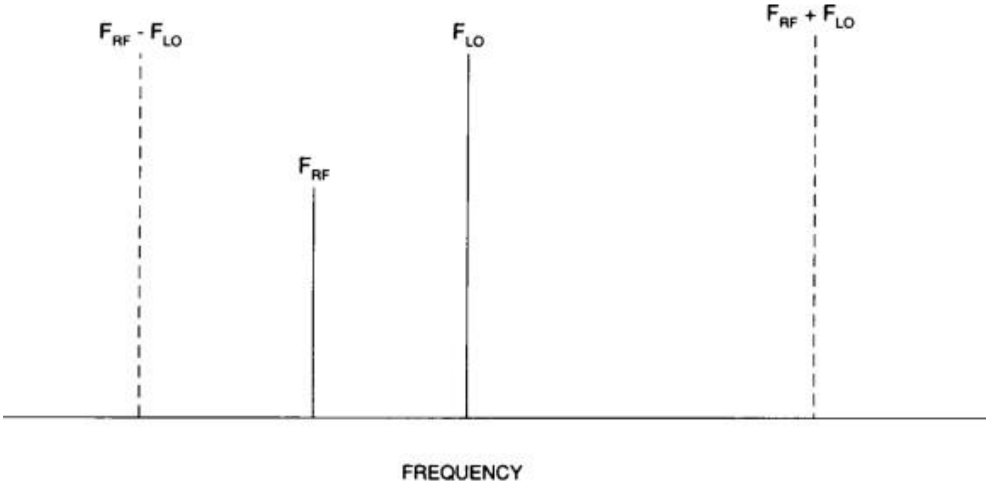


Figure 2.1 Heterodyning’s Effect on the Frequency Spectrum [25]

Mathematically, the phenomena can simply be represented as,

$$F_O = F_{LO} \pm F_{RF} \tag{2.1}$$

In (2.1), F_O is the total spectrum of the mixed signals, F_{RF} represents the RF frequency and F_{LO} is the frequency of the local oscillator. The mixed output essentially has two components in frequency domain, located at $F_{RF} - F_{LO}$ and $F_{RF} + F_{LO}$. In radio applications, the mixed signal is low pass filtered, then the information hidden inside the lower frequency component is demodulated further to obtain the message, as the modulated message still remains unchanged after heterodyne mixing [26]. For this case, there is no modulated message to be demodulated. Assuming that $F_{RF} = F_{LO}$, such that the lower frequency component is DC and the high frequency component is two times the input frequencies, using a low pass filter whose cutoff frequency is below the high frequency image, Real and Imaginary components of the received signal can be found, which is explained further in detail in the Lock-In Detection section.

Essentially, in this thesis, multiple signals are present, where each of these signals has a frequency equal to the resonance frequency of the individual resonators in the system. The signal consisting of various frequencies is mixed with its distinct frequency components respectively inside the FPGA. This mixing is explained in detail in the further sections. In fact, one should notice that the resultant signal consists of multiple frequency components, and filtering out the DC component requires careful design. Otherwise, the frequency components present in the signal as a result of this mixing phenomena would disrupt the effort for detecting the DC portion of the mixed signal. Additionally, the resonance frequencies of these resonators are chosen such that the mixing’s effect on demodulation of the DC component is minimized.

2.1.2 Phase Locked Loop (PLL)

When the two input signals have the same frequency, it is correct to assume that there will be a constant phase difference between the signals. Fundamentally using this phenomenon, a PLL’s purpose is to match the phase or frequency of a periodic input signal with respect to an internal reference signal using a closed loop system. In this thesis, closed loop controller is responsible for adjusting the frequency of the reference signal which is indirectly induced by the loop, until the reference phase difference requirement between the output and input of the resonator is satisfied. According to [27], a simple PLL consists of three basic parts, voltage-controlled oscillator (VCO), phase detector (PD) and an optional loop filter (LF), as is seen in the Figure 2.2.

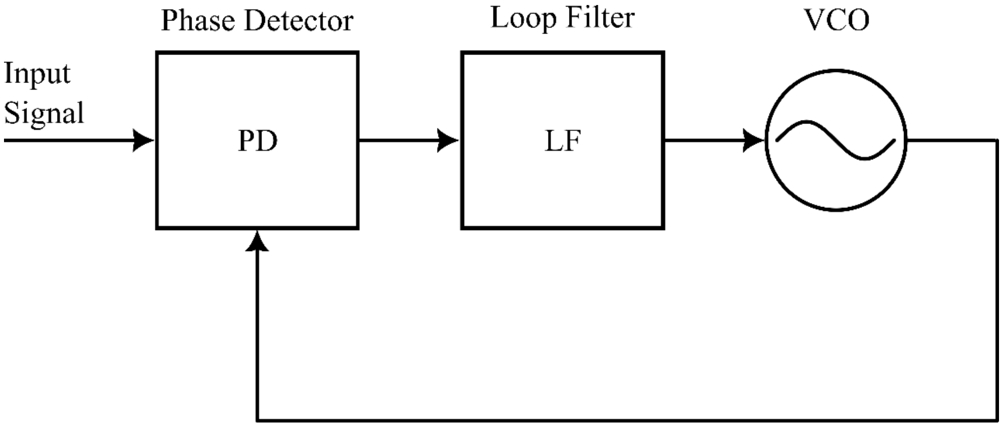


Figure 2.2. PLL Basic Block Diagram

The use case and configurations of PLLs vary, where in this thesis a Design Under Test (DUT) is desired to be operated at its resonance/maximum SNR frequency, where the phase difference between the output and the input of the DUT is desired to be held at -90° or any desired reference phase value, by varying the DUT input frequency, i.e., the VCO output. The signals to be phase locked from both at the input and output side of the DUT is fed into a phase detector along with the quadrature signals from the VCO. Signals are mixed as in the Heterodyning section, and the output is filtered with an IF filter whose frequency is at 0 Hz. The phase difference between the output and input side of the DUT is compared inside a phase comparator with the reference phase value. The error is fed into a controller for driving the VCO. The loop stabilizes when the phase comparator error is sufficiently low. The proposed block diagram is given in Figure 2.3 ,where rest of the design is built on.

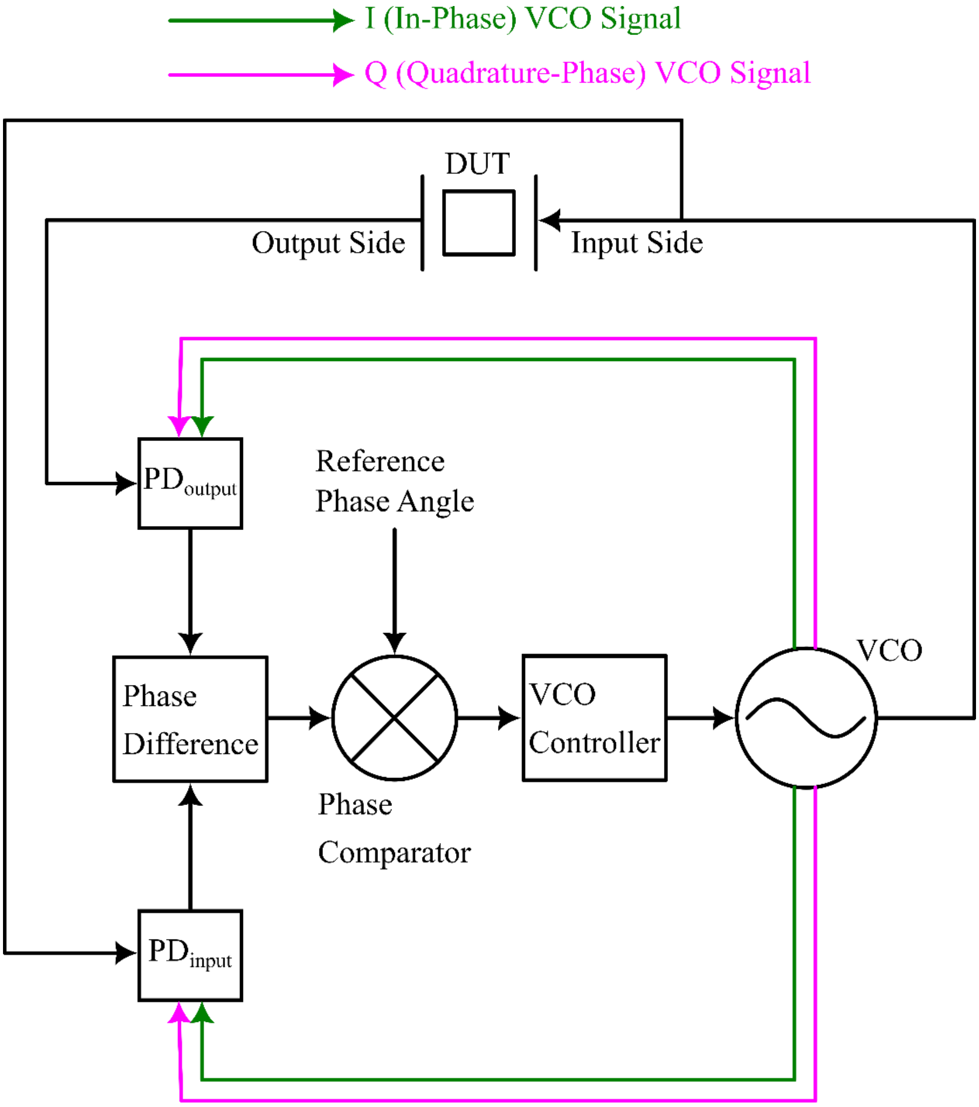


Figure 2.3. Analog Version of the Proposed PLL

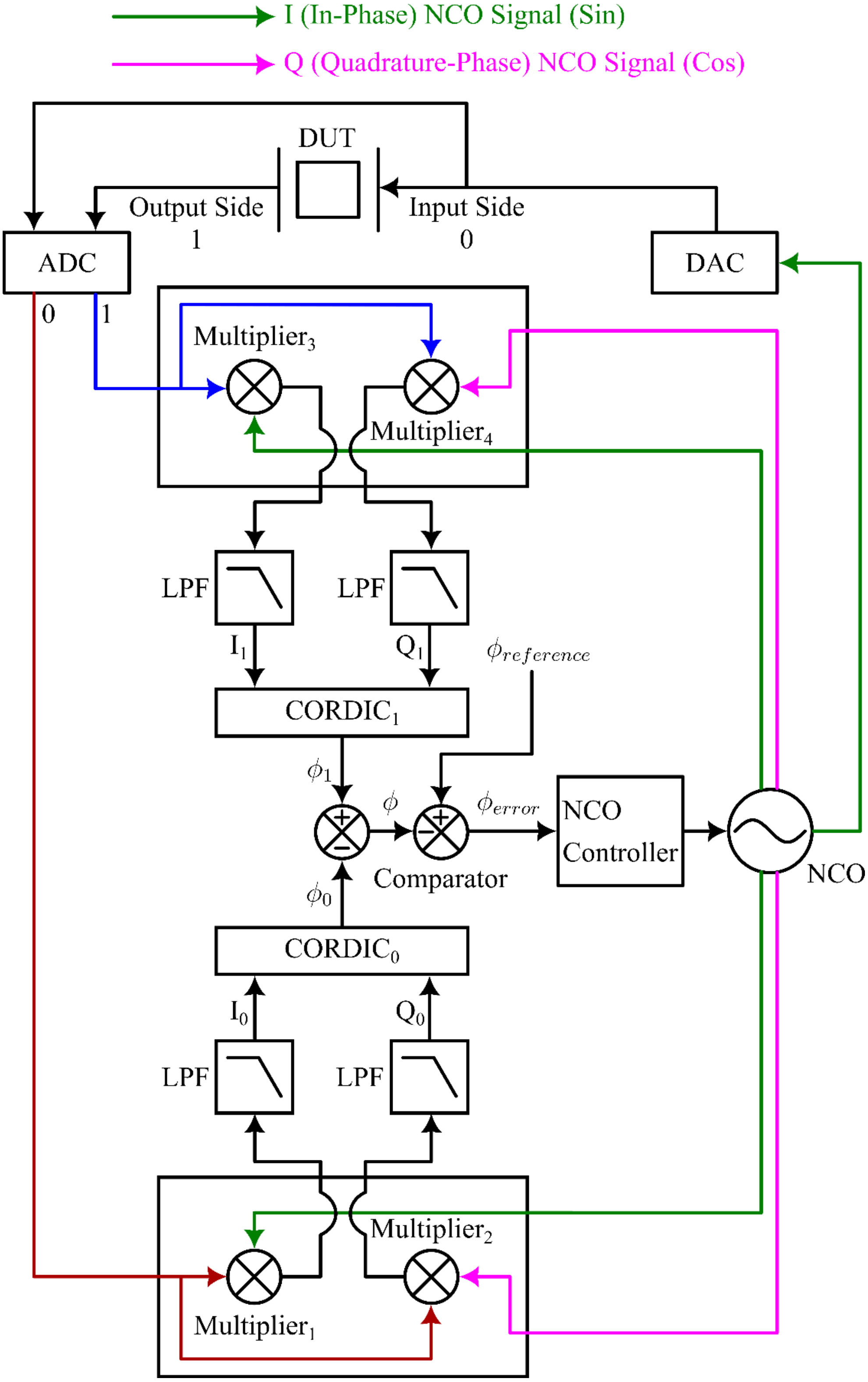
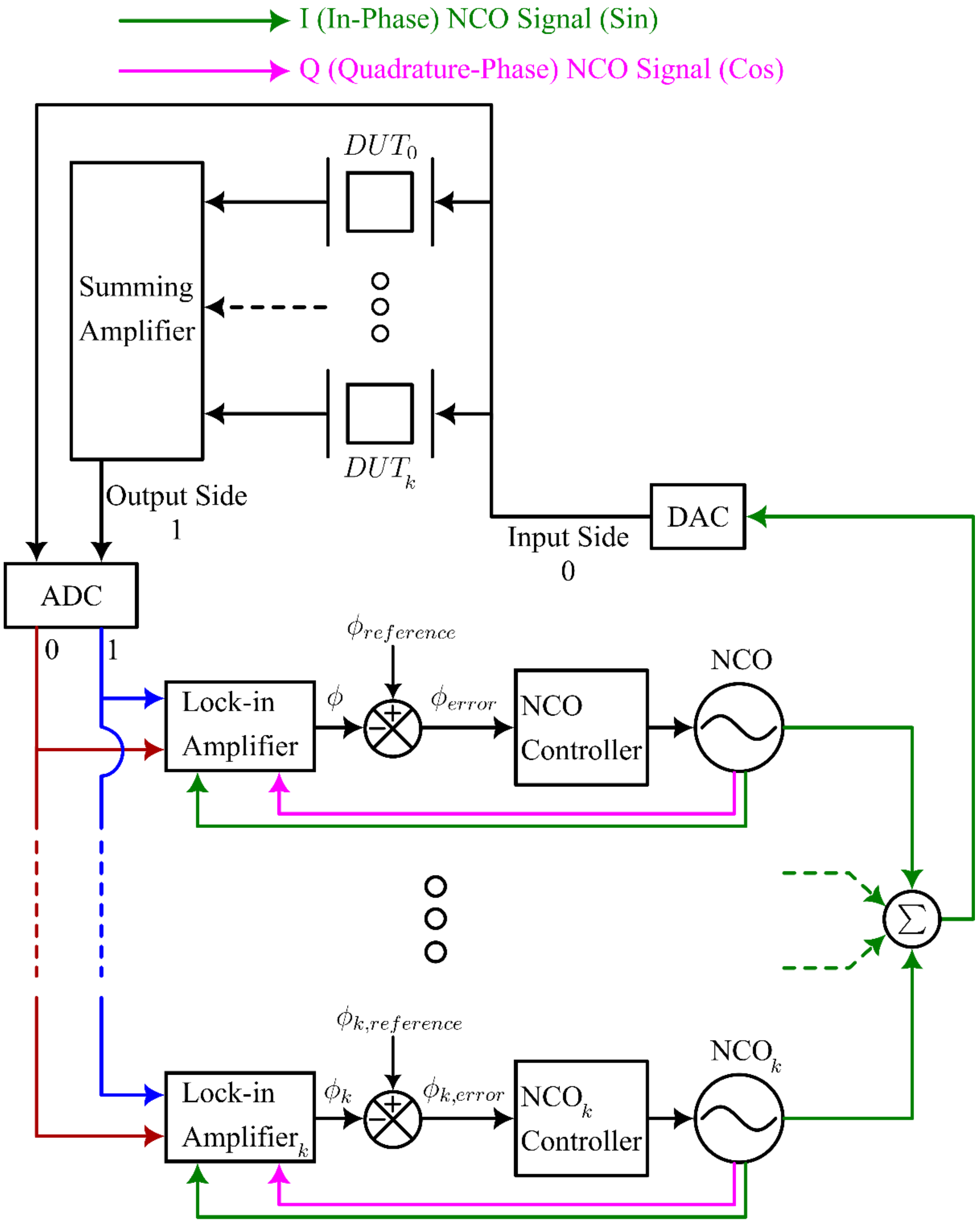


Figure 2.4. Digital PLL Implementation Inside the FPGA

Although the explanation has been done on the analog version of the PLL for convenience, digital equivalents of the building blocks such as VCO, PD and LF inside the FPGA design are Numerically Controlled Oscillator (NCO), Lock-in Amplifier in a closed loop configuration with an Integral Controller while LF is not implemented at all. The digital version of the Figure 2.3 is given in Figure 2.4. Constituting blocks are explained in detail in the following sections. Similar to the Digital PLL for single DUT, a multi DUT resonating loop is implemented as well. The design proposal is drawn in Figure 2.5, where the previous design is changed slightly for multi device setting. Since the design for single resonator is easily scalable inside the FPGA for most of the building blocks merely repeat themselves, single device setting is first implemented before the multi device PLL. However, the idea remains still the same hence the scalability.



$$k = \{0, 1, 2, \ldots, D-1\} \ , \begin{matrix} k \in \mathbb{Z}_0^+ \\ D \in \mathbb{Z}^+ \end{matrix}$$

Figure 2.5. Digital PLL Implementation for Multi DUT

2.1.3 Lock-In Detection

This block is responsible for the phase detection and NCO mentioned in Phase Locked Loop (PLL) section. It is actually the open loop version of the Digital Phase Locked Loop given in Figure 2.4. Dating back to the beginning of the 20th century [28], in order to extract the amplitude and phase information from a noisy signal, lock-in amplifiers have been used. In principle, heterodyne detection algorithms are in effect for lock-in amplifiers as mentioned before, where the input signal is mixed with a reference signal before being subjected to filtering to get rid of the high frequency components. This technique is still commonly found [26] in digital communication and radar systems. Lock-in amplifiers share the similar approach as in QAM and QPSK which are used for state-of-the-art high-speed wireless digital communication systems like 5G and Wi-Fi 6 [29][30]. The message is demodulated in a way similar to the lock-in detection. Implemented block diagram of the lock-in amplifier is given in Figure 2.6 for further discussion.

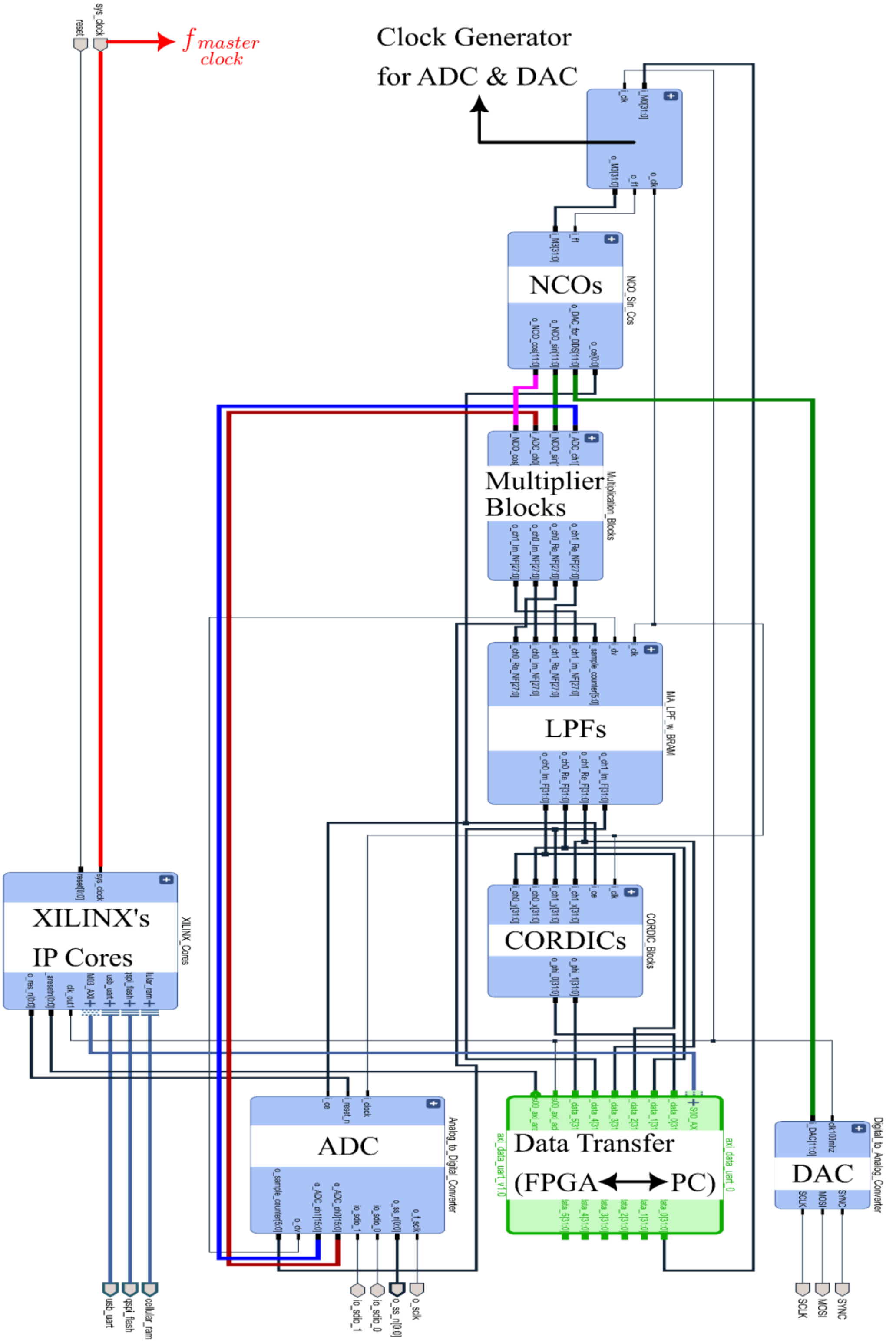


Figure 2.6. Lock-In Amplifier Block Diagram

As mentioned before, Figure 2.6 works in open loop configuration in the given configuration which is made closed loop in the following sections. Low-level design is done on Xilinx’s Cmod A7-35T: Breadbordable Artix-7 FPGA Module, using Vivado Software with VHDL programming language. All of the cores given in Figure 2.6 are written from the ground up using VHDL, except the XILINX’s IP Cores. Clock generator provides the necessary clocks to the ADC and DAC for serial communication. Furthermore, the NCO word for signal frequency to be locked-in is provided through the Data Transfer block into the NCOs. NCO block generates the quadrature sine and cosine waves at the lock-in frequency and sends the digital words to the Multiplier Blocks, where they are multiplied with the ADC readings obtained from the input and output sides of the DUT. These multiplications are asynchronous and use combinational logic with the help of limited DSP resources inside the FPGA. In total there are 4 multipliers inside the Multiplier Blocks. The results are sent to Random Access Memory (RAM) based Low Pass Filters (LPFs) and summed/integrated continuously over a certain sample length. Removing the AC components from the outputs of Multiplier Blocks, 4 LPFs in total send their results to 2 CORDIC Blocks. As the result of these LPFs represent Real and Imaginary components of the signal in phasor domain, their corresponding phase angles are calculated inside the CORDIC blocks in a single clock cycle. CORDIC Blocks work as phase detectors. The phase angles are sent out through Data Transfer block from the FPGA through UART protocol to a PC in real time, where the results are read out and saved for further evaluation. Note that the NCO controller mentioned in the previous section is not present in this configuration and later added to the design. Finally, XILINX’s Intellectual Property (IP) Cores are generated automatically by the program and are necessary for the programming and initialization of the FPGA and its peripherals such as RAM, Read Only Memory (ROM) and UART.

The operations performed in time domain are explained below and the phasor-time domain representation of the involved signals are given in Figure 2.7.

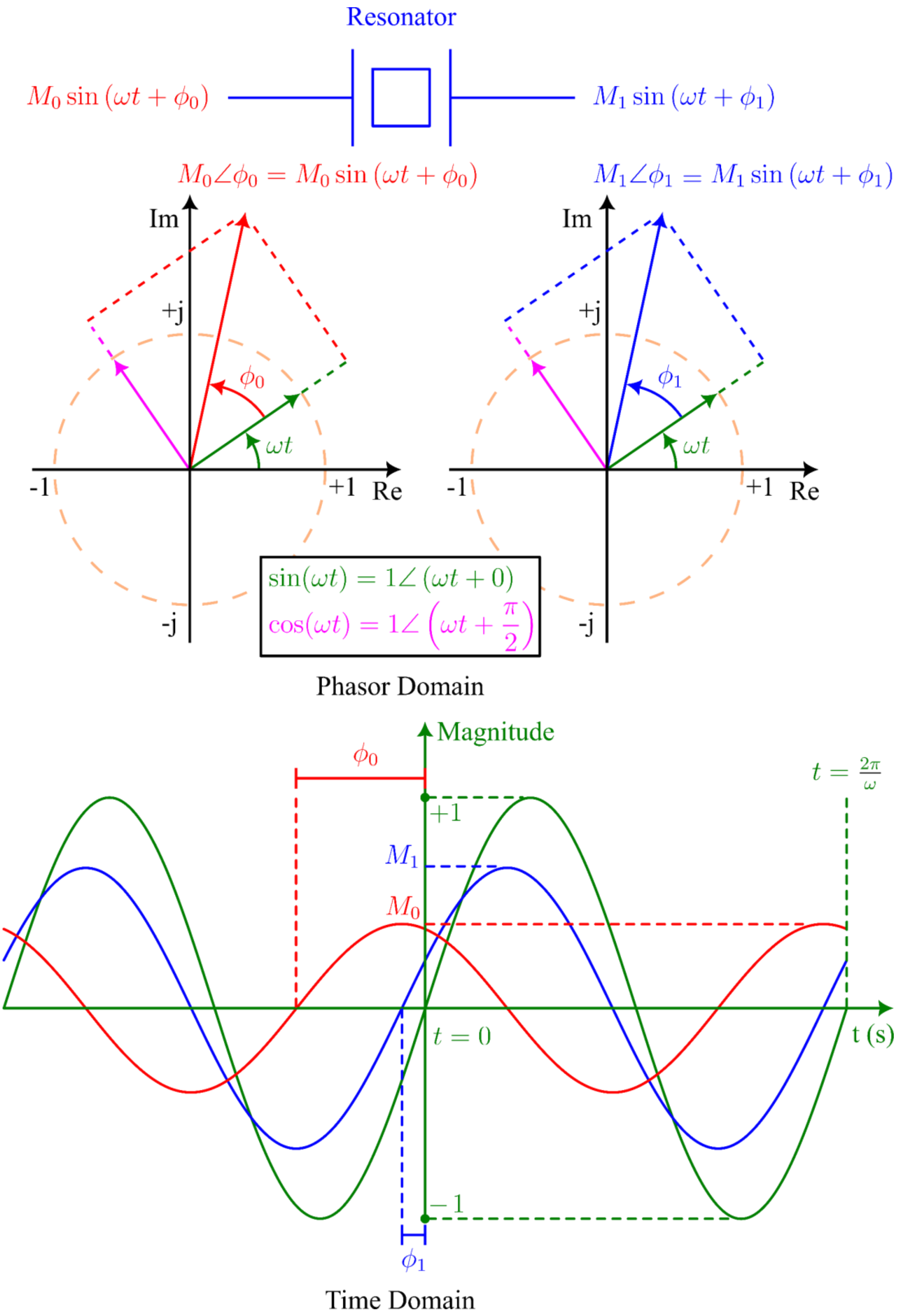

Resonator
$M_0 \sin(\omega t + \phi_0)$
$M_1 \sin(\omega t + \phi_1)$
$M_0\angle\phi_0 = M_0 \sin(\omega t + \phi_0)$
$M_1\angle\phi_1 = M_1 \sin(\omega t + \phi_1)$
Im
+j
-j
-1
+1
Re
ϕ_0
ϕ_1
ωt
$\sin(\omega t) = 1\angle(\omega t + 0)$
$\cos(\omega t) = 1\angle\left(\omega t + \frac{\pi}{2}\right)$
Phasor Domain
Magnitude
ϕ_0
$t = \frac{2\pi}{\omega}$
$+1$
M_1
M_0
$t = 0$
t (s)
-1
ϕ_1
Time Domain

Figure 2.7. Phasor and Time domain representation of the signals

Recalling some trigonometric identities before proceeding any further,

$$\begin{aligned}\sin(\mathrm{A})\sin(\mathrm{B}) &= \frac{1}{2}\left[\cos(\mathrm{A}-\mathrm{B})-\cos(\mathrm{A}+\mathrm{B})\right]\\ \cos(\mathrm{A}+\mathrm{B}) &= \cos(\mathrm{A})\cos(\mathrm{B})-\sin(\mathrm{A})\sin(\mathrm{B})\end{aligned} \tag{2.2}$$

The signal applied at the DUT input is the DDS signal which is generated inside the NCO digitally,

$$\mathrm{Im}\left[M_0 e^{j(\omega t+\phi_0)}\right] = M_0 \sin(\omega t+\phi_0) \tag{2.3}$$

Multiplying (2.3) which is the signal received from the ADC, with the following reference signals respectively results in,

$$\begin{aligned}&M_0 \sin(\omega t+\phi_0)\sin(\omega t)\\ &M_0 \sin(\omega t+\phi_0)\cos(\omega t)\end{aligned} \tag{2.4}$$

Manipulating (2.4) with the identities given at (2.2) ,

$$\frac{M_0}{2}\left[\cos(\phi_0)-\left[\cos(2\omega t)\cos(\phi_0)-\sin(2\omega t)\sin(\phi_0)\right]\right] \tag{2.5}$$

$$\frac{M_0}{2}\left[-\sin(\phi_0)-\left[\cos(2\omega t)\sin(\phi_0)+\sin(2\omega t)\cos(\phi_0)\right]\right] \tag{2.6}$$

Integrating (2.5) for a sufficiently long time would yield the quadrature components I and Q , which is done by the RAM based LPF moving average blocks or CIC filters covered in the next sections,

$$\int_{t=0}^{t=\frac{2\pi n}{\omega}} \left\{ M_0 \left[\cos(\phi_0) - \underbrace{\left[\cos(2\omega t)\cos(\phi_0)-\sin(2\omega t)\sin(\phi_0)\right]}_{\text{These terms will be equal to zero after integration.}} \right] \right\} dt \quad ,n\in\mathbb{Z}^+ \tag{2.7}$$

$$I_0 = \frac{2M_0\pi n}{\omega}\cos(\phi_0) \tag{2.8}$$

Similarly, following the same procedure for (2.6),

$$\int_{t=0}^{t=\frac{2\pi n}{\omega}} \left\{ M_0 \left[-\sin(\phi_0) - \underbrace{\left[\cos(2\omega t)\sin(\phi_0) + \sin(2\omega t)\cos(\phi_0)\right]}_{\text{These terms will be equal to zero after integration.}} \right] \right\} dt \quad , n \in \mathbb{Z}^+ \tag{2.9}$$

$$Q_0 = -\frac{2M_0\pi n}{\omega}\sin(\phi_0) \tag{2.10}$$

Both (2.8) and (2.10) are used for phase calculation at (2.11). The digital equivalent of this calculation is performed inside the CORDIC Blocks.

$$\phi_0 = ATAN2(y,x) = ATAN2(Q_0, I_0) \tag{2.11}$$

Note that the phase is the angle between the signal and the internal reference signal. Therefore, in order to calculate the phase difference between the output and the input of DUT, the same calculations should be performed for the DUT output signal, resulting in,

$$\phi_1 = ATAN2(y,x) = ATAN2(Q_1, I_1) \tag{2.12}$$

Although the input side driving the resonator denoted with the subscript 0 is expected to have no phase difference at the first time, in real life application there is always a phase difference. As a result, phase angles both at the input and output side of the resonator should be calculated separately by comparing them with the quadrature reference signals generated internally. Then, the exact phase difference between the output and the input of the DUT would be found as,

$$\phi = \phi_1 - \phi_0 \tag{2.13}$$

The following sections will focus on implementing the abovementioned arithmetic inside the FPGA using various digital signal processing algorithms. However, the main concept remains the same, to extract phase and frequency information from a given signal.

2.1.4 Direct Digital Synthesis (DDS)

DDS is a method where a signal with a dynamically adjustable frequency and phase is produced digitally and converted to an analog signal through a Digital to Analog Converter (DAC)[31]. It is called a Numerically Controlled Oscillator (NCO), if there is no DAC. Known to be the digital equivalent of a VCO, DDS block is used to generate the synchronized quadrature reference signals, sine and cosine waves, inside the DSP multiplication blocks for lock-in detection. It is also used for generating the serial communication clocks of ADC and DAC for maximum throughput.

DDS in principle uses a phase accumulator inside the NCO component, where a counter is incremented as a pulse is received from a clock source. The accumulated value is stored in a register after each pulse. The increment amount is equal to the binary tuning word, M, and the output register is used as an input to a Look Up Table (LUT) stored inside the FPGA Programmable Read Only Memory (PROM). The LUT inside the PROM contains $^1\!/_4$ of a sine-wave and outputs the sine-wave word according to the instantaneous value of the accumulator. The value is sent to the DAC to produce the desired driving signal. The LUT generator is written in MATLAB and pasted inside the FPGA PROM. The code is given in Appendix A. The reason only a quarter of the sine wave is stored is because of the symmetry of the sine wave, the remaining values can be derived from the LUT and phase accumulator’s output. Hence, less FPGA logic is consumed. The flowchart for DDS implementation is given in Figure 2.8. Processes inside the flowchart are identified as CL (Combinational Logic) and SL (Sequential Logic). The former one does not require a clock to be executed while the latter one requires an input clock, where the execution of the process takes place in the rising edge of this input clock. Also, the SLs are processes which are executed at the same rising edge. In fact, this is one of the main distinctions of VHDL from higher level languages such as C.

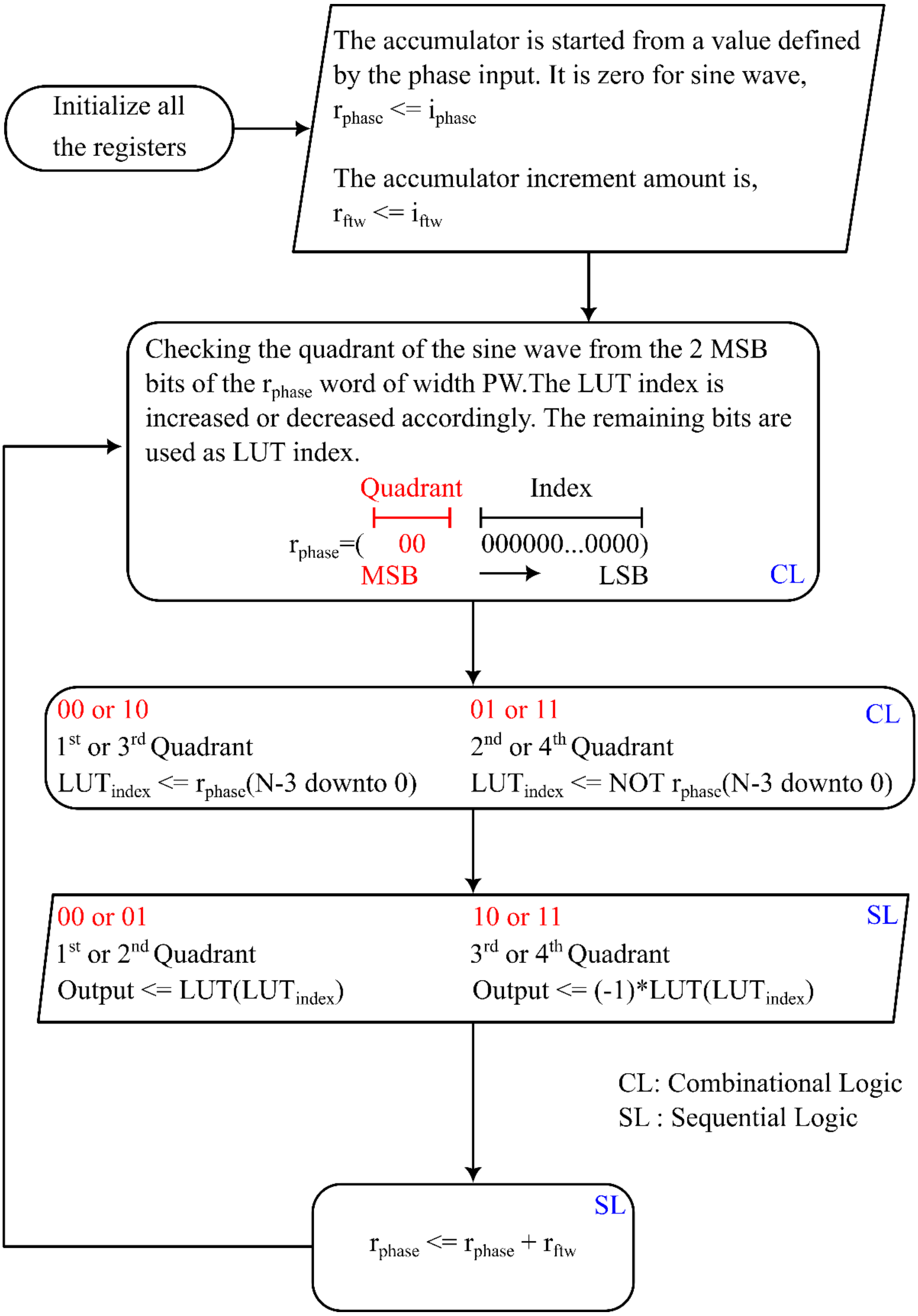

Initialize all the registers
The accumulator is started from a value defined by the phase input. It is zero for sine wave,
r_{phase} <= i_{phase}
The accumulator increment amount is,
r_{ftw} <= i_{ftw}
Checking the quadrant of the sine wave from the 2 MSB bits of the r_{phase} word of width PW.The LUT index is increased or decreased accordingly. The remaining bits are used as LUT index.
Quadrant
Index
r_{phase}=(00 000000...0000)
MSB
LSB
CL
00 or 10
1st or 3rd Quadrant
LUT_{index} <= r_{phase}(N-3 downto 0)
01 or 11
2nd or 4th Quadrant
LUT_{index} <= NOT r_{phase}(N-3 downto 0)
CL
00 or 01
1st or 2nd Quadrant
Output <= LUT(LUT_{index})
10 or 11
3rd or 4th Quadrant
Output <= (-1)*LUT(LUT_{index})
SL
CL: Combinational Logic
SL : Sequential Logic
SL
r_{phase} <= r_{phase} + r_{ftw}

Figure 2.8. Quarter LUT DDS Flowchart

The simple equation used for DDS is given as, where M is the tuning word, f_{clock} is the reference clock frequency, and n is the phase accumulator's word width. The resolution of the DDS is enhanced by choosing a wider phase accumulator, where n is increased. Indeed, as n is increased, systems with high Q can be resonated in a stable manner. Otherwise, the loop can become un-locked from the resonance easily,

$$f_{out} = \frac{M \times f_{clock}}{2^n} \tag{2.14}$$

In this design, both sine and cosine waves are simultaneously generated thanks to FPGA's parallel processing capability, allowing the generation of these signals after each reference clock cycle. The phase accumulators r_{phase}, of sine and cosine DDS generators start from values which are $\frac{2^n}{4}$ apart, as sine and cosine waves are expected to have a $\frac{1}{4}$ cycle phase difference in time domain as shown in Figure 2.9. Even though accumulator width is $n = 32$, there is no need to store a LUT with a depth of 2^n. Instead, a LUT with a depth of 16, also named as phase width, is defined. As a result, LUT only uses the 16 most significant bits for LUT index value. However, it does not change the fact that sine and cosine NCOs have a phase difference of a quarter cycle. Their ftws are incremented the same amount defined by M or r_{ftw}, causing them to operate at the same frequency. Moreover, AW stands for amplitude width and represents the word width of each of the stored signed value inside the LUT. The LUT generator code is given in Appendix A. AW is taken as 12 bits as the ADC is able to produce samples of 12-bit width. Hence, multiplication between the ADC samples and NCO words will result in 24 bits wide value.

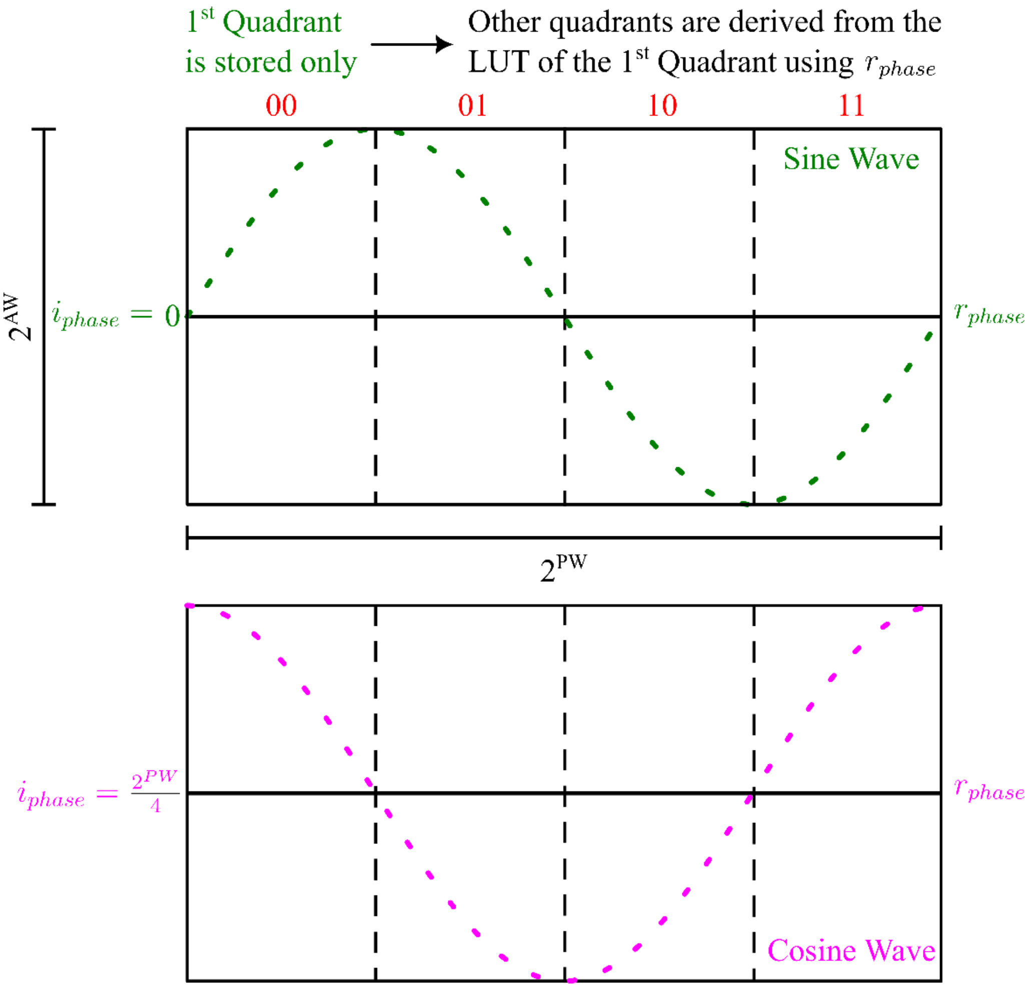

1st Quadrant is stored only
Other quadrants are derived from the LUT of the 1st Quadrant using r_{phase}
00
01
10
11
Sine Wave
2^{AW}
$i_{phase} = 0$
r_{phase}
2^{PW}
$i_{phase} = \frac{2^{PW}}{4}$
r_{phase}
Cosine Wave

Figure 2.9. Sine and Cosine Wave Phase Difference with Phase Tuning Word

With the same logic, DDS based clock generators are also implemented and used in the design. Clock generators are simpler to implement as they do not require any LUT as the MSB of the phase accumulator is the output. Clock frequencies with a precision of $\frac{f_{master\ clock}}{2^n}$ are possible and used throughout the design. Detailed view of the designed DDS blocks from Figure 2.6 is given in Figure 2.10.

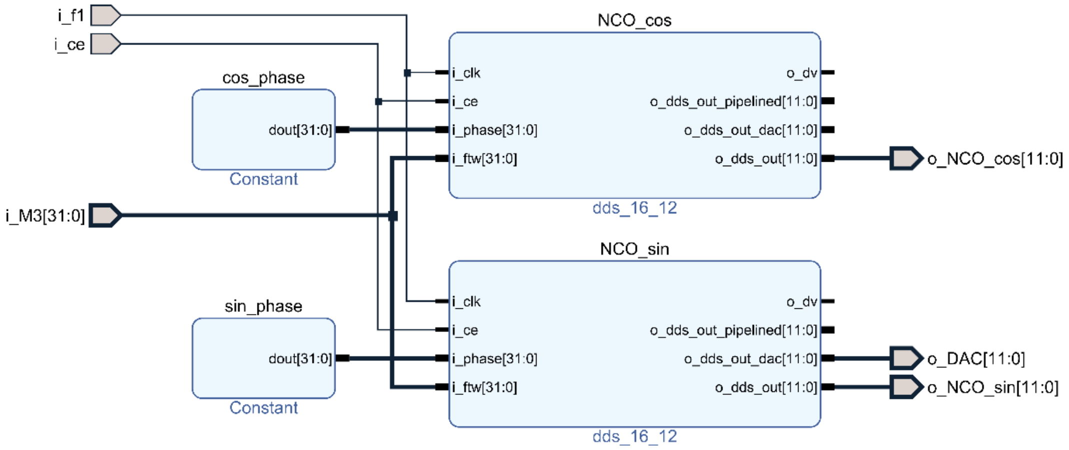
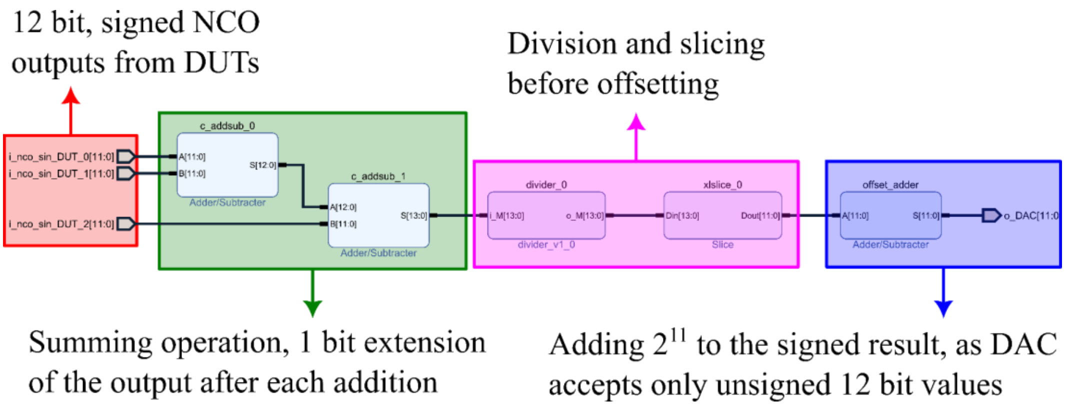


Figure 2.10. DDS Blocks in VIVADO

The DDS blocks in Figure 2.10 employing NCO algorithm inside are used for driving multi resonator systems through a single DAC. It is implemented by copying the blocks as separate processes. FPGA makes sure that they are executed simultaneously as well. The DDS outputs from each of these separate blocks are used for lock-in calculation of individual resonators while the in-phase components of these DDS blocks are added together and supplied as an input to the DAC. Essentially, the DAC output consists of multiple sine waves instead of a single one for single resonator configuration. The implemented DDS design for multi resonator systems is given in Figure 2.11. NCO outputs coming from the individual DUTs are summed and the result is manipulated for the 12-bit DAC. After each addition operation, bit growth is considered, hence the output register is extended 1 bit. Afterwards, in an effort to prepare the output for 12-bit DAC, which accepts positive unsigned registers as its input, the summation is divided and sliced such that when 2^{11} offset is added to the result, overflow does not occur. The addition of offset allows the DAC to output the summation along with a DC component, which is filtered out with an analog low pass filter during the experiments.



12 bit, signed NCO outputs from DUTs

Division and slicing before offsetting

Summing operation, 1 bit extension of the output after each addition

Adding 2^{11} to the signed result, as DAC accepts only unsigned 12 bit values

Figure 2.11. DDS Block Configuration for a Multi Resonator System

2.1.4.1 Feed Through Cancellation

It is known that resonators can be modeled as RLC circuits with feedthrough and parasitic capacitances. The model is shown in the following figure,

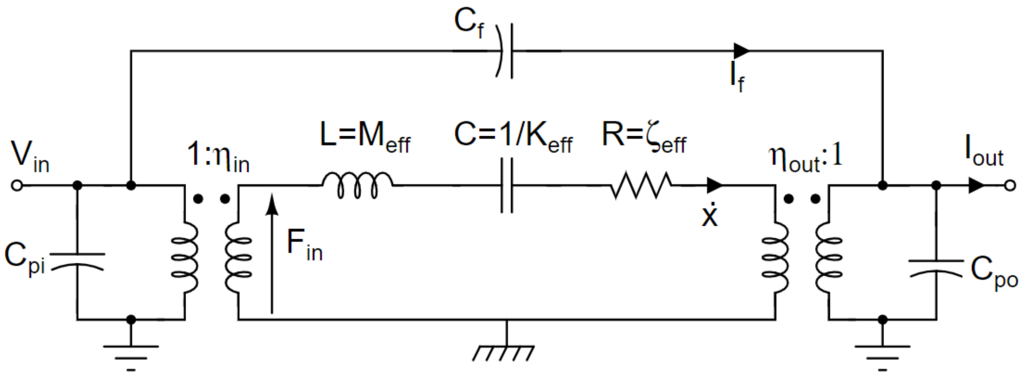


Figure 2.12. RLC Model of MEMS resonators [24]

As a result of these capacitors, while driving a resonator, the driving signal leaks to the sense side, due to the fact that both drive and sense nodes being conductors with some capacitance in between. Since the feedthrough capacitance changes according to both the frequency of the driving signal and external disturbances such as temperature, MEMS resonators have to be designed to accommodate a suppression mechanism for this parasitic effect. As a result of these capacitive effects, the maximum SNR point turns out to be at a different location than $\phi = -90^{\circ}$,if left uncompensated. Generally, fully differential drive method is used where two drive signals that are 180° apart is fed to separate electrostatic actuation pads present near the resonator. These pads are usually placed on specific locations so as not to affect the mode shape of the resonator during operation. Since both of these signals are going to be exposed to the same feedthrough capacitance as they are both virtually connected to the output of the resonator, overall effect of the feedthrough capacitance is expected to cancel out. Utilizing the parallel processing capability of FPGAs, similar to creating in-phase and quadrature-phase signals, one can create 180° phased driving signal inside an NCO and feed it to a second DAC, achieving fully differential driving.

2.1.5 Serial Peripheral Interface (SPI)

SPI communication protocol is commonly used for high-speed communication between devices. The communication is synchronous and full duplex, allowing a single master device (FPGA) to drive many slave devices such as an ADC or DAC. The dual independent channel 12-bit ADC used in this thesis has a maximum sampling rate of 1 Million Samples per Second (MSPS), and AD7476A chip inside communicates with a Serial Peripheral Interface (SPI). Similarly, the 12-bit DAC constitutes an AD5628 chip with the same communication protocol. Serial communication algorithm is written in VHDL where the testbench output obtained from Xilinx's Vivado Software is shown in Figure 2.13. Note that the timings are given for relative comparison only and do not represent the actual timing values in nanoseconds (ns). In reality, one should consider the device specific timing requirements. This figure is only for showing that the SPI protocol works properly. CPOL and CPHA are device specific parameters and indicate whether the data is sent in/out at the rising edge or falling edge of the serial clock, o_sclk and o_ss pin's expected state at the beginning and at the end of transmission. Other than that, io_ pins express the serial data pins that are clocked in/out with respect to the serial clock. According to Figure 2.13, a new sample is received after exactly $17.5t_{SCLK}$, where t_{SCLK} is the serial clock necessary for SPI protocol and is provided by the FPGA master, to the slave ADC.

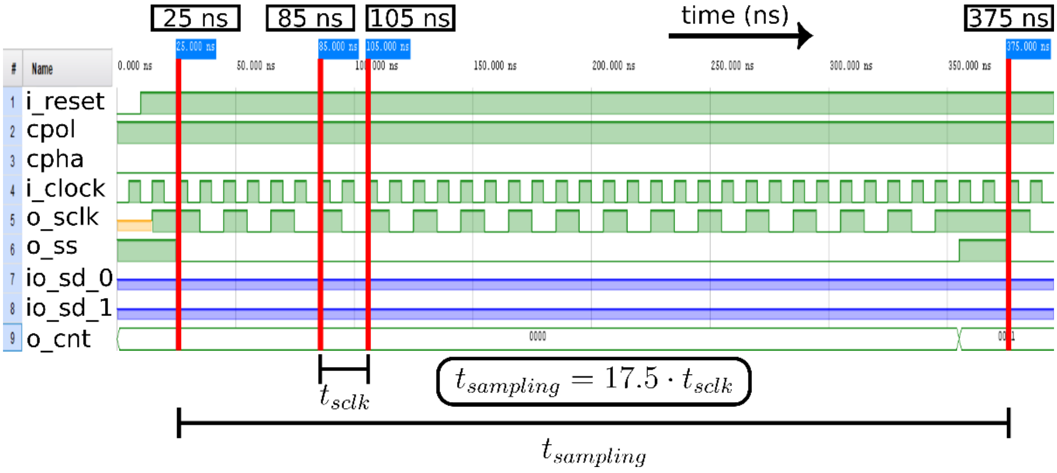


Figure 2.13. SPI Interface Testbench

A generic SPI protocol written in VHDL is turned into IP Block in Vivado for better visual comprehension. ADC and DAC both make use of these custom IP Blocks while communicating with the FPGA. The communication is performed at the highest rate possible with this setup. IP Blocks are shown in Figure 2.14.

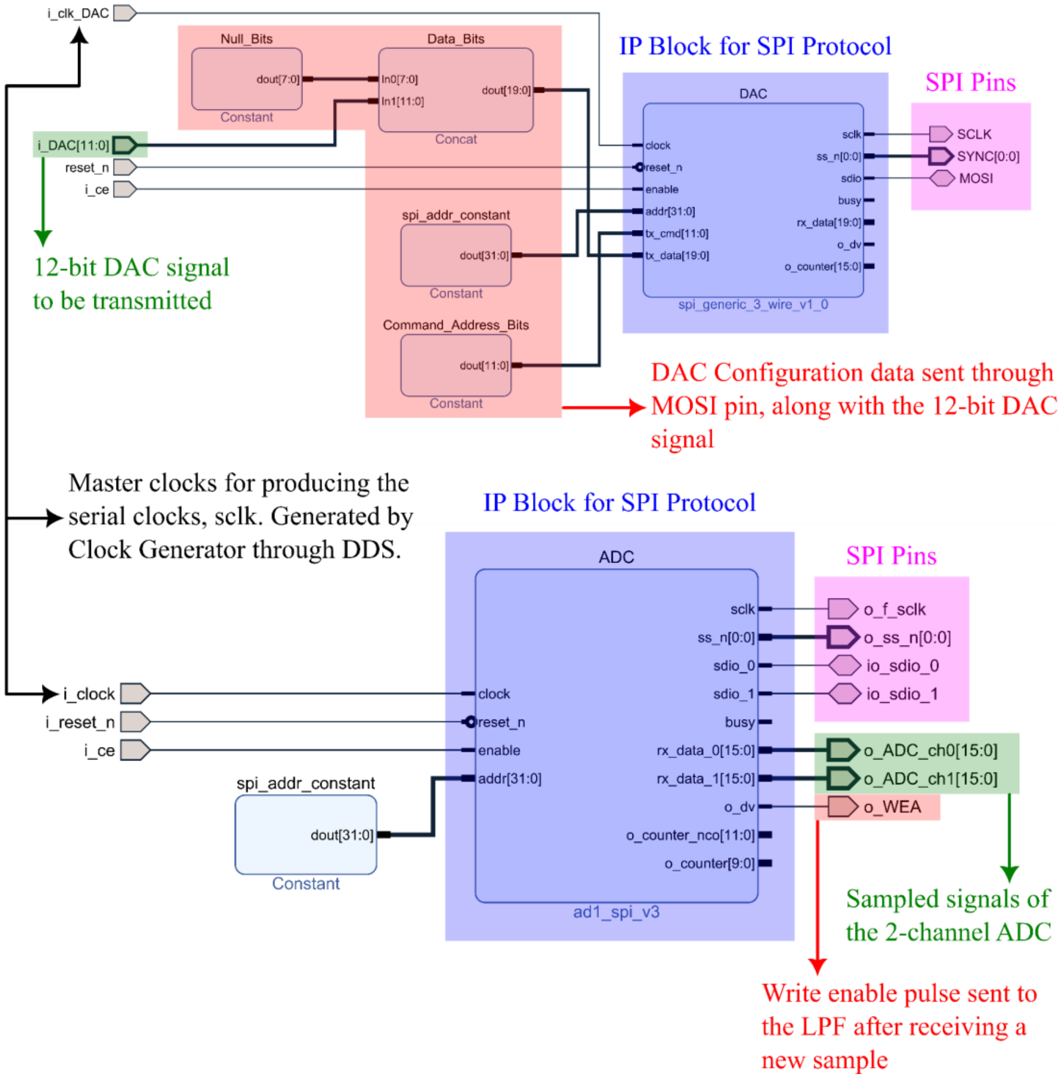


Figure 2.14. SPI Blocks Implemented for ADC and DAC

2.1.6 BRAM Based Moving Average Low Pass Filter

In order to realize the integral operations mentioned in (2.7) and (2.9) inside the FPGA, computationally efficient and multiplierless low pass filter (LPF) with moving average is implemented. This is especially important as otherwise for every tap of the filter, a distinct coefficient is both needed to be stored in the hardware and a multiplication operation is needed to be performed. This is not sustainable because the hardware multipliers are extremely limited resources, and the filter length becomes extremely limited. Essentially, an LPF is used to extract the magnitude and phase information from the mixed signals by means of averaging/taking the integral of a periodic signal in digital domain. Increasing the filter length makes sure that the integration filters out the AC component of the signal properly. Additionally, the moving average portion of this filter provides us with the opportunity of keeping the average of the latest samples at all times, while updating the result constantly based on the most recent sample.

The purpose is to create a filter which constantly averages the latest N many samples. As the new sample is introduced inside the average, the oldest sample is taken out from the average. The average is updated at each sample. It is made sure the averaged signal contains the least number of unwanted frequencies, leakage, which would have resulted in the average being fluctuated because of the uneven cycles fitted inside the window. Although total cycles fitted inside the average is not even, increased tap number takes care of the effect of unevenness. The most basic way of implementing a moving average filter is to use an FIR filter with unity filter coefficients, expressed in the following way.

$$y[n] = \frac{1}{N}\sum_{k=0}^{N-1} \underset{=1}{a[k]}\, x[n-k] \tag{2.15}$$

Although the filter described in (2.15) is a moving average filter as well, the amount of resources necessary for this filter is high due to the fact that N many adders are necessary, and the samples are to be delayed with the use of flip-flops, making it a resource intensive filter.

On the other hand, the moving average can be realized with another, much more practical approach. With this filter, the samples are summed for once. Furthermore, after adding the first N data set, the average is updated by adding the newest sample to and subtracting the oldest sample from the filter output. However, this filter requires the use of BRAMs, making them limited as the bandwidth of the filter decreases. The difference equation for this filter is written as below. N is the length of the filter,

$$y[n]=\frac{1}{N}\left(y[n-1]+x[n]-x[n-N]\right) \tag{2.16}$$

Even though eqn. (2.16) resembles an IIR filter where a recursive averaging takes place, the input is taken as is and not multiplied with a weight constant. Also, unlike an IIR filter, instead of taking a portion of the old average to calculate the new output, previous average is taken as a whole. The only difference is subtracting the oldest sample. Let us recall the IIR filter's difference equation as well[32],

$$y[n]=a_1 y[n-1]+b_0 x[n] \tag{2.17}$$

In order to implement (2.16), a Random-Access Memory (RAM) is necessary as mentioned before, where the most recent N many samples are stored in. For this purpose, Block RAM (BRAM) readily found in the FPGA fabric is used. Therefore, instead the recursive averager whose impulse response is a boxcar function is implemented using BRAM.

Writing the z-domain expression of the filter in (2.16).

$$Y(z)=\frac{1}{N}\left(Y(z)z^{-1}+X(z)-X(z)z^{-N}\right) \tag{2.18}$$

Rearranging it,

$$H(z)=\frac{Y(z)}{X(z)}=\frac{1}{N}\frac{1-z^{-N}}{1-z^{-1}} \tag{2.19}$$

One can find the frequency response of the filter described in (2.19) with the following way,

$$\text{Let } z = e^{j\omega} \text{ where } \underset{\substack{\text{Normalized}\\ \text{Frequency}}}{\omega} = \frac{f_{\substack{sampled\\ signal}}}{f_{sampling}} = \underbrace{\boxed{0\ldots \quad \underset{\substack{\text{Nyquist}\\ \text{Frequency}}}{\pi}}}_{\substack{\text{Range of Interest}\\ \text{for the Response}}}\ldots 2\pi \qquad (2.20)$$

The frequency response of the moving average filter described in (2.16) can be found with a simple MATLAB script given in Appendix B. The response is in Figure 2.15. Due to the shape of response, these filters are also known as sinc filters [33].

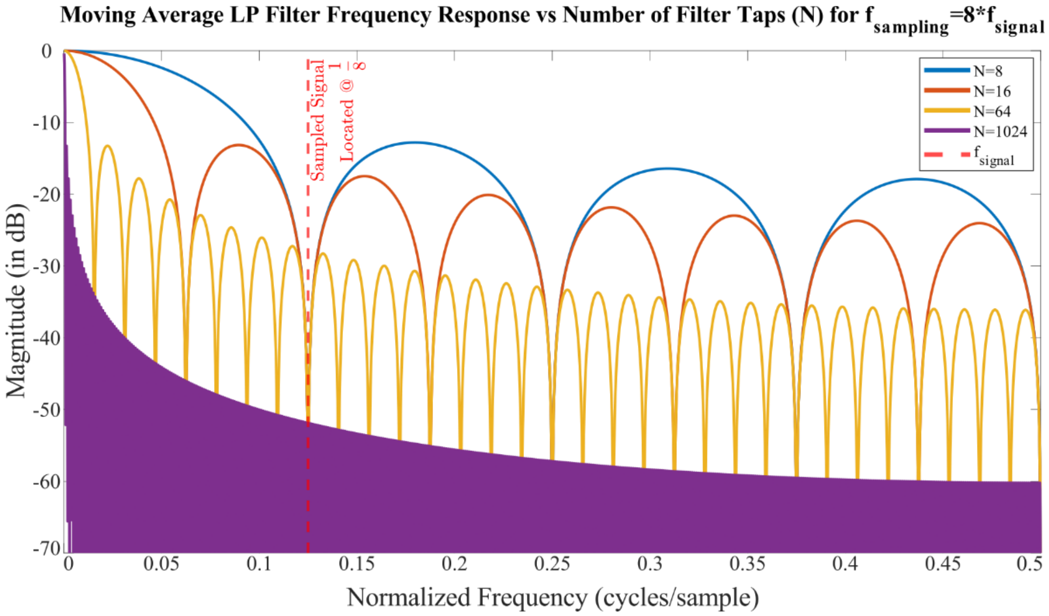


Figure 2.15. Frequency Response of the Filter

The horizontal axis is given as the normalized frequency. Frequency of an example sampled signal is marked at normalized frequency of $\omega = \frac{1}{8}\frac{cycles}{sample}$ on Figure 2.15 located on top of the filter zero. Nevertheless, filtered-out signal is rarely on top of the location of filter zero in z-domain. It is seen that the signal attenuation gets better with the increase in filter length from 16 to 1024. Stopband attenuation dramatically increases while transition occurs more steeply, Also, the passband region becomes smaller as the cutoff frequency decreases, filtering out most of the periodic signals except the DC component of the input, which is necessary for lock-in detection.

Normally, another process is performed to converge the filter to an ideal brick-wall filter. However, it is neither necessary nor desired for this case. Additionally, the higher frequency component 2ω present inside the integral terms in (2.7) and (2.9) correspond to another zero location of the filter, attenuating that component further as well.

As a final step, updating the average based on the newest sample is done using BRAM mentioned before. Block memory generator is used to instantiate the BRAM. Averaging operation is performed based on the timing diagrams provided by the manufacturer, Xilinx [34]. Figure 2.16 shows the read first mode timing diagram as each sample is written to the BRAM before being replaced with the latest sample. In order to do that, the BRAM is operated at the read first mode, allowing read from the address first, before the latest sample is written into that same address inside the memory. The necessary signals to be provided and their relations are explained in the following sentences. WEA enables the writing operation provided as a pulse from the ADC after receiving a new sample. DINA is the multiplication result obtained as a result of combinational logic inside the FPGA using DSP tiles. This is the value written inside the BRAM which is updated simultaneously with the newly received sample from the ADC. Additionally, ADDRA is provided by the ADC, which is essentially a counter being incremented with each new sample. Also DOUTA is equivalent to $y\left[n-1\right]$ term inside (2.16), where DINA is $y[n]$ and n is the ADDRA signal. ENA input is always high as the BRAM is operated at Always Enabled mode. The configuration is Single Port RAM where the write & read widths are equal to the multiplication results equal to the summation of widths of the multiplied registers. In other words, if two numbers with a width of A and B are multiplied, the output will have a final width of $A+B$. The depth of the BRAM is equal to the filter length, low pass filter can easily be adjusted by means of varying the $BRAM_{depth}$, i.e., the total number of samples used in calculating the moving average.

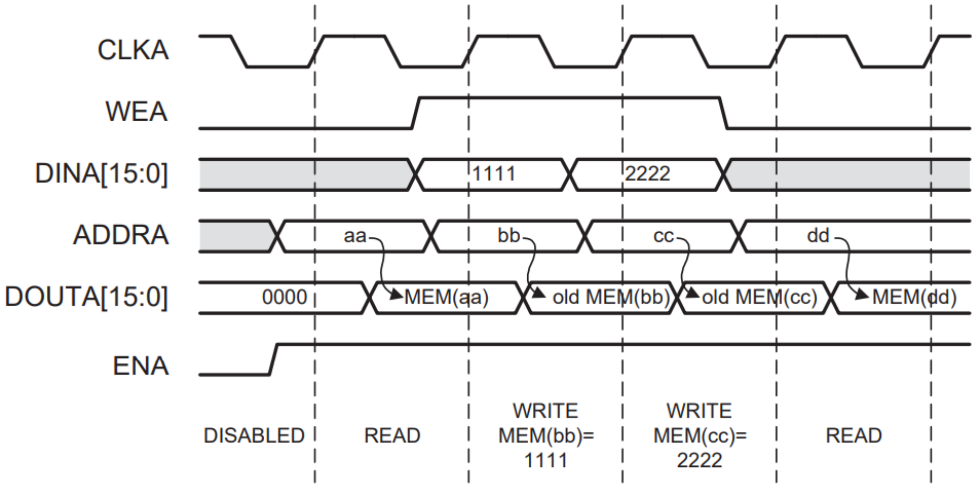


Figure 2.16. BRAM Read Only Mode Timing Diagram [34]

As a final consideration, WEA pin, responsible from the write enable operation into the BRAM, and BRAM master clock CLKA should have a certain clock relation expressed vaguely in Figure 2.16. The relation between the frequency of those pins chosen as follows inside the FPGA and the VHDL code is written accordingly,

$$f_{CLKA} >= 4 \cdot f_{WEA} \tag{2.21}$$

Considering the WEA signal is actually equal to the sampling rate of the ADC, WEA pin is basically a clock where at each of its rising edge a new sample is accepted, and the address of the BRAM is updated. f_{WEA} and f_{CLKA} are in-phase as well, as they are derived from a common master clock using DDS Clock Generator IP Block.

The IP Block configuration for the filter elements is given in Figure 2.17. It can be seen that for each resonator, there are in total 4 Blocks responsible from BRAM utilization and 4 Blocks from moving average calculations. The filtered-out results are sent out to CORDIC blocks for phase angle calculations. The signal prefixes o_ and i_ are for output and input, _ch#_ for the channel number, _Re_ and _Im_ represent whether the result belongs to the Real or Imaginary part of the signal in phasor domain and _NF, _F represent whether the signal is not filtered or filtered.

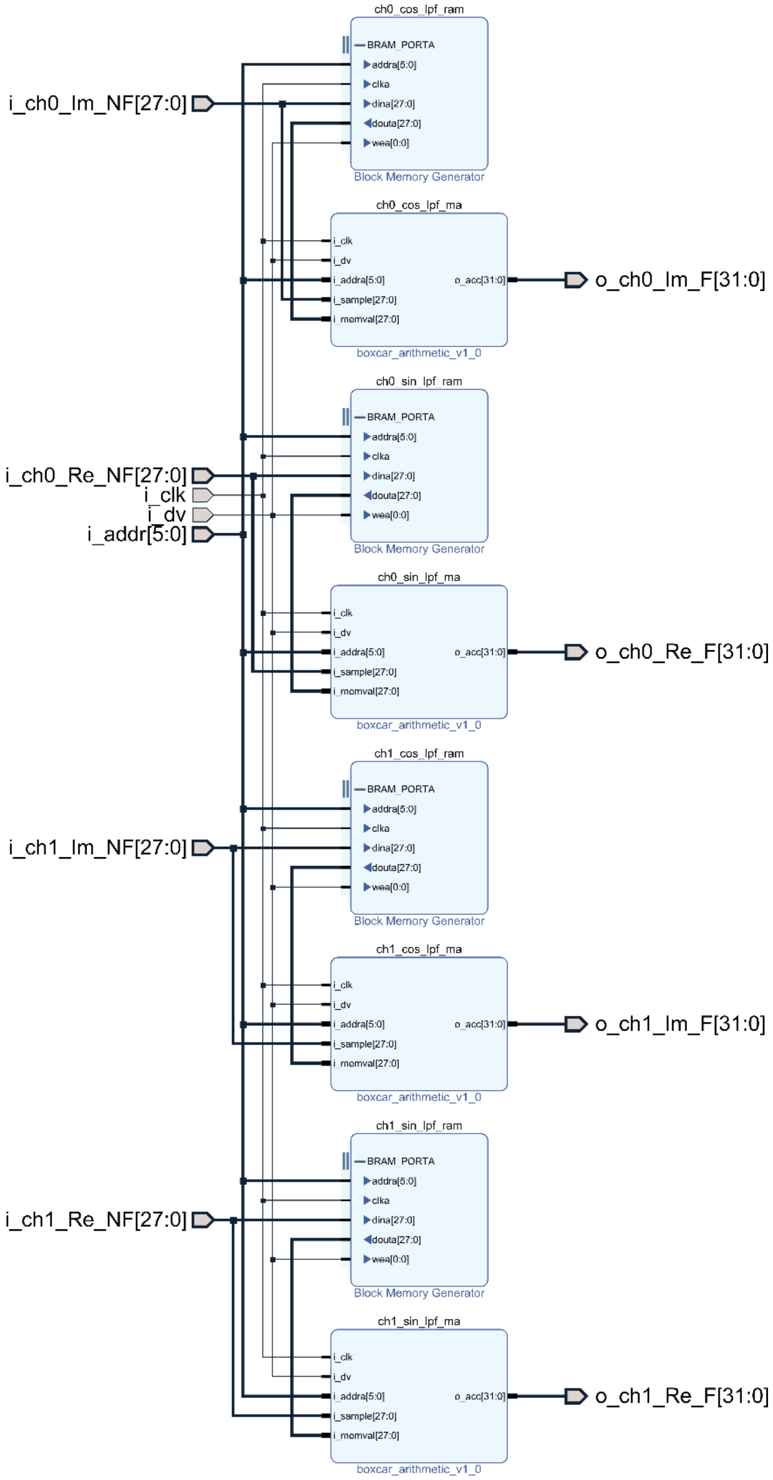


Figure 2.17. Filtering Blocks in Vivado

2.1.7 CIC Filter Based Low Pass Filter

In the case of multi-device lock-in with PLL, due to increased number of filters, more aggressive filtering is required, as well as more RAM area. Cascaded Integrator Comb Filters (CIC) are used for recursive averaging mentioned in (2.16) without a BRAM, the filter is updated at every Nth samples and has the same frequency and impulse response, while only suffering from register growth as the number of samples taken is increased [35]. CIC filters have the same sinc function resembling frequency response as the one in BRAM Based Moving Average Low Pass Filter section. Achieving the same attenuation with only one adder, one subtracter, two delay elements and a decimator, they are much simpler to implement in hardware. The design parameters are selected such that pass-band attenuates most of the high frequency components, and the stop-band is below a certain dB.

Introduced by Hogenauer in 1981 [36], z-domain transfer function is the same as (2.19). Z-domain transfer function is given in (2.22), where N is the total filter length/delay.

$$H_{CIC}(z) = \frac{1-z^{-N}}{1-z^{-1}} \tag{2.22}$$

Certain manipulations are performed on the transfer function such that it is computationally efficient and satisfies the original transfer function. The manipulated version is seen in the simplified block diagram of a decimating CIC filter in Figure 2.18.

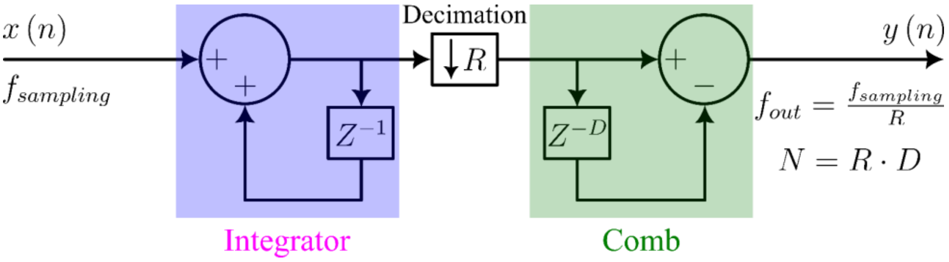


Figure 2.18. Single stage, single differential delay CIC filter [37]

CIC filter delay N is dependent on both the differential delay D and down sampling ratio R through the following relation,

$$N = R \cdot D \tag{2.23}$$

Changing either R or D affects the number of zeros present on the unit circle in z-domain. Hence, the decision should be made accordingly. It is also worth noting down that the output register should be adjusted so as to prevent overflow errors. As a result, the following equation should be taken into account while designing the filter[37]. R is the decimation ratio and D is the differential delay of the filter. For this thesis, M and D are taken as 1 while R is the design parameter.

$$width_{output} = width_{input} + \left[log_2 \left(R \cdot D \right) \right] \tag{2.24}$$

As it is known that both the ADC and DAC has a resolution of 12 bits, their resulting multiplication is going to have a total width of $width_{input} = 24\,bits$. For a differential delay $D = 1$ and decimation ratio of $R = 2^{16}$, the output register should have a width of $width_{output} = 40\,bits$. An example of the frequency response of the filter implemented is given in Figure 2.19. The horizontal axis stands for the normalized frequency which when multiplied with the sampling frequency, yields the frequency with the unit of sampling frequency. Vertical axis shows the magnitude of the filter in dB. Decimation ratio R is chosen as $2^{16} = 65536$. Hence, the filter length is equal to R, where in total 65536 equally spaced filter zeroes on unit circle in z-domain exist. Stopband and half of the bandwidth are shown on the first figure. The bottom figure shows the point where the signal attenuation is below -70 dB on a zoomed-out view of the frequency response. For a sampling rate of $10^6\,Hz$, normalized frequencies of the marked data are as follows. It is clear that this filter is effective for filtering out the AC components from the mixed signals of the resonators operating at kHz region. In fact, as long as the resonators operate at least 20 Hz apart from each other, this filter is able to filter the beat frequencies which result as a result of mixing multiple resonator signal in a single channel. In other words, image frequencies exist after heterodyning the resonating frequencies of multiple DUTs.

$$\begin{aligned}
F_1\left(Cutoff\ Frequency\right) &= \underbrace{\frac{6.74155}{10^6}\frac{cycles}{sample}}_{f_1}\cdot 10^6\,\frac{samples}{second} \cong 6.74\,Hz \\
F_2\left(Stopband\ Frequency\right) &= \underbrace{\frac{1.23415}{10^5}\frac{cycles}{sample}}_{f_2}\cdot 10^6\,\frac{samples}{second} \cong 12.34\,Hz \\
F_3\left(Frequency\,@-70\,dB\right) &= \underbrace{0.0150538\,\frac{cycles}{sample}}_{f_3}\cdot 10^6\,\frac{samples}{second} \cong 15\,kHz
\end{aligned} \tag{2.25}$$

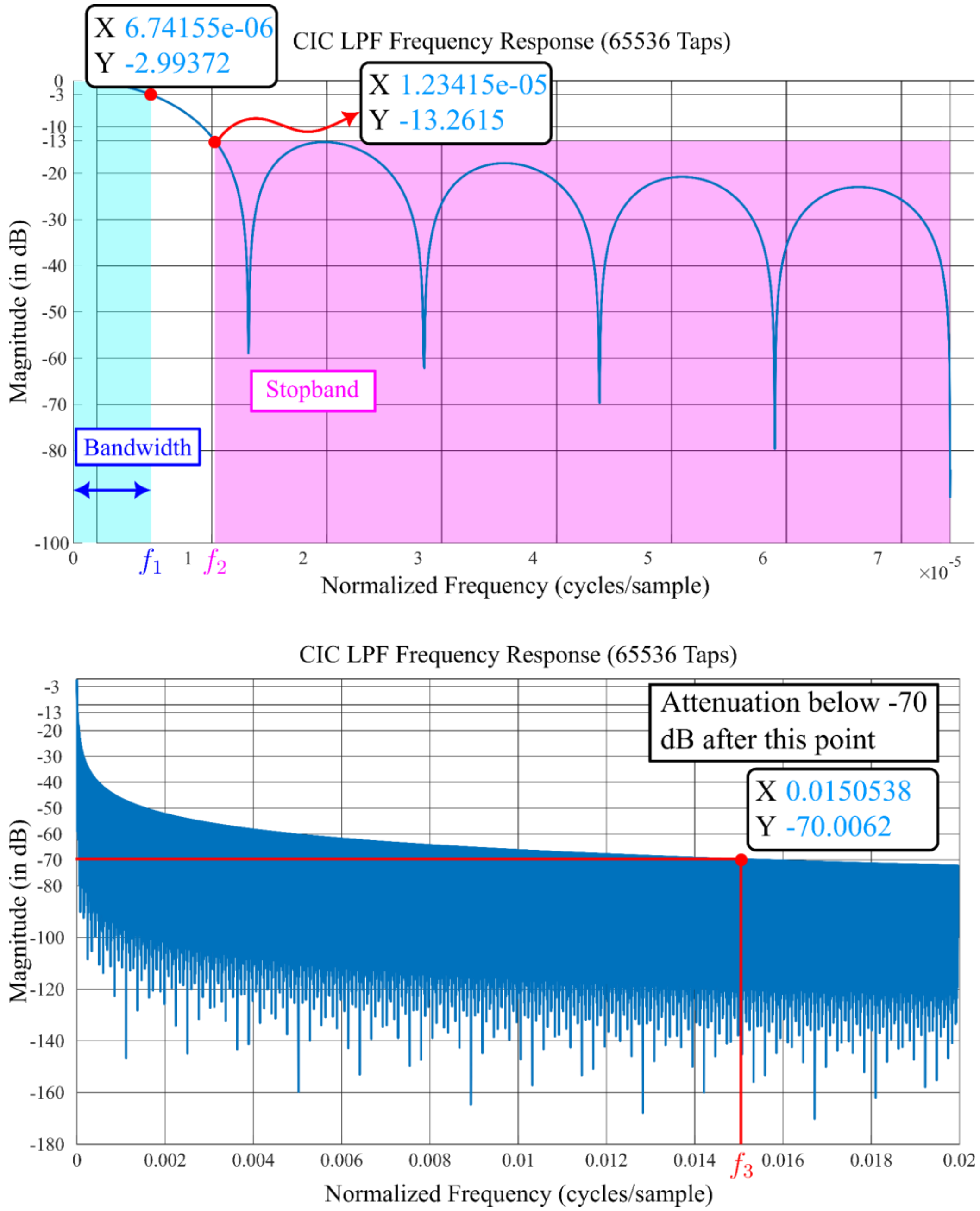


Figure 2.19. CIC Filter Frequency Response

IP blocks created in Vivado based on the VHDL source code are shown on Figure 2.20. Similar to Figure 2.17. Filtering Blocks in Vivado, for every resonator there are 4 filtering elements here as well. The only difference is lack of RAM usage.

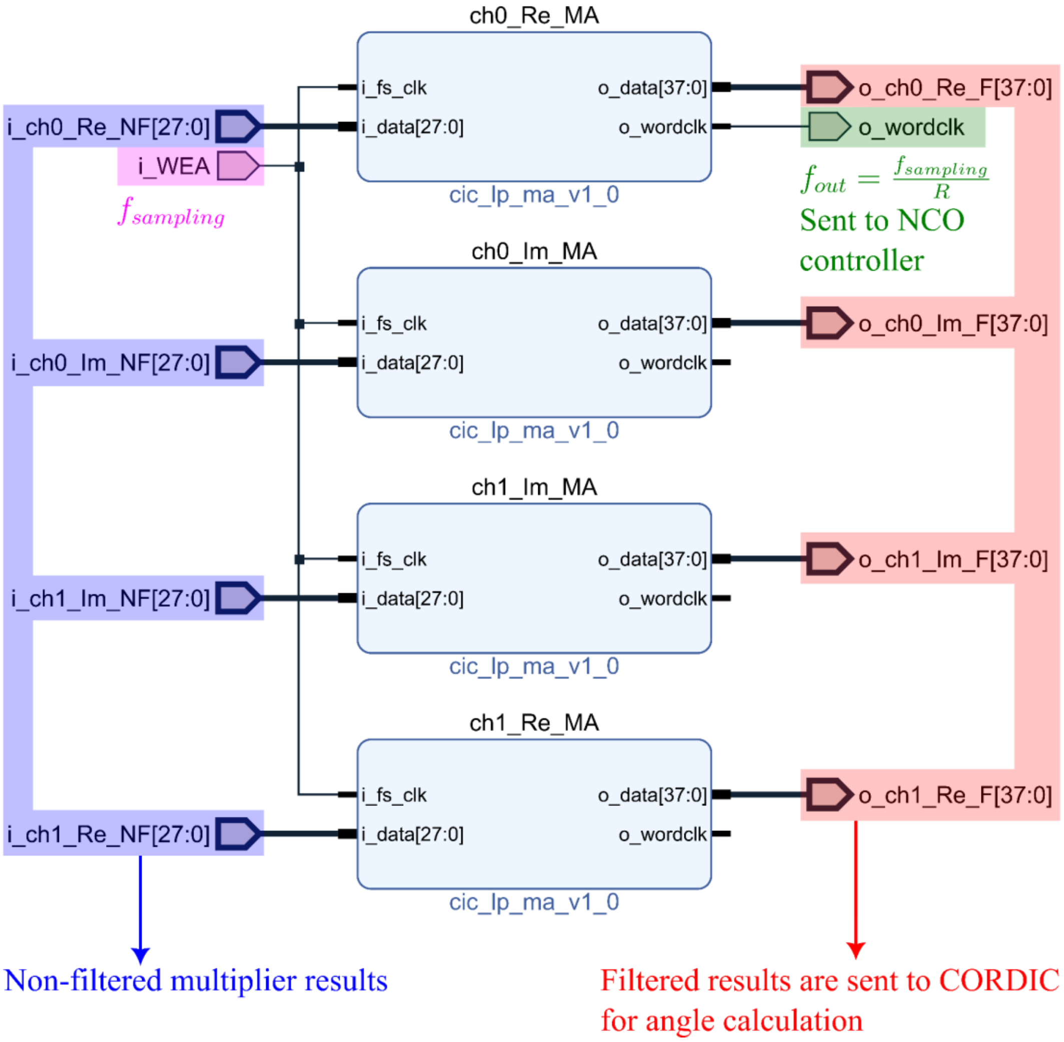


Figure 2.20. CIC IP Blocks in Vivado

2.1.8 COordinate Rotation Digital Computer (CORDIC)

Originally developed for real-time airborne computation for identifying the relative position of targets on the radar, CORDIC has been used for calculating trigonometric identities precisely, using only adders, shifters and ROM/LUT [38]. This is especially useful for FPGAs, as they only possess those components. The CORDIC algorithm is used in this thesis for calculating the phases defined in (2.11) and (2.12) , using quadrature components found, as a result of LPF. Although the CORDIC can be used for calculating any trigonometric function, for lock-in detection ATAN2 is the function of choice. ATAN2 accepts two cartesian inputs (x, y) and outputs the phase in polar coordinates. The vector defined by these cartesian inputs is as follows,

$$\vec{r} = x + jy \tag{2.26}$$

If this vector $\vec{r}$ were to be rotated on $x - y$ plane with a known angle of θ, it would look like the following expressions.

$$x' + jy' = (x + jy)e^{j\theta} = (x + jy)(\cos\theta + j\sin\theta) \tag{2.27}$$

Rearranging,

$$x' + jy' = (x\cos\theta - y\sin\theta) + j(x\sin\theta + y\cos\theta) \tag{2.28}$$

Dividing both sides with $\cos\theta$,

$$\frac{x'}{\cos\theta} + j\frac{y'}{\cos\theta} = (x - y\tan\theta) + j(y + x\tan\theta) \tag{2.29}$$

CORDIC suggests the rotation of the vector $\vec{r}$ recursively around the origin, where $x \& y$ coordinates are updated after every rotation, until the vector is rotated the desired amount. The recursive version of (2.29) is as follows,

$$x_{i+1} = x_i - y_i \tan\theta_i \tag{2.30}$$

$$y_{i+1} = y_i + x_i \tan\theta_i \tag{2.31}$$

Knowing that multiplication or division with the power of 2 is a shifting operation in digital design, one can manipulate the above written expressions into ones more compatible with FPGAs. Let,

$$\tan\theta_i = 2^{-i} \quad \text{where} \quad i = 0, 1, 2, \ldots, \#ofRotations \tag{2.32}$$

In that case, the rotation angles for each iteration respectively would be as follows for a vector lying inside the 1st quadrant, where R is the number of CORDIC rotations which is taken as 28 for this thesis,

$$\begin{array}{lll} i=0 & \tan\theta_0 = 1 & \theta_0 = 45^\circ \\ i=1 & \tan\theta_1 = 1/2 & \theta_1 = 45/2^\circ \\ i=2 & \tan\theta_2 = 1/4 & \theta_2 = 45/4^\circ \\ \vdots & \vdots & \vdots \\ i=R & \tan\theta_R = 1/2^R & \theta_R = 45/\left(2^R\right)^\circ \end{array} \tag{2.33}$$

Depending on the initial position of the vector, whether it is inside the white, blue, green or yellow region given in Figure 2.21, initial rotation amounts are $-45^\circ, -135^\circ, -225^\circ, -315^\circ$ respectively. Thus, θ_0 is either $45^\circ, 135^\circ, 225^\circ$ or 315°.

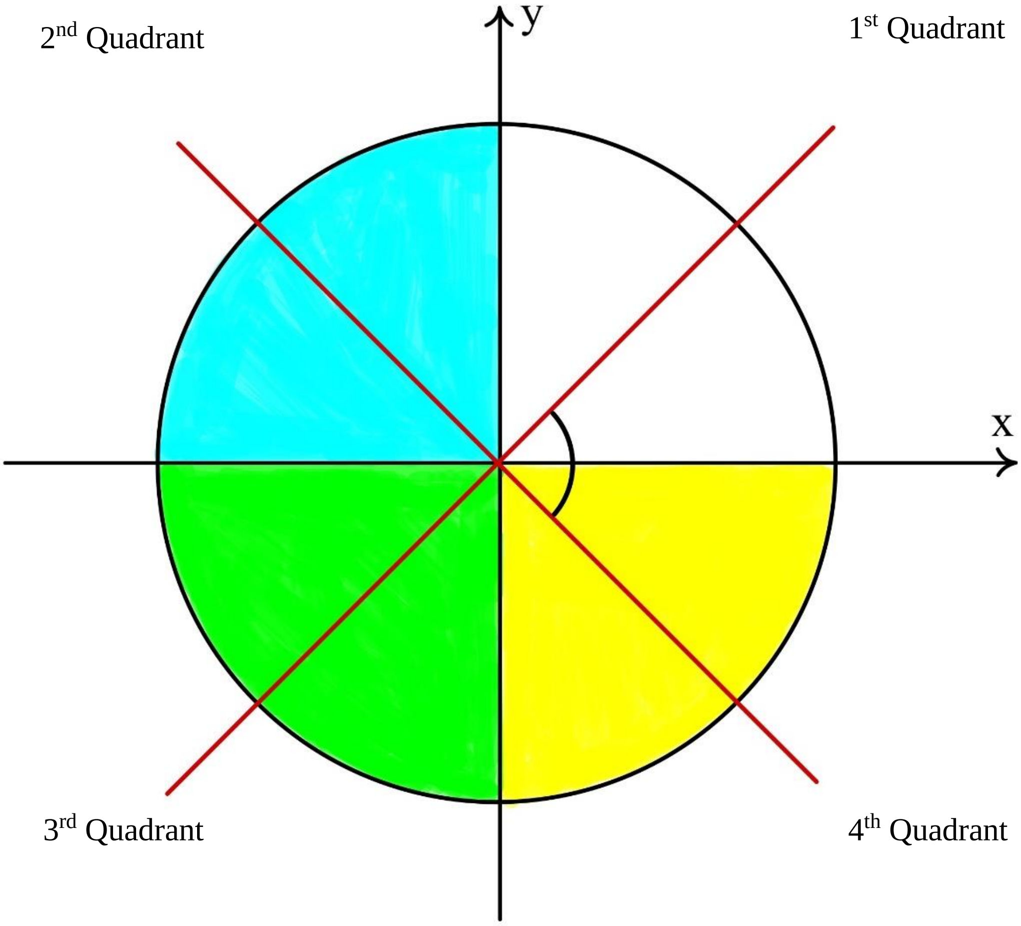


Figure 2.21. CORDIC Initial Rotation

Based on whether y coordinate of the vector is positive or negative, the CORDIC phase is accumulated and outputted in the end.

$$\theta_{CORDIC} = \sum_{i=0}^{R-1} sign(y_i)\theta_i \tag{2.34}$$

The whole algorithm is shown on a flowchart in Figure 2.22. Based on the flowchart, VHDL implementation is written. The flowchart suggests a loop to be used, which would have a loop execution time if it had been implemented in a computer. However, within the FPGA, the whole loop can be executed within one clock cycle due to parallel execution capability of the digital hardware. Since the operations are performed in digital domain, one should map the angle θ to a register value. This is because the values stored inside the FPGA are not represented as decimal values. Thus, quantization is done the following way where PW is the phase width which is 31 for this thesis,

$$360^\circ = 2^{PW} = 2^{31} \Rightarrow 90^\circ = \frac{2^{PW}}{4} = 2^{29} \tag{2.35}$$

The angles in (2.33) are mapped as in (2.35) by multiplying them with 2^{PW} and rounding them to the nearest integer. These final values are stored in a LUT inside the FPGA. The LUT content is generated with a simple MATLAB code given in Appendix C. Final CORDIC angle register can be converted back into angle in degrees, by dividing the result with $2^{PW} = 2^{31}$.

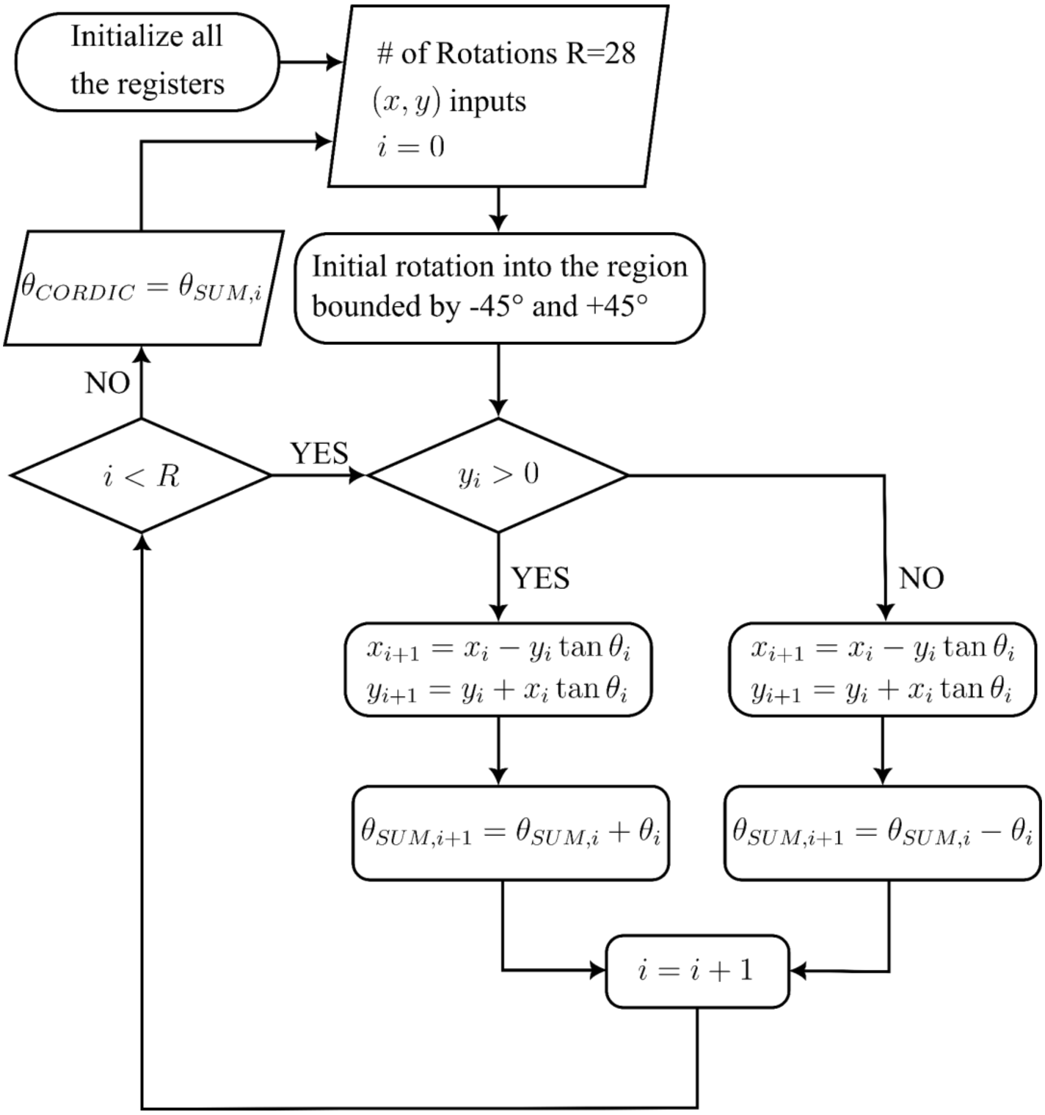

Initialize all the registers
# of Rotations R=28
(x, y) inputs
$i = 0$
$\theta_{CORDIC} = \theta_{SUM,i}$
Initial rotation into the region bounded by -45° and +45°
NO
YES
$i < R$
$y_i > 0$
YES
NO
$x_{i+1} = x_i - y_i \tan\theta_i$
$y_{i+1} = y_i + x_i \tan\theta_i$
$x_{i+1} = x_i - y_i \tan\theta_i$
$y_{i+1} = y_i + x_i \tan\theta_i$
$\theta_{SUM,i+1} = \theta_{SUM,i} + \theta_i$
$\theta_{SUM,i+1} = \theta_{SUM,i} - \theta_i$
$i = i + 1$

Figure 2.22. CORDIC Flowchart

Flowchart in Figure 2.22 is implemented inside the FPGA using Vivado. The implemented design can be seen in Figure 2.23. _x and _y suffixes represent the Real and Imaginary parts of the incoming signal in phasor domain. o_phi_0 & o_phi_1 signals represent the input side and output side phase values respectively, whose phase values are calculated as with respect to the internally generated NCO signals. The outputs are then sent out to the controller block for phase error calculation.

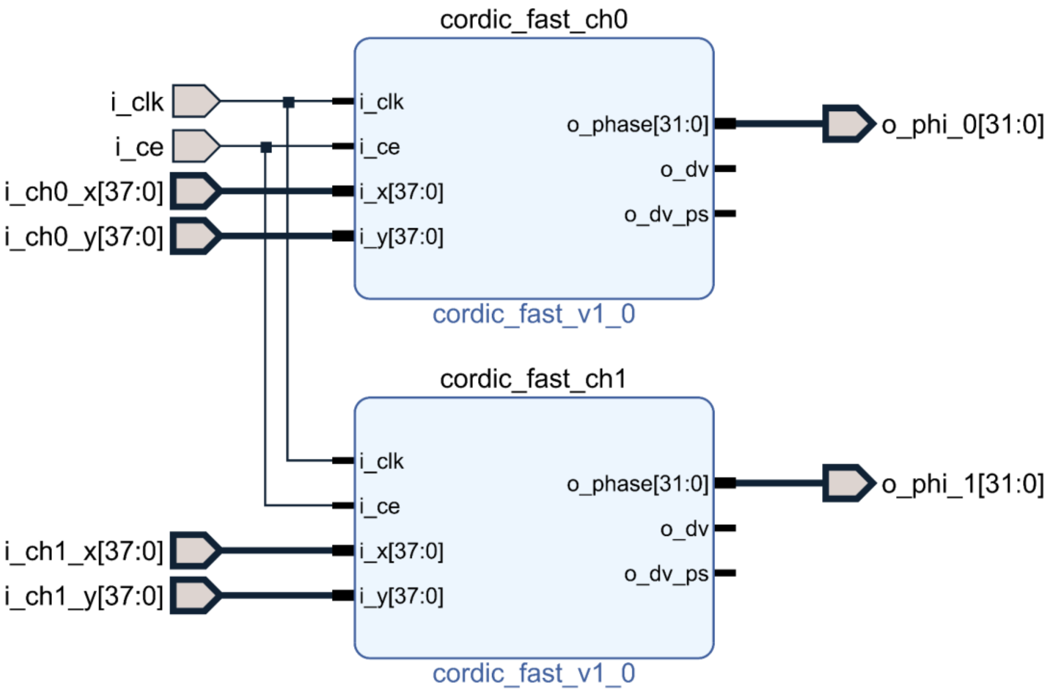


Figure 2.23. CORDIC Blocks in Vivado

Simulation results prove that in fact the CORDIC angle calculations take one clock cycle after the inputs are latched in at the rising edge of the clock. The test bench that is run for 70 ns is given in Figure 2.24. The input clock has a period of 10 ns. In order to show that the algorithm works in every quadrant, distinct x & y values are provided at the rising edge of the input clock. CORDIC block general parameters are defined as IW (input width), Nstages (Number of CORDIC Stages), WW (word width used inside the algorithm) and PW (Phase width). WW is chosen much greater than the IW such that the register growth due to calculations does not affect the algorithm. It is clear from the figure that after the states of x & y change, the CORDIC phase result becomes valid at the next rising edge of the input clock. Since the CORDIC block is expected to update its output whenever the moving average low pass filter’s output is valid, this speed is more than enough for the application at hand.

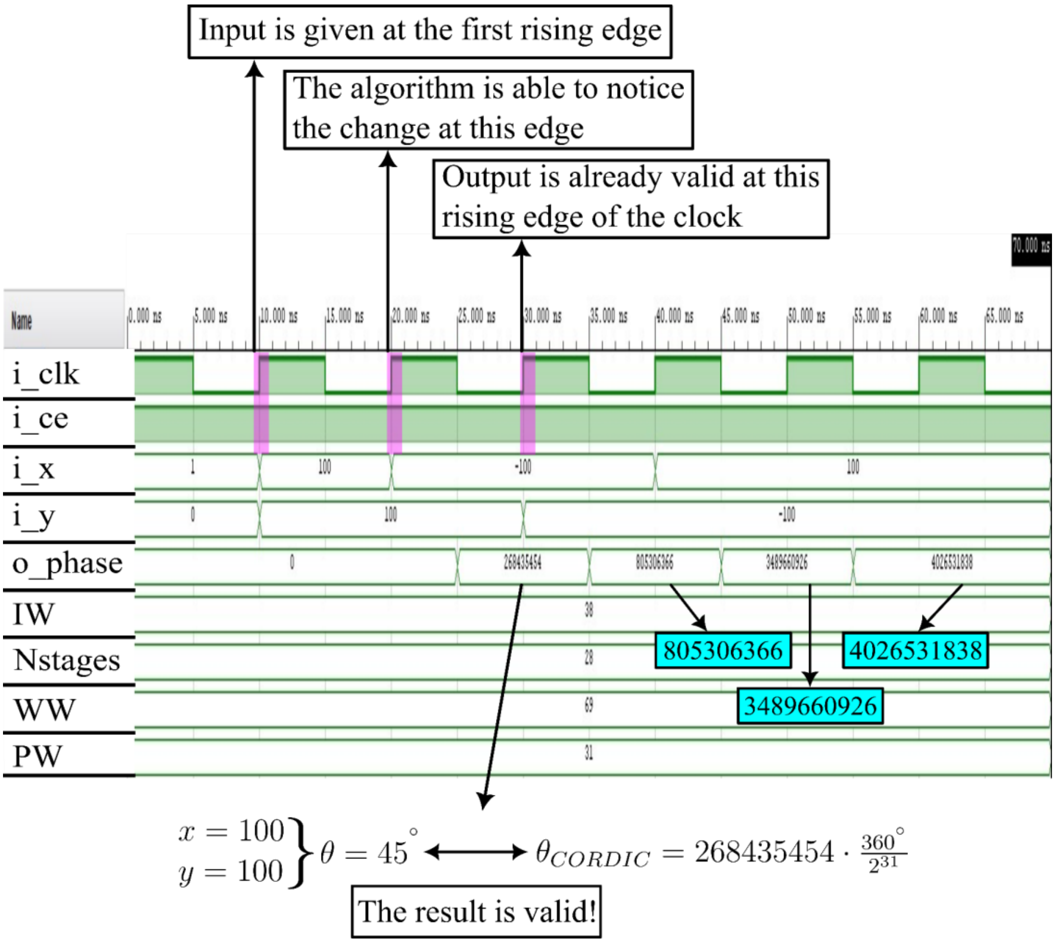


Figure 2.24. CORDIC Block Simulation Result in Vivado

The true error, considering the simulation in Figure 2.24, calculated for $x = 100, y = 100$ is as follows,

$$\varepsilon_{true} = \frac{\phi_{true} - \phi_{simulation}}{\phi_{true}} = \frac{\frac{2^{31}}{8} - 268435454}{\frac{2^{31}}{8}} = 7.45 \cdot 10^{-9} \qquad (2.36)$$

It is obvious that the angle is calculated with an extremely low true error using only adders, shifters and LUTs. However, it may be more desirable to use a CORDIC with a worse true error, where such an extremely low true error might be unnecessary. Although the error decreases with the increased number of CORDIC rotations, FPGA resources gets consumed a lot more. As a result, for a specific design, CORDIC rotations should be optimized.

2.1.9 Closed Loop Controller

To lock onto a frequency for which the phase is equal to the reference phase defined by the user, a controller is to be designed. Thankfully, due to the linear nature of the resonators, frequency and phase relation around the region of interest, resonance, is linear. As a result, an Integral-controller (I-controller) is more than enough to be used inside the NCO Controller shown Figure 2.4. Nevertheless, the phase response around the resonance is to be determined beforehand for each resonator so as to adjust the controller gain of the I controller as each resonator's response is expected to be different, hence an independent control loop for each resonator is required. Since the gain of the I-controller essentially converts the phase error into frequency, its unit can be expressed as, where K_I is the gain of the controller,

$$K_I = \frac{frequency}{phase\,error} \tag{2.37}$$

The gain of the I-controller is equal to the slope of the frequency vs phase-error graph shown in Figure 2.25. The frequency word found from the graph corresponding to the reference phase is fed back into DDS. One should note that in order for these loops to work, the initial driving frequency should lie inside this linear response region, close to the maximum SNR or resonance frequency. One can think of the search for resonance algorithm as the bisection method used for numerical root finding, where the initial guess is critical for finding the root of a curve. Although bisection method could have been used to find the necessary driving frequency, there is no need for root searching algorithms as the frequency corresponding to the reference phase can easily be found by a simple arithmetic operation. The line tangent to the linear region of the phase response happens to be the DDS frequency vs phase-error graph shown in Figure 2.25. It is derived from the phase angle vs DDS frequency graph shown in the same figure. The line equation is written as follows where $a\,\&\,b$ are the points on phase response curve between which the response is linearized. Point r on the line represents the reference, whose phase angle is provided by us.

$$\frac{f-f_r}{\phi-\phi_r}=\frac{f_a-f_b}{\phi_a-\phi_b}=K_I \rightarrow f=K_I\left(\phi-\phi_r\right)+f_r \tag{2.38}$$

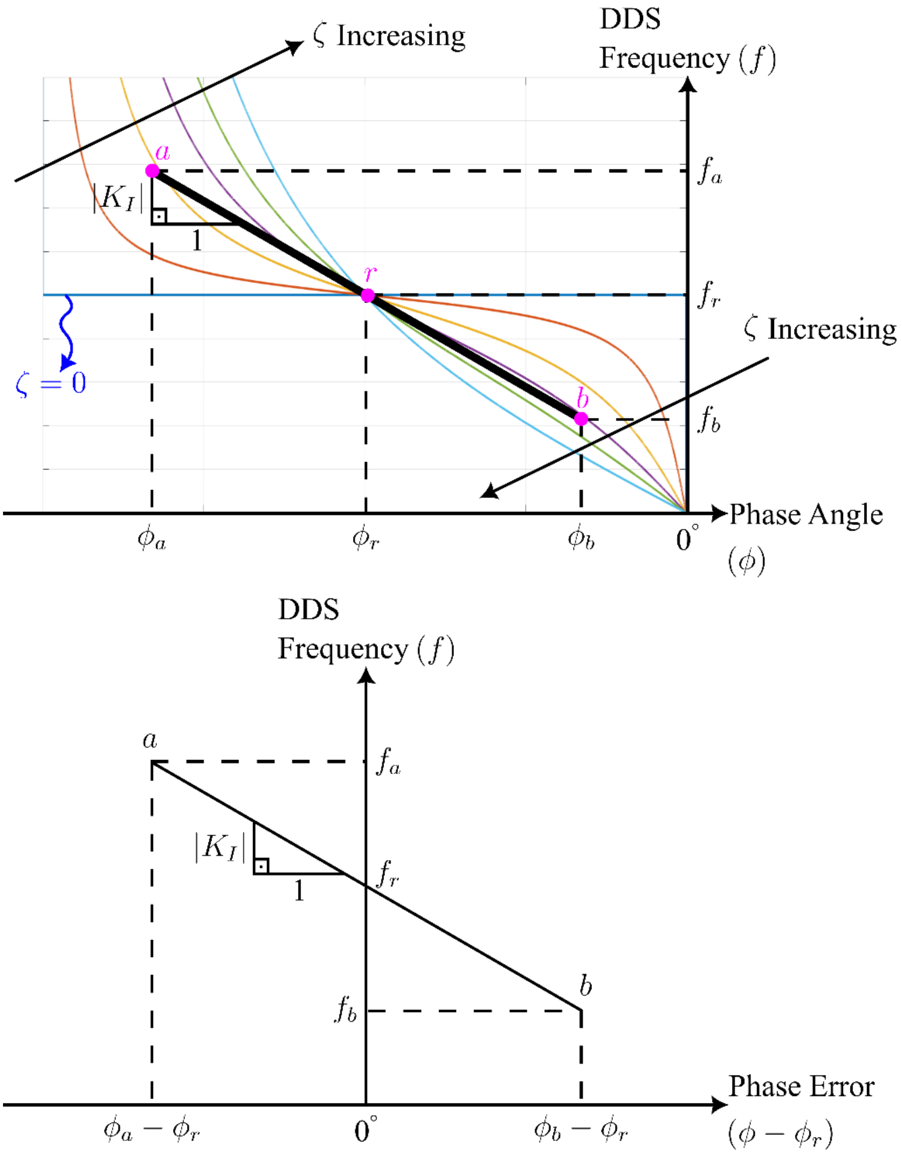


Figure 2.25. Frequency vs Phase Angle and Frequency vs Phase Error

One should note that after finding the controller gain K_I equal to the slope, the reference frequency, f_r, corresponding to the reference phase given as an input , ϕ_r, is easily derived from (2.38) where,

$$f_r = f - K_I\left(\phi - \phi_r\right) \qquad ,-1 < K_I < 0 \tag{2.39}$$

That is because f is the known DDS word, and ϕ is the instantaneous phase calculated by the lock-in algorithm. The only unknown is f_r which is to be calculated by the algorithm inside the FPGA.

The slope of the line shown on Figure 2.25 has a slope of K_I, which is bounded as,

$$-1 < K_I < 0 \tag{2.40}$$

The reason is that around the resonance/maximum SNR point, the change in phase is more than the unit change in frequency. As a result, the phase is highly sensitive to the change in frequency, hence the expression (2.40) is valid. In fact, thanks to this property, resonance detection based on frequency change is superior to that of amplitude-based detection. In order to implement the equation (2.39) inside the FPGA using VHDL, division operation should be performed due to (2.40). Since division in digital domain is not a straightforward process, certain manipulations should be done. The procedure is as follows.

Division of a number A with B in a digital environment is done with multiplication and shifting operators. Manipulating the division, the following way will not change the result analytically.

$$Result = \frac{A}{B} = \frac{A}{B}\frac{2^n}{2^n} \tag{2.41}$$

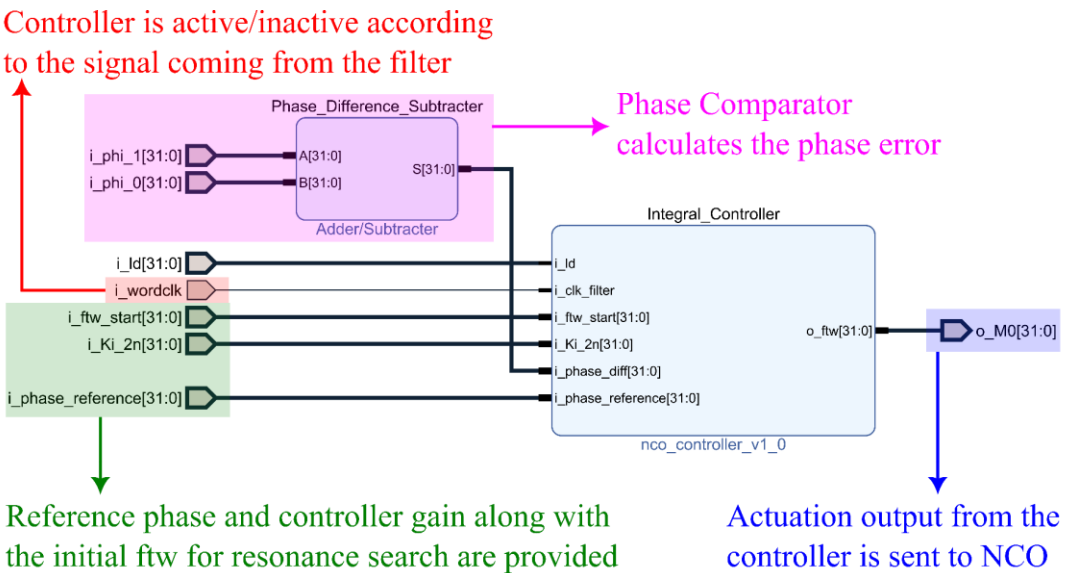
Multiplication with 2^n having a register width of n maps the decimal value of $\frac{1}{B}$ to a register value of width n. This n bit number is multiplied with A of width n, causing the result to have a width of $2n$. Then, division with $\frac{1}{2^n}$ shifts the result n bit to the right. As a result, integer part of the $\frac{A}{B}$ is found. The operation is shown as,

$$\frac{A}{B} = \underset{\text{Multiplication}}{A \cdot \frac{2^n}{B}} \cdot \underset{\text{Shift Right}}{\frac{1}{2^n}} \tag{2.42}$$

Applying the same procedure to (2.39) where $n = 32$ for this work,

$$f_r = f - \underbrace{\underbrace{K_I 2^n}_{\text{Multiplication for mapping}} \left(\phi - \phi_r\right)}_{\text{The result with a width of } 2n} \underbrace{\frac{1}{2^n}}_{\text{Shifting the result } n \text{ bit to right}} \tag{2.43}$$

Note that f_r & f are related to DDS ftw input M_0, and ϕ & ϕ_r angles are stored as raw CORDIC angles described in (2.35). Thus, it is more convenient to find the constant K_I in terms of $\frac{DDS \text{ ftw}}{\text{Raw } CORDIC \text{ Angle}}$ for the sake of digital signal processing. As it is difficult to find the constant from the frequency response for different systems, it is better to find the constant with trial and error as is done in this thesis. Controller constant in fact affects the response of the system, making the response either slower or faster to the disturbances. NCO Controller IP Block is shown in Figure 2.26.


Controller is active/inactive according to the signal coming from the filter

Phase Comparator calculates the phase error

Reference phase and controller gain along with the initial ftw for resonance search are provided

Actuation output from the controller is sent to NCO

Figure 2.26. NCO Integral Controller IP in Vivado

In the case of a multiple resonator configuration, for each of the DUT, a separate controller is in effect where each of them drives their corresponding DDS Blocks. However, the idea remains the same, hence distinct controller constants are provided by the user for the specific multi resonator system. The controller is integrated into the overall design as shown in Figure 2.27. The name given to the controller in this case is NCO_Controller, where it produces the control output for quadrature and in-phase NCO signals with certain frequencies. The controlling action is determined inside the controller based on the output of CORDIC_Blocks and the reference phase difference value provided by the user.

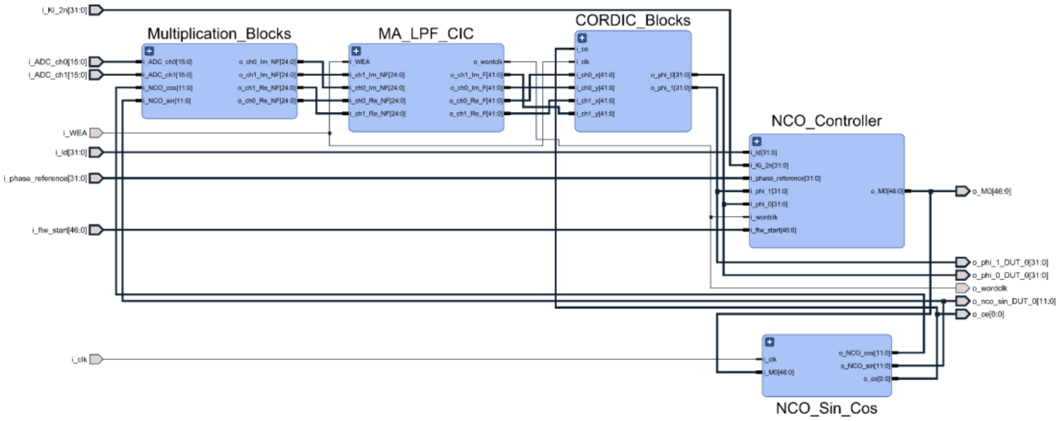


Figure 2.27. Closed loop design for a single DUT

The closed loop design given in Figure 2.27 is repeated and put together as shown in Figure 2.28. The design is capable of resonating three devices simultaneously. Since the design is highly scalable, the only limiting factor for driving more resonators is the FPGA resources available. In fact, the design for multiple DUTs had to be implemented on a more capable FPGA such as XC7Z020-CLG484 available on Zedboard. Although a breadbordable FPGA, Cmod-A7-35T was more than necessary for a single resonator, it was not capable for a bigger design. Additionally, for quartz resonators, DDS is upgraded from 32-bit to 47-bit version so that the resolution of the NCO is below μHz . This way, high quality factor quartz resonators are kept in a resonance loop with a better frequency resolution, where oscillation around the resonance is possible with a much better steady state error.

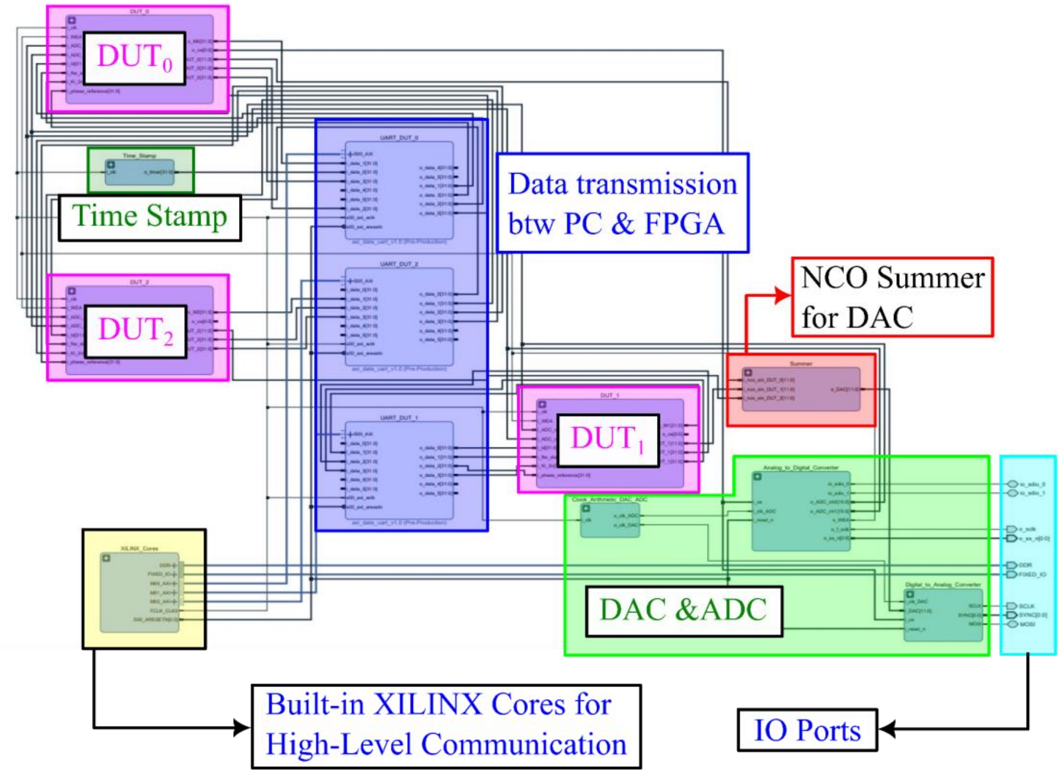
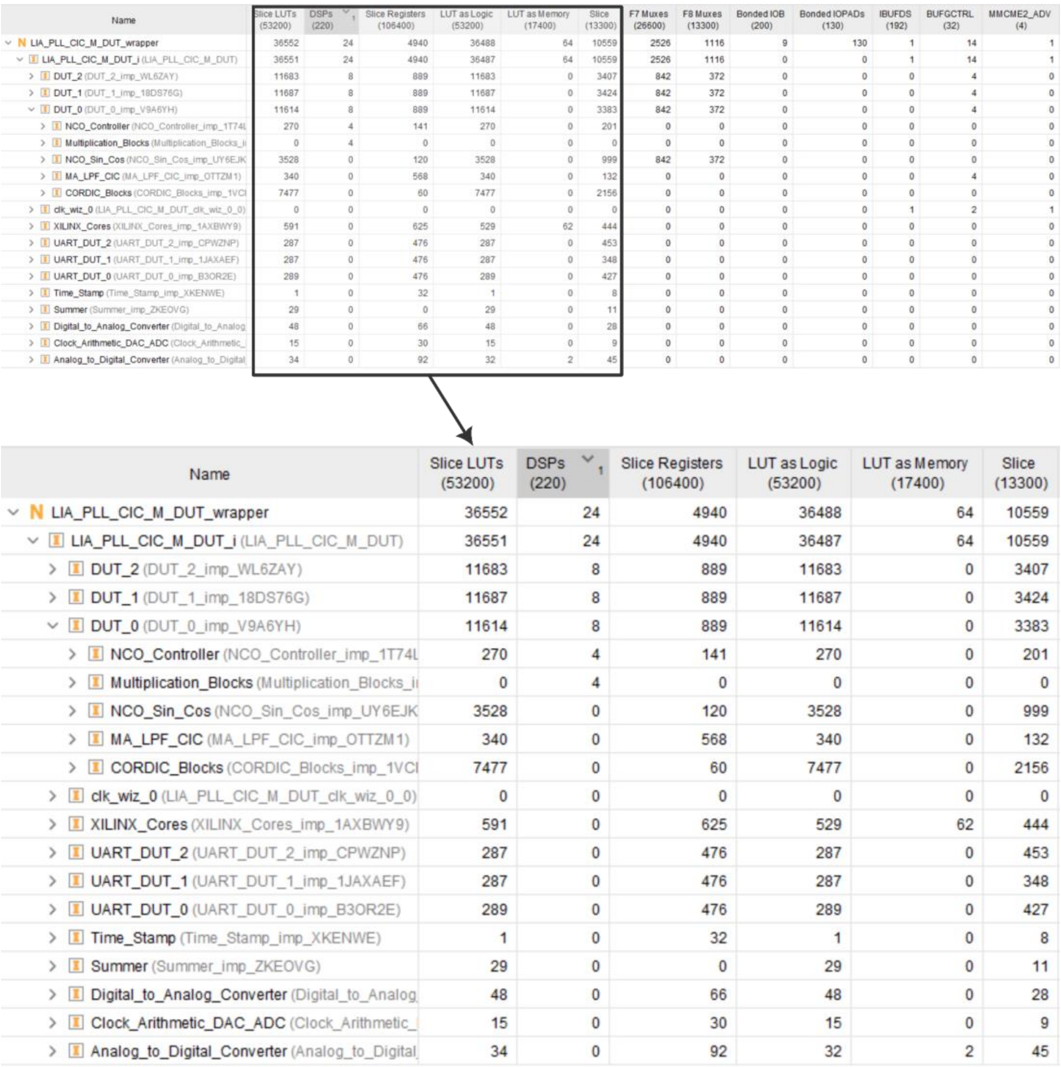

DUT$_0$
Time Stamp
DUT$_2$
Data transmission btw PC & FPGA
NCO Summer for DAC
DUT$_1$
DAC &ADC
Built-in XILINX Cores for High-Level Communication
IO Ports

Figure 2.28. Multi-DUT resonating design in Vivado

Resource utilization report is given as a table in Figure 2.29 for the Multi-DUT resonating design. As expected, CORDIC and NCO algorithms consume a lot more resources than the other IP blocks.



Name	Slice LUTs (53200)	DSPs (220)	Slice Registers (106400)	LUT as Logic (53200)	LUT as Memory (17400)	Slice (13300)	F7 Muxes (26600)	F8 Muxes (13300)	Bonded IOB (200)	Bonded IOPADs (130)	IBUFDS (192)	BUFGCTRL (32)	MMCME2_ADV (4)
LIA_PLL_CIC_M_DUT_wrapper	36552	24	4940	36488	64	10559	2526	1116	9	130	1	14	1
LIA_PLL_CIC_M_DUT_i (LIA_PLL_CIC_M_DUT)	36551	24	4940	36487	64	10559	2526	1116	0	0	1	14	1
DUT_2 (DUT_2_imp_WL6ZAY)	11683	8	889	11683	0	3407	842	372	0	0	0	4	0
DUT_1 (DUT_1_imp_18DS76G)	11687	8	889	11687	0	3424	842	372	0	0	0	4	0
DUT_0 (DUT_0_imp_V9A6YH)	11614	8	889	11614	0	3383	842	372	0	0	0	4	0
NCO_Controller (NCO_Controller_imp_1T74L	270	4	141	270	0	201	0	0	0	0	0	0	0
Multiplication_Blocks (Multiplication_Blocks_i	0	4	0	0	0	0	0	0	0	0	0	0	0
NCO_Sin_Cos (NCO_Sin_Cos_imp_UY6EJK	3528	0	120	3528	0	999	842	372	0	0	0	0	0
MA_LPF_CIC (MA_LPF_CIC_imp_OTTZM1)	340	0	568	340	0	132	0	0	0	0	0	4	0
CORDIC_Blocks (CORDIC_Blocks_imp_1VCI	7477	0	60	7477	0	2156	0	0	0	0	0	0	0
clk_wiz_0 (LIA_PLL_CIC_M_DUT_clk_wiz_0_0)	0	0	0	0	0	0	0	0	0	0	1	2	1
XILINX_Cores (XILINX_Cores_imp_1AXBWY9)	591	0	625	529	62	444	0	0	0	0	0	0	0
UART_DUT_2 (UART_DUT_2_imp_CPWZNP)	287	0	476	287	0	453	0	0	0	0	0	0	0
UART_DUT_1 (UART_DUT_1_imp_1JAXAEF)	287	0	476	287	0	348	0	0	0	0	0	0	0
UART_DUT_0 (UART_DUT_0_imp_B3OR2E)	289	0	476	289	0	427	0	0	0	0	0	0	0
Time_Stamp (Time_Stamp_imp_XKENWE)	1	0	32	1	0	8	0	0	0	0	0	0	0
Summer (Summer_imp_ZKEOVG)	29	0	0	29	0	11	0	0	0	0	0	0	0
Digital_to_Analog_Converter (Digital_to_Analog	48	0	66	48	0	28	0	0	0	0	0	0	0
Clock_Arithmetic_DAC_ADC (Clock_Arithmetic_	15	0	30	15	0	9	0	0	0	0	0	0	0
Analog_to_Digital_Converter (Analog_to_Digital	34	0	92	32	2	45	0	0	0	0	0	0	0

Figure 2.29. Resource Utilization Report

CHAPTER 3

CHARACTERIZATION AND TESTS

In this chapter, the digital signal processing algorithms brought together inside the FPGA are put under test. In doing so, lock-in amplification and frequency response capabilities are demonstrated. System setup, the results of these tests and the methodology are explained in detail. First of all, an RLC circuit is configured to be a position sensing device as a proof of concept for the following sections. Secondly, the design inside the FPGA is tweaked such that it is capable of taking the frequency response of a quartz crystal as they tend to have high quality factor. After proving the system works on a smaller scale, lock-in based DPLL algorithm is utilized for resonating multiple DUTs with the previously used setups. Frequency response and lock-in performance of the design are presented. Finally, it is shown that resonant bolometers can be driven with a similar fashion for measuring thermal changes in the environment.

3.1 A Simple RLC Circuit Based Position Sensor as a Proof of Concept

In this section, the digital phase locked loop employing lock-in amplifier is employed for converting a simple RLC circuit shown in Figure 3.1 into a highly precise and sensitive position sensor. As a matter of fact, bolometers are known to have been assumed as RLC circuits for simplification in some analyses. Therefore, being a good approximation of a resonance-based sensing device, this position sensing device has shown us, whether the digital design proposed in the previous chapter has been successful in achieving all its promises or not.

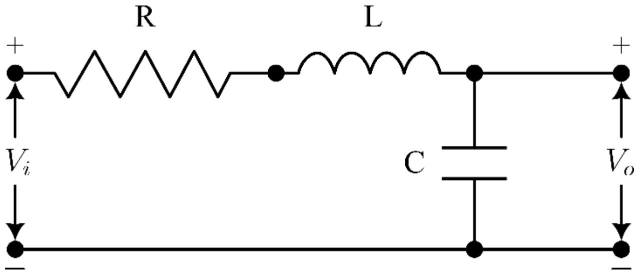


Figure 3.1. Simple RLC Circuit

The transfer function of the RLC system is known to be,

$$\frac{V_o}{V_i} = \frac{{}^{1}\!/_{LC}}{s^2 + \frac{R}{L}s + \frac{1}{LC}} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{3.1}$$

The natural frequency is shown to be related to the inductance and capacitance of the circuit. The resonance frequency although affected by the resistance, it is dependent on the natural frequency as well. In this circuit, a variable ferrite core inductor is used as an inductor element. The cylindrical metal core is displaced along the hollow core. As the displacement causes a change in the inductance value, affecting the resonance frequency of the circuit. This way, the change in resonance is related to position change of the metal core. Resonance is always achieved through the digital phase locked loop. This setup is used as a proof of concept for tracking the resonance of bolometers in the following sections as the idea of phase tracking remains the same. As a matter of fact, one can claim that the position sensor proposed in this chapter is some kind of low-cost version of LVDT (Linear Variable Differential Transformer), which is a precise position measuring electromechanical transducer [39].

3.1.1 System Setup and Circuit Design

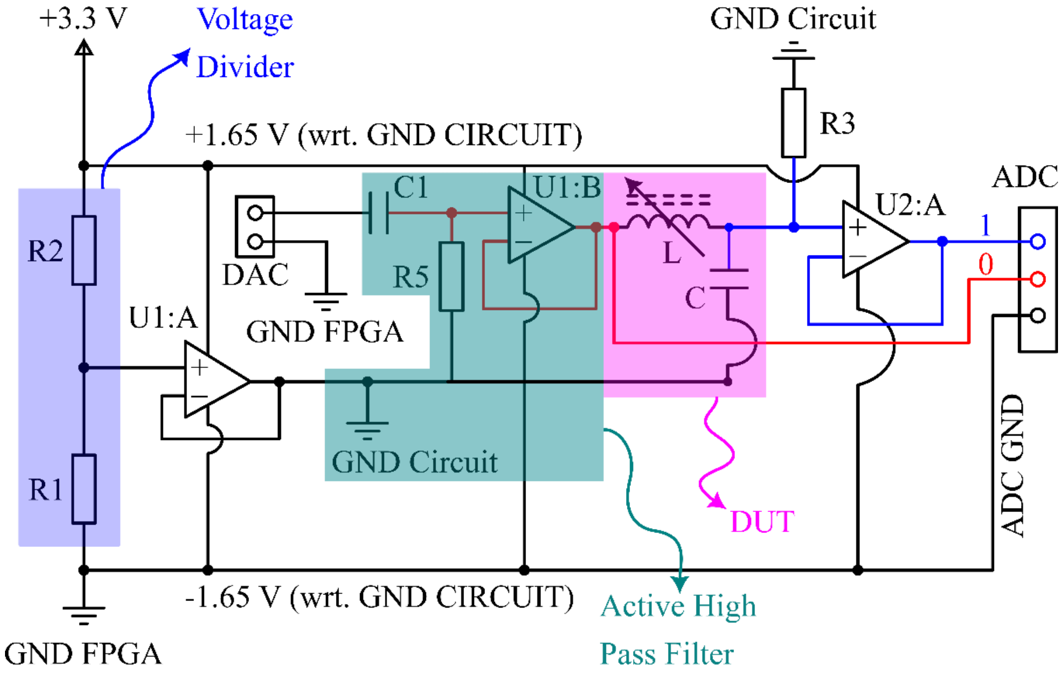


Figure 3.2. Circuit Diagram with RLC as DUT

The FPGA model used in this setup is a breadbordable Cmod-A7-35T. Additionally, ADC is a Digilent's Pmod AD1: Two 12-bit A/D Inputs, and the DAC used is Digilent's Pmod DA4: Eight 12-bit D/A Outputs. The FPGA is powered through USB connection to the PC, where the rest of the circuit is powered using the FPGA's onboard power outputs. Op-amps used in the circuit are of type LF353 and used as voltage divider and for isolating the DUT(s) from rest of the system. Hence, they are utilized as voltage followers. Since the input side of the op-amps are of high-impedance while the output sides are of low-impedance, ADC readings are performed through the low-impedance sides of the op-amps.

3.1.2 Position Measurement

The shift in resonance frequency as a result of the displacement of the solenoid core is related to position information. First, the position relation is performed for an open loop case, where the NCO controller is not in effect. In other words, the system is driven at a fixed frequency while the phase difference output constantly fluctuates due to the changing nature of phase response as a result of resonance shifting. However, it should be noted that the system is not operating at its maximum SNR point, hence the sensitivity of position sensing is expected to be different for each location. For the closed loop configuration, as the system is kept always at either resonance or maximum SNR, i.e., at the reference phase difference, sensitivity is at its highest value, for the phase response is steeper. The resolution of the sensing system is measured as well which is found to be decent for a system this simple.

The frequency response of the RLC DUT is obtained using Keysight’s DSOX1204G Digital Oscilloscope. There are two different frequency response measurements, one for the case where the rod is fully inserted inside the solenoid with a prefix of rod_ ,and the other is for the fully pulled out rod with a prefix no_rod. The shifting of resonance behavior is obvious and is expected to provide a decent amount of range for the measurements. The maximum SNR occurs around 500 Hz when the rod is fully inserted inside the variable ferrite core inductor, while it is about 1200 Hz if the ferrite core is not inside. The frequency response for both cases is given in Figure 3.3. Both the change in magnitude and phase diagrams are seen on the figure. As expected from equation (3.1) the resonance frequency increases as the ferrite rod is pulled outwards, decreasing the inductance of the system. The proposed range of operation for maximum SNR is shown on the figure as well. In that case, the reference phase value should be selected differently than $\phi_{reference} = -90^{\circ}$ for maximum sensitivity as the phase response steepness is greater around the maximum SNR points where $\phi_{reference} = -40^{\circ}$.

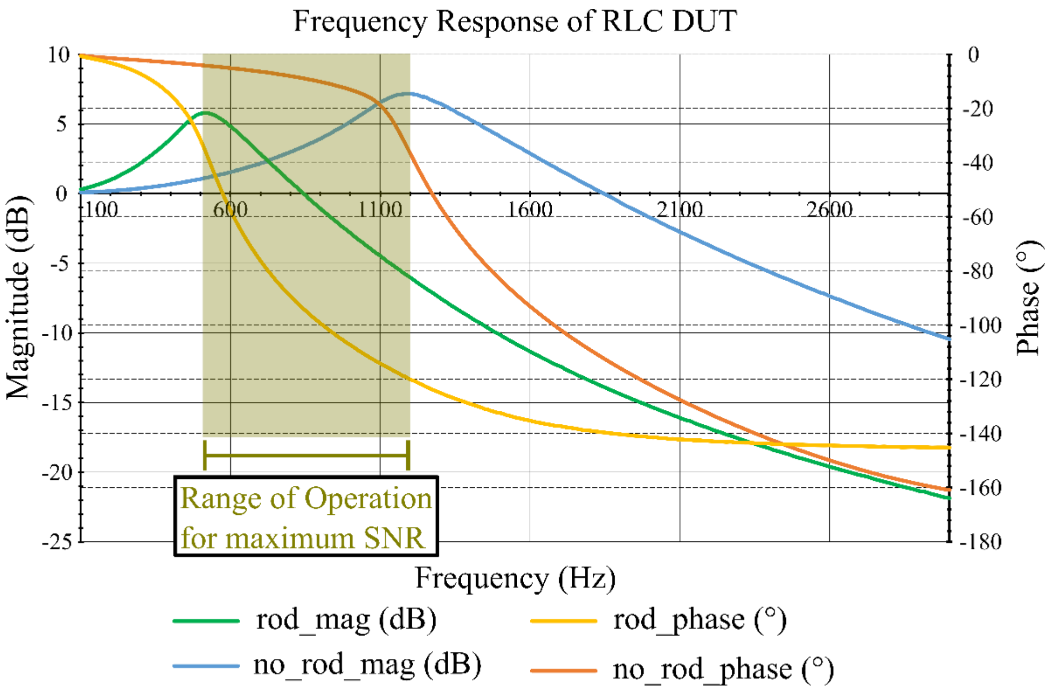


Figure 3.3. Frequency Response of RLC DUT

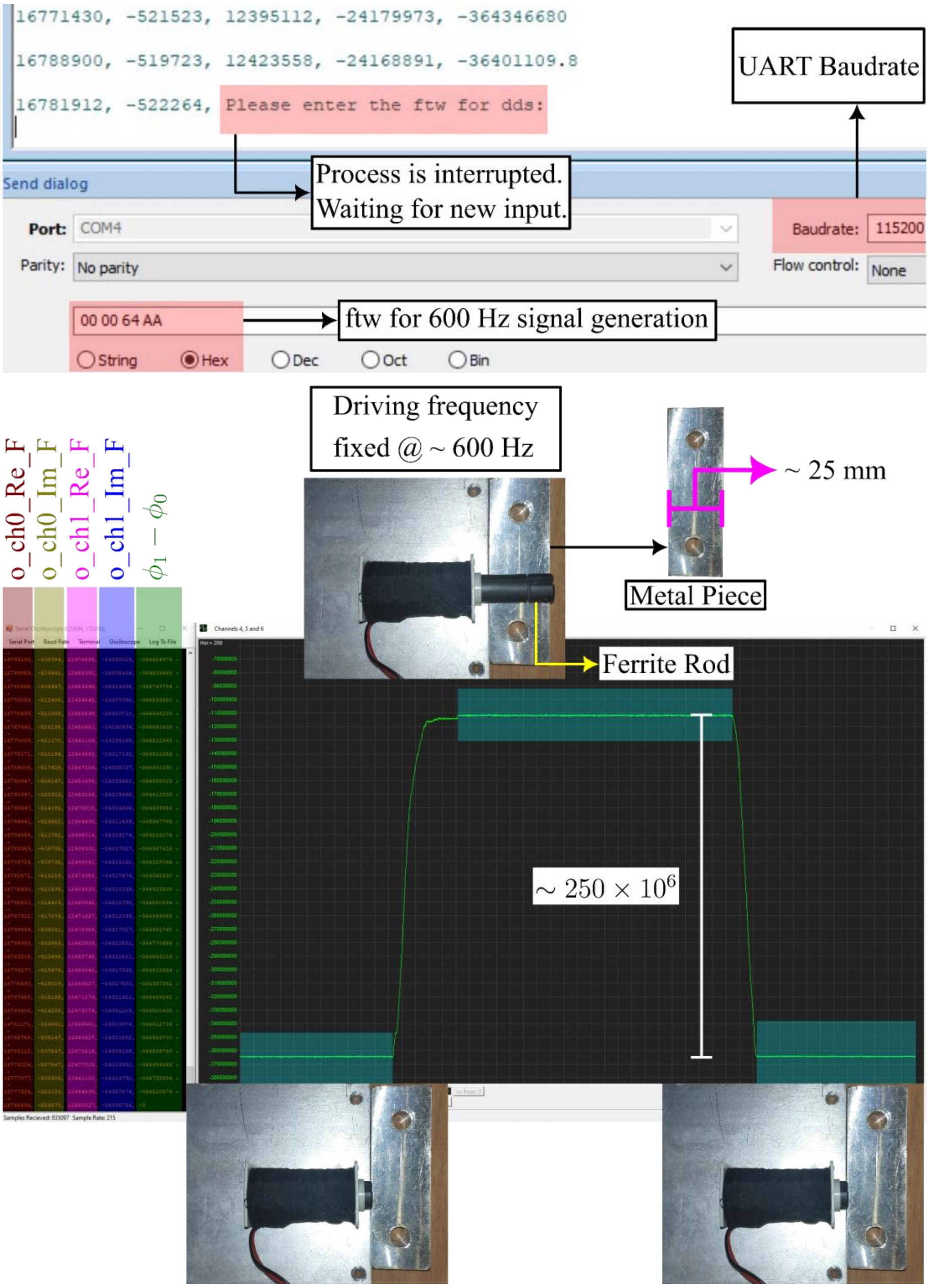
3.1.2.1 Open Loop Configuration

In this configuration, the NCO controller is off, and the system is driven at a fixed frequency. The DDS inside the FPGA is programmed by sending the ftw through UART protocol as a hexadecimal value. As a result, the FPGA sends out CORDIC input values along with the calculated CORDIC phase difference between the output and input shown in Figure 3.2. The ferrite rod is pushed manually on a rectangular metal piece for a total distance of 25 mm. The rod is then returned to its fully retracted original position. The process is depicted in Figure 3.4.

UART based user interface along with the system outputs is given in Figure 3.4. System response described above as a result of the ferrite core’s displacement is clearly seen. By default, the system starts from a predetermined DDS ftw. By sending 1, with a baud rate of 115200, the high-level process responsible from data transmission between the FPGA and PC is interrupted, waiting for the new DDS ftw.

Note that the low-level FPGA fabric keeps on working during this period, the interrupt only affects the user interface running at a higher level. The system is desired to be driven at 600 Hz, which lies in the beginning of the range of operation shown in Figure 3.3. Using the equation (2.14), ftw is calculated as 25,769.804 for a master clock of 100 MHz and a ftw width of 32. As the ftw can only attain an integer value, the ftw should be rounded to the nearest integer. In that case, ftw is selected as 25,770 resulting in a signal generation with a frequency of 600.004 Hz when plugged back into the (2.14). Hence, the ftw is safely selected as 25,770 for the NCO inside the DDS algorithm. Its 4-byte hexadecimal equivalent is equal to 00 00 64 AA. The driving signal has a peak-to-peak voltage of 0.5 V. The CORDIC phase difference is outputted in its raw format through UART. Moreover, the phase difference at the initial position is around -60° when the raw CORDIC data is converted into degrees according to (2.35). The result shows a correlation with the phase value measured at 600 Hz by Keysight’s DSOX1204G, shown in Figure 3.3. While the range is observed to be at an order of 250 million, the output fluctuates in the order of 1.5 million as shown in Figure 3.5 if the core does not get actuated.

This experiment’s whole purpose is to show the capability of the digital design implemented in this thesis. It shows how a simple, extremely low cost RLC circuit can be converted into a position sensing device. Thus, while evaluating the sensor performance parameters, it should be noted down.



```
16771430, -521523, 12395112, -24179973, -364346680
16788900, -519723, 12423558, -24168891, -36401109.8
16781912, -522264, Please enter the ftw for dds:
```

Port: COM4

Parity: No parity

Baudrate: 115200

Flow control: None

00 00 64 AA

○ String ● Hex ○ Dec ○ Oct ○ Bin

Figure 3.4. System Response and User Interface

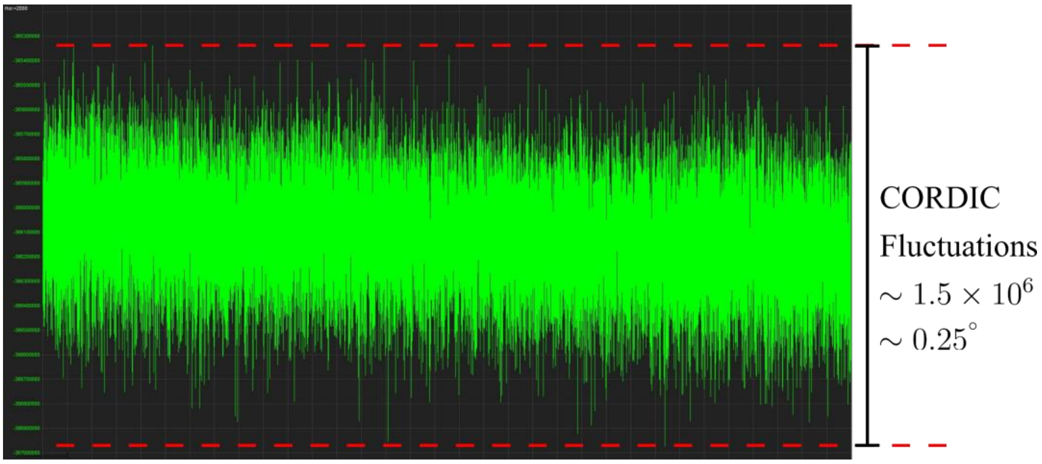


Figure 3.5. Output Fluctuations at a Fixed Position, raw CORDIC output [40]

Input and output signals are measured through Digilent's Analog Discovery 2 Device and given in Figure 3.6. The yellow sinusoid signal is quantized according to the digital design. Although quantization at much more points is possible for the DDS is capable of that, it is not necessary for this design as explained in the previous chapter. Other than that, blue signal is the signal measured at the output of the system according to the circuit schematic given in Figure 3.2. One can notice the change in phase difference between the yellow and blue signals, as the ferrite core is displaced.

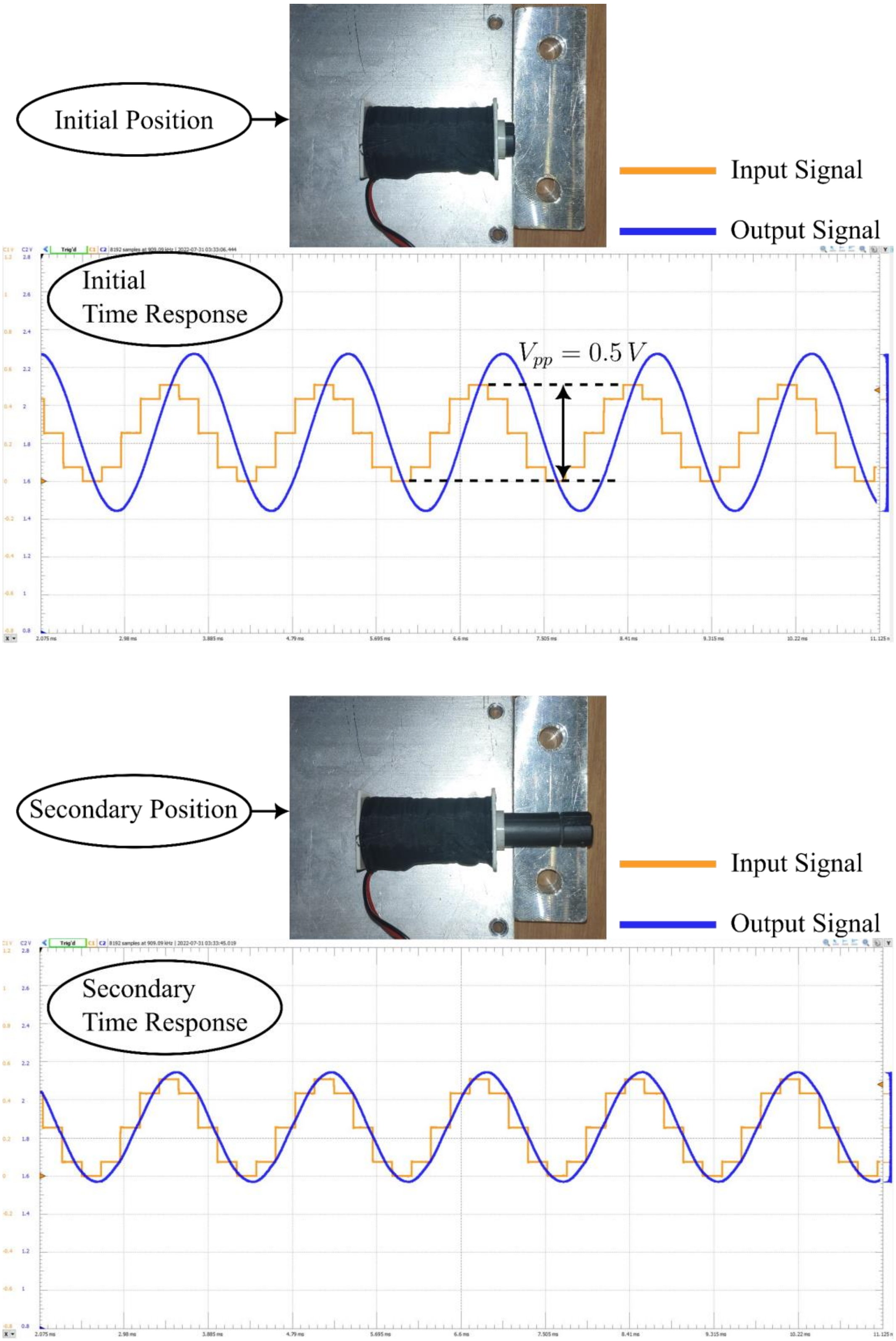


Figure 3.6. Oscilloscope Readings

3.1.2.2 Closed Loop Configuration with DPLL

In this part, the NCO controller is active where the digital PLL tries to lock onto the reference phase difference value given by the user through UART. The controller gain is provided by the user and is configurable as well. The system starts by default from 500 Hz, where the DDS inside the FPGA adjusts the driving frequency until the reference phase difference is achieved. The details are given in Closed Loop Controller part of the DESIGN chapter.

After consecutively sending the initial ftw, reference phase for the NCO Controller and the NCO Controller gain, the system gets activated. Raw data coming from the UART is collected by the PC and plotted. The ferrite core of the solenoid is dropped from a height of around 25 mm. The response of the DDS and the change in phase angle is shown in Figure 3.11. In order to lock onto the reference phase, DDS frequency is adjusted dynamically. The system is operated around its maximum SNR point corresponding to the -40° according to Figure 3.3. Although operation around -90° is possible, maximum sensitivity is achieved this way. Figure 3.11 shows how good the PLL is able to lock onto the reference phase. It takes around 150 ms for the system to reject the disturbances. The data sent over UART for adjusting the controller gain is empirically determined and chosen to be $100,000 = K_I 2^{32}$.

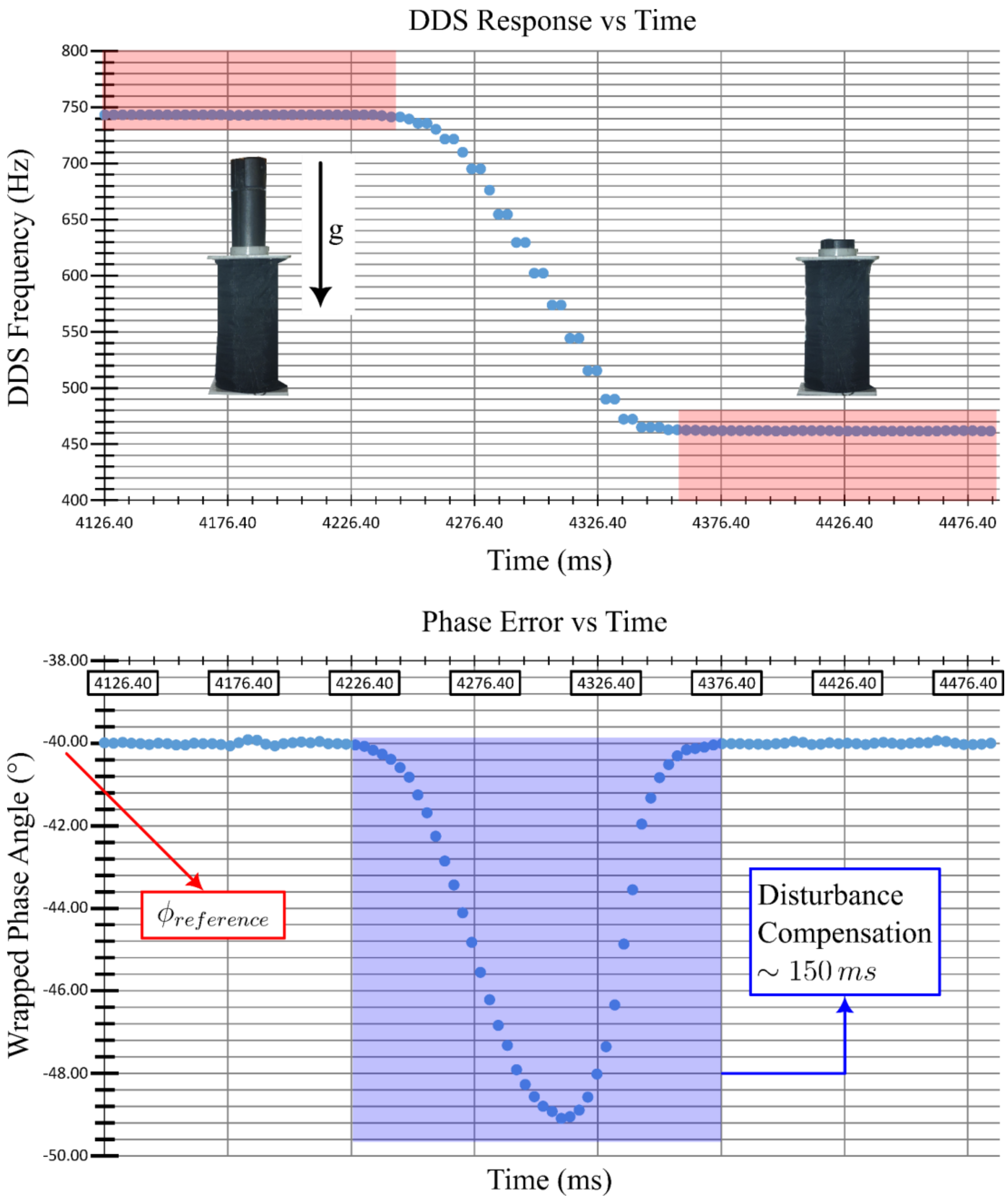


Figure 3.7. System Response for a Closed Loop RLC System

In order to calculate the resolution of the sensor, the raw data coming from the FPGA is inspected as depicted in Figure 3.8.

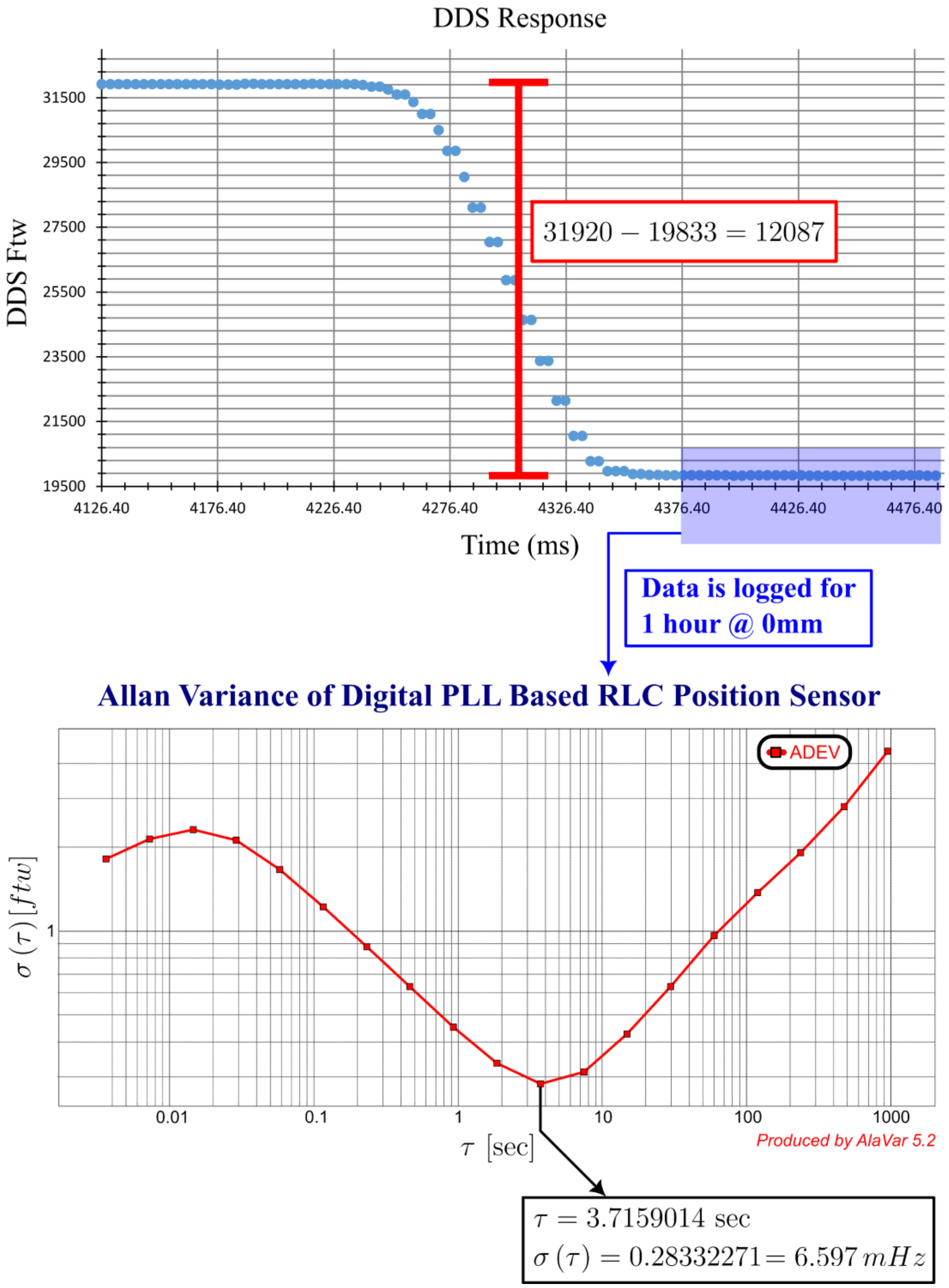


Figure 3.8. DDS Response with the Raw Data from the FPGA

As a result, with the help of digital PLL, resolution as low as $\pm 0.293\,\mu m$ with an integration time of 3.72 seconds is obtained according to the Allan Variance,

$$\text{Resolution} = \frac{0.28332271}{31920-19833} 25mm \approx 0.6\,\mu m \tag{3.2}$$

Oscilloscope readout is given in Figure 3.9. For comparison, the same reading is shown on the same figure for $\phi_{difference} = -90^{\circ}$, where it is proven that in fact the maximum SNR occurs around -40° as the output signal is amplified more.

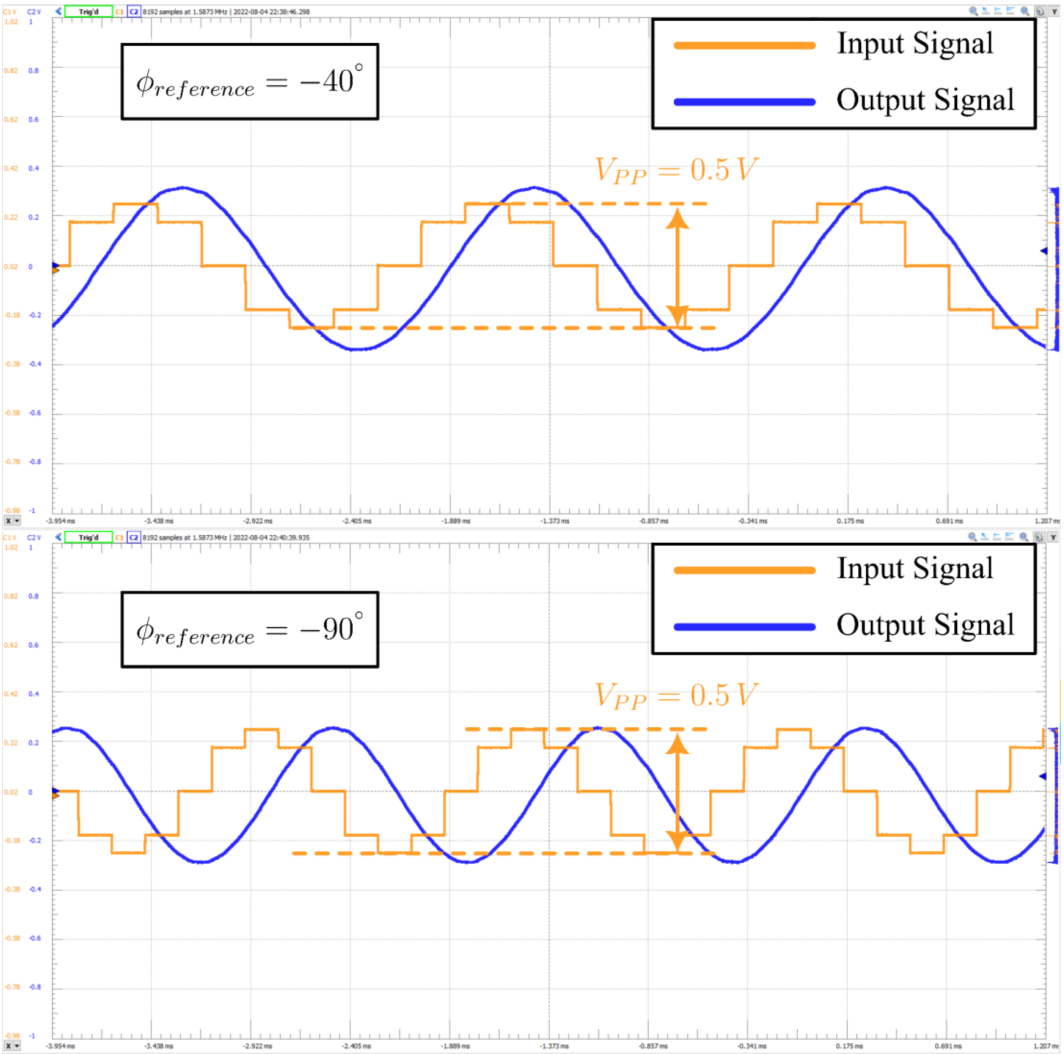


Figure 3.9. Oscilloscope Reading for the Closed Loop RLC System

Finally, the system response for a step input is determined where the reference phase is changed from -90° to -40°. As a result, the DDS response as well as the change in phase angle are measured and plotted in Figure 3.10.

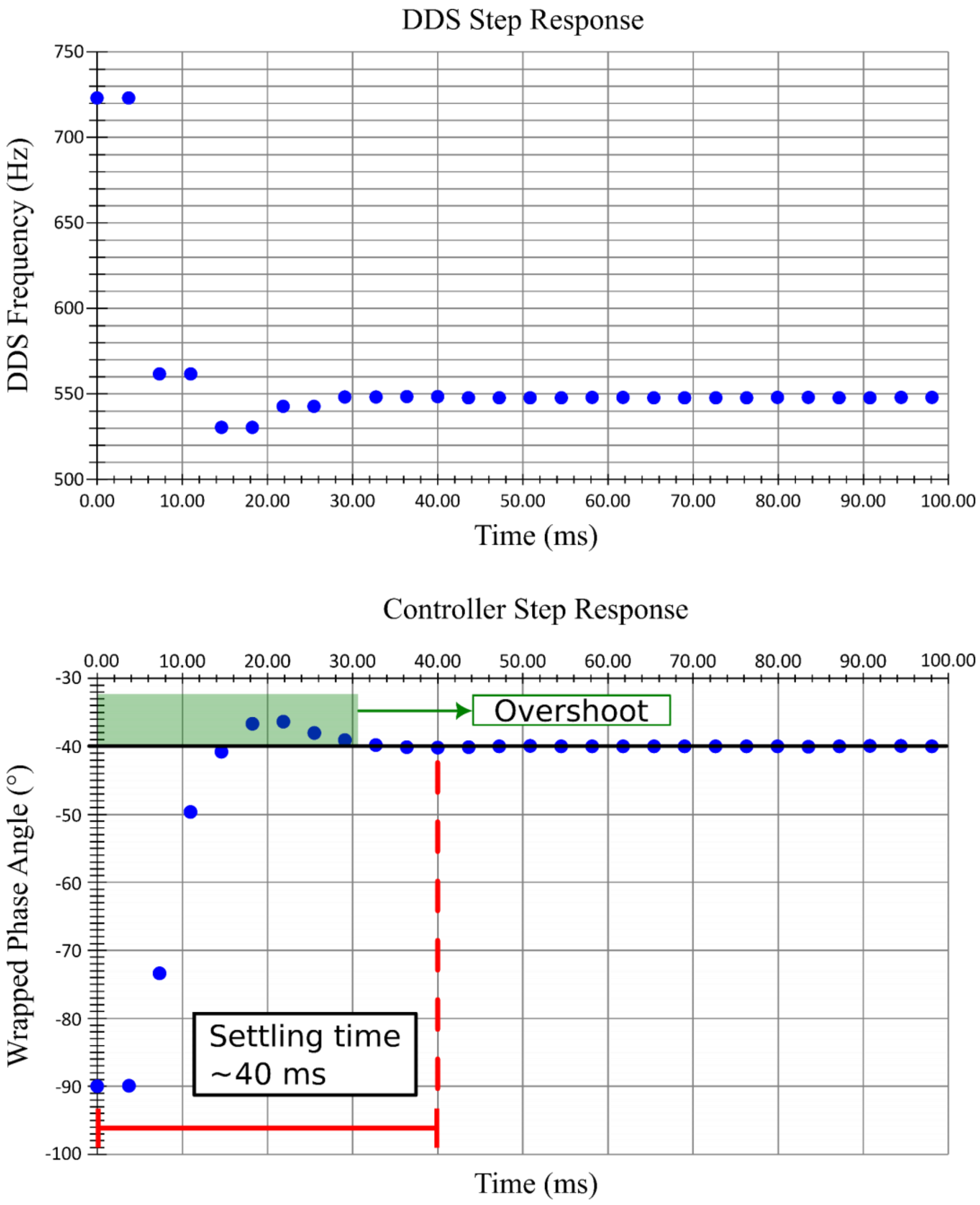


Figure 3.10. Step Response of the Closed Loop RLC System

As it can be seen, the underdamped controller has a settling time of around 40 ms, where the overshoot amount is acceptable. One can try to find the controller gain to achieve a critically damped system. However, this response is enough and shows that the system is stable.

3.2 Characterization of a Resonator and Implementation of DPLL for Multiple DUTs

In this section characterization of a quartz resonator is performed for demonstration purposes; additionally, quartz crystals with distinct resonating frequencies are simultaneously kept at resonance. The circuit and digital hardware design are similar to the previous section with some design changes.

3.2.1 Frequency Response of a Quartz Crystal

A quartz resonator with a resonance frequency around 60 kHz is taken as the DUT. Its frequency response is taken around this frequency. The frequency response is taken using the lock-in amplifier based digital PLL proposed in this thesis. The user interface accepts the starting, stopping frequencies, frequency increment amount and settling cycles after which the received data is accepted as valid as frequency changes. Those inputs are sent through the UART protocol to the FPGA, starting the program inside. The frequency, magnitude and phase data are transmitted to a PC through UART protocol. The PC saves the received stream of data from the FPGA. This saved data is then taken to Microsoft’s Excel Software for drawing the Bode Diagram of this quartz resonator. The circuit diagram is given in Figure 3.11.

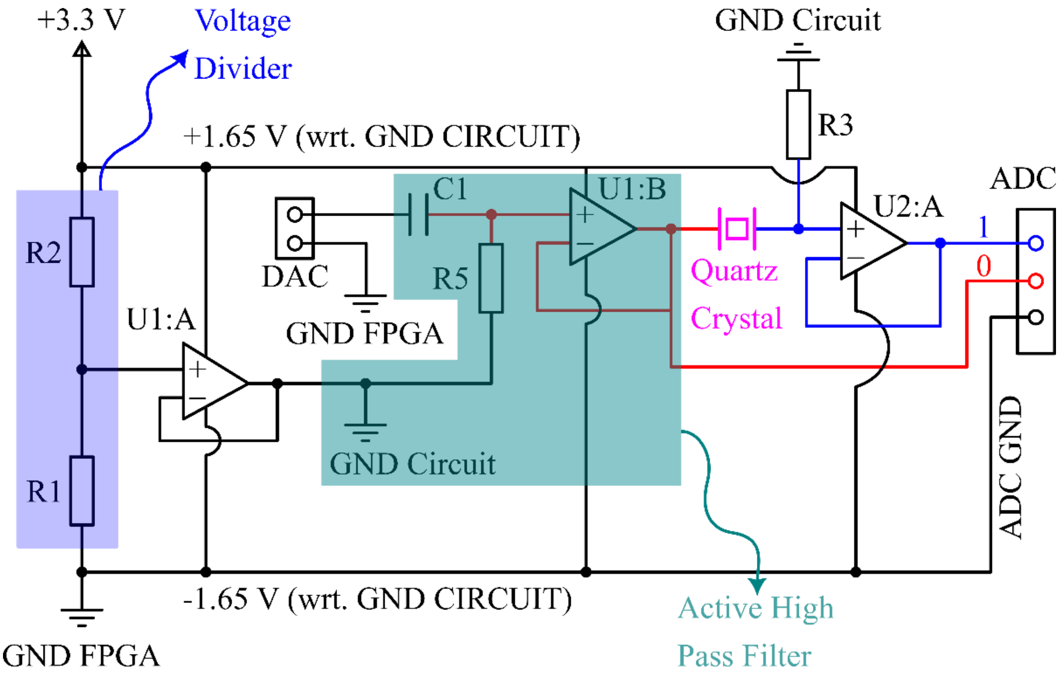


Figure 3.11. Circuit Diagram with Quartz Crystal as DUT

The magnitude and phase plots of the crystal is provided in Figure 3.12. The vertical axis is given in ftw where its frequency conversion equation is given on the figure as well. Other than that, it is clear that the maximum SNR point inside the blue region, corresponding to the maximum of the magnitude diagram has a phase value different than -90° which is actually around -50° mainly due to feedthrough capacitance. As a result, the commonly known resonance indication where the phase difference is $\phi = -90^{\circ}$ does not correspond to the maximum magnitude in the diagram as is shown in red area on the figure. Hence, the necessity of locking onto a desired phase value given by the user is obvious while operating the system at its maximum SNR value. The digital design proposed in this thesis makes this possible. Resonance and anti-resonance peaks are clearly seen in both the magnitude and phase diagrams. The vertical axis is given in ftw where its frequency conversion equation is given on the figure as well. Other than that, it is clear that the maximum SNR point inside the blue region, corresponding to the maximum of the magnitude diagram has a phase value different than -90° which is actually around -50°. Similarly, the commonly known resonance indication where the phase difference is $\phi = -90^{\circ}$ does not correspond to the maximum magnitude in the diagram as is shown in red area on the figure. As a result, the necessity of locking onto a desired phase value given by the user is obvious while operating the system at its maximum SNR value. The digital design proposed in this thesis makes this possible.

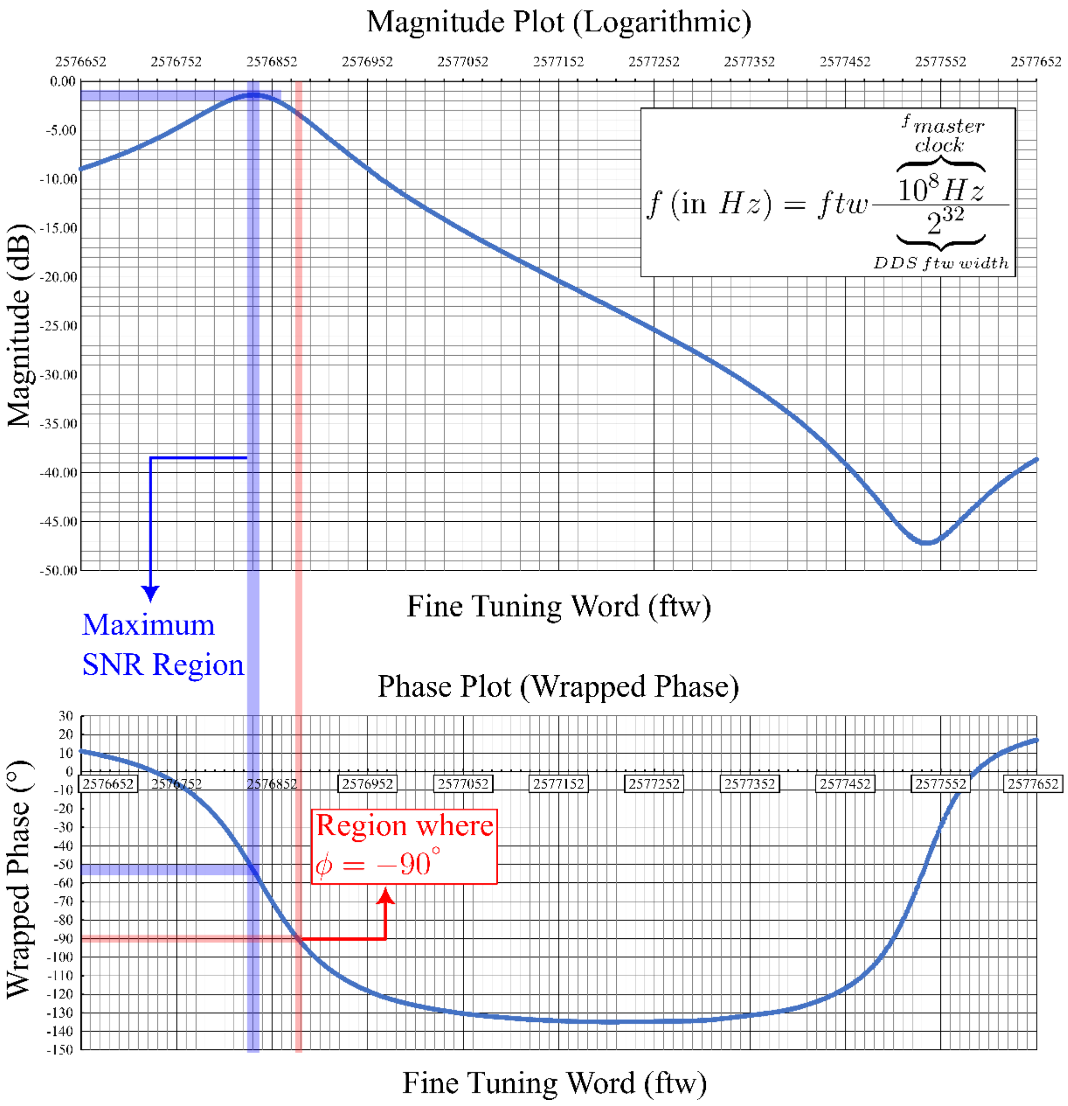


Figure 3.12. Bode Plot of the Quartz Crystal

The region around the maximum SNR is inspected closely, and the quality factor is determined based on the obtained data. As a result, quality factor of the resonator is found to be around 19,000 which is expected from quartz crystals. Note that although quite small for quartz crystals, feedthrough capacitance does exist, hence the difference between maximum SNR and $\phi = -90^\circ$ region.

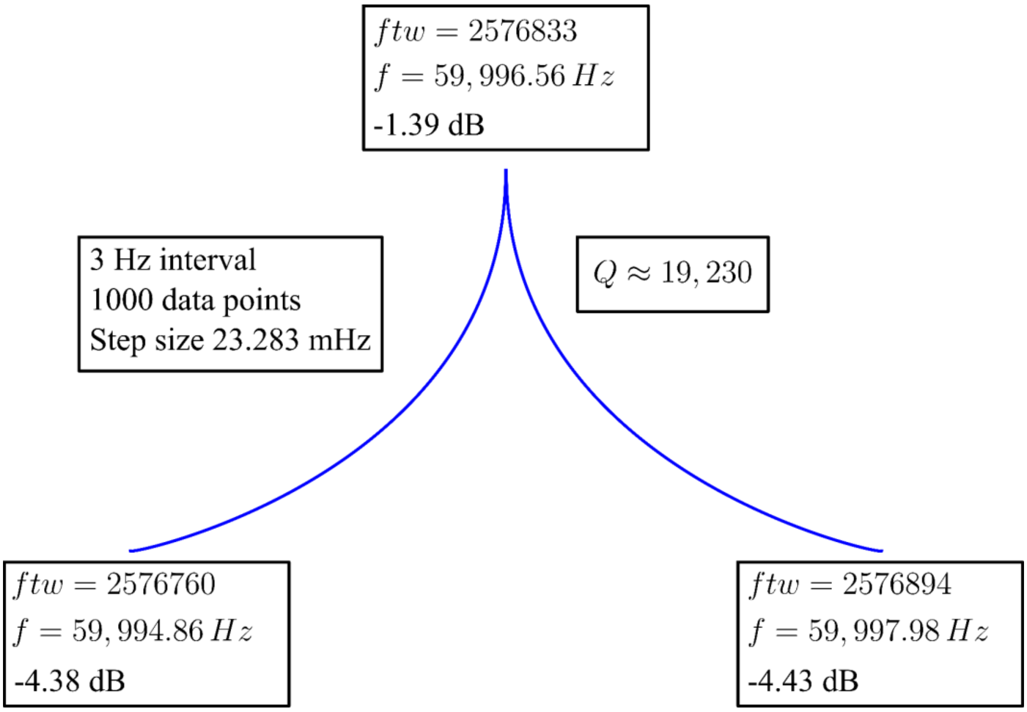


Figure 3.13. Quality factor of a Quartz crystal

3.2.2 Resonating Three Quartz Crystals with Distinct Frequencies

In this section, lock-in based DPLL is configured for multi device operation and for proving that the system in fact works as expected, three quartz resonators with 60.000 kHz, 65.536 kHz and 77.500 kHz are resonated and their outputs are logged in. Allan variance of the resonators is plotted for sensitivity analysis. Fast Fourier Transform (FFT) results obtained from the oscilloscope readings are shown to prove that in fact the signal has three distinct components necessary for resonating three DUTs. Feed through capacitance of these crystals are in pF level, hence feed through cancellation is not necessary for such small capacitances. A photograph of the system setup along with the resonator array is given in Figure 3.14, where the resonance frequencies decrease from left to right. As the analog readout circuitry is the same for both RLC position sensor and quartz resonators, the same breadboard is used for the tests. Cmod A7-35T [41] is used for RLC position sensor application. However, for multiple DUTs, a more capable Zedboard’s FPGA XC7Z020CLG484-1 [42] is needed as the resources available are not sufficient for the specific design in this thesis. For both of the applications in the test chapter, analog readout circuitry is powered by the respective FPGAs used for that application. Analog Discovery 2 is used as an oscilloscope to measure both the input and output side of the DUT(s). FPGAs are supplied with an external clock, which is from an oven controlled clock oscillator (OCXO). Keysight’s 33600A Waveform Generator is used for this purpose, where clock jitter less than 0.5 ps is guaranteed. Using an OCXO as a reference clock makes more precise temperature measurements possible as the master clock is guaranteed to be not affected by the change in environment temperature. Otherwise, environment temperature would have affected both the master clock and the DUTs, which is not desired. Overall system setup view is provided in Figure 3.15, where multiple DUTs are resonated with the Zedboard and OCXO as reference clock. Analog Discovery 2’s user interface shows both the time and frequency domain contents of the measured signals, proving in fact the resonance condition is achieved.

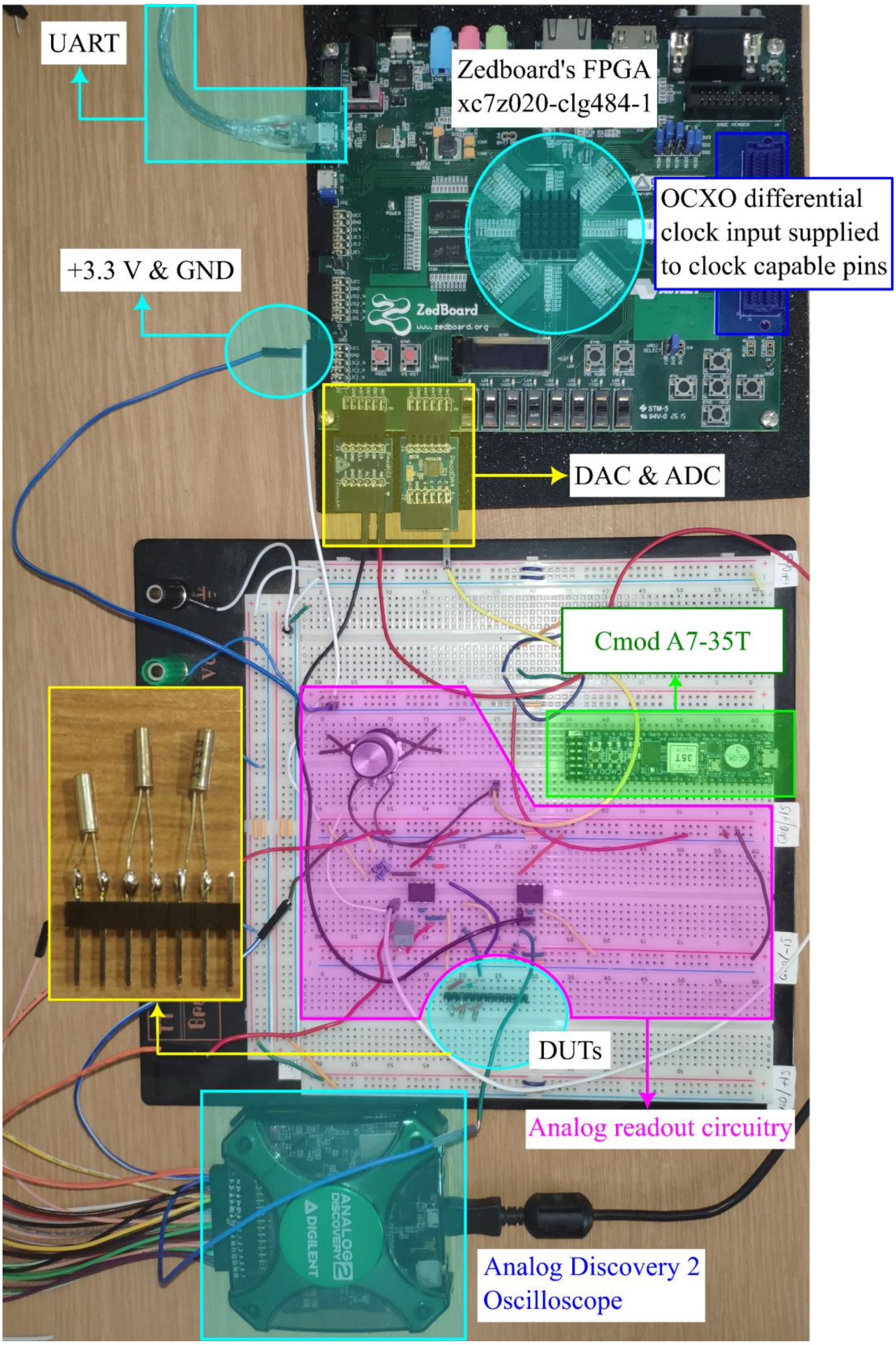


Figure 3.14. Top view of the sytem setup

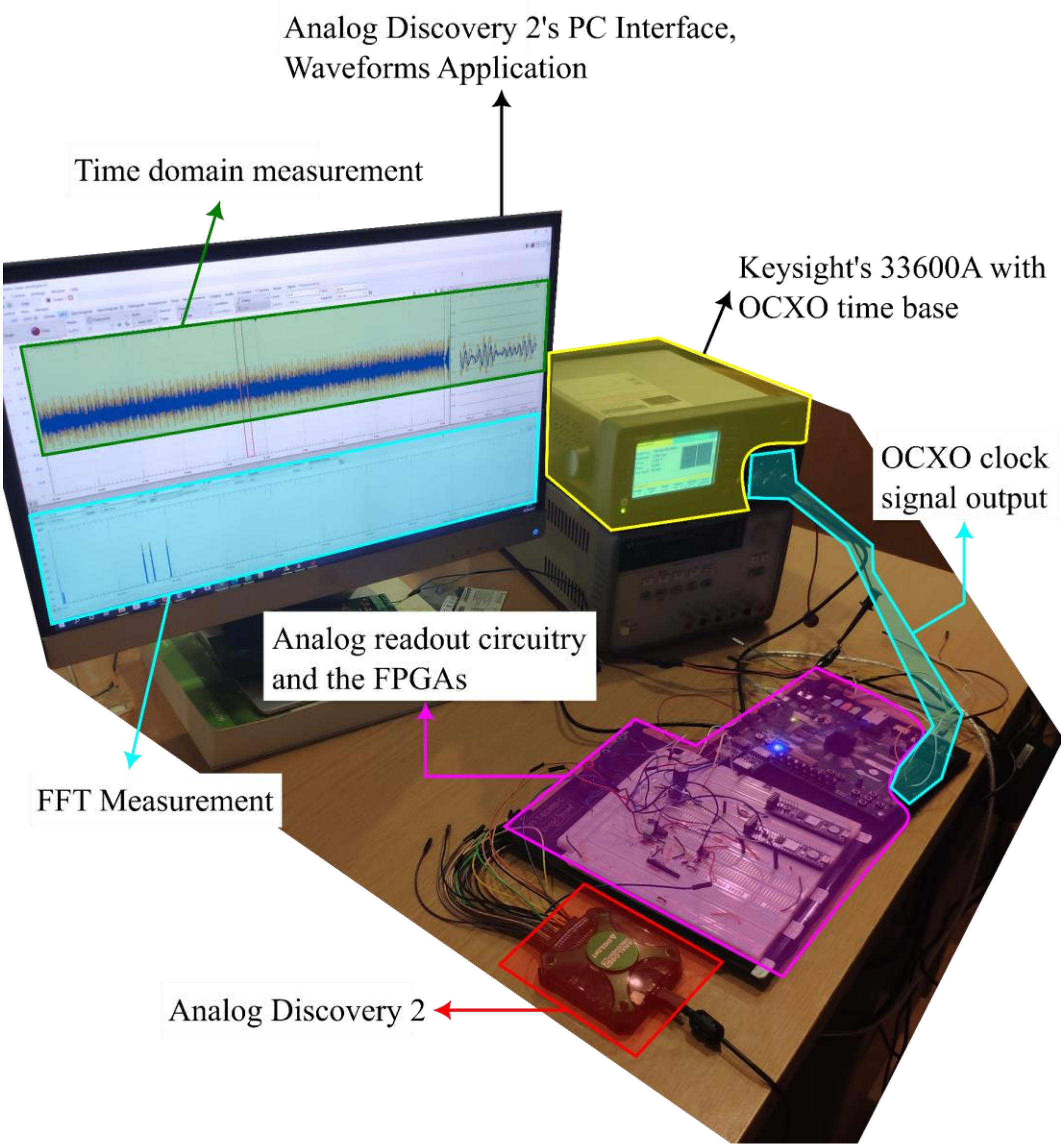
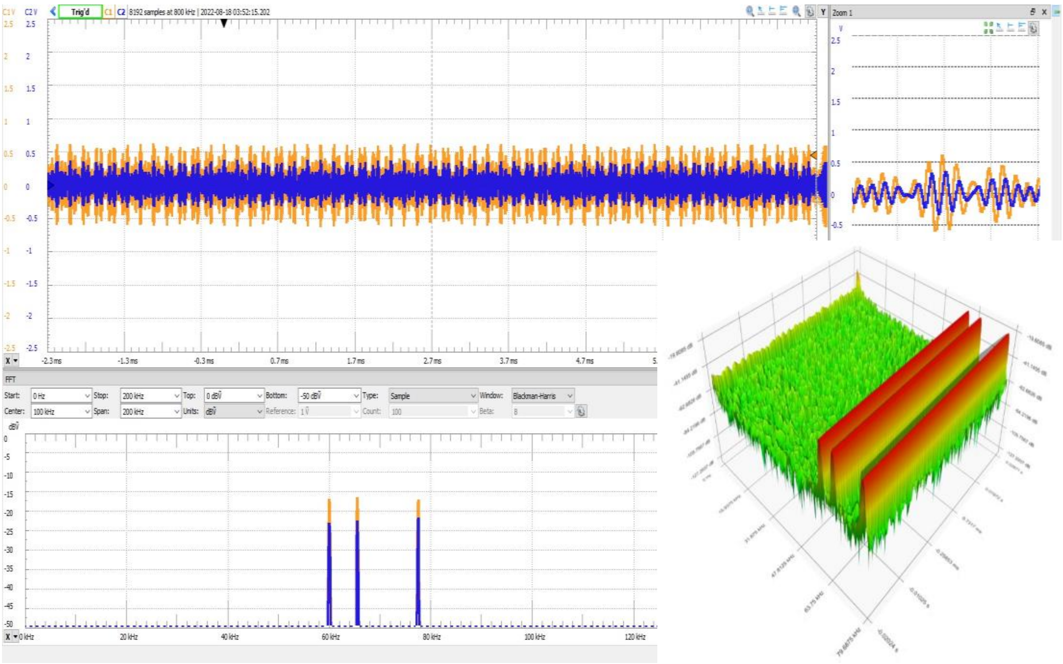


Figure 3.15. Overall system setup

Analog Discovery 2’s instrument panel is given in the following figure, where the FFT result on the bottom left shows frequency components of both the yellow input, and blue output signals of the quartz crystals. As expected, the time domain signal on the top side contains three sinusoidal terms, corresponding to the resonance frequencies of the DUTs. The rainbow chart is provided in the same figure for better clarity of the signal content.



Trig'd

8192 samples at 800 kHz | 2022-08-18 03:52:15.202

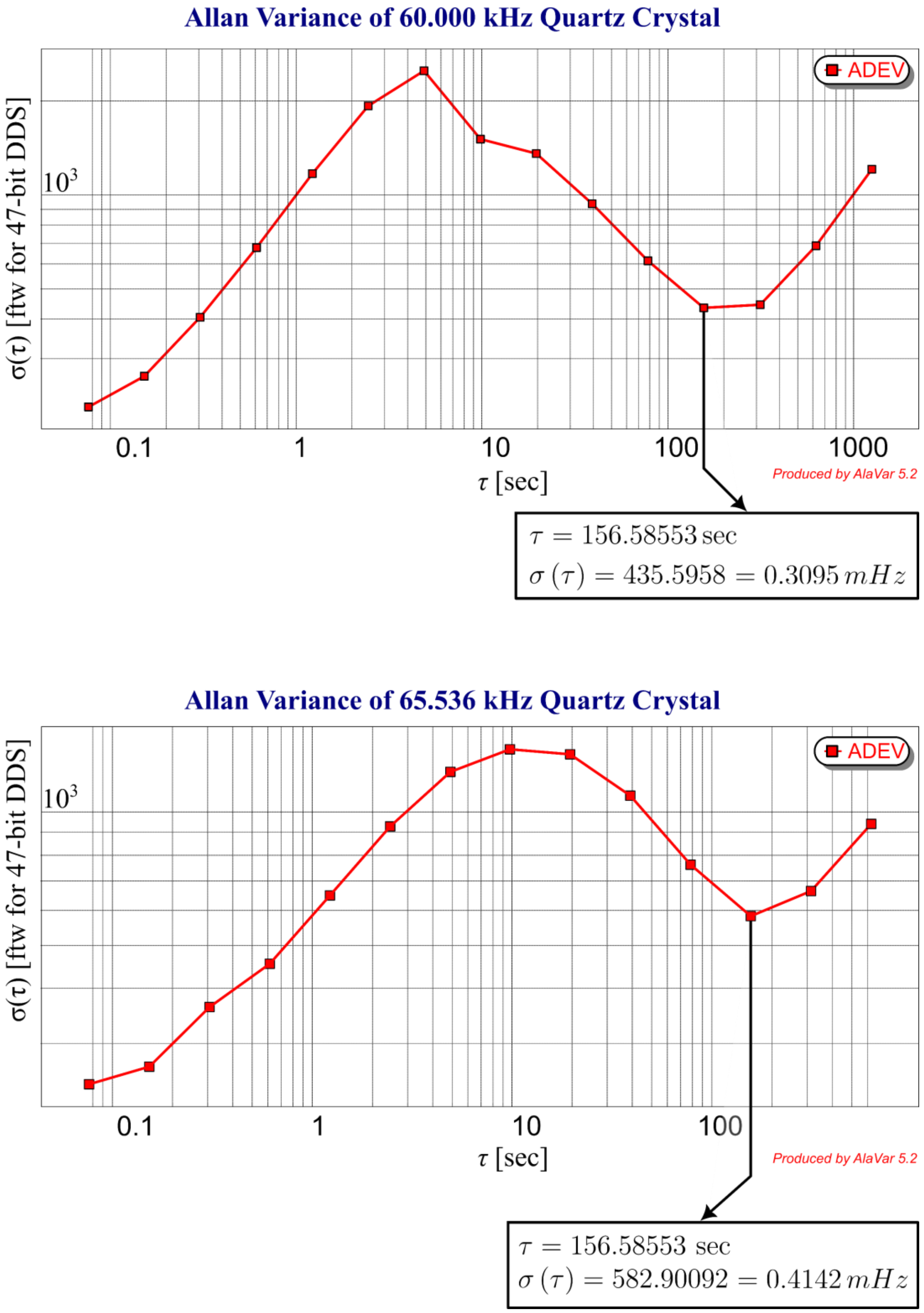
Zoom 1

FFT

Start: 0 Hz
Stop: 200 kHz
Top: 0 dBV̂
Bottom: -50 dBV̂
Type: Sample
Window: Blackman-Harris
Center: 100 kHz
Span: 200 kHz
Units: dBV̂
Reference: 1 V̂
Count: 100
Beta: 8

Figure 3.16. Oscilloscope reading

Allan Variance of the resonators is measured through the logged data coming from the FPGA. The test is performed in atmospheric conditions; hence temperature control of the environment has not been possible. As a result, effect of the temperature is imminent and causes change in the Allan curve. However, the data set is chosen around the region where the temperature does not fluctuate for an hour.



Allan Variance of 60.000 kHz Quartz Crystal

Allan Variance of 65.536 kHz Quartz Crystal

Figure 3.17. Allan Variance of the 60.000 kHz and 65.536 kHz Resonators

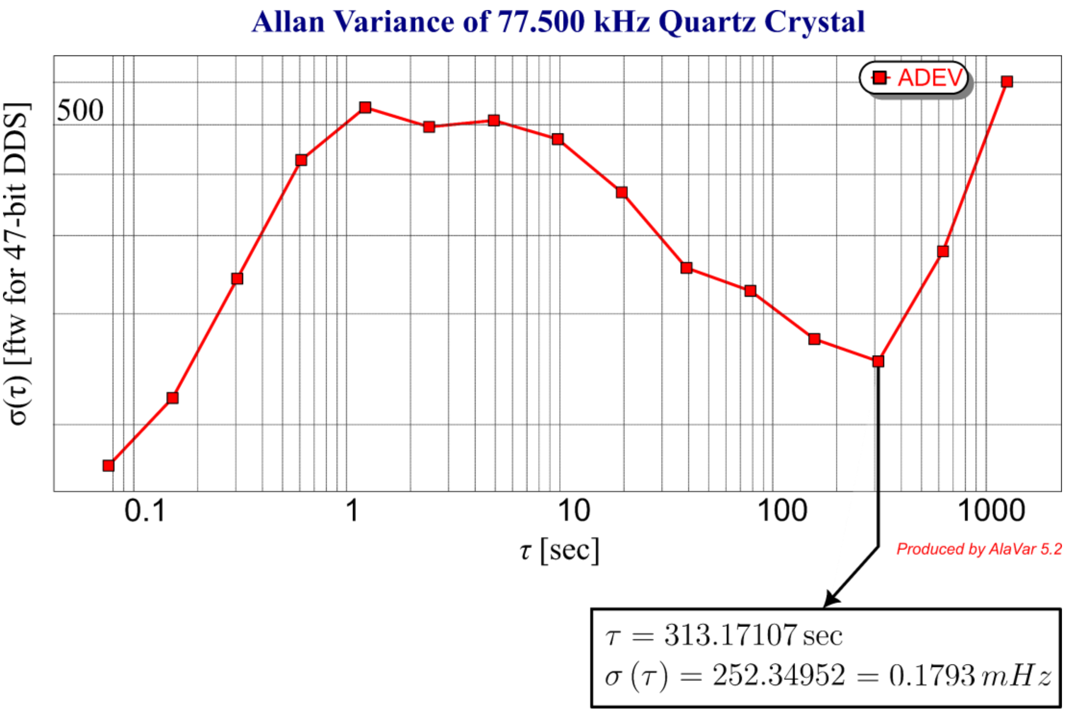


Figure 3.18. Allan Variance of the 77.500 kHz Quartz Resonator

CHAPTER 4

CONCLUSION AND FUTURE WORKS

This study presents a lock-in based DPLL readout mechanism, which is designed, simulated, implemented and tested using an FPGA. The aim is to operate any kind of resonance based transducer with the building blocks presented in this thesis. In the 2nd Chapter, lock-in principles along with the DPLL structure is explained. After proposing the overall design, constituting blocks and special digital signal processing algorithms are explained, where simulations and implemented design in software is presented. Hardware design both for resonating single and multiple DUTs is presented. Afterwards, 3rd Chapter demonstrates the tests performed on simple resonating devices such as RLC circuit for precise position sensing and high-quality factor quartz crystals for resonance detection. The overall conclusions as a result of this thesis are expressed below:

- Fundamental principles necessary for closed loop, resonance maintaining design is covered in detail. Frequency, time and z-domain operations necessary for implementing such a design is proven to be possible with FPGAs.
- Capability of synthesizing a sinusoidal wave with a dynamically adjustable frequency as low as $0.71\mu Hz$ is demonstrated with the help of a 47-bit DDS signal produced inside the FPGA. This capability is proposed to be used for feed through cancellation in resonant MEMS transducers by means of producing same frequency signals with different phase values. Moreover, parallel processing capability of FPGAs is used for generation of signals constituting more than one frequency component inside.
- Realizing serial communication algorithms on low-level design is demonstrated to be useful for transmission of data at the highest rate possible. Z-domain LPF designs are implemented using either RAM or CIC filter. It is

shown that both of them have almost the same characteristics. Comparison between those two filters is done where the former one is found to be useful for single resonator loops while the latter one is found to be more useful for resonating multiple DUTs.

- CORDIC algorithm is designed from the ground up, and test benches are written in simulation environment to prove its performance. The algorithm is used mainly for phase detection between the sense and drive ports. The algorithm is capable of finding the angle with an extremely low true error as much as $7.45 \cdot 10^{-9}$.
- A closed loop integral controller is used to lock onto resonance. The controller input is based on the phase error calculated from the CORDIC block outputs. Controller gain is shown to be dependent on the quality factor of the resonating structure both in the design chapter and tests. Step response of the controller is demonstrated as well.
- A simple low-cost RLC circuit with a variable ferrite core inductor is shown to be capable of measuring displacement with a resolution as low as $0.6\,\mu m$ for an integration time of around 3.71 seconds, according to Allan Variance analysis. It seems to be on par with most of the LVDT transducers used in the industry in terms of resolution, considering how low cost the device is [43].
- The building blocks in the design chapter are configured in a way, such that the hardware design is capable of taking the frequency response of a quartz crystal. Dynamic range of the design is shown to be capable of showing both the resonance and antiresonance characteristics.
- Multiple DUTs are resonated simultaneously with separate lock-in detection and integral controller. As a result, it is proposed that the same readout technique is capable of resonating microbolometer array where the change in resonance is related to temperature change.

The performance of the overall design can be enhanced with the following methods in the future:

- The design should be optimized to accommodate much more DUTs, where the overall system still works with sufficient performance. For example, optimizing the CORDIC algorithm would allow FPGA resources to be utilized more efficiently.
- The tests should be performed in a temperature-controlled environment for more accurate Allan Variance measurements.
- A generic Z-domain transfer function of the overall system can be determined to better select controller gain and design a more sophisticated controller
- A more detailed test-bench can be prepared for simulating the blocks before implementing the design on hardware. In fact, unforeseen errors can rise up which otherwise would have been impossible if the tests are performed solely on hardware.
- An external computer RAM can be used and a RAM controller can be written to use this peripheral for the moving average, resulting in less utilization of FPGA resources.
- More professional circuits (PCBs) can be manufactured and circuit analyses can be performed with proper software simulations for the electronic components to mitigate the effect of electronics related noise and errors.

REFERENCES

[1] S. R. Bistafa, "Euler first theory of resonance," *Arch Hist Exact Sci*, vol. 76, no. 3, pp. 207–221, May 2022, doi: 10.1007/S00407-021-00280-5/FIGURES/9.

[2] R. M. Langdon, "Resonator sensors-a review," *I. Phys. E: Sci. Instrum*, vol. 18, 1985.

[3] K. E. Petersen, "Silicon as a Mechanical Material," *Proceedings of the IEEE*, vol. 70, no. 5, pp. 420–457, 1982, doi: 10.1109/PROC.1982.12331.

[4] W. H. Ko, "Trends and frontiers of MEMS," *Sens Actuators A Phys*, vol. 136, no. 1, pp. 62–67, May 2007, doi: 10.1016/J.SNA.2007.02.001.

[5] C. T. C. Nguyen, "MEMS technology for timing and frequency control," *IEEE Trans Ultrason Ferroelectr Freq Control*, vol. 54, no. 2, pp. 251–270, Feb. 2007, doi: 10.1109/TUFFC.2007.240.

[6] "From Technologies to Markets Status of the MEMS Industry 2021 Market and Technology Report 2021 Sample," 2021, Accessed: Aug. 08, 2022. [Online]. Available: www.yole.fr

[7] O. Brand, I. (Electrical engineer) Dufour, S. M. Heinrich, and F. Josse, "Resonant MEMS : fundamentals, implementation and application".

[8] S. Ilyas and M. I. Younis, "Resonator-based M/NEMS logic devices: Review of recent advances," *Sens Actuators A Phys*, vol. 302, p. 111821, Feb. 2020, doi: 10.1016/J.SNA.2019.111821.

[9] M. A. Hopcroft *et al.*, "Using the temperature dependence of resonator quality factor as a thermometer," *Appl Phys Lett*, vol. 91, no. 1, p. 013505, Jul. 2007, doi: 10.1063/1.2753758.

[10] A. Rahafrooz and S. Pourkamali, “Rotational mode disk resonators for high-Q operation in liquid,” *Proceedings of IEEE Sensors*, pp. 1071–1074, 2010, doi: 10.1109/ICSENS.2010.5690808.

[11] K. Azgın, “LORENTZ KUVVETİ TABANLI, TINLAYAN ve TİTREŞİM GENLİĞİ ÖLÇÜMÜ İLE ÇALIŞAN BİR MEMS MANYETOMETRE,” *Uludağ University Journal of The Faculty of Engineering*, vol. 22, no. 3, pp. 61–80, Dec. 2017, doi: 10.17482/uumfd.325923.

[12] A. Rogalski, *Infrared and Terahertz Detectors, Third Edition*. CRC Press, 2019. doi: 10.1201/b21951.

[13] H. Wada *et al.*, “Design and performance of 256x256 bolometer-type uncooled infrared detector,” in *Infrared Detectors and Focal Plane Arrays V*, Jul. 1998, vol. 3379, no. 22, p. 90. doi: 10.1117/12.317635.

[14] M. Piller, N. Luhmann, M.-H. Chien, and S. Schmid, “Nanoelectromechanical infrared detector,” Sep. 2019, p. 1. doi: 10.1117/12.2528416.

[15] J. R. Vig, R. L. Filler, and Y. Kim, “Uncooled IR imaging array based on quartz microresonators,” *Journal of Microelectromechanical Systems*, vol. 5, no. 2, pp. 131–137, Jun. 1996, doi: 10.1109/84.506201.

[16] N. Inomata, L. Pan, M. Toda, and T. Ono, “Temperature-depended mechanical properties of microfabricated vanadium oxide mechanical resonators for thermal sensing,” *Jpn J Appl Phys*, vol. 55, no. 3, Mar. 2016, doi: 10.7567/JJAP.55.037201.

[17] M. H. Chien, M. Brameshuber, B. K. Rossboth, G. J. Schütz, and S. Schmid, “Single-molecule optical absorption imaging by nanomechanical photothermal sensing,” *Proc Natl Acad Sci U S A*, vol. 115, no. 44, pp. 11150–11155, Oct. 2018, doi: 10.1073/pnas.1804174115.

[18] J. Jeong, S. Kumagai, I. Yamashita, Y. Uraoka, and M. Sasaki, “Micromechanical IR thermal detector using torsional oscillation: Improvement of resonator profile for high sensitivity,” *Jpn J Appl Phys*, vol. 54, no. 4S, p. 04DE07, Apr. 2015, doi: 10.7567/jjap.54.04de07.

[19] Y. Zhang, S. Hosono, N. Nagai, S.-H. Song, and K. Hirakawa, “Fast and sensitive bolometric terahertz detection at room temperature through thermomechanical transduction,” *J Appl Phys*, vol. 125, no. 15, Apr. 2019, doi: 10.1063/1.5045256.

[20] “FPGA Fundamentals - NI.” https://www.ni.com/en-tr/innovations/white-papers/08/fpga-fundamentals.html (accessed Jul. 30, 2022).

[21] Z. Gao, B. Zhou, Y. Li, Q. Wei, and R. Zhang, “Design of Digital Demodulation Circuit for Closed Loop Control of Resonant MEMS Gyroscopes,” *Proceedings of IEEE Sensors*, vol. 2019-October, Oct. 2019, doi: 10.1109/SENSORS43011.2019.8956796.

[22] “Window Types: Hanning, Flattop, Uniform, Tukey, and Exponential.” https://community.sw.siemens.com/s/article/window-types-hanning-flattop-uniform-tukey-and-exponential (accessed Jul. 14, 2022).

[23] A. Breitenbach, “Against spectral leakage,” *Measurement*, vol. 25, no. 2, pp. 135–142, Mar. 1999, doi: 10.1016/S0263-2241(98)00074-8.

[24] R. Abdolvand, B. Bahreyni, J. Lee, and F. Nabki, “Micromachined Resonators: A Review,” *Micromachines (Basel)*, vol. 7, no. 9, p. 160, Sep. 2016, doi: 10.3390/mi7090160.

[25] J. J. Carr, “Radio Receiver Basics,” *The Technician’s EMI Handbook*, pp. 163–195, Jan. 2000, doi: 10.1016/B978-075067233-7/50016-8.

[26] C. Wang, Y. Qu, and Y. P. T. Tang, “IQ quadrature demodulation algorithm used in heterodyne detection,” *Infrared Phys Technol*, vol. 72, pp. 191–194, Sep. 2015, doi: 10.1016/J.INFRARED.2015.08.003.

[27] F. M. Gardner, *Phaselock techniques*. Wiley-Interscience, 2005.

[28] W. C. Michels and N. L. Curtis, “A Pentode Lock-In Amplifier of High Frequency Selectivity,” *Review of Scientific Instruments*, vol. 12, no. 9, p. 444, Dec. 2004, doi: 10.1063/1.1769919.

[29] T. J. Rouphael, “Common Digital Modulation Methods,” *Signal Processing for Software-Defined Radio*, pp. 25–85, Jan. 2009, doi: 10.1016/B978-0-7506-8210-7.00003-5.

[30] E. J. Oughton, W. Lehr, K. Katsaros, I. Selinis, D. Bubley, and J. Kusuma, “Revisiting Wireless Internet Connectivity: 5G vs Wi-Fi 6,” *Telecomm Policy*, vol. 45, no. 5, p. 102127, Jun. 2021, doi: 10.1016/J.TELPOL.2021.102127.

[31] “All About Direct Digital Synthesis | Analog Devices.” https://www.analog.com/en/analog-dialogue/articles/all-about-direct-digital-synthesis.html (accessed Jul. 06, 2022).

[32] R. SCHAFER, J. McCLELLAN, and M. YODER, “DSP first: a multimedia approach,” 1998, Accessed: Jul. 14, 2022. [Online]. Available: http://repo.unikadelasalle.ac.id/index.php?p=show_detail&id=2770&keywords=

[33] S. Smith, S. Linearity, S. Fidelity, and C. Decompositions, *The Scientist and Engineer’s Guide to Digital Signal Processing By Steven W. Smith, Ph. D.* Accessed: Jul. 14, 2022. [Online]. Available: http://www.dspguide.com/ch24/1.htm

[34] “pg058-blk-mem-gen.pdf • Viewer • Documentation Portal.” https://docs.xilinx.com/v/u/en-US/pg058-blk-mem-gen (accessed Jul. 18, 2022).

[35] E. B. Hogenauer, “An Economical Class of Digital Filters for Decimation and Interpolation,” *IEEE Trans Acoust*, vol. 29, no. 2, pp. 155–162, 1981, doi: 10.1109/TASSP.1981.1163535.

[36] E. B. Hogenauer, “An Economical Class of Digital Filters for Decimation and Interpolation,” *IEEE Trans Acoust*, vol. 29, no. 2, pp. 155–162, 1981, doi: 10.1109/TASSP.1981.1163535.

[37] “A Beginner’s Guide To Cascaded Integrator-Comb (CIC) Filters - Rick Lyons.” https://www.dsprelated.com/showarticle/1337.php (accessed Aug. 13, 2022).

[38] J. E. Voldert, “The CORDIC Trigonometric Computing Technique*,” 1959. doi: 10.1109/TEC.1959.5222693.

[39] “Linear Variable Differential Transformer (LVDT) Basics | TE Connectivity.” https://www.te.com/usa-en/products/sensors/position-sensors/intersection/lvdt-tutorial.html (accessed Aug. 15, 2022).

[40] “Serial oscilloscope – x-io Technologies.” https://x-io.co.uk/serial-oscilloscope/ (accessed Sep. 13, 2022).

[41] “Cmod A7 - Digilent Reference.” https://digilent.com/reference/programmable-logic/cmod-a7/start?redirect=1 (accessed Sep. 13, 2022).

[42] “ZedBoard - Digilent Reference.” https://digilent.com/reference/programmable-logic/zedboard/start (accessed Sep. 13, 2022).

[43] P. Ripka, “Magnetic Sensors: Principles and Applications,” *Encyclopedia of Materials: Science and Technology*, pp. 1–11, Jan. 2007, doi: 10.1016/B978-008043152-9.02158-8.

APPENDICES

A. LUT Generator for DDS

```
clc, clear all, close all
fid='C:/Users/Graviton/Desktop/THESIS_SnrYlmz_2097665/dds_16_12_lut.txt';
pw=16; %phase width
aw=12; %amplitude width
max_amplitude = 2^(aw-1)-1;
%quarter sine-wave lookup table
fileID=fopen(fid,'w');
for i=1:1:2^(pw-2)

    output(i)=round(max_amplitude*sin((i-1)*pi()/(2^(pw-1))));
    fprintf(fileID,'to_signed(%d,AMPL_WIDTH),\n',output(i));
    %fprintf('to_signed(%d,AMPL_WIDTH),\n',output(i));

end

fclose('all');
```

Output File Content (Only a portion of the file is given due to sheer size of the lookup values):

```
to_signed(0,AMPL_WIDTH),
to_signed(0,AMPL_WIDTH),
to_signed(0,AMPL_WIDTH),
to_signed(1,AMPL_WIDTH),
to_signed(1,AMPL_WIDTH),
to_signed(1,AMPL_WIDTH),
to_signed(1,AMPL_WIDTH),
to_signed(1,AMPL_WIDTH),
to_signed(2,AMPL_WIDTH),
to_signed(2,AMPL_WIDTH),
to_signed(2,AMPL_WIDTH),
to_signed(2,AMPL_WIDTH),
to_signed(2,AMPL_WIDTH),
to_signed(3,AMPL_WIDTH),
to_signed(3,AMPL_WIDTH),
to_signed(3,AMPL_WIDTH),
to_signed(3,AMPL_WIDTH),
to_signed(3,AMPL_WIDTH),
to_signed(4,AMPL_WIDTH),
.
.
.
to_signed(2047,AMPL_WIDTH),
to_signed(2047,AMPL_WIDTH),
to_signed(2047,AMPL_WIDTH),
to_signed(2047,AMPL_WIDTH),
```

B. Frequency Response of the Moving Average Low Pass Filter

```
clc, clear all, close all
N=8;

r=0.001:pi()/100000:pi(); %cycle/sample if r/(2*pi) i.e. normalized freq.
z=exp(1).^(j.*r);
Mag8=abs((1-z.^-N)./(1-z.^-1));
Mag16=abs((1-z.^-16)./(1-z.^-1));
Mag64=abs((1-z.^-64)./(1-z.^-1));
Mag1024=abs((1-z.^-1024)./(1-z.^-1));

plot(r/(2*pi()),[20*log10(Mag8/N);20*log10(Mag16/16);20*log10(Mag64/64);20*
log10(Mag1024/1024)],'LineWidth',3);
xline(1/8,'--r',{'Sampled Signal','Located @
$\displaystyle\frac{1}{8}$'},'interpreter','latex','LineWidth',3);
xlabel('Normalized Frequency (cycles/sample)')
ylabel('Magnitude (in dB)')
ylim([-70 0])
title('Moving Average LPF Frequency Response vs Number of Filter Taps (N) for
f_s_a_m_p_l_i_n_g=8*f_s_i_g_n_a_l')
legend('N=8','N=16','N=64','N=1024','f_s_i_g_n_a_l')
```

C. LUT for CORDIC Algorithm

```
clc, clear all, close all
fid='C:/Users/Graviton/Desktop/THESIS_SnrYlmz_2097665/dds_cordic_lut.txt';
pw=31; %the width of the output phase,32 is the maximum number that one can
select since to_signed accept integers btw
%-2.147.483.648 and 2.147.483.647, hence
%log2(2147483647*360/315)=31.19 meaning 31 is the maximum pw one can
%choose without exceeding this limit
nstages=28;  %number of cordic stages(the amount of total rotations)
fileID=fopen(fid,'w');
for i=1:1:nstages

    output(i)=round(atand(1/2^i)*2^pw/360);
    fprintf(fileID,'to_signed(%d,PW),\n',output(i))

end

fclose('all');
```

Output File Content :

```
to_signed(158466703,PW),
to_signed(83729454,PW),
to_signed(42502378,PW),
to_signed(21333666,PW),
to_signed(10677233,PW),
to_signed(5339919,PW),
to_signed(2670123,PW),
to_signed(1335082,PW),
to_signed(667543,PW),
to_signed(333772,PW),
to_signed(166886,PW),
to_signed(83443,PW),
to_signed(41722,PW),
to_signed(20861,PW),
to_signed(10430,PW),
to_signed(5215,PW),
to_signed(2608,PW),
to_signed(1304,PW),
to_signed(652,PW),
to_signed(326,PW),
to_signed(163,PW),
to_signed(81,PW),
to_signed(41,PW),
to_signed(20,PW),
to_signed(10,PW),
to_signed(5,PW),
to_signed(3,PW),
to_signed(1,PW)
```