# DESIGN AND IMPLEMENTATION OF HIGH FREQUENCY SIGNAL GENERATOR BASED ON PHASE LOCKED LOOP

**MUTINDA BONIFACE MUSYOKA (B.Ed. Sci)**

**I56/20697/2020**

**A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF DEGREE OF MASTER OF SCIENCE (ELECTRONICS AND INSTRUMENTATION) IN THE SCHOOL OF PURE AND APPLIED SCIENCES OF KENYATTA UNIVERSITY**

**NOVEMBER 2023**

## DECLARATION

This thesis is my original work and has not been presented for degree or other awards in any other university.

Signature.....Date.....

Boniface Mutinda

Department of Physics

Kenyatta University

## SUPERVISORS

We confirm that the work reported in this thesis was carried out by the candidate under our supervision.

Signature.....Date.....

Dr. Mathew Munji

Department of Physics

Kenyatta University

Signature.....Date.....

Dr. Raphael Nyenge

Department of Physics

Kenyatta University

## DEDICATION

To my parents, Isaac Mutinda and Mary Mueni, and siblings, Judith Kithia, Elizabeth Mwikali, Emmaculate Nzisa and Antony Muema.

## ACKNOWLEDGMENTS

I would like to take this opportunity to thank my supervisors, Dr. Mathew Munji and Dr. Raphael Nyenge, for all of the assistance and support they provided me while I was conducting my research. You have been a great source of knowledge for me; without you, I would never have been able to get to this point.I would also like to take this opportunity to thank the staff and academic members of the Department of Physics at Kenyatta University for the assistance they provided and the words of encouragement they offered.

I want to appreciate the Kenyatta University Electrical Engineering Department chairman and staff for allowing me to carry out measurements using a spectrum analyser in the Electronics lab.

I'd like to take this opportunity to thank my comrades Marcelina Kinyumu, Paul Angira, Daniel Ndengwa, and Mary Ndimuwaki for their humble time and the positive atmosphere of sharing knowledge and helping one another. It's been a pleasure working with you guys.

I would like to take this opportunity to convey my appreciation to my entire family, and in particular to my parents, for their unflinching support and encouragement throughout this entire process of my research. You have modeled the man I am today, and I cannot thank you enough. I also want to thank all my friends who have always been there for me in times of hardship.

In conclusion, I would like to express my gratitude to the Almighty God for the unwavering support and guidance that He has provided throughout my life.

## TABLE OF CONTENTS

## LIST OF FIGURES

## LIST OF TABLES

## LIST OF ABBREVIATIONS AND ACRONYMS

**AC** Alternating Current

**ADC** Analog to Digital Converter

**BJT** Bipolar Junction Transistor

**CE** Chip Enable

**CLK** Clock

**CMOS** Complementary Metal-Oxide-Semiconductor

**CP** Charge Pump

**CPU** Central Processing Unit

**CRO** Cathod Ray Oscilloscope

**CSR** Cycle Slip Reduction

**DAC** Digital Analogue Converter

**DC** Direct Current

**DDS** Direct Digital Synthesizer

**FPGA** Field Programmable Gate Array

$w_{OSC}$ Frequency of Oscillation

**GPS** Global Positioning System

**HCMOS** High Density Complimentary Metal-Oxide-Semiconductor

**IC** Integrated Circuit

**ICSP** In Circuit Serial Programming

$I_{CP}$ Charge Pump Current

**IF** Intermediate Frequency

**ISM** Industrial Scientific and Medical

**LC** Inductor Capacitor

**LCD** Liquid Crystal Display

**LE** Load Enable

**LFMS** Linear Frequency Modulated Signal

**LPF** Low Pass Filter

**LSB** Less Significant Bit

**MIPS** Million Instructions Per Second

**MOSFET** Metal-Oxide-Semiconductor Field-Effect Transistor

**MSB** Most Significant Bit

**MUXOUT** Multiplexer Output

**NMOS** N channel Metal-Oxide-Semiconductor

**PC** Personal Computer

**PFD** Phase Frequency Detector

**PLL** Phase Locked Loop

**PMOS** P channel Metal-Oxide-Semiconductor

**ppm** parts per million

**RADAR** Radio Detecting and Ranging

**RBW** Resolution Bandwidth

**RC** Resistor Capacitor

**RFB** Feedback Resistors

**RFC** Radio Frequency Choke

**RISC** Reduced Instruction Set Computer

**SFCW** Step Frequency Continuous Waves

$T_{SW}$ Switching Speed

**TTL** Transistor-Transistor Logic

**USB** Universal Serial Bus

**VBW** Video Bandwidth

**VCO** Voltage Controlled Oscillator

**VCXO** Voltage Controlled Crystal Oscillator

**VFB** Feedback Voltage

**VREF** Reference Voltage

**Wi-Fi** Wireless Fidelity

**WIMAX** Wireless Inter-operability for Microwave Access

## ABSTRACT

A signal generator is an electronic test instrument that has a wide range of applications. Some of these applications include testing systems in cellular communications, radar, microstrip antennas and testing components in electronics labs. For a signal generator to be used in these applications, it should generate high frequencies, have a low level of phase noise, and also have a fast locking time. The purpose of this research was to design and build a signal generator that operates on the principle of a phase-locked loop, has the capacity to generate frequencies ranging from 35 MHz to 3 GHz, and has a low degree of phase noise. The simulation of the phase-locked loop synthesizer was done using the ADIsimPLL design tool. The phase locked loop chip used in simulation was the ADF4351 from Analog Devices. In the design, we implemented a loop filter of the third order and chose a reference frequency of 10 MHz. The layout of the phase locked loop was simulated, and the results showed that the optimal values for loop bandwidth and phase margin were 10 kHz and 45°, respectively. Following the simulation, the optimal values for each of the loop filter's components were analyzed and calculated. The signal generator was built by integrating the phase locked loop synthesizer and a keypad shield with an Arduino UNO microcontroller. The ADF4351 was programmed via Serial Peripheral Interface (SPI) to enable the changing of frequencies using the keypad shield. The nature of the generator's signal was investigated using a cathode ray oscilloscope in the 35-100 MHz frequency range. The testing was also done for 101-3000 MHz using a spectrum analyser. The level of phase noise was calculated at 35 megahertz, 387 megahertz, 1 gigahertz, 2 gigahertz, and 2.9 gigahertz at 1, 10, 100, and 1000 kilohertz. The amount of phase noise that was acquired after experimental work was higher than the level obtained after simulation. For example, at the output frequency of 387 MHz, the experimental phase noise was -104.2 dBc/Hz while the simulated was -126 dBc/Hz at 100 kHz offset frequency. The reason for this is that the phase noise contribution in the simulation was only from the phase locked loop components while in the experimental, in addition to the phase noise from the PLL components, there were other sources of phase noise while carrying out the experiment. The rise in output frequency was also accompanied by an increase in phase noise. The reason for this is that the signal generator was built with the concept of a phase locked loop which implements the idea of frequency multiplication by dividing along the feedback loop with the use of a counter. This concept raised the phase noise by 20 multiplied by the logarithm of the number of counter. The maximum spur appeared at the third harmonic and was found to be -18.6 dBc, while the minimum spur appeared at the fourth harmonic and was found to be -44.5 dBc.

# CHAPTER ONE

# INTRODUCTION

This chapter gives background information on the research subject. It also provides the statement of the research problem, its rationale and research objectives.

## 1.1 Background of the Study

A signal generator is an instrument that produces waveforms that are either repeating or non-repeating and can be used in testing electronics devices (Qi *et al.*, 2015). In a laboratory for electronics and in experimental courses, this device can be used for a broad variety of tasks, such as characterizing digital and analog systems (Muteithia, 2014). Numerous methods can be used for generating waveforms in signal generators. These methods include the resistor-capacitor (RC) oscillator, direct digital synthesizers (DDS), inductor-capacitor (LC) oscillator, crystal oscillator, Phase Locked Loop (PLL) synthesizers, as well as various variants of these technologies (Vankka, 2000).

For this study, a frequency synthesizer based on PLL was used to generate frequencies. This is due to the fact that PLL generates frequencies with a wide output range, a low level of phase noise, high accuracy and a fast locking period when designed well (HOSSEINI and Masoumi, 2017).

A frequency synthesizer is an electrical circuit that can generate a broad range of frequencies from a single frequency source (Fox, 2002). When a frequency synthesizer uses PLL technology to generate its output , it is called an indirect frequency synthesiser. A PLL is a control circuit with feedback that is frequency or phase-sensitive and synchronizes the output signal of the Voltage Controlled Oscillator (VCO) with a reference signal (Shurender *et al.*, 2013). PLL synthesizers are commonly used in communication systems in wireless transmission because they have excellent frequency resolution and their ability to lock quickly in time(Patel and Sharma, 2010). Figure 1.1 shows the main components of a PLL, which include a phase or frequency detector

(PFD), filter, Charge Pump (CP), feedback divider (1/N) and VCO (del Rocío Ricardez-Trejo *et al.*, 2017).

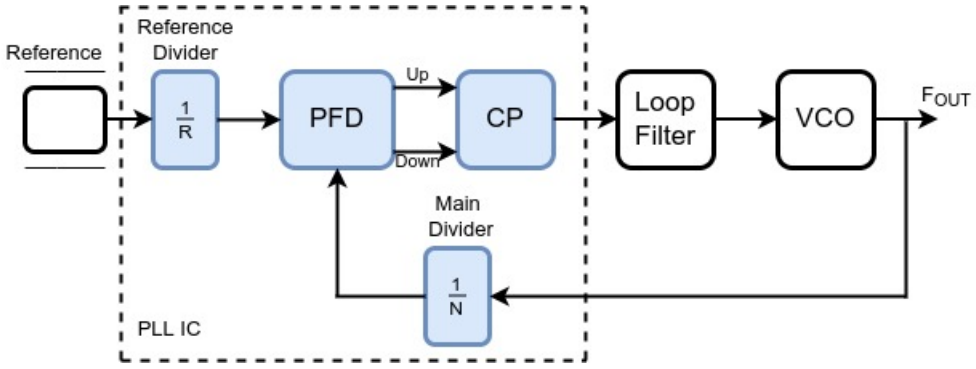


**Figure 1.1:** Schematic representation of the primary elements comprising a frequency synthesizer

A comparison is made by the phase or frequency detector, between the frequency from the reference signal and the frequency that is generated by the VCO. It creates two output signals, "UP" and "DOWN," depending on the deviation of the phase. The CP combines the signals generated by the PFD and provides an output. This output is then sent on to the low pass filter so that spurs generated by the PFD can be reduced. The CP also changes the signal from a phase or frequency state to a voltage state, which is used to vary the VCO. When the phase of the two signals has a difference of zero, the system is said to be locked.

A signal is sent out by the VCO, the nature of which is determined by the voltage that the CP produces. When the signal at the output of the PFD is outrageous, the voltage output of the CP component rises, which in turn causes an increase in the output frequency of the VCO. Similarly, the output of the VCO drops if there is a low level of PFD output. After that, the frequency that the VCO generated is sent to the PFD to recalculate the fluctuation in phase, which ultimately results in the creation of a locked-loop frequency control system (Barret, 1999). For the phase locked loop to generate high frequencies and of wide range, the concept of frequency multiplication is carried out by using an N counter to divide along the feedback loop. Equation 1.1 can be used to calculate the output frequency of the system.

$$F_{OUT} = N \times F_{REF} \tag{1.1}$$

where, $F_{OUT}$ is the frequency at which the VCO produces its output, N is a feedback division ratio, and $F_{REF}$ is the frequency at which the reference produces its output

It is possible to adjust the divider so that the division ratio, N, takes the form of either a whole number or a fraction (Mutinda *et al.*, 2023). This can result in a PLL with either an integer value of N or a fractional value of N (Harney, 2009). An integer N-PLL synthesizer's resolution relies on a reference frequency to function properly. As a result, the reference frequency is divided down by R to obtain a high-frequency resolution, which results in the frequency resolution shown in equation 1.2.

$$F_R = \frac{F_{REF}}{R} \tag{1.2}$$

where, $F_R$ represents the frequency resolution, $F_{REF}$ stands for the frequency of reference, and R stands for the division ratio for reference.

Following this, the frequency at which the signal is produced can be calculated using equation 1.3.

$$F_{OUT} = \frac{N}{R} F_{REF} \tag{1.3}$$

A significant amount of phase noise is typically present in the signals produced by an integer N-PLL. This is due to the fact that, in accordance with the frequency multiplication principle, there is an increase of $20log(N)$ dB in the phase noise of the signal (Barret, 1999). The resolution in a fractional N-PLL which is a fractional part of the resulting frequency produced by the phase detector is attained. This is made possible by an inner circuitry that allows the magnitude of N to vary incrementally when the PLL is in lock condition. When the divider is switched between two consecutive values in the correct division, an ordinary division ratio is realized as N and an arbitrary function y/z.

The frequency of the output is calculated using equation 1.4.

$$F_{OUT} = F_{REF}\left(N + \frac{Y}{Z}\right) \tag{1.4}$$

where, N, Y and Z are integers, N is the feedback partition ratio, Y is the value of fraction register and Z is the value of modulus register (Barret, 1999).

## 1.2 Statement of the Problem

As time goes on, the number of applications that use microwave signals, such as mobile communication, satellite communication, radar, microstrip antennas and Wi-Fi, continues to grow. Therefore, there is a need to design and implement a high frequency signal generator with a low level of phase noise ($<$100 dBc/Hz), a fast locking time ($\leq$1 ms) and the ability to generate frequencies up to the microwave range.

The goal of this work was to create a PLL-based signal generator that can generate frequencies in the 35 MHz–3 GHz range to be used in testing devices operating in this frequency range.

## 1.3 Justification of the Study

In this research, the signal generator was designed using the concept of PLL. This is because phase-locked loop (PLL) synthesizers generate stable and clean signals. They also produce frequencies that have a wide output range compared to frequencies generated using the DDS concept. PLL also has a fast locking time and provides a good platform for the implementation of a signal generator in terms of performance and power consumption.

## 1.4 Objectives

### 1.4.1 General Objective

The primary aim of this research was to design and implement a phase locked loop based signal generator which can generate frequencies in the range of 35 MHz–3 GHz.

### 1.4.2 Specific Objectives

i. To simulate and design a PLL synthesizer, reference frequency, low pass filter, and VCO using the ADIsimPLL design software.

ii. To build a high frequency signal generator following the results obtained after simulation.

iii. To test the functioning of the signal generator.

iv. To carry out measurements and analysis of phase noise and spurs.

## 1.5 Significance of the Study

This study aimed at designing a signal generator that could output frequencies in the microwave range. This signal generator can be applied in many areas, such as testing components in electronics laboratories, characterization of microstrip antennas and testing systems in a wide variety of applications, which include Wireless Fidelity (Wi-Fi)and Wireless Inter-operability for Microwave Access (WiMAX). Signal generators can help in assessing the performance of satellite ground station receivers. It can also be used in broadcast transmitters and receivers and in testing system in Global Positioning System (GPS).

# CHAPTER TWO

# LITERATURE REVIEW

## 2.1 Introduction

This chapter provides an overview of the earlier studies that were conducted on direct digital synthesis and phase locked loop based signal generators. Various simulation tools that have been used in designing phase lock loop system are also discussed in this chapter.

## 2.2 Direct Digital Synthesis based Signal Generators

Li *et al.* (2012) proposed an l-band synthesizer system with high agility and large bandwidth using a direct digital synthesis (DDS) and a scheme of frequency multiplication. Their proposed synthesizer first produces a baseband linear frequency modulated signal (LFMS) using DDS, and then the signal produced is added to a frequency multiplication system that can transform the baseband signal into l-band LFMS with a range of 1424 MHz-1950 MHz, which is quite low.

Bonfanti *et al.* (2003) presented a PLL based on DDS, that had an output frequency of 2.4 GHz with an 80-MHz tuning range. Their design achieved a high-frequency resolution of 1 kHz, a fast switching time of 3 μs and spectral purity of -50 dBc. The main weakness of this device was the high power consumption of the DDS-DAC blocks.

Du *et al.* (2017) presented a field programmable gate array (FPGA) and DDS based signal generator. The FPGA was used to provide the settling period order for the DDS. They obtained a 1.4 gigahertz frequency with a good resolution of approximately 190 picohertz. The level of phase noise calculate at a 1.2 gigahertz was found to be -65 dBc/Hz at a 1 Hz offset frequency, and a settling time of 150 ns was also obtained.

## 2.3 Phase Locked Loop Generators

Hwang *et al.* (2000) proposed a digitally controlled PLL to be applied in clock synthesis. Their design achieved a fast clock time and a frequency range of 300 MHz-800 MHz, but due to the resolution limitation of the DAC and the low sensitivity of the frequency detector, its jitter was very high, 149 ps p-p. Yang *et al.* (2012) designed an ultra-broadband frequency synthesizer that had a high resolution and low power consumption. In their design, DDS was used as the reference for the broadband PLL. Using their approach, they were able to obtain broadband ranging from 137.5 MHz to 4.4 GHz with 128 μHz resolution. However, this approach is costly because it requires many components. Shurender *et al.* (2013) designed a PLL-based indirect frequency synthesis for the s band frequency. They minimized spurious signals by narrowing the bandwidth of the filter. However, this resulted in a longer lock time.

Handique and Bezboruah (2015) designed and analysed a PLL based synthesizer operating in the range of 128 to 256 megahertz which can be applied in communication systems that are wireless. They recorded phase noise of –80.9 dBc/Hz and –107.8 dBc/Hz at an offset frequency of 10 and 1000 kilohertz, respectively. Fajar *et al.* (2019) designed a fractional based signal generator for detecting a high-resolution object or subsoil structure. They used an ADF4351 synthesizer, which was controlled by a microcontroller. Their signal generator generated output signals in the range of 0.5 GHz – 1.5 GHz.

Kameche and Feham (2013) designed and simulated an integer-N frequency synthesizer for exchanging data over short distances, operating in 2400 MHz–2480 MHz range with a 1 MHz frequency step. This design had a switching time of about 50.83 $\mu$s. Despite this, the design resulted in a phase noise of -82 dBc/Hz when the offset was adjusted to 1 MHz, which is an extremely high value. Rout *et al.* (2014) performed an analysis and developed a 1 GHz PLL in order to achieve rapid acquisition of phase and frequency.

They discovered that the PLL settling time was largely determined by the type of PFD design that was utilized, in addition to the charge pump and the loop filter. They were able to accomplish a lock time of 280.6 ns at a frequency of 1 GHz by making the appropriate choices for these parameters. Kumar and Shrimali (2022) constructed a subcutaneous medical device that uses a frequency synthesizer based on a phase-locked loop of the second order for monitoring and performing medical diagnoses in a human body. The weakness of their design was that it had only one output frequency of 450 MHz.

## 2.4 Simulation of PLL

Sun (2013) did research on designing and debugging a phase locked loop synthesizer. He recommended that designers of PLL based synthesizer use the ADIsimPLL tool software for simulation. Shurender *et al.* (2013) designed an S-band frequency synthesizer for microwave applications using fractional-N PLL. They used ADIsimPLL software to simulate and design PLL. They also simulated the total phase noise contributed by PFD, filter and the VCO.

del Rocío Ricardez-Trejo *et al.* (2017) implemented a Matlab -based PLL simulator for modeling and simulating blocks of PLL based microwave generators. Telba *et al.* (2004) described Matlab Simulink simulation method for noise and jitter in PLL. They studied two methods of reducing jitter, either by using a PLL that has a small band filter or with a narrow voltage controlled crystal oscillator (VCXO) by simulation using Matlab Simulink.

HOSSEINI and Masoumi (2017) designed and implemented an ultra-high band and high precision synthesizer.They simulated the system with the ADIsimpPLL software so that they could calculate the level of phase noise and the settling period. They also applied the simulation to realizing and avoiding mistakes that occur during designing.

According to the previous research, many challenges result when designing a PLL to operate at high frequencies. The goal of this study was to identify the lowest value of phase noise that could be achieved. By calculating the ideal value of the loop bandwidth for the filter that was going to be implemented in the system, the phase noise was brought down significantly. After running simulations of the PLL synthesizer with a number of different loop bandwidths, the optimal loop bandwidth that produced the lowest level of phase noise was chosen.

# CHAPTER THREE

# THEORETICAL BACKGROUND

## 3.1 Introduction

This section offers an overview of the theoretical underpinnings of the primary components of the signal generator that is built on a PLL. Under the context of a phase lock loop system, this chapter further delves into phase noise and the degree of spurs.

## 3.2 Components of a Phase Locked Loop

### 3.2.1 Reference Crystal Oscillator

A reference crystal oscillator should have high frequency stability. For this reason, quartz crystals are used since they can overcome some factors that affect an oscillator's stability, such as variations in temperature, load and DC power supply. Piezoelectric effect is the ability of a crystal to generate an electrical charge in reaction to applied mechanical force, and it is reversible. This effect is utilized by Quartz Crystal Oscillators, and it is what allows them to produce oscillations. For the crystal to function properly in an electronic circuit, it is given the form of a capacitor by being sandwiched in the middle of two metal plates. Figure 3.1 illustrates the symbol of a crystal oscillator.

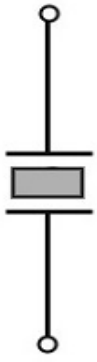

**Figure 3.1:** A symbol that resembles a crystal

The most commonly used crystal oscillator circuit is the transistor pierce crystal oscillator, which is shown in Figure 3.2. The connections of resistors $R_1$, $R_2$, and $R_E$ work

together to form a voltage-divider which outputs a stable signal. The capacitor $C_E$ is what enables the AC bypass for the emitter resistor. The Radio Frequency Choke (RFC) coil is what generates the DC bias and also decouples AC signal that may be present on the power lines from having an effect on the output signal. The coupling capacitor C eliminates any DC current that might flow between the collector and the base.

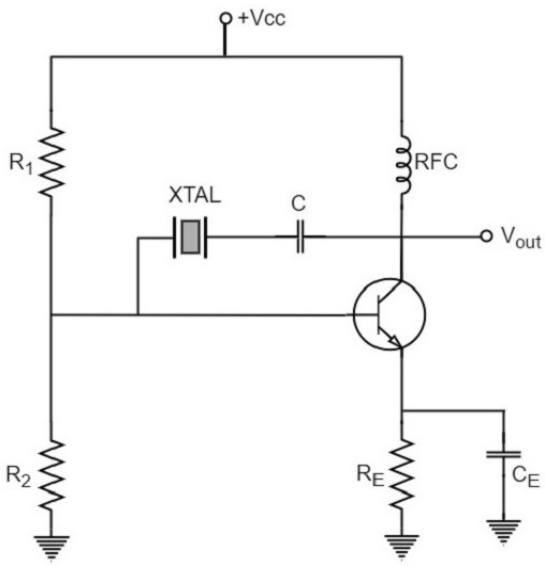


**Figure 3.2:** Transistor pierce crystal oscillator circuit

The term ”stability” refers to the oscillator’s capacity to keep the frequency of its output signal unchanged over a considerable period of time. The stability is usually expressed in parts per million (ppm) which can be represented in the form of frequency as shown in equation 3.1 (Lacoste, 2014).

$$Variation\ in\ Hz = \frac{f \times ppm}{10^6} \tag{3.1}$$

where, f is the central frequency, which is measured in hertz (Hz), and ppm is the frequency variation, which is measured in parts per million.

### 3.2.2 Phase / Frequency Detector (PFD)

Memory components such as flip-flops are utilized in the construction of the tri-state PFD, which is the commonly used phase detector. An edge-triggered PFD that is

made up of two D flip-flops and a NAND gate in the reset path is shown in Figure 3.3 (Debnath, 2015). The undetected phase difference range, sometimes known as dead zones, is decreased when the reset path contains a delay element.

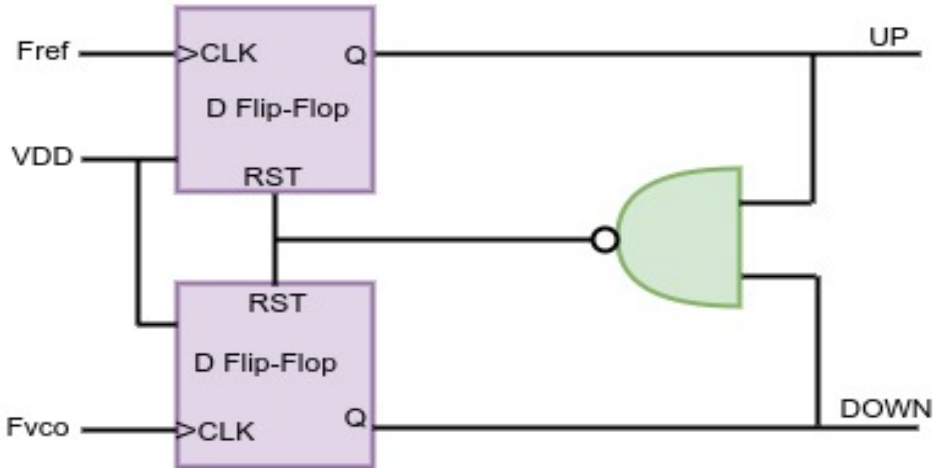


**Figure 3.3:** A PFD Constructed using Edge-Triggered D Flip-Flops

When the reference signal is ahead of the phase of the VCO output, the PFD will provide an “Up” error signal. In a similar fashion, a “Down” error signal is produced whenever the VCO output is ahead of the reference signal.

### 3.2.3 Charge Pump

To produce the current signal Icp, which is equivalent to the phase error signal generated by the PFD, the charge pump is utilized in conjunction with the PFD. Figure 3.4 depicts a diagrammatic representation of a fundamental charge pump. A charge pump consists of two switches, and the "Up" and "Down" pulses on the PFD are the ones that control those switches. When the "Up" signal is high and the “Down” signal is low, switch SW1 is active, switch SW2 is disabled, and a current called IUp flows through the load capacitance. This charges the capacitance to the supply voltage. On the other hand, when the “Down” signal is high, current IDown is sunk, which ultimately results in the load capacitor being discharged. If “Up” and “Down” signals are both high, the current is sourced and sinked equally, and if low, the switches are turned off, and in both cases, there is no current in the load (Behzad, 2012).

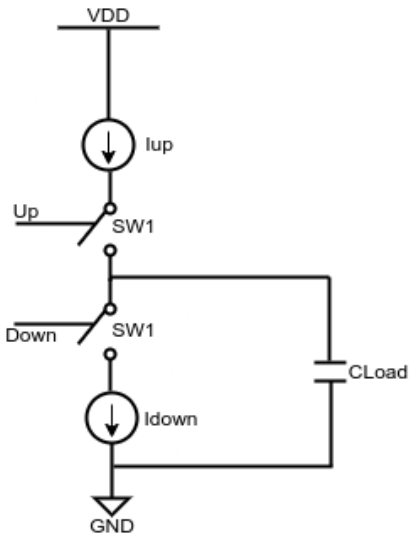


**Figure 3.4:** Charge pump Architecture

### 3.2.4 Loop Filter

It is the job of the filter to remove the phase inaccuracy and the disruption introduced by the higher frequency terms from the output of the PFD. There is flexibility in the design of the filter, which can either be active or passive, as shown in Figure 3.5. Active filters were mostly used in old fashioned PLLs in order to improve the lock in ranges and help match the voltage swings of the PFD with the VCO. Because there are no lock-in range issues in modern PLLs that are integrated with charge pumps, passive filters are the best option. Because of how the new PLL designs are constructed, active filters are no longer required. This is due to the fact that active devices raise both the cost and the complexity of the system(Banerjee, 2017).

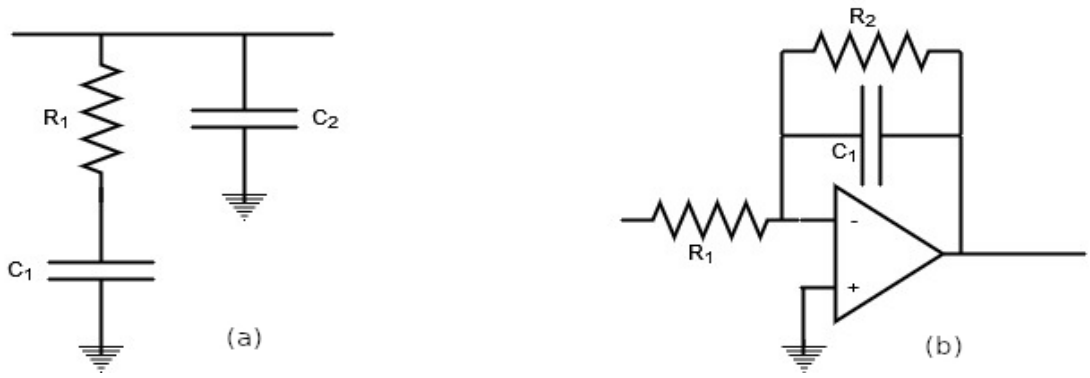


**Figure 3.5:** (a) Passive loop filter (b) Active loop filter

When designing a loop filter, it is essential to select the appropriate filter architecture, phase margin, filter order, and loop bandwidth. We get to know the order of the loop filter by determining the number of poles. Raising the order of the filter will improve the attenuation response of the filter, which will help in the filtering of spurs. Adding more components raises the order, but each of those components has their own thermal noise, which adds to the overall phase noise. The order should be carefully selected because using higher orders may reduce the phase margin at unity gain, which causes instability.

Instability is checked by analyzing the bode plot of the system. Figure 3.6 shows a sample bode plot. At unity gain frequency, the difference between the phase of the loop gain and -180 degrees is referred to as the phase margin. When the phase margin is increased too much, the lock time is lengthened; nevertheless, the instability increases when the phase margin is decreased too much. Phase margin is generally chosen between 40° and 55° (Banerjee, 2017).

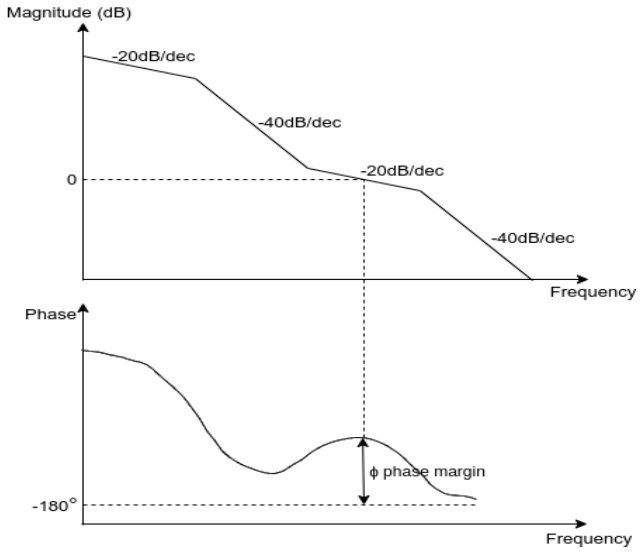


**Figure 3.6:** Stability analysis by Bode plot

The open loop transfer function of a second-order filter has one zero and two poles, which means that the filter is always stable. When applied to very far apart frequencies, a pole and a zero will produce phase shifts of -90 degrees and 90 degrees, respectively.

This indicates that the phase margin can never be 0 if they are positioned in the appropriate locations, which makes the loop stable. It is possible that the phase shift will be more than -180 degrees at unity if the order is raised, as this will result in the addition of new poles. This results in the beginning of oscillations that are not desired. Therefore, it is important to carefully place the poles and zeros of higher order filters to avoid causing instability.

Another crucial element that contributes to the overall performance of the PLL is the loop bandwidth, which is the point at which the output signal is reduced to -3 dB relative to the input signal. The relationship between bandwidth and the switching speed is given by equation 3.2.

$$T_{SW} \propto \frac{1}{\omega_n \xi} \tag{3.2}$$

where ωn is the loop bandwidth, $T_{SW}$ is the switching speed, and ξ is the damping factor.

As a result, a reduction in bandwidth causes the system to lock up for an extended period of time and also results in an increase in the attenuation of spurs.

### 3.2.5 Voltage Controlled Oscillator

It generates a stable output frequency from a reference that operates at a significantly lower frequency. The most common forms of oscillators found on integrated circuits are ring oscillators and LC-tuned oscillators.

LC- tuned oscillators have a satisfying level of phase noise at gigahertz frequencies (Yoon, 2004). As can be seen in Figure 3.7, they are made up of inductors and capacitors that are connected in parallel to a current source in order to produce an LC tank circuit, which is also referred to as an LC resonator circuit. If it is designed appropriately, the

tank should be able to produce the desired output oscillation. Equation 3.3 is what decides how much oscillation there is in the output.

$$w_{osc} = \frac{1}{\sqrt{LC}} \tag{3.3}$$

where, $w_{OSC}$ is the frequency of oscillation, LC is the Inductance-Capacitance

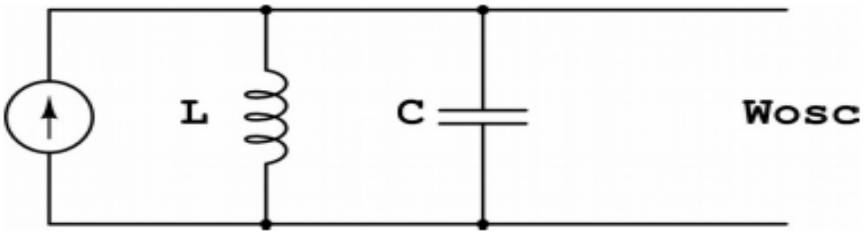


**Figure 3.7:** LC tank circuit

A ring oscillator is known to be a good power consumer, and they also have large tuning ranges. However, they have poor phase noise in the gigahertz frequency range in comparison to the LC-tuned oscillators. As can be seen in Figure 3.8, they are usually constructed using an odd number of delay stages.

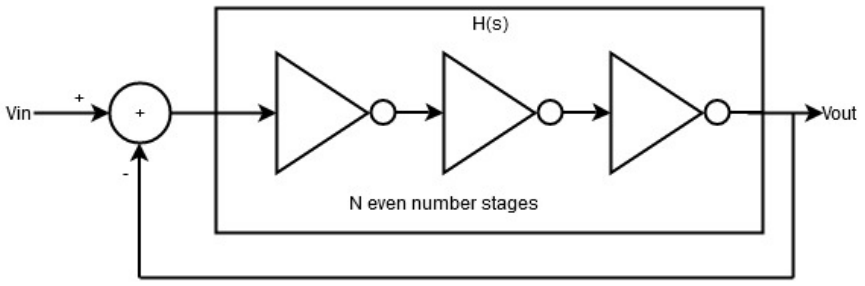


**Figure 3.8:** Ring oscillator

The gain of the voltage controlled oscillator (KVCO) is the extent to which the output frequency of the VCO changes for a change in input voltage, and the units are expressed in MHz/V. A fundamental plot of control voltage vs frequency of oscillation is shown in Figure 3.9, and this plot is used to determine the VCO gain. The VCO gain, or sensitivity, is determined by equation 3.4.

$$K_{VCO} = \frac{\omega_2 - \omega_1}{V_2 - V_1} \tag{3.4}$$

where, $\omega_2$ and $\omega_1$ is the final and initial voltage respectively, and $V_2$ and $V_1$ is the final and initial voltage respectively

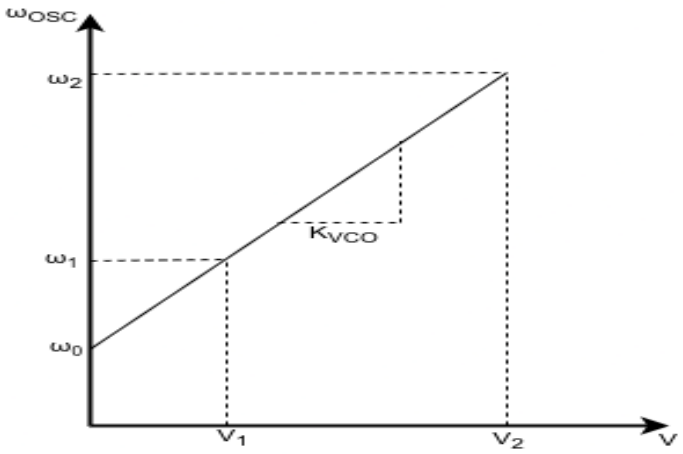


**Figure 3.9:** Control voltage versus output frequency

### 3.2.6 Programmable Frequency Divider

A frequency divider can be defined as any system that is capable of dividing an output frequency by a certain number. Digital counters, which are manufactured using low frequency processes such as CMOS, are utilized as divisions for VCO frequencies that are lower than 200 MHz. Because a pure CMOS counter cannot function at higher frequencies, prescalers, which are fixed higher frequency dividers, are utilized to bring the frequency of the VCO down to a range that is more amenable to use by the counters (Bölücek, 2009).

Figure 3.10 depicts a single modulus prescaler, which includes both a fixed value prescaler and a counter. The value of the prescaler is set to P, and the counter has been programmed to read “y”. Equation 3.5 provides the division rate, which can be altered by making adjustments to the value of y in the counter. One of the limitations of this divider is that the division ratio can only be determined using integer multiples of P

(Bölücek, 2009).

$$N = y \times p \tag{3.5}$$

where N is the total divider value, y is the Y counter value and p is the prescaler value.

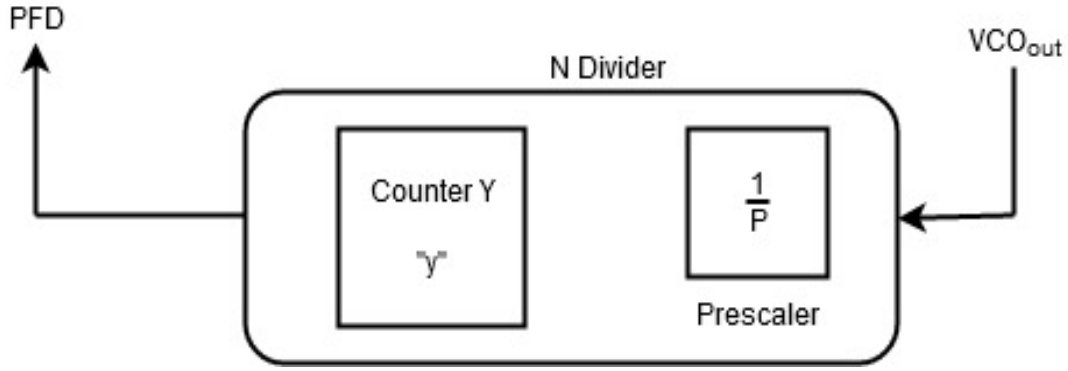


**Figure 3.10:** Single modulus prescaler

Dual modulus prescalers on the other hand are designed to increase the resolution of the division. The design utilizes just one prescaler with the size P and a value of P+1. This value is achieved by inserting a pulse swallow function directly in front of the prescaler, as demonstrated in Figure 3.11. The A counter and the B counter begin counting at the same precise moment. The pulse swallow function is disengaged once the P+1 prescaler has completed its "a" cycles of operation and the A counter has reached zero. The B counter, which has a prescaler value of P, keeps counting for b-a cycles. The division ratio is calculated as shown in equation 3.6. A limitation that states b value must be higher than or equal to "a" value must be satisfied for the operation to be carried out correctly. In the event that this requirement is not fulfilled and the incorrect N value is reached, the counter will prematurely reset before the A value hits zero (Bölücek, 2009).

$$N = a(P+1) + P(b-a) = Pb + a \tag{3.6}$$

where, N is the total divider value, a is number of cycles for A counter, b is the

number of cycles for B counter, P is the prescaler value.

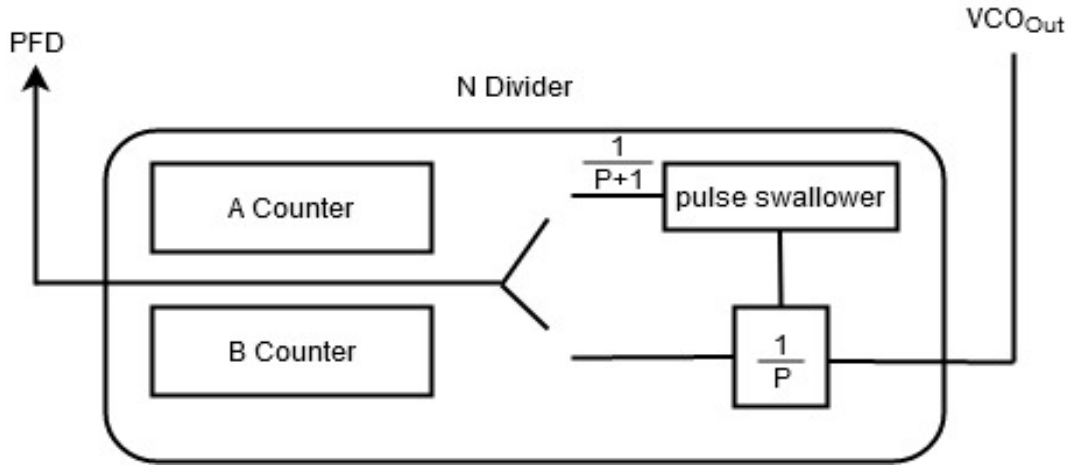


**Figure 3.11:** A block diagram of a dual modulus prescaler

### 3.3 Fractional-N Phase Locked Loop

The calculations and the working of fractional-N PLLs are similar to those of integer N PLLs. The only time a distinction is made is during the process of putting the mechanism for division into action. This is due to the fact that fractional division must be obtained through the dynamic switching between at least two integer values. There are no devices available that can divide by fractional values.When we divide the total number of F cycles by N+1 for K times and by N for F-K times, the result is a fraction, which can be shown in equation 3.7 (Barret, 1999).

$$N_T = \frac{K(N+1)+N(F-K)}{F} = N + \frac{K}{F} \tag{3.7}$$

where $N_T$ is the total divider value, N is the feedback division ratio, K is the fraction register and F is the modulus register.

The fractional-N PLL is integrated with a fractional accumulator, as shown in Figure 3.12, which changes the N value dynamically when the PLL is locked. When the accumulator reaches its maximum capacity, the carry out is activated, and the value of the A counter is increased by one. The value of F indicates the number of reference cycles that are contained within each complete fractional divide cycle. On the other

hand, the value of K indicates the number of times the device will divide by N+1 during each full fractional divide cycle. In the event that we take into consideration a dual modulus divider, for every K times during F cycles, if a is increased by one, then N is likewise increased by one (Barret, 1999).

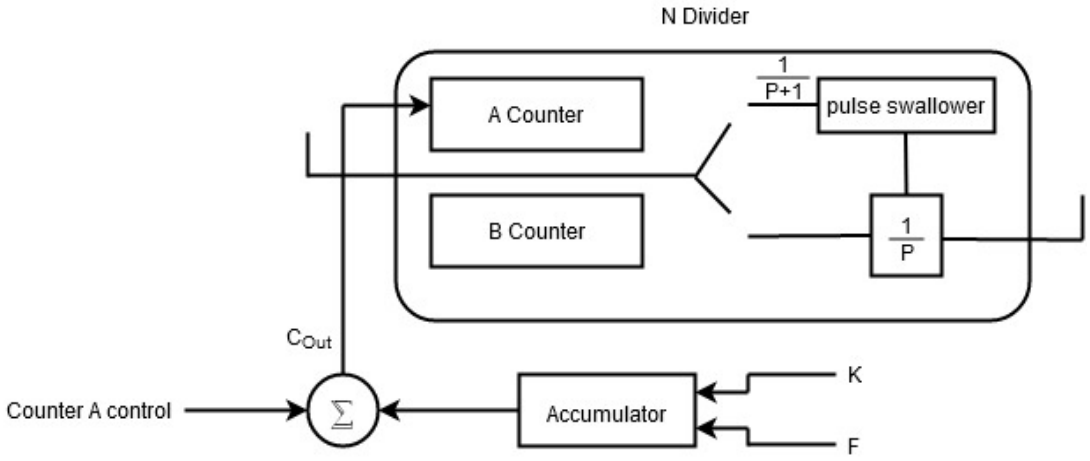


**Figure 3.12:** The operation of the accumulator

When N is incremented by 1, it causes an abrupt change in phase on a periodic basis, as shown in Figure 3.13, which leads to a spurious signal known as a fractional spur. These spurs fall into the loop bandwidth since their frequencies are low, and they cause poor working of the synthesizer if not eliminated.

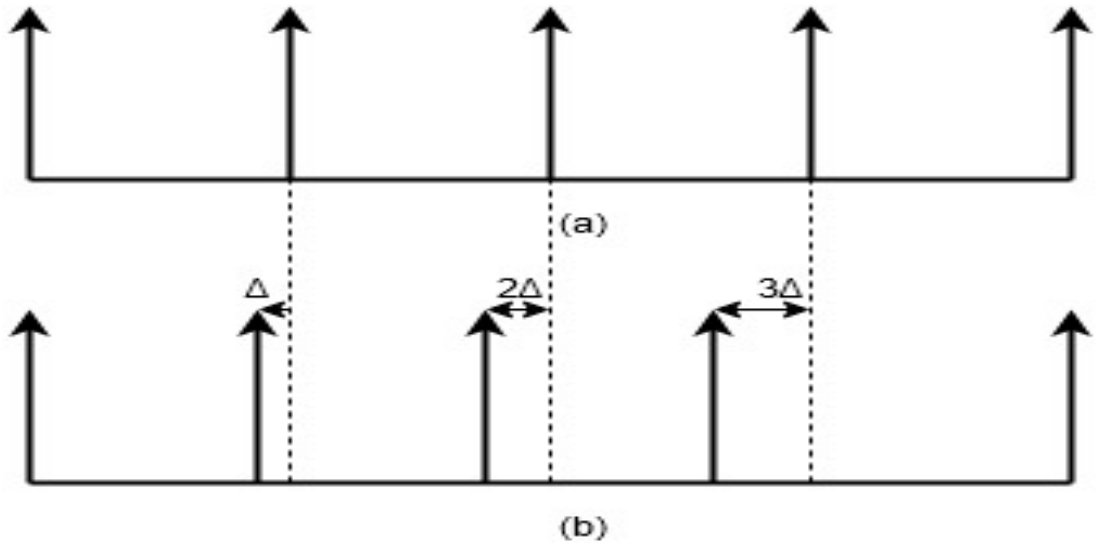


**Figure 3.13:** (a) The output that is desired from the divider (b) A sudden shift in phase resulting from the operation of a fractional divider of 1/4

Fractional spurs can be minimized by using analog techniques, such as current compensation in the loop filter and the introduction of phase delay at the phase detector, but it is not an efficient method. The efficient method to minimise fractional spurs is to use delta sigma ($\Sigma$-$\Delta$) modulators in the PLL, which make compensation in the digital domain (Banerjee, 2017).

### 3.3.1 Delta Sigma ($\Sigma$-$\Delta$) Modulators in Fractional Phase Locked Loop

Switching between two integer values and calculating the average of those values in order to arrive at a fractional value is the fundamental logic that underpins the operation of fractional N synthesizers, as was mentioned earlier. The switching between two integers is actually a first order $\Sigma$-$\Delta$ modulator although not common. In the vast majority of instances, $\Sigma$-$\Delta$ modulators begin with the second order, which toggles in and out of four integer values. In a nutshell, the output of an nth order $\Sigma$-$\Delta$ modulator jumps back and forth between $2^n$ integers.

The fractional spurs are improved with this design by increasing the number of division rates, which gets rid of the sudden phase change that generates the spurs in the first place. Increasing the division rate also pushes the fractional spurs caused by the divider to a higher frequency, enabling the loop filter to eliminate them easily (Bölücek, 2009). The first order delta sigma is modelled using a quantization noise or quantization error E(z) which is added to a divider as shown in Figure 3.14.

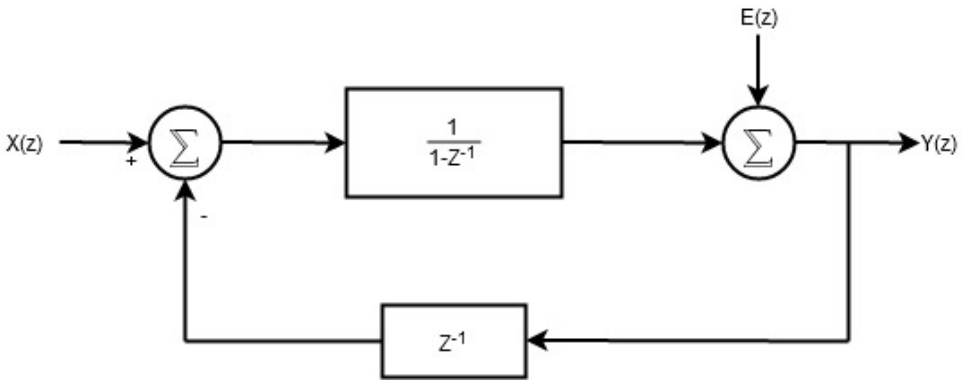


**Figure 3.14:** 1st order delta sigma modulator

Equation 3.8 determine the transfer function of the system depicted in figure 3.14.

$$Y(z) = X(z) + E(z)(1 - Z^{-1}) \tag{3.8}$$

where Y(z) is the output function, X(z) is the input function, E(z) is the quantization error and $Z^{-1}$ is one clock delay for feedback path.

This equation shows that previous value of phase error is subtracted from the present value and then reflected to the output which is actually a digital high pass filtering operation. If modulator order is increased, quantization noise is pushed to higher frequencies to be filtered by the loop filter more easily.

## 3.4 Phase Noise

This is the noise that is present on the output signal, and it is created by erratic changes in the signal's phase. The phase noise is measured in decibels with respect to carrier power per hertz (dBc/Hz) (Banerjee, 2017).

### 3.4.1 Phase Noise Modeling of Phase-Locked-Loop

For modeling PLL phase noise, each noise source in the system is examined individually, and the transfer of every noise source to the output is computed. This is done so that the noise may be accurately modeled. Figure 3.15 depicts a phase noise model. In the model, the noise from the blocks is indicated in form of an angle and a current, where $\theta_{REF}$, $\theta_{PFD}$, $I_{NCP}$, $\theta_{VCO}$ and $\theta_{NDIV}$ denote the reference noise, PFD noise including the reference divider noise, CP noise, VCO noise and the N divider noise, respectively.

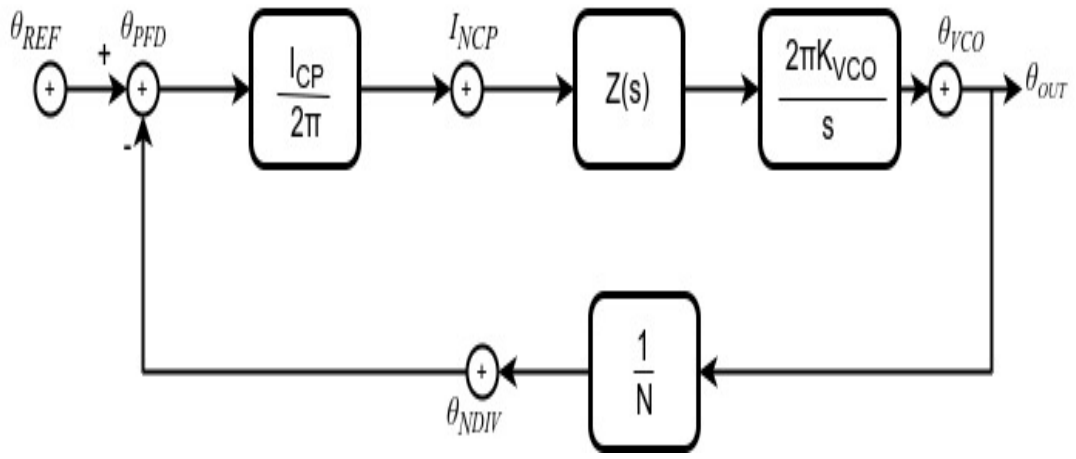


**Figure 3.15:** A block diagram showing a Linear phase domain PLL

As indicated in equation 3.9, the product of the CP transfer function, the low pass filter transfer function, and the VCO transfer function determines the forward transfer function of the system represented in Figure 3.15. Using the feedback transfer function that is defined in equation 3.10, one may express the transfer function of each phase noise source using the equations that are illustrated in 3.11 to 3.13 (Bölücek, 2009).

$$G(s) = \frac{I_{CP}}{2\pi} Z(s) \frac{2\pi K_{VCO}}{s} \tag{3.9}$$

$$H(s) = \frac{1}{N} \tag{3.10}$$

where, G(s) is the forward transfer function, H(s) is the feedback transfer function, $I_{CP}$ is the charge pump current, $K_{VCO}$ is the VCO gain, Z(s) is the filter function, N is the value of feedback divider and s is the complex frequency plane.

$$T_{REF}(s) = T_{PFD}(s) = T_{NDIV}(s) = \frac{G(s)}{1 + G(s)H(s)} \tag{3.11}$$

$$T_{NCP}(s) = \frac{2\pi}{I_{CP}} \frac{G(s)}{1 + G(s)H(s)} \tag{3.12}$$

$$T_{VCO}(s) = \frac{1}{1 + G(s)H(s)} \tag{3.13}$$

where $T_{REF}$, $T_{PFD}$, $T_{NCP}$, $T_{VCO}$ and $T_{NDIV}$ is the transfer function of the reference, PFD, charge pump, VCO and N divider respectively.

In the event that noise is introduced at one of the sources, the transfer function that corresponds to that source is used to multiply the noise.In their transfer functions, the reference noise, N divider noise, PFD noise, and charge pump noise can all be observed to share a common component. As shown in Figure 3.16, the in-band noises are most prominent at lower offsets, whereas the VCO noise is most prominent at higher offsets, where fc, fz and fp is cutoff frequency, zero location, and pole location, respectively.

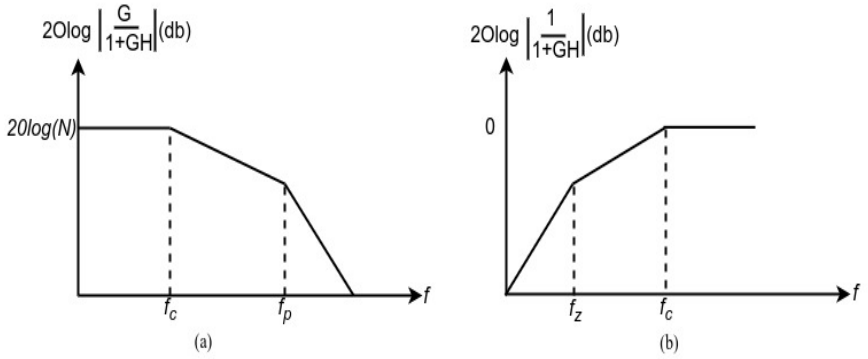


**Figure 3.16:** Plot showing the common factor transfer function multiplying (a) the CP, the PFD, the reference, and the N divider noise (b) the VCO noise.

When frequency goes to zero, G(s) approaches to infinity and the expression in figure 3.16 (a) reduces to 20 times the natural logarithm of N in dB as shown in equation 3.14 while the expression in figure 3.16 (b) reduces to 0 (Banerjee, 2017).

$$\lim_{G(s)\to\infty} \left( \frac{NG(s)}{N + G(s)} \right) = N \tag{3.14}$$

where, G(s) is the forward transfer function, and N is the value of feedback divider.

As a result, the noise of the VCO that occurs inside the bandwidth of the loop is

negligibly low, whilst the in band noise sources are multiplied by $20log(N)$.

## 3.5 Phase-Locked-Loop Spurs

Spurs are introduced into a signal as a result of a concentration of noise at a certain offset from the carrier, which results in interference with the signal (Banerjee, 2017). Spurs are of several types, depending on their causes.

### 3.5.1 Reference Spurs

These spurs can be seen occurring at multiples of the frequency of the PFD. The mismatches and current leakages that occur in the charge pump of the PLL are the primary factors that contribute to these spurs (Banerjee, 2017). While the PLL is at its locked state, the charge pump is usually in tri-state, and in this state, the charge pump experiences a small parasitic leakage that causes modulation on the tuning line of the VCO. Mismatching the charge pump is another key source of the reference spurs that should be considered, which comes as a result of the difference in sinking current and sourcing current in charge pump. The current is sourced by a PMOS device and sunk at a different speed by a NMOS device, resulting in current spikes at the PFD frequency device (Bölücek, 2009).

### 3.5.2 Fractional Spurs

These spurs only occur in the fractional-N synthesizers. These spurs are caused by accumulator or $\Sigma$-$\Delta$ modulator operation. Fractional PLL relies on averaging the results of two or more integer N values instead of dividing by fractional numbers directly. The sudden modifications that result from this averaging lead to the formation of fractional spurs that correspond to the channel spacing or fractions of it (Bölücek, 2009).

### 3.5.3 Crosstalk spurs

The crosstalk spurs are caused by some external signal. An example is a Crystal reference crosstalk spur caused by excessive gain in the crystal inverter structure and appears at an offset frequency same as the crystal reference frequency from the carrier. Other crosstalk spurs are known as external crosstalk which are any spurs that occur and are not related to the PLL. This spur may be caused by the power supplies, computer screens, or any other devices connected to the circuit. The spectrum analyzer may also cause it(Banerjee, 2017).

## 3.6 Voltage Regulators

This circuit type generates and keeps a constant output voltage, irrespective of the change that occurs in the input voltage or the load conditions. Linear regulators constitute the most common and important kind of voltage regulators.

### 3.6.1 Linear Regulators

. These regulators are controlled by a high-gain operational amplifier. As can be seen in Figure 3.17, the primary elements that make up a linear regulator are an amplifier, a feedback resistor, an output driver transistor, and a reference voltage unit. Linear regulators are able to keep the output voltage at a constant level by managing how much resistance the driver transistor presents to the controlling circuit. An amplifier receives the feedback voltage ($V_{FB}$) from a feedback resistor ($R_{FB}$), which is achieved through the division of the output voltage by R1 and R2.

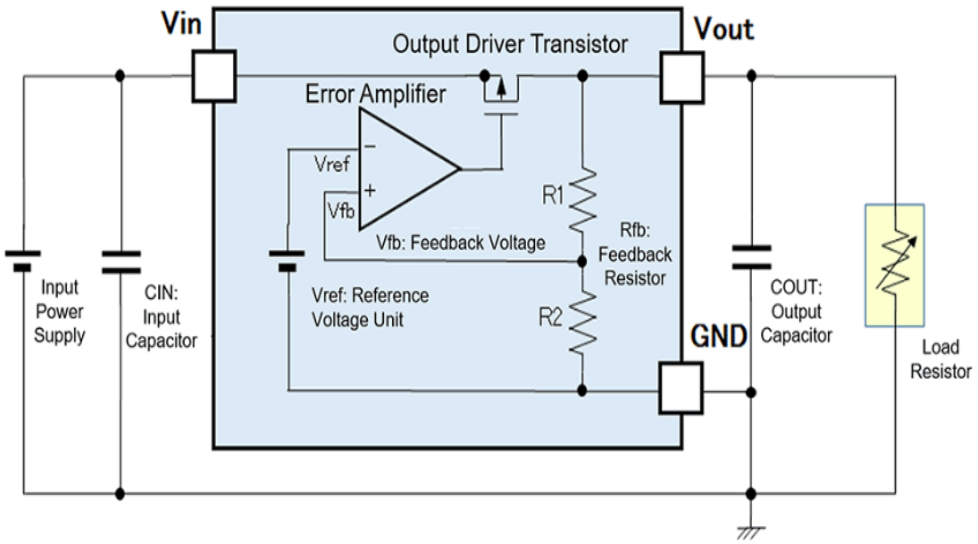


**Figure 3.17:** The basic setup of a linear regulator

The implementation of a reference voltage unit consists of taking the difference in voltage that is present between the thresholds of two MOS transistors. The level of voltage that is produced by it ranges anywhere between 0.6 and 1.0 V. The resistance of the output driver transistor is adjusted by the negative feedback circuit after a comparison is made between the feedback voltage and the reference voltage ($V_{REF}$). This is done to ensure that there is no difference in voltage between the two. As demonstrated in equation 3.15, the VREF and $V_{FB}$ are held constant at an equal value in a normal state (Lecturer, 2014).

$$V_{FB} = V_{REF} \tag{3.15}$$

The $V_{FB}$ is calculated using equation 3.16.

$$V_{FB} = V_{OUT}\left(\frac{R_2}{R_1 + R_2}\right) \tag{3.16}$$

The output voltage is given by equation 3.17 after simplifying equation 3.16 using equation 3.15.

$$V_{OUT} = V_{REF}\left(\frac{R_1 + R_2}{R_2}\right) \tag{3.17}$$

where $V_{FB}$ is the feedback voltage, $V_{REF}$ is the reference voltage, $V_{OUT}$ is the output voltage and $R_1$ and $R_2$ are resistors.

# CHAPTER FOUR

# MATERIALS AND METHODS

## 4.1 Introduction

This section explains the simulation tool used in the design of the signal generator, as well as the specifications used. This chapter also contains an explanation of the working principle of the components that were employed in the design. This chapter provides an overview of the procedures used to implement the signal generator.

## 4.2 System Design

The system was designed by simulation, which followed the specifications of the proposed signal generator. Simulation is an important part of the electronic system design process because it provides an early understanding of the design while also assisting in verifying that the design will function as intended. In this work, simulation was also used to determine the system behavior of the synthesizer. The goal was to create a low phase noise signal generator capable of generating frequencies ranging from 35 MHz to 3 GHz by determining the optimum bandwidth and phase margin through simulation.

### 4.2.1 The Signal Generator Specifications

The specifications were identified after realizing the need to have a signal generator that can generate a wide frequency range with high resolution and low noise at a low cost. The specifications were broken down into two distinct groups, the functional requirements, and the non-functional requirements.

Functional requirements specifications define what the signal generator is supposed to do. They include:

1. Allowing for control of registers stored on the chip through a three-wire interface.

2. To output an integer or a fractional frequency.

3. To generate a sine waveform.

4. To facilitate the selection of the output frequency.

Non-functional requirements define the constraints of the signal generator, which enable us to meet the functional requirements. Table 4.1 lays out all of these requirements.

**Table 4.1:** PLL based signal generator non-functional requirements

| **Parameter** | **Specification** |
|---|---|
| Frequency Range | 35 MHz to 3 GHz |
| Resolution | $\leq$ 1 kHz |
| Tuning speed | $<$ 1 ms |
| Phase Noise range | -90 dBc/Hz to -100 dBc/Hz |
| | at 100 kHz offset frequency |
| Reference frequency range | 10 MHz to 20 MHz |

### 4.2.2 Simulation Tool used

The simulation tool used in designing the PLL synthesizer was ADIsimPLL™. The reason for choosing this tool is that it is comprehensive, easy to use, and readily available for free. This program makes it possible to do simulations for all of the important non-linear phenomena that significantly affect PLL performance. These effects include frequency transients and precise models for phase noise, among others. The user can make changes to the PLL's design parameters, such as the VCO sensitivity, loop bandwidth, component values, and phase margin, and see those changes reflected in real-time in the simulation results. This allows the user to easily tailor and optimize the design to meet their unique needs.

### 4.2.3 Phase-Locked-Loop Chip

The PLL chip used in the simulation was the ADF4351 from Analog Devices shown in Figure 4.1. This chip has 32 pins and is a Lead Lead Frame Chip Scale Package (LFCSP) measuring $5mm \times 5mm$ in body and $0.75mm$ in height.

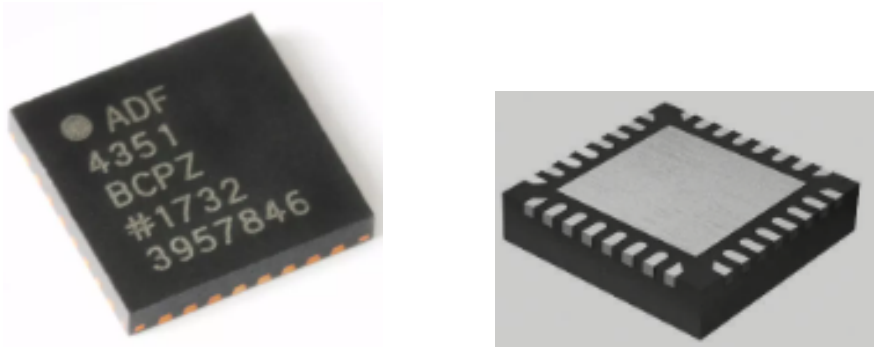


**Figure 4.1:** ADF4351 phase locked loop chip (Devices, 2008)

This chip was chosen because it has the potential to enable the construction of integer-N or fractional-N PLL frequency synthesizers when combined with an external loop filter and a reference frequency. This chip contains a VCO with a frequency range of 2.2–4.4 gigahertz. The divide-by-1/-2/-4/-8/-16/-32/-64 circuits are also integrated within the device. A standard 3-wire interface is used to control all of the registers that are located on the chip. This chip requires a power source that is between 3.0 V and 3.6 V in order to function properly.

Figure 4.2 provides a representation of the functional block diagram for the ADF4351. The fact that this particular synthesizer has increased noise performance, as well as frequency resolution, is the primary advantage that it offers. The ADF4351 is equipped with a modulus (MOD) that may be set to any integer value between 2 and 4095 and a third-order delta sigma modulator. The range of possible MOD values is from 2 to 4095. While operating in low spur mode, a modulus value of at least 50 is permissible.

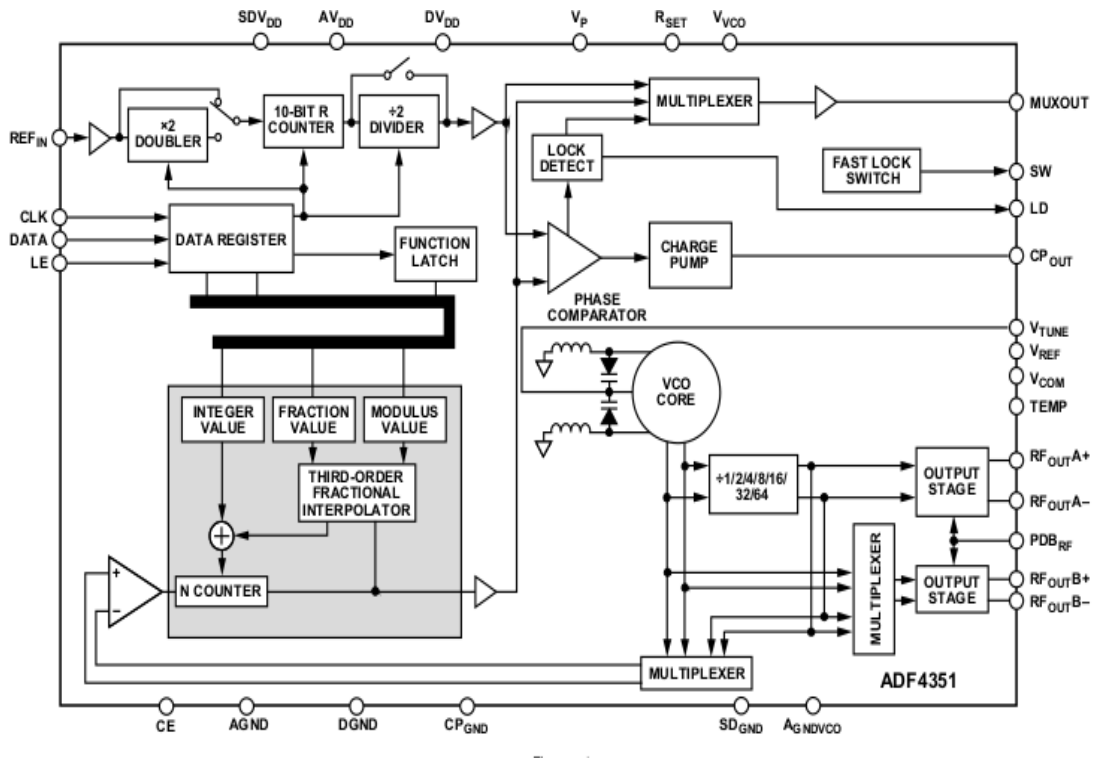


**Figure 4.2:** An illustration of the ADF4351's functional block diagram (Devices, 2008)

The ADF4351 chip also includes a reference frequency doubler part, which enables the reference frequency to be doubled and improves phase noise performance of the system. The reference signal is divided in half using the divide-by-2 component. This results in the PFD frequency having a duty cycle of fifty percent, which is necessary for the effective operation of Cycle Slip Reduction (CSR). CSR makes it possible to achieve shorter lock times at the expense of lower noise effectiveness.

#### 4.2.3.1 Reference Input Section

The reference frequency of 10 MHz from a crystal oscillator was used in the simulation of the PLL. This frequency travels via the reference input stage of the ADF4351, which is seen in Figure 4.3. The switches (SW1 and SW2) are in their normal closed positions, whilst switch SW3 is in its typical open position. When the power is turned off to the PLL, the SW3 switch is closed, while the SW1 and SW2 switches remain open. During the power-down process, there will be no loading on the REFIN pin because of this

setting.

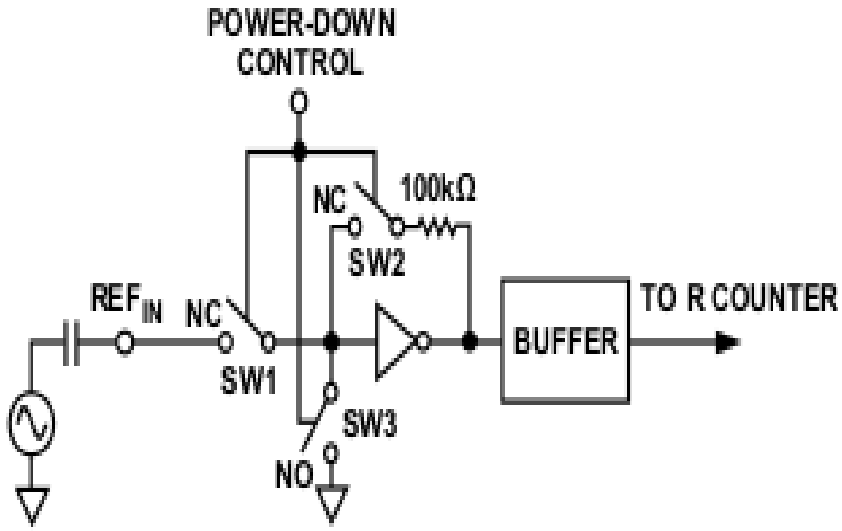


**Figure 4.3:** Reference Input Stage in ADFF351 (Devices, 2008)

#### 4.2.3.2 ADF4351 Dividers

Along the PLL feedback path, a division ratio can be achieved with the help of the N divider. The division ratio is obtained from the INT, FRAC, and MOD values that are used to construct this divider, as depicted in Figure 4.4; these values define the ratio of the division.

The output frequency is given by equation 4.1.

$$RF_{OUT} = f_{PFD}\left(INT + \left(\frac{FRAC}{MOD}\right)\right) \tag{4.1}$$

where the frequency of the VCO output is denoted by RFOUT, the 16 bit counter divider is denoted by INT, the numerator of the fractional division is denoted by FRAC, MOD denotes the preset fractional modulus, and fPFD is the frequency of the phase detector, which is calculated using equation 4.2 (Devices, 2008).

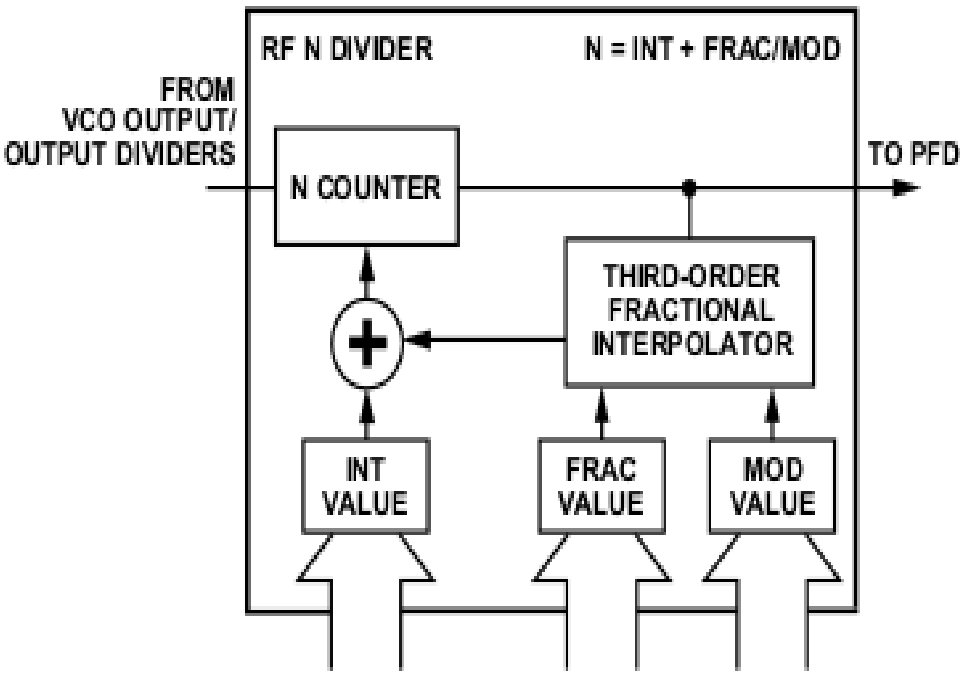


**Figure 4.4:** ADF4351 RF N Divider (Devices, 2008)

$$f_{PFD} = REF_{IN}\left(\frac{1+D}{R(1+T)}\right) \tag{4.2}$$

where REFIN is the frequency at which the reference counter is operating, D is the bit that doubles the reference frequency, R is the division ratio of the reference counter, and T is the bit that divides the reference frequency by two.

#### 4.2.3.3 Multiplexer Output

The multiplexer output of the ADF4351's primary function is to provide the user with the ability to access the internal points located on the chip, and the state of this output is determined by the bits located in Register 2. The MUXOUT section is shown in Figure 4.5 in block diagram form.

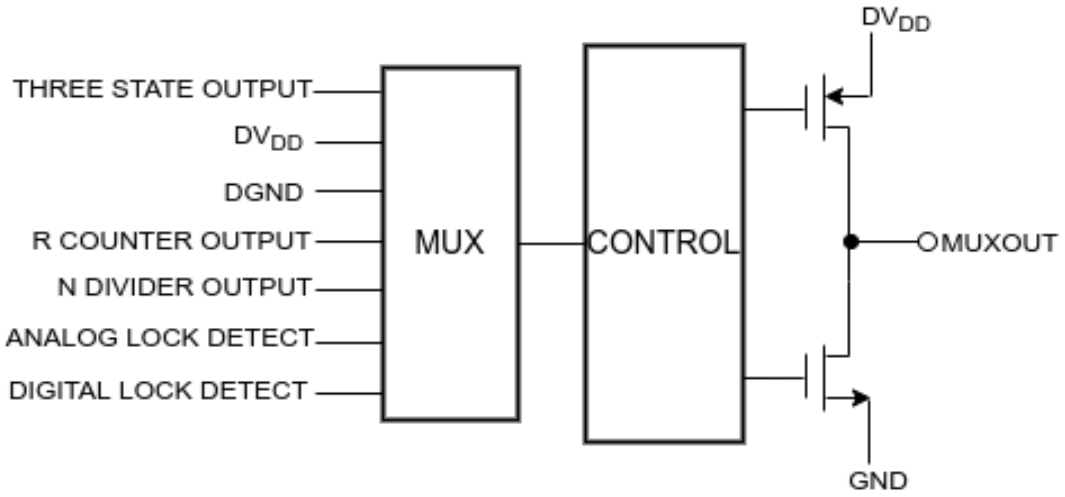


**Figure 4.5:** Schematic diagram of multiplexer output

#### 4.2.3.4 Input Shift Registers in ADF4351

The digital component of the ADF4351 consists of a 16-bit N counter, a 10-bit R counter, a 12-bit fraction counter, and a 12-bit modulus counter. Moreover, there is a 12-bit fraction counter. Data is clocked into the 32-bit shift register whenever the CLK signal experiences a rising edge, with the most significant bit (MSB) being clocked in first. The status of the three control bits (C3, C2, and C1) in the shift register controls which latch the data is destined for once it has been transferred from the shift register to one of the six latches on the rising edge of LE. It is clear from looking at Figure 4.6, which depicts the timing diagram, that the control bits are the three least significant bytes (DB2, DB1, and DB0).

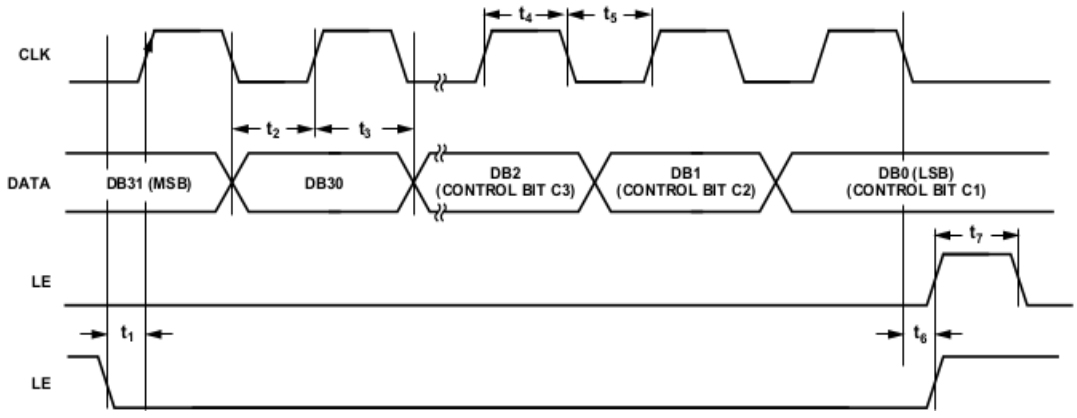


**Figure 4.6:** Timing diagram showing how data is clocked in registers (Devices, 2008)

Table 4.2 displays the truth table for these bits.

**Table 4.2:** The truth table for the Control Bits C3, C2, and C1.

| Control Bits | | | Registers |
|---|---|---|---|
| C3 | C2 | C1 | |
| 0 | 0 | 0 | Register 0 (R0) |
| 0 | 0 | 1 | Register 1 (R1) |
| 0 | 1 | 0 | Register 2 (R2) |
| 0 | 1 | 1 | Register 3 (R3) |
| 1 | 0 | 0 | Register 4 (R4) |
| 1 | 0 | 1 | Register 5 (R5) |

#### 4.2.3.5 Voltage Controlled Oscillator integrated in ADF4351

As can be seen in Figure 4.7, the VCO core is composed of three independent VCOs. Each of these VCOs uses 16 overlapping bands to span a wide frequency range without producing a significant amount of phase noise, spurs, or sensitivity in the VCO itself. The VCO and band choose logic will automatically select the appropriate VCO when the power supply is first turned on or whenever Register zero is refreshed. It takes ten PFD cycles, multiplied by the output of the R counter, which is the value of the band choice clock divider, to complete both the VCO and the band selection.

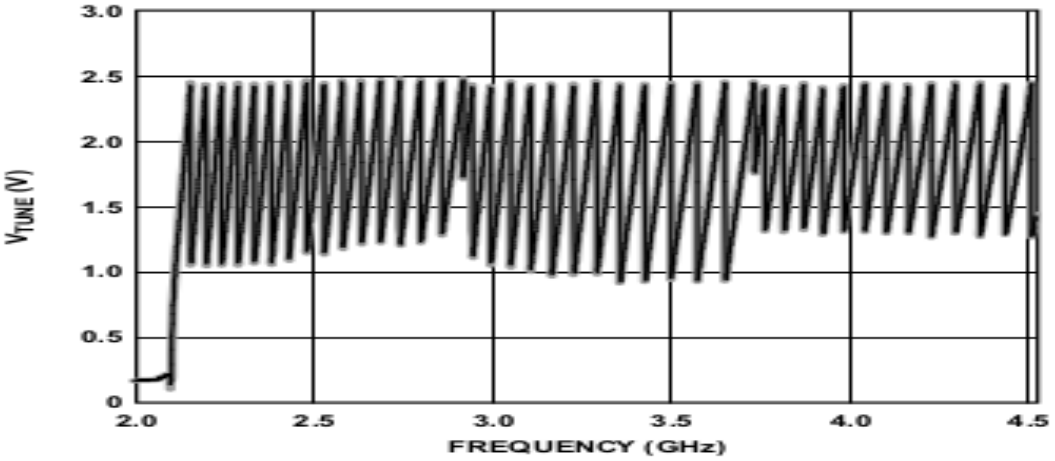


**Figure 4.7:** Voltage- Frequency characteristics of the VCO (Devices, 2008)

As the tuning voltage shifts inside a band as well as across bands, the sensitivity ($K_V$) of the VCO shifts accordingly, displaying a range of values. Because it is the value that is closest to an average, the sensitivity value of 40 MHz/V is the most accurate value to utilize in applications that encompass a large frequency range. Figure 4.8 illustrates how sensitivity changes depending on the frequency of the VCO as well as the average value for the frequency band.

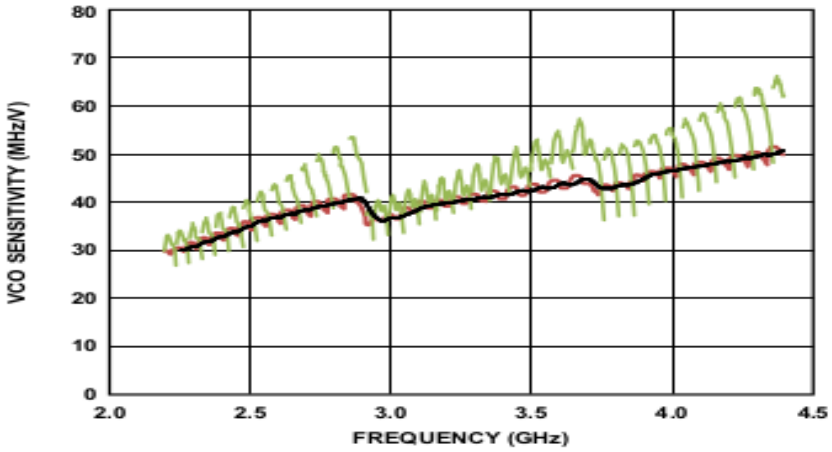


**Figure 4.8:** VCO Sensitivity (KV) vs. Frequency (Devices, 2008)

#### 4.2.3.6 Output Stage of the ADF4351

According to the diagram in Figure 4.9, the $RF_{OUT}A+$ and $RF_{OUT}A-$ output pins of the ADF4351 are wired to the collectors of an NPN differential set. This pair is driven by the outputs of the VCO through the buffer.

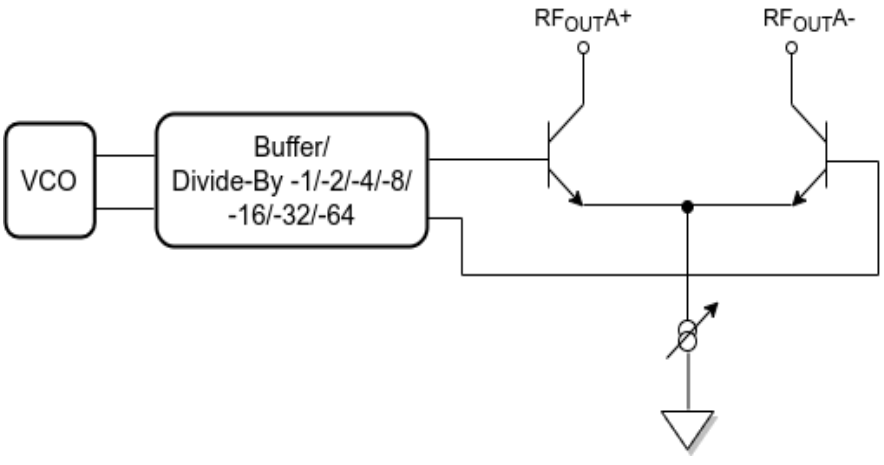


**Figure 4.9:** Output stage of ADF4351 PLL Chip

### 4.2.4 Loop Filter

The loop filter was designed by simulating different loop bandwidths and phase margins using the ADIsimPLL simulation tool. The primary purpose of the low pass filter that was utilized was to get rid of the undesirable high frequency components. These components had the potential to upset the PLL's balance by over modulating the VCO.In addition to that, the charge pump output was converted into a voltage by the filter, which was then utilized to tune the VCO (Bölücek, 2009). A passive filter of the third order with the architecture depicted in Figure 4.10 was chosen as the one to use. A third order filter was selected because it improves filter attenuation response, which helps in the filtering of spurs.

Equation 4.3 can be used to determine the impedance of the loop filter shown in Figure 4.10 (Banerjee, 2017).

$$Z(s) = \frac{1 + sT_2}{s(C_1 + C_2 + C_3)(1 + sT_1)(1 + sT_3)} \tag{4.3}$$

where T2 is a zero given by equation 4.4, T1 and T3 are poles given by equations 4.5 and 4.6 respectively.

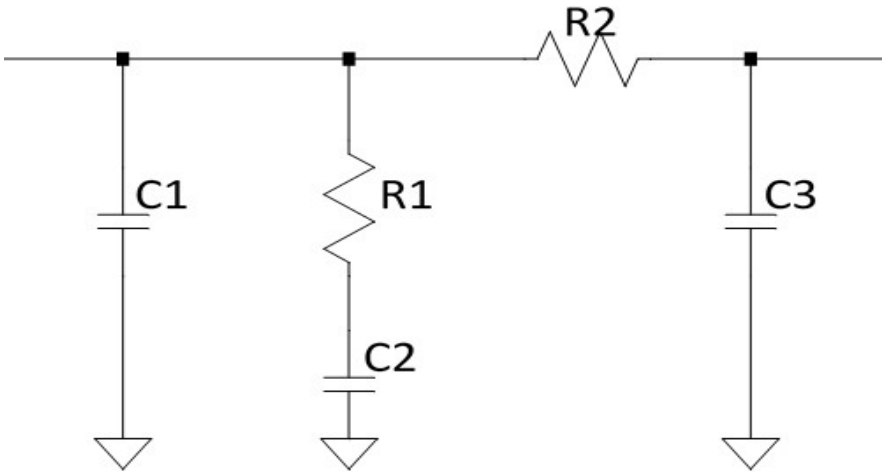


**Figure 4.10:** The structure of a third-order passive loop filter

$$T_2 = R_2 C_2 \tag{4.4}$$

**Table 4.3:** A Generic Routh array

| | | | | |
|---|---|---|---|---|
| $s^n$ | $a_0$ | $a_2$ | $a_4$ | ... |
| $s^{n-1}$ | $a_1$ | $a_3$ | $a_5$ | ... |
| $s^{n-2}$ | $b_1 = \frac{a_1 a_2 - a_0 a_3}{a_1}$ | $b_2 = \frac{a_1 a_4 - a_0 a_5}{a_1}$ | ... | ... |
| $s^{n-3}$ | $c_1 = \frac{b_1 a_3 - a_1 b_2}{b_1}$ | ... | ... | ... |
| $s^0$ | ... | ... | ... | ... |

$$T_1 \approx \frac{C_1 C_2 R_2}{C_1 + C_2 + R_2} \tag{4.5}$$

$$T_3 \approx R_3 C_3 \tag{4.6}$$

### 4.2.5 Checking the Stability of Phase-Locked-Loop

If a system's closed loop transfer function has poles that are all negative in real terms, then that system is considered to be stable. The Routh's stability criteria approach was utilized in order to carry out the task of determining whether or not the PLL system is stable. With the use of this method, it is possible to check, without the need to explicitly compute the poles of the PLL closed loop transfer function, to see if they are all located in the left-hand plane. In addition, the Routh array is where the coefficients of the characteristic equation are placed in a table. This table is referred to as the Routh array, and it can be seen in Table 4.3. It is essential and sufficient for the values in the first column of the Routh's array to be positive for a system to be considered stable (Clark, 1992). Taking into consideration a system, the characteristic equation of which is presented in equation 4.7, Routh's array can be generated using the coefficients of the equation as shown in table 4.2.

$$q(s) = a_0 s^n + a_1 s^{n-1} + a_2 s^{n-2} ... + a_{n-1} s + a_n \tag{4.7}$$

In the PLL system, the open loop transfer function is represented by the equation 4.8.

This system uses a third order loop filter.

$$G(s) = \frac{NK(1+sT_2)}{s^2(A_2s^2+A_1s+A_0)} \tag{4.8}$$

where

$$K = \frac{K_{PD}K_{VCO}}{N} \tag{4.9}$$

Equation 4.10 describes the PLL's transfer function for the closed loop of the system.

$$\frac{G(s)}{1+\frac{G(s)}{N}} = \frac{NK(1+sT_2)}{s^4A_2+s^3A_1+s^2A_0+KsT_2+K} \tag{4.10}$$

Therefore, equation 4.11 contains the solution of the PLL system's characteristic equation, which may be found in the denominator of equation 4.10.

$$g(s) = s^4A_2+s^3A_1+s^2A_0+KsT_2+K \tag{4.11}$$

### 4.3 Building ADF4351 based Signal Generator

The signal generator was built by adding a section of a microcontroller that controls the PLL digitally through the SPI interface, as shown in Figure 4.11. The microcontroller used in this work was the Arduino UNO. In addition, an LCD keypad shield was interfaced to the microcontroller so that it would be possible to write to and read from the registers of the PLL device. This led the selection of output frequency by pressing the buttons on the keypad and reading the output frequency on the LCD.

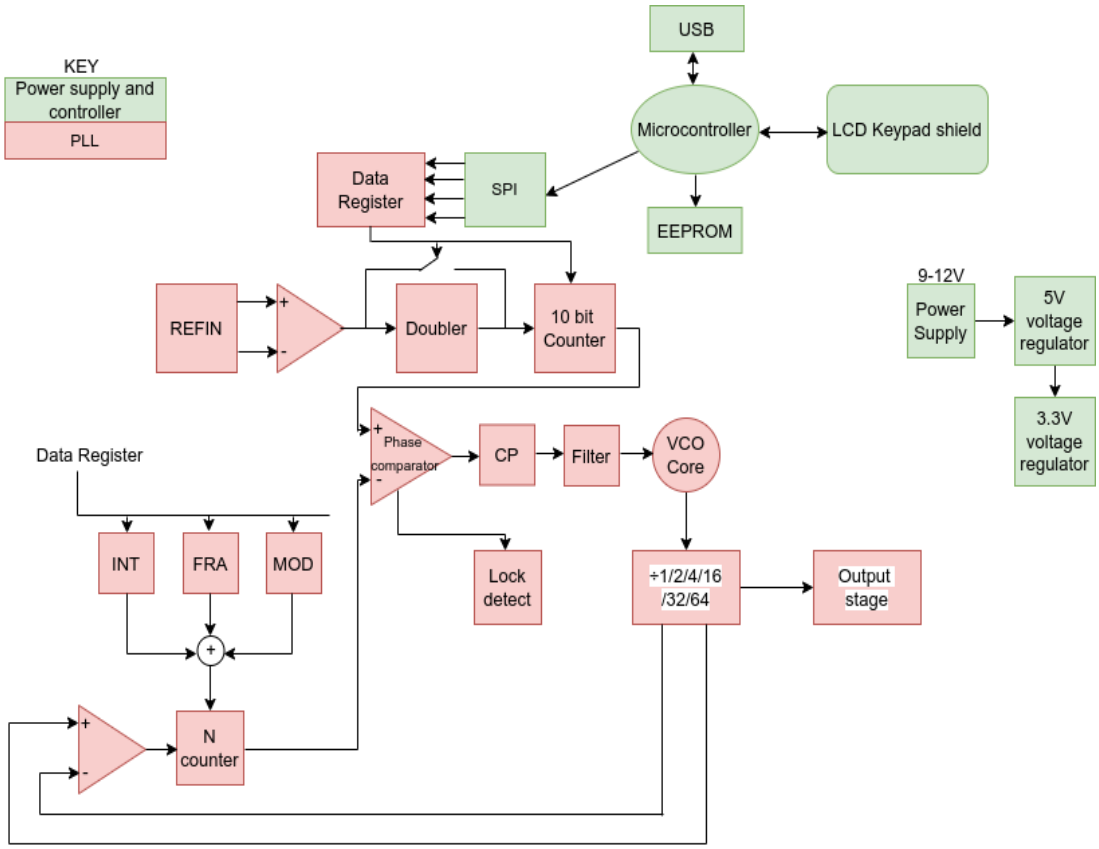


**Figure 4.11:** A block diagram of PLL based signal generator

### 4.3.1 Reference Crystal Oscillator

The ECS-2200X, seen in Figure 4.12 and produced by ECS Inc. International, was the reference crystal oscillator utilized in the construction of the signal generator.

**Figure 4.12:** 10 MHz ECS-2200X crystal oscillator (ECS, nd)

This crystal oscillator was selected because it produces a 10 MHz output frequency with a stability of 50 ppm. This crystal can drive both High-Density Complimentary Metal Oxide Semiconductor (HCMOS) and Transistor–transistor logic (TTL), and it

also features a tri-state enable/disable capability in pin 8. The pinout of this crystal oscillator is shown in Figure 4.13 and summarized in Table 4.4.

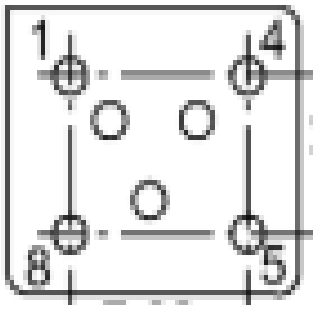


**Figure 4.13:** A picture of ECS-2200X crystal oscillator pinout (ECS, nd)

**Table 4.4:** Pin connections of ECS-2200X crystal oscillator

| Pin | Connection |
|---|---|
| 1 | Tri-state |
| 4 | Ground |
| 5 | Output |
| 8 | +5 V DC |

### 4.3.2 Arduino UNO Platform

This microcontroller board features an ATmega328P as its core component, as depicted in Figure 4.14. It includes a power jack, a reset switch, an In-Circuit Serial Programming (ICSP) header, and a USB connection. Furthermore, it boasts 14 digital input/output pins. Moreover, it features six analog inputs, a ceramic resonator operating at 16 MHz, and a reset button. This platform was chosen for this work because;

1. When compared to other platforms, the cost of Arduino boards is reasonable, and its integrated development environment (IDE) is easily accessible and free to download online.

2. Arduino programming software is a cross-platform that runs on many operating systems.

3. It is an open source platform, therefore allowing experienced designers to extend and improve it.

4. It is easy to connect to a computer through a USB port.

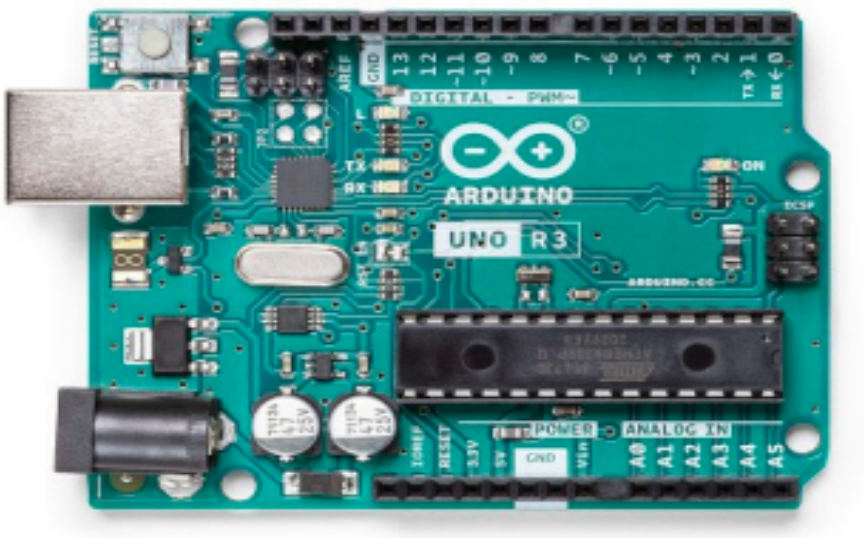


**Figure 4.14:** The Arduino UNO board (Arduino, nd)

#### 4.3.2.1 ATmega328P Microcontroller

This is the microcontroller that was used to vary the frequencies of the signal generator. It is an 8-bit, high performance, low power microcontroller from Atmel®AVR® family. This microcontroller has 28 pins, as shown in Figure 4.15, and is also integrated with an ADC, which makes it capable of handling both digital and analogue inputs.

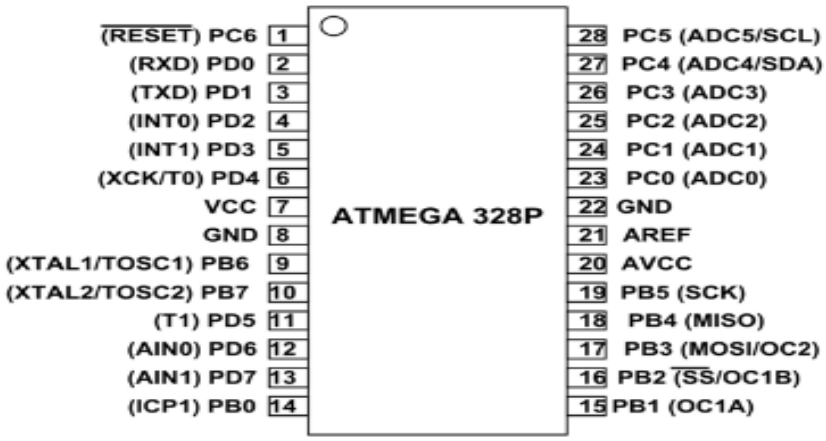


**Figure 4.15:** Pin diagram of ATmega328P Microcontroller (Atmel, 2015)

A sophisticated Reduced Instruction Set Computer (RISC) architecture underpins this microcontroller, which features 131 high-performance instructions in total, and has 32×8 general purpose registers. Because it achieves a throughput of 1 MIPS per MHz, the designer is able to optimize the power consumption while maintaining the desired level of processing speed. Figure 4.16 presents a diagrammatic representation of the internal architecture of the microcontroller.The Central Processor Unit, often known as the CPU, is like the brain of the microcontroller. It is responsible for regulating the execution of the program by accessing the storage, conducting calculations, controlling the peripherals, and responding to interrupts.

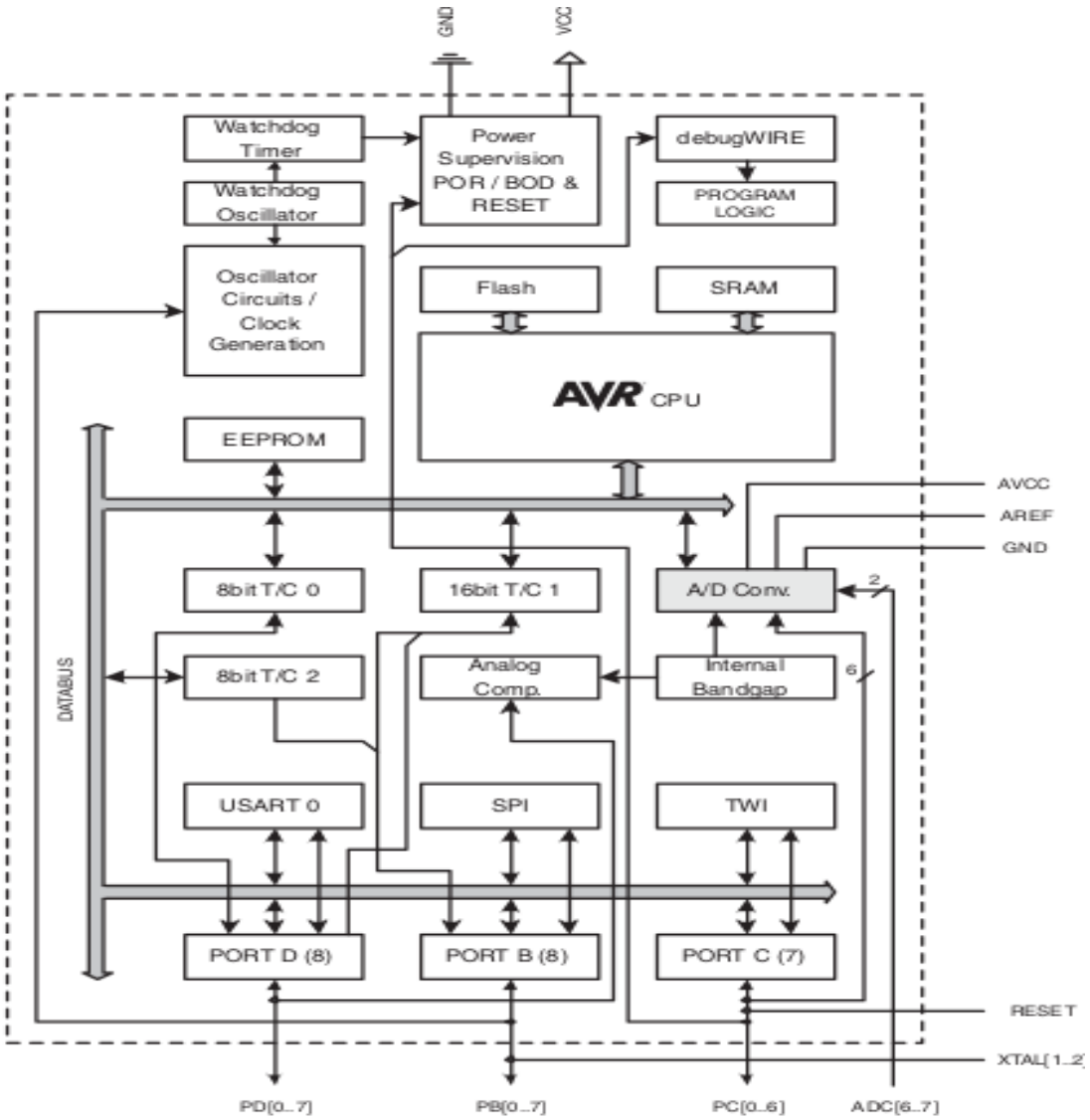


**Figure 4.16:** Block diagram of ATmega328P Microcontroller (Atmel, 2015)

### 4.3.3 LCD Keypad Shield

This LCD Keypad shield was selected for this work because it is compatible with Arduino boards. It also provides a straightforward design, which makes it easier for users to navigate the available options on the menu and select the appropriate one. As can be seen in Figure 4.17, this shield features a 2x16 LCD display in addition to six momentary push buttons labeled choose, up, right, down, and left, as well as a reset button. The Arduino board is interfaced with the LCD through pins 4, 5, 6, 7, 8, 9 and 10. The contrast of the LCD is varied using a potentiometer. This shield requires a 5 volt power supply, which can be supplied through the USB supply or using an external adapter.

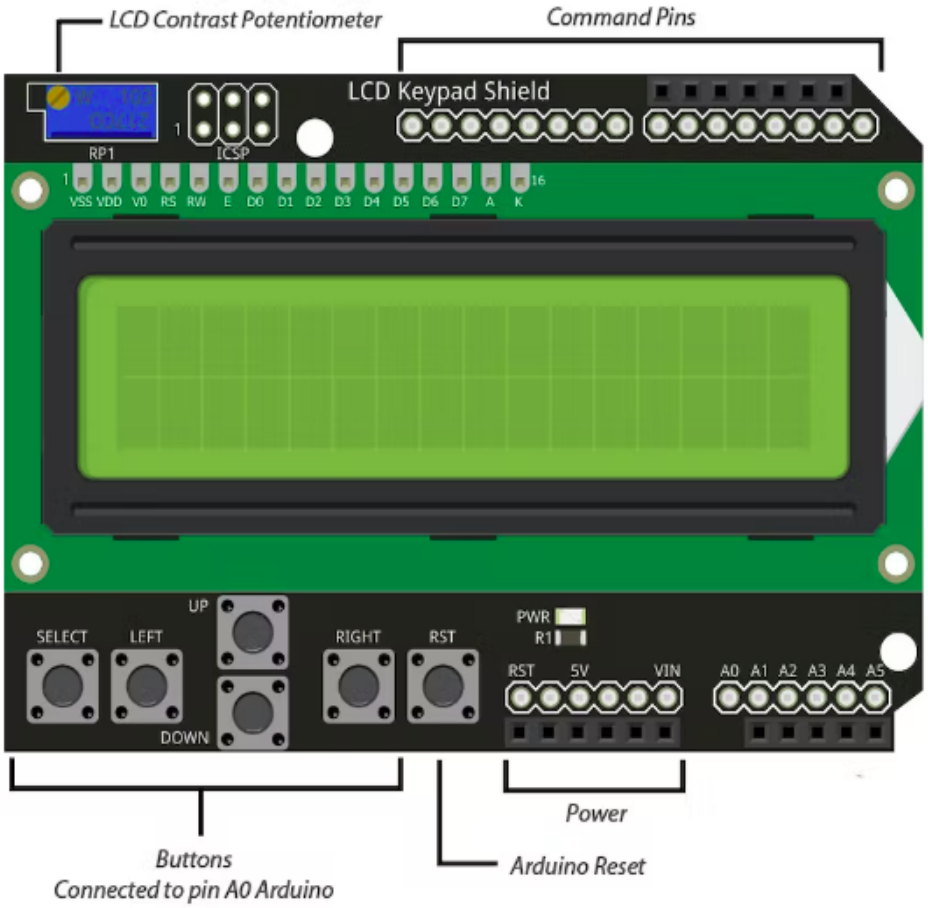


**Figure 4.17:** LCD Keypad shield (ElectroPeak, 2019)

The LCD Keypad shield pin-out and its connection to Arduino is summarized in Table 4.5.

As all of the switches on the shield are connected to analog Pin 0, the ADC is required in order to read them. The five switches are labeled select, up, right, down, and left respectively. In line with a resistive splitting circuit in the shield that is used to identify the key, a value is returned to the A0 pin whenever a key is pressed. Table 4.6 contains a summary of this information.

**Table 4.5:** Connection of LCD Keypad shield to Arduino Pins

| Arduino Pin | Shield Pin |
|---|---|
| 8 | LCD RS |
| 9 | LCD Enable |
| 7 | LCD D7 |
| 6 | LCD D6 |
| 5 | LCD D5 |
| 4 | LCD D4 |
| 10 | LCD Backlight |
| A0 | Buttons |

**Table 4.6:** Reading LCD Keypad Shield Keys

| Key | A0 Value |
|---|---|
| RIGHT | 0-60 |
| UP | 60-200 |
| DOWN | 200-400 |
| LEFT | 400-600 |
| SELECT | 600-800 |

### 4.3.4 Power Supply

The system's power supply was from a variable DC source, which was fixed to output 9V as this was within the range of the microcontroller. Since the reference oscillator circuit requires 5 V, a voltage regulator was employed to get 5 V from the supply of

9 V. This was accomplished with the help of the LM7805 voltage regulator, which is an integrated circuit that consists of three pins, as illustrated in Figure 4.18. The input pin is used for receiving the fluctuating voltage, the ground pin is used to establish a ground, and the output pin supplies a constant 5 V output. The LM7805 contains on-chip circuitry that can avoid damage in the case that it overheats or receives an excessive amount of load current. Moreover, it can supply a load current of up to 1.5 A.

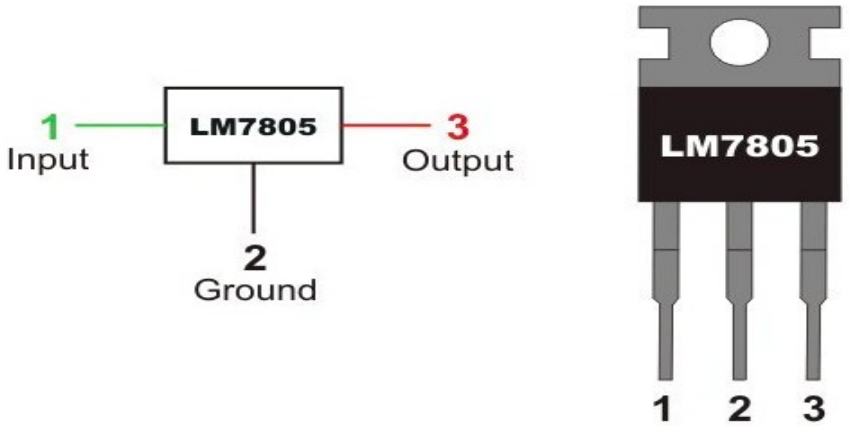


**Figure 4.18:** A picture of LM7805 and its pinout (Electronics, 2017)

According to its datasheet, the ADF4351 PLL chip requires a supply voltage of 3.3 V. Therefore, the LD1117aV33 voltage regulator was employed to get 3.3 V from the 5 V output.The LD1117A is a low drop voltage regulator that has the capacity to supply an output current of up to 1 A. Pin 1 is the ground, Pin 2 is the output, which supplies a constant 3.3 V, and Pin 3 is the input pin. The acceptable input voltage is in the range of 4–15 V. Figure 4.19 provides a visual representation of the pinout diagram.


**Figure 4.19:** 4.18: A picture of LD1117aV33 (Jordan, 2010)

## 4.4 Testing and Measurements of the Signal Generator using Spectrum Analyser

The measurements were taken using a spectrum analyser because it is designed with routines incorporated into the software that make the testing easier. This device also provides a more convenient and accurate method of measuring phase noise compared to other methods.

A Spectrum Analyzer is a piece of testing equipment that is utilized in the process of determining the relationship between the magnitude of an input signal and the frequency that falls within the capabilities of the instrument. The frequency appears on the horizontal (X) axis, while the magnitude is on the vertical (Y) axis. The phase noise of a signal can be measured with the help of this instrument, which does so by displaying the amount of noise power at a specific frequency that is offset from the carrier frequency. The spectrum analyser used in this work was the GSP-830.

### 4.4.1 GSP-830 Spectrum Analyser

The GSP-830 is a high performance spectrum analyzer that has a frequency range from 9 kHz all the way up to 3 GHz for its output. With a phase noise of -80 dBc/Hz for a 1 GHz frequency at a 20 kHz offset frequency, the GSP-830 is one of the most modern spectrum analyzers available today. It also has a USB host port on the front panel, which supports flash drives for saving setup information and capturing screens. A picture of the GSP-830 is shown in Figure 4.20.

The spectrum analyser has a frequency key and a span key for setting the frequency scale. Either the center-and-span method or the start-and-stop method can be used to set the scale. The center-and-span method involves defining the center point as well as the frequency range that surrounds it, while the start-and-stop method involves defining the beginning and the end of the frequency range. This instrument also has special span settings for full and zero spans.


**Figure 4.20:** A picture of GSP-830 Spectrum Analyser (Instrument, nd)

The Resolution Bandwidth (RBW) of a spectrum analyzer defines the width of the intermediate frequency (IF) filter used to separate peaks of different signals.Making the RBW narrow increases the capability of the equipment to separate signals with close frequencies. Narrow RBW, on the other hand, lengthens the sweep time over a given frequency span. The GSP-830 has a RBW of 3 kHz, 30 kHz, 300 kHz, and 4 MHz.

The smoothness of the trace on the display is determined by something called the ”Video Bandwidth,” or ”VBW.” When paired with RBW, VBW specifies the ability to separate the signal of interest from the noise in its surroundings or peaks that are close to it. GSP-830 has a VBW in the range of 10 Hz–1 MHz in 1-3 steps (Instrument, nd).

### 4.4.2 Testing the Signal Generator

Testing of the prototype signal generator was done in order to verify whether it could generate the expected frequencies and also determine the kind of waveforms produced. An examination of the output spectrum was carried out with the help of the spectrum analyser between the frequencies of 35 MHz and 3 GHz. A Cathod Ray Oscilloscope (CRO) was used to determine the output waveform. The CRO was capable of producing frequencies up to 100 MHz at their highest setting. As a result, only a spectrum analyzer was utilized for the testing of frequencies higher than 100 MHz. The test setup included

a laptop, USB cable, Arduino Uno microcontroller, Keypad shield, PLL system, CRO and spectrum analyser as depicted in Figure 4.21.

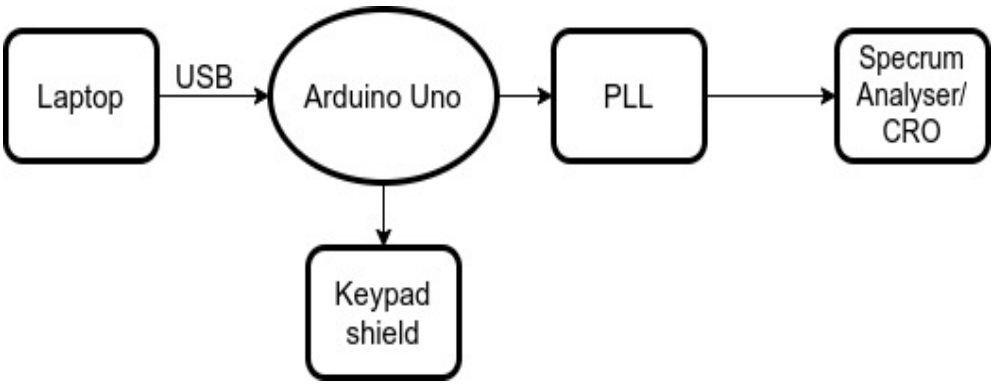


**Figure 4.21:** Hardware test setup

### 4.4.3 Measuring Phase Noise using GSP-830 Spectrum Analyser

In the frequency domain, the amount of phase noise is evaluated based on the amount of noise power present in a specific bandwidth. Since the standard bandwidth is 1 Hz, the measurements of a wider bandwidth were converted to the value for a 1 Hz bandwidth. The value of the noise is related to the carrier level since it varies by the number of decibels down on the carrier. As a direct consequence of this, the power of the noise was measured in decibels relative to the carrier (dBc). Because the noise level shifts in response to changes in the distance between the carrier and the signal, the offset from the carrier must be specified. As a result, the phase noise was quantified as the decibels below the carrier in a 1 Hz bandwidth at a certain frequency offset (dBc/Hz at x offset frequency), as demonstrated by equation 4.12.

$$Phase\ Noise = (L_C - L_O) - 10\log_{10}(RBW) \tag{4.12}$$

where, $L_c$ is the level of the carrier in dBm, $L_o$ is the level at offset frequency in dBm and RBW is the resolution bandwidth of the filter in Hz.

### 4.4.4 Measuring Spurs and Harmonics using GSP-830 Spectrum Analyser

Spurs are any unwanted signal generated outside the frequency band of interest. The concentration of spurs occurs at a particular offset from the carrier. The fundamental frequency is multiplied by an integer number of times to find the harmonics, which are a subset of spurs. Harmonics are generated when a signal with frequency $f > 0$ passes through a non-linear component (Rohde and Schwarz, 2012). In order to evaluate the amount of harmonics and spurs present in the output signal, a spectrum analyzer was utilized. Using the key for setting the full span, the scale was readjusted so that it could record all of the harmonics.

Recording the power levels at integer multiples of the fundamental frequency allowed for the level of the harmonics to be calculated and determined. Because it fluctuates by the number of decibels that are lower than the carrier, the value of the spurs is directly proportional to the level of the carrier. As a direct consequence of this, spurs were measured in decibels in relation to the carrier (dBc). The spurs were determined using equation 4.13.

$$Spurs = C_{Power} - O_{Power} \tag{4.13}$$

where $C_{power}$ is the level of the carrier in dBm and $O_{power}$ is the level of occurrence of spurs at a specific offset frequency.

# CHAPTER FIVE

# RESULTS AND DISCUSSION

## 5.1 Introduction

The outcomes of the simulations and experiments are presented and summarized in this chapter. The testing of the signal generator is covered in this chapter. The phase noise obtained after experimental work is discussed and compared to that obtained during simulation as well as the results obtained by other researchers. The level of spurs in the signal generator is outlined and discussed in this chapter.

## 5.2 Simulation of the Phase Locked Loop

The simulation tool known as ADIsimPLL was responsible for producing these results. The PLL synthesizer was designed using an ADF4351 PLL chip from analog devices, a 10 MHz reference frequency, and a third order loop filter.

### 5.2.1 Reference Crystal Oscillator

In the simulation, the reference crystal oscillator was utilized to produce a frequency of 10 MHz for use as a reference. The phase noise of the oscillator is described in Table 5.1, which covers offset frequencies ranging from 1 Hz to 100 MHz.

**Table 5.1:** Phase noise of the reference crystal oscillator

| Offset Frequency (Hz) | Phase Noise (dBc/Hz) |
|---|---|
| 1 | -70.00 |
| 10 | -90.00 |
| $1 \times 10^2$ | -110.00 |
| $1 \times 10^3$ | -130.00 |
| $1 \times 10^4$ | -150.00 |
| $1 \times 10^5$ | -166.99 |
| $1 \times 10^6$ | -170.00 |
| $1 \times 10^7$ | -170.00 |
| $1 \times 10^8$ | -170.00 |

The reference crystal oscillator was modelled as a custom reference oscillator for the simulation purpose. Figure 5.1 depicts the phase noise plot of the modelled crystal oscillator. The phase noise decreased with an increase in offset frequency until it reached 1 MHz, after which it remained constant for higher offset frequencies. This constancy indicated that the phase noise remained at its minimum for the simulated reference crystal oscillator.

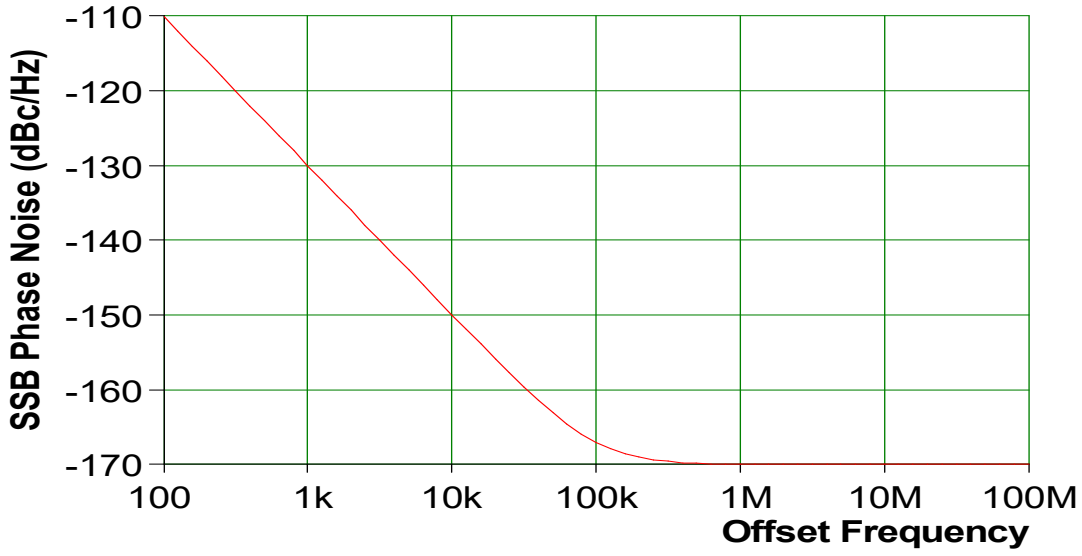


**Figure 5.1:** The phase noise of the reference oscillator after simulation

### 5.2.2 Voltage Controlled Oscillator (VCO)

An ADF4351 PLL IC was utilized in the simulation; this particular IC features an integrated voltage-regulated oscillator (VCO). The phase noise of the VCO is represented in Table 5.2, and it ranges from 1 kHz offset frequency up to 100 MHz offset frequency.

**Table 5.2:** Specification of the voltage controlled oscillator

| Offset Frequency (Hz) | Phase Noise (dBc/Hz) |
|---|---|
| $1 \times 10^3$ | -65.92 |
| $1 \times 10^4$ | -100.57 |
| $1 \times 10^5$ | -130.98 |
| $1 \times 10^6$ | -150.96 |
| $1 \times 10^7$ | -157.08 |
| $1 \times 10^8$ | -157.08 |

The fluctuation in phase noise that occurs with the offset frequency is illustrated in Figure 5.2. The phase noise decreased with an increase in offset frequency until it reached 10 MHz, where it remained constant for higher offset frequencies showing that -157.08 dBc/Hz was the minimum phase noise of the simulated voltage controlled oscillator.

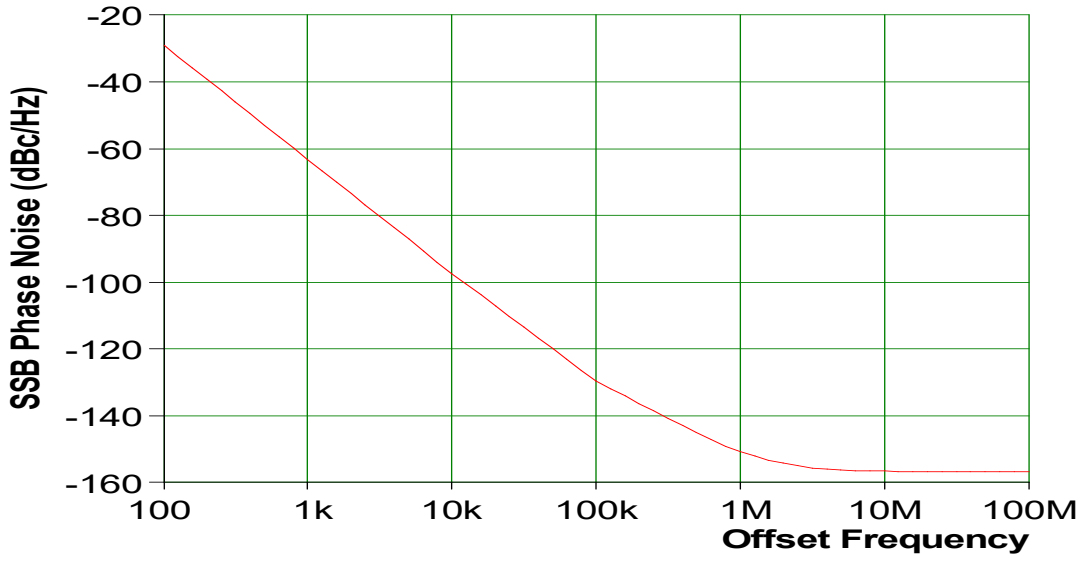


**Figure 5.2:** phase noise performance of integrated VCO

### 5.2.3 Loop Filter Design

#### 5.2.3.1 Loop Bandwidth Optimization

The phase locked loop design was simulated for 2.5 kilohertz, 5 kilohertz, 10 kilohertz, 20 kilohertz, 30 kilohertz, 40 kilohertz, and 50 kilohertz bandwidths at a 100 kHz offset frequency, as shown in Table 5.3. The phase noise and lock time of the each loop bandwidth were obtained and recorded as shown in Table 5.3. The narrow bandwidth had low phase noise but a long lock time. Wider loop bandwidths reduced the lock time with the expense of increased phase noise. An optimum value of loop bandwidth was chosen as 10 kHz because the lock time was less than 1 ms, which agreed with our specification, and the phase noise improved by more than 8 dBc/Hz compared to that of the 20 kHz phase margin.

**Table 5.3:** Simulated Phase Noise and Lock Time for various Loop Bandwidths at 100 kHz offset frequency

| Loop Bandwidth (kHz) | Phase Noise (dBc/Hz) | Lock time($\mu$s) |
|---|---|---|
| 2.5 | -129.53 | 2740 |
| 5 | -129.02 | 1410 |
| 10 | -125.96 | 753 |
| 20 | -117.15 | 426 |
| 30 | -110.59 | 317 |
| 40 | -105.97 | 263 |
| 50 | -102.62 | 230 |

Simulations were done for the same loop bandwidths at 100 hertz, 1 kilohertz, 10 kilohertz, 100 kilohertz and 1 megahertz offset frequencies, and their phase noise characteristics are shown in Figure 5.3. It was discovered that the phase noise in a bandwidth of 2.5 kHz and 5 kHz had a very high value when the offset frequency was low, but a very low value when the offset frequency was large. The phase noise was not very noticeable at low offset frequencies for bandwidths greater than 10 kHz, but it

became more noticeable at higher offset frequencies.

The 10 kHz bandwidth was found to be optimum because it had the best phase noise characteristics compared to the rest. This is because it had low phase noise at low offset frequencies in comparison to bandwidths of 2.5 kHz and 5 kHz, and it had low phase noise at higher offset frequencies in comparison to bandwidths of 20 kHz, 30 kHz, 40 kHz, and 50 kHz, as shown in Figure 5.3. Another reason for this is that it had low phase noise at higher offset frequencies. The phase noise of 10 kHz bandwidth at 1 MHz offset frequency was -150.80 dBc/Hz and was much lower compared to the phase noise of -82dBc/Hz at an offset frequency of 1MHz obtained by Kameche and Feham (2013).

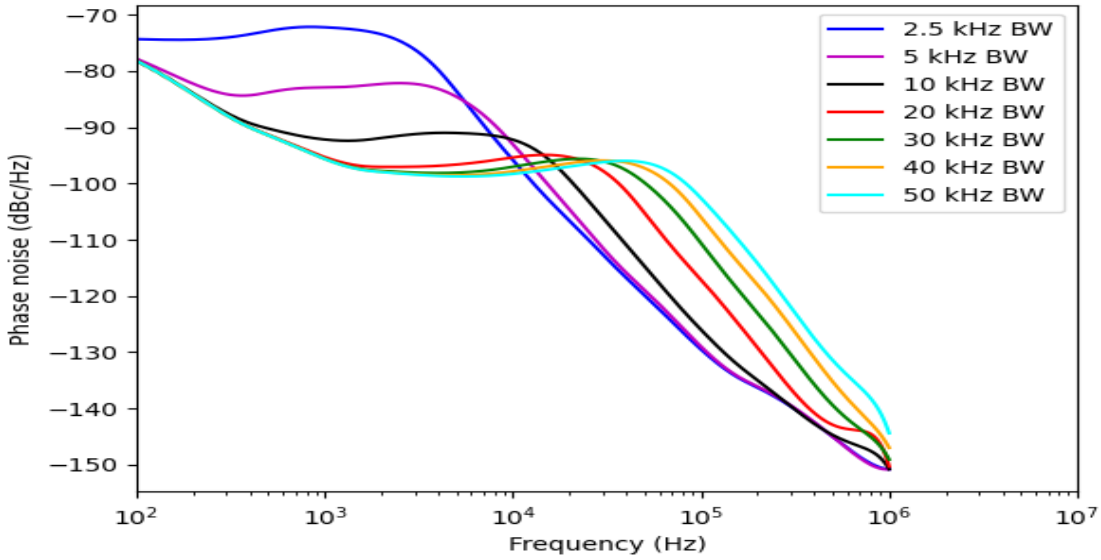


**Figure 5.3:** Phase noise simulation for different loop bandwidths

#### 5.2.3.2 Phase Margin Optimization

The phase noise obtained for 35°, 40°, 45°, 50°, 55° and 60° with a 5° phase margin values was -127.25 dBc/Hz, -126.69 dBc/Hz, -125.96 dBc/Hz, -124.95 dBc/Hz, -123.64 dBc/Hz and -121.66 dBc/Hz respectively as summarized in Table 5.4. The phase noise of the optimum phase margin was -125.96 dBc/Hz, and it was low compared to the phase noise obtained by Shurender *et al.* (2013) of -112.4 dBc/Hz at the same phase

margin.

**Table 5.4:** Simulated Phase Noise for various phase margin

| Phase Margin (degrees) | Phase Noise (dBc/Hz) |
|---|---|
| 35 | -127.25 |
| 40 | -126.69 |
| 45 | -125.96 |
| 50 | -124.95 |
| 55 | -123.64 |
| 60 | -121.66 |

Figure 5.4 illustrates the fluctuation of the phase noise with regard to the phase margin. As can be seen in figure 5.4, it was noticed that there was a correlation between an increase in phase margin and an increase in the phase noise. As a result, the optimal value for phase margin was decided to be 45 degrees since, below this phase, the loop became unstable, and when compared to that of the 50° phase, the improvement in phase noise was greater than 1 decibel per hertz.

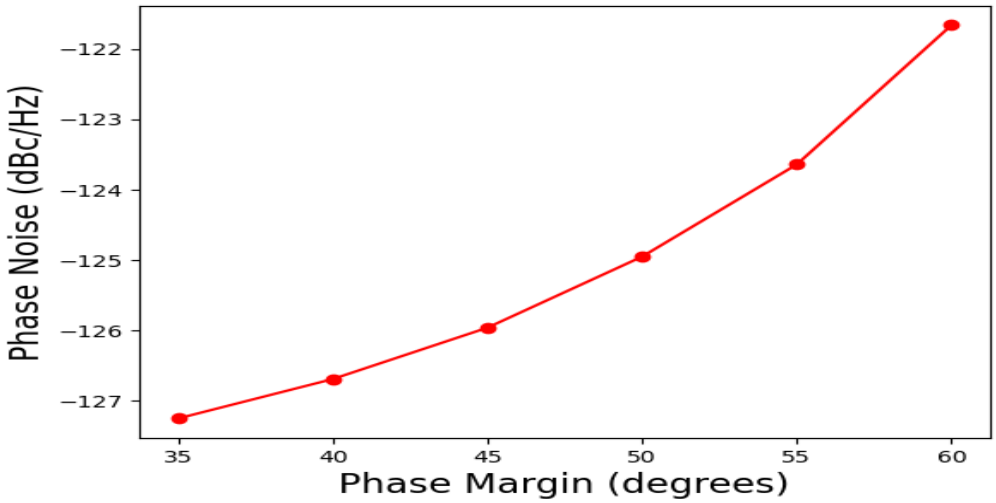


**Figure 5.4:** Phase noise variation with respect to the phase margin

After running a simulation, the optimal values for the loop filter's resistors, capacitors, phase margin, and bandwidth are presented in Table 5.5 below. These values were

determined by the results of the simulation. These values were converted to the standard values that are available in the market by the simulation software.

**Table 5.5:** Components of the loop filter, both optimized and standard,

| Filter Components | Optimized Value | Standard Value |
|---|---|---|
| Loop Bandwidth | 10 kHz | 10 kHz |
| Phase Margin | 45° | 47° |
| C1 | 291 pF | 270 pF |
| R1 | 11.3 kΩ | 11 kΩ |
| C2 | 3.96 nF | 3.90 nF |
| R2 | 23 kΩ | 22 kΩ |
| C3 | 133 pF | 120 pF |

### 5.2.4 System Behaviour

Figure 5.5 presents a diagrammatic representation of the simulated PLL system. The control input of the voltage controlled oscillator is connected to the loop filter, which in turn is connected to the charge pump output pin. In order to establish the charge pump output current, a resistor with a value of 5.1 kΩ is connected between pin 22 and ground. The reference frequency is also linked to the reference input port of the charge pump.

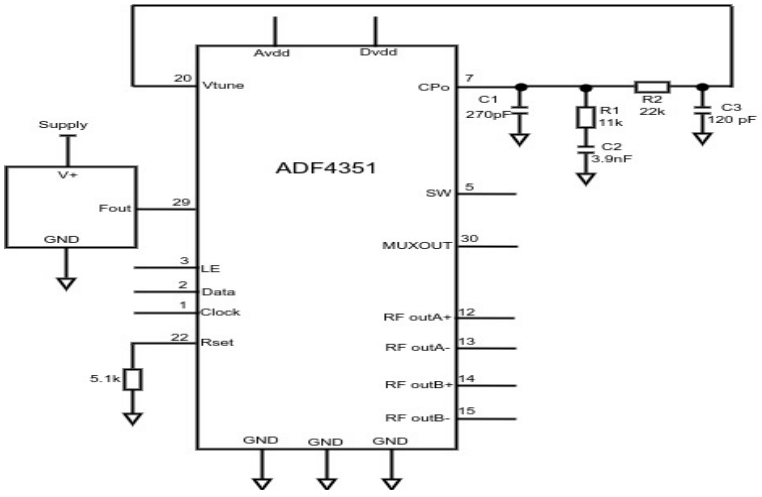


**Figure 5.5:** Phase Locked Loop Schematic using ADIsimPLL

The bandwidth and phase margin of the design of the phase locked loop may be seen in Figure 5.6, which displays the open-loop gain plot. At the selected loop bandwidth of 10 kHz (the crossover frequency), the gain decreases to nearly 1 (0 dB), as shown in Figure 5.6, and the phase reaches its maximum value at -135.00°. This is the point when the gain is at its lowest. This phase corresponds to the phase margin of 45° (180° - 135°) which is the selected phase margin. This means that the PLL system is stable because the magnitude of the open loop gain drops to 0 dB before the phase has reached 180°.

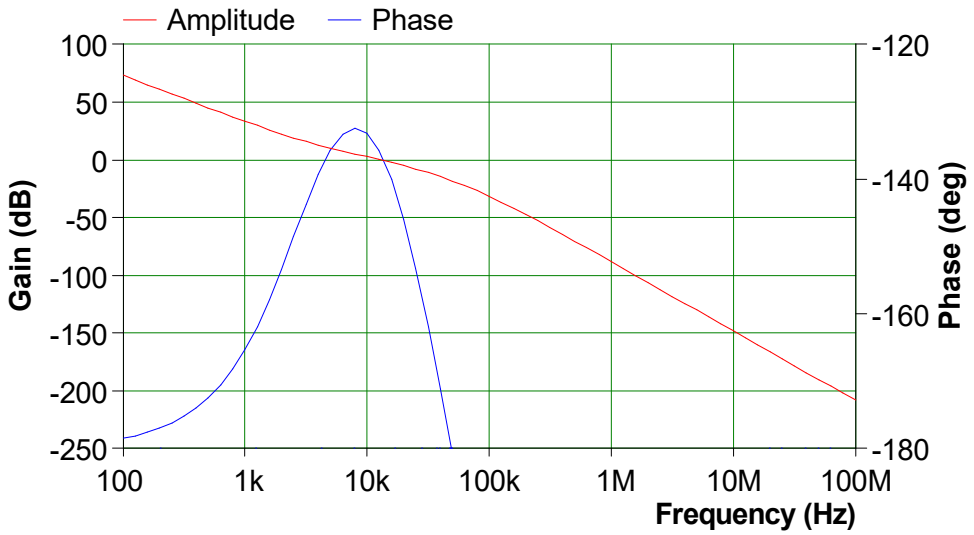


**Figure 5.6:** Open loop transfer function at 387 MHz

Figure 5.7 depicts the closed loop gain transfer functions as well as the phase transfer functions. It was discovered that the transfer function in the closed loop had a sizable amplitude inside the loop bandwidth of the synthesizer. This was one of the discoveries made. This resulted in an increase in the noise that was coming from the reference, dividers, PFD, and charge pump as a result of the fact that it was multiplied by the closed loop transfer function and reflected to the output. The gain remained constant, as shown in figure 5.7, until it reached the loop bandwidth, after which it dropped rapidly. Consequently, the gain plot for the closed loop may be utilized to trace the phase noise attenuation of each component's respective contribution. Outside of the loop bandwidth, the phase noise of the VCO was able to have an effect.

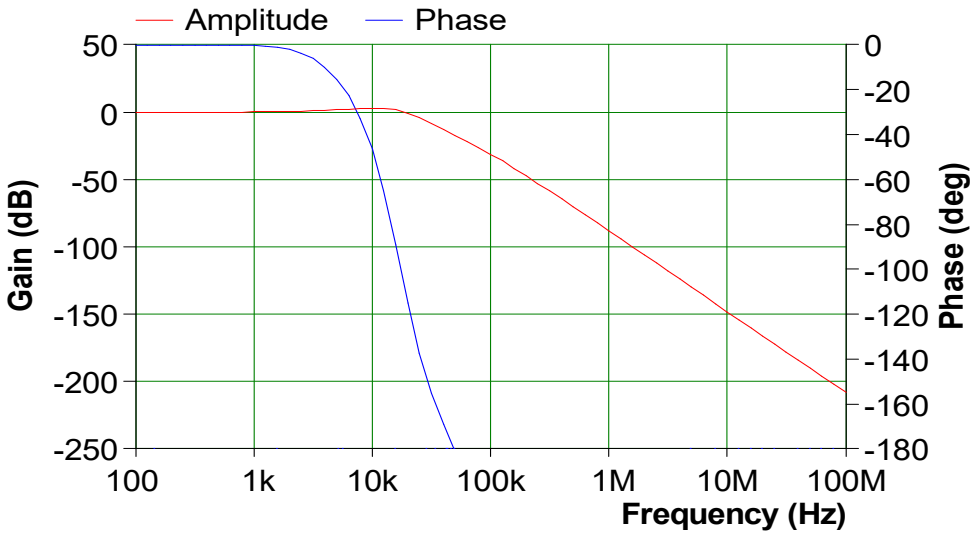


**Figure 5.7:** The closed loop transfer function at 387 MHz

### 5.2.5 Output Phase Noise

The overall noise response of each component of the PLL, including the reference, chip, filter, and VCO, is detailed in Table 5.6, along with the phase noise that each component of the PLL contributes. The total phase noise for 100 hertz, 1 kiloherz, 10 kiloherz, 100 kiloherz and 1 megahertz offset frequencies was obtained as -78.18 dBc/Hz, -92.17 dBc/Hz, -92.20 dBc/Hz, -126.00 dBc/Hz, and -150.80 dBc/Hz, respectively, as shown in Table 5.6.

**Table 5.6:** The Phase noise simulation results (all findings are in dBc/Hz)

| Frequency (Hz) | Reference | Chip | Filter | VCO | Total |
|---|---|---|---|---|---|
| $1 \times 10^2$ | -78.24 | -98.45 | -133.50 | -100.70 | **-78.18** |
| $1 \times 10^3$ | -98.03 | -98.95 | -113.60 | -94.98 | **-92.17** |
| $1 \times 10^4$ | -115.60 | -96.65 | -100.30 | -95.37 | **-92.20** |
| $1 \times 10^5$ | -168.80 | -132.80 | -130.50 | -129.50 | **-126.00** |
| $1 \times 10^6$ | -228.70 | -189.70 | -168.80 | -150.80 | **-150.80** |

The plot of the simulated phase noise of each component and the total phase noise is shown in Figure 5.8. The plot shows that the phase noise of the reference dominated the

characteristics at the earlier stages of the loop bandwidth, up to about 1 kHz. Because the gain of the closed loop transfer function was high in that region, the phase noise of the chip and loop filter was effective as expected in the region between this frequency (1 kHz) and the loop bandwidth. This was due to the fact that the loop bandwidth was in that region. A flattening out of the slope may be seen at the loop bandwidth. This is because the phase noise addition of the reference was fairly low, whereas the phase noise component of the chip, filter, and VCO started to drop at the loop bandwidth.

Beyond the loop bandwidth, it was seen that the phase noise of the VCO became the dominant factor, which was to be expected, while the phase noise of other components continued to drop continually. This was the case even though the loop bandwidth remained unchanged. When the frequency was greater than 100 kHz, the overall response replicated the VCO's phase noise.

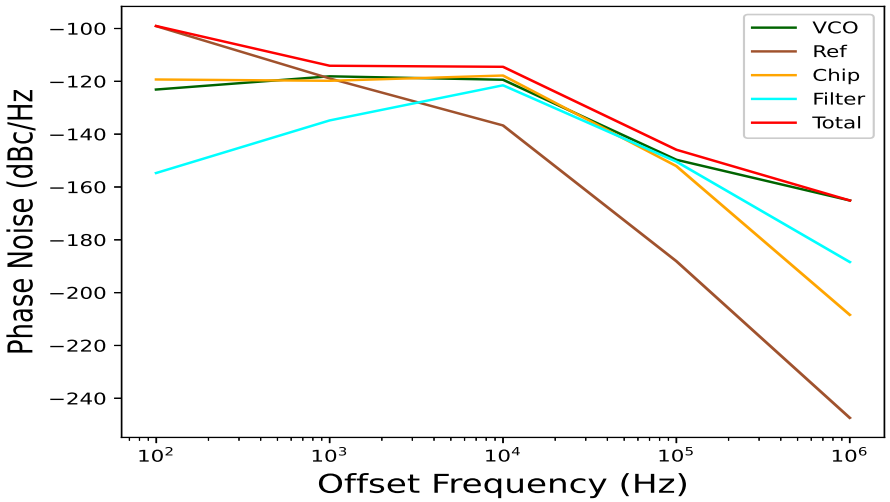


**Figure 5.8:** The contribution of phase noise from system components and the overall response at 387 MHz

Figure 5.9 illustrates the reference spurs that emerged as a consequence of the non-linear behavior of the phase frequency detector located in the signal path. It can be seen from the plot that the maximum spurs was of -300 dBc which is much lower compared to that of -82 dBc obtained by Kameche and Feham (2013). The first spur that occurred was a phase detector spur of -300 dBc. This is because it appeared at an offset frequency, the

same as the phase detector frequency. The other spurs were modulated spurs and they were of -300 dBc.

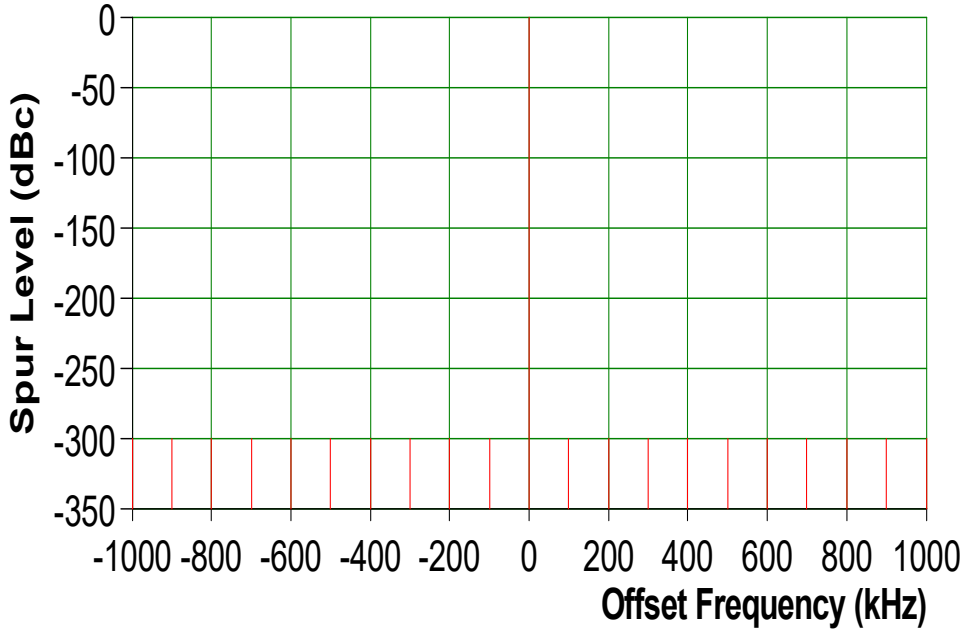


**Figure 5.9:** Reference spurs at 387 MHz

### 5.2.6 Transient Response of Phase Locked Loop

As can be seen in Figure 5.10, a simulation of the transient response was carried out in order to demonstrate how the PLL's output frequency varies. The PLL switched from 35 MHz-3 GHz and the lock time taken by the system to settle at 3 GHz was approximately 753 $\mu$s. The locking time obtained was lower compared to 2.869 ms obtained by HOSSEINI and Masoumi (2017) and agrees with the one obtained by Shurender *et al.* (2013) of 119.5 $\mu$s. The frequency first increased until it reached the peak frequency of about 3.18 GHz, and the absolute value of overshoot was 0.18 GHz. The rise time was approximately 0 seconds, while the peak time was approximately 9.27 $\mu$s. After reaching the peak frequency, there was damping and repeated overshoot and undershoot (ringing) until the system reached a final frequency.

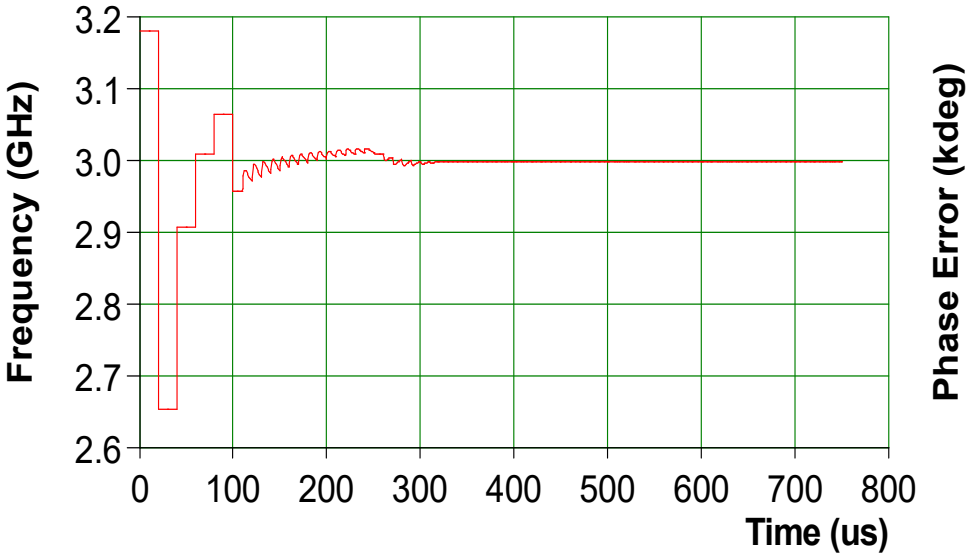


**Figure 5.10:** Transient frequency characteristics of simulated PLL

### 5.2.7 Checking the Stability of the PLL system Using Routh's Stability Criterion

The values of the coefficients were computed with the assistance of a Python script, which, together with the specifics of the computations, is provided in the appendix. These values are;

A0 = 4.2900000000000001e-09

A1 =2.2908600000000003e-14

A2 = 3.057912e-20

T2 = 4.29e-05

K = 31.415926535897935

The values of the Routh array were computed based on the information provided in Table 5.7, and the first column of the Routh array contained values that were positive. As a direct consequence of this, the phase-locked loop system was stable, and the left side of the s-plane contained all of the poles of the PLL closed-loop transfer function.

**Table 5.7:** Phase-locked loop stability according to Routh's criteria

| | | | |
|---|---|---|---|
| $s^4$ | $3.06 \times 10^{-20}$ | $4.29 \times 10^{-9}$ | 31.42 |
| $s^3$ | $2.29 \times 10^{-14}$ | $1.35 \times 10^{-3}$ | 0 |
| $s^2$ | $4.29 \times 10^{-9}$ | 31.42 | 0 |
| $s^1$ | $1.06 \times 10^{-3}$ | 0 | 0 |
| $s^0$ | 31.42 | 0 | 0 |

## 5.3 Implementation of the signal Generator

After simulation, the signal generator circuit was designed using KiCad software. The circuits designed were for power supply, the control of the synthesizer registers and keypad shield by the microcontroller.

### 5.3.1 Power supply design

Figure 5.11 provides a representation of the circuit diagram for the power module construction. LM7805 voltage regulator was used in regulating the 15 V input to a 5 V output, which was used to power the microcontroller. LD1117av33 voltage regulator converted the 5 V from LM7805 to 3.3 V, which was used to power the ADF4351 chip.

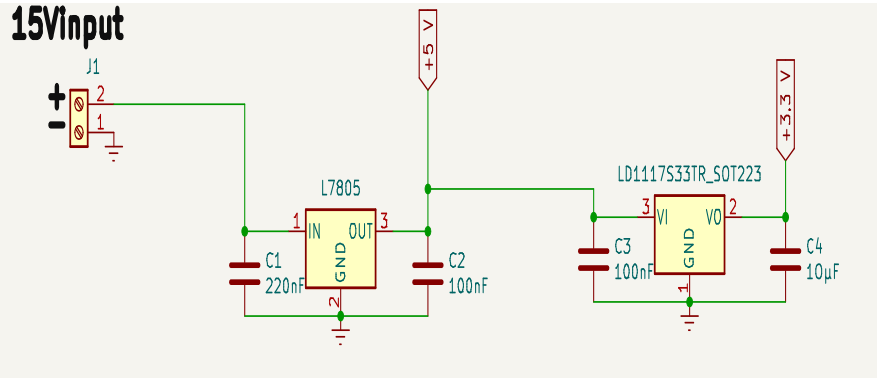


**Figure 5.11:** Circuit diagram of the power section design

### 5.3.2 Synthesizer Control Design

Figure 5.12 shows the synthesizer and how it was controlled by an Atmega 328P microcontroller. The 10 MHz reference frequency was fed to the ADF4351 synthesizer by capacitive coupling. The charge pump's output was connected to a third order passive filter with a 10 kHz bandwidth and a 45° phase margin. After that, a connection was made between the output of the filter and the input of the VTUNE. The Serial Peripheral Interface (SPI) module of the Atmega 328P was used to set the registers of the synthesizer. Because the microcontroller operates at 5 V, a voltage divider was constructed using 560 Ω and 1 kΩ to regulate the voltage to 3.3 V for the ADF4351's CLK, DATA, and load enable inputs.

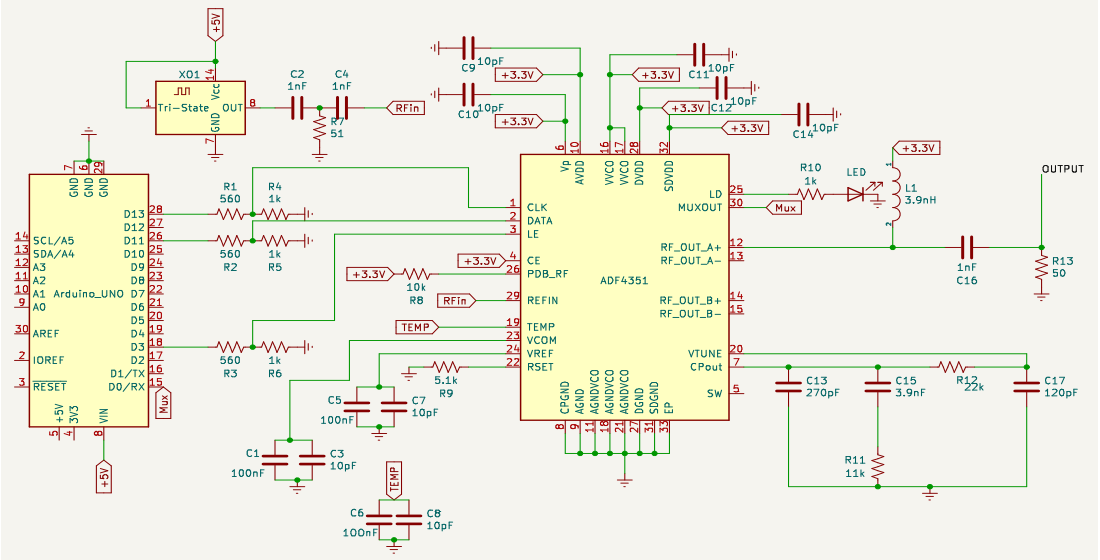


**Figure 5.12:** Circuit diagram of the PLL chip control design

### 5.3.3 Keypad Shield Control

As can be seen in Figure 5.13, this shield features a 2x16 LCD display in addition to six momentary press buttons labeled select, up, right, down, and left, as well as a reset button. The Arduino board is interfaced with the LCD through pins 4, 5, 6, 7, 8, 9 and 10 as shown in figure 5.13. Because all of the controls are connected to analog Pin 0, an ADC was utilized in order to read their values, and a potentiometer was used in order to adjust the level of contrast displayed on the LCD.

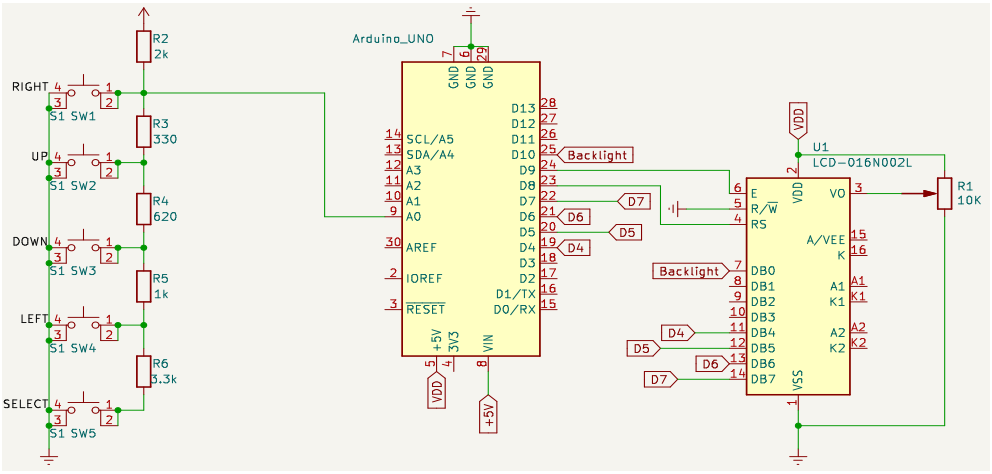


**Figure 5.13:** Circuit diagram of the keypad shield controlled by arduino

## 5.4 Tests and Measurements

### 5.4.1 Testing the Signal Generator

The testing of the signal generator was done with a GSP-830 spectrum analyzer and Cathode Ray Oscilloscope (CRO). The output spectrum of the signal generator is depicted in Figure 5.14, and the signal generator had a minimal frequency of 35 MHz. This signal's output power was measured to be -12.8 dBm.

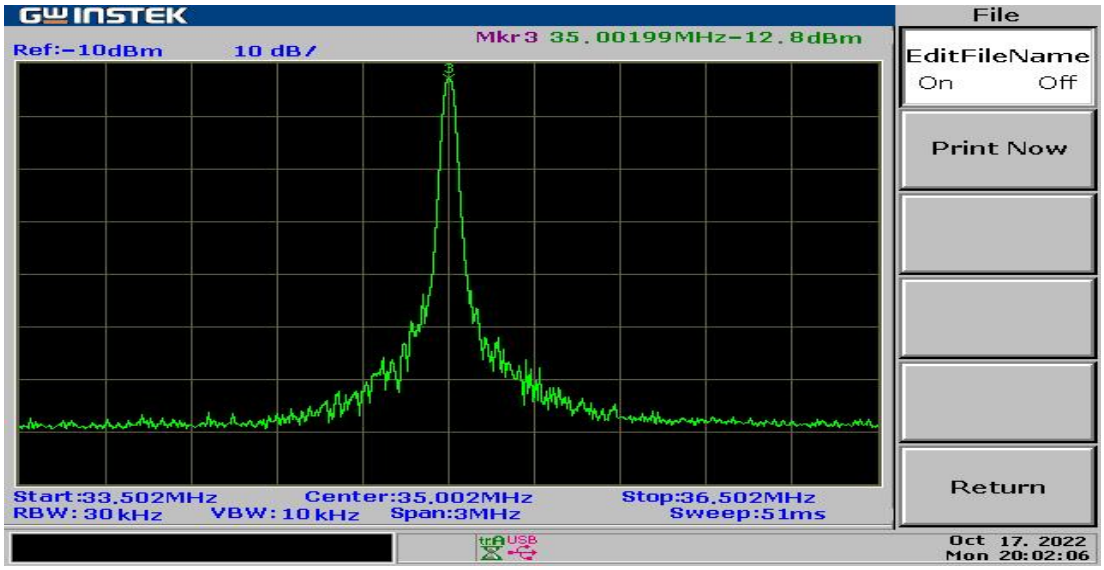


**Figure 5.14:** Output spectrum at 35 MHz frequency on a spectrum Analyser

The testing of the signal was also done at 35 MHz frequency on a CRO. As can be seen

in Figure 5.15, a sinusoidal waveform was observed, and its output frequency was 34.97 MHz, while its magnitude was 272 mV.

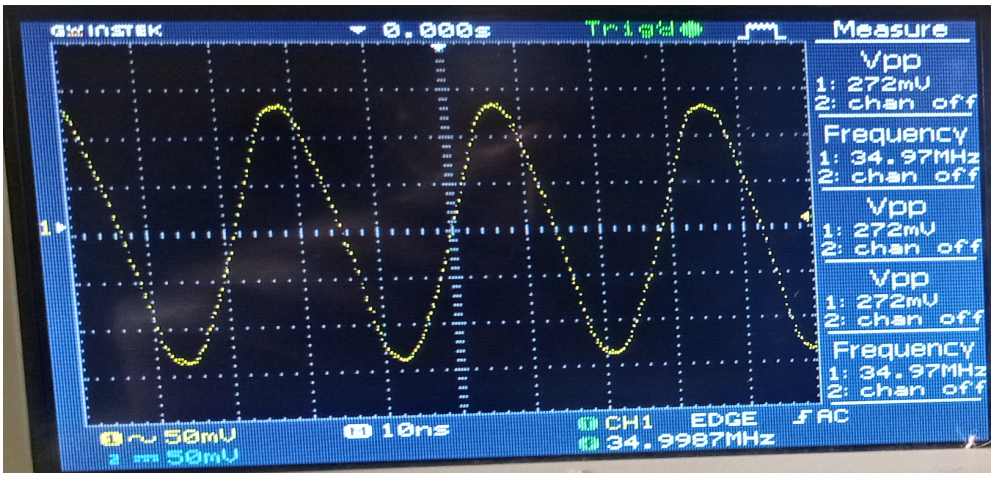


**Figure 5.15:** Signal output from a CRO at the frequency of 35 MHz

The signal generator was also evaluated on a spectrum analyzer operating at 90 MHz, and the results are presented in Figure 5.16 as the output spectrum. The level of the output power was observed at -15.9 dBm.

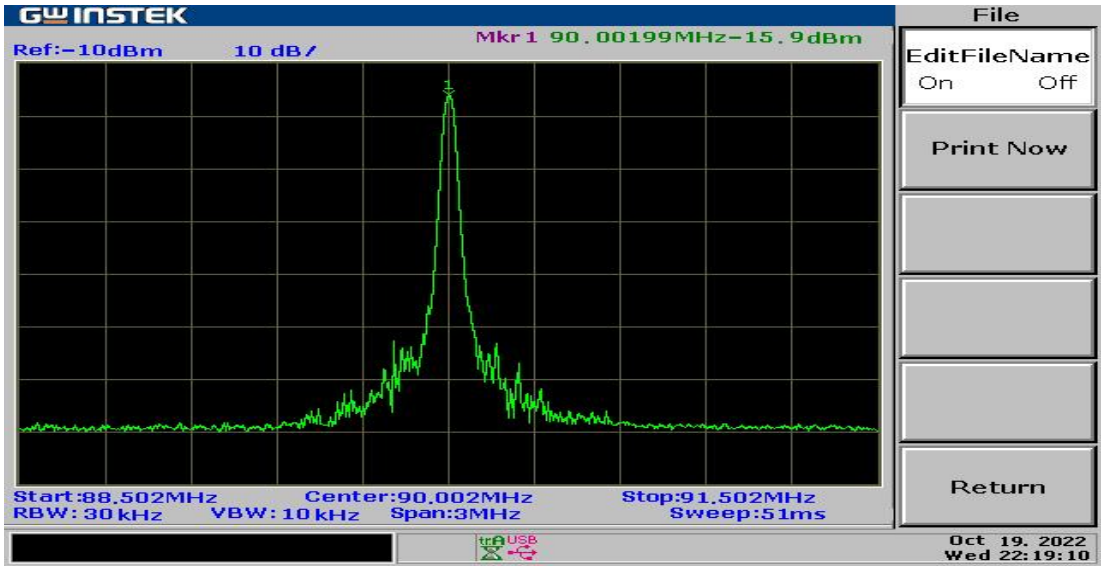


**Figure 5.16:** Output spectrum at 90 MHz frequency on a spectrum Analyser

The testing of the signal for 90 MHz output frequency was also done on a CRO. As can be seen in Figure 5.17, a sinusoidal waveform was observed, and its amplitude measured 1.28 V from peak to peak.

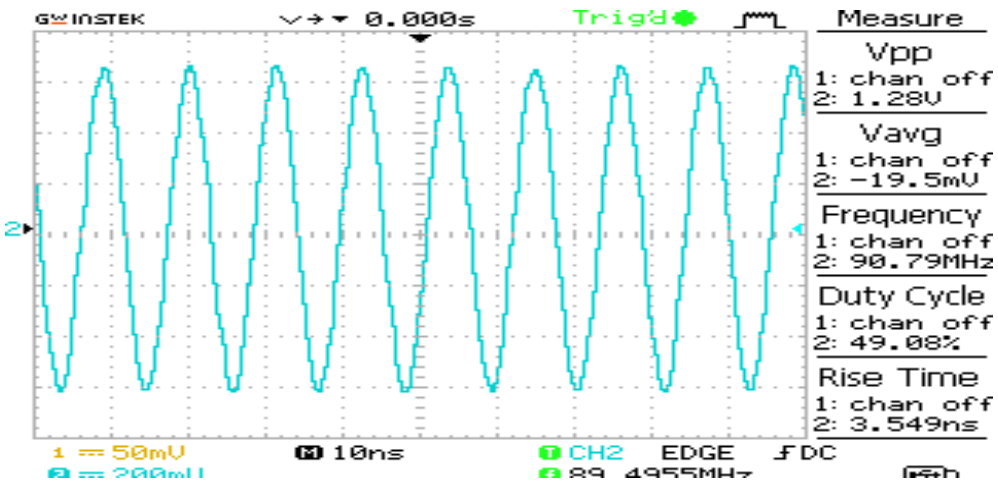


**Figure 5.17:** Output signal at 90 MHz frequency on a CRO

The testing was also done at 1734.5 MHZ output frequency. An output spectrum shown in Figure 5.18 with an output power of -44.3 dBm was observed. The output signal was not tested at this frequency because the CRO has an output range of 0 – 100 MHz.

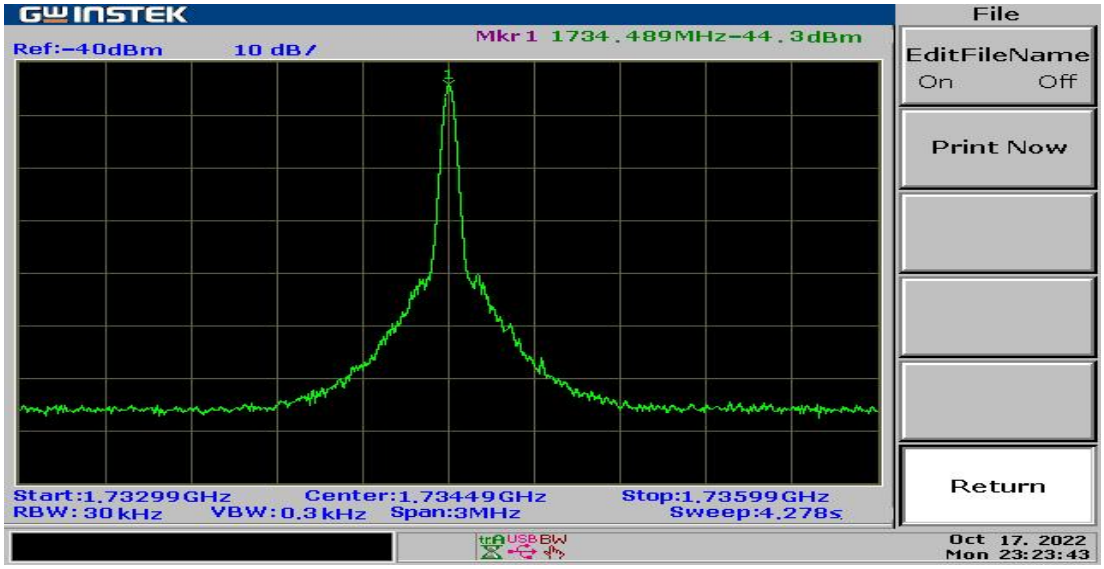


**Figure 5.18:** Output spectrum at 1734.5 MHz frequency on a spectrum Analyser

The maximum frequency tested in the spectrum analyser was of 2900 MHz and its output spectrum is shown in Figure 5.19 with an output power of -58.4 dBm.

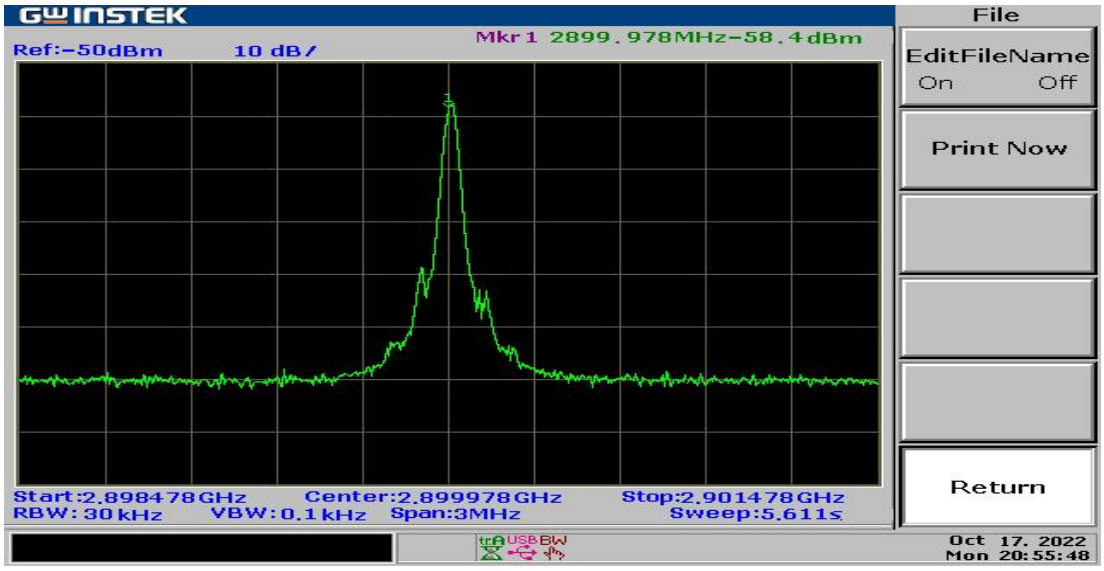


**Figure 5.19:** Output spectrum at 2900 MHz frequency on a spectrum Analyser

### 5.4.2 Phase Noise Measurements

The GSP-830 spectrum analyzer was also utilized in the conduct of the phase noise measurements. We chose the least resolution bandwidth of the GSP-830 spectrum analyzer, which is 3 kHz, to cut down on the amount of noise the analyzer was contributing. The experimental phase noise was calculated for 35.00 MHz, 387.00 MHz, 1.00 GHz, 2.00 GHz, and 2.90 GHz frequencies at 1 kHz, 10 kHz, 100 kHz and 1 MHz offset frequencies as shown in Table 5.8. The phase noise values obtained after simulation for 35.00 MHz, 387.00MHz, 1.00 GHz, 2.00 GHz and 2.90 GHz frequencies at 1 kHz, 10 kHz, 100 kHz, and 1 MHz offset frequencies are also summarized in Table 5.8.

The experimental phase noise seems to agree with the one obtained by Handique and Bezboruah (2015) where they reported a phase noise of below 100 dBc/Hz at an offset frequency of 1 MHz in their study about design, fabrication and analysis of 1.1 GHz PLL synhesizer. HOSSEINI and Masoumi (2017) obtained a phase noise of below 100 dBc/Hz at an offset frequency of 100 kHz, agreeing with our results as in Table 5.8.

**Table 5.8:** Experimental and Simulated Phase noise at 35.00 MHz, 387.00 MHz,1 GHz, 2 GHz and 2.9 GHz

| **Offset Freq (kHz)** | **Experimental Phase Noise (dBc/Hz)** | | | | | **Simulated Phase Noise (dBc/Hz)** | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 35 MHz | 387 MHz | 1 GHz | 2 GHz | 2.9 GHz | 35 MHz | 387 MHz | 1 GHz | 2 GHz | 2.9 GHz |
| 1 | - 94.2 ± 1.2 | -86.1 ± 0.5 | - 76.5 ± 0.5 | -72.1 ± 2.5 | - 71.2 ± 1.0 | - 114.1 | - 92.1 | - 79.5 | - 73.5 | - 74.6 |
| 10 | - 95.7 ± 2.0 | - 90.3 ± 1.5 | - 82.7 ± 1.0 | - 77.9 ± 1.5 | - 74.8 ± 2.0 | - 114.5 | - 92.2 | - 84.2 | - 78.2 | - 75.4 |
| 100 | - 106.4 ± 1.5 | - 104.2 ± 2.0 | - 102.6 ± 1.0 | - 102.2 ± 1.0 | - 100.7 ± 2.6 | - 145.9 | - 126.0 | - 119.1 | - 113.0 | - 108.4 |
| 1000 | - 115.7 ± 3.0 | - 112.8 ± 1.0 | - 112.1 ± 0.5 | - 111.1 ± 1.0 | - 110.7 ± 1.5 | - 165.1 | - 150.8 | - 143.2 | - 137.5 | - 133.9 |

A comparison was made between the phase noise measured experimentally at 35 MHz and the simulated phase noise, as shown in Figure 5.20. It is clear from looking at the graph that the experimental phase noise was far higher than the phase noise that was generated by the simulation. This is due to the fact that the phase noise contribution in the simulation was only from the phase locked loop components (Loop filter, PLL chip, reference oscillator, and the VCO), whereas in the experiment, in addition to the phase noise from the PLL components, there were other sources of phase noise while carrying out the experiment. These sources include the spectrum analyser, power source, microcontroller and connecting wires.

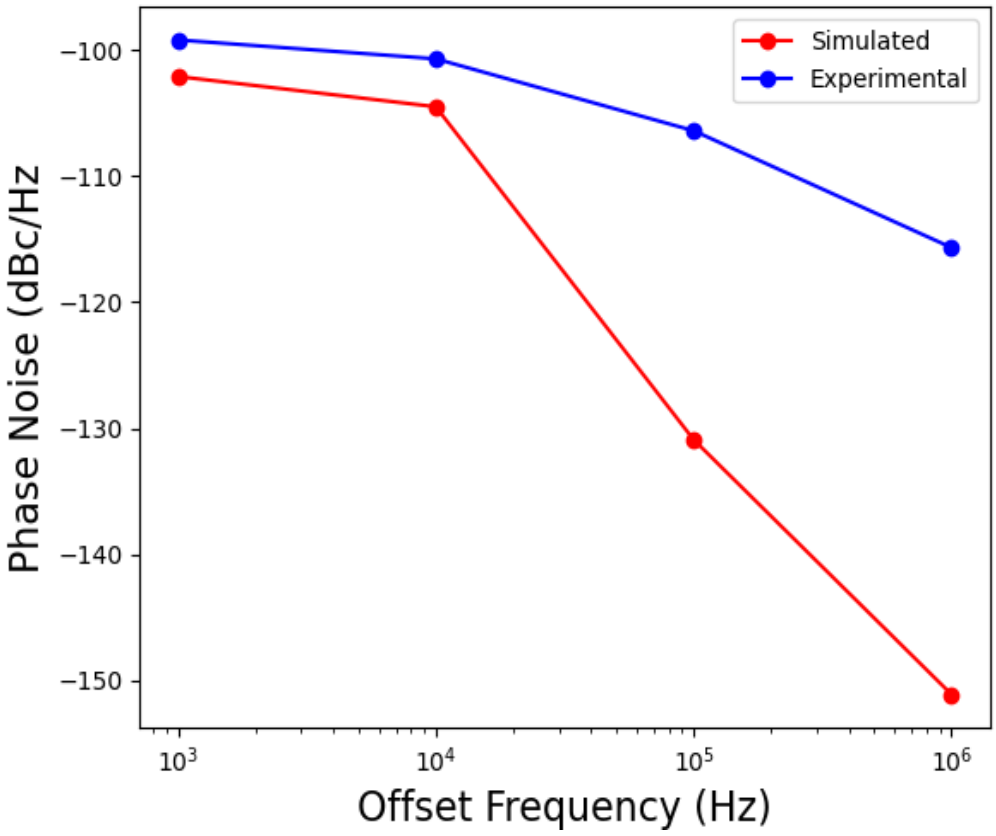


**Figure 5.20:** A comparison of the phase noise of simulated and experimental data at 35 MHz

The phase noise comparison was also done at 387.00 MHz, 1 GHz, 2 GHz and 2.9 GHz, as shown in Figure 5.21. The experimental phase noise was higher than the simulated phase noise. The experimental and simulated phase noise were almost the same at 1 kHz and 10 kHz offset frequencies. This is because the simulated phase noise increased in this region as it was within the selected loop bandwidth of 10 kHz during simulations, and most of the phase noise from the PLL components were dominant in this region.

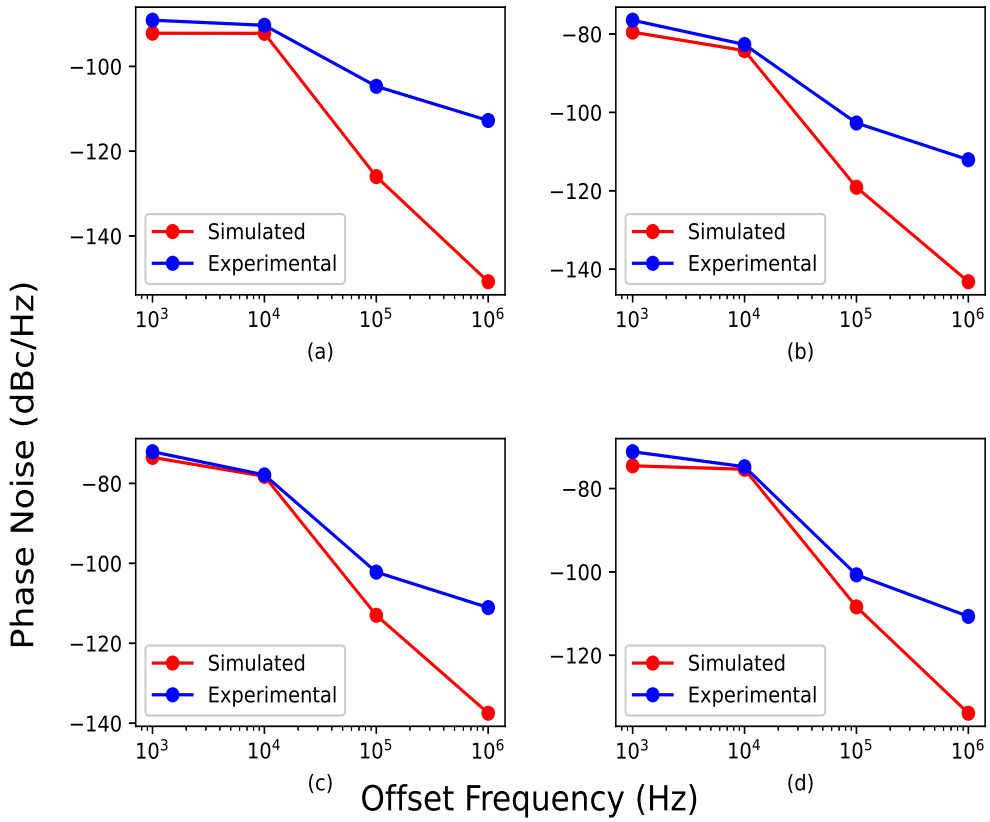


**Figure 5.21:** A comparison of the phase noise of simulated and experimental data at (a) 387 MHz, (b) 1 GHz, (c) 2 GHz and (d) 2.9 GHz

The experimental phase noise was seen to increase with an increase in the output frequency, as shown in Figure 5.22. The same observation was also made by Yang *et al.* (2012) in their design of Ultra-broadband, High Resolution Frequency Synthesizer. This is due to the fact that the signal generator was built with the concept of a phase locked loop. The phase-locked loop implements the idea of frequency multiplication by dividing along the feedback loop with the use of a counter. This allows the generated frequencies to have a wide range and a high frequency. Therefore, as the frequency was multiplied to generate higher frequencies, the phase noise was raised by 20 multiplied by the logarithm of the number of counter, as explained in the introduction chapter.

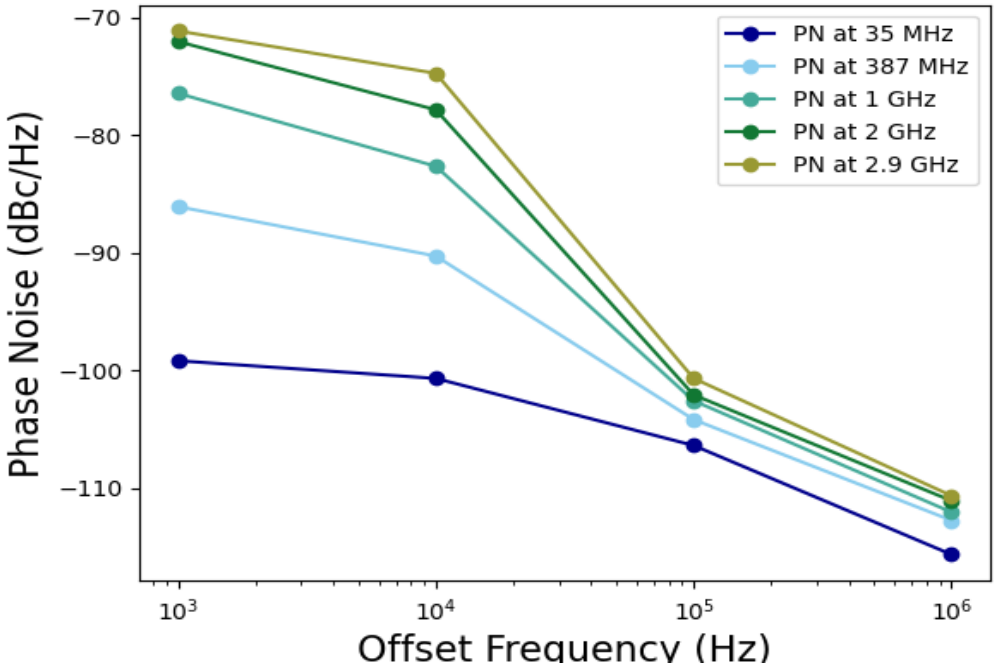


**Figure 5.22:** Phase Noise at 35 MHz, 387 MHz, 1 GHz, 2 GHz and 2.9 GHz output frequencies

### 5.4.3 Spurs Measurements

Most of the spurs observed at 387.00 MHz output frequency were harmonics, as shown in Figure 5.23. At full span, we were able to observe up to the fifth harmonic. The levels of the first, second, third, fourth and fifth harmonics were -33.1 dBm, -75.7 dBm, -51.7 dBm, -77.6 dBm, -66.3 dBm and -77.2 dBm, respectively, as summarized in Table 5.9. Their respective values with respect to the carrier were also calculated and are summarized in column 3 of Table 5.9.

Other spurs were observed at the 387.00 MHz output frequency, as shown in Figure 5.23. These spurs appeared at 50 MHz, 100 MHz, 600 MHz, 940 MHz, 2320 MHz and 2700 MHz frequencies, and after calculation, their level with respect to the carrier was found to be -40.2 dBc, -41.9 dBc, -38.5 dBc, -43.3 dBc, -44.1 dBc and -37.4 dBc, respectively, as shown in Table 5.9.

**Table 5.9:** Level of spurs at 387.00MHz output frequency

| Frequency (MHz) | Level (dBm) | Level (dBc) |
|---|---|---|
| 50 | -73.3 | -40.2 |
| 100 | -75.0 | -41.9 |
| 387 | -33.1 | 0 |
| 600 | -71.6 | -38.5 |
| 774 | -75.7 | -42.6 |
| 940 | -76.4 | -43.3 |
| 1161 | -51.7 | -18.6 |
| 1548 | -77.6 | -44.5 |
| 1935 | -66.3 | -33.2 |
| 2320 | -77.2 | -44.1 |
| 2700 | -70.5 | -37.4 |

These spurs could be contributed by the reference frequency oscillator because they appeared at multiples of input reference frequency of 10 MHz. Another possible cause could be the channel spacing used while designing the PLL of 100 kHz. This is because these spurs also appeared at multiples of the channel spacing.

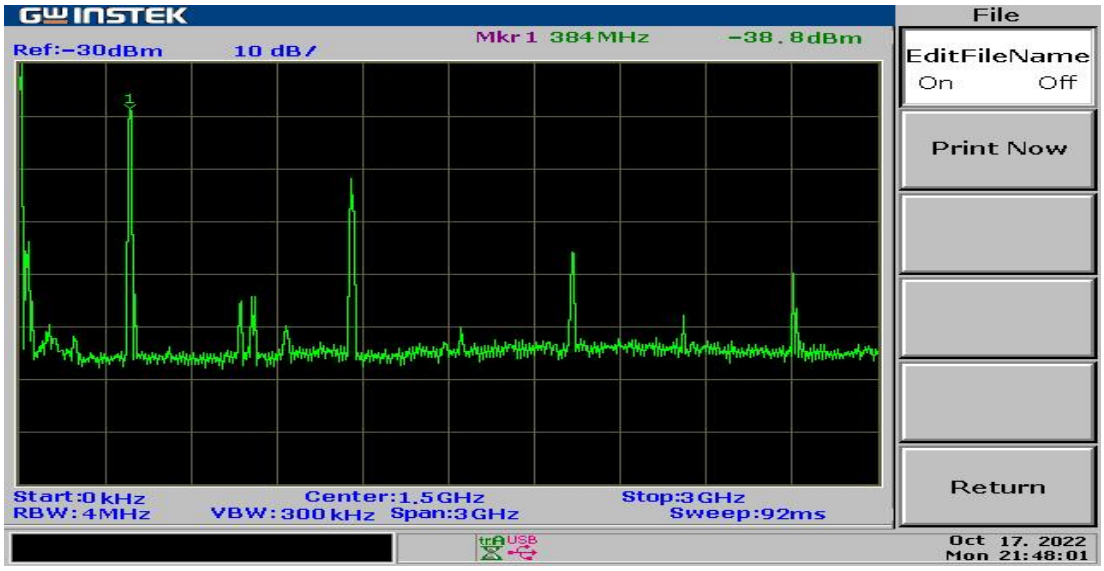


**Figure 5.23:** Level of spurs at 387.00 MHz output frequency

# CHAPTER SIX

# CONCLUSIONS AND RECOMMENDATIONS

## 6.1 Introduction

The designed signal generator produced stable output frequencies and exhibited good phase noise performance across a broad spectrum of frequencies. This chapter summarizes the most important takeaways from my research. Also, it draws attention to the areas in which we believe additional investigation is warranted to enhance the performance of the signal generator.

## 6.2 Conclusions

This research was conducted to design and construct a signal generator based on a phase-locked loop capable of generating frequencies in the range of 35 MHz–3 GHz with a low phase noise level. The simulations of the signal generator were done using ADIsimPLL design software. In the simulation, the phase locked loop chip selected was ADF4351 from analog devices because it can generate frequencies in the required range. The reference frequency of 10 MHz was used in the simulation. The phase locked was simulated at different values of loop bandwidth, and an optimum value of 10 kHz was obtained.

The simulations were also run with other values of phase margin, and after examining the phase noise and lock time that the results produced, the optimal value of 45° was determined to be used.This work examines the noise produced by each phase-locked loop component and presents the results of that analysis along with their impacts. The output frequency at 387 MHz had a phase noise of -126 dBc/Hz at a 100 kHz offset and a locking time of 753 $\mu$s for a 2065 MHz frequency jump.

The standard values of a third order filter components at 10 kHz loop bandwidth and 45° phase margin were obtained as 270 PF, 3.9 nF, 120 pF, 11 k$\Omega$ and 22 k$\Omega$. A phase locked

loop synthesizer was built using the components designed by simulation. The signal generator was built by integrating the phase locked loop synthesizer and a keypad shield with Arduino UNO microcontroller. This microcontroller was used to program and control the ADF4351 chip via Serial Peripheral Interface (SPI) to enable the changing of frequencies using a keypad shield.

The nature of the signal produced by the generator was tested using a cathode ray oscilloscope for the 35–100 MHz frequency range, and we observed a sinusoidal wave whose output power changed as the output frequency was varied. The spectrum analyser was used for testing frequencies higher than 100 MHz since the maximum output of the CRO was 100 MHz. Similarly, the output power changed as the frequency increased. The level of phase noise obtained after calculation was higher than the level obtained after simulation, and the phase noise also increased with an increase in output frequency. The maximum spur appeared at the third harmonic and was found to be -18.6 dBc, while the minimum spur appeared at the fourth harmonic and was found to be -44.5 dBc.

### 6.3 Recommendations

The signal generator designed in this work can be applied in electronics lab for testing RF components such as Wifi, GPS and wireless communication systems, during designing and debugging. This signal generator can be used in characterizing microstrip antennas in the lab.

The following areas are suggested for further research:

i. A module can be included in the signal generator design to enable the selection of variable peak to peak amplitude of the output frequency.

ii. The signal generator can be modified to broaden the output frequency range in order to expand the generator's range of potential applications.

## REFERENCES


Arduino (n.d). Arduino uno r3-datasheet.pdf. https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf. (Accessed on 10/02/2022).

Atmel (2015). Atmega328p datasheet. https://datasheet.octopart.com/ATMEGA328P-AUR-Atmel-datasheet-141421474.pdf. (Accessed on 10/02/2022).

Banerjee, D. (2017). *Pll Performance, Simulation, and Design 5th Edition*. Dog Ear Publishing.

Barret, C. (1999). Fractional/integer-n pll basics. *http://www. ti. com/lit/an/swara029/swara029. pdf*.

Behzad, R. (2012). *RF microelectronics*. Prentice hall.

Bölücek, M. A. (2009). Design and implementation of low phase noise phase locked loop based local oscillator. Master's thesis, Middle East Technical University.

Bonfanti, A., Amorosa, F., Samori, C., and Lacaita, A. L. (2003). A dds-based pll for 2.4-ghz frequency synthesis. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 50(12):1007–1010.

Clark, R. N. (1992). The routh-hurwitz stability criterion, revisited. *IEEE Control Systems Magazine*, 12(3):119–120.

Debnath, S. (2015). *Design of fractional-N frequency synthesizer for 2.4/5 GHz wireless local area network*. PhD thesis, Rutgers University-Graduate School-New Brunswick.

del Rocío Ricardez-Trejo, M., Gutiérrez-Martínez, C., Torres-Fórtiz, J. A., and Meza-Pérez, J. (2017). Modeling and simulation of phase-locked loops (pll) microwave

generators using matlab/simulink® basic blocks. In *2017 IEEE URUCON*, pages 1–4. IEEE.

Devices, A. (2008). Adf4351 datasheet and product info. https://www.analog.com/en/products/adf4351.html#product-overview. (Accessed on 8/02/2022).

Du, Y., Li, W., Ge, Y., Li, H., Deng, K., and Lu, Z. (2017). Note: A high-frequency signal generator based on direct digital synthesizer and field-programmable gate array. *Review of Scientific Instruments*, 88(9):096103.

ECS (n.d). ecs-2200x.pdf. https://ecsxtal.com/store/pdf/ecs-2200x.pdf. (Accessed on 8/10/2022).

Electronics, E. (2017). Ecstuff4u for electronics engineer: Lm7805 pinout — introduction — feature — diagram — application. https://www.ecstuff4u.com/2019/09/lm7805-pinout.html. (Accessed on 10/06/2022).

ElectroPeak (2019). Using 1602 lcd keypad shield w/ arduino [w/ examples] - arduino project hub. https://create.arduino.cc/projecthub/electropeak/using-1602-lcd-keypad-shield-w-arduino-w-examples-e02d95. (Accessed on 11/02/2022).

Fajar, Nur, C., Arseno, and Dharu (2019). Design and realization step frequency continuous wave generator for ground penetrating radar using phase-locked loop. In *2019 IEEE 13th International Conference on Telecommunication Systems, Services, and Applications (TSSA)*, pages 13–16. IEEE.

Fox, A. (2002). Pll synthesizers (ask the applications engineer). *Analog Dialogue*, 36(3).

Handique, J. and Bezboruah, T. (2015). Design, fabrication and analysis of a 1.1 ghz phase-locked loop frequency synthesizer for wireless communication systems. *International Journal of Electronics and Electrical Engineering*, 3(1):19–23.

Harney, A. (2009). Designing high-performance phase-locked loops with high-voltage vcos. *19 Automobile Tail-Lamp and Brake-Lamp Controller*, page 13.

HOSSEINI, T. Y. and Masoumi, N. (2017). Design and implementation of an ultra-wide band, high precision, and low noise frequency synthesizer. *Information Systems and Telecommunication*, 5(4):209–216.

Hwang, I., Lee, S., Lee, S., and Kim, S. (2000). A digitally controlled phase-locked loop with fast locking scheme for clock synthesis application. In *2000 IEEE International Solid-State Circuits Conference. Digest of Technical Papers (Cat. No. 00CH37056)*, pages 168–169. IEEE.

Instrument, G. W. (n.d). Gsp-9330. `https://www.instrumentation2000.com/pub/media/pdf/gw-gsp-9330-manual.pdf`. (Accessed on 11/02/2022).

Jordan, A. (2010). 3.3v voltage regulator 950ma ld1117v33 (ld33v). `https://www.twinschip.com/LD1117V33-3.3V-Voltage-Regulator-LD33V`. (Accessed on 10/07/2022).

Kameche, S. and Feham, M. (2013). Designing and simulating a 2.4 ghz integer-n frequency synthesizer with 1 mhz frequency step. *International Journal of Computer Applications*, 66(3):9–12.

Kumar, D. and Shrimali, H. (2022). Design and implementation of a second order pll based frequency synthesizer for implantable medical devices. *Integration*.

Lacoste, R. (2014). *Robert lacoste's the darker side: Practical applications for electronic design concepts*. Newnes.

Lecturer, S. (2014). What is a linear regulator (ldo regulator)? — design supports — nisshinbo micro devices. `https://www.nisshinbo-microdevices.co.jp/en/design-support/basic/02-linear-regulator.html`. (Accessed on 12/02/2022).

Li, Q., Yang, D., Mu, X. H., and Huo, Q. L. (2012). Design of the l-band wideband lfm signal generator based on dds and frequency multiplication. In *2012 International Conference on Microwave and Millimeter Wave Technology (ICMMT)*, volume 4, pages 1–4. IEEE.

Muteithia, W. M. (2014). *Design And Development Of An FPGA Based DDFS Signal Generator*. PhD thesis, University of Nairobi.

Mutinda, M., Munji, K., and Nyenge, L. (2023). Implementation of a wide band and low noise signal generator based on phase locked loop. *African Journal of Pure and Applied Sciences*, 4(1):50–59.

Patel, G. S. and Sharma, S. (2010). Comparative study of pll, dds and dds-based pll synthesis techniques for communication system. *International Journal of Electronics Engineering*, 2(1):35–40.

Qi, J., Sun, Q., Wu, X., Wang, C., and Chen, L. (2015). Design and analysis of a low cost wave generator based on direct digital synthesis. *Journal of Electrical and Computer Engineering*.

Rohde and Schwarz (2012). Measurement of harmonics using spectrum analyzers. `https://scdn.rohde-schwarz.com/ur/pws/dl_downloads/dl_application/application_notes/1ef78/1EF78_2E.pdf`. (Accessed on 11/02/2022).

Rout, P. K., Panda, B. P., Acharya, D. P., and Panda, G. (2014). Analysis and design of 1 ghz pll for fast phase and frequency acquisition. *International Journal of Signal and Imaging Systems Engineering*, 7(1):30–37.

Shurender, S., Srividhya, K., Mantharachalam, V., Suresh, K., and Habiba, M. U. (2013). Design of s-band frequency synthesizer for microwave applications. *International Journal of Innovation and Applied Studies*, 3(4):1110–1115.

Sun, R. (2013). How to design and debug a phase-locked loop (pll) circuit. *Analog Dialogue*, 47(9).

Telba, A., Noras, J., Abou El Ela, M., and Almashary, B. (2004). Simulation technique for noise and timing jitter in phase locked loop. In *Proceedings. The 16th International Conference on Microelectronics, 2004. ICM 2004.*, pages 501–504. IEEE.

Vankka, D. J. (2000). *Digital Synthesizers: Theory, Design and Applications*. PhD thesis, Helsinki University of Technology.

Yang, L., Tian, L., and Hong, W. (2012). An ultra-broadband, high resolution frequency synthesizer. In *2012 International Conference on Microwave and Millimeter Wave Technology (ICMMT)*, volume 4, pages 1–4. IEEE.

Yoon, S. (2004). *LC-tank CMOS voltage-controlled oscillators using high quality inductors embedded in advanced packaging technologies*. Georgia Institute of Technology.

## APPENDIX A

## PYTHON SCRIPT

**Filter components**

C1=270e-12
C2=3.9e-9
C3=120e-12
R1=11e3
R2=22e3

**PLL parameters**

Rfout=387e6
fref=10e6
Kvco=40e6
Kpd=3e-5
N = Rfout/fref
k = (Kvco*Kpd)/N

**Time constants**

T1= (c1*c2*R1)/(c1+c2+c3)
T2= R1*c2
T3= R2*c3

**Filter coefficients**

A0= c1+c2+c3
A1= A0*(T1+T3)
A2= A0*(T1*T3)

# APPENDIX B

## MICROCONTROLLER CODE

```
%%

#include <LiquidCrystal.h>
#include <EEPROM.h>
#include <SPI.h>
#define ADF4351_LE 3

LiquidCrystal lcd(8, 9, 4, 5, 6, 7);
byte poscursor = 0; //current cursor position 0 to 15
byte line = 0; // current LCD display line 0 or 1
byte memory,RWtemp; // EEPROM memory number
uint32_t registers[6] =  {0x4580A8, 0x80080C9, 0x4E42
   , 0x4B3, 0xBC803C, 0x580005} ; // 437 MHz with ref
    at 25 MHz
int address,modif=0,WEE=0;
int lcd_key = 0;
int adc_key_in  = 0;
int timer = 0,timer2=0; // used to measure the
   duration of a key press
unsigned int i = 0;

double RFout, REFin, INT, PFDRFout,
   OutputChannelSpacing, FRACF;
double RFoutMin = 35, RFoutMax = 4400, REFinMax =
   250, PDFMax = 32;
unsigned int long RFint,RFintold,INTA,RFcalc,PDRFout,
    MOD, FRAC;
byte OutputDivider;byte lock=2;
unsigned int long reg0, reg1;

int read_LCD_buttons()
{adc_key_in = analogRead(0);      // read the value
   from the buttons
  if (adc_key_in < 790)lcd.blink();
  if (adc_key_in < 50)return btnRIGHT;
  if (adc_key_in < 195)return btnUP;
  if (adc_key_in < 380)return btnDOWN;
  if (adc_key_in < 555)return btnLEFT;
  if (adc_key_in < 790)return btnSELECT;

  return btnNONE;  // keys not pressed}

//SP Frequency Display on LCD
void printAll ()
{lcd.setCursor(0, 0);
  lcd.print("RF = ");
```

```
40        if (RFint < 100000) lcd.print("␣");
41        if (RFint < 10000)  lcd.print("␣");
42        lcd.print(RFint/100);lcd.print(".");
43        RFcalc=RFint-((RFint/100)*100);
44        if (RFcalc<10)lcd.print("0");
45        lcd.print(RFcalc);
46        lcd.print("␣MHz");
47        lcd.setCursor(0,1);
48        if (WEE==0) {lcd.print("REE=");}
49        else {lcd.print("WEE=");}
50        if (memory<10)lcd.print("␣");
51        lcd.print(memory,DEC);
52        if  ((digitalRead(2)==1))lcd.print("␣LOCKED␣");
53        else lcd.print("␣NOLOCK␣");
54        lcd.print(PFDRFout,DEC);
55        lcd.setCursor(poscursor,line);  }
56
57      void WriteRegister32(const uint32_t value)   //
           Program a 32bit register
58      { digitalWrite(ADF4351_LE, LOW);
59        for (int i = 3; i >= 0; i--)          // loop on 4
             x 8 bits
60        SPI.transfer((value >> 8 * i) & 0xFF); // shift,
             byte masking and send via SPI
61        digitalWrite(ADF4351_LE, HIGH);
62        digitalWrite(ADF4351_LE, LOW);  }
63
64      void SetADF4351()  // Program all registers of the
           ADF4351
65      { for (int i = 5; i >= 0; i--)  // ADF4351
           programming starting with R5
66        WriteRegister32(registers[i]);}
67
68      // SP write Long word (32 bits) in EEPROM between
           address and address+3
69      void EEPROMWritelong(int address, long value)
70      { //Decomposition of the long (32bits) in 4 bytes
71
72        byte four = (value & 0xFF);
73        byte three = ((value >> 8) & 0xFF);
74        byte two = ((value >> 16) & 0xFF);
75        byte one = ((value >> 24) & 0xFF);
76
77        //Write 4 bytes to EEPROM memory
78        EEPROM.write(address, four);
79        EEPROM.write(address + 1, three);
80        EEPROM.write(address + 2, two);
81        EEPROM.write(address + 3, one);}
```

```
82
83      // SP reading Long word (32 bits) in EEPROM located
           between address and address+3
84      long EEPROMReadlong(long address)
85      {//Read the 4 bytes from the eeprom memory.
86        long four = EEPROM.read(address);
87        long three = EEPROM.read(address + 1);
88        long two = EEPROM.read(address + 2);
89        long one = EEPROM.read(address + 3);
90        //Returns long(32bits) using 0, 8, 16 and 24 bit
             shift and masks
91        return ((four << 0) & 0xFF) + ((three << 8) & 0
             xFFFF) + ((two << 16) & 0xFFFFFF) + ((one << 24)
              & 0xFFFFFFFF);  }
92      //Setup
93      void setup() {
94        lcd.begin(16, 2); // two 16 characters lines
95        lcd.display();
96        analogWrite(10,255); //LCD brightness
97        Serial.begin (19200); //  Serial to the PC via
             Arduino "Serial Monitor"  at 9600
98        lcd.print("␣␣␣GENERATOR␣␣␣");
99        lcd.setCursor(0, 1);
100       lcd.print("␣␣␣␣ADF4351␣␣␣␣␣");
101       poscursor = 7; line = 0;
102       delay(1000);
103       lcd.setCursor(0, 0);
104       lcd.print("␣␣␣BONIFACE␣␣␣␣");
105       delay(1000);
106       pinMode(2, INPUT);  // PIN 2 in input for lock
107       pinMode(ADF4351_LE, OUTPUT);   // Setup pins
108       digitalWrite(ADF4351_LE, HIGH);
109       SPI.begin();
110       SPI.setDataMode(SPI_MODE0);
111       SPI.setBitOrder(MSBFIRST);
112
113
114       void loop()
115
116
117       INTA = (RFout * OutputDivider) / PFDRFout;
118       MOD = (PFDRFout / OutputChannelSpacing);
119       FRACF = (((RFout * OutputDivider) / PFDRFout) -
             INTA) * MOD;
120       FRAC = round(FRACF); // We round the result
121       registers[0] = 0;
122       registers[0] = INTA << 15;
123       FRAC = FRAC << 3;
```

```
124       registers[0] = registers[0] + FRAC;
125       registers[1] = 0;
126       registers[1] = MOD << 3;
127       registers[1] = registers[1] + 1 ; // added address
            "001"
128       bitSet (registers[1], 27); // Prescale to 8/9
129       bitSet (registers[2], 28); // Digital lock == "110"
             on b28 b27 b26
130       bitSet (registers[2], 27); // digital lock
131       bitClear (registers[2], 26); // digital lock
132       SetADF4351();  // Program all registers of the
            ADF4351
133       RFintold=RFint;modif=0;
134       printAll();  // LCD display}
135     lcd_key = read_LCD_buttons();  // read the buttons
136     switch (lcd_key)               // Select action
137     {case btnRIGHT: //Right
138       poscursor++; // cursor to the right
139       if (line == 0) {
140         if (poscursor == 9 ) {
141           poscursor = 10;
142           line = 0; }
143         if (poscursor == 12 ) {
144           poscursor = 0; line = 1; }; //if cursor right}
145       if (line == 1) {
146         if (poscursor == 1 ) {poscursor = 5; line = 1; }
              //if cursor on the memory digit
147         if (poscursor == 6 ) {poscursor = 15; line = 1; }
148         if (poscursor==16) {poscursor=5; line=0;};      }
149       lcd.setCursor(poscursor, line);
150       break;
151       case btnLEFT:
152       poscursor--;
153       if (line == 0) {
154         if (poscursor == 4) {poscursor = 15; line = 1;
            };
155         if (poscursor == 9) {   poscursor = 8; line=0;}}
156       if(line==1){
157         if (poscursor==255) {poscursor=11; line=0;};
158         if (poscursor==4) {poscursor=0; line=1;};
159         if (poscursor==14) {poscursor=5; line=1;};}
160       lcd.setCursor(poscursor, line);
161       break;
162       case btnUP:
163       if (line == 0)
164       { if (poscursor == 5) RFint = RFint + 100000 ;
165         if (poscursor == 6) RFint = RFint + 10000 ;
166         if (poscursor == 7) RFint = RFint + 1000 ;
```

```
167         if (poscursor == 8) RFint = RFint + 100 ;
168         if (poscursor == 10) RFint = RFint + 10 ;
169         if (poscursor == 11) RFint = RFint + 1 ;
170         if (RFint > 440000)RFint = RFintold;}
171
172       if (poscursor==15){
173         if( PFDRFout==10){PFDRFout=25;} //FREF setting
174         else if ( PFDRFout==25){PFDRFout=10;}
175         else PFDRFout=25;// in case PFDRF different from
              10 and 25
176         modif=1;  }
177       if( (poscursor==0) && (WEE==1))WEE=0;
178       else if ((poscursor==0) && (WEE==0))WEE=1;        }
179     printAll();
180     break; // fin bouton up
181     case btnDOWN:
182     if (line == 0) {
183       if (poscursor == 5) RFint = RFint - 100000 ;
184       if (poscursor == 6) RFint = RFint - 10000 ;
185       if (poscursor == 7) RFint = RFint - 1000 ;
186       if (poscursor == 8) RFint = RFint - 100 ;
187       if (poscursor == 10) RFint = RFint - 10 ;
188       if (poscursor == 11) RFint = RFint - 1 ;
189       if (RFint < 3450) RFint = RFintold;
190       if (RFint > 440000)  RFint = RFintold;
191       break;}
192     if (line == 1)
193
194   }// Fin LCD keys
195   do { adc_key_in = analogRead(0); delay(1);} while (
        adc_key_in < 900); // wait key release
196   delay (10);timer++; // inc timer
197   if (timer>1000){lcd.noBlink();timer=0;} // cursor off
198 }    %%
```

# APPENDIX C

## HARDWARE TEST SETUP