A Novel Tuneable Microwave Frequency Synthesizer Employing Cascaded PLL Architecture for Balanced Phase Noise and Lock Time Optimization in 5G/6G Networks

Third Year Individual Project – Final Report

April 2025

Advait Paranjpe
11017873

Supervisor:
Dr Zhipeng Wu

Main Body Word Count:
10808

Table of Contents

Abstract

This research presents the design, development, and experimental validation of a novel microwave frequency synthesizer built on a cascaded phase-locked loop architecture. The work addresses the critical challenges encountered in high-frequency communication systems by achieving a balanced trade-off between low phase noise, rapid frequency lock times, and cost-effective integration. Comprehensive theoretical analysis, careful printed circuit board design, and rigorous testing reveal that the new synthesizer not only meets but, in many cases, exceeds the performance benchmarks set by established industry solutions. Key improvements include optimized high frequency board design principles, refined loop filter design, and a fully integrated, user-friendly software control interface, all of which contribute to enhanced signal integrity and reduced jitter. The experimental results, substantiated by direct comparisons with simulation data and existing commercial hardware, highlight the potential of this approach to support the demanding requirements of modern fifth generation and emerging sixth-generation communication networks. The research findings suggest that the cascaded phase-locked loop synthesis provides a scalable and robust solution that can be further refined with advanced integration techniques and adaptive control strategies, paving the way for future innovations in high-performance radio frequency systems.

Key Words: Microwave Frequency Synthesizer, Cascaded Phase-Locked Loop, Phase Noise, Lock Time, 5G/6G Networks

Declaration of originality

I hereby confirm that this dissertation is my own original work unless referenced clearly to the contrary, and that no portion of the work referred to in the dissertation has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Intellectual property statement

i. The author of this thesis (including any appendices and/or schedules to this thesis) owns certain copyright or related rights in it (the "Copyright") and s/he has given The University of Manchester certain rights to use such Copyright, including for administrative purposes.

ii. Copies of this thesis, either in full or in extracts and whether in hard or electronic copy, may be made *only* in accordance with the Copyright, Designs and Patents Act 1988 (as amended) and regulations issued under it or, where appropriate, in accordance with licensing agreements which the University has from time to time. This page must form part of any such copies made.

iii. The ownership of certain Copyright, patents, designs, trademarks and other intellectual property (the "Intellectual Property") and any reproductions of copyright works in the thesis, for example graphs and tables ("Reproductions"), which may be described in this thesis, may not be owned by the author and may be owned by third parties. Such Intellectual Property and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property and/or Reproductions.

iv. Further information on the conditions under which disclosure, publication and commercialisation of this thesis, the Copyright and any Intellectual Property and/or Reproductions described in it may take place is available in the University IP Policy (see http://documents.manchester.ac.uk/DocuInfo.aspx?DocID=24420), in any relevant Dissertation restriction declarations deposited in the University Library, and The University Library's regulations (see http://www.library.manchester.ac.uk/about/regulations/_files/Library-regulations.pdf).

Acknowledgements

I would like to acknowledge the very helpful EEE Help Lab on the bottom floor of the Nancy Rothwell Building. Specifically, Matthew Newman who was willing to give advice when asked.

Chapter 1: Introduction

1.1 Introduction

Over the last two decades, the growth in personal mobile devices has been unmistakable [1]. As the number of personal devices has surged, so too has the demand for rapid and reliable wireless network technologies. Users now rely on these networks for streaming services, gaming platforms, vehicular communications, and a broad range of other high-bandwidth applications, all of which require low latency and stable connectivity [2].

This escalating demand has driven innovations in wireless network design, ultimately leading to the development of 5G networks. In response, infrastructure is being built around the world to implement functional and reliable 5G networks. These networks leverage technologies such as millimetre-wave spectrum, massive MIMO antennas, and network virtualization to meet the complex requirements of modern applications [2]. A typical 5G system (Figure 1) is composed of:

- **User Equipment (UE):** These are the end-user devices – like smartphones and IoT modules – that access 5G services wirelessly.
- **Radio Access Network (RAN):** Base stations (gNodeBs) and associated equipment that connect UEs over the air using the 5G New Radio (NR) interface [3].
- **5G Core Network (5GC):** This is the cloud-native core that manages subscriber information, mobility, and data routing. It authenticates users, establishes reliable connections, and provides access to various services [3]. Built on a service-based architecture (SBA), the core incorporates virtualized network functions separates control and user planes to offer enhanced flexibility [2].
- **Edge Computing (MEC):** Multi-access Edge Computing deploys distributed computing servers at the network's edge, often alongside RAN sites. This placement allows applications and content to be hosted closer to end-users, reducing latency and alleviating backhaul load by processing data locally [4].

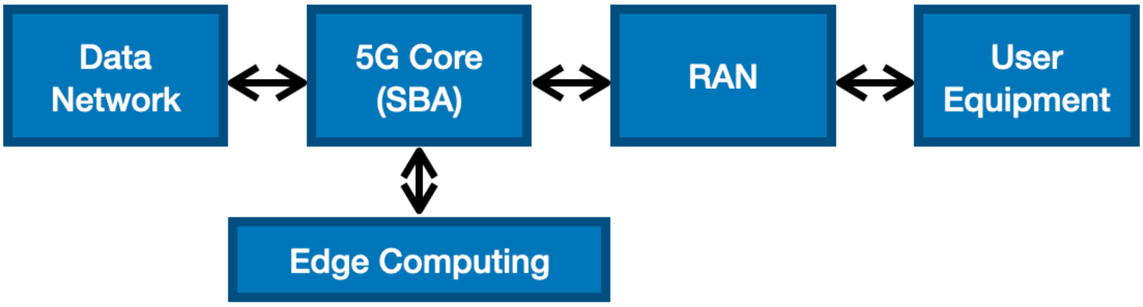


Figure 1: High-level representation of 5G network architecture. Note, MEC servers reside at base stations or RAN sites.

Together, these components enable 5G networks to deliver gigabit speeds. The cloud-based core and RAN enable flexible scaling and network slicing, making 5G a marked improvement over earlier generations [2], [3].

The system architecture underpinning these techniques involves complex signal processing chains that handle high data rates, dynamic interference conditions, and flexible spectrum allocations. Within this architecture, superheterodyne transceivers remain a central element due to their reliability in upconversion and downconversion tasks. They mix incoming or outgoing signals with a local oscillator (LO), converting signals to more manageable intermediate frequencies (IF) [5]. Figure 2 illustrates the block diagram of a typical superheterodyne transceiver.

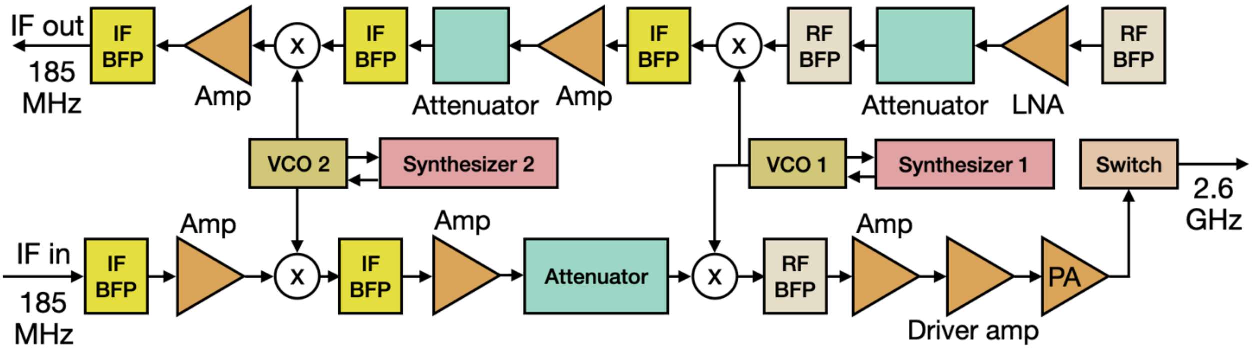


Figure 2: Block diagram representing a typical superheterodyne transceiver [5]

At the heart of every superheterodyne transceiver is a Microwave Frequency Synthesizer (MFS), which provides the stable, low-noise LO signals required for accurate frequency conversion [6]. This essential component becomes even more critical when considering that 5G networks operate over diverse and often fragmented bands – from sub-1 GHz coverage layers to millimetre-wave bands exceeding 20 GHz – and must adapt rapidly to changing channel conditions and service demands. As a result, the MFS must switch frequencies quickly and precisely, ensuring low latency, high throughput, and robust interference management across the entire spectrum [7], [8].

Yet this requirement for agile and precise frequency switching presents challenges. Conventional MFS designs face trade-offs among phase noise, frequency lock time, and cost. High-performance architectures may be too large or expensive for some deployments, while simpler designs can compromise signal purity or lock times. These issues become especially pressing in massive MIMO arrays and other multi-channel scenarios, where each transceiver path demands a clean, rapidly switchable LO source [9].

1.2 Aim

Therefore, to address these issues, the primary aim of this investigation is to design, develop, and implement a novel MFS with a cascaded phase-locked-loop (PLL) architecture and fully integrated software control, all encapsulated with a user-friendly interface. The cascaded PLL approach is particularly attractive because it permits dedicated optimization of each processing stage, improving frequency agility and reducing overall complexity without incurring prohibitive costs [10]. To verify the systems performance, the synthesizer will be rigorously tested, with attention given to metrics such as phase noise, jitter and lock time.

To translate these aims into actionable steps, a set of specific objectives have been established. These objectives detail the design, fabrication, debugging, and performance analysis phases of this paper, ensuring that each element of the research is systematically addressed.

1.3 Objectives

- Design a schematic using Altium Designer for the chosen architecture
- Export the schematic to an Altium Designer PCB layout and design the layout
- Compile and send appropriate files to a manufacturing team to fabricate the PCB
- Debug the PCB using an external debugger to program the registers of the PLL chip.
- Write firmware which utilises the MCU's built in timer to analyse frequency lock time.
- Measure key performance metrics of the MFS, including phase noise, lock time and jitter.

1.4 Paper Structure

The remainder of this paper is structured as follows. First, a high-level overview of the proposed synthesizer architecture, highlighting the cascaded PLL structure, will be presented. This will be followed by a detailed review of the existing literature to verify whether the gaps identified in Section 1.1 are appropriate. This will be followed by a detailed theoretical background section, providing essential knowledge on PLL structures, high-speed digital PCB design considerations, and SPI communication protocols. The subsequent Initial Architecture Design chapter will be grounded in this theoretical discussion. Finally, the results obtained from the physical PCB will be analysed thoroughly, focusing on key performance metrics including phase noise, RMS jitter and frequency lock times. The following literature review not only expands upon the gaps outlined in Section 1.1 but also substantiates the need for the novel cascaded PLL architecture proposed in this paper.

1.5 High-level System Overview

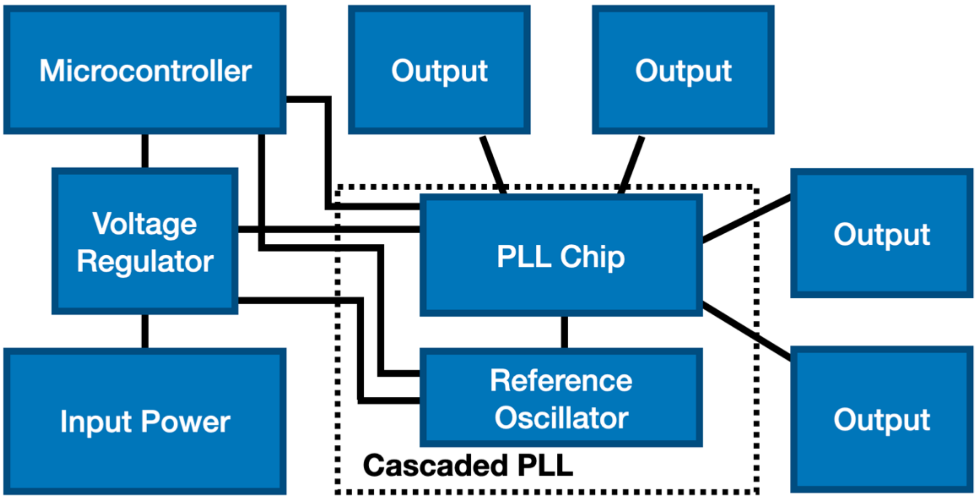


Figure 3: High-level Block Diagram of System Architecture

Figure 3 represents the high-level system architecture, highlighting the cascaded PLL architecture.

1.6 Literature Review

To set the stage for the proposed approach seen in Figure 3, the following literature review examines recent developments in MFS technologies. Despite significant advances in key 5G components such as network virtualization and RF front-end designs, it is the performance of the MFS that ultimately constrains these systems.

Direct Digital Synthesis (DDS)-based solutions offer rapid tuning and fine frequency resolution but typically exhibit inferior phase noise and stability at high frequencies compared to PLL-based designs, which are crucial for reliable 5G communication [14]. Similarly, Optoelectronic Oscillators (OEOs) provide exceptional spectral purity but come with increased complexity, larger physical size, and integration difficulties [15]. Given these trade-offs, a cascaded PLL synthesizer incorporating an integrated Voltage-Controlled Oscillator (VCO) appears as a balanced and practically implementable solution for compact, high-performance 5G systems.

The paper *"Design of Frequency Synthesizer Based on PLL Architecture at X-band"* [11] demonstrates a relevant PLL-based synthesizer achieving stable and low-noise microwave frequency generation. Unlike this paper's integrated cascaded PLL approach, the referenced design uses discrete components – a separate VCO, PLL, and external crystal reference oscillator. Although flexible, this discrete architecture increases complexity and introduces design uncertainties due to intricate component interactions. Additionally, its frequency range (9.5–10.8 GHz) and lock-time optimization are limited [11]. In contrast, this paper's chosen system architecture, optimised using specialised software (TCIS Pro), widens the frequency range (100

MHz - 9.8 GHz), and aids in achieving faster frequency lock times through register level programming.

Another relevant design is introduced by the paper *"Ultra-Low Phase Noise and Frequency Agile X-Band Frequency Synthesizer Based on a Phase Noise Optoelectronic Oscillator (OEO) and Direct Digital Synthesizer (DDS)"* [12]. This hybrid approach combines DDS's rapid frequency tuning capability – derived digitally through a phase accumulator and lookup tables [13] – with the ultra-low phase noise advantages of OEO technology. While the paper reports impressive phase noise performance (-139 dBc/Hz, decibels relative to the carrier per hertz, at 10 kHz offset), this design suffers from significant drawbacks in complexity, size, and cost due to photonic integration. Such limitations restrict practical use in compact and cost-sensitive 5G architectures.

The paper *"A fast-switching low-spurious 6–18 GHz hybrid frequency synthesizer"* [14] combines DDS and PLL technologies. This hybrid achieves rapid switching (6 µs) and low spurious emissions, benefitting from DDS's speed and PLL's spectral purity [15]. Nevertheless, the DDS-PLL integration adds complexity and cost. Although performance is promising, the additional complexity compared to a simpler cascaded PLL architecture could introduce integration difficulties, especially in compact 5G systems. Therefore, a critical open question explored in this paper is whether a simpler cascaded PLL architecture can achieve similar or superior results in terms of phase noise, switching speed, and cost efficiency.

Further exploring cascaded PLL techniques, the paper *"A Dual-Band 28/38 GHz Cascaded Phase-Locked Loop Circuit Design"* [16] introduces a dual-mode quadrature VCO architecture for dual-band operation. It employs a two-stage PLL (a charge-pump PLL followed by a subsampling PLL) to achieve improved phase noise by removing frequency dividers. While the design delivers excellent phase noise (~ -115 dBc/Hz at 1 MHz offset), it lacks rapid frequency agility and broader tuning capability beyond two fixed frequency bands. Moreover, the quadrature and dual-mode approach adds complexity in integration and PCB layout, potentially increasing overall system risk.

Lastly, the paper *"A 28-GHz Quadrature Fractional-N Frequency Synthesizer for 5G Transceivers With Less Than 100-fs Jitter Based on Cascaded PLL Architecture"* [10] implements a sophisticated fractional-N cascaded PLL combined with a quadrature subsampling PLL stage. This design achieves exceptional jitter (<100 fs) and impressive phase noise (~ -112 dBc/Hz at 1 MHz offset) performance, making it highly suitable for advanced modulation schemes such as 256-QAM [17]. Despite these strengths, the fractional-N and subsampling PLL implementation significantly increases PCB design complexity and cost. Thus, this the architecture explored in this paper aims to balance performance with practical PCB implementation considerations, addressing cost and complexity concerns without sacrificing essential performance metrics.

In summary, the in-depth literature review confirms that the gaps speculated in Section 1.1 persist: existing MFS designs consistently suffer from trade-offs among complexity, integration challenges, limited frequency agility, suboptimal phase noise performance, and high implementation costs. Recognizing these persistent limitations, this paper proposes that the novel solution of a cascaded PLL synthesizer is well-suited to address these challenges. This design aims to improve frequency agility, reduce design complexity, and achieve more seamless practical integration, thereby directly addressing the gaps identified in current research and forming the foundation of the present investigation.

Following the identification of these gaps, Chapter 2 provides an in-depth exploration of PLL technology, detailing the operational principles and design considerations that serve as the cornerstone of the proposed synthesizer design.

Chapter 2: Phase-Locked Loops - Theoretical Background

A Phase-Locked Loop (PLL) is a closed-loop feedback control system designed to synchronize the phase and frequency of an oscillator with a given reference signal. PLLs are fundamental components in frequency synthesis, clock generation, and signal synchronization across various high-performance applications. The importance of PLL technology lies in its ability to offer precise frequency control, low phase noise, and minimal jitter. This chapter outlines the fundamental operating principles of PLLs, providing a theoretical foundation for subsequent discussions on design methodologies and performance optimization strategies [18], [19].

2.1 Basic PLL Components and Structure

A PLL comprises four core components that collectively synchronize an oscillator's output with a reference input [18]:

- **Phase Detector (PD):** Compares the phase (and frequency) of the reference and feedback signals, producing an error signal proportional to their difference. Common architectures include XOR-based, mixer-based, and phase-frequency detectors (PFDs). The PD's accuracy and sensitivity directly influence the PLL's overall performance [18].
- **Loop Filter (LF):** Processes and smooths the error signal into a stable control voltage, determining the PLL's dynamic response – particularly loop bandwidth and damping

behaviour. Proper filter design ensures rapid, stable locking with minimal overshoot and suppressed noise [18].

- **Voltage-Controlled Oscillator (VCO):** Generates a frequency dependent on the control voltage from the loop filter. Key parameters include tuning sensitivity and frequency range. Adjusting the VCO input reduces phase error until synchronization with the reference is achieved [18].
- **Frequency Divider:** Scales the VCO output frequency to a lower value suitable for direct phase comparison with the reference. Integer-N dividers use simple integer ratios, whereas fractional-N dividers employ more sophisticated division schemes for finer frequency resolution [18].

2.2 Operational Principles

The fundamental operation of a PLL can be viewed as a dynamic, continuous control process. The following summarises how the individual components identified in Section 2.1 function together. Initially, the phase detector compares the phases (and frequencies) of the reference and scaled feedback signals, generating an error signal proportional to the detected difference. Subsequently, the loop filter smooths this error signal, producing a stable control voltage that modulates the VCO frequency, reducing phase differences iteratively. When the PLL reaches equilibrium, the VCO output frequency and phase remain closely synchronized with the reference, maintaining any residual phase errors within permissible limits [18], [19].

Key operational concepts for a PLL include [20].

- **Lock Range**: The frequency range over which the PLL stays synchronized to the reference.
- **Capture Range**: The frequency range over which the PLL can establish lock.
- **Loop Dynamics**: Determined by the loop filter’s design parameters and damping factor.

2.3 Key Performance Metrics

PLLs are typically evaluated using several interconnected metrics, each critically affecting system performance in high-frequency communications and related applications [20]:

- **Phase Noise**: Measured in dBc/Hz, phase noise characterizes short-term instability and frequency fluctuations around the main carrier frequency. High phase noise adversely affects signal integrity, increases bit error rates, and reduces overall spectral efficiency.
- **Root Mean Square (RMS) Jitter**: RMS jitter quantifies temporal uncertainty in the PLL’s output signal. It is derived by integrating phase noise over a defined frequency offset range and converting the result into a time-domain measure. Minimizing jitter is essential in timing-critical applications such as digital communications and precision timing circuits.

- **Lock Time**: Lock time indicates the duration necessary for a PLL to transition from an unlocked state to stable phase synchronization after perturbations in the reference frequency. Lock time is mainly governed by the loop filter bandwidth and damping characteristics.

Optimizing PLL performance involves careful consideration of these interrelated metrics. Enhancements in one performance measure typically involve compromises in another, underscoring the importance of balanced and application-specific PLL design strategies [20], [21].

2.4 Practical Considerations and Design Trade-Offs

Designing effective PLL systems involves addressing practical constraints, implementation technology limitations, and careful navigation of performance trade-offs among key metrics.

A primary practical consideration is the design of the loop filter. Selecting a broader loop bandwidth results in shorter lock times, improving dynamic responsiveness. However, this also permits increased phase noise penetration, deteriorating the spectral purity of the output signal. Conversely, a narrower loop bandwidth reduces phase noise but increases lock time. Additionally, choosing the loop filter order substantially influences transient performance and stability [18], [20], [21].

The quality of the reference signal is another essential consideration. Because any reference signal imperfections propagate through the PLL's feedback loop, utilizing a stable, low-noise reference source is vital for optimal system performance. Attention to power supply noise mitigation, decoupling techniques, and careful layout strategies are key for maintaining reference quality [21].

Lastly, frequency dividers add complexity in fractional-N PLL designs. Although fractional dividers enable finer frequency control compared to integer dividers, they add additional noise and spurs that designers must manage through circuit optimization and architectural decisions [20].

In summary, effective PLL design demands a holistic approach informed by both theoretical principles and empirical data. System-level requirements such as high-speed operation, minimal jitter, rapid frequency locking, and low phase noise performance drive critical design decisions, highlighting the need for thorough consideration of both theoretical and practical constraints. The next chapter outlines theoretical and practical constraints that go into designing the PCB layout, which directly influence the performance of the PLL structure.

Chapter 3: Microwave Frequency PCB Design Background

High-speed digital design is an engineering discipline that relies on a solid theoretical framework to ensure both functionality and optimized performance in modern PCBs. Given the inherently creative nature of this process, engineers must be well-versed in the fundamental principles that govern signal integrity and power distribution at high speeds. As such, this paper will dissect five critical theoretical domains: Inductance, Crosstalk, Power distribution, Layer Stack Design and Transmission Line Modelling [22].

3.1 Inductance

Inductance arises from the relationship between current (I), and magnetic flux (Φ), as expressed in Equation (1) [23]:

$$\Phi = L \cdot I \tag{1}$$

Changing current during high-speed switching causes the magnetic flux (Φ) to change proportionally to inductance (L). According to Faraday's Law of Electromagnetic Induction, this change in flux induces an electromotive force (EMF) that opposes the change in current:

$$\varepsilon = -\frac{d\Phi}{dt} = -L\frac{dI}{dt} \tag{2}$$

Here, ε is the induced EMF (in Volts), and $\frac{dI}{dt}$ is the rate of change of current. The negative sign indicates that the induced EMF opposes the change in current, as per Lenz's Law [23]. In high-speed PCB designs, signals switch between logic levels in nanoseconds, Therefore, even small parasitic inductances (L) can then generate significant back EMF, causing voltage spikes and transient over-voltages. This high back EMF often leads to critical PCB issues, as shrinking rise times magnify the effect of even minimal inductances. Minimizing inductance (and thus back EMF) is critical. PCB designers achieve this by reducing loop area and optimizing current paths [22].

A key strategy is reducing loop area and optimize current paths to minimize inductive effects. One effective strategy is routing signal traces over a return plane, so their magnetic fields cancel each other. Even if another route appears shorter, at high frequencies the return current follows the path with the lowest loop impedance, dominated by inductive reactance. Minimizing loop area directly reduces inductance and magnetic flux, cutting down back EMF and enhancing signal integrity. Consequently, placing the return path beneath the signal trace provides the least inductive route and the best overall performance [22], [23].

3.2 Crosstalk

However, controlling loop inductance is only part of the puzzle; designers must also address crosstalk, another key factor that can degrade signal integrity in high-speed PCB layouts. In essence, high-speed systems rely on rapid, often gigahertz-level signal transitions – commonly referred to as “aggressors”– to drive circuit logic. These fast-changing signals can induce EMI in adjacent traces, thereby creating “victims” that experience unwanted noise. Left unaddressed, this crosstalk can degrade signal integrity, cause timing errors, and undermine the overall performance of densely populated PCB layouts. There are three major causes of crosstalk: Shared Impedance, Mutual Inductance and Mutual Capacitance. It is worth breaking these three down into further detail to grasp the implications of crosstalk on high-speed digital PCBs [22], [23].

3.2.1 Types of Crosstalk

When multiple circuits share a single power source or via, they share a common impedance. At high frequencies, the inductive component of this shared path becomes significant – rapid switching in one circuit can induce voltage seen by another, introducing noise.

Similarly, mutual capacitance arises when adjacent traces share an electric field, enabling noise from one to couple into another. This coupling induces a displacement current that can distort waveforms, cause timing errors, and push voltage levels beyond acceptable thresholds. As frequency increases, the lower capacitive reactance intensifies this effect. Additionally, mutual inductance occurs when a changing current in one trace generates a magnetic field that induces a voltage in a nearby trace. At high frequencies, these rapid current changes can produce induced voltages, leading unwanted signal fluctuations. Together, these parasitic effects – shared impedance, mutual capacitance, and mutual inductance – can severely compromise signal integrity in high-speed PCB designs [22], [23].

3.2.2 Mitigation Strategies for Crosstalk

Crosstalk in high-speed digital systems can be minimized with careful PCB layout and power distribution. Segregating power rails, using dedicated supply paths, and reducing inductance in shared return or power traces all help mitigate noise from common impedances. Spacing traces further apart, incorporating solid ground planes, and routing high-speed lines away from sensitive signals lower mutual capacitance and inductance. Keeping loop areas small and parallel traces short also reduces coupling. All these techniques have been implemented in this paper's PCB design to preserve signal integrity and minimize crosstalk [22], [23].

3.3 Power distribution

The next critical element in high-speed digital logic circuit design is the power distribution network that supplies various components across the board. Rapid switching causes fast voltage transitions and sudden current bursts. Decoupling capacitors act as local charge reservoirs – quickly discharging to supply current surges and recharging during low activity. This localized support maintains voltage stability and reliable circuit operation [22], [23]

In practice, every capacitor exhibits parasitic inductance and resistance due to its leads, internal structure, and mounting pads. Illustrated in Figure 4, these parasitic elements cause the capacitor's impedance to vary across different frequencies. At its self-resonant frequency, the inductive and capacitive reactance's cancel each other, causing the capacitor to transition from capacitive behaviour to inductive behaviour. Beyond this point, the device effectively acts more like an inductor. Consequently, no single capacitor can provide optimal performance across all frequencies. This is why a tiered decoupling scheme – employing capacitors of various values and self-resonant frequencies – is essential for stable power delivery and robust high-speed circuit operation [22], [23], [24].

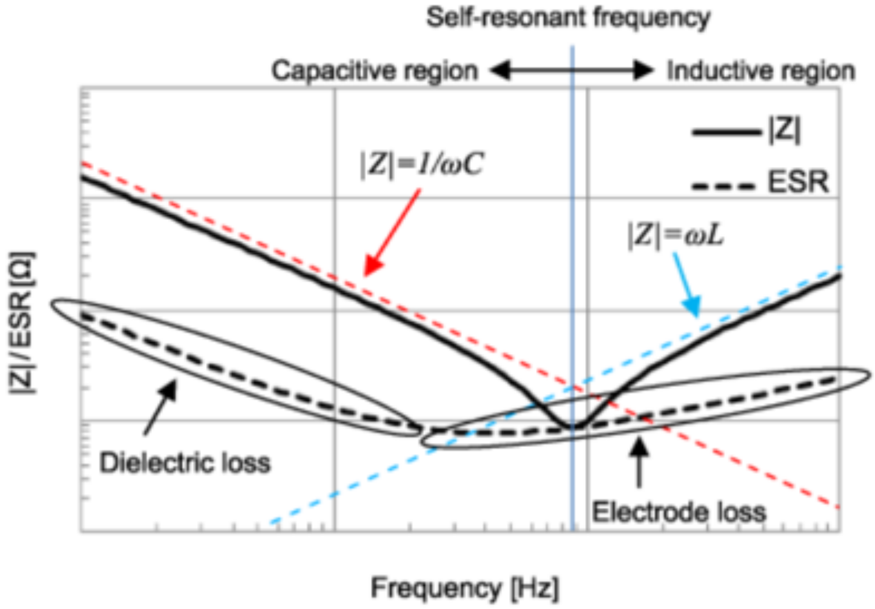


Figure 4: Plot of a capacitors behaviour as the frequency increases [23]

Another method for supplying transient current bursts involves placing solid, continuous power and ground planes adjacent within the PCB. This arrangement effectively forms a distributed capacitor across the board, offering an additional localized energy reserve at higher frequencies. By reducing the distance between the planes, designers minimize inductance and enhance the board's ability to rapidly deliver current where it is needed [22], [23].

3.4 Layer Stack Design

This neatly leads on to layer stack design within a PCB. There are six common practices to follow when deciding upon a layer stack up [23]:

- **Signal Layer Adjacent to a Reference Plane**
  Ensuring each signal layer directly borders a solid ground (or power) plane minimizes the loop area for return currents. This reduces crosstalk, improves impedance control, and enhances signal integrity.
- **Tightly Coupled Power and Ground Planes**
  Placing power and ground planes close together forms a distributed capacitor that helps stabilize supply voltages at high frequencies. The reduced plane spacing also lowers inductance and improves decoupling performance.
- **High-Speed Signals Buried Between Reference Planes**
  Embedding critical traces between reference planes creates a shielding effect, reducing EMI and further stabilizing impedance. This is crucial for preserving signal integrity in high-speed designs.
- **Multiple Ground Planes for Lowest Possible Impedance**
  Multiple ground planes lower the overall return-path impedance, helping to maintain clean reference levels. This approach reduces noise coupling, supports more efficient current return paths, and eases EMI/EMC compliance.
- **Mechanically Balanced Stack-Up**
  A symmetrical or balanced layer arrangement prevents warping during PCB fabrication and thermal cycling. Consistency in layer distribution also maintains predictable impedance characteristics across the board.

3.5 Transmission Line Modelling

At the switching speeds of highspeed systems, the wave-like nature of signal propagation becomes crucial, and designers must account for wave reflections, impedance mismatches and more. Maintaining signal integrity involves ensuring that the rapid transitions between high and low logic levels are both distinct and reliably detected. Thus, what truly defines a high-speed system is not simply the fundamental clock frequency, but the rapid rise and fall times of its signals. A square wave can be viewed as a sum of multiple higher-order harmonics. The sharper the edges of the pulse, the more high-frequency components are required to recreate that shape. Consequently, any design that incorporates rapid signal transitions – regardless of its fundamental frequency – demands careful consideration of high-speed effects [25].

3.5.1 Voltage as a Wave

As a signal travels along a PCB trace, it reflects at endpoints until it settles. This behaviour distinguishes lumped circuit models – which assume instantaneous voltage across the circuit – from distributed line models that account for wave propagation. When a signal's rise time is long relative to the trace length, the voltage is nearly uniform, and a lumped model suffices. However, if the rise time is short compared to the trace length, different segments reach different voltages simultaneously, requiring a distributed model. [23], [25]. The concept of "electrical length" (L_e) – the distance a signal transitions between voltage levels – helps determine the appropriate model: if the trace is much shorter than (L_e), lumped analysis is valid; if comparable or longer, a distributed approach is needed. A general rule of thumb to follow in this distinction is the following [23]:

$$Distributed\ Line\colon \frac{L_e}{6} < physical\ length \tag{3}$$

Dealing with distributed transmission lines, two key parameters must be defined: propagation velocity $\left(v_p\right)$ and propagation delay (t_d). The propagation velocity describes how quickly an electromagnetic wave travels along a conductor, in meters per second. The propagation delay is the time it takes for the wave to traverse a specified distance on the line, in seconds per meter. [23]. Both parameters depend on the effective relative permittivity $\left(\varepsilon_{eff}\right)$ of the medium surrounding the conductor. This relationship is approximated by second portion of Equation 4 [23].

$$v_p = \frac{1}{t_d} = \frac{c}{\sqrt{\varepsilon_{eff}}} \tag{4}$$

Where c is the speed of light in a vacuum. To decide whether to treat a line as lumped or distributed, one calculates its electrical length by multiplying the propagation velocity by the signal's rise time. However, in very high-speed designs, it is generally advisable to treat all traces as distributed, since even minor disturbances can have a significant impact on performance [23].

3.5.3 Transmission Lines

In Section 3.5.2, lumped and distributed models were distinguished, noting that the primary difference lies in the length of the trace relative to the signal's rise time. Conceptually, any trace can be divided into an infinite series of infinitesimally small segments, each treated as a lumped element. This segmentation demonstrates how distributed modelling naturally extends from repeated lumped approximations along the trace. When examining each lumped segment, the roles of the inductor and capacitor become apparent. As illustrated in Figure 5, when a voltage is applied at the trace input, the inductor initially resists the change in current. Once this is overcome, current flows and charges the capacitor until it reaches the applied voltage. This sequence repeats for each subsequent segment along the trace, effectively propagating the signal until it reaches the end of the line. These are the building blocks of the concept: “transmission line” [23], [25]

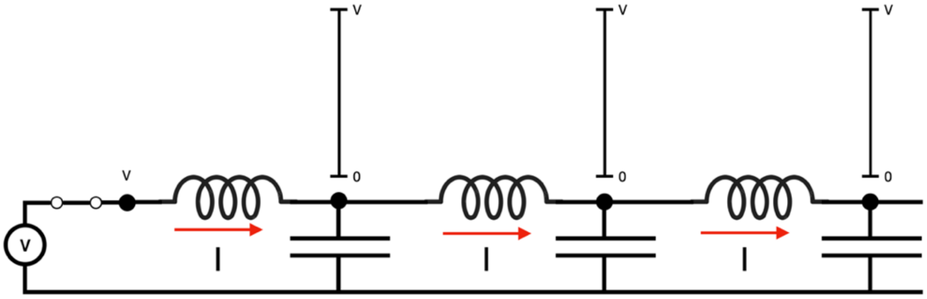


Figure 5: Lumped line segments of a transmission line

The characteristic impedance, denoted Z_0 is key for a transmission line [23]. Figure 6 illustrates a segment, with an inductor L, a capacitor C, and a load Z_0 representing the next segment of the line.

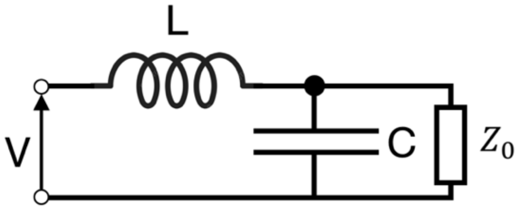


Figure 6: One element of the lumped transmission line abstraction

Assuming each segment has the same characteristic impedance Z_0, Z_0 can be placed in parallel with the capacitor and in series with the inductor, as illustrated in Figure 6. The impedance looking into this single segment is therefore:

$$\frac{V}{I} = Z_0 = j\omega L + \ Z_0 || \frac{1}{j\omega C}$$

Rearranging and solving for Z_0, an expression is obtained containing both L and C.

$${Z_0}^2 = \frac{L}{C} + j\omega L Z_0$$

As each lumped segment is made infinitesimally small, L and C also become very small. In this limit, the $j\omega L Z_0$ term is negligible compared to the $\frac{L}{C}$ term, yielding:

$${Z_0}^2 = \frac{L}{C}$$

Thus, the characteristic impedance simplifies to [23], [25]:

$$Z_0 = \sqrt{\frac{L}{C}} \tag{5}$$

For an ideal transmission line, Z_0 depends solely on its per-unit-length inductance (L) and capacitance (C), independent of frequency. However, impedance mismatches create discontinuities that reflect signals – degrading high-frequency signal integrity. The reflection coefficient, gamma (Γ), quantifies these reflections as the ratio of reflected to incident voltage (or the negative ratio for current) at a junction between impedances Z_2 and Z_1 [23],

$$\Gamma = \frac{V^-}{V^+} = -\frac{I^-}{I^+} = \frac{Z_2 - Z_1}{Z_2 + Z_1} \tag{6}$$

where V^+ and I^+ are the voltage and current traveling into the junction, and V^- and I^- are those reflected back. Keeping Γ small – or ideally zero by matching impedances – minimizes reflections and helps maintain signal integrity [23].

3.5.4 Reflection Mitigation Strategies

To minimize reflections, the source and load should match the transmission line's characteristic impedance, typically achieved by using termination resistors at one or both ends (see Figure 7). Each termination configuration offers distinct pros and cons, outlined in Table 3.5.4.1 below.

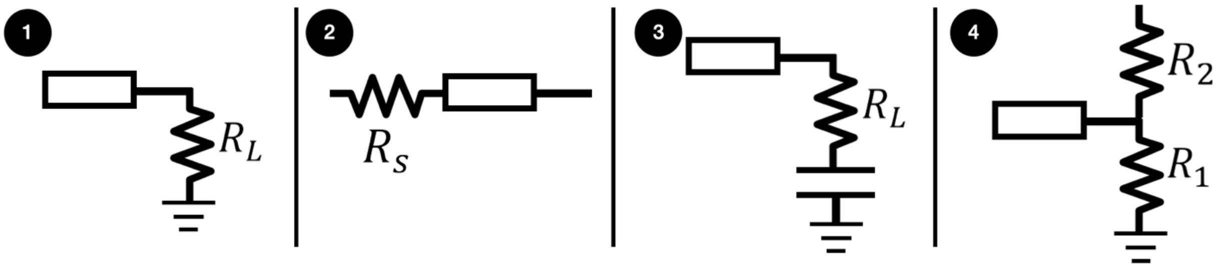


Figure 7: Common source and load termination resistor topologies

Termination	Advantages	Disadvantages
1. Parallel	• Terminate to PWR/GND • R_L easy to determine • Only one additional component	• Power dissipated in R_L at all times • High power requirement
2. Series	• One component • No DC load	• Can be difficult to optimise R_S • Absorbs reverse reflection
3. AC	• No DC drain • No DC load	• C difficult to optimise • Potential timing problems
4. Thevenin	• Pullup/pulldown resistors can improve noise margins • Steady current flow/dissipation	• Continuous current flow through resistors • R_1 and R_2 can be complex

Table 3.5.4.1

By selecting the right termination strategy and carefully matching impedances, designers can substantially reduce reflections and maintain cleaner waveforms at higher frequencies.

3.5.5 Crosstalk in Transmission Lines

For high-speed signals, maintaining a consistent geometry – especially a stable width-to-height ratio between the trace and its reference plane – is crucial for preserving the characteristic impedance (see Equation 7). Nevertheless, mutual inductance between adjacent traces can induce unwanted coupling, degrading signal integrity (see Section 3.2.1). A method to reduce this is to increase the spacing (D) between traces. Given that these traces typically have a well-defined height (h) above their reference plane, one can *estimate* the resulting crosstalk using the following relationship [23]:

$$Crosstalk = \frac{K}{1 + \left(\frac{D}{h}\right)^2} \tag{7}$$

Here, K is a constant that depends on the specific geometry and materials. In typical designs, the ratio (D/h) is sufficiently large that the crosstalk drops quickly as the lines are spaced further apart. For instance, separating traces by roughly $4h$ may reduce crosstalk to around 6 percent, which is generally acceptable in high-speed systems.

3.5.5.1 Near End and Far End Crosstalk

Until now, crosstalk has been treated as a lumped phenomenon, overlooking the wave-like propagation in distributed transmission lines. In high-speed systems, crosstalk pulses travel along adjacent lines, adding complexity to how crosstalk manifests [22], [23], [25].

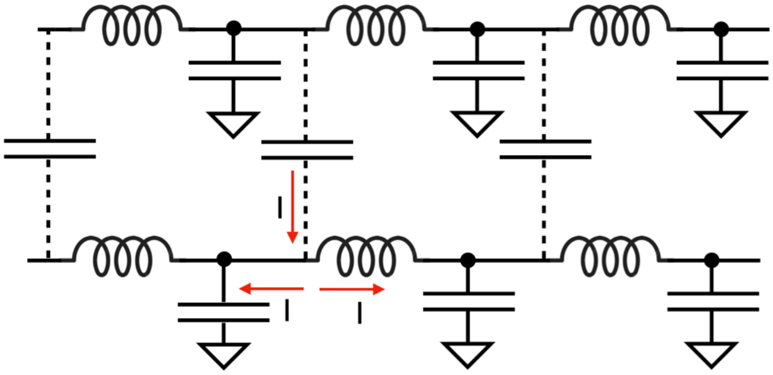


Figure 8: NEXT and FEXT representation

Figure 8 shows two parallel transmission lines models (see Section 3.5.3). The dotted black lines with capacitors represent the mutual capacitance between them. As a voltage wave travels along the primary line, each segment induces a current in the secondary line. These induced signals propagate in both directions, resulting in near-end crosstalk (NEXT) and far-end crosstalk (FEXT) [23].

This concludes the theoretical background on high-speed PCB design. Next, Serial Peripheral Interface communication protocols are examined to provide a full theoretical background on all information and techniques used in the development of the MFS proposed in this paper.

Chapter 4: Software Theoretical Background

In this paper, the microcontroller and peripheral components communicate via the Serial Peripheral Interface (SPI) – a synchronous, full-duplex protocol that uses a dedicated clock line for data transfers. SPI is especially suitable for high-speed designs because it supports higher clock rates than protocols like I2C [26].

4.1 SPI Architecture

SPI follows a master-slave model. The master generates the clock (SCLK), initiates and terminates communication, and selects the slave via a chip-select ($\overline{CS}$) line. The slave responds when $\overline{CS}$ is low and exchanges data with the master via two lines: MOSI (Master Out, Slave In) and MISO (Master In, Slave Out). This straightforward structure allows full-duplex communication at high speeds and can easily accommodate multiple slaves [26].

4.2 SPI Signal Lines

SPI communication typically has four dedicated signal lines [26]:

- Chip Select ($\overline{CS}$): Active-low line from master to slave enabling communication
- Serial Clock ($SCLK$): Clock line from master to slave
- Master Out, Slave In ($MOSI$): Data sent from master to the slave
- Master Out, Slave In ($MISO$): Data sent from slave to master

4.3 Data Transmission

During an SPI transfer, bits shift out serially between the master's and slave's shift registers. If the master only needs to transmit, it can ignore incoming data; for receiving, it sends dummy data to clock in the slave's output. Two key parameters – clock polarity (CPOL) and clock phase (CPHA) – govern how data is sampled. Each can be 0 or 1, yielding four clock modes [26].

4.4 Summary

SPI excels in several areas [26]:

- **High-Speed Capability**: It supports high clock frequencies, ideal for rapid data transfers.
- **Simplicity and Low Overhead**: Straightforward hardware and protocol reduce complexity.
- **Flexibility**: Multiple slaves are easily integrated by assigning each its own chip-select line.

Because of these advantages, SPI is used to ensure reliable, high-speed communication. Combining the hardware design principles from Chapter 3 with the SPI communication strategies, the next chapter presents the practical development of the synthesizer's initial architecture.

Chapter 5: Initial Architecture Development

5.1 Schematic Design

The initial stage in system design involves selecting compatible, cost-effective components. For this paper the key components are:

- LMX2592 RF Synthesiser with integrated VCO (LMX) [27]: Chosen for its wide frequency range, excellent phase noise, and low jitter – crucial for high-performance 5G.
- LMK62E2-100M Low Jitter Oscillator (LMK) [28]: Provides a stable 100 MHz reference, ensuring the LMX operates optimally.
- STM32F401RE Microcontroller (STM32) [29]: Selected for its high computational performance and ease of interfacing.
- NCP1117LP Voltage Regulator (NCP) [30]: Converts a 5 V USB input to a compatible 3.3 V supply for all key components.
- ST-Link V2 Debugger [31]: Facilitates efficient debugging and development.

These components were chosen based on their performance, compatibility, and cost efficacy. The next step is developing the schematic layout to ensure proper interconnections among these devices.

5.1.2 Schematic Layout

The entire PCB was drawn using Altium Designer. To have a detailed look at the schematic, refer to Section 1.1 and 1.2 in the appendix. Additionally, it is important to verify that the design aligns with critical datasheet recommendations to guarantee optimal performance.

5.1.3 Datasheet Recommendations/Justifications

The LMX datasheet [27] recommends several key practices for stable, low-noise operation.

Pin/Feature	**Recommended Practice and Implementation**	**Datasheet Section**
VDD pins	Each VDD pin should have a 0.1 μF bypass capacitor placed close to the pin to minimise high-frequency noise, plus 1 μF and 10 μF capacitors on the same rail for bulk decoupling.	Table 5.1
Reference Clock Input	A series 0.1 μF capacitor is suggested to block DC offset and ensure proper AC coupling.	Table 5.1
Differential Outputs	For differential outputs, 50 Ω resistors and 100 pF capacitors to be connected in parallel to match impedance and reduce reflections	Section 8.2

Table 5.1.3.1

Similarly, the LMK datasheet and the capacitor–inductor frequency graph (see Figure 4 in Section 3.3) [28] [24] recommends key practices for stable, low-noise operation.

Pin/Feature	**Recommended Practice and Implementation**	**Datasheet Section**
Power Supply Decoupling	Place bypass capacitors on the VDD pin to minimise high frequency noise. A 0.1 nF capacitor is used for high frequency decoupling (>100 MHz), supplemented by bulk 1nF and 10 nF capacitors.	Section 7.2.1.2
Clock Differential Output	150 Ω termination resistors are suggested on the pair of clock outputs to minimise ringing and reduce reflections going into the critical LMX chip.	Section 6
Clock Enable	A 10 $K\Omega$ pullup resistor to the enable the LMK chip.	Table 4.1

Table 5.1.3.2

The STM32 datasheet [29] also recommends key practices for correct functionality:

Pin/Feature	**Recommended Practice and Implementation**	**Datasheet Section**
VDD pins	Every VDD pin should have a 0.1 μF, bypass capacitor to filter high frequency noise along with 1 μF and 10 μF bulk capacitors for overall power stability	Figure 43
NRST	Should have an external 10 $k\Omega$ pullup resistor with a 0.1 μF capacitor in parallel to enable debugging	Figure 32
BOOT Configuration	The BOOT0 must have an external 10 $k\Omega$ pulldown resistor in series to the pin and tied to ground to ensure program runs from flash memory	Section 3.12
VCAP Configuration	A 4.7 μF bypass capacitor must be placed in series with the VCAP pin to stabilise the internal voltage of the chip.	Figure 2

Table 5.1.3.3

The NCP datasheet [30] also recommends configuration for stable voltage regulation:

Pin/Feature	**Recommended Practice and Implementation**	**Datasheet Section**
VDD pins	Connect 10 μF capacitors in parallel to both the input and output lines of the voltage regulator for voltage stability.	Figure 1

Table 5.1.3.4

Loop filter capacitor values were calculated using PLLatinum Sim, designed for the LMX2592. Initially, the “Low Spur” option was mistakenly selected instead of “Optimize Jitter,” an error corrected in the second iteration. Thus, a third-order filter was chosen for its optimal phase noise performance under the “Low Spur” option (see Appendix section 1.3). The simulated loop filter values were: $C: 1.5, 33, 33\ nF$ and $R: 120, 330\ \Omega$. The SPI interface was set up by routing STM32’s PB12, PB13, and PB15 to the LMX’s chip select (CS), serial clock (SCLK), and MOSI pins, respectively, ensuring accurate communication. After completing the schematic and layout an Electrical Connectivity Check confirmed that no nets were left floating. The design was then ready to export to the PCB editor to begin the PCB design process.

5.2 PCB Design

High-performance gigahertz systems demand meticulously designed PCBs for maintaining signal integrity, controlling parasitic effects, and ensuring robust power distribution. This section outlines the PCB design for the MFS using principles from Chapter 3: controlled trace impedance, optimized layer stack-up, strategic component placement, and effective decoupling. Annotated layout screenshots further link theory with practice.

5.2.1 Layout and Routing Considerations

Maintaining a consistent 50-ohm characteristic impedance is critical in RF and high-speed designs, ensuring compatibility with RF cables and SMA connectors [25]. Using Altium Designer's trace width calculator with the PCB's dielectric constant, trace thickness, etc., it was determined that 6.22 mils is optimal for signal traces, minimizing reflections and signal degradation [32].

Different traces connecting different nets have specific requirements:

- Power Traces: Via-to-power plane connections use a wider width of 10 mils for improved current carrying capacity and reduced voltage drop.
- Differential Pairs: Calculated separately, differential pairs were designed with a 5.7 mil trace width to ensure minimal coupling losses [25], [27].

These layout and routing decisions create a robust, high-performance PCB capable of operating reliably at gigahertz frequencies. Next, layer stack-up – essential for controlled impedance and effective power distribution – is addressed.

5.2.2 Layer Stack-up

A four-layer stack-up was selected to balance performance and cost – more layers increase expense, while fewer compromise high-speed performance [22], [23]. The PCB uses FR-4 (dielectric constant 4.5) with 1 oz copper on the outer layers for reliable mechanical strength and high-frequency behaviour [33]. Using Altium Designer's Layer Stack Manager, each signal layer is optimized to maintain a 50 Ω characteristic impedance, preserving signal integrity [23], [33].

This design ensures mechanical balance to reduce warping and leverages closely coupled ground and power planes to form interplane capacitance for effective high-frequency decoupling. Additionally, the top signal layer is dedicated to critical high-frequency traces – minimizing impedance discontinuities – while the bottom layer accommodates lower-frequency or less sensitive signals [23], [25]. Each layer is detailed in Table 5.2.2.1 below.

Layer Position	Purpose	Benefits	Considerations
Top Signal Layer	High-speed signal routing layer	Provides uninterrupted routing for critical signals; minimal via transitions for impedance control	Prioritize high-frequency signals to avoid impedance discontinuities
Ground Plane	Reference plane/Return current path	Maintains low impedance for return currents; supports controlled impedance routing; enhances signal integrity	Must be continuous and closely coupled with the power plane
Power Plane	Power distribution	Provides stable voltage and creates interplane capacitance with the ground plane for decoupling at high frequencies	Ensure minimal noise and proper coupling with the ground plane
Bottom Signal Layer	Lower frequency or non-critical signal routing	Offers additional routing space; allows for separation of less critical signals from high-speed areas	May include longer trace lengths, so impedance matching is less critical

Table 5.2.2.1

5.2.3 Component Proximity and Placement

With the layer stack-up optimized, the focus shifts to component proximity and placement to minimize interference and ensure uniform signal paths across the board.

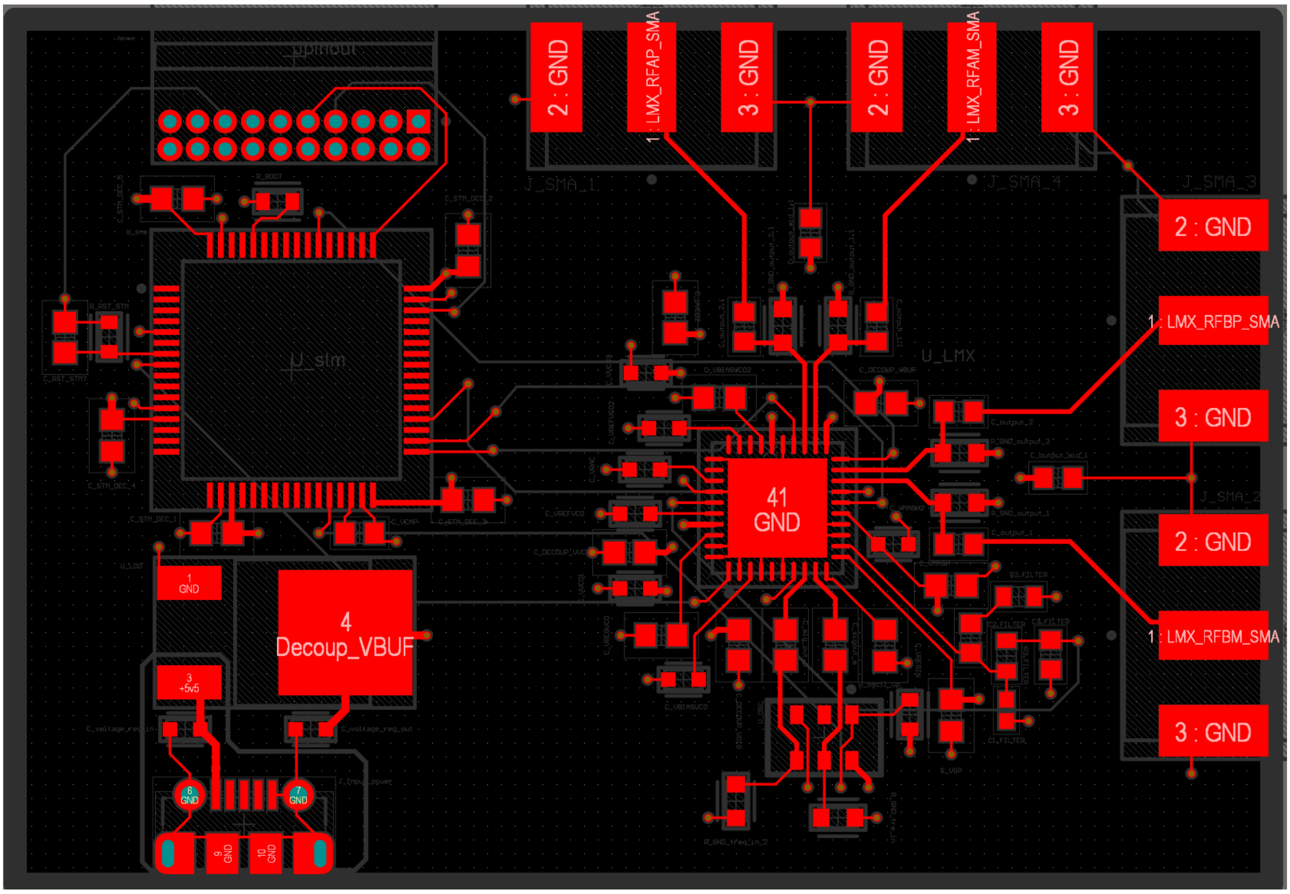


Figure 9: Top Layer PCB Architecture

Figure 9 shows the PCB's component placement strategy. The central LMX chip is placed near the board's centre to minimize trace lengths and interference. Its two differential outputs connect to four symmetrically arranged SMA connectors, ensuring equal trace lengths and minimal skew for optimal signal integrity.

The reference oscillator is positioned close to the LMX to shorten differential traces and reduce noise coupling. In contrast, the STM32 microcontroller and NCP regulator – operating at lower frequencies – are located near the USB power input and routed mainly on the bottom layer. The pin header is similarly placed to connect low-frequency signals without affecting high-speed routing. Decoupling capacitors, which are discussed next, are strategically placed close to each device's power pins.

5.2.4 Capacitor Placement (Tiered Decoupling)

A tiered decoupling strategy is implemented to reduce noise and parasitic effects in high-speed systems. As shown in Figure 9, small-value capacitors (targeting high-frequency noise) are placed closest to the device pins, while larger-value capacitors (addressing lower-frequency components) are positioned slightly farther away [22], [23]. This arrangement ensures effective decoupling across a broad frequency range, even as capacitor behaviour shifts at higher frequencies [24]. The next step is to optimize the via and grounding strategy for robust power distribution.

5.2.5 Vias and grounding strategy

The via diameter was kept small (16 mils) to reduce parasitic inductance, and each component has its own through-hole via connection to the ground or power plane, minimizing shared impedance, capacitance, and inductance. Ground polygon pours on both board sides further enhance low impedance return paths. Additionally, as shown in the bottom left of Figure 7, a power plane separation was implemented. The +5 V pad of the NCP regulator and the micro-USB connector are isolated within a dedicated 5 V section with rounded corners to minimize noise coupling into the 3.3 V domain. With the grounding strategy finalized, focus is shifted to configuring the SPI communication protocol for precise control between the microcontroller and the MFS.

5.2.6 SPI Communication Protocol

As discussed in Chapter 4, SPI was selected for its high-speed data transmission, precision control, and simplicity. The LMX datasheet [27] confirms SPI as the compatible protocol. The following flow diagram in Figure 10 illustrates how the STM32 updates the LMX's register values; see Appendix section 1.4 for details.

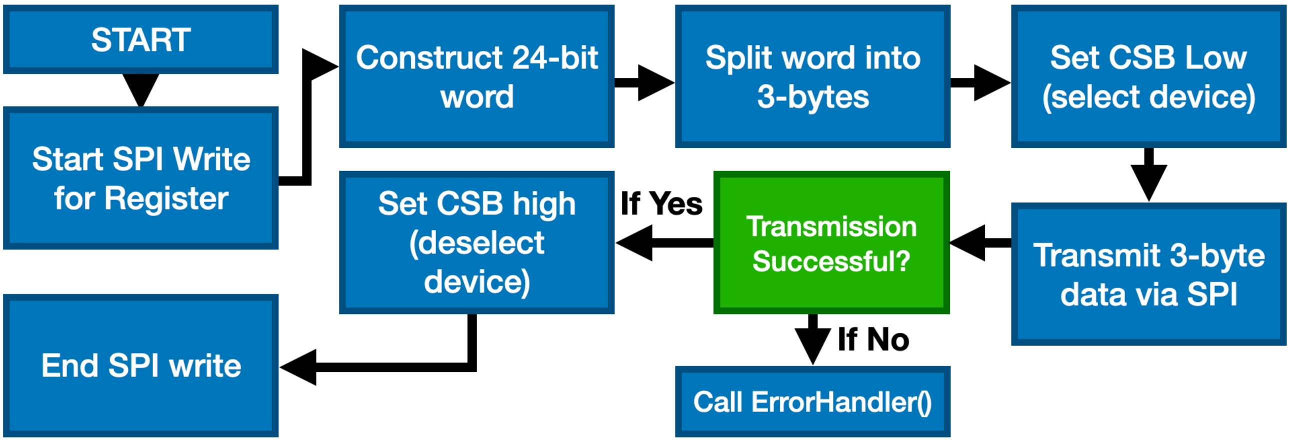


Figure 10: SPI Protocol Flow Chart

Specific registers in the LMX chip dictate its performance. Texas Instruments' "TCIS Pro" software calculates the register adjustments for frequency switching. This paper's software stores these configurations in look-up tables, allowing users to enter a desired frequency via a user-friendly interface for accurate system output. For details, see Appendix Section 1.5.

Together, these design choices establish a robust foundation for the PCB design, ensuring optimal system performance through careful layout, controlled impedance, and strategic component placement.

5.3 Hardware Measurement Methodology

After finalizing the design and manufacturing the PCB, the synthesizer's performance was benchmarked using the *Agilent Technologies CXA Signal Analyser N9000A (9 kHz – 26.5 GHz)* [34]. Phase noise data was collected at carrier frequencies of 1 GHz, 3 GHz, and 5 GHz, with noise measured at offsets from 100 Hz to 10 MHz to capture a wide range of performance.

5.3.1 Measurement Setup and Parameters

To measure phase noise, the spectrum analyser is centred on the chosen output carrier frequency, and the noise is recorded at multiple offsets (e.g., 100 Hz, 1000 Hz etc.). The noise is the difference in power level (dB) between the carrier peak and the power at the offset frequency from the carrier. Two main parameters govern the measurement [35]:

- Resolution Bandwidth (RBW): Defines the filter bandwidth over which noise power is integrated. A narrow RBW provides finer frequency resolution but can raise the noise floor by capturing less total noise power. A wide RBW integrates more noise power but may obscure subtle details in the noise profile.
- Video Bandwidth (VBW): Smooths rapid fluctuations in the displayed signal. Typically set equal to or slightly below the RBW for a stable trace without distorting the measurement.

Since phase noise is typically reported in dBc/Hz, the measured noise power must be normalized to a 1 Hz bandwidth. A sample measurement is illustrated below in Figure 11.

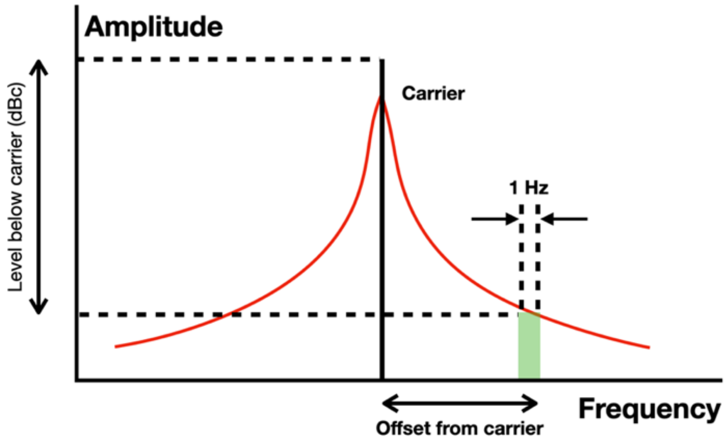


Figure 11: Phase noise measurement at frequency offset [35]

5.3.2 Phase Noise Calculation

The raw data from the analyser shows noise power across the selected RBW. To convert this into a normalized phase noise level, Equation 8 is used [35], [36]:

$$Phase\ Noise\ \left(\frac{dBc}{Hz}\right) = Measured\ Delta\ (dB) - 10 \times \log_{10}(RBW\ \times\ 1.2) \qquad (8)$$

In this equation: $10 \times \log_{10}(RBW)$ converts the noise power integrated over the RBW into a per-hertz value. An additional multiplier of 1.2 provides a correction that more accurately reflects the true phase noise of the synthesizer [35].

5.3.3 Why the Correction is Necessary

The 1.2 correction factor is crucial for addressing the inherent limitations and sampling behaviour of the spectrum analyser's ADC [35] :

- Discrete Sampling and Square-Pulse Response: The ADC samples the incoming signal in discrete steps, leading to a square-shaped measurement response rather than the ideal Gaussian distribution. The 1.2 factor approximates the difference in area between these two shapes, ensuring that the final phase noise values are neither under- nor overestimated.
- Spectrum Analyzer Resolution: Any analyser has finite resolution bandwidth and filtering effects that can distort the recorded noise profile. The correction factor helps account for these non-idealities so that measurements align more closely with real-world performance.

By applying this correction, a more precise view of the noise profile at each offset frequency is obtained, allowing consistent comparisons across different carriers (1 GHz, 3 GHz, 5 GHz) and paving the way for further jitter analysis, by integrating under the phase noise curve.

5.4 Jitter Analysis

In this section, the method used to measure the oscillator's timing jitter is detailed. Based on the methodology outlined by UCLA and the practical approach from DigiKey [37], [38], the process involves two main steps: integrating the phase noise over the frequency range of interest and converting that integrated value into an RMS jitter.

5.4.1 Integration Limits and Measurement

For the measurement, the phase noise from the carrier out to a 10 MHz offset on the upper sideband was recorded. In PLL systems, phase noise is typically symmetric around the carrier. Therefore, although only the upper sideband was measured, it can be assumed that the lower sideband contributes a similar amount of noise. The UCLA analysis recommends integrating over the full noise spectrum; for systems where both sidebands contribute equally, integrating from the carrier to the upper limit and then doubling the result provides a good approximation of the total noise power [37]. However, since the datasheet of the LMX [27] only considers the upper side band, this paper will not account for jitter across the entire spectrum, such that appropriate comparisons can be made.

5.4.2 Converting Integrated Phase Noise to RMS Jitter

Once the integrated phase noise (in rad^2) over the measured offset range is obtained, the RMS phase jitter in radians is given by [38]:

$$RMS\ Phase\ Jitter\ (rad) = \sqrt{2A} \tag{9}$$

where (A) is the integrated phase noise over the measured range (after converting the phase noise data from dBc/Hz to a linear scale). To convert the phase jitter to time jitter (in seconds), divide by the oscillator's angular frequency $(2\pi f_0)$ [38].

$$RMS\ Time\ Jitter\ (s) = \frac{\sqrt{2A}}{2\pi f_0} \tag{10}$$

Based on the measurements from the carrier to a 10 MHz offset on the upper sideband, the above equations can be used to estimate the total RMS jitter. This simplified method provides a reliable estimate of the oscillator's timing performance while keeping the analysis straightforward [37]. Note, integration was performed using Python's NumPy trapezoidal function. Discrepancies between the RMS jitter values computed in this paper and those from simulation may stem from differences in integration techniques, rather than from inherent physical limitations or improvements in the designed MFS architecture. After phase noise measurement, attention shifts to lock time – the speed at which the system stabilizes after a frequency change.

5.5 Lock Time Measurements

The lock time program is designed for simplicity and precision, using the STM32's 16 MHz internal timer [29] to achieve microsecond-level accuracy that meets the LMX datasheet requirements [27]. The measurement sequence begins with setting the frequency and continues until the feedback pin (MUXout) registers a high voltage, indicating lock (see Appendix Section 1.4). When measuring, the system transitions directly between adjacent frequencies (e.g., 3 GHz to 4 GHz), simulating real-world frequency hopping.

5.5.1 Limitations of the lock time measurement method

While the sequential measurement method offers a practical framework for evaluating PLL lock times during frequency hopping, it presents limitations. Direct frequency transitions without re-initializing the loop can leave residual offsets and partially settled loop conditions, potentially inflating the measured lock times by carrying over transient states from the previous frequency. The reliance on the MUXout signal as a lock indicator further complicates the evaluation, as this signal may not precisely reflect the moment the PLL achieves true phase and frequency alignment. Moreover, inherent microcontroller processing delays – including timer overhead and interrupt latency – can introduce timing inaccuracies that undermine the claimed microsecond-level precision. Finally, by focusing solely on the lock time metric, this method may overlook other critical dynamic behaviours such as transient overshoots, spurious signals, and the overall stability of the loop, particularly under rapid, multi-hop operational conditions [19], [36].

Chapter 6: Results

Phase noise is crucial for high-performance synthesizers and PLLs, directly affecting signal purity, timing accuracy, and overall reliability [35]. It quantifies short-term phase fluctuations, with elevated phase noise degrading communication quality, increasing bit error rates, and causing timing uncertainties. In this design, phase noise is a key metric for evaluating the cascaded PLL architecture: at lower offsets, flicker noise and reference quality dominate, while at higher offsets, the VCO's intrinsic noise floor and measurement limitations prevail [36], [39]. This chapter provides an in-depth analysis of measured phase noise at 5 GHz (with supporting data for 1 GHz and 3 GHz in Appendix Section 2.1. The obtained data is compared with simulated values which are taken from PLLatinum, which uses the LMX EVM board [40]. This sets the stage for linking phase noise performance to RMS jitter and lock time.

6.1 Measured Phase Noise

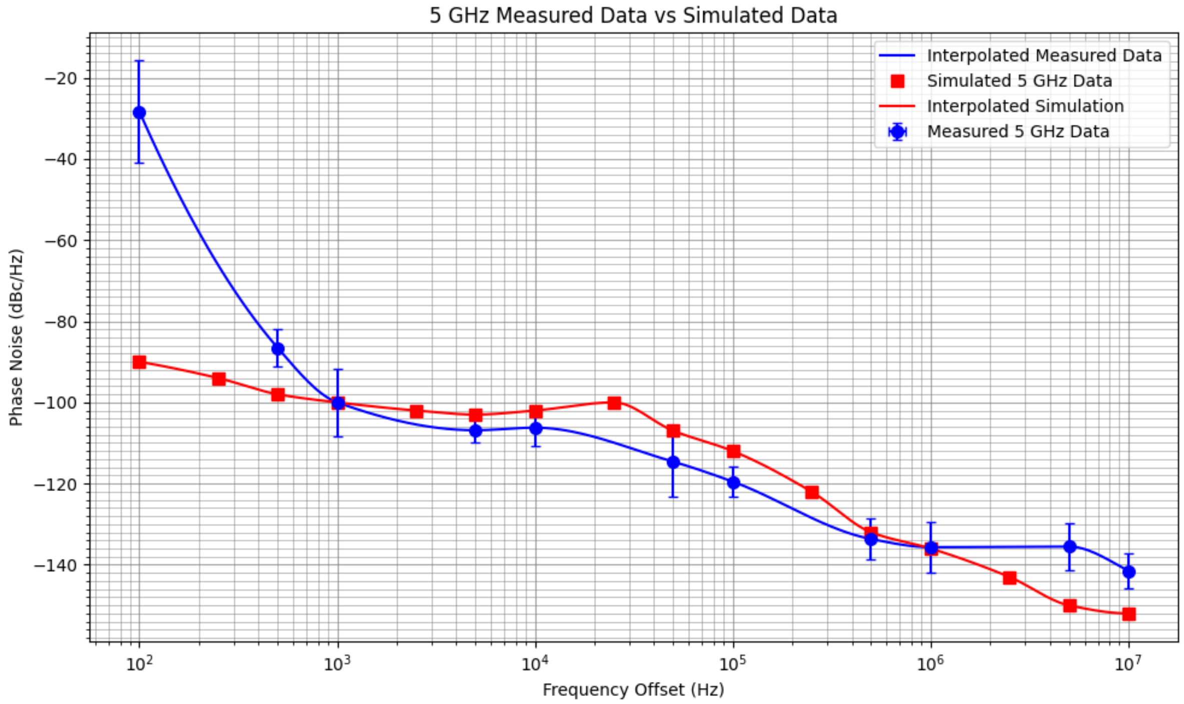


Figure 12: Phase Noise (dBc/Hz) as a function of frequency offset (Hz), 5 GHz

Figure 12 illustrates the phase noise profile of the MFS at a 5 GHz carrier, compared with the expected performance generated by the simulation tool (PLLatinum). For clarity, this section is divided into three parts corresponding to the different offset frequency regions: lower (≤1 kHz), mid (10 kHz to 1 MHz), and high (>1 MHz). Data from 1 GHz and 3 GHz measurements show similar trends and are discussed in Appendix 2.1. The following begins by examining the lower offset region, where the initial observations revealed discrepancies between measured and simulated data.

6.1.1 Lower Frequency Offsets

At offsets below 1 kHz, the measured phase noise was higher than anticipated as illustrated in Figure 10, suggesting that flicker noise or reference noise was dominating this region. A detailed review of the MFS design revealed that certain power supply decoupling capacitors were inadvertently omitted, leaving the reference signal effectively unfiltered. As a result, reference noise overwhelmed the loop filter within its bandwidth (~34 kHz) [40], [41]. For instance, according to the LMX EVM User guide [40], for a 100 MHz reference signal, the phase noise should be below –150 dBc/Hz at a 10 kHz offset; otherwise, the reference noise will significantly degrade the in-band phase noise at the RF output. In summary, the elevated phase noise at low offsets (-28 dBc/Hz when -95 dBc/Hz is expected) can be attributed to inadequate decoupling and unfiltered reference noise – a problem that can be remedied by proper capacitor placement and improved power supply design. The next section examines how the phase noise behaviour improves in the mid-frequency range.

6.1.2 Mid Frequency Offsets

From 10 kHz to 1 MHz, the measured phase noise closely tracked the simulated data, in fact demonstrating lower phase noise values than the simulation, as seen in Table 6.1.2.1, which summarizes the primary metrics below. This improvement is largely attributed to the cascaded PLL architecture, which appears to be highly effective in reducing phase noise within that range.

Data Type	Phase noise at offset frequencies (dBc/Hz)					
	100 Hz	1 kHz	10 kHz	100 kHz	1 MHz	10 MHz
Measured	-28.32	-99.99	-106.22	-119.52	-135.67	-142.52
Simulated	-90.6	-100.3	-102.4	-113.4	-136.4	-152.2

Table 6.1.2.1

Each empirical measurement is an average of five trials and carries a standard deviation uncertainty due the phase noise being located about a mean, but with additional minimum and maximum noise bands. Nevertheless, the measured data matches the simulation, indicating correct functionality. These findings reinforce the benefits of a cascaded PLL architecture, with performance gains at higher offset frequencies [10]. Having established that the mid-frequency region performs as expected, the discussion now turns to the high-offset region, where instrument limitations become evident.

6.1.3 High Frequency Offsets

An important consideration when analysing the data is the intrinsic noise floor of the spectrum analyser. For each major offset frequency, the analyser imposes a minimum noise level below which it cannot measure. This limitation is summarized in Table 6.1.3.1 below [34]:

Offset	Specification	Typical
1 kHz	–98 dBc/Hz	–103 dBc/Hz
10 kHz	–102 dBc/Hz	–110 dBc/Hz
100 kHz	–108 dBc/Hz	–110 dBc/Hz
1 MHz	–130 dBc/Hz	–130 dBc/Hz
10 MHz		–145 dBc/Hz nominal

Table 6.1.3.1

If the analyser's specified noise floor is higher (in absolute value) than the LMX's optimal simulated phase noise at a given offset, then any measured result near or below that limit reflects the analyser's noise floor rather than the synthesizer's true performance. The measurement is dominated by the instrument's limitation. For example, seen in Table 6.1.2.1, the measured phase noise at 100 kHz offset is -119.52 dBc/Hz, which is lower than the specified noise floor of the analyser. This could be a result of incorrect readings. Consequently, the board may be performing at or above simulation at high offsets, but that cannot be confirmed with the current measurement setup.

In summary, the measured phase noise across low, mid, and high offsets validates the cascaded PLL architecture but underscores the need for improved reference decoupling. Although some uncertainty arose from output frequency fluctuations, the results still align with simulations. At higher offsets, the analyser's noise floor limits visibility into the MFS's true performance. Building on these observations, the next sections explore how integrated phase noise translates into jitter and the PLL's lock time under various conditions.

6.2 Jitter Analysis

RMS jitter is a direct consequence of phase noise. As discussed in Section 5.4, RMS jitter is a key figure of interest for evaluating a phase noise profile because lower phase noise generally translates into lower RMS jitter. Table 6.2.1 summarizes the measured RMS jitter results and compares them with the corresponding simulated values.

Data Type	**RMS Jitter (Upper side band only)**		
	1 GHz	3 GHz	5 GHz
Measured	104200 fs	16620 fs	5628 fs
Simulated	83.51 fs	82.26 fs	95.87 fs

Table 6.2.1

As shown in Table 6.2.1, the RMS jitter values differ significantly in magnitude across frequencies. For example, the jitter at 1 GHz is measured at 104200 fs, while at 5 GHz it is 5628 fs. According to Equation 10, RMS jitter depends on the carrier frequency, scaling inversely with frequency. Consequently, higher carrier frequencies (e.g., 5 GHz) tend to exhibit lower jitter.

Since RMS jitter is calculated by integrating the phase noise over frequency, higher phase noise in the low-offset region increases the overall area under the phase noise curve, resulting in higher RMS jitter. This issue can be traced back to the omission of decoupling capacitors on the reference oscillator's power supply pins, which leads to increased noise in the lower offset range.

Another notable observation is the discrepancy between the measured and simulated RMS jitter values. This difference may stem from measurement limitations or the noise floor of the test equipment. Because RMS jitter is obtained by integrating phase noise over a given bandwidth, any residual or instrument noise can disproportionately affect the measurement – especially at higher frequencies where the actual jitter may be very low. In such cases, the noise floor of the measurement setup can mask the true performance of the reference oscillator [38].

6.3 Lock Time Analysis

As the final parameter of interest, Table 6.3.1 summarizes the measured lock times for the MFS during transitions between various frequencies. These measurements follow the sequential lock-time method described in Section 5.5, where an STM32's 16 MHz internal timer tracks the interval from issuing a frequency change command until the feedback pin (MUXout) indicates a locked state. Under this approach, transitioning directly from one frequency to the next can expose any residual loop conditions or noise-related effects that prolong settling.

Frequency	Measured Lock Time (μs)	Simulated Lock Time (μs)
1 GHz	956.3125	628.4
2 GHz	2754.4375	633.2
3 GHz	3299.3125	636.0
4 GHz	1914.0625	536.6
5 GHz	932.6875	746.9
6 GHz	1123.4375	640.8
7 GHz	4090.75	291.1
8 GHz	2319.3125	331.3
9 GHz	2276.9375	359.4

Table 6.3.1

These results reveal a broad range of lock times – from less than 1 ms at certain frequencies to over 4 ms in more challenging transitions – highlighting how larger hops or higher noise levels can significantly affect PLL acquisition. In particular, inadequate power supply decoupling on the reference oscillator increases the overall noise floor, forcing the PLL to continuously correct for rapid, unpredictable fluctuations. At low offset frequencies, this heightened noise is especially problematic, as the PLL's proportional-integral (PI) controller must work harder to maintain lock, leading to slower lock times. Conversely, as offset frequencies grow, the voltage-controlled oscillator (VCO) response becomes more dominant, partially mitigating the noise impact [36].

Overall, the interplay between sequential frequency transitions, reference noise, and loop filter dynamics underscores the importance of robust decoupling and clean reference signals. Even with microsecond-level timing precision, residual offsets from prior states and elevated noise can prolong lock times and degrade phase noise performance, demonstrating the value of careful design considerations in both the measurement setup and the PLL hardware itself.

Chapter 7: Architecture Optimisation

After analysing the results, it was clear that further optimizations were necessary to achieve or exceed simulated performance. The simulated values were obtained using a Texas Instruments evaluation board for the LMX, which is not fully optimized in its current configuration. Although the board has the potential for full optimization, its present setup limits performance. To fully exploit the cascaded PLL architecture, robust and optimized RF circuit design techniques must be applied. In addition to the essential inclusion of power supply decoupling capacitors for the reference oscillator, the following optimizations were implemented to maximize the LMX's performance.

7.1 Key Improvements

- **Vias Under Pads:** Capacitors are placed under vias to reduce trace inductance – improving signal integrity and reducing high-speed reflections.
- **Optimized Decoupling Capacitors:** Selected using impedance-frequency function to optimise filtering across a frequency span, reducing noise and enhancing power integrity.
- **2nd-Order Loop Filter:** Implemented a 2nd-order loop filter (verified with PLLatinum) for a more stable and responsive filtering effect compared to a 3rd-order design.
- **Back-of-Board Capacitors:** Located decoupling capacitors on the board's backside to minimize the path through vias, ensuring a cleaner power supply and very low inductance.
- **Dedicated Low-Noise LDO:** Introduced a separate Low Dropout Regulator for each major component on the board to ensure stable power distribution while minimizing noise.
- **Optimized Termination Resistor Placement:** Reoriented termination resistors on transmission lines for improved impedance matching and reduced signal reflections.
- **50-Ohm Impedance Matching for Vtune:** Employed 50-ohm impedance for the Vtune trace short to minimize signal degradation and reflections.
- **Differential Pair Trace Width Calculation:** Applied calculations based on impedance requirements to ensure proper signal propagation and improved integrity.
- **Curved Traces:** Used curved traces for high-speed signals to reduce reflections
- **Coplanar Waveguide for Differential Pairs:** Utilized coplanar waveguide structures to maintain controlled impedance and enhance high-speed signal integrity.

Aesthetic Improvements:

- **Pinout Header:** Improved layout for easier component accessibility.
- **Clearer Labelling:** Enhanced labels for better board readability and easier troubleshooting.
- **Logo and Name:** Added a personal logo and board name for clearer identification.
- **LEDs:** Integrated LEDs to provide immediate visual feedback on board status.

7.1.1 Top Layer Optimised Board

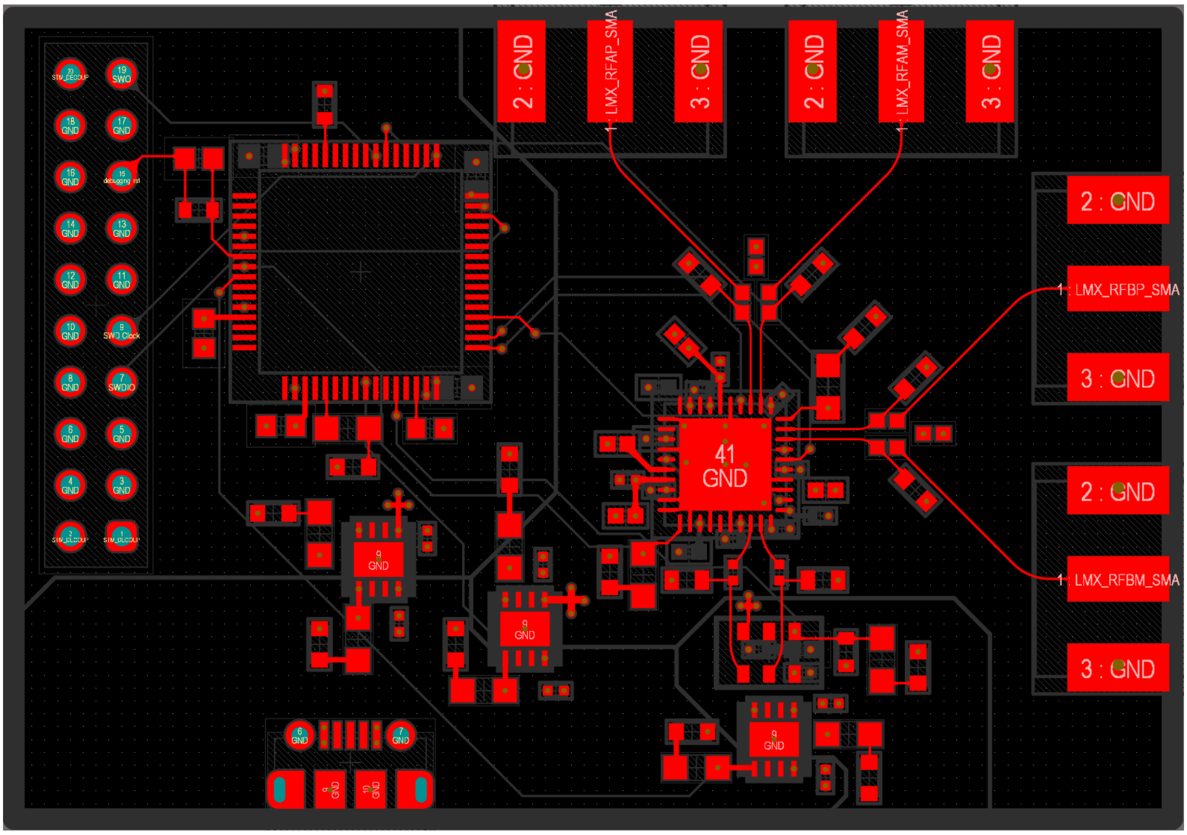


Figure 13: Top Layer Optimised Board

7.1.2 Bottom Layer Optimised Board

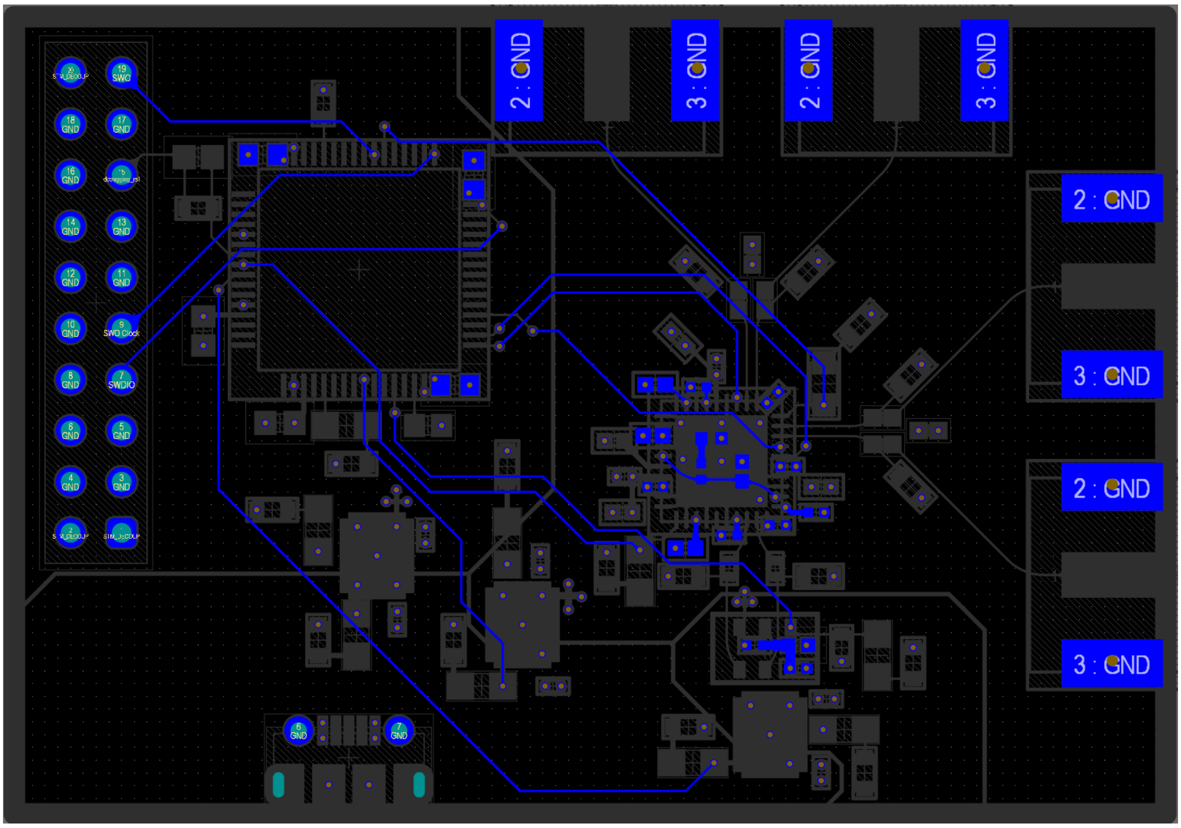


Figure 14: Bottom Layer Optimised Board

See Section 4 of the Appendix for all images relevant to the Optimised Board Design.

Chapter 8: Optimised Architecture Results

The procedure outlined in Section 5.3 was followed to obtain data for the second iteration of board and compared against the simulation. This is illustrated in Figure 13 below.

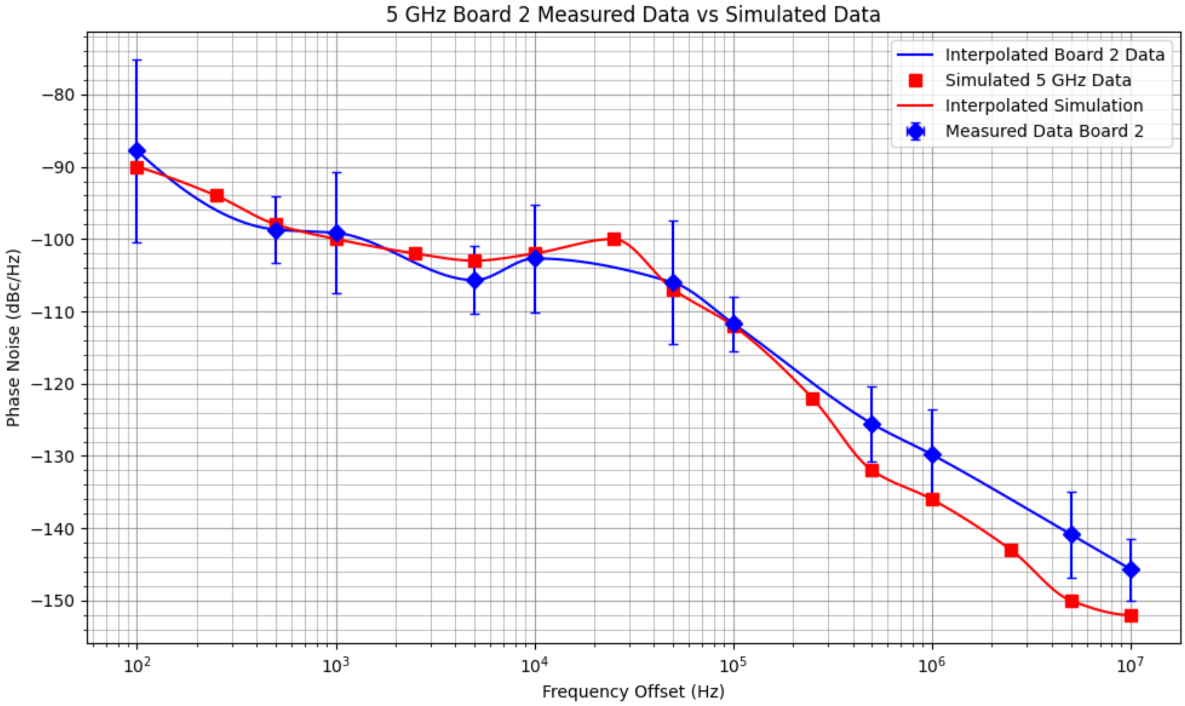


Figure 15: Second Iteration Board phase noise and Simulated Phase noise comparison

8.1 Phase Noise Analysis

At lower frequency offsets, the incorporation of well-placed power supply decoupling capacitors led to a marked reduction in phase noise, bringing the measured performance in line with simulation predictions. Moving into the mid-frequency range, the results become even more impressive – the second iteration's measured data clearly matches the simulated performance, though limited by the noise floor of the spectrum analyser. With a better measurement device, the second iteration of the board would surpass the simulation. Moreover, at higher frequency offsets, the measured performance does not exceed the simulation; instead, it reaches the intrinsic noise floor of the spectrum analyser (see Table 6.1.3.1). This suggests that the measurements in this region are limited by the analyser rather than reflecting any shortfall in the cascaded PLL structure itself. Overall, these findings confirm that the optimized board is a superior solution compared to existing market offerings, especially in the low to mid-frequency ranges where the enhancements are most apparent.

8.2 RMS Jitter Analysis

Additionally, the RMS jitter derived from the measured data is lower than that obtained from simulation. This improvement is largely attributable to the enhanced phase noise performance in the mid-frequency offset range of the second board iteration, which results in a smaller integrated noise area and, consequently, reduced RMS jitter. The differences are made clear in Table 8.2.1 below.

Data Type	**RMS Jitter (fs) (Upper side band only)**		
	1 GHz	3 GHz	5 GHz
Measured	78.30 fs	74.60 fs	88.35 fs
Simulated	83.51 fs	82.26 fs	95.87 fs

Table 8.2.1

8.3 Lock time Analysis

Frequency	**Measured Lock Time (μs)**	**Simulated Lock Time (μs)**
1 GHz	733.3522	628.4
2 GHz	680.3871	633.2
3 GHz	692.3457	636.0
4 GHz	595.2746	536.6
5 GHz	810.3678	746.9
6 GHz	674.4913	640.8
7 GHz	319.3042	291.1
8 GHz	412.1345	331.3
9 GHz	423.6902	359.4

Table 8.3.1

As shown in Table 8.3.1, the second board iteration achieved notably improved lock times, although they did not fully align with simulations. The LMX datasheet outlines further optimizations—such as programming FCAL_FAST and ACAL to reduce VCO calibration time, adjusting FCAL_CLK_DIV and ACAL_CMP_DLY for shorter lock intervals, and bypassing amplitude measurement via VCO_IND_AC. Additional registers like ACAL_VCO_IDAC_RANGE, ACAL_VCO_IDAC_START, and VCO_SEL_START, along with a wider loop bandwidth, can further enhance performance. However, because this paper prioritizes phase noise optimization over lock time, these measures were not implemented, though they are viable if needed [27].

Chapter 9: Conclusion

This paper set out to design, develop and implement a novel MFS with a cascaded PLL architecture and fully integrated software controlled, all encapsulated with a user-friendly interface. The design was geared toward achieving a balanced trade-off between low phase noise, rapid lock times, and cost-effective integration for 5G applications. The research spanned theoretical analysis, meticulous PCB design, and rigorous experimental validation, allowing for a comprehensive evaluation of the system's performance against both simulated models and benchmarks reported in the literature.

A comprehensive analysis and benchmarking of both the first and second iterations of the MFS reveal that the developed board not only outperforms Texas Instruments' EVM board [40] but also meets, and at times, exceeds industry and literature standards. It is important to note that the true low noise performance of the solution could not be fully quantified, as measurements were limited by the noise floor of the spectrum analyser rather than the inherent noise characteristics of the circuit. Table 9.1 presents the final findings for a clear comparison between the simulated values, the first iteration of the board, the second iteration of the board, and current literature values.

	Phase noise at offset frequencies (dBc/Hz)					
Board	100 Hz	1 kHz	10 kHz	100 kHz	1 MHz	10 MHz
Iteration 1	-28.32	-99.99	-106.22	-119.52	-135.67	-142.52
Iteration 2	-87.79	-99.16	-102.69	-111.69	-129.81	-145.67
Simulation	-90.6	-100.3	-102.4	-113.4	-136.4	-152.2
Literature	-90	-92	-106	-112	-125	-138

Table 9.1

(Literature values from [10] for best comparison as it employs a cascaded PLL architecture)

In summary, compared to established literature, the proposed design offers competitive phase noise performance and reduced jitter while maintaining a reasonable lock time. Although certain results are still influenced by measurement limitations (notably in the high frequency offset region due to the spectrum analyser's noise floor), the experimental data validate the hypothesis that a cascaded PLL architecture can achieve a well-balanced performance in a compact, cost-effective design.

Returning to the motivation outlined in the Section 1.1, the pursuit of achieving such a low phase noise has real world implications. As outlined MFS lie at the heart of any 5G systems, meaning optimisations in their performance are critical in enhancing the entire signal processing chain that follows. This paper verifies that a cost-effective, fully integrated cascaded PLL architecture enables MFSs to meet the robust performance demands of 5G and future 6G systems.

Looking ahead, the evolution of 6G networks will demand tighter integration, enhanced energy efficiency, and further miniaturization of RF components. Future research should explore advanced integration techniques, adaptive control strategies for dynamic network conditions, and novel materials to reduce interference and power loss. Additionally, AI-driven adaptive networks and innovative beamforming techniques present promising avenues for extending this work [42].

Thus, achieving low phase noise is not just a technical specification – it is a fundamental enabler that strengthens the entire communication system. The design and optimizations presented in this paper confirm that by employing a cascaded PLL architecture, it is possible to meet the rigorous performance standards required in today's and tomorrow's wireless communication networks.

References

[1] ‘Smartphone sales worldwide 2007-2023’, Statista. Accessed: Mar. 08, 2025. [Online]. Available: https://www.statista.com/statistics/263437/global-smartphone-sales-to-end-users-since-2007/

[2] P. Palagummi, ‘A Quarterly Publication of ACCS’, *ACCS*, vol. 2, no. 3, Dec. 2022, Accessed: Mar. 08, 2025. [Online]. Available: https://journal.accsindia.org/show.article.php?id=63

[3] V. Sritapan, ‘5G Security Evaluation Process Investigation: Version 1, May 2022’, CISA, 1, May 2022. Accessed: Mar. 08, 2025. [Online]. Available: https://www.cisa.gov/sites/default/files/publications/5G_Security_Evaluation_Process_Investigation_508c.pdf#:~:text=%E2%80%A2%205G%20RAN%20supports%20all,network%20supports%20interworking%20with%20other

[4] ETSI, ‘ETSI - Multi-access Edge Computing - Standards for MEC’, Multi-access Edge Computing (MEC). Accessed: Mar. 08, 2025. [Online]. Available: https://www.etsi.org/technologies/multi-access-edge-computing

[5] W. H. Huang Jianyi Zhou, Zhiqiang Yu, Lina Cao, Liang Ma, and Fei, ‘Transceiver Supports 8 × 8 MIMO Systems’, Microwaves & RF. Accessed: Mar. 08, 2025. [Online]. Available: https://www.mwrf.com/technologies/analog/mixed-signal/article/21845496/transceiver-supports-8-8-mimo-systems

[6] U. Rohde, *The design of modern microwave oscillators for wireless applications : theory and optimization*. Hoboken, NJ : J. Wiley, 2005.

[7] F. Pedersen, ‘5G spectrum for local industrial networks’. Accessed: Mar. 08, 2025. [Online]. Available: https://www.ericsson.com/en/reports-and-papers/white-papers/5g-spectrum-for-local-industrial-networks

[8] E. Dahlman, *5G/5G-advanced : the new generation wireless access technology*, Third edition. London, United Kingdom ; San Diego, CA : Academic Press, 2024.

[9] L. Kuai, W. Hong, J. Chen, and H. Zhou, ‘A Frequency Synthesizer for LO in Millimeter-wave 5G Massive MIMO System’, in *2019 IEEE Asia-Pacific Microwave Conference (APMC)*, Dec. 2019, pp. 1014–1016. doi: 10.1109/APMC46564.2019.9038807.

[10] W. El-Halwagy, A. Nag, P. Hisayasu, F. Aryanfar, P. Mousavi, and M. Hossain, ‘A 28-GHz Quadrature Fractional-N Frequency Synthesizer for 5G Transceivers With Less Than 100-fs Jitter Based on Cascaded PLL Architecture’, *IEEE Trans. Microw. Theory Tech.*, vol. 65, no. 2, pp. 396–413, Feb. 2017, doi: 10.1109/TMTT.2016.2647698.

[11] D.-Y. Huang, X.-W. Zhu, X.-L. Yang, L. Qiu, and S.-H. Yan, ‘Design of Frequency Synthesizer Based on PLL Architecture at X-band’, in *2024 International Conference on*

Microwave and Millimeter Wave Technology (ICMMT), May 2024, pp. 1–3. doi: 10.1109/ICMMT61774.2024.10672512.

[12] H. Peng *et al.*, ‘Ultra-Low Phase Noise and Frequency Agile X-Band Frequency Synthesizer Based on a Phase Locked Optoelectronic Oscillator’, in *2019 Joint Conference of the IEEE International Frequency Control Symposium and European Frequency and Time Forum (EFTF/IFC)*, Apr. 2019, pp. 1–3. doi: 10.1109/FCS.2019.8856046.

[13] ‘Understanding Direct Digital Synthesis (DDS)’. Accessed: Mar. 08, 2025. [Online]. Available: https://www.ni.com/en/shop/electronic-test-instrumentation/waveform-generators/understanding-direct-digital-synthesis--dds-.html

[14] S. Biswas and V. Revathi, ‘A fast-switching low-spurious 6–18 GHz hybrid frequency synthesizer’, in *2015 IEEE MTT-S International Microwave and RF Conference (IMaRC)*, Dec. 2015, pp. 312–315. doi: 10.1109/IMaRC.2015.7411404.

[15] ‘Single-Chip Direct Digital Synthesis vs. the Analog PLL | Analog Devices’. Accessed: Mar. 08, 2025. [Online]. Available: https://www.analog.com/en/resources/analog-dialogue/articles/dds-vs-analog-pll.html

[16] W. Chen and T. Xia, ‘A Dual-Band 28/38GHz Cascaded Phase Locked Loop Circuit Design’, in *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, Oct. 2020, pp. 1–5. doi: 10.1109/ISCAS45731.2020.9180886.

[17] jacquelineh96, ‘Selecting VCOs for Clock Timing Circuits – A System Perspective - Mini-Circuits Blog’. Accessed: Mar. 09, 2025. [Online]. Available: https://blog.minicircuits.com/selecting-vcos-for-clock-timing-circuits-a-system-perspective/

[18] ‘Phase-Locked Loops for Analog Signals | Zurich Instruments’. Accessed: Apr. 01, 2025. [Online]. Available: https://www.zhinst.com/de/resources/phase-locked-loops

[19] R. E. Best, *Phase-locked loops : design, simulation, and applications*, 4th ed. New York ; London : McGraw-Hill, 1999.

[20] ‘Phase-Locked Loop (PLL) Fundamentals | Analog Devices’. Accessed: Mar. 31, 2025. [Online]. Available: https://www.analog.com/en/resources/analog-dialogue/articles/phase-locked-loop-pll-fundamentals.html

[21] W. Li and J. Meiners, ‘Introduction to phase-locked loop system modeling’, 2000.

[22] A. Belous and V. Saladukha, *High-speed digital system design : art, science and experience*. Cham : Springer, 2020.

[23] H. W. Johnson and M. Graham, *High-speed digital design : a handbook of black magic*. Upper Saddle River, N.J. : PTR Prentice Hall, 1993.

[24] ‘ksim’. Accessed: Apr. 01, 2025. [Online]. Available: https://ksim3.kemet.com/capacitor-simulation

[25] D. Pozar, *Microwave engineering*, 3rd ed. New York ; Chichester : Wiley, 2005.

[26] A. Stoicescu, ‘Getting Started with Serial Peripheral Interface (SPI)’, 2021.

[27] ‘LMX2592 High Performance, Wideband PLLatinum^{TM} RF Synthesizer With Integrated VCO’. Texas Instruments, Aug. 2022. [Online]. Available: https://www.ti.com/lit/ds/symlink/lmx2592.pdf?ts=1741522499622&ref_url=https%253A%252F%252Fwww.mouser.com%252F

[28] ‘LMK62XX High-Performance Low Jitter Oscillator’. Texas Instruments, Dec. 2023. [Online]. Available: https://www.ti.com/lit/ds/symlink/lmk62e2-100m.pdf?ts=1741632639259

[29] ‘STM32F401xD STM32F401xE’. STMicroelectronics, Jan. 2025. [Online]. Available: https://www.st.com/resource/en/datasheet/stm32f401re.pdf

[30] ‘NCP1117LP’. On Semiconductor, Mar. 2014. [Online]. Available: https://docs.rs-online.com/1c68/0900766b812e2781.pdf

[31] ‘UM1724 User manual’. STMicroelectronics, Aug. 2020. [Online]. Available: https://www.st.com/resource/en/user_manual/um1724-stm32-nucleo64-boards-mb1136-stmicroelectronics.pdf

[32] Z. Peterson, ‘Impedance Control: How to Specify Your Requirements for PCB Manufacturers’, Altium. Accessed: Mar. 11, 2025. [Online]. Available: https://resources.altium.com/p/pcb-manufacturing-and-impedance-control-how-specify-your-requirements

[33] ‘Defining the Layer Stack’, Altium Documentation. Accessed: Mar. 11, 2025. [Online]. Available: https://www.altium.com/documentation/altium-designer/defining-layer-stack

[34] ‘CXA X-Series Signal Analyzer N9000A Datasheet’. Keysight Technologies, Apr. 09, 2018.

[35] P. Denisowski, ‘Understanding Phase-Noise Measurement Techniques’, Microwaves & RF. Accessed: Mar. 20, 2025. [Online]. Available: https://www.mwrf.com/technologies/test-measurement/article/21268395/rohde-schwarz-understanding-phase-noise-measurement-techniques

[36] F. M. Gardner, *Phaselock techniques*, 3rd ed. Hoboken, N.J. : Wiley-Interscience, 2005.

[37] Y. Zhao and B. Razavi, ‘Phase Noise Integration Limits for Jitter Calculation’, in *2022 IEEE International Symposium on Circuits and Systems (ISCAS)*, Austin, TX, USA: IEEE, May 2022, pp. 1005–1008. doi: 10.1109/ISCAS48785.2022.9937231.

[38] ‘Converting Oscillator Phase Noise to Time Jitter’, DigiKey. Accessed: Mar. 21, 2025. [Online]. Available: https://www.digikey.co.uk/en/articles/converting-oscillator-phase-noise-to-time-jitter

[39] A. Hajimiri, ‘A General Theory of Phase Noise in Electrical Oscillators’.

[40] ‘LMX2592EVM High Performance, Wideband PLLatinum^{TM} RF Synthesizer Evaluation Board Operating Instructions User’s Guide’. Texas Instruments, Dec. 2015.

[41] R. Woogeun and Z. Yu, *Phase-Locked Loops: System Perspectives and Circuit Design Aspects*, 1st edition. Newark: Wiley, 2023.

[42] H. Tataria, M. Shafi, A. F. Molisch, M. Dohler, H. Sjöland, and F. Tufvesson, ‘6G Wireless Systems: Vision, Requirements, Challenges, Insights, and Opportunities’, *Proc. IEEE*, vol. 109, no. 7, pp. 1166–1199, Jul. 2021, doi: 10.1109/JPROC.2021.3061701.

Appendix

1.1 Schematic Designs Board 1 Page 1

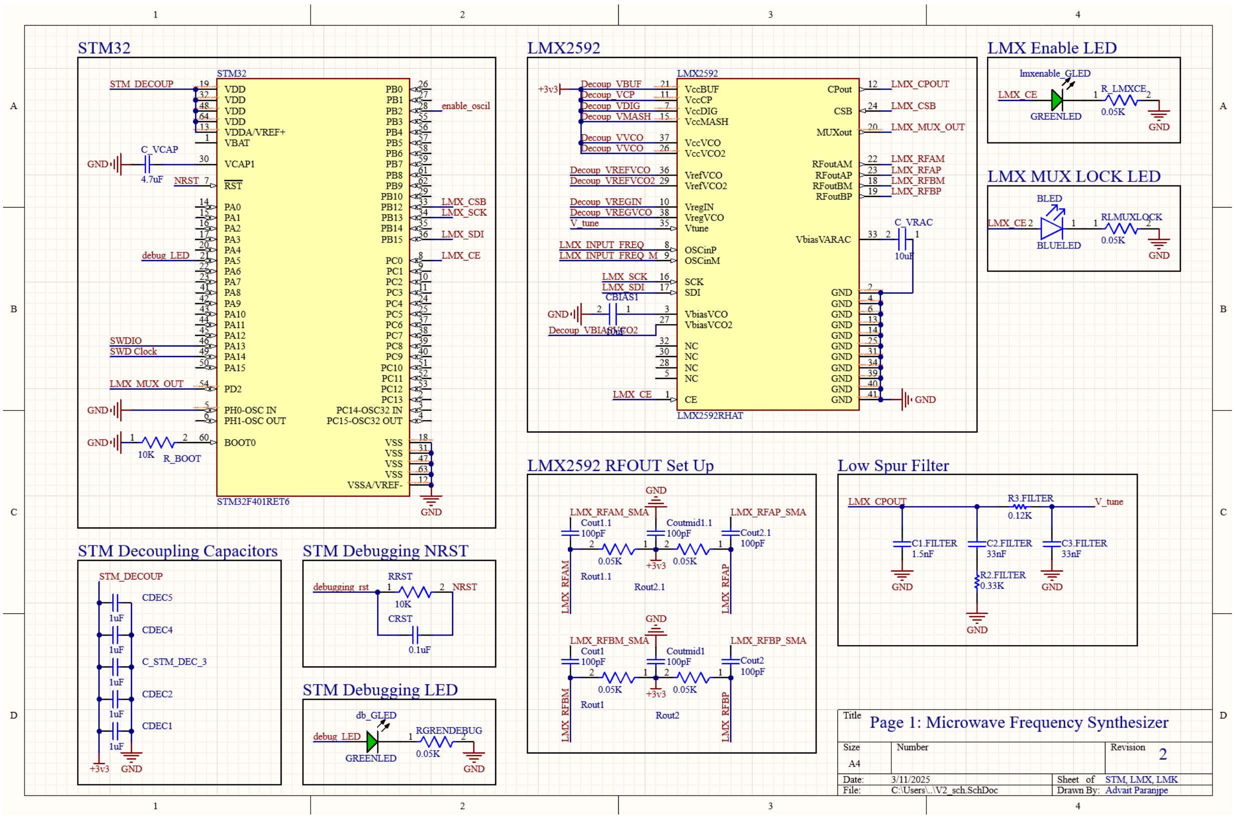


Figure 16: Architecture Schematic Page 1

1.2 Schematic Design Board 1 Page 2

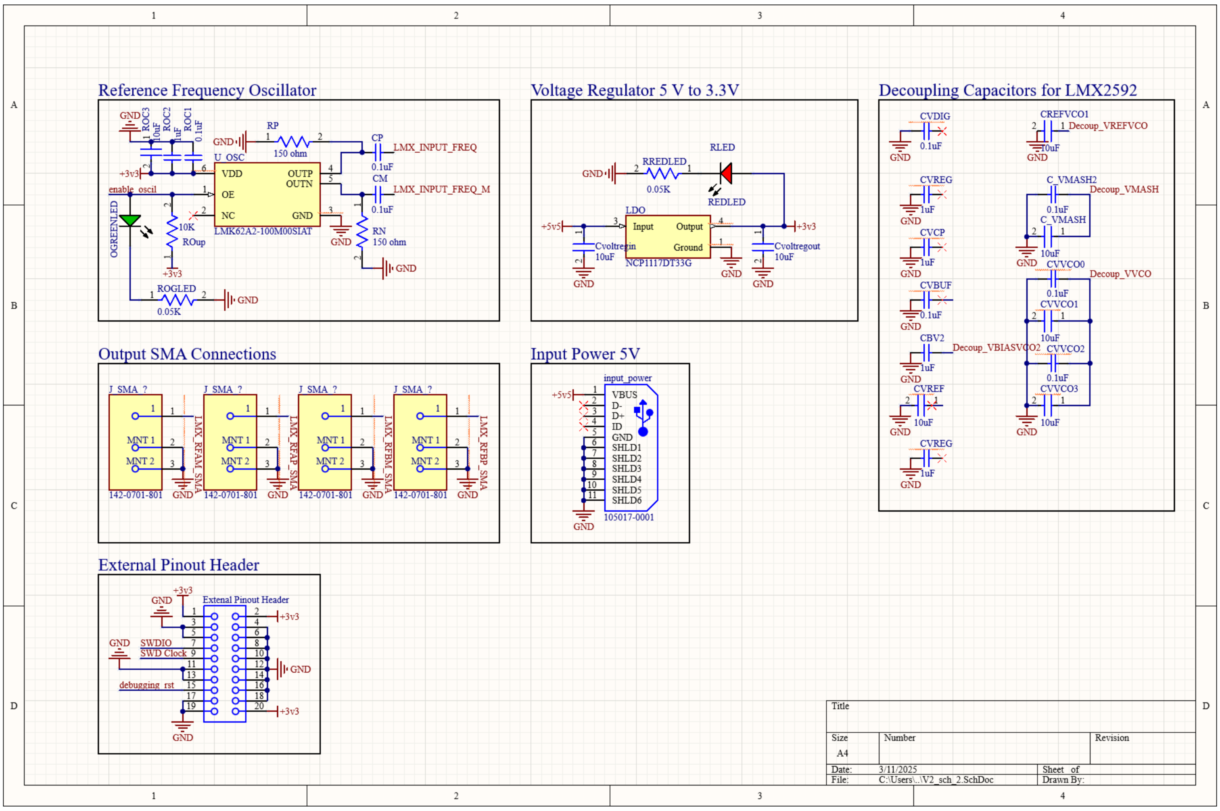
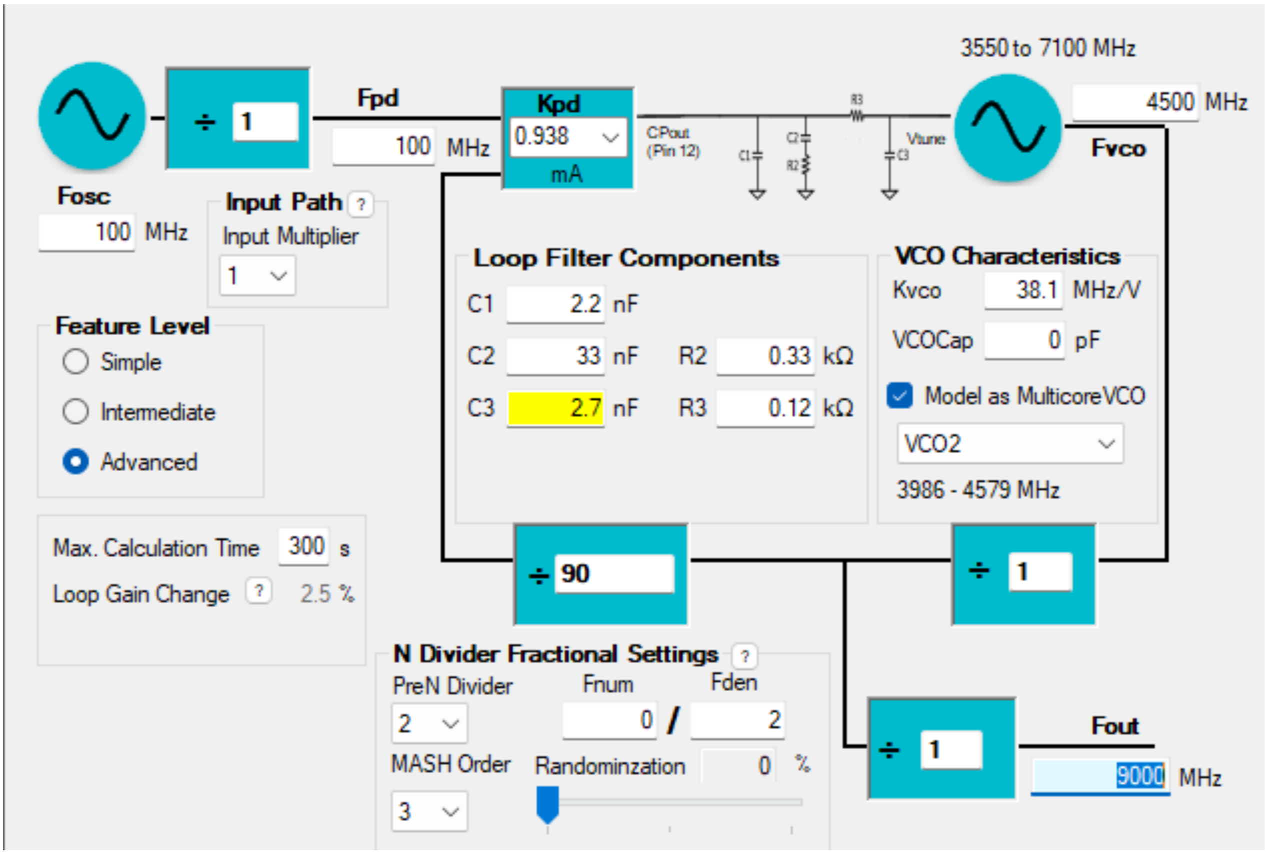


Figure 17: Architecture Schematic Page 2

1.3 Optimise Spurs Loop Filter Setup



Fosc 100 MHz

÷ 1

Input Path
Input Multiplier 1

Fpd 100 MHz

Kpd 0.938 mA

Fvco 4500 MHz

Feature Level
○ Simple
○ Intermediate
● Advanced

Max. Calculation Time 300 s
Loop Gain Change 2.5 %

Loop Filter Components

C1	2.2	nF			
C2	33	nF	R2	0.33	kΩ
C3	2.7	nF	R3	0.12	kΩ

VCO Characteristics
Kvco 38.1 MHz/V
VCOCap 0 pF
☑ Model as MulticoreVCO
VCO2
3986 - 4579 MHz

÷ 90

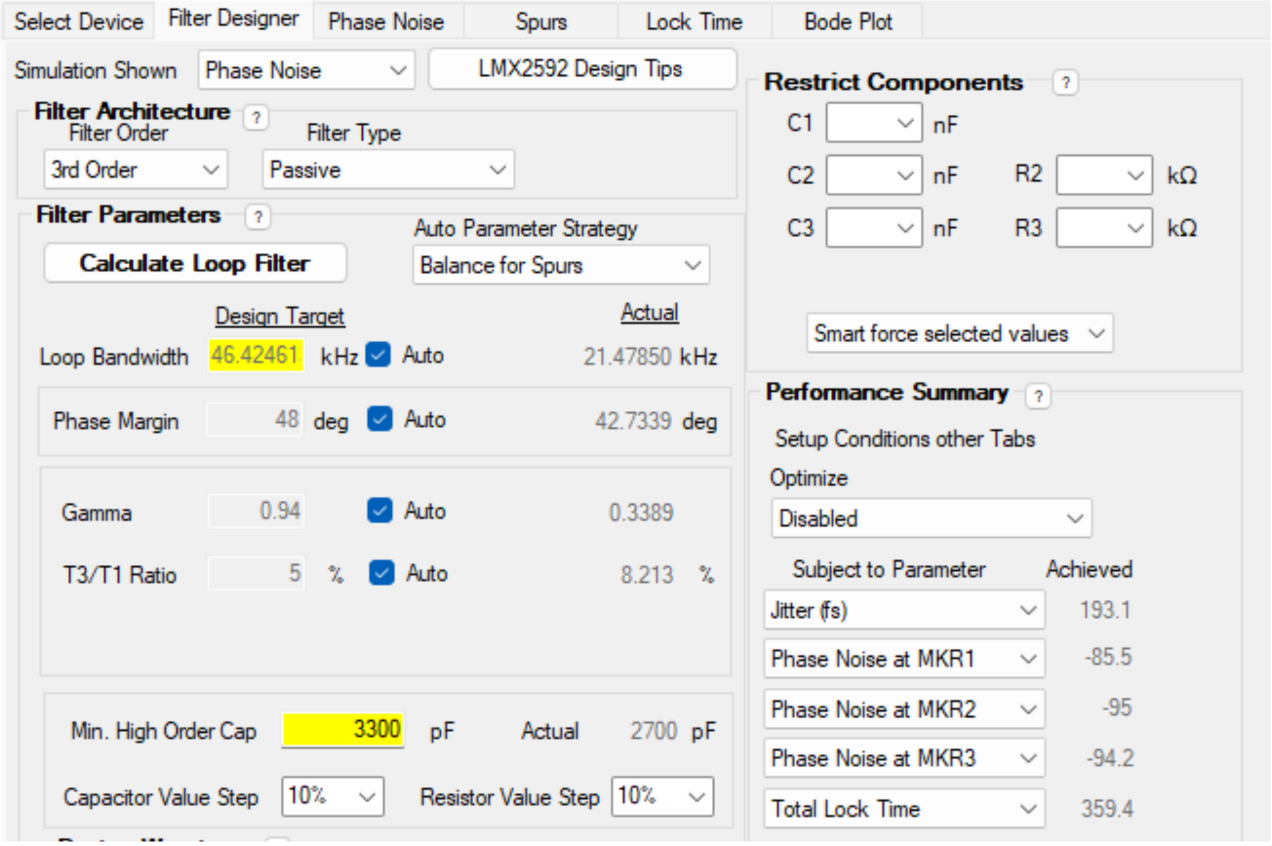
÷ 1

N Divider Fractional Settings
PreN Divider 2
Fnum 0 / Fden 2
MASH Order 3
Randominzation 0 %

÷ 1

Fout 9000 MHz

Figure 18: Capacitor and Resistor values for 3rd Order Low Spur Filter



Simulation Shown: Phase Noise

Filter Architecture
Filter Order: 3rd Order
Filter Type: Passive

Filter Parameters
Auto Parameter Strategy: Balance for Spurs

	Design Target			Actual
Loop Bandwidth	46.42461	kHz	☑ Auto	21.47850 kHz
Phase Margin	48	deg	☑ Auto	42.7339 deg
Gamma	0.94		☑ Auto	0.3389
T3/T1 Ratio	5	%	☑ Auto	8.213 %

Min. High Order Cap 3300 pF Actual 2700 pF
Capacitor Value Step 10% Resistor Value Step 10%

Restrict Components
C1 nF
C2 nF R2 kΩ
C3 nF R3 kΩ
Smart force selected values

Performance Summary
Setup Conditions other Tabs
Optimize: Disabled

Subject to Parameter	Achieved
Jitter (fs)	193.1
Phase Noise at MKR1	-85.5
Phase Noise at MKR2	-95
Phase Noise at MKR3	-94.2
Total Lock Time	359.4

Figure 19: Loop bandwidth and balance for spurs option selected

1.4 SPI Communication Protocol Program and Lock time measurement Program

https://github.com/advaitParanjpe/microwave_frequency_synthesizer.git

----

CustomBoard V3 - LMX2592 Initialization and PLL Lock Time Measurement

----

This repository contains two example implementations for interfacing with the LMX2592 frequency synthesizer on a custom board based on the STM32F4 microcontroller series. The projects demonstrate the following features:

1. Basic LMX2592 Initialization:
   - Programs the LMX2592 registers via SPI using values exported from TICS Pro.
   - Configures key control signals such as chip enable (CE), chip select (CSB), and the reference oscillator enable.

2. Advanced Initialization with PLL Lock Time Measurement and SWO Debug Output:
   - Extends the basic initialization with:
     • SWO (Single Wire Output) redirection of printf via ITM_SendChar.
     • High-resolution timer (TIM1) configuration to measure the PLL lock time.
     • External interrupt (EXTI) on PD2 (MUXOUT) to capture the moment when the PLL locks.
   - After the PLL lock is detected, the captured timer counter value (in ticks) is sent over the SWO for debugging purposes.

Table of Contents

• License and Disclaimer

Hardware Requirements

• Microcontroller: An STM32F4 series MCU (e.g., STM32F401xE).

• Frequency Synthesizer: LMX2592.

• CustomBoard V3 Connections:

- SPI2: Used to program the LMX2592.
- LMX2592 Control Signals:
  • Chip Enable (CE): PC0.
  • Chip Select (CSB): PB12.
  • Reference Oscillator Enable: PB2.
- PLL Lock Indicator (MUXOUT): PD2 (used in the advanced example).
- SWO Debug Output: ITM is used to redirect printf messages.
- Additional GPIOs: For optional LEDs and status monitoring (e.g., LED on PA2).

Software Requirements

• IDE: STM32CubeIDE (or an equivalent tool that supports STM32 development).

• Compiler: ARM GCC toolchain (arm-none-eabi-gcc).

• Libraries: STM32 HAL libraries.

• Debug/Programmer: ST-Link.

Project Structure

• main.c:

Contains the application code with two variants:

- Basic Version: Initializes the LMX2592 with exported register values.
- Advanced Version: In addition to LMX2592 initialization, includes:
  • SWO retargeting of printf.
  • Timer (TIM1) initialization.
  • PLL lock time measurement via an EXTI interrupt on PD2.

• Peripheral Initialization Routines:

 - SPI (MX_SPI2_Init): Sets up the SPI interface for LMX2592 communication.

 - GPIO (MX_GPIO_Init): Configures control signals and LED outputs.

 - Timer (MX_TIM1_Init): Configures TIM1 for high-resolution timing (advanced version).

Build and Flash Instructions

----

1. Import the Project:

 - Open the project in STM32CubeIDE and verify that the target MCU (e.g., STM32F401xE) is correctly selected.

2. Build the Project:

 - Build the project using the "Build All" command. Ensure that no stray text or extra comments interfere with the build.

3. Flash the Firmware:

 - Connect your ST-Link debugger.

 - Verify that all hardware connections (SPI, SWD, power, etc.) are secure.

 - Program the device using the ST-Link utility or via STM32CubeIDE.

4. Debug and Monitor:

 - For the advanced version, open the SWO viewer (or ITM console) to see the printf output showing the PLL lock time in timer ticks (and optionally, the converted time).

Detailed Code Explanation

LMX2592 Initialization:

• SPI Write Function:

The function LMX2592_WriteReg(uint8_t address, uint16_t data) formats a 24-bit write command (1 control bit + 7-bit address + 16-bit data) and transmits it over SPI2.

• Register Settings Array:

The array LMX2592_RegSetting initRegs[] holds register addresses and corresponding data settings exported from TICS Pro for a 30 MHz output configuration.

- Initialization Routine:
  The function LMX2592_Init_30MHz() enables the device via the CE pin (PC0) and programs the LMX2592 registers sequentially with a small delay between writes.

PLL Lock Time Measurement:

- Timer Configuration (TIM1):
  TIM1 is set up in up-counting mode without a prescaler so that each tick provides high-resolution timing. With a 16 MHz clock, each tick is approximately 62.5 ns.

- External Interrupt (EXTI) on PD2:
  PD2 is configured as an input with an EXTI interrupt. When the LMX2592 indicates PLL lock (MUXOUT goes high), an interrupt is triggered.

- Interrupt Callback:
  In the HAL_GPIO_EXTI_Callback() function, the current counter value of TIM1 is captured and stored in pll_lock_time, and a flag (lock_captured) is set to indicate that the PLL has locked.

- SWO Debug Output:
  The standard library function printf is retargeted via ITM_SendChar so that the measured lock time can be output over the SWO interface for debugging.

---

Customization

---

- Register Values:
  If a different output frequency or configuration is required, update the values in the initRegs array in LMX2592_Init_30MHz().

- Timer Settings:
  Adjust TIM1 settings (e.g., prescaler, period) if a different timing resolution is needed or if the MCU clock configuration changes.

• GPIO and Peripheral Configurations:
  Modify the GPIO pin assignments in MX_GPIO_Init() to match any changes in the custom board hardware.

Troubleshooting

• Programming and Debugging Issues:
  - Verify wiring and connector integrity between the ST-Link and the target board.
  - Check that the MCU does not reconfigure SWD pins prematurely.
  - Ensure that the BOOT0 pin is set correctly so the MCU boots from main flash memory.

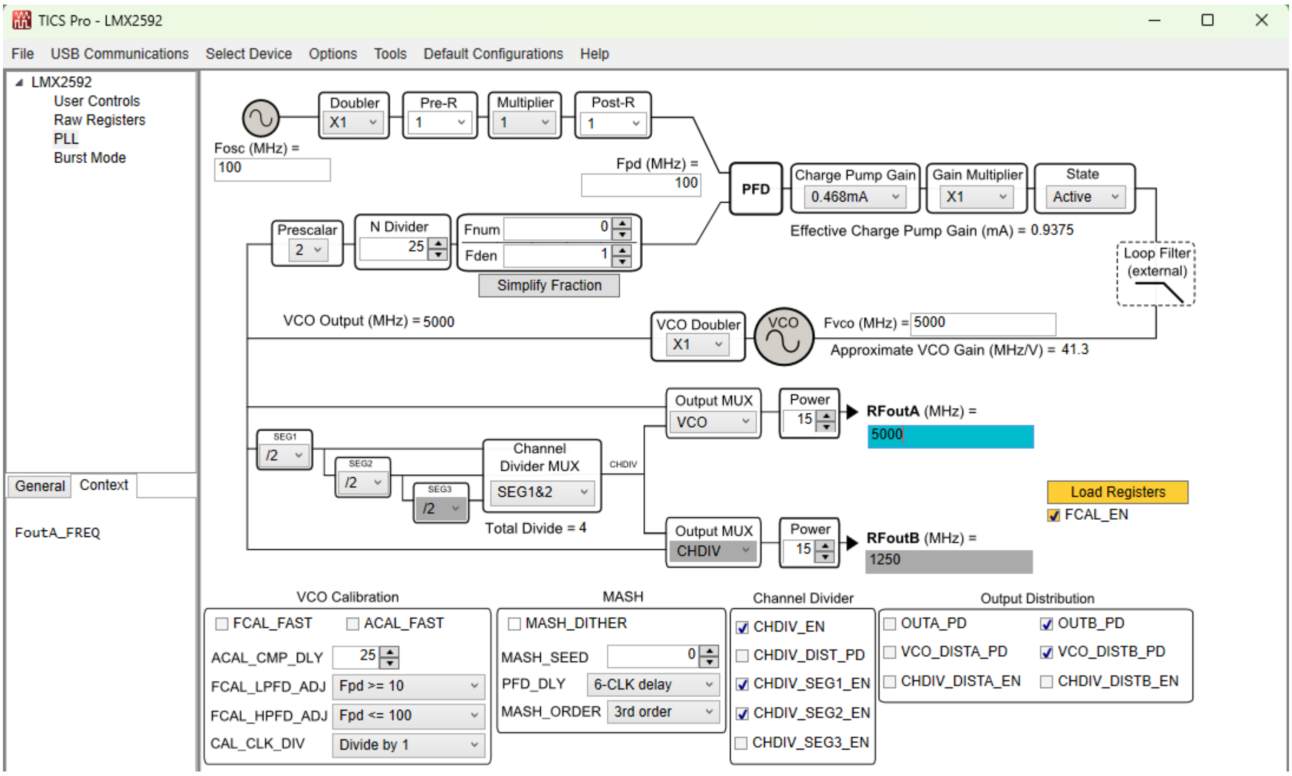
• PLL Lock Issues:
  - Ensure the LMX2592 register settings are correct for your desired frequency and that external components (oscillators, filters) are functioning properly.
  - Confirm that the MUXOUT pin is properly connected to PD2 and that the EXTI interrupt is enabled.

• SWO Output Missing:
  - Verify that SWO configuration is enabled in your debugger settings.
  - Ensure that the ITM stimulus port is active and properly connected.

License and Disclaimer

This code is provided by STMicroelectronics and is licensed under the terms stated in the LICENSE file in the root directory of this software component. If no LICENSE file is provided, the code is provided AS-IS.

1.5 LMX Register Update Program



Fosc (MHz) = 100
Doubler X1
Pre-R 1
Multiplier 1
Post-R 1
Fpd (MHz) = 100
Charge Pump Gain 0.468mA
Gain Multiplier X1
State Active
Effective Charge Pump Gain (mA) = 0.9375
Prescalar 2
N Divider 25
Fnum 0
Fden 1
VCO Output (MHz) = 5000
VCO Doubler X1
Fvco (MHz) = 5000
Approximate VCO Gain (MHz/V) = 41.3
Output MUX VCO
Power 15
RFoutA (MHz) = 5000
SEG1 /2
SEG2 /2
SEG3 /2
Channel Divider MUX SEG1&2
Total Divide = 4
Output MUX CHDIV
Power 15
RFoutB (MHz) = 1250
☑ FCAL_EN

VCO Calibration
☐ FCAL_FAST ☐ ACAL_FAST
ACAL_CMP_DLY 25
FCAL_LPFD_ADJ Fpd >= 10
FCAL_HPFD_ADJ Fpd <= 100
CAL_CLK_DIV Divide by 1

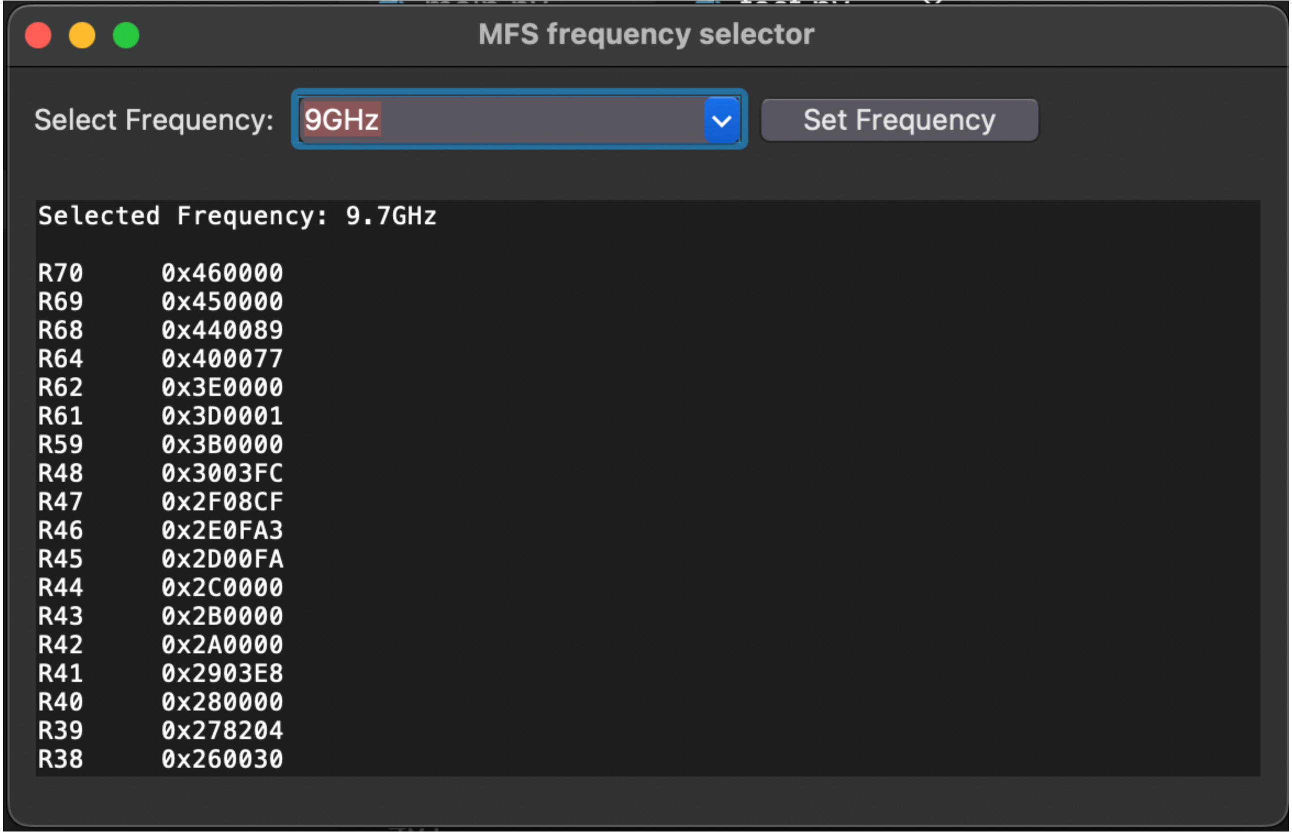
MASH
☐ MASH_DITHER
MASH_SEED 0
PFD_DLY 6-CLK delay
MASH_ORDER 3rd order

Channel Divider
☑ CHDIV_EN
☐ CHDIV_DIST_PD
☑ CHDIV_SEG1_EN
☑ CHDIV_SEG2_EN
☐ CHDIV_SEG3_EN

Output Distribution
☐ OUTA_PD ☑ OUTB_PD
☐ VCO_DISTA_PD ☑ VCO_DISTB_PD
☐ CHDIV_DISTA_EN ☐ CHDIV_DISTB_EN

Figure 20: Register value exports for 5 GHz setup

The link for the GUI: https://github.com/advaitParanjpe/microwave_frequency_synthesizer.git



Select Frequency: 9GHz

```
Selected Frequency: 9.7GHz

R70     0x460000
R69     0x450000
R68     0x440089
R64     0x400077
R62     0x3E0000
R61     0x3D0001
R59     0x3B0000
R48     0x3003FC
R47     0x2F08CF
R46     0x2E0FA3
R45     0x2D00FA
R44     0x2C0000
R43     0x2B0000
R42     0x2A0000
R41     0x2903E8
R40     0x280000
R39     0x278204
R38     0x260030
```

Figure 21: User-friendly GUI for frequency selection

2.1 Results for 1 GHz and 3 GHz

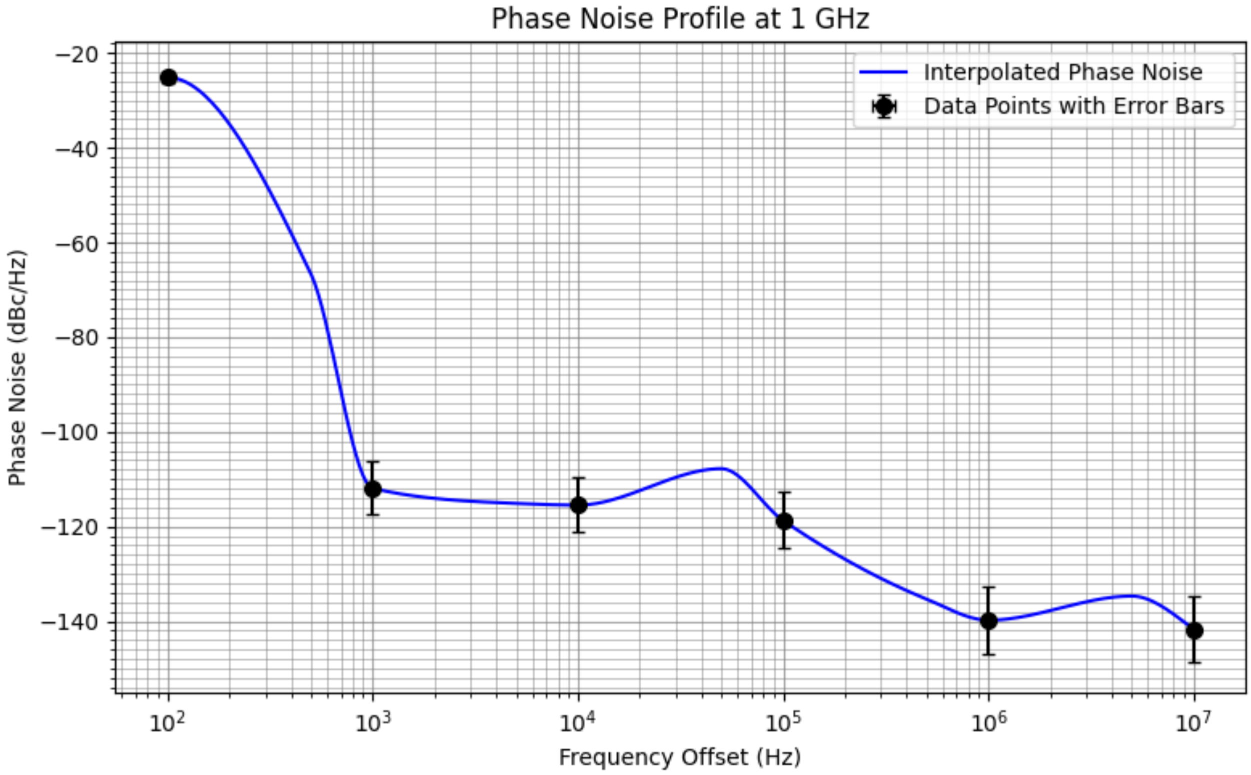


Figure 22: Phase noise profile at 1 GHz

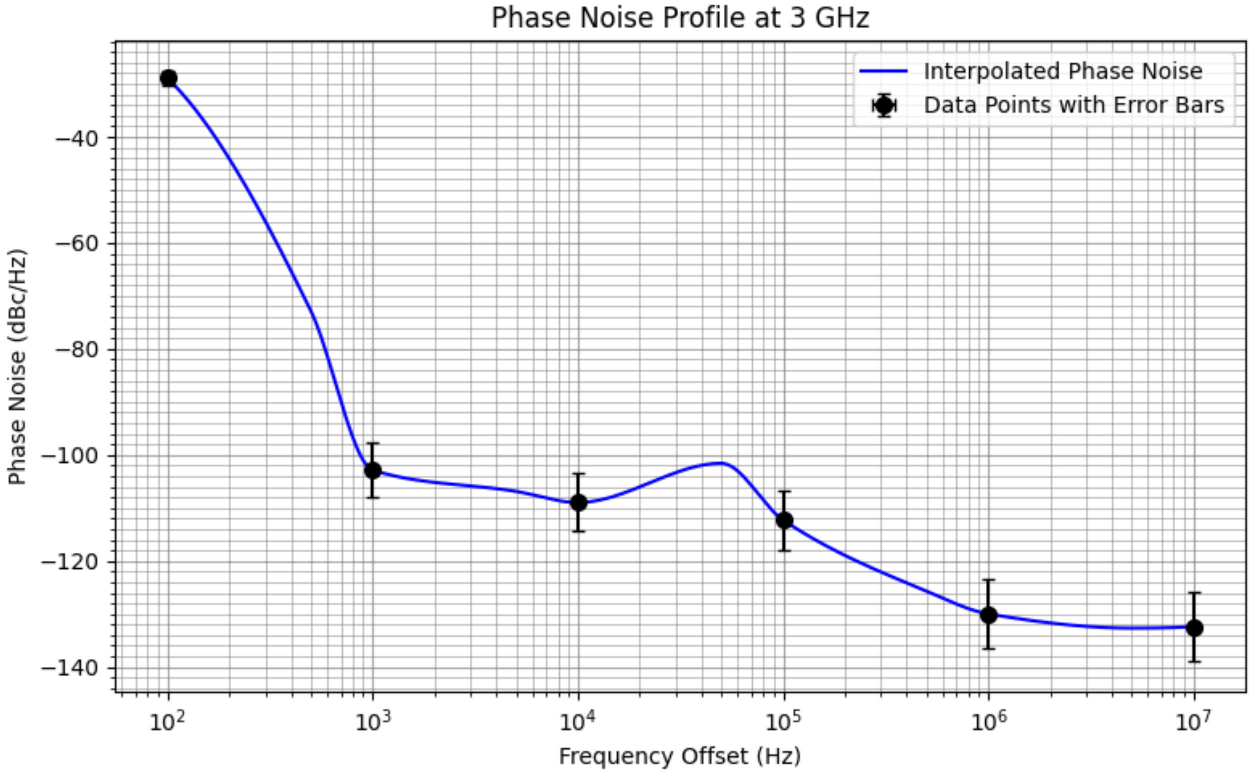


Figure 23: Phase Noise Profile at 3 GHz

Clearly, both 1 GHz and 3 GHz produce similar phase noise curves as the 5 GHz curve which is carefully analysed in Section 6.1 of the main report. Thus, only the 5 GHz phase noise profile was analysed.

3.1 Image of Board Iteration 1

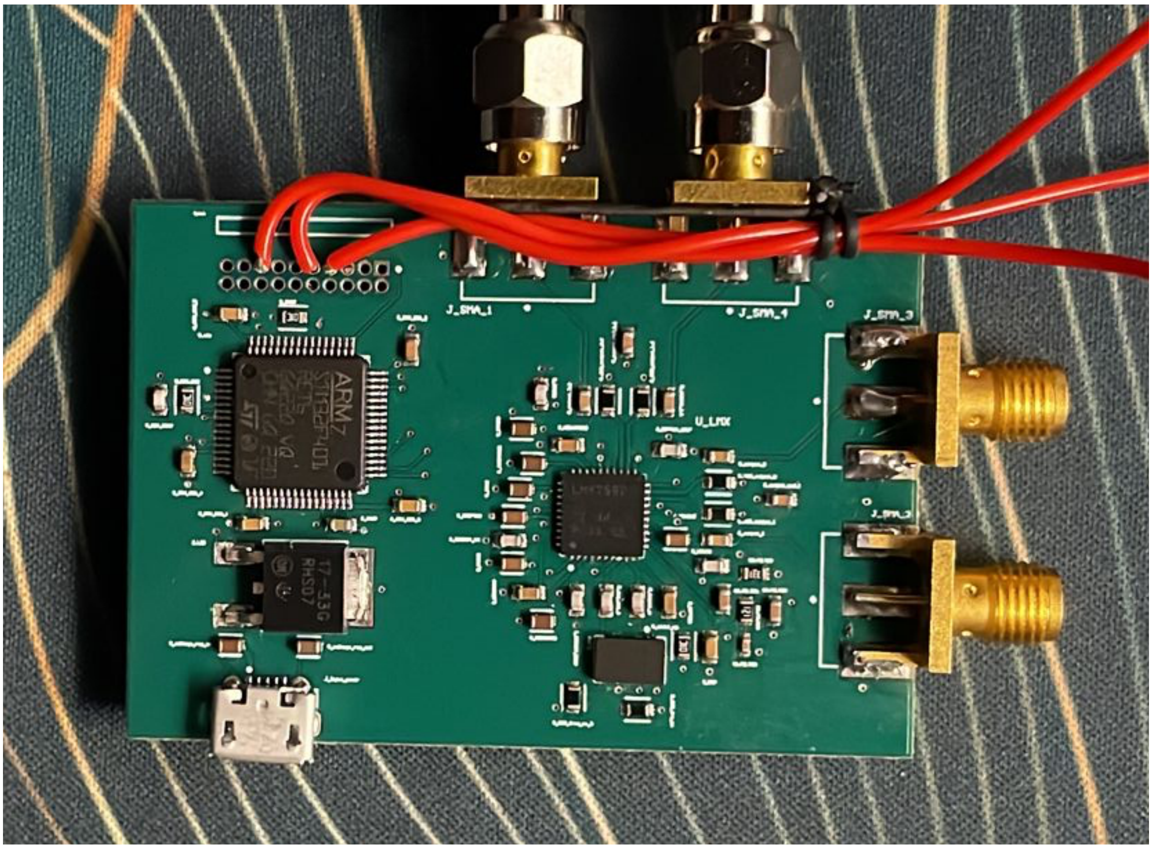


Figure 24: Board Iteration 1

3.2 Image of Board Iteration 2 Front

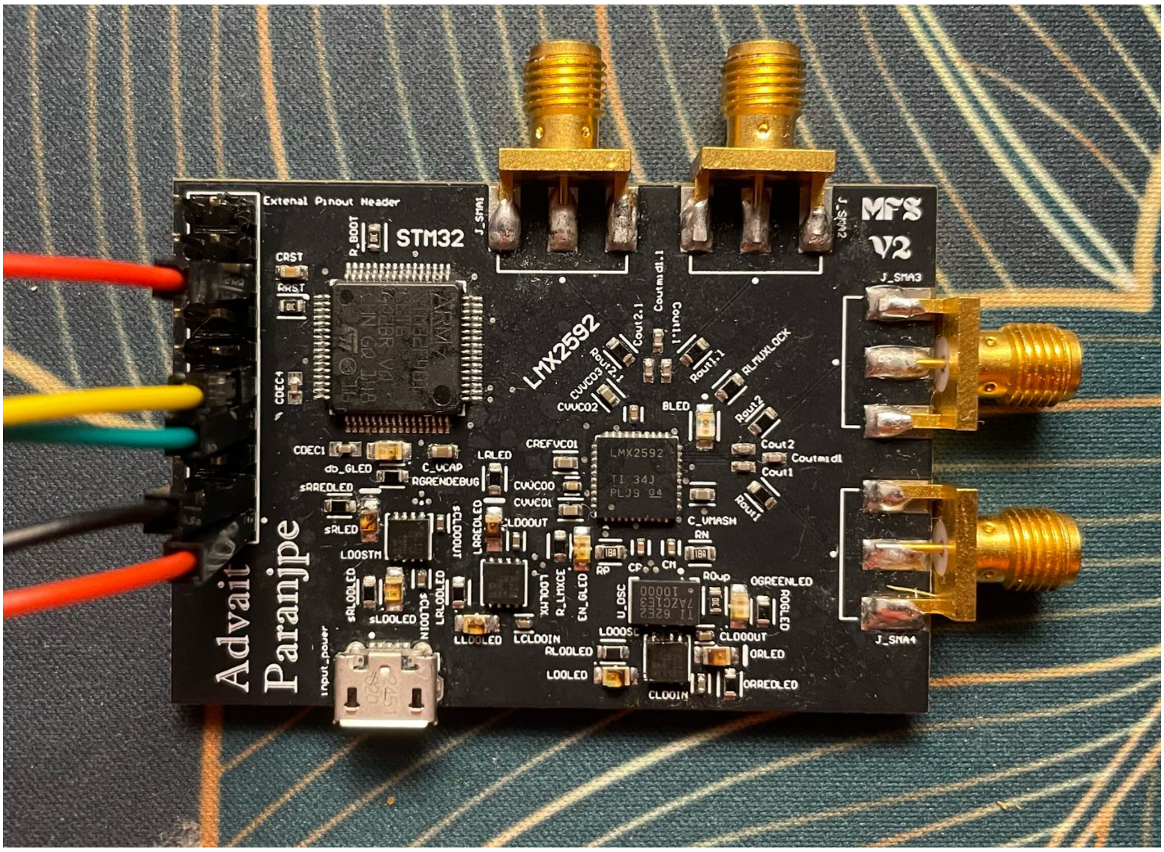


Figure 25: Board 2 Front

3.3 Image of Board Iteration 2 Back

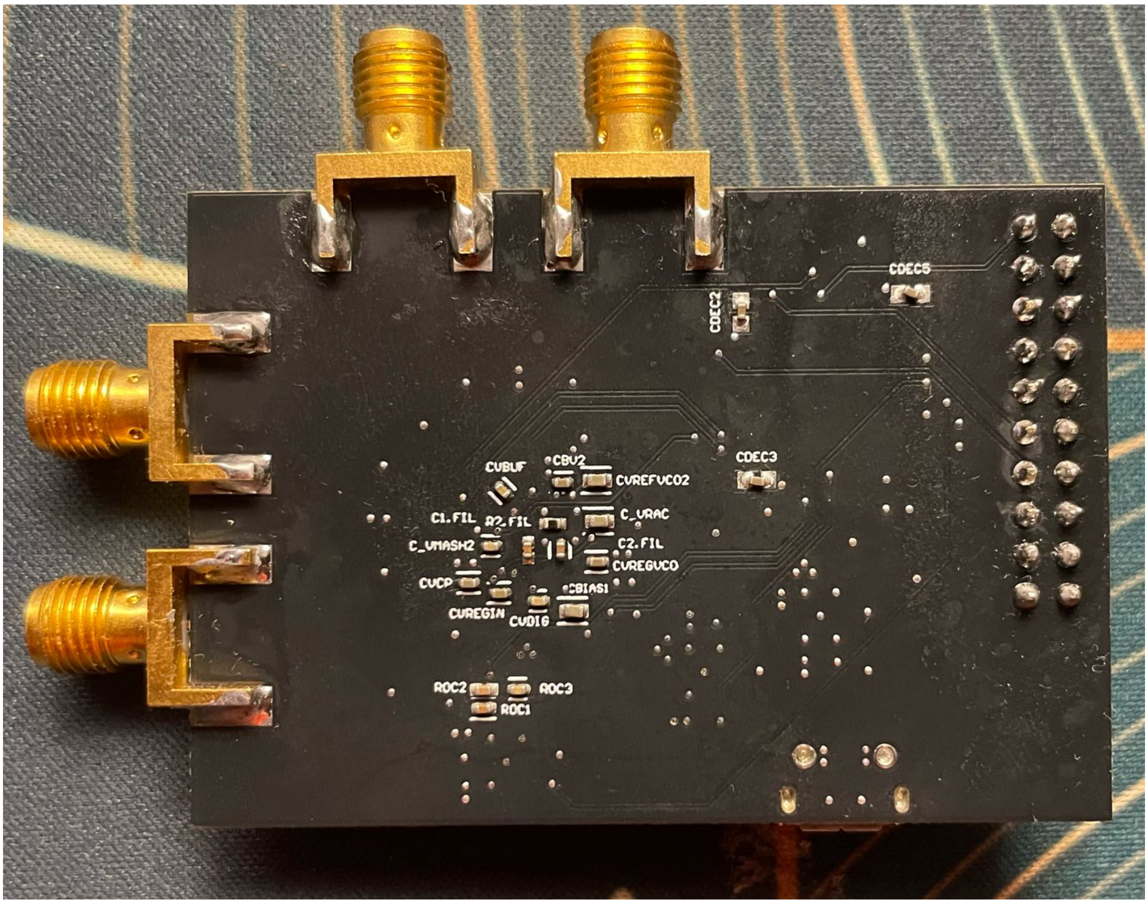


Figure 26: Board 2 Back

4.1 Schematics of Optimised Board Layout

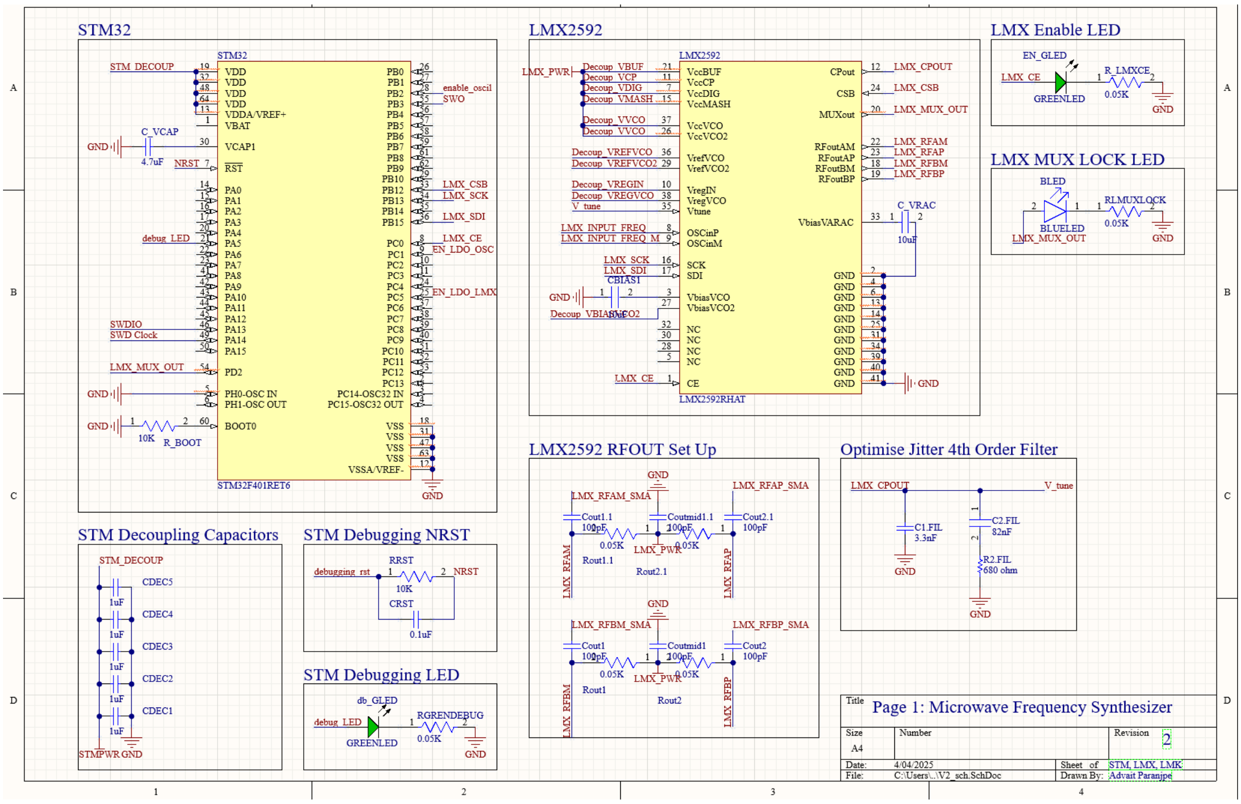


Figure 27: Page 1 of Optimised Board Schematic

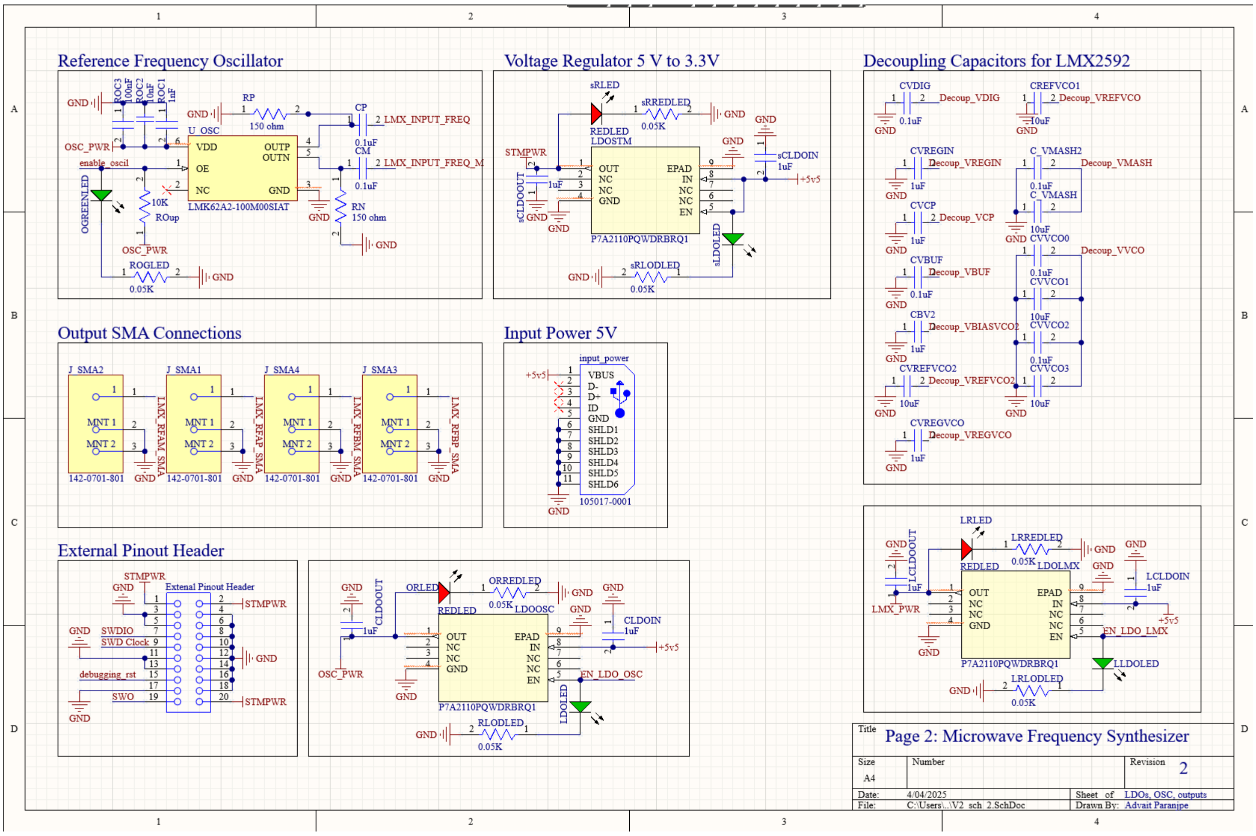
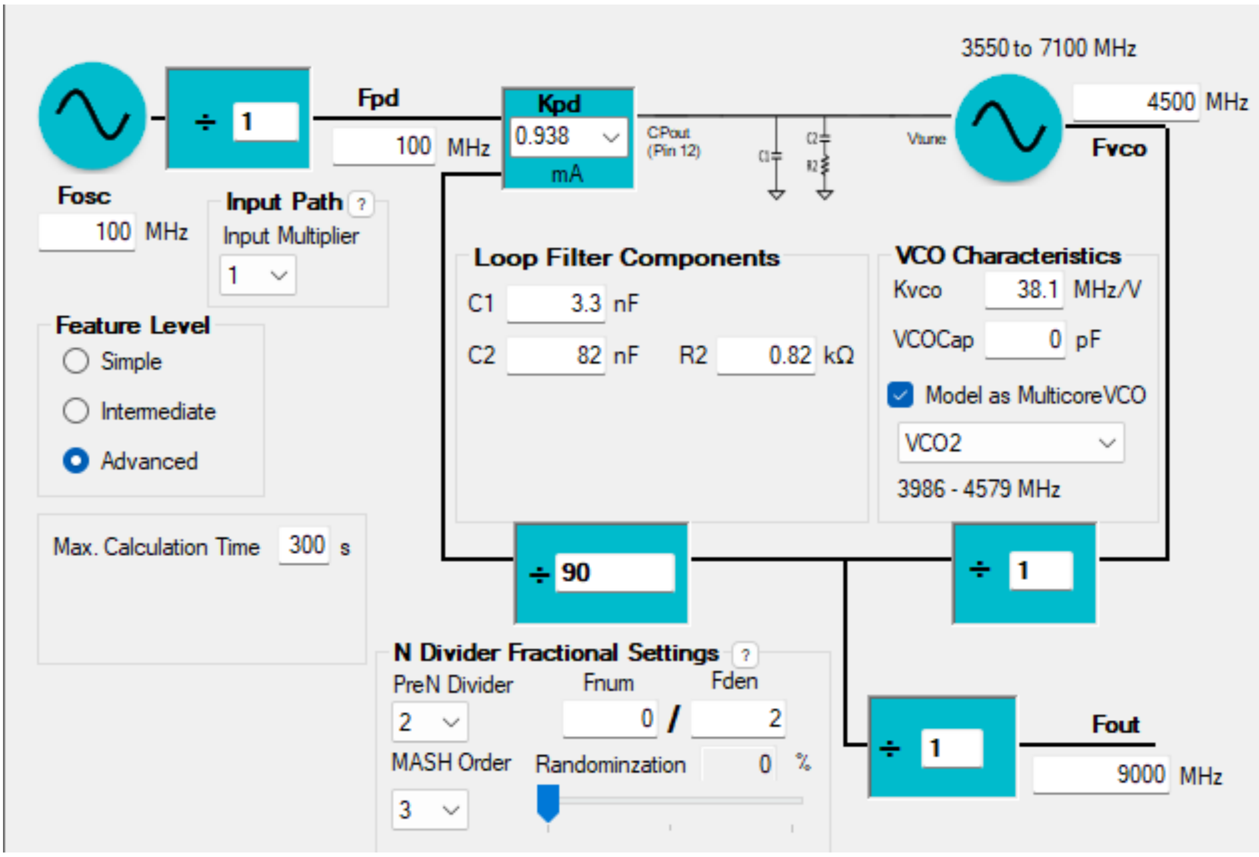

Reference Frequency Oscillator
Voltage Regulator 5 V to 3.3V
Decoupling Capacitors for LMX2592
LMK62A2-100M00SIAT
P7A2110PQWDRBRQ1
Output SMA Connections
Input Power 5V
External Pinout Header
142-0701-801
105017-0001
Title
Page 2: Microwave Frequency Synthesizer
Size
A4
Number
Revision
2
Date: 4/04/2025
File: C:\Users\...\V2_sch_2.SchDoc
Sheet of LDOs, OSC, outputs
Drawn By: Advait Paranjpe

Figure 28: Page 2 of Optimised Board Schematic

4.2 PLLatinum Sim Configurations for Optimised Board Setup



Fosc 100 MHz

Input Path: Input Multiplier 1

Fpd 100 MHz

Kpd 0.938 mA

Fvco 4500 MHz

Feature Level
- ○ Simple
- ○ Intermediate
- ● Advanced

Max. Calculation Time 300 s

Loop Filter Components

C1	3.3 nF		
C2	82 nF	R2	0.82 kΩ

VCO Characteristics

Kvco 38.1 MHz/V

VCOCap 0 pF

☑ Model as MulticoreVCO

VCO2

3986 - 4579 MHz

N Divider Fractional Settings

PreN Divider 2

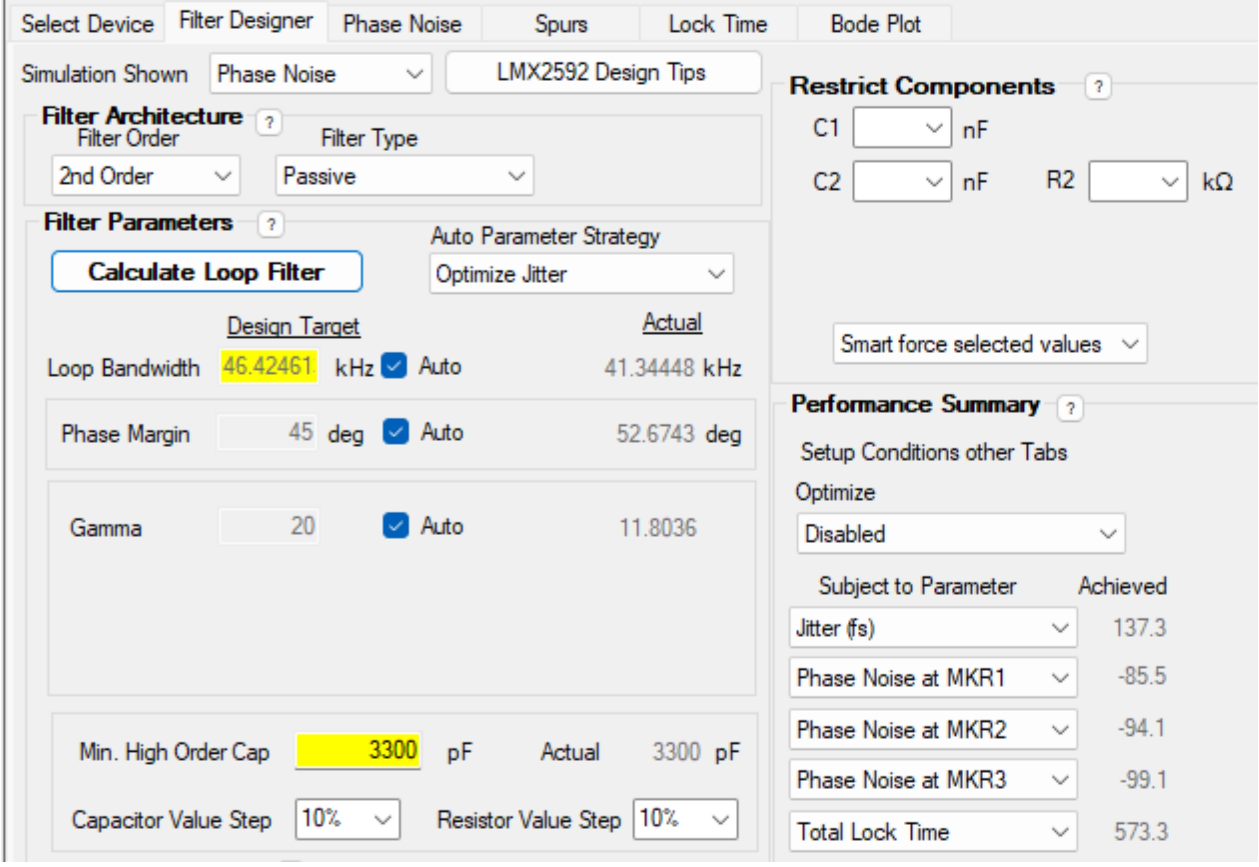
Fnum 0 / Fden 2

MASH Order 3

Randominzation 0 %

Fout 9000 MHz

Figure 29: Capacitor and Resistor values for 2nd Order Optimised Jitter Filter



Simulation Shown: Phase Noise

Filter Architecture: Filter Order 2nd Order; Filter Type Passive

Filter Parameters

Auto Parameter Strategy: Optimize Jitter

	Design Target		Auto	Actual
Loop Bandwidth	46.42461 kHz	☑	Auto	41.34448 kHz
Phase Margin	45 deg	☑	Auto	52.6743 deg
Gamma	20	☑	Auto	11.8036

Min. High Order Cap 3300 pF Actual 3300 pF

Capacitor Value Step 10% Resistor Value Step 10%

Restrict Components

C1 nF

C2 nF R2 kΩ

Smart force selected values

Performance Summary

Setup Conditions other Tabs

Optimize: Disabled

Subject to Parameter	Achieved
Jitter (fs)	137.3
Phase Noise at MKR1	-85.5
Phase Noise at MKR2	-94.1
Phase Noise at MKR3	-99.1
Total Lock Time	573.3

Figure 30: Loop bandwidth and Optimise Jitter option selected

5.1 Project Outline

Abstract:

The aim of this project is to design and develop a *microwave frequency synthesizer (MFS)*, for 5G applications. The key objective is to build an MFS which can be dynamically tuned using bespoke control algorithms, optimising connectivity. Theoretical research on MFS design, simulations, and implementation of an MFS that can be adjusted using sophisticated control algorithms form the basis of this project. Upon implementation, key metrics such as frequency stability, phase noise and output power will be measured using equipment like a spectrum analyser. The project aims to contribute to the growing need for stabilised and flexible 5G networks. The final deliverable will be a functional prototype of an MFS, with software capable of adjusting the synthesizers parameters in real-time.

Introduction:

As 5G technology becomes more prevalent within our society, the need for optimising these systems is ever increasing, such that next generation systems have a solid foundation to build upon. What facilitates these communication systems are microwave frequency synthesizers, which are critical in producing stable, tuneable frequencies. As highlighted by Pozar, “precise frequency control is vital for maintaining seamless connectivity and enabling the rapid data transmission required by modern communication networks”. Consistent frequency generation directly impacts data throughput and user experience, making it a cornerstone of advanced telecommunication infrastructure.

Although traditional frequency synthesizers are effective, they often rely on manual tuning and lack ability to adapt dynamically. As noted by Chenakin, “this lack of flexibility is a significant drawback when designing systems that need to handle the increasingly complex demands of 5G technology”. This project addresses these challenges by introducing bespoke control algorithms aimed at automating and optimizing MFS performance for 5G applications, moving beyond the constraints of manual intervention; crucial for supporting applications such as augmented reality, autonomous vehicles, and smart city technologies.

Thus, this project will address both the theoretical and practical implementation. The expected outcome is a functional adaptive MFS prototype, paving the way for seamless connectivity and rapid data transmission in next-generation networks.

Project Scope and Process (for Gantt Chart):

Milestones	Description	Objectives	Success Criteria/Deliverables
Theory: 30 days	Develop an understanding of MFS design and control algorithms for tuning the MFS.	Build knowledge on MFS architecture and control algorithms to adjust frequency and stability.	A comprehensive literature review and theoretical understanding of key MFS components (e.g., PLLs, VCOs) and control algorithm principles.
Design Stage 1: 21 days	Design the MFS using simulation software and develop basic control algorithms using Python.	Create **three** preliminary designs for the MFS in simulation, including basic control algorithms for real-time tuning.	Successfully implement basic MFS designs in simulation software with functional control algorithms coded in Python.
Simulation: 8 days	Simulate the MFS's in software tools like Keysight ADS and validate control algorithms using Python.	Simulate the MFS's behaviour and test control algorithms for their ability to adjust key parameters.	Simulation results validate that control algorithms can successfully adjust frequency, phase noise, and stability.
Design Stage Physical: 34 days	Finalize the design and integrate control algorithms into hardware.	Design and fabricate the physical MFS based on simulation results. Integrate control algorithms into the hardware.	PCB design and fabrication are complete, and control algorithms are integrated with the MFS hardware.
Testing: 14 days	Evaluate performance metrics using RF/microwave test equipment like spectrum analysers.	Measure frequency stability, phase noise, and output power. Test the software's real-time control capabilities.	The hardware meets performance benchmarks for frequency stability and noise levels. Real-time software control is verified through testing.

Table 1.2 Project Scope and Key Deliverables (see Gantt chart also)

Given the iterative nature of development, the project scope may evolve based on design and implementation findings. Any adjustments will be coordinated with the Project Supervisor to align with the project's objectives.

Further innovation (if time permits):

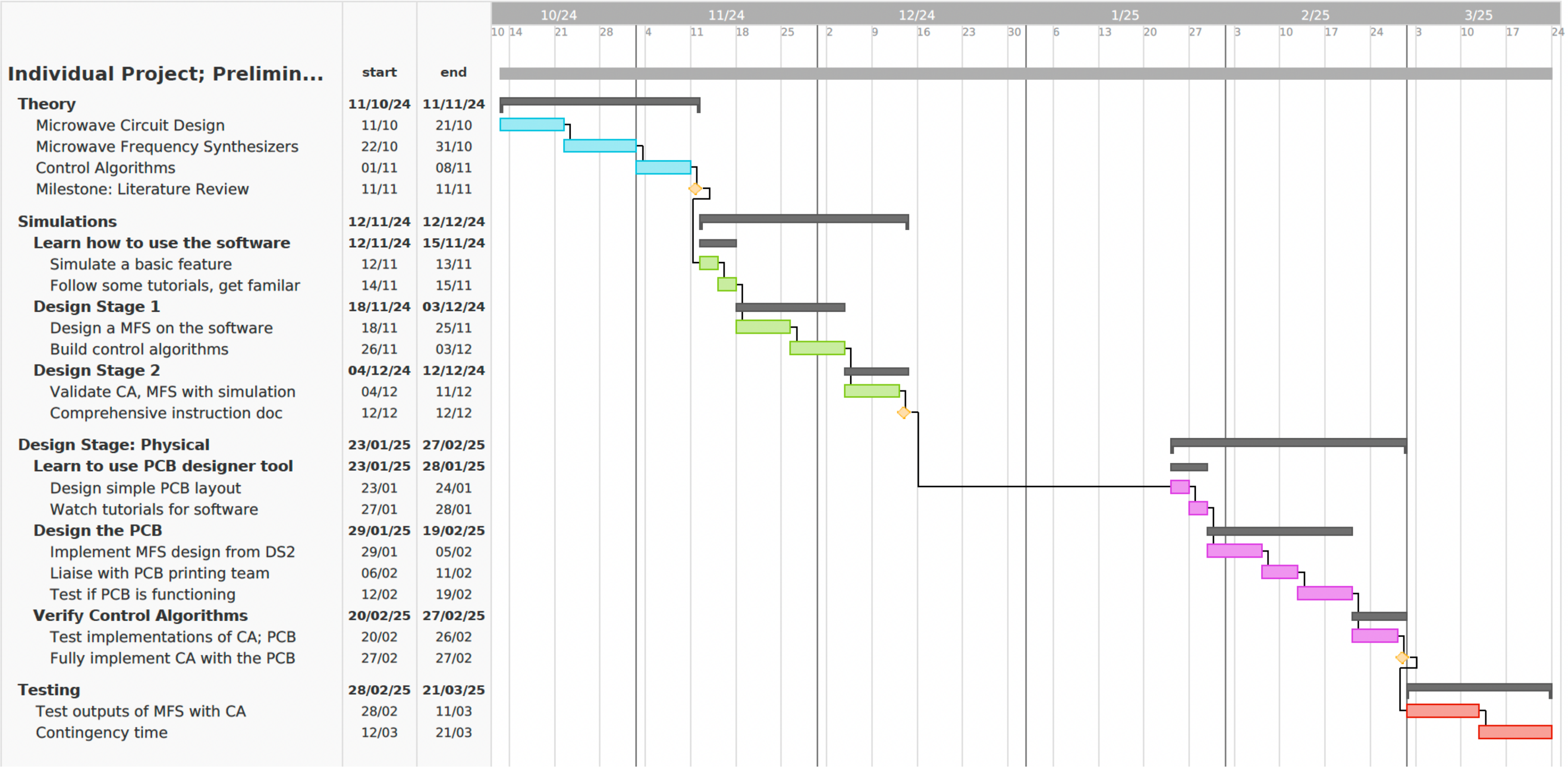
While the focus of the project is to develop a functional MFS, with control algorithms to allow real-time tuning, incorporating machine learning could further optimise the process. If time permits, I will explore how data-driven machine learning models could predict optimal tuning frequencies based on environmental conditions or network congestion. This enhancement would push the boundaries of what 5G systems are capable of and contribute to future research of next generation communication systems.

5.2 Health and Safety Risk Assessment

Hazard	Risk	Mitigation
Electrical Shock	Exposure to live wires or circuits.	Use insulated tools, wear rubber-soled shoes, disconnect power when handling circuits.
Soldering Burns & Fumes	Burns from soldering iron, exposure to harmful fumes.	Wear protective gloves and glasses, ensure proper ventilation or use fume extraction.
Microwave Radiation	Harm from exposure to microwave frequencies.	Shield equipment, maintain safe distance, use regulated testing equipment.
Ergonomic Issues	Eye strain, repetitive strain injuries (RSI), poor posture.	Follow ergonomic setup guidelines, take breaks, stretch regularly.
Fire Hazard	Overheating components or improper handling of hot tools.	Keep fire extinguisher nearby, ensure flammable materials are removed, monitor equipment closely.
First Aid & Emergency Response	Injuries from burns, shocks, or other accidents.	Ensure a first aid kit is present

Table 4.2.1 Health and Safety Risk Assessment

5.3 Initial Project Plan (Gantt Chart)



Individual Project; Prelimin...	start	end
Theory	**11/10/24**	**11/11/24**
Microwave Circuit Design	11/10	21/10
Microwave Frequency Synthesizers	22/10	31/10
Control Algorithms	01/11	08/11
Milestone: Literature Review	11/11	11/11
Simulations	**12/11/24**	**12/12/24**
Learn how to use the software	**12/11/24**	**15/11/24**
Simulate a basic feature	12/11	13/11
Follow some tutorials, get familar	14/11	15/11
Design Stage 1	**18/11/24**	**03/12/24**
Design a MFS on the software	18/11	25/11
Build control algorithms	26/11	03/12
Design Stage 2	**04/12/24**	**12/12/24**
Validate CA, MFS with simulation	04/12	11/12
Comprehensive instruction doc	12/12	12/12
Design Stage: Physical	**23/01/25**	**27/02/25**
Learn to use PCB designer tool	**23/01/25**	**28/01/25**
Design simple PCB layout	23/01	24/01
Watch tutorials for software	27/01	28/01
Design the PCB	**29/01/25**	**19/02/25**
Implement MFS design from DS2	29/01	05/02
Liaise with PCB printing team	06/02	11/02
Test if PCB is functioning	12/02	19/02
Verify Control Algorithms	**20/02/25**	**27/02/25**
Test implementations of CA; PCB	20/02	26/02
Fully implement CA with the PCB	27/02	27/02
Testing	**28/02/25**	**21/03/25**
Test outputs of MFS with CA	28/02	11/03
Contingency time	12/03	21/03

5.4 Risk Register

Project Title:	Development of a frequency synthesizer with software control		Submission Date:	08/10/2024	
Student Name:	Advait	Paranjpe			

Project Risk	Severity			Potential			Score (Severity x Potential)	Mitigation Measures
	L	M	H	L	M	H	L=1, M=2, H=3	
PCB Design and Fabrication Delays		X			X		4	Early PCB design submission, extend simulation phase to refine algorithms until hardware is available.
Control Algorithm Ineffectiveness		X			X		4	Fall back on simpler, control algorithms if initial versions fail to meet expectations.
Hardware Malfunction During Testing			X		X		6	Test individual components, have spare parts, use software simulations to identify issues before hardware testing.
Time Management		X				X	6	Prioritize core functionalities, use buffer periods in the timeline, focus on primary deliverables to ensure project completion.
Component Availability Issues		X			X		4	Identify alternative suppliers early in the project and list equivalent components
Frequency Interference Issues		X				X	6	Conduct testing in a controlled environment or use shielding techniques to minimize interference. Relocate testing if necessary.
Software Integration Problems		X			X		4	Perform thorough testing of software modules before integration. Maintain modular code architecture for quick issue isolation.
Unexpected Power Failures or Data Loss			X	X			3	Regular data backups and use a UPS for equipment. Save progress and use version control.
Miscommunication with Supervisor	X				X		2	Schedule regular meetings, summarize meeting points in written form for clarity.
Minor Software Bugs	X				X		2	Follow best coding practices, conduct unit tests regularly, use version control to quickly identify and address bugs.

5.5 CPD

Continuing Professional Development Log

Name: *Advait Paranjpe*

Current and recent CPD activity:

CPD Activity Title	Description	Dates	CPD Hours
Learning how to use Altium Designer	Since I was working on RF design, I had to ensure I was consistently executing best practices when using the Altium Design Software. I am now proficient in using the software.	15/10/2024 – 21/03/2025	100 Hours
Learning how to solder very small footprint components	I had to use new techniques such as using solder paste and sticking the component on and connecting the solder via a heat gun. I also became more competent in regular soldering.	05/02/2025 – 21/03/2025	8 Hours
Participated in Hack-a-bot	Interfaced with the Sony integrated AI camera to where I generated a custom dataset and programmed a custom ML model to detect ASL and output in speech the exact letter being signed	29/03/2025 – 30/03/2025	30 Hours
Learning how to design a PCB	Learning the fundamentals behind PCB design allowed me to harness Altium Designer to use best PCB design practices.	01/10/2024 – 15/01/2025	75 Hours

Planned and Future CPD activity:

CPD Activity Title	Description	Skills addressed	Dates
Internship	Software Engineering Internship at NESO	It will allow me to understand how companies work on larger code bases, as well as improve my proficiency in Python and C++ programming.	01/07/2025
Spanish Lessons	Learning the Spanish Language	I will increase my proficiency from beginner to intermediate in Spanish since I will be moving to California where it would be useful to know Spanish	01/07/2025