University of California
Santa Barbara

Mixed signal control techniques for Optical Frequency Synthesis

A dissertation submitted in partial satisfaction
of the requirements for the degree

Doctor of Philosophy
in
Electrical and Computer Engineering

by

Akshar Jain

Committee in charge:

Professor Luke Theogarajan, Chair
Professor John Bowers
Professor Forrest Brewer
Professor Clint Schow

December 2021

The Dissertation of Akshar Jain is approved.

Professor John Bowers

Professor Forrest Brewer

Professor Clint Schow

Professor Luke Theogarajan, Committee Chair

November 2021

Mixed signal control techniques for Optical Frequency Synthesis

by

Akshar Jain

Acknowledgements

It takes more than one voice to make a harmony, and I know for a fact that I would not have completed this journey without the love and support of the countless individuals around me. However, this work would not have been possible without the constant and unwavering support of my advisor Professor Luke Theogarajan. As other member of the Biomimetic Circuits and Nanosystems group will attest, he has always been present to help in our hour of need; nothing could demonstrate this more than the fact that he spent as much time in the office as we did, during all our tapeouts - working just as hard as us but also picking us up and lifting us with words of encouragement in our moments of despair. He took a big risk when he accepted me into the group, and I hope I have repaid the faith that he showed in my abilities.

I would also like to thank all the members of my committee. Professor John Bowers has been an exemplary Project Manager, and if it wasn't for him keeping me on my toes and keeping me honest - I would have probably taken twice as long to finish. I would also like to thank him for opening his lab to me and providing all the support and his time during the project. There is not a single part of this work, that is not in some shape or form been influenced by the lessons I learnt from Professor Forrest Brewer's classes. Despite never having met me, Professor Clint Schow graciously accepted me request to be on my committee, and I am ever so grateful for all the warm messages and support he has provided me with during this process.

I would like to thank all of the members of the Biomimetic Circuits and Nanosystems group - if it was not for them, this work would be nigh impossible. I would like to thank Aaron Bluestone for taking me under his wing when I first started and teaching my practically everything I know about locking lasers and testing in the lab. I will always cherish the memories we have from our countless all-nighters and will always remember

how you always had a joke ready to lighten the mood in our moments of despair. I would also like to thank Sean McCotter, who even though started out as a student I taught, ended teaching up me more about circuit design than I could have anticipated. The academic 2018-19 will forever be my favorite year in Graduate school, for all the wonderful memories I shared inside and outside the lab with Sean, Alex Nguyen-Le and Sarah Grundeen. I will be eternally grateful to have the privilege of calling the three of you my friends. I would also like to thank the Undergraduate interns in our lab, Nicole Griffin and Ryan Kaveh, who have now gone on to do such great things in their own careers.

And finally, my loving parents and family, who have been the biggest supporters throughout my academic journey. I would also like to acknowledge all my close friends at UCSB - Bharath Vishwanath, Kalyan Tej, Bharat Chandrahas, Kruthika Sivakumar and Ravishankar Muniraju. They have been ever present to provide me with any help that I may need and I am so thankful to have friends such as them in my life.

And finally, my girlfriend Leah Monica. Words cannot express the gratitude I feel for her constant love and support through this journey. She has coasted the ups and downs of this journey, right by my side. She has felt every one of my failures and has been the first to celebrate every one of my successes. It would be an understatement to say that I would not have completed this journey without her.

Curriculum Vitæ

Akshar Jain

Education

2021	Ph.D. in Electrical and Computer Engineering (Expected), University of California, Santa Barbara.
2017	M.S. in Electrical and Computer Engineering, University of California, Santa Barbara.
2015	B.S. in Electronics and Communication Engineering, National Institute of Technology, Karnataka.

Experience

2016-21	Graduate Student Researcher, Biomimetic Circuits and Nanosystems Group, UCSB
2019	Gradudate Student Intern, Intel Corporation, Hillsboro, OR
2020	Gradudate Student Intern, Intel Corporation, Hillsboro, OR

Publications

- A. Bluestone, **A. Jain**, N. Volet, D. Spencer, S. Papp, S. Diddams, J. Bowers, and L. Theogarajan, "Heterodyne-based hybrid controller for wide dynamic range optoelectronic frequency synthesis", *Optics Express*, vol. 25, no. 23, pp. 29086-29097, 2017.
- P. Srinivasan, N. Griffin, P. Joshi, D. Thakur, A. Nguyen-Le, S. McCotter, **A. Jain**, M. Saeidi, P. Kulkarni, J. Eisdorfer, J. Rothman, C. Montell, L. Theogarajan, "An Autonomous Molecular Bioluminescent Reporter (AMBER) for voltage imaging in freely moving animals", *bioRxiv* 845198; doi: https://doi.org/10.1101/845198

Abstract

Mixed signal control techniques for Optical Frequency Synthesis

by

Akshar Jain

Beginning from their bulky and power-hungry implementations in the early 20th century, microwave synthesizers have now progressed to occupying nearly every aspect of our lives. Despite initially finding a place only in military and fringe scientific applications, these synthesizers can now be found, in some shape or form, in every electronic device we use.

Optical frequency synthesizers (OFS) find applications in the field of metrology, molecular spectroscopy, navigation, optical communication, and LiDAR. These synthesizers have the same technological disrupt potential that microwave synthesizers did in the previous century, however, most demonstrations of frequency synthesizers often involve unwieldy implementations that reside on expensive optical benches and consume several Watts of power.

To increase their applicability, it is important to reduce their Size, Weight and Power consumption (SWaP). Silicon Photonics technology, which is compatible with Complementary Metal Oxide Semiconductor (CMOS) foundry processes, offers a viable solution to this problem of integration and mass production; however, miniaturizing these devices also makes them prone to fabrication variation and environmental fluctuations during operation.

This works focuses on the challenges faced during the design and implementation of the electronics required to stabilize and control these intricate systems, and discusses three specific implementations of OFS that involve varying degrees of integration. It first

presents a Printed Circuit Board (PCB) prototype that demonstrates laser frequency synthesis with parts-per-trillion stability. It then two Application Specific Integrated Circuits (ASICs) designed in 130nm and 55nm CMOS processes, that attempt to tackle the SWaP limitations of the board level prototype. Finally, it discusses the difficulties that arise during the design and fabrication of these ASICs, and addresses the challenges faced during the testing of these circuits in conjunction with complex optical systems to achieve synthesis. Hardware and software solutions are presented at every level of the system – beginning from the PCBs that house these ASICs, continuing through the Digital Signal Processing (DSP) implemented on Field Programmable Gate Arrays (FPGA) and finally to the Graphical User Interface (GUI) designed to make interfacing with these systems easier.

The final part of this research then shifts focus to an alternative and novel method of OFS that attempts to relatively stabilize two laser systems with offsets up to THz in frequency, without the use of high-speed electronics. A board level prototype of this system achieves this feat, accompanied by a software interface that allows turn key operation of this system, enabling production of arbitrary microwave frequency signals with the click of a button.

Contents

Chapter 1

Optical Frequency Synthesis

1.1 Introduction

Few technological advancements have been as beneficial and disruptive to all aspects of our daily lives as the ones ushered in by the development of microwave frequency (1GHz to 1000GHz) synthesizers in the 1940s. These synthesizers can vary in size ranging from a tiny integrated circuit (IC) used in mobile devices to full rack-mountable electronic systems. Initially restricted to military applications, such as Radio detection and ranging (Radar) and for communication, these systems are now omnipresent in our world - from radios to bluetooth and WiFi devices. And with further deployment of "5G" technologies and the Internet of Things (IoTs), they will only increase in number.

Despite the complexities involved in implementing these systems, they can be abstracted to look like simple control systems with negative feedback as shown in Figure 1.1. We encounter control systems of this nature constantly in our everyday lives. Take for instance a car's cruise control setting. Relating this example to the blocks in Figure 1.1, at the center of the system is the plant - the car. The output of interest for this plant is the car's speed. This speed is compared against a reference: the target speed.

This difference or error is fed to the loop's filter, the computer in the car, which responds to the measured difference in speed and controls the accelerator or brakes accordingly. Now, if this was a driver controlling the speed instead of relying on the cruise control, they would be the filter and their reaction speed would be the time constant associated with this filter. The "negative" part of the negative feedback loop comes from the fact that the filter responds negatively to the difference in measured speed to the target speed - when the car is above the target speed - it slows down and vice versa, when it is below the target speed - it speed ups. In this discussion, the astute reader will notice that the block labelled β was skipped over. β is commonly referred to as the feedback factor and is the fraction of the output signal that is referred back to the error block. In this example, $\beta = 1$, meaning the speedometer would show the actual speed of the car.

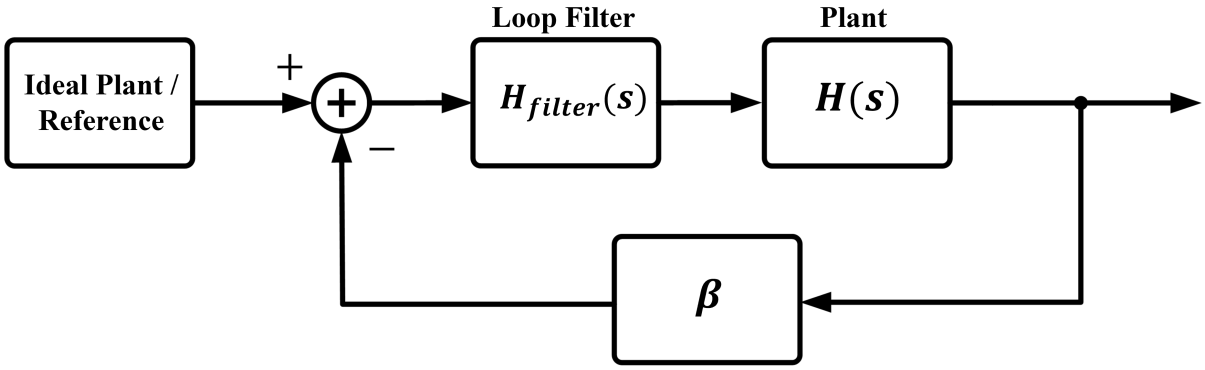


Figure 1.1: Generic Feedback system

Taking this example forward, let us look at the microwave frequency synthesizer shown in Figure 1.2. Synthesizers like this can be found in FM/AM tuners, or in phones and computers where they produce the clock signal for their processors. In this example, our plant is a voltage controlled oscillator (VCO). Oscillators are electronic devices that produce an electrical output that varies periodically with time. In this case, just like a car's accelerator and brake control its speed, the oscillator's output period (or its

frequency, which is mathematically the inverse of its period) can be changed using a control voltage. The reference for the synthesizer here is typically a quartz crystal or even an ultra pure Radio Frequency (RF) signal generated by a Cesium clock [1]. These sources are chosen as the reference for the frequency synthesizer because of their spectral purity, and how stable and periodic these clock sources can be (the Cesium fountain atomic clock in [1] had an uncertainty about 1×10^{-15} in the year 2000, but as of the year 2013, the uncertainty has been reduced to 3×10^{-16}, meaning it will take more than a 100 million years for it to gain or lose a second!). Unfortunately, the trade off made for their purity is that they also have a fixed output frequency (in the order of 10s of MHz). Synthesizer oscillators need to oscillate with frequencies much higher than that (FM broadcasting stations in America are assigned to 101 channels, designated from 87.9 to 107.9 MHz and the clock frequencies in processors are in excess of a GHz). This is where the feedback factor β comes into the picture. As shown in Figure 1.2, a divider is used that divides the output frequency of the oscillator by a factor N which then brings the frequency down to a value comparable to the reference frequency. Once the loop has settled and reached a steady state (presuming everything worked correctly and in a timely fashion), the output frequency will be an exact multiple of the reference frequency by a factor N ($F_{OUT} = N.F_{REF}$). By changing the divide value - N, the output frequency of the loop can be changed. This is why these loops are called synthesizers, by changing N, different frequencies can be synthesized.

Apart from making sure that the output frequency is a multiple of the reference, these loops carry out another very important function. As mentioned earlier, the references chosen for these systems are highly periodic and do not lose time over several years. The VCO on the other hand is highly susceptible to noise sources (partly owing to its tunability). Because of negative feedback, the loop can correct these frequency errors and attempt to make the loop's oscillator oscillate as perfectly as the reference. However, the

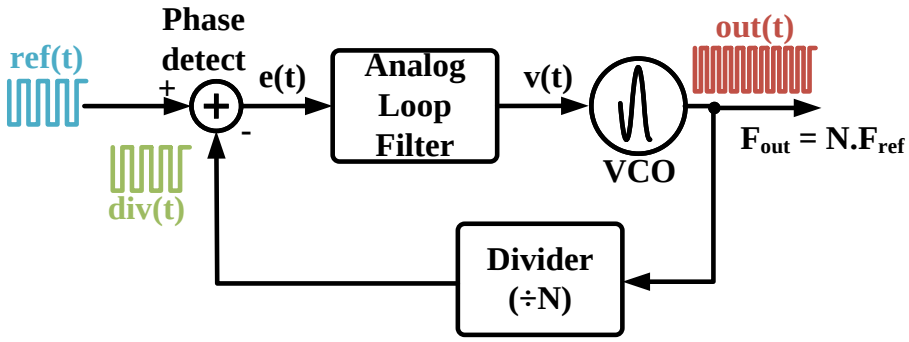


Figure 1.2: Generic Phase Locked Loop

amount of noise the loop can filter is limited by its response time and so, the output can never be as pure as the reference. Since the loop only can correct errors that are slower than its response time, errors that are too fast cannot be corrected (a more advanced reader will recognize this as the loop's bandwidth).

The synthesizers similar to the one shown in Figure 1.2 are referred to as "Phase Locked Loops" (PLLs), since the quantity measured by the difference block shown in the figure is the phase difference. In a PLL, an error signal is produced by measuring the phase difference between the reference and the feedback signal and then fed to the loop filter. Thus, when the loop has settled, the two phases are "locked" to each other and thus, move together.

Optical frequency synthesizers (OFS) have the potential to create a technological revolution in the same way microwave synthesizers did in the 20th century. They can be applied in the fields of metrology [2, 3, 4, 5], spectroscopy [6, 7, 8], navigation [9, 10], microwave photonics, and precision LiDAR [11, 12, 13]. Here, instead of the frequency of an electrical output, the output of interest is that of a laser. For all the applications listed in this manuscript, the lasers being used have a wavelength in the C-band ($\lambda \approx 1550$ nm and f = $\sim$193THz)

Traditional electronics are only capable of processing signals up to few 10s of GHz and so, the same method of frequency division that was used in the case of a PLL cannot be used to bring the laser's frequency down to the order of a few MHz. Thus, the most demanding part of an optical frequency synthesizer is this "link" between the optical and microwave domain. Several efforts were made in the 1960's by Javan et al. to multiply the microwave frequencies up using several non linear devices (diodes for a few THz and then nonlinear crystals for 100s of THz). These devices, however, as one would expect, were highly unreliable and quite bulky. Research interests in the early 1990s then shifted to dividing the output frequency of the lasers (similar to PLLs) using higher-order optical non-linearities. The real breakthrough in this field came in the late 90s when the Hänsch group demonstrated stable Ti:sapphire mode-locked lasers for metrology applications. These mode-locked lasers could produce "Optical Frequency Combs" that would finally provide the missing link. Deeper discussions about how frequency combs operate and can be produced [14] are beyond the scope of this manuscript, but the next section will briefly discuss what an optical frequency comb is and how they can be used to finally solve the problem of dealing with frequencies that are ∼193THz. There have been several demonstrations of OFS using frequency combs, and some have even gone to the extent of demonstrating synthesis at the chip scale on both native III-V and heterogeneous silicon platforms.

Optical Frequency Combs

An optical frequency comb is an optical spectrum of equally spaced lines in the frequency domain (shown in Figure 1.3) that often spans several THz. Each "comb" line can be thought of as a source of laser light located at a fixed distance from its neighbors. This distance is called the repetition rate of the comb, f_r. The frequency of the n^{th} comb line can be given by the formula:

$$f_n = f_0 + n.f_r \tag{1.1}$$

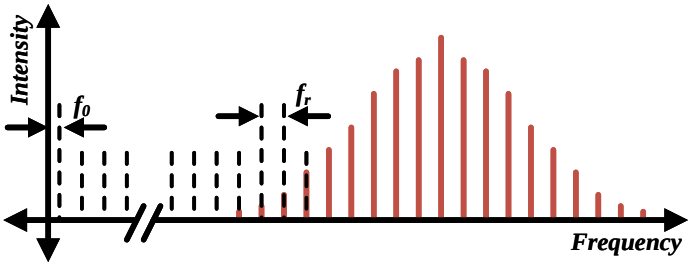


Figure 1.3: Optical Frequency Comb

where the term f_0 is known as the carrier envelope offset frequency (this may also be referred to as f_{ceo}). The source of this offset frequency is beyond the scope of this discussion, however, there are several ways to calculate this offset frequency.

Before moving forward, a refresher on the concept of "heterodyning" is necessary. A heterodyne (often called a "beat note") is a signal frequency created by a process called heterodyning. In most applications, this process involves mixing two signal frequencies f_1 and f_2 such that the output of the mixer contains two signals - one containing the sum of the individual frequencies $f_1 + f_2$ and the other containing the difference $f_1 - f_2$. These mixers are usually followed by a filter that filters out the higher frequency term and leaving only the difference.

Now, if the span of the comb is more than an octave (the highest comb line has a frequency greater than twice the smallest frequency), this offset can be calculated using a non-linear device called a Second Harmonic Generator (SHG). An SHG, as the name

suggests, doubles the frequency of the optical signal at its input. Once the second harmonic of the frequency comb has been attained, an optical heterodyne can be performed on it with the original frequency comb. This procedure is illustrated in Figure 1.4, where the resulting beat note's frequency is given below [5]:

$$f_{beat} = f_2 - f_1 \tag{1.2}$$

$$= (n_2 f_r + f_0) - 2 * (n_1 f_r + f_0) \tag{1.3}$$

$$= (n_2 - 2 * n_1) f_r + f_0 \tag{1.4}$$

$$= f_0 \text{ (for } n_2 = 2 * n_1) \tag{1.5}$$

And so, the offset frequency f_{ceo} or f_0 can be directly calculated from the beat note [15] [16]. Once the offset frequency has been found, the value of each of the variables in Equation (1.1) is known and the frequency of each comb tooth can be precisely calculated. Apart from measuring f_0, a substantial amount of work has been done to lock this offset frequency to a microwave source and even to completely remove this offset frequency.

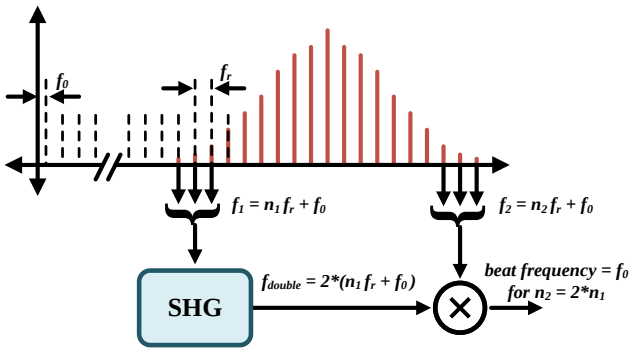


Figure 1.4: Illustration of f_0 calculation using $f - 2f$ generation with an SHG

So how does this frequency comb and the precise knowledge of the frequency of each comb tooth help us bring our tunable laser's frequency down to the microwave regime?

The answer is heterodyning, the technique introduced in the previous section. If a tunable laser is optically heterodyned with a frequency comb and the frequency difference between the laser and its nearest comb tooth is within the microwave regime, the instantaneous frequency of the laser is calculable. This frequency can be given by the formula below:

$$f_{laser} = f_0 + n.f_r + f_{beat} \tag{1.6}$$

For Equation (1.6) to work, n needs to be known, i.e. what comb line the tunable laser is close to. This can be easily achieved by generating a tuning map (Look up table) for the laser and by having a rough idea of the laser's frequency.

In this manuscript, Chapter 2 and Chapter 3 are dedicated to discussing two particular hardware architectures that process the beat note resulting from the heterodyne mix of the tunable laser and the frequency comb and use it to "lock" the laser's instantaneous frequency.

Wavemeter based frequency generation

While OFS with frequency combs demonstrates incredibly impressive performance, it still involves the use of expensive lab instrumentation and high speed electronics. If there were applications that did not require a laser to possess absolute stability but rather wished to have two lasers locked to each other with offsets of more than 100s of GHz, it would be inefficient to lock both these lasers to comb teeth. At the same time, simply processing the beat note of these lasers directly would once again be obstructed by the speed limitations of conventional electronics. The solution to this problem has actually been available since the late 19th century, and is given by the field of optical interferometry.

Figure 1.5: Basic block diagram relating the effect of a path difference to the interference pattern at the output

Consider the block diagram shown in Figure 1.5. In it, a tunable laser's output power is split in half using a "50:50 coupler." Half of this light is sent along an optical fiber of path length, "L." The second half is sent along an optical fiber of path length, "$L + \Delta L$." At the end of both of these fibers is another 50:50 coupler which is used to perform a heterodyne mix between the two lights. The resulting light is then converted to a voltage using the photoreceiver circuit shown in the figure. Since the frequencies of both lights are the same, the output will be at 0Hz (DC). This DC voltage, however, displays a peculiar relationship with the wavelength of the light. As illustrated in 1.6(a), if the wavelength of the laser stays constant, the voltage coming out of the photoreceiver circuit remains constant. However, if the wavelength of the laser is ramped (1.6(b)), a sinusoidal response to the change in wavelength is observed and this sinusoid has a fixed repetition rate.

As the wavelength of the laser is changed while keeping the path difference between the two couplers constant, the phase of the wave interfering inside the coupler changes by the relation $sin\left(\frac{2\pi}{\lambda_{laser}}\Delta L\right)$ - resulting in the sinusoidal variation. The argument of the sine function also gives us an insight into the reason behind the repetition rate. When the ratio $\left(\frac{\Delta L}{\lambda_{laser}}\right)$ is an odd multiple of $\frac{1}{2}$, the two waves meet destructively and constructively, when it is an even multiple of $\frac{1}{2}$.

Chapter 4 describes an architecture and its board level implementation that uses this

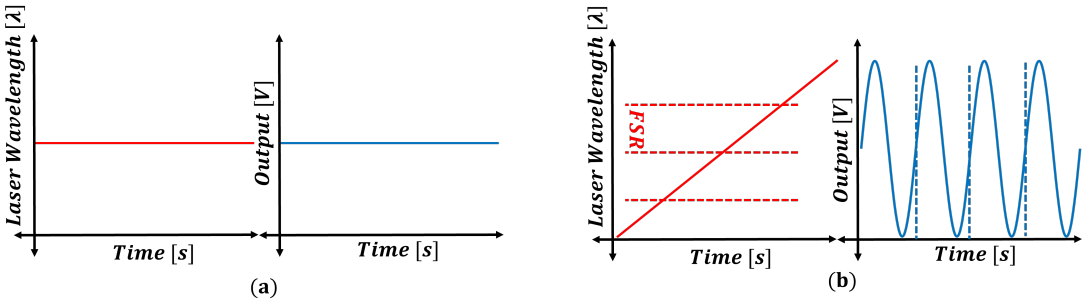


Figure 1.6: Output amplitude vs. Time (a) as the wavelength is held constant (b) wavelength is varied over time

phenomenon to lock two lasers with GHz offsets without the use of Radio Frequency (RF) electronics.

1.2 Thesis Organization

In Chapter 2, we first describe a board level implementation that uses off-the-shelf components and achieves heterodyne locks that control a laser's output frequency with mHz precision. We then proceed to discuss an Application Specific Integrated Circuit (ASIC) designed to shrink all the off-the-shelf components and integrate them on one Complementary Metal Oxide Semiconductor (CMOS) chip. In Chapter 3, we improve upon this Integrated Circuit (IC) with a chip that consumes less power and has a smaller silicon footprint. In Chapter 4, we move on to discuss a completely different and novel method of synthesizing microwave frequencies using low power lasers and low speed electronics. Finally, we conclude this work by comparing and contrasting all these techniques and discuss the cost of these implementations versus the benefits they offer.

Chapter 2

Heterodyne-based controller for OFS

From the previous chapter, readers should broadly understand how a frequency comb can help synthesize optical frequencies. This chapter will now look into further detail how these synthesizers are implemented, and try to get a better sense of the loop dynamics and some of the practical challenges involved in locking lasers. We start by giving an overview of the locking architecture and discuss conventional Opto-Electronic Phase Locked Loop (OEPLL) architectures. These architectures have a major drawback when utilized to lock a laser to a frequency comb, and so we will look at a mixed signal solution to the problem that uses both analog and digital circuitry to address this issue. We will finally examine an ASIC implementation of this system and conclude the chapter by discussing the locking results achieved using the ASIC.

2.1 Overview of Architecture

Figure 2.1 shows the optical spectrum of the tunable laser (shown in blue) and a frequency comb as they are combined using a "50:50 coupler" (also sometimes referred to as a "3dB coupler"). As the tunable laser moves close to one of the comb teeth, it

produces a beat note with frequency equal to $f_{beat} = f_n - f_{laser}$, where f_n is the frequency of the n^{th} comb tooth.

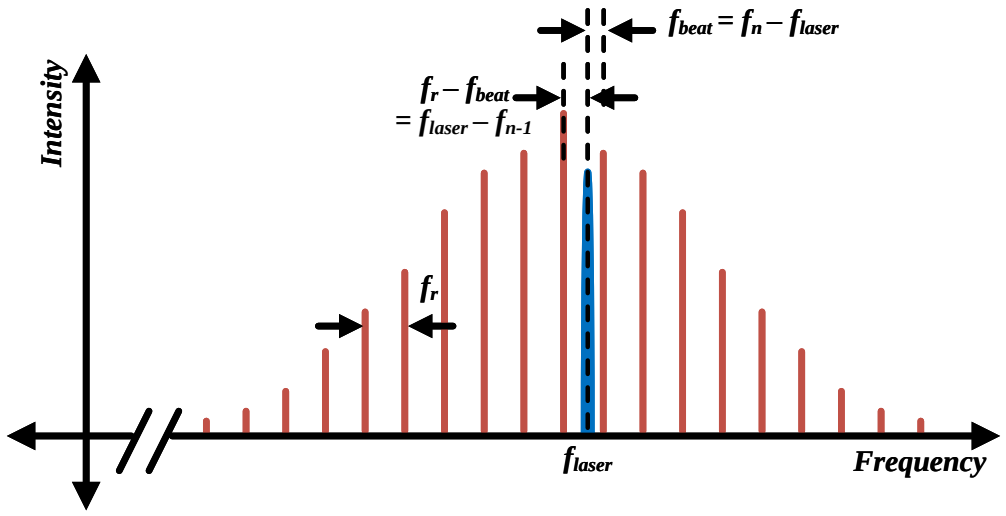


Figure 2.1: Tunable Laser lined up between two comb teeth. If $f_{beat} \approx \frac{f_r}{2}$, we can potentially lock to the wrong comb line, since $f_{laser} - f_{n-1} \approx f_n - f_{laser}$,

Figure 2.2, shows us a block diagram of a typical OPLL[17]. In this case, light is taken from the frequency comb and combined with light from our tunable laser using a coupler. This coupled light is then fed into a photoreceiver (Photo-diode (PD) + TransImpedance Amplifier (TIA)) which converts light to a voltage. This voltage signal is then mixed with an RF source whose frequency is equal to our targeted beat note frequency. The result of this mixer is then passed through an appropriately designed loop filter (LF) to generate the necessary servo signal for the laser's current source. Under ideal operations, OEPLLs designed using this architecture perform exceedingly well and can produce extraordinary locking results.

A problem, however, arises when it is desirable to have continuous tuning across the

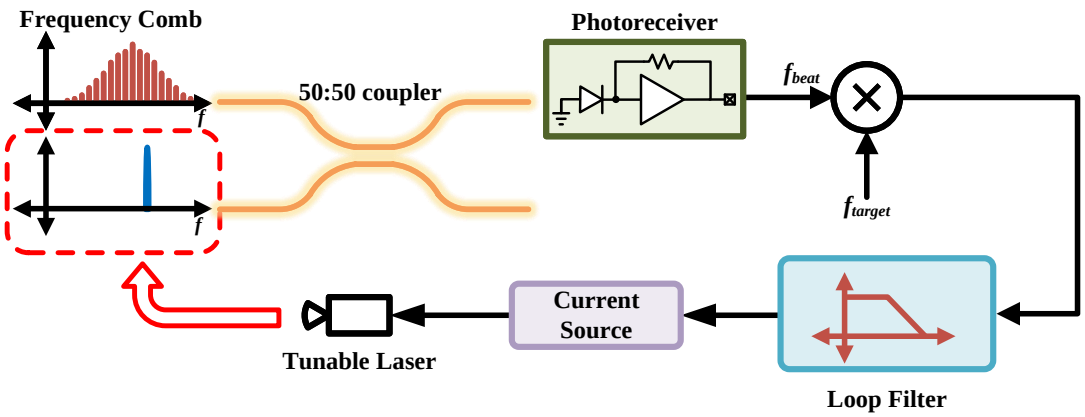


Figure 2.2: Simplified block diagram of a conventional Optical Phase Locked Loop (OPLL)

entire range of the frequency comb. When the locked laser wishes to switch from a lower comb tooth to a higher comb tooth and is exactly between two comb teeth, the beat notes from the two comb lines are so close to each other that it can be impossible to distinguish the two. In such a case, it is possible to make an error and lock to the wrong comb tooth.

The solution to this problem is by using an In-phase and Quadrature Mixer (Demodulator) at the receiver instead of just a single mixer. This method is similar to a "Costas Loop", which is a PLL commonly used in communication systems to recover the carrier signal from an incoming signal. There's been significant work done in this field, and we even have a fully integrated OEPLLs that is used to implement a 40Gbit/s coherent optical receiver.

The method described in [17], while highly effective, consumes $\approx 2 - 3W$ of power and involves the design of complex high-speed electronics. The OEPLL also implements a homodyne lock, which means that the beat frequency would only be equal to the Local Oscillator (LO) frequency. These LOs are typically implemented on-chip as PLLs and

are often integer-N PLLs. The result of that is that PLLs can only change their output frequencies as integer multiples of their own reference clock. To get non-integer multiples (or as they're called in literature fractional multiples), they need to be Fractional-N PLLs. Fractional-N PLLs involve the implementation of highly complex sub-parts and have several noise limitations of their own.

To ease the constraints placed on the electronics, as well as to reduce the power consumed, this work implements a heterodyne lock with the receiver. This means that the beat note is going to be locked at a frequency that is not equal to the LO frequency. The implementation described in the next section is done in such a way that the frequency difference between the beat note and the LO can be of a fractional value, while the LO can be an integer-N PLL.

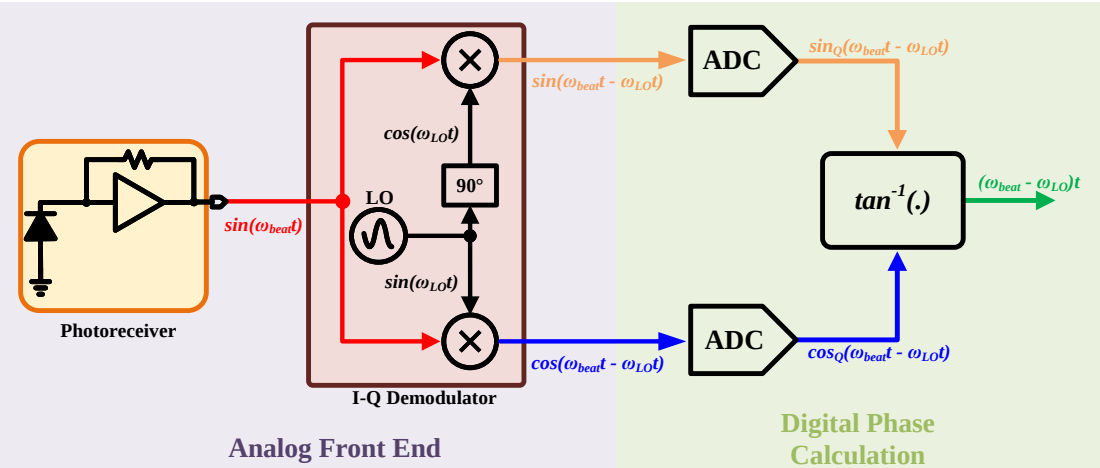


Figure 2.3: Simplified block diagram illustrating the Analog Front-End and phase calculation using the I-Q outputs of the demodulator

Figure 2.3 shows a simplified block diagram of the front end of the receiver. Similar to a Costas Loop, it produces Inphase and Quadrature components using two mixers with LO signals that are 90° apart in phase. However, instead of adding the two components

and relying on a small-angle approximation to determine the phase, these outputs are digitized using 2 Analog to Digital Converters (ADCs) and the phase rotation is measured simply by calculating the tan^{-1} of these two components.

There's two advantages of this implementation over a Costas loop implementation: First, the output phase measurement is completely independent of the photoreceiver's output power. If the laser or the comb output power changes and produces a commensurate change in the beat note power, the tan^{-1} operation cancels out these amplitude changes. The second advantage is better explained pictorially using Figure 2.4.

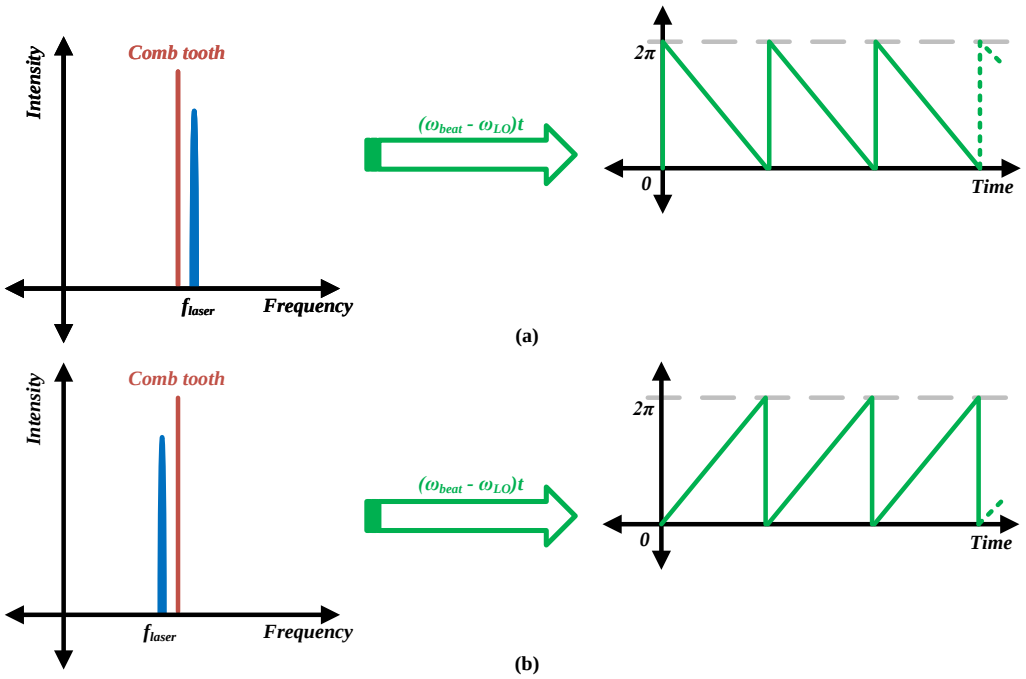


Figure 2.4: Plots demonstrating how the instantaneous phase unwraps when the laser frequency is (a) higher than a comb line (b) lower than a comb line. In situations when the laser is between two comb teeth, it now becomes easy to differentiate the beat note from the lower tooth vs. the higher one.

Figure 2.4 shows the instantaneous phase measured by the tan^{-1} block in Figure 2.3,

when the laser frequency is higher than the comb line (Figure 2.4(a)) and when it is lower (Figure 2.4(b)). As can be clearly seen from the plots shown in green, the measured phase rotation of the beat note is different in sign. The consequence of this is that it is now possible to differentiate between the beat notes that the laser produces with a higher and a lower comb line.

The next section discusses a board level implementation of this system. The system described uses off-the-shelf components for processing the laser beat note, and then sample it using a high speed Analog to Digital Converter (ADC). This sampled signal is used for Digital Signal Processing (DSP) and is used to control the laser and lock it to a reference.

2.2 Board Level Prototype

The work presented here was previously published in *Optics Express*[18]. In this testing setup, a stable bench-top commercial laser is used as the optical reference instead of using a frequency comb. As depicted in Figure 2.5, the light resulting from the mixture of this reference with the tunable laser is fed to a "light-wave converter", the Agilent 11982a, which converts this optical light to an RF voltage signal. This RF signal is divided down using two frequency dividers - a prescale-by-2 divider and then a divide-by-8. Previous synthesis experiments showed that the RF beat note demonstrated extraordinary frequency jitter that would often be too fast for the loop bandwidth and its magnitude would be greater than the loop's acquisition range. Using dividers has an additional benefit of averaging the beat note's phase noise and improving the beat note linewidth seen by the synthesizer's receiver.

The divided down signal is then input to an I-Q Demodulator (Analog Device's ADRF6820). This demodulator IC features a local oscillator capable of outputting fre-

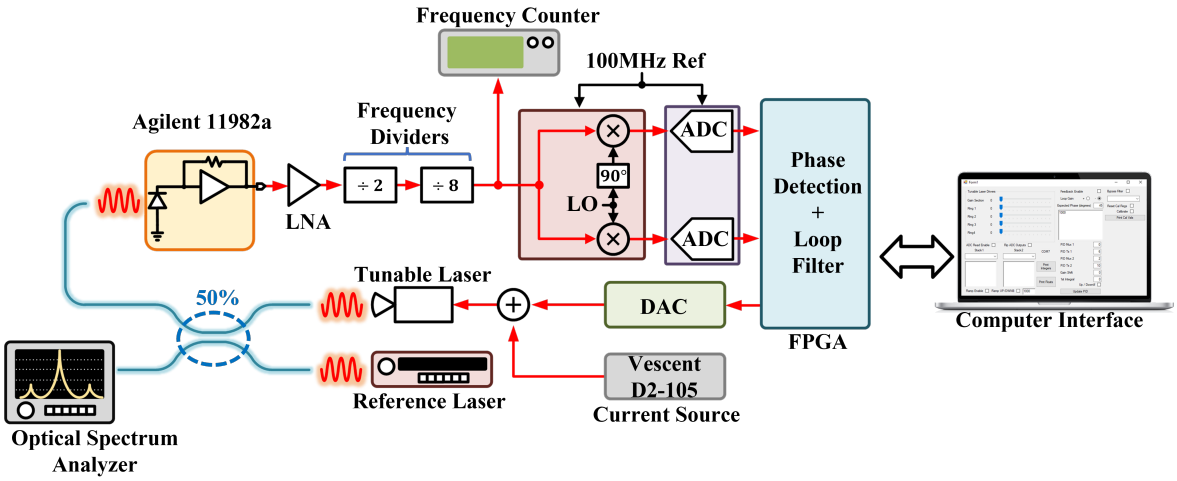


Figure 2.5: Block diagram depicting all the parts required to lock a laser to an optical reference

quencies ranging from 695 MHz to 2700 MHz [19]. This board also produces the I-Q signals described in the previous section to determine the laser's location with respect to a comb line (in this case, a reference laser that will be emulating a comb line). These I-Q signal's are then sampled using a high-speed ADC - the ADC of choice was Texas Instrument's ADS4449 [20]. This dual channel pipeline ADC was capable of sampling the demodulator signals at an astonishing rate of 250MSPs. Once sampled, these signals were then fed to a Field Programmable Gate Array (FPGA) [21], that performed as the "brains" of the loop and calculated the laser's instantaneous phase and implemented a loop filter that produced a response that would stabilize the laser and keep it locked. This signal is output through a high speed Digital to Analog Converter (DAC) that is fed to the servo input of a commercial laser driver. Through extensive experimentation and after countless conversations with our fellow colleagues, we believe that in the case of low noise locking experiments, no other current source offers a noise performance better than Vescent Photonics' D2-105 [22]. We use this laser driver to control the gain section of the laser, which in turn controls the laser's output frequency.

The next section covers all the Digital Signal Blocks (DSP) implemented inside the FPGA, followed by a discussion of some notable results that were obtained.

2.2.1 Digital Signal Processing (DSP)

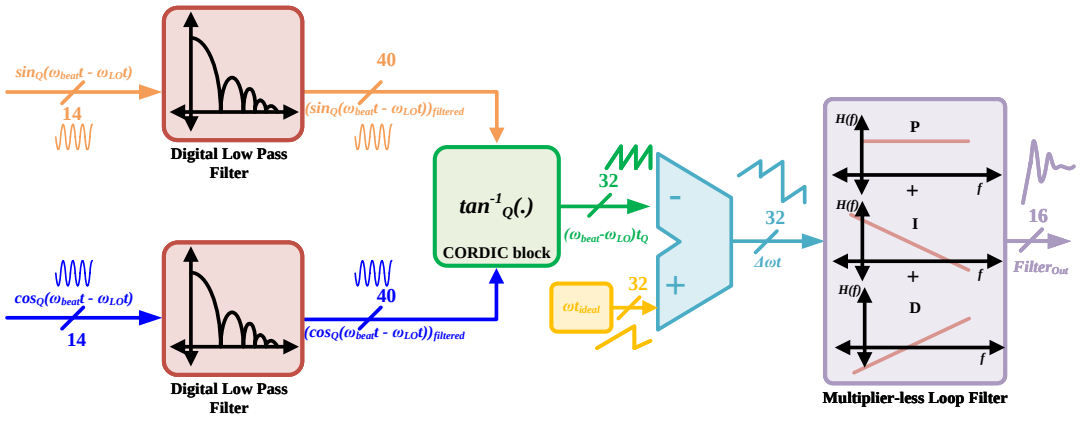


Figure 2.6: Digital Signal Processing (DSP) blocks required to process the sampled demodulator outputs and to generate the control signal for the Digital to Analog Converter (DAC)

Figure 2.6 outlines all the steps that are required to use the digitized version of the demodulated outputs and determine the control word necessary to stabilize the laser. During our measurements, we noticed random glitches at the output of the ADC that were not necessarily present coming out of the mixer. These glitches do not provide any information necessary to the system, rather they add external errors to the system that may not otherwise be present. Luckily, these glitches could be simply filtered out as higher frequency terms using a high order digital low pass filter. The filter outputs are then fed to a COordinate Rotation DIgital Computer (CORDIC) [23] block that calculates the tan^{-1} of the two signals. These hardware implementations have been around for a long time and were often used in calculators in the late 20th century. The

primary novelty of these blocks is that they compute the result of complex trigonometric functions iteratively and without the use of expensive multiply or divide operations. The output of the CORDIC block, which is our measured phase, is then compared to the ideal phase that we want our laser to rotate with. The phase error block generates this error block which is finally fed to a loop filter. To filter the error signal, a simple Proportional-Integral-Differentiator (PID) controller is implemented. This PID has filter coefficients that are strictly limited to powers of 2. This limitation ensures that every multiplication or division operation in the filter is now a left shift or right shift operation.

50th Order Low Pass Filter (LPF)

All the DSP required for our loop is carried out on a Zedboard which features a Xilinx Zynq-7000 All Programmable SoC and 7 series programmable logic [21]. To synthesize the filter for this FPGA, we use a commercially available Computer Aided Design (CAD) software called "Vivado" [24]. One of the benefits of this software is that it allows the user to easily drag and drop Intellectual Property (IP) that implement a Finite Impulse Response (FIR) Filter and the only input it requires is a Xilinx Coefficient File containing the filter coefficients. We use Matlab's Filter Designer Tool to generate this filter and use it in the Vivado Tool.

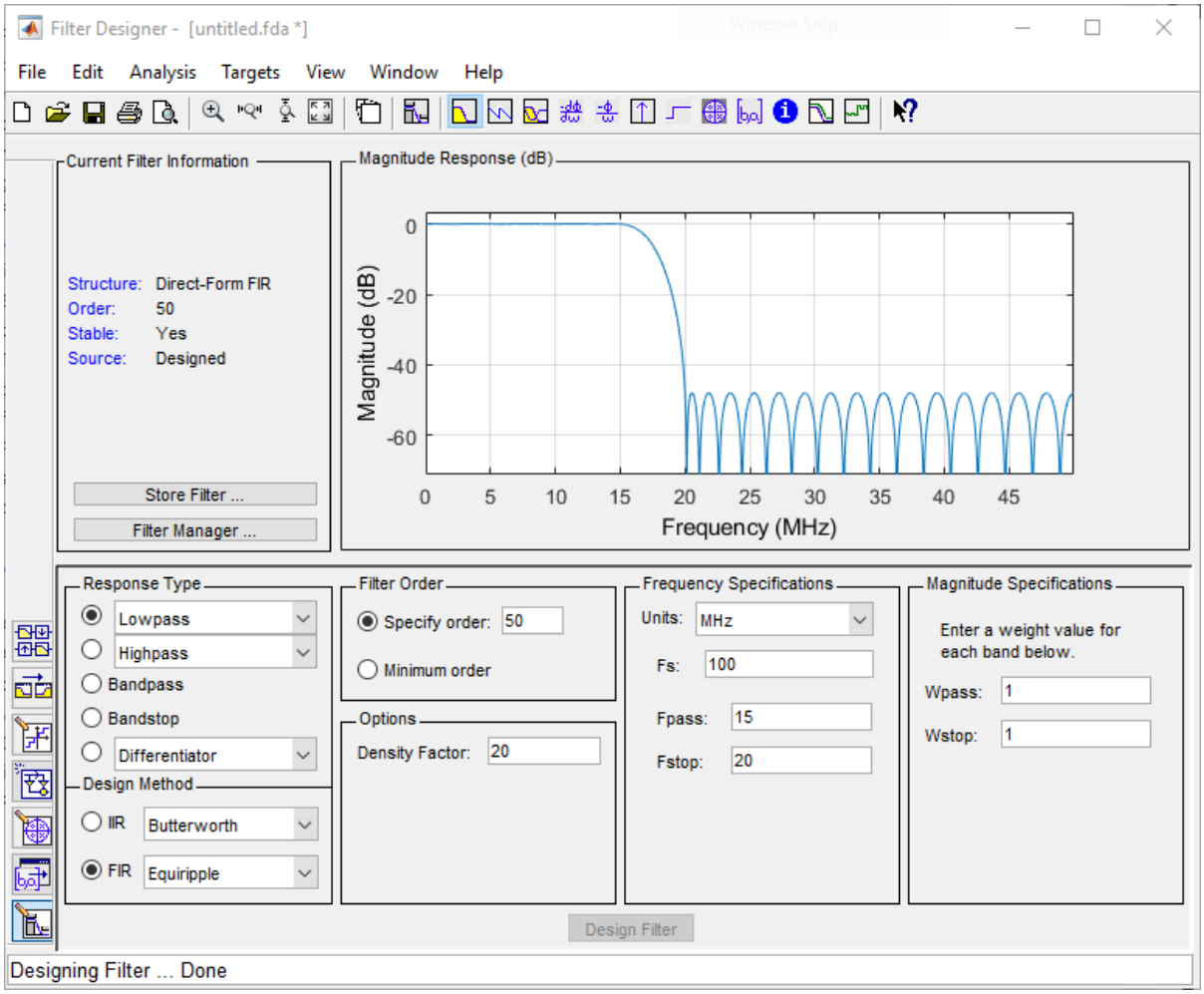


Figure 2.7: Filter Designer tool used in MATLAB to create a Low Pass - Equiripple Filter with a stop frequency of 20MHz, a pass frequency of 15MHz, and a sampling frequency of 100MHz.

Figure 2.7 shows the Filter Designer tool available through MATLAB. As can be seen from the figure, to generate the desired Filter coefficients, one has to simply choose the Filter topology they desire and fill in the prompts available through the interface to implement a filter with the intended specifications.

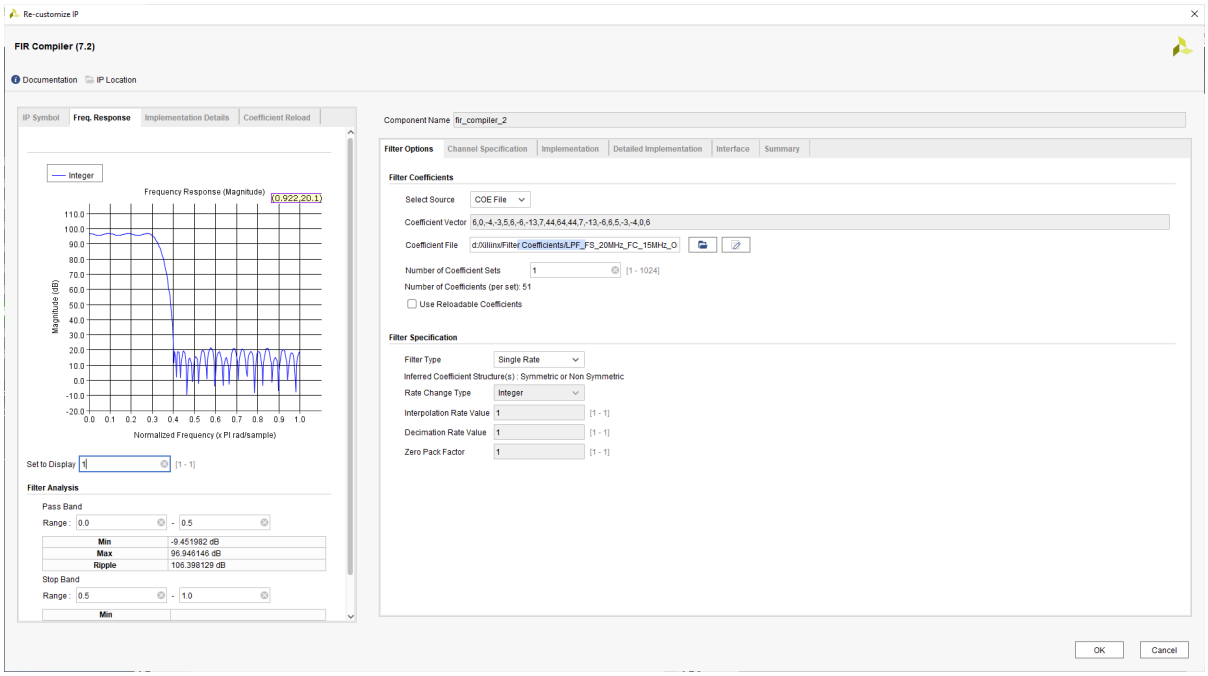


Figure 2.8: Screenshot of Vivado Window that allows us to import the Filter coefficients generated using MATLAB to implement them in Hardware

Figure 2.8 shows the Vivado Graphical User Interface (GUI), where the Xilinx Coefficient File created using MATLAB is entered. As can be seen by comparing the two plots in Figs 2.7 and 2.8, the filter coefficients have been imported successfully, and the "pass" and "stop" frequencies of the two Bode plots match perfectly.

CORDIC implementation for tan^{-1} computation

To compute the instantaneous phase of the IF signal, the I and Q signals are used as two arguments for the CORDIC. This block, used in the vectoring mode, is a slight variation of the architecture used in the seminal work done by Volder et al. [25]. Instead of using expensive multiplication and division operations to compute the arctangent, the algorithm in Figure 2.9 formulates the problem in such a way that it can be computed

iteratively with the use of simultaneous shift and add blocks, thereby substantially reducing area and power consumption. The traditional implementation of the CORDIC algorithm has a latency of 16 cycles and a throughput of 1 output (instantaneous phase) every 16 cycles. This implementation can further be improved by pipelining the computation. The pipelined architecture still has a latency of 16 clock cycles but has an improved throughput of 1 output (instantaneous phase) every clock cycle.

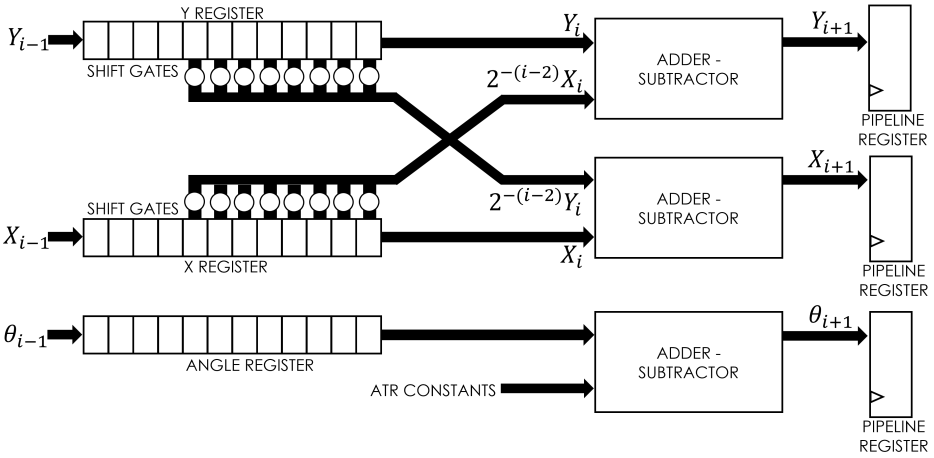


Figure 2.9: Digital Implementation of the CORDIC block that is used to calculate the instantaneous phase of the laser beat note [25]

Phase error calculation

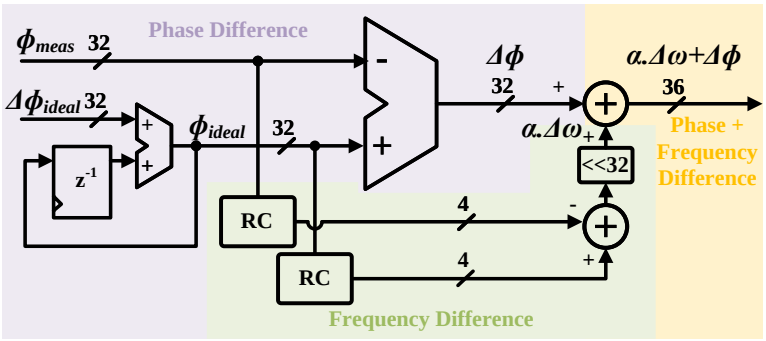


Figure 2.10: Digital Implementation of the Phase and Frequency Error Calculation block. RC: Rotation Counter

Figure 2.10 shows a functional diagram depicting the DSP that is used to calculate the phase and frequency error between the tunable laser and the microwave source. The ideal reference for the system is generated using an extremely stable RF clock reference. This RF source is used to clock an accumulator that accumulates a desired $\Delta\Phi_{ideal}$. The difference between this accumulated phase and the measured phase (output of the tan^{-1} block), yields the phase difference. However, as mentioned earlier, laser beat notes tend to have a lot of frequency jitter, and can make large frequency jumps. In such a case, it is desirable to have a loop with a large acquisition range. In conventional PLLs, when trying to implement synthesizers, these constraints require one to move from a simple Phase Detector (PD) to a Phase Frequency Detector (PFD). Highlighted in green in Figure 2.10 is the approach taken to convert the PD to a PFD. Two extra counters called "Rotation Counters" (labelled as RC in the figure) are also used. These counters count every time the phase of either the measured or the actual phase rolls over ($2\pi \implies 0$ transition). A difference of how often these roll over gives us a sense of the frequency difference between the two signals. Readers should note that even though these are implemented as two separate counters in the figure, in reality, this functionality is implemented as an UP/DOWN counter. When the ideal phase rolls over, the counter is incremented, and vice versa, when the measured phase rolls over, the counter is decremented.

Multiplier-less Loop Filter

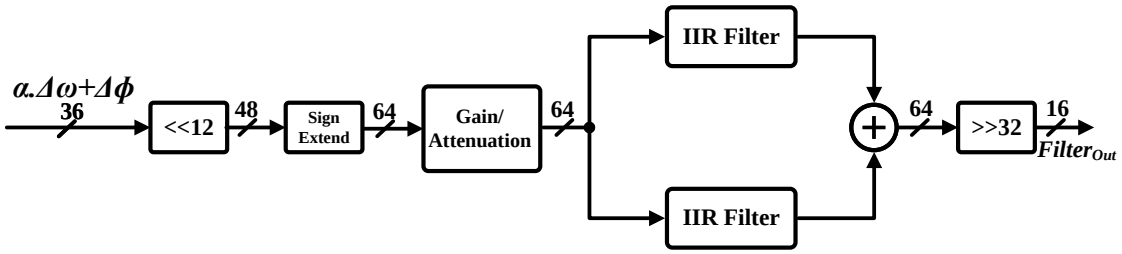


Figure 2.11: Loop Filter implementation. IIR: Infinite Impulse Response

The loop filter for the system is implemented as shown in Figure 2.11. Here, the 36 bits coming out of the phase error block are taken and scaled up by using a left shift operation ($<<<$ 12) to align the MSB to be in the 48th bit position. The reason to do this is so as to have sufficient bit space in the LSBs and MSBs, and so that no quantization errors are encountered during the filtering operations. Apart from the re-scaling, a gain/attenuation block is added in the signal chain. This block is necessary to increase or decrease the overall loop gain of the loop. The scaled input is then fed to two Infinite Impulse Response (IIR) filters whose outputs are summed together to get the final output. Once the filtering operations have been completed, the output is scaled back down to 16 bits. The reason it needs to be 16 bits is because that is the resolution of our DAC, but this output can easily be quantized to support a different resolution by appropriately choosing the right shift value.

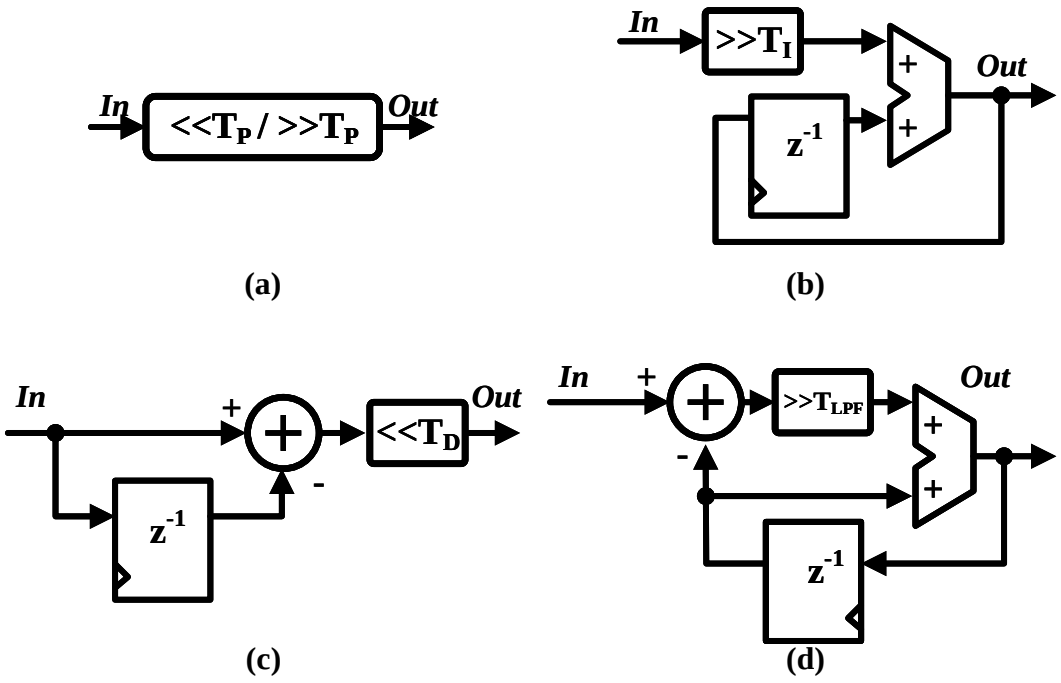


Figure 2.12: Infinite Impulse Response (IIR) Filter Implementation of (a) Proportional Control (b) Integrator (c) Differentiator (d) Low Pass Filter

Figure 2.12 shows the four options that are available to configure the IIR filter as. These filters can be set to be: (a) a proportional block, (b) an integrator, (c) a differentiator, or (d) a low pass filter. The outputs of each of these filters can be taken and summed together at the output to get a desirable frequency response. The novelty of these filter implementations is that all the filter coefficients involved in the various topologies are restricted to powers of 2. This restriction makes it so that every multiplication or division operation in the filter can be implemented as a simple shift left or shift right operation.

For the Integrator depicted in Figure 2.12(b), the Z-transform is given by Equation (2.1), and its magnitude response is shown in Figure 2.13. As is clear from the figure, the 0dB point for the filter moves to a lower location as the right shift value is changed

from $0 \rightarrow 5$.

$$H_{INT}(z) = \frac{1}{2^{T_I}} \frac{1}{1 - z^{-1}} \tag{2.1}$$

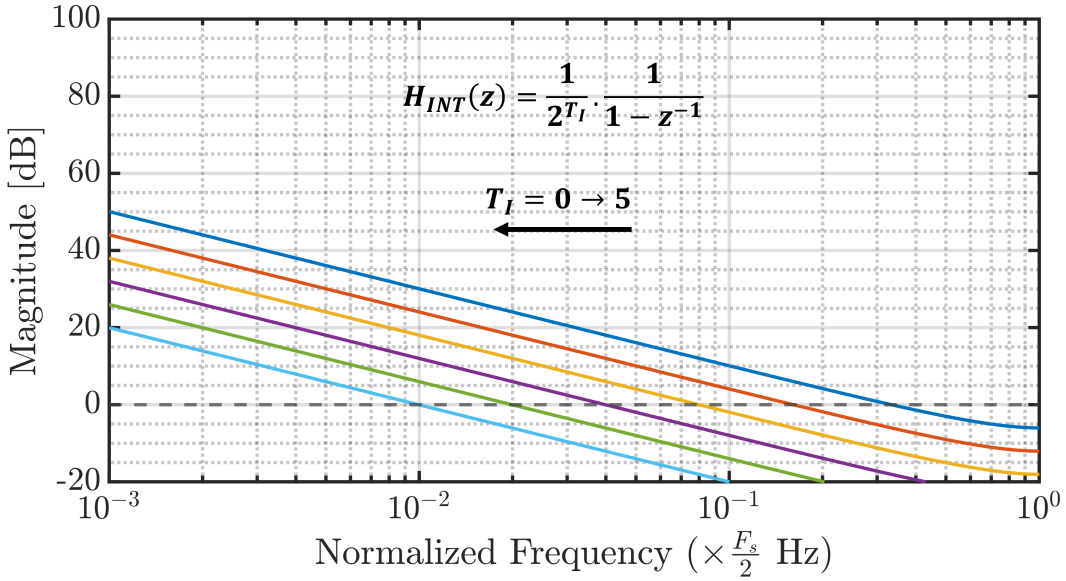


Figure 2.13: Magnitude Response of an Integrator as we change the filter coefficient T_I

Similar to Integrator, the Differentiator's (Figure 2.12(c)) Z-transform is given by Equation (2.2), and its magnitude response is given by Figure 2.14. The 0dB point for this filter also moves to a lower location as the right shift value is changed from $0 \rightarrow 5$.

$$H_{DIFF}(z) = 2^{T_D}(1 - z^{-1}) \tag{2.2}$$

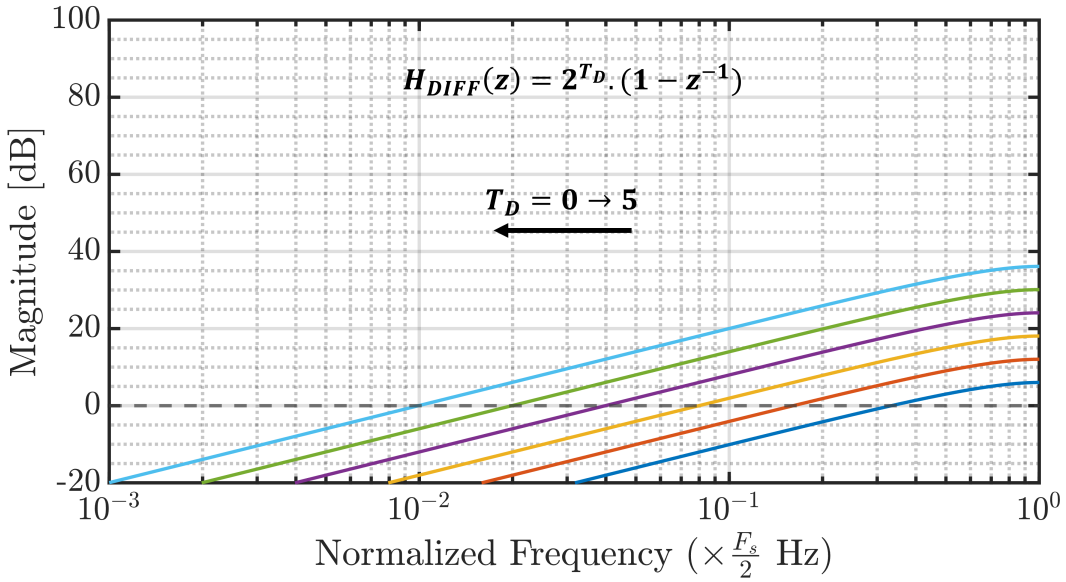


Figure 2.14: Magnitude Response of a Differentiator as we change the filter coefficient T_D

Similar to the previous cases, the Z-Transform the Low Pass Filter (Figure 2.12(d)) is given by Equation 2.3, and its magnitude response is given by Figure 2.15. One thing to note with the LPF, is that its magnitude at DC is 0dB, and by changing the value of the coefficient T_{LPF}, the location of its dominant pole (or 3-dB point) is pulled in.

$$H_{LPF}(z) = \frac{1}{2^{T_{LPF}}} \frac{1}{\left(1 - z^{-1}\left(1 - \frac{1}{2^{T_{LPF}}}\right)\right)} \tag{2.3}$$

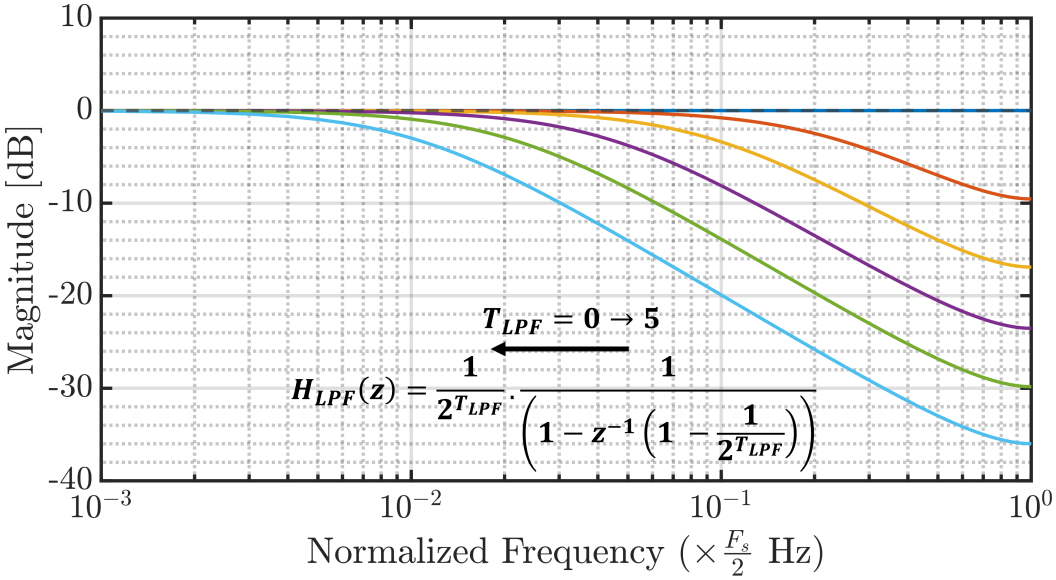


Figure 2.15: Magnitude Response of a Low Pass Filter as we change the filter coefficient T_{LPF}

2.2.2 Measurement Results

We will now present the measurement results obtained from the board level synthesizer with parts arranged as shown in Figure 2.5. During the measurements, the laser's beat note with the comb tooth (in this case, it is a reference laser emulating a comb tooth) was the measurement of primary interest. The beat note is converted to an electrical signal via a photoreceiver and further divided down for reasons mentioned earlier.The output of the frequency dividers is measured using a Frequency counter (Keysight 53230A). The first measurement to be made is the absolute stability of the laser. To measure this,the laser is set to a constant frequency set-point and its frequency deviations are observed over time. Figure 2.16 shows a measured made for a span of 20,000 secs (slightly more than 5.5 hours). The gate time for the frequency counter during this measurement was set to be 1 sec. As can be seen from the figure, the frequency deviation here is less a

100mHz and the standard deviation from the mean is 626μHz. The readers should note that the data presented in Figure 2.16 is the measurement made by the frequency counter *after* being multiplied by a factor of 16. This is done to account for the deviations divided by the frequency dividers in the path. Figure 2.17 shows the power spectrum of this beat note as measured by an electrical spectrum analyzer.

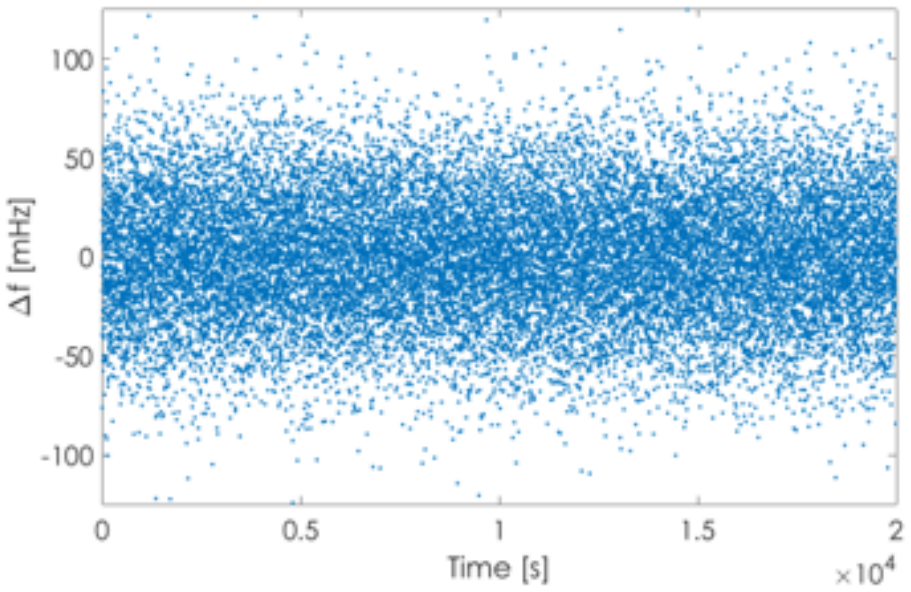


Figure 2.16: Current synthesizer output f_{out} deviation for 1-s gate time shows a maximum deviation of +/-100mHz

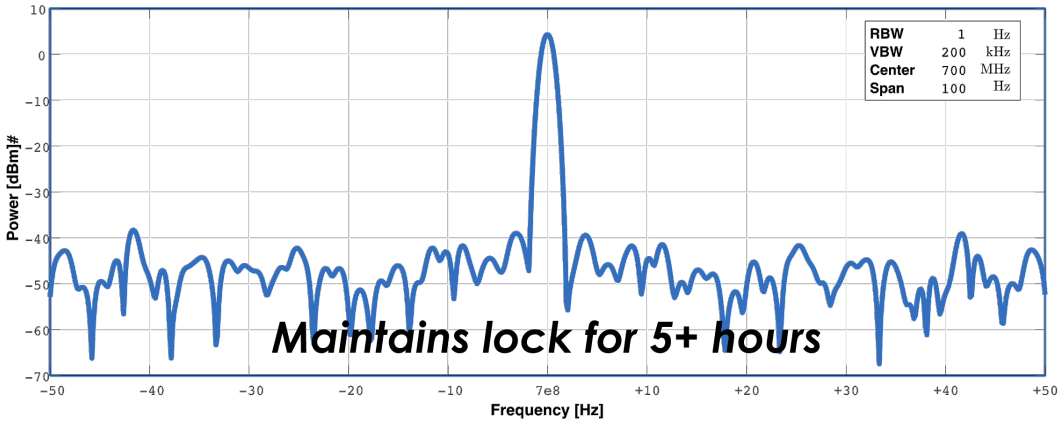


Figure 2.17: Power spectrum of the locked beat note

A frequency synthesizer would not be called so without its ability to 'synthesize' frequencies. Figure 2.18 shows the synthesizer stepping its output frequency in steps of less than 1 Hz. Readers are reminded that the absolute frequency of the laser at this point is 193,137,199,825,231.4 Hz and so, this is precision in the order of 1 part in nearly 200 billion!

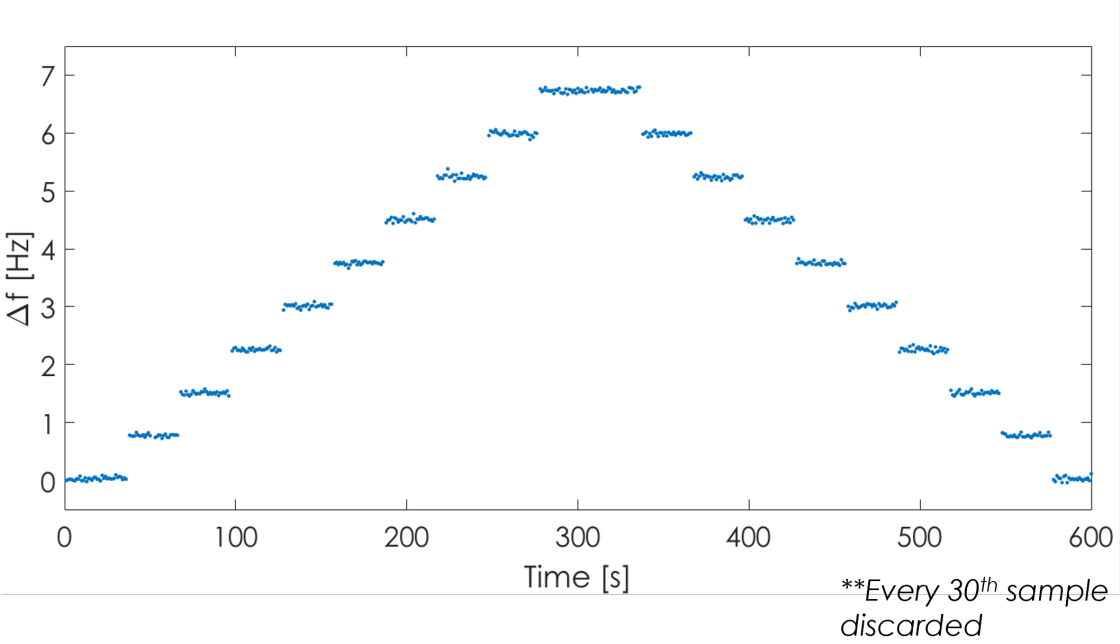


Figure 2.18: Bidirectional linear ramp of the synthesizer via step control of the laser offset PLL setpoint (100-ms gate)

2.3 ASIC Integration

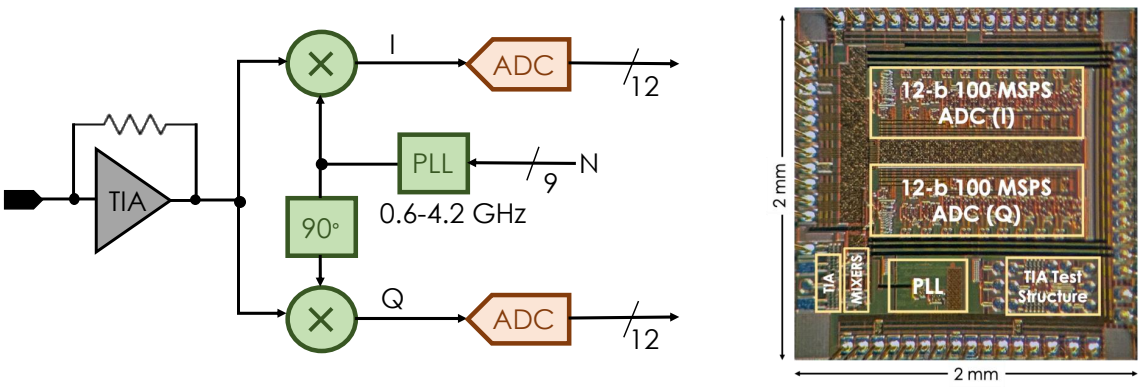


Figure 2.19: (a)Simplified schematic of signal chain (b) Photograph taken of the Integrated Circuit (IC) after fabrication

To reduce our size, weight, area, and power (SWAP), it was decided to implement the prototype circuit from the previous section as an Integrated Circuit (IC). The IC implementation offers several benefits over the board level implementation: apart from the obvious miniaturization benefit, a substantial boost in our power efficiency is also attained. Since the mixer, local oscillator (LO) and analog to digital converters (ADCs) are all integrated in the same chip, several of the power-hungry Low Noise Amplifiers (LNAs) that were present in the board level implementation can now be gotten rid of. The trans-impedance amplifier (TIA) used in the design also gives ability to sense photo currents that are in the order of nAs and improves the overall sensitivity of the system.

Figure 2.19 illustrates the signal chain implemented in the IC. The first circuit in the chain is the TIA which senses the photo-current coming out of the photo-diode whose job it is to convert the light coming from the beat note of the tunable laser with the reference laser into a current. This TIA converts the photo-current to voltage which can then be mixed down using two I-Q mixers who get their LO signals from an on-chip Phase Locked Loop (PLL). This PLL was designed to generate RF frequencies ranging from 600 MHz - 4.2 GHz, and generates two outputs that are 90° apart in phase.

Once mixed down by the mixer and filtered by two 1^{st} order RF filters, the Intermediate Frequency (IF) signals are then fed to two 12-bit 100 Mega Samples per sec (MSPs) ADCs. In order to meet the bit resolution and the sampling frequency requirements, these were implemented as pipeline ADCs. This was similar to the case at the board level with the only difference being in the sampling frequency of the off-the-shelf component (250MSPs, in that case). Once digitized, the outputs of these ADCs are then fed to an FPGA where they are digitally processed the same way as in case of the board level implementation.

A majority of the design and layout for this chip was carried out jointly by Sean McCotter and me. The TransImpedance Amplifier was designed and laid out by our

colleague Robert Costanzo at the University of Virginia, and the PLL was designed by our Professor Luke Theogarajan.

The following sections will describe how these blocks were implemented in further detail and discuss some of the trade-offs that were made in the design process.

2.3.1 Trans-impedance Amplifier (TIA)

The transimpedance amplifier used in the signal chain has been designed by our colleague Robert Costanzo and is the same as the one referenced in [26]. This can be seen in Figure 2.20, and consists of a common emitter amplifier with a negative feedback amplifier. This amplifier stage is then followed by a CMOS amplifier and active balun that generates differential signals for on-chip processing.

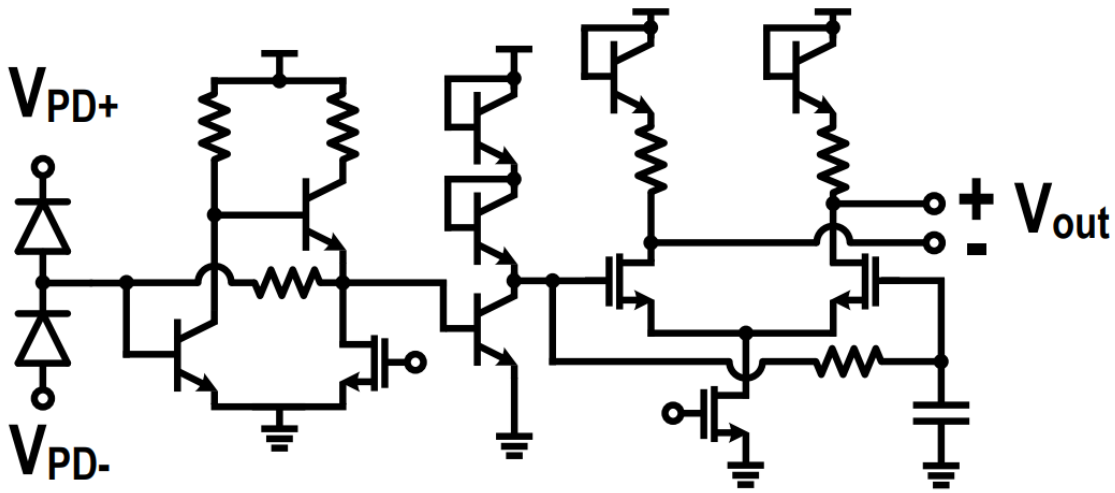


Figure 2.20: Circuit level schematic of the TIA. Biasing and output buffer circuitry omitted.

2.3.2 Phase Locked Loop (PLL)

The PLL used for this application requires a very wide tuning range of $\approx DC-4$ GHz. One way to support this range, is to fabricate several LC-tank oscillators in parallel and

multiplex between them depending on the frequency range you're operating in. Owing to their large inductor size, LC-tank oscillators consume a lot of area (especially at lower frequencies), and so this alternative would consume a lot of area. An added disadvantage of this topology would be that as oscillator is changed, it would also be required to multiplex between different loop filter configurations, which would mean that our design would require a lot of calibration time and would not be as robust, and susceptible to Process, Voltage and Temperature (PVT) variation.

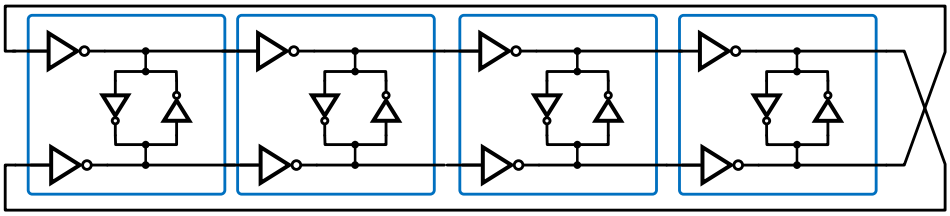


Figure 2.21: PLL Voltage Controlled Oscillator (VCO) - 4 stage pseudo-differential ring oscillator

All of these design considerations lead to the use of the ring oscillator structure shown in Figure 2.21. While not truly differential, this structure offers a very wide tuning range and a very linear frequency response to the applied input voltage. We can tune the output frequency of this oscillator by modulating the supply voltage to the pseudo-differential inverter pair. However, since the control voltage to the oscillator is now applied to a current hungry low-impedance node, using this ring oscillator necessitates the use of a voltage regulator after the loop filter (shown in Figure 2.22). This is different from the

case of a conventional PLL topology, where the output of the loop filter could be directly fed to the Voltage controlled oscillator (VCO). The linear regulator used in this case, in conjunction with the large PMOS driver driving the VCO input behaves as a two-stage amplifier, and is inherently unstable. This inherent instability needs to be compensated and its output pole needs to be adjusted in such a way that to the overall PLL loop, these poles are practically invisible. The loop filter and linear regulator compensation is carried out in such a way that their output poles are derived automatically based on the PLL's output frequency.

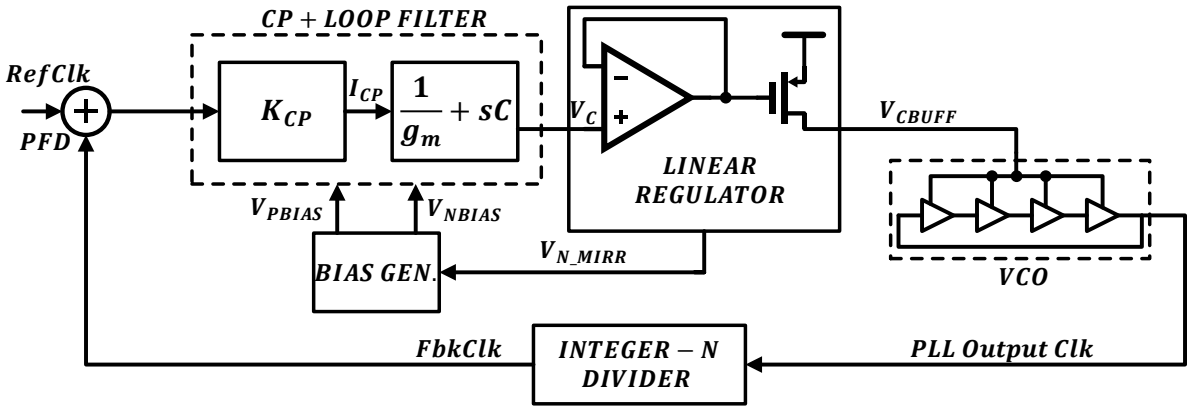


Figure 2.22: Simplified schematic of PLL topology

Closed Loop Response of a generalized PLL

The frequency response of a PLL can be analyzed using continuous time approximations as long as the loop bandwidth is a decade or more below the operating frequency of the loop. This bandwidth constraint has an added benefit that high-order poles that

exist due to the delay around the sampled feedback loop are virtually invisible to the loop, thereby ensuring the loop's stability.

Since there are two integrators present in the loop (charge pump current integrated by the loop filter and the frequency integration at the VCO output to get the phase response), the PLL is a second order system. The input output relationship of this feedback loop is given by:

$$P_O(s) = \left(P_{REF}(s) - \frac{P_O(s)}{N}\right) I_{CP} \left(R_{LF} + \frac{1}{sC_{LF}}\right) \frac{K_V}{s} \tag{2.4}$$

Where, I_{CP} is the charge pump current (A), R_{LF} and C_{LF} are the loop filter resistor and capacitors respectively, and K_V is the VCO gain (Hz/V). The closed loop response is then given by:

$$\frac{P_O(s)}{P_{REF}(s)} = \left(\frac{1}{N} + \frac{s}{I_{CP}(R_{LF} + 1/(C_{LF}))K_V}\right)^{-1} \tag{2.5}$$

$$\frac{P_O(s)}{P_{REF}(s)} = \frac{N(1 + sR_{LF}C_{LF})}{1 + sR_{LF}C_{LF} + s^2/(I_{CP}/C_{LF}.K_V/N)} \tag{2.6}$$

Comparing this to a second order system:

$$\frac{P_O(s)}{P_{REF}(s)} = N.\frac{1 + 2\zeta(s/\omega_N)}{1 + 2\zeta(s/\omega_N) + (s/\omega_N)^2} \tag{2.7}$$

we get the damping factor, ζ:

$$\frac{1}{2}\cdot\sqrt{\frac{1}{N}I_{CP}K_V R_{LF}^2 C_{LF}} \tag{2.8}$$

and the loop bandwidth, ω_N:

$$\frac{2\zeta}{R_{LF}C_{LF}} \tag{2.9}$$

For a second order system, when $\zeta = 1$, it is said to be critically damped and when this factor much greater than 1, the system is said to be over-damped.

For a typical PLL, I_{CP}, R_{LF},C_{LF}, and K_V are all constant and so the loop bandwidth and damping factors are constant as well. As mentioned before, for the loop filter to be stable the loop filter needs to be set at least a decade or more below the operating frequency. This means that if the loop bandwidth is fixed, it needs to be fixed for the worst case condition i.e. the slowest PLL operating frequency (lowest divide).

A PLL adjusts its output frequency when it is disturbed. What this means is that when the output frequency is varied, the phase error that results from this variation accumulates for several cycles till the loop corrects this change. The number of cycles this error accumulates is equal to the operating frequency divided by the loop bandwidth. For this reason, it is desirable to have loop bandwidth be as close to the reference frequency as possible. However, this becomes a problem when the loop bandwidth is fixed and has to be conservatively set a decade below the lowest operating frequency, which can oftentimes be as low as the reference frequency itself.

In the seminal work done by [27], the author proposed a method to keep both ζ, and the ratio of ω_n/ω_{REF} constant so as to have the best possible jitter performance.

To get a constant ζ across different operating frequencies, the authors proposed to

change the charge pump current I_{CP} proportional to the bias current to the delay element. Similarly, if we can set R_{LF} to vary inversely to the square root of the bias current, we can have a constant ζ across operating regions.

In the same way, since the output frequency is proportional to the square root of the bias current, we have a constant tracking bandwidth that is directly proportional to the ratios of the delay bias current and the bias current producing the R_{LF}.

Voltage controlled oscillator gain, K_V

For an inverter in the VCO shown in Figure 2.21, the propagation delay through a delay cell is given by:

$$t_{pd} = \int_{t_1}^{t_2} dt = \int_{0}^{V_{ctrl}/2} \frac{C_{load}(Vout)}{i_{ds}(Vout)} dVout \tag{2.10}$$

Assuming a constant load capacitance and an average current between constantly flowing between $t = 0$ and $t = t_{PLH}$

$$t_{pd} = \frac{C_{load}\Delta V}{I_{inv}} \tag{2.11}$$

where $I_{inv} = 0.5 * (i_{ds}(0) + i_{ds}(t_{PLH}))$

Defining this delay with respect to the supply voltage, V_{ctrl}:

$$t_{pd} = \frac{C_{load}.V_{ctrl}}{2.I_{inv}(V_{ctrl})} \tag{2.12}$$

Here, the average current, I_{inv} is a function of the supply.

With this definition of propagation delay we can calculate the operating frequency of

the VCO as:

$$F_{op} = \frac{1}{2.n.t_{pd}} = \frac{I_{inv}(V_{ctrl})}{C_{load}.V_{ctrl}.n} = \frac{I_{reg}(V_{ctrl})}{C_{load}.V_{ctrl}.n^2} \tag{2.13}$$

where n is the number of stages in the VCO.

To get the VCO gain, K_V from this we need to differentiate F_{op} w.r.t. V_{ctrl}

$$K_V = \frac{dF_{op}}{dV_{ctrl}} \tag{2.14}$$

$$K_V = \frac{d}{dV_{ctrl}}\left(\frac{I_{reg}(V_{ctrl})}{C_{load}.V_{ctrl}.n^2}\right) \tag{2.15}$$

$$= \frac{dI_{reg}}{dV_{ctrl}}\frac{1}{n^2 V_{ctrl} C_{inv}} - \frac{I_{reg}}{n^2 V_{ctrl}^2 C_{inv}} \tag{2.16}$$

$$= \frac{g_{m.reg}}{n^2 V_{ctrl} C_{inv}} - \frac{I_{reg}}{n^2 V_{ctrl}^2 C_{inv}} \tag{2.17}$$

Thus, the VCO gain, K_V is given by:

$$K_V = \gamma \frac{g_{m,reg} V_{ctrl} - I_{reg}}{V_{ctrl}^2 C_{inv}} \tag{2.18}$$

where the constant γ is related to the number of VCO stages by:

$$\gamma = \frac{1}{n^2} \tag{2.19}$$

Loop Filter Resistor, R_{LF}

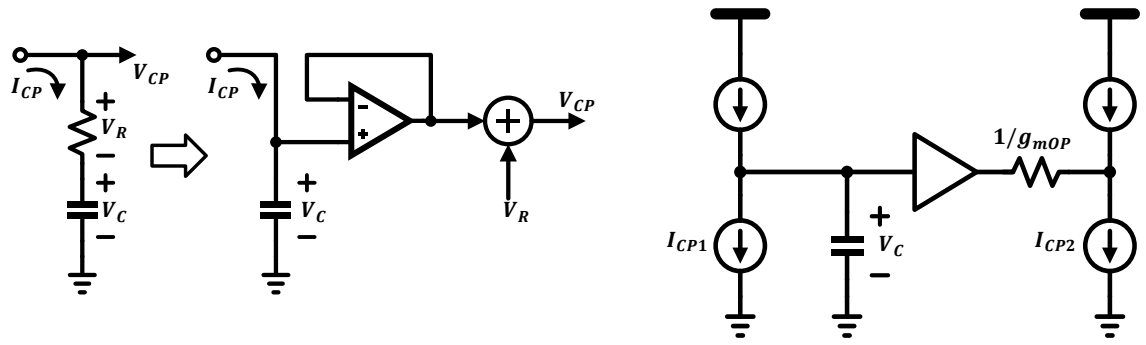


Figure 2.23: Implementing the stabilizing zero

Figure 2.23(a) shows a simplified schematic of how a typical second order loop filter is implemented in a conventional PLL. The first capacitor integrates the current output by the charge pump. Since this capacitor adds an additional pole at DC (and thus, makes the PLL loop unstable), we need a zero-ing resistor as shown in the figure to improve the phase margin right before the unity gain frequency. The voltage drops across both, the capacitor, and the resistor are then fed to the linear regulator which then drives the VCO.

Figure 2.23(b) shows an alternate way to generate this control voltage, V_C. Instead of using one charge pump to push/pull current across a resistor-capacitor pair, one can use two separate charge pumps to generate the potential drops across the resistor and the capacitor separately, and then add them up.

In [27] and [28], the authors first showed how the schematic in Figure 2.23(c) can be used to generate this control voltage. If s unity gain amplifier with an output resistance of $1/g_{m,LF}$ was used to buffer the capacitor voltage, and an appropriately sized charge pump copy at the output of this amplifier was added, the second charge pump can be used to control the location of this zero. An important detail that is often omitted from this discussion, however, is that this unity gain amplifier has to be a single-stage amplifier

for the output voltage to truly vary as $1/g_{m,LF}$ when a current is applied at the output. This is important because using a single stage amplifier then limits the drive capability, and output swing of the amplifier; which in turn, limits the size and tuning range of the VCO.

$$R_{LF} = \frac{1}{g_{m,LF}} = \alpha \frac{1}{g_{m,reg}} \tag{2.20}$$

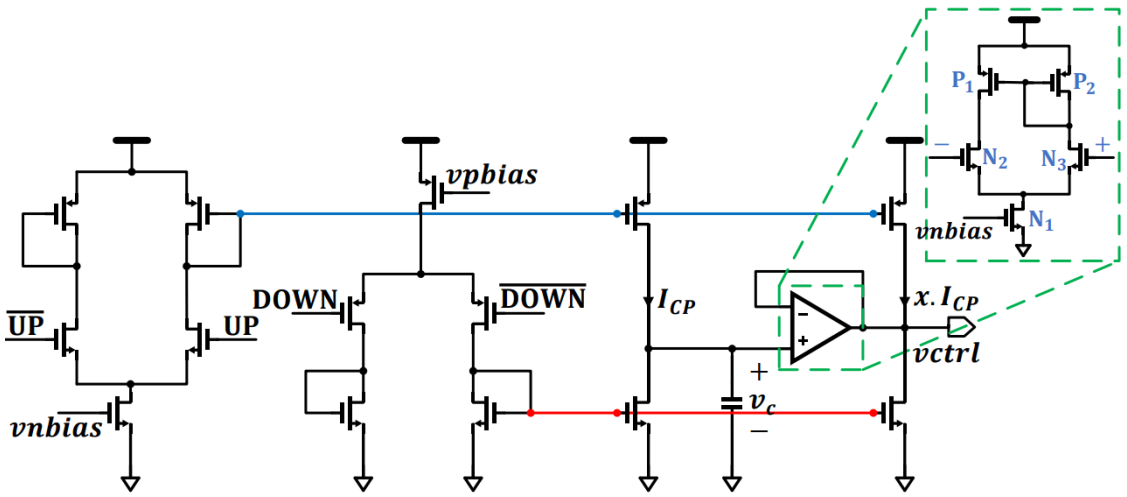


Figure 2.24: Circuit implementation of the charge pump and loop filter

Figure 2.24 shows the actual circuit implementation of the charge pump and loop filter. Both, the $UP/\overline{UP}$ and $DOWN/\overline{DOWN}$ signals are used to drive differential loads and the current is mirrored to create the charge pump. Using both signals allows the current from the tail current sources to always flow and the mirror circuitry avoids any possible coupling from the sharp edges of the $UP/\overline{UP}$ and $DOWN/\overline{DOWN}$ signals.

As shown in Figure 2.23(c), the first charge pump integrates the current onto a capacitor which is then fed to an amplifier configured as a unity gain buffer. In this amplifier, the input nmos, $N2$ provides the $1/g_m$ resistor. An important point to note here is that the *vnbias* and *vpbias* voltages are common between the charge pump biases and the amplifier bias, in addition to this, all the transistors used as current sources here are integral multiples of the same unit transistor. What this means is that the current sourced by all the current sources are all integral multiples of the same current.

Linear Regulator

The linear regulator used here is similar to the topology used in [29]. A simplified schematic of this Regulator can be seen in Figure 2.25. As can be seen from the figure, the amplifier driving the PMOS driver with a resistive load, is effectively a 2-stage amplifier. As with the other second order systems mentioned previously, this is inherently an unstable system, and needs to be compensated. Conventionally, this 2-stage amplifier is compensated as shown in Figure 2.25(a). Adding a large compensation capacitor, C_C creates a 'dominant pole' and the compensation resistor, R_C provides a left-half plane (LHP) zero that provides a phase bump right before the open loop unity gain frequency.

As shown by Figure 2.25(b), the same technique can be used in the loop filter to implement R_C by buffering the voltage across C_C and forcing a mirrored fraction of the driver current into the output of the buffering amplifier.

A detailed analysis of the loop gain transfer function can be found in [29]. It shows that the we can show that the damping of the loop is independent of the unity-gain bandwidth and only depends on geometrical ratios of the devices and capacitors used.

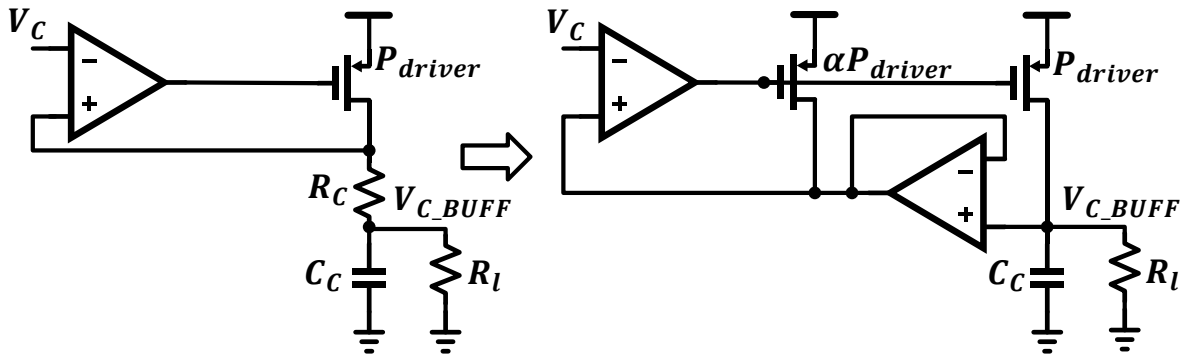


Figure 2.25: Simplified schematic of the linear regulator used in the PLL

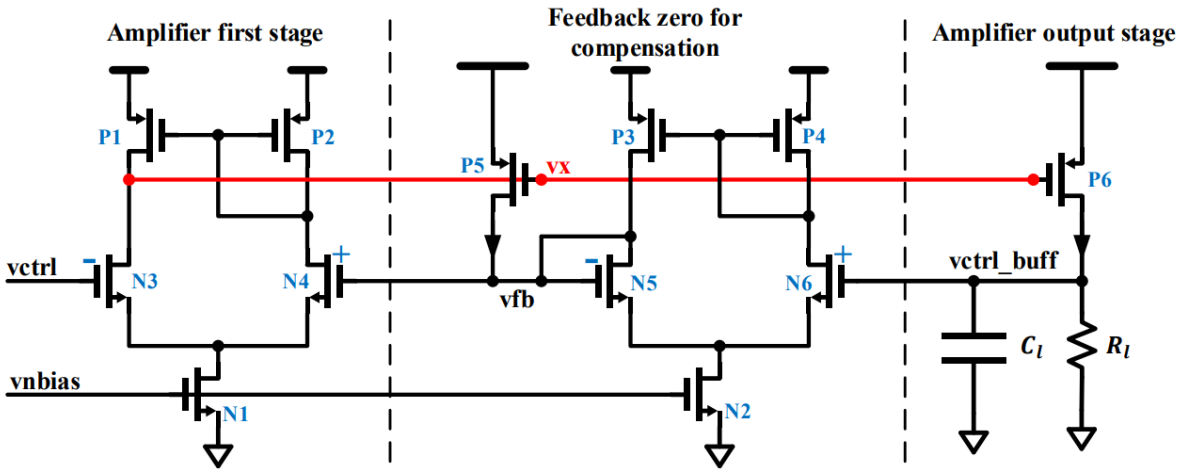


Figure 2.26: Circuit implementation of the linear regulator and its compensation

Bias Generator

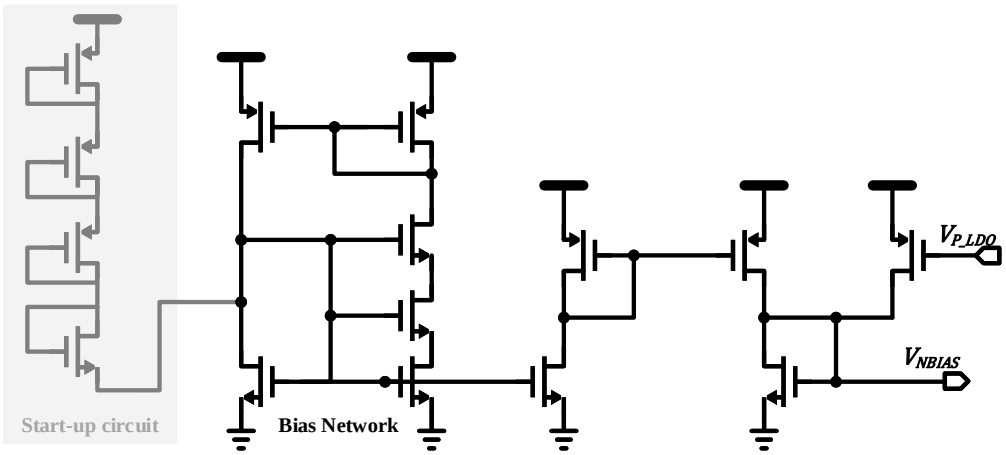


Figure 2.27: Circuit implementation of the Bias Generator

Figure 2.27 shows how the biases for the charge pump and loop filter are generated. The bias generator uses a β-multiplier topology to generate an initial bias during startup, which prevents the bias generator from collapsing. However, the majority of the current used to generate the output 'vnbias' is generated by using the LDO output voltage. This ensures that the same amount of current flows through the nmos generating the bias current as through the VCO.

These bias voltages are then used to generate the charge pump current, I_{CP}, which is given by:

$$I_{CP} = \beta I_{reg} \tag{2.21}$$

Level Shifter

The control voltage for our oscillator is the supply for the delay cells, and so the output voltage of the oscillator varies from 0 to V_{ctrl}. However, all subsequent circuits

that use the output of this oscillator expect a full scale signal going from 0 to V_{supply}. The circuit shown in Figure 2.28 is used to complete this function.

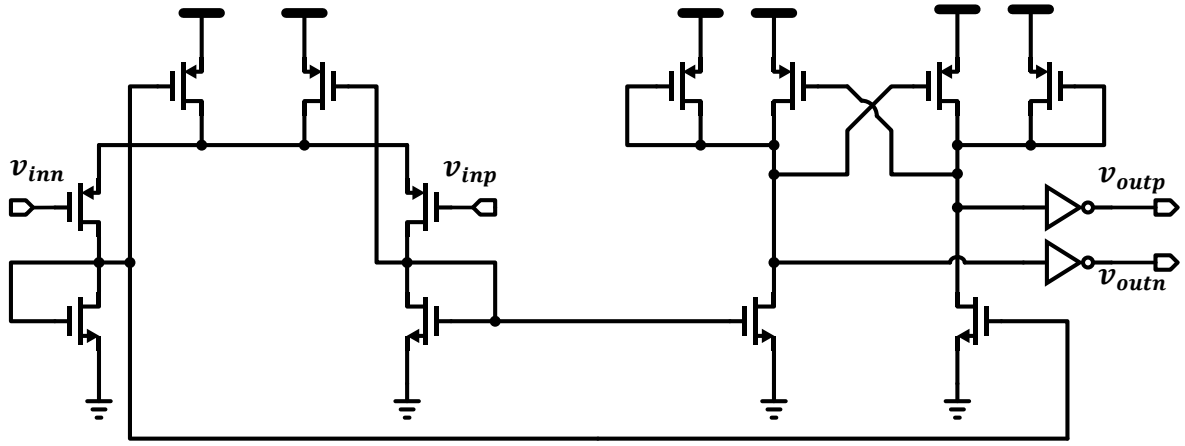


Figure 2.28: Circuit implementation of the Level Shifter

The circuit is a two stage amplifier with a low gain first stage and high gain output stage with "infinite" differential load resistors. The first stage is a fully differential version of the self biased amplifier first introduced in [30].

Adaptive Bandwidth PLL - Analysis

Putting all the pieces together, it can now be shown how the bandwidth of the loop tracks the PLL's output frequency. Before beginning, a few facts need to be reiterated. The bias generator generates the bias for all analog circuits in the design using the supply current to the VCO. This means that the current flowing through the charge pump and loop filter amplifiers are proportional to the VCO supply current.

Inserting values from Equations 2.21, 2.18, and 2.20 into 2.8, and 2.9, we get:

$$\zeta = \frac{1}{2}\sqrt{\frac{1}{N}\beta I_{reg}\gamma\frac{g_{m.reg}V_{ctrl} - I_{reg}}{V_{ctrl}^2}\alpha^2\frac{1}{g_{m,reg}^2}\frac{C_{LF}}{C_{inv}}} \tag{2.22}$$

$$\zeta = \frac{1}{2}\sqrt{y\frac{\left(g_{m.reg}\frac{V_{ctrl}}{I_{reg}} - 1\right)}{\left(g_{m,reg}\frac{V_{ctrl}}{I_{reg}}\right)^2}} \tag{2.23}$$

Where y is a constant given by:

$$y = \alpha^2\beta\gamma\frac{C_{LF}}{NC_{inv}} \tag{2.24}$$

The $g_{m,reg}$ will be larger than the load resistor given by $R_{Load} = \frac{V_{ctrl}}{I_{reg}}$. Therfore $g_{m,reg}\frac{V_{ctrl}}{I_{reg}} \gg 1$, which yields

$$\zeta \cong \frac{1}{2}\sqrt{y\frac{I_{reg}}{g_{m,reg}V_{ctrl}}} \tag{2.25}$$

Similarly,

$$\frac{\omega_N}{\omega_{REF}} = \frac{N}{2\pi F}.\frac{2\zeta}{R_{LF}C_{LF}} \tag{2.26}$$

$$= \frac{Nn^2V_{vctrl}C_{inv}}{2\pi I_{reg}}.\frac{2\zeta}{C_{LF}R_{LF}} \tag{2.27}$$

$$= \frac{1}{2\pi}.\frac{Nn^2V_{vctrl}}{I_{reg}}\frac{C_{inv}}{C_{LF}}\frac{g_{m,reg}}{\alpha}.\sqrt{\alpha^2\beta\gamma\frac{C_{LF}}{NC_{inv}}\frac{I_{reg}}{g_{m,reg}V_{ctrl}}} \tag{2.28}$$

$$= \frac{1}{2\pi}\sqrt{\frac{NC_{inv}}{C_{LF}}\frac{\beta}{\gamma}\frac{g_{m,reg}V_{ctrl}}{I_{reg}}} \tag{2.29}$$

$$= \frac{1}{2\pi}\sqrt{z\frac{g_{m,reg}V_{ctrl}}{I_{reg}}} \tag{2.30}$$

Where z is a constant given by:

$$z = \frac{NC_{inv}}{C_{LF}} \frac{\beta}{\gamma} \tag{2.31}$$

In the analysis above, we assume that the PLL has locked to the correct frequency and so the operating frequency is N times the reference frequency.

$$\omega_{ref} = 2\pi \left(\frac{F_{op}}{N} \right) \tag{2.32}$$

Here, F_{op} is the same as defined in eq. 2.13

Adaptive Bandwidth PLL - Case Study

In the case of the PLL implemented on chip,

$$R_{LF} = \frac{1}{g_{m,follower}} = \alpha \frac{1}{g_{m,reg}} \rightarrow \alpha = \frac{64}{4}$$

$$I_{CP} = \beta I_{reg} \rightarrow \beta = \frac{1}{64}$$

$$\frac{C_{LF}}{C_{inv}} \cong 1000$$

For our case, we have a stage VCO,

$$\gamma = \frac{1}{2^2} = \frac{1}{4}$$

For a divide value with $N = 30$, ζ will be given by:

$$y = \alpha^2 \beta \gamma \frac{C_{LF}}{NC_{inv}}$$

$$y = \left(\frac{64}{4}\right)^2 . \left(\frac{1}{64}\right) . \left(\frac{1}{4}\right) . \left(\frac{1000}{30}\right) \cong 33.34$$

$$z = \frac{NC_{inv}}{C_{LF}} \frac{\beta}{\gamma}$$

$$z = \frac{30}{1000} . \frac{4}{64} = 1.875 \times 10^{-3}$$

Thus, for our PLL the damping factor and Bandwidth ratio are given by:

$$\zeta \cong \frac{1}{2} \sqrt{33.34 \frac{I_{reg}}{g_{m,reg} V_{ctrl}}}; \frac{\omega_N}{\omega_{REF}} = \frac{1}{2\pi} \sqrt{1.875 \times 10^{-3} \frac{g_{m,reg} V_{ctrl}}{I_{reg}}}$$

Figure 2.29(a) and (b) show how these values vary with the control voltage. The function $\frac{I_{reg}}{g_{m,reg} V_{ctrl}}$ is extracted using a SPECTRE simulation.

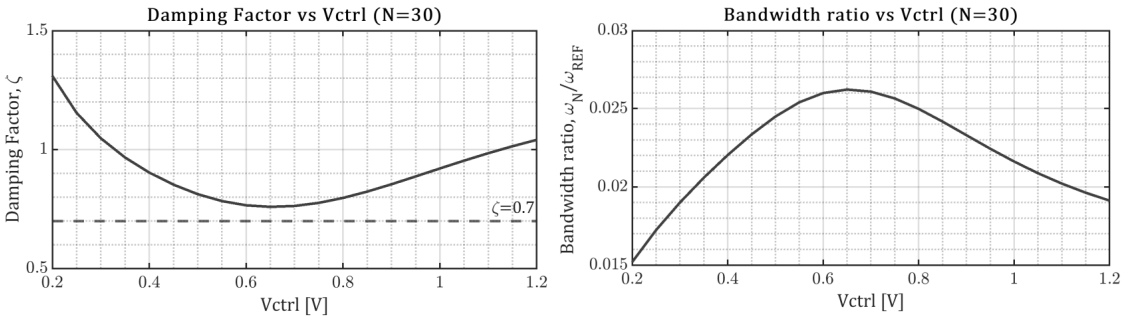


Figure 2.29: (a) Damping Factor variation as a function of control voltage, V_{ctrl} (b) Bandwidth ratio variation as a function of control voltage, V_{ctrl}

As can be seen from 2.29(a), the damping factor ζ never goes below 0.7 and even though the bandwidth ratio is not constant, as it would be with older technologies, it does not vary wildly with different operating voltages.

2.3.3 Mixer - Gilbert Cell

The mixer topology used in the heterodyne receiver is shown in Figure 2.30. This commonly used topology has been analyzed extensively in [31], and works by effectively multiplying the incoming Radio Frequency (RF) and Local Oscillator (LO) voltages - equivalent to a convolution in the frequency domain.As described in [31], these active mixers perform the mixing operation in three steps: they convert the incoming RF voltage to a current, they steer the RF current using the LO and then convert the Intermediate Frequency (IF) current to a voltage. Here, the conversion gain can be given by the ratio of the Current-Voltage (I/V) gain at the output (output resistor) to the Voltage-Current (V/I) gain at the input (transconductance of the input stage).

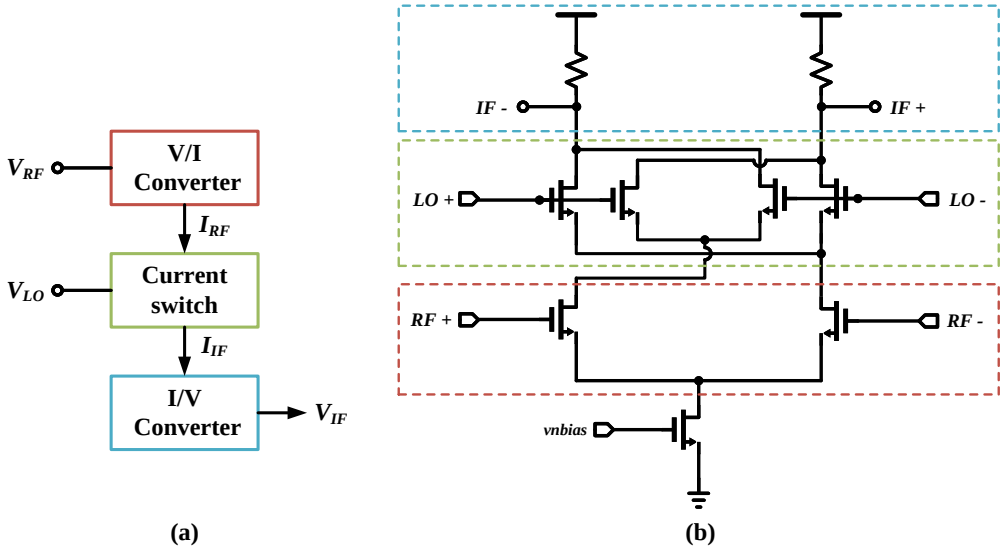


Figure 2.30: (a) Functional description of Mixer cell (b) Circuit implementation of the double balanced mixer

Mathematical description

Let us assume that the incoming RF signal is a single tone sine wave and is given by:

$$V_{RF} = cos(\omega_{RF} t) \tag{2.33}$$

For ideal operation, we want an LO signal that is a square wave which can be described as:

$$V_{LO} = \begin{cases} +1 & \text{if } \mathrm{mod}(t, \frac{1}{f_{LO}}) < \frac{1}{2*f_{LO}} \\ -1 & \text{otherwise} \end{cases} \tag{2.34}$$

The fourier transform for this function is given by:

$$V_{LO} = \frac{4}{\pi} \sum_{n=1,3,5,..}^{+\infty} \frac{1}{n} sin(n\omega_{LO}t) \tag{2.35}$$

As mentioned before, a mixer performs a convolution operation in the frequency domain and multiplies these two voltages together. And so the output can be described as:

$$V_{IF} = \frac{4}{\pi} \sum_{n=1,3,5,..}^{+\infty} \frac{1}{n} \left(sin(n\omega_{LO}t).cos(\omega_{RF}t)\right) \tag{2.36}$$

$$= \frac{2}{\pi} \sum_{n=1,3,5,..}^{+\infty} \frac{1}{n} \left(sin(n\omega_{LO}t + \omega_{RF}t) + sin(n\omega_{LO}t - \omega_{RF}t)\right) \tag{2.37}$$

The desirable term from Equation (2.37) - $sin(\omega_{LO}t + \omega_{RF}t)$, can easily be separated from the other terms by using a low pass filter with a corner frequency appropriately placed much lower than ω_{LO} and ω_{RF}. An additional benefit of using the topology given in Figure 2.30(b) has an added benefit of having a fully differential output. This kind of output minimizes unwanted non-linearities and switching effects (coupling capacitances, common-mode gain).

2.3.4 12-bit 100MSPs Analog to Digital Converter (ADC)

From the board level prototype of the overall system, it was known that to get Hz level precision at least 12 bits of resolution was needed. At the same time, because the Intermediate Frequency (IF) signal coming out of the mixer has a max frequency of 25MHz, the sampling frequency needs to be at least 100MHz.

Given these specifications, and the technology available, the ADC was decided to be a pipeline ADC. The sampling frequency is too high for an over-sampled data converter

such as a Delta-Sigma ADC. And while there have been demonstrations of 100MSPs Successive Approximation ADCs (SAR ADCs), it was not a feasible option to design in the process available(BiCMOS8HP - 130nm).

Design Overview

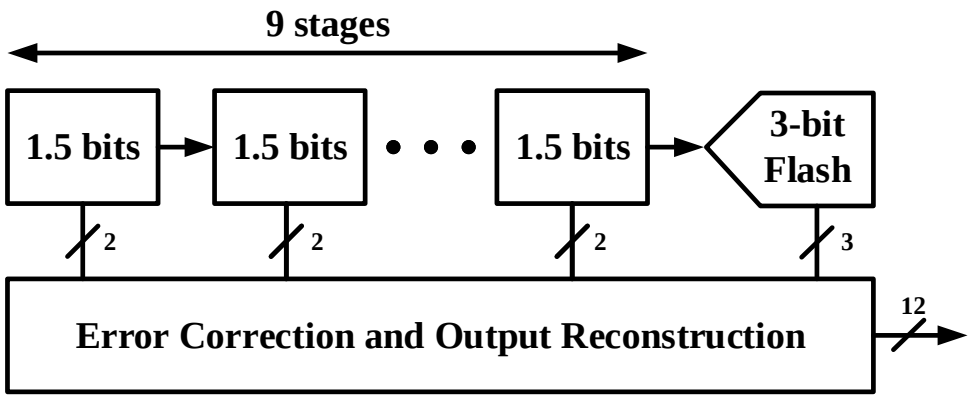


Figure 2.31: Overall Architecture of 12 bit Pipeline ADC

Figure 2.31 shows the overall architecture of the pipeline ADC implemented. The ADC consists of nine successive 1.5-bit stages followed by a final stage which is a 3-bit Flash ADC. The results of each of these stages is directly output off-chip where they are combined to give a 12-bit output which corresponds to the digital representation of the analog input sampled by the ADC.

Figure 2.32 shows a block diagram representation of the various parts of the 1.5-bit stage. The input to each stage is fed to both a sample and hold block and to a 3-level quantizer. The outputs from both of these blocks are fed to the heart of the pipeline ADC, the mDAC.

The mDAC uses these inputs and gives an output corresponding the following formula:

$$V_{OUT} = 2 * V_{IN} + B.V_{REF} \tag{2.38}$$

where B depends on the output of the 3-level quantizer, and is given by:

$$B = \begin{cases} +1 & \text{if } X = 1 \\ 0 & \text{if } Y = 1 \\ -1 & \text{if } Z = 1 \end{cases}$$

This Equation can be easily derived by analyzing the circuit implementation of the mDAC (shown in Figure 2.33; single ended version shown for simplicity). The entire operation takes two non-overlapping phases of a clock ϕ_S and ϕ_{res}. During the sampling phase, ϕ_S, the two sampling switches close and sample the input on two capacitors given by C_S and C_F.

In the next phase of the clock, the residual amplification phase, ϕ_{res}, the capacitor C_F is put in feedback and due to the conservation of charge, the charge from the capacitor C_S is transferred to it. The amount of charge transferred depends on the quantizer decision and so, it is proportional to $V_{IN} + V_{REF}$, V_{IN} or $V_{IN} - V_{REF}$.

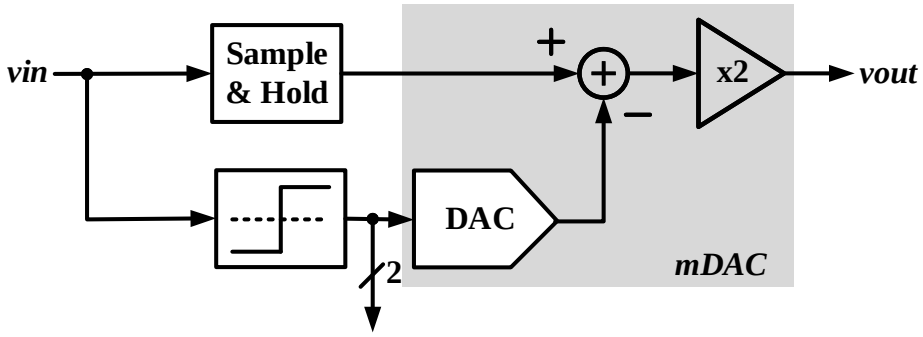


Figure 2.32: 1.5 bit per stage implementation

Each of these stages has an extra 0.5-bit of redundancy built into them which relaxes the requirements of each stage's quantizer and opamp by half an LSB. Figure 2.34 [32] shows the sources of some of these non-idealities and shows how they might affect the quantizer decision correspondingly.

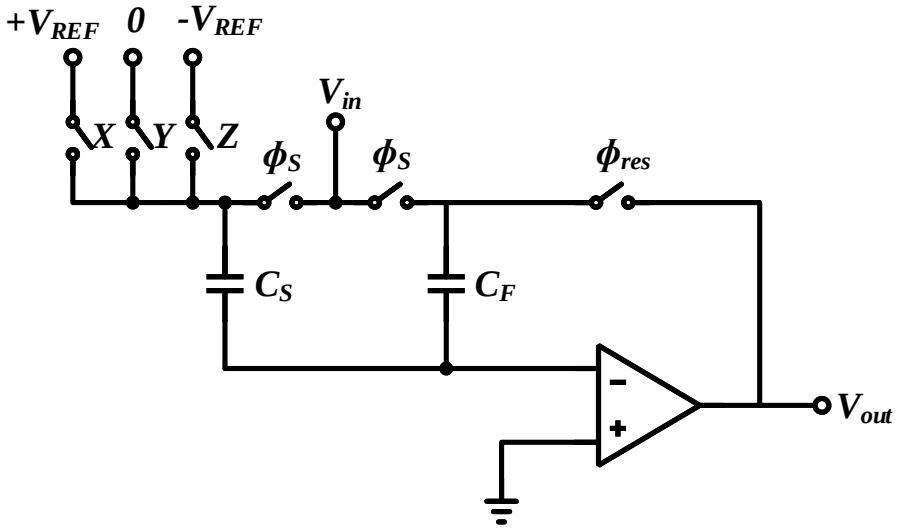


Figure 2.33: Circuit implementation of 1.5 bit stage mDAC

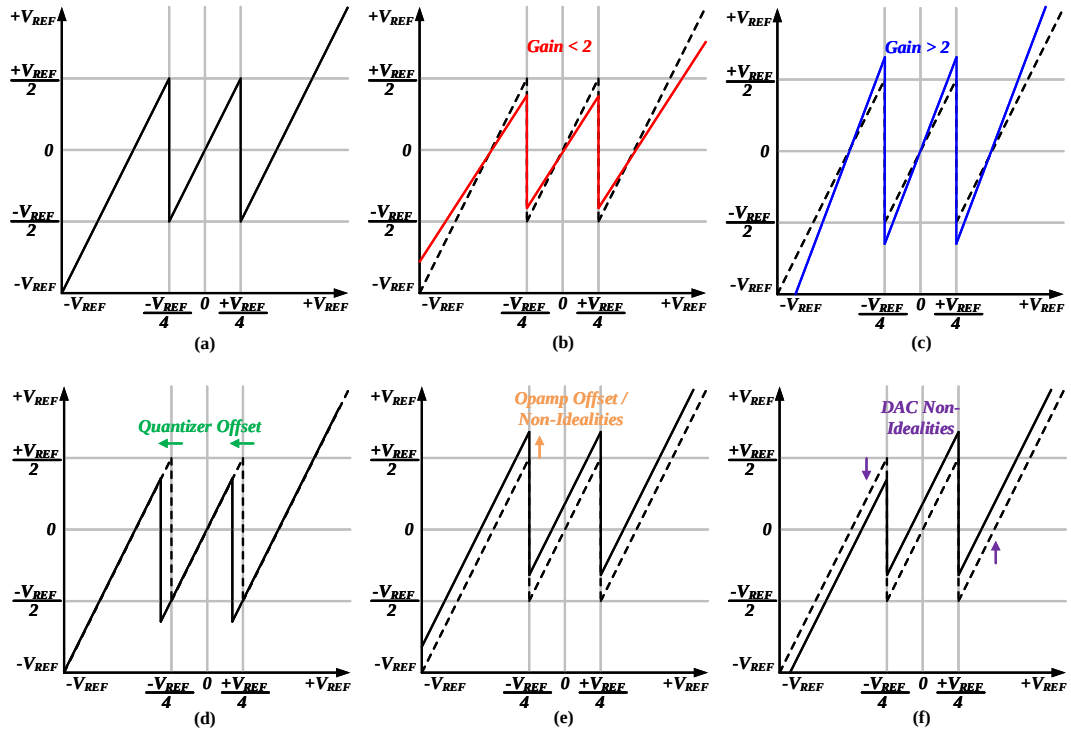


Figure 2.34: 1.5 bit per stage non-idealities

The idea behind this redundancy is essentially to defer decisions about signals that are not too small or too big to the next stage. [32] illustrates this very clearly in Figure 2.35. Figure 2.35 (a) shows ideal operation with an input of 0.1V. In this case the correct decision is made in each stage of the pipeline and follows the behavior described in Equation 2.38. Figure 2.34 (b) shows what happens when there is an incorrect quantizer decision made in the second stage. Here, as long as the error is less than $\frac{1}{2}LSB = \frac{V_{REF}}{4}$, as illustrated by the output bits, we still get the correct answer.

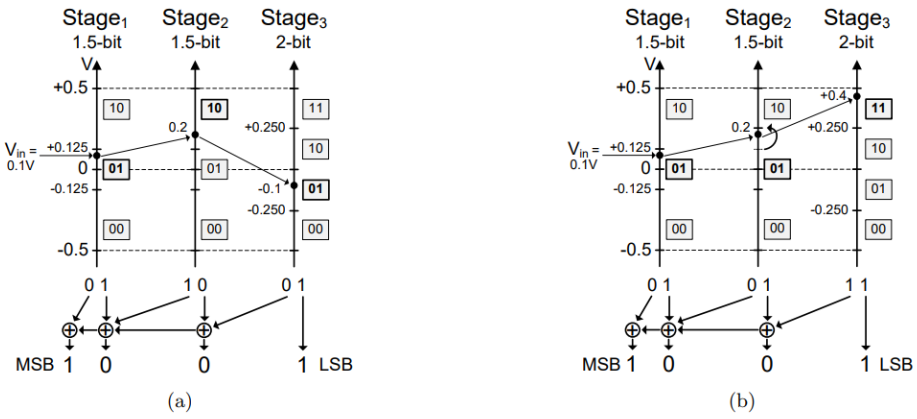


Figure 2.35: Digital Error Correction (a) Ideal Operation (b) Operation with an error present in the quantizer [32]

Design Methodology

For the ADC, the first stage has the strictest requirements if we intend to hit the required Effective Number of Bits (ENOB). For our design, we operate using a $2.5V$ supply and have a $V_{REF} = 1V$. This sets the quantization noise floor as $V_{n,rms} = \frac{V_{LSB}}{\sqrt{12}} = 70.48uV$.

If we want this noise to be our dominant noise source, this sets the capacitor size.

$$\frac{kT}{C_S} < 70.48uV \implies C_S = 972fF(80^{\circ}C) \tag{2.39}$$

The drawback with capacitors this big is that it restrains our operating frequency. We still need the output of the mDAC to settle within half a clock period. It also restricts the maximum allowed switch resistance given by:

$$R = \frac{T_S}{2C_S(N+1)ln(2)} \implies 515\Omega \tag{2.40}$$

The maximum tolerable error (as mentioned in the previous section) is $\frac{V_{LSB}}{2}$. This determines the gain required for our amplifier. The minimum error is given by:

$$\frac{V_{LSB}}{2} > \frac{1}{1+A} \implies A > 78dB \tag{2.41}$$

Similar to the case with the switch resistance, we would like the amplifier output to settle within half an LSB. This gives us the bandwidth requirement:

$$BW > \frac{2(N+1)ln(2)}{2\pi T_S} = 575MHz \tag{2.42}$$

Operational Transconductance Amplifier (OTA)

As mentioned in the previous section, it was required to design an amplifier with a DC Gain of at least $78dB$ and a Gain Bandwidth Product (GBP) of at least 575MHz. Even though in the previous section we demonstrated mDAC operation using a single ended amplifier, in practice, a fully differential amplifier was preferred. This is because fully differential amplifiers have the ability to reject any common-mode noise present in our circuit.

Folded Cascode Amplifier

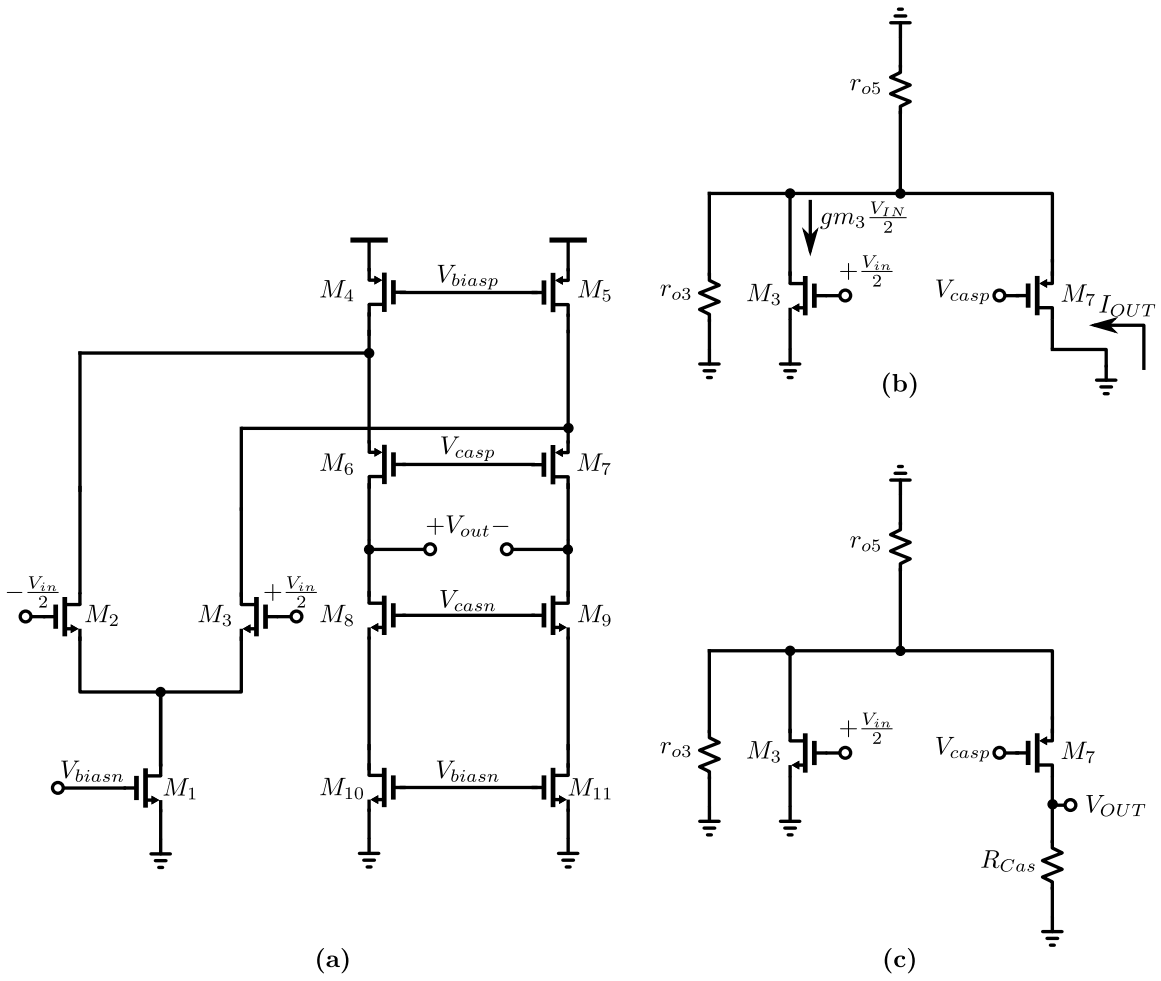


Figure 2.36: (a) Circuit Implementation of Folded Cascode Amplifier (b) Small signal model of half-circuit with output shorted to ground (c) Equivalent circuit with output open

Figure 2.36(a) shows the circuit implementation of a folded cascode amplifier. This topology has the small signal gain of a telescopic cascode amplifier but does not suffer from the same voltage swing and headroom limitations as it. To determine the small signal voltage gain of this amplifier, Figs 2.36(b) and (c) need to be analyzed. Figure 2.36(b) helps determine the equivalent G_m, while Figure 2.36(c) demonstrates a way

to calculate the output resistance R_{OUT}. Combining these two, the overall gain of the amplifier $|A_V| = G_m * R_{OUT}$ can be determined.

As can be seen from Figure 2.36(b), most of the output current, I_{OUT} flows through the drain of the input transistor, M_3. This is because the impedance looking into the transistor M_7 ($gm_7^{-1}||r_{o7}$) is less than $r_{o3}||r_{o5}$. And so, $G_m \approx gm_3$.

Similarly, analyzing 2.36(c), gives the output resistance of the circuit.

$$R_{OUT} = R_{cas}|| \left((r_{o3}||r_{o5})(1 + gm_7 r_{o7})\right) \tag{2.43}$$

where R_{cas} is the resistance seen in to the drain of M_9. This resistance is the cascode resistance, which is the drain resistance of the transistor M_{11} amplified by the gain of the transistor M_9.

$$R_{cas} = r_{o11}(1 + gm_9 r_{o9}) \tag{2.44}$$

Gain Boosting

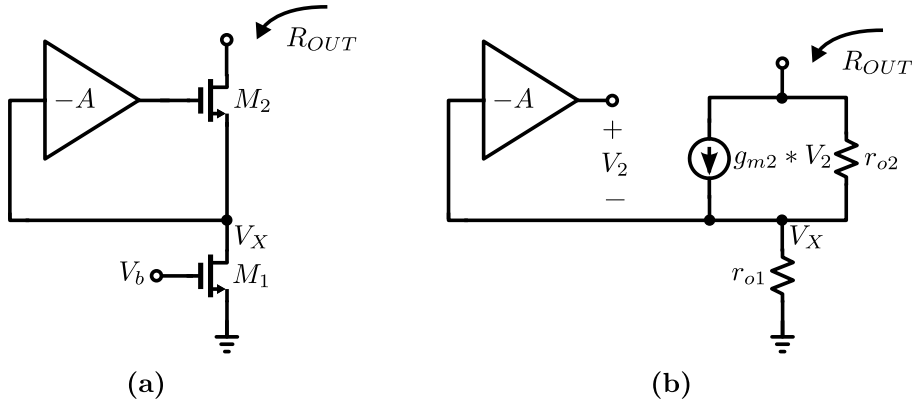


Figure 2.37: (a) Circuit implementation of gain boosting stage (b) Small signal model of the circuit

Figure 2.37(a) shows a commonly used technique used to boost output impedance of a cascode device. Figure 2.37(b) is the equivalent small signal model of the same circuit. Applying Kirchhoff's Current Law (KCL) to the V_X node, we get the following relation:

$$\frac{V_X}{r_{o1}} = g_{m2} * V_2 + \frac{V_{OUT} - V_X}{r_{o2}} \tag{2.45}$$

$$V_2 = -V_X(A + 1) \tag{2.46}$$

Substituting Eq 2.46 into Eq 2.45, we get:

$$\frac{V_{OUT}}{r_{o2}} = V_X \left(\frac{1}{r_{o1}} + \frac{1}{r_{o2}} + g_{m2}(1 + A) \right) \tag{2.47}$$

$$V_{OUT} = V_X \left(\frac{r_{o2}}{r_{o1}} + 1 + g_{m2} r_{o2} (1 + A) \right) \tag{2.48}$$

The current flowing through r_{o1} is the same as the output current:

$$I_{OUT} = \frac{V_X}{r_{o1}} \tag{2.49}$$

Thus,

$$R_{OUT} = \frac{V_{OUT}}{I_{OUT}} = r_{o1} \left(\frac{r_{o2}}{r_{o1}} + 1 + g_{m2} r_{o2} (1 + A) \right) \tag{2.50}$$

Since r_{o1} is very close in magnitude to r_{o2} the increase in output resistance is close to

$$R_{OUT} = r_{o1} \left(2 + g_{m2} r_{o2} (1 + A) \right) \approx r_{o1} (g_{m2} r_{o2} (1 + A)) \tag{2.51}$$

As can be seen from Eq 2.51, the output resistance of the cascode amplifier is increased even further by the gain of the feedback amplifier biasing the cascode device.

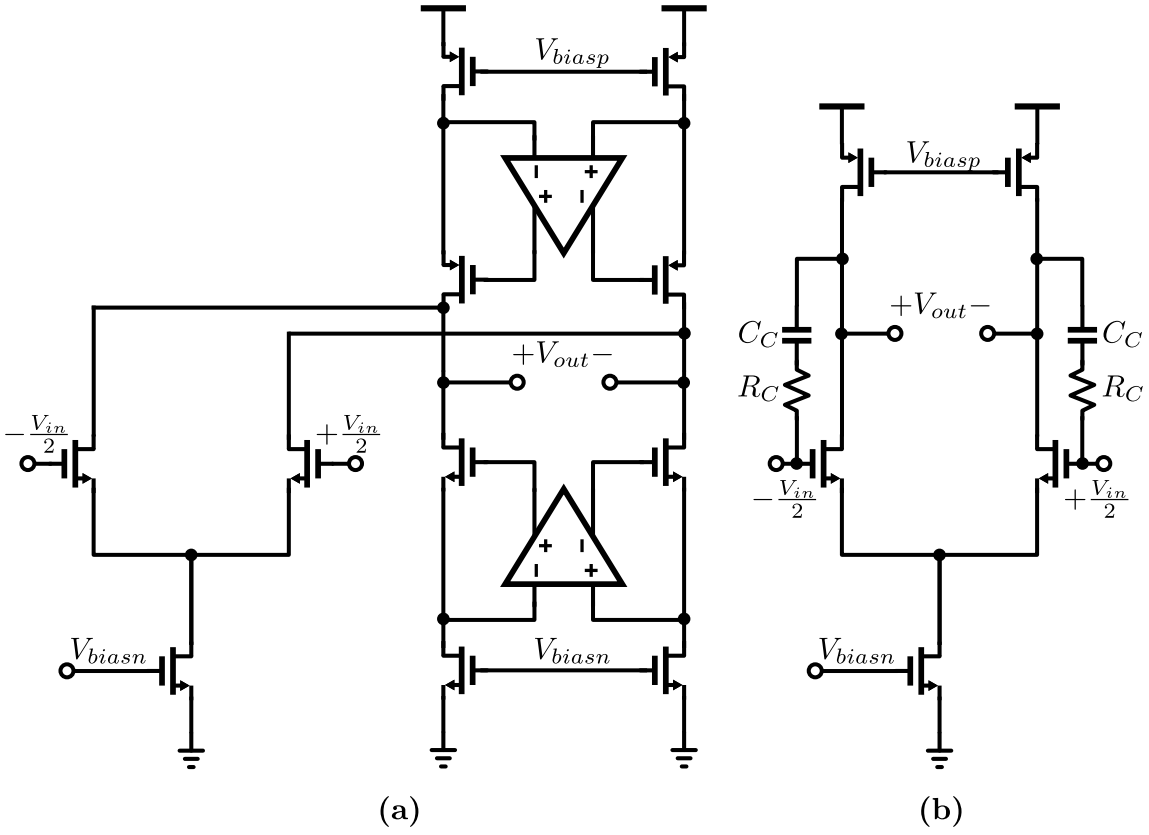


Figure 2.38: Circuit Implementation of OTA stages (a) First stage - Regulated cascode (b) Second stage - Fully differentially common source amplifier

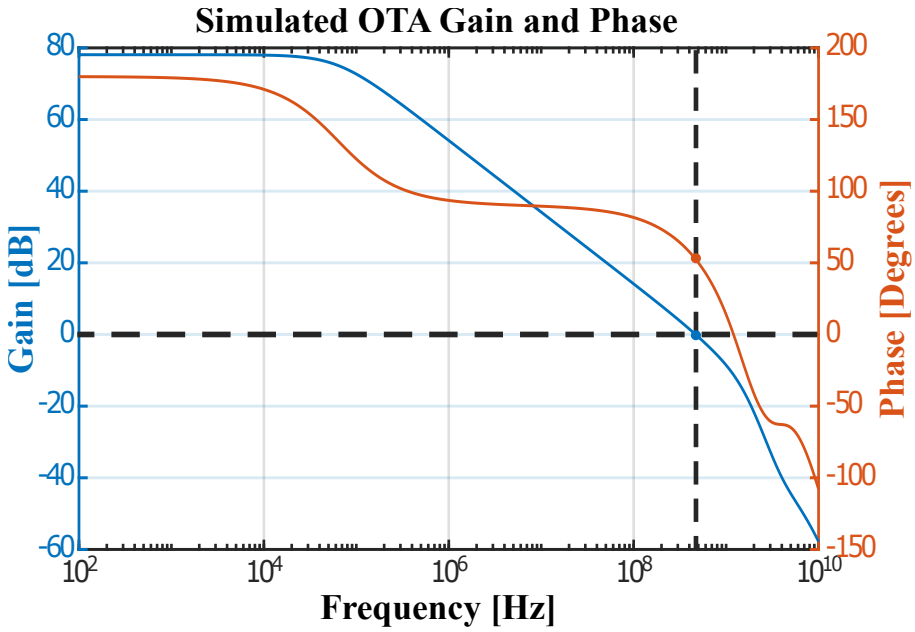


Figure 2.39: AC Simulation Results: Bode Plots showing Gain and Phase Response of the OTA

Comparator for Three-Level Quantizer

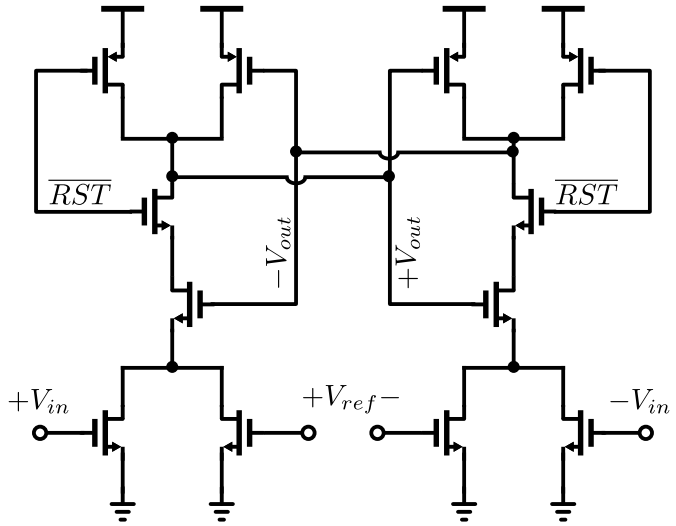


Figure 2.40: Circuit implementation of comparator used to make quantizer decisions

The comparator used to make quantizer decisions in the pipeline stage is shown in Figure 2.40. This design was first used in [33], and is a variation of the comparator used in [34] without a preamplifier. This comparator is ideal for use in low resolution pipeline stages (1.5 bit/stage). The bottom four input transistors operate in the triode region, and adding the inner transistors allows us to vary the comparator threshold.

2.3.5 Measurement Results

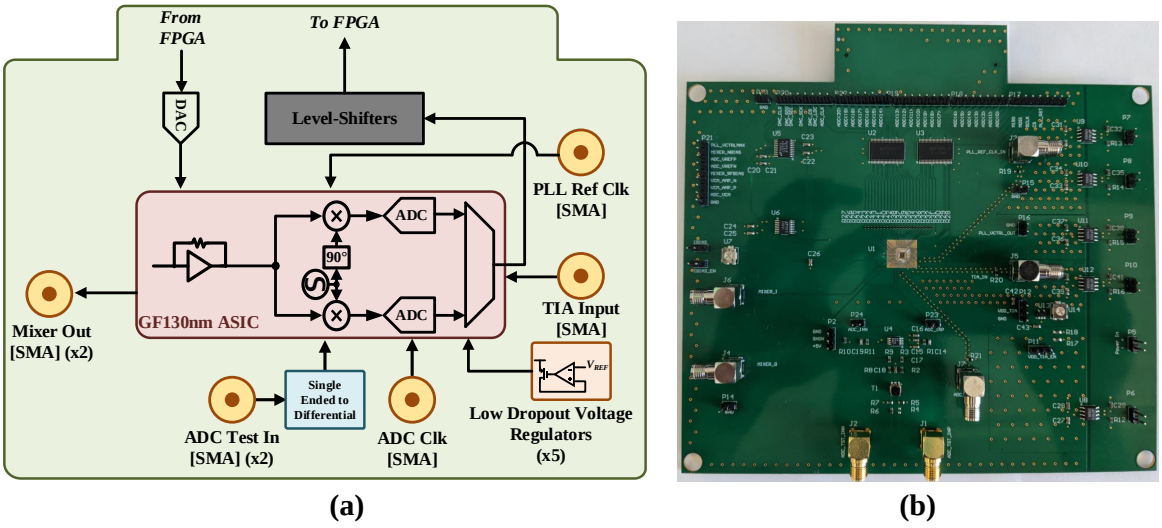


Figure 2.41: (a) Simplified schematic of Test PCB used for testing (b) Photograph of Test PCB

Figure 2.41(a) shows a simplified schematic of the layout of the Test PCB used to test the ASIC. The board was manufactured using the Chip-on-board (CoB) method where the ASIC was wire-bonded directly to the circuit board. All the analog biases for the ASIC were externally provided using a low noise 16-bit Digital to Analog Convertor (DAC) (LTC2656) programmed using a 4 pin Serial Peripheral Interface (SPI). The

outputs of the ADC are provided by low voltage (1.2V) devices inside the ASIC, while the Field Programmable Gate Array (FPGA) requires LVCMOS signalling (2.5V), we use level shifters (SN74AVC16T245) at the output of the ADC that sufficiently level-up the outputs of the ADC. The ASIC requires one 1.5V Digital supply, while it also requires a 2.5V digital and analog supply. To ensure low noise performance and to minimize the number of external power supplies required, we use several Low Drop-out (LDO) voltage regulators (LT1763) that supply these power supplies to the ASIC.

To test the ASIC's ability to be used as a synthesizer, it is used in the same configuration as shown in Figure 2.5. However, this integrated IC completely replaces the I-Q demodulator board (ADRF6820) and the high speed ADC (ADS4449). The output of our ASIC is a 21 bit dual date rate output and a clock signal, with results from each ADC written out on opposite edges of the clock. This data is processed by an FPGA which applies error correction and interprets the output as 12 bit words corresponding to the outputs sampled by the ADC.

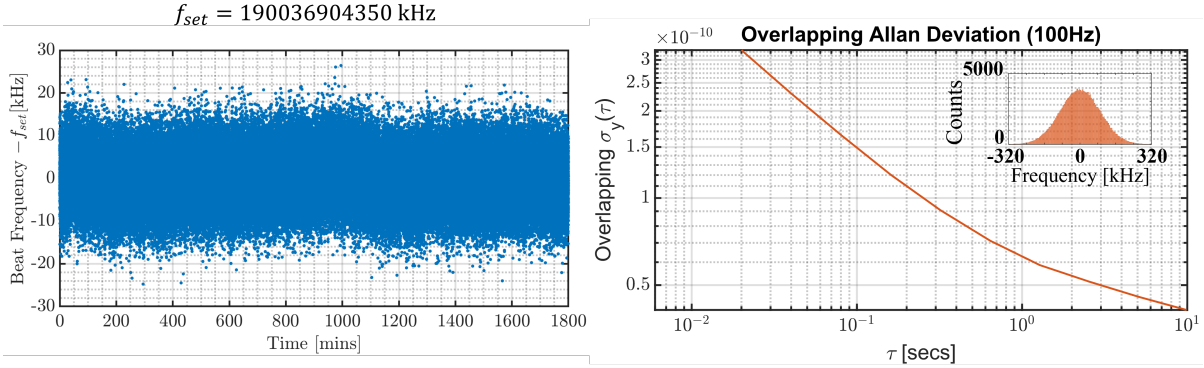


Figure 2.42: Current synthesizer output f_{out} deviation for 30 mins at 10-ms gate time shows

Figure 2.42(a) shows the frequency vs. time output of the laser when measured for 30 mins. This output is directly measured by the frequency counter (Keysight 53230A).

The maximum deviation seen in this case is +/-20kHz, orders of magnitude worse than the measurements made using board level components, where our deviation was in the order of mHz. However, a look at Figure 2.42(b), which shows the Allan deviation of our locked beat note, still shows a $1/\tau$ slope over longer averaging times. Please note that for the purpose of measuring the Allan Deviation, we multiply the deviations measured by the frequency counter by a factor of 16, since in this case, it is important to calculate the absolute stability of the laser.

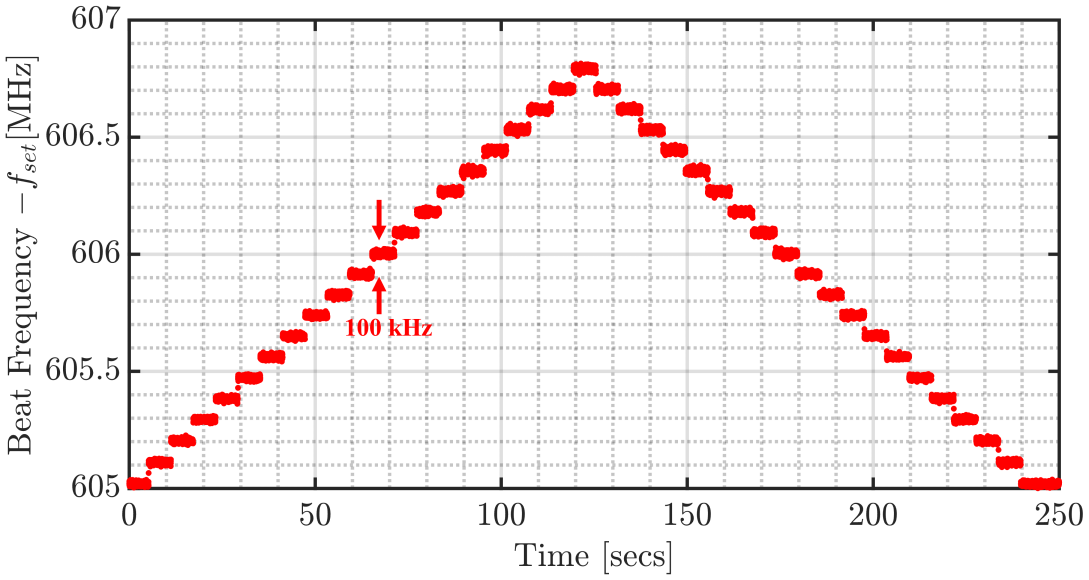


Figure 2.43: Bidirectional linear ramp of the synthesizer via step control of the laser offset PLL setpoint (10-ms gate time)

Similarly to case with the board level prototype, it is also important to demonstrate a synthesizer's ability to produce arbitrary frequencies. Figure 2.43 shows how the synthesizer can take steps of <100kHz and these steps are bidirectional.

2.4 Summary

	Freq. Detection using ADCs	
Level of Integration	Discrete	Integrated
Power [W]	2.1 (1.4+0.7)	0.314
Frequency Dev. [Hz]	+/-60m	+/- 320k
Allan Dev. [τ @ 1sec]	$< 10^{-16}$	$6.5*10^{-11}$
Step Size [Hz]	<1	100k
Range [MHz]	695-2700	600-2000

Figure 2.44: Table comparing the performance results between the Board-Level and Integrated Implementations

The table shown in Figure 2.44 succinctly summarizes all the trade-offs that we make when we fabricated our ASIC implementation. As can be seen from the first two rows of the table, we obtain tremendous benefits from integrating our Heterodyne-based controller as an ASIC. However, due to complexities involved in Analog Design and an ADC's susceptibility to noise, we are only able to get an Effective Number Of Bits (ENOB) of 9 bits. This reduction in bit resolution greatly affects the final frequency stability we are capable of achieving.

Equation (2.52) gives a rough first-order estimate of the frequency deviation we expect with an N-bit ADC and an overall loop bandwidth, f_{loop}, where T_{avg} is gate-time of the Frequency Counter.

$$\Delta f \propto \frac{\frac{45^\circ}{360^\circ * 2^N} * f_{IF}}{\sqrt{f_{loop} * T_{avg}}} \tag{2.52}$$

As can be seen from Equation (2.52), as we drop N, we increase the frequency deviation we observe. At the same time, reducing the sampling frequency of our ADC, indirectly reduces our loop bandwidth, f_{loop}, further increasing our frequency deviation.

A major shortcoming of this architecture is the fact that we have two use two power hungry and area intensive ADC's for our I-Q demodulation scheme. This results in us occupying twice the area and twice the power. Figure 2.19 clearly shows that a majority of the area inside the chip is occupied by our two ADCs.

The second short coming of this architecture is that we are required to use Frequency Dividers in our loop. Laser beat notes typically move by several 10s of MHz, and so if they were directly fed to input to ADC, we would violate the Nyquist Criterion that avoids aliasing of our input. By adding frequency dividers, we guarantee that the sampled version of our signal is truly the signal that is applied to our Front End. While these frequency dividers improve our stability, they also divide down a wide bandwidth of white noise within our loop bandwidth.

In the next chapter, we will address some of these shortcomings and work to implement an All-Digital design that requires very little calibration, and is very robust to noise sources within the circuit.

Chapter 3

Frequency Detection using Time to Digital convertor (TDC)

While the ASIC in the previous section offers several benefits, and offers extremely small resolutions, its cost is the area taken and power dissipated by the Analog to Digital Converter (ADC). As described in the previous section, the noise requirements of the system necessitated the use of large on chip capacitors for sampling at every stage of the ADC. This resulted in the use of larger currents to drive these capacitors, which increased the power dissipated by the system.

This motivated us to find a solution to measure the instantaneous phase of the Intermediate Frequency (IF) signal without the use of a power-hungry ADC. The inspiration for our solution was found in All-Digital Phase Locked Loops (ADPLLs) which solve a very similar problem using a Time to Digital Converter (TDC). Instead of digitizing the IF signal and computing the instantaneous phase off-chip using the CORDIC block, it is possible to directly measure the phase difference between the IF signal and the reference signal and send it out of the ASIC. This is similar to the case of Phase Frequency Detector (PFD) in a Phase Locked Loop (PLL), where the time difference between the two

clock edges is converted directly to voltage using a current that stays 'ON' for that time duration and integrated onto a capacitor. In the case of a TDC, that time is directly reported as a digital value.

The TDC offers several benefits over the ADC approach to measuring phase. It consumes a fraction of the area that is occupied by the ADC, owing to the lack of any large capacitors and current sources in its design. It is also much more robust to process and temperature variations compared to an ADC. The only process-sensitive part in the entire design is the delay ring, which consists of simple current starved inverters tied in a ring fashion. The effect of process variation on these circuits is very easy to characterize and calibrate. As will be described in a later section, the calibration mechanism for the TDC is also a lot simpler compared to the ADC. Adjusting the current biases for the current starved inverters can get the delay that is desired. Finally, a TDC is an all digital design, which means that it can be completely synthesized and can be much more readily adopted to other technologies, and for future designs.

Even though this IC is discussed after the ADC chip in this manuscript, chronologically it was actually designed and fabricated first. A major part of this IC's architecture and schematic design was done by colleague Aaron J Bluestone and this IC is discussed comprehensively in his dissertation [35]. The TIA design and layout were completed by our colleague Robert Constanzo from the University of Virginia [36]. The PLL used in this chip was designed by our advisor Prof. Theogarajan, and Aaron and I worked together to complete the layout of this chip. We both worked together on the initial testing of this chip, and work was continued after Aaron's graduation to lock a laser using this chip. In the next section, we will discuss the various design blocks employed in this chip for completeness (for a detailed explanation of all the functional blocks readers are encouraged to refer to [35]), and then discuss the locking results that were obtained using this chip.

3.1 ASIC Integration

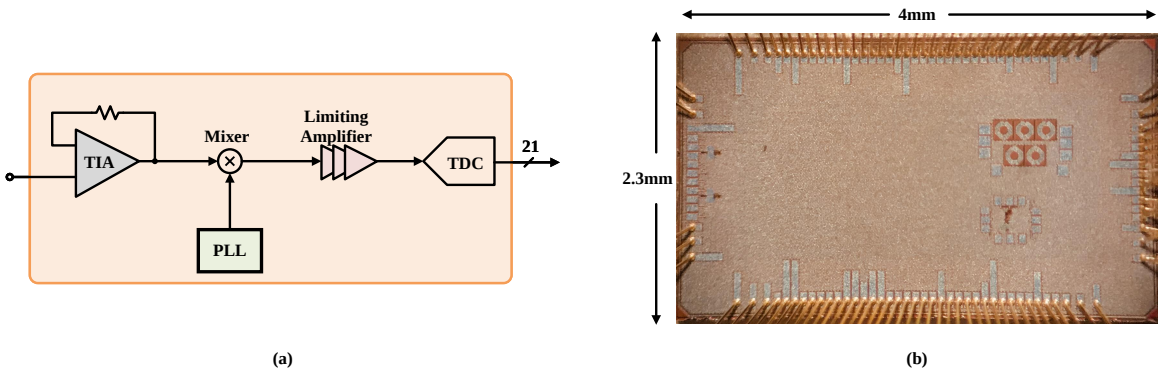


Figure 3.1: (a)Simplified schematic of signal chain (b) Photograph taken of the Integrated Circuit (IC) after fabrication

Figure 3.1(a) shows the signal chain present on the signal chain on the Integrated Circuit (IC). Similar to the IC fabricated using Global Foundries' BiCMOS8HP process, it has a Radio-Frequency (RF) front end consisting of a Trans-Impedance Amplifier (TIA), Mixer and a Phase Locked Loop (PLL) that generates the Local Oscillator (LO) signal for the mixer.

The TIA converts the beat note photo current emitted by the photo diode to a voltage. This RF signal can range anywhere from DC-8GHz, and so it needs to be down-converted to an Intermediate Frequency (IF) signal for processing by the Time to Digital Convertor (TDC). This is done by a purely CMOS double balanced mixer which receives its LO signal directly from the on-chip PLL.

The next few sections will describe each of these parts in further detail, and finally show measurements results from them, and the final locking results.

3.1.1 Trans-impedance Amplifier

Just as in the case of the IC in the previous chapter, the TIA used in our design was designed by our colleague Robert Costanzo at the University of Virginia, and is described in detail in [36]. The TIA design, shown in Figure 3.2, consists of a current reuse path that is applied to a conventional Regulated Cascode (RGC) TIA in order to offer improvements in transimpedance performance.

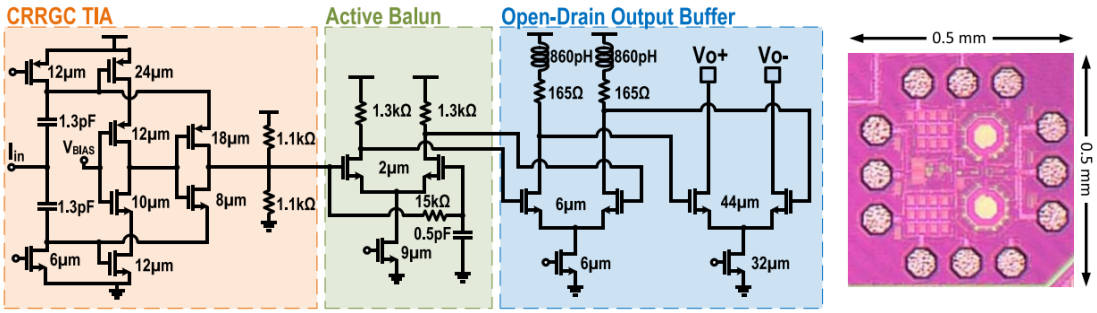


Figure 3.2: Left: full schematic of the implementation of the CRRGC TIA, including active balun and open-drain output buffer. The biasing circuits and references for the current tails are omitted. All devices are minimum length. Right: microphotograph of the CRRGC TIA on a 65-nm GP CMOS chip.

3.1.2 Phase Locked Loop

The phase locked loop (PLL) in this IC is almost identical to the one implemented in the previous chapter. The only few differences between the two structures are the Voltage controlled oscillators (VCO), the level shifters, and the divider. The only difference between the two VCO's is the fact that instead of a 2 stage VCO implemented in the previous case, we implement a 3 stage VCO. The next few subsections will describe the level shifters, and the divider in more detail. For all other blocks used in the PLL, we encourage the readers to refer to the "Phase Locked Loop" section in the previous chapter.

Level Shifter

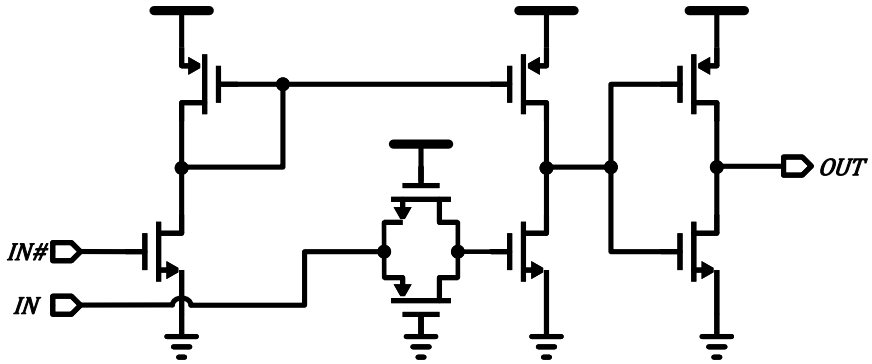


Figure 3.3: Circuit implementation of the Level Shifter

The level shifter for the PLL needs to operate from DC-8GHz, and the two stage level shifter topology used in Figure 2.28 does not work at higher frequencies. For the level shifter, the topology used is shown in Figure 3.3. Since the VCO output frequency is modulated using its supply voltage, the output swing of the VCO output at lower frequencies can potentially be much lower than an inverter threshold. In those cases, the current mirror that uses the $IN\#$ input generates a very small current that slowly charges the capacitor at the input of the inverter which in turn generates the OUT signal.

Divider

The PLL designed for this IC has an operating range from DC-8GHz. This means that the divider has to divide signals with frequencies up to 8GHz down to the reference frequency (typically 10-100 MHz). A simple solution to this is to use a prescale divide-by-N (usually, N=2 or 4), where the input is first divided down to manageable frequencies and ease the constraints on the programmable divider. The drawback of this approach is that the smallest frequency step that the PLL can now take is $N.f_{ref}$, where N is the

prescale divide value. For PLLs where finer frequency steps are needed, or frequency steps at least equal to f_{ref}, another approach is to use a "Pulse-swallow counter" or a "Dual-Modulus divider". The topology for this type of divider can be seen in Figure 3.4(a). Instead of using a fixed divide-by-N prescale divider at the input, the core idea behind this divider is to switch between using an N or N+1 divider (labeled as $M/M+1$ in the figure, where M stands for Modulus in that case). As illustrated in Figure 3.4(a), in these types of implementations, there are two counters that count up to A and B respectively, where $A < B$. So, for A cycles of the divided clock, a prescale divide value of $M+1$ is used, and for the remaining cycles $(B-A)$ the prescale value is M. The final number of divide cycles for the input clock can be given by Equation (3.2).

$$\text{Divide cycles} = A * (M+1) + (B-A) * M \tag{3.1}$$

$$= B * M + A \tag{3.2}$$

Thus, by appropriately choosing the values of A and B, integer values for the final divide can now be achieved.

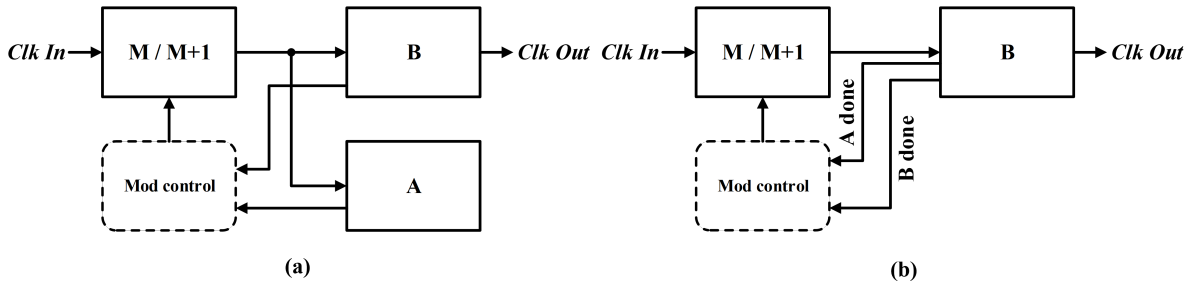


Figure 3.4: (a) Traditional implementation of a dual modulus divider (b) Current Implementation of a dual modulus divider

Our contribution to this divider is to use one counter instead of two and simply count up to B. When this counter value hits the programmable value of A, we use it to generate a control signal that switches the modulus (Figure 3.4(b)).

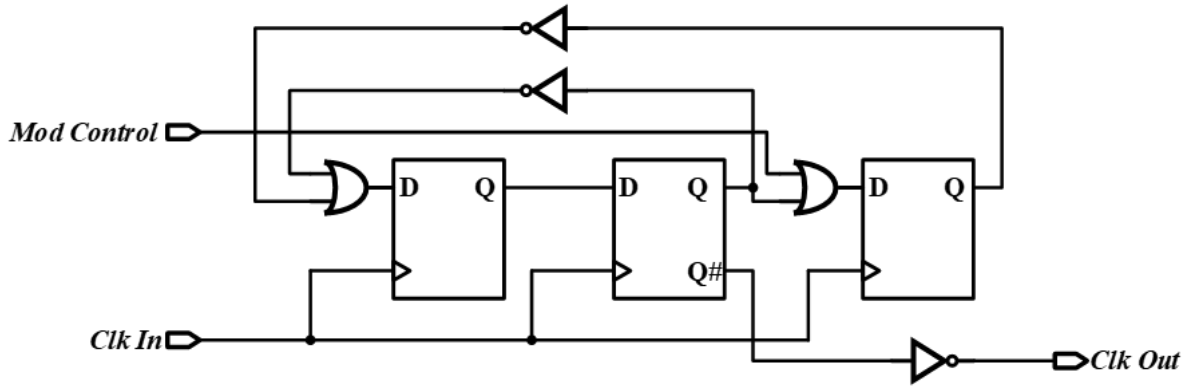


Figure 3.5: Circuit Implementation of 4/5 Dual Modulus Divider

For our IC, our modulus of choice was to use $M = 3$. Figure 3.5, shows the circuit implementation of our 4/5 Dual Modulus Divider. This divider then feeds its scaled output clock to a Programmable 6-bit counter to give us the divide value we desire.

3.1.3 CMOS Gilbert Cell Mixer

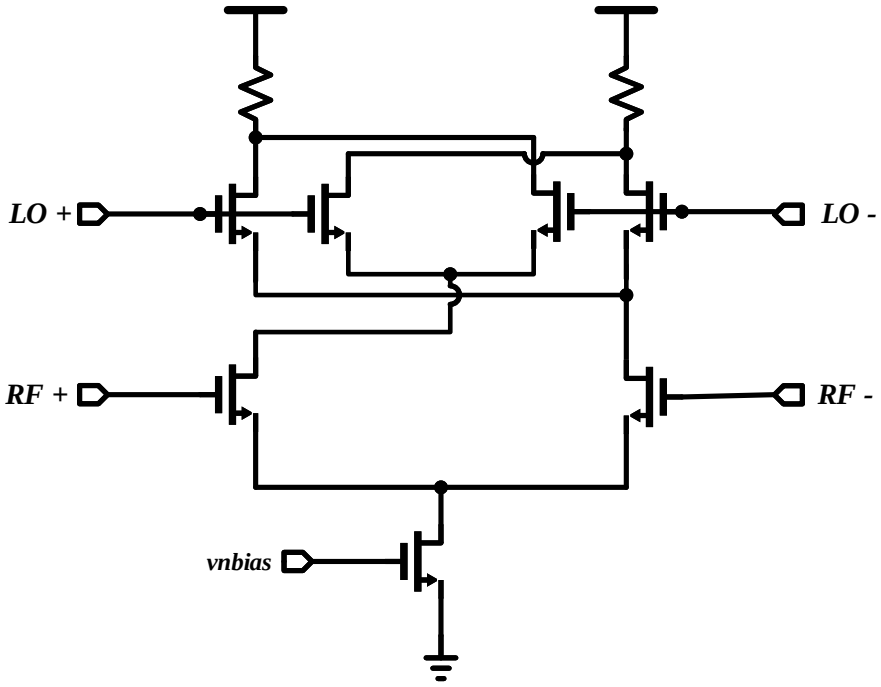


Figure 3.6: Circuit implementation of the double balanced mixer

Figure 3.6 shows a CMOS implementation of a double balanced mixer. An astute reader will observe that this mixer has the same circuit topology as the mixer used in the previous chapter, with CMOS transistors replacing the BJTs. For further information on how this topology mixes the RF signal of interest to baseband, the reader is encouraged to refer to the "Mixer - Gilbert Cell" section of the previous chapter.

3.1.4 Baseband Amplifier and Limiting Amplifier

After the mixer,two operations on the IF signal need to be accomplished. Post-mixing, the IF signal could be in the order of several hundred μVs and contain several harmonics of the IF signal. This small signal needs to be amplified, and made into a full

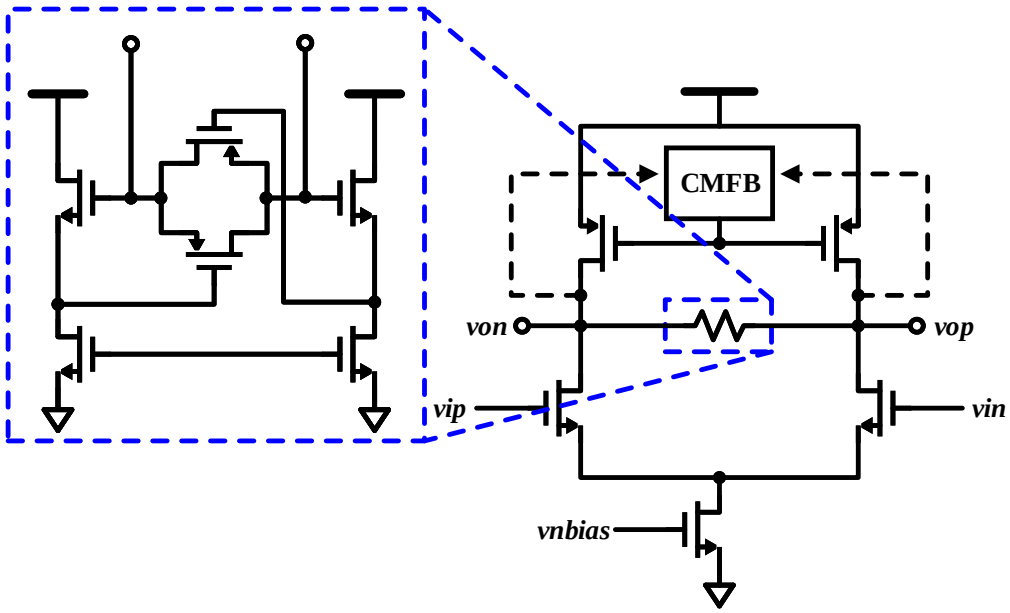


Figure 3.7: Circuit implementation of the double balanced mixer

swing signal. At the same time, the higher order harmonics need to be filtered so that they do not interfere with the signal processing.

This is accomplished in two stages. For the first stage, a tunable bandwidth (BW) differential amplifier is used. This differential amplifier not only provides the gain that is needed after the mixer, but also helps filter the undesirable high frequency components coming out of the mixer.

The gain of this first stage tunable amplifier, A is given by:

$$A = g_m \left(r_o p || r_o n || \frac{R_{lin}}{2} \right)$$

If it is ensured that the $\frac{R_{lin}}{2} << r_o p, r_o n$, then this gain term becomes

$$A = g_m \left(\frac{R_{lin}}{2} \right)$$

Similarly, the Bandwidth (BW) of this tunable amplifier is given by:

$$f_{3dB} = \frac{1}{\pi R_{lin} C_{load}}$$

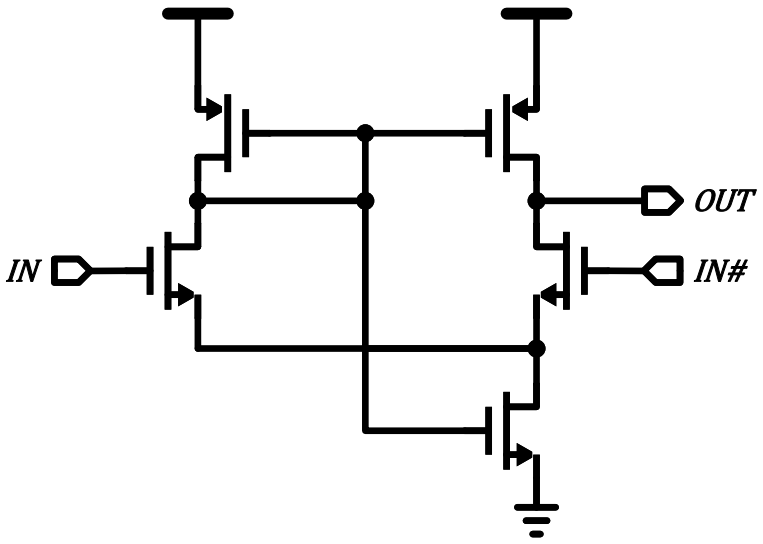


Figure 3.8: Circuit implementation of the Chappell Amplifier

After sufficient pre-amplification and filtering, the baseband signal in this case is converted to a rail-to-rail digital signal that is sent to the Time to Digital Convertor (TDC). For this, a simple Chappell Amplifier shown in Figure 3.8 is used that was first seen in [30]. The reason to choose this amplifier is because it is self biased, and has the versatility to operate at different temperatures and supply voltages.

3.1.5 Time to Digital Converter

A Time to Digital Converter (TDC) is an electrical circuit that measures the time difference between two clock edges that arrive at different times. Figure 3.9(a) illustrates an example where two clock edges arrive at different times. The circuit shown in Figure

3.9(b) would be a simple but inefficient implementation of a TDC, which would help find the difference between the arrival times of the two clock signals. Let the first arriving clock edge be called the "faster edge" and the later arriving one the "slower edge". In the TDC implemented in Figure 3.9(b), the faster edge is passed through a delay line with uniformly spaced buffers and it is ensured that each buffer has a constant delay of τ. The output of each of the buffer along the delay line is tapped and fed to the D input of D-Flipflop (DFF). Using the slower edge as the clock input for all of these flip flops would then give the requested TDC measurement every clock cycle. The system yields the TDC value in the form of a thermometer code which can be converted to a binary value that would give the solution. To understand how the TDC works, one can imagine the measurement operation as a race between the two rising edges. As the faster edge propagates through the delay line, it flips each of the buffer outputs to a "1". However, if the slower edge arrives before the faster edge can propagate through the entire delay line, it will sample a series of 1s followed by 0s. These 1s correspond to the distance the faster edge travelled along the delay line before it was sampled by the slower edge. Here, the resolution of our TDC is the delay of buffer cell in the delay line and our dynamic range is the number of buffers and DFFs in our system. A consequence of this is that if an N-bit TDC was wanted, the number of these cells required would be 2^N.

Instead of using a linear delay line Yu et al. [37] increased the dynamic range of their TDC by configuring their delay line as a ring. As opposed to simply counting the number of delay cells it took for the slower edge to catch up with the faster edge, the number of laps it takes for the slower edge to catch the faster edge can be counted. The resolution of the TDC is further increased by transmitting the slower edge through another ring and configuring the delay of its buffers to be slightly faster than the buffers in the delay line transmitting the faster edge. A consequence of this adjustment is that the resolution of the TDC is now the difference of the propagation delay of the two buffer cells, which

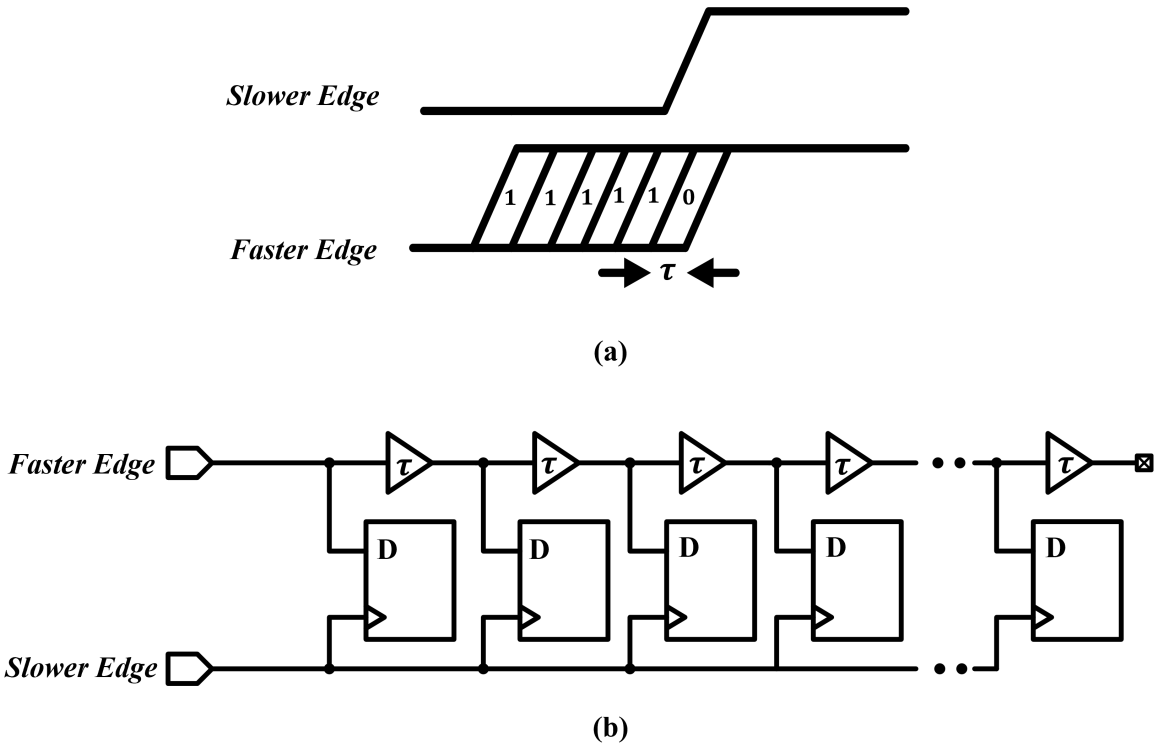


Figure 3.9: (a) Illustration of two rising edges arriving at different times (b) Block diagram of a simple Time to Digital Converter (TDC) that could measure the time difference between the two edges with a timing resolution equal to the delay of each inverter

can be tuned to be much smaller than the propagation delay of the buffer cells. TDCs implemented in this fashion are often referred to as Vernier Ring TDCs (VRTDCs). The TDC implemented in this IC is a VRTDC and was greatly inspired by the work done in [37], and an extensive analysis of this specific VRTDC and its implementation can be found in [35].

Conventional implementations of TDCs expect the two input frequencies at its input to have the same frequency. However, as discussed in Chapter 2, a laser's beat note can often jump by multiple MHz. Our solution to this problem is illustrated in Figure 3.10. We make two changes to our system's architecture: 1. We no longer expect the reference clock and the feedback clock to be the same frequency, we ensure that the feedback clock

is always faster than the reference clock; 2. Instead of arbitrating which signal arrived first, we always have the reference clock edge enter the fast ring. We place a counter at the input of the feedback input and count the number of edges before a feedback edge arrives. Control circuitry only lets the feedback clock enter the slow ring after the reference clock has entered the fast one. These two alterations now free us from the restriction that the feedback frequency needs to be close to the reference frequency and increase our acquisition range. In the next section, we will discuss how we use the output of this counter in conjunction with our TDC output to get a measure of our beat note frequency.

Frequency Error Calculation

As mentioned in the previous section, to facilitate a wider range of operation, a counter is used instead of a frequency divider before the input of the TDC (illustrated in Figure 3.10). Figure 3.11 shows a timing diagram of the TDC under normal operating conditions. The TDC yields a result, τ_N every reference cycle, T_{REF}. In addition, it also measures the beat count, B, which is the number of "Intermediate Frequency" cycles between each reference cycle. This counter is reset every rising edge of the reference clock.

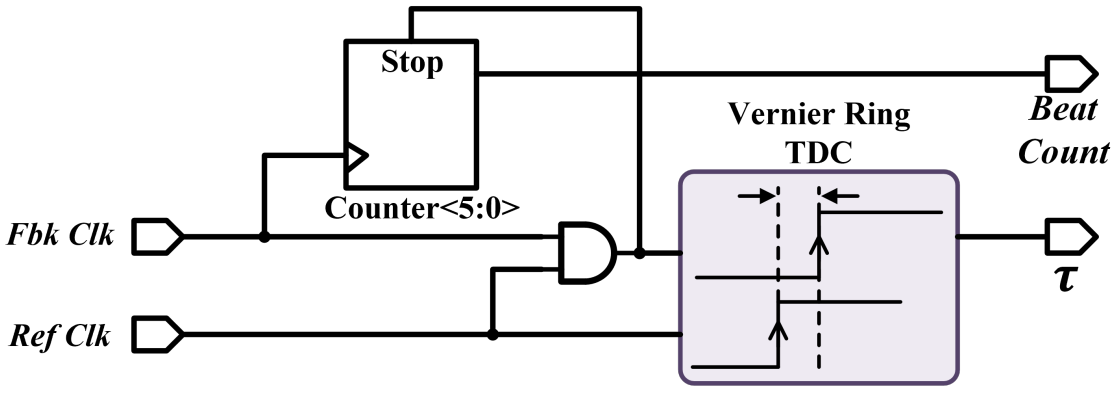


Figure 3.10: Simplified circuit diagram of extended range TDC

Based on all the information we receive from the TDC, the time period, T_{IF} of the incoming IF signal is easily calculable.

$$T_{IF} * B = (T_{REF} + \tau_{N+1} - \tau_N) \tag{3.3}$$

$$T_{IF} = \frac{T_{REF} + \tau_{N+1} - \tau_{N+1}}{B} \tag{3.4}$$

$$T_{IF} * B - T_{REF} = \tau_{N+1} - \tau_N \tag{3.5}$$

However, careful examination of Equation (3.4) reveals that correct calculation of the IF time period requires the use of an expensive divider operation - these operations on the FPGA not only require a lot of resources but also have a high latency.

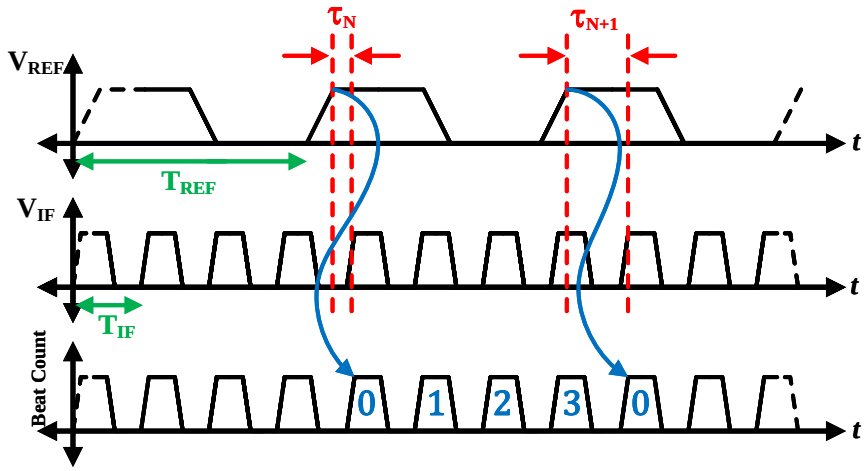


Figure 3.11: Timing diagram illustrating TDC operation

For the loop, it is required to produce an error signal that is proportional to the difference in the time period of the IF signal and our reference signal. An ideal version

of this signal would be given by:

$$(T_{error})_{ideal} = T_{IF} - T_{REF} \tag{3.6}$$

However, as mentioned before, calculating the T_{IF} is expensive. So, instead of calculating the error exactly, an error term that is proportional to the ideal error by a constant is used.

This error can be acquired by subtracting $(B-1) * T_{REF}$ from Equation 3.5.

$$(T_{error})_{actual} = \tau_{N+1} - \tau_N - (B-1) * T_{REF} \tag{3.7}$$

$$(T_{error})_{actual} = T_{IF} * B - T_{REF} - (B-1) * T_{REF} \tag{3.8}$$

$$(T_{error})_{actual} = B * (T_{IF} - T_{REF}) \tag{3.9}$$

$$(T_{error})_{actual} = B * (T_{error})_{ideal} \tag{3.10}$$

$$(T_{error})_{actual} \propto (T_{error})_{ideal} \tag{3.11}$$

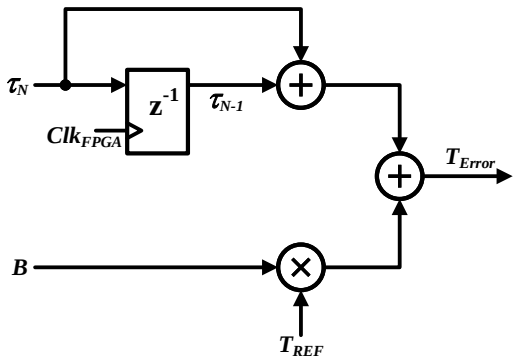


Figure 3.12: Block diagram of the Digital Signal Processing required to calculate the time error

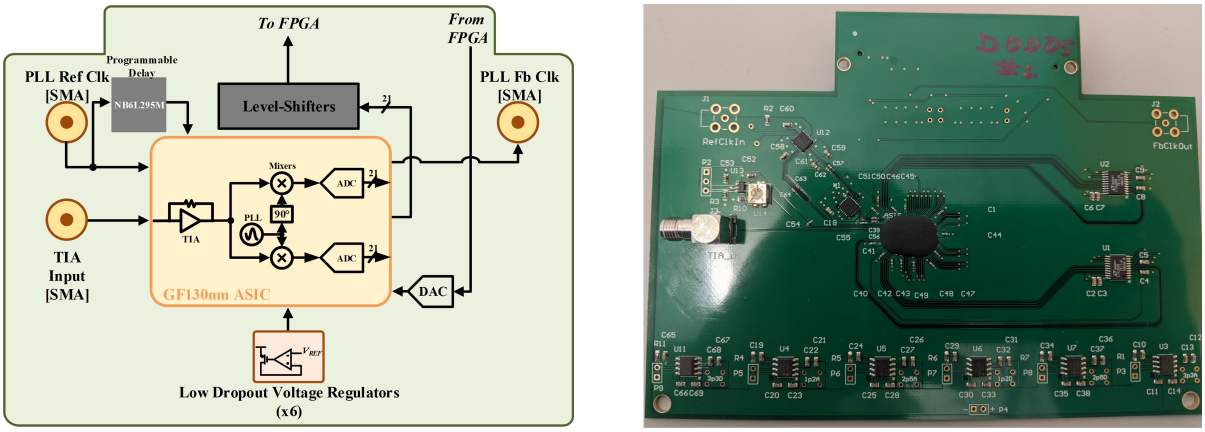


Figure 3.13: (a) Simplified schematic of Test PCB used for testing (b) Photograph of Test PCB

3.1.6 Test PCB

Figure 3.13(a) shows a simplified schematic of the layout of the Test PCB used to test the ASIC. The board was manufactured using the Chip-on-board (CoB) method where the ASIC was wire-bonded directly to the circuit board. All the analog biases for the ASIC were externally provided using a low noise 16-bit Digital to Analog Convertor (DAC) (LTC2656) programmed using a 4 pin Serial Peripheral Interface (SPI). The outputs of the ADC are provided by low voltage (1.2V) devices inside the ASIC, while the Field Programmable Gate Array (FPGA) requires LVCMOS signalling (2.5V), level shifters (SN74AVC16T245) are used at the output of the ADC that sufficiently level-up the outputs of the ADC. The ASIC requires one 1.2V Digital supply, while it also requires a 2.5V digital and analog supply. To ensure low noise performance and to minimize the number of external power supplies required, several Low Drop-out (LDO) voltage regulators (ADM7154) are used that supply these power supplies to the ASIC. Finally, to test the functionality of the TDC, and calibrate the fast and slow delay lines a programmable delay IC (NB6L295M) is used.

3.2 Measurement Results

Full loop frequency synthesis by bypassing RF front end

To test the ASIC's ability to be used as a synthesizer, it is used in the same configuration as shown in Figure 2.5. However, as shown in Figure 3.14, instead of using an I-Q Demodulator and an ADC, a single mixer is used to mix down the beat note to a manageable Intermediate Frequency (IF) and a Time to Digital Converter (TDC) is used to calculate the frequency of this IF signal. For an initial experiment just to prove that a TDC can be used in a locking experiment, the TDC inside the ASIC is employed and the RF Front End (TIA, Mixer and PLL) is completely bypassed.

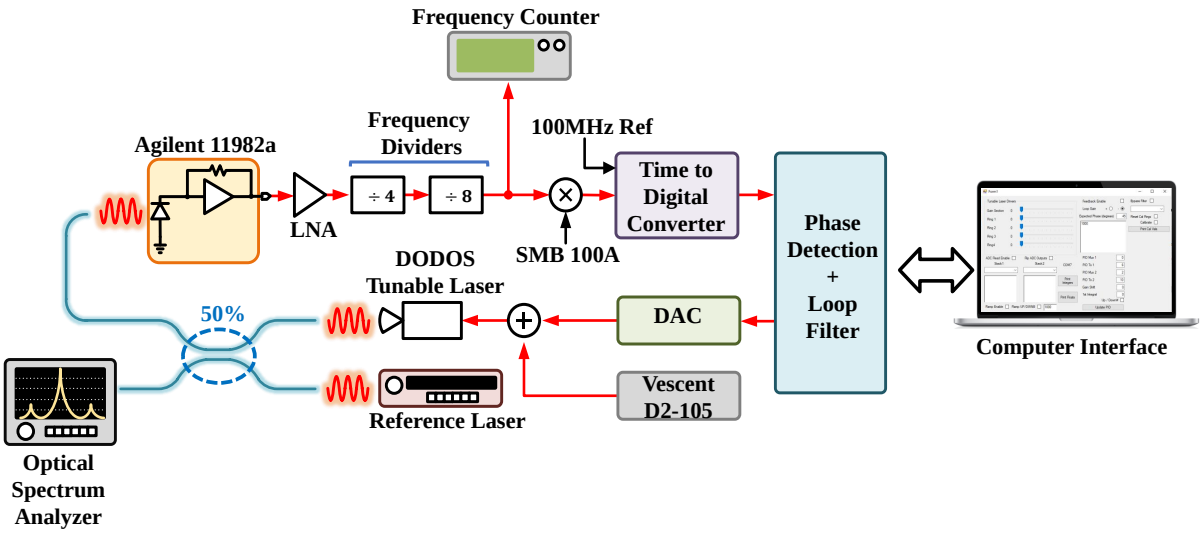


Figure 3.14: Block diagram of Full loop used for frequency synthesis

An actual photograph of this setup can be seen in Figure 3.15. This figure illustrates all the extra RF components that are required to process the RF beat note before being mixed down to an IF signal.

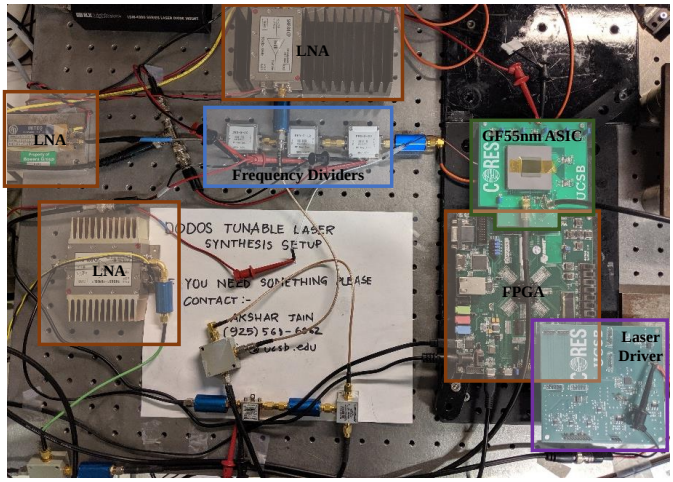


Figure 3.15: Photograph of Full loop used for frequency synthesis

Figure 3.16(a) shows the frequency vs. time output of the laser when measured for 300 secs. This output is directly measured by the frequency counter (Keysight 53230A). The maximum deviation seen in this case is +/-64Hz. Note that the frequency deviation already includes a factor of 32. The power spectrum associated with this beat note can be seen in Figure 3.17.

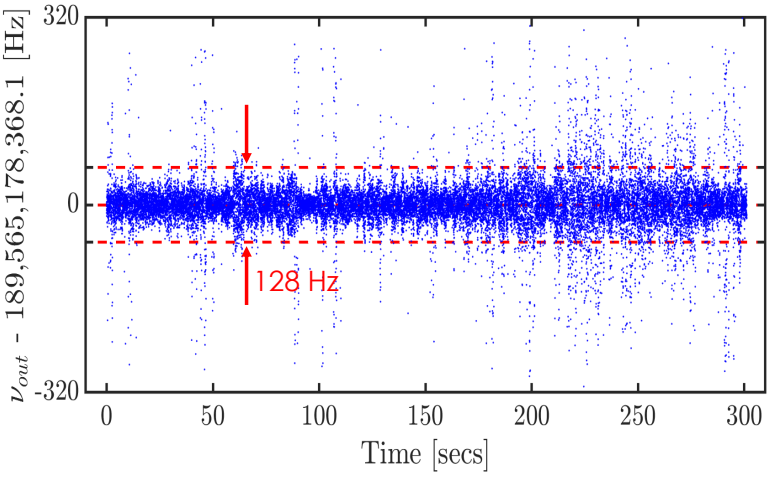


Figure 3.16: Current synthesizer output ν_{out} deviation for 300s at 10-ms gate time shows

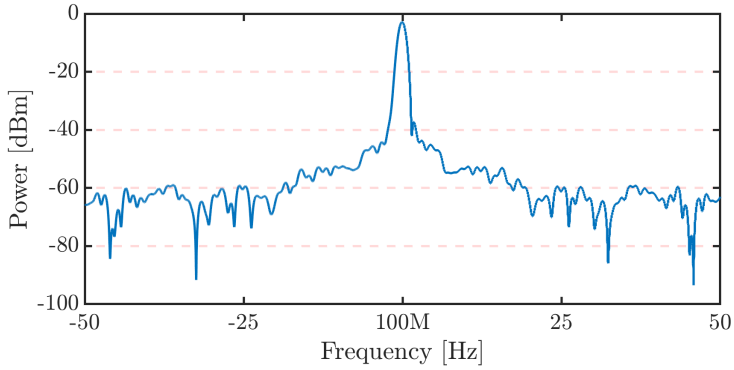


Figure 3.17: Power spectral density of the locked tunable laser plotted on an ESA (Resolution bandwidth of 1 Hz and span of 100 MHz)

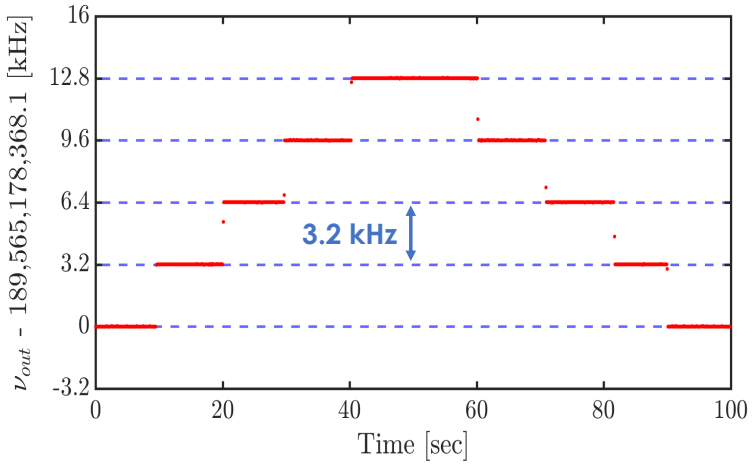


Figure 3.18: Bidirectional linear ramp of the synthesizer via step control of the laser offset PLL setpoint (100-ms gate)

To demonstrate the synthesizer capabilities of the loop, the set-point is moved by changing the Local Oscillator (LO) signal in steps of 3.2kHz (this is the signal used to mix down the beat note to IF). Figure 3.18 shows frequency counter measurements that measure this beat note vs. time.

Full loop frequency synthesis

Now that it has been demonstrated that it is possible to use a TDC instead of the ADC-based architecture to lock a laser to an optical reference, we now present the results of our synthesis efforts when we use the RF signal chain inside the ASIC, without the help of external RF components such as frequency dividers, Low Noise Amplifiers (LNAs), and mixers. Figure 3.19 is a simplified block diagram of our entire signal chain. The output from the Agilent Lightwave converter is now directly fed to the IC which mixes the beat note down to baseband frequencies using internal RF electronics.

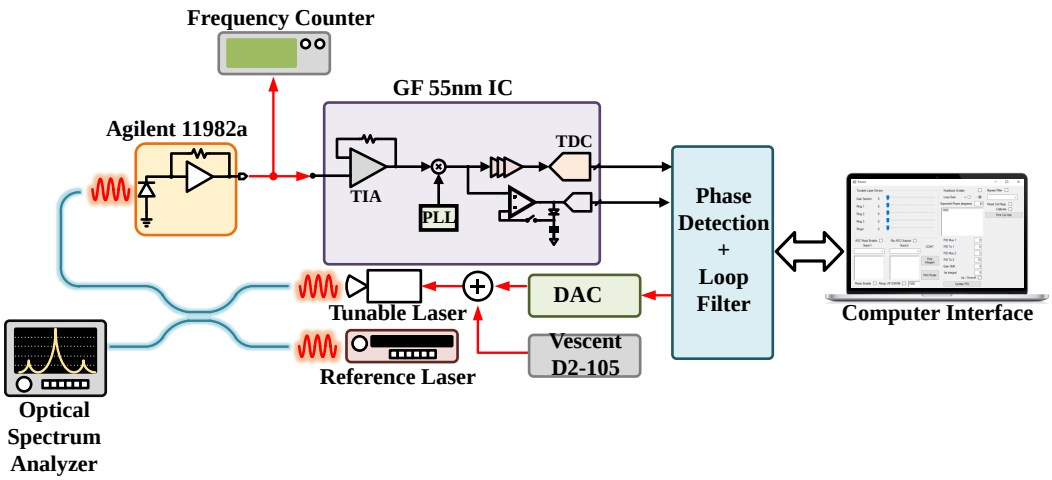


Figure 3.19: Block diagram of Full loop used for frequency synthesis

An actual photograph of this setup can be seen in Figure 3.20. This figure illustrates how all the extra RF components that are required to process the RF beat note before being mixed down to an IF signal are no longer required for frequency synthesis.

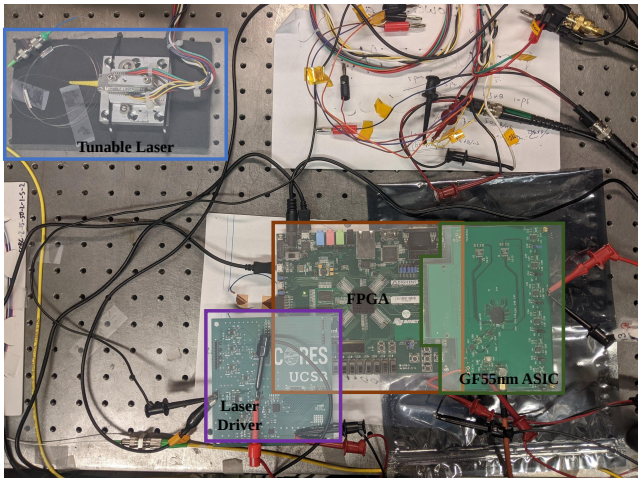


Figure 3.20: Photograph of Full loop used for frequency synthesis. Note the absence of LNAs, Frequency Dividers, and an external RF Mixer

Figure 3.21(a) shows the frequency vs. time output of the laser when measured for 3600 secs. This output is directly measured by the frequency counter (Keysight 53230A). The maximum deviation seen in this case is +/-4kHz. We use this frequency data to calculate the Allan Deviation of our synthesizers output (shown in Figure 3.21(b)).

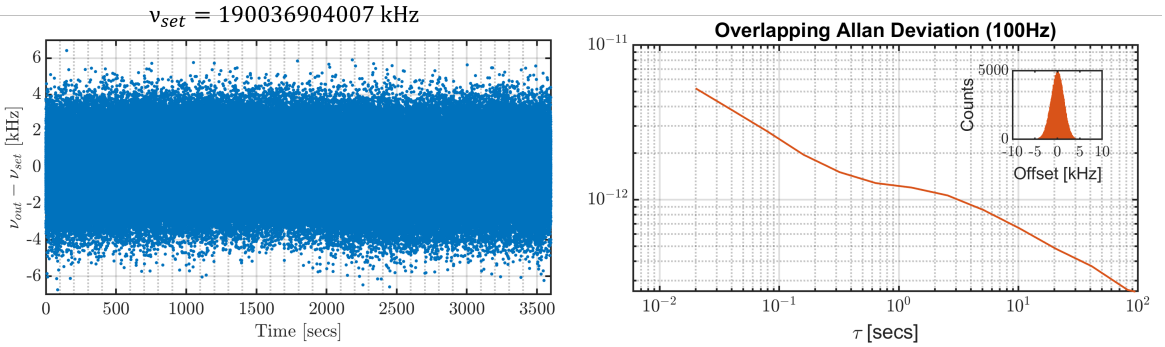


Figure 3.21: Current synthesizer output ν_{out} deviation for 3600s at 10-ms gate time shows

As with previous synthesizer experiments, we now demonstrate the synthesis ability of the ASIC. Figure 3.22(a) shows bidirectional control of the output frequency. There are two mechanisms that use to alter the output frequency. The first way is to step the LO frequency used to down-mix the beat note, these step sizes are equal to the frequency

of our PLL's reference clock (for our ASIC's PLL, this reference was set to 25MHz). The second technique is to alter the expected beat note in our frequency calculation algorithm. This moves the output with step sizes equal to the TDC's reference frequency. Figure 3.22(a) shows both these steps and demonstrates our ability to move the beat note in both directions. Figure 3.22(b) shows the synthesizers ability to generate beat notes over a wide frequency range. Here, we move the beat note from 3GHz to 4GHz.

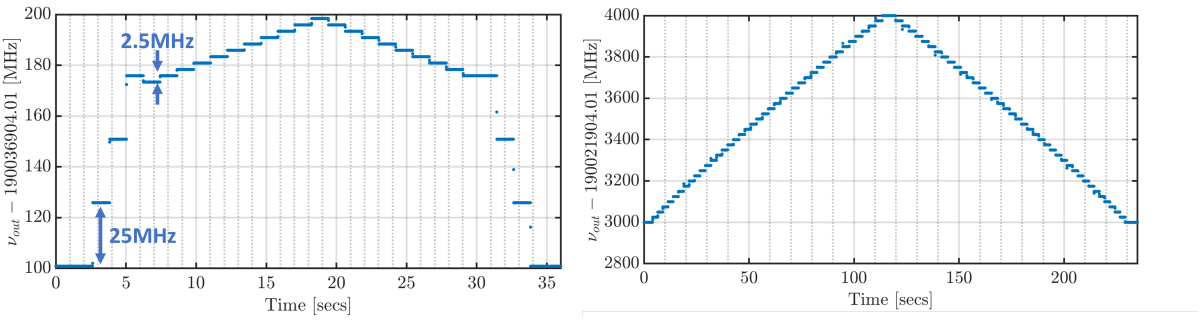


Figure 3.22: Bidirectional linear ramp of the synthesizer via step control of the laser offset PLL setpoint (100-ms gate)

Finally, to prove the synthesizers ability to generate arbitrary frequencies we use the loop to move the laser in way that spells out "UCSB" when we plot the laser's instantaneous frequency vs. time.

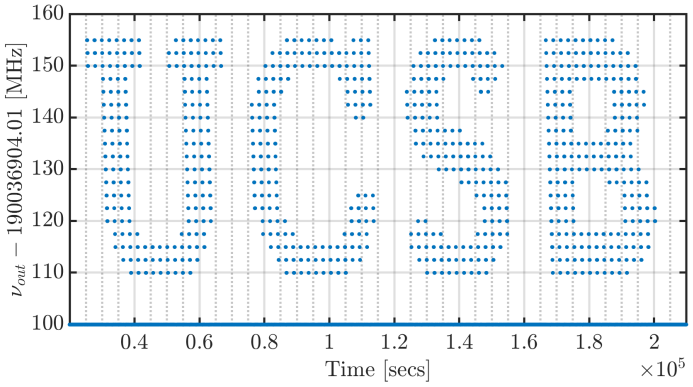


Figure 3.23: Synthesizer setpoint varied over time to display "UCSB" Logo

3.3 Summary

	Freq. Detection using ADCs		GF55nm ASIC
Level of Integration	Discrete	Integrated	Integrated
Power [W]	2.1 (1.4+0.7)	0.314	0.212
Frequency Dev. [Hz]	+/-60m	+/- 320k	+/-4k
Allan Dev. [τ @ 1sec]	$< 10^{-16}$	$6.5*10^{-11}$	$< 10^{-12}$
Step Size [Hz]	<1	100k	2.5M
Range [MHz]	695-2700	600-2000	500-7000

Figure 3.24: Table comparing the performance results between the Board-Level and Integrated Implementations of the ADC-based approach vs. the TDC-based approach

As demonstrated by the table shown in Figure 3.24, the ASIC implemented in this chapter outperforms the GF130nm ASIC in nearly every respect. One of the primary

benefits of this design is the All-Digital nature of the Time to Digital Converter (TDC). Being a digital circuit, it is extremely robust to noise sources within the IC and process variations during the fabrication process. The only calibration required with this circuit is to tune the delay cells within the Vernier-Ring TDC core.

Compared to ADC based implementation, we also reduce the total area and power consumed by our circuit. As can be seen in Figure 3.25, the TDC covers only a fraction of the total area covered by one ADC.

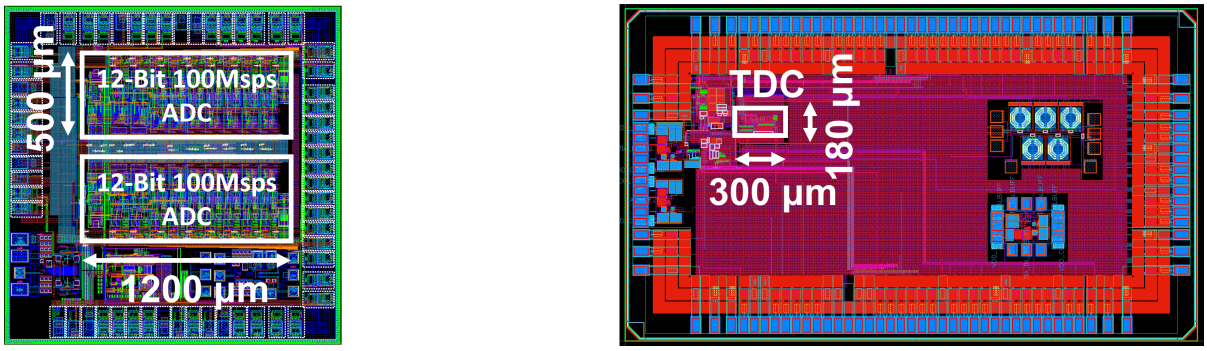


Figure 3.25: Comparison of the area covered by the ADC vs. the area covered by the TDC

The price we pay, however, to get rid of I-Q demodulation is that we now lose the ability to differentiate between the desirable beat note and undesirable beat note from Figure 2.1. Without I-Q demodulation, if the laser moves too quickly to a different comb tooth, we can inadvertently lock to the wrong comb-line.

The final drawback of this system is that similar to a conventional OEPLL, our minimum step size is now limited to integral multiples of our TDC. If we wanted smaller steps, we would have to implement a Fractional-N PLL on chip that would increase the complexity and noise of our design.

Despite the drawbacks relating to the minimum step size, the gains accomplished by eliminating the frequency dividers in our system are tenfold. This omission not only reduces the overall power and area of our system, but also removes the noise the dividers

would normally divide within the bandwidth of our system.

Offset locking to a comb offers absolute laser stability and we have demonstrated some spectacular results in the previous two chapters; however, this architecture also involves the use of locked combs using two resonators (SiN and SiO_2 based) that need to be stabilized and locked to each other, before a laser being locked to one of them. In the next chapter, we will deal with a synthesizer that is not concerned with the absolute stability of our lasers but is rather interested in locking two lasers with >100GHz offsets between them.

Chapter 4

Frequency detection using a 3x3 wavemeter

As mentioned at the beginning of this manuscript, the architectures described in Chapter 2 and Chapter 3 are incredibly efficient solutions for applications that demand optical light with "absolutely" stable frequencies, however, if an application cared only about the relative stability of two lasers locked together - the solution described in the previous chapters would perhaps be superfluous. In this chapter, we look at a novel new way of generating microwave signals of arbitrary frequencies first described by T. Komljenovic et al.[38] and further add to this body of work by leveraging some of the tricks and techniques we have used in the previous chapters.

While the work done in [38], demonstrates stable locking results and the ability to create arbitrary frequencies, it still falls shy of its promise of delivering multiple GHz frequencies. The work done in this chapter aims to offer an alternative locking architecture that uses DSP to calculate the wavelength of the laser and aims to address some of the pitfalls of using a purely analog approach to laser locking.

The next section will start by explaining how the "wavemeter" was constructed and

motivate the reasons behind the design choices. The locking technique used in [38] will then be described followed by a discussion of some of its shortcomings. We will finally discuss our locking architecture and end the discussion by showcasing some of our results and review important observations made over the course of the experiment.

4.1 System Overview

The architecture shown in Figure 1.5 offers a viable solution to the problem of locking a laser's wavelength. As can be seen from Figure 1.6(a), in the ideal case where the laser's wavelength does not change with time, the output of the interferometer also does not change with time. Moreover, a linear change in the laser's wavelength results in a sinusoidal change in the interferometer's output. This fact can be used to lock the laser's wavelength to a particular value by monitoring the interferometer output. As with the other negative feedback loops discussed in the previous chapters, if the output of the interferometer moves in a direction, the laser needs to be controlled in such a way that the output moves back to its original set-point. Thus, by locking this output voltage, the laser's wavelength will be indirectly locked. This lock however, is completely dependent on the fact that the laser's output power stays constant and does not change with time. If the laser's power changes, it will inadvertently change the output of the interferometer and will incorrectly be interpreted as a change in laser's wavelength.

The solution to this problem is to use a 3x3 light coupler instead of a 2x2 coupler and use the topology used in Figure 4.1. If designed correctly, 3x3 couplers provide a phase shift of 120° between the outputs of its arms [39].

Figure 4.2 shows the wavemeter output as the wavelength is varied linearly with time. The phase shift between these arms makes it possible to generate quadrature signals, which can then be used to unwrap the phase. This calculation provides robustness

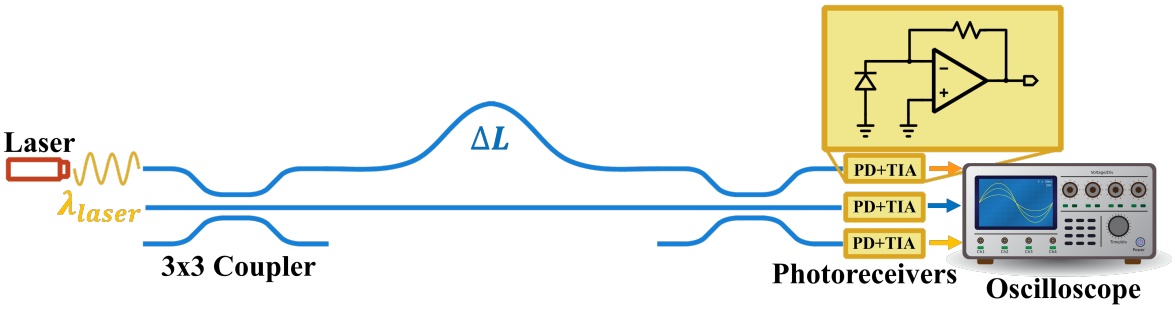


Figure 4.1: Interferometer created similar to 1.5 but made using a 3x3 coupler instead of a 2x2 coupler

against any variations in the output power of the laser. This is because an increase in laser power will affect all the channels equally and so the phase calculation from the quadrature signals should then nullify the effect of these variations.

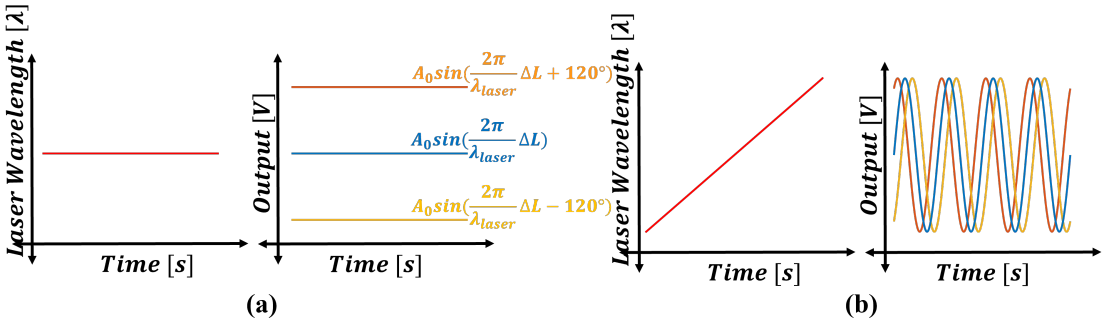


Figure 4.2: The output of the wavemeter as (a) the wavelength is held constant (b) Wavelength is changed linearly with time. Using a 3x3 coupler now gives us access to three sinusoidal outputs that have a phase difference of 120° between each other

Figure 4.3 shows this unwrapped phase and its relationship with the laser frequency. Phase response as the one shown in Figure 4.3 helps precisely determine the laser's frequency and its direction of drift. As is already deducible from the figure, if two servo loops with different time constants are employed, it is possible to potentially lock the laser with MHz-level resolution but at the same time possess GHz level offsets without the use of any high-speed or Radio-Frequency (RF) electronics.

As mentioned in Chapter 1, the interferometer response repeats itself periodically with a period determined by the ratio $\frac{\Delta L}{\lambda_{laser}}$. This period is called the Free Spectral Range (FSR) of the Wavemeter. With prior knowledge of the wavemeter FSR, the "slow loop" keeps track of the number of rotations in the phase and makes it possible to move the laser to GHz offsets with respect to the reference set point. Once the slow loop has brought the laser to the correct set-point, the "fast loop" can be used to lock the laser to a specific phase value with an extremely high degree of accuracy, thereby giving MHz-level resolution.

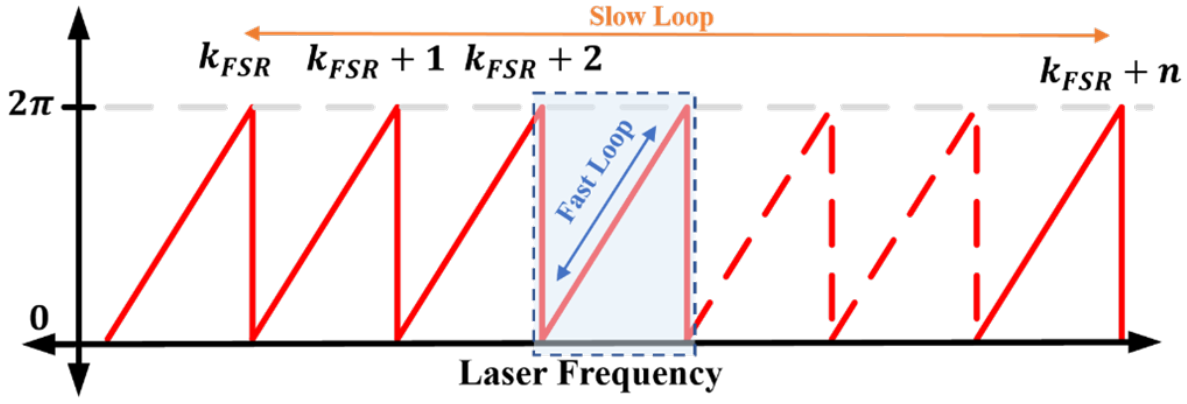


Figure 4.3: Tuning laser frequency using the Free Spectral Range of a 3x3 wavemeter

4.2 Controller Architecture

As mentioned in the preceding section, the frequency control system consists of two feedback loops with differing time constants. The loop with the larger time constant is dubbed the "slow loop" and can be seen in Figure 4.4. The slow loop actively measures the instantaneous phase of the wavemeter output and at the same time keeps track of its rotation history using a binary counter. The counter increments or decrements every time it encounters a phase rotation based on its direction. The output of this rotation counter

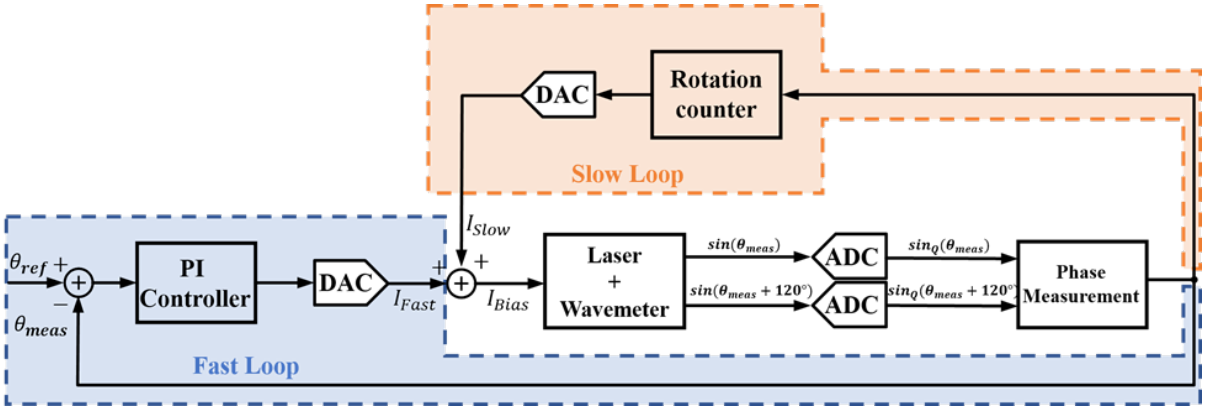


Figure 4.4: Block Diagram of Controller Architecture

determines the large signal set point for the laser and thus it drives a Current Digital to Analog Converter (DAC) with a very wide output range (50mA-300mA). With the large frequency offset taken care of, a traditional Proportional-Integral (PI) controller is used to tightly lock the instantaneous phase output of the wavemeter to a specific phase value. The aim is to have the bandwidth of this loop $> 1MHz$, thereby providing exceptionally high long-term stability and potentially kHz-level stability. All the digital signal processing (DSP) for the loop will be carried out using Digilent's Zedboard FPGA evaluation board which features a Xilinx Zynq 7000 FPGA/ARM SoC. To keep other noise factors to a minimum custom laser drivers are employed and a 12-bit Analog to Digital Converter (ADC) to measure the instantaneous phase.

4.3 Digital Signal Processing

All the digital signal processing (DSP) for the control loop is implemented using the Xilinx Zynq-7000 all programmable SoC (AP SoC). Although this FPGA features 100 DSP slices which can carry out up to 18x25 bit multiply and accumulate operations, a multiplication operation is carried out only once and all the multiplication and division operations have been replaced with left shift and right shift operations respectively.

4.3.1 Wavelength calculation from 3x3 coupler outputs

Once the coupler outputs are sampled by the Analog to Digital Converters (ADCs), a few simple trigonometric manipulations can be implemented to calculate the In-Phase (I) and Quadrature (Q) components as shown below. The In-Phase component is calculated as shown in Eq. (1) and the quadrature component is calculated by simply multiplying the $sin(\theta_{meas})$ term with $sin(120)$. These I-Q components are used to calculate the instantaneous phase using an FPGA implementation of the CORDIC algorithm. Instead of using expensive multiplication and division operations, the CORDIC algorithm iteratively computes the arc tangent using shift and add blocks, and with the help of a look up table. Figure 4.5 illustrates the implementation of this algorithm in our FPGA.

$$sin(\theta_{meas} + 120^\circ) = sin(\theta_{meas})cos(120^\circ) + cos(\theta_{meas})sin(120^\circ)$$

$$\implies sin(\theta_{meas} + 120^\circ) - sin(\theta_{meas})cos(120^\circ) = cos(\theta_{meas})sin(120^\circ)$$

$$\implies \theta_{meas} = tan^{-1}\left(\frac{sin(\theta_{meas})sin(120^\circ)}{cos(\theta_{meas})sin(120^\circ)}\right)$$

The I-Q computation is the only time a multiplication operation is used in the control loop. The multiplication with $sin(120^\circ)$ can be reduced to a right shift of 1 bit, however, multiplication with $cos(120^\circ)$ will still use one DSP slice.

4.3.2 Proportional Integral (PI) Controller

Fig.4.6 shows the implementation details of the Proportional-Integral controller. Similar to the filters implemented in Chapter 2, all the filter coefficients are limited to powers of 2. Implementing the filter in this manner converts all the multiplication and division

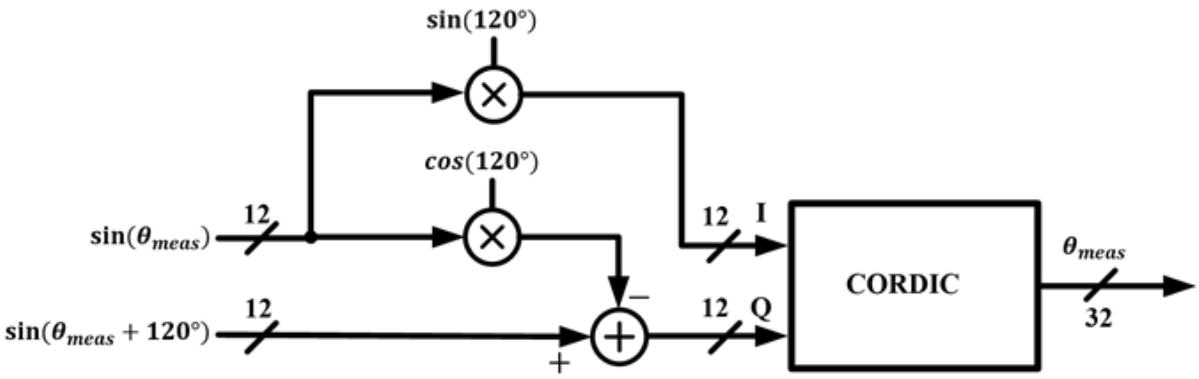


Figure 4.5: Block Diagram of Instantaneous Phase Calculation

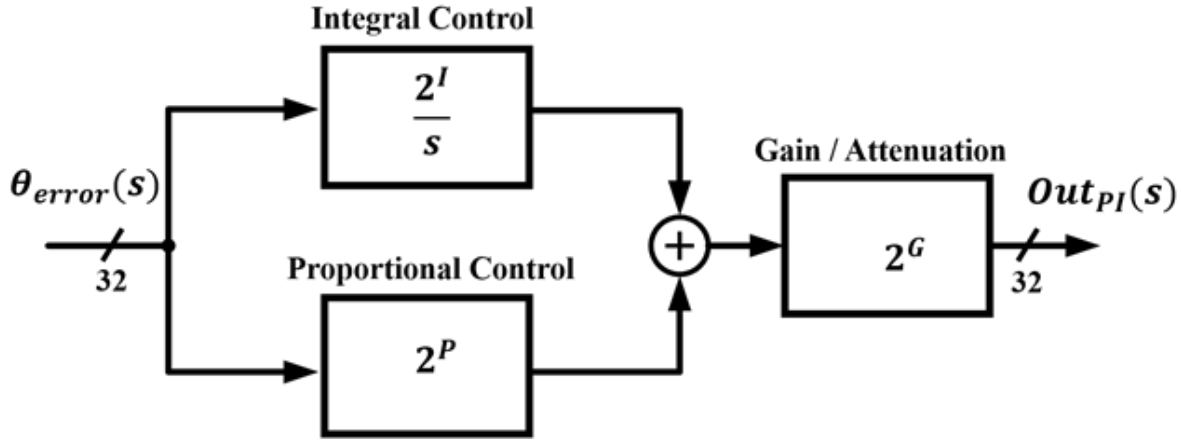


Figure 4.6: Block Diagram of Proportional Integral Controller

operations to simple shift operations, which avoids using the expensive DSP slices mentioned earlier.

4.3.3 Fast loop simulation using MATLAB

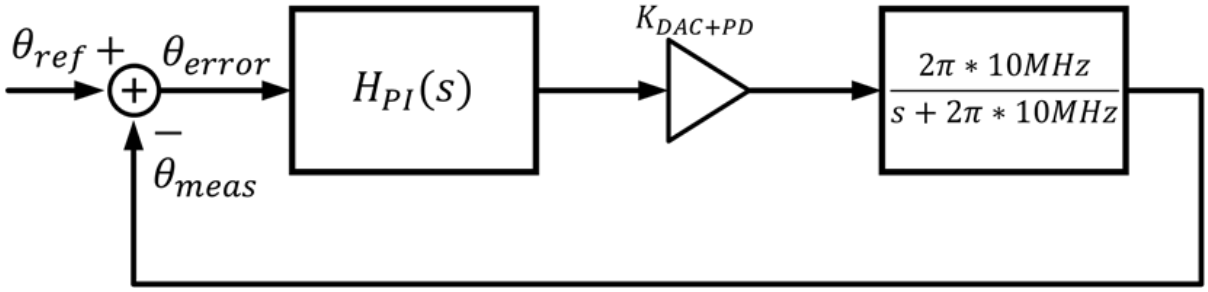


Figure 4.7: Simplified Block Diagram of the Fast Loop

Figure 4.7 shows a simplified block diagram of the simulation setup used to determine the PI coefficients that will stabilize the loop. A first-order low pass filter with a corner frequency of 10MHz emulates the laser modulation upper limit in the lock setup. This limitation comes from us using discrete components and cables to power the laser. A linear gain block models the laser phase output response and incorporates all other gain terms arising from the DAC within the same block. In practice, a calibration step needs to be run and the DAC output needs to be ramped to precisely measure this gain value. However, by adjusting the gain of the PI controller, the contribution of this term can be cancelled out. Finally, a block modelling the output response of our PI controller itself is added which will contribute to a pole and a zero in the loop. For the simulation, the PI coefficients are set so that the loop bandwidth is 2MHz (12.56M rad/s) while at the same time it is designed to posses a healthy phase margin . For the PI Filter (shown in Figure 4.6), the coefficients are set as P=-6, I=22, and G=5. Figure 4.8 shows the frequency response of the Fast loop for these settings and its corresponding transient response is shown in Figure 4.9. It can be seen that the loop has a bandwidth of 1.59MHz (11M rad/s) with a healthy phase margin of 75°. The transient simulation of the model shows a settling time close to 150ns with a transient response that would be consistent with a

phase margin $> 45°$.

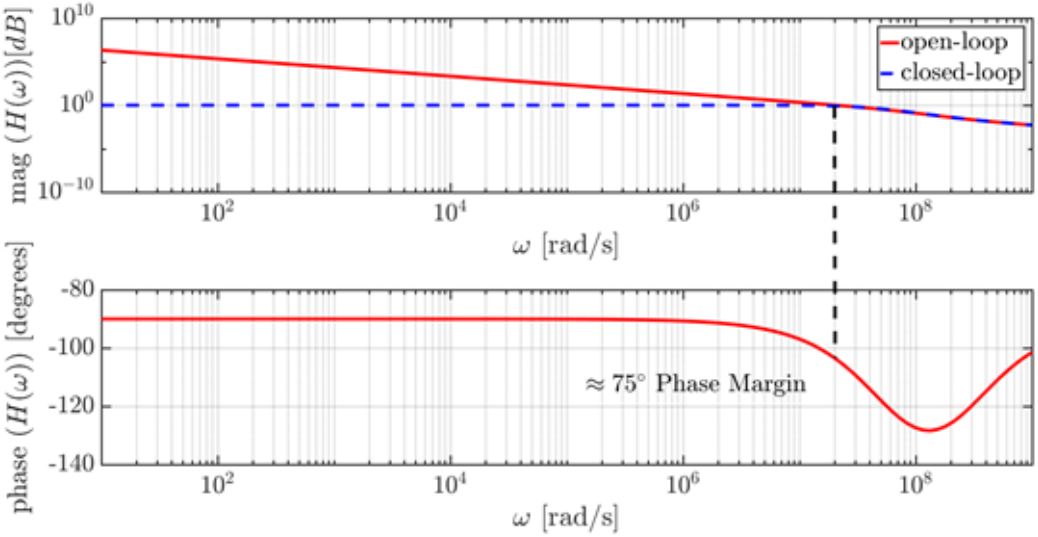


Figure 4.8: Fast loop Frequency Response

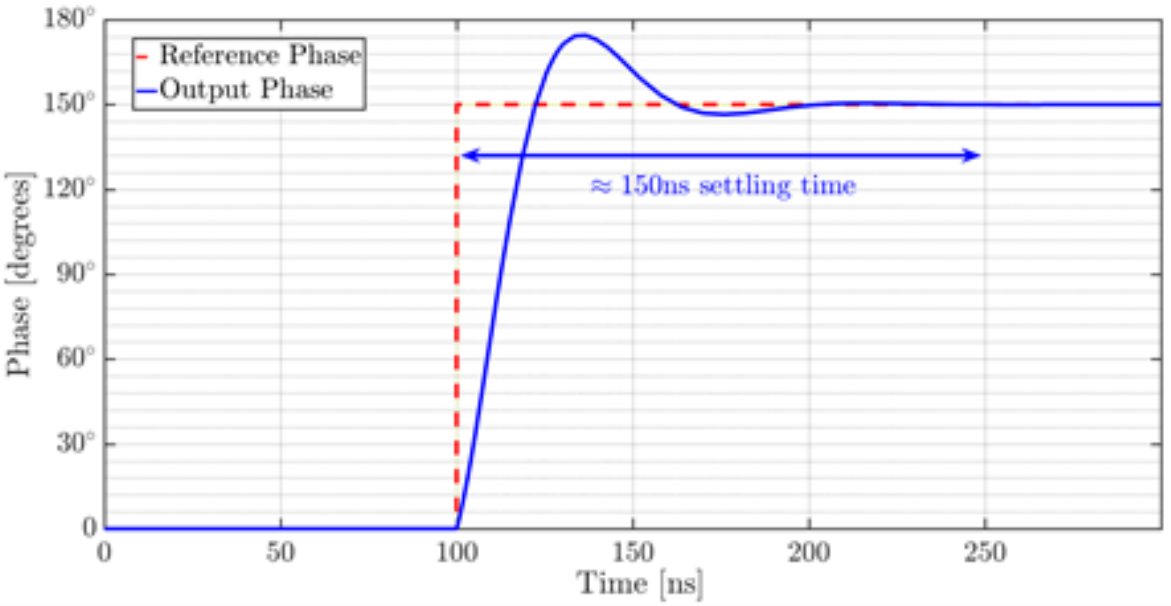


Figure 4.9: Fast loop Transient Response

4.4 Graphical User Interface and ARM Cortex A-9 processor

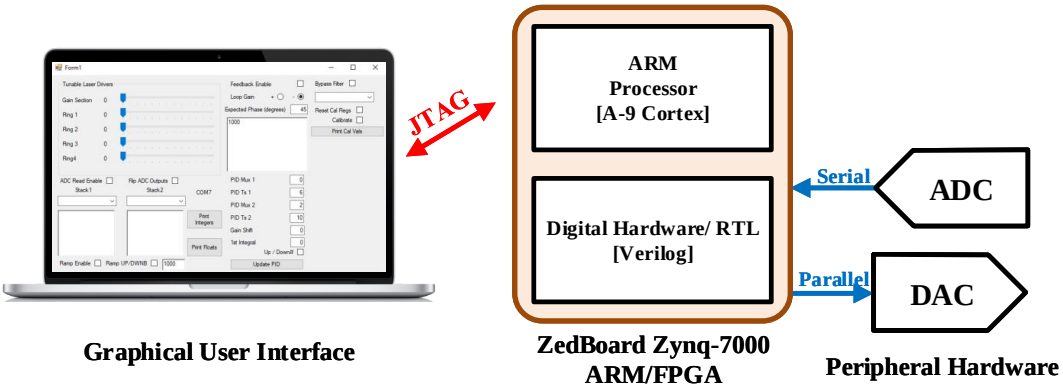


Figure 4.10: Graphical User Interface (GUI) written using Visual Basic interacts with the ARM Cortex A-9 processor which controls low level RTL hardware written in Verilog

In addition to its programmable logic core, the Zynq-7000 SOC also has a dual-core ARM Cortex A9 processor. The processor allows interface with lower-level Verilog code using an interface written in C and all the I/O periphery is treated as register mapped IO. As illustrated in Figure 4.10, the Graphical User Interface (GUI) communicates to the ARM processor using a JTAG cable which in turn controls the lower-level Verilog code using the AXI interface. The Verilog code interfaces with various I/O peripherals that enables the FPGA to carry out the digital signal processing we discussed in the preceding section. For Analog to Digital conversion (ADC), the MAX 11192 Eval board is used which houses a MAX 11192 IC. This ASIC consists of two 2MSPS SAR ADCs which have a resolution of 12 bits each. On the other end, a custom board is being used as the laser driver. This board features the MAX 5885 for its Digital to Analog Conversion (DAC). This 16-bit 100MSPS DAC generates the feedback signal to the laser, but also debug

various other signals along the signal chain. To conceal the complexities of the underlying code and to make the system easier to operate, a Graphical User Interface (GUI) designed using Visual Basic communicates with the FPGA and associated peripherals. As can be seen in Figure 4.11, the GUI can be used to change various laser biases and alter the Proportional-Integral-Differential (PID) coefficients of the overall servo loop. The GUI makes it possible to bypass the loop filter and view intermediate signals along the signal chain. This feature is used in the next section to ensure that the internal Verilog code is correctly calculating the phase difference between the arms of the wavemeter.

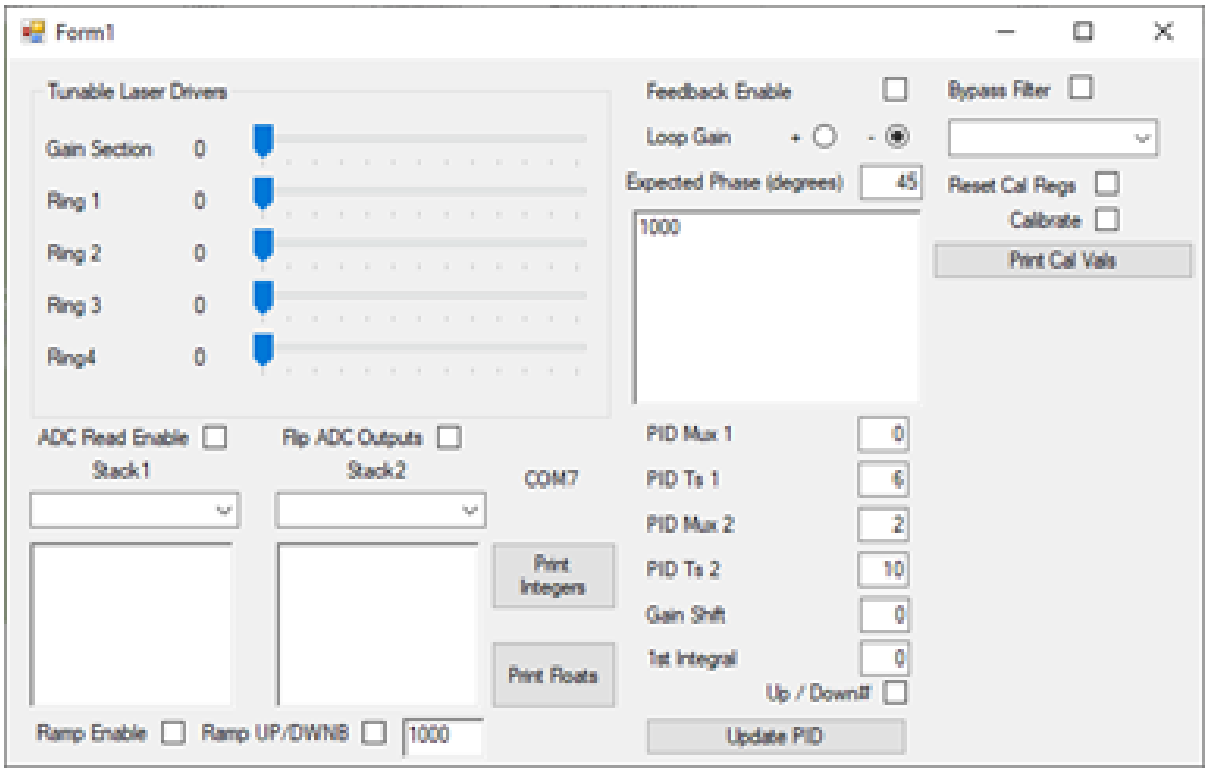


Figure 4.11: Graphical User Interface (GUI)

4.5 Custom Photo Receiver Board - version 1

To get a precise estimate of the phase difference between the wavemeter arms is, the output power of each wavemeter arms need to be converted to its voltage equivalent. Figure 4.12(a) shows a simplified schematic of the photo-receiver circuit being employed. The circuit consists a photodiode-TIA combination that then drives an opamp buffer. The Printed Circuit Board (PCB) also consists an optional path that utilizes a tunable potentiometer instead of the TIA. This enables us to change the transimpedance gain of the circuit should the need arise to do so. Finally, in order to measure the power output of 3 wavemeter arms, the PCB uses the same circuit thrice – one for each arm of the wavemeter.

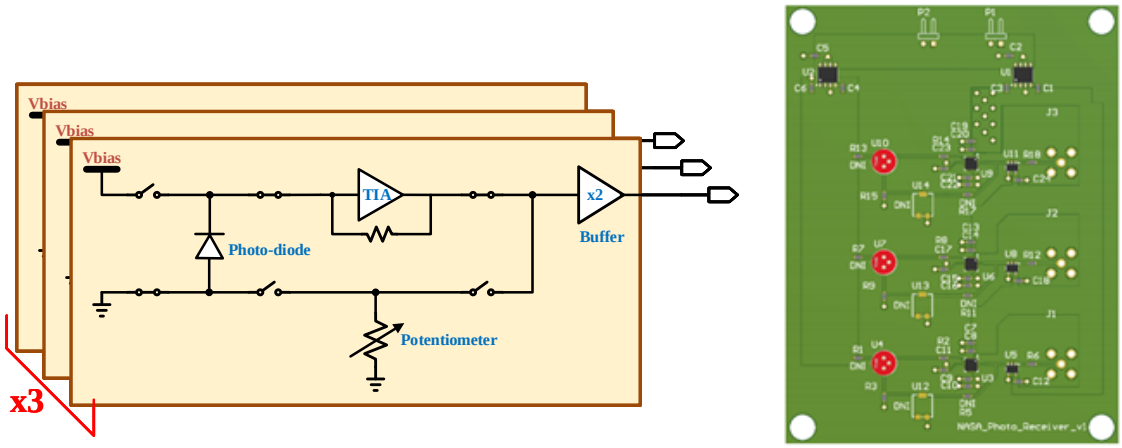


Figure 4.12: (a)Simplified schematic of photo receiver circuit (b) 3D render of the PCB

4.6 Custom Photo Receiver Board - version 2

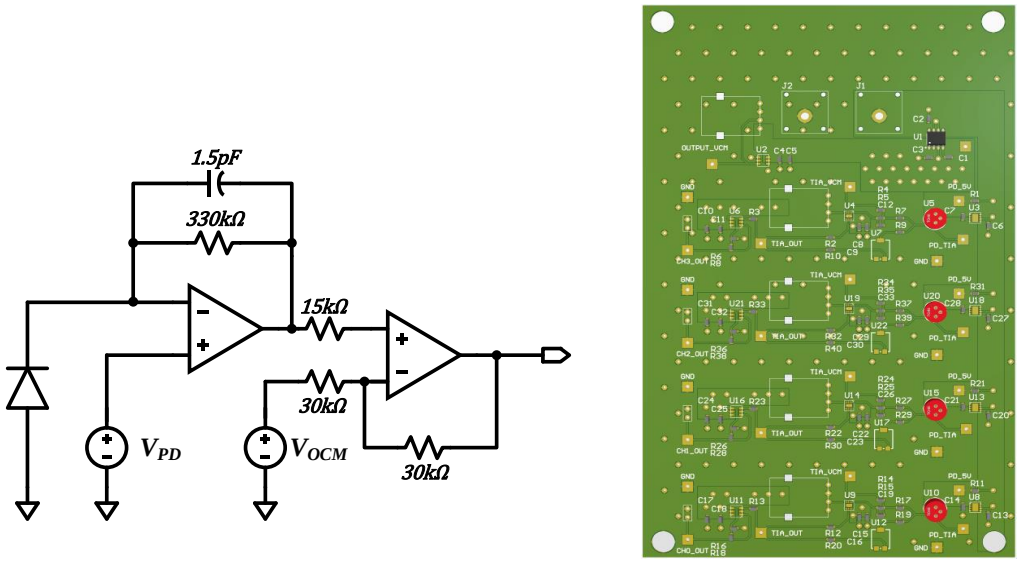


Figure 4.13: (a)Simplified schematic of photo receiver circuit (b) 3D render of the PCB

Based on measurements from the first version of the board, new version of the photoreceiver board was designed. The new schematic for this board can be seen in Figure 4.13(a). A Transimpedance amplifier (TIA) with a 330kOhm resistor as the feedback is used to convert the photocurrent to a voltage. On the positive terminal of the TIA, a voltage bias now sets our photodiode bias. This is an improvement over the previous implementation as the negative feedback now ensures that the voltage bias is now fixed and does not change with the photocurrent. In addition to the TIA there is another amplifier which adjusts the common mode of the TIA output. The reason to add this is that the Analog to Digital Converter (ADC) has reference voltage of 2V, which means that any input greater than 2V is not correctly sampled by the ADC. The final board

can be seen in Figure 4.13(b), where we have 4 copies of the same circuit shown in Figure 4.13(a).

4.7 Measurement Results

4.7.1 Instantaneous Phase Measurement

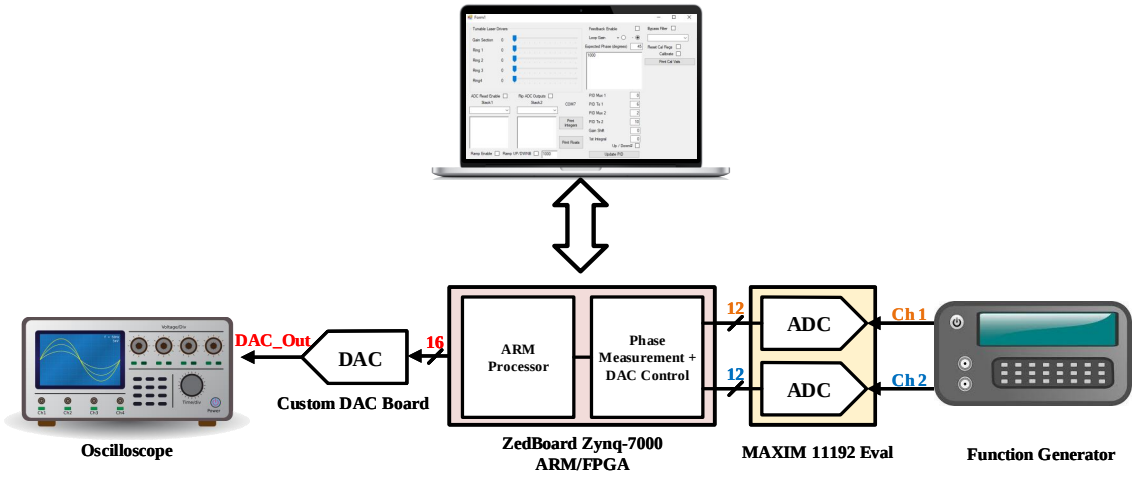


Figure 4.14: Schematic of test setup used to measure the instantaneous phase difference through the wavemeter

Figure 4.14 shows the experimental setup used to ensure correct hardware operation and to verify that the digital signal processing is calculating the instantaneous phase accurately. An Arbitrary Waveform Generator generates two sine waves that have a phase difference of 120°, which are then sampled by the MAX 11192, which consists of two 12-bit 2MSPS ADCs. The digitized versions of these two signals are then processed to generate the quadrature signals required by the CORDIC algorithm to calculate the instantaneous phase. This 32-bit instantaneous phase is then truncated to a 16-bit value which is then output by a high-speed DAC (MAX 5885) for viewing.

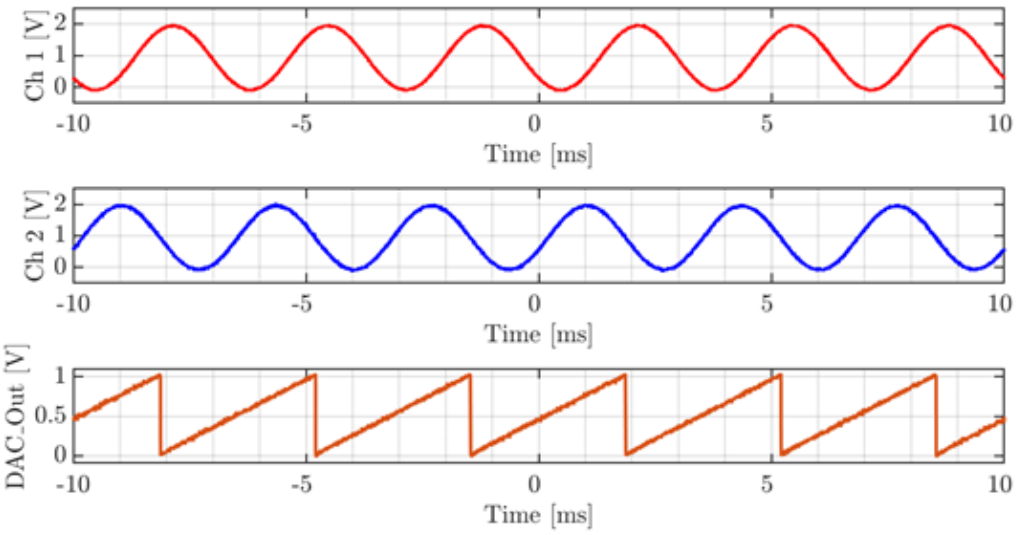


Figure 4.15: Measured outputs verifying correct phase measurement operation

Figure 4.15 shows the various traces measured using an oscilloscope and verify correct operation. The traces shown in red and blue are emulating the outputs of the wavemeter that are then fed into the ADC. The trace plotted in orange is the output of the CORDIC block which, as you would expect, is a ramp function.

4.7.2 Incorrect Phase Measurement due to non idealities

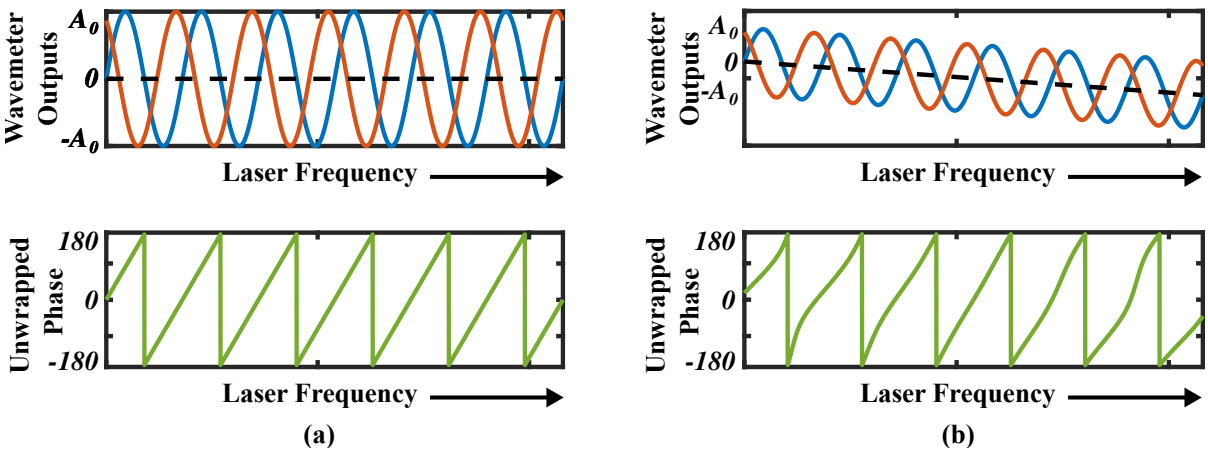


Figure 4.16: Simulation results (a) Under Ideal operation the wavemeter has a constant DC common mode across multiple FSRs (b) The DC common mode in this case has a frequency dependence which introduces an error in our phase unwrapping

Figure 4.16(a) shows simulation results when the phase unwrapping DSP block receives two ideal sine waves that are perfectly 120° apart. Here, the two sine waves are perfectly centered at a DC common mode of 0 and both sinewaves have the same amplitude. The output of the DSP block is shown below the sine waves. As can be seen from the figure, the block perfectly unwraps the phase from the two sine waves and gives the graph shown in green.

Unfortunately, this is not what the output of the wavemeter looks like coming out of the integrated wavemeter. The integrated wavemeter's output power has a dependency on the laser's frequency and so, as the laser's frequency is swept, the DC common mode of the wavemeter's output moves over time (Figure 4.16 (b)). This change in common mode introduces an error in the phase measurement and causes the output of the DSP block to have a "swiggly" nature to it.

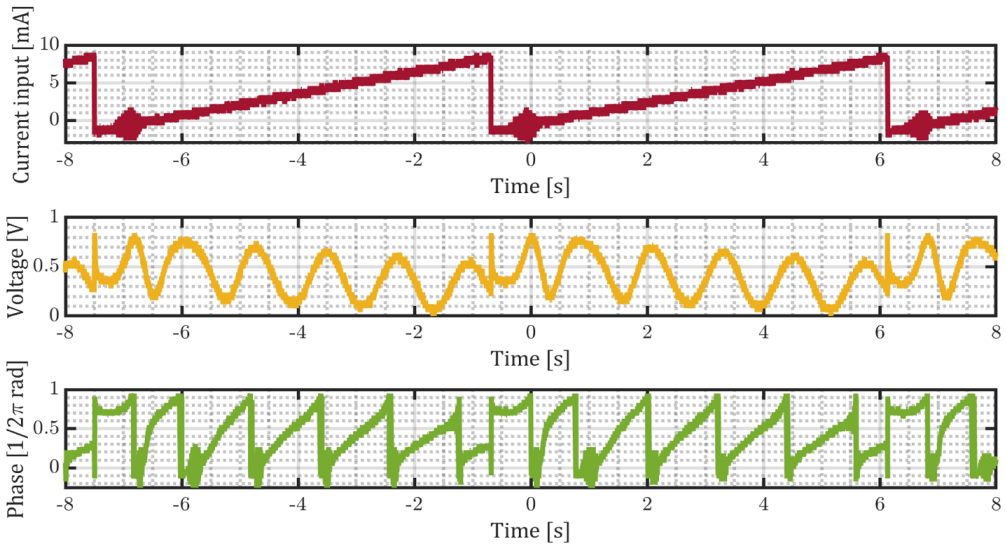


Figure 4.17: Measurement Results showing the wavemeter output and the unwrapped phase

4.7.3 Long Term Stability

Figure 4.18 is a simplified schematic of the setup used to lock two lasers to the wavemeter. The setup used here is very similar to the previous experiment, however, instead of using a function generator to generate inputs to the ADC, the outputs of the wavemeter are fed to the photoreceiver board. The voltages from the board are then used to calculate the instantaneous phase of the lasers through the wavemeter. A simple PID control loop tries to control this instantaneous phase and lock it to a specified set-point. To verify the quality of the lock, a 3dB coupler is used to mix the light coming out of each laser and the beat note is observed.

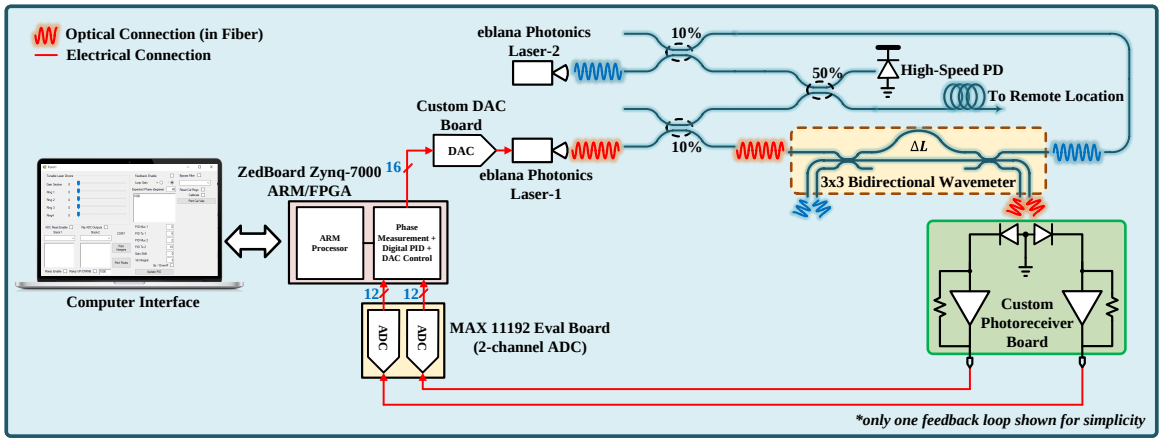


Figure 4.18: Simplified schematic of Full Loop

Figure 4.19 shows the actual setup for the synthesizer set up in the lab.

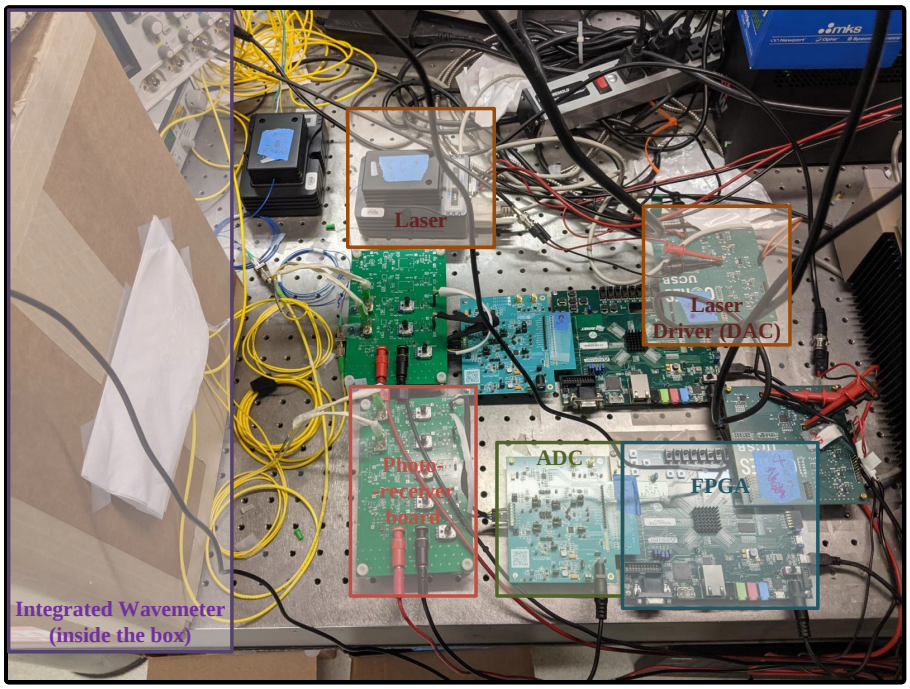


Figure 4.19: Photograph of Full Loop

Figs 4.20(a) and (b) display the beat note of the synthesizer after it has locked

the amplitudes coming out of the wavemeter. The wavemeter has been sealed inside a cardboard box filled with packing peanuts to isolate it from temperature variations from the surroundings. However, peculiar behaviour is observed from the beat notes during different parts of the day. The laser's being used for our lock have a linewidth in the order of a MHz and so expect MHz level variations in the beat note is to be expected. However, earlier in the morning, as lab ambient temperatures were increasing, it was noticed that the locked beat note was increasing and this linear increase was >10MHz. Similarly at night, when temperatures were decreasing, the beat note would follow a downward trajectory.

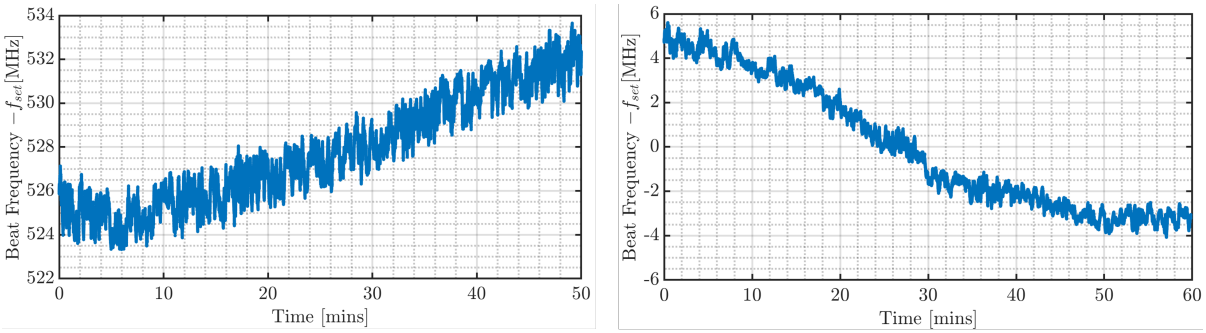


Figure 4.20: Frequency vs. Time plot of beat note between the lasers measured with wavemeter placed inside a box during (a) day time when lab temperatures were rising (b) night time when lab temperatures were falling

To ameliorate the variations caused due changes in ambient temperature, the packaged wavemeter is mounted on a heat sink attached to a ThermoElectric cooler (TEC). An illustration of this new setup can be seen in Figure 4.21.

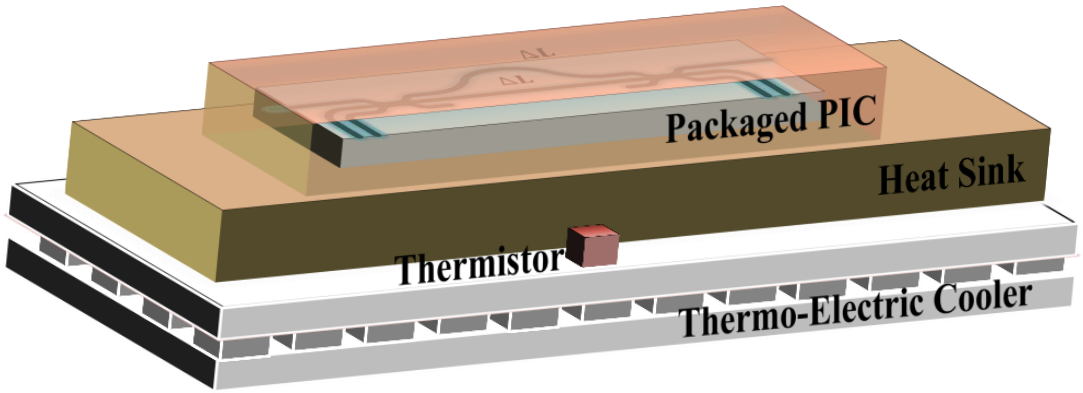


Figure 4.21: Illustration of setup used to stabilize the integrated wavemeter's temperature

As a result of this temperature stabilization, a marked improvement is seen in the drift of synthesizer output. Figure 4.22 shows a Frequency vs. Time plot of the synthesizer output after it has been locked to a fixed point along the wavemeter and the wavemeter's temperature has been stabilized using a TEC. As can be seen from the figure, the wavemeter output no longer drifts by 10s of MHz over an hour and stays centered around a fixed mean.

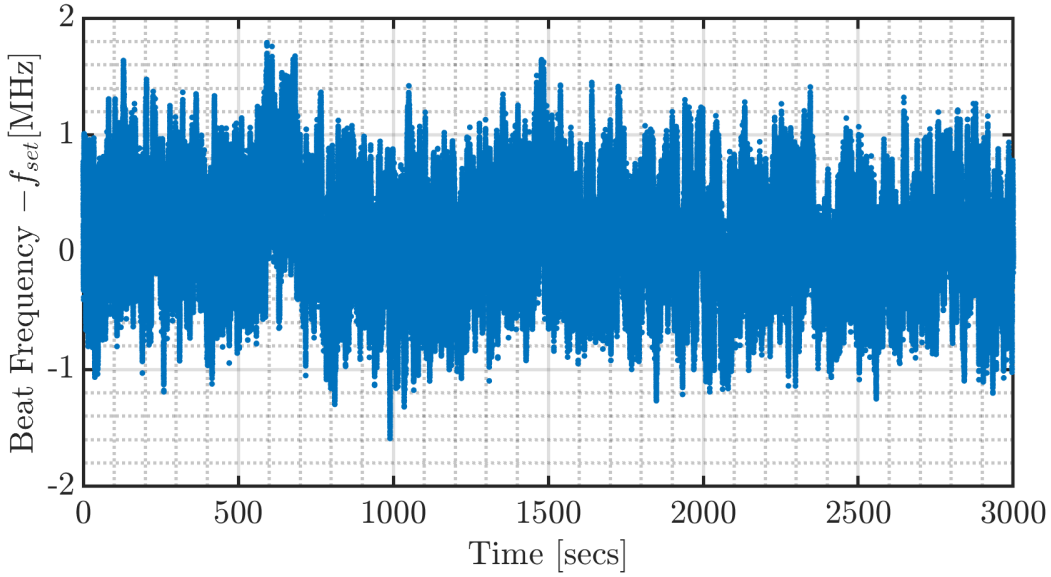


Figure 4.22: Frequency vs. Time plot of beat note between the lasers measured for 3000 secs with wavemeter resting on a temperature controlled stage (Gate time for the frequency counter = 10ms)

Figure 4.23 shows a histogram of this long term measurement. Looking closer at Figure 4.22 from 0 to 1000 secs, it can be seen that the beat note is locked to a higher frequency than its later set point. This can also be seen in our histogram in Figure 4.23, which appears as if a Gaussian Bell curve has been dragged lower.

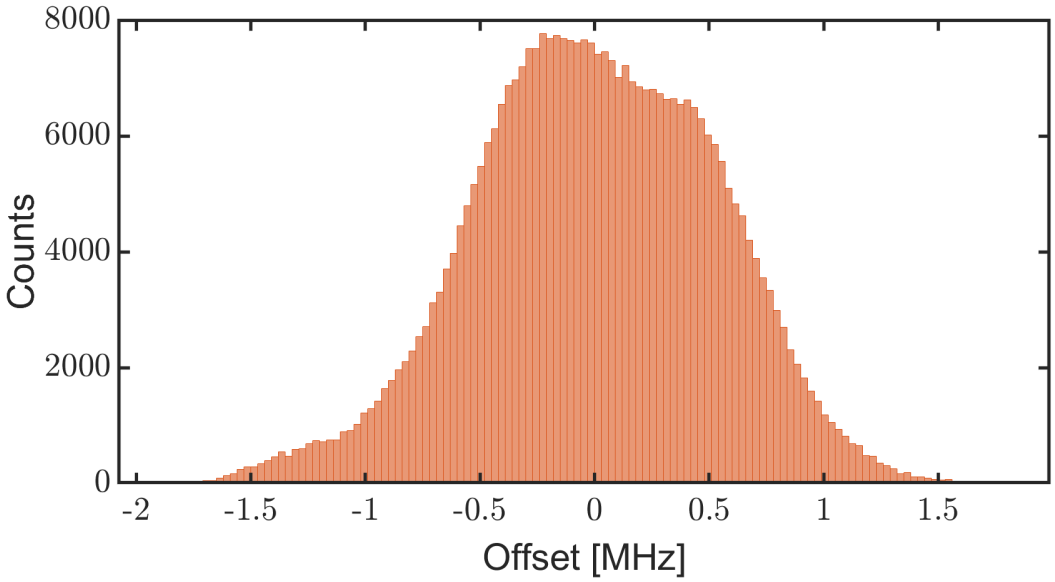


Figure 4.23: Histogram of beat note between the lasers measured for 3000 secs with wavemeter resting on a temperature controlled stage (Gate time for the frequency counter = 10ms)

While our temperature controlled stage works to stabilize the wavemeter's temperature, there is still a large time constant associated with the temperature loop as the wavemeter is still separated from the TEC (where the thermistor is mounted) by a large heat sink and the wavemeter's package itself.

4.7.4 Large Frequency Steps

This section illustrates the real advantage of this locking scheme. By utilizing the same low speed electronics that give MHz offset frequencies between the two lasers, it is possible to use them to separate the lasers precisely with an offset of 100s of GHz (even THz). Figure 4.24 shows the instantaneous beat note between the two lasers as one of them is moved 2 FSRs away from their set point in steps of 5°. However, the laser does frequency does not move linearly proportional to our input stimulus. The reason for this non-linearity is the error introduced to the phase measurement algorithm due to the

varying DC common mode of the wavemeter outputs (shown in Figure 4.17). It can be seen that the shape of the frequency graph precisely matches the shape of the measured phase and that is because they are both inverse operations of each other.

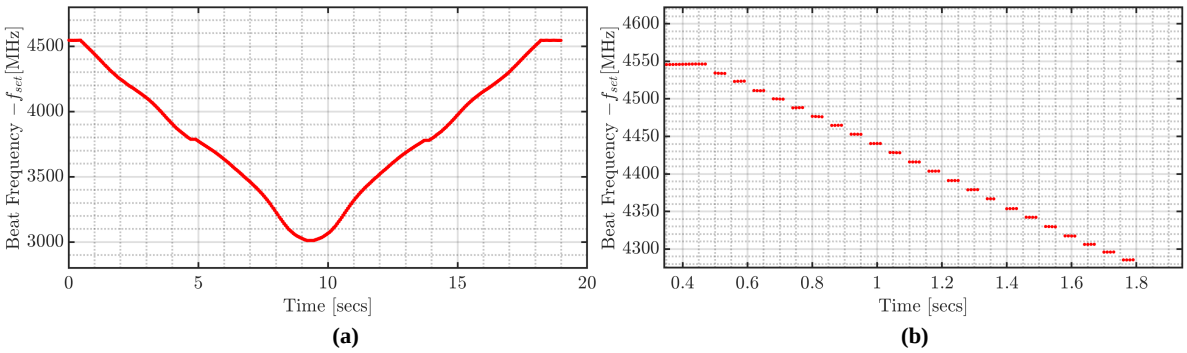


Figure 4.24: (a) Frequency counter data measuring the instantaneous beat note of the two lasers as they are moved multiple Free Spectral Ranges (FSRs) away from each other (b) Zoomed in view displaying individual step sizes

There are several solutions to address this problem, and one of them is to simply run an initial calibration run and create a Look Up Table (LUT) of all the indices and their corresponding beat frequencies. Once stored,this LUT can be used to step through the desired output frequencies. For the results shown in Figure 4.24 we are stepping through 2 FSRs of the wavemeter by taking small steps of size equal to 5° and storing their corresponding frequencies in a LUT. This LUT can then be used to go to the desired output frequency as is illustrated in Figure 4.25 where we are now taking linear steps through the entire range of the wavemeter.

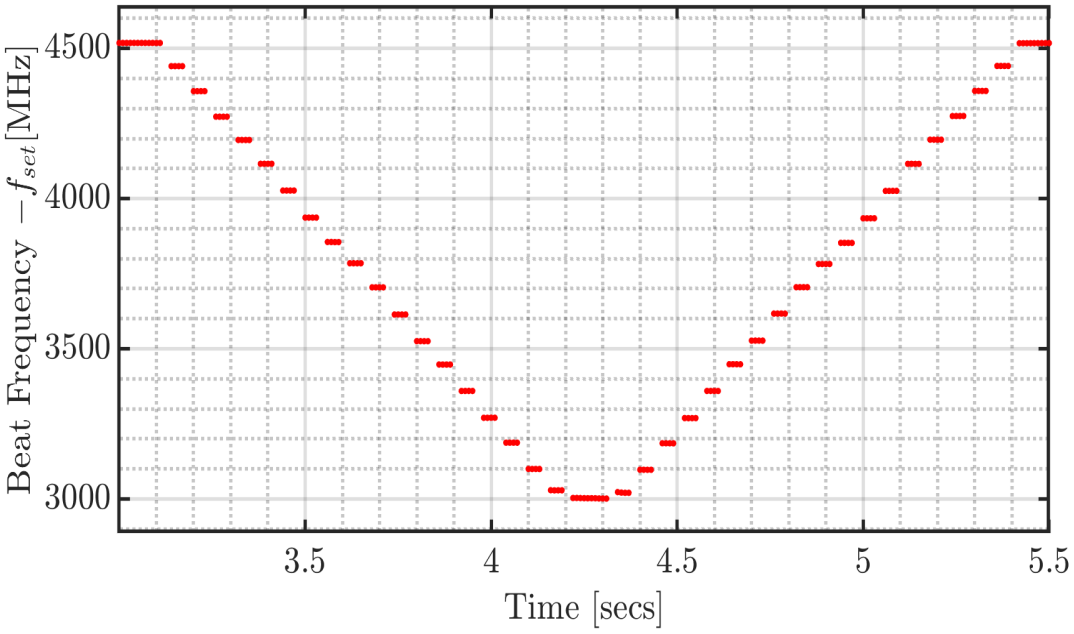


Figure 4.25: (a) Frequency counter data measuring the instantaneous beat note of the two lasers as they are moved multiple Free Spectral Ranges (FSRs) away from each other based on values from a Look Up Table (LUT)

4.8 Summary

In this chapter, we showed us how using a 3x3 coupler helps us generate quadrature components that can then be used to calculate the phase difference through our wavemeter. We show that by locking this phase difference to a constant value, we can lock our lasers to each other.

	Freq. Detection using ADCs		GF55nm ASIC	Wavemeter
Level of Integration	Discrete	Integrated	Integrated	Discrete
Power [W]	2.1 (1.4+0.7)	0.314	0.212	0.180
Frequency Dev. [Hz]	+/-60m	+/- 320k	+/-4k	+/-1.5M
Allan Dev. [τ @ 1sec]	$< 10^{-16}$	$6.5*10^{-11}$	$< 10^{-12}$	–
Step Size [Hz]	<1	100k	2.5M	~5M
Range [MHz]	695-2700	600-2000	500-7000	>100GHz

Figure 4.26: Table comparing the performance results with respect to the other synthesizers mentioned in this work

The table shown in Figure 4.26 might make it seem that the synthesizer we have implemented here does not perform as well as our previous implementations. However, one of the reasons our locked stability looks so much worse, is that the laser used in this implementation has a Lorentzian linewidth that is three orders of magnitude larger than our other lasers. And so we expect to see the deviation that we do.

However, the real strength of this implementation is the wide tuning range that this wavemeter based approach has to offer. Using the same low-speed electronics, we can move the lasers multiple FSRs away from each other and potentially get >100GHz offsets. The only drawback from this implementation is the wavemeter's susceptibility to temperature variations.

If we compare our implementation to the commercial bench-top source sold by [40], we provide orders of magnitude better frequency stability, while providing potential to demonstrate the same tuning range.

In the next chapter, we will look at some ways that this implementation can be improved upon in the future. We will also compare and contrast our various implementations, and discuss the trade-offs that need to be made when we design a synthesizer for a particular application.

Chapter 5

Conclusion

5.1 Future work

As with most scientific endeavors, there is always more work to be done and any system can always be improved upon. This section will discuss some of the work that has been left unfinished, and list some of the ways to improve upon the results that have been achieved thus far.

5.1.1 Fully integrated frequency synthesizer

The work done in this manuscript was funded by the Defense Advanced Research Projects Agency (DARPA) project named Direct On-Chip Digital Optical Synthesizer (DODOS). The aim of this project was to fabricate a fully integrated Optical Frequency Synthesizer capable of producing an optical output with Hz-level precision in a package with a volume less than $1cm^3$.

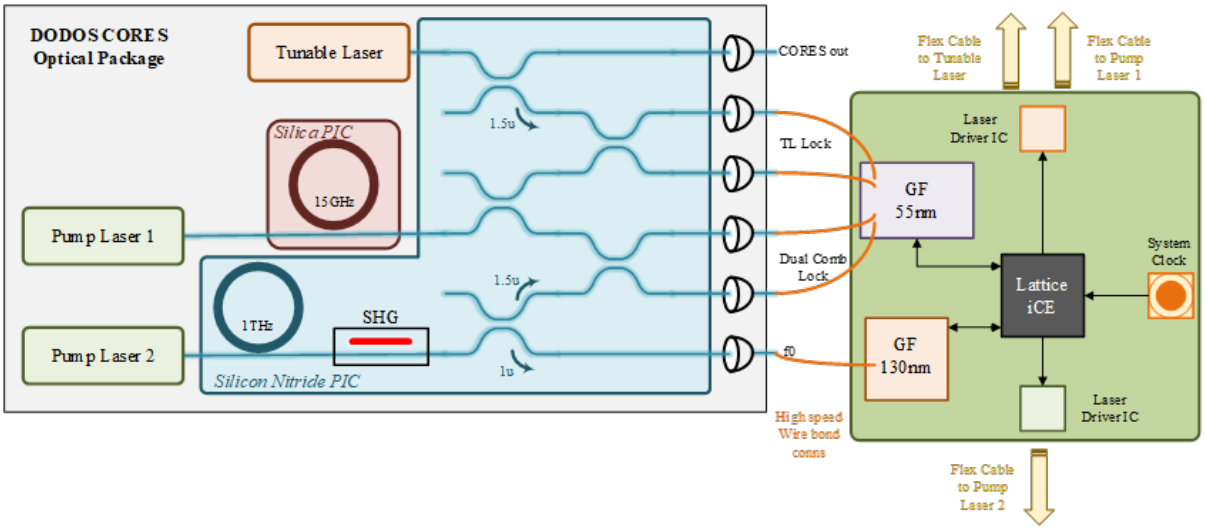


Figure 5.1: Simplified block diagram of DODOS Optical Synthesizer with Final DODOS PCB

Figure 5.1 shows a simplified block diagram of the final system package. This package includes an optical package that interfaces with an electrical PCB via flex cables. The optical package consists of three lasers, two micro-ring resonators required to generate two optical frequency combs with a repetition rate of 1THz and 15GHz respectively. The assembly also includes the Second Harmonic Generator and all the couplers required to generate our beat notes.

This optical assembly was directed wirebonded to a Printed Circuit Board (PCB) with the two ASICs (from Chapter 2 and Chapter 3) directly wirebonded to the board in a CoB package. The final PCB is a 6-layer Rigid-Flex PCB consisting of all the electronic circuitry that is necessary to drive the two pump lasers, and the widely tunable laser that produces the system output. In addition to the laser drivers, the PCB also features two custom ASICs (GF130nm and GF55nm) designed to process the photo-diode outputs generated by the optical package. The PCB also features a Field Programmable Gate Array (FPGA) that carries out all the control loop digital signal processing and facilitates an interface to communicate with the ASICs. Finally, the PCB provides a means to

provide an ultra-stable Radio Frequency (RF) system clock, which can be then used as reference the entire system locks to.

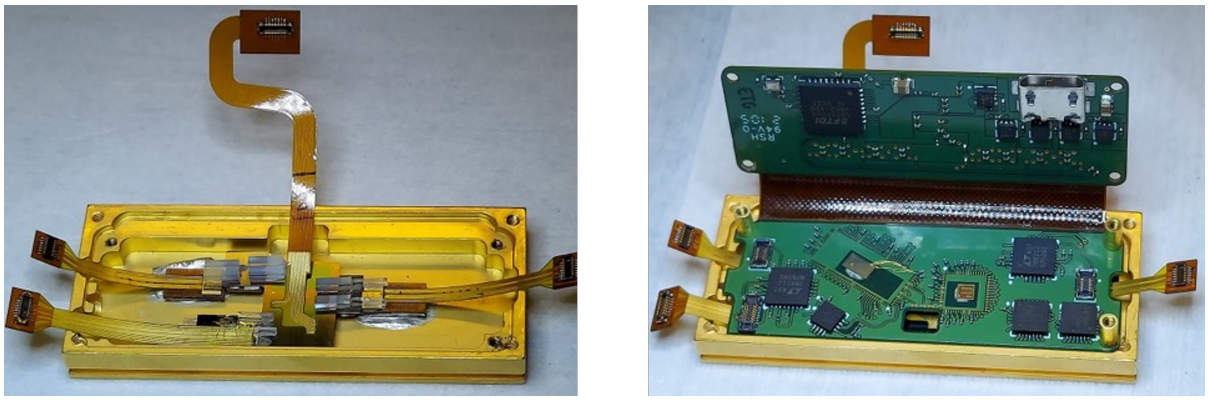


Figure 5.2: Actual Photograph of Final Assembly (Photographs taken by David Kinghorn [41])

The laser driver ICs used to drive the lasers is Linear Technology's LTC2662. This IC is a five-channel 16/12-bit current Digital to Analog Converter (DAC). The IC can be configured to have output ranges from a list of values (300mA, 200mA, 100mA, 50mA, 25mA, 12.5mA,6.25mA and 3.125mA). This makes it ideal to drive the tunable laser which needs five bias currents – gain section, SOA, phase section, and 2 rings. The area required by these ICs is still smaller than using two separate single-channel current DACs, and so use another IC to drive the two pump lasers.

While the system has been fully assembled, further testing is still required to demonstrate all the pieces working together in unison.

5.1.2 Wavemeter based lock

In the experiments with the wavemeter, it was attempted to stabilize the wavemeter's temperature by mounting a packaged version of it on a temperature controlled stage. As can be seen from Figure 4.22, this provided some stabilization to the frequency beat

note. However, if the Photonic IC (PIC) housing the wavemeter structure was directly mounted on a Peltier cooler or a Thermo-Electric Cooler (TEC) along with a thermistor right next to it, it might be possible to obtain results that could potentially be several orders of magnitude better.

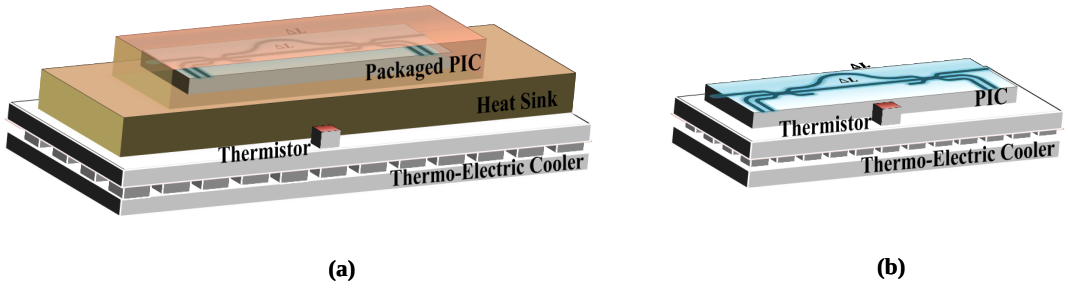


Figure 5.3: (a) Current setup used to stabilize the wavemeter's temperature. The packaged Photonic IC (PIC) is mounted on heat sink which sits on a ThermoElectric Cooler (TEC) (b) Better stabilization can be achieved by packaging the PIC directly on a TEC with a thermistor mounted nearby for precise temperature control

5.1.3 Using a Delta-Sigma $\Delta - \Sigma$ ADC

To measure the outputs of the wavemeter, the optical light is converted a voltage using the TIA on photoreceiver board. This voltage is then digitized using a Successive Aproximation ADC (SAR ADC). Once digitized, these signals are digitally processed to calculate the phase difference through the wavemeter. This signal is processed inside the FPGA and the digital output of the loop filter is then converted to an analog value using a high precision DAC.

For the system described here, the sampled analog values are very close to DC and do not require high speed Nyquist data converters. In this case, a Delta Sigma ADC would actually be the perfect candidate for data conversion. Moreover, a lot of work has been recently done to use Delta Sigma streams directly for Logical [42] and Algebraic operations [43]. Using these one can directly use the stream for feedback error calculation

and then use the output stream to directly drive a laser using a 1-bit current DAC.

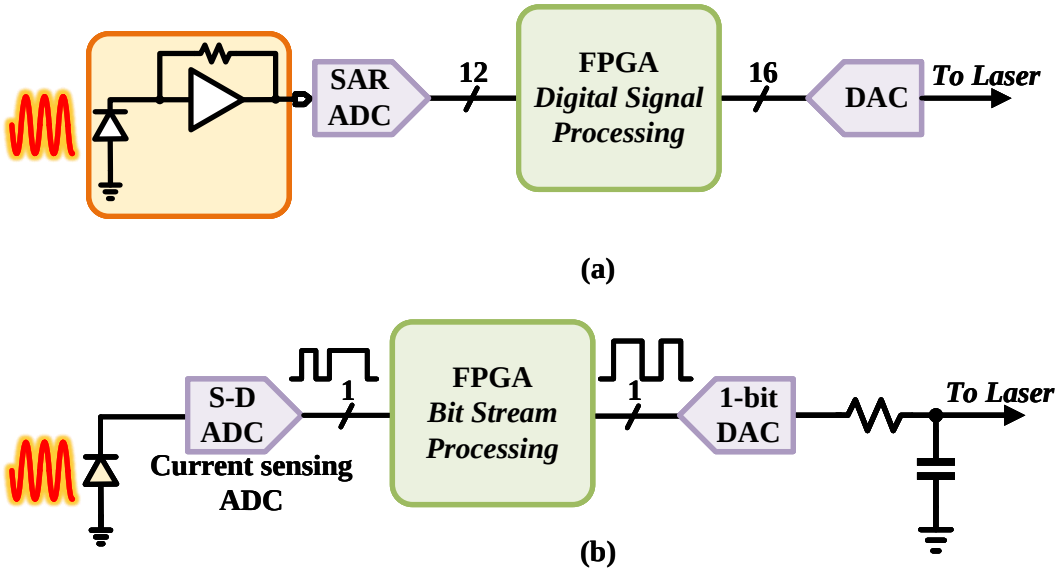


Figure 5.4: (a) Current setup used for wavelength stabilization (b) By replacing our 12-bit SAR ADC, we can simplify our DSP and also eliminate the need for a TIA and high resolution output DAC

Apart from simplifying the signal processing, the Sigma Delta ADC can be implemented as current sensing ADCs. These ADCs have been shown to have sensitivities down to 100pAs [44], and thus can fully eliminate the need for a TransImpedance Amplifier (TIA), further reducing the power consumed.

5.1.4 Using ultra low linewidth lasers

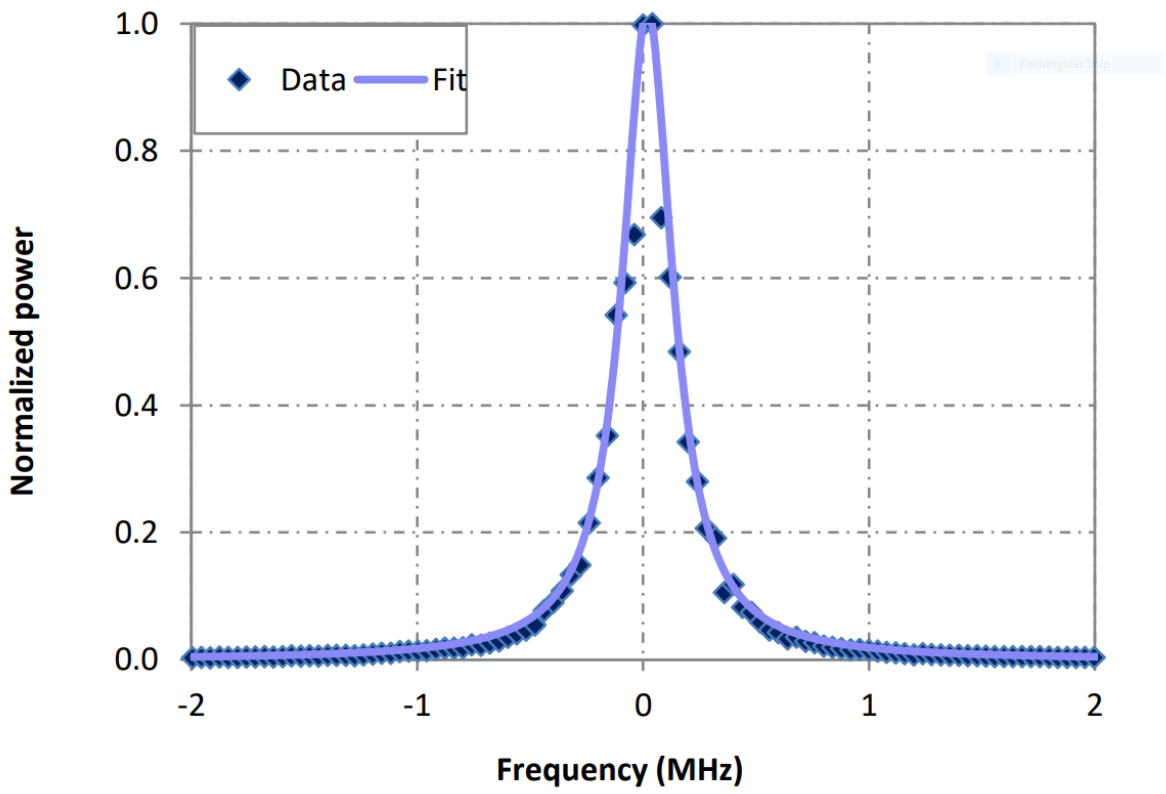


Figure 5.5: Linewidth of the laser being used for the locking experiment (138kHz)

Figure 5.5 shows the Lorentzian fit of the Linewidth spectrum of the laser being used for the lock described in Chapter 4. Comparing the histogram of the steady state lock shown in Figure 4.23 to this plot demonstrates that the linewidth of the laser is another area that is limiting the precision of the locking setup. Remarkable advances have recently been in made in the development of ultra linewidth lasers that display continuous tuning across the C-band [45] [46]. Using these lasers with integrated wavemeters of longer delays are the next step towards achieving higher levels of stability. Since the laser beat note is not part of the full loop in this system, any improvement to our loop gain or improvements in measurement, do not improve the beat note stability and thus, the

only direct way to improve this stability to use Ultra-low linewidth lasers, that correlate directly to a more stable beat note.

5.2 Conclusion

In this work, we have described the design and implementation of four different synthesizers. There are several ways we can classify them and their performance. In Chapter 2 and Chapter 3, we discussed synthesizer topologies that used an optical frequency comb as their reference. These combs are used to bring laser frequencies that are typically nearly 200 THz down to microwave frequencies, and enables us to translate the stability of microwave references to the optical domain. In Chapter 2, we began by building our first prototype using off-the-shelf board level components that allowed us to optimize every single part of the signal chain for maximum performance. As a result of these optimizations, we restricted the laser frequency deviations to a standard deviation of 626μHz from the mean. This implementation allowed us to control the laser's output frequency with Hz-level precision, thereby providing us with a stepping accuracy of one part in nearly 200 billion!

Motivated by the success of our board level implementation, we then attempted to reduce the 'Size, Weigh And Power' (SWaP) of our system. The entire system was taken and integrated into a 2mm*2mm Integrated Circuit (IC). The IC fabricated using Global Foundries 130nm BiCMOS process, and featured a fully integrated Radio Frequency (RF) front end. Designed to operate from frequencies starting from 50MHz to 2.5GHz, these front end featured a TransImpedance Amplifier, an RF Mixer and a Phase Locked Loop (PLL) that generates the Local Oscillator (LO) signal for the mixer. In this architecture, we require both the InPhase and Quadrature (I-Q) components of the mixed down beat note, the PLL was designed to produce two outputs that were phase shifted by 90° with

respect to each other, and the mixer was implemented as an I-Q demodulator.These I-Q Intermediate Frequency (IF) signals were fed to two pipeline Analog to Digital converters (ADCs) that were designed to operate at a sampling frequency of a 100MHz and with a resolution of 12-bits each. To test the system, we assembled the test Printed Circuit Board (PCB) with the ASIC wirebonded directly to the board. While the results from this integrated system are not at par with our board level synthesizer, they still offer stabilities comparable to other state of the art synthesizers [47].

In Chapter 3, we describe an alternate electronic synthesizer, that does not require the use of an I-Q demodulator, and a high speed and high resolution ADC. This purely digital IC uses a Time to Digital Converter and a digital counter to make an estimate of the beat note's IF frequency and uses that information to lock it to an optical reference. We present a novel algorithm that does the error measurement in the time domain and manages to achieve lock without using a digital divider. This architecture was first tested by bypassing our RF signal and only using a TDC. The results from this experiment were quite encouraging and showed great promise for the fully integrated version. The results from the fully integrated version (locking experiment carried out using integrated TIA, Mixer and PLL) once again failed to reach the highs demonstrated by our board level implementation, but still performed as well as commercial synthesizers. This IC also did not depend on any external RF Low Noise Amplifiers (LNAs) or Frequency Dividers to process the laser's beat note and so, had a substantially lower energy footprint than the board level implementation. The tradeoff made for low power operation, however, was that we were no longer capable of stepping the laser with Hz level precision. The smallest step size of the synthesizer is now restricted to be an integral multiple of either the PLL's step frequency or the TDC's reference frequency.

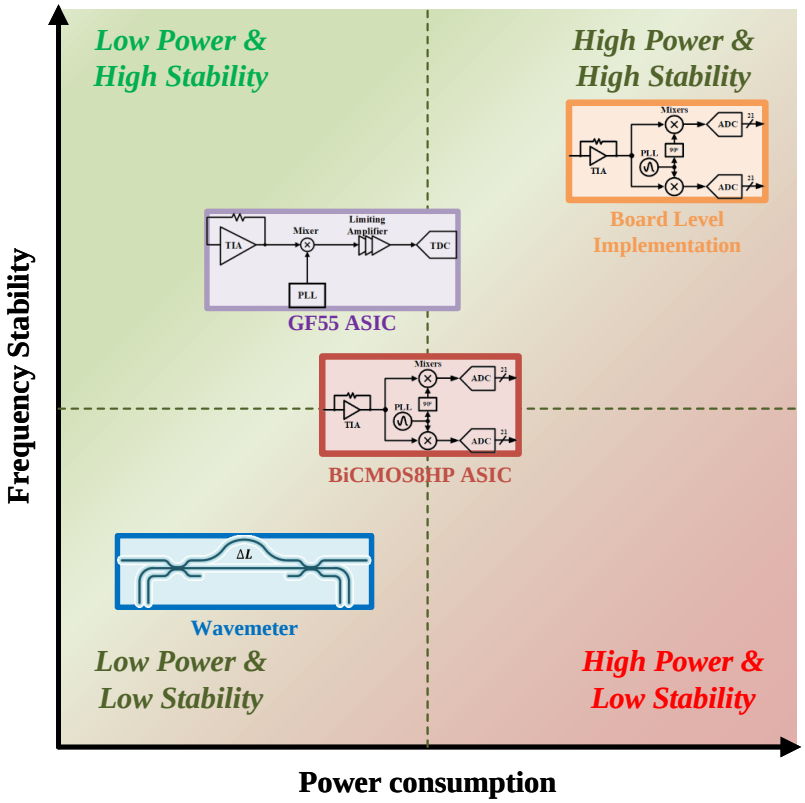


Figure 5.6: A relative comparison between the four synthesizers described using a Frequency Stability vs. Power Consumed plot

Finally in Chapter 4, we address a completely different and novel way of generating THz optical frequencies without the use of complex frequency combs or other high speed electronics. Introduced in [38], these synthesizers generate arbitrary microwave frequency signals by locking two lasers to the wavemeter. As the delay through the wavemeter changes, the two lasers move together with the wavemeter thereby keeping the beat note frequency constant. We improved on this system by implementing the loop digitally and allowing us to now provide a viable path to move the lasers away from each other by multiple wavemeter Free Spectral Ranges (FSRs), thereby providing offsets equal to multiple GHz. Ultimately, the stability of this beat note was limited by the temperature

of the wavemeter. The wavemeter is extremely sensitive to temperature variations, and its sensitivity was found to be the biggest hurdle in achieving stabilities similar to the synthesizers mentioned in the previous chapters.

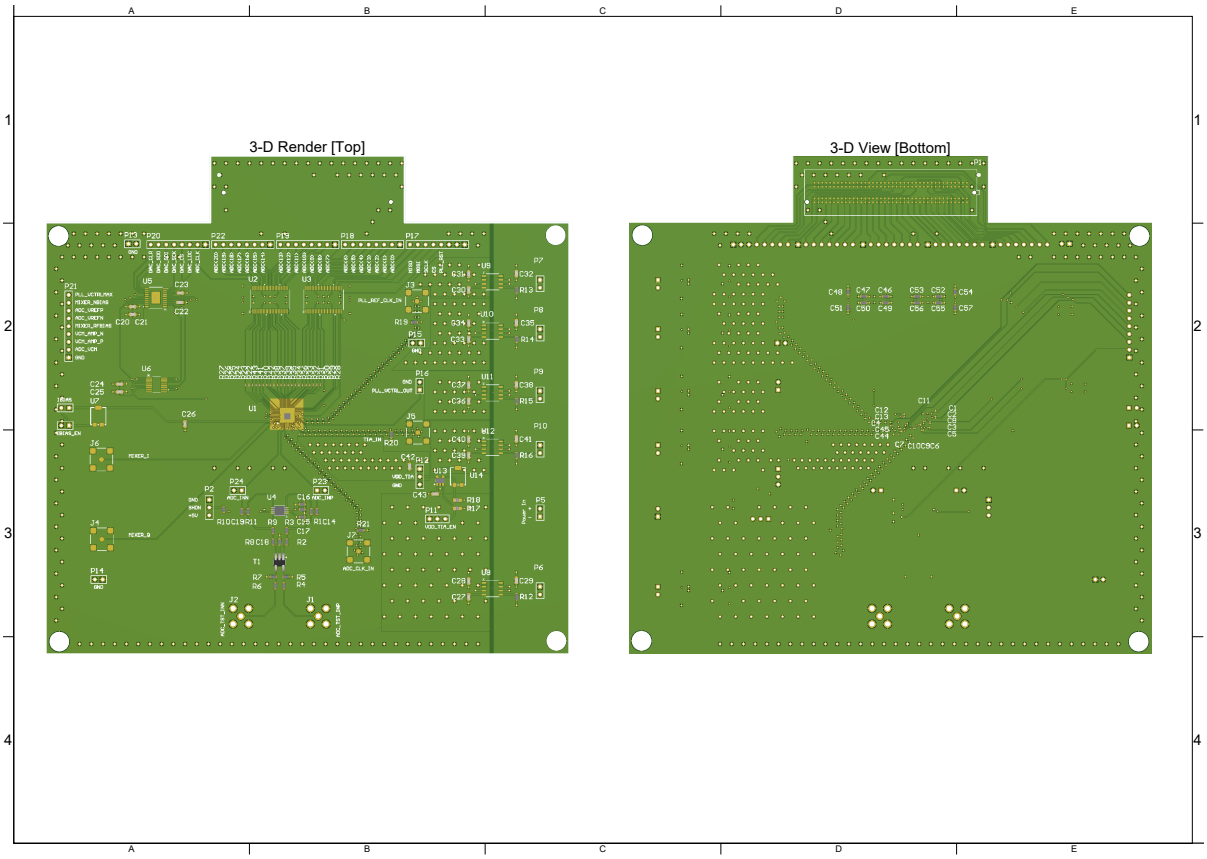
Figure 5.6 is a graphical representation of the tradeoffs between all four implementations discussed in this manuscript. The figure plots the relative stabilities of these implementations as a function of the power they consume. We start off by placing the board level implementation we described in Chapter 2 in the top right corner. Here in this corner, we place the synthesizer that consumes the most power while providing the best results. We next place the ASIC implementation of this synthesizer somewhere in the middle of this chart, as it consumes lower power than the board implementation, but offers substantially lower stabilities. We improve on this ASIC implementation with the IC from Chapter 3, which consumes even lower power while offering an order of magnitude better performance. Finally, we place our wavemeter based synthesizer at the bottom left of the chart, since this implementation consumes lower power than all of the other implementations.

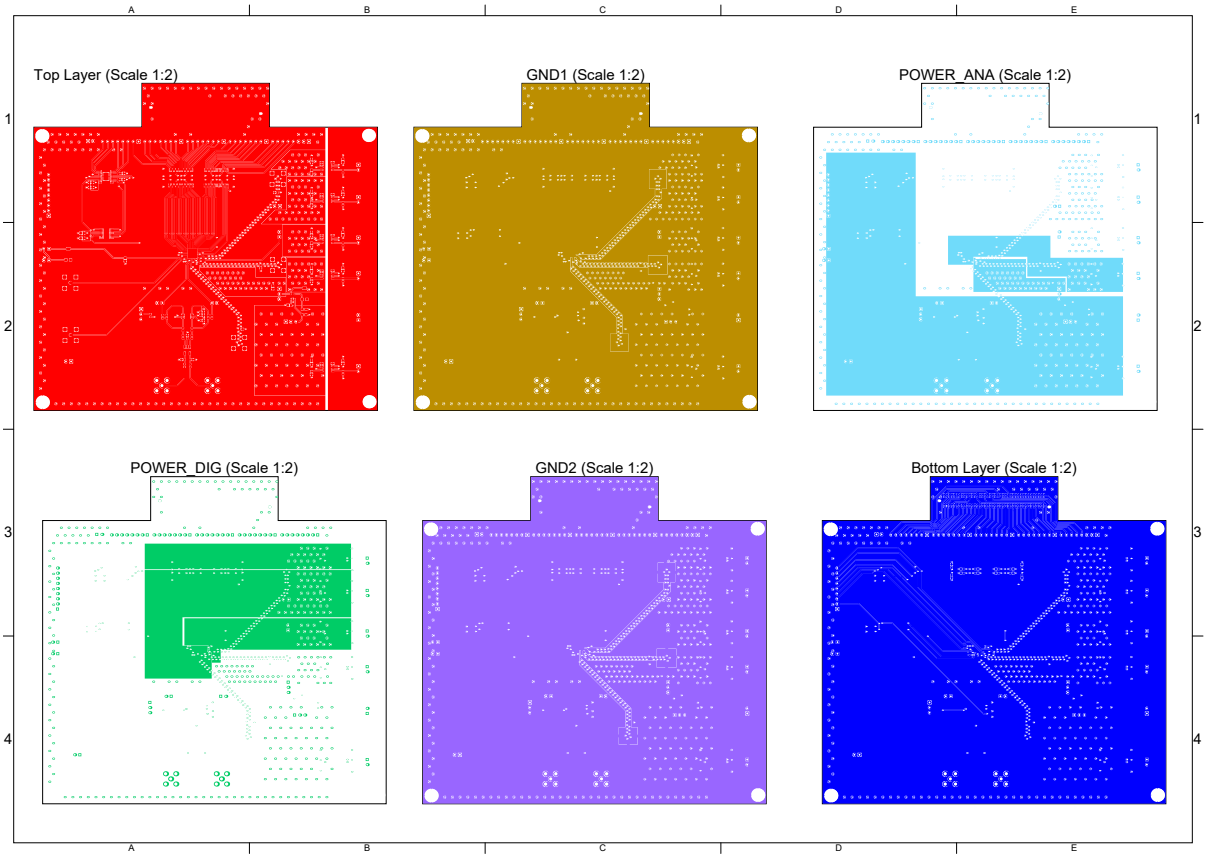
Appendix A

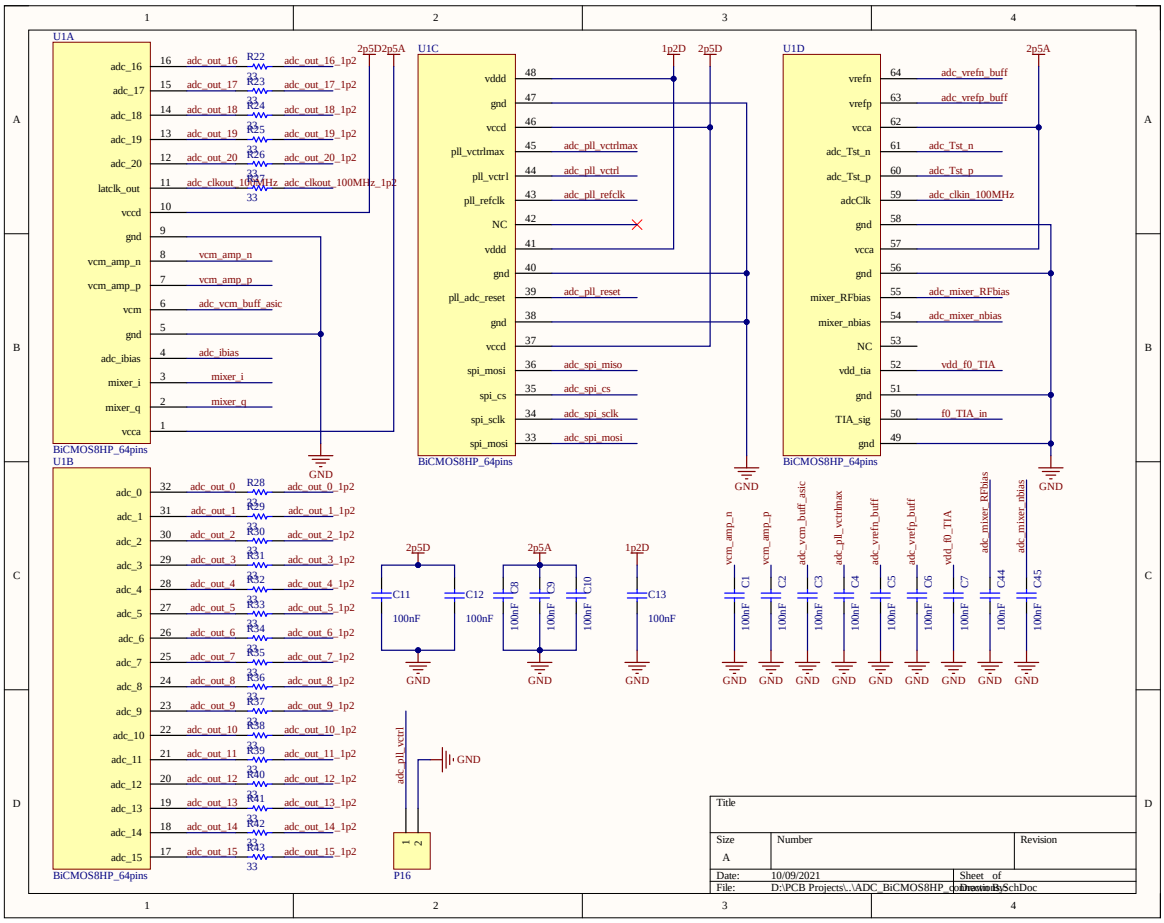
Appendix Title

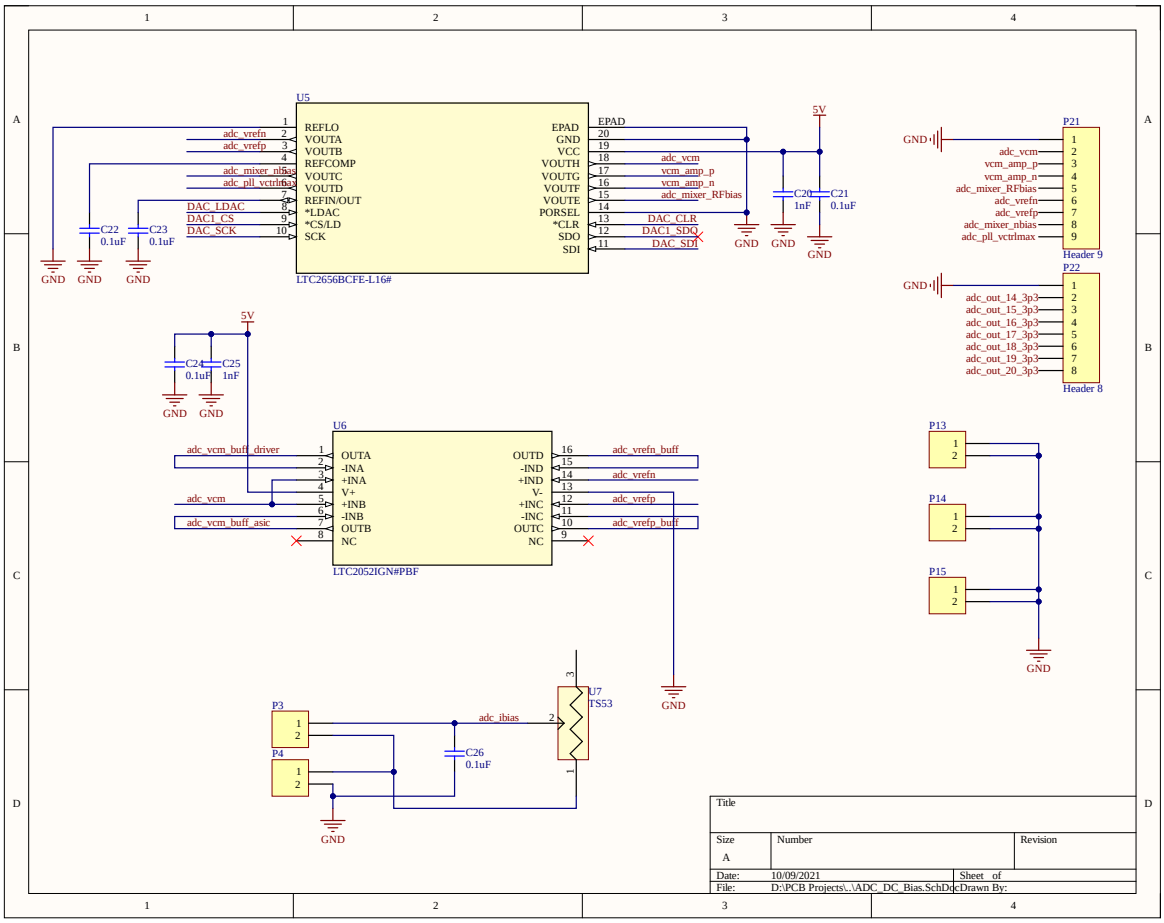
In the appendix, we include all the design files that were used to fabricate the Test Printed Circuit Boards (PCBs). We include the Layer stackup for each of these PCBs as well as their schematics.

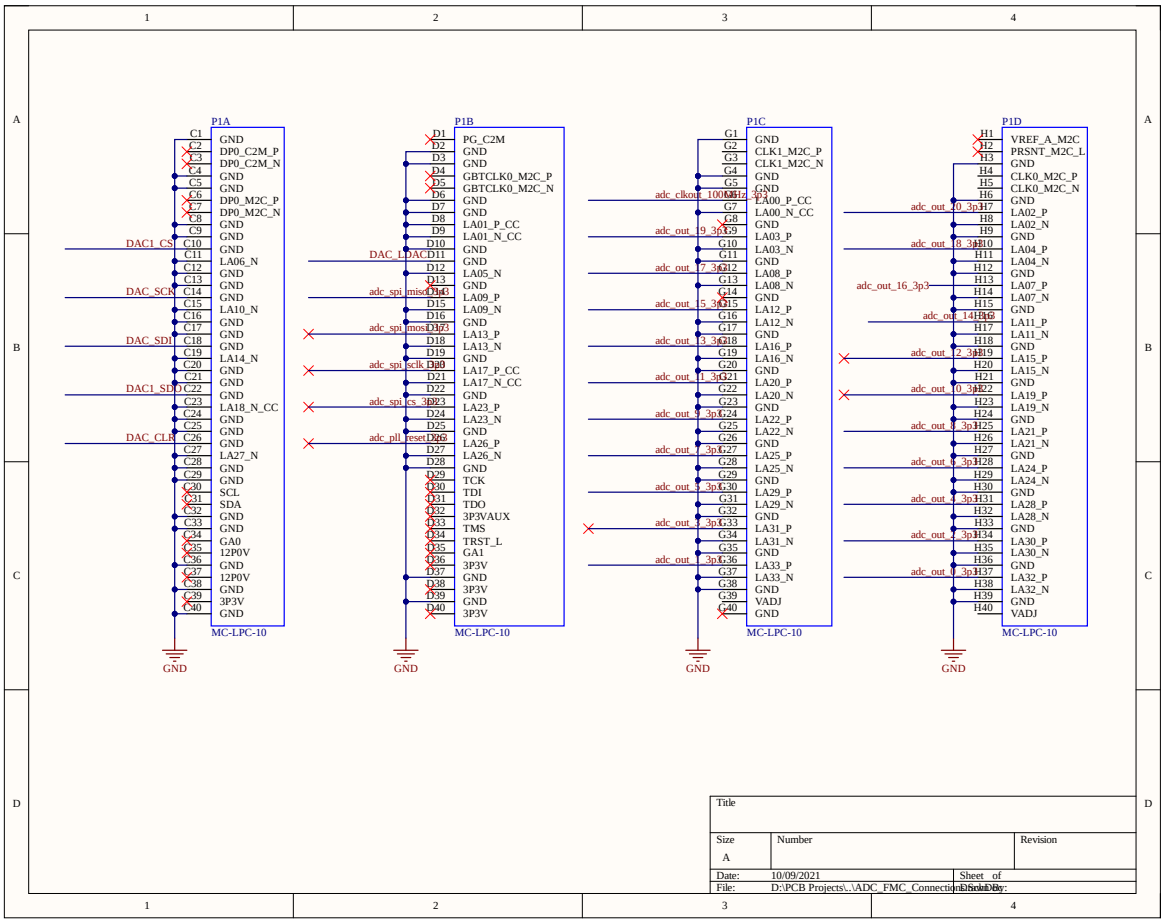
A.1 BiCMOS8HP Test PCB Design Files

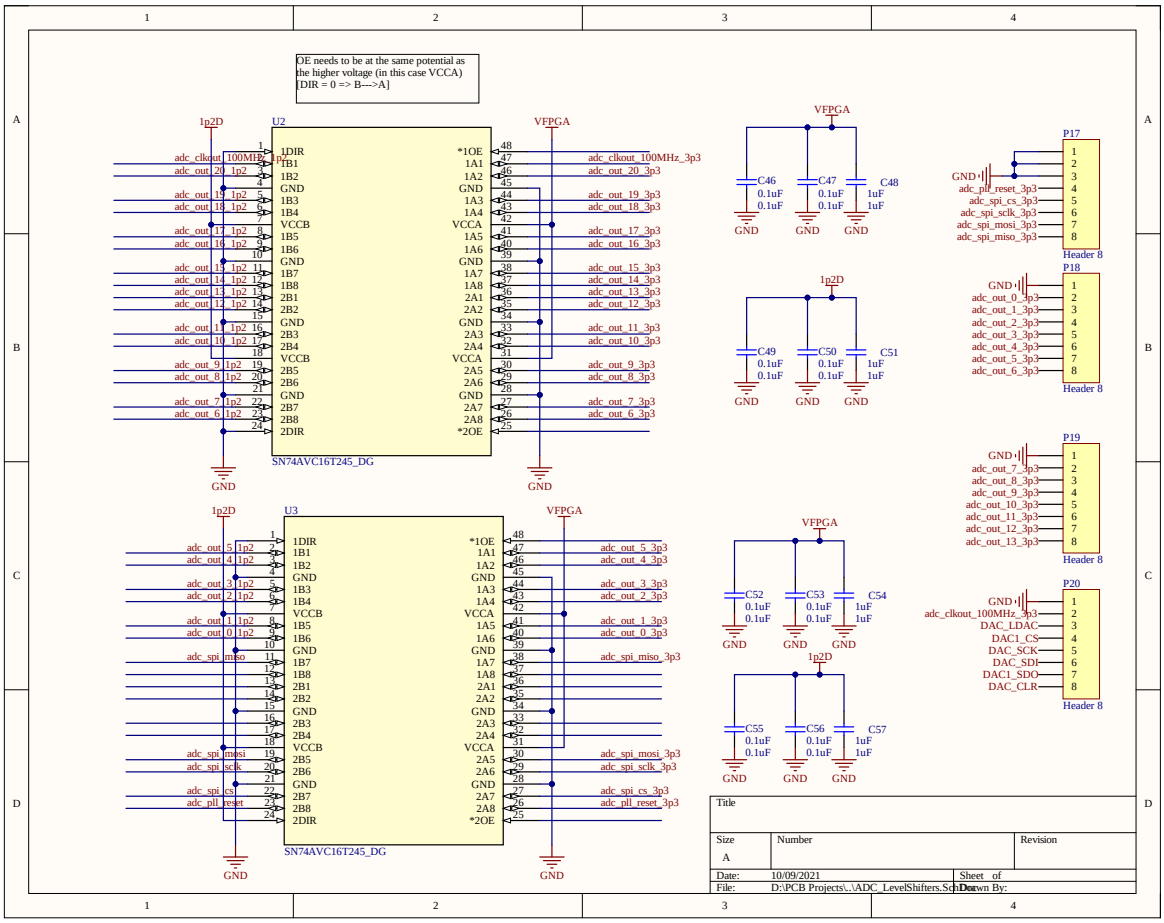


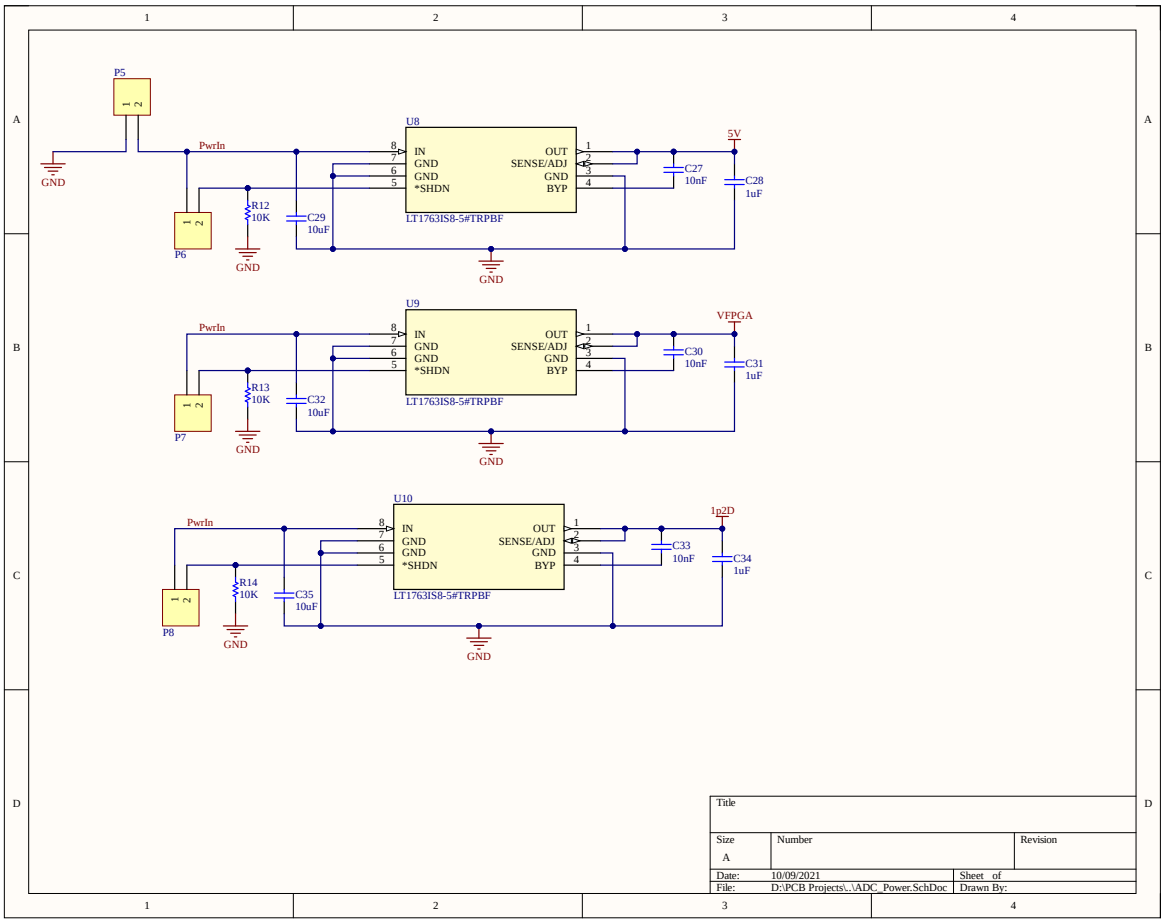

A
B
C
D
E
1
2
3
4
Top Layer (Scale 1:2)
GND1 (Scale 1:2)
POWER_ANA (Scale 1:2)
POWER_DIG (Scale 1:2)
GND2 (Scale 1:2)
Bottom Layer (Scale 1:2)

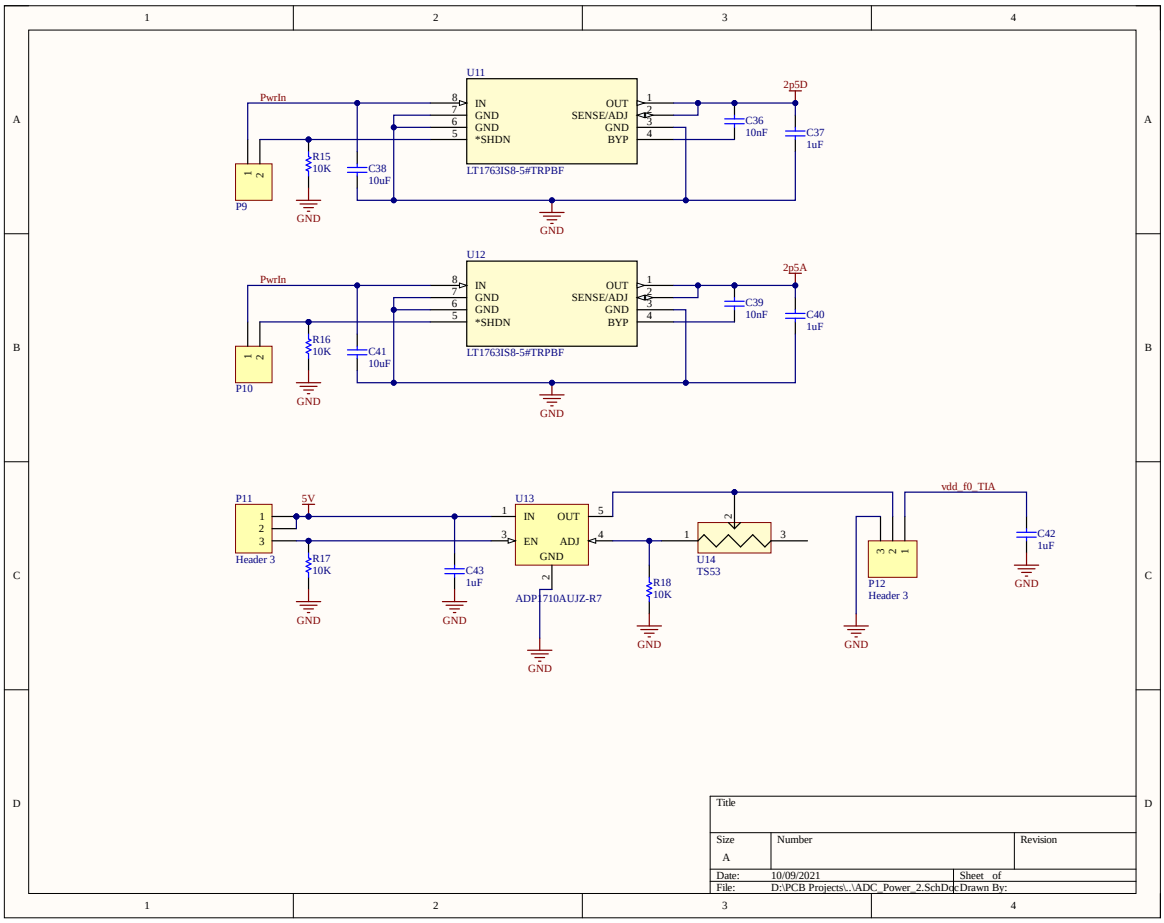

U1A
BiCMOS8HP_64pins
U1B
BiCMOS8HP_64pins
U1C
BiCMOS8HP_64pins
U1D
BiCMOS8HP_64pins
GND
2p5D
2p5A
1p2D
P16
Title
Size
A
Number
Revision
Date: 10/09/2021
File: D:\PCB Projects\..\ADC_BiCMOS8HP_down.SchDoc

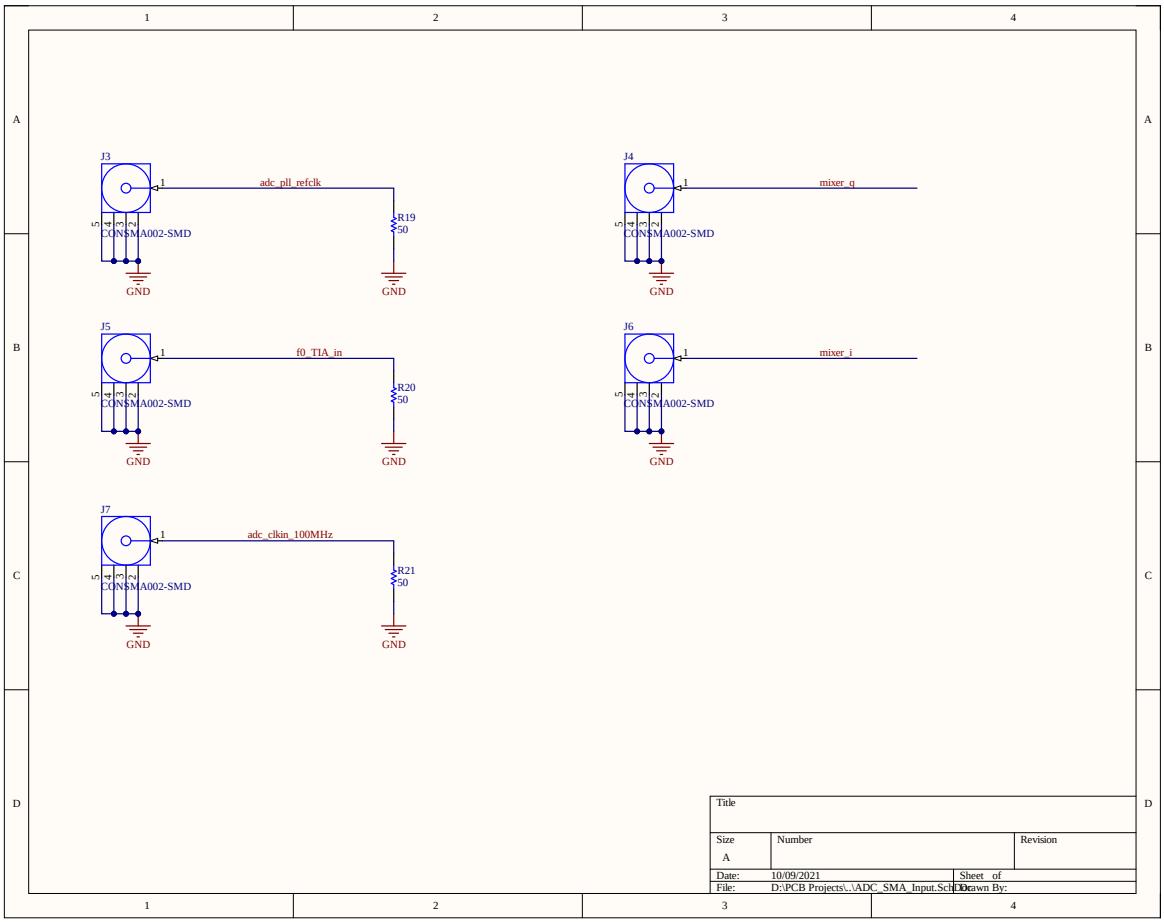

U5
REFLO
VOUTA
VOUTB
REFCOMP
VOUTC
VOUTD
REFIN/OUT
*LDAC
*CS/LD
SCK
EPAD
GND
VCC
VOUTH
VOUTG
VOUTF
VOUTE
PORSEL
*CLR
SDO
SDI
LTC2656BCFE-L16#
adc_vrefn
adc_vrefp
adc_mixer_nbias
adc_pll_vctrlmax
DAC_LDAC
DAC1_CS
DAC_SCK
adc_vcm
vcm_amp_p
vcm_amp_n
adc_mixer_RFbias
DAC_CLR
DAC1_SDO
DAC_SDI
5V
GND
C20 1nF
C21 0.1uF
C22 0.1uF
C23 0.1uF
C24 0.1uF
C25 1nF
C26 0.1uF
P21
Header 9
adc_vcm
vcm_amp_p
vcm_amp_n
adc_mixer_RFbias
adc_vrefn
adc_vrefp
adc_mixer_nbias
adc_pll_vctrlmax
P22
Header 8
adc_out_14_3p3
adc_out_15_3p3
adc_out_16_3p3
adc_out_17_3p3
adc_out_18_3p3
adc_out_19_3p3
adc_out_20_3p3
U6
OUTA
-INA
+INA
V+
+INB
-INB
OUTB
NC
OUTD
-IND
+IND
V-
+INC
-INC
OUTC
NC
LTC2052IGN#PBF
adc_vcm_buff_driver
adc_vcm
adc_vcm_buff_asic
adc_vrefn_buff
adc_vrefn
adc_vrefp
adc_vrefp_buff
P13
P14
P15
P3
P4
adc_ibias
U7
TS53
Title
Size
A
Number
Revision
Date: 10/09/2021
Sheet of
File: D:\PCB Projects\..\ADC_DC_Bias.SchDoc
Drawn By:

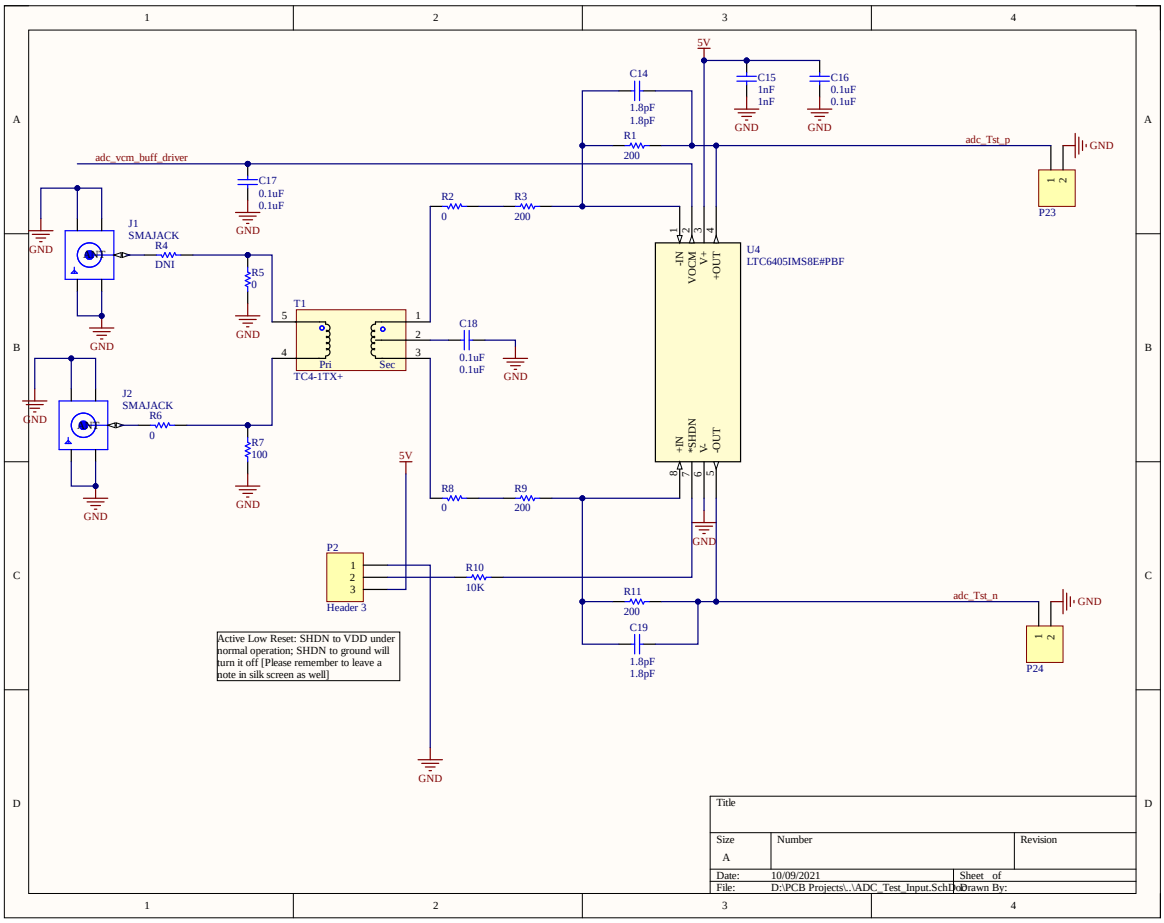

P1A
P1B
P1C
P1D
MC-LPC-10
GND
DAC1_CS
DAC_SCK
DAC_SDI
DAC1_SDO
DAC_CLR
DAC_LDAC
adc_spi_misc_3p3
adc_spi_mosi_3p3
adc_spi_sclk_3p3
adc_spi_cs_3p3
adc_pll_reset_3p3
adc_clkout_100MHz
adc_out_19_3p3
adc_out_17_3p3
adc_out_15_3p3
adc_out_13_3p3
adc_out_11_3p3
adc_out_9_3p3
adc_out_7_3p3
adc_out_5_3p3
adc_out_3_3p3
adc_out_1_3p3
adc_out_20_3p3
adc_out_18_3p3
adc_out_16_3p3
adc_out_14_3p3
adc_out_12_3p3
adc_out_10_3p3
adc_out_8_3p3
adc_out_6_3p3
adc_out_4_3p3
adc_out_2_3p3
adc_out_0_3p3
Title
Size
A
Number
Revision
Date: 10/09/2021
Sheet of
File: D:\PCB Projects\..\ADC_FMC_Connection.SchDoc
Drawn By:


OE needs to be at the same potential as the higher voltage (in this case VCCA)
[DIR = 0 => B--->A]
U2
U3
SN74AVC16T245_DG
VFPGA
1p2D
GND
C46 C47 C48 C49 C50 C51 C52 C53 C54 C55 C56 C57
0.1uF
1uF
P17
P18
P19
P20
Header 8
adc_clkout_100MHz_3p3
adc_out_20_3p3
adc_out_19_3p3
adc_out_18_3p3
adc_out_17_3p3
adc_out_16_3p3
adc_out_15_3p3
adc_out_14_3p3
adc_out_13_3p3
adc_out_12_3p3
adc_out_11_3p3
adc_out_10_3p3
adc_out_9_3p3
adc_out_8_3p3
adc_out_7_3p3
adc_out_6_3p3
adc_out_5_3p3
adc_out_4_3p3
adc_out_3_3p3
adc_out_2_3p3
adc_out_1_3p3
adc_out_0_3p3
adc_spi_miso_3p3
adc_spi_mosi_3p3
adc_spi_sclk_3p3
adc_spi_cs_3p3
adc_pll_reset_3p3
DAC_LDAC
DAC1_CS
DAC_SCK
DAC_SDI
DAC1_SDO
DAC_CLR
Title
Size
A
Number
Revision
Date: 10/09/2021
Sheet of
File: D:\PCB Projects\..\ADC_LevelShifters.SchDoc
Drawn By:

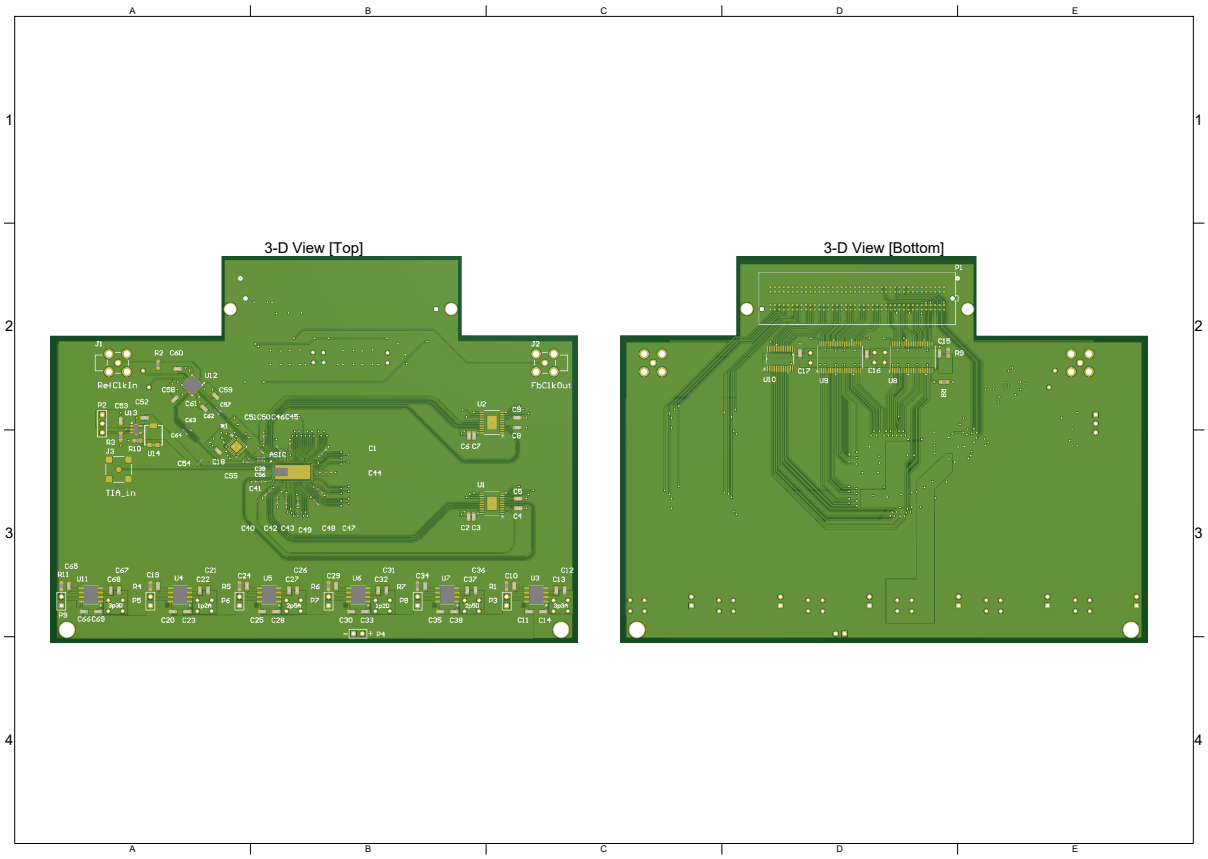

P5
PwrIn
GND
U8
IN
GND
GND
*SHDN
OUT
SENSE/ADJ
GND
BYP
LT1763IS8-5#TRPBF
5V
C27
10nF
C28
1uF
R12
10K
C29
10uF
P6
GND
U9
VFPGA
C30
10nF
C31
1uF
R13
10K
C32
10uF
P7
U10
1p2D
C33
10nF
C34
1uF
R14
10K
C35
10uF
P8
Title
Size
A
Number
Revision
Date: 10/09/2021
Sheet of
File: D:\PCB Projects\..\ADC_Power.SchDoc
Drawn By:

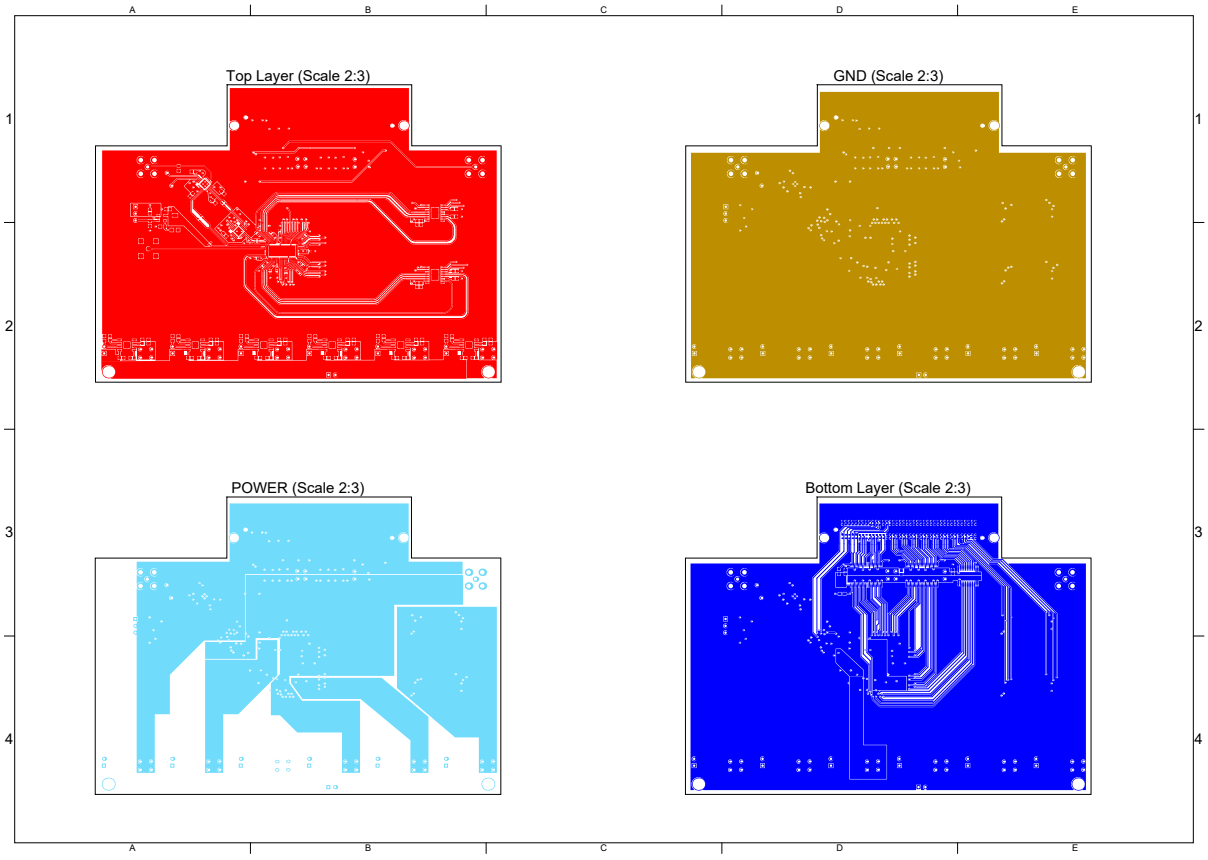

U11
PwrIn
IN
GND
GND
*SHDN
OUT
SENSE/ADJ
GND
BYP
LT1763IS8-5#TRPBF
2p5D
C36
10nF
C37
1uF
R15
10K
C38
10uF
P9
GND
U12
PwrIn
IN
GND
GND
*SHDN
OUT
SENSE/ADJ
GND
BYP
LT1763IS8-5#TRPBF
2p5A
C39
10nF
C40
1uF
R16
10K
C41
10uF
P10
GND
P11
Header 3
5V
R17
10K
C43
1uF
U13
IN
EN
OUT
ADJ
GND
ADP1710AUJZ-R7
R18
10K
U14
TS53
vdd_f0_TIA
P12
Header 3
C42
1uF
Title
Size
A
Number
Revision
Date: 10/09/2021
Sheet of
File: D:\PCB Projects\..\ADC_Power_2.SchDoc
Drawn By:

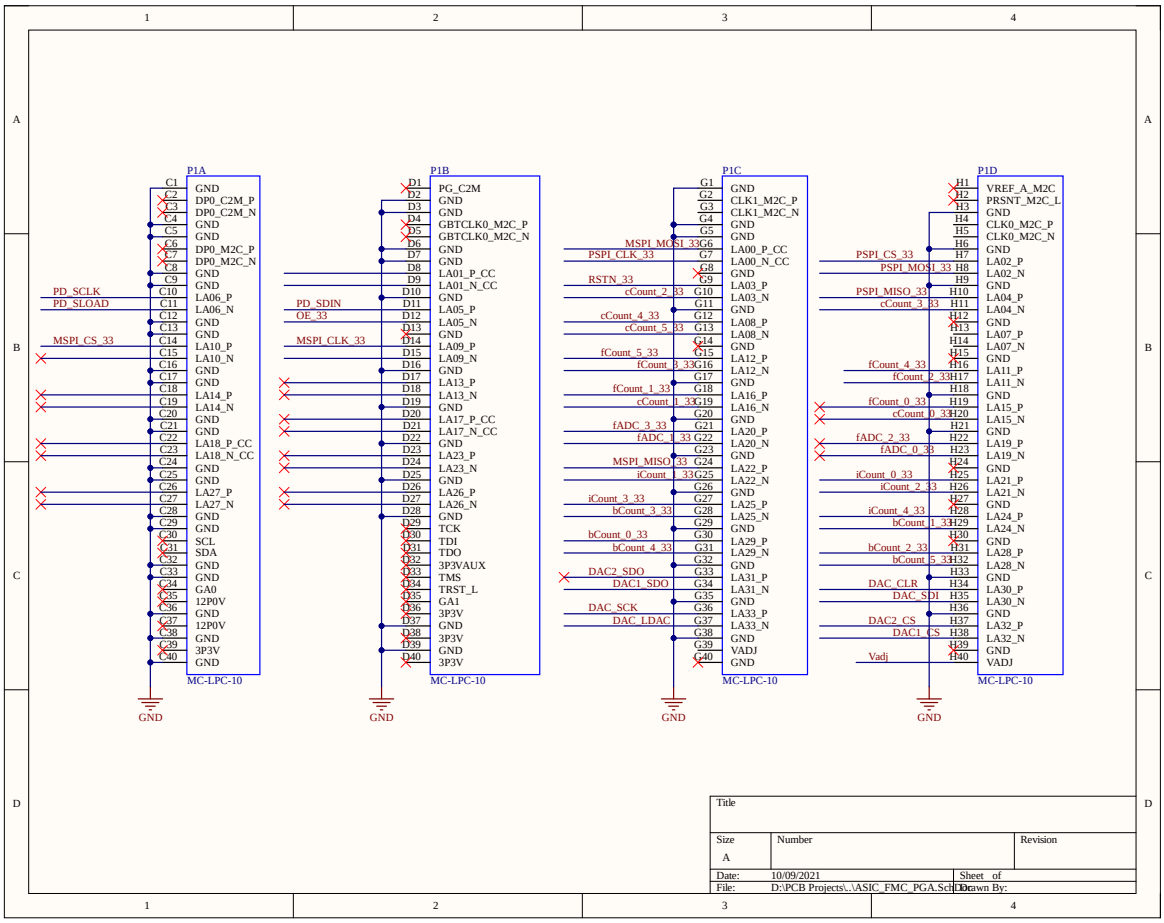

J3
adc_pll_refclk
R19
50
CONSMA002-SMD
GND
J4
mixer_q
J5
f0_TIA_in
R20
J6
mixer_i
J7
adc_clkin_100MHz
R21
Title
Size
A
Number
Revision
Date: 10/09/2021
Sheet of
File: D:\PCB Projects\..\ADC_SMA_Input.SchDoc
Drawn By:

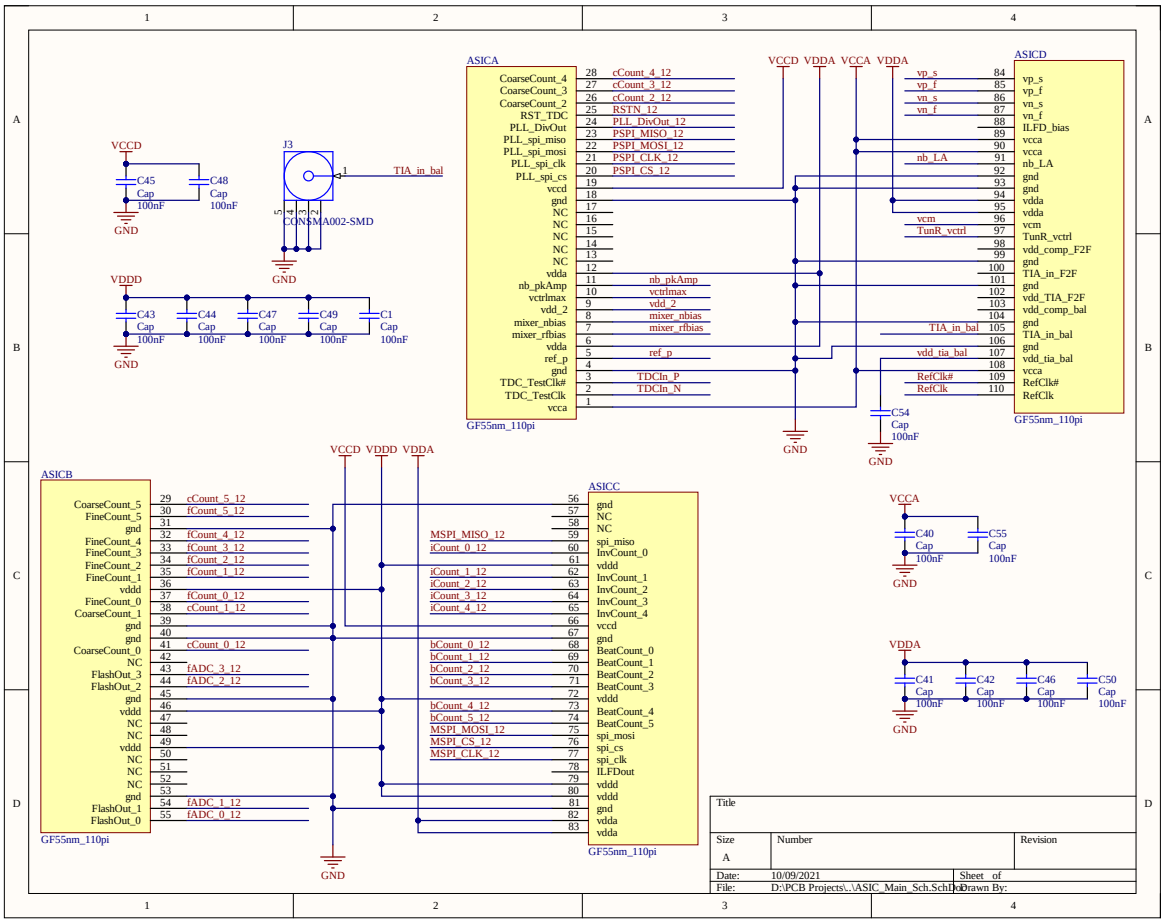

5V
C14
1.8pF
1.8pF
C15
1nF
1nF
C16
0.1uF
0.1uF
GND
R1
200
adc_Tst_p
P23
adc_vcm_buff_driver
C17
0.1uF
0.1uF
R2
0
R3
200
J1
SMAJACK
R4
DNI
R5
0
T1
Pri
Sec
TC4-1TX+
C18
0.1uF
0.1uF
U4
LTC6405IMS8E#PBF
-IN
VOCM
V+
+OUT
+IN
-SHDN
V-
-OUT
J2
SMAJACK
R6
0
R7
100
R8
0
R9
200
P2
Header 3
R10
10K
R11
200
C19
1.8pF
1.8pF
adc_Tst_n
P24
Active Low Reset: SHDN to VDD under normal operation; SHDN to ground will turn it off [Please remember to leave a note in silk screen as well]
Title
Size
A
Number
Revision
Date: 10/09/2021
Sheet of
File: D:\PCB Projects\..\ADC_Test_Input.SchDoc
Drawn By:

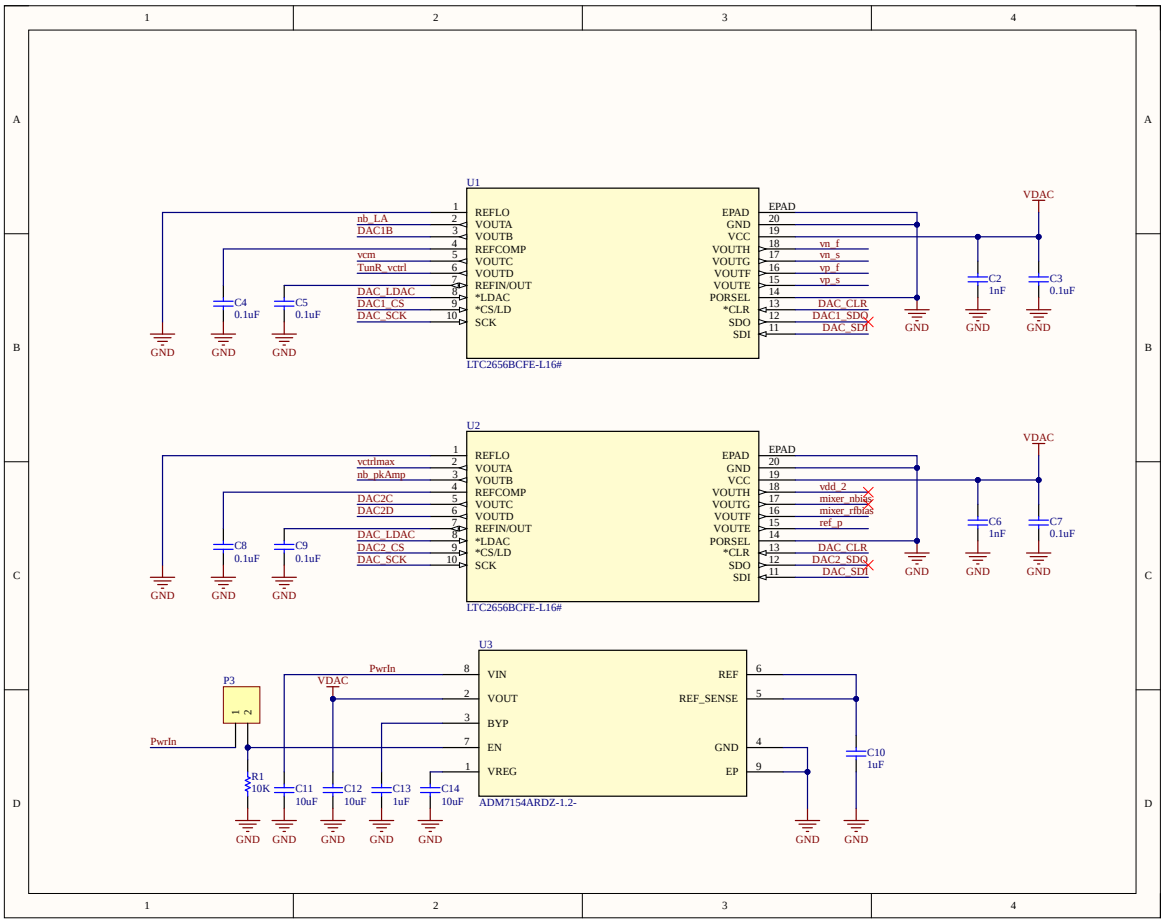
A.2 GF55nm Test PCB Design Files

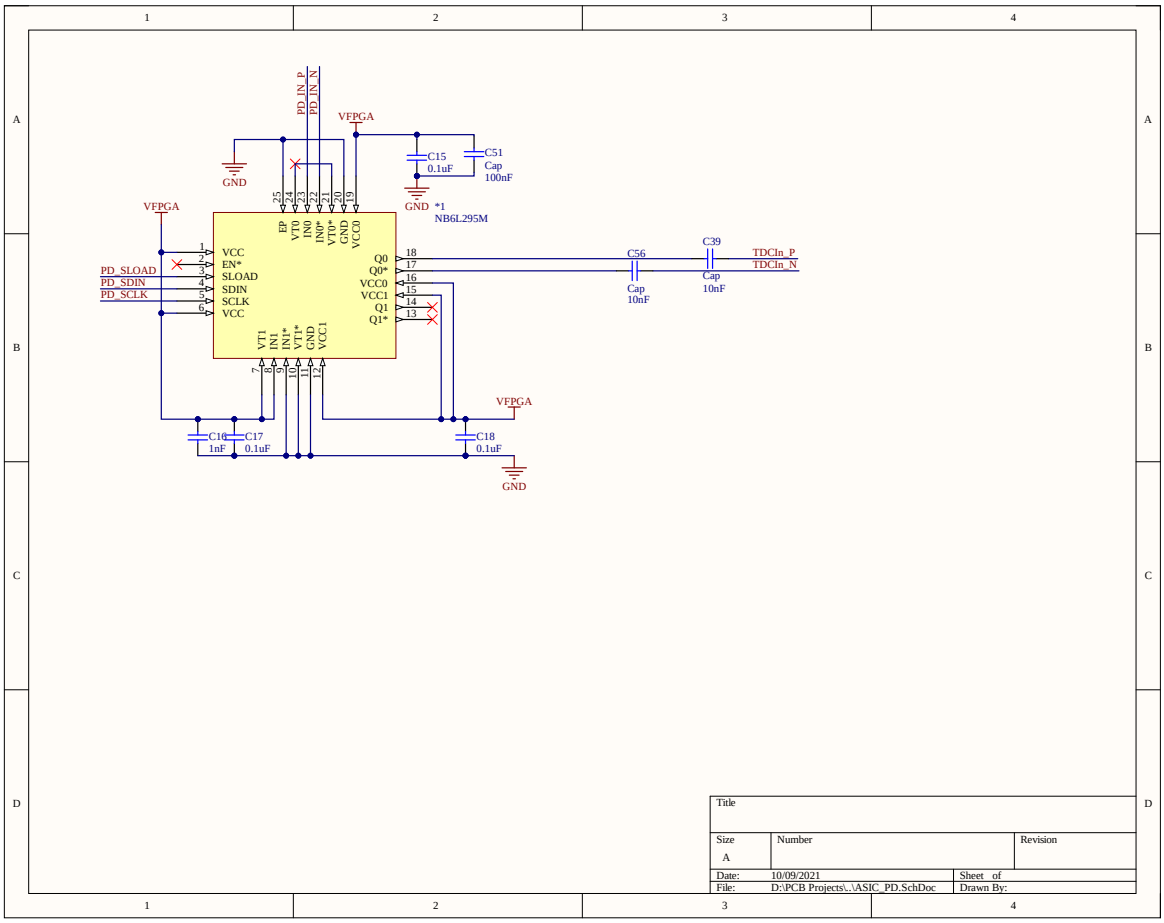


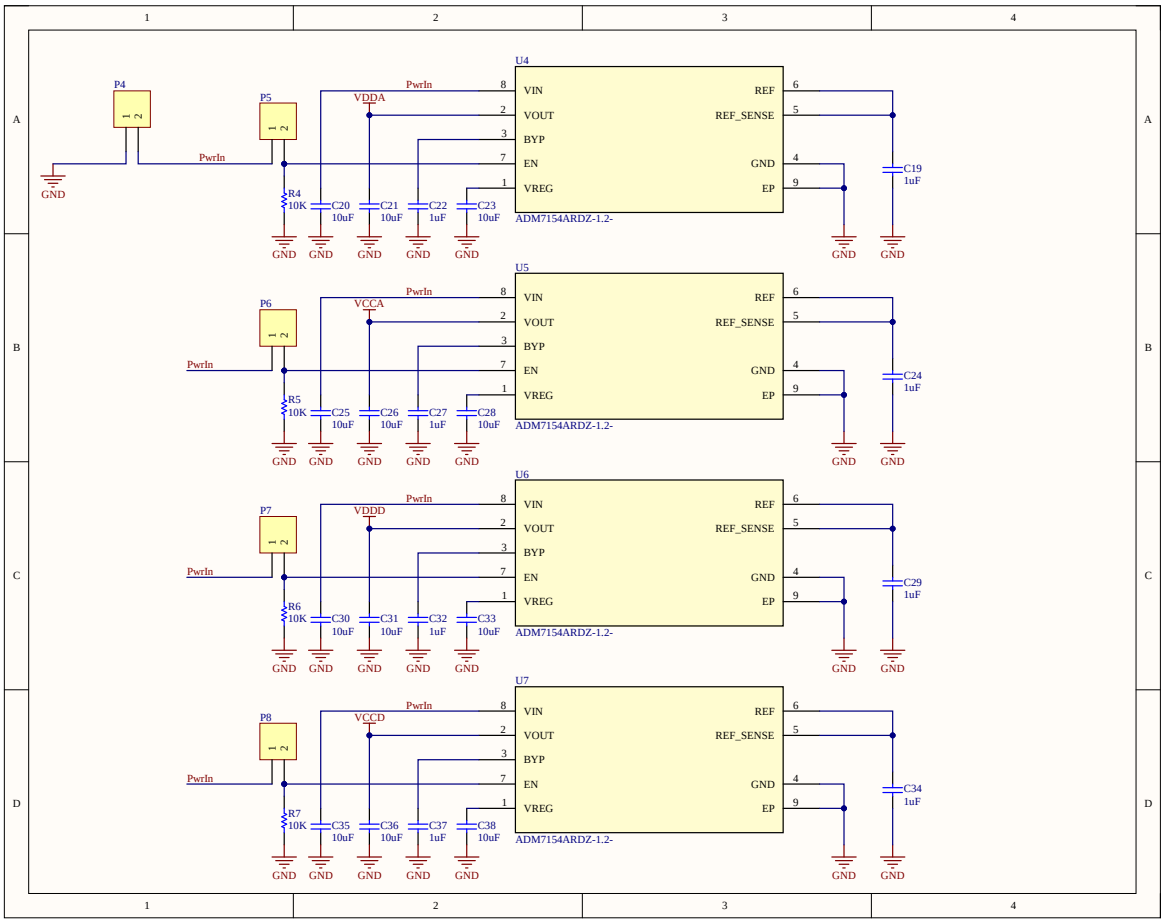

A
B
C
D
E
1
2
3
4
Top Layer (Scale 2:3)
GND (Scale 2:3)
POWER (Scale 2:3)
Bottom Layer (Scale 2:3)

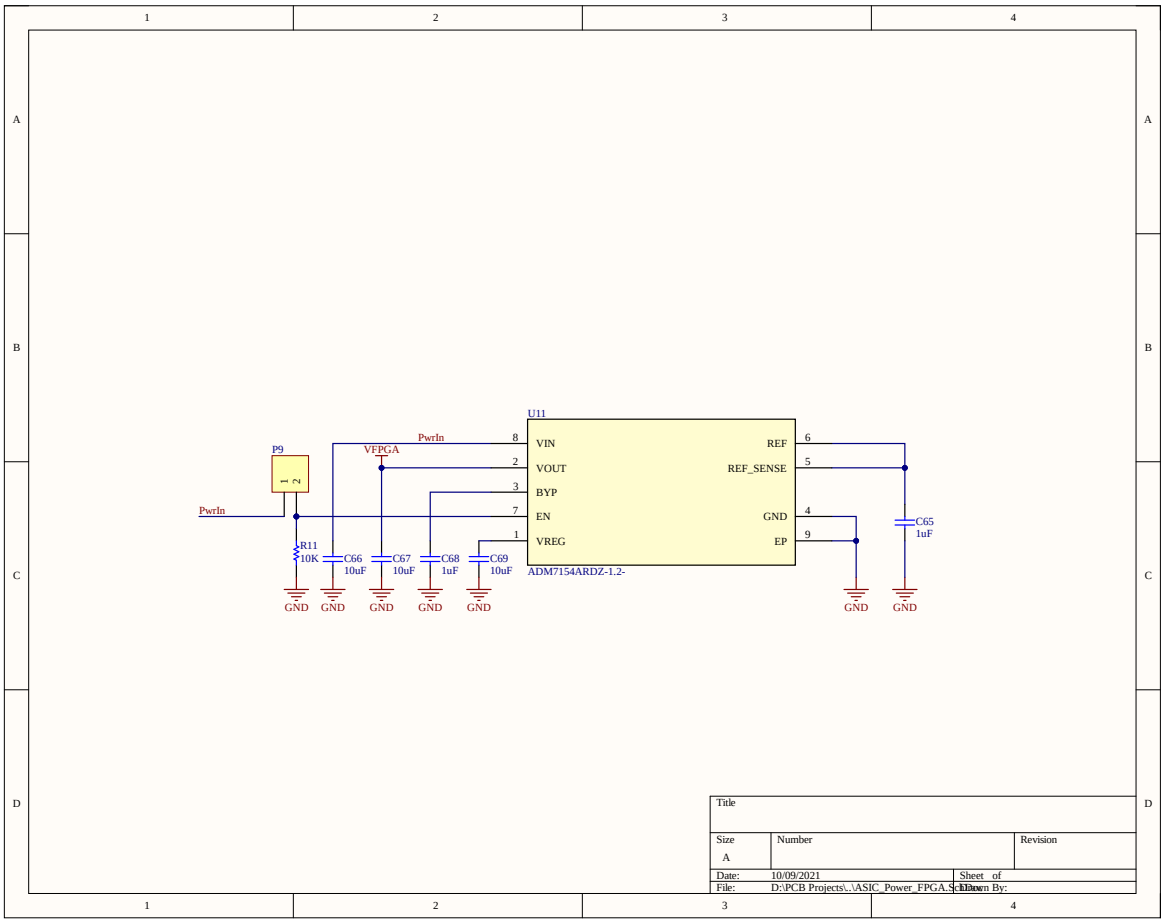

P1A
C1 GND
C2 DP0_C2M_P
C3 DP0_C2M_N
C4 GND
C5 GND
C6 DP0_M2C_P
C7 DP0_M2C_N
C8 GND
C9 GND
C10 LA06_P
C11 LA06_N
C12 GND
C13 GND
C14 LA10_P
C15 LA10_N
C16 GND
C17 GND
C18 LA14_P
C19 LA14_N
C20 GND
C21 GND
C22 LA18_P_CC
C23 LA18_N_CC
C24 GND
C25 GND
C26 LA27_P
C27 LA27_N
C28 GND
C29 GND
C30 SCL
C31 SDA
C32 GND
C33 GND
C34 GA0
C35 12P0V
C36 GND
C37 12P0V
C38 GND
C39 3P3V
C40 GND
MC-LPC-10
PD_SCLK
PD_SLOAD
MSPI_CS_33
GND
P1B
D1 PG_C2M
D2 GND
D3 GND
D4 GBTCLK0_M2C_P
D5 GBTCLK0_M2C_N
D6 GND
D7 GND
D8 LA01_P_CC
D9 LA01_N_CC
D10 GND
D11 LA05_P
D12 LA05_N
D13 GND
D14 LA09_P
D15 LA09_N
D16 GND
D17 LA13_P
D18 LA13_N
D19 GND
D20 LA17_P_CC
D21 LA17_N_CC
D22 GND
D23 LA23_P
D24 LA23_N
D25 GND
D26 LA26_P
D27 LA26_N
D28 GND
D29 TCK
D30 TDI
D31 TDO
D32 3P3VAUX
D33 TMS
D34 TRST_L
D35 GA1
D36 3P3V
D37 GND
D38 3P3V
D39 GND
D40 3P3V
MC-LPC-10
PD_SDIN
OE_33
MSPI_CLK_33
GND
P1C
G1 GND
G2 CLK1_M2C_P
G3 CLK1_M2C_N
G4 GND
G5 GND
G6 LA00_P_CC
G7 LA00_N_CC
G8 GND
G9 LA03_P
G10 LA03_N
G11 GND
G12 LA08_P
G13 LA08_N
G14 GND
G15 LA12_P
G16 LA12_N
G17 GND
G18 LA16_P
G19 LA16_N
G20 GND
G21 LA20_P
G22 LA20_N
G23 GND
G24 LA22_P
G25 LA22_N
G26 GND
G27 LA25_P
G28 LA25_N
G29 GND
G30 LA29_P
G31 LA29_N
G32 GND
G33 LA31_P
G34 LA31_N
G35 GND
G36 LA33_P
G37 LA33_N
G38 GND
G39 VADJ
G40 GND
MC-LPC-10
MSPI_MOSI_33
PSPI_CLK_33
RSTN_33
cCount_2_33
cCount_4_33
cCount_5_33
fCount_5_33
fCount_3_33
fCount_1_33
cCount_1_33
fADC_3_33
fADC_1_33
MSPI_MISO_33
iCount_1_33
iCount_3_33
bCount_3_33
bCount_0_33
bCount_4_33
DAC2_SDO
DAC1_SDO
DAC_SCK
DAC_LDAC
GND
P1D
H1 VREF_A_M2C
H2 PRSNT_M2C_L
H3 GND
H4 CLK0_M2C_P
H5 CLK0_M2C_N
H6 GND
H7 LA02_P
H8 LA02_N
H9 GND
H10 LA04_P
H11 LA04_N
H12 GND
H13 LA07_P
H14 LA07_N
H15 GND
H16 LA11_P
H17 LA11_N
H18 GND
H19 LA15_P
H20 LA15_N
H21 GND
H22 LA19_P
H23 LA19_N
H24 GND
H25 LA21_P
H26 LA21_N
H27 GND
H28 LA24_P
H29 LA24_N
H30 GND
H31 LA28_P
H32 LA28_N
H33 GND
H34 LA30_P
H35 LA30_N
H36 GND
H37 LA32_P
H38 LA32_N
H39 GND
H40 VADJ
MC-LPC-10
PSPI_CS_33
PSPI_MOSI_33
PSPI_MISO_33
cCount_3_33
fCount_4_33
fCount_2_33
fCount_0_33
cCount_0_33
fADC_2_33
fADC_0_33
iCount_0_33
iCount_2_33
iCount_4_33
bCount_1_33
bCount_2_33
bCount_5_33
DAC_CLR
DAC_SDI
DAC2_CS
DAC1_CS
Vadj
GND
Title
Size A
Number
Revision
Date: 10/09/2021
Sheet of
File: D:\PCB Projects\..\ASIC_FMC_PGA.SchDoc
Drawn By:

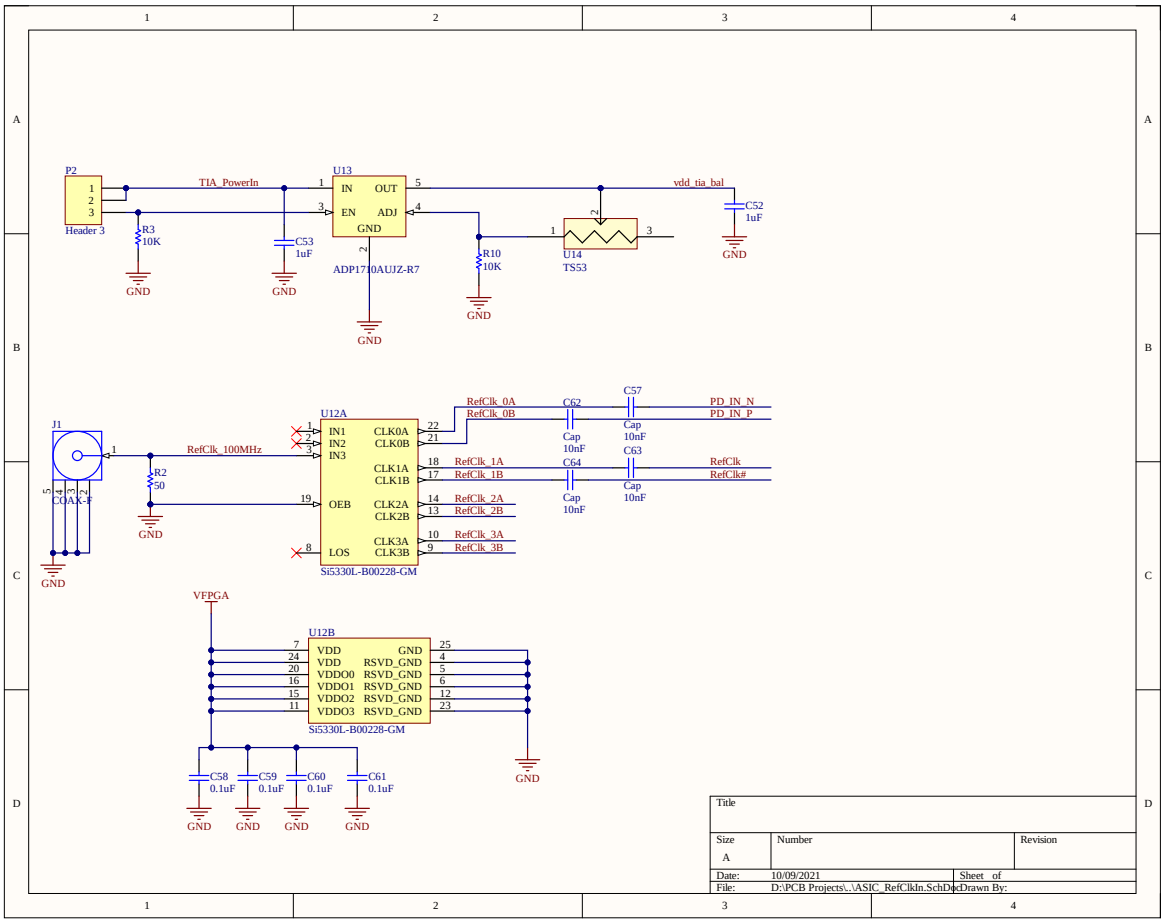

ASICA
ASICB
ASICC
ASICD
GF55nm_110pi
VCCD
VDDD
VDDA
VCCA
GND
J3
TIA_in_bal
CONSMA002-SMD
C45 Cap 100nF
C48 Cap 100nF
C43 Cap 100nF
C44 Cap 100nF
C47 Cap 100nF
C49 Cap 100nF
C1 Cap 100nF
C54 Cap 100nF
C40 Cap 100nF
C55 Cap 100nF
C41 Cap 100nF
C42 Cap 100nF
C46 Cap 100nF
C50 Cap 100nF
Title
Size A
Number
Revision
Date: 10/09/2021
Sheet of
File: D:\PCB Projects\..\ASIC_Main_Sch.SchDoc
Drawn By:

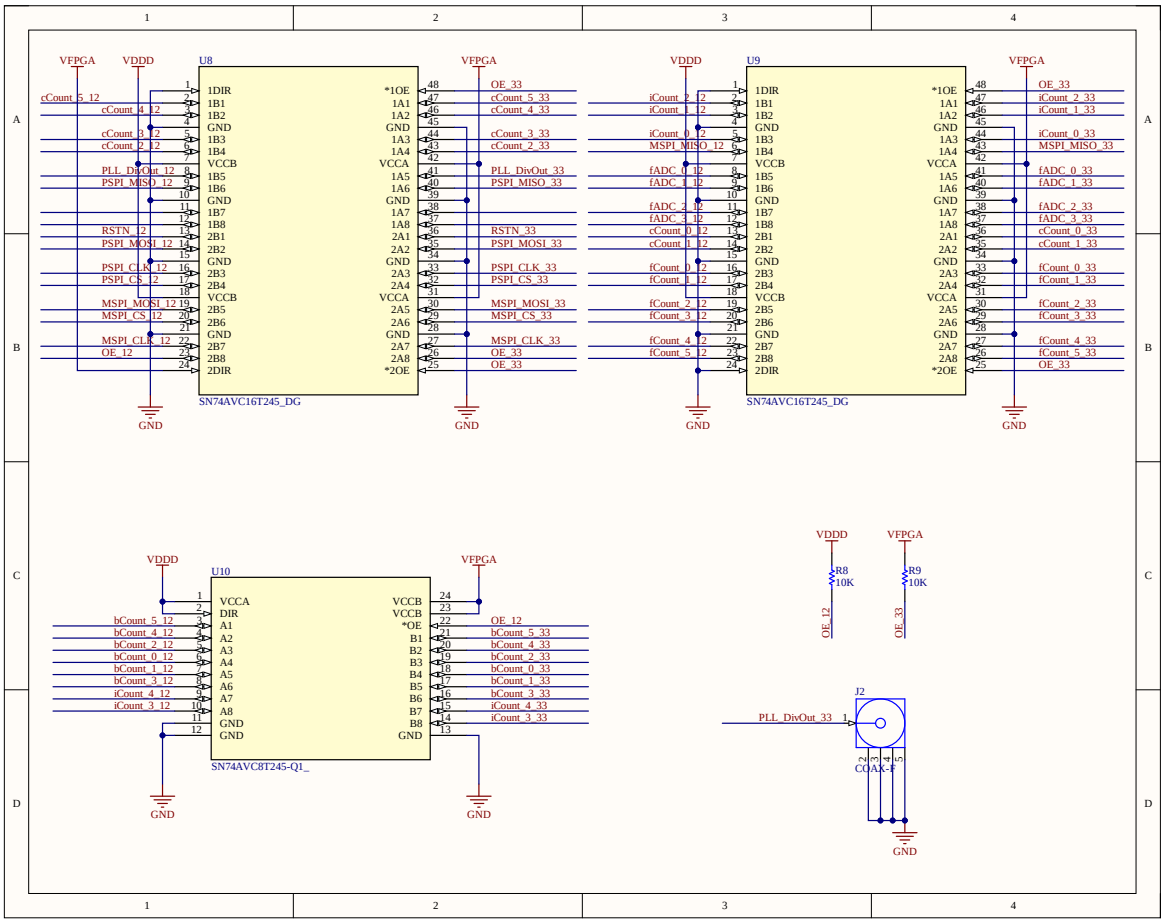

U1
REFLO
VOUTA
VOUTB
REFCOMP
VOUTC
VOUTD
REFIN/OUT
*LDAC
*CS/LD
SCK
EPAD
GND
VCC
VOUTH
VOUTG
VOUTF
VOUTE
PORSEL
*CLR
SDO
SDI
nb_LA
DAC1B
vcm
TunR_vctrl
DAC_LDAC
DAC1_CS
DAC_SCK
vn_f
vn_s
vp_f
vp_s
DAC_CLR
DAC1_SDO
DAC_SDI
VDAC
C2 1nF
C3 0.1uF
C4 0.1uF
C5 0.1uF
LTC2656BCFE-L16#
U2
vctrlmax
nb_pkAmp
DAC2C
DAC2D
DAC_LDAC
DAC2_CS
DAC_SCK
vdd_2
mixer_nbias
mixer_rfbias
ref_p
DAC_CLR
DAC2_SDO
DAC_SDI
C6 1nF
C7 0.1uF
C8 0.1uF
C9 0.1uF
LTC2656BCFE-L16#
U3
VIN
VOUT
BYP
EN
VREG
REF
REF_SENSE
GND
EP
PwrIn
P3
R1 10K
C10 1uF
C11 10uF
C12 10uF
C13 1uF
C14 10uF
ADM7154ARDZ-1.2-


PD_IN_P
PD_IN_N
VFPGA
GND
C15
0.1uF
C51
Cap
100nF
GND
*1
NB6L295M
EP
VT0
IN0
IN0*
VT1*
GND
VCC0
VCC
EN*
SLOAD
SDIN
SCLK
VCC
Q0
Q0*
VCC0
VCC1
Q1
Q1*
VT1
IN1
IN1*
VT1*
GND
VCC1
PD_SLOAD
PD_SDIN
PD_SCLK
C56
Cap
10nF
C39
Cap
10nF
TDCIn_P
TDCIn_N
C16
1nF
C17
0.1uF
C18
0.1uF
Title
Size
A
Number
Revision
Date: 10/09/2021
Sheet of
File: D:\PCB Projects\..\ASIC_PD.SchDoc
Drawn By:

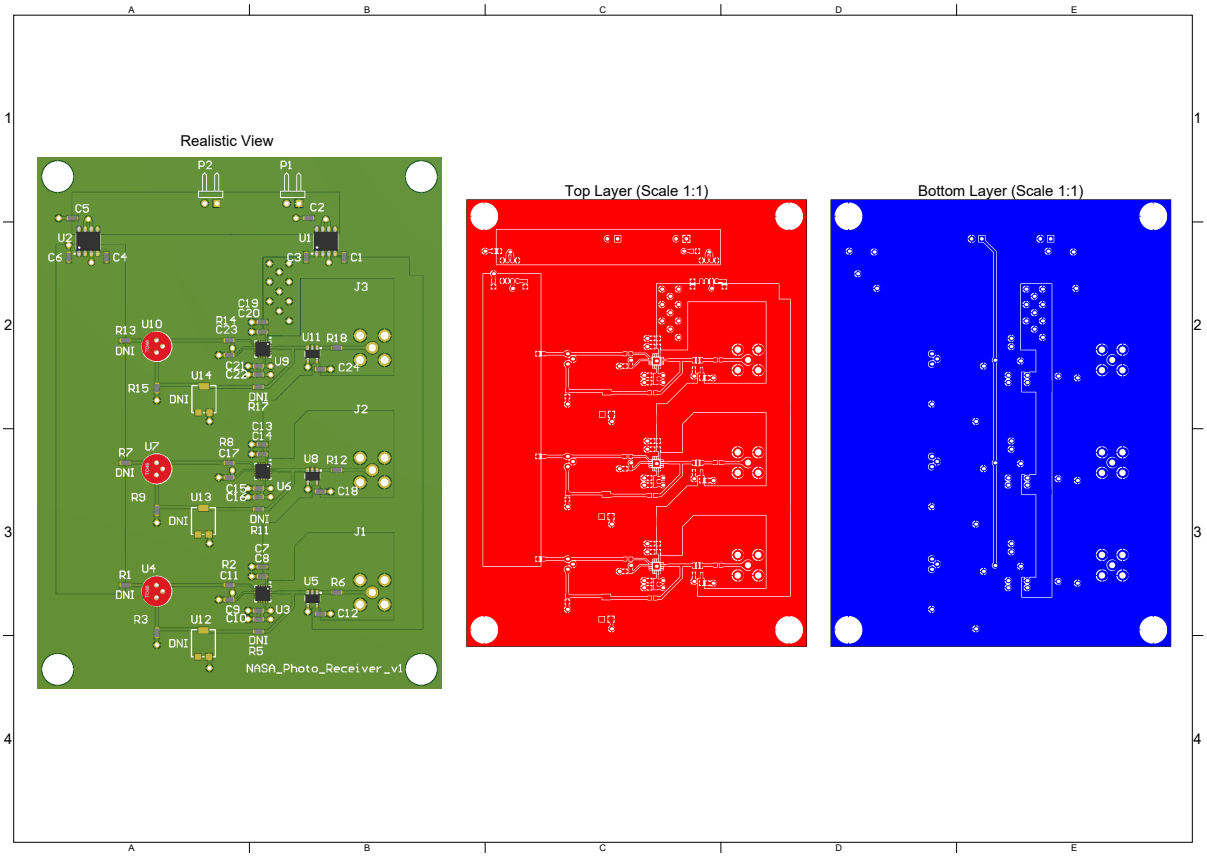

U4
U5
U6
U7
VIN
VOUT
BYP
EN
VREG
REF
REF_SENSE
GND
EP
ADM7154ARDZ-1.2-
PwrIn
VDDA
VCCA
VDDD
VCCD
P4
P5
P6
P7
P8
R4
R5
R6
R7
10K
C19
C20
C21
C22
C23
C24
C25
C26
C27
C28
C29
C30
C31
C32
C33
C34
C35
C36
C37
C38
10uF
1uF

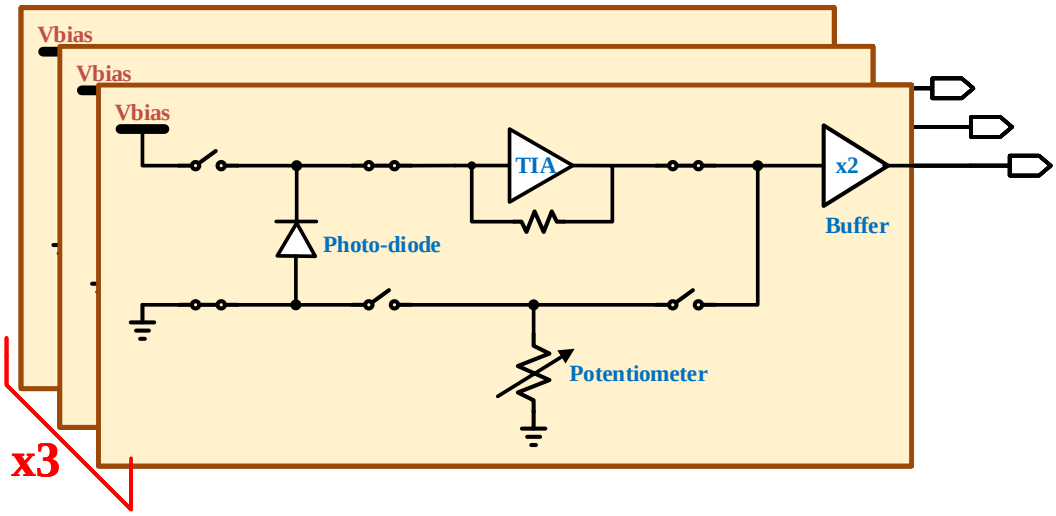

U11
VIN
VOUT
BYP
EN
VREG
REF
REF_SENSE
GND
EP
ADM7154ARDZ-1.2-
P9
PwrIn
VFPGA
R11
10K
C66
10uF
C67
10uF
C68
1uF
C69
10uF
C65
1uF
GND
Title
Size
A
Number
Revision
Date: 10/09/2021
File: D:\PCB Projects\..\ASIC_Power_FPGA.SchDoc
Drawn By:

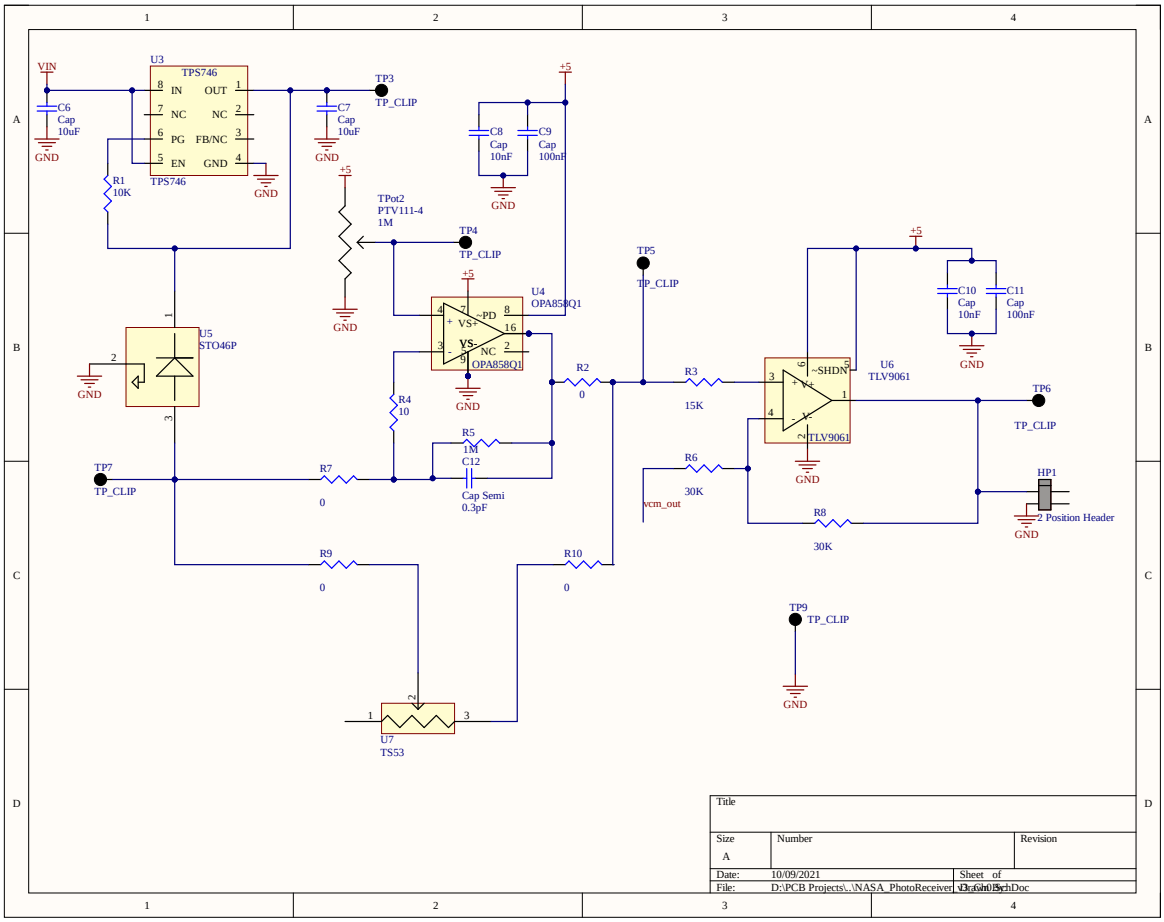

P2
Header 3
TIA_PowerIn
R3
10K
C53
1uF
U13
IN
OUT
EN
ADJ
GND
ADP1710AUJZ-R7
R10
10K
U14
TS53
vdd_tia_bal
C52
1uF
J1
COAX-F
RefClk_100MHz
R2
50
U12A
IN1
IN2
IN3
OEB
LOS
CLK0A
CLK0B
CLK1A
CLK1B
CLK2A
CLK2B
CLK3A
CLK3B
Si5330L-B00228-GM
RefClk_0A
RefClk_0B
RefClk_1A
RefClk_1B
RefClk_2A
RefClk_2B
RefClk_3A
RefClk_3B
C62
C64
C57
C63
Cap
10nF
PD_IN_N
PD_IN_P
RefClk
RefClk#
VFPGA
U12B
VDD
VDDO0
VDDO1
VDDO2
VDDO3
RSVD_GND
C58
C59
C60
C61
0.1uF
Title
Size
A
Number
Revision
Date: 10/09/2021
Sheet of
File: D:\PCB Projects\..\ASIC_RefClkIn.SchDoc
Drawn By:

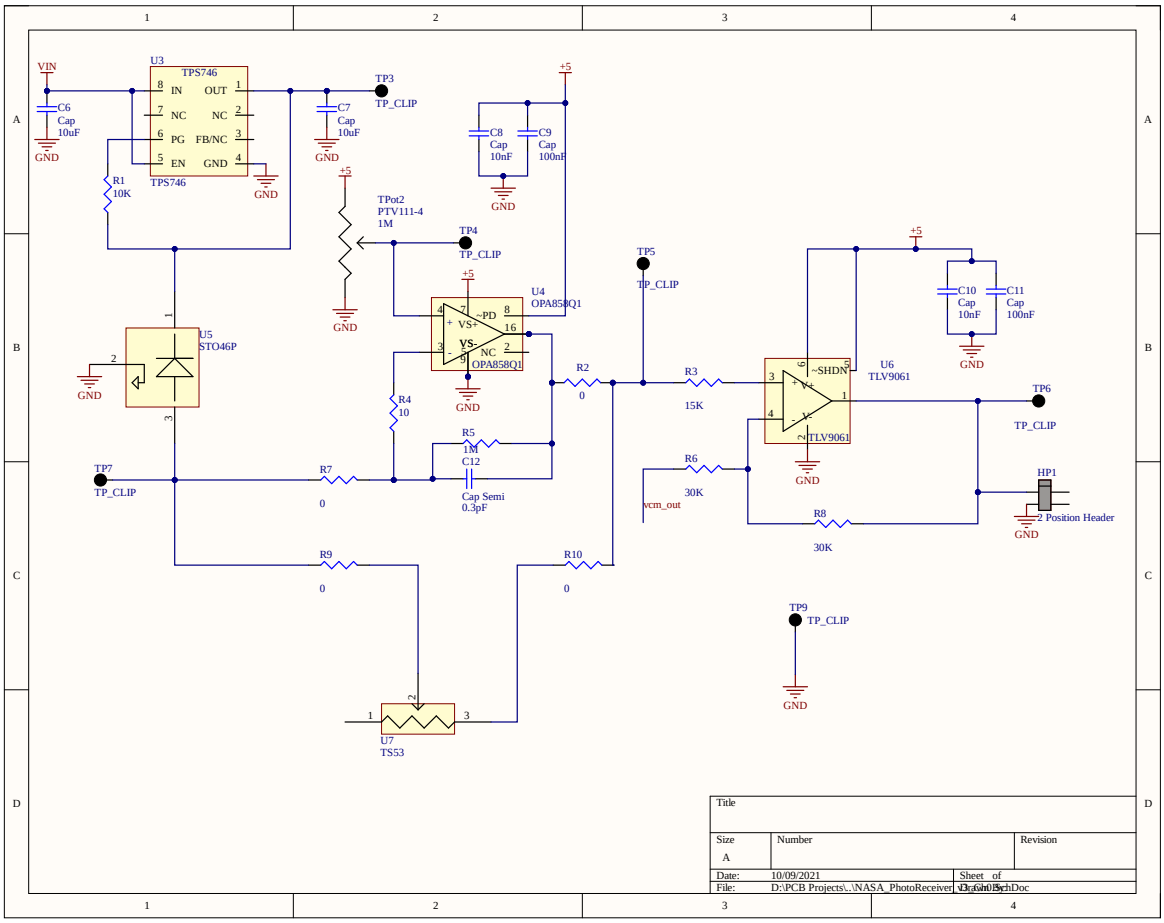

U8
SN74AVC16T245_DG
U9
SN74AVC16T245_DG
U10
SN74AVC8T245-Q1_
R8
10K
R9
10K
J2
COAX-F
VFPGA
VDDD
GND

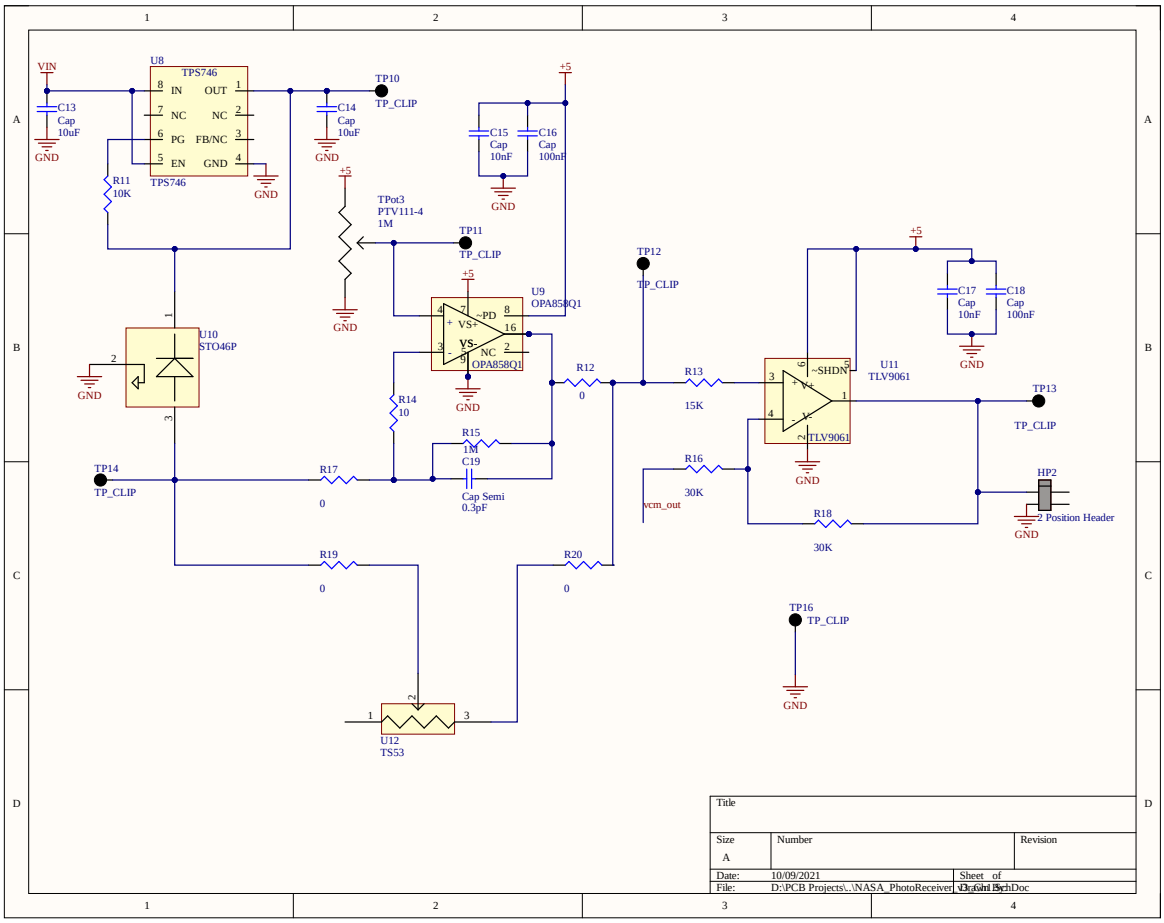
A.3 NASA Photoreceiver version 1 - PCB Design Files

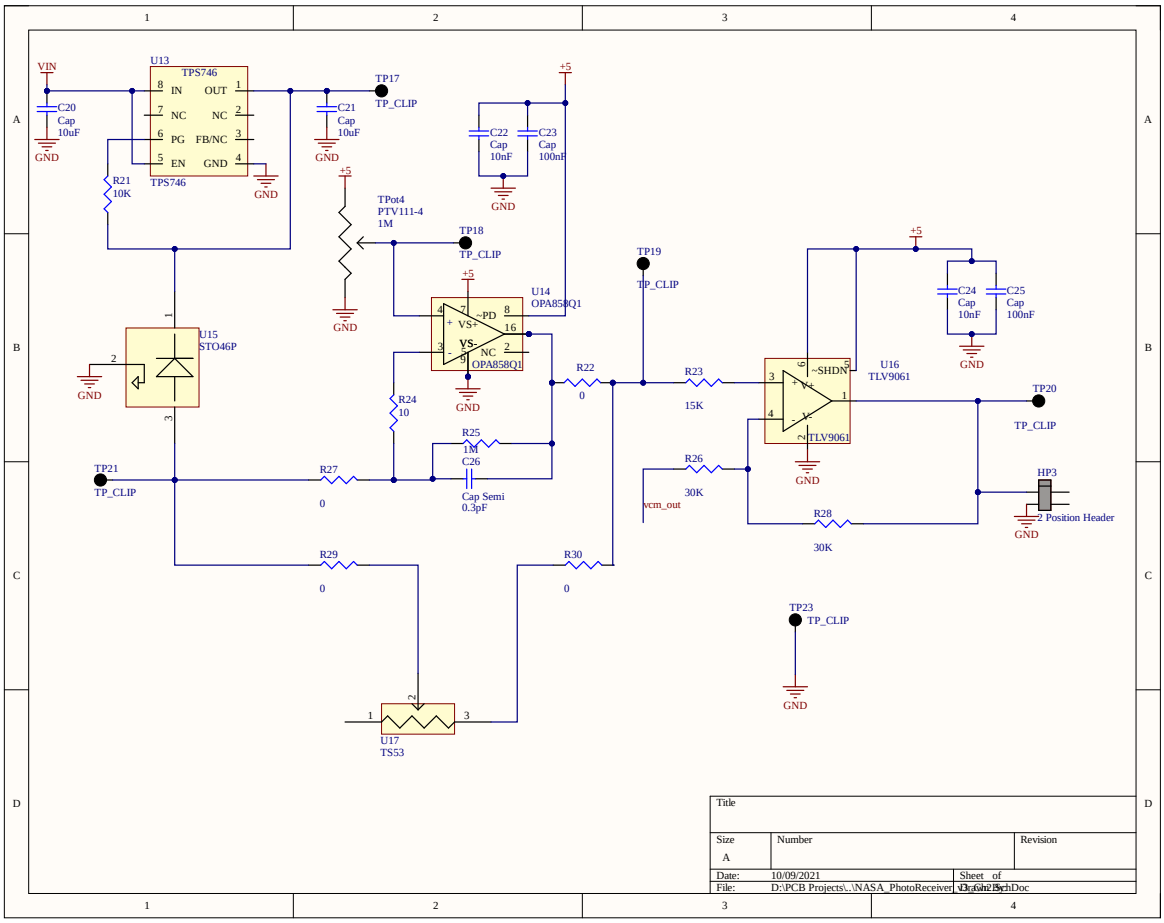


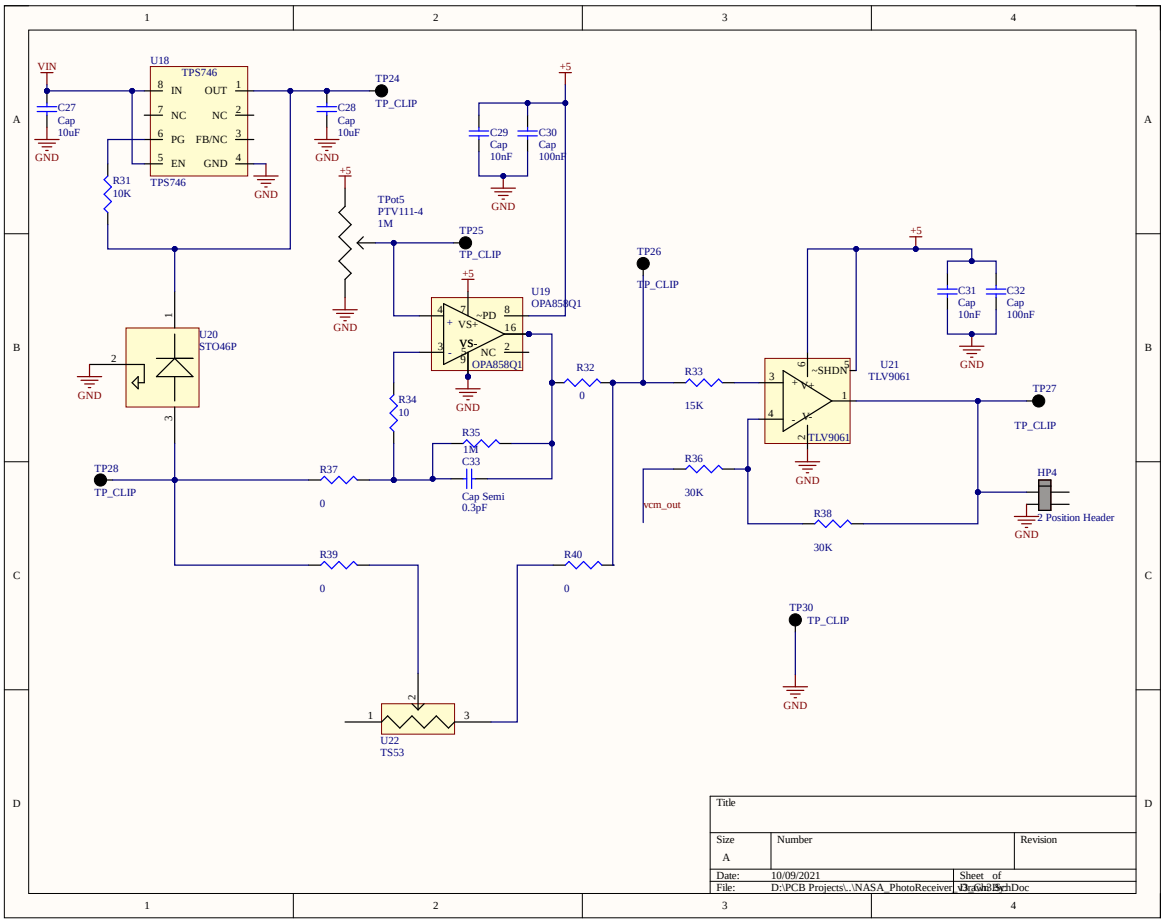

Vbias
Vbias
Vbias
TIA
x2
Buffer
Photo-diode
Potentiometer
x3

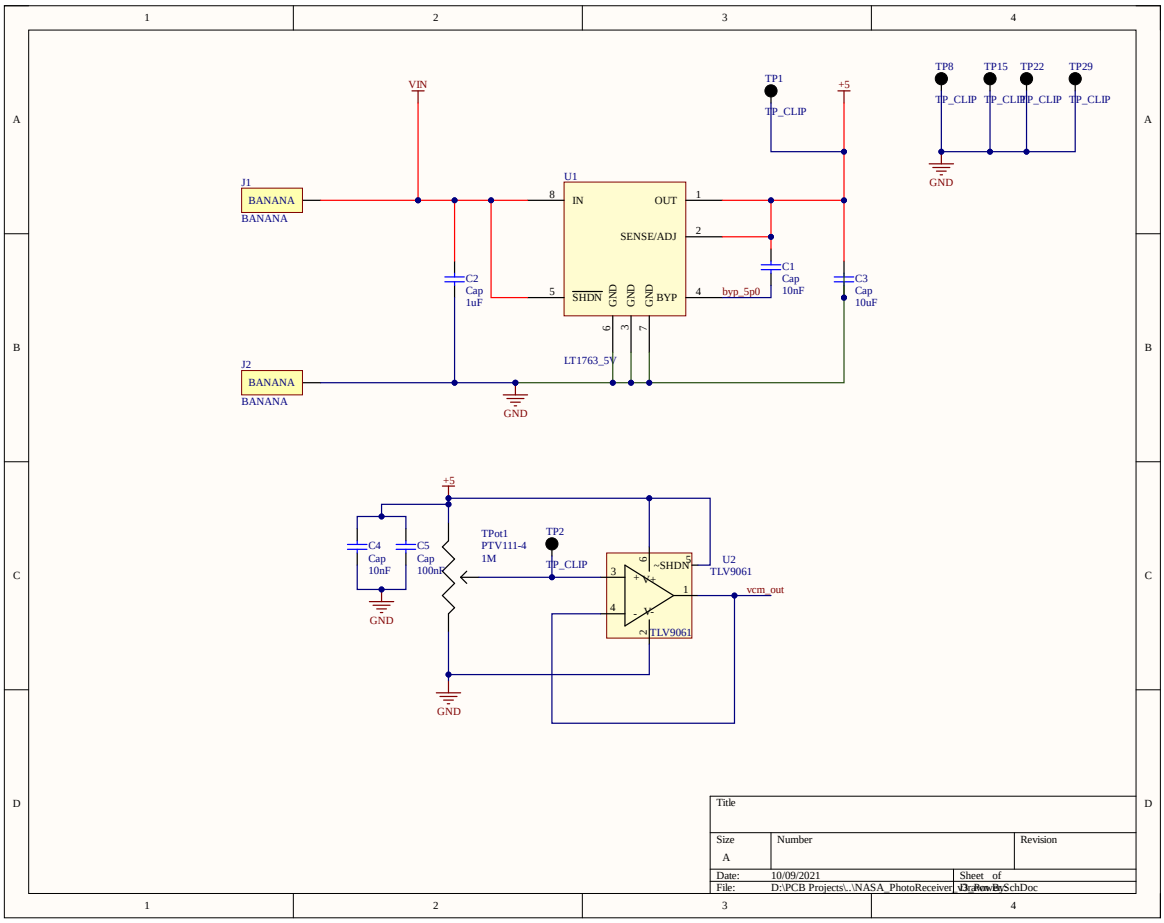

VIN
U3
TPS746
C6
Cap
10uF
GND
R1
10K
TP3
TP_CLIP
C7
C8
Cap
10nF
C9
Cap
100nF
TPot2
PTV111-4
1M
TP4
U4
OPA858Q1
U5
STO46P
R2
0
TP5
R3
15K
U6
TLV9061
C10
C11
100nF
TP6
R4
10
R5
1M
C12
Cap Semi
0.3pF
TP7
R6
30K
vcm_out
HP1
2 Position Header
R8
30K
R7
R9
R10
U7
TS53
TP9
Title
Size
A
Number
Revision
Date:
10/09/2021
File:
D:\PCB Projects\..\NASA_PhotoReceiver_Gain05.SchDoc


VIN
U3
TPS746
IN
OUT
NC
NC
PG
FB/NC
EN
GND
C6
Cap
10uF
C7
Cap
10uF
TP3
TP_CLIP
C8
Cap
10nF
C9
Cap
100nF
+5
R1
10K
TPot2
PTV111-4
1M
TP4
TP_CLIP
U4
OPA858Q1
~PD
VS+
VS-
NC
OPA858Q1
U5
STO46P
R4
10
R5
1M
C12
Cap Semi
0.3pF
TP7
TP_CLIP
R7
0
R2
0
TP5
TP_CLIP
R3
15K
U6
TLV9061
~SHDN
V+
V-
~TLV9061
C10
Cap
10nF
C11
Cap
100nF
TP6
TP_CLIP
R6
30K
vcm_out
R8
30K
HP1
2 Position Header
R9
0
R10
0
U7
TS53
TP9
TP_CLIP
Title
Size
A
Number
Revision
Date: 10/09/2021
Sheet of
File: D:\PCB Projects\..\NASA_PhotoReceiver_Grant05.SchDoc


VIN
U8
TPS746
IN
OUT
NC
NC
PG
FB/NC
EN
GND
C13
Cap
10uF
C14
Cap
10uF
C15
Cap
10nF
C16
Cap
100nF
R11
10K
TP10
TP_CLIP
+5
TPot3
PTV111-4
1M
TP11
TP_CLIP
U9
OPA858Q1
~PD
VS+
VS-
NC
U10
STO46P
R12
0
TP12
TP_CLIP
R13
15K
U11
TLV9061
~SHDN
V+
V-
~TLV9061
C17
Cap
10nF
C18
Cap
100nF
TP13
TP_CLIP
R14
10
R15
1M
C19
Cap Semi
0.3pF
TP14
TP_CLIP
R17
0
R16
30K
vcm_out
R18
30K
HP2
2 Position Header
R19
0
R20
0
TP16
TP_CLIP
U12
TS53
Title
Size
A
Number
Revision
Date: 10/09/2021
File: D:\PCB Projects\..\NASA_PhotoReceiver_Graphic.SchDoc
Sheet of


VIN
U13
TPS746
IN
OUT
NC
NC
PG
FB/NC
EN
GND
C20
Cap
10uF
C21
Cap
10uF
C22
Cap
10nF
C23
Cap
100nF
R21
10K
TP17
TP_CLIP
TPot4
PTV111-4
1M
TP18
TP_CLIP
TP19
TP_CLIP
U14
OPA858Q1
~PD
VS+
VS-
NC
U15
STO46P
R22
0
R23
15K
U16
TLV9061
~SHDN
V+
V-
~TLV9061
C24
Cap
10nF
C25
Cap
100nF
TP20
TP_CLIP
R24
10
R25
1M
C26
Cap Semi
0.3pF
TP21
TP_CLIP
R27
0
R26
30K
vcm_out
HP3
2 Position Header
R28
30K
R29
0
R30
0
TP23
TP_CLIP
U17
TS53
Title
Size
A
Number
Revision
Date: 10/09/2021
File: D:\PCB Projects\..\NASA_PhotoReceiver_Grant2.SchDoc
Sheet of


VIN
U18
TPS746
IN
OUT
NC
NC
PG
FB/NC
EN
GND
C27
Cap
10uF
C28
Cap
10uF
TP24
TP_CLIP
C29
Cap
10nF
C30
Cap
100nF
R31
10K
TPot5
PTV111-4
1M
TP25
TP_CLIP
U19
OPA858Q1
~PD
VS+
VS-
NC
OPA858Q1
U20
STO46P
R32
0
TP26
TP_CLIP
R33
15K
U21
TLV9061
~SHDN
V+
V-
~TLV9061
C31
Cap
10nF
C32
Cap
100nF
TP27
TP_CLIP
R34
10
R35
1M
C33
Cap Semi
0.3pF
TP28
TP_CLIP
R37
0
R36
30K
vcm_out
HP4
2 Position Header
R38
30K
R39
0
R40
0
TP30
TP_CLIP
U22
TS53
Title
Size
A
Number
Revision
Date: 10/09/2021
File: D:\PCB Projects\..\NASA_PhotoReceiver_Grandboard.SchDoc
Sheet of


VIN
TP1
TP_CLIP
+5
TP8
TP15
TP22
TP29
TP_CLIP
GND
J1
BANANA
BANANA
U1
IN
OUT
SENSE/ADJ
SHDN
GND
GND
GND
BYP
byp_5p0
C1
Cap
10nF
C2
Cap
1uF
C3
Cap
10uF
LT1763_5V
J2
BANANA
BANANA
GND
+5
C4
Cap
10nF
C5
Cap
100nF
TPot1
PTV111-4
1M
TP2
TP_CLIP
~SHDN
U2
TLV9061
vcm_out
TLV9061
GND
Title
Size
A
Number
Revision
Date:
10/09/2021
Sheet of
File:
D:\PCB Projects\..\NASA_PhotoReceiver_V3.1.SchDoc
Drawn By:

Bibliography

[1] V. Gerginov, "NIST-F1 Cesium Fountain Atomic Clock." `https://www.nist.gov/pml/time-and-frequency-division/time-realization/primary-standard-nist-f1`, 2009. [Online; accessed 12-Oct-2021].

[2] T. Udem, J. Reichert, R. Holzwarth, and T. W. Hänsch, *Optical frequency metrology*, *Nature* (2002).

[3] A. A. Madej, L. Marmet, and J. E. Bernard, *Rb atomic absorption line reference for single sr+ laser cooling systems*, *Applied Physics B* **67** (1998) 229–234.

[4] T. Udem, J. Reichert, R. Holzwarth, and T. W. Hänsch, *Absolute optical frequency measurement of the cesium d_1 line with a mode-locked laser*, *Phys. Rev. Lett.* **82** (May, 1999) 3568–3571.

[5] D. J. Jones, S. A. Diddams, J. K. Ranka, A. Stentz, R. S. Windeler, J. L. Hall, and S. T. Cundiff, *Carrier-envelope phase control of femtosecond mode-locked lasers and direct optical frequency synthesis*, *Science* **288** (2000), no. 5466 635–639, [https://www.science.org/doi/pdf/10.1126/science.288.5466.635].

[6] F. Keilmann, C. Gohle, and R. Holzwarth, *Time-domain mid-infrared frequency-comb spectrometer*, *Opt. Lett.* **29** (Jul, 2004) 1542–1544.

[7] I. Coddington, N. Newbury, and W. Swann, *Dual-comb spectroscopy*, *Optica* **3** (Apr, 2016) 414–426.

[8] M. J. Thorpe, K. D. Moll, R. J. Jones, B. Safdi, and J. Ye, *Broadband cavity ringdown spectroscopy for sensitive and rapid molecular detection*, *Science* **311** (2006), no. 5767 1595–1599, [https://www.science.org/doi/pdf/10.1126/science.1123921].

[9] P. Trocha, M. Karpov, D. Ganin, M. H. P. Pfeiffer, A. Kordts, S. Wolf, J. Krockenberger, P. Marin-Palomo, C. Weimann, S. Randel, W. Freude, T. J. Kippenberg, and C. Koos, *Ultrafast optical ranging using microresonator soliton frequency combs*, *Science* **359** (2018), no. 6378 887–891, [https://www.science.org/doi/pdf/10.1126/science.aao3924].

[10] W. Guanhao, L. Lei, X. Shilin, L. Guoyuan, C. Zhijian, and Z. Zebin, *Synthetic wavelength interferometry of an optical frequency comb for absolute distance measurement*, *Scientific Reports* (2018).

[11] E. Baumann, F. R. Giorgetta, J.-D. Deschênes, W. C. Swann, I. Coddington, and N. R. Newbury, *Comb-calibrated laser ranging for three-dimensional surface profiling with micrometer-level precision at a distance*, *Opt. Express* **22** (Oct, 2014) 24914–24928.

[12] J. Ye, *Absolute measurement of a long, arbitrary distance to less than an optical fringe*, *Opt. Lett.* **29** (May, 2004) 1153–1155.

[13] W. C. Swann and N. R. Newbury, *Frequency-resolved coherent lidar using a femtosecond fiber laser*, *Opt. Lett.* **31** (Mar, 2006) 826–828.

[14] N. Singh, M. Xin, N. Li, D. Vermeulen, A. Ruocco, E. S. Magden, K. Shtyrkova, P. T. Callahan, E. Ippen, F. X. Kärtner, and M. R. Watts, *Silicon photonics optical frequency synthesizer - spofs*, in *Conference on Lasers and Electro-Optics*, p. ATh4I.2, Optical Society of America, 2019.

[15] T. J. Kippenberg, R. Holzwarth, and S. A. Diddams, *Microresonator-based optical frequency combs*, *Science* **332** (2011), no. 6029 555–559, [https://www.science.org/doi/pdf/10.1126/science.1193968].

[16] V. Brasch, E. Lucas, J. D. Jost, M. Geiselmann, and T. J. Kippenberg, *Self-referenced photonic chip soliton kerr frequency comb*, *Light: Science & Applications* **6** (Jan, 2017) e16202–e16202.

[17] H. chul Park, M. Lu, E. Bloch, T. Reed, Z. Griffith, L. Johansson, L. Coldren, and M. Rodwell, *40gbit/s coherent optical receiver using a costas loop*, *Opt. Express* **20** (Dec, 2012) B197–B203.

[18] A. Bluestone, A. Jain, N. Volet, D. T. Spencer, S. B. Papp, S. A. Diddams, J. E. Bowers, and L. Theogarajan, *Heterodyne-based hybrid controller for wide dynamic range optoelectronic frequency synthesis*, *Opt. Express* **25** (Nov, 2017) 29086–29097.

[19] "Adrf6820: 695 mhz to 2700 mhz, quadrature demodulator with integrated fractional-n pll and vco," analog devices, rev. c, 2016." `https://www.analog.com/media/en/technical-documentation/data-sheets/ADRF6820.pdf`.

[20] "Ads4449 quad-channel, 14-bit, 250-msps, low-power adc." `https://www.ti.com/product/ADS4449`.

[21] "Zedboard." `https://www.avnet.com/wps/portal/us/products/avnet-boards/avnet-board-families/zedboard/`.

[22] "vescent." https://vescent.com/us/d2-105-laser-controller.html.

[23] J. Volder, *The cordic computing technique*, in *Managing Requirements Knowledge, International Workshop on*, vol. 1, (Los Alamitos, CA, USA), p. 257, IEEE Computer Society, mar, 1959.

[24] "Vivado 2021.1 - programming and debug." https://www.xilinx.com/support/documentation-navigation/design-hubs/dh0011-vivado-programming-and-debug-hub.html.

[25] J. E. Volder, *The cordic trigonometric computing technique*, *IRE Transactions on Electronic Computers* **EC-8** (1959), no. 3 330–334.

[26] R. Costanzo, Q. Yu, X. Shen, J. Gao, A. Beling, and S. M. Bowers, *Low-noise balanced photoreceiver with waveguide sin photodetectors and sige tia*, in *2020 Conference on Lasers and Electro-Optics (CLEO)*, pp. 1–2, 2020.

[27] J. Maneatis, *Low-jitter process-independent dll and pll based on self-biased techniques*, *IEEE Journal of Solid-State Circuits* **31** (1996), no. 11 1723–1732.

[28] S. Sidiropoulos, D. Liu, J. Kim, G. Wei, and M. Horowitz, *Adaptive bandwidth dlls and plls using regulated supply cmos buffers*, in *2000 Symposium on VLSI Circuits. Digest of Technical Papers (Cat. No.00CH37103)*, pp. 124–127, 2000.

[29] L. S. Theogarajan, *A low-power fully implantable 15-channel retinal stimulator chip*, *IEEE Journal of Solid-State Circuits* **43** (2008), no. 10 2322–2337.

[30] B. Chappell, T. Chappell, S. Schuster, H. Segmuller, J. Allan, R. Franch, and P. Restle, *Fast cmos ecl receivers with 100-mv worst-case sensitivity*, *IEEE Journal of Solid-State Circuits* **23** (1988), no. 1 59–67.

[31] B. Razavi, *RF Microelectronics (2nd Edition) (Prentice Hall Communications Engineering and Emerging Technologies Series)*. Prentice Hall Press, USA, 2nd ed., 2011.

[32] M. Figueiredo, J. Goes, and G. Evans, *Reference-Free CMOS Pipeline Analog-to-Digital Converters*. Springer, 01, 2013.

[33] T. Cho and P. Gray, *A 10 b, 20 msample/s, 35 mw pipeline a/d converter*, *IEEE Journal of Solid-State Circuits* **30** (1995), no. 3 166–172.

[34] A. Yukawa, *A cmos 8-bit high-speed a/d converter ic*, *IEEE Journal of Solid-State Circuits* **20** (1985), no. 3 775–779.

[35] A. J. Bluestone, "A versatile mixed-signal controller for optoelectronic frequency synthesis." https://escholarship.org/uc/item/38k2z5kj.

[36] R. Costanzo and S. M. Bowers, *A current reuse regulated cascode cmos transimpedance amplifier with 11-ghz bandwidth*, *IEEE Microwave and Wireless Components Letters* **28** (2018), no. 9 816–818.

[37] J. Yu, F. F. Dai, and R. C. Jaeger, *A 12-bit vernier ring time-to-digital converter in 0.13 ¡formula formulatype="inline"¿¡tex notation="tex"¿µm¡/tex¿ ¡/formula¿ cmos technology*, *IEEE Journal of Solid-State Circuits* **45** (2010), no. 4 830–842.

[38] T. Komljenovic, B. Szafraniec, D. Baney, and J. E. Bowers, *Stable arbitrary frequency generator*, *J. Lightwave Technol.* **35** (Nov, 2017) 4897–4902.

[39] R. Priest, *Analysis of fiber interferometer utilizing 3 × 3 fiber coupler*, *IEEE Journal of Quantum Electronics* **18** (1982), no. 10 1601–1603.

[40] "Tera scan." `https://www.toptica.com/products/terahertz-systems/frequency-domain/terascan/`.

[41] "Pro precision process and reliabity llc." `https://proprecisionprocess.com/`.

[42] A. Klein, *Logic operators on delta-sigma bit-streams*, *Mathematical and Computational Applications* **23** (2018), no. 1.

[43] A. Klein and W. Schumacher, *Algebraic operations on delta-sigma bit-streams*, *Mathematical and Computational Applications* **23** (2018), no. 3.

[44] M. Saeidi and L. Theogarajan, *A current-to-digital adc for low-noise high-precision applications*, pp. 667–670, 08, 2020.

[45] B. Li, W. Jin, L. Wu, L. Chang, H. Wang, B. Shen, Z. Yuan, A. Feshali, M. Paniccia, K. J. Vahala, and J. E. Bowers, *Reaching fiber-laser coherence in integrated photonics*, *Opt. Lett.* **46** (Oct, 2021) 5201–5204.

[46] W. Jin, Q.-F. Yang, L. Chang, H. Wang, B. Shen, M. A. Leal, L. Wu, M. Gao, A. Feshali, M. Paniccia, K. J. Vahala, and J. E. Bowers, *Hertz-linewidth semiconductor lasers using cmos-ready ultra-high-q microresonators*, *Nature Photonics* (2021).

[47] M. Xin, N. Li, N. Singh, A. Ruocco, Z. Su, E. S. Magden, J. Notaros, D. Vermeulen, E. P. Ippen, M. R. Watts, and F. X. Kärtner, *Optical frequency synthesizer with an integrated erbium tunable laser*, *Light: Science & Applications* **8** (Dec, 2019) 122.