DIGITAL FILTER AND ANALOG LDO DESIGN FOR ADPLL IN 180 nm CMOS TECHNOLOGY

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF
ENGINEERING AND NATURAL SCIENCES
OF ISTANBUL MEDIPOL UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF
MASTER OF SCIENCE
IN
ELECTRICAL, ELECTRONICS ENGINEERING AND CYBER SYSTEMS

By

Zehra Nur Oktay

June, 2021

DIGITAL FILTER AND ANALOG LDO DESIGN FOR ADPLL IN 180nm CMOS TECHNOLOGY

By Zehra Nur Oktay

30 June 2021

We certify that we have read this dissertation and that in our opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Master of Science.

_____

Assoc. Prof. Dr. Hakan Doğan (Advisor)

_____

Assoc. Prof. Dr. Tufan Coşkun Karalar

_____

Assist. Prof. Dr. Hüseyin Şerif Savcı

Approved by the Graduate School of Engineering and Natural Sciences:

_____

Assoc. Prof. Dr. Yasemin Yüksel Durmaz

Director of the Graduate School of Engineering and Natural Sciences

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

Signature :

Name, Surname: ZEHRA NUR OKTAY

ACKNOWLEDGEMENT

During my master's education I have received a great deal of support and assistance. First and foremost, I would like to express my sincere gratitude to my advisor Assoc. Prof. Hakan Doğan, for his continuous support, patience, help, insightful comments and suggestions at every stage of my research project. His absolute knowledge and broad experience have always encouraged me to learn more in all the time of my academic research. I am also deeply grateful to Dr. Mustafa Aktan for his continuous unwavering technical support on my study. His expertise was invaluable, and his intellectual feedback made this research an enlightening experience for me. I would like to acknowledge my colleague, team mate and best friend Tamer Eren for his greatest support and knowledge. Every time I had difficulty, he was there to help and motivate me. Finally, I would like to thank my family and friends for their patience and support during my masters.

This thesis founded by ASELSAN.

Zehra Nur Oktay

June, 2021

CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

Ω : Resistance
V : Volts
f : Frequency
Hz : Hertz

ABBREVIATIONS

ADPLL : All Digital Phase Locked Loop
APLL : Analog Phase Locked Loop
BJT : Bipolar Junction Transistor
BP : Band Pass
BR : Band Reject
CMOS : Complementary Metal Oxide Semiconductor
DC : Direct Current
DCO : Digital Controlled Oscillator
DLF : Digital Loop Filter
DPLL : Digital Phase Locked Loop
DRAM : Dynamic Random Access Memories
DRC : Design Rule Check
DSP : Digital Signal Processing
EEPROM : Programmable Read-Only Memories
FIR : Finite Impulse Response
HP : High Pass
IIR : Infinite Impulse Response
IRT : Impulse Response Truncation
LCD : Liquid Crystal Displays
LDO : Low Dropout Regulator
LF : Loop Filter
LP : Low Pass
LPF : Low Pass Filter
LPLL : Linear Phase Locked Loop
LVS : Layout Versus Schematic
PD : Phase Detector
PFD : Phase Frequency Detector
PLL : Phase Locked Loop
PSRR : Power Supply Rejection Ratio
PEX : Parasitic Extraction
PVT : Process Voltage Temperature
RFID : Radio Frequency Identification
RHP : Right Half Plane
SC : Switched Capacitor
SPLL : Software Phase Locked Loop
TDC : Time to Digital Converter
VCO : Voltage Controlled Oscillator

ADPLL İÇİN 180 nm CMOS TEKNOLOJİSİNDE DİJİTAL FİLTRE VE ANALOG DÜŞÜK BIRAKMA REGÜLATÖRÜ TASARIMI

ÖZET

Zehra Nur Oktay

Elektrik-Elektronik Mühendisliği, Yüksek Lisans

Tez Danışmanı: Doç. Dr. Hakan Doğan

Eş Danışman: Dr. Mustafa Aktan

Haziran, 2021

PLL, kararlı bir referans sinyalinin frekansını ve fazını bir osilatör çıkış sinyali ile senkronize eden elemanlar tarafından oluşturulmuş bir kontrol sistemidir. 21. Yüzyılda PLL, cep telefonları, radyo yayınları, televizyonlar, Wi-Fi yönlendiriciler, telsizler ve profesyonel iletişim sistemleri gibi cihazlarla insanların günlük yaşamlarında yaygın olarak kullanılmaktadır. PLL, LF, VCO ve PD gibi farklı bloklardan oluşur. Bu tezde ise, PLL'in 2 bloğu olan dijital filtre ve düşük bırakma regülatörü tasarımı anlatılmaktadır. Bu tez 5 farklı bölümden oluşmaktadır. İlk bölüm olan giriş kısmında, projenin kısa özetine, tasarım hedeflerine ve tezin bölümlerine yer verilmiştir. 2. Bölümde PLL, filtre ve LDO devrelerinin özellikleri, sınıflandırmaları ve kullanımları ile ilgili literatür taramasına yer verilmiştir. Tezin 3. Bölümünde ise, filtre ve LDO'nun uygulanması, tasarım yöntemleri ve hesaplamaları verilmiştir. 4. Bölümde her iki tasarımın sonuçları şekil ve tablolarla desteklenerek verilmiştir. Son olarak, tez 5. Bölümde sonuçlandırılmıştır. Filtre tasarımı için IIR dijital filtre tasarım topolojisi kullanılmıştır, çünkü IIR filtreler, geri besleme mekanizması sayesinde diğer metotlara göre çok daha az sayıda katsayı ile gerçekleştirilmektedir. Ayrıca, IIR filtrelerinin katsayıları istenen tasarım özelliklerine göre çok kolay ayarlanabilmektedir. Öte yandan, projede hızlı kilitlenme süresine sahip olan ve PLL'deki gürültüyü minimize eden bir filtre tasarımı gerekmektedir. Bu iki farklı özellik bant genişliğine göre ayarlandığından farklı katsayılara ve dolayısıyla farklı bant genişliklerine sahip iki filtre tasarlanmıştır. Öte yandan, LDO tasarımındaki hata yükselticisi, kazancı artırmak için diferansiyel yükseltici ve ortak kaynak yükseltici olmak üzere 2 farklı aşamadan oluşur. Hata yükselticisinde istenen en önemli durum, istenen bant genişliğine sahip kararlı bir tasarıma sahip olmaktır. Bunu sağlamak için tasarımda nulling dirençli Miller kompanzasyon tekniği kullanılmıştır. Filtre tasarımı ilk olarak Verilog HDL'de uygulanmıştır ve Cadence Virtuoso'ya aktarılmıştır. Filtrenin serimi, Cadence SOC Encounter tarafından oluşturulmuştur. Düşük bırakma regülatörü devresi Cadence Virtuoso yazılımında 180 nm CMOS teknolojisi ile tasarlanmıştır. LDO'nun serimi, Cadence'de manuel olarak tasarlanmıştır. Her iki tasarımın serimi de Calibre ile doğrulanmıştır.

Anahtar sözcükler: Filtre, LDO, IIR, PLL, CMOS.

DIGITAL FILTER AND ANALOG LDO DESIGN FOR ADPLL IN 180 nm CMOS TECHNOLOGY

ABSTRACT

Zehra Nur Oktay

MSc in Electrical, Electronics Engineering and Cyber Systems

Advisor: Assoc. Prof. Hakan Doğan

Co-Advisor: Dr. Mustafa Aktan

June, 2021

PLL is a control system built by elements that synchronize frequency and phase of a stable reference signal with an oscillator output signal. In 21st century, PLL is widely and essentially used in one's daily life in wireless and radio devices such as mobile phones, broadcast radios, televisions, Wi-Fi routers, walkie talkie radios and professional communication systems. PLL is built by different blocks such as LF, VCO and PD. In this work, 2 blocks of PLL, which are filter and LDO, were designed. Thesis was formed by 5 Chapters. In the 1st Chapter, introduction, which has brief summary of the proposed work, and goals of the design, is given. In the 2nd Chapter, literature review of PLL, filter and LDO is given. In the 3rd Chapter implementation, design methods of the filter and LDO are given. In Chapter 4, results of both of the designs were given in figures and tables. Finally, the thesis was concluded in the 5th Chapter. IIR digital filter design topology is used for filter design because IIR filters are realized with reasonable numbers of coefficients thanks to its feedback mechanism. Also, IIR filters' coefficients are adjustable according to design specifications. To have fast lock time and to eliminate noise in the PLL, two filters with different coefficients, which works one by one, were designed in the project. Moreover, error amplifier in the LDO consists of 2 stages, which are differential amplifier and common source amplifier, to boost the gain. One of the most important concerns in the error amplifier is to have a stable design with desired bandwidth. Miller compensation technique with nulling resistor is used for LDO design to maintain stability. The filter design is first implemented in Verilog HDL and imported to Cadence Virtuoso. Layout of the filter is generated by Cadence SOC Encounter. LDO circuit is designed with 180 nm CMOS technology in Cadence Virtuoso software. Layout of the LDO were designed manually in Cadence. Both of the design's layouts were verified with Calibre.

Keywords: Filter, LDO, IIR, PLL, CMOS.

CHAPTER 1

1. INTRODUCTION

PLL is a control system that synchronizes frequency and phase of a stable reference signal with an oscillator output signal. PLL is mainly built by different blocks, DCO, TDC and LPF.

The aim of this thesis is designing digital loop filter and LDO in the PLL system.

PLL needs not only fast lock time but also suppressing noise in the system. Fast lock time can be achieved with high bandwidth of filter, on the other hand suppressing noise can be achieved with low bandwidth of the filter. To have fast lock time and low sigma delta noise in the PLL, two different filters were used in the system.

Moreover, in the PLL, LDO was used to regulate the supply of the DCO. The goal is to design stable LDO with at least 40MHz bandwidth with at most 3dB noise introduction to the system. Stability was guaranteed by Miller compensation technique.

Chapter 2 shows the literature review about PLL systems, loop filter, and LDO, also their types and working principles. The implementation of proposed IIR digital loop filter and LDO are given in Chapter 3. In Chapter 4, results of both of the designs are given. Finally, the thesis concluded in Chapter 5.

CHAPTER 2

2. THEORITICAL PART

2.1. Basics of PLL

Phase-locked loop (PLL) is a fundamental control system that is commonly used among numerous fields in electronic applications such as communications, instrumentation, control systems, multimedia apparatus, radio frequency and wireless applications [1], [2].

Therefore, PLL is widely used in one's daily life as an essential electronic block in wireless and radio devices such as mobile phones, broadcast radios, televisions, Wi-Fi routers, walkie talkie radios and professional communication systems [2]. However, one of the most important functions of PLL is the frequency synthesizer which PLL generates discrete frequencies [3].

The working principle of PLLs is based on synchronizing frequency and phase of a stable reference signal with an oscillator output signal. After synchronization achieved, PLL is said to be locked, and in the locked state, phase error, which is the phase difference between two synchronized signals, should be ideally zero or remains constant. Even if phase error increases, it is reduced by the PLL's control mechanism and the output signal's phase is locked to the reference signal's phase. Mainly, a traditional PLL is built by 3 functional blocks, a voltage-controlled oscillator (VCO), phase detector (PD) and a loop filter (LF). Hata! Başvuru kaynağı bulunamadı. shows the block diagram of a PLL [4].

PD has an output voltage VPD and gives a high frequency component that measures the phase difference between reference signal and oscillator output signal. This measured phase difference goes through a loop filter, and at the output of LF error signal is obtained, which controls the VCO. Finally, VCO generates the output signal whose phase and frequency is proportional to VCO's input error signal [5].

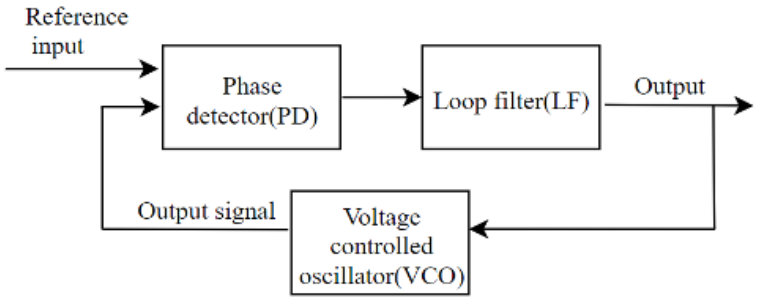

Figure 2.1: Block diagram of PLL [4].

The input and output of the PLL are given in the equation 2.1 and 2.2 [3].

$$V_{in} = V_A cos(wt) \tag{2.1}$$

$$V_{out} = V_B cos(wt + \phi) \tag{2.2}$$

2.1.1. PLL types

As it can be seen from the **Figure 2.2**, PLL can be categorized in different classifications according to their applications and orders.

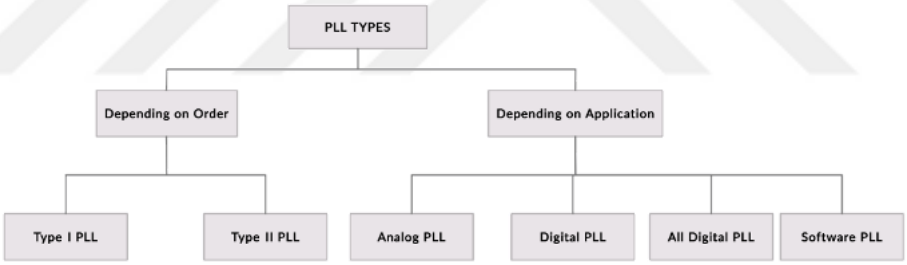

Figure 2.2: The PLL types are given.

The different applications result in different requirements for parameters such as bandwidth, the desired high frequency roll-off, transient behavior, PLL stability, and PLL has different types and orders according to these parameters. The PLL type is dependent on the number of poles at origin (number of integrators) in the open loop transfer function.

2.1.1.1. PLL types depending on the order

There are 2 major types of PLLs depending on their order, Type I and Type II PLL. In the type I PLL there is only 1 pole at the origin [6]. The phase diagram of type I PLL is given in **Figure 2.3.**

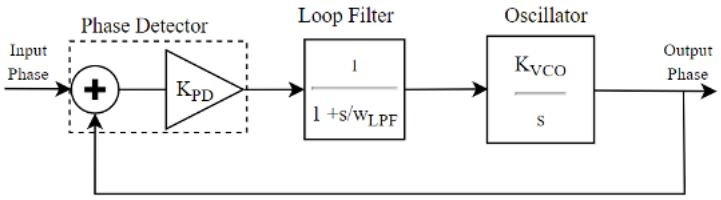

Figure 2.3: Type I PLL phase diagram [7].

Also, the transfer function of this system is given in equation 2.3:

$$H(s) = \frac{K_{PD}K_{VCO}}{\frac{s^2}{\omega_{LPF}}+s+K_{PD}K_{VCO}} \quad \textbf{(2.3)}$$

PD's gain is represented as K_{PD}. Damping ratio is given in equation 2.4:

$$\zeta = \frac{K_{PD}K_{VCO}}{\frac{s^2}{\omega_{LPF}}+s+K_{PD}K_{VCO}} \quad \textbf{(2.4)}$$

Type I PLL is relatively easier to design with understandable frequency response. However, there are some drawbacks hence it is generally not suitable for modern RF synthesizers. First of all, damping ratio, which defines stability, is directly dependent on ω_{LPF} and causes tradeoff between stability and VCO frequency modulation. Namely, VCO frequency modulation is done by ripple control, and this is achieved by low ω_{LPF} value, which causes insufficient stability. Also, acquisition rate is insufficient, meaning that VCO output frequency might not be equated to the input frequency when there is a huge difference between them initially. To eliminate these drawbacks, type II PLLs can be utilized [6].

The type I PLL is improved by adding a frequency detector and charge pump to the design, and it is called type II PLL. Frequency detector improves the acquisition range by giving more precise results about the difference between reference and output signal. On the other hand, adding only a frequency detector to the system might not always lock the system. Hence, the best solution is adding a phase frequency detector (PFD) to the system, which works as a frequency detector when the frequencies are different, and works as a phase detector when the frequencies are matched.

The phase diagram of type II PLL is given in **Figure 2.4** [6], [7].

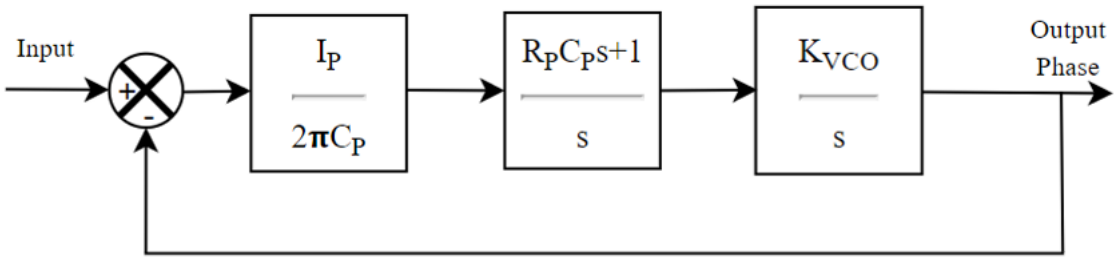

Figure 2.4: Type II charge pump PLL [6].

Open loop transfer function of this system is given in equation 2.5.

$$G(s) = \frac{I_P}{2\pi}\left(R_P + \frac{1}{C_P S}\right)\frac{I_P K_{VCO}}{s} \tag{2.5}$$

Close loop transfer function of the system can be calculated from equation 2.6.

$$H(s) = \frac{G(s)}{1+G(s)} = \frac{\frac{I_P K_{VCO}}{2\pi C_P}(R_P C_P S+1)}{S^2+\frac{I_P}{2\pi}K_{VCO}+R_P S+\frac{I_P}{2\pi C_P K_{VCO}}} \tag{2.6}$$

The damping ratio of the system is given in equation 2.7.

$$\zeta = \frac{R_P}{2}\sqrt{\frac{I_P C_P K_{VCO}}{2\pi}} \tag{2.7}$$

Phase diagram and concept of phase frequency detector is given in **Figure 2.5.** As it can be observed from the figure, the PFD circuit has two inputs and outputs. Inputs are A and B, outputs are QA and QB. When input A is 1, output QA is equalized to 1, and when input B is 1 the output QA is equalized to 0.

In the first case, shown in **Figure 2.5-a**, when frequencies are different, if phase of A (B) is greater than phase of B (A), output QA (QB) produces results as pulses and QB (QA) is zero. On the other hand, when frequencies are equal to each other, shown in **Figure 2.5-b**, the pulses with widths equal to the frequency difference are produced at the output QA or QB [6], [7].

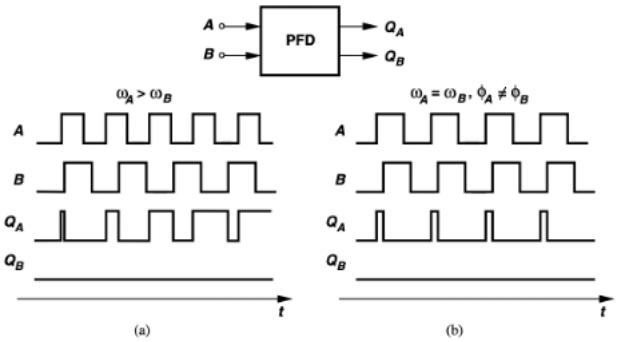

Figure 2.5: PFD response to the different inputs [6].

2.1.1.2. PLL types depending on the application

There are mainly 4 types of PLLs which are, analog phase-locked loop (APLL) also known as linear phase locked loop (LPLL), digital phase-locked loop (DPLL), all digital phase-locked loop (ADPLL), and software phase-locked loop (SPLL) [8]. Even though APLL and DPLL has a similar working principle, each PLL type has a different analogy and behavior.

Firstly, as is evident from its name, APLL is built by purely analog blocks. PD is realized by analog four quadrant multiplier. Passive or active RC filter is used as a LF, and VCO is used as an output signal generator of PLL. On the other hand, DPLL consists of certain digital device such as PD that contains digital gates e.g. XOR gate or JK-flip flop, and the rest of the components are analog.

Moreover, ADPLL consists of all digital blocks therefore PD and LF is obtained by digital components. Additionally, SPLL can be implemented by a computer program instead of implementing on a hardware [4].

Even though their implementations are different, all of these PLL types have a common working principle in general. The oscillation frequency is controlled by the PD output. As the oscillation frequency changes, PD's phase error output is adjusted to correct for the frequency. As a result, PD's output voltage changes, and the oscillator frequency is locks to the desired value. The feedback clock is a divided version of the oscillator output and the division is done by the feedback divider. Finally, the loop filter averages the error signal in the loop and ensures stability of the loop [8].

In a summary, PLL types and its components are given in **Table 2.1.**

Table 2.1: PLL Types.

	Phase Detector	Loop Filter	Oscillator
APLL	Analog four quadrant multiplier	Analog low pass filter (LPF)	Voltage controlled oscillator (VCO)
DPLL	Digital PD with digital gates	LPF	VCO
ADPLL	TDC	Digital low pass filter	Digitally controlled oscillator (DCO)
SPLL	Implemented in Software	Implemented in Software	Implemented in Software

2.1.2. Benefits of all digital pll and comparison with the other types

The improvements in the size of CMOS circuits, which are in nanometer range, have concluded greater timing accuracy, power and density with lower price per performance in digital circuits. However, the CMOS size improvement has affected analog circuits negatively because of the lower supply voltage and higher leakage at the gate.

Moreover, an analog PLL circuit can operate in phase domain only when it is in the locked state. Also, phase domain modelling can only be obtained during locked conditions [9]–[12]. For these reasons, digitally assisted designs or fully digital designs have been taking the place of analog designs [13]. Furthermore, all-digital phase locked loops (ADPLL) are more efficient than analog PLLs in several fields such as area, scalability, testability, and programmability [2], [14]–[21]. Instead of using a phase detector and a passive analog filter, time-to-digital converters and more cost-effective digital loop filters are used in ADPLLs. This replacement results in reduced area and advanced compatibility between all the digital blocks. Furthermore, the testability and programmability of the circuit is improved [12].

In digital PLLs, digital phase detectors provide digital output which has the time difference information between the reference signal and the feedback signal. The digital phase detectors are free from thermal noise, drift and charge pump leakage.

The output of the phase detector goes through the digital filter, and the incoming signal is filtered and integrated at the digital filter. The advantage of the digital filter is that its

parameters can be changed according to the desired application, which makes the system more flexible.

2.1.3. All digital pll working principle

The all-digital PLL is built from entirely digital functional blocks. Major ADPLL blocks are TDC, PD and the LF and they are implemented digitally, also VCO is implemented as a digital DCO [4]. In short, ADPLL is a mixed signal system which offers increased testability, programmability, stability, portability, low power consumption, lower noise, and higher flexibility compared to the other PLL technologies [9], [22]–[24].

The basic ADPLL block diagram is given in **Figure 2.6.** As it can be seen from the figure, there are three main blocks: TDC, digital loop filter, and DCO. TDC block compares, measures and digitizes the phase difference between the feedback signal and the reference signal. DLF decreases the effects of noise generated by the TDC and produces an output according to the digitized phase error which is produced by the TDC. After that, DCO produces a signal whose frequency is proportional to DLF's output. [9], [25]. The resolution of TDC and sensitivity of DCO should be adjusted very carefully because these two blocks degrade the purity of the output significantly by adding quantization noise to the system [9].

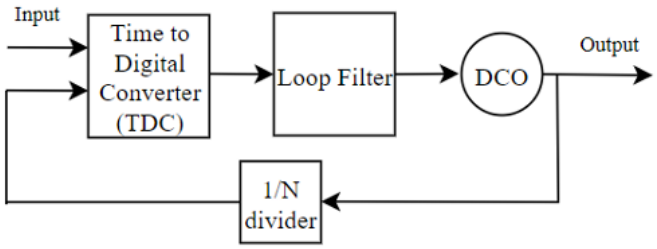

Figure 2.6: The block diagram of all digital PLL [25].

The first part of this thesis focuses on the design of the Digital Loop Filter and below, details of the filter design are given.

2.2. Basics of Loop Filter

The circuits' or system's performance is directly proportional to the performance of the filter because it averages the control signal and stabilizes the system. Basically, the filter's duty is to pass desired signals from input to output without distortion and block the other

signals. Moreover, in PLL designs, loop filters are one of the most important blocks because incorrect filter design might result in instability in the PLL [8].

Filters are used in widely distinctive areas with many different purposes. First of all, filters are used for noise suppression in noisy signals, like received radio signals, signals in television cameras or infrared imaging devices, biomedical signals like brain, heart, or neurological signals, signals from analog media, like analog magnetic tapes. The next area filters are used is frequency range enhancement, e.g. treble, bass control and graphic equalizers in audio systems, and edges of images. Besides, in communication applications filters are used to limit the bandwidth to prevent interference from the neighboring channels and helps prevent aliasing in sampling. In addition, filters remove or attenuate specific frequencies, they repel DC components of signals, also attenuate interferences from a power line. Lastly, digital filters can do differentiation, integration and Hilbert transform operations [26].

As it was mentioned, PLL is a feedback system therefore it provides suppression of noise and undesirable signals in a specific frequency range. This frequency range is adjusted inside the feedback loop, and input signals are not affected within these frequencies. [27]. In PLLs, loop filters average the phase detector output voltage without affecting the usual working principle of the system, which is basically comparing two phases between input and output. For example, PD usually produces noise at the reference frequency, and this noise should be filtered out before going into the oscillator. Without loop filters, VCO is insufficiently modulated, and produces jitter. Taking everything into account, loop filters have a very critical function in PLL circuits [8].

In **Figure 2.7** first order low-pass RC loop filter schematic is given and the frequency response of this filter is given in equation 2.8 [8]:

$$F(s) = \frac{1}{1+sCR} \qquad \textbf{(2.8)}$$

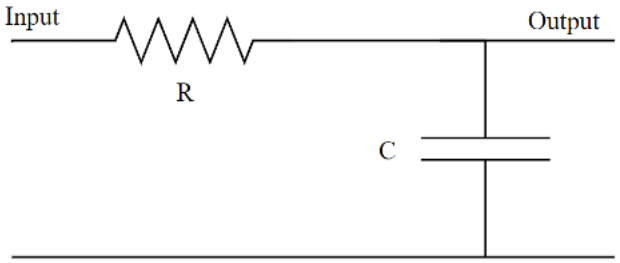

Figure 2.7: First order low-pass RC loop filter circuit [10].

Filters are divided into different categories based on their working principles. First category is made up of FIR (finite impulse response) and IIR (infinite impulse response) depending on the filter's impulse response. Depending on the dimensions, there are 2 different types, one dimensional (1D), which is to process signals, and two dimensional (2D), which is to process images, filters. Lastly, filters are classified according to their frequency response, that are low pass (LP), which block high frequencies and pass low frequencies up to desired cut-off frequency, high pass (HP), which block low frequencies and pass high frequencies, band pass (BP), which only pass a desired frequency range and block other frequencies, and band reject (BR) that only ban a specific frequency range and pass other frequencies [28]–[30].

In PLLs, a low pass filter is used as the loop filter. Loop filter's duty is basically suppressing random noise and high-frequency components generated by the phase detector. LF's output is also the input of the VCO, that is a dc control signal [29].

Filtering a signal results in change in the frequency spectrum, and shaping and controlling of the desired signal is done [28].

Overall, loop filters in PLLs are divided into 2 subcategories, analog loop filters and digital loop filters. Analog filters have 2 types: passive and active analog filters. **Figure 2.7** is an example of a passive filter, and it is made up of passive components such as resistors, inductors and capacitors. The main advantage of passive filters is they do not consume any power, but these filters are bulky. Active filters on the other hand contain active components like operational amplifiers (op-amps). An example of an active filter schematic is given in **Figure 2.8.** In these filters, op-amps are used with passive components to obtain filtering with desired frequency response. The filter performance is dependent on the op-amp performance, so that the signal quality is affected by the gain, slew rate, bandwidth and other properties of the op-amp. Also, op-amps introduce noise to the system [8].

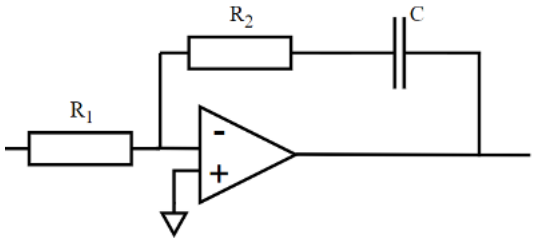

Figure 2.8: The circuit diagram of active loop filter.

Analog filters are commonly used in noise reduction, video signal enhancement and graphic equalizers, and are built by all-analog electronic components such as resistor capacitors and operational amplifiers. While designing analog filters, resistor, capacitor and inductor values need to be calculated [30]. These components introduce noise, nonlinearities, limited dynamic range, variations in component values, lack of flexibility, and imperfect repeatability in the analog filter circuits [26]. Analog filters are applied to continuous time signals and consequently their outputs are continuous time signals, and the analog filter's range is infinite. However, in real life, limitations in the frequency response of components used in the filter design will cause the filter frequency response to be limited. [26].

On the other hand, digital filters are applied to discrete time signals, and outputs are discrete time signals. In digital filters, there is a limitation of frequency range, that is half of the sampling rate, so it is finite. Also, digital filters have a digital processor to be able to do numerical calculations like addition, multiplication and data movement [26]. Digital filters have higher frequency accuracy than analog filters, and linear phase. They are flexible, limitlessly repeatable, easy to design and simulate. Moreover, adaptive filtering can be done with digital filters. Besides, they are not affected by internal noise, and their dynamic range does not have limitation problems. Digital filters are very useful to implement pure time delays, time-varying filters, and adaptive filters. Design of digital filters is done by calculating coefficient values and these take the place of passive components in the digital filters [30].

Digital filter design can easily be done in several steps. Beforehand, filter specifications are determined according to desired specifications, then, filter parameters, amplitude and phase response are calculated. As part of these parameters, cut-off frequencies, pass-band and stop band width, overall gain, stop band attenuation, and ripples should be very well determined. Moreover, depending on the requirements, appropriate windowing functions such as Butterworth, Chebyshev, Kaiser etc. should be realized [28]. Digital filters are

not affected from drift and temperature changes, indeed they are extremely stable. Moreover, digital filters are more functional than analog filters, and are able to process signals in different ways such as they can adapt changes, since the filter specifications can be changed very easily and quickly. Finally, instead of using complex analog circuits, fast DSP processors can undertake sophisticated filter designs.

As it was mentioned before, digital filters are mainly divided into 2 subcategories depending on the impulse response length of the filter, finite impulse response (FIR) and infinite impulse response (IIR).

2.2.1. Finite impulse filters

As its name applies FIR filters' response is in a finite period, it does not have feedback in the filter circuit, and are able to perform substantial things such as obtaining a perfect linear phase response that analog filters cannot perform. Also, FIR filters are suitable for the implementations where there are not any memory limits and there is enough performance to perform computations [30]. FIR filters settle to zero in finite time [31].

In the **Figure 2.9**, the block diagram of the FIR filter can be seen.

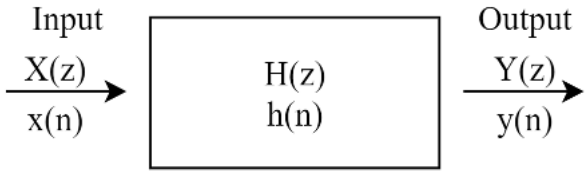

Figure 2.9: Block diagram of FIR digital filter [32].

FIR's output is given in equation 2.9 and 2.10:

$$Y(z) = X(z)H(z) \quad \textbf{(2.9)}$$

$$y(n) = x(n) * h(n) \quad \textbf{(2.10)}$$

The transfer function of a typical FIR filter is given in equation 2.11 [28].

$$\mathrm{H(z)} = \sum_{n=0}^{N-1} h(n)z^{-n} \quad \textbf{(2.11)}$$

N is the length of the filter and h(n) is the impulse response of the filter.

Finally the frequency response of FIR filter in frequency domain is given in equation 2.12, and w_k is given in equation 2.13 [28].

$$\mathrm{H}(w_k) = \sum_{n=0}^{N} h(n) e^{-jw} k^n \quad \textbf{(2.12)}$$

$$w_k = \frac{2\pi k}{N} \quad \textbf{(2.13)}$$

The structure of FIR filter is built by series connected delay elements, z^{-1}, with desired filter coefficients. In **Figure 2.10**, structure of FIR filters is given. Each delay element is multiplied with filter coefficients, and these results are added up at the output.

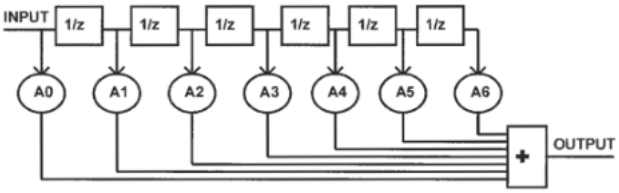

Figure 2.10: FIR filter design diagram [9].

FIR filter types are quite successful to realize in a wide range of frequencies by adjusting weight of the filter coefficients and the number of filter taps. Despite the fact that FIR filters perform substantially, they need a large number of multiply-accumulates which causes a need for fast and efficient DSPs. FIR filters have a wide range of applications in different areas such as audio and biomedical signal enhancement applications. Using FIR filters in the system have several important benefits over IIR filters, in some areas namely FIR filters are more stable, are able to have a perfect linear phase, have flexibility to change the magnitude response, and have an easier implementation [26],[28].

As it was mentioned, FIR has a finite response, and it eventually settles to zero because it does not have a feedback mechanism. However, in some applications of FIR filter, e.g. moving average filter, the feedback is applied to the system with finite impulse response. Even though this application of FIR filter has a feedback system and each Nth prior sample fed back at each cycle, impulse response is still finite and finally reaches to zero [31].

There are 4 types of FIR filter, Type I, Type II, Type III and Type IV.

Type I filters are even order filters and its initial phase is zero which results in a constant phase delay. The symmetry can be seen from impulse response of the filter in equation 2.14. Type I is a convenient design for low pass, high pass, band pass, band stop, and multiband filters. Type II filters also have zero initial phase with odd order and symmetry condition can be seen from the filter's impulse response equation 2.14.

$$h[n] = h[N - n] \quad \textbf{(2.14)}$$

Type II is a convenient design for low pass and bandpass filters. Type III filters also are even order filters and their initial phase $\pi/2$ which provides constant group delay. The anti-symmetry can be seen from impulse response equation 2.15.

$$h[n] = -h[N-n] \quad \textbf{(2.15)}$$

The type IV filters are odd order filters with an initial phase $\pi/2$ which provides constant group delay. The anti-symmetry can be seen from impulse response equation 2.15. Type III and Type IV are mostly used for differentiators and Hilbert transformers [26]. Fundamentally FIR filters have several design methods. One of the first two methods are windowing method, which is a simple and convenient method, but designed filter's order might be higher than its possible minimum order. Least squares design method, that is done in the frequency domain, minimizes integral square error. Another design method is equiripple design method, which gives the best result in terms of ripple in the frequency domain. This method is based on numerical optimization via well developed software tools, and it provides less complexity with desired results.

Impulse response truncation (IRT), which can be used in LP, HP, BP, BS, multiband filters, differentiators, and Hilbert transformers, is another easy method to use for design. This method is applied by the computation and trimming of the finite length of the ideal filter's impulse response, namely it is basically truncating the Fourier series of a periodic function mathematically. Therefore, while it goes through Gibbs phenomenon, which is the condition of constant tolerance parameters regardless of filter's order and shape, it is useful to minimize the integral of frequency-domain square error.

Least-squares method is another design method and it is considered as an alternative method of windowing. This method adjusts its desired parameters and weighting function by trial and error. It is a preferable method because it provides a minimum order filter with desired parameters. This method requires advanced processing techniques which can be achieved with specific design software that are not easily accessible. Hence, even though least-squares method is a convenient and handy method, it is not generally preferred due to absences of the software tools [26].

Advantages of the FIR filter can be given as follows. Firstly, as it was mentioned FIR filters have a perfectly linear phase, and this provides distortion free filtering, and

frequencies of the signal are shifted the same amount therefore their relative phase relationship is preserved.

Secondly, even though there are many advantages of having feedback in the filter design, it genuinely causes filter response to be unstable which is a serious issue. FIR filter's feedback free design always remains stable, and this feature gives an enormous advantage over IIR filter. Next, FIR filter has an arbitrary magnitude response, and it results in relatively easier customization for the filter design. Last advantage is, FIR filters can be applied to fixed points relatively easier than IIR filters, and effects of quantization are easier to undergo. FIR filter also has some disadvantages. First of all, more coefficients are necessary obtaining sharp and distinct cut off regions, and having more coefficients results in the need for large amounts of memory and multiply-accumulate operations. The requirement of more coefficients has a domino effect, and it results in higher latency in FIR filters. Again, this might cause large group delay, and it might result in failure to obtain desired stability. For all of these reasons, FIR filters might not be suitable for real-time closed-loop control applications. Lastly, unlike IIR filters, FIR filters do not have analog equivalents, so the filter cannot be easily transferred from its analog equivalent [32].

2.2.2. Infinite impulse filters

The second type of digital filters, IIR filters, fundamentally imitates the analog filter's performance, and uses feedback which results in its impulse response to be an infinite period of time. Unlike FIR filters, IIR filters are realized with lesser numbers of coefficients thanks to its feedback mechanism. IIR filters are widely used in various fields like audio equalization, biomedical sensor signal processing, smart sensors and RF applications [30].

IIR filters have an infinite response, which is provided by a feedback topology in the filter circuit. The block diagram of a typical IIR filter is given in **Figure 2.11** and this figure can be represented as in equation 2.16.

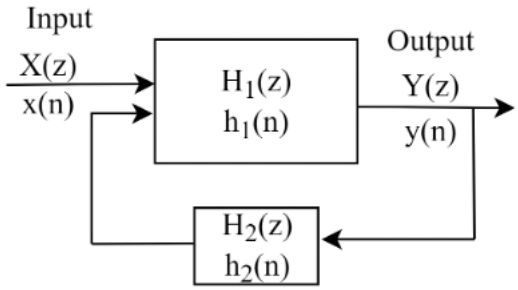

Figure 2.11: The block diagram of IIR filter.

$$y(n) = \sum_{n=0}^{M} p_k x(n-k) - \sum_{n=0}^{N} q_k y(n-k) \quad \textbf{(2.16)}$$

pk and qk are the filter's coefficients, M and N are the filter coefficients. Finally, the transfer function is given in equation 2.17.

$$H(z) = \frac{\sum_{k=0}^{M} p_k z^{-k}}{\sum_{k=1}^{N} q_k z^{-k}} \quad \textbf{(2.17)}$$

The structure of IIR filter is given in **Figure 2.12**. z^{-1} in the figure represents delay element like flip flops. This biquad structure in IIR filters provides application of different types of filters like lowpass, highpass, bandpass and bandstop. The transfer function of the IIR filter in **Figure 2.12** is given in equation 2.18.

$$H(z) = \frac{Y(z)}{X(z)} = \frac{A_0 + A_1 z^{-1} + A_2 z^{-2}}{1 - B_1 z^{-1} - B_2 z^{-2}} \quad \textbf{(2.18)}$$

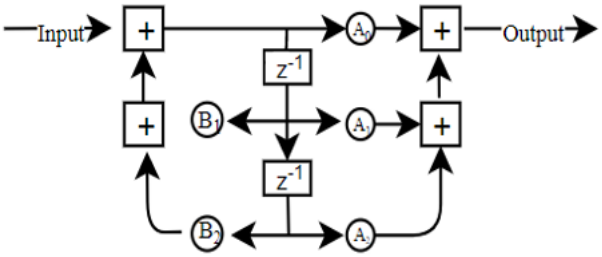

Figure 2.12: The first order IIR filter structure [26].

Analog to digital filter transformation techniques

Bilinear transformation

As it was mentioned, bilinear transformation is used to transform analog filters to equivalent digital filters, which is basically converting s domain (analog domain) to z domain (digital domain), and analog frequency response H(s) to digital frequency

response H(z). As it can be seen from **Figure 2.13**, z plane is circular plane that is from 0 to +- infinity, and s plane is x-y coordinate system with a real and imaginary axis.

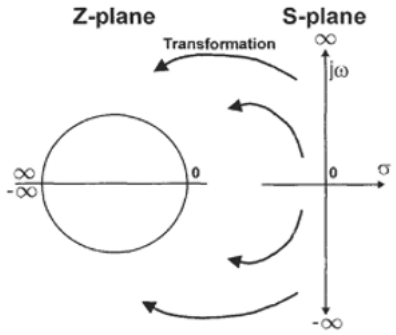

Figure 2.13: The s plane to z plane transformation diagram [26].

Pre-warping

Simple bilinear transformation might not give an exact desired transfer function, and the pre-warping technique can give the wanted frequency transformation. To obtain this, the following relationship, which is given in equation 2.19 and 2.20, between analog and digital filters should be applied.

$$w_{analog} = \tan\left(\frac{w_{digital}}{2}\right) \quad \textbf{(2.19)}$$

$$w_c = 2\pi\left(\frac{f_c}{f_s}\right) \quad \textbf{(2.20)}$$

where w_c is the cutoff, w_{analog} is the analog cutoff frequency. The relationship in equation 2.20 should be satisfied in the s plane before bilinear transform is applied [8]. IIR filters' coefficients are relatively easier to obtain because it requires less computations. IIR filter's feedback mechanism provides it with a memory because previous outputs are fed back to the filter at each cycle. Consequently, designing an IIR filter is not an easy task, it is even more complex than designing a FIR design.

During the design of IIR filter, coefficients are obtained first by realizing equivalent analog filters, then bilinear transform, or impulse invariance methods are implemented to obtain the desired results.

For the design, one of the filter types, which are Butterworth, Chebyshev-I, Chebyshev-II and elliptic filter, can be chosen according to the desired functions of the filter. Each of these filter types has their own ripple characteristics. For example, Butterworth filter is monotone in all frequencies, and it has the highest order, Chebyshev-I has monotone response in the stop band and equiripple in the pass band, Chebyshev-II has exact opposite

ripple effect of Chebyshev-I, and lastly an elliptic filter has equirippIe response in all bands, and it has the smallest order. The desired type should be chosen depending on the requirements of the desired filter [26].

IIR filters can form different types when they are used together. The cascaded filter's transfer function is each IIR filter's transfer function multiplied together, and parallel combination's output is their transfer functions added together.

There are various advantages of IIR filters. Firstly, implementation cost for these types of filters are low thanks to their feedback mechanism. IIR filters require a lesser number of coefficients and memory than FIR filters do for similar filter requirements. Secondly, IIR has a low latency and real time control, hence they can be used in high speed RF applications [33].

Lastly, IIR filters have an analog equivalent which enables the easy transform an analog filter into an equivalent digital IIR filter. Standard transformation procedures are available, and it can be done easily [26].

On the other hand, there are several disadvantages of IIR filters. First of all, IIR filters have nonlinear phase characteristics particularly around 3dB cut-off frequencies. Secondly, if IIR filters are applied to fixed points, further detailed analysis such as scaling and numeric overflow need to be performed. Next, compared to FIR filters, IIR filters are not as suitable and flexible for nonstandard frequency responses. Transferring IIR filter from its analog equivalent is an easy and straightforward process, but other than this approach, it is very complex to design IIR filters. Lastly, because of its feedback paths, IIR filters are more unstable than FIR filter.

All in all, IIR filters are widely used, especially in memory limited applications and where linear phase response requirement is insignificant [26].

2.3. Basics of LDO

Low Dropout Regulators (LDO) are the main component that is widely used in portable electronic applications and provide a straightforward and cheap method to basically regulate output voltages of circuits which have high input supply voltages. The design and implementation of LDOs is very simple, and it is widely used in wide range of electronic systems.

LDOs are not only used in DC circuits, but also high frequency devices because of their fast response to output current or input voltage, which affects the performance in high frequencies.

LDO takes a voltage, which can be very close to the desired output voltage, converts it to a controlled, stable, low noise DC output voltage. LDO provides a low difference between the input to output voltage, hence the name Low Dropout [34].

Regulators, which mainly regulate and condition power, are divided into 3 categories, switched- capacitor (SC) regulators, linear regulators and switching regulators.

Switched- capacitor (SC) regulators generate desired regulated output voltage from the input voltage by transferring charge in between capacitors. Its function is like a charge pump because it uses capacitors to store energy and transfer charge from input to the output. SC regulators can increase, decrease or change polarity of the input voltage, and these regulators are used in simple applications such as electrically erasable radio-frequency identification (RFID) tags, programmable read-only memories (EEPROMs), flash memories, liquid crystal displays (LCD), dynamic random access memories (DRAM), diode emitters, and in energy harvesting circuits such as solar cells, thermoelectric generators or piezoelectric sensors, which does not require too strict specifications. In **Figure 2.14**, block diagram for a SC regulator is given. In this circuit, according to the switch's position, which is controlled by V_{clk} phase, the capacitors C1 and Cl_{oad} transfer the charge. S1 and S2 work in opposite phases. For example, in the first clock phase S1 is closed and S2 is opened, therefore steady state voltage on the C1 is input voltage V_{IN}. In the second clock phase, S1 is opened and S2 is closed, V_{IN} and the voltage on the capacitor add up to make the output $2V_{IN}$ [35], [36].

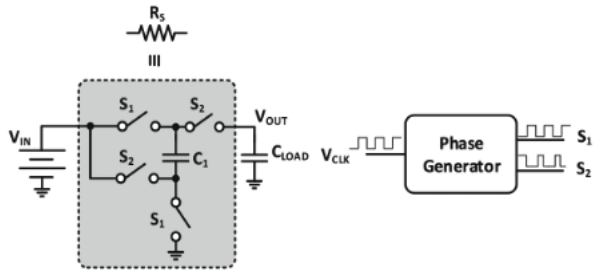

Figure 2.14: Switched capacitor regulator's block diagram [36].

In switching regulators, voltage is regulated by energy transfer to the load by storing energy on an inductor. As it can be seen from **Figure 2.15** that the switching regulators include a transistor switch, which is used to charge up the inductor, a low pass filter,

which filters out the pulse width modulated signal to generate an average output, and a regulation loop, which is realized by a control block to set the output voltage with pulse width modulation. Switching regulators can achieve high efficiencies up to 95%. However, to achieve these high efficiencies, high quality factor off-chip inductors, which are expensive, are used [37].

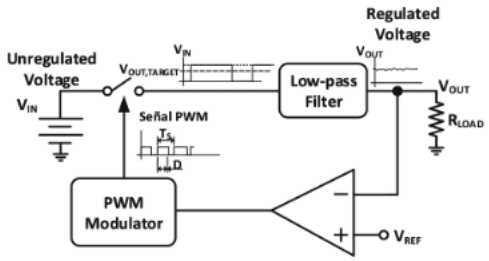

Figure 2.15: Block diagram of switching regulator [37].

The third type of the regulators is the linear regulators, which uses an error amplifier to copy a reference voltage and regulate the output. In **Figure 2.16**, classical linear regulator topology is given. There is an output pass element, which is a transistor, and it provides current for the load. Secondly, there is a feedback network, which feeds the sampled output voltage to the input where it is compared with a reference voltage. Feedback network can be realized by resistor division, and an error amplifier compares the reference voltage to the feedback voltage. Consequently, the error amplifier keeps the output voltage constant by modulating the pass element's controlling voltage [38].

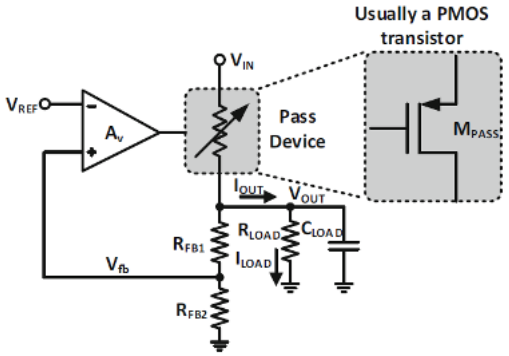

Figure 2.16: Linear regulator block diagram circuit [38].

The dropout voltage is the difference between input and output voltage. This dropout voltage in linear regulators has 2 important roles, firstly it defines the minimum unregulated input voltage which meets the desired functions, and secondly it determines the power dissipation in the LDO and the efficiency.

In comparison, switching regulators are more energy efficient than linear regulators. On the other hand, linear regulators have many advantages, such as having a lower output noise, giving faster response to input and output transients. Also, they don't require bulky inductors, so they are smaller, cheaper and easy to design. Even though switching regulators have very high efficiency, linear regulators have a satisfying efficiency if the voltage drop thorough the regulation is relatively small. Besides, in low power applications, the power lost might be considered as negligible, so LDO efficiency is not very crucial in such cases.

As a conclusion, the main advantages of linear regulators are, well-regulated output voltage, low noise level, lower PCB area and less complexity. At the same time, the output voltage can be as high as unregulated input voltage. Linear regulators are very useful in applications that require a high performance because these applications seek fast transient responses to the changes in the current load and input voltage, also they show sensitivity to power supply deteriorations.

Linear regulators can be examined in 2 types: standard linear regulators and low dropout linear regulators (LDOs). Linear voltage regulators work accurately when there is a large amount of input to output voltage drop. Therefore, the linear voltage regulator needs high input voltage power supply which causes low power efficiency. On the other hand, even if input and output voltage levels are close to each other, LDO functions accurately, and its power efficiency is not negatively affected.

Further, LDO is able to function at a low supply voltage. LDO usually works with a voltage reference that results in a less supply dependence and less temperature drift reference voltage. The reference voltage can either be a bandgap reference, which is built by parasitic vertical BJTs, or threshold-voltage difference that can be applied in different methods like flat-band voltage difference that has changing gate materials, selective channel implant and work-function difference with different gate doping.

The design of LDOs should be done carefully with respect to given design criteria; input and output voltage range, load current requirement, load, temperature, output accuracy, dropout voltage, which is the input and output voltage difference, output noise, power supply rejection ratio (PSRR), quiescent current, which is the input and output current difference, efficiency, which is depended on quiescent current, transient response, line

regulation, which is having a specific value of output voltage regardless of load current variations.

LDO in analog devices need to be stable in specific voltage values and temperatures. Ideally, the output voltage is stable, and is not affected by temperature changes. Additionally, LDO's dropout voltage, which is the input output voltage difference, is expected to be as low as possible. Dropout voltages as small as 200mV is common in LDO designs. LDOs also can be used as post regulators, meaning that high efficiency switcher's output can be regulated for noise filtering and stability.

To maintain LDO's stability compensation is needed. The compensation can be done internally, which is done by connecting a capacitor at the internal nodes to have a dominant pole at this node, or externally, which is done by connecting a large capacitor to the output with the purpose of having the dominant pole at this node. Even though they have insufficient response to the fast load-dumps, on-chip compensated linear regulators are useful type of regulators due to their low cost. They are mainly used in low power applications [39].

In brief, linear regulators are divided into different categories according to their applications: high or low power, externally or internally compensated, and low dropout or high dropout regulators.

LDO stability effects the performance of the system, and in the proposed work, LDO is connected to the DCO directly. While giving regulated voltage to DCO, LDO should be stable in order not to degrade DCO's performance. In the LDO, dominant pole compensation technique is done to maintain stability of the circuit. This technique is basically done by adding a large capacitor to a large impedance node [39], [40].

CHAPTER 3

3. EXPERIMENTAL PART

3.1. Design Calculations of Filter

In **Figure 3.1**, the block diagram of the proposed PLL is given. General transfer function of the IIR filter is given in equation 3.1, an and bn in the equation are filter's coefficients [41].

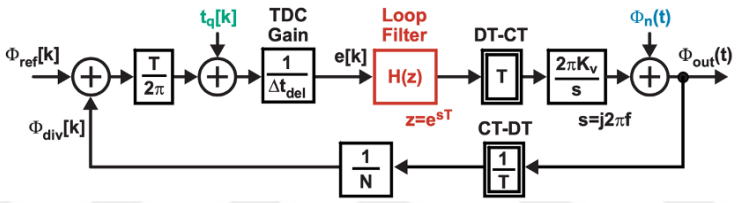

Figure 3.1: The block diagram of proposed PLL [42].

$$H(z) = \frac{B(z)}{A(z)} = \frac{b_0 z^N + b^1 z^{N-1} + b_2 z^{N-2} + \cdots + b_N}{z^N + a^1 z^{N-1} + \cdots + a_N} = \frac{b_0 + b^1 z^{-1} + \cdots + b_N z^{-M}}{1 + a_1 z^{-1} + \cdots + a_N z^{-N}} \quad \textbf{(3.1)}$$

Open loop transfer function of the PLL is given in equation 3.2.

$$A(z) = \frac{KT^2}{s^2 T^2} \frac{1 + \frac{s}{w_z}}{1 + \frac{s}{w_s}} \quad \textbf{(3.2)}$$

K in the equation, which is the open loop gain of PLL, is given in equation 3.3. w_p is given in equation 3.4, and the ratio of w_{cp} to w_o is given in equation 3.5.

$$K = \frac{{w_o}^2}{1 + \frac{1}{Q}\frac{w_o}{w_{cp}}} = \frac{w_o * Q}{\frac{Q}{w_o} + \frac{1}{w_{cp}}} = \frac{{w_o}^2}{1 + \frac{1}{Q} \times \frac{w_o}{w_z - 1}} \quad \textbf{(3.3)}$$

$$w_p = w_o \frac{1}{Q} + \frac{w_{cp}}{w_o} \quad \textbf{(3.4)}$$

$$\frac{w_{cp}}{w_o} = \frac{\frac{w_z}{w_o}}{1 - \left(\frac{1}{Q}\frac{w_z}{w_o}\right)} \quad \textbf{(3.5)}$$

w_{cp} equals to equation 3.6.

$$w_{cp} = \frac{w_z}{1-\left(\frac{w_z}{w_{co}Q}\right)} \quad \textbf{(3.6)}$$

2nd order transfer function of the PLL is given in equation 3.7.

$$A(s) = \frac{T}{\Delta tdel}\frac{N}{K_v}\frac{1}{s}H(s) \quad \textbf{(3.7)}$$

Equation 3.8 is obtained by equating 3.2 and 3.7 to each other.

$$A(s) = \frac{T}{\Delta tdel}\frac{N}{K_v}\frac{1}{s}H(s) = \frac{KT^2}{s^2T^2}\frac{1+\frac{s}{w_z}}{1+\frac{s}{w_s}} \quad \textbf{(3.8)}$$

Therefrom, H(s) is given in equation 3.9.

$$H(s) = \frac{\Delta tdel}{T}\frac{N}{K_v}\frac{K}{s}\frac{1+\frac{s}{w_z}}{1+\frac{s}{w_s}} \quad \textbf{(3.9)}$$

The relationship between z and s domain, which is for mapping s domain to z domain, is given in equation 3.10.

$$z = e^{st} \quad \textbf{(3.10)}$$

This equation can be converted to the form in equation 3.11 and eventually 3.12 at low frequencies.

$$z^{-1} = 1 - sT \quad \textbf{(3.11)}$$

$$s = \frac{1-z^{-1}}{T} \quad \textbf{(3.12)}$$

From these equations, type 2 second order filter's transfer function H(z) is given in equation 3.13, and filter pole zero locations are given in terms of a1 and b1 in equations 3.14 and 3.15. The filter open loop gain Klf is given in equation 3.16.

$$H(z) = K_{lf}\frac{1}{1-z^{-1}}\frac{(1-b_1z^{-1})}{1-a_1z^{-1}} \quad \textbf{(3.13)}$$

$$a_1 = \frac{1}{1+w_pT} \quad \textbf{(3.14)}$$

$$b_1 = \frac{1}{1+w_zb_1T} \quad \textbf{(3.15)}$$

$$K_{lf} = \frac{T}{T/N}\frac{K}{K_v}\frac{w_p}{w_z}\frac{a_1}{b_1}T \quad \textbf{(3.16)}$$

On the other hand, plugging equation 3.12 into H(z), the frequency response in s domain is obtained in equation 3.17.

$$H(s) = K_{lf} \frac{1-b_1}{1-a_1} \frac{1}{sT} \frac{1+s\frac{b_1 T}{1-b_1}}{1+s\frac{a_1 T}{1-a_1}} \quad \textbf{(3.17)}$$

Zero (w_z) and pole (w_p) locations are given in equation 3.18 and 3.19,

$$w_z = \frac{1-b_1}{b_1 \times T} \quad \textbf{(3.18)}$$

$$w_p = \frac{1-a_1}{a_1 \times T} \quad \textbf{(3.19)}$$

Bandwidth w_o is calculated from equation 3.20.

$$w_o = \frac{w_p + w_z}{4Q}\left(1 + \sqrt{1 - \left(\frac{4Q}{w_p + w_z}\right)^2 w_p w_z}\right) \quad \textbf{(3.20)}$$

The proposed filter's transfer function is given in equation 3.21 calculated from equation 3.1 and 3.13. Filter gain is given in equation 3.22. a_1 and b_1 are given in equation 3.23 and 3.24 respectively.

$$H(z) = k_2(1 + k_1)alpha \frac{1}{1-z^{-1}} \frac{1-\frac{k_1}{1+k_1}z^{-1}}{1-(1-alpha)z^{-1}} \quad \textbf{(3.21)}$$

$$K_{lf} = k_2 \times (k_1 + 1) \times alpha \quad \textbf{(3.22)}$$

$$a_1 = 1 - alpha \quad \textbf{(3.23)}$$

$$b_1 = \frac{k_1}{1+k_1} \quad \textbf{(3.24)}$$

Open loop gain, K given in equation 3.3, can also be written in equation 3.25 now, and dt is TDC resolution.

$$\mathrm{K} = \frac{\mathrm{K_v}}{\mathrm{N\Delta tdel}} \frac{1-\mathrm{b_1 z^{-1}}}{1-\mathrm{a_1 z^{-1}}} \mathrm{K_{lf}} = \frac{\mathrm{K_v} \times \mathrm{Klf\_mult} \times \mathrm{Scale}}{\mathrm{N} \times \mathrm{dt}} \quad \textbf{(3.25)}$$

k_2 in equation 3.22 is given in 3.26,

$$k_2 = \frac{1-b_1}{1-a_1} K_{lf} \quad \textbf{(3.26)}$$

Finally, NΔtdel is given in equation 3.27, and M in the equation is number of ring oscillator stages.

$$N\Delta tdel = \frac{T}{2M} \quad \textbf{(3.27)}$$

In the equations, k_1 and k_2 coefficients are used for determining filter gain and zero position. K_{lf} is the open loop gain, wo is the bandwidth, w_z is the zero frequency, and wp

is the pole frequency, T is reference period which is 1/f_ref . Also, 1/Δtdel is the TDC gain and N is the divider ratio. Also, in a butterworth filter, $w_{co}=w_o$ and Q= 0.707 [42].

3.2. Design Methods of Filter

The flow diagram of the filter in equation 3.13 is given in **Figure 3.2**. z^{-1} term in the diagram represents delay for the pole and the zero, and in Verilog code it can be realized with D flip flops. As it can be seen from the formulas, it is quite easy to change the filter's coefficients which are basically pole and zero locations, open loop gain etc.

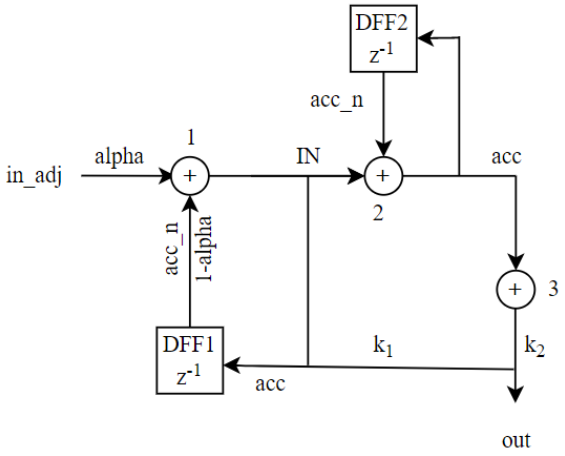

Figure 3.2: Block diagram of loop filter.

In the project, the goal is designing a programmable filter whose coefficients changes according to the 8 bits input tune. The different coefficients regarding to tune input is given in **Table 3.1** and **Table 3.2**.

Optimum f_o value, which gives the minimum noise in the system, was found by phase noise graph given by CPPsim. In CPPsim f_{ref} was given as 20MHz, TDC noise was given as -93 dBc/Hz, DCO noise was given as -111 dBc/Hz. In **Figure 3.3**, phase noise of PLL is given. To obtain minimum noise in PLL, the intersection point of VCO noise and detector noise, which is TDC noise, is taken as f_o frequency that is around 125 kHz. Low f_o values are desirable to maintain noise in the system because it suppress the sigma delta noise as it can be seen in **Figure 3.3**. On the other hand, low frequencies of f_o result in long lock time in PLL. In **Figure 3.4**, step response of PLL is given. It settles in 30µs which is a long time and not desirable.

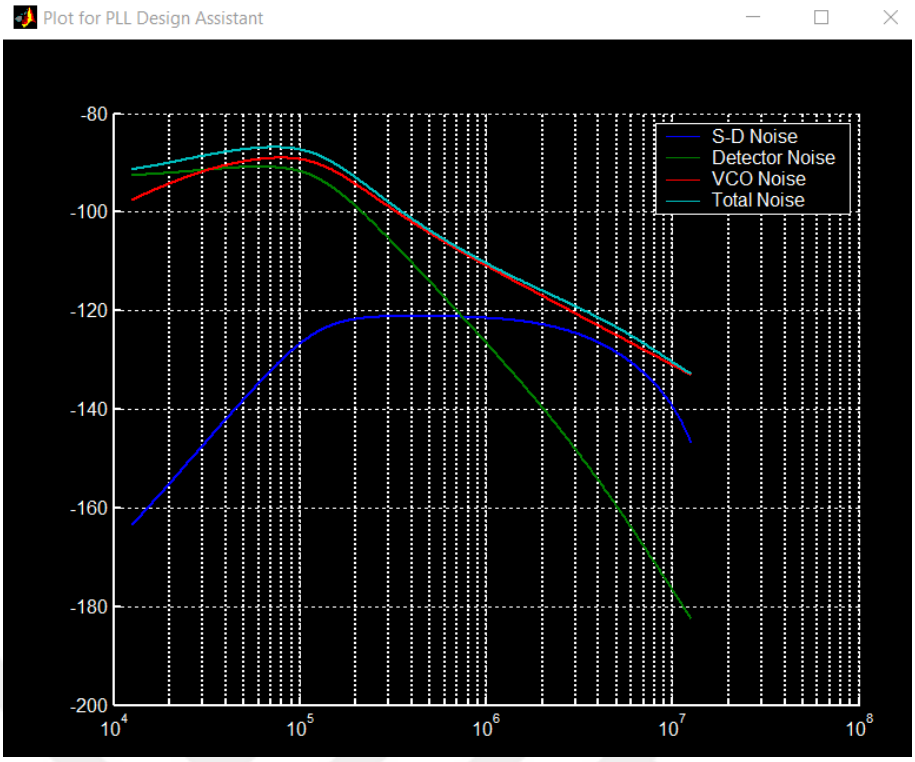

Figure 3.3: CPPsim noise analysis of PLL.

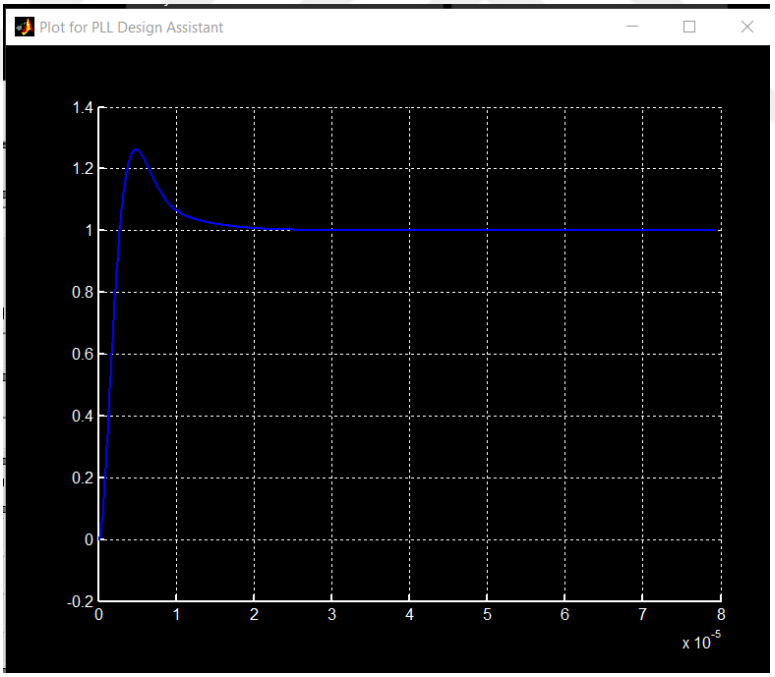

Figure 3.4: Step response of PLL for f_o=125kHz.

In **Figure 3.5**, step response of high f_o, which is 660kHz, is given. As it can be seen from **Figure 3.5**, lock time is around 6µs, which is 5 times lower than previous value. This is desirable for fast lock time of PLL. However, in this case the noise of the system is too high and not desirable.

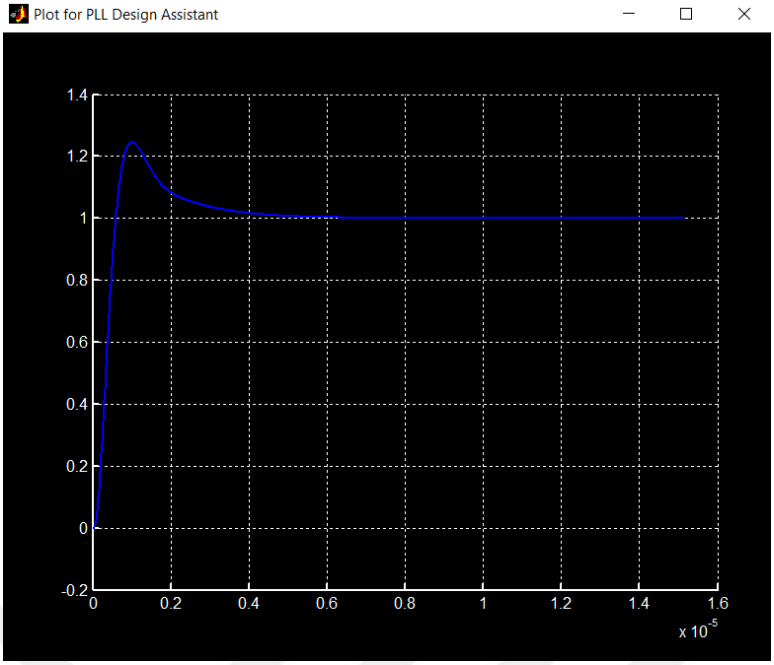

Figure 3.5: Step response of PLL for f_o=660kHz.

To have fast lock time and low sigma delta noise in the PLL, two filters were used in the system. Coarse filter is used for having a fast lock time in the system, and that is provided by high bandwidth. Fine filter is used for suppressing noise in PLL, and that is provided by low bandwidth. Zero and pole frequencies were calculated from k1 and alpha values, and these frequencies was chosen according to two concerns. First, f_z and f_p should give f_o from equation 3.20. Second, all the values of k1 and alpha was chosen as multiple or divisor of two to reduce the hardware cost because these values can be realized by shifting operations in Verilog, and there is no need to use multiplier in the system. Also, to find gain values, first of all PLL loop gain, K, was calculated from equation 3.3, and by equating this result to equation 3.25 gain of filter which is K_{LF}=Klf_mult×scale was obtained.

The first row of the **Table 3.1** was calculated, according to equate f_o to 660 kHz. First, k1 is decided as 32, alpha is 0.25 and f_{ref} is 20 MHz (T_{ref}=50 ns). b1 is calculated as 0.9696 from equation 3.15. Zero value w_z is calculated as 99 kHz from equation 3.18. Value a1 is 0.75 from equation 3.23, and w_p is calculated as 1061 kHz from equation 3.19.

K is calculated from 3.3 as 2.055×1012. This result was equated to equation 3.25. K_v is DCO gain 3.5×106. N is division ratio, 56 and dt, TDC resolution, is 22.26 ps. Therefrom, K_{LF}, gain of the filter, is 7.3×10^{-4}. In the design this value is represented as multiplication of Klf_mult and scale values. Klf_mult is chosen as 1, and scale is chosen as 2^{-10}. The

other rows of the table were calculated with same idea. Klf_mult value was chosen as 1, 5, 9 or 17 to obtain cost minimum hardware because these values can be obtained by shifting operations with 1 in Verilog, and there is no need to use multiplier in the system.

In **Table 3.1** and **Table 3.2**, filter coefficients according to different tune input is given. After obtaining all the values for different tune input values, Verilog code was built using these results following the block diagram in **Figure 3.2**.

Table 3.1: The constants of the fine filter for Q=0.707.

Tune	k_1	alpha	f_z(kHz)	f_p(kHz)	f_0(kHz)	f_z/f_0	Scale	K_{lf} Mult
2'b00	32	0.25	99	1061	660	0.15	2^{-10}	1
2'b01	64	0.125	50	455	274	0.18	2^{-11}	5
2'b10	128	0.0625	25	212	125	0.19	2^{-12}	9
2'b11	256	0.03125	13	102	65	0.20	2^{-13}	17

Table 3.2: The constants of the coarse filter for Q=0.707.

Tune	k_1	alpha	f_z(kHz)	f_p(kHz)	f_0(kHz)	f_z/f_0	Scale	K_{lf} Mult
2'b00	32	0.25	99	1061	660	0.15	2^{-15}	1
2'b01	64	0.125	50	455	274	0.18	2^{-16}	5
2'b10	128	0.0625	25	212	125	0.19	2^{-17}	9
2'b11	256	0.03125	13	102	65	0.20	2^{-18}	17

DCO has 2 different varactors for fine and coarse tuning. For this reason, to be in accordance with both types of DCO tuning, two types of filters were designed as coarse and fine filters. The coarse filter requires wider bandwidth to cover incoming signals, and at required frequencies its function was degraded so it needs lower values of gain to maintain stability. While their logic and working principle is exactly the same as each other, their gains are different, meaning that only the scale module of the filter is different in the Verilog code. Loop filter is designed at 5 different modules in Verilog; lpf, zero, klf_multiplication, scale and trim modules, and Verilog codes were given in the appendix A. In lpf module, poles of the filter were calculated. DFF1 and summation number 1 in **Figure 3.2** were realized in lpf module. The block diagram of lpf module is given in

Figure 3.6. Delay, z^{-1}, were obtained by D-flipflop. Summation number 1 in **Figure 3.2**, which is responsible for calculating coefficients of the poles, were realized by a multiplexer. 14 bits input came to the filter from the TDC's output. The input is expanded to 24 bits by adding redundant bits. The coefficients change only according to first two bits of the tune. For example, if tune is 00, alpha becomes 0.25. In each case of tune, the operation is the same in which input signal IN_ADJ is multiplied by alpha, and delayed version of input is multiplied by 1-alpha and these two results are added to obtain the output. Addition and multiplication operations were realized by shifting operators in Verilog code to avoid unnecessarily large sizes that is caused by many multiplication and addition gates.

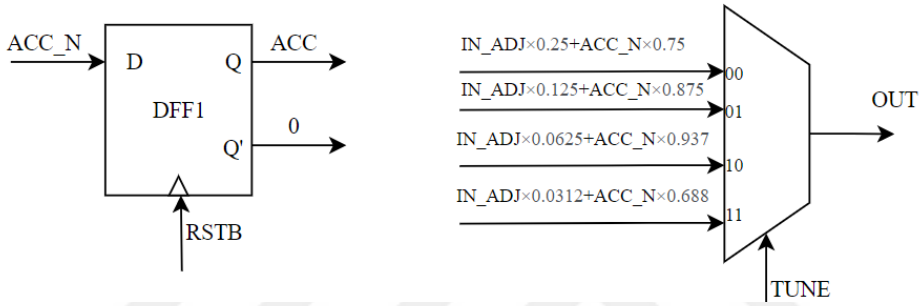

Figure 3.6: Block diagram of lpf module.

The second Verilog block zero1 is built for calculating the zeros. The output of the lpf is the input of this block. The second Verilog module zero1 was built for calculating the zeros. The output of lpf is the input of this block. DFF2 and summation number 2 and 3 in **Figure 3.2** were realized by zero module. The block diagram of zero module is given in **Figure 3.7**. Delay, z^{-1}, were obtained by D-flipflop. Input was multiplied by k1 coefficient according to different tune values, and it is done by a multiplexer. Summation number 2 and 3 in **Figure 3.2**, which is responsible for calculating coefficients of the zeros, were realized by adding IN, with_k1 and acc_n. Therefore, output of zero1 which is the numerator of equation 3.13, was obtained.

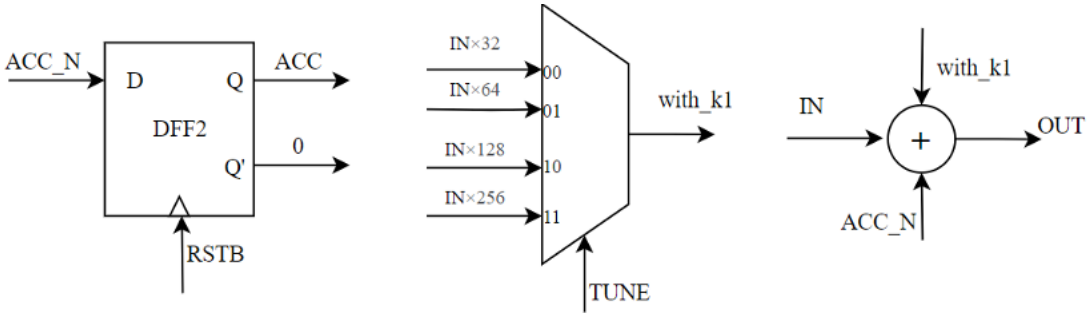

Figure 3.7: Block diagram of zero module.

Other modules in the filter are klf_multiplication and scale modules. These two modules are built for the same purpose which is deciding Klf value in the given formulas. The output of zero1 module is input to klf_multiplication. According to tune, input is multiplied by 1, 5, 9 or 17 in klf_multiplication, and the output is obtained. This operation was obtained by multiplexer, and the block diagram is given in **Figure 3.8.**

Again, these operations are realized by shifting operators in Verilog to avoid excessive hardware.

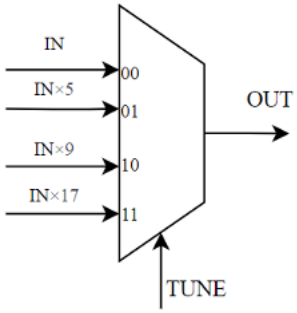

Figure 3.8: Block diagram of klf_multiplication module.

The output of klf_multiplication is input of scale module. In this module, again, according to tune, input is multiplied by 2^{-15}, 2^{-16}, 2^{-17}, 2^{-18}. In this way, K_{lf} can be controlled and it can have 16 different values. This operation was obtained by multiplexer, and the block diagram is given in **Figure 3.9**.

As it was mentioned, all the modules of the fine filter are same with coarse filter except scale module. In the fine filter design, scale input, which is output of klf_multiplication module, is multiplied by 2^{-10}, 2^{-11}, 2^{-12}, 2^{-13}. This operation was obtained by multiplexer, and the block diagram is given in **Figure 3.10**.

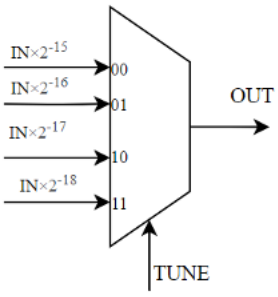

Figure 3.9: The block diagram of scale module in fine filter.

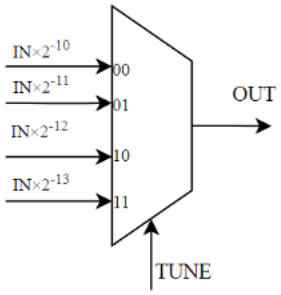

Figure 3.10: The block diagram of scale module in coarse filter.

The input of the trim module is the output of the scale module. Trim's flow diagram is given in **Figure 3.11**. The goal of trim is mapping output from 0 to 256.

The output of the filter module is 8bits, and output is from 0 to 28. The most significant bit of the scale module output is the sign bit. Least significant bits are trimmed by the trim module, so that the output is mapped from 0 to 256. If the input is less than -128, the output is mapped to 0, and if the input is greater than 127, it is mapped to 255.

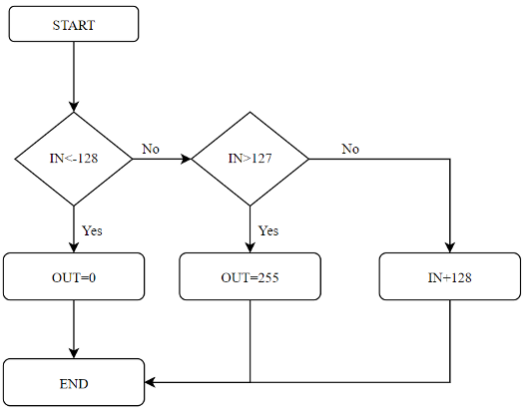

Figure 3.11: Flow diagram of trim module.

The results and outputs of the filter Verilog codes are given in results section.

Once the filter design was finalized in Verilog, the codes were imported to Cadence using the place and route tool to get the gate level schematics.

While testing the coarse filter in Cadence, filter inputs are added as it is shown in **Figure 3.12**. Tune inputs were buffered for signal integrity. As a reset input, ANDed RSTB and inverted PH_LOCK signal was used. Each input bit of the filter goes through and operation with reset input of the filter by and gate. The idea in that is saving power, when rstbf input is zero, result of and gate is zero therefore filter is turned off. Finally, the clock signal, CK, of the both filters was obtained by latch-and based integrated clock gating. Clock gating is used to reduce dynamic power dissipation. Two filters are used in the

design, coarse and fine filter, and clock gating mechanism allows one filter to be operational while the other is turned off. The clock gating circuit design for fine and coarse are given in **Figure 3.12 and 3.13**. Latch is used with AND gate to obtain effect of positive edge triggered flip flop. Enable signal is PH_LOCK in the design, and to avoid propagation of the clock signal due to possible hazard in PH_LOCK, glitches, latch is used. The difference in clock gating of the fine and coarse filter is, enable signal PH_LOCK. In fine filter PH_LOCK is used as enable, and in coarse filter inverted PH_LOCK is used therefore only one of the filters are operating at the same time. Basically, if PH_LOCK is 0, coarse tuning filter is working, if PH_LOCK is 1, fine tuning filter is working.

In the top module, tune is 8 bits random input signal and each module takes different 2 bits of tune. Bit 0 and 1 go to lpf module to decide alpha, bit 2 and 3 go to zero module to decide k1 coefficient, bits 4 and 5 go to klf_multiplication module to decide numerator of the gain of the filter. Lastly, bits 6 and 7 go to scale module to decide denominator of the gain. In this way, all of the filter specifications, its gain, pole zero locations and filter response, can be decided independently and freely according to needs by applying required tune input.

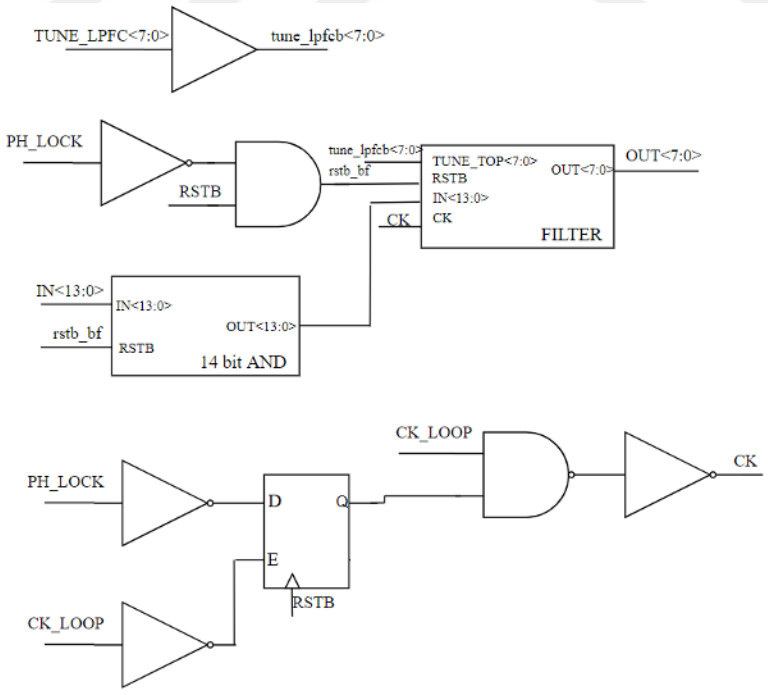

Figure 3.12: The top module schematic of the coarse loop filter.

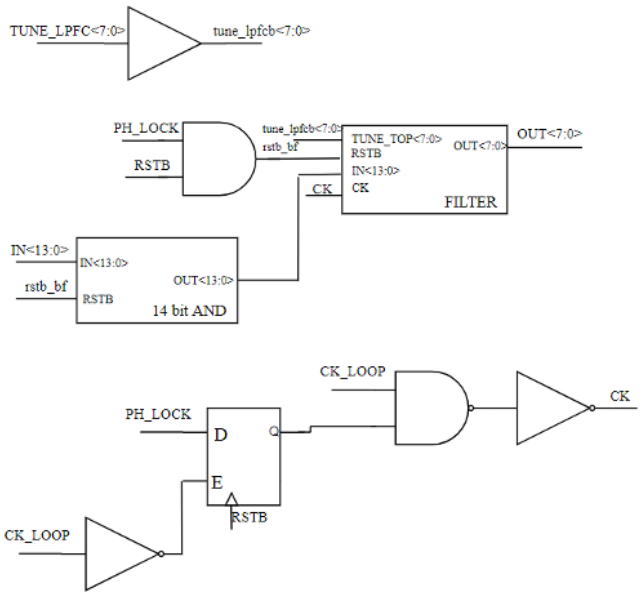

Figure 3.13: The top module schematic of the fine loop filter.

Coarse filter top module is shown in **Figure 3.12**, and fine filter top module is shown in **Figure 3.13**.

3.3. Design Methods of LDO

First, error amplifier of an LDO was designed. Error amplifier design considerations were given in **Table 3.3.** The circuit was designed in 180nm technology with 3.3V power supply. Phase margin greater than 45 degrees is necessary for stability, and bandwidth higher than 40MHz is required to cover the input signal that is coming from TDC at 20MHz (Nyquist rate $2\times f_{TDC}$).

Table 3.3: Error amplifier design considerations.

Technology (nm)	Phase margin (degrees)	Power supply (V)	Bandwidth (MHz)
180	> 45	3.3	> 40

The block diagram of error amplifier is given in **Figure 3.14**. The design consists of differential amplifier and common source amplifier. 2 stage amplifier configuration was used to boost the gain. Circuit design of error amplifier is given in **Figure 3.15**.

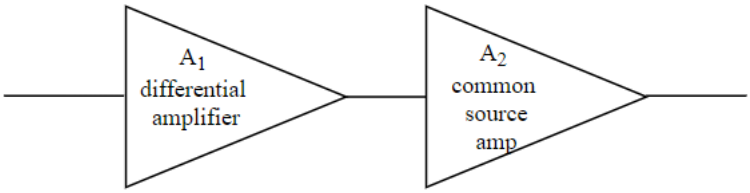

Figure 3.14: Block diagram of error amplifier.

The miller compensation technique was used for pole splitting by moving dominant pole to lower frequencies, therefore introducing a dominant pole to the system [43]. Miller capacitance, Cc, introduces large input capacitance, and introduces output capacitance equal to Cc. Therefore, dominant pole in the circuit is obtained. Cc also introduces right half plane zero (RHP) to the system. Phase shift and the magnitude are increased by RHP zero, therefore stability is poorly affected by RHP, and its effect cannot be ignored because this RHP zero gives higher loop gain magnitude with more negative loop phase shift. This problem can be controlled and eliminated with 2 approaches, first of them is to place a unity gain buffer at the feedback path, or to place a nulling resistor in series with the miller capacitor. In the proposed work, nulling resistor was placed in series with the miller capacitance to solve the problem. The nulling resistor is able to independently control the zero location. As a result, this method provides sufficient stability.

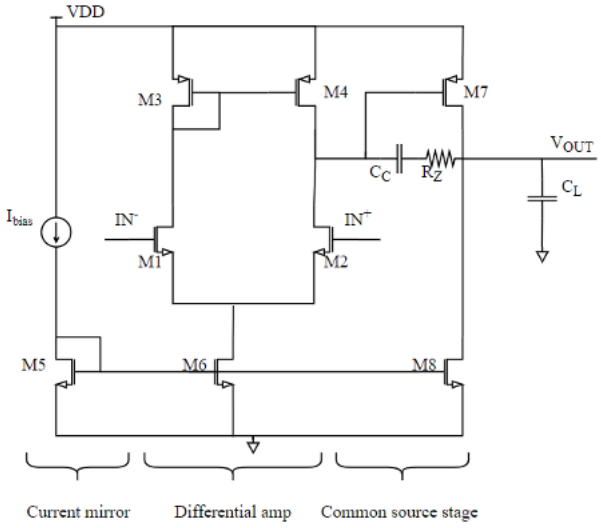

Figure 3.15: Error amplifier circuit design.

The error amplifier gain is given in equation 3.28.

$$A_v = A_1 \times A_2 \tag{3.28}$$

The first and second stage gains are given in equation 3.29 and 3.30 respectively.

$$A_1 = -g_{m1} \times (r_{o1} // r_{o2}) \tag{3.29}$$

$$A_2 = -g_{m6} \times (r_{o6} // r_{o7}) \quad \textbf{(3.30)}$$

Dominant pole location is given in equation 3.31.

$$f_{p1} = \frac{1}{2\pi R_1 g_{m2} R_2 C_C} \quad \textbf{(3.31)}$$

Second pole location is given in equation 3.32.

$$f_{p2} = \frac{g_{m2}}{2\pi C_2} \quad \textbf{(3.32)}$$

Zero location is given in equation 3.33.

$$f_{z1} = \frac{g_{m2}}{2\pi C_C} \quad \textbf{(3.33)}$$

R_1 and R_2 is given in equation 3.34 and 3.35 respectively.

$$R_1 = r_{o1} // r_{o2} \quad \textbf{(3.34)}$$

$$R_2 = r_{o6} // r_{o7} \quad \textbf{(3.35)}$$

C_1 and C_2 are given in equation 3.36 and 3.37.

$$C_1 = C_{gd2} + C_{db2} + C_{gd4} + C_{db4} + C_{gs6} \quad \textbf{(3.36)}$$

$$C_2 = C_{db6} + C_{db7} + C_{gd7} + C_L \quad \textbf{(3.37)}$$

The design parameters of error amplifier are given in **Table 3.4.**

Table 3.4: The design parameters of op-amp.

Devices	W(µ)	L(µ)	Finger	Multiplier
M1	10	1	10	2
M2	10	1	10	2
M3	6	1	5	2
M4	6	1	5	2
M5	4	1	1	1
M6	4	1	16	2
M7	6	1	10	2
M8	4	1	32	1

In LDO design, output voltage was fed back to input of error amplifier by feedback network. Fed back voltage was compared with the reference voltage, and error voltage was generated according to the difference. The LDO design was given in **Figure 3.16.**

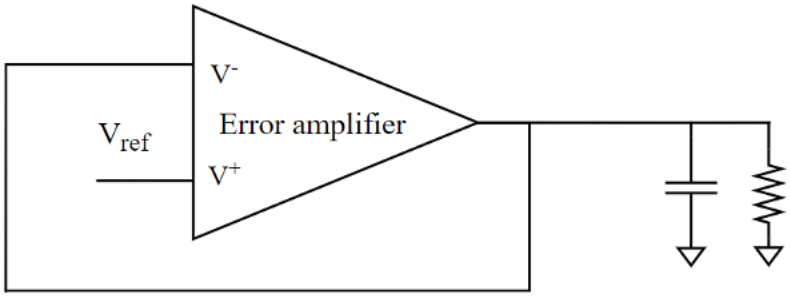

Figure 3.16: LDO design configuration.

In the PLL, this LDO is used to regulate the supply of the DCO. LDO has to work properly at DCO frequencies of 400MHz and 560MHz, and it should not degrade the Phase Noise of the DCO by more than 3dB.

The LDO is tested in 3 different conditions. First, it was tested in the worst-case scenario with 10pF capacitance and 1K load resistance. In worst case scenario, the output impedance was increased, and pole splitting is degraded, meaning that two poles came closer to each other. Secondly, the LDO was tested with DCO block when it is oscillating at 400MHz and at 560MHz for transient result and for noise contribution as in **Figure 3.17**. At each condition, PVT corner simulations were done. DCO phase noise analysis was also done when it was fed with the LDO. Also, phase margin, bandwidth, gain, supply and total current results are collected for PVT corners. All these results are given in the results part of the thesis. Finally, layout of the LDO was finished and PEX analysis was done to observe nonideal effects of the parasitics in the functioning of the circuit. These results are also given in the results part.

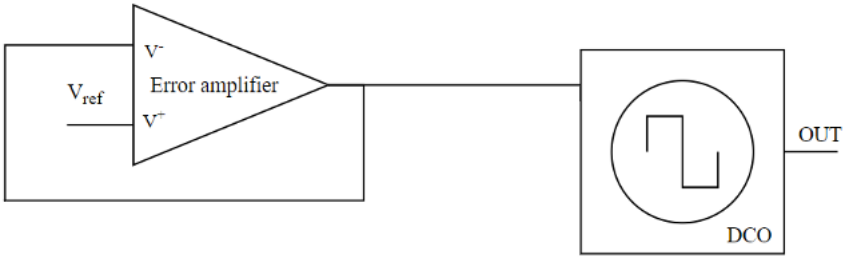

Figure 3.17: Testbench of LDO with DCO.

CHAPTER 4

4. RESULTS AND DISCUSSION

4.1. Results of Digital Filter

To test the filter, T=5000ns square signal is applied, and output signal, out_total, is plotted in Verilog simulation. From **Figure 4.1** and **Figure 4.2**, low pass characteristics of the filter can be observed. High frequency components of the square input signal were filtered out by designed loop filter. Filter output waveform is changing according to the relationship between period of input signal (T) and period of filter (Tf). If Tf > T, square input wave is given in **Figure 4.1**, filter time constant is Tf = 2.79µs, and it means input signal's period is larger than Tf (T>Tf). If T ≥ Tf , square input wave is converted to triangular wave. This can be seen in **Figure 4.2,** where Tf = 4.7µs, which is almost equal to T, and it gives triangular shape output.

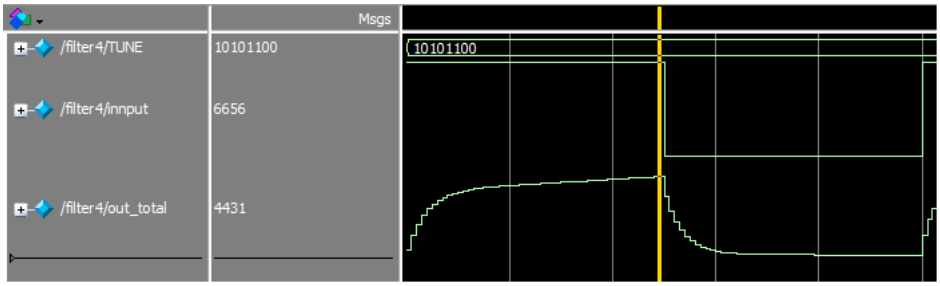

Figure 4.1: Fine filter output signal when tune is 10101100.

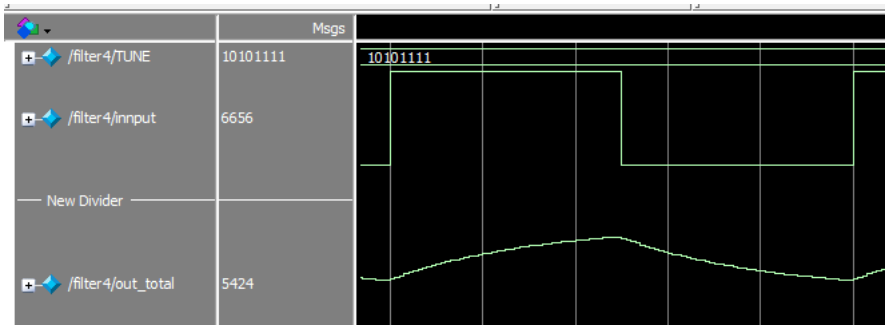

Figure 4.2: Fine filter output signal when tune is 10101111.

Layout of the coarse filter and fine filter are given in **Figure 4.3** and **Figure 4.4**.

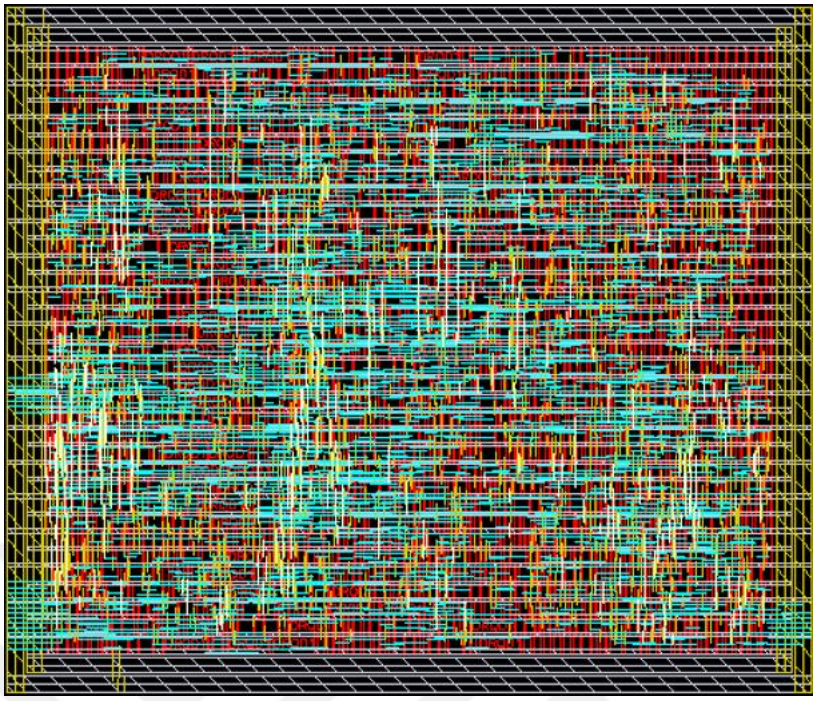

Figure 4.3: Layout of coarse filter.

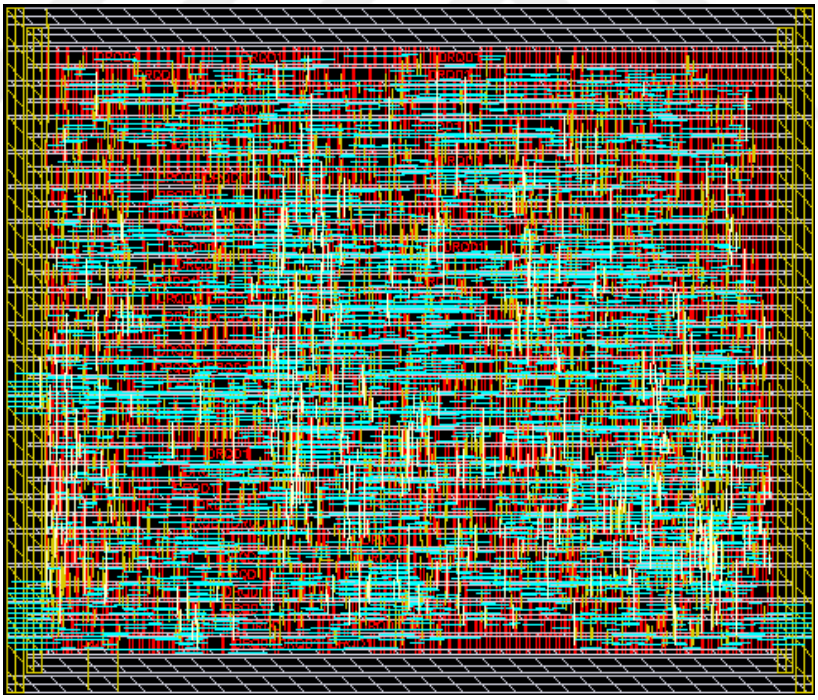

Figure 4.4: Layout of fine filter.

4.2. Results of LDO

4.2.1. Results of the error amplifier

The analysis of the error amplifier was done in Cadence. First, the error amplifier was tested in worst case scenario, and design configuration is given in **Figure 3.14**. The AC response is given in **Figure 4.5**, and bode plot of the loop is given in **Figure 4.6**. In **Table 4.1**, results for worst case scenario is given at nominal condition and PVT corners. In typical conditions phase margin is 57 degrees, bandwidth is 70MHz and gain is 61dB. Current that passes through M7 is 2mA.

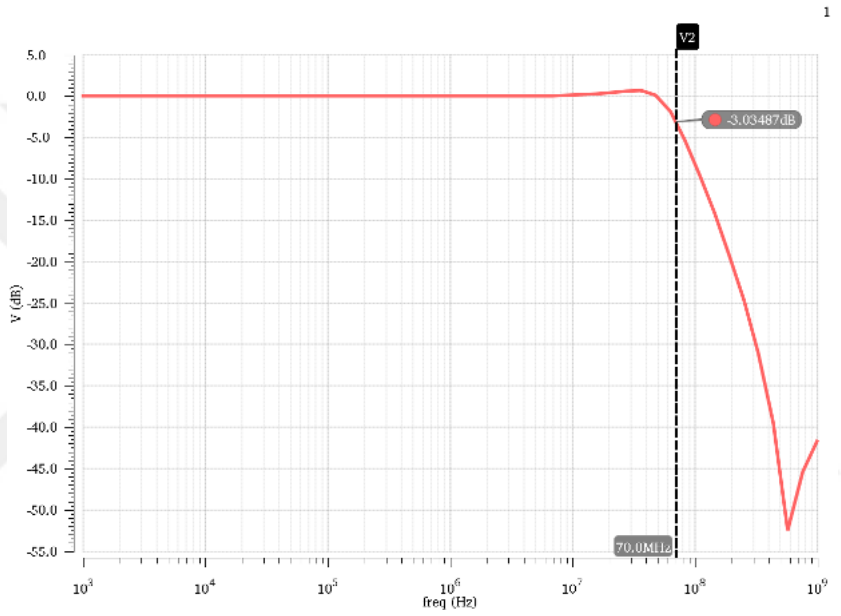

Figure 4.5: AC response of the circuit in worst case scenario.

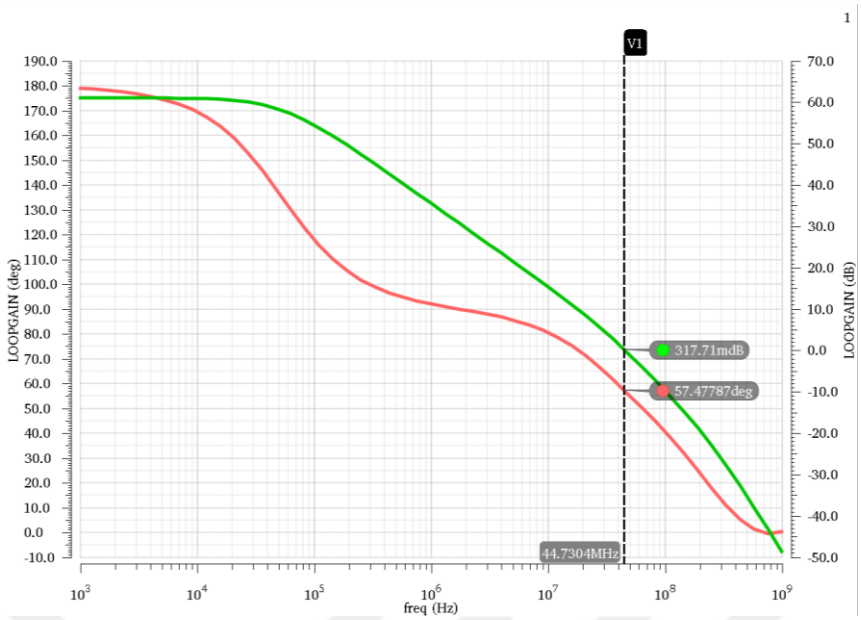

Figure 4.6: Bode plot of circuit in worst case scenario.

Table 4.1: Results of circuit at nominal conditions and PVT corners in worst case scenario.

	TYPICAL	FAST FAST	FAST FAST	SLOW SLOW
Temp(°C)	27	-41	-200	71
PM (degrees)	57	61	40	50
Bandwidth (MHz)	70	100	462	59
Gain (dB)	61	65	76	56
Supply (V)	3.3	3.63	3.63	2.97
Total Current (mA)	2	2.3	2.3	1.88

Power supply rejection ratio (PSRR) graph of LDO is given in **Figure 4.7**. PSRR shows LDO's rejection of the ripples seen in the input. In **Figure 4.7** the attenuation amount provided by LDO at different frequencies is given. At low frequencies, PSRR is -61 dB, it means LDO can successfully reject ripples from input. The worst PSRR, which is -4.45 dB, is obtained at 64 MHz.

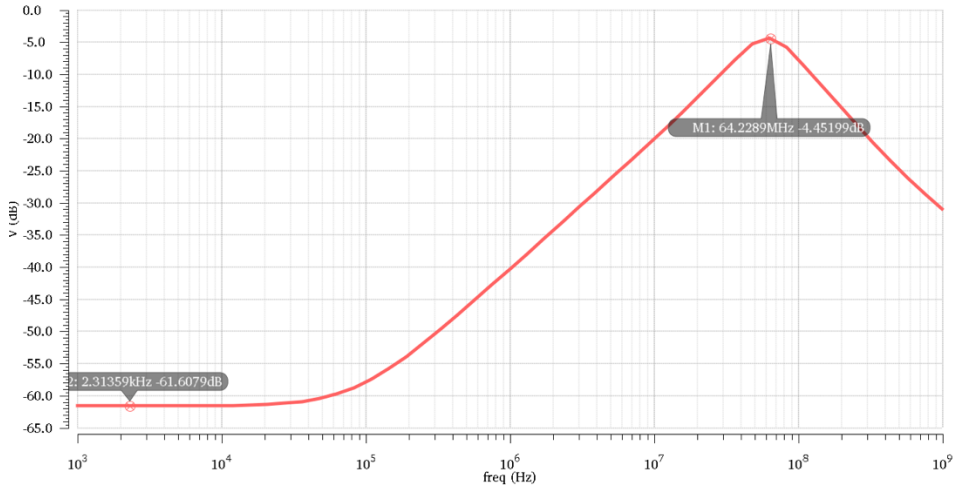

Figure 4.7: PSRR graph of LDO.

4.2.2. Results of the LDO

LDO was tested with the DCO, where transient output voltage of the circuit and LDO's impact on the noise performance was observed. These results were given in this section. In **Figure 4.8**, transient output response of LDO is given. The voltage variation at the output was caused by sudden increase in the load current that is shown in **Figure 4.9**. The transient voltage until 2.4ns is 1.8V and it was constant because the load current was constant until 2.4ns. However, right after 2.4ns, load current has increased suddenly, and as a result sharp drop at transient voltage, 0.06V, was seen because LDO's negative feedback loop cannot operate as fast as load current change.

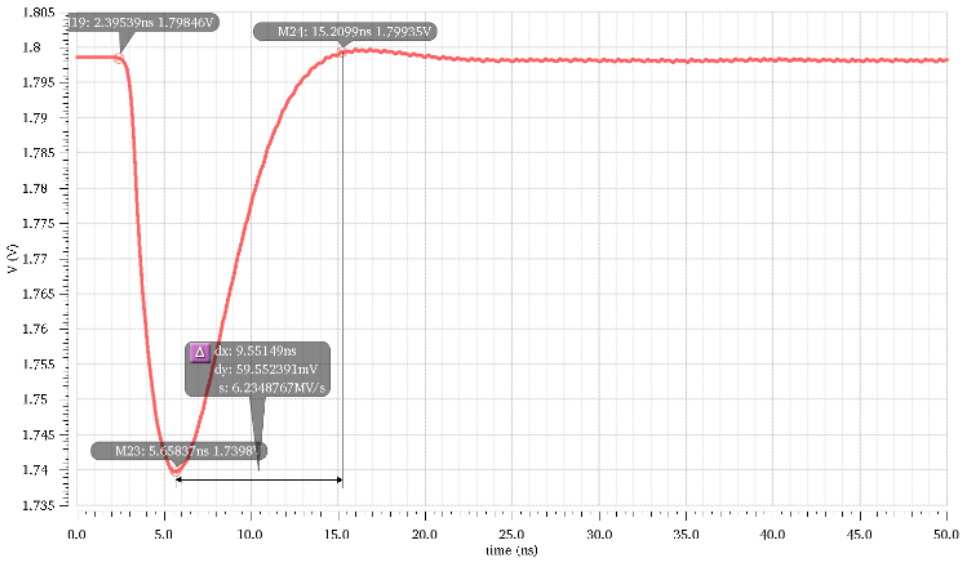

Figure 4.8: Transient voltage output of LDO.

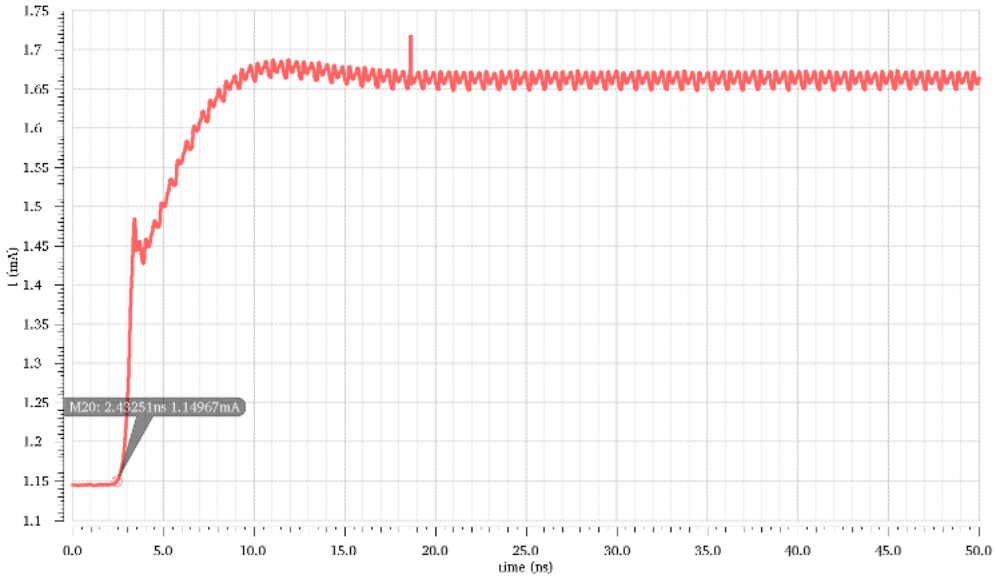

Figure 4.9: Transient current in LDO.

Moreover, in **Figure 4.10** transient response of LDO to higher load current value was given. Load current was given in **Figure 4.11**. Higher voltage variation at the output was obtained because the load current is higher in this case.

The output transient voltage dropped by 0.09V, this is higher than previous case because in this case LDO's negative feedback loop needed to respond to a higher current value.

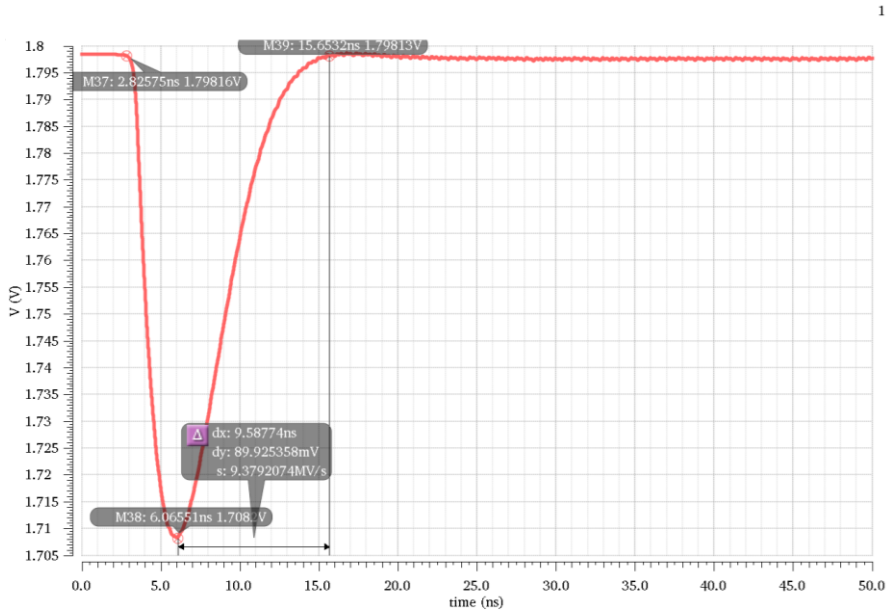

Figure 4.10: Transient voltage output of LDO.

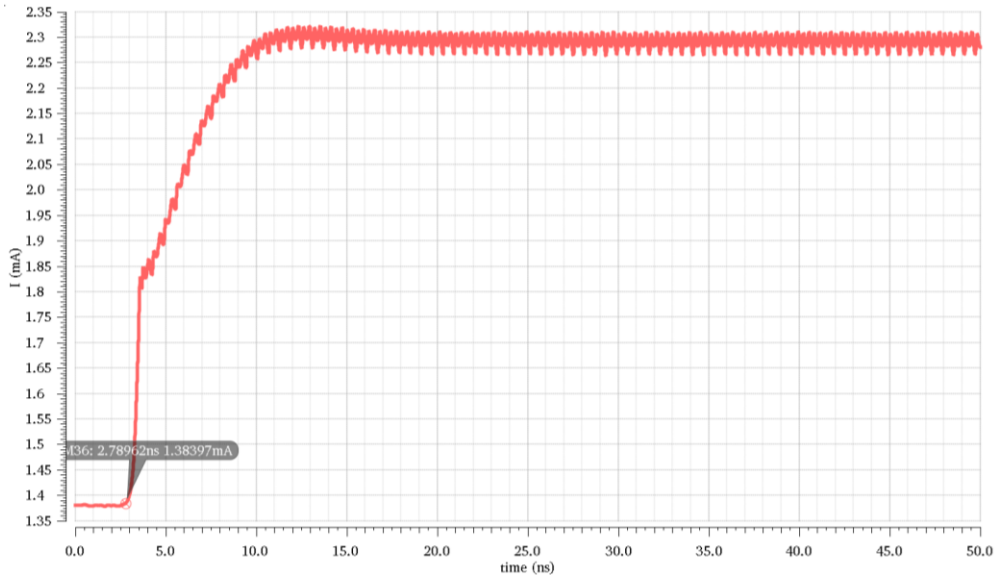

Figure 4.11: Transient current in LDO.

Moreover, the noise analysis was given from 10kHz to 21MHz to observe the noise effect of the LDO on the DCO. In **Figure 4.12**, the noise values for DCO block and noise values for DCO with LDO were given. The noise values at each frequency were compared, and the LDO added no more than 2dB noise to the system.

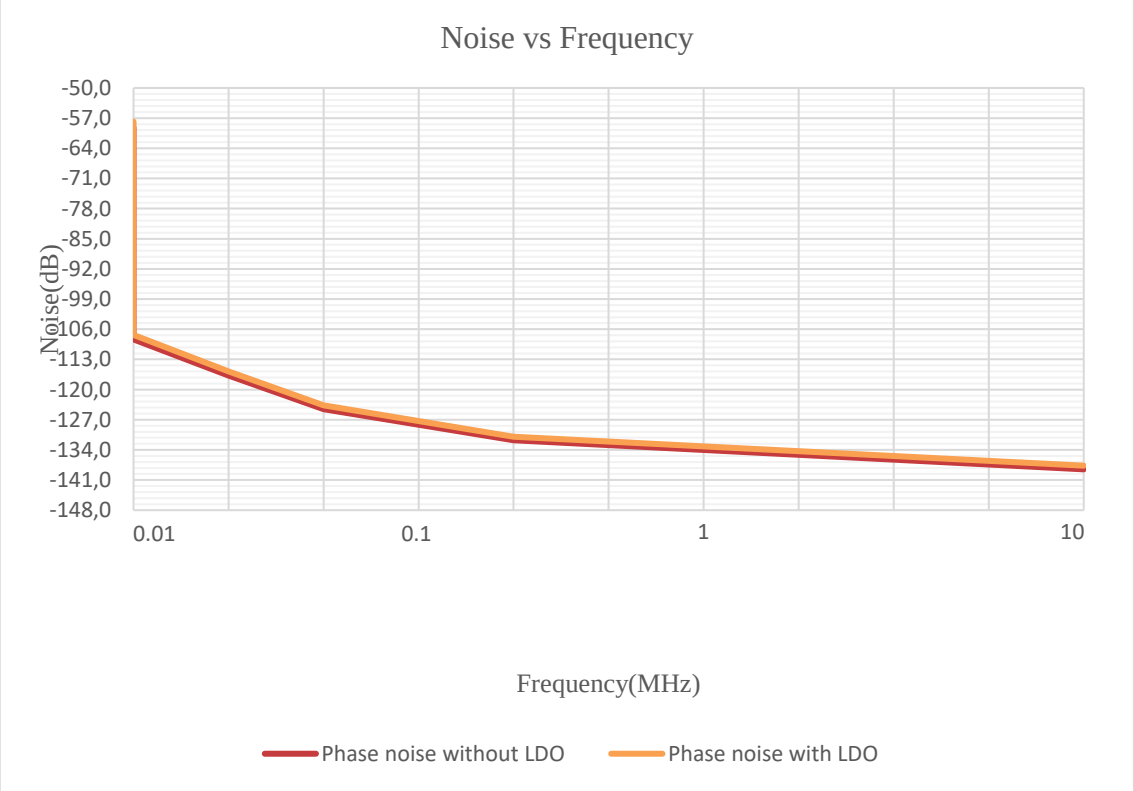

Figure 4.12: Noise vs frequency graph of LDO with DCO.

Finally, the layout drawing of the LDO circuit was prepared. While making layouts, the area has been kept as small as possible and the metals of the nodes with excessive current are made thick. Layout configuration is given in **Figure 4.13**.

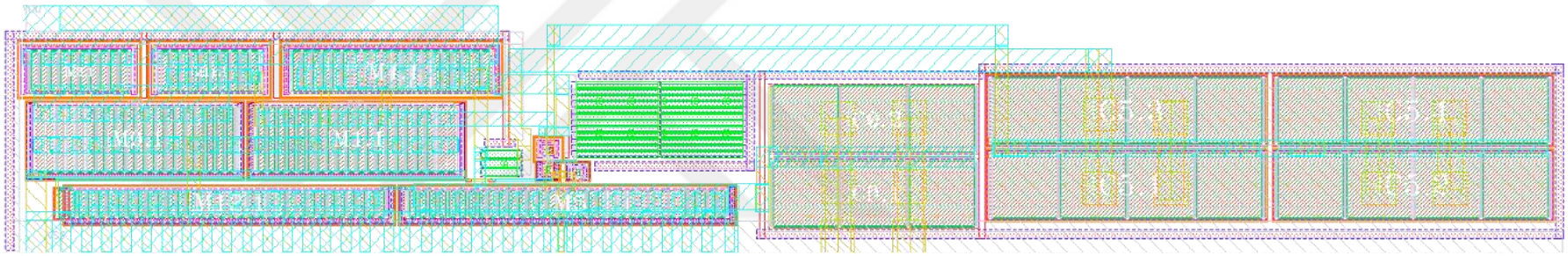

Figure 4.13: LDO layout schematic.

CHAPTER 5

5. CONCLUSIONS AND FUTURE WORK

In this thesis, the design and implementation of 14-bit digital filter and analog LDO design for ADPLL in 180nm CMOS technology is presented. Thesis consist of 5 Chapters. The first chapter starts with introduction of the thesis. The 2nd Chapter consist of literature review of PLL, its subblocks, filter, digital filter, its types, finally advantages and disadvantages of digital filter. The 3rd chapter consist of implementation and design methods of the proposed digital filter and LDO. In the 4th chapter results of digital filter and LDO are given.

Digital IIR filter is preferred in the project because it is very easy to change the specifications of the filter according to needs of the project, meaning that it is easy to program. IIR filter requires low memory, it has low cost and low latency.

The software of the filter design is completed in Verilog. Schematic design and layout are obtained in Cadence Virtuoso. Calibre tool in Cadence is used to complete DRC, LVS and PEX results.

LDO schematic and layout design are completed in Cadence Virtuoso. Mentor Calibre tool in Cadence is used to complete DRC, LVS and PEX results.

BIBLIOGRAPHY

[1] Mao-Fu Lai and M. Nakano, "Special section on phase-locked loop techniques," *IEEE Trans. Ind. Electron.*, vol. 43, no. 6, pp. 607–608, Dec. 1996.

[2] T. Olsson and P. Nilsson, "A digitally controlled PLL for SoC applications," *IEEE J. Solid-State Circuits*, vol. 39, no. 5, pp. 751–760, May 2004.

[3] *Pll Performance, Simulation and Design - Dean Banerjee - Google Kitaplar*. .

[4] R. E. Best, "Phase-Locked Loops," McGraw-Hill Education, 2007.

[5] M. Karimi-Ghartemani and M. R. Iravani, "A new phase-locked loop (PLL) system," in *Proceedings of the 44th IEEE 2001 Midwest Symposium on Circuits and Systems. MWSCAS 2001 (Cat. No.01CH37257)*, 2001, vol. 1, pp. 421–424.

[6] B. Razavi and R. F. Microelectronics, *Prentice Hall (1998), cover pages*. and.

[7] T. Shiva and M. R. Mahajan, "A Review Paper On Generalization and Parameter Variation of Different PLL structures," *Int. J. Adv. Inf. Sci. Technol. ISSN*, vol. 5, no. 1, 2016.

[8] S. Winder and S. Winder, "Analog and digital filter design," p. 450, 2002

[9] J. Wang, "Modeling And Implementation of All-Digital Phase-Locked Loop Based on Vernier Gated Ring Oscillator Time-to-Digital Converter."

[10] Y. Taur, "CMOS design near the limit of scaling," *IBM J. Res. Dev.*, vol. 46, no. 2–3, pp. 213–222, 2002.

[11] D. Buss, "Technology in the Internet Age," in *Digest of Technical Papers - IEEE International Solid-State Circuits Conference*, 2002, no. SUPPL., pp. 12-13+360.

[12] A. J. Annema, B. Nauta, R. Van Langevelde, and H. Tuinhout, "Designing outside rail constraints," in *Digest of Technical Papers - IEEE International Solid-State Circuits Conference*, 2003, vol. 47, pp. 96-97+466.

[13] Y. Park and D. D. Wentzloff, "An all-digital PLL synthesized from a digital standard cell library in 65nm CMOS," in *Proceedings of the Custom Integrated Circuits Conference*, 2011.

[14] W. Grollitsch, R. Nonis, and N. Da Dalt, "A 1.4psrms-period-jitter TDC-less fractional-N digital PLL with digitally controlled ring oscillator in 65nm CMOS," in *Digest of Technical Papers - IEEE International Solid-State Circuits Conference*, 2010, vol. 53, pp. 477–479.

[15] M. S. W. Chen, D. Su, and S. Mehta, "A calibration-free 800 MHz fractional-N digital PLL with embedded TDC," in *IEEE Journal of Solid-State Circuits*, 2010, vol. 45, no. 12, pp. 2819–2827.

[16] A. V. Rylyakov, J. A. Tierno, D. Z. Turker, J. O. Plouchart, H. A. Ainspan, and D. Friedman, "A modular all-digital PLL architecture enabling both 1-to-2GHz and 24-to-32GHz operation in 65NM CMOS," in *Digest of Technical Papers - IEEE International Solid-State Circuits Conference*, 2008, vol. 51, pp. 516–518.

[17] J. A. Tierno, A. V. Rylyakov, and D. J. Friedman, "A Wide Power Supply Range,

Wide Tuning Range, All Static CMOS All Digital PLL in 65 nm SOI," *IEEE J. Solid-State Circuits*, vol. 43, no. 1, pp. 42–51, 2008.

[18] E. Temporiti, C. Weltin-Wu, D. Baldi, M. Cusmai, and F. Svelto, "A 3.5 GHz wideband ADPLL with fractional spur suppression through TDC dithering and feedforward compensation," in *IEEE Journal of Solid-State Circuits*, 2010, vol. 45, no. 12, pp. 2723–2736.

[19] T. Tokairin, M. Okada, M. Kitsunezuka, T. Maeda, and M. Fukaishi, "A 2.1-to-2.8GHz all-digital frequency synthesizer with a time-windowed TDC," in *Digest of Technical Papers - IEEE International Solid-State Circuits Conference*, 2010, vol. 53, pp. 470–471.

[20] J. Borremans, K. Vengattaramane, V. Giannini, and J. Craninckx, "A 86MHz-to-12GHz digital-intensive phase-modulated fractional-N PLL using a 15pJ/shot 5ps TDC in 40nm digital CMOS," in *Digest of Technical Papers - IEEE International Solid-State Circuits Conference*, 2010, vol. 53, pp. 480–481.

[21] C. M. Hsu, M. Z. Straayer, and M. H. Perrott, "A low-noise wide-BW 3.6-GHz digital ΔΣ fractional-N frequency synthesizer with a noise-shaping time-to-digital converter and quantization noise cancellation," in *IEEE Journal of Solid-State Circuits*, 2008, vol. 43, no. 12, pp. 2776–2786.

[22] J. Dunning, G. Garcia, J. Lundberg, and E. Nuckolls, "An All-Digital Phase-Locked Loop with 50-Cycle Lock Time Suitable for High-Performance Microprocessors," *IEEE J. Solid-State Circuits*, vol. 30, no. 4, pp. 412–422, 1995.

[23] I. Hwang, S. Lee, S. Lee, and S. Kim, "A digitally controlled phase-locked loop with fast locking scheme for clock synthesis application," in *Digest of Technical Papers - IEEE International Solid-State Circuits Conference*, 2000, pp. 168–169.

[24] T. Y. Hsu, C. C. Wang, and C. Y. Lee, "Design and analysis of a portable high-speed clock generator," *IEEE Trans. Circuits Syst. II Analog Digit. Signal Process.*, vol. 48, no. 4, pp. 367–375, Apr. 2001.

[25] F. Andersson and A. Johnson, "Modeling and Characterization of All-Digital Phase-Locked Loop," 2010.

[26] B. Porat, "A Course in Digital Signal Processing," p. 632, 1997.

[27] W. Stapelbroek, "Higher order feedback loop for a pulse width modulator," 2006.

[28] R. V. Ravi, K. Subramaniam, T. V. Roshini, S. P. B. Muthusamy, and G. K. D. Prasanna Venkatesan, "Optimization algorithms, an effective tool for the design of digital filters; a review," *J. Ambient Intell. Humaniz. Comput.*, vol. 1, p. 3, Sep. 2019.

[29] G. C. Hsieh and J. C. Hung, "Phase-locked loop techniques - A survey," *IEEE Trans. Ind. Electron.*, vol. 43, no. 6, pp. 609–615, 1996.

[30] *Mixed-signal and DSP Design Techniques - Analog Devices Inc., Engineeri, Analog Devices, inc - Google Kitaplar*. .

[31] H. Soni, D. Sharma, P. Dubey, and M. Tech, *FINITE IMPULSE RESPONSE FILTER DESIGN AND THEIR APPLICATION: A REVIEW*. .

[32] A. Roychowdhury, “FIR Filter Design Techniques.”

[33] R. Pal, “Comparison of the design of FIR and IIR filters for a given specification and removal of phase distortion from IIR filters,” in *International Conference on Advances in Computing, Communication and Control 2017, ICAC3 2017*, 2018, vol. 2018-Janua, pp. 1–3.

[34] M. Day, “Understanding Low Drop Out (LDO) Regulators.”, 2006.

[35] A. Castaldo, “Switching regulator fundamentals Application Report Switching regulator fundamentals,” 2012.

[36] Hanh, P. Le, and P. K. Wright, “Design Techniques for Fully Integrated Switched-Capacitor Voltage Regulators,” 2013.

[37] S. Maniktala, *New York Chicago San Francisco Athens London Madrid Mexico City Milan New Delhi Singapore Sydney Toronto Second Edition Switching Power Supply Design & Optimization*. McGraw-Hill Education, 2014.

[38] J. A. Valle-Mayorga, A. Rahman, and H. A. Mantooth, “A SiC NMOS linear voltage regulator for high-temperature applications,” *IEEE Trans. Power Electron.*, vol. 29, no. 5, pp. 2321–2328, May 2014.

[39] J. M. Hinojo, C. Luján Martínez, and A. Torralba, *Internally Compensated LDO Regulators for Modern System-on-Chip Design*. Springer International Publishing, 2018.

[40] *CMOS Analog Circuit Design - Phillip E. Allen, Douglas R. Holberg - Google Kitaplar*. .

[41] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing Principles, ~ l ~ o r i t h m i , and Applications Third Edition*. .

[42] M. H. Perrott, “Tutorial on Digital Phase-Locked Loops CICC 2009,” 2009. .

[43] R. Gomez, “ScholarWorks@UARK Design of Two-Stage Operational Amplifier using Indirect Feedback Frequency Compensation.”

APPENDIX A

Appendix A1

Coarse Filter Verilog Code

```
module lpf2 (OUT, IN, TUNE, CK, RSTB) ;
 output  signed [23:0] OUT ;
 input signed [13:0] IN;
 input [1:0] TUNE;
 input CK;
 input RSTB;
 reg signed [23:0] ACC, ACC_N,TEMP1;
 reg signed [23:0] OUT ;
 wire signed [23:0] IN_ADJ;
// pole
 assign IN_ADJ = {IN,10'b0};
 always @(posedge CK or negedge RSTB)
  if (RSTB == 1'b0)
   ACC_N <= 0;
  else
   ACC_N <= ACC;
 always @(*)
  case (TUNE) //tune'a göre pole belirleme
  2'b00 : TEMP1 = (IN_ADJ-ACC_N)>>>2;
  2'b01 : TEMP1 = (IN_ADJ-ACC_N)>>>3;
  2'b10 : TEMP1 = (IN_ADJ-ACC_N)>>>4;
  2'b11 : TEMP1 = (IN_ADJ-ACC_N)>>>5;
  endcase
always@(*) begin
ACC=TEMP1+ACC_N;
end
  always@(*)
  begin
   OUT = ACC;
  end
```

```
 endmodule

module zero12 (OUT, IN, CK, RSTB, TUNE) ;
 output reg signed [32:0] OUT ;
 input signed [23:0] IN;
 input CK ;
 input RSTB ;
 input[1:0] TUNE;
 reg signed [32:0] acc, acc_n, with_k1;

 always @(posedge CK or negedge RSTB)
  if (RSTB == 1'b0)
   acc_n <= 0;
  else
   acc_n <= acc;
// input signal*k1
always@(*) begin
 case(TUNE)
 2'b00: with_k1=(IN<<<5);
 2'b01: with_k1=(IN<<<6);
 2'b10: with_k1=(IN<<<7);
 2'b11: with_k1=(IN<<<8);
 endcase end
 always @(*) begin
  acc = IN + acc_n;
  OUT = acc+ with_k1;
  // overflow protection for integrator
  if (acc > $signed(32'h3FFF_FFFF))
   acc = $signed(32'h3FFF_FFFF) ; // most positive 31b number
  else if (acc < $signed(32'hC000_0000))
   acc = $signed(32'hC000_0000) ; // most negative 31b number
 end
endmodule

module klf_multiplication2(IN,OUT,TUNE);
```

```
input signed[32:0] IN;
input [1:0] TUNE;
output reg signed[37:0]OUT;
always@(*)
 begin
 case(TUNE)
 2'b00: OUT=(IN<<<0);
 2'b01: OUT=((IN<<<2)+IN);
 2'b10: OUT=((IN<<<3)+IN);
 2'b11: OUT=((IN<<<4)+IN);
endcase end
endmodule
module ascale12(IN,OUT,TUNE);
input signed[37:0] IN;
input [1:0] TUNE;
output reg signed[12:0]OUT;
always@(*)begin
 case(TUNE)
 2'b00: OUT={IN[37],IN[36],IN[35],IN[34:25]};
 2'b01: OUT={IN[37],IN[37],IN[36],IN[35:26]};
 2'b10: OUT={IN[37],IN[37],IN[37],IN[36:27]};
 2'b11: OUT={IN[37],IN[37],IN[37],IN[37:28]};
endcase
end
endmodule

module trim2 (OUT, IN);
 output reg [7:0] OUT;
 input signed [12:0] IN;
 always @(*) begin
  if (IN<-128)
   OUT=0 ;
  else if(IN>127)
   OUT=255 ;
  else
```

```
   OUT={~IN[12],IN[6:0]}; //in + 8'd128
 end
endmodule

//coarse filter
`timescale 1ns/1ns
module filter3 ();
reg clk;
wire [13:0] d1 = 14'b01101000000000;
wire [23:0]out_p;
wire [32:0]out_z;
wire [12:0]out_total;
wire [7:0] out_trim;
wire [37:0] out_klf;
reg [13:0] innput;
reg rstb;
reg clk2;
wire [7:0] TUNE;
initial begin
 clk = 1 ;
 clk2=1;
end
initial begin
 rstb=0;
 #20;
 rstb=1;
end
// clock generator
always #5000 clk = ~clk ;
//assign a=d1;
always #50 clk2=~clk2;
always@(*)
 innput<=clk*d1;
lpf2 u_lpf333 (.OUT(out_p), .IN(innput), .TUNE(TUNE[1:0]), .CK(clk2), .RSTB(rstb));
```

```
zero2 u_zero333 (.OUT(out_z), .IN(out_p), .TUNE(TUNE[3:2]), .CK(clk2), .RSTB(rstb));
klf_multiplication2 u_klf1333 (.OUT(out_klf), .IN(out_z), .TUNE(TUNE[5:4]));
ascale2 u_scale333 (.IN(out_klf), .OUT(out_total), .TUNE(TUNE[7:6]));
trim2 trim_u333 (.OUT(out_trim), .IN(out_total));
endmodule

// Library - Layout_filter, Cell - FILT_COURSE, View - schematic
// LAST TIME SAVED: Apr 21 16:01:03 2021
// NETLIST TIME: Apr 21 16:01:30 2021
`timescale 1ns / 1ps
module FILT_COURSE ( OUT, CK_LOOP, IN, PH_LOCK, RSTB, TUNE_LPFC);
input  CK_LOOP, PH_LOCK, RSTB;
output [7:0]  OUT;
input [7:0]  TUNE_LPFC;
input [13:0]  IN;
// Buses in the design
wire  [7:0]  tune_lpfcb;
wire  [13:0]  net021;

INVD1 I8 ( .ZN(ph_lckb), .A(PH_LOCK));
INVD1 I74 ( .ZN(net77), .A(CK_LOOP));
LRQD1 I75 (.RB(RSTB), .Q(net76), .D(ph_lckb),
    .E(net77));
ND2D1 I76 ( .ZN(net75), .A1(CK_LOOP),
    .A2(net76));
INVD3 I0 (  .ZN(net020), .A(PH_LOCK));
INVD3 I77 ( .ZN(CK), .A(net75));
filter2 I6 ( .OUT(OUT[7:0]), .CK(CK),
    .IN(net021[13:0]), .RSTB(rstb_bf), .TUNE_TOP(tune_lpfcb[7:0]));
AND_14bit_rstb I7 (  .OUT(net021[13:0]),
    .IN(IN[13:0]), .RSTB(rstb_bf));
AN2D1 I10 (.Z(rstb_bf), .A1(net020),
    .A2(RSTB));
BUFD2 I55[7:0] ( .Z(tune_lpfcb[7:0]), .A(TUNE_LPFC[7:0]));
```

```
endmodule
```

Fine Filter Verilog Code

```
module lpf (OUT, IN, TUNE, CK, RSTB) ;
 output  signed [23:0] OUT ;
 input signed [13:0] IN;
 input [1:0] TUNE;
 input CK;
 input RSTB;
 reg signed [23:0] ACC, ACC_N, TEMP1;
 reg signed [23:0] OUT ;
 wire signed [23:0] IN_ADJ;
// pole
 assign IN_ADJ = {IN,10'b0};
 always @(posedge CK or negedge RSTB)
  if (RSTB == 1'b0)
   ACC_N <= 0;
  else
   ACC_N <= ACC;
 always @(*)
  case (TUNE) //tune'a göre pole belirleme
  2'b00 : TEMP1 = (IN_ADJ-ACC_N)>>>2;
  2'b01 : TEMP1 = (IN_ADJ-ACC_N)>>>3;
  2'b10 : TEMP1 = (IN_ADJ-ACC_N)>>>4;
  2'b11 : TEMP1 = (IN_ADJ-ACC_N)>>>5;
  endcase
always@(*) begin
ACC=TEMP1+ACC_N;
end
  always@(*)
  begin
   OUT = ACC;
```

```
 end
 endmodule

module zero1 (OUT, IN, CK, RSTB, TUNE) ;
 output reg signed [32:0] OUT ;
 input signed [23:0] IN;
 input CK ;
 input RSTB ;
 input[1:0] TUNE;
 reg signed [32:0] acc, acc_n, with_k1;

 always @(posedge CK or negedge RSTB)
  if (RSTB == 1'b0)
   acc_n <= 0;
  else
   acc_n <= acc;
// input signal*k1
always@(*) begin
 case(TUNE)
 2'b00: with_k1=(IN<<<5);
 2'b01: with_k1=(IN<<<6);
 2'b10: with_k1=(IN<<<7);
 2'b11: with_k1=(IN<<<8);
 endcase end
 always @(*) begin
  acc = IN + acc_n;
  OUT = acc+ with_k1;
  // overflow protection for integrator
  if (acc > $signed(32'h3FFF_FFFF))
   acc = $signed(32'h3FFF_FFFF) ; // most positive 31b number
  else if (acc < $signed(32'hC000_0000))
   acc = $signed(32'hC000_0000) ; // most negative 31b number
 end
endmodule
module klf_multiplication(IN,OUT,TUNE);
```

```
input signed[32:0] IN;
input [1:0] TUNE;
output reg signed[37:0]OUT;
always@(*)
 begin
 case(TUNE)
 2'b00: OUT=(IN<<<0);
 2'b01: OUT=((IN<<<2)+IN);
 2'b10: OUT=((IN<<<3)+IN);
 2'b11: OUT=((IN<<<4)+IN);
endcase end
endmodule
module ascale1(IN,OUT,TUNE);
input signed[37:0] IN;
input [1:0] TUNE;
output reg signed[17:0]OUT;
always@(*)begin
 case(TUNE)

 2'b00: OUT={IN[37],IN[36],IN[35],IN[34:20]};
 2'b01: OUT={IN[37],IN[37],IN[36],IN[35:21]};
 2'b10: OUT={IN[37],IN[37],IN[37],IN[36:22]};
 2'b11: OUT={IN[37],IN[37],IN[37],IN[37:23]};
endcase
end
endmodule
module trim (OUT, IN);
 output reg [7:0] OUT;
 input signed [17:0] IN;
 always @(*) begin
  if (IN<-128)
   OUT=0 ;
  else if(IN>127)
   OUT=255 ;
  else
```

```
   OUT={~IN[17],IN[6:0]}; //in + 8'd128
 end
endmodule
//fine filter
`timescale 1ns/1ns
module filter4 ();
reg clk;
wire [13:0] d1 = 14'b01101000000000;
wire [23:0]out_p;
wire [32:0]out_z;
wire [17:0]out_total;
wire [7:0] out_trim;
wire [37:0] out_klf;
reg [13:0] innput;
reg rstb;
reg clk2;
wire [7:0] TUNE;
initial begin
 clk = 1 ;
 clk2=1;
end
initial begin
 rstb=0;
 #20;
 rstb=1;
end
// clock generator
always #5000 clk = ~clk ;
//assign a=d1;
always #50 clk2=~clk2;
always@(*)
 innput<=clk*d1;
lpf u_lpf3334 (.OUT(out_p), .IN(innput), .TUNE(TUNE[1:0]), .CK(clk2), .RSTB(rstb));
zero1 u_zero3334 (.OUT(out_z), .IN(out_p), .TUNE(TUNE[3:2]), .CK(clk2), .RSTB(rstb));
```

```
klf_multiplication u_klf13334 (.OUT(out_klf), .IN(out_z), .TUNE(TUNE[5:4]));
ascale1 u_scale3334 (.IN(out_klf), .OUT(out_total), .TUNE(TUNE[7:6]));
trim trim_u3334 (.OUT(out_trim), .IN(out_total));
endmodule
// Library - Layout_filter, Cell - FILT_FINE, View - schematic
// LAST TIME SAVED: Apr 21 13:51:36 2021
// NETLIST TIME: Apr 21 15:22:21 2021
//FLT_FINE
`timescale 1ns / 1ps
module FILT_FINE ( OUT, CK_LOOP, IN, PH_LOCK, RSTB, TUNE_LPFC
    );
input  CK_LOOP, PH_LOCK, RSTB;
output [7:0]  OUT;
input [13:0]  IN;
input [7:0]  TUNE_LPFC;
// Buses in the design
wire  [7:0]  tune_lpfcb;
wire  [13:0]  net69;

INVD1 I78 (  .ZN(net68), .A(CK_LOOP));
LRQD1 I79 ( .RB(RSTB), .Q(net67), .D(PH_LOCK),
    .E(net68));
INVD3 I80 (  .ZN(CK), .A(net66));
ND2D1 I81 (  .ZN(net66), .A1(CK_LOOP),
    .A2(net67));
filter I6 ( .OUT(OUT[7:0]), .IN(net69[13:0]),
    .CK(CK), .RSTB(rstb_bf), .TUNE_TOP(tune_lpfcb[7:0]));
AND_14bit_rstb I7 ( .OUT(net69[13:0]),
    .IN(IN[13:0]), .RSTB(rstb_bf));
AN2D1 I10 (.Z(rstb_bf), .A1(PH_LOCK),
    .A2(RSTB));
BUFD2 I55[7:0] ( .Z(tune_lpfcb[7:0]), .A(TUNE_LPFC[7:0]));

endmodule
```

CURRICULUM VITAE

Name Surname : Zehra Nur Oktay

EDUCATION:

B.Sc. : 2019, Medipol University, Engineering and Natural Sciences, Biomedical Engineering

B.Sc. (Double Major) : 2020, Medipol University, Engineering and Natural Sciences, Electrical-Electronics Engineering

PROFESSIONAL EXPERIENCE AND REWARDS:

- 2015-2020 Fall Semester, Honorary Student, Istanbul Medipol University
- Ranked 3rd in class of 2019 Biomedical Engineering

OTHER PUBLICATIONS, PRESENTATIONS, AND PATENTS:

- "6 GHz Low Noise Amplifier design with 65nm CMOS for 5G/6G Applications", 2020 12. ELECO, IEEE

DIGITAL FILTER AND ANALOG LDO DESIGN FOR ADPLL IN 180 nm CMOS TECHNOLOGY

ORIGINALITY REPORT

11%	8%	6%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Eastern Mediterranean University Student Paper	1%
2	etd.lsu.edu Internet Source	<1%
3	idoc.pub Internet Source	<1%
4	repository.tudelft.nl Internet Source	<1%
5	Submitted to CSU, San Jose State University Student Paper	<1%
6	University of Tennessee, Knoxville Publication	<1%
7	digitalcommons.uri.edu Internet Source	<1%
8	www.electronics-notes.com Internet Source	<1%
	www.ir.nctu.edu.tw	