DESIGN OF FRACTIONAL-N FREQUENCY SYNTHESIZER FOR 2.4/5 GHZ WIRELESS LOCAL AREA NETWORK

By

SUBRATA DEBNATH

A thesis submitted to the

Graduate School-New Brunswick

Rutgers, The State University of New Jersey

In partial fulfillment of the requirements

For the degree of

Master of Science

Graduate Program in Electrical and Computer Engineering

Written under the direction of

Dr. Michael Caggiano

And approved by

_____

_____

_____

_____

New Brunswick, New Jersey

October 2015

ABSTRACT OF THE THESIS

Design of Fractional-N Frequency Synthesizer for 2.4/5 GHz Wireless Local Area Network

by SUBRATA DEBNATH

Thesis Director:

Dr. Michael Caggiano

Frequency synthesizers are widely being used for generating local oscillators for majority of RF, wireless, communication, and navigation systems for the last few decades. Phase-locked-loops (PLL) on the other hand are one of the fundamental portions of any digital/mixed-signal devices in addition to the previously mentioned systems. In this thesis work, a PLL based fractional-N frequency synthesizer for 2.4 GHz and 5 GHz wireless local area network (WLAN) in 0.18 μm CMOS-RF process has been proposed. With the adoption of a MASH 1-1-1 delta-sigma modulator facilitating fractional division ratios through a programmable divider, the frequency synthesizer differs from its integer-N counterpart in its higher reference frequency, wider loop bandwidth, faster settling time,

and better phase noise and suppression of spurious tones. The synthesizer consists of several blocks, including a wide range LC-tuned voltage controlled oscillator (VCO), divide by 16 – 252 programmable divider, dead-zone free phase-frequency detector (PFD), low mismatch high swing cascode charge pump (CP), 3rd order loop filter (LF), and a 3rd order MASH delta-sigma modulator (DSM)—all of which have been designed and constructed in both transistor and layout levels. SPICE (BSIM3) level simulations have been performed for all the individual blocks as well as the complete frequency synthesizer for extracting transient, DC, periodic-steady-state, and phase-noise analyses results. Overall, with 1.2 V supply voltage, the 0.628 mm X 0.594 mm fractional-N frequency synthesizer achieves “locked” state in approximately 2 μs and produces approximately -111 dBc/Hz phase noise at 1 MHz offset (excluding the MASH modulator) while consuming about 20.76 mW of power.

Acknowledgments

I would like to take this opportunity to express my sincere gratitude towards my advisor and thesis director, Dr. Michael Caggiano for accepting me as his supervised graduate student. I would like to thank him for mentoring and guiding me throughout the course of this thesis and helping me with all sorts of technical insights I needed from him. It has been a challenging project and without his support and supervision, it would not be completed.

I would also like to thank Dr. Sigrid McAfee for her motivational and professional guidance. Because of her genuine advice, I am now more aware of the nuts and bolts of the engineering community and have developed the confidence to apply myself better towards any difficult projects. Last but not least, I would like to thank ECE department's systems administrator, John Scafidi for all his efforts in keeping the workstations up and running, and facilitating my allowance of all the resource hungry simulations.

Table of Contents

Abstract ii

Acknowledgments iv

Table of Contents v

List of Figures viii

List of Tables xiii

Chapter 1: Introduction 1

1.1 Wireless Communication, WLAN, and Frequency Synthesis 1

1.2 Motivation 7

1.3 Thesis Organization and Technology Overview 8

Chapter 2: System Level Analysis of Fractional-N Frequency Synthesizer 11

2.1 Introduction 11

2.2 Basics of Charge Pump PLL 12

2.3 Fractional-N Frequency Synthesis 19

2.4 Summary 25

Chapter 3: Voltage Controlled Oscillator 26

3.1 Introduction 26

3.2 Selection of Architecture 27

3.3 LC VCO Basics: LC Tank, Negative Resistance, and Phase Noise. 30

3.4 Circuit Design 40

3.5 Simulation Results 50

3.6 Summary 58

Chapter 4: Phase-Frequency Detector 59

4.1 Introduction 59

4.2 Phase-Frequency Detector Basics 60

4.3 Circuit Design 65

4.4 Simulation Results 71

4.5 Summary 77

Chapter 5: Charge Pump 78

5.1 Introduction 78

5.2 Charge Pump Basics 79

5.3 Circuit Design 85

5.4 Simulation Results 90

5.5 Summary 94

Chapter 6: Loop Filter 95

6.1 Introduction 95

6.2 Design Methodology 96

6.3 Simulation Results 104

6.4 Summary 108

Chapter 7: Programmable Frequency Divider 109

7.1 Introduction 109

7.2 Programmable Frequency Divider Basics 110

7.3 Circuit Design 114

7.4 Simulation Results 126

7.5 Summary 131

Chapter 8: MASH 1-1-1 Delta-Sigma Modulator 132

8.1 Introduction 132

8.2 Delta-Sigma Modulator Basics 133

8.3 Circuit Design 141

8.4 Simulation Results 148

8.5 Summary 153

Chapter 9: Simulation of Complete Synthesizer 154

Chapter 10: Conclusion and Future Improvements 163

References 168

Appendix 175

List of Figures

1.1	Typical Architecture of A Transceiver Front-End	3
1.2	Architecture of Integer-N Frequency Synthesizer	4
1.3	Architecture of Fractional-N Frequency Synthesizer	5
2.1	PLL Linear Model	12
2.2	Type-II Charge Pump PLL	14
2.3	3^{rd} Order Loop Filter	16
2.4	Poles and Zeros of 3^{rd} Order PLL	16
2.5	Transimpedance Plot of 3^{rd} Order Loop	17
2.6	Basic Fractional-N Frequency Synthesizer	19
2.7	3^{rd} Order Delta-Sigma Modulator	23
3.1	Ring Oscillator	28
3.2	LC Tank Circuit	28
3.3	Oscillator as A Negative Feedback System	29
3.4	Control Voltage vs. Output Frequency	31
3.5	Lossy LC Tank Circuit with Active Negative Resistance	32
3.6	Loss Models of On-Chip Inductor and MOS Varactor	33
3.7	(a) NMOS-PMOS Cross-Coupled Pairs, (b) PMOS only pair, (c) NMOS only pair	36
3.8	Phase Noise	37
3.9	Phase Noise vs. Offset Frequency	38
3.10	NMOS-PMOS Cross-Coupled LC VCO	40
3.11	Switchable Capacitor Bank	41
3.12	LC Tank Symmetric Small Signal Model	42
3.13	MOS Transistor Parasitic Capacitors	46
3.14	Threshold Voltage Referenced Self Biasing Circuit	49

3.15	Layout of Voltage Controlled Oscillator	51
3.16	Transient Simulation Showing Start-Up and Differential Outputs	52
3.17	Zoomed-In Transient Signals Showing Periodic Measurements	52
3.18	Output Spectrum of VCO at 4.98 GHz	53
3.19	VCO's Eye Diagram showing Jitter	54
3.20	Phase Noise Measurement for 4.77 GHz Oscillation	55
3.21	Phase Noise Measurement for 5.9 GHz Oscillation	55
3.22	VCO Tuning Using only MOS Varactors	56
3.23	VCO Tuning Using MOS Varactors and Capacitor Bank	57
4.1	Phase-Frequency Detector using Edge Triggered D Flip-Flops	60
4.2	PFD State Machine Diagram	61
4.3	PFD Timing Diagram	62
4.4	PFD's Ideal Behavior	63
4.5	PFD's Non-Ideal Behavior	63
4.6	Down Error Spikes Generated by Dead-Zone	64
4.7	D Flip-Flop Used in Conventional PFD	65
4.8	Pass Transistor Based PFD	67
4.9	(a) 2 Input NAND Gate, (b) 3 Input NAND Gate	68
4.10	Buffer Used in Pass Transistor PFD	68
4.11	(a) Reference Leading VCO, (b) VCO Leading Reference, (c) Reference and VCO Aligned	70
4.12	Lengths of Up and Down Signals	71
4.13	PFD Phase Noise at 1 MHz Offset	72
4.14	(a) Reference Leading VCO, (b) VCO Leading Reference, (c) Reference and VCO Aligned	73
4.15	Lengths of Up and Down Signals of Improved PFD	74
4.16	Pass Transistor PFD Phase Noise at 1 MHz Offset	74
4.17	Layout of the D Flip Flop Based PFD	76

4.18	Layout of the Pass Transistor PFD	76
5.1	(a) Charge Pump Model, (b) Basic CMOS Implementation	79
5.2	Current Steering Charge Pumps (a) Basic (b) with Op-Amp (c) NMOS Switch Only	81
5.3	Aligning "Up" and "Down" Signals with Their Complements	84
5.4	NMOS Switch Only High Swing Cascode Charge Pump	85
5.5	(a) Simple Current Mirror (b) High Swing Cascode Current Mirror	86
5.6	Self Biased Beta Multiplier Current Reference	88
5.7	Matched Charge (Yellow) and Discharge (Green) Currents	91
5.8	Four Different Modes of Operation	92
5.9	Charging (Pink) and Discharging (Green) of the Load Capacitor	92
5.10	Current Pulses and Charging Voltage	93
5.11	Output Current Noise	93
5.12	High Swing Cascode Charge Pump and Self Biased Reference Layout	94
6.1	Passive 3rd Order Loop Filter	96
6.2	Simple Linear Model of PLL	97
6.3	Loop Gain and Phase Margin (Behavioral)	104
6.4	Loop Filter's Transfer Function	105
6.5	Step Response for "Up" and "Down" Currents	105
6.6	Input and Output Noises of the Loop Filter	106
6.7	Phase Noise Gain of Loop Filter	107
6.8	Layout of Loop Filter	108
7.1	Frequency Divider Architecture	111
7.2	Waveform of 2/3 Prescaler Operation	112
7.3	CML D-Latch	113
7.4	CML Divide-By-2 Circuit	115
7.5	Beta Multiplier Reference for 500 μA	119

7.6	Differential to Single Converter and Buffer	119
7.7	(a) Divide-By-2 (b) Divide-By-4 Prescaler	120
7.8	Cascaded 2/3 Programmable Divider Architecture	122
7.9	Programmable Divider with Scaling Logic	122
7.10	Divide by 2/3 Prescaler	124
7.11	Simple D Latch for 2/3 Prescaler	125
7.12	Divide by 4 Prescaler Layout	125
7.13	Transient Simulation for Divide by 4 Prescaler	126
7.14	Transient Analysis for Divide by 2/3 Prescaler	127
7.15	Divide by 16	128
7.16	Divide by 63	128
7.17	Different Division Ratios for 00111	129
7.18	Phase Noise for Division Ratio of 36 at 1 MHz	130
7.19	Programmable Divider Layout	130
8.1	Linear Model of General Delta-Sigma Modulator	134
8.2	Linear Model of the First Order Delta-Sigma Modulator	135
8.3	Hardware Implementation of First Order Delta-Sigma Modulator/Accumulator	136
8.4	Linear Model of 3rd Order Delta-Sigma Modulator	137
8.5	MASH 1-1-1 Delta-Sigma Modulator	141
8.6	24-Bit Accumulator (First Stage)	142
8.7	24-Bit Pipeline Adder (Accumulator for 2nd and 3rd Stage)	143
8.8	4-Bit CLA Block Diagram	144
8.9	(a) Full-Adder for CLA (b) Carry Look-Ahead Logic	145
8.10	(a) Error Cancellation Network (b) Mapping Logic 3 (c) Mapping Logic 1 (d) Mapping Logic 2	147
8.11	Error Corrected Accumulator Overflows Representing 8 Levels of Output	148

8.12 Transient Analysis of Accumulator/Carry Look-Ahead Adder 149
8.13 Randomized Noise Shaping of MASH 1-1-1 150
8.14 Layout of MASH 1-1-1 Modulator 151
8.15 Output Noise Power Spectral Density of MASH 1-1-1 (Behavioral) 152
8.16 Output Noise Power Spectral Density of MASH 1-1-1 (SPICE) 152
9.1 Transient Simulation for Division Ratio of 124 154
9.2 Reference Frequency, Divided Frequency and Control Voltage, 156
(a) Before Settling, (b) After Settling.
9.3 (a) Spectrum of 4.96 GHz Output from VCO, 157
(b) Spectrum of 2.48 GHz Output from Divide-by-2 Prescaler
9.4 Transient Simulation for Division Ratio of 137.5 158
9.5 (a) Spectrum of 5.5 GHz Output from VCO, (b) 55 dB attenuation of 159
first Harmonic at 2.75 GHz
9.6 Phase Noise of Frequency Synthesizer at 1 MHz Offset 160
9.7 Eye Diagram of Frequency Synthesizer 161
9.8 Layout of Complete Fractional-N Frequency Synthesizer 161

List of Tables

2.1	Fractional-N Frequency Synthesizer Design Specifications	24
3.1	VCO Design Goals and Parameters	43
3.2	Equations Used in VCO Design	44
3.3	VCO Design Variables	47
4.1	Conventional PFD Design Variables	69
4.2	Pass Transistor PFD Design Variables	69
4.3	PFD Performance Comparison	75
5.1	Charge Pump and Current Reference Design Variables	88
6.1	Initial Design Parameters	97
6.2	Phase Transfer Functions	98
6.3	Initial Design Parameters	103
7.1	Design Variables for High Speed CML Divider	117
8.1	Mapping Delta-Sigma Modulator Output Bits to Decimal Levels	139
A.1	List of Channels in 2.4 and 5 GHz Bands	181

Chapter 1

Introduction

1.1 Wireless Communication, WLAN, and Frequency Synthesis

With the emergence of technological breakthroughs based on wireless networks and scaling of VLSI processes, the already connected world has become much faster and far more dependent on the connectivity than what it was even half a decade ago. Wireless communication has become such an essential part modern society that losing wireless internet for several hours may have substantial adverse effects on personal and organizational operations. Although, there is a large number of different topologies of wireless communication, Wi-Fi is the key player when it comes to connectivity within a local area.

Wi-Fi in contemporary times is a universally recognized term, generally referring to the 802.11ac/b/g/n wireless local area network defined by IEEE standards [1]. With the adoption of 2.4 GHz, 3.6 GHz, 5 GHz, and 60 GHz ISM bands, the Wi-Fi Alliance has brought upon a whole set of network protocols, that keep us connected to that precious wireless router and constantly provide us with a carrier to send and receive information through. Wireless LAN and wireless devices are constantly being used from the stock market of Wall Street to the local coffee shops, to

the rural schools in South Africa [2], and even to the ICD/pacemakers for the heart patients [3]. Without wireless connectivity, perhaps this thesis work would also not have been thought of—let alone be completed.

A key device in any wireless communication system is the transceiver (amalgamation of transmitter and receiver). One of the crucial components of a transceiver is its local oscillator, with which the signal of interest gets mixed to and is up-converted/down-converted for further processing [4]. In modern radio/RF/digital communication and navigation systems, this local oscillator is supplied using a device named phase-locked-loop (PLL). In general, phase-locked-loops are devices that take a reference frequency (usually from a crystal oscillator), and with the utilization of negative feedback, generate a very stable output oscillation. The generated oscillation is then distributed throughout the entire transceiver [4], [5]. But what if there was a way, by which a single reference frequency could have been used to generate multiple output frequencies covering multiple channels in multiple bands? This is where the concept of frequency synthesis comes into play.

PLL based frequency synthesizers have gained tremendous amount of popularity in the last two decades. Known for their frequency multiplying abilities, frequency synthesizers provide transceivers with a local oscillator, utilization of which may be spanned across multiple channels, if not bands [4], [8]. Synthesizers are also recognized for their

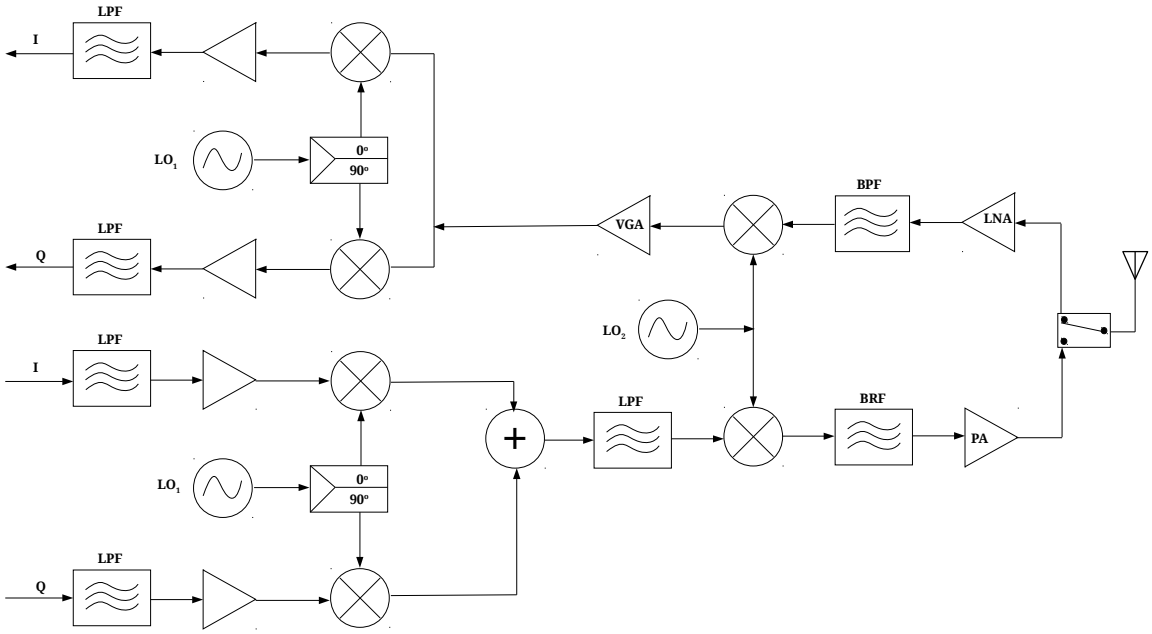

Figure 1.1: Typical Architecture of A Transceiver Front-End [6], [7]

ability in providing an output, stability of which is quite similar to the slower reference frequency. In this thesis, a frequency synthesizer has been developed that uses a 40 MHz reference frequency and generates differential output frequencies ranging from 2.38 GHz to 2.95 GHz, and from 4.77 GHz to 5.9 GHz. This wide range of frequencies cover all the channels within the 2.4 GHz and 5.9 GHz bands. Table A.1 includes the list of channels in the 2.4 GHz and 5 GHz bands.

The concept of frequency synthesis can be easily explained using the diagram provided in figure 1.2. In the figure, it can seen that a simple PLL based synthesizer has been constructed using multiple blocks, namely the phase detector, the charge pump and the low pass filter, the voltage controlled oscillator (VCO), and the feedback divider. The VCO is a

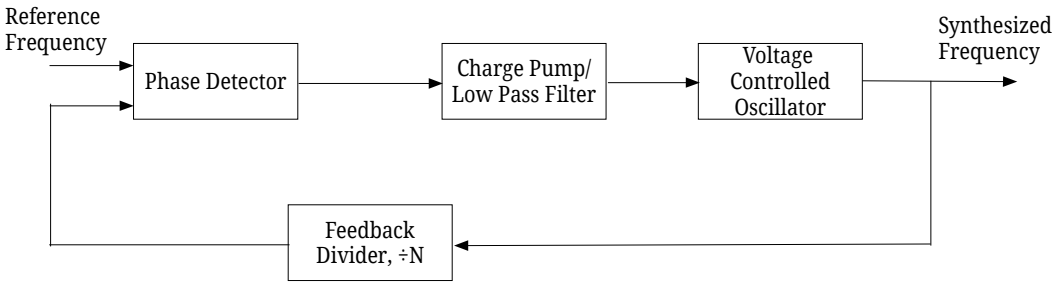

Figure 1.2: Architecture of Integer-N Frequency Synthesizer

free running device which generates an output oscillation predetermined by its design. However, it includes a mechanism to tune its oscillation within a certain frequency range. The output of the VCO is then divided by the feedback divider with a division ratio of N. The phase of the divided signal then gets compared to the phase of the reference frequency at the phase detector, which generates error signals based on the phase difference between the two signals. The error signals afterward gets smoothed out by the charge pump and the low pass filter, filtering out any unwanted high frequency components. The smoothed out signal is then applied to tune the VCO in a way that the phases of the two signals at the phase detector match and the loop achieves a locked state [4], [9]. This cycle of phase detection and VCO modulation continue until the system decides to modify the output frequency by modifying the division ratio.

The operation described above is the basis of a conventional integer-N frequency synthesizer. It takes a fixed valued reference frequency and generates integer multiples of that frequency at the output.

One, however, may wonder how an integer-N frequency divider could take a 40 MHz reference and generate an output of 5.5 GHz (one of the channels in 5 GHz band). As the division ratio would be 137.5, in simple words, it cannot. In real world phase-locked-loop based systems, it is impossible for a divider to achieve a constant fractional division ratio. In addition, over the years it has also been a matter of concern that during phase detection process, the integer-N synthesizers produce considerable amount of spurious tones that generate noise at the output and tune the VCO unnecessarily [10]. This is where the importance of an improved frequency synthesizer was deemed necessary and hence the use of fractional-N frequency synthesizers started growing.

The concept of fractional-N frequency synthesizer is very similar to that of its integer-N frequency counterpart, except the feature that fractional-n can generate an output frequency, that is a fractional

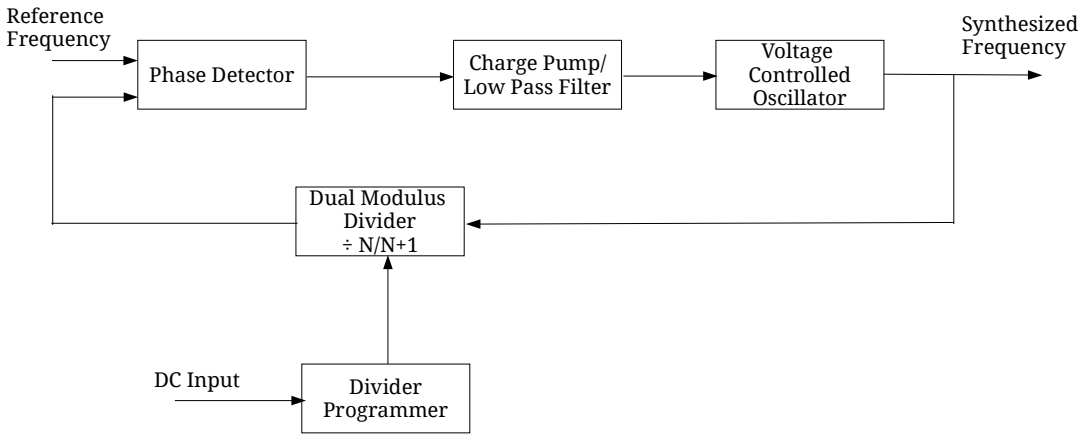

Figure 1.3: Architecture of Fractional-N Frequency Synthesizer

multiple of the input reference frequency. A conceptual diagram of the fractional-N frequency synthesizer has been provided in figure 1.3.

The fractional-N synthesizer makes use of a dual-modulus divider that divides the output of the VCO by either N or N+1. The divider programmer/controller (modulator) on the other hand, facilitates this process of changing the division ratio periodically. In this way, although the divider cannot divide the VCO output by a fractional number, because of periodic division by N and N+1, the average division ratio becomes a fractional number. Over a period of time, this arrangement of periodic dual-modulus division synthesizes an output that is a fractional multiple of the input reference frequency [4], [11]. In addition, fractional-N frequency synthesizers can achieve a faster locking time and fill the deficiency of spurious tone suppression that impacts the performance of integer-N frequency synthesizers [11], [12]. Detailed discussion on this matter will be provided in the subsequent chapters.

1.2 Motivation

Being so readily available and being so tightly integrated in our daily lives, we often do not realize how broad and critical of a subject matter wireless communication is. Frequency synthesizers today are being used not only in RF and wireless LAN devices, but also in GPS navigation systems, baseband devices, satellite televisions, and etc. However, my motivation to start working on this particular topic did not stem from any special interest in RF communication or RF circuit design—but instead from the desire to learn about phase-locked-loops for analog/mixed-signal systems. Nevertheless, as I kept researching on the topic of PLLs and its applications, two specific applications struck my interest—namely frequency synthesis, and clock and data recovery. At the end it appeared that a frequency synthesizer is truly a mixed-signal system, and working on such a project will benefit me with substantial experience in analog, digital, mixed-signal, and RF systems. In addition, having been developing myself as an integrated circuit designer, this was a perfect opportunity to implement my transistor level circuit design knowledge into practice and develop an industry grade monolithic mixed-signal integrated circuit. Furthermore, having strong interest in the integrated circuit design industry, it was essential that I had undertaken a considerably critical project during my graduate studies. All in all, from the start to end, working on the thesis project was pleasurable and I have garnered a great deal of useful knowledge and experience.

1.3 Thesis Organization and Technology Overview

This thesis has been divided into 10 separate chapters. The chapters go into nitty-gritty details of all the components comprising the fractional-N frequency synthesizer. Following is a list of the chapters including their brief summaries.

- Chapter 1: In this chapter the reader is introduction to frequency synthesizers and their application in wireless communication systems.
- Chapter 2: Here the reader will be provided with a system level analysis of the synthesizer including details of linear models of each blocks and their respective noise analysis. System specification of 2.4 GHz and 5 GHz WLAN application will also be discussed in this chapter.
- Chapter 3: The basics of the voltage controlled oscillator, its circuit/layout design techniques, and simulated performance will be discussed in this chapter.
- Chapter 4: The reader will be presented with discussions about two different topologies of the phase frequency detector, their implementations, and a comparison between their simulated performance in this chapter.
- Chapter 5: In the fifth chapter, the reader will gain understanding of the charge pump circuit. Discussion will be

provided on its non-idealities and techniques to improvement them. And finally, simulated performance supporting the claims will be discussed.

- Chapter 6: This chapter discusses one of the most important blocks of the fractional-N synthesizer, the loop filter. Design methodology of the passive loop filter, its calculated and measured transimpedence transfer function, and input/output noise will also be presented.
- Chapter 7: In this chapter, a two part presentation will take the reader through the operating principles, circuit design techniques, and simulated performance of the programmable frequency divider.
- Chapter 8: This chapter will deal with a complicated yet highly essential all digital device, the delta-sigma modulator. Basic analysis and hardware implementation methods will be provided, including the modulator's impact which creates the distinction between the fractional-N synthesizer and the integer-N synthesizer.
- Chapter 9: This chapter will present system level layout and simulation results for the entire frequency synthesizer. Comparative analysis between intended performance and actual results will also be provided.
- Chapter 10: Finally, the thesis will come to an end in this chapter with some concluding remarks and ideas for future improvements.

All the circuits presented in this thesis have been developed in a 0.18 μm CMOS-RF process supported by IBM. For almost all transistors level designs and analyses, software packages from Cadence Design Systems have been used. Schematic capture and mask layout drawings were done in the Virtuoso-XL platform while simulations were performed with Spectre-RF [13]. The frequency synthesizer being such a large circuit, a lot of different methodologies were undertaken in order to meet convergence during simulations. Additionally, due to having access to the simulator in a system with rather limited resources, some performance parameters needed to be loosened. Doing so certainly helped in achieving convergence and generated expected results; however, it also reduced accuracy of the results by a small fraction.

Chapter 2

Fractional-N Frequency Synthesizer

2.1 Introduction

In this chapter, basics of phase-locked-loop and fractional-N frequency synthesizer have been discussed. Analysis of linear models, and operation of individual blocks and their expected outcome will be some of topics of discussion here. Comparative analysis of the development of fractional-N synthesizer from a simple PLL will also be included in this chapter. At the end of the chapter, system level specification for a 2.4/5 GHz fractional-N synthesizer will be provided.

2.2 Basics of Charge Pump PLL

As described in chapter 1, we know that a simple PLL is made up of a phase detector, a loop filter, and a voltage controlled oscillator. Converting the simple PLL into a linear model, we can derive the following figure.

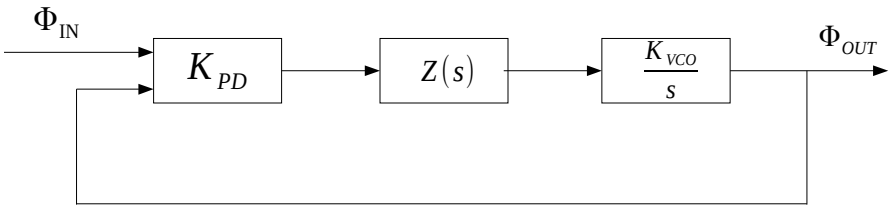

Figure 2.1: PLL Linear Model

Looking at the linear model, we find that the gain of phase detector has been defined as K_{PD}, transfer function of the loop filter is Z(s), and the gain of the VCO is K_{VCO}/s. From this continuous time model, we can derive the open loop transfer function of the loop as [24], [40]:

$$G(s) = K_{PD}.Z(s).\frac{K_{VCO}}{s} = \frac{K_{PD}.Z(s).K_{VCO}}{s} \quad (2.1)$$

For this basic PLL, we assume the loop filter is a simple 1st order RC low pass filter that will suppress the ripples on the control signal modulating the VCO.

$$Z(s) = \frac{1}{1+RCs} \tag{2.2}$$

From equations (2.1) and (2.2), we can then derive the closed-loop transfer function as [4], [40]:

$$H(s) = \frac{\Phi_{out}}{\Phi_{IN}}(s) = \frac{G(s)}{G(s)+1} = \frac{K_{PD}.K_{VCO}}{RCs + s + K_{PD}.K_{VCO}} \tag{2.3}$$

From the above equation we see that, the open loop transfer function has only one pole at the origin due to the VCO. This type of PLL exhibiting an ideal integrator characteristic is known as type-I PLL. Razavi suggests that, expressing the open loop transfer function in terms of natural frequency and damping factor, it can be written as [4]:

$$H(s) = \frac{\omega_n^2}{s^2+2\zeta\omega_n s+\omega_n^2} \tag{2.4}$$

where,

$$\text{damping factor, } \zeta = 0.5\sqrt{\frac{\frac{1}{RC}}{K_{PD}.K_{VCO}}} \tag{2.5}$$

$$\text{and, natural frequency, } \omega_n = \sqrt{K_{PD}.K_{VCO}.\frac{1}{RC}} \tag{2.6}$$

It has been suggested, if at start-up the input and output phases are greatly unequal, because of the 1st order loop filter's frequency suppression characteristic and the ideal integrator's behavior, the simple PLL will not likely acquire a locked state [4], [24]. Because of this reason, a more advanced structure of PLL needs to be investigated. Figure 2.2 shows a simple structure of a type-II charge pump PLL.

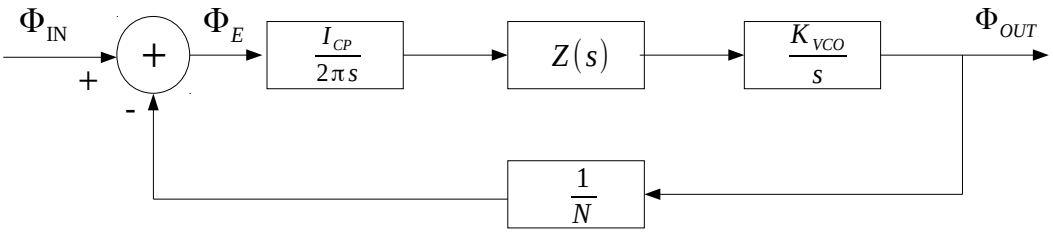

Figure 2.2: Type-II Charge Pump PLL

From the figure above, we find that the phase detector of the simple PLL has been replaced with a combination of phase detector and charge pump, and gain K_{PD} has been replaced by $I_{CP}/2\pi s$. In this configuration, we again find the closed-loop gain of the PLL as [4], [40]:

$$H(s) = \frac{\frac{I_{CP}.K_{VCO}}{2\pi C}.(RCs+1)}{s^2 + \frac{I_{CP}}{2\pi}.K_{VCO}.Rs + \frac{I_{CP}}{\pi C}.K_{VCO}} \qquad (2.7)$$

where,

$$\text{damping factor, } \zeta = \frac{R}{2}\sqrt{\frac{I_{CP}.C.K_{VCO}}{2\pi}} \quad (2.8)$$

$$\text{and, natural frequency, } \omega_n = \sqrt{\frac{I_{CP}.K_{VCO}}{2\pi C}} \quad (2.9)$$

In contrast to type-I PLL, we find that there are now two closed-loop poles at:

$$\omega_{p1} = -\frac{1}{RC} \quad (2.10)$$

$$\omega_{p2} = -\frac{R.I_{CP}.K_{VCO}}{2\pi} \quad (2.11)$$

Because of this two pole system (hence the name, type-II), the system will behave as an oscillatory system containing two ideal integrators. Now, compared to the type-I PLL, if one of the integrators can demonstrate some loss, stability within the system can be established. In this case, the problem that the type-I PLL had with stability and ripple suppression, can not only be improved, but the dependency they have on each other can completely be eliminated.

In modern PLLs or PLL based systems (such as frequency synthesizers), majority of times a type-II charge pump PLL is used. Although most systems are based on higher order loops rather than only 1st

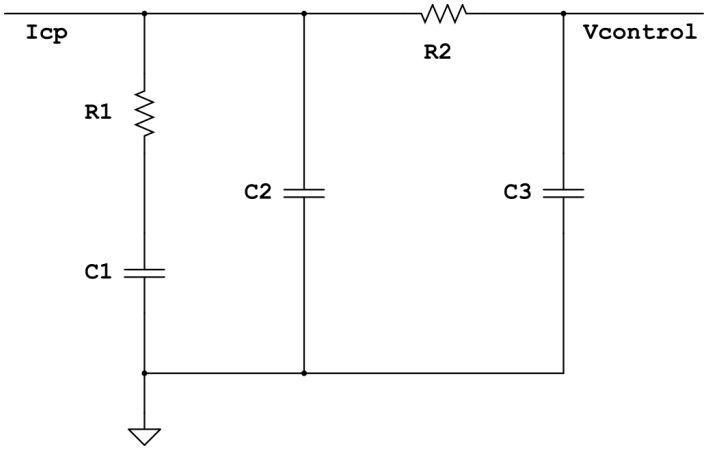

Figure 2.3: 3rd Order Loop Filter

order, even the basic type-II PLL has much advantage over type-I PLLs. In this thesis, the frequency synthesizer has been developed based on a 3rd order PLL, where the loop filter is a 3rd order passive low pass filter. Illustrated in figure 2.3, the 3rd order loop filter generates three poles and a zero which have also been shown in figures 2.4 and 2.5.

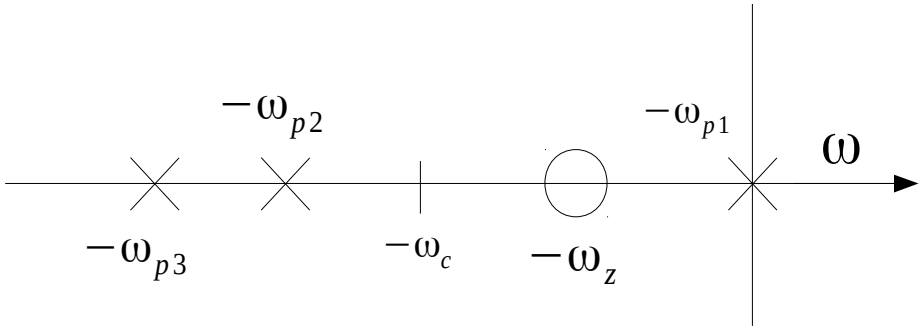

Figure 2.4: Poles and Zeros of 3rd Order PLL

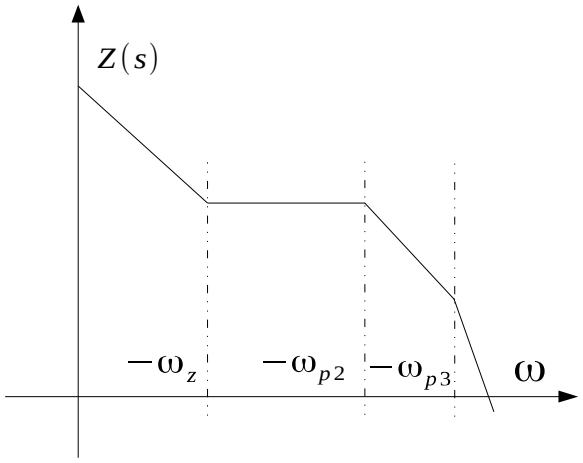

Figure 2.5: Transimpedance Plot of 3rd Order Loop [24]

Going back to the analysis for the charge pump PLL, we can find that the closed-loop transfer function in terms of natural frequency and damping factor is:

$$H(s) = N\frac{2\zeta\omega_n s + \omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \qquad (2.12)$$

Shu et al., suggest that the relationship between the – 3-dB unity gain frequency and the natural frequency can be given as:

$$\omega_{\text{-3dB}} = \left(\sqrt{(2\zeta^2 + 1) + \sqrt{(2\zeta^2 + 1)^2 + 1}}\right).\omega_n \qquad (2.13)$$

Both Razavi, and Shu et al., also suggest that, for the loop to experience a critically damped or overdamped response, a typical choice of damping factor is: √2/2. In addition, for the 3rd order system, using its poles

and zeros provided by the transfer function below [24]:

$$Z(s) = \frac{1}{s}.\frac{\frac{1 + sR_1.C_1}{C_1 + C_2.C_3}}{1 + s.\frac{R_1.C_1(C_2+C_3) + R_2.C_3(C_1+C_2)}{C_1 + C_2 + C_3} + s^2.\frac{R_1.R_2C_1C_2C_3}{C_1 + C_2 + C_3}} \quad , \quad (2.14)$$

we can find the phase margin as:

$$\Phi_m = \tan^{-1}\frac{\omega_{\text{-3dB}}}{\omega_z} - \tan^{-1}\frac{\omega_{\text{-3dB}}}{\omega_{p2}} - \tan^{-1}\frac{\omega_{\text{-3dB}}}{\omega_{p3}} \quad (2.15)$$

Furthermore, the locking time of the PLL can be given as [24], [40]:

$$T_L = -\frac{\ln(\epsilon\sqrt{1 - \zeta^2})}{\zeta\omega_n} \quad , \quad (2.16)$$

where ε is an empirical term defined by the errors in phase and frequency of the reference and the VCO's output oscillation. Based on all the variables derived prior to this point, the loop stability and the loop coefficients are determined. An example of the 3rd order loop filter designed for this thesis has been provided in details in chapter 6.

2.3 Fractional-N Frequency Synthesis

Now that we have some basic understanding of type-II charge pump PLL, we can go ahead and analyze the fractional-N frequency synthesizer. Provided in figure 2.6 is a basic implementation of the fractional-N frequency synthesizer.

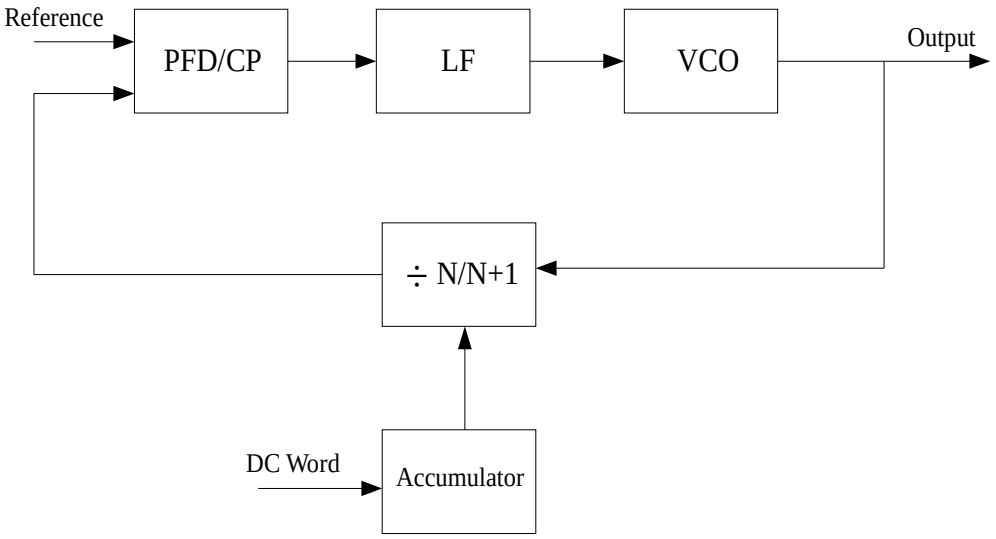

Figure 2.6: Basic Fractional-N Frequency Synthesizer

As discussed in chapter 1, the main difference between an integer-N and a fractional-N frequency synthesizer is in their implementation of feedback divider. Instead of dividing continuously by a constant division ratio, the fractional-N synthesizer divides VCO output by N and N+1 periodically. Perhaps, the most crucial characteristic of the fractional-N synthesizer is its capability of spurious tone and phase noise

reduction. In addition, the fractional-N synthesizer improves locking time with a wider loop bandwidth and smaller division ratio [12].

Heart of a fractional-N synthesizer is its feedback divider and division control circuit/accumulator. The accumulator takes an n-bit DC input and produces an overflow bit at every predetermined cycle. In this arrangement, the feedback divider divides the VCO output by N for a certain number of cycles, and by N+1 for another certain number of cycles. To illustrate this phenomenon, an example from Texas Instruments has been adopted [12]. In the following example, the output of the VCO can be express as [12], [40]:

$$F_{VCO} = F_{REF}.\left(N + \frac{K}{F}\right) \qquad (2.17)$$

where, F_{REF} is the reference frequency

N is the division ratio

K is the predetermined cycle, and

F is the fractional resolution with respect to the reference frequency.

For a 480 KHz reference and division ratio of 2000, if we require a 30 KHz channel spacing, we can determine the following parameters.

First of all, we get F_{VCO} = 2000 x 480 Khz = 960 MHz. Then in order to generate the 30 KHz resolution (meaning subsequent F_{VCO} = 960.03 MHz), we need to change N from 2000 to 2001 one out of every 16 reference cycles. Which means, F = 16, K = 1. In this way, the average division ratio will be:

$$\frac{15 \text{ x } 2000 + 1 \text{ x } 2001}{16} = 2000 + \frac{1}{16} = 2000.0625$$

In this way, the VCO output becomes: F_{VCO} = 960.03 MHz.

At this stage we find that, compared to integer-N synthesizers, resolution of frequencies that can be achieved with fractional-N synthesizers, are much smaller. What that means in terms of a PLLs stability is that, fractional-N synthesizers can use a reference frequency that is orders above the reference for integer-N synthesizers. With a larger reference, we can also achieve a loop bandwidth that is at least twice as wide as the one intended for integer-N synthesizers. Having a wider loop bandwidth, spurs can be eliminated with much better efficiency. Also, having a wider loop bandwidth means that, the PLL can achieve a locking time that is faster than that of integer-N synthesizers. Furthermore, because of the larger reference frequency, the division ratio would be lesser that in turn would generate less phase noise [4], [12].

The basic implementation of the fractional-N synthesizer includes an accumulator. The accumulator can be regarded as a 1st order Delta-Sigma modulator. The function of a 1st delta-sigma modulator is akin to an accumulator that generates an overflow bit to modulate the divider at every predetermined cycle. This technique of fractional division, however, has some disadvantages. Since the division is now occurring at a fractional resolution, fractional spurs are generated that move through the VCO and distort the output [32], [40]. To eliminate this behavior, usually implementation of a higher order delta-sigma modulator is desired.

The function of a delta-sigma modulator is to produce randomized and oversampled output by shaping its quantization noise. Since a 1st order modulator has DC inputs, there is no way to shape the quantization noise. If, however, 2 or more 1st order modulators are cascaded, the modulator generates a randomized switching for the dual-modulus divider such that spurious tones/signals are diminished. So, instead of dividing by N+1 every 10th cycle out of 16 cycles, if the cycle number is randomized, a fractional spur will not dominate. Nevertheless, the fractional division ratio will remain the same. A figure of a 3rd order delta-sigma modulator is shown in figure 2.7.

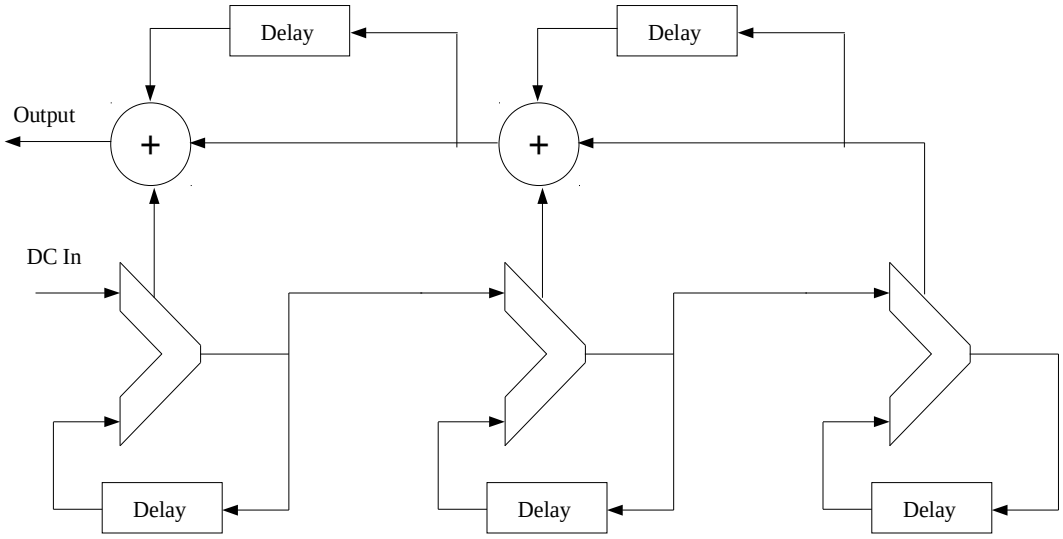

Figure 2.7: 3rd Order Delta-Sigma Modulator

Detailed analysis and design procedure of the 3rd order delta-sigma modulator has been provided in chapter 8. Having understood the basics of phase-locked-loop and fractional-N frequency synthesizer, we now need to design our synthesizer according to IEEE 802.11ac/b/g/n standards. Table 2.1 shows the requirements of a 2.4/5 GHz WLAN frequency synthesizer [1], [49].

Table 2.1: Fractional-N Frequency Synthesizer Design Specifications

Phase Noise at 1 MHz Offset	≤ - 107 dBc /Hz
Spur Level at Reference Offset	≤ - 54 dB
2.4 GHz Frequency Range	2.412 GHz – 2.484 GHz (5 MHz Channel Spacing)
5 GHz Frequency Range	5.035 GHz – 5.825 GHz (20 MHz Channel Spacing)
Lock Time	≤ 10 μs
Reference Frequency	40 MHz
Charge Pump Current	100 μA
Loop Bandwidth	600 KHz
Phase Margin	55^{O}
VCO Gain	760 Mhz/V
Prescaler Division Ratio	4
Programmable Division Ratio	16 – 63
Loop Filter and Delta-Sigma Modulator Order	3^{rd}

2.4 Summary

We have discussed the basics of phase-locked-loop and fractional-N frequency synthesizer in this chapter. Understanding how the system worked in phase domain is essential for designing synthesizers. Adequate explanations were provided using linear models, simple examples, and system level diagrams. Beginning from the next chapter, design specifications provided in table 2.1 will be taken into consideration and appropriate design procedures will be taken in order to meet the system specifications.

Chapter 3

Voltage Controlled Oscillator

3.1 Introduction

This chapter discusses the basic principles, the design, and the simulation results of the LC-tuned Voltage Controlled Oscillator (VCO). For any phase-locked-loop based system, the VCO is undoubtedly the most important block. It is the component that generates the desired stable output frequency from a much slower reference frequency. In this chapter, discussions has been provided on the design steps taken to construct an appropriate VCO for WLAN application, and understand how this single oscillator block is able to generate a range of different frequencies to cover all the channels in the 2.4 GHz and 5 GHz bands.

3.2 Selection of Architecture

Most modern PLLs make use of on-chip passive components to construct their VCOs. The two most preferred types of on-chip VCOs are: 1) Ring Oscillators and 2) LC-tuned Oscillators. Upon the determination of the intended application, area and power budget, and phase noise requirements, it was apparent that the appropriate architecture for the WLAN frequency synthesizer would the LC-tuned Oscillator. LC VCOs are historically known to provide satisfactory phase noise performance at gigahertz frequencies—something that is a determining factor for reliable operation of the frequency synthesizer [14], [15].

A ring oscillator has simpler form than an LC, and usually is constructed of even number of delay stages (inverters) as shown in Figure 3.1. Ring oscillators are known to have large tuning ranges, something that is suitable for our intended application. They are also known for their low area coverage, and low power consuming qualities. However, ring oscillators exhibit phase noise performances that cannot be overlooked in gigahertz range synthesizers, and are much poorer compared to their LC-tuned counterparts. According to IEEE 802.11 standards, it is necessary for a 5 GHz VCO to supply an output oscillation, phase noise of which should be less than or equal to -107 dBc/Hz at 1 MHz offset [1], [16]. It would have been overly complicated to design a ring oscillator to achieve that kind of phase noise performance while covering the frequency range.

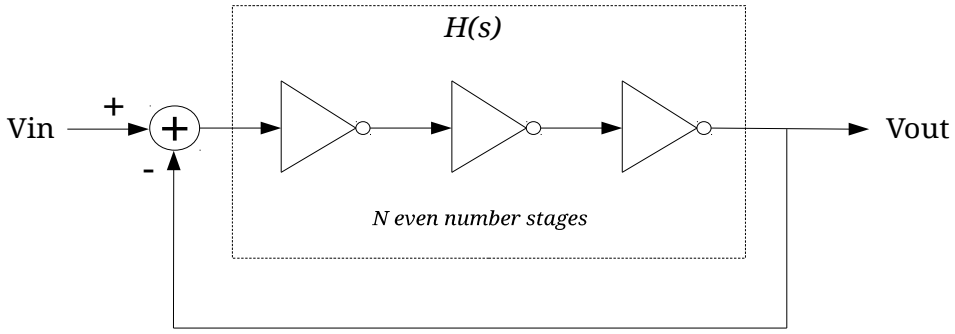

Figure 3.1: Ring Oscillator

As the name suggests, LC-tuned VCOs are made up of inductors and capacitors placed in parallel to a current source. This parallel arrangement of the two passive components is called an LC tank circuit. Figure 3.2 shows a basic LC-tank, also known as LC resonator circuit. If designed correctly satisfying necessary constraints, the tank can generate the intended output oscillation. The output oscillation can be given as:

$$\omega_{osc} = \frac{1}{\sqrt{(L.C)}} \qquad (3.1)$$

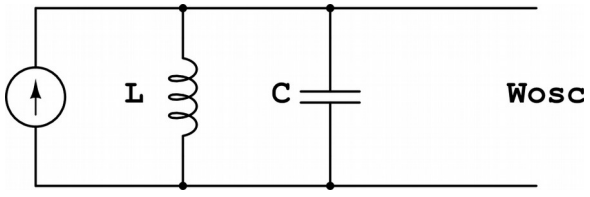

Figure 3.2: LC Tank Circuit

Design of a well-constructed LC-tuned oscillator depends on a lot of different variables and predetermined factors—including quality factors of the on-chip components, controllable capacitance, proper sizing of active devices, and etc.

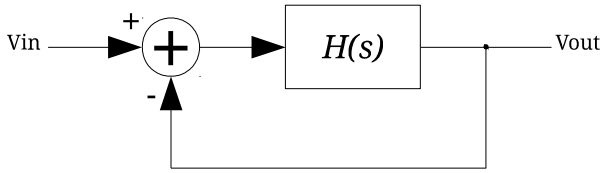

Figure 3.3: Oscillator as A Negative Feedback System

3.3 LC VCO Basics: LC Tank, Negative Resistance, and Phase Noise

Oscillators, or more specifically voltage controlled oscillators, are a study of their own. It is beyond the scope of this thesis to do a comprehensive analysis of the VCO—however, a stripped down analysis of the LC VCO is presented in the following sections. In its most basic definition, an oscillator is a negative feedback system that feeds and amplifies its own noise back to itself and grows to such an extent that it eventually creates a periodic output signal. Depending on the closed-loop gain, amplitude of noise will rise only up to a certain level where it will lower and stabilize the loop's gain to generate a stable oscillation. As depicted in figure 3.3, the closed-loop negative feedback system follows the Barkhausen's Criteria, and exhibits unity gain magnitude and phase shift of 180^{o} as it passes through H(s). Razavi suggests that a VCO can be viewed as a badly designed negative feedback amplifier which has a negative phase margin with one zero and two imaginary poles at ±jω [4].

$$\frac{V_{out}}{V_{in}}(s) = \frac{H(s)}{1 + H(s)} \tag{3.2}$$

$$|H(s = j\omega)| = 1 \tag{3.3}$$

$$\angle H(s = j\omega) = 180^{o} \tag{3.3}$$

As the name suggests, beside the function of oscillation, the primary objective of a VCO is to tune its output frequency using a control

voltage. Figure 3.4 shows a basic plot of control voltage vs. frequency of oscillation. Often expressed as K_{VCO}, the gain or sensitivity of a VCO is given as:

$$K_{VCO} >= \frac{\omega_2 - \omega_1}{V_2 - V_1} \qquad (3.5)$$

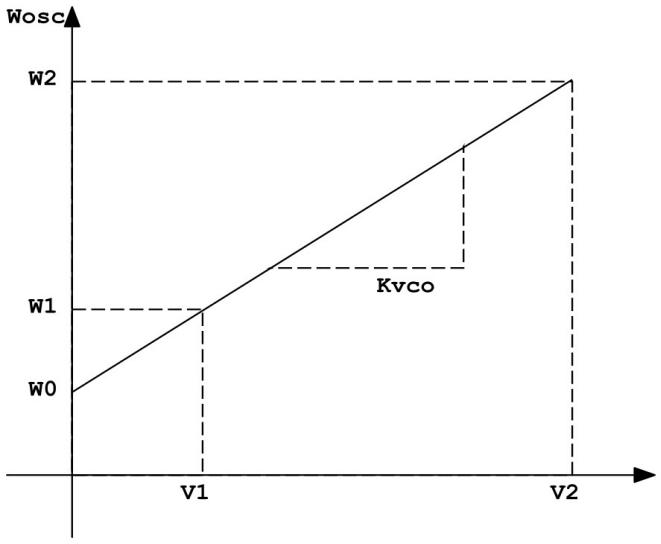

Figure 3.4: Control Voltage vs. Output Frequency

As previously mentioned, LC-tuned VCOs are typically constructed of on-chip components (spiral inductors, metal-insulator-metal capacitors, MOS varactors, etc.). These on-chip components have some resistive loss associated with them. Figure 3.5 shows a simplified diagram of the associative loss with an added negative resistance which is needed in order to achieve successful oscillation.

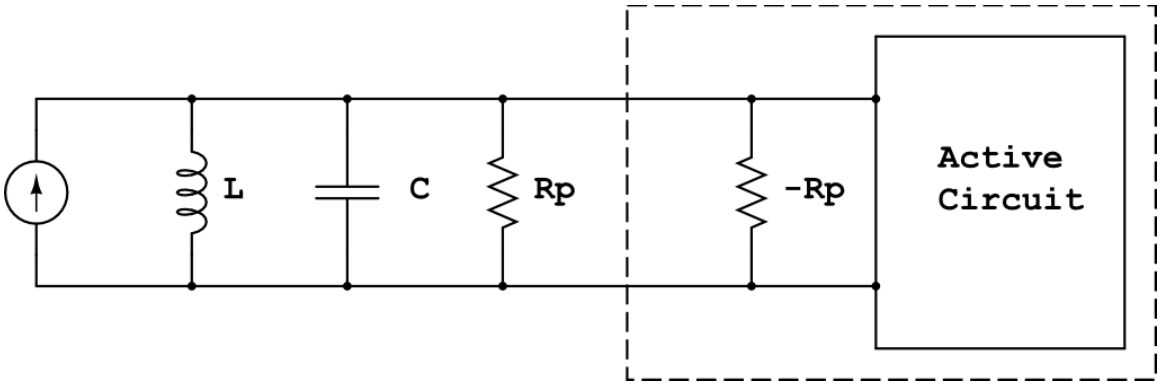

Figure 3.5: Lossy LC Tank Circuit with Active Negative Resistance

We can observe from figure 3.5 that, the cumulative loss associated with the inductor and the capacitor has been represented with R_P, and the active portion generated negative resistance, $-R_P$. Because of this loss, the resonance of the tank will suffer from an exponential decay; and either oscillation will not begin, or it will not sustain [15], [16], [17]. To allay this issue, an active circuit needs to be attached that will provide a negative resistance, $-R_P$. This negative resistance will then try to replenish the exponential loss and in turn will generate and/or sustain the oscillation. Since the output is expected to be differential, cross-coupled pairs of NMOS, or PMOS, or both are used to generate a transconductance, g_m that acts as the negative resistance [4]. Since the transconductance is supposed to eliminate the effect of the resistive loss, the product of both should be equal to or more than 1. Although in practice, designers use a g_m that is much larger than the loss for achieving robustness and eliminating phase noise better.

$$g_m R_P >= 1 \qquad (3.6)$$

Inductors are one of the crucial elements of LC VCOs. Using foundry provided design considerations and constraints, designers in general make use of spiral structures to model and fabricate their inductors. Commonly, the top metal layer of the process is used to fabricate the inductor so that, it minimizes the magnetic coupling from the substrate and hence achieve a good quality factor. Other methods such as parallel

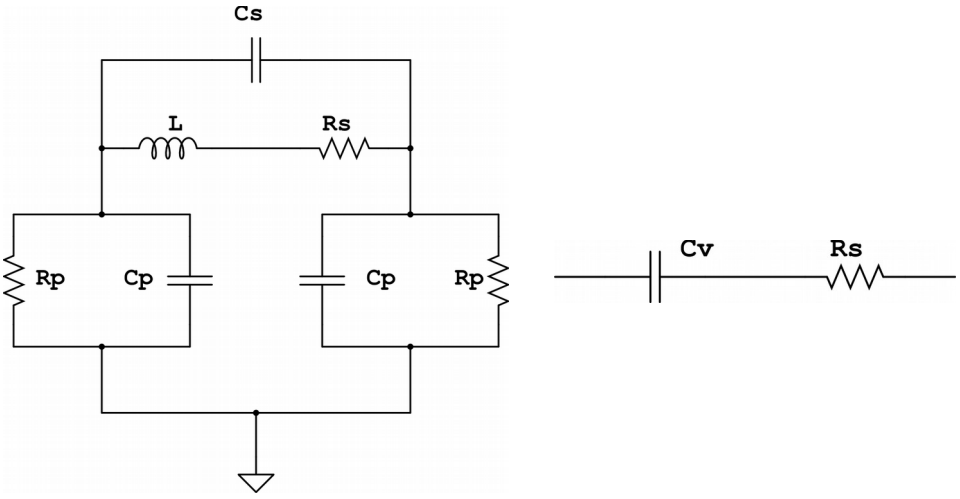

Figure 3.6: Loss Models of On-Chip Inductor and MOS Varactor

placement of multi-layer devices, symmetric devices, use of bond wires, and etc. are also taken into consideration when designing inductors as they may add to the improvement of the quality factor [15], [17], [18]. Even though in many instances PLL designers design their own inductors

suitable for their applications, it was beyond the scope of this paper to come up with an inductor design. In this thesis, an inductor was chosen from the foundry's model guide that provides the highest quality (Q) factor at a close offset from the center frequency.

Similar to the choice of the inductor, a suitable capacitor model also needs to be chosen—paired with which the inductor will generate the intended oscillation. Since the inductance is fixed, to vary oscillation, we need to have a variable capacitive component. CMOS processes have the capability to fabricate variable capacitors, called MOS Varactors. Varactors are tunable with the application of a DC control voltage [4], [17]. For simplicity and satisfactory performance, an inversion region NMOS varactor was chosen for this thesis [19]. A varactor is basically a source-drain connected MOS transistor that provides variable capacitance as the gate voltage is varied. Using the peak Q frequency of the inductor, a varactor model was chosen which has the highest quality factor within the desired capacitance/frequency range. The loss models of both the on-chip inductor and the varactor are shown in figure 3.6. It should be noted that, in an actual LC tank, the parallel capacitance consists of not only the varactor's capacitance, but also the load capacitance, the parasitic capacitance of the inductor, the parasitic capacitance of the NMOS and PMOS pairs, and the capacitance of any switchable capacitors used for discrete tuning. In that case we may reiterate equation (3.1) as:

$$\omega_{osc} = \frac{1}{\sqrt{L(C + C_v)}} \tag{3.7}$$

where C_V is the varactor's capacitance and C consists of capacitances from the load, the parasitics, and the switchable capacitors mentioned earlier. For simplicity, we may combine all the capacitances, including the varactor capacitance and express them as C_{tank}. Based on the maximum and minimum tank capacitance, we may define the output frequency range as:

$$\omega_{osc,min} >= \frac{1}{\sqrt{L_{tank} C_{tank,max}}} \tag{3.8}$$

$$\omega_{osc,max} <= \frac{1}{\sqrt{L_{tank} C_{tank,min}}} \tag{3.9}$$

There are several techniques a designer may adopt in order to build the negative resistance generating active circuit. Figure 3.7 shows the three particular types of cross-coupled differential pairs that are generally used as the active portion in an LC VCO: 1) NMOS only pair, 2) PMOS only pair, and 3) NMOS-PMOS cross-coupled pairs [4], [17].

An NMOS-PMOS cross-coupled topology was chosen for this thesis. The reason behind the choice was to ensure that the common mode voltage level at the output of the MOS devices is at least half of the supply voltage. Additionally, compared to the other two topologies, the output

voltage swing doubles, and the tail current has less modulating effects on the varactors.

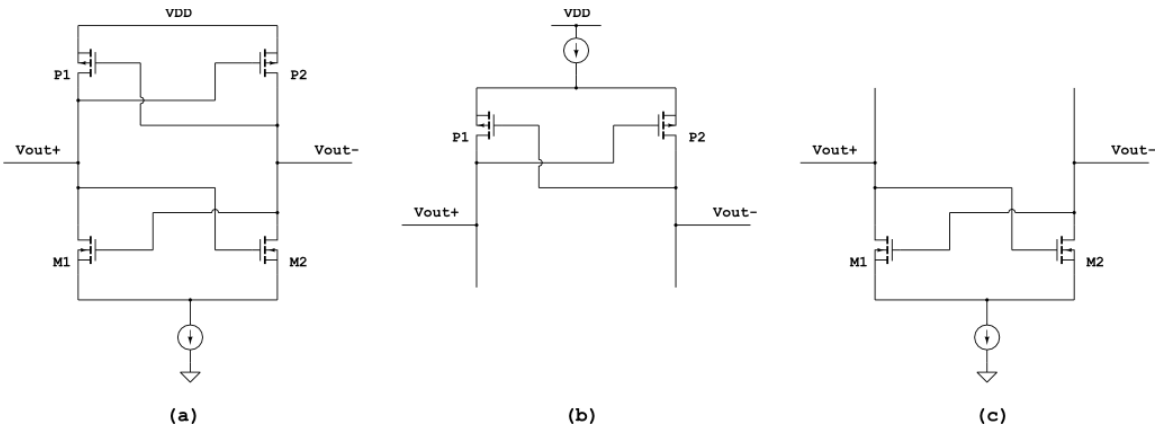

Figure 3.7: (a) NMOS-PMOS Cross-Coupled Pairs, (b) PMOS only pair, (c) NMOS only pair

Illustrated in figure 3.8 is another very important specification of a VCO's performance, the Phase Noise. Heuristically derived by Leeson with equation (3.10), phase noise can be defined as a fluctuation at the VCO's output which is fast, momentary, and irregular, and is typically caused by instabilities either in the VCO itself and/or in the noisy devices used to construct the VCO [21], [22], [23]. Phase noise is an expression in the frequency domain while its time domain expression is known as Jitter —which appears rather frequently in literature dealing with communication, signal processing, digital systems, and etc. Voltage and current sources, thermal noise, flicker noise, and shot noise of passive devices, all can contribute to the phase noise at the output of the VCO and may collectively pull down the free-running frequency; hence causing

instability. Large enough phase noise not only may cause the synthesizer to lose stability, but also add delays in the synthesizer's locking time and may provide the transceiver's mixer with a very noise local oscillator.

$$\{\Delta\omega\} = 10.\log\left\{\frac{2FkT}{P_s}.\left[1+\left(\frac{\omega_0}{2Q_{tank}\Delta\omega}\right)^2\right].\left(1+\frac{\Delta\omega_{1/f^3}}{|\Delta\omega|}\right)\right\} \qquad (3.10)$$

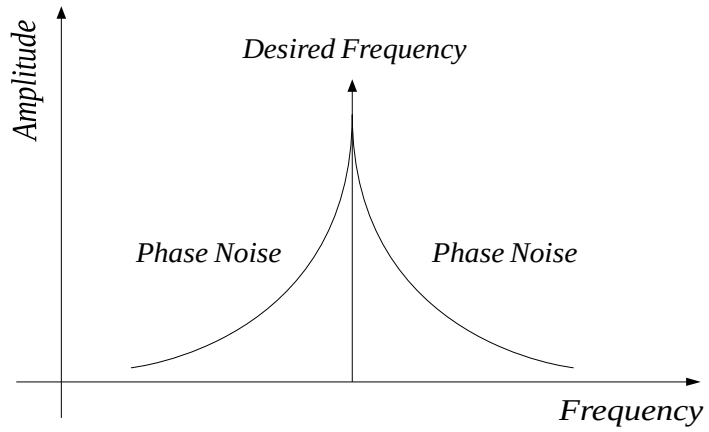

Figure 3.8: Phase Noise

In the equation above:

- F is device noise number/noise factor,
- k is Boltzmann's constant,
- T is absolute temperature,
- P_s is the average power dissipated in the tank's lossy resistance,
- ω_0 is the center frequency of the VCO,
- Q_{tank} is the effective quality factor of the LC tank, also known as

loaded Q,

- $\Delta\omega$ is the frequency offset from the carrier and,
- $\Delta\omega_{1/f^3}$ is the frequency of the corner between the $1/f^3$ and $1/f^2$ regions as shown in figure 3.9.

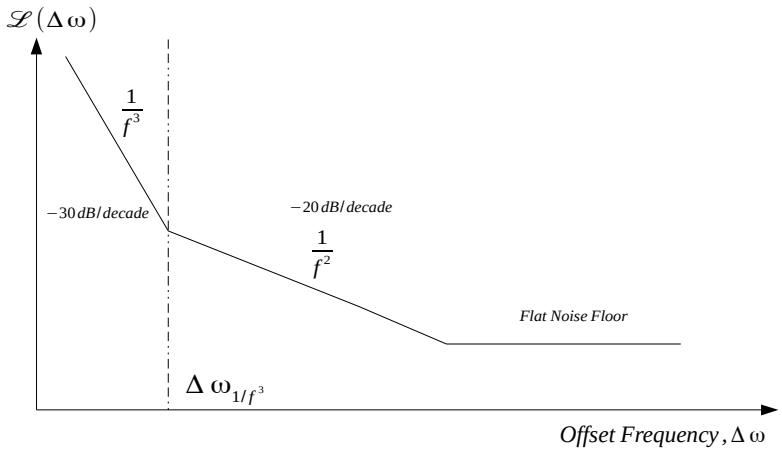

Figure 3.9: Phase Noise vs. Offset Frequency

From the simple model of LC tank represented in figure 3.5 and assuming $\Delta\omega << \omega_0$,

$$Z(\omega_0+\Delta\omega) \approx \frac{1}{g_{tank}} . \frac{1}{1 + j.2.Q_{tank}\frac{\Delta\omega}{\omega_0}} \qquad (3.11)$$

Assuming current noise of the parallel resistance is:

$$\frac{\overline{i_n^2}}{\Delta\omega} = 4FkTg_{tank} \qquad (3.12)$$

We find the phase noise under the $1/f^2$ region as:

$$\mathscr{L}\{\Delta\omega\} = 10.\log\left[\frac{2FkT}{P_s}.\left(\frac{\omega_0}{2Q_{tank}\Delta\omega}\right)^2\right] \qquad (3.13)$$

Shu et al. and Hajimiri et al. describe the portion of phase noise under the $1/f^3$ area to be completely empirical and point out that, both the phase noises under $1/f^3$ and $1/f^2$ regions are results of upconversion of VCO's noise due to some specific nonlinearities associated with it [21], [24].

3.4 Circuit Design

Figure 3.10 shows the VCO circuit that was implemented in this thesis for the fractional-N frequency synthesizer. Here we can see that the LC VCO includes an inductor, two MOS varactors, and a switchable capacitor bank all in parallel to the cross-coupled NMOS-PMOS pair active circuit. Unlike ring oscillators, LC VCOs are incapable of achieving a large (about 1.2 GHz for this synthesizer) frequency range using only varactors. For this reason, in order to improve tuning range, discrete tuning needs to

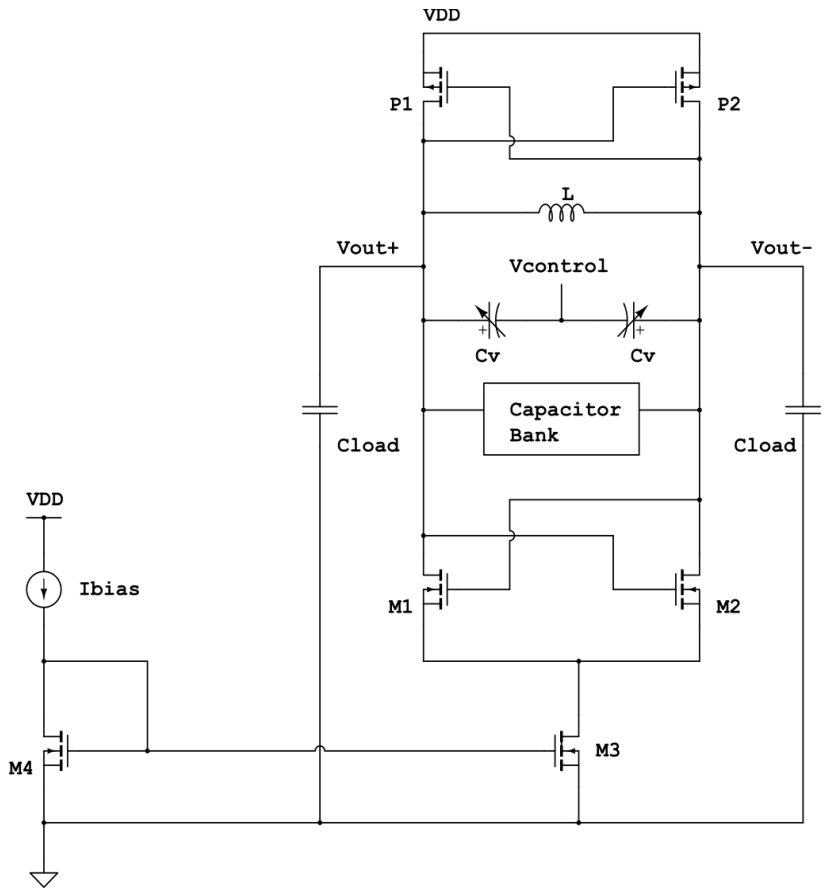

Figure 3.10: NMOS-PMOS Cross-Coupled LC VCO

be performed using a switchable capacitor bank. Usually a set of NMOS devices are used to switch the capacitors. Compared to the other components in the tank, the capacitor bank and the NMOS switches do not add significant parasitic capacitances when the switches are turned off—because of which, the calculations in the following sections have not taken those parasitics into consideration.

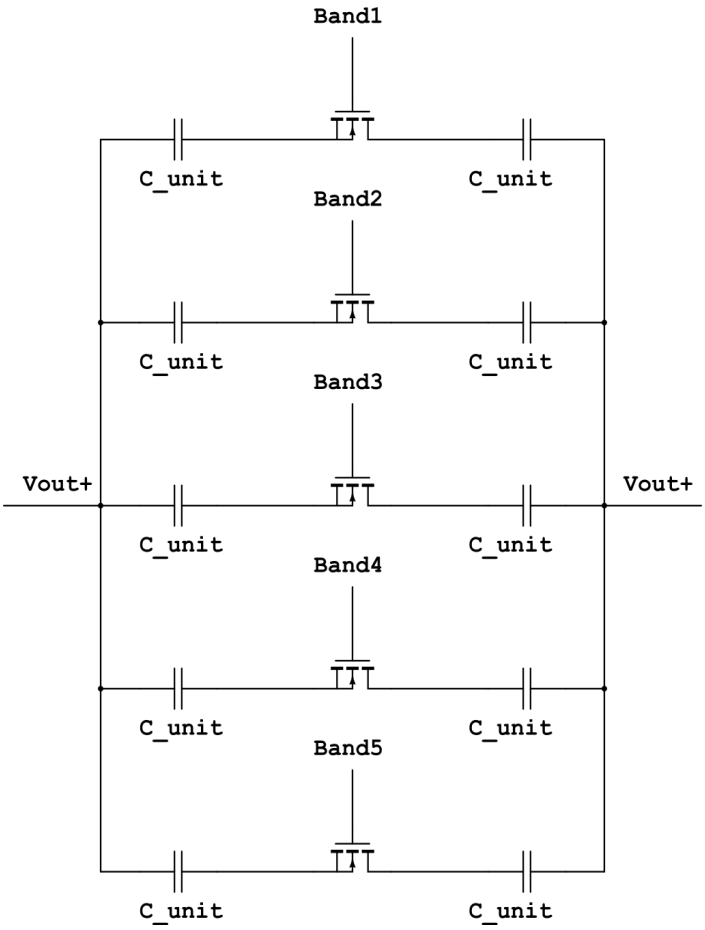

Figure 3.11: Switchable Capacitor Bank

The LC tank shown in figure 3.10 can be further divided and symmetrically redrawn to simplify calculation of the parasitics. Figure 3.12 shows the redrawn small signal model of the LC tank; associative calculations of which are provided in table 3.2. Since the VCO needs to cover all the channels in the 2.4 GHz and the 5 GHz frequency bands, it should generate an output frequency that can be tuned from 4.824 GHz to 5.825 GHz (frequency will be divided by 2 later for 2.4 GHz channels). The initial parameters and design goals are provided in table 3.1. With the specifications in consideration, the VCO was designed using the equations in table 3.2 and the design methodology provided subsequently.

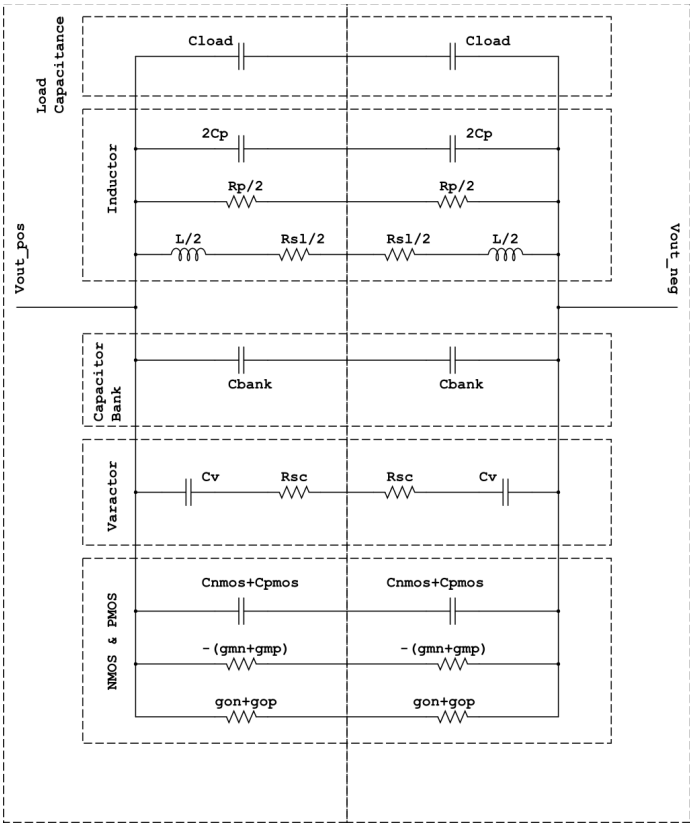

Figure 3.12: LC Tank Symmetric Small Signal Model [20], [21], [22], [23]

Table 3.1: VCO Design Goals and Parameters

Initial Design Parameters	
Tuning Range	4.824 GHz – 5.825 GHz
Phase Noise at 1 MHz offset	≤ - 107 dBc/Hz
Output Voltage Amplitude	600 mV – 800 mV
Control Voltage Range	0 V – 1 V
Process	180 nm CMOS-RF
Supply Voltage, V_{DD}	1.2 V
Bias Current, I_{bias}	1.5 mA
Load Capacitance, C_{load}	50 fF
Oxide Thickness, T_{ox}	4.45 nm (NMOS), 4.60 nm (PMOS)
Transconductance Parameter, μC_{ox}	354 $\mu A/V^2$ (NMOS), 68 $\mu A/V^2$ (PMOS)
Threshold Voltage, V_t	426 mV(NMOS), 379 mV (PMOS)

To begin the design, we start off by choosing an appropriate inductor with large enough Q factor for reduced noise and frequency considerations. We also find the self resonant frequency, SRF, and the peak Q frequency of the chosen inductor. Using the inductance and the maximum frequency of the LC tank, we choose a varactor model; again with a high enough Q factor. From the equation of Q_L and Q_c, we will then

find the parallel capacitance, and loss resistance of the tank circuit. At this stage, we will also estimate the transconductance of the tank. Using the tail current of the VCO, we will then size the NMOS and PMOS transistors accordingly to generate a transcondutance which is at least twice as much as the transconductance estimated for the LC tank. Following this step, we will go back to calculate the parasitics of tank again; however, this time by also adding the parasitics of the transistors. Similarly, we will recalculate the transconductance of the tank, but this time including the output transconductances of the transistors [4], [14], [20]. If the updated values of the tank capacitance and transconductance are within the design constraints, it is preferred to implement the circuit on a SPICE simulator and find out the simulated results. If not, we may have to go back to either modify the varactor model or modify the transistor sizes to meet our design requirements.

Table 3.2: Equations Used in VCO Design

Quality Factor of Inductor, Q_L	$\frac{\omega_L L}{R_{sl}} = \frac{R_{pl}}{L\omega_L}$ where $\omega_L = 2 \text{ x } \pi \text{ x } 5.8\,GHz$
Quality Factor of Varactor, Q_c	$\frac{1}{\omega_c C_v R_{sc}}$, where $\omega_c = 2 \text{ x } \pi \text{ x } 5.8\,GHz$
Tank Transconductance, g_{tank}	$\frac{g_{oN}+g_{oP}+g_v+g_L}{2}$

Active Transconductance, g_{active}	$\frac{g_{mN}+g_{mP}}{2}$
Tank Inductance, L_{tank}	L
Tank Capacitance, C_{tank}	$\frac{C_{PMOS,pair}+C_{NMOS,pair}+C_L+C_v+C_{load}}{2}$
MOS Capacitance, $C_{NMOS} = C_{PMOS}$	$C_{gs}+C_{db}+4C_{gd}$
MOS Pair Capacitance, $C_{NMOS,pair} = C_{PMOS,pair}$	$\frac{C_{gs}}{2}+\frac{C_{db}}{2}+2C_{gd}$
Inductor Parasitic Capacitance, C_L	$C_{sL}+C_{pL}$ or $\cong \frac{1}{2 \text{ x } \pi \text{ x } SRF^2 L}$
Transistor Gate-Source Capacitance, C_{gs}	$\cong \frac{2}{3} C_{ox} W L$, in saturation
Transistor Drain-Body/Gate-Body Capacitance, $C_{db} = C_{gd}$	$\cong 0.3 \frac{fF}{\mu m}$
MOS Output Transconductance, $g_{oN} = g_{oP}$	$\cong \lambda_{N/P} I_D$
Varactor Transconductance, g_v	$\omega^2 C_v^2 R_{sc}$
Inductor Transconductance, g_L	$\frac{1}{R_p} + \frac{R_s}{(L\omega_L)^2}$
MOS Transconductance, $g_{mN} = g_{nP}$	$\frac{2I_D}{V_{gs}-V_t}$ Or $\sqrt{2\mu_{n/p}.C_{ox}.\frac{W}{L}I_D}$
MOS Drain Current, I_D	$\frac{\mu_{n/p}.C_{ox}.\frac{W}{L}}{2}(V_{gs}-V_t)^2$, ignoring channel length modulation

Using table 3.1, table 3.2, and figure 3.12 we may come up with a set of VCO design variables such as inductance, MOS varactor capacitance range, and transistor sizes. Table 3.3 lists the variables used for the design of VCO for this thesis. It should be noted that these variables are dependent on a lot of approximations, such as considering that all the transistor

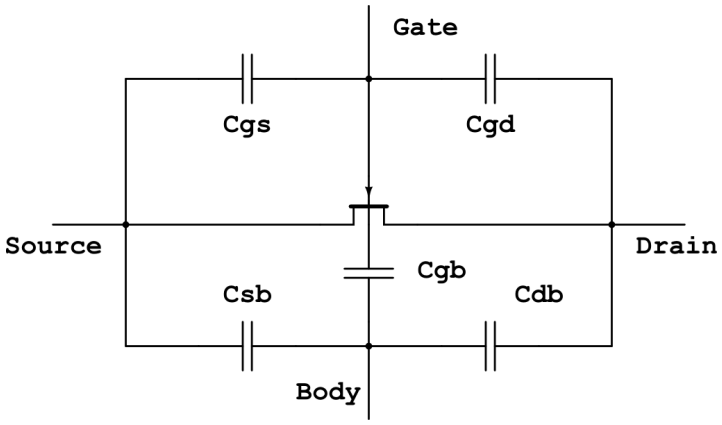

Figure 3.13: MOS Transistor Parasitic Capacitors

operate in saturation region, and that they are long enough to experience constant channel length modulations, and etc. Parameters such as channel length modulation, gate-body/drain-body capacitances are not only process dependent, but also dependent on transistor's regions of operation. For those specific parameters, NMOS and PMOS devices were characterized using parametric SPICE simulations, and averaged values have been used in the design process.

Table 3.3: VCO Design Variables

Spiral Inductor Parameters	
Inductance, L	1.58 nH
Quality Factor, Q_L	20.2
Self Resonance Frequency, SRF	12 GHz
Peak Q_L Frequency, f_{QL}	5.8 GHz
MOS Varactor Parameters	
Varactor Capacitance, C_v	0.26 pF (Vg = 0 V), 0.32 pF (Vg = 1 V) Minimum Varactor Capacitance = 0.26 pF x 2 = 0.52 pF
Quality Factor, Q_c	40
VCO Core	
M_1, M_2	$\frac{W}{L} = \frac{20}{0.18}$
P_1, P_2	$\frac{W}{L} = \frac{110}{0.18}\frac{\mu m}{\mu m}$
M_3, M_4	$\frac{W}{L} = \frac{68}{1}\frac{\mu m}{\mu m}$
Capacitor Bank	
Unit Capacitance	128 fF
MIM Capacitor Size	$\frac{W}{L} = \frac{8}{8}\frac{\mu m}{\mu m}$

NMOS Switches	$\frac{W}{L} = \frac{36}{0.18}\frac{\mu m}{\mu m}$
Self Biased Current Reference	
P_1, P_2	$\frac{W}{L} = \frac{112}{1}\frac{\mu m}{\mu m}$
P_3	$\frac{W}{L} = \frac{174}{1}\frac{\mu m}{\mu m}$
M_1	$\frac{W}{L} = \frac{148}{1}\frac{\mu m}{\mu m}$
M_2	$\frac{W}{L} = \frac{88}{1}\frac{\mu m}{\mu m}$
R, replaced with NMOS	$\frac{W}{L} = \frac{40}{1}\frac{\mu m}{\mu m}$

In addition to the VCO core, a threshold voltage referenced self biasing circuit has been added to bias the core. In figure 3.14, the M_1 transistor is usually made wide enough so that its gate to source voltage becomes equal to the threshold voltage that fells across the resistor [25], [26]. Dividing the threshold voltage with the resistance, R, a suitable current can be generated. In the actual schematic, the resistor has been replaced with an NMOS transistor to eliminate/reduce thermal noise generated by the resistor. Although, the biasing circuit can provide a sufficient supply independent current, a more robust design such as beta multiplier reference circuit or a bandgap voltage reference can be used to ensure supply, temperature, and process independence.

$$I_{bias} = \frac{V_{thN}}{R} \qquad (3.14)$$

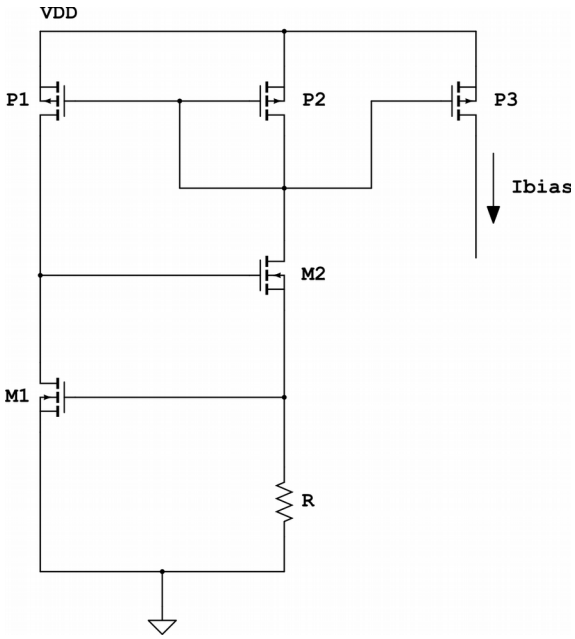

Figure 3.14: Threshold Voltage Referenced Self Biasing Circuit

3.5 Simulation Results

With the device values and sizes from table 3.3, the schematic of the VCO core and the self biasing circuit were constructed in a SPICE simulator for simulation and analysis. Few different types of analysis, such as DC simulation for power consumption, transient simulation for time domain waveforms, periodic steady state simulation for frequency tuning ranges and phase noise, have been performed to assess the performance of the VCO. In addition to the transistor level construction, the circuit has also been developed in physical/layout level. The simulation results presented in this section are results of analyses performed in layout level, which provide more accurate results compared to the transistor level design.

First of all, DC simulation has been performed to find whether all the transistors were operating in the saturation region and to find if they are generating the intended transconductance in parallel with the LC tank. Transistor sizes needed to be tuned accordingly to equate transconductances of the PMOS and NMOS pairs. According to Hajimiri et al., equal transconductances are necessary for the cross-coupled pairs to reduce phase noise and to generate equal voltage swings at the differential outputs [21]. With 1.2 V supply voltage, the total power consumption at the highest output frequency has been found out to be 3.7 mW.

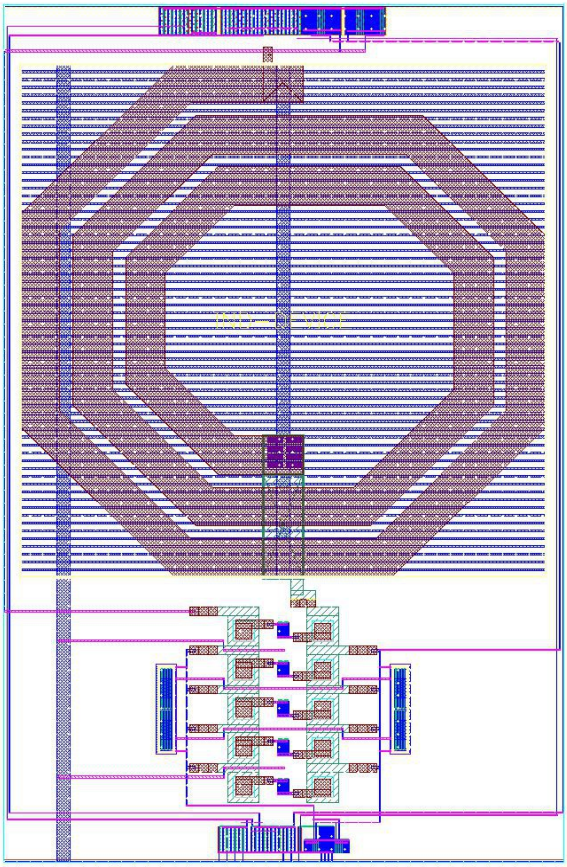

Figure 3.15: Layout of Voltage Controlled Oscillator

Layout of the VCO is shown in figure 3.15. It is evident from the layout that, the reason for LC VCOs to consume so much area in contrast to ring oscillators, is their inductors. In order to reduce PVT variations, common centroid technique was employed during the placement of the cross-coupled pairs. Back annotation has also been performed to achieve a layout simulation performance close to transistor level design by reducing wire parasitics. The area of the whole VCO including the self biasing was measured to be 0.213 mm X 0.335 mm.

Transient simulation for 4.98 GHz output has been performed afterward; results of which are shown in figure 3.16 and 3.17. In a physical circuit, the VCO will start oscillating assisted by its noise; which however, is unavailable in SPICE level simulations. In this case, to initiate oscillation and in order to aid convergence, an initial condition was needed to be set.

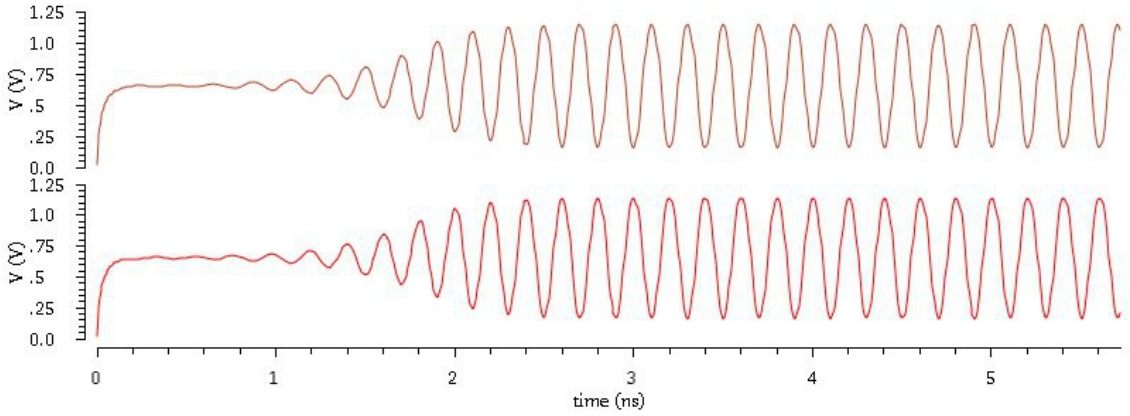

Figure 3.16: Transient Simulation Showing Start-Up and Differential Outputs

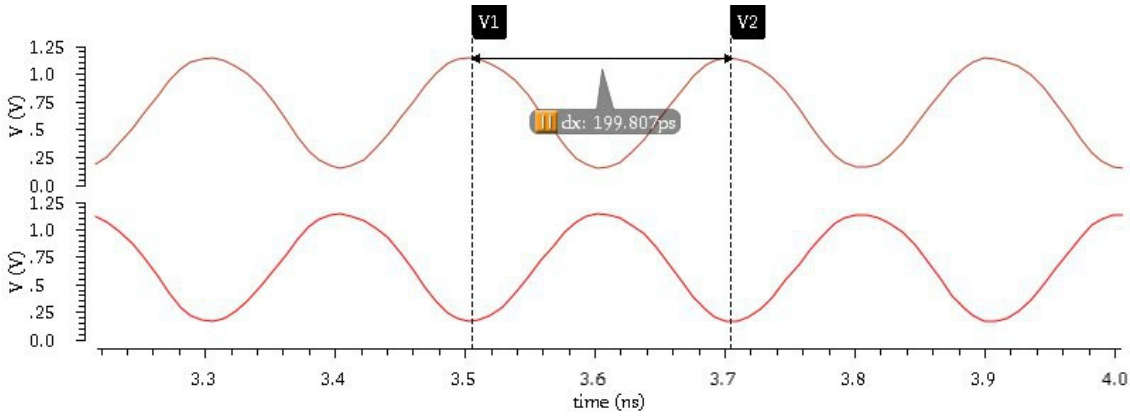

Figure 3.17: Zoomed-In Transient Signals Showing Periodic Measurements

Furthermore, fast Fourier transform (FFT) has been performed on the transient waveforms to generate the spectrum of the oscillation. Figure 3.18 shows the simulated spectrum depicting the phase noise issue illustrated in figures 3.8 and 3.9.

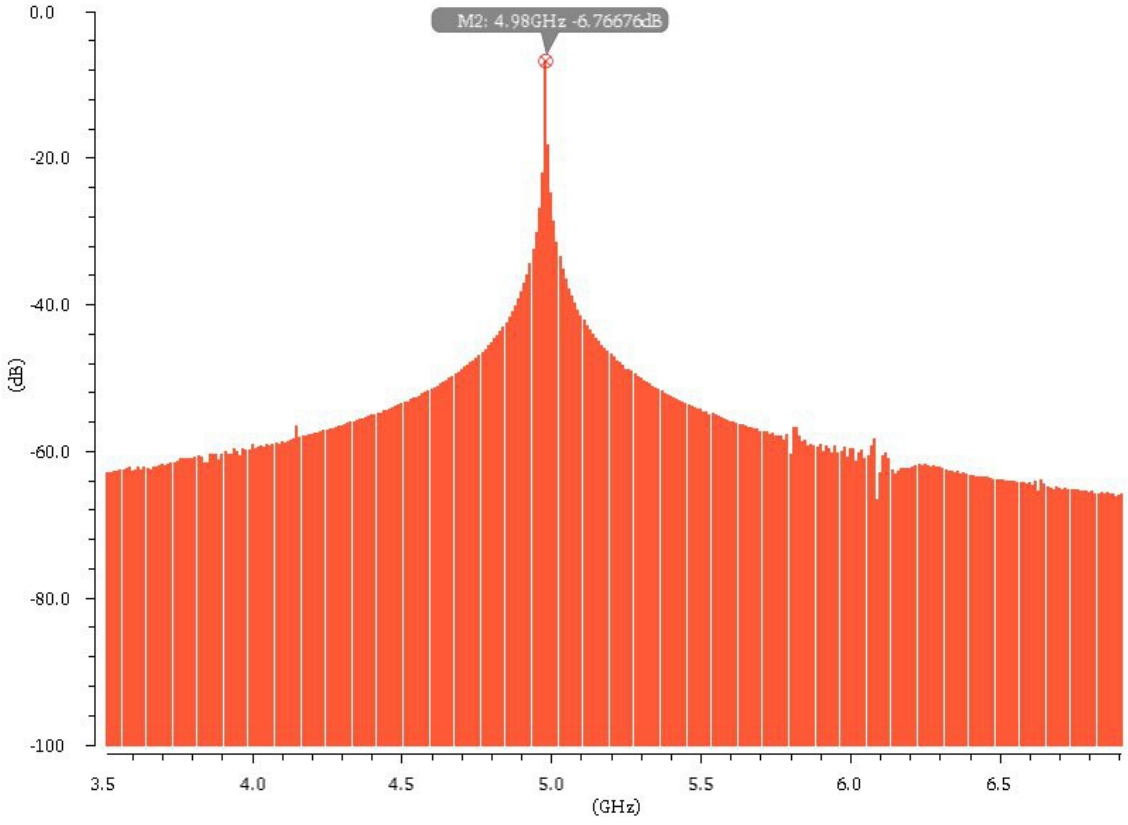

Figure 1.18: Output Spectrum of VCO at 4.98 GHz

From the transient waveforms, an eye diagram has also been extracted to identify the VCOs phase noise/jitter in time domain. The eye diagram in figure 3.19 shows the phase offsets for one period after the VCO established stability.

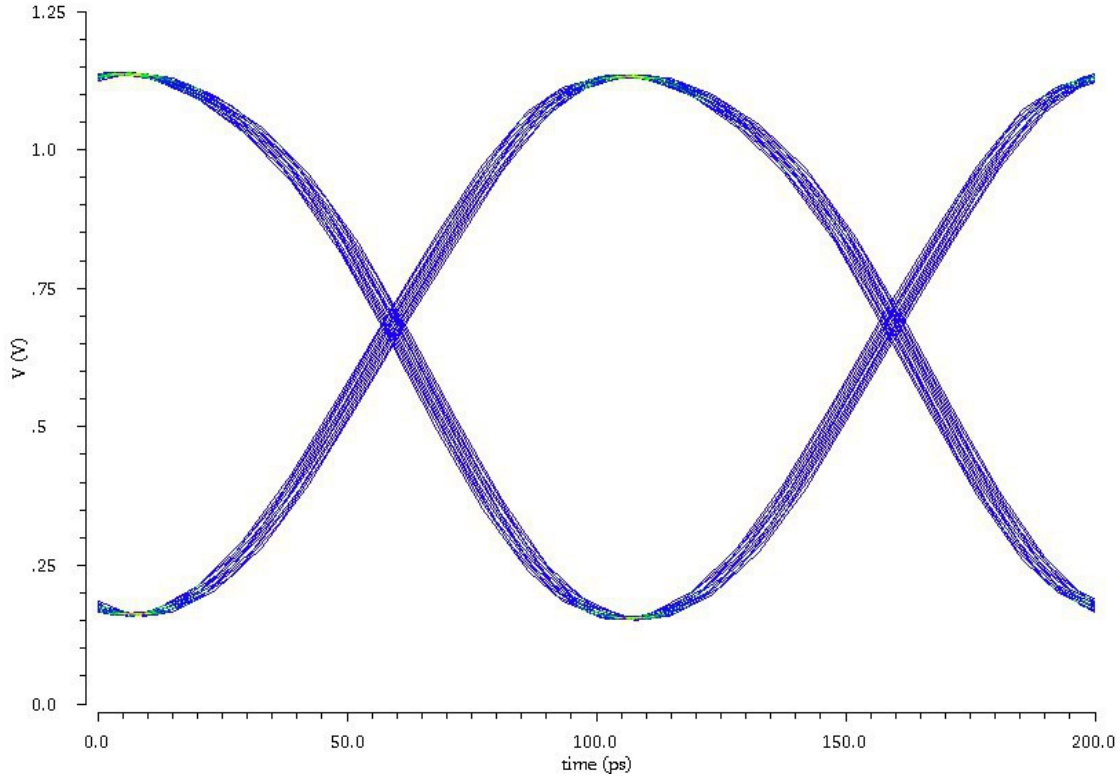

Figure 3.19: VCO's Eye Diagram showing Jitter

Traditionally, phase noise is simulated using periodic steady state analysis, which illustrate phase noise in a frequency domain. In figures 3.20 and 3.21, the best (4.77 GHz) and the worst (5.9 GHz) phase noises exhibited by the VCO are presented. Phase noises in both cases were measured at 1 MHz frequency offset and found to be -118 dBc/Hz and -115 dbc/Hz respectively. Additional steps have also been taken in order to minimize phase noise, such as modification of tail current, improvement of CMOS transconductances, choice of different inductors and capacitors with different Q factors, and etc.

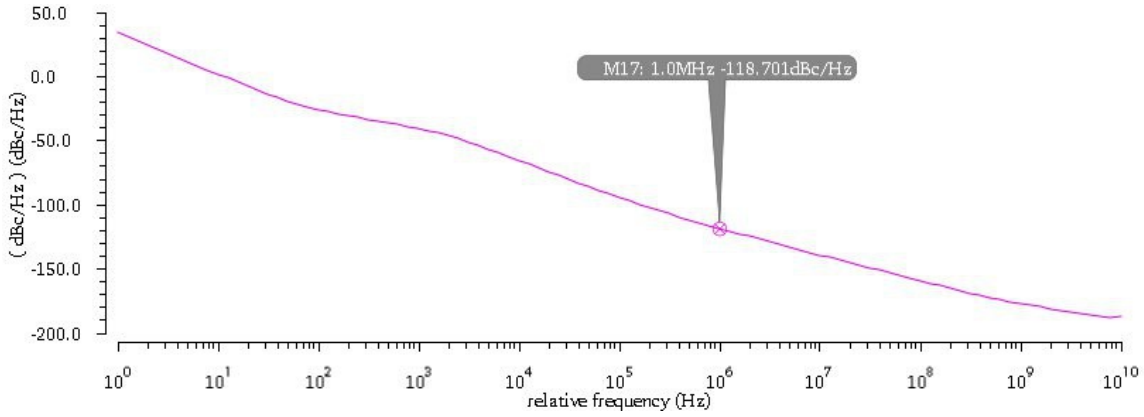

Figure 3.20: Phase Noise Measurement for 4.77 GHz Oscillation

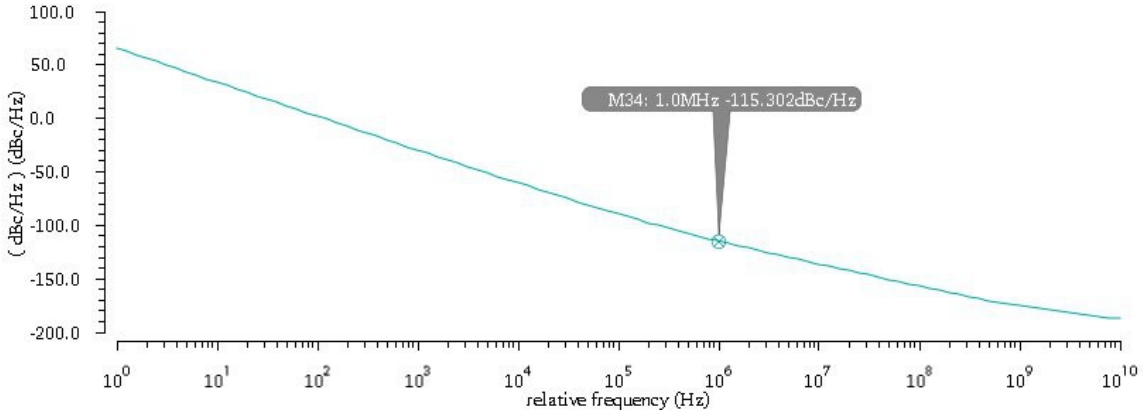

Figure 3.21: Phase Noise Measurement for 5.9 GHz Oscillation

Finally, another parametric periodic steady state analysis was performed to determine the coarse and fine tuning capabilities of the VCO. In figure 3.22 we can observe the fine tuning of the VCO being done by modifying the control voltage of the varactor in a linear fashion. Increasing

the control voltage by 100 mV per step, a frequency range of 220 MHz (from 5.06 GHz to 5.28 GHz) was achieved.

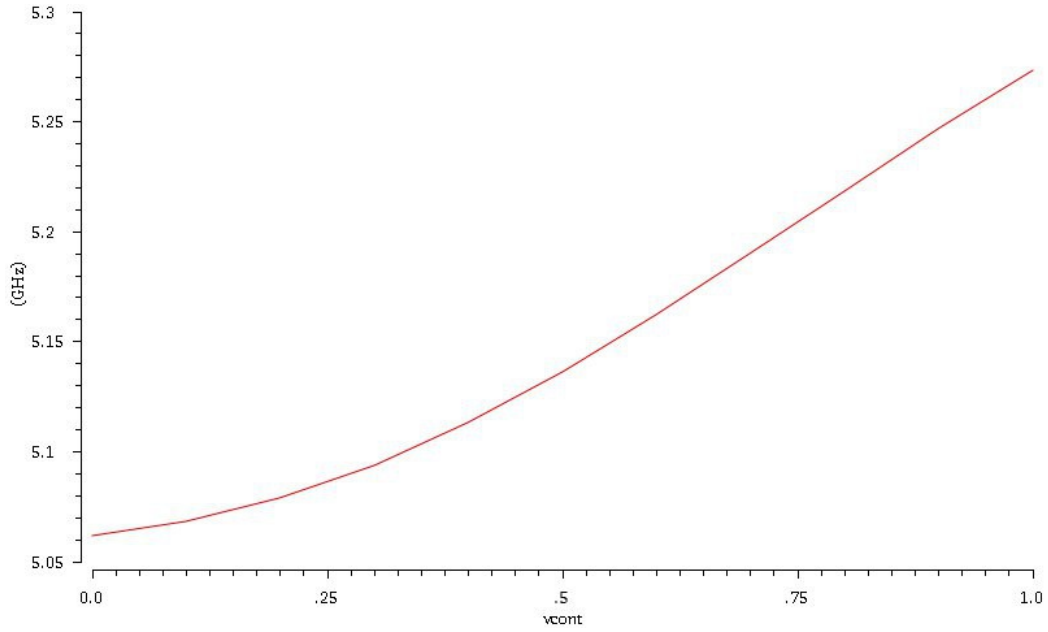

Figure 3.22: VCO Tuning Using only MOS Varactors

The coarse tuning capability of the VCO is provided in figure 3.23. With the use of the capacitor bank and the MOS varactors, discrete levels of fine tuning were performed. Fine tuning method used previously has been applied in conjunction with parametric analysis for the capacitor bank in order to exhibit the VCO's complete tuning range. From 4.77 GHz to 5.9 GHz, the output frequency range of the VCO was found to be 1.13 GHz; something that covers all the channels in the 2.4 GHz (after division) and the 5 GHz bands.

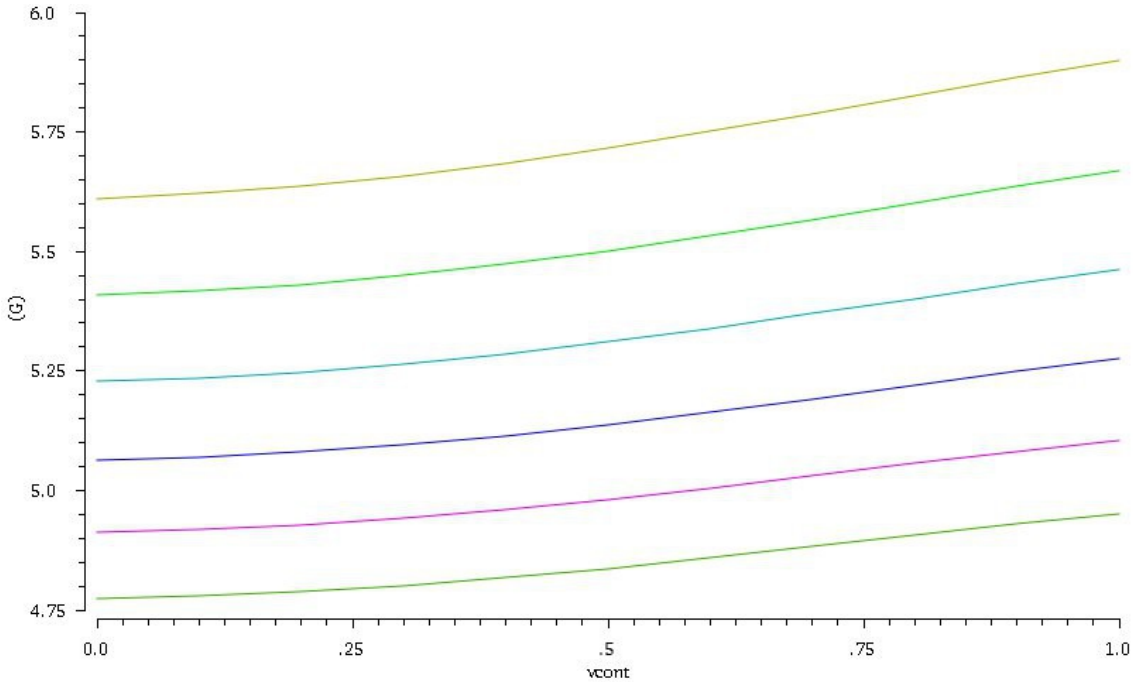

Figure 3.23: VCO Tuning Using MOS Varactors and Capacitor Bank

3.6 Summary

In this chapter, we discussed the basics of LC-tuned oscillator, its operating principles, phase noise, tuning range, and overall design considerations for WLAN applications. Schematics, layout, and their respective simulated performance have also been included. The 0.213 mm X 0.335 mm VCO was able to generate a tunable oscillation from 4.77 GHz to 5.9 GHz with the worst phase noise performance of -115 dBc/Hz at 1 MHz offset. The VCO consumes about 3.7 mW of power with a 1.2 V supply voltage. Observing the performance characteristics, it is evident that the VCO has been constructed well enough to be used with the intended fractional-N frequency synthesizer.

Chapter 4

Phase-Frequency Detector

4.1 Introduction

A phase detector or a phase-frequency detector (PFD) is typically the first block used in a PLL or a PLL based system. The objective of the detector is to compare the phases of reference frequency and the divided frequency, and generate an error signal proportional to their differences. In this chapter we discuss the operational principles, the characteristics, the implementation, and the performance of the PFD designed for the fractional-N frequency synthesizer.

4.2 Phase-Frequency Detector Basics

There are several different types of phase detectors that have been employed in PLL designs throughout history, but the most common type is the tri-state phase-frequency detector. In general, this type of PFDs have some sort of memory elements in them, such as edge-triggered D or J-K flip-flops/latches, and follow the functionality illustrated in the state machine diagram in figure 4.2 [9], [24]. Figure 4.1 shows a conventional implementation of the PFD using two D flip-flops and a NAND (or AND depending on the topology of the flip-flops) gate in their reset paths. The delay element is inserted to reduce “dead-zones”, a non-ideal phenomenon discussed later in the chapter.

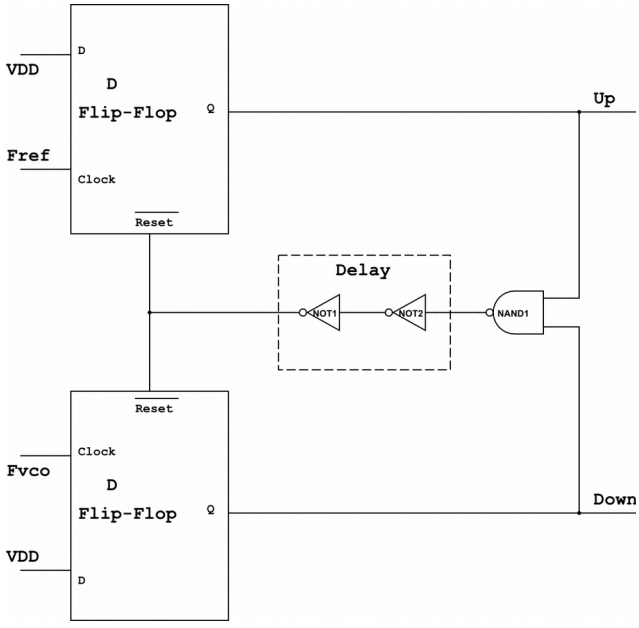

Figure 4.1: Phase-Frequency Detector using Edge Triggered D Flip-Flops

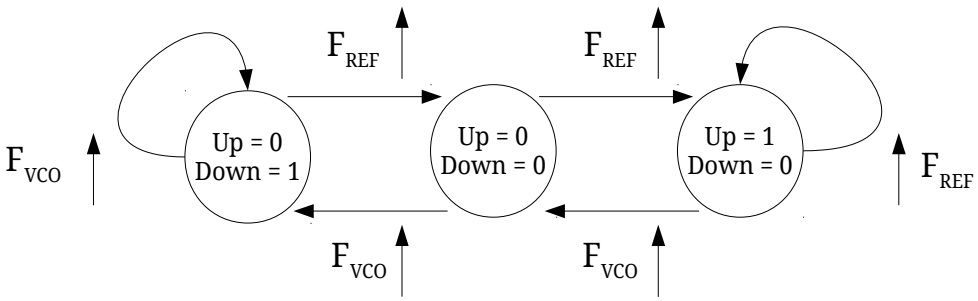

Figure 4.2: PFD State Machine Diagram

Looking at figures 4.1 and 4.2, it can be deduced that the PFD generates an “Up” error signal whenever the phase of the VCO/divider output is lagging the reference signal. Similarly, it generates a “Down” error signal when the reference signal is lagging the VCO output. The operation of the PFD can be further illustrated using the timing diagram in figure 4.3, where we can see conditions at which the “Up” and “Down” signals are generated. What it means in terms of the synthesizer's operation is that, when “Up” is high, the VCO is instructed to oscillate faster in order to cover the phase difference, and when “Down” is high, the VCO is instructed to oscillate slower.

In practice, PFDs are generally used in conjunction with charge pumps, function of which is to convert the error signals into current outputs and eventually use the error currents to modulate the VCO [9]. As previously mentioned, the output of the PFD is proportional to the phase differences between the reference and the VCO output. This difference lies between -2π and 2π. Since the charge-pump will be used with the PFD, the

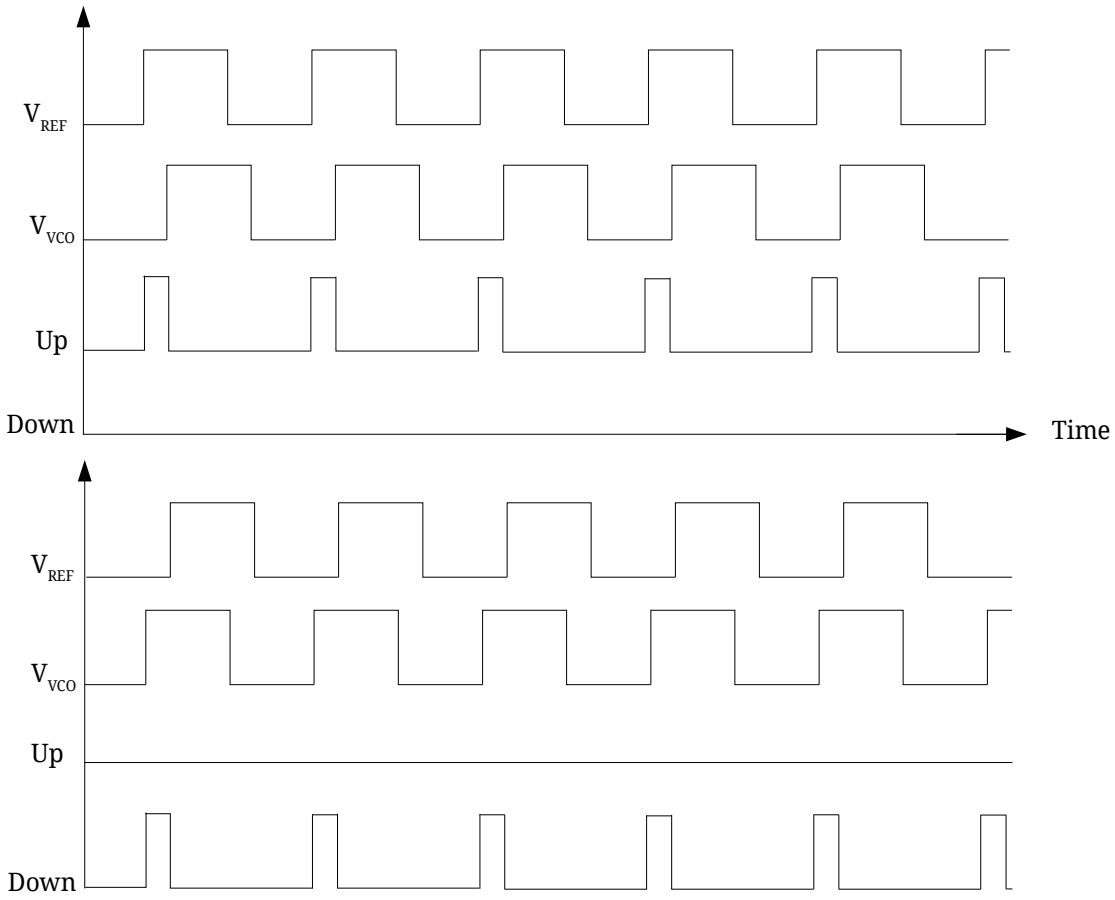

Figure 4.3: PFD Timing Diagram

modulo 2π phase difference pattern will push or pull the charge pump current proportional to the phase difference. Figure 4.4 shows the ideal behavior of the PFD in phase domain [20], [27], [28].

In a physical circuit, PFDs exhibit some non-ideal characteristics. The most commonly observed one is the generation of the dead-zone regions [4], [9], [27]. Dead-zone regions are caused by very small phase errors and time delays within the PFD. Dead-zones have a tendency of unnecessarily modulating the VCO, adding jitter, and reducing

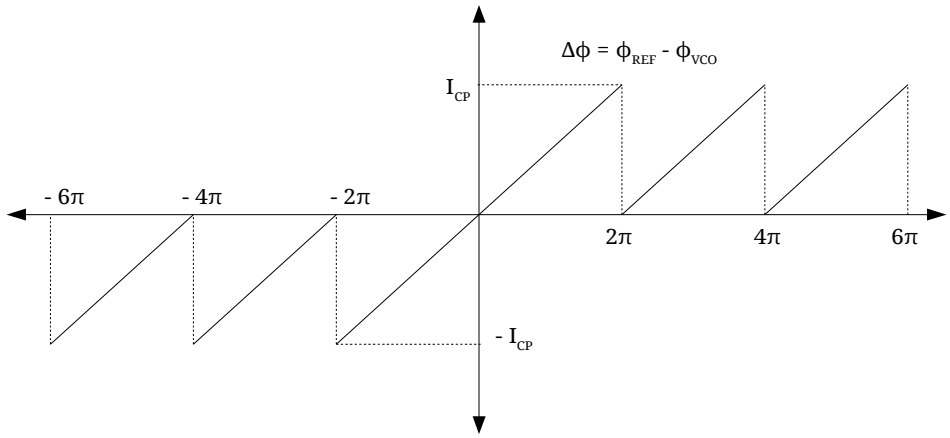

Figure 4.4: PFD's Ideal Behavior

loop gain—all of which ultimately results in delays in PLL's lock time and potential instability [20], [29]. Figures 4.5 and 4.6 illustrate the dead-zone characteristic. It can be seen from the time-domain behavior of the PFD on figure 4.6 that, a dead-zone region is exhibited with unwanted “Down” error spikes that most probably will result in spurs in the spectrum.

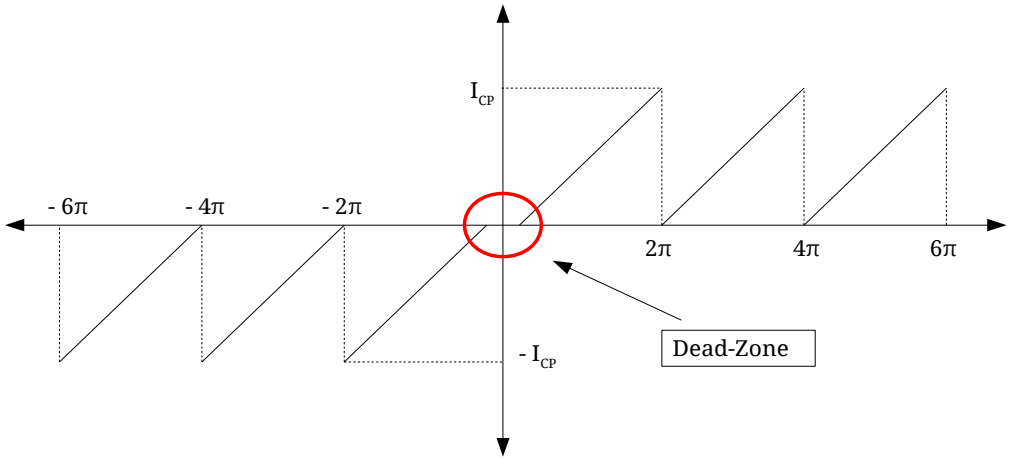

Figure 4.5: PFD's Non-Ideal Behavior

Since dead-zones are a very common issue in any type of PFD, different methods have been employed in PFD design to either reduce or eliminate the dead-zone region completely. On figure 4.1, we saw that a delay element was added in the reset path of the D Flip-Flops. It is done so that, a minimum length of “Up” or “Down” pulse is forced into the PFD and the spikes caused by small phase errors will be suppressed in locked condition [9], [24], [29].

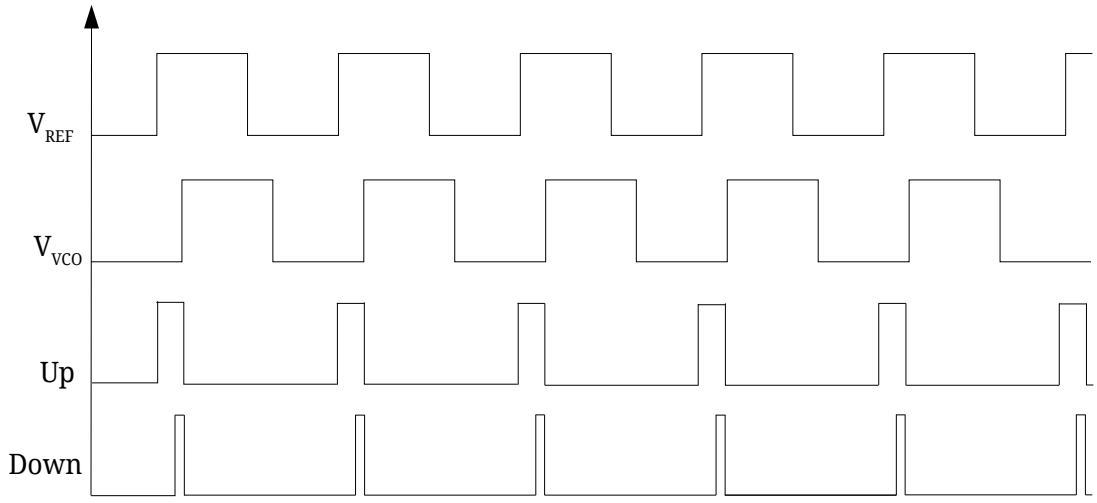

Figure 4.6: Down Error Spikes Generated by Dead-Zone

Another solution would be to use a certain topology that can lengthen the reset path, or eliminate the need of a reset path. This thesis implements a phase-frequency detector that has eliminated the reset path, resulting in higher frequency of operation, lower power consumption, and coverage of a much lower area.

4.3 Circuit Design

In this section, two different types of PFDs have been implemented in SPICE. The D flip-flop based one on figure 4.1 has been implemented first and then a pass transistor based one without a reset path has been implemented. After that, performances of both implementations have been compared to find which design suits best for the intended WLAN application. It was found that the pass transistor based design worked with much more robustness and have completely eliminated the generation of short error spikes. Although simulations of both designs will be compared later in the chapter, operation of only the improved design will be discussed in this section.

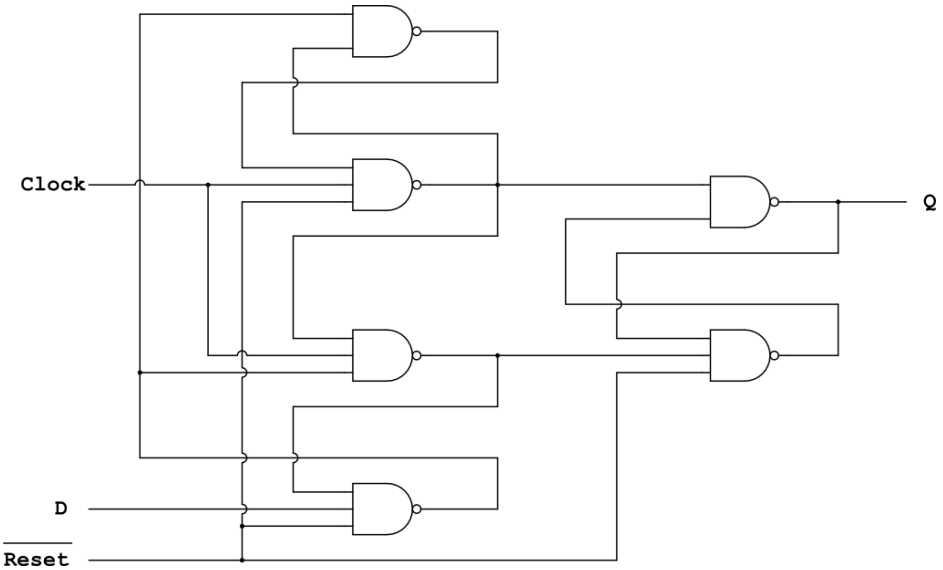

Figure 4.7: D Flip-Flop Used in Conventional PFD

Shown in figure 4.8, we can see that the PFD has a similar structure to that of the D flip-flop based one; however, instead of relying on the “Up” and “Down” signals to trigger the reset of the flip-flops, it relies on the N and P-type pass transistors controlled by the reference and the VCO output signals, and hence producing the exact “Up” and “Down” phase errors [29], [30]. To simply describe the working principle, when both the reference and the VCO output are at low, the node between PMOS P_1, NMOS M_1, and NMOS M_3, and the node between PMOS P_2, NMOS M_2, and NMOS M_3 will be pulled high. When either the reference or the VCO output becomes high, the PMOS devices will be turned off and the node voltages will be passed through the NMOS (M_1 or M_2) devices pulling “Up” and/or “Down” signals to high. In the case when reference voltage is high but the VCO voltage is low, the “Up” signal will be pulled high and “Down” will stay low. The “Up” signal will stay high until VCO output goes up, turning the NMOS device M_3 on, which will then pull the node voltage down and eventually “Up” node low. Similarly, operation of the “Down” signal can be described at the condition when VCO voltage becomes high before the reference voltage. As pass transistor logic states—NMOS devices are worse than PMOS devices for passing logic high and will have a threshold voltage drop [31]. Because of this reason static inverter based buffers have been added to for full scale output.

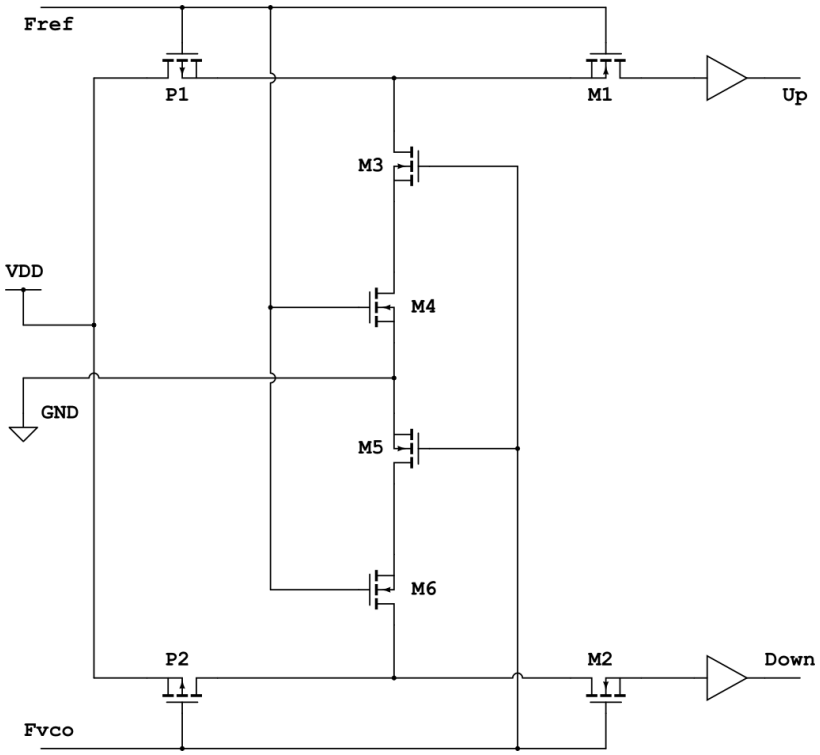

Figure 4.8: Pass Transistor Based PFD

According to Majeed et al., in this design the error signals are not fed back into the PFD, eliminating the delay of reset trigger [29]. Since there is no reset path, for small phase errors short spikes are not generated —which in turn also eliminates the need of a delay element. Tables 4.1 and 4.2 respectively show the CMOS design variables used for designing the conventional and the pass transistor based PFDs.

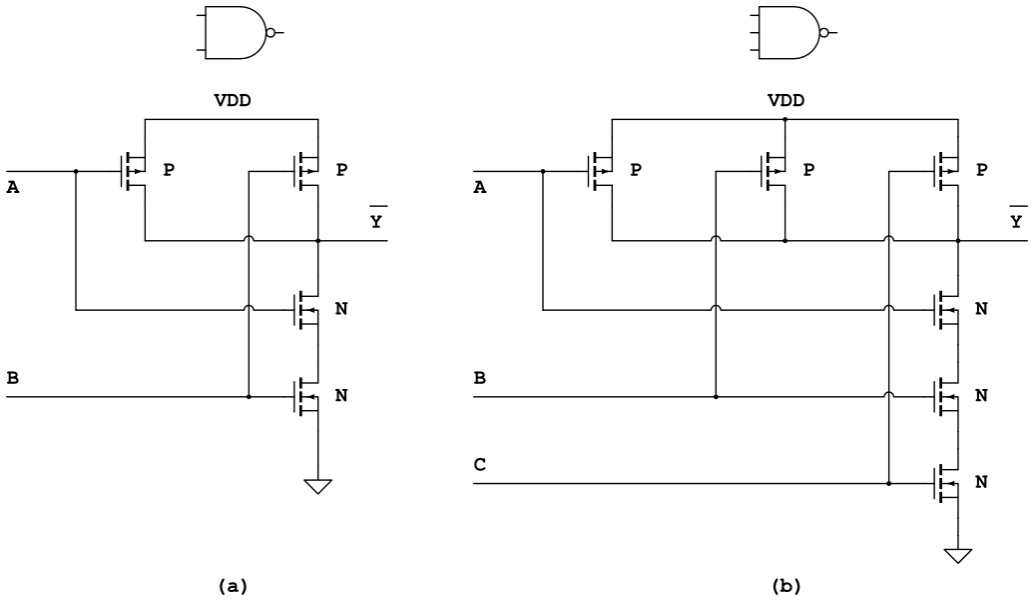

Figure 4.9: (a) 2 Input NAND Gate, (b) 3 Input NAND Gate

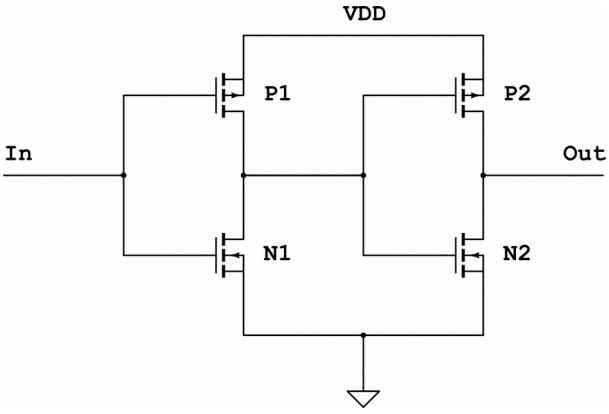

Figure 4.10: Buffer Used in Pass Transistor PFD

Table 4.1: Conventional PFD Design Variables

2 Input NAND Gate	
PMOS, P	$\frac{W}{L} = \frac{2}{0.18}$
NMOS, N	$\frac{W}{L} = \frac{2}{0.18}$
3 Input NAND Gate	
PMOS, P	$\frac{W}{L} = \frac{2}{0.18}$
NMOS, N	$\frac{W}{L} = \frac{3}{0.18}$

Table 4.2: Pass Transistor PFD Design Variables

PMOS, $P_1 = P_2$	$\frac{W}{L} = \frac{0.8}{0.18}$
NMOS, $N_1 = N_2$	$\frac{W}{L} = \frac{3}{0.18}$
NMOS, $N_3 = N_4 = N_5 = N_6$	$\frac{W}{L} = \frac{1.8}{0.18}$
Buffers	
PMOS, P_1	$\frac{W}{L} = \frac{2}{0.18}$

PMOS, P_2	$\frac{W}{L} = \frac{4}{0.18}$
NMOS, N_1	$\frac{W}{L} = \frac{1}{0.18}$
NMOS, N_2	$\frac{W}{L} = \frac{2}{0.18}$

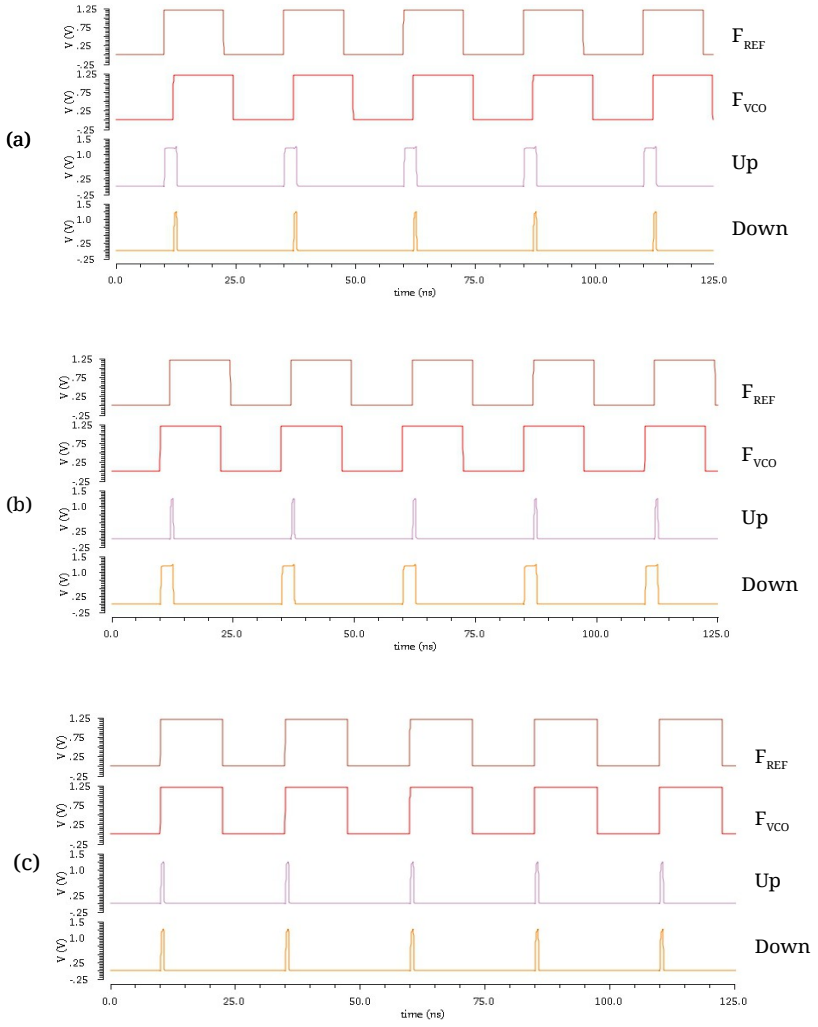

Figure 4.11: (a) Reference Leading VCO, (b) VCO Leading Reference, (c) Reference and VCO Aligned

4.4 Simulation Results

Similar to the previous chapter, several different types of simulations were performed in assessing the performance of the PFD; including DC, transient, periodic steady state, and phase noise analysis. In the following few sections simulated results of both different designs of PFDs will be presented and compared.

Transient simulation results of the conventional PFD for 40 MHz reference and VCO output is shown in figure 4.11 and figure 4.12. We can see that the PFD works as intended resulting in actual “Up” and “Down” error signals and unwanted spikes.

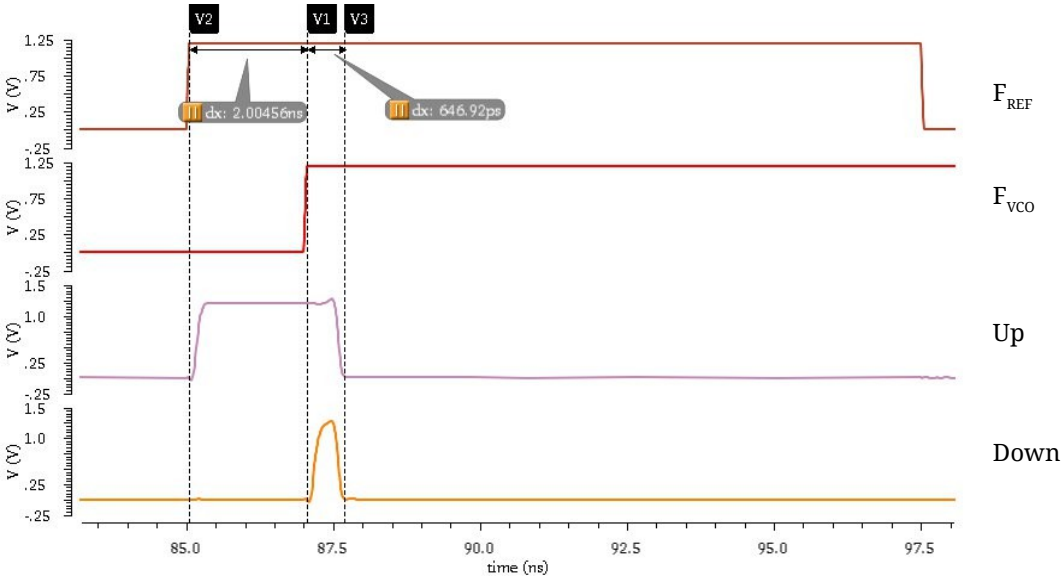

Figure 4.12: Lengths of Up and Down Signals

We can observe in figure 4.12 that for a VCO output lagging the reference signal, the length of “Up” error signal is about 2.6 ns, while the Down spike is about 650 ps. The best phase noise performance of the PFD has been measured to be -157 dBc/Hz at 1 MHz offset while the worst was -139 dBc/hz.

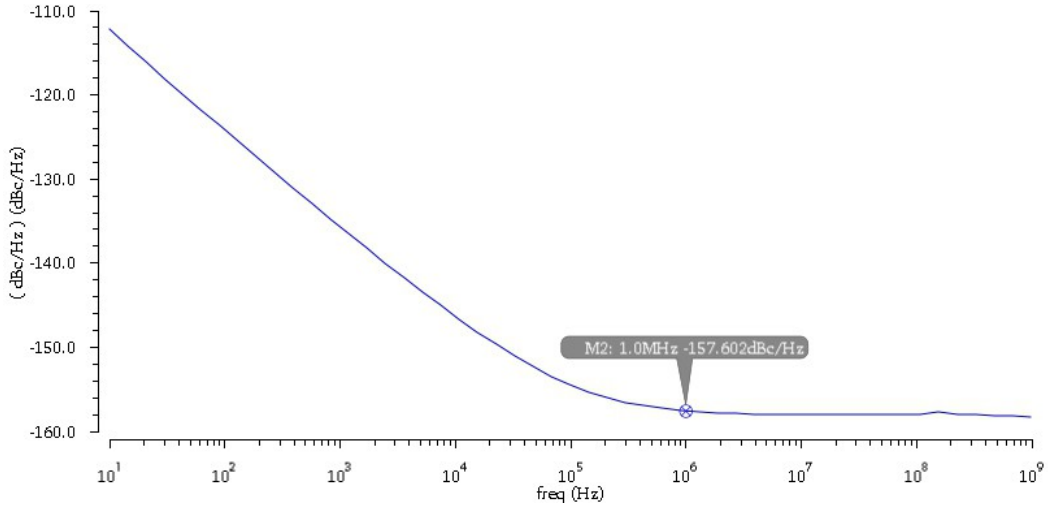

Figure 4.13: PFD Phase Noise at 1 MHz Offset

Simulations were performed again for the improved pass transistor based PFD, and the transient simulation results have been extracted and provided in figures 4.14 and 4.15.

It is evident from the simulation results in figure 4.14 that this implementation of the PFD does not generate any kind of non-ideal spikes in all three modes of operation. It is also observable from figure 4.15 that the lengths of the error “Up” and “Down” signals have been reduced

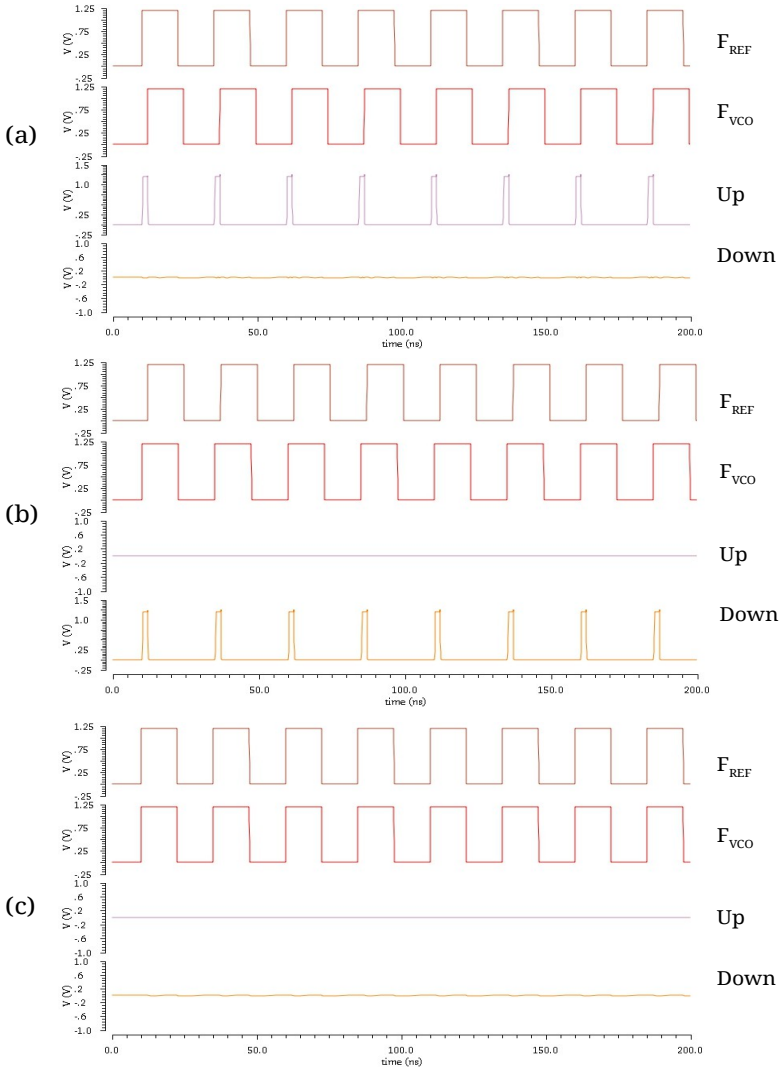

Figure 4.14: (a) Reference Leading VCO, (b) VCO Leading Reference, (c) Reference and VCO Aligned

resulting in less overall power consumption. Although, because of the output buffers reduction of lengths were little more than expected, it should not drastically affect the performance of the synthesizer other than slightly increasing the settling time. Comparing that with the performance

of the conventional PFD, it is superior and will not generate reference spurs resulting in unwanted VCO modulation and increased jitter.

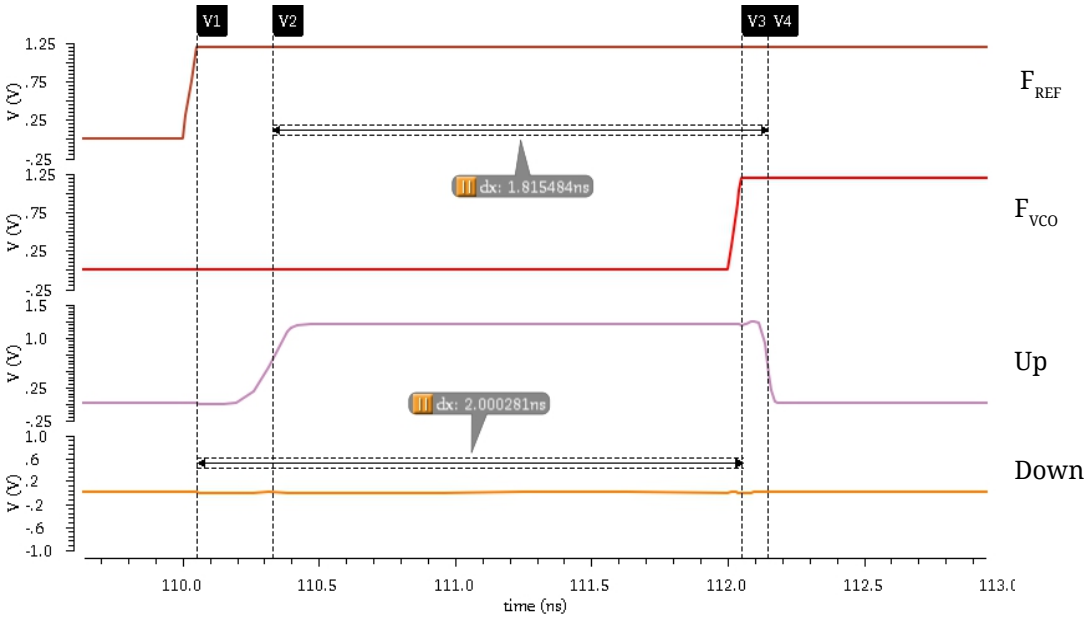

Figure 4.15: Lengths of Up and Down Signals of Improved PFD

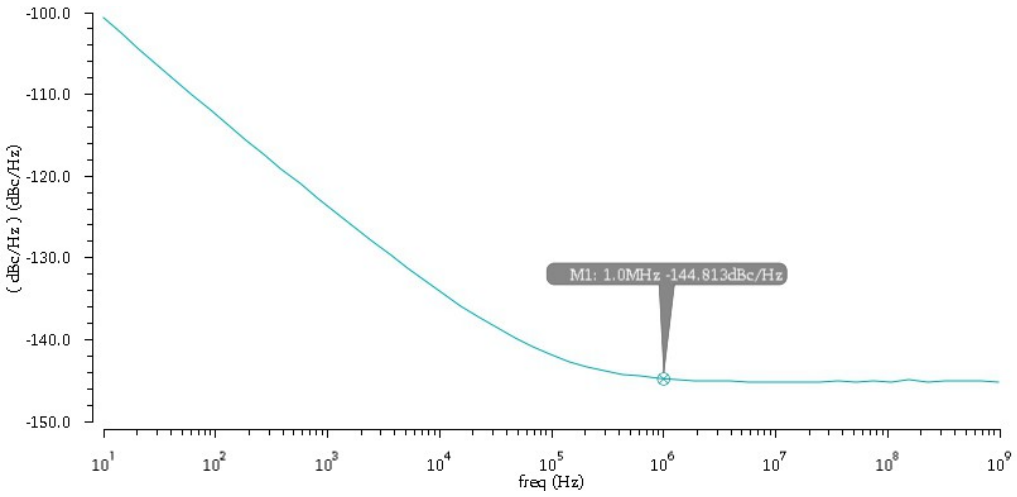

Figure 4.16: Pass Transistor PFD Phase Noise at 1 MHz Offset

In terms of phase noise, the pass transistor PFD perform worse than the conventional PFD. The best phase noise for this implementation has been measured to be -144 dBc/Hz at 1 MHz offset, while the worst is -136 dBc/Hz. However, in terms of power consumption, high frequency operation, and layout area, the pass transistor topology surpasses the performance of the conventional PFD. A summary of the performance comparison has been provided in the table 4.3.

Table 4.3: PFD Performance Comparison

Conventional PFD	Pass Transistor PFD
Power Consumption = 106 µW	Power Consumption = 3.27 µW
Layout Area = 54.9 µm X 33.82 µm	Layout Area = 11.21 µm X 16.27 µm
Worst Phase Noise at 1 MHz Offset = -139 dBc/Hz	Worst Phase Noise at 1 MHz Offset = -136 dBc/Hz

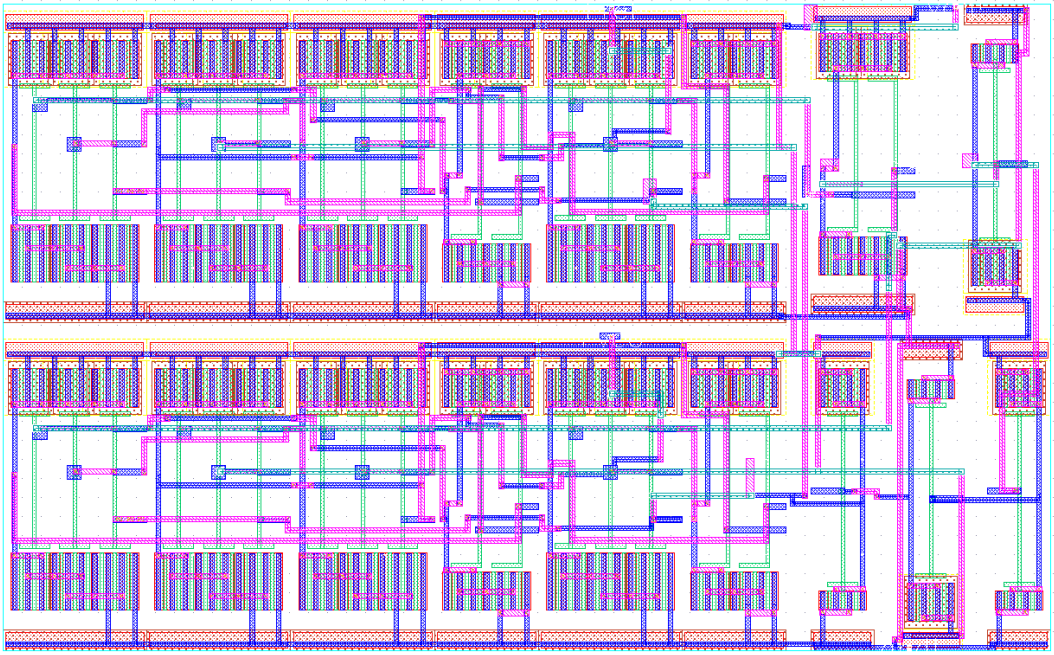

Figure 4.17: Layout of the D Flip Flop Based PFD

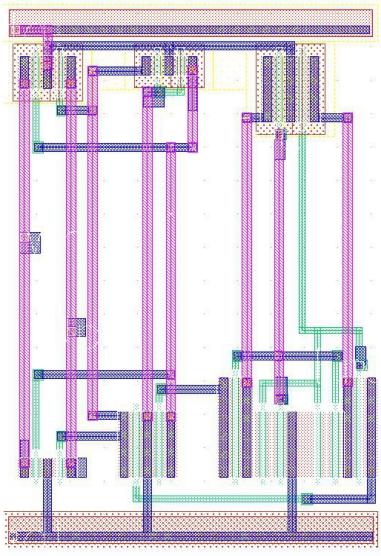

Figure 4.18: Layout of the Pass Transistor PFD

4.5 Summary

The basic operations of the phase-frequency detector, its ideal and non-ideal characteristics, transistor level implementation of two different architectures, and simulation of both implementations have been discussed in this chapter. Based on the performance of both PFDs, it is obvious that for the WLAN frequency synthesizer the pass transistor based PFD would be the best choice. Although some performance characteristics, such as the phase noise of the preferred topology, are not as great as the conventional topology, the slight increase in jitter and lock time should be overpowered by the elimination of the reference spurs and unwanted spikes caused by the dead-zone region.

Chapter 5

Charge Pump

5.1 Introduction

In this chapter we discuss the charge pump, an essential PLL block, which when used in conjunction with the phase-frequency detector, will produce current signals corresponding to the phase error signals generated by the PFD. This is a fundamental element in any kind of analog/mixed-signal/RF PLLs/frequency synthesizers as the control voltage of the VCO depends heavily on the consistent performance of the charge pump. This chapter also discusses some of the non-ideal effects that need to be accounted for while designing a functional charge pump circuit for a well performing frequency synthesizer.

5.2 Charge Pump Basics

The architecture of the charge pump and its most basic implementation have been shown in figure 5.1. In the simplest terms, a charge pump consists of two switches that are controlled by the “Up” and “Down” pulses generated by the PFD. We know from previous discussions that “Up” and “Down” signals should not be high simultaneously (except for non-ideal spikes)—which means only one of the two switches will be ON at a time. When the “Up” signal is high and the “Down” is low, SW_1 will be turned on, SW_2 will be turned off, and current I_{Up} will flow through the load capacitance and charge it up to the supply voltage. Similarly, when

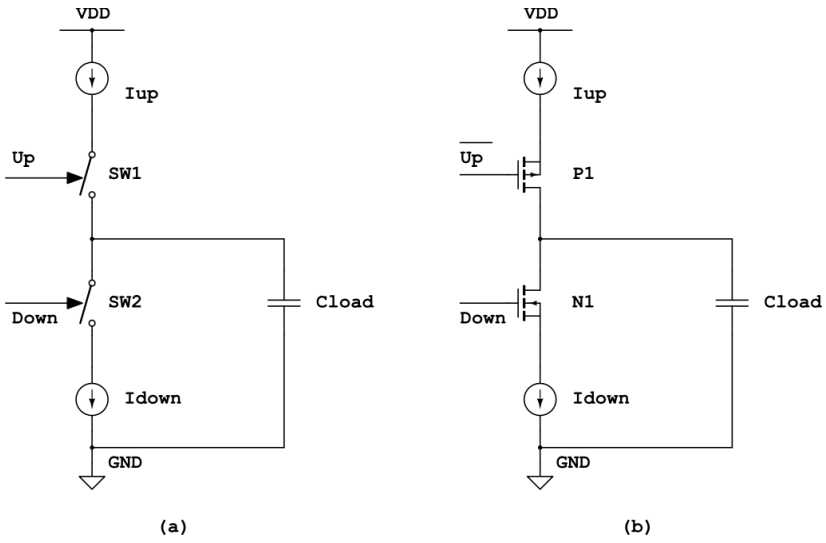

Figure 5.1: (a) Charge Pump Model, (b) Basic CMOS Implementation

“Down” signal is high, current I_{Down} will be sinked from the load capacitance to discharge it. If both “Up” and “Down” are high (equal

current will be sourced and sinked into the load) or low (switches turned off), the load should not experience any changes in current [4], [9], [32]. Figure 5.1(b) shows a basic implementation of the charge pump using a PMOS device switch for the current source and NMOS device switch for the current sink. Since PMOS devices turn on at low gate voltage state, complementary “Up” signal is used for switching [9], [37].

A well constructed charge pump should exhibit several different features, such as, equality in charge and discharge currents, compensation of “Up” and “Down” pulse skews and mismatches, minimization of charge injection and clock feed-through, and etc [4], [36], [37]. Due to the fact that the PFD designed for this thesis generates stable phase error pulses and exhibit minimum amount of dead-zones, focus was given more towards minimizing non-idealities, other than timing mismatch, during the charge pump design. In addition to the charge pump structure, a simple but sturdy design of beta multiplier current reference was constructed for biasing the charge pump.

Besides the ideal charge pump shown in figure 5.1, many different structures of charge pumps have been invented over the years; each of which has its own benefits. Charge pump architectures can be either fully differential, or differential input and single ended output, or single ended input and differential output [24]. For our synthesizer, a differential input and single ended output charge pump was deemed

suitable as the VCO's control voltage is single ended and it does not require an additional loop filter. Furthermore, “Up” and “Down” skews could also be alleviated with differential input.

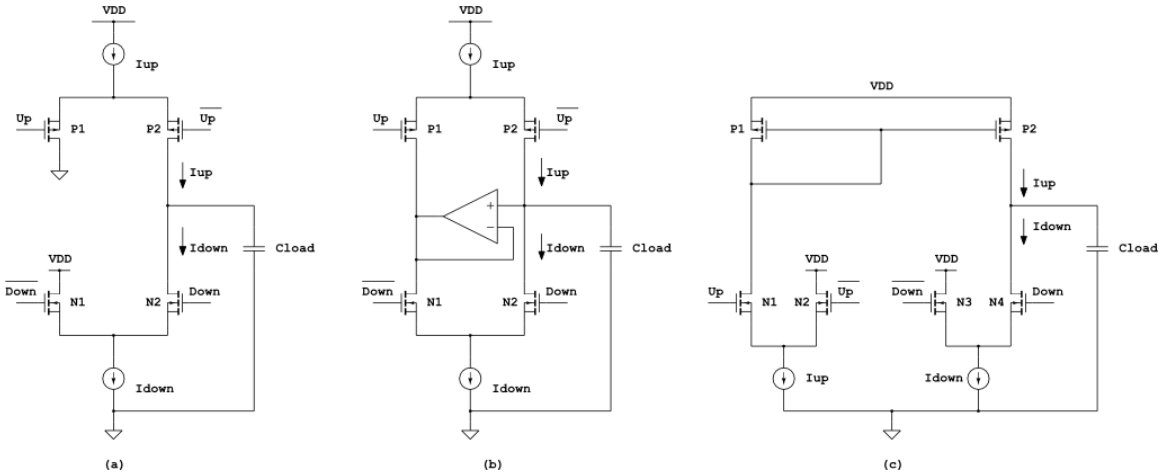

Figure 5.2: Current Steering Charge Pumps (a) Basic (b) with Op-Amp (c) NMOS Switch Only

The charge pump implemented with PMOS and NMOS switches suffers from switching time and current mismatches due to the uneven behavior of the devices. Additionally, the devices exhibit unequal leakage currents and unequal channel length modulations which can result in even more mismatches in the “Up” and “Down” currents. To mitigate these non-ideal effects, current steering technique has been adopted by designers [24], [36]. Figure 5.2 features some of the basic current steering charge pump architectures, of which, a variation of the architecture in 5.2 (c) was implemented for this thesis.

A current steering charge pump has the ability to improve switching speeds by reducing switching transients using both the phase error signals and their complements [4]. The two pairs of complementary signals make sure that the output produced by the charge pump does not stay in transition after the switch has been turned off. This reduces unwanted charging and discharging of the output load and hence reducing unwanted VCO tuning [38]. Another advantage of using a current steering charge pump is to reduce charge injection produced by the PMOS and NMOS switches. Since both the transistors have some charge in their inversion layer (unequal charges because of different dimensions and overdrive voltages), and they do not cancel each other out, these charges travel with the control voltage and unnecessarily tune the VCO [4], [5], [40].

The charge pump designed here has a similar structure to that of figure 5.2 (c). We can see from the figure that the charge pump has been implemented using four NMOS switches and a PMOS current mirror. When “Up” signal is high, current I_{Up} through PMOS P_1 is mirrored at PMOS P_2 which charges the load capacitor [24], [38]. Similarly, when “Down” signal is high, the current mirror is cut-off and the load capacitor is discharged with current sink I_{Down}. Since both currents are handled using NMOS switches, their timing characteristics, and their switching characteristics are far better compared to the basic implementation. Additionally, the architecture is simple enough and does not require an additional Op-Amp similar to figure 5.2 (b). Moreover, a low power Op-Amp is incapable of

producing a rail-to-rail voltage necessary for tuning the VCO—something which can be easily achieved with some improvements on the NMOS switch only current steering technique.

Another thing that needs to consider regarding the current steering method is that, the simple current mirror constructed using the P-type devices, has a tendency to produce inconsistent current. This is primarily due to its low output impedance, which in practice cannot sustain a constant current operation throughout different operating conditions [41]. Because of these fluctuations, ripples and/or glitches may appear at the loop filter input which may cause the PLL/frequency synthesizer to have a noisy output. In addition, output swings will be limited because of low output resistance—which in turn may cause charge and discharge current mismatches similar to that of the basic PMOS and NMOS switch charge pump. Nevertheless, this situation can be improved using a better current mirror structure such as, a high swing cascode structure. Cascode mirrors have high output impedances and produce constant currents across many operating conditions [38], [39].

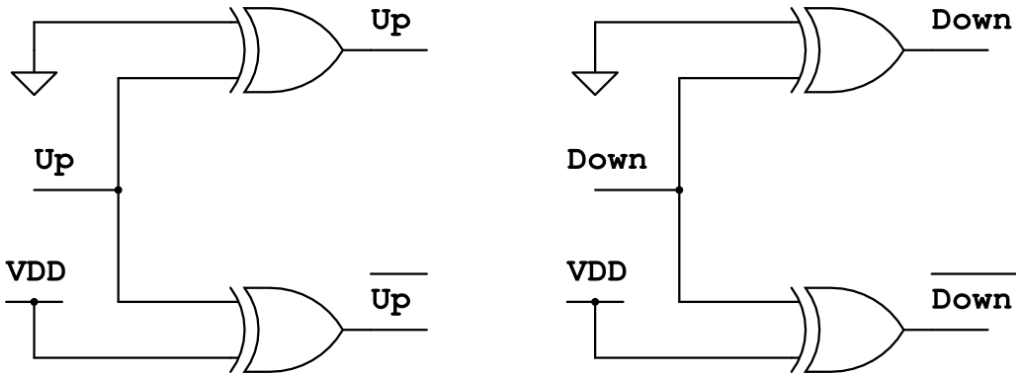

Figure 5.3: Aligning "Up" and "Down" Signals with Their Complements

Previously, it was mentioned that the PFD designed for the WLAN synthesizer has stable phase error pulses and introduces very low amount of noise/jitter into the VCO. However, the problem of clock skew of the complementary signals affecting the switching characteristics of the differential pairs still remains. To remedy this problem, few different methods can be approached. Figure 5.3 shows the method that was used in this work for generating the actual signal and its complement with the least amount of skew [24]. With this method, using only four XOR (exclusive OR) gates, we can ensure complete alignment of “Up” with its complementary signal, and “Down” with its complementary signal.

5.3 Circuit Design

The high swing cascode charge pump proposed by Zhang et al., using NMOS only switch has been illustrated in figure 5.4 [38]. This highly stable design that can achieve excellent matching of charge and discharge currents, was adopted for our synthesizer. Having a high output impedance, a very constant current output can be achieved with this design. Furthermore, due to having cascode mirrors for both “Up” and “Down” currents, rail-to-rail swings can be easily achieved [38], [41].

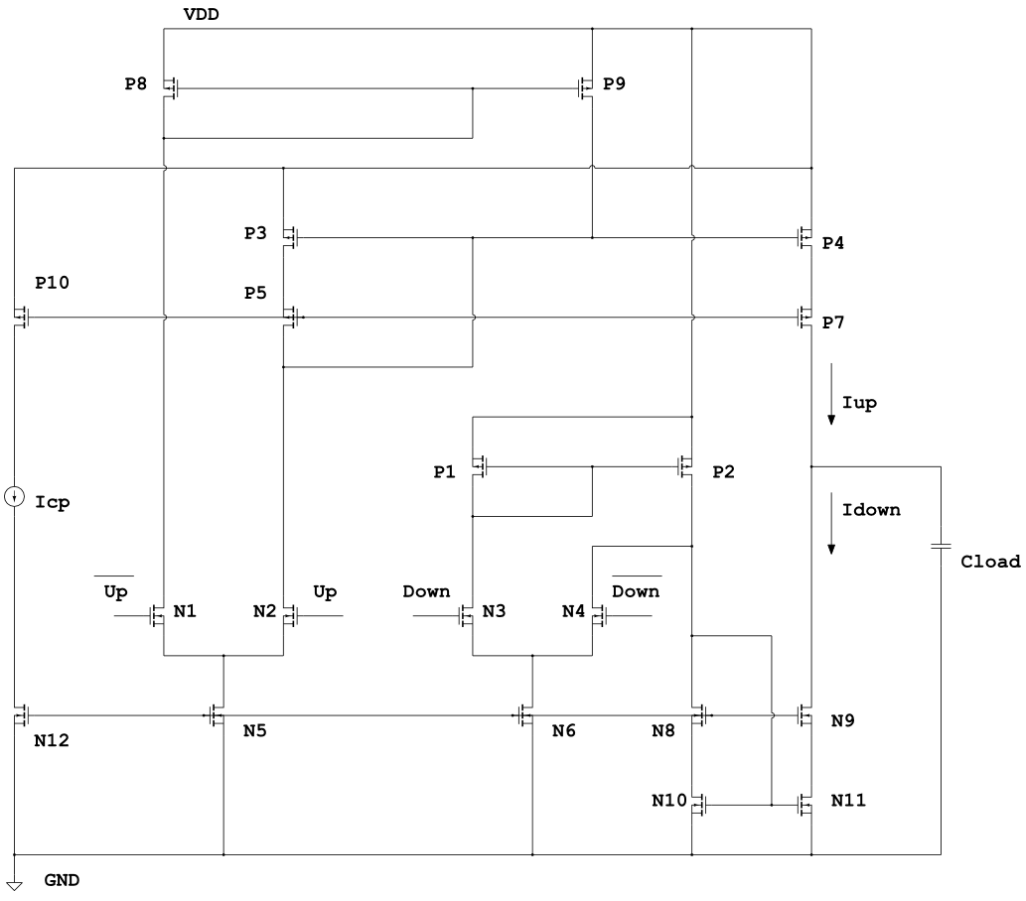

Figure 5.4: NMOS Switch Only High Swing Cascode Charge Pump

The working principle of this circuit is almost identical to that of the simpler NMOS switch only current steering charge pump in figure 5.2 (c). Using PMOS devices P_1 and P_3 to P_7, the cascode current mirror for charging ("Up" current, I_{Up}) has been constructed. Similarly for discharging ("Down" current, I_{Down}), the cascode current mirror was constructed using NMOS devices N_8 to N_{12}. Figure 5.5 shows the contrast between a simple current mirror and a high swing cascode current mirror. Looking at the current mirror structures, we can derive that the output resistance of the cascode mirror is $g_m r_o$ times more than the simple current mirror [41].

Output resistance basic current mirror: r_{o2}

Output resistance cascode current mirror: $g_m r_{o3} r_{o5} \approx g_m r_o^2$

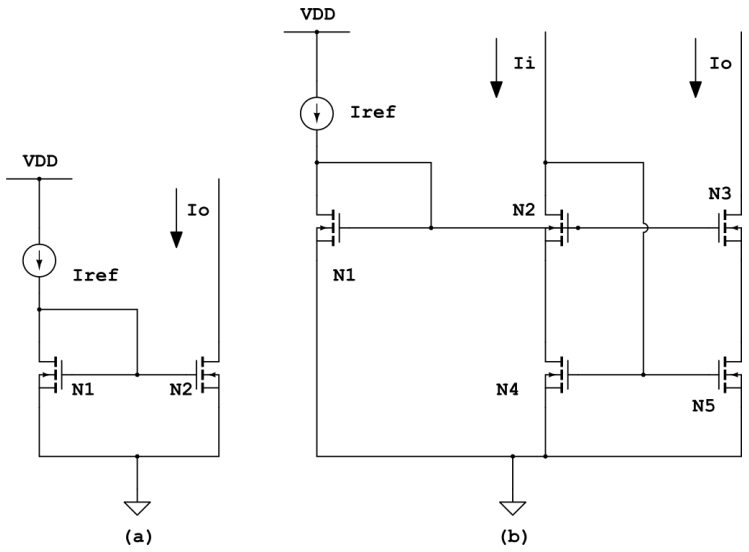

Figure 5.5: (a) Simple Current Mirror (b) High Swing Cascode Current Mirror

Going back to the schematic of the charge pump, we find that two simple PMOS current mirrors constructed with P_1, P_2, and P_8, P_9 have been placed at the gate of the cascode mirrors. This was done to reduce output transients by either pulling the P_3, P_4 node up to V_{DD} when “Up” is low, or to pulling the N_{10}, N_{11} node down to GND when “Down” is low [38]. Doing so provided a more accurate transfer of charge and discharge current and further improved mismatches.

To provide the charge pump with a reference current I_{CP}, a self biased beta multiplier current reference has been designed. Figure 5.6 shows the schematic of the current reference that generates a supply independent current of 100 μA. We can observe from the schematic that the threshold voltages of N_1 and N_2 subtract out, which also make this design suitable if there are any process and/or temperature shifts [9], [42]. Furthermore, the resistor R has been replaced with an NMOS device, g_m of which can be controlled and can be used to modify the reference current as needed. Using PMOS P_3, current source for NMOS current mirror is provided and using the NMOS N_3, current sink for PMOS mirror is provided. Table 5.1 has been compiled to present all the design variables of the charge pump and the beta multiplier reference. It should be noted that the 100 μA charge pump current was determined during the system level analysis which works well with the VCO gain, K_{VCO}.

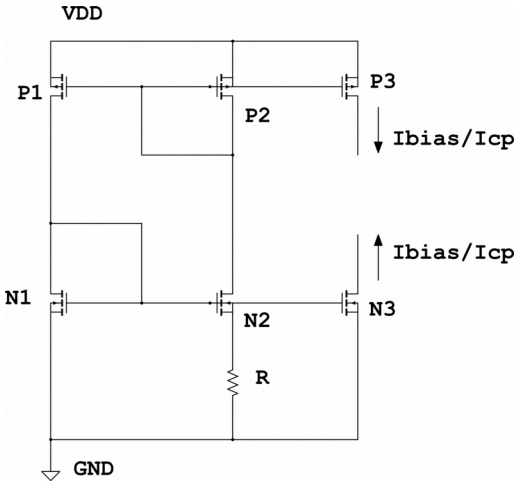

Figure 5.6: Self Biased Beta Multiplier Current Reference

Table 5.1: Charge Pump and Current Reference Design Variables

Charge Pump	
PMOS, $P_8 = P_9$	$\frac{W}{L} = \frac{0.6}{0.36}$
PMOS, $P_1 = P_2$	$\frac{W}{L} = \frac{18}{0.18}$
PMOS, $P_3 = P_4 = P_5 = P_7$	$\frac{W}{L} = \frac{13}{0.36}$
PMOS, P_{10}	$\frac{W}{L} = \frac{4}{0.6}$
NMOS, $N_1 = N_2 = N_3 = N_4$	$\frac{W}{L} = \frac{2}{0.18}$

NMOS, $N_5 = N_6$	$\frac{W}{L} = \frac{3}{0.55}$
NMOS, $N_8 = N_9 = N_{10} = N_{11}$	$\frac{W}{L} = \frac{11}{0.36}$
NMOS, N_{12}	$\frac{W}{L} = \frac{2}{0.36}$
Beta Multiplier Reference	
PMOS, $P_1 = P_2$	$\frac{W}{L} = \frac{24}{1}$
PMOS, P_3	$\frac{W}{L} = \frac{16}{1}$
NMOS, $N_1 = N_2$	$\frac{W}{L} = \frac{12}{1.2}$
NMOS, N_3	$\frac{W}{L} = \frac{33}{0.86}$
Resistor/NMOS, R	$\frac{W}{L} = \frac{6}{1}$

5.4 Simulation Results

A test bench has been built for all simulations using the phase difference outputs from the phase-frequency detector as inputs to the charge pump, and an ideal load capacitor of 10 pF to the output. Using DC analysis, and switching only one differential pair at a time, the amount of the “Up” and “Down” currents flowing to and pulling from the load capacitor were verified. The DC mismatch between the two currents has been found to be almost non-existent. In addition, DC simulation revealed that the charge pump and the self biased reference, when both “Up” and “Down” pulses are active, consume only about 800 μW of power with a supply voltage of 1.2 V.

Following the DC simulation, a transient simulation for 100 ns with 5 ns “Up” and/or “Down” pulses was performed to observe the mismatches between the two currents. It can be seen on figure 5.7 that the charging and discharging currents are almost equal at 100 μA, and switching has little to no delay in settling. Moreover, for a 5 ns voltage pulse, the current pulse was measured to be about 4.8 ns, which is sign of satisfactory switching characteristics.

Figure 5.8 illustrates the charge pump's four different modes of operation. When either “Up” or “Down” pulses are switching exclusively, the load capacitor respectively becomes fully charged or becomes completely discharged. In situations when both pulses are generated

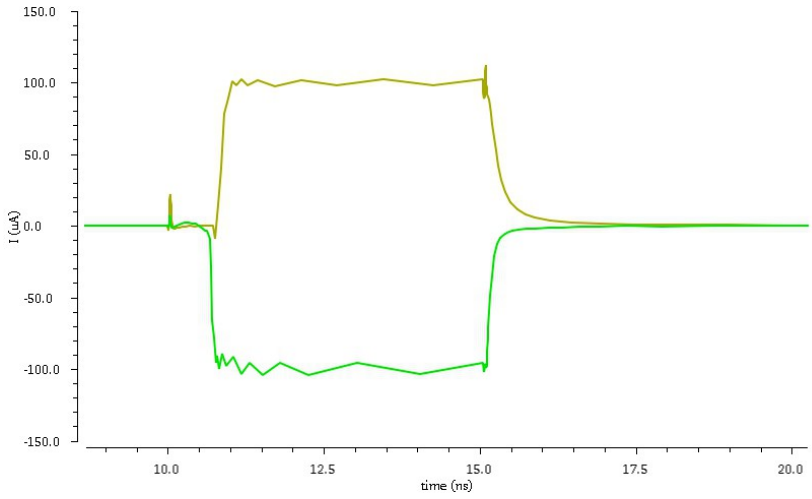

Figure 5.7: Matched Charge (Yellow) and Discharge (Green) Currents

concurrently, or due to PLL's locking no pulses are generated, the load capacitor holds a stable voltage of approximately 600 mV. This data exhibits the charge pump's fast switching, and mismatch minimization capabilities. Further analysis on figure 5.9 determine that the load capacitor's charge and discharge voltages match at approximately 600 mV —which supports the reliability of the charge pump's performance.

Also, on figure 5.10 we can observe the accurate charging of the load capacitor only during the switching. The charging curve has almost no noticeable glitches/ripples and the switching current is quite fast. In addition, it affirms that leakage currents and clock feed-through are not affecting the current output as it would have if it were only a simple charge pump.

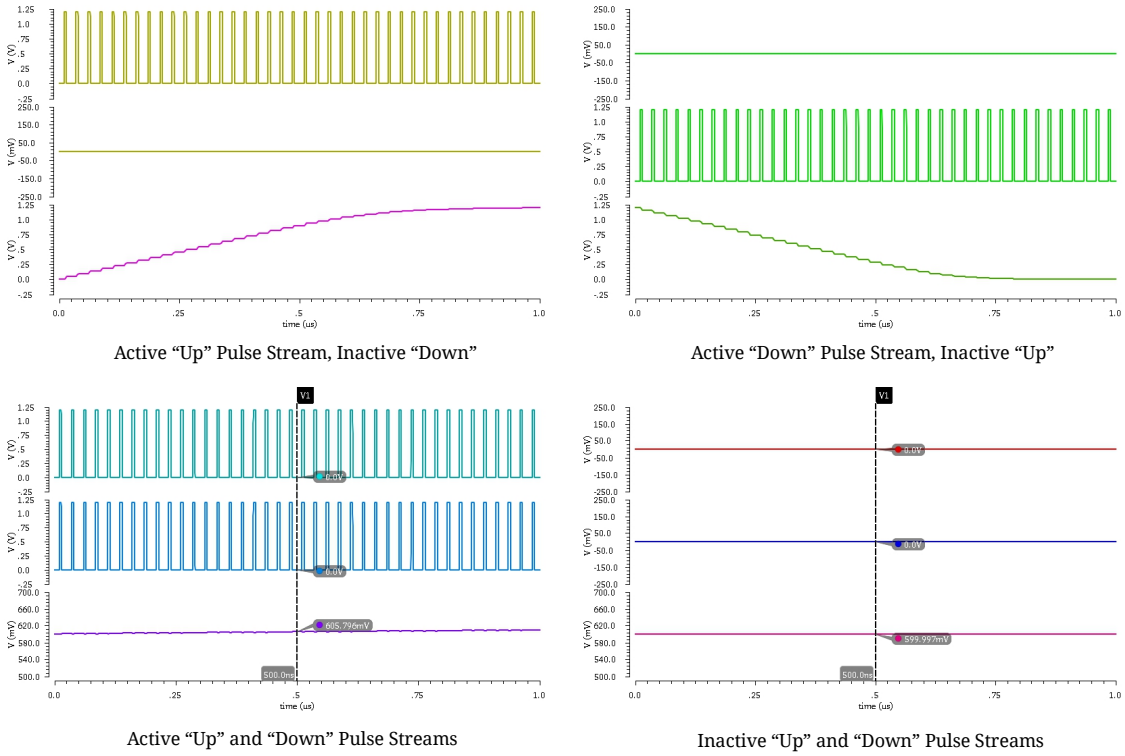

Figure 5.8: Four Different Modes of Operation

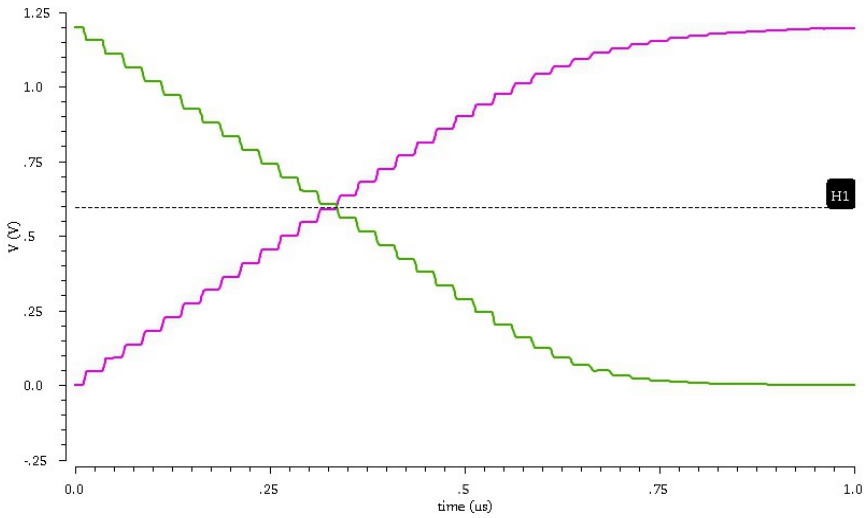

Figure 5.9: Charging (Pink) and Discharging (Green) of the Load Capacitor

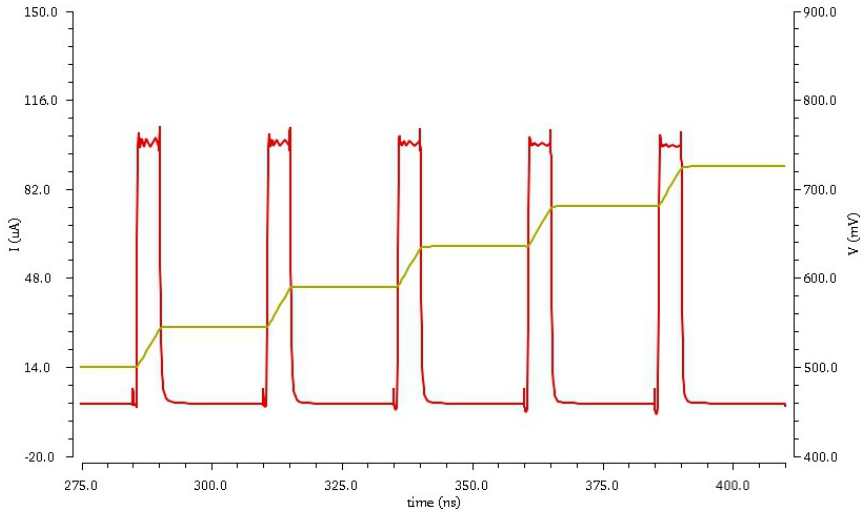

Figure 5.10: Current Pulses and Charging Voltage

Finally, figure 5.11 shows the current noise level within the frequency region of our frequency synthesizer's operation. At 6 MHz, the noise level is at -243 dBc, which is low enough to meet the synthesizer's requirements.

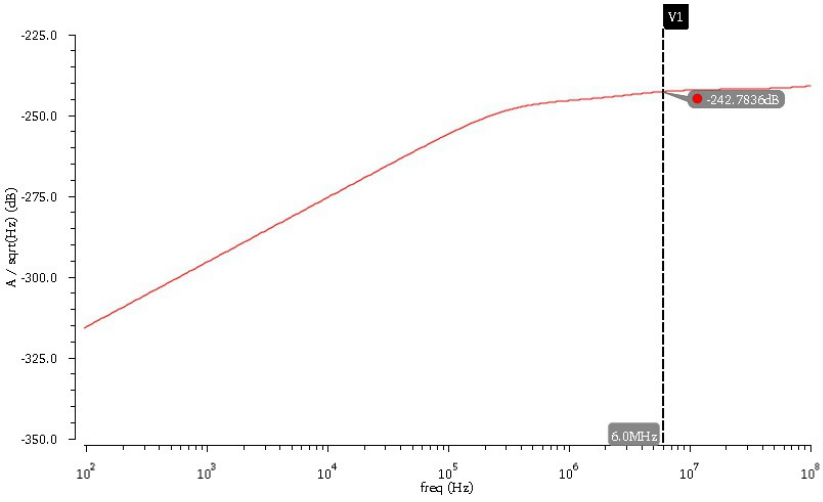

Figure 5.11: Output Current Noise

5.5 Summary

In this chapter we discussed about the basics, the non-idealities, the design, and the simulation of a high swing cascode charge pump circuit. For our WLAN application, it was determined that a charge pump needs to be implemented for 100 µA current. The highly stable design of the 57.48 µm X 31.42 µm (figure 5.12) charge pump has been found to consume about 800 µW of power and have the least amount of current mismatch. Moreover, with the high swing cascode architecture, the charge pump was able to achieve a constant current output throughout all four modes of operation.

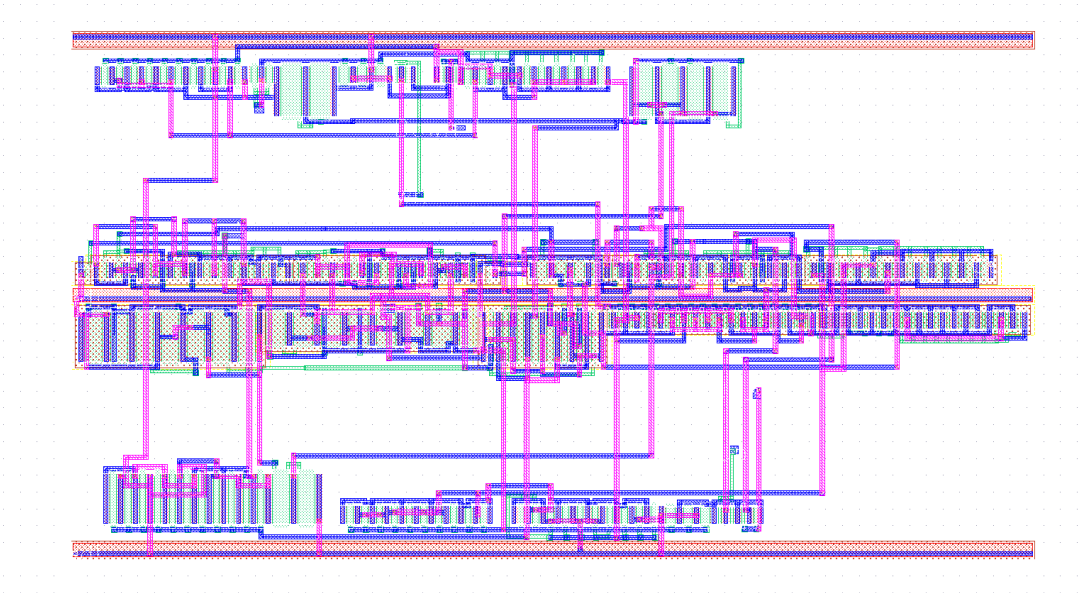

Figure 5.12: High Swing Cascode Charge Pump and Self Biased Reference Layout

Chapter 6

Loop Filter

6.1 Introduction

One of the most important parts of designing a PLL or a frequency synthesizer is to design its loop filter. The whole synthesizer's performance including its open and closed-loop gains, phase margin, and most importantly, stability depends on the loop filter's coefficients. Since most stability analyses have been included in the 2nd chapter, this chapter will deal with some basic functionality and the design methodology of the 3rd order loop filter for our 3rd order type-II PLL synthesizer.

6.2 Design Methodology

The primary objective of a loop filter is to filter out any unwanted high frequency components that may over-modulate the VCO and throw the PLL out of balance [50]. Even if the VCO tends to stay stable, the unwanted noises may still pass through a badly designed loop filter and appear as spurious tones at the output of the frequency synthesizer. Based on the design specifications of a PLL, a loop filter may either be active or passive. In many low frequency applications, the preferred choice of loop filter would be an active filter, constructed of operational amplifiers. Since our fractional-N synthesizer operates at gigahertz range, in terms of power budget and complexity, it would be expensive to use active components. With all constraints considered, a passive 3rd order loop filter, illustrated in figure 6.1, has been designed in this thesis.

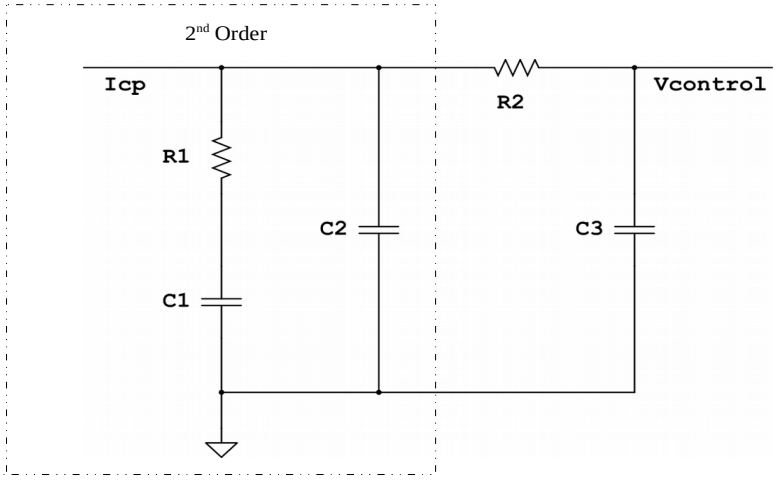

Figure 6.1: Passive 3rd Order Loop Filter

A typical design process of a passive loop filter require several initial parameters from the system level analysis of the frequency synthesizer [51]. In the past few chapters, we have come across the following design parameters, which will be used to generate the appropriate values of the resistive and capacitive components.

Table 6.1: Initial Design Parameters

VCO Gain, K_{VCO}	180 Mhz/V
PFD/Charge Pump Constant, K_{ϕ}	100 μA/2π rad
Reference Frequency, F_{REF}	40 MHz
Maximum Output Frequency, F_{VCO}	5.825 GHz
Highest Division Ratio, N	~ 146

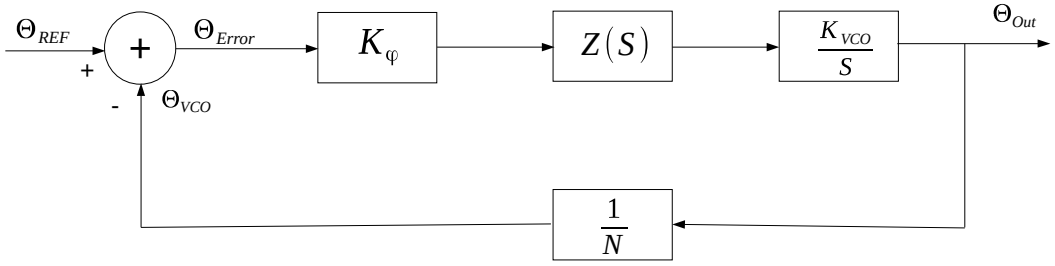

Figure 6.2: Simple Linear Model of PLL [51]

From the simple linear model of PLL in figure 6.2, we can determine the equations (slightly different than the ones in chapter 2) listed table 6.2.

Table 6.2: Phase Transfer Functions

Forward Loop Gain, G(s)	$\frac{\Theta_{Out}}{\Theta_{Error}} = K_{\varphi}.Z(s).\frac{K_{VCO}}{s}$
Reverse Loop Gain, H(s)	$\frac{\Theta_{VCO}}{\Theta_{Out}} = \frac{1}{N}$
Open Loop Gain, H(s) G(s)	$\frac{\Theta_{VCO}}{\Theta_{Error}} = K_{\varphi}.Z(s).\frac{K_{VCO}}{N\,s}$
Closed Loop Gain, $\Theta_{Out}/\Theta_{REF}$	$\frac{G(s)}{[1+H(s).G(s)]}$

Using methodology provided by Texas Instruments, Banarjee, and Shu et al., the loop filter's parameters were calculated [51], [36], [24]. First of all, the schematic provided in figure 6.1 show that without components R_2 and C_3, the filter would act as a 2nd order filter. For a second order system, we can find the filter's impedance as:

$$Z(s) = \frac{s.C_1.R_1 + 1}{s^2.C_1.C_2.R_1 + s.C_1 + C_2} \qquad (6.1)$$

From equation (6.1), we can define the time constants for determining the pole and zero frequencies of the filter's transfer function:

$$T_1 = R_1\frac{C_1.C_2}{C_1 + C_2} \qquad (6.2)$$

$$T_2 = R_1.C_1 \qquad (6.3)$$

Now, in terms of frequency ω, the time constants T_1 and T_2, and the parameters from table 6.2, we can derive the open loop gain of the PLL as:

$$G(s)H(s) = \frac{-K_{\varphi}.K_{VCO}(1 + j.\omega.T_2)}{\omega^2.C_2.N(1 + j.\omega.T_1)}.\frac{T_1}{T_2} \qquad (6.4)$$

It is evident from equation (6.4) that the phase of the loop filter will be contingent upon the poles and the zeros. In this case, we can find the phase margin as:

$$\varphi(\omega) = \tan(\omega.T_2).\tan(\omega.T_1) + 180^o \qquad (6.5)$$

In order to find the relationship between the PLL's loop bandwidth and the time constants, the derivative of equation (6.5) needs to be equated to zero, which then gives us:

$$\omega_p = \frac{1}{\sqrt{T_1.T_2}} \qquad (6.6)$$

Additionally, phase margin needs to be determined to ensure loop stability—which can be found by equating the open loop gain to 1. Doing so, we find from equation (6.4) that:

$$C_2 = \frac{K_{\varphi}.K_{VCO}.T_1}{\omega_p^2.N.T_2}.\frac{1 + j.\omega_p.T_2}{1 + j.\omega_p.T_1} \qquad (6.7)$$

From the synthesizer's specifications we know the loop bandwidth ω_p and the phase margin ϕ_p. With these parameters using the equations in this chapter so far, we derive the time constants as:

$$T_1 = \frac{sec(\varphi_p) - \tan(\varphi_p)}{\omega_p} \tag{6.8}$$

$$T_2 = \frac{1}{\omega_p^2 . T_1} \tag{6.9}$$

We then derive the values of the of resistor and the two capacitors in the 2nd order filter in terms of time constants:

$$C_2 = \frac{T_1}{T_2} . \frac{K_\varphi . K_{VCO}}{\omega_p^2 . N} . \sqrt{\frac{1 + (\omega_p . T_2)^2}{1 + (\omega_p . T_1)^2}} \tag{6.10}$$

$$C_1 = C_2 . (\frac{T_2}{T_1} - 1) \tag{6.11}$$

$$R_1 = \frac{T_2}{C_2} \tag{6.12}$$

Now, going back to the 3rd order loop filter in figure 6.1, because of an added resistor and a capacitor, we find that there is an additional time constant associated with it. These added components add an extra degree of attenuation, expressed as:

$$\text{Attn} = 10 \log[(2\pi . F_{REF} . R_2 . C_3)^2 + 1] \tag{6.13}$$

The transimpedance transfer function of the 3rd order filter is:

$$Z_T(s) = \frac{Z(s).\frac{1}{s.C_3}}{Z(s) + R_3 + \frac{1}{s.C_3}} \tag{6.14}$$

Because of this extra time constant, the 2nd time constant will also be affected. In which case we find,

$$T_3 = R_2.C_3 \tag{6.15}$$

$$T_3 = \sqrt{\frac{10^{\left(\frac{Attn}{10}\right)} - 1}{(2\pi.F_{REF})^2}} \tag{6.16}$$

$$T_2 = \frac{1}{\omega^2.(T_1 + T_3)} \tag{6.17}$$

And, the open loop gain in equation (6.4) becomes:

$$G(s).H(s) = \frac{-K_\varphi.K_{VCO}(1 + j.\omega.T_2)}{\omega^2.C_2.N(1 + j.\omega.T_1)} \cdot \frac{T_1}{T_2} \cdot \frac{1}{1 + j.\omega.T_3} \tag{6.18}$$

At this stage, looking at equations (6.13) to (6.17), we find that the open loop unity gain frequency is:

$$\omega_c = \frac{\tan(\varphi).(T_1 + T_3)}{(T_1 + T_3)^2 + T_1.T_3}\left[\sqrt{1 + \frac{(T_1 + T_3)^2 + T_1.T_3}{(\tan(\varphi).(T_1 + T_3))^2}} - 1\right] \tag{6.19}$$

And the 2nd capacitor becomes:

$$C_2 = \frac{T_1}{T_2} \cdot \frac{K_\varphi . K_{VCO}}{\omega_c^2 . N} \cdot \sqrt{\frac{1 + \omega_c^2 . T_2^2}{(1 + \omega_c^2 . T_1^2)(1 + \omega_c^2 . T_3^2)}} \qquad (6.20)$$

According to the authors of the references [24], [36], [51], for satisfactory loop stability we should have:

$$C_3 = \frac{C_2}{10} \qquad (6.21)$$

$$R_2 = \frac{T_3}{C_3} \qquad (6.22)$$

Again, for stability reasons, Shu et al., suggests that the loop bandwidth should be at least 1/10th of reference frequency. So, in order to achieve a reasonable settling time, we find the loop bandwidth as:

$$\omega_p = \frac{8\pi}{T_L} \qquad (6.23)$$

Finally, for a phase margin (ϕ_p) of 55^o, added attenuation of 20 dB, and a settling time (T_L) of 10 μs, and using the equations for the 3rd order filter, the design variables of the loop filter were derived. The values are presented in table 6.3. It should be noted that, to take advantage of the fractional-N topology, the loop bandwidth was specified as 600 KHz, which is 50% more than the calculated value.

Table 6.3: Initial Design Parameters

T_1	8.363 X 10^{-8} s
T_3	3.959 X 10^{-8} s
ω_c	2.51 X 10^{6} rad/s
T_2	1.289 X 10^{-6} s
C_2	4.189 pF
C_1	60.38 pF
R_1	21.35 kΩ
C_3	418.8 fF
R_3	94.51 kΩ

6.3 Simulation Results

In general, the loop filter is used directly with the phase-frequency detector and the charge pump. What it means that, there are only a few parameters, that can be simulated from a testbench of the loop filter. Ideal behavioral results for the loop gain (transfer function) and the phase margin are shown in figure 6.3. Then we have determined the transfer function of the loop filter in SPICE simulator in order to compare the performance with the ideal behavior, results of which are in figure 6.4. Two different analyses, namely periodic transfer function (PXF) and periodic phase noise (PNoise), were performed to find the accuracy of the transfer function between two different algorithms.

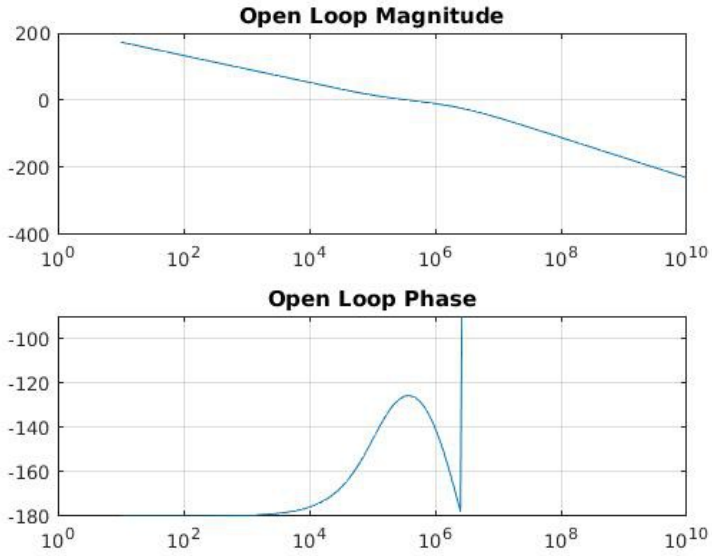

Figure 6.3: Loop Gain and Phase Margin (Behavioral)

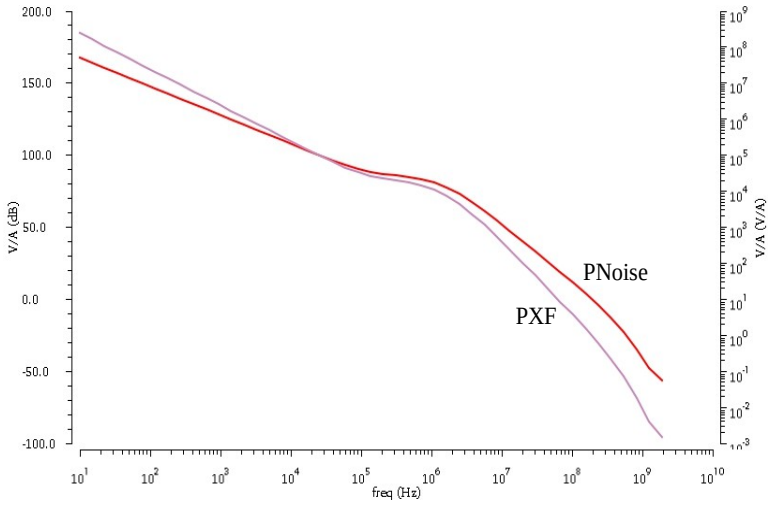

Figure 6.4: Loop Filter's Transfer Function

In figure 6.5, we see the filter's damping factor through step responses for -100 µA, and 100 µA currents.

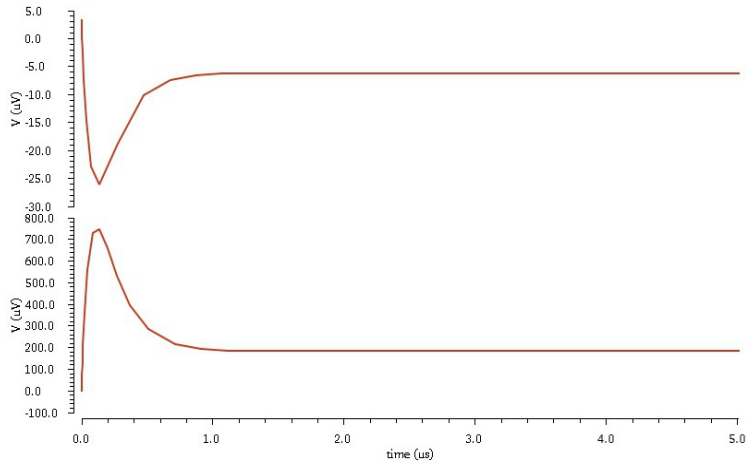

Figure 6.5: Step Response for "Up" and "Down" Currents

Because of the phase noises contributed by the PFD and the charge pump, as well as the spikes of the charge/discharge currents being pushed or pulled out of the loop filter, the filter will suffer from some input

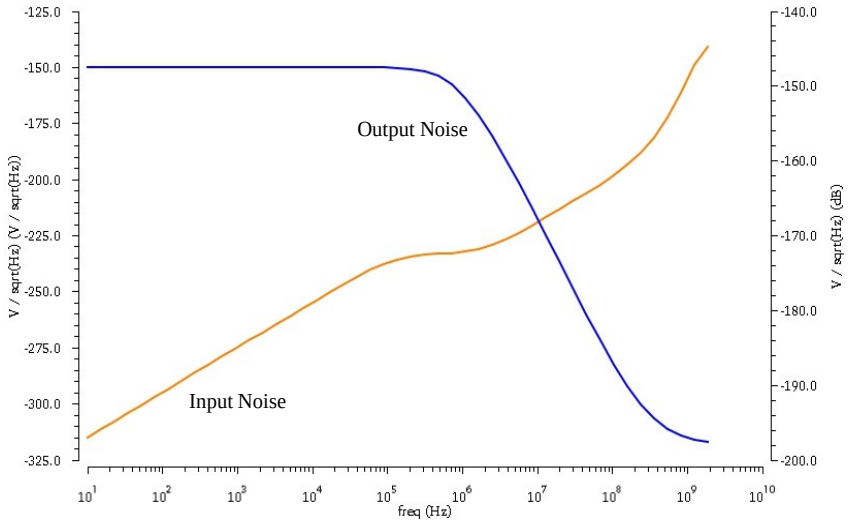

Figure 6.6: Input and Output Noises of the Loop Filter

referred noise, and generate some output noise for the subsequent stages of the PLL. Presented in figure 6.6 we see that, the input noise experienced by and the output noise generated by the loop filter are within a reasonable range of less than -150 dB.

Finally, loop filters are also known for their phase noise gain—which typically and is unfortunately, larger compared to the rest of the synthesizer. However, due to having a loop bandwidth close to recommended and implementing a fractional-N architecture, we can assume that the performance of the phase noise gain should be capable of sustaining a stable and low jitter output at the VCO while meeting the specification for WLAN standards. The phase noise gain performance is provided in figure 6.7.

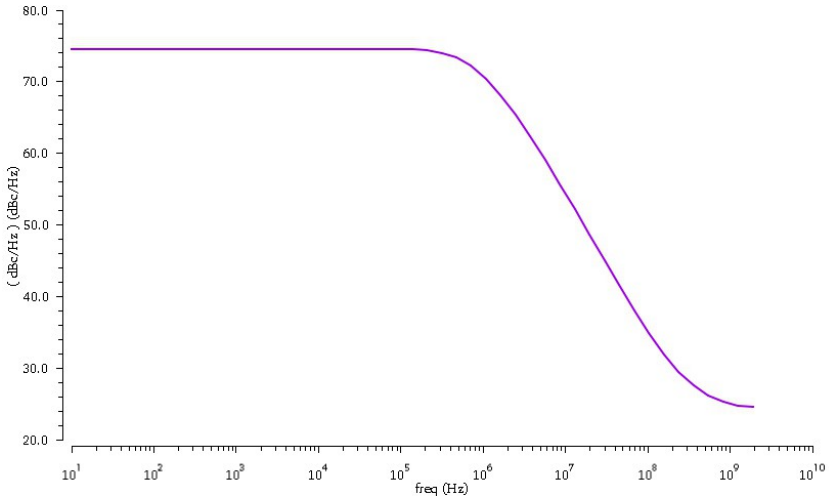

Figure 6.7: Phase Noise Gain of Loop Filter

Besides these previous analyses, DC simulation was also performed, which indicated that the worst case power consumption of the passive filter is approximately, 465 μW. We can also see the layout of the loop filter spanning across an area of 0.068 mm X 0.139 mm in figure 6.8.

6.4 Summary

In this chapter we presented a basic design procedure for the passive 3rd order loop filter for a type-II PLL based frequency synthesizer. Inspired by works of multiple authors, the design methods of the loop filter and its simulated performance have been carefully detailed here. It should be noted that this chapter is directly related to the stability analysis performed in chapter 2. So, in order to better understand the equations provided, reference to chapter 2 should to made.

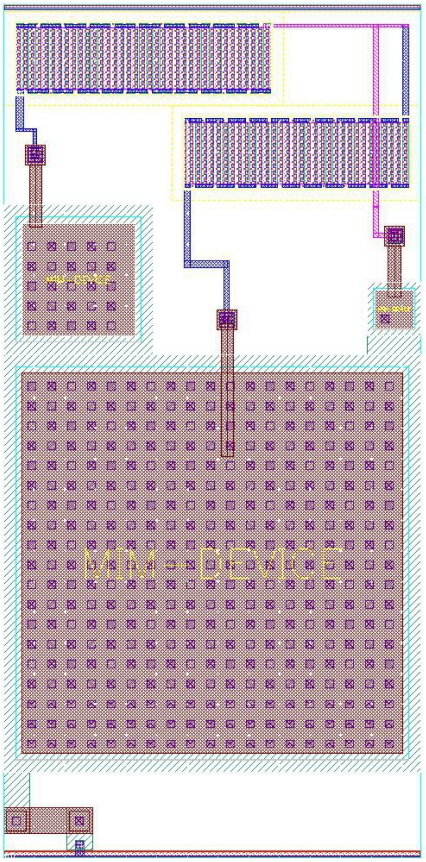

Figure 6.8: Layout of Loop Filter

Chapter 7

Programmable Frequency Detector

7.1 Introduction

Frequency dividers are essential for a phase-locked-loop to achieve frequency synthesis abilities. This is the block that can replicate the stability of the low reference frequency into the synthesizer's high output frequency. Although the frequency divider can be built for a single divide ratio, in this chapter we discuss how the ratio can be made programmable and how this programmability is employed to cover all the frequency ranges in the 2.4 GHz and the 5 GHz wireless LAN bands. Discussions on design, simulation, and performance of different components of the frequency divider will also be presented in this chapter.

7.2 Programmable Frequency Divider Basics

Frequency divider refers to any system that has the ability to divide a certain frequency by a certain number. Due to the divider being a bridge between the PFD and the VCO, the divider needs to operate at the frequency that the VCO generates [12]. As previously discussed, in order to lock the synthesizer, the divider needs to divide the VCO frequency by a certain number that will generate a frequency equal to the 40 MHz reference frequency. With that logic, we can infer that the division ratio "N", is actually being multiplied to the reference frequency in order to generate the desired output at the VCO. For example, if our communication system require a local oscillator of 5.76 GHz, for a 40 MHz reference oscillator, we need to implement a divide ratio N of 144.

Now that we understand the concept of the frequency divider, we also need to understand the principle of its programmability. The channels in the 2.4 GHz WLAN band range from 2.412 GHz to 2.484 GHz, and the 5 GHz channels range from 5.035 MHz to 5.825 GHz. The VCO designed for this work has an output frequency range of 4.77 GHz to 5.9 GHz. So, in order to cover all the channels in both bands using a 40 MHz reference, it can be deduced that the divider's division ratio should range between 60.3 and 145.625.

Since we are implementing a fractional-N frequency synthesizer, we are expected to have a divide ratio that is a fractional

number. Although in real life situations, we cannot have a frequency divider that can divide by a fractional number—however, as discussed in chapter 2, the divider needs to switch its divide ratio periodically to behave as a fractional divider. The programmable divider that has been implemented for this work has two unique parts. Illustrated in figure 7.1, we notice that the frequency divider consists of a divide-by-4 prescaler and a programmable divider. It should be noted that not all of the division ratios are programmable—reason of which are to facilitate the advantage of the fractional-N architecture and to reduce the phase noise/jitter that will follow the large number of divide ratios [12], [24], [44]. The CML divider/prescaler is built using current mode logic (CML) and operates at the same high frequency as the VCO's output. Since CMOS logic gates typically cannot operate at such high frequencies, the frequency needs to be scaled down to a suitable range using the prescaler.

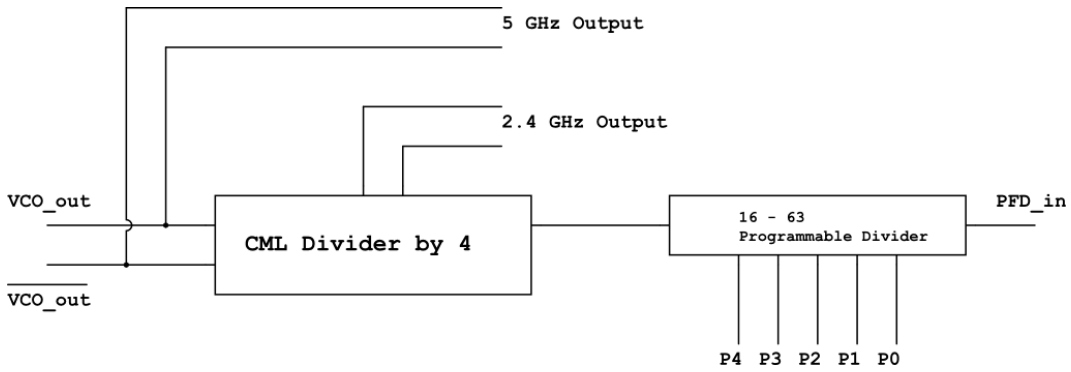

Figure 7.1: Frequency Divider Architecture

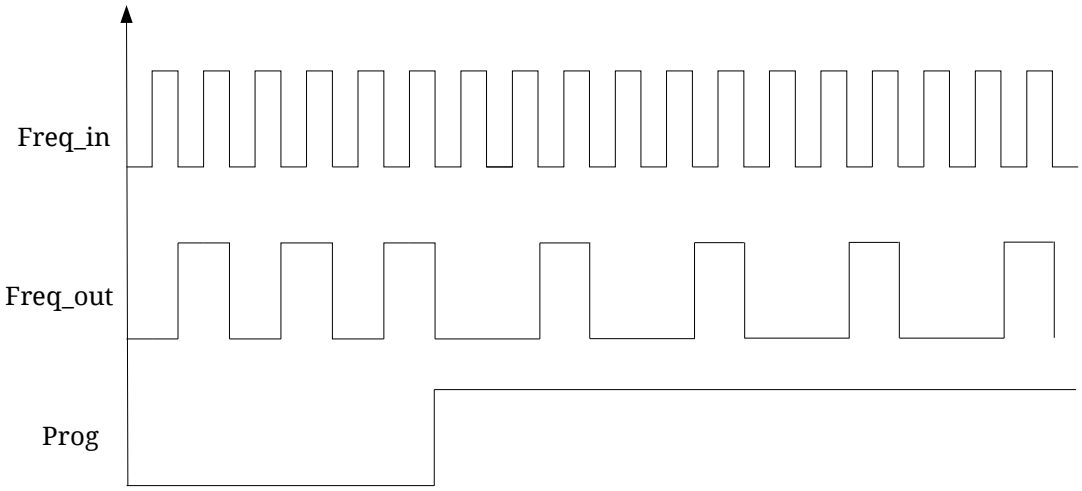

Figure 7.2: Waveform of 2/3 Prescaler Operation

The prescaler also take care of generating the 2.4 GHz channels and reduces the number of programming bits. The programmable divider that comes afterward, has been built using CMOS logic blocks, which comfortably operate at any frequency equal to or less than 1.5 GHz. With a 5 bit program code, the programmable divider can divide the input frequency with divide ratios ranging from 16 to 63. In conjunction with the prescaler, the range of divide ratio ultimately ranges from 64 to 252.

Another concept that should be understood before going into the details of the sub-blocks, is the concept of dual-modulus division. The programmable divider consists of several 2/3 dual-modulus prescalers—function of which are to divide the input frequency either by 2 (N) or by 3 (N+1). By periodically dividing the input frequency with both 2 and 3, we can achieve the fractional division ratio. So, using a combination of several

2/3 prescalers we can achieve a fractional division ratio solving the frequency resolution issues with the integer-N architecture [5], [12]. The concept of 2/3 dual-modulus prescaler has been illustrated in figure 7.2. We observe that when the “Prog” signal is low, the prescaler divides the input signal by 2, and when it is high, the division ratio becomes 3.

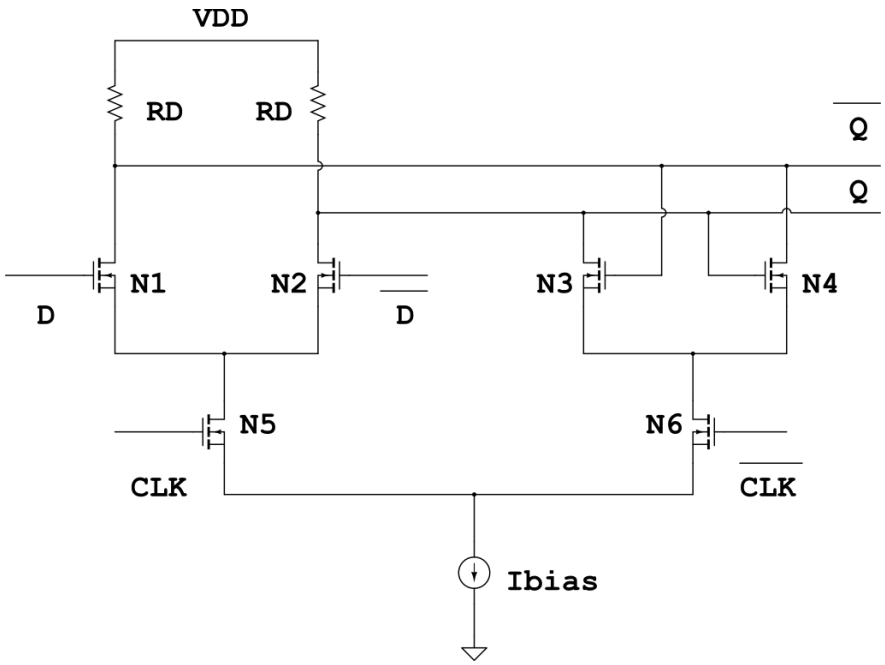

Figure 7.3: CML D-Latch

7.3 Circuit Design

The high speed portion of the frequency divider is a divide by 4 prescaler made of CML. The prescaler consists of two cascaded divide by 2 blocks, each block consisting of two CML latches put in a regenerative negative feedback loop. Figure 7.3 shows the structure of the CML D latch, that is used as both the master and the slave latches in a divide-by-2 D flip-flop. The latch is made up of a differential pair, a regenerative pair, and a clocked pair [4], [45]. The circuit also includes two load resistances at the drains of the differential pair and the regenerative pair in order to improve gain at certain stages of the latch's operation. One thing to consider while designing any CML gate is that the input at the gates must be differential and must have large enough swing for the gates to generate stable outputs [46]. As our VCO's output swings from 800 mV to 1 V, the CML latches should be designed such that, they operate reasonably within that common mode input range and generate sufficient output swings for the subsequent stages. The design process should also take into account the input capacitance of the latch that was used as the load capacitance for the VCO, and the output capacitance used as the input capacitance for next stage.

The operating principle of the CML latch has two modes: sense and regenerative [4], [46]. During the sense mode when CLK is high (N_5 ON) and $\overline{\text{CLK}}$ is low (N_6 OFF), the differential pair (N_1, N_2) senses the differential inputs and amplifies the difference between the two signals at Q and $\overline{\text{Q}}$

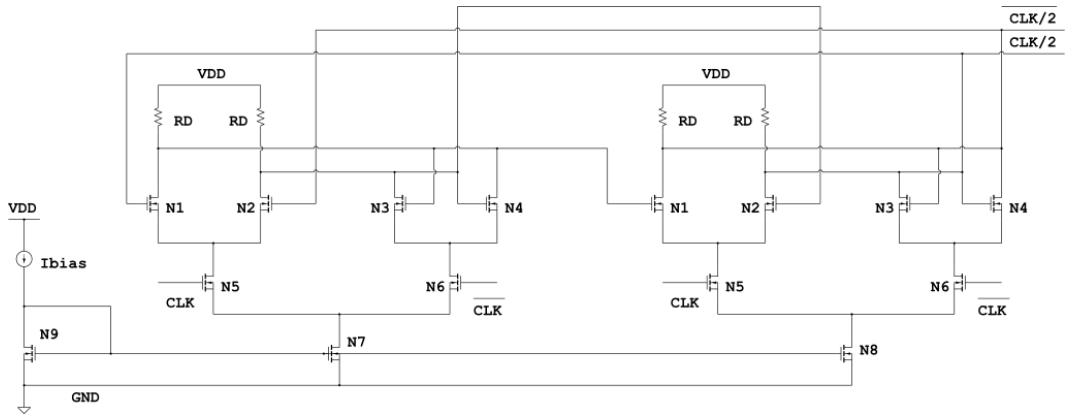

Figure 7.4: CML Divide-By-2 Circuit

nodes. When CLK goes down and CLK goes up, the circuit changes to regenerative mode. At this point the regenerative pair (N_3, N_4) latches onto the Q and $\overline{Q}$ signals produced during the sense mode and regeneratively amplifies them and holds the state. The state is held until CLK goes up again turning N_5 ON and turning N_6 OFF. The high speed operation of the circuit can be realized by understanding that during the transition from sense mode to latch mode, the regenerative pair keeps Q and $\overline{Q}$ in amplification, reducing the dependency on the CLK signal to arrive at the precise moment. Nonetheless, to ensure fast response throughout different modes of operation, the bias current, the NMOS devices, and the load resistances need to be valued and sized appropriately. The divide-by-2 master-slave configuration using CML D latches in a negative feedback loop has been illustrated in figure 7.4. The design procedure to implement such a configuration is as follows.

Observing from the figure, we find that the output of the first (master) latch connects to the input of the second (slave) latch, and the output of the second latch connects to the input of the first latch. Having known only a few design parameters, Razavi's method has been adopted to design the negative feedback configuration [4], [45]. To start off, the bias current has been set to 500 μA and $R_D I_{bias}$ has been set to approximately 500 mV. Transistor N_7, N_8, and N_9 were sized such that they suffer from the least amount of channel length modulation and have large enough transconductance to supply a stable DC current into the loads. After that, the differential pair (N_1, N_2) has been sized in a way that complete switching can be obtained for a differential input of 500 mV. With sizes that produce a small signal gain of more than unity, the regenerative pair has been designed. Finally, latch pair N_5, N_6 were designed to steer maximum amount of current to the correct direction (sense or regenerative) of the circuit in order to meet amplitude requirement of the output swings. Using a SPICE simulator thereupon, the circuit has been put to test and after a few trial and error iterations, the design variables in table 7.1 were obtained. Similar to the charge pump block of the frequency synthesizer, the current source for the CML blocks were implemented using beta multiplier reference (figure 7.5) topology—design variables of which are also provided in table 7.1.

Table 7.1: Design Variables for High Speed CML Divider

CML Divide-By-2 Circuit	
NMOS, $N_1 = N_2$	$\frac{W}{L} = \frac{6.4}{0.18}$
NMOS, $N_3 = N_4$	$\frac{W}{L} = \frac{6.4}{0.18}$
NMOS, $N_5 = N_6$	$\frac{W}{L} = \frac{18}{0.18}$
NMOS, $N_7 = N_8$	$\frac{W}{L} = \frac{56}{1}$
NMOS, N_9	$\frac{W}{L} = \frac{32}{1}$
Resistor, R_D	$R = 1.5K\Omega$
Beta Multiplier Reference	
PMOS, P_1	$\frac{W}{L} = \frac{33}{1}$
PMOS, P_2	$\frac{W}{L} = \frac{21}{1}$
PMOS, P_3	$\frac{W}{L} = \frac{24}{0.5}$
PMOS, N_1	$\frac{W}{L} = \frac{9}{1}$
PMOS, N_2	$\frac{W}{L} = \frac{27}{1}$
Resistor, R (replaced with NMOS)	$\frac{W}{L} = \frac{33}{1}$

Differential to Single Converter and Buffer	
PMOS, $P_1 = P_2$	$\frac{W}{L} = \frac{2}{0.18}$
PMOS, P_3	$\frac{W}{L} = \frac{3}{0.18}$
NMOS, $N_1 = N_2$	$\frac{W}{L} = \frac{3}{0.18}$
PMOS, P_4	$\frac{W}{L} = \frac{0.6}{0.18}$
PMOS, P_5	$\frac{W}{L} = \frac{1.2}{0.18}$
PMOS, P_6	$\frac{W}{L} = \frac{2}{0.18}$
PMOS, P_7	$\frac{W}{L} = \frac{6}{0.18}$
NMOS, N_4	$\frac{W}{L} = \frac{1}{0.18}$
NMOS, N_5	$\frac{W}{L} = \frac{2.4}{0.18}$
NMOS, N_6	$\frac{W}{L} = \frac{1}{0.18}$
NMOS, N_7	$\frac{W}{L} = \frac{3}{0.18}$

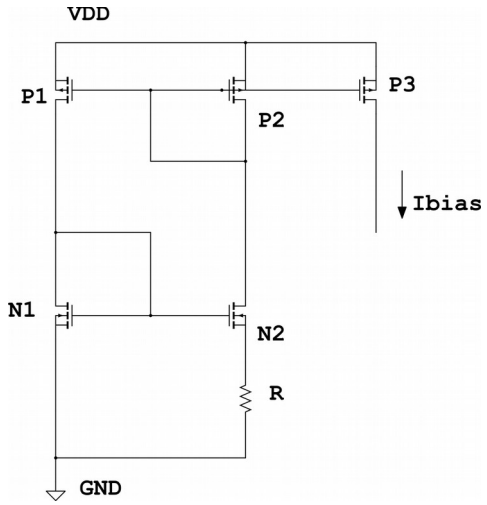

Figure 7.5: Beta Multiplier Reference for 500 uA

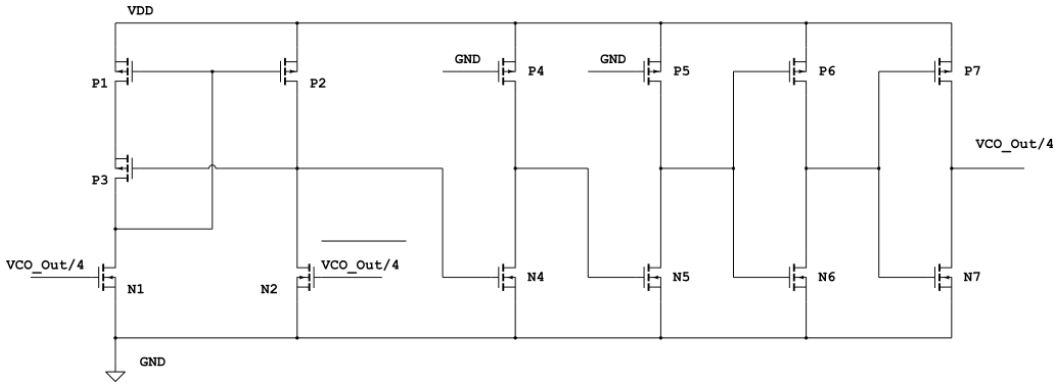

Figure 7.6: Differential to Single Converter and Buffer

Since CML gates have differential output, the signals need to be converted to single ended for further processing via the CMOS logic gates. Figure 7.6 shows a method of implementing a high speed differential to single ended converter using only 3 PMOS devices and 2 NMOS devices [47]. When the input at NMOS N_1 transitions from low to high and input at N_2 transitions from high to low, the node voltage between the gate

terminals of P_1 and P_2 fells, turning both the PMOS devices ON. At this point the output node is at V_{DD} which turns P_3 OFF. Since P_3 is OFF, the node between N_1 and P_1 is open, eliminating any possibilities of current leakage. Thus switching between V_{DD} and GND occurs at the output as the input voltages at the NMOS devices keep transitioning from high to low and from low to high. Furthermore, to smooth out the output signal into stable square waves, four back to back inverters have been added as buffer, out of which two have been implemented using pseudo-NMOS logic [31]. In figure 7.7, the whole structure of the divide-by-4 prescaler circuit has been presented.

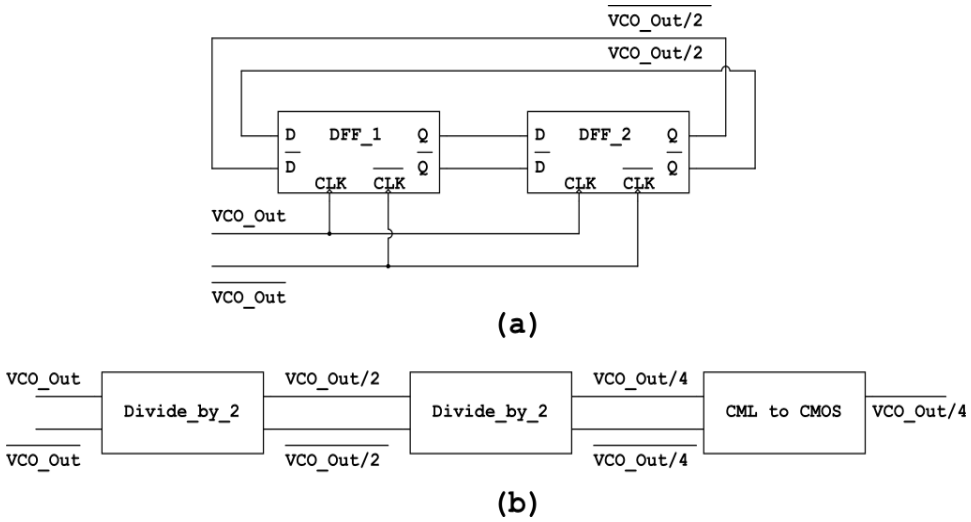

Figure 7.7: (a) Divide-By-2 (b) Divide-By-4 Prescaler

Next step in the frequency divider design process is to design the programmable divider. As mentioned previously, the feedback divider's division ratio range should be between 60.3 and 145.625. Since the prescaler took care of division ratio of 4, the rest needs taken care of by the programmable divider—division ratio of which should range between 15.075 and 36.406. In addition, the frequency resolutions for both 2.4 and 5 GHz channels also need to met. In order to meet all the requirements, a programmable divider consisting of 5 cascaded stages of 2/3 dividers has been implemented, and can be seen on figures 7.8 and 7.9.

Similar to ripple counters, this cascaded architecture in figure 7.8 has a minimum and a maximum division ratios which are achieved by controlling the program bits P_0 to $P_{(n-1)}$.

Minimum division ratio = 2^N

Maximum division ratio = $2^{N+1} - 1$

For example, for a 5 cascaded 2/3 counters, and programming bits, $P_0\ P_1\ P_2\ P_3\ P_4$ = 0 1 1 0 1, we can calculate division ratio as [5], [36]:

$$N = 2^N + P_{N-1}2^{N-1} + P_{N-2}2^{N-2} + \ldots. + P_1 2 + P_0$$

$$N = 2^5 + 1.2^4 + 0.2^3 + 1.2^2 + 1.2^1 + 0.1 = 54$$

Now, although we have established division ratio range from 16 to 63, the division ratio of 15, which is essential for 2.4 GHz band, is missing.

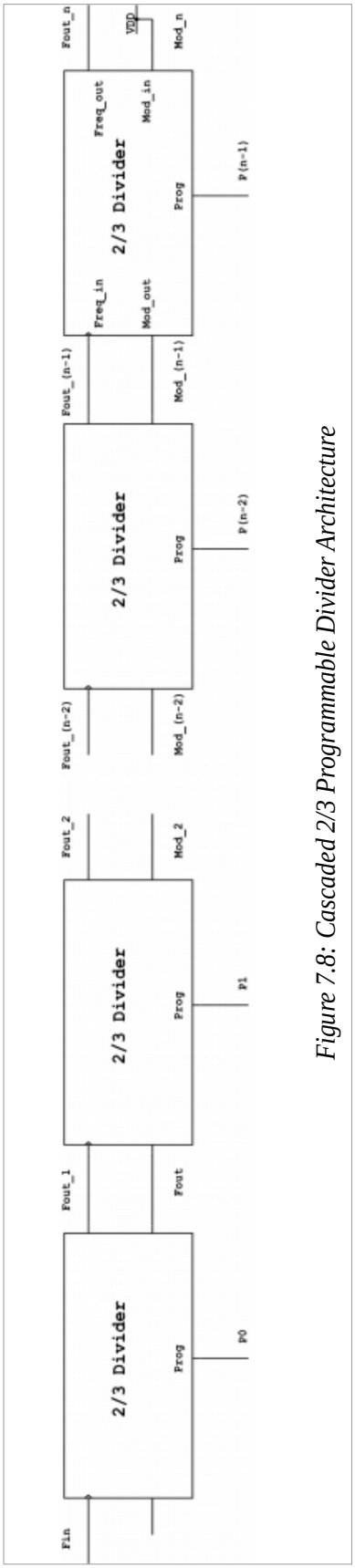

Figure 7.8: Cascaded 2/3 Programmable Divider Architecture

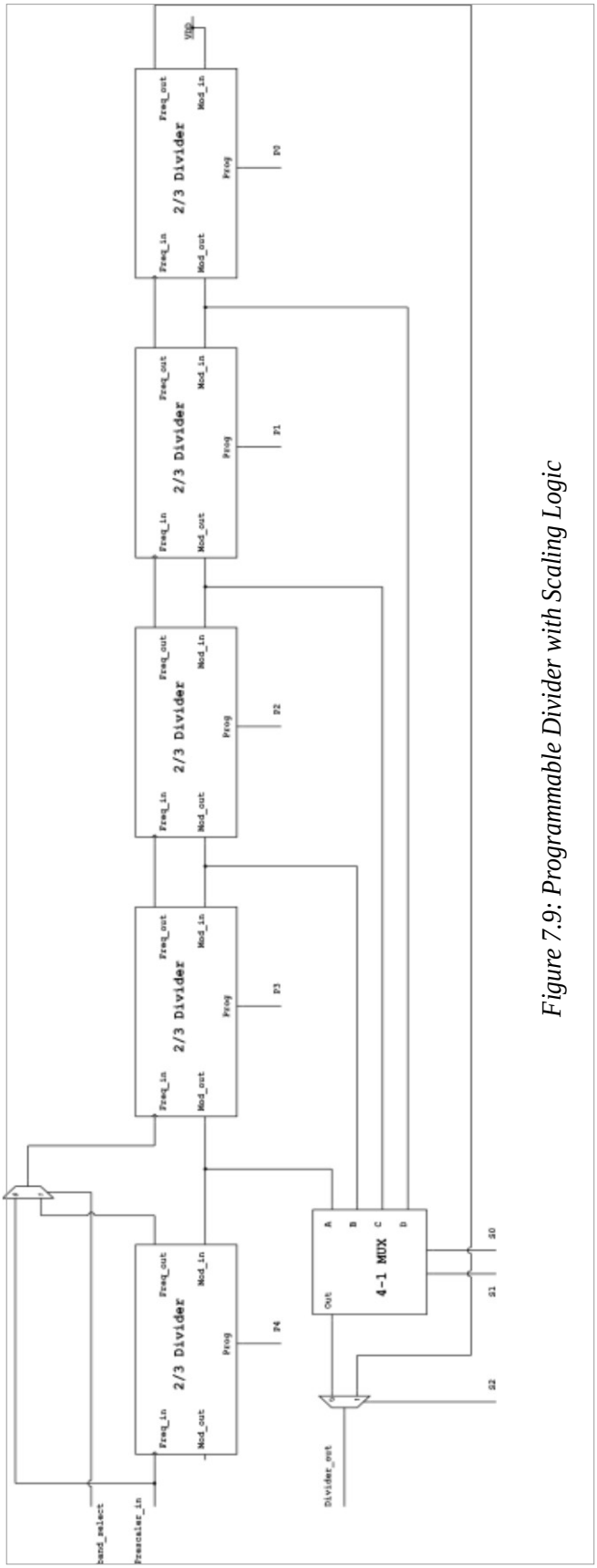

Figure 7.9: Programmable Divider with Scaling Logic

To resolve this issue, we need to apply the output from the prescaler, where there is a divide by 2 block, to down-convert the 5 GHz frequency range. For instance, to generate a 2.424 GHz output, we need to set the programmable divider to divide by 30.3, multiplying with which the prescaler will give us an output frequency of 4.848 GHz. Following that, we can take the output from the first stage of the CML prescaler and provided with the 2.424 GHz differential signals.

In addition to the arrangement for division ratio of 15, some extra control circuitry has been added to the programmable divider to reduce jitter and power consumption [5], [48]. Illustrated in figure 7.9, multiplexers have been added to divide the input frequency by either four stages of 2/3 dividers, or by 5 stages. Understanding from the previous discussions, we know that for 2.4 GHz channels, we do not need more than 4 stages, or division ratio between 15 and 31. However, in order to divide the 5 GHz channels, it is necessary that we have 5 stages of 2/3 dividers. The “band select” bit at the 2-to-1 multiplexer controls which bandwidth the synthesizer wants the divider to operate in. In the end, outputs from all stages have been connected to another set of multiplexers to accurately control which signal should travel to the phase frequency detector.

Lastly, the structure of the 2/3 prescaler has been presented in figure 7.10. Using the “Prog” bit, the divider is set to either divide by 2, or divide by 3. Moreover, since the prescalers are added in cascaded fashion,

an end-of-cycle logic has also been implemented [5], [32], [40]. Whenever, the first prescaler starts dividing by 3, the “Mod_out” bit becomes high at the end of conversion and becomes low when the cycle starts again. This provides the benefit of acquiring a signal (Mod_out), phase of which is matches the phase of the reference frequency at the PFD. Because of modulating the “Prog” bit, the output frequency of the prescaler will have a wider pulse train and will be unable to match the phase of the reference frequency. To let the previous stage know that the next stage will divide by 3, a “Mod_in” bit is sent to the previous stage where its end-of-cycle logic has already been activated. Similarly, if the “Prog” bit of the previous stage is also high, the prescaler will swallow one of its cycles providing another division ratio of 3. However, if the “Prog” bit low, the end-of-cycle logic will be re-clocked back to the next stage and the 2nd prescaler will keep dividing by 2, keeping only one stage of division by 3.

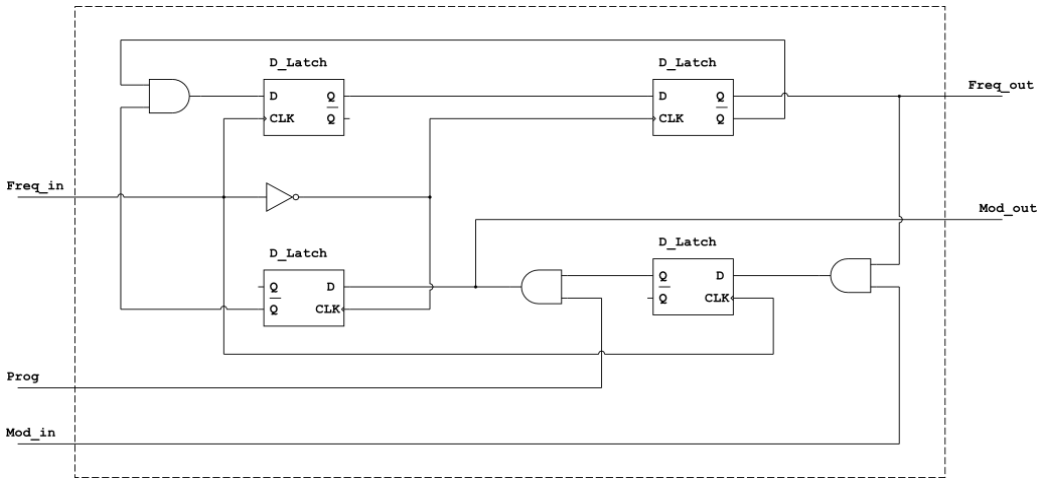

Figure 7.10: Divide by 2/3 Prescaler

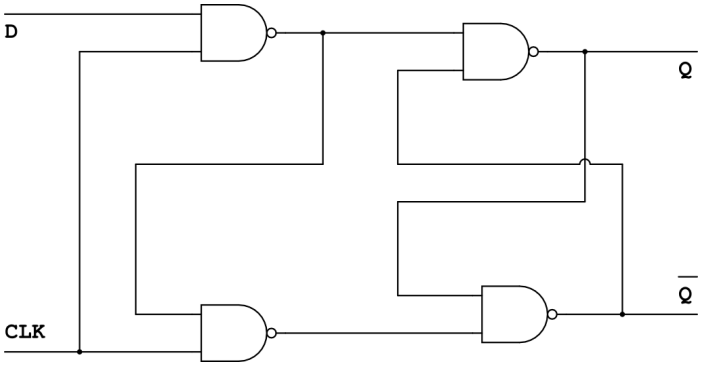

Figure 7.11: Simple D Latch for 2/3 Prescaler

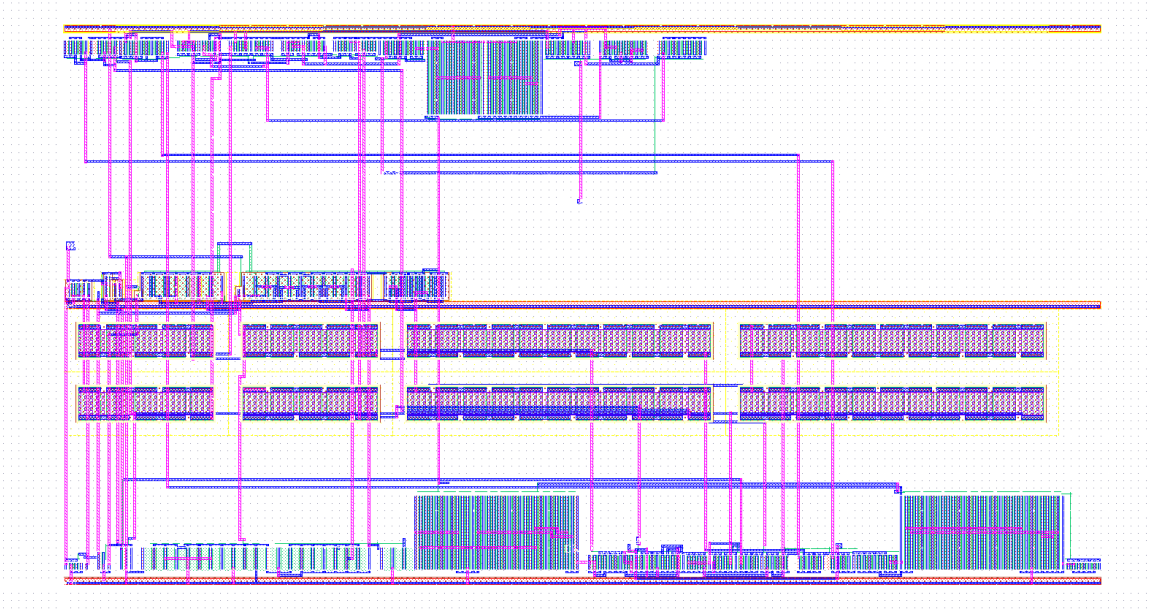

Figure 7.12: Divide by 4 Prescaler Layout

7.4 Simulation Results

Transient simulation results for CML divide by 4 prescaler has been presented in figure 7.13. Here we see that for an input frequency of 6 GHz, the prescaler generates an output frequency of 1.5 GHz. Furthermore, we also observe that for differential sine waves generated by the VCO, the output at the prescaler is a single ended square wave.

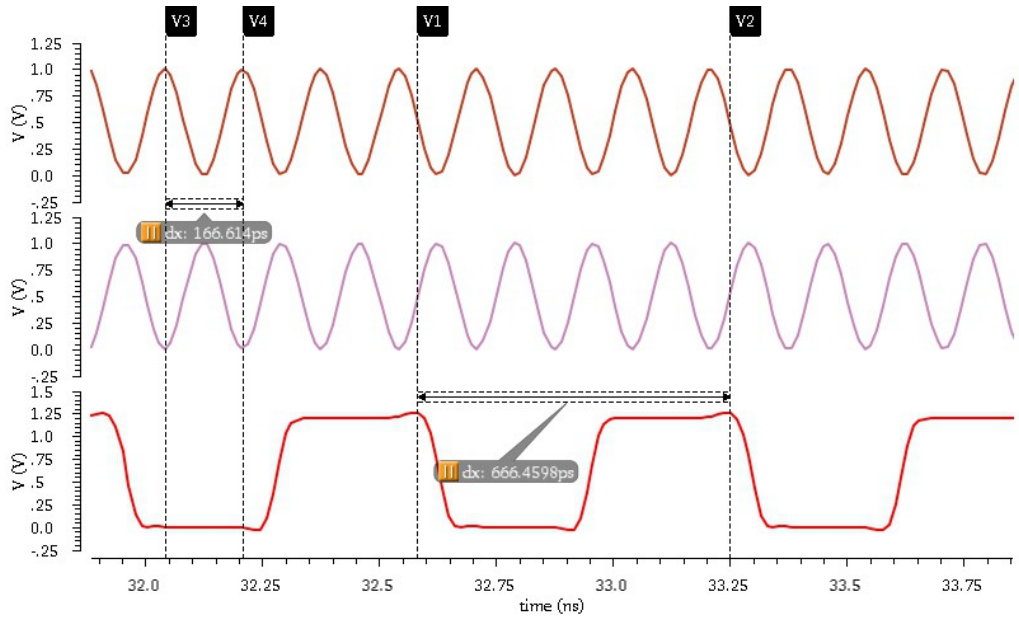

Figure 7.13: Transient Simulation for Divide by 4 Prescaler

Running a DC simulation with a supply voltage of 1.2 V, the total power consumption of the prescaler has been measured to be 2.5 mA. Although quite high, but we can justify the power consumption as the prescaler required large enough swing for input frequencies ranging up to 5.9 GHz, and mirrored the 500 µA current into 4 CML latches. Shown in figure 7.12, the layout of the prescaler takes 0.139 mm X 0.075 mm area.

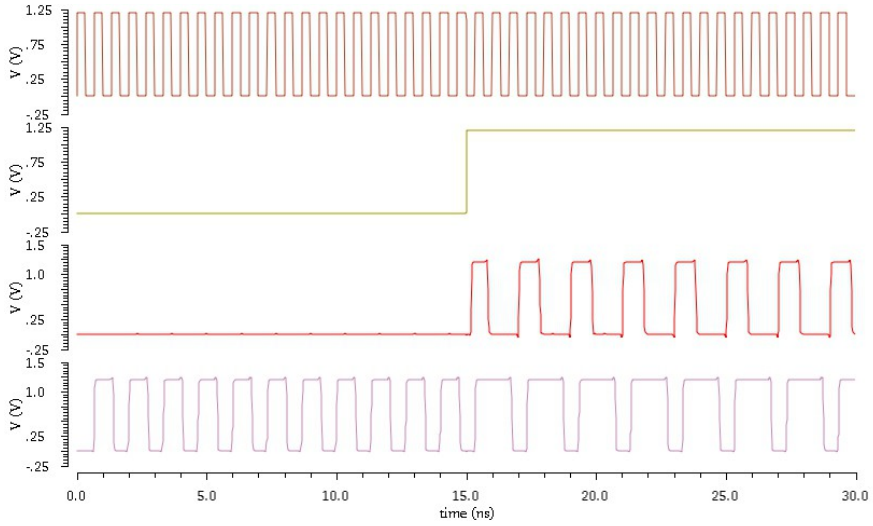

Figure 7.14: Transient Analysis for Divide by 2/3 Prescaler

Similar simulations were performed for the programmable divider. First of all, transient simulation was performed to understand the operation of the divide by 2/3 prescaler. From the results in figure 7.14, we see that when the “Prog” bit (yellow) is low, the divider divides the input signal (orange) by 2. As soon as the “Prog” bit becomes high, the input starts being divided by 3 and the end-of-cycle logic starts generating the “Mod_out” signal (Red). So, during the time “Prog” bit is high, instead of using the “Freq_out” bit as output, “Mod_out” bit is used for matching the phase of the reference frequency at the PFD.

Two different transient analyses have been performed to find the programmable dividers lowest and highest divide ratios. In addition, a divide by 39 simulation was performed to observe the different division ratios in different stages of the cascaded divider. Figure 7.15 shows the

input signal with a period of 1.5 GHz being divided by 16 to achieve an output of 93.75 MHz.

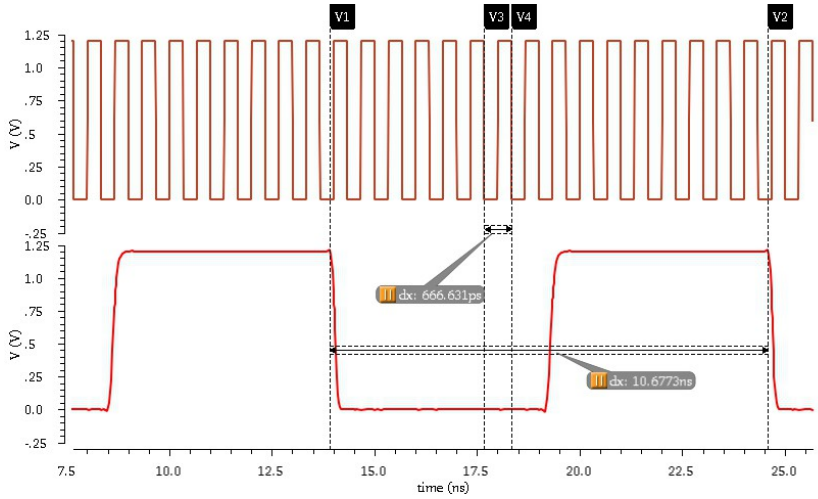

Figure 7.15: Divide by 16

On figure 7.16, we can see the division ratio of 63, which was acquired by pulling all the programming bits to high and taking the output from the most significant prescaler.

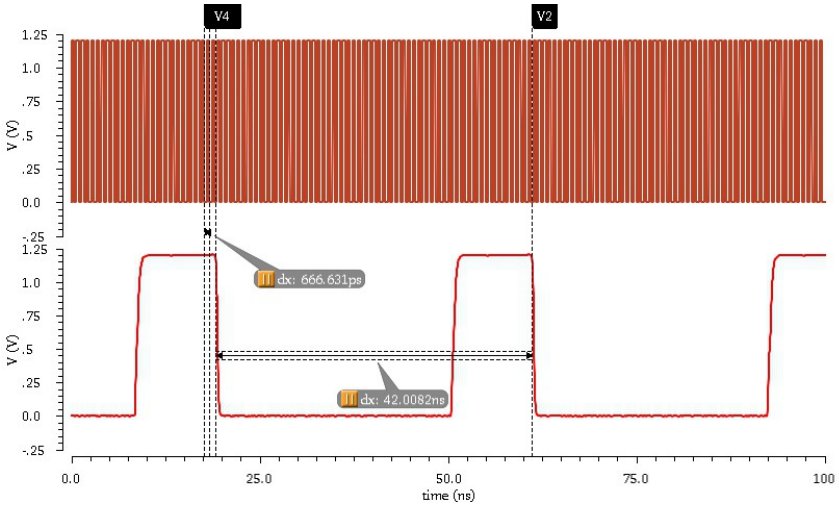

Figure 7.16: Divide by 63

Finally, on figure 7.17, we see a presentation of different division ratios at different stages of the programmable divider. For this simulation, the program code that was used: $P_4\ P_3\ P_2\ P_1\ P_0 = 0\ 0\ 0\ 1\ 1$.

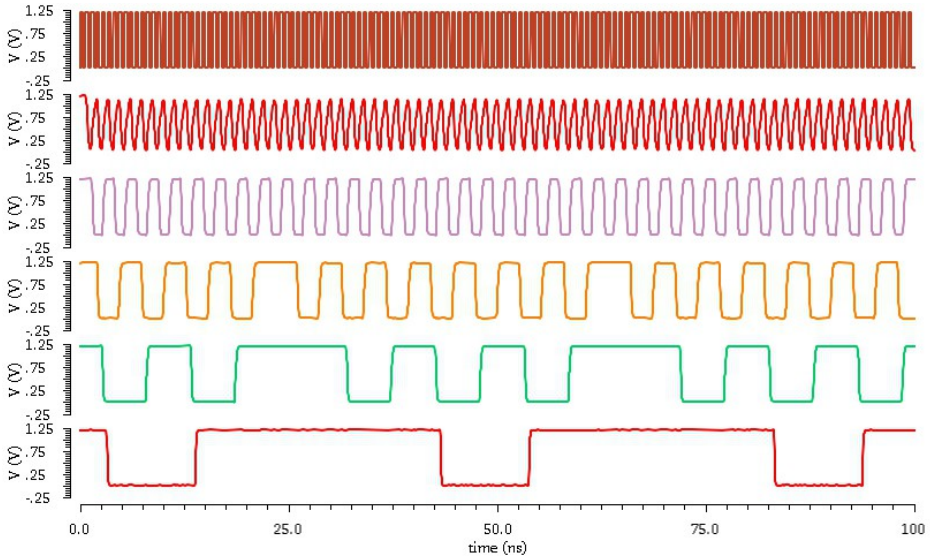

Figure 7.17: Different Division Ratios for 00111

Frequency dividers are notorious for their phase noise contribution in any PLL based system. And due to programmability, phase noise contribution may vary from little to quite large. Since we only care up to division ratio 36.406, a periodic steady state and a phase noise simulations were ran for a division ratio of 36. Results of the simulations can be seen on figure 7.18 and it can be deduced that, the programmable divider performs very well generating a phase noise of only about -158 dBc/Hz at 1 MHz offset.

DC simulation has also been performed to find the total power consumption. For generating the highest amount of divide ratio (63) for a

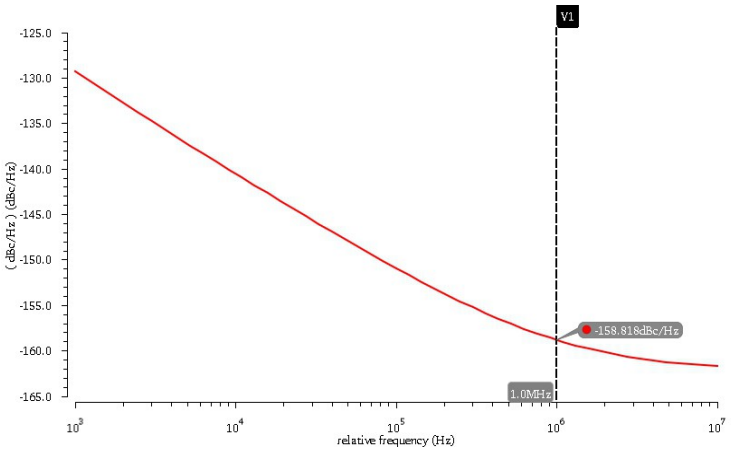

Figure 7.18: Phase Noise for Division Ratio of 36 at 1 MHz

1.5 GHz input signal, the programmable divider consumes about 490 µW, which is significantly good and allowed us to distribute more of the power budget towards designing the CML prescaler. The layout of the divider included in figure 1.19, takes up 0.130 mm X 0.102 mm area.

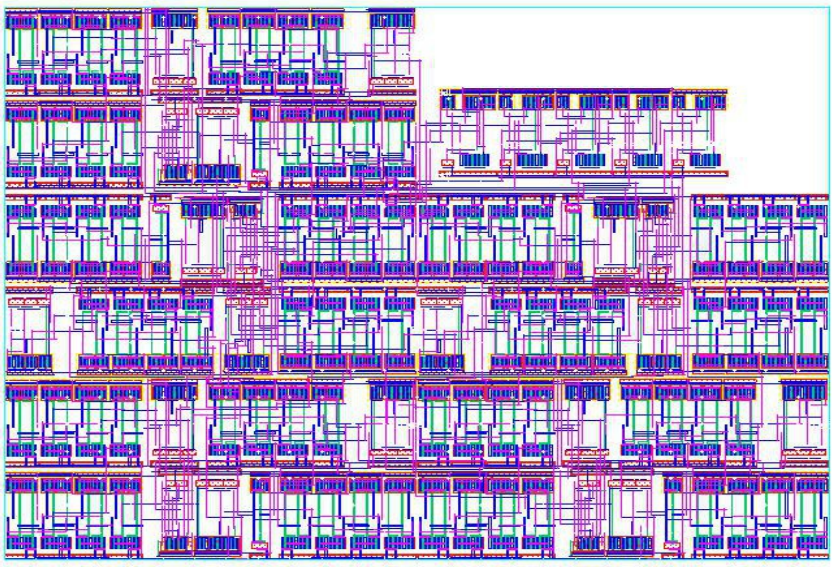

Figure 7.19: Programmable Divider Layout

7.5 Summary

In this chapter, we discussed the structure and the principles of the feedback/frequency divider. The designed divider consists of a high speed CML prescaler and a low speed programmable divider. Overall, the frequency divider can generate division ratios from 64 to 252 and can be programmed to be used with the VCO and PFD to generate output signals covering all channels in the 2.4 GHz and the 5 GHz bands.

Chapter 8

MASH 1-1-1 Delta-Sigma Modulator

8.1 Introduction

In recent years, use of accumulator based multi-stage modulators have become quite popular in the fields of fractional-N frequency synthesis. Optimized in regards to spurious tone suppression, the 3rd order Delta-Sigma modulator designed in this thesis provides the fractional-N synthesizer with a frequency resolution of approximately 2.4 Hz. Basic operational principles and hardware implementation of the MASH 1-1-1 Delta-Sigma modulator will be discussed in this chapter. Simulation results and digital implementation techniques will also be covered.

8.2 Delta-Sigma Modulator Basics

In the last two decades or so, different methods have been used to facilitate fractional-N frequency synthesis. While the integer-N side has been relying on manual techniques, a few self modulating digital techniques have also been proposed over the years that require much less effort in meeting the fractional resolution of different wireless standards. Delta-sigma modulators are one of such. Conventionally, division control of fractional-N synthesizers are controlled using single stage digital accumulators. While achieving better performance in terms of spur suppression and better stability than the integer-N synthesizers, fractional-N synthesizers with single accumulators still produce noticeable amount of spurs at the interval when the overflow bit from the accumulator modulates the dual-modulus divider [32], [40]. To overcome this issue, multi-stage noise shaping (MASH) property of delta-sigma modulators (DSM) and its decimation filters have gained popularity and are being employed today in fractional-N frequency synthesis. Delta-Sigma modulators are known for their oversampling characteristics; and with the inclusion of noise shaping, an unconditionally stable output may be acquired through multi-stage implementation [24], [52]. In figure 8.2 and 8.3, a simple single stage 1st order delta-sigma modulator/accumulator is presented—however, for the purpose of this thesis, a 3rd order delta-sigma modulator (MASH 1-1-1) is utilized that relies on its randomized overflow bits from three cascaded accumulators.

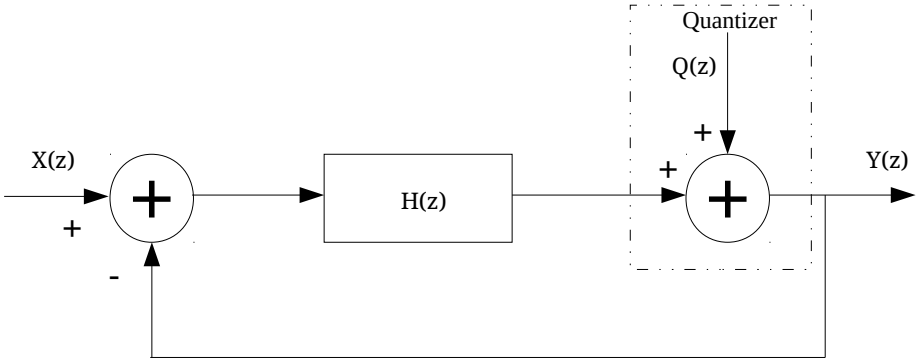

Figure 8.1: Linear Model of General Delta-Sigma Modulator

With the explanation provided by Lee et al., a general delta-sigma modulator can be treated as a linear model depicted in figure 8.1. It can be observed that, with an input signal X(z) and a quantization noise Q(z), the signal transfer function is expressed as:

$$S_{TF}(z) = \frac{H(z)}{1 + H(z)} \quad , \qquad \text{when} \quad Q(z) = 0 \tag{8.1}$$

In addition, the noise transfer function is given as:

$$N_{TF}(z) = \frac{1}{1 + H(z)} \quad , \qquad \text{when} \quad X(z) = 0 \tag{8.2}$$

Furthermore, the output signal Y(z) can be expressed in the frequency domain as:

$$\begin{aligned} Y(z) &= S_{TF}(z).X(z) + N_{TF}(z).Q(z) \\ &= \frac{H(z)}{1 + H(z)}.X(z) + \frac{1}{1 + H(z)}.Q(z) \end{aligned} \tag{8.3}$$

From the equations above, we analyze that the noise transfer function is proportional to the poles of H(z) so that, when H(z) becomes too large, the noise diminishes to zero. We also see that the output signal is dependent on proper choice of H(z) and quantization noise. Usually, a unity gain H(z) is chosen such that, the signal transfer function is unity across the band of interest and noise transfer function generates a high pass response. In such conditions, in-band quantization noise will be reduced keeping the output unaltered.

The hardware implementation of the first order delta-sigma modulator/accumulator proposed by Bourdi et al., has been presented and discussed in the following two figures [32]. In time domain, we find:

$$\begin{aligned} u(n) &= X(n) - C(n-1) \\ v(n) &= u(n) + v(n-1) \end{aligned} \tag{8.4}$$

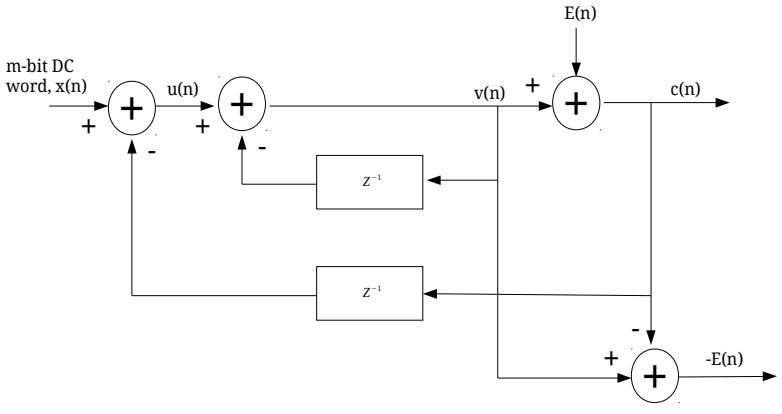

Figure 8.2: Linear Model of the First Order Delta-Sigma Modulator

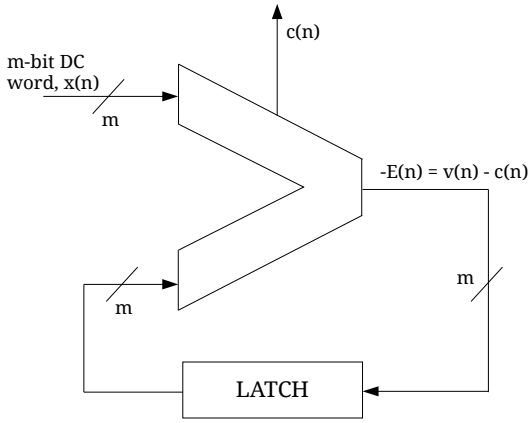

Figure 8.3: Hardware Implementation of First Order Delta-Sigma Modulator/Accumulator

Using quantization noise, E(n) = C(n) – v(n) we write:

$$v(n) = X(n) - C(n-1) + v(n-1) \tag{8.5}$$

So, from equations (8.4) and (8.5), we finally derive:

$$v(n) = X(n) - E(n-1) \tag{8.6}$$

Now as the cycle of accumulation and quantization progresses, the contents of the overflow and error signal follow each other. In this case, when quantization Q(n) becomes 1, overflow C(n) becomes 1, and error signal -E(n) becomes 1.

As previously mentioned, for this thesis a 3rd order cascaded modulator has been implemented. On figure 8.4, we see the linear model of the hardware implementation of the modulator. Higher-order modulators such as the MASH modulators, employ feedback to improve noise shaping

at low frequencies while maintaining a reasonable stability. Since the 3rd order modulator is constructed using 1st order modulators, quantization noise from each modulator contributes to the progressive noise shaping, filtering out high frequency noises.

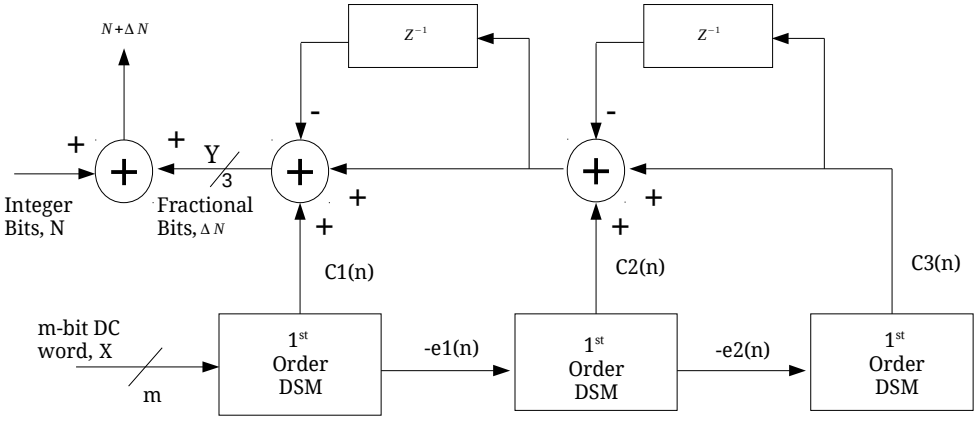

Figure 8.4: Linear Model of 3rd Order Delta-Sigma Modulator

Using the analysis of the 1st order modulator above, and assuming each 1st order modulator in figure 8.4 has one-bit quantizer Q(i), we can derive the expression for the output of the 3rd order modulator as:

$$Y(z) = X(z) + Q_1(1 - z^{-1}) + [-Q_1 + Q_2(z)].(1 - z^{-1})(1 - z^{-1}) + [-Q_2 + Q_3(z)].(1 - z^{-1})(1 - z^{-1})^2 \quad (8.7)$$

$$= X(z) + (1 - z^{-1})^3.Q_3(z) \quad (8.8)$$

$$= X(z) + H(z).Q_3(z) \quad (8.9)$$

Here, H(z) refers to the high frequency component of the cumulative noise transfer function [53], [54]. We can see from figure 8.4, that the output from the MASH modulator is the fractional part of the divisor that will control/modulate the programmable divider. With the addition of the integer part, the output frequency of the frequency synthesizer can be express as:

$$F_{VCO} = (N + \Delta N).F_{REF} \tag{8.9}$$

Where, F_{VCO} is the output frequency of the VCO,

F_{REF} is the frequency of the reference,

N is the integer part of the division ratio, and

ΔN is the fractional part of the division ratio.

The fractional part of the division ratio depends on the number of input bits, or DC input word of the delta-sigma modulator. The fractional resolution becomes smaller and more precise as the number of inputs become higher. For this thesis a 24-bit DC word delta-sigma modulator has been implemented for which we can estimate the frequency resolution to be about 2.4 Hz. The fractional part can be further expressed as:

$$F_{VCO} = (N + \frac{K}{F}).F_{REF} \tag{8.10}$$

Where K is the decimal representation of the input DC word and F is the decimal representation of the resolution of the input bits. For example, using K = 00000000100000100010100 10 and N = 133, for a reference frequency of 40 MHz, we can calculate the VCO frequency as:

$$F_{VCO} = \left(133 + \frac{33874}{1677216}\right) 40 \text{ x } 10^{6}$$

$$F_{VCO} = 5.321\, MHz$$

Since MASH 1-1-1 topology produces 3 overflow bits from the noise shaping accumulators, they need to be properly mapped to represent the correct output sequence [32], [55]. For a 3rd order MASH 1-1-1 topology, the output levels vary from -3 to +4. In order to be added to the integer division ratio, these 8 levels of output needs to be mapped into 3 bits. A typical mapping implements 2's complement algorithm—representation of which is provided in table 8.1.

Table 8.1: Mapping Delta-Sigma Modulator Output Bits to Decimal Levels

Output Levels	Y_2	Y_1	Y_0
-3	1	0	1
-2	1	1	0
-1	1	1	1
0	0	0	0

1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0

Another very commonly attributed characteristic of the delta-sigma modulator is the power spectral density of the instantaneous phase error seen at the output of the modulator and the input of the phase frequency detector [24], [32], [40]. Running fast Fourier transform over a highly sampled period of transient response, and normalizing by a factor of $N/2\pi$, the power spectral density of the delta-sigma modulator follows a rise up of 60 dB/decade. Simulated results of the power spectral density along with the other analyses will be presented later in the chapter.

8.3 Circuit Design

The MASH modulator designed for the synthesizer is a purely digital circuit. Using methods proposed by Bourdi et al., and Kim et al., implementation of the cascaded 3rd order modulator has been one of the daunting parts of this thesis [32], [55]. Not having access to foundry provided standard cells, ASIC implementation could not be carried out. Instead, the whole system was constructed in device level which took a considerable amount of time for debugging and reconstruction. To start off, figure 8.5 shows the system level diagram of the MASH where three first order 24-bit digital accumulators have been cascaded (hence the name, MASH 1-1-1). In addition, the system also includes 24-bit clocked latches,

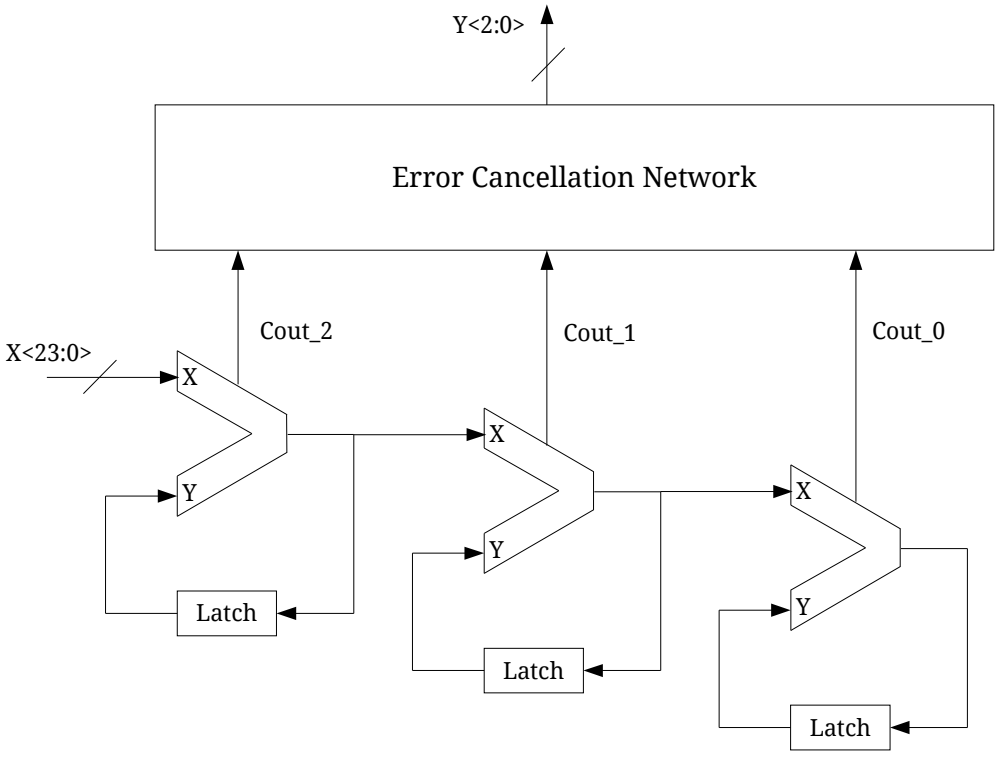

Figure 8.5: MASH 1-1-1 Delta-Sigma Modulator

and an error cancellation network, where the 3-bit overflow from accumulators is converted into 8 output (- 3 ~ + 4) levels. The node between two accumulators, where the latch picks up the data, represents quantization error—an essential parameter in order to generate randomized overflow bits.

Due to high speed requirement of the synthesizer, the 24-bit accumulators/pipeline adders have been designed using 8-bit carry look-ahead adders (CLA) [31]. Figure 8.6 and 8.7 show the schematic of the 1st and subsequent stages of the accumulators. On figure 8.6, we see that the data is carried over to the last stage in 3 clock cycles. This is done to synchronize the output of the CLAs so that, all of the 24-bit output arrives

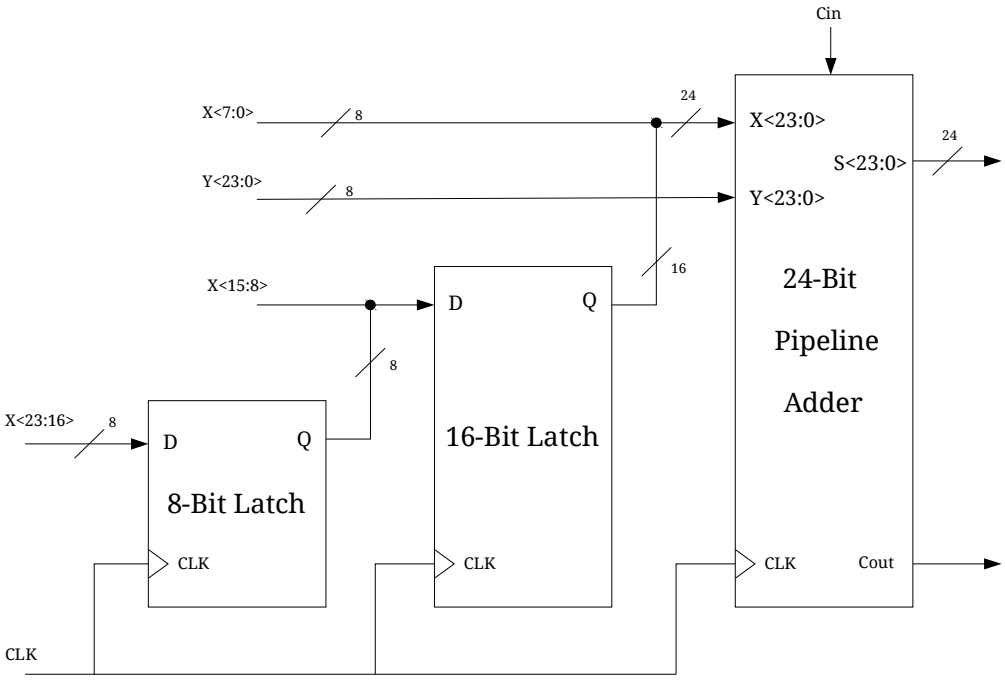

Figure 8.6: 24-Bit Accumulator (First Stage)

at the 2nd accumulator at the same time. Since 2nd and 3rd stages take input from 1st and 2nd stages respectively, and within the previous three clock cycles quantization errors have already started to accumulate, the 2nd and 3rd accumulators do not need to have similar synchronization.

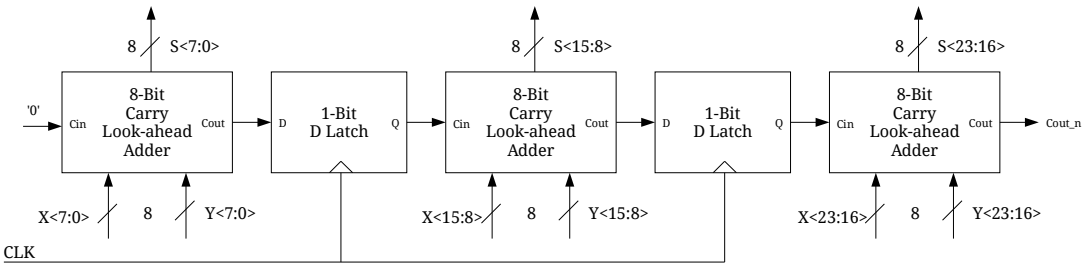

Figure 8.7: 24-Bit Pipeline Adder (Accumulator for 2nd and 3rd Stage)

We also observe in the figures that the carry from each of the 8-bit CLAs are being propagated to the next stages' “carry ins” through a D-latch, which ensures correct results at the output when the proper input is available from the previous stages. The “carry out” bit from each of the 24-bit accumulators are the overflows that will be converted for our division control.

Comprehensive analysis of the CLA is beyond the scope of this thesis; however, a basic operational principle is being provided for the readers' understanding. For the sake of simplicity, a system level diagram of a 4-bit CLA (design of 8-bit CLAs has been carried out in the actual circuit) is provided in figure 8.8, while the schematic is provided in figure

8.9. For a 4-bit carry ripple adder, the most critical path would be the path from the “carry in” of the 1st bit adder to the “carry out” of the 4th bit adder. Since one stage's “carry in” is dependent on the “carry out” of the previous stage, speed of operation is heavily compromised. For the carry look-ahead adder, we see that each 1-bit adder is made to produce a propagate P and a generate G signals that are calculated simultaneously with the summation bits [31]. Because of this technique, although the CLA starts at the same speed as a carry ripple adder, generation of the “carry outs” becomes about 4 times faster as the operation continue—resulting in a faster mode of operation. One disadvantage of CLAs is that, although the circuit operates at a high speed, the complexity of designing the circuit for higher number of bits is large. In order to ease the design process, CLAs are typically built in 4-bit groups. The group as a whole output a group propagate and a group generate signals, which are carried over to the subsequent stages to construct a higher bit CLA.

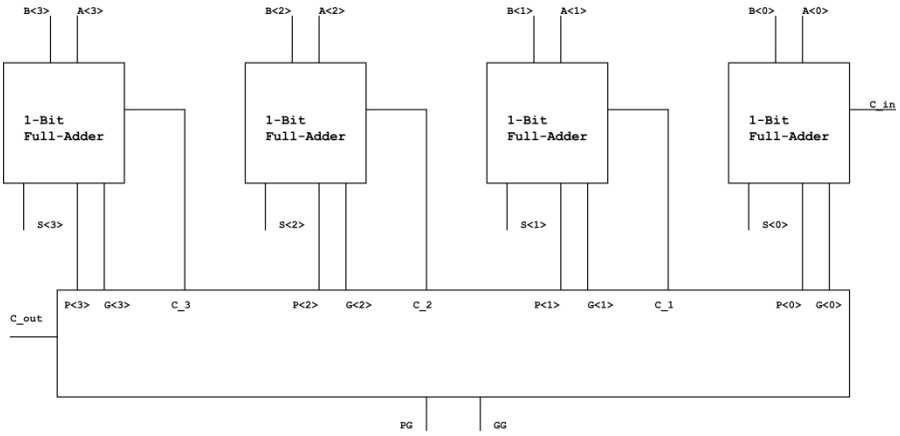

Figure 8.8: 4-Bit CLA Block Diagram

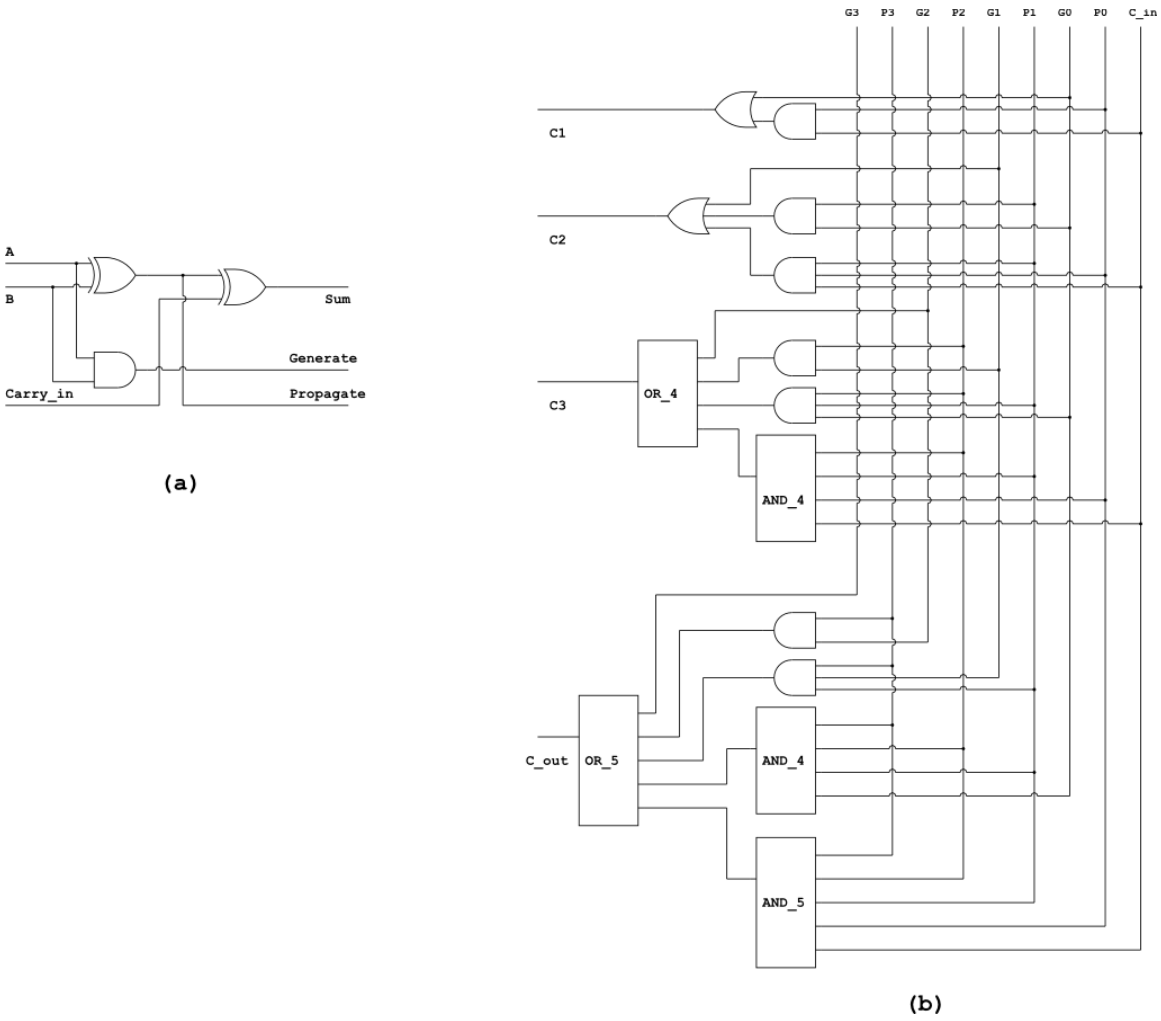

Figure 8.9: (a) Full-Adder for CLA (b) Carry Look-Ahead Logic

The error cancellation network generates 8-levels of output from the 3 overflow bits of the accumulators and uses an algorithm based on 2's complement [32], [40], [55]. Presented in figure 8.10, it can be seen that the network uses two different mapping logic blocks which together generate the 2's complement mapping of the 3-cycle latched overflows. Again, the clocked latches are to synchronize the correct sequence of bits at

the mapping logic due to the first stage of the accumulation being three cycle slower. Each of the latch also represent 1st to 3rd order MASH modulator outputs. Further details regarding the mapping logic and implementation of the 2's complement algorithm may be acquired from [32] and [55]. The “mapping logic 1” and the “mapping logic 2” blocks implement the following functions:

Mapping Logic 1:

$$M_1=[B.[(\overline{A}.\overline{C})+(A.C)]+(\overline{B}.A.\overline{C})+(\overline{A}.C)] \quad (8.11)$$

$$M_2=[B.(A+C)+(A.C)] \quad (8.12)$$

Mapping Logic 2:

$$M_1=A+(\overline{B}.C) \quad (8.13)$$

$$M_2=(A.\overline{B}.\overline{C})+(\overline{A+B}.C)+(A.B) \quad (8.14)$$

$$M_3=(\overline{B}.C)+(B.\overline{C}) \quad (8.15)$$

Using the equations above and due to the connectional mechanism of the 2nd and 3rd accumulators, error correction is performed with weighted differentiators [32]. Figure 8.10 (c) and 8.10 (d) illustrate the implementation of this correctional network.

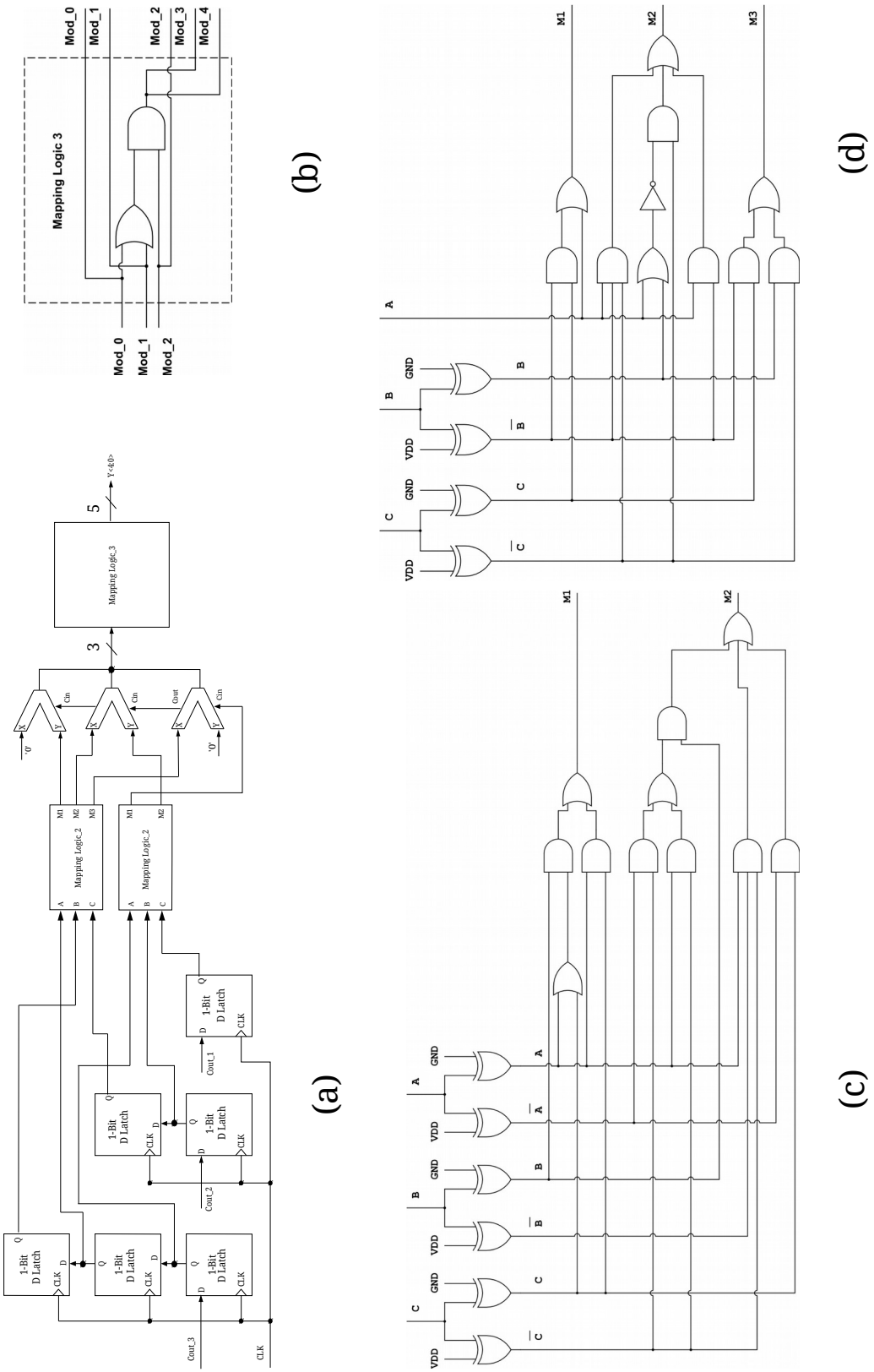

Figure 8.10: (a) Error Cancellation Network (b) Mapping Logic 3 (c) Mapping Logic 1 (d) Mapping Logic 2

8.4 Simulation Results

Transient simulations have been performed for the MASH modulator testbench. Since MASH produces a randomized output based on its noise shaping functionality, the output signals have no particular sequence. However, using a 3-bit ideal digital-to-analog converter in its testbench, the modulator can be observed to produce 8 distinct levels (- 3 ~ + 4) of output.

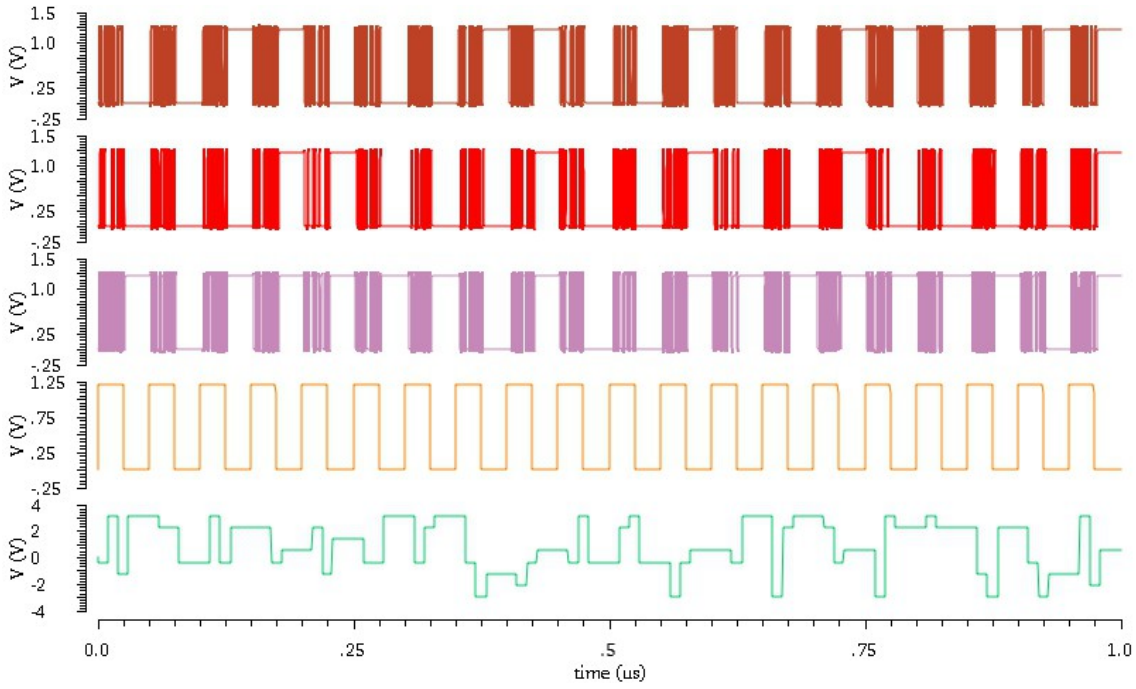

Figure 8.11: Error Corrected Accumulator Overflows Representing 8 Levels of Output

Figure 8.12 illustrates the function of the pipeline CLA with a simple transient simulation. The latching and aligning of the carry bit along with the sum bit from one CLA to the next CLA can be observed in the figure.

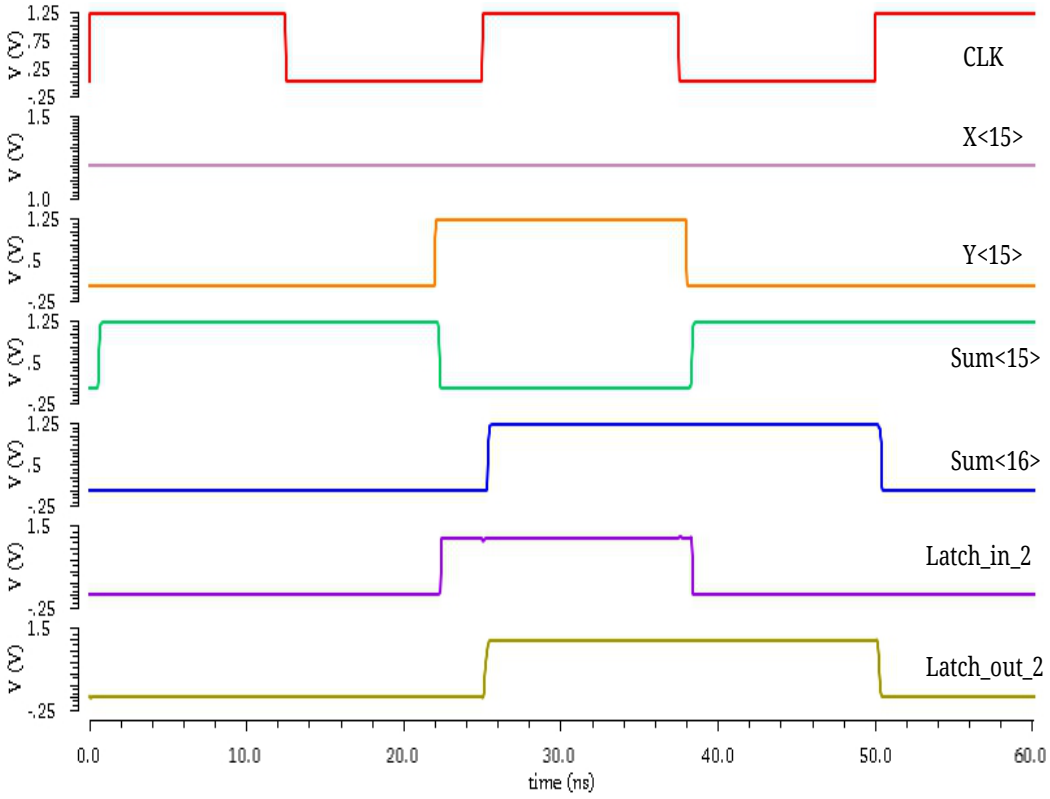

Figure 8.12: Transient Analysis of Accumulator/Carry Look-Ahead Adder

In addition, simulation for the power spectral density of the output noise has also been performed using MATLAB (figure 8.15) and the output at -40 dB appears to be meeting the expected results.

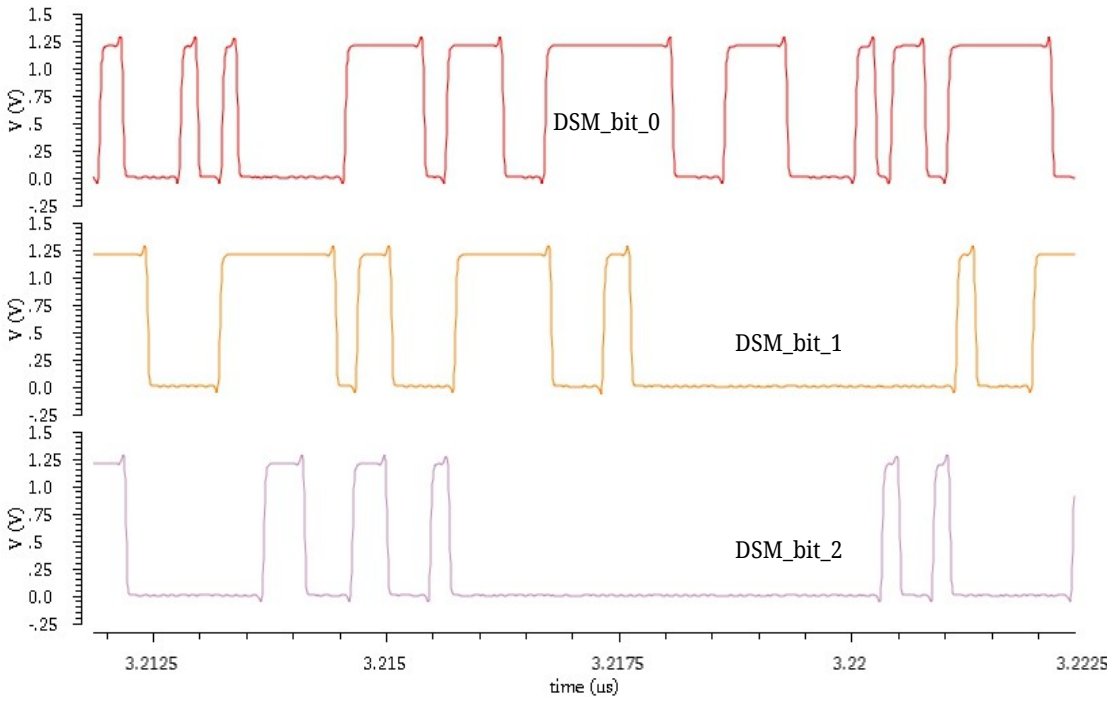

Figure 8.13: Randomized Noise Shaping of MASH 1-1-1

After running a mid-tolerant DC analysis, the average power consumption of the MASH came out to be 15.3 mW, which is quite high compared to the rest of the synthesizer blocks. But given the high number of gates and high frequency performance, it appeared justifiable. Figure 8.14 shows the layout of the Delta-Sigma Modulator covering an area of 0.374 mm X 0.593 mm.

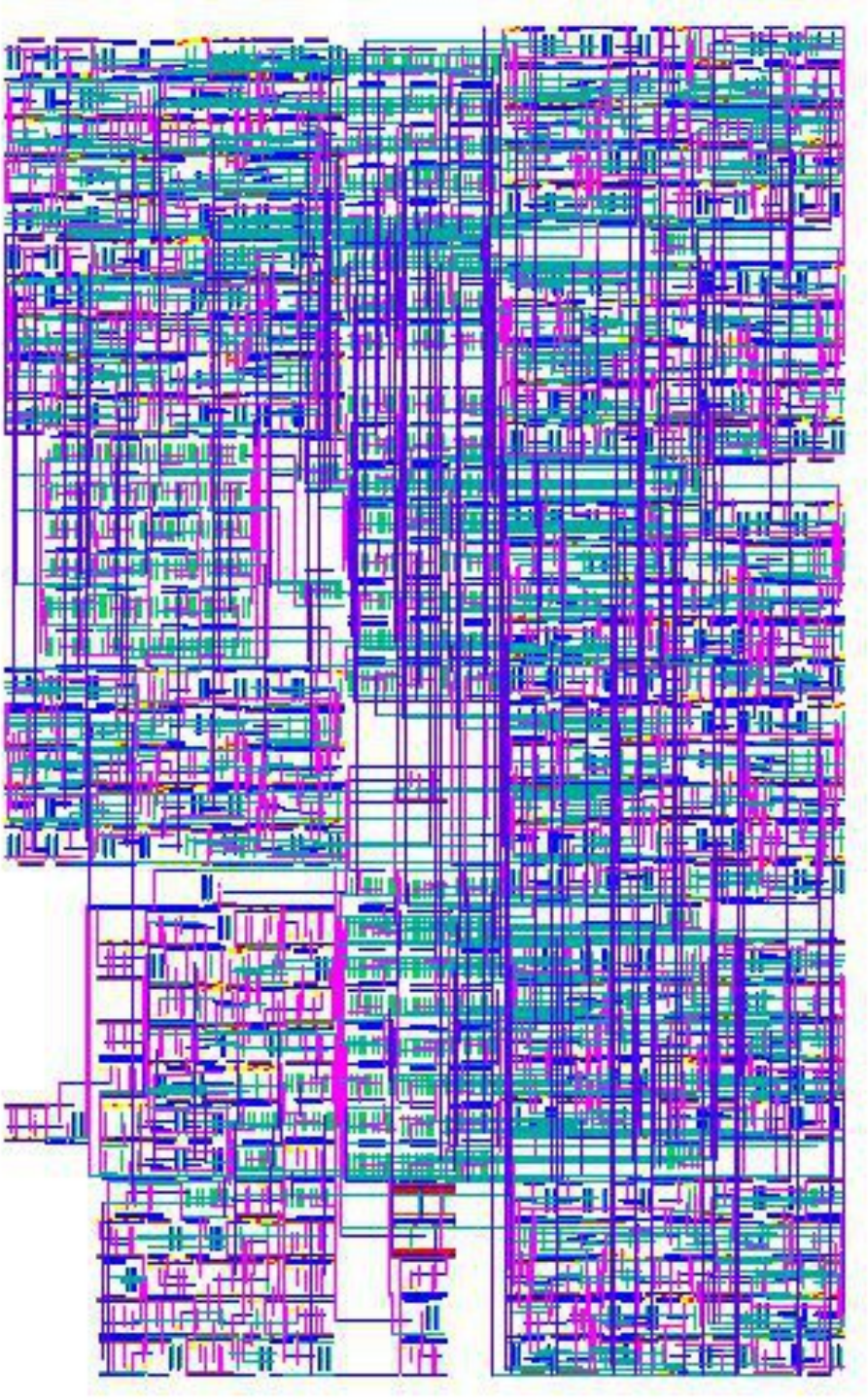

Figure 8.14: Layout of MASH 1-1-1 Modulator

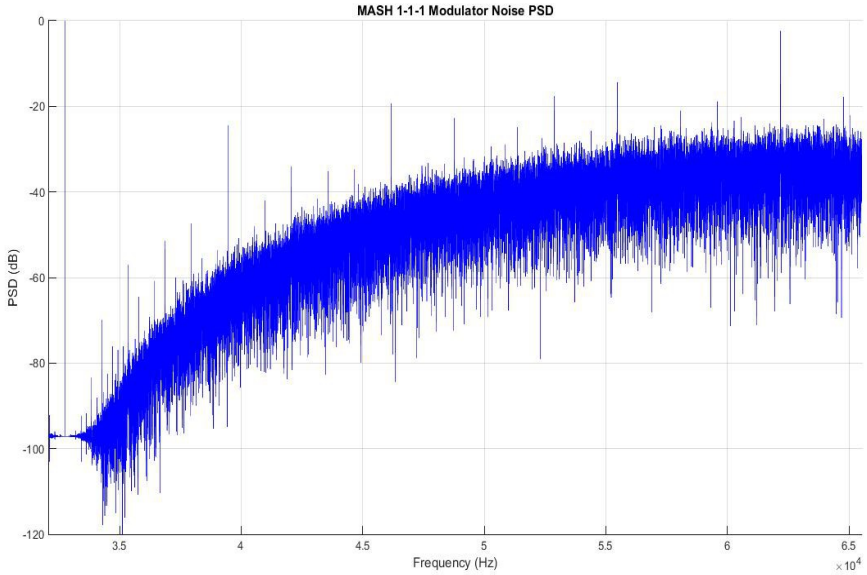

Figure 8.15: Output Noise Power Spectral Density of MASH 1-1-1 (Behavioral)

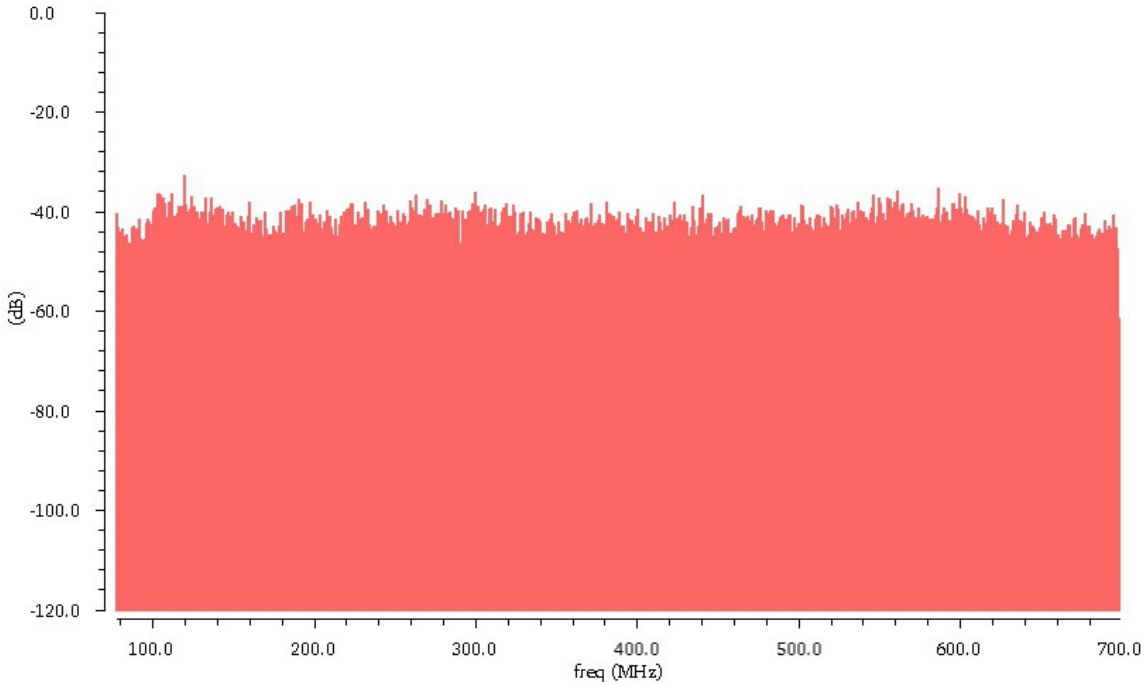

Figure 8.16: Output Noise Power Spectral Density of MASH 1-1-1 (SPICE)

8.5 Summary

In this chapter we discussed the basics and hardware implementation of the MASH 1-1-1 Delta-Sigma modulator. Constructed using 3 stages of 24-bit accumulators and a robust mapping logic, the MASH produces 8 output levels with which the synthesizer can achieve a frequency resolution of approximately 2.4 Hz. In addition, the power spectral density of its phase error is measured to be approximately -40 dB at the loop bandwidth.. The average power consumption for the modulator is 15.3 mW while it covers an area of 0.374 mm X 0.593 mm.

Chapter 9

Simulation of Complete Synthesizer

SPICE level simulation of the complete fractional-N frequency synthesizer is quite a challenging process. The reason why many researchers completely depend on behavioral simulations is that, SPICE level simulations require tremendous amount of time and effort to achieve desired results while being heavily resource hungry. Although some system level analyses were provided in last few chapters, a successful SPICE simulation was one of the primary goals of this thesis. In many instances, due to lack of sufficient computational memory, some analyses needed to

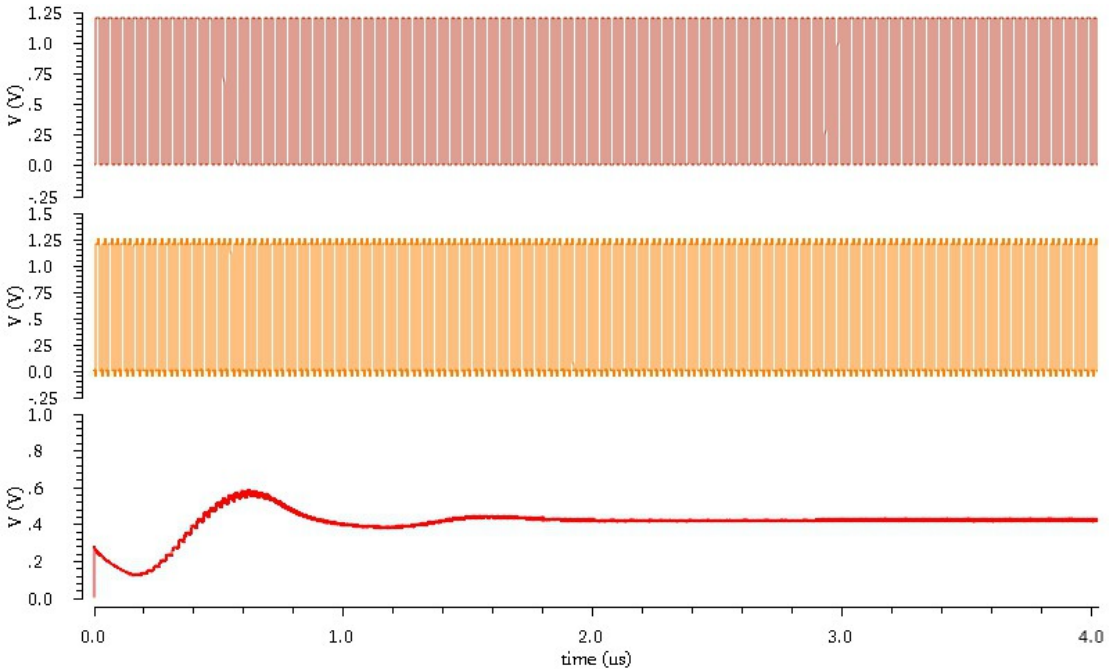

Figure 9.1: Transient Simulation for Division Ratio of 124

be performed with low to medium tolerance. In other instances, quicker algorithms of the simulator had to be used in order to avoid system instability and loss of results. In the cases of integer division, to perform a single transient simulation of 5 μs it took about 9.5 hours. When it came to fractional division, single transient analysis of 3 μs took little over 12 hours. Fortunately, as we can see from figures 9.1 and 9.4, the frequency synthesizer achieved a moderately stable locked state within approximately 2 μs.

On figures 9.1 and 9.2, we see the VCO frequency is being divided by 124, phase of which is then being compared to the phase of the 40 MHz reference frequency. Before the phases are equal, we see the control voltage of the VCO is either rising or falling. As the phases become equal, that is moment when the synthesizer achieves the locked state and the control voltage stops modulating. Spectrs of both the VCO's output and its divided by 2 frequency (frequency of interest in 2.4 GHz band) on figure 9.3 show that, they are experiencing a few spurs; which however, have been attenuated enough to minimize disruption of the lock state.

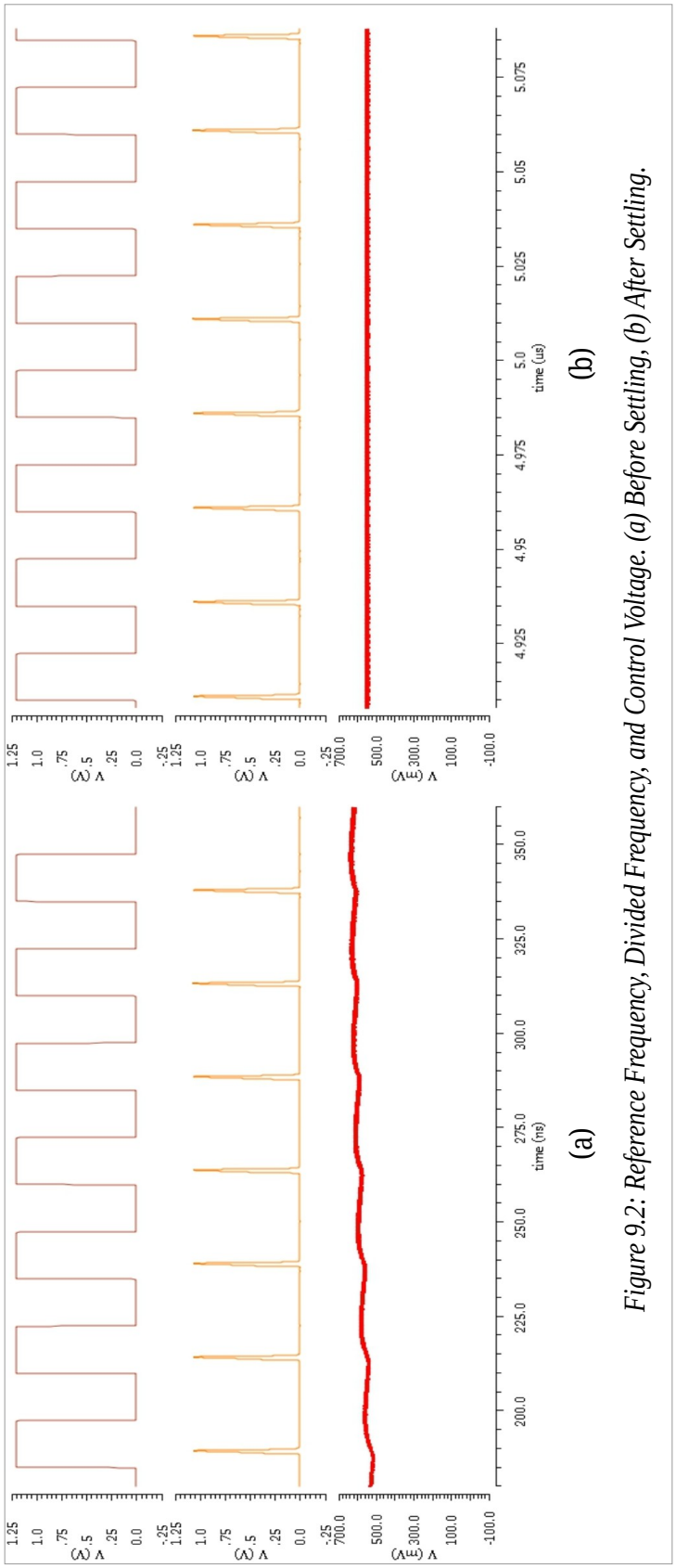

Figure 9.2: Reference Frequency, Divided Frequency, and Control Voltage. (a) Before Settling, (b) After Settling.

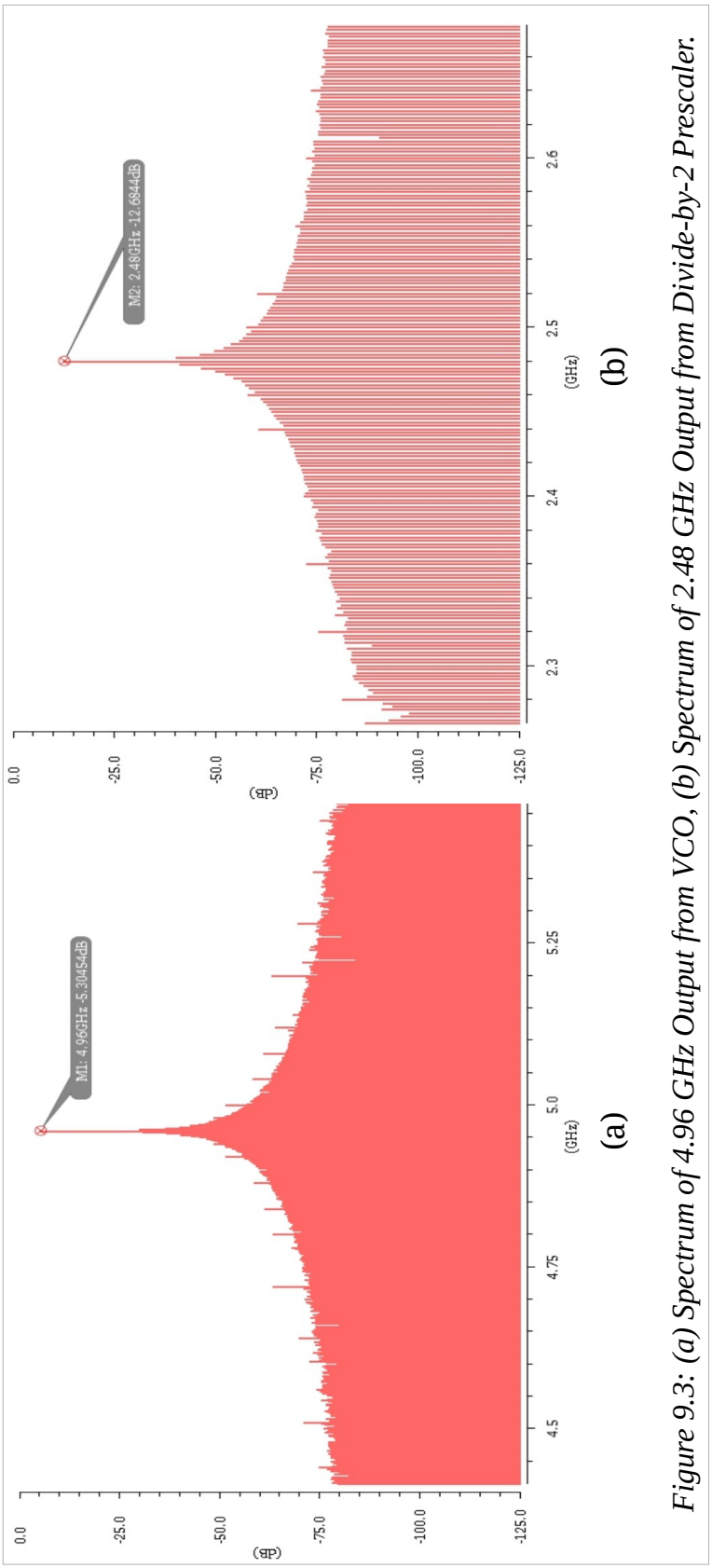

Figure 9.3: (a) Spectrum of 4.96 GHz Output from VCO, (b) Spectrum of 2.48 GHz Output from Divide-by-2 Prescaler.

Figure 9.4 and 9.5 show the transient simulation results for the fractional division ratio. With the integer bits set to 00001 and the DC input to the MASH modulator set to 10000000000000000000000, the 5.5 GHz output from the 40 GHz reference frequency was achieved. Figure 9.4 shows that the loop achieves lock almost at the same interval as the integer division, and figure 9.5 shows that the difference of power between the fundamental frequency and its 2nd harmonic is about 55 dB. We also observe that during fractional division, spurs have been suppressed greatly. However, the total noise floor due to the addition of the modulator has increased compared to integer division.

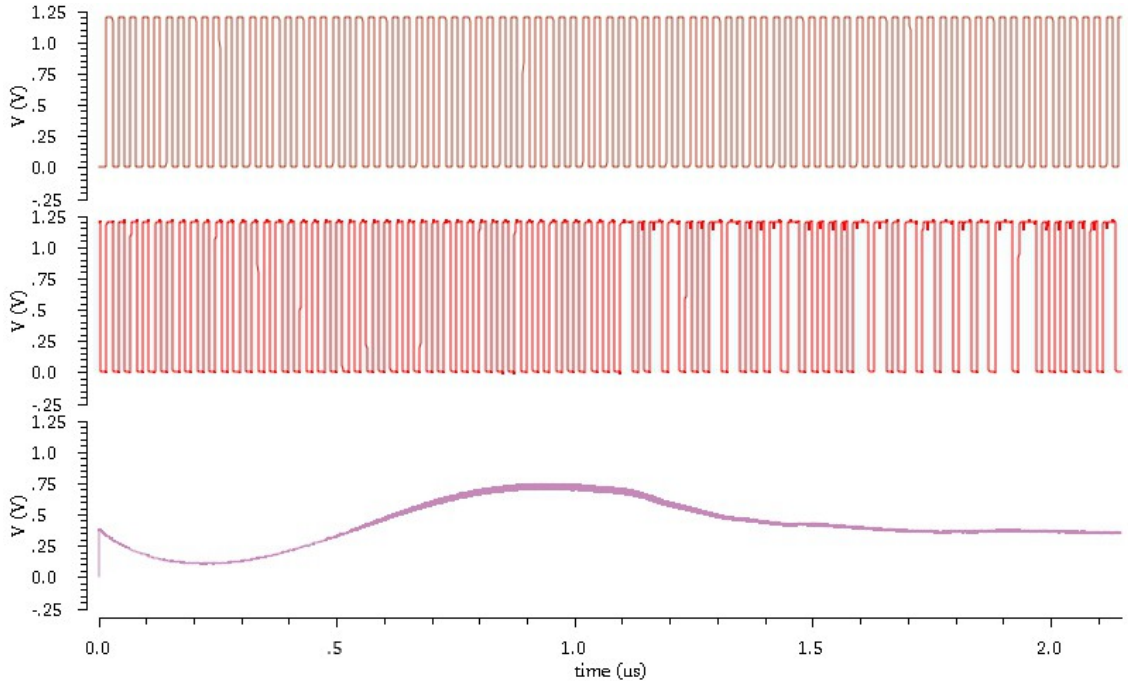

Figure 9.4: Transient Simulation for Division Ratio of 137.5

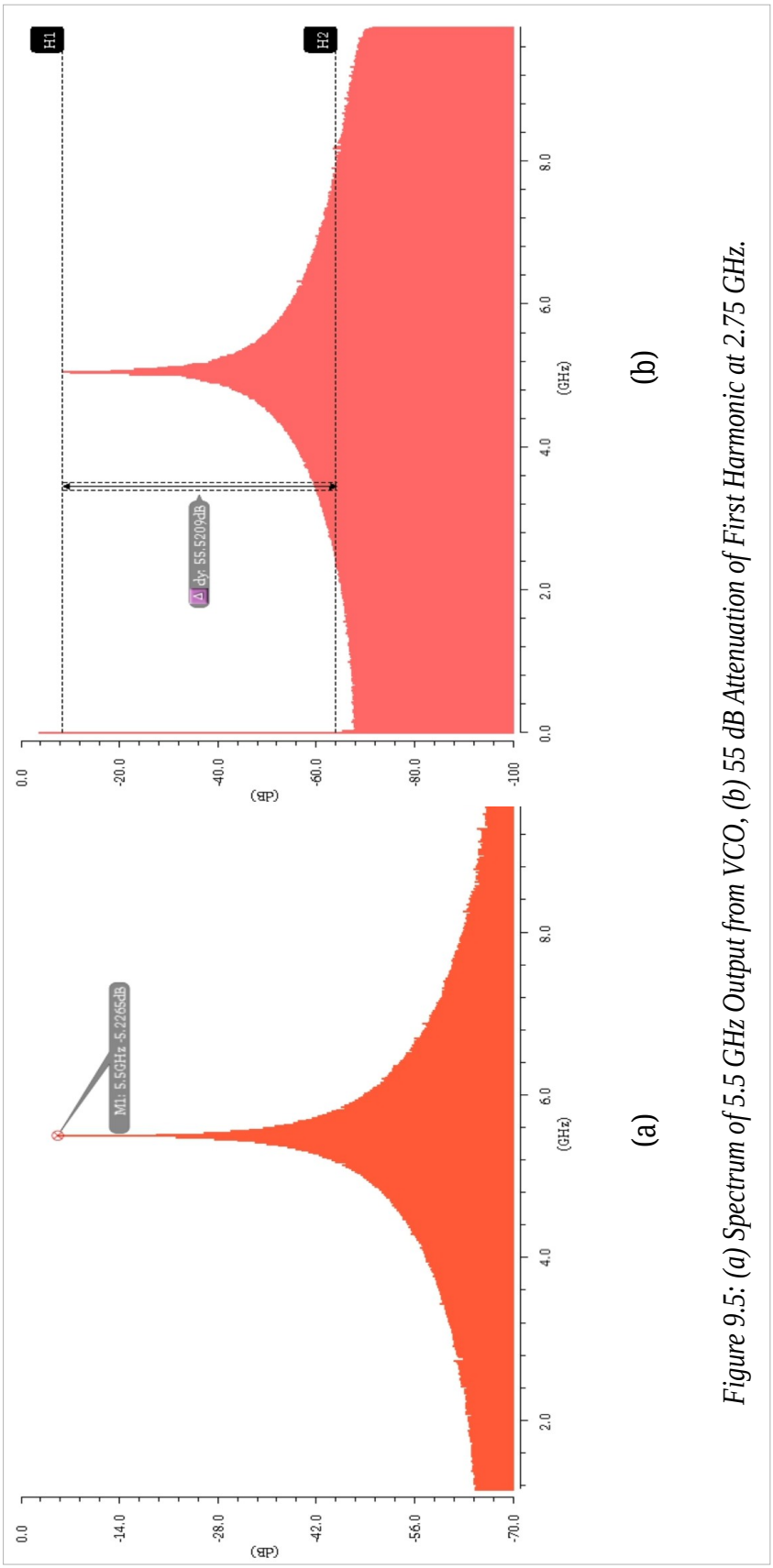

Figure 9.5: (a) Spectrum of 5.5 GHz Output from VCO, (b) 55 dB Attenuation of First Harmonic at 2.75 GHz.

Phase noise performance is illustrated in Figure 9.6. For integer-N division without the modulator, the synthesizer achieves -111.02 dBc/Hz at 1 MHz offset, which is acceptable given the specification in chapter 2. Due to converge and unavailability of suitable algorithm, the same analysis could not be performed for a fractional division. The jitter performance can be seen from the eye diagram in figure 9.7. Compared to what was seen in the case of the VCO, phase noise and jitter performance of the synthesizer fares worse due to the noise contribution from all the other blocks in the synthesizer. However, since phase noise is within the acceptable range, it can be assumed that the jitter is not going to drastically affect the performance of the system.

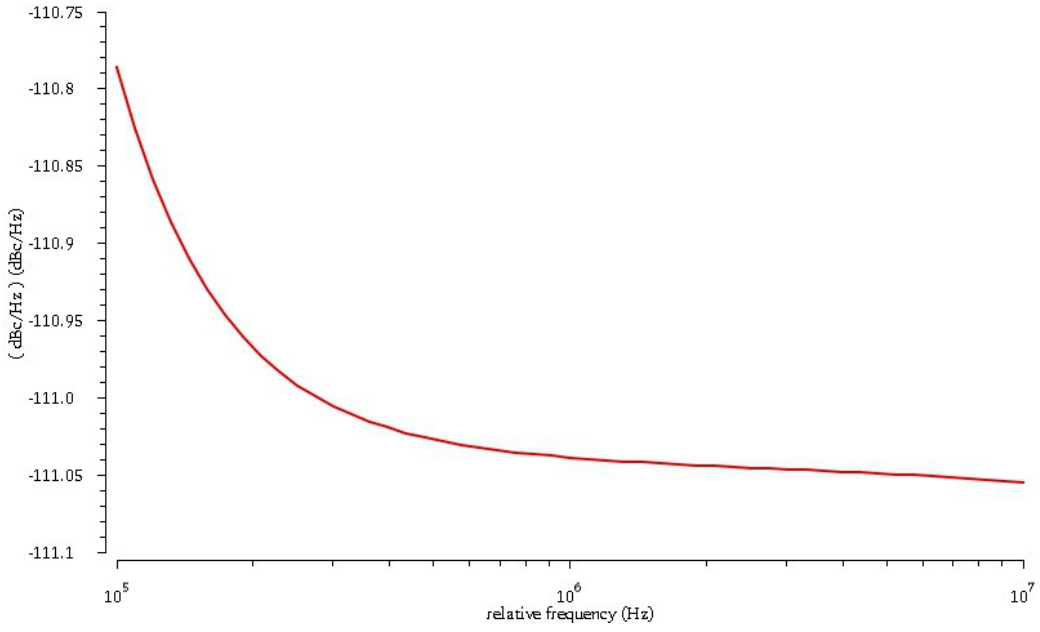

Figure 9.6: Phase Noise of Frequency Synthesizer at 1 MHz Offset

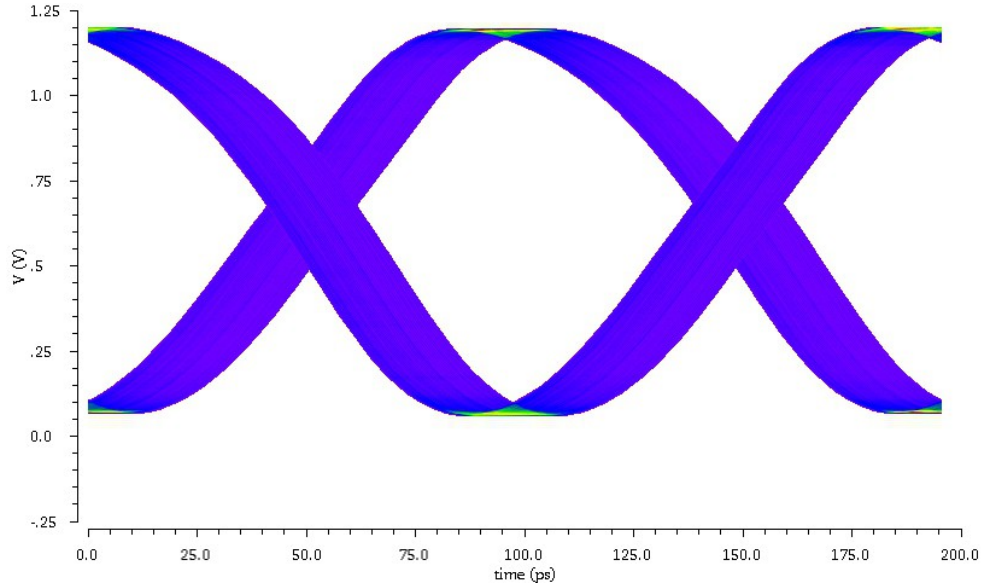

Figure 9.7: Eye Diagram of Frequency Synthesizer

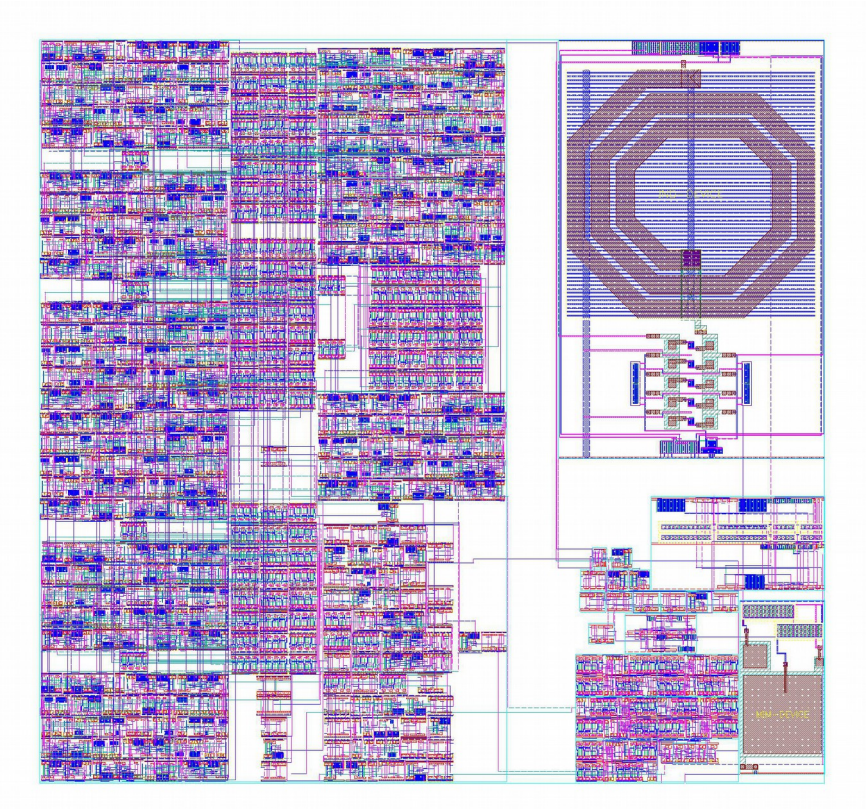

Figure 9.8: Layout of Complete Fractional-N Frequency Synthesizer

Finally, the complete layout of the frequency synthesizer, covering a total area of 0.628 mm X 0.594 mm, is shown in figure 9.8. Again, due to insufficient convergence, a DC simulation for the complete synthesizer could not be performed. However, estimation has been made using the DC simulation results from the previous chapters. Adding the power consumptions of all the individual blocks, it was estimated that in average the synthesizer consumes about 20.76 mW of DC power.

Chapter 10

Conclusion and Future Improvements

In this thesis, a monolithic fractional-N frequency synthesizer for 2.4 GHz and 5 GHz Wireless LAN application has been presented. The fractional-N synthesizer has demonstrated expected behavior across all its sub-blocks and measured satisfactory performance in accordance to the 802.11ac/b/g/n WLAN standards.

The thesis began with the introduction to phase-locked-loops and PLL based frequency synthesizers, their applications, and their importance in wireless communication in chapter 1. Brief comparison between integer-N and fraction-N frequency synthesizers, motivation to take on such a project, and short summaries of subsequent chapters have also been provided. In chapter 2, a complete analysis of 3rd order, type-II PLL based fractional-N frequency synthesizer has been presented. This analysis includes linear models of different blocks of the synthesizer, their mathematical explanation, open and closed loop transfer functions of the PLL, and operational principles of the fraction-N sythesizer.

Beginning with chapter 3 and ending in chapter 8, presentations of all different modules/blocks, including the voltage controlled oscillator (VCO), the phase-frequency detector (PFD), the charge

pump (CP), the loop filter (LF), the programmable frequency divider (FD), and the MASH 1-1-1 delta-sigma modulator (DSM) have been provided. Detailed explanations of circuit topologies/architectures, choice of design variables using reasonable assumptions, expected outcomes, and etc. are included in these chapters. To further support the design topologies and their expected outcomes, numerous low and high performance SPICE simulations have been performed and results have been presented. To end each of these chapter, mask layout of each block have been included, specifying how much area in a silicon wafer each block will cover.

Transient and phase noise simulation results of the complete fractional-N frequency synthesizer including delta-sigma modulator, and its layout are presented in chapter 9. The 0.628 mm X 0.594 mm frequency synthesizer has been found to acquire a lock time of approximately 2 μs for a fractional division ratio of 137.5. In an average case scenario, for a reference frequency of 40 MHz, the synthesizer was able to produce an output frequency of 5.5 MHz with a phase noise performance of – 115 dBc/Hz at 1 MHz offset.

There is a lot of room for improvements for the frequency synthesizer presented in this thesis. First of all, the reference frequency used for the designed synthesizer provides a constant oscillation. In order to add an increased flexibility on the choice of reference frequency, a reference divider can be added [32]. Although, sufficient care need to be

taken in order to meet the phase noise requirements which may increase due to the added reference division.

Since the frequency synthesizer is intended to be used in a wireless communication systems, it should be noted that many different WLAN transceiver topologies make use of quadrature oscillation generated by its local oscillator. The VCO in this thesis produces only a differential output—which may not meet the requirements of every wireless devices. In consideration of having a rather universal solution to all 802.11 systems, either the VCO must be constructed to produce quadrature oscillation, or an extra circuitry, such as a polyphase filter, should be added to generate the quadrature outputs [4], [33], [34].

Another improvement that could bring more robustness in this synthesizer would be implementation of a differential charge pump along with an added loop filter. A differential topology will provide better matching of the “Up” and “Down” currents and while reducing spurious tones generated at the output of the VCO. Additionally, an operational amplifier could be employed to add more stability and rigidity in the charge pump's output currents, while keeping a constant phase noise generation across a wider bandwidth [20], [35]. Differential loop filters will also benefit with better phase noise performance and let designers to be more flexible with their choice of loop filter components [12]. Nonetheless, because of added number of passive components, the area coverage of the

loop filters will increase. But loop filters can always be taken off the chip and the extra area could be used to build a higher order modulator.

Instead of having cascaded dual-modulus prescalers for the programmable divider, a pulse-swallow architecture may be adopted. Although it may not provide the synthesizer with a wide range of division ratios, it could provide more flexibility in the choice of integer division ratios—and if designed with the appropriate variables, may suppress fractional spurs better [36]. Nevertheless, the complexity of designing such circuit, and its added power consumption and area coverage, were some of the reason why the pulse-swallow architecture was not chosen for this thesis.

Improved layout techniques could be used to facilitate better area coverage, better matching (in order to allow unexpected PVT variations), added ESD protection, and prevention of unexpected latch-ups. To further improve latch-up prevention, spacing between different devices should be adequate and a large enough number of substrate contacts should be included. Furthermore, guard-rings may be added around the PMOS and the NMOS devices so that they do not form the unexpected parasitic thyristors.

Finally, as mentioned earlier, a higher order delta-sigma modulator will provide more output values, and perhaps could be utilized with an improved frequency divider to add more frequency resolution to

the synthesizer. With that feature, the fractional-N frequency synthesizer may have the capability to be used across all bands in the 802.11 standards. Additionally, if a frequency multiplier is added at the output stages of the synthesizer, perhaps the same synthesizer can be used over a much wider bandwidth covering a lot more wireless applications.

References

[1] “Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications”, IEEE Std. 802.11, 2012. [Online]. Available: http://standards.ieee.org/getieee802/download/802.11-2012.pdf.

[2] Southafrica.info, “Rural South Africa to get free wi-fi”, 2015. [Online]. Available: http://www.southafrica.info/about/education/ecape-wifi-260515.htm.

[3] H. Burri and D. Senouf, “Remote monitoring and follow-up of pacemakers and implantable cardioverter defibrillators”, *Europace*, vol. 11, no. 6, pp. 701-709, 2009.

[4] B. Razavi, *RF microelectronics*. Upper Saddle River, NJ: Prentice Hall, 2012.

[5] “Architectures for RF Frequency Synthesizers”, *The International Series in Engineering and Computer Science*, 2003.

[6] W. Kang, W. Hong and J. Zhao, “Optimized Transceiver Aids Broadband Wireless Access”, *Mwrf.com*, 2015. [Online]. Available: http://mwrf.com/systems/optimized-transceiver-aids-broadband-wireless-access.

[7] AnySilicon, “Wireless Transceivers RF CMOS Front Ends – AnySilicon”. [Online]. Available: http://anysilicon.com/wireless-transceivers-rf-cmos-front-ends/.

[8] J. Lee, K. Kim, J. Kwon, S. Lee, J. Kim and S. Lee, “A 3.8-5.5-GHz Multi-Band CMOS Frequency Synthesizer for WPAN/WLAN Applications”, *IEEE Custom Integrated Circuits Conference 2006*, 2006.

[9] R. Baker, *CMOS*. Hoboken, N.J.: IEEE Press/Wiley, 2010.

[10] V. Kratyuk, P. Hanumolu, Un-Ku Moon and K. Mayaram, “A Low Spur Fractional-N Frequency Synthesizer Architecture”, *2005 IEEE International Symposium on Circuits and Systems*, 2005.

[11] M. Kozak and I. Kale, *Oversampled delta-sigma modulators*. Boston: Kluwer Academic, 2003.

[12] C. Barrett, *Fractional/Integer-N PLL Basics*. Dallas: Texas Instruments, 1999.

[13] *Virtuoso Spectre Circuit Simulator and Accelerated Parallel Simulator RF Analysis User Guide*, 14th ed. San Jose: Cadence Design Systems, Inc., 2014.

[14] S. Yoon, “LC-tank CMOS Voltage-Controlled Oscillators using High Quality Inductors Embedded in Advanced Packaging Technologies”, Ph. D, Georgia Institute of Technology, 2004.

[15] T. Dellsperger, “Design of 5 GHz VCO in CMOS”, M. Sc., Swiss Federal Institute of Technology Zurich, 2002.

[16] A. Bavisi, S. Dalmia, M. Swaminathan and F. Ayazi, “A 802.11a WLAN oscillator with high Q embedded passives on laminate-type organic package”, *Proceedings of the 16th International Symposium on Power Semiconductor Devices & IC's*, 2003.

[17] A. Berny, “Analysis and Design of Wideband LC VCOs”, Ph.D, University of California, Berkeley, 2006.

[18] D. Solanki, R. Chandel, T. Alam and A. Nishad, “Design and Analysis of LC-VCO Using Mems Spiral Inductor”, *International Journal of Micro and Nano Systems*, vol. 2, no. 1, pp. 47-51, 2011.

[19] *CMRF7SF Model Reference Guide*. IBM Microelectronics Division, 2013.

[20] G. Gal, “Design of Fractional-N Phase Locked Loops for Frequency Synthesis from 30 to 40 GHz”, M. Eng, McGill University, 2012.

[21] A. Hajimiri and T. Lee, “A general theory of phase noise in electrical oscillators”, *IEEE J. Solid-State Circuits*, vol. 33, no. 2, pp. 179-194, 1998.

[22] B. Razavi, *Phase-locking in high-performance systems*. Piscataway, N.J.: IEEE, 2003.

[23] S. Jade, “Design of Bluetooth Frequency Synthesizer”, M.S., San Jose State University, 2005.

[24] E. Sánchez-Sinencio and K. Shu, *CMOS PLL Synthesizers*. [New York]: Springer Science+Business Media, Inc., 2005.

[25] G. Nadkarni, *Design of Voltage References BMR and BGR*. Georgia Institute of Technology, 2013.

[26] R. Baker, “CMOSedu bad design 4”, *Cmosedu.com*, 2012. [Online]. Available: http://cmosedu.com/cmos1/bad_design/bad_design4/bad_design_4.htm.

[27] V. Manthena, “Ultra Low Power CMOS Phase-Locked Loop Frequency Synthesizers”, Ph. D, Nanyang Technological University, 2011.

[28] M. Sabbagh, “0.18µm Phase/Frequency Detector and Charge Pump Design for Digital Video Broadcasting for Handheld's Phase-Locked-Loop Systems”, M.S., Ohio State University, 2007.

[29] K. Majeed and B. Kailath, “Low power, high frequency, free dead zone PFD for a PLL design”, *2013 IEEE Faible Tension Faible Consommation*, 2013.

[30] M. Mansuri, D. Liu and C. Yang, “Fast frequency acquisition phase-frequency detectors for Gsamples/s phase-locked loops”, *IEEE J. Solid-State Circuits*, vol. 37, no. 10, pp. 1331-1334, 2002.

[31] N. Weste and D. Harris, *CMOS VLSI design*. Boston: Addison Wesley, 2011.

[32] T. Bourdi and I. Kale, *CMOS single chip fast frequency hopping synthesizers for wireless multi-gigahertz applications*. Dordrecht: Springer, 2007.

[33] *Fractional-N Frequency Synthesizer*. Norwood: Analog Devices, Inc., 2012.

[34] R. Li, *RF circuit design*. Hoboken, New Jersey: John Wiley & Sons, Inc., 2012.

[35] T. Soldner, “Design of a Delta-Sigma Fractional-N PLL Frequency Synthesizer at 1.43GHz”, M.S., University of Minnesota, 2012.

[36] D. Banerjee, *PLL performance, simulation, and design*. Indianapolis, IN.: Dog Ear Publishing, 2006.

[37] M. Jalali, A. Bakhtiar and S. Mirabbasi, “A charge-pump with a high

output swing for PLL and CDR applications", *Proceedings of the 8th IEEE International NEWCAS Conference 2010*, 2010.

[38] C. Zhang, T. Au and M. Syrzycki, "A high performance NMOS-switch high swing cascode charge pump for phase-locked loops", *2012 IEEE 55th International Midwest Symposium on Circuits and Systems (MWSCAS)*, 2012.

[39] S. Chen, Z. Li and Q. Li, "An improved high speed charge pump in 90 nm CMOS technology", *2011 IEEE 13th International Conference on Communication Technology*, 2011.

[40] X. He, "Low Phase Noise CMOS PLL Frequency Synthesizer Design and Analysis", Ph.D., University of Maryland, College Park, 2007.

[41] P. Hasler, *Current Mirrors*. Georgia: Georgia Institute of Technology, 2000.

[42] A. Kitagawa, *Voltage References*. Kanazawa: Kanazawa University.

[43] Y. Kuo and R. Weng, "A Dynamic-Logic Frequency Divider for 5-GHz WLAN Frequency Synthesizer", *2007 17th International Conference Radioelektronika*, 2007.

[44] C. Quemada, G. Bistué and I. Adin, *Design methodology for RF CMOS phase locked loops*. Boston: Artech House, 2009.

[45] R. Desikachari, "High-Speed CMOS Dual-Modulus Prescalers for Frequency Synthesis", M.S., Oregon State University, 2003.

[46] M. Perrott, *High Speed Frequency Dividers, Phase Detectors, Charge Pumps, and Loop Filter Design*. 2008.

[47] S. Wadhwa, “Design of a low power high speed differential to single ended converter using feedback”, *Embedded*, 2013. [Online]. Available: http://www.embedded.com/design/power-optimization/4408179/Design-of-a-low-power-high-speed-differential-to-single-ended-converter-using-feedback-.

[48] M. Perrott, *High Speed Frequency Dividers*. Massachusetts: Massachusetts Institute of Technology, 2003.

[49] H. Luong and G. Leung, *Low-voltage RF CMOS frequency synthesizers*. New York: Cambridge University Press, 2004.

[50] I. Poole, “PLL Loop Filter”, *Radio-electronics.com*. [Online]. Available: http://www.radio-electronics.com/info/rf-technology-design/pll-synthesizers/loop-filter-design.php.

[51] *An Analysis and Performance Evaluation of a Passive Filter Design Technique for Charge Pump PLL’s*. National Semiconductor, 2001.

[52] C. Chen, J. Ho, W. Liou and R. Hsiao, “A 5.2GHz CMOS fractional-n frequency synthesizer with a MASH delta-sigma modulator”, *2008 51st Midwest Symposium on Circuits and Systems*, 2008.

[53] Y. Zhu, Z. Shao and W. Pang, “A generalized MASH architecture in fractional-N synthesizer”, *Proceedings. 7th International Conference on Solid-State and Integrated Circuits Technology, 2004.*, 2004.

[54] T. Xu and M. Condon, “Comparative study of the MASH digital delta-sigma modulators”, *2009 Ph.D. Research in Microelectronics and Electronics*, 2009.

[55] S. Kim, M. Keel, K. Lee and S. Kim, “CMOS Delta-Sigma Frequency Synthesizer with a New Frequency Divider and a Simplified MASH Structure”, *Journal of the Korean Physical Society*, vol. 41, no. 6, pp. 967 - 973, 2002.

[56] E. Atalla, E. Hegazi and M. Marzouk Ibrahim, “On the Design of Error Cancellation Network for MASH Delta-Sigma Frequency discriminators”, *2007 IEEE International Symposium on Circuits and Systems*, 2007.

[57] R. McConnell, “Basic MASH 111 Sigma Delta Simulation - File Exchange - MATLAB Central”, *Mathworks.com,* 2013. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/24296-basic-mash-111-sigma-delta-simulation.

[58] B. Gilbert, “Phase Locked Loop Synthesis and Simulation”, *Mathworks.com*, 2009. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/23588-phase-locked-loop-synthesis-and-simulation/content/pll_simulation.m.

[59] I. Poole, “Wi-Fi / WLAN Channels, Frequencies, Bands & Bandwidths”, *Radio-electronics.com*. [Online]. Available: http://www.radio-electronics.com/info/wireless/wi-fi/80211-channels-number-frequencies-bandwidth.php.

Appendix

A.1 MATLAB Program for Delta-Sigma Modulator's PSD

The following program was created by Rick McConnell and it was extracted from from Mathworks community website [57].

```
NumberSamples = 2^24;
BusSize = 24; %bits
Fraction = 0.55; %usable 0 to 1
FractionInternal = 2^BusSize * Fraction;
AccumulatorBits = 24; %bits
AccumulatorSize = 2^AccumulatorBits;

C1(1:NumberSamples) = 0; %Carry out of the first accumulator
C2(1:NumberSamples) = 0; %Carry out of the 2nd accumulator
C3(1:NumberSamples) = 0; %Carry out of the 3rd accumulator
U1(1:NumberSamples) = 0; %output of the 1st accumulator
U2(1:NumberSamples) = 0; %output of the 2nd accumulator
U3(1:NumberSamples) = 0; %output of the 3rd accumulator
Yout1(1:NumberSamples) = 0; %output to the divider for 1
stage DSM
Yout2(1:NumberSamples) = 0; %output to the divider for 2
stage DSM
```

```
Yout3(1:NumberSamples) = 0; %output to the divider for 3
stage DSM

for index = 3:NumberSamples
    U1(index) = FractionInternal+U1(index – 1);
    U2(index) = U1(index - 1) + U2(index – 1);
    U3(index) = U2(index - 1) + U3(index – 1);

    if U1(index) > AccumulatorSize
        C1(index) = 1; %carry 1
        U1(index) = U1(index) – AccumulatorSize;
    end

    if U2(index) > AccumulatorSize
        C2(index) = 1; %carry 2
        U2(index) = U2(index) – AccumulatorSize;
    end

    if U3(index) > AccumulatorSize
        C3(index) = 1; %carry 3
        U3(index) = U3(index) – AccumulatorSize;
    end

    %The output is the overflow from accumulator 1, plus the
     difference of the overflow from accumulator 2, plus the
     2nd derivative of the overflow from accumulator 3
```

```
    Yout3(index) = C1(index) + C2(index) - C2(index - 1) +
     C3(index) - 2 * C3(index - 1) + C3(index - 2); %output
     to the divider - 3 stages
    Yout2(index) = C1(index) + C2(index) - C2(index - 1);
     %output to the divider - 2 stages
    Yout1(index) = C1(index); %output to the divider - 1
     stages
end

MeanFrac = mean(Yout3);
fprintf('\nMeanFracMASH = %1.4f\n', MeanFrac)
figure(1)
SignalFreq1 = 20 * log10(abs(fft(Yout1)));
SignalFreq2 = 20 * log10(abs(fft(Yout2)));
SignalFreq3 = 20 * log10(abs(fft(Yout3)));
hold on
grid on
axis([3.21e4 NumberSamples -120 0]);
plot(fftshift(SignalFreq3) - max(SignalFreq3), 'b')
title('MASH 1-1-1 Modulator Noise PSD')
ylabel('PSD (dB)')
xlabel('Frequency (Hz)')
```

A.2 MATLAB Program for Loop Gain and Phase Margin

The following program was created by Ben Gilbert and it was extracted from from Mathworks community website [58].

```
C1 = 4.189e-12;
C2 = 60.38e-12;
C3 = 418.8e-15;
C4 = 0;
R2 = 21.35e3;
R3 = 94.51e3;
R4 = 0;

% Conversion of parameters to more convenient units
Kpd = 100e-6 / 2 / pi; % phase detector gain
Kvco = 180e6 * 2 * pi; % vco gain

% Plot Setup
% Generates logarithmic spaced points for the calculations
fplotstart = 10; % Hz
fplotstop = 10E9; % Hz
plotpoints = 100;
pltfreqs = [];

for ppts = 0:plotpoints
    pltfreqs = [pltfreqs fplotstart * 10 ^ (ppts /
```

```
     plotpoints * log10(fplotstop / fplotstart))];
end

% Loop Poll's Polynomial Coefficients
A0 = C1 + C2 + C3 + C4;
A1 = C2 * R2 * (C1 + C3 + C4) + R3 * (C1 + C2) * (C3 + C4) +
     C4 * R4 * (C1 + C2 + C3);
A2 = C1 * C2 * R2 * R3 * (C3 + C4) + C4 * R4 * (C2 * C3 * R3
     + C1 * C3 * R3 + C1 * C2 * R2 + C2 * C3 * R2);
A3 = C1 * C2 * C3 * C4 * R2 * R3 * R4;
T2 = R2 * C2;

% Filter Transfer Function
Zfilt = @(s)  (1 + s .* T2) ./ s ./ (A3 .* s .^3 + A2 .*
     s .^2 + A1 .* s + A0); % filter transfer function
zfilt = @(f) Zfilt(2 * pi * 1i * f); % filter transfer
     function as a function of frequency

% VCO Transfer Function
Gvco = @(s) Kvco ./ s; % vco transfer function
gvco = @(f) Gvco(2 * pi * 1i * f); % vco transfer function
     expressed as a function of frequency

% Forward Path Transfer Function
G = @(s)  Kpd * Gvco(s) .* Zfilt(s); % forward path transfer
     function
```

```
% Reverse (Feedback) Transfer Function
H = 40e6 / 5825e6; % feedback path transfer function

% Open Loop Transfer Function
GH = @(s) G(s) * H; % open loop transfer function
gh = @(f) GH(2 * pi * 1i * f); % open loop transfer function
     expressed as a function of frequency

% Bode Plot
figure;
    subplot(2,1,1);
        semilogx(pltfreqs, 20 * log10(abs(gh(pltfreqs))));
        grid on;
        title('Open Loop Magnitude');
    subplot(2,1,2);
        semilogx(pltfreqs, angle(gh(pltfreqs)) .* 180 / pi);
        grid on;
        title('Open Loop Phase');
        ylim([-180 180]);

% Find open loop bandwidth and phase margin
ghdB = @(f) 20 * log10(abs(gh(f)));
bandwidth = fzero(ghdB, [fplotstart fplotstop]); % find
     bandwidth numerically
pm = 180 + angle(gh(bandwidth)).* 180 / pi;
```

A.3 2.4 GHz and 5 GHz Channels

The following table has been extracted from an article written by Ian Poole on the Radio-Electronics website [59].

Table A.1: List of Channels in 2.4 and 5 GHz Bands

2.4 GHz Channels	
Channel Number	*Frequency (MHz)*
1	2412
2	2417
3	2422
4	2427
5	2432
6	2437
7	2442
8	2447
9	2452
10	2457
11	2462
12	2467
13	2472

14	2484
5 GHz Channels	
Channel Number	*Frequency (MHz)*
36	5180
40	5200
44	5220
48	5240
52	5260
56	5280
60	5300
64	5320
100	5500
104	5520
108	5540
112	5560
116	5580
120	5600
124	5620
128	5640

132	5660
136	5680
140	5700
149	5745
153	5765
157	5785
161	5805
165	5825