

Design of Fractional-N Phase Locked Loops For Frequency Synthesis From 30 To 40 GHz

George Gal



Department of Electrical & Computer Engineering
McGill University
Montreal, Canada

October 2012

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment
of the requirements for the degree of Master of Engineering.

© 2012 George Gal

Abstract

High-frequency fractional-N PLLs in CMOS technology in the 30 to 40 GHz are very difficult to design when considering power, area, phase noise requirements and frequency range of operation. One of the difficulties is to synthesize the loop filter of the PLL such that it meets the phase noise characteristics using the information available for all the components that make up the PLL. At the same time, predicting the phase noise output of the PLL using extracted layout results takes a long time to simulate and often the solution does not converge, thereby lengthening the design cycle. This thesis proposes a new methodology for designing high performance wide-band fractional-N PLLs in the 30-40 GHz range. The method begins by first designing the phase-frequency detector/charge-pump, voltage-controlled oscillator and frequency divider circuit for realization in a specific CMOS technology. The method of choice mixes insight deemed from both a theoretical and simulation perspective. Next, the loop filter is derived based on the layout extracted behaviour of each component. Once complete, all components of the PLL are described using the high-level description language of Verilog-A available in the Cadence tool set over its full range of operating characteristics. Ideally, these components would be fabricated first and characterized afterward. The Verilog-A description of the PLL enables a fast and efficient simulation of the complete PLL in a closed-loop configuration. This latter steps allows further optimization of the overall design. Two chips have been fabricated; one in a $0.13\ \mu\text{m}$ CMOS process from IBM and another in a 65 nm CMOS process from TSMC. One chip contain the design of a 28 GHz VCO and another containing the design of a programmable frequency divider circuit. Experimental results for both chip are provided.

Résumé

Les systèmes de boucle à phase asservie fabriqués dans une technologies de CMOS et pour de hautes fréquences se situant entre 30 et 40 GHz tout en respectant divers contraintes tels que la puissance électrique requise, la surface occupée sur la puce, les exigences de bruit de phase ainsi que la plage de fréquences à couvrir constituent un défi majeur de conception. Une des difficultés consiste à synthétiser le filtre de la boucle du système de boucle à phase asservie à partir des caractéristiques des composantes faisant parties du système afin de rencontrer les exigences imposées sur le bruit de phase. Les simulations basées sur les circuits des composants extraites du "layout" pour prédire le bruit de phase de la boucle à phase asservie sont d'autant plus longues et sujettes à des problèmes de convergences, augmentant ainsi le temps requis pour leur conception. Ce mémoire de maîtrise propose une méthodologie visant l'optimization du bruit de phase pour les systèmes de boucle à phase asservie opérant dans les fréquences de 30 à 40 GHz. La première étape consiste à concevoir le détecteur de phase et fréquence, l'oscillateur contrôlé en voltage, ainsi que le diviseur de fréquence dans une technologie de CMOS choisie. La deuxième étape se base sur la théorie et les résultats provenant de la simulation du circuit extrait en "layout" afin de dériver le filtre principal. Une fois la structure du filtre établis, les composantes, idéalement fabriquées sur la puce, sont caractérisées et ensuite modélisées dans un langage de haut-niveau tel que le Verilog-A. Cette étape permet d'extraire la performance générale du système en boucle fermée tout en réduisant le temps de simulation, permettant ainsi de se concentrer sur l'optimization du système dans son ensemble. Deux puces ont été fabriquées; une dans la technologie d'IBM 130 nm et l'autre en 65 nm de TSMC. La première puce contient le circuit d'un oscillateur contrôlé en voltage, et la seconde, le circuit d'un diviseur à haute fréquence. Les résultats expérimentaux de ces deux puces sont présentés ainsi que leur integration dans le modèle de haut niveau.

Acknowledgments

First I would like to thank my parents, my aunts and grand-parents for their dedication to education and moral support throughout my studies. I would like to thank my brother as well for the moral support and help with software issues, etc. Secondly, I would like to thank professor Gordon Roberts for letting me join his lab and giving me the opportunity to work on a very challenging project. I would also like to thank him for his mentoring and time he spend with us trying to solve issues we faced. Thirdly, I thank my friend and colleague Jean-Samuel Chenard for his numerous advices on linux, and printed circuit board design, along with being morally supportive. I would also like to thank all the colleagues in MACS lab, for numerous advices or simply as being good friends and for the fun we all had together.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Thesis Contributions/Literature Review	2
1.3	Thesis Overview	3
2	Background Theory of Phase Locked Loops	5
2.1	System Perspective of A Standard PLL	5
2.2	PLL Components	6
2.3	PLL Phase-Domain Model	9
2.3.1	PLL Phase-Domain Transfer Function	9
2.4	Design For Specific PLL Transfer Function	11
2.4.1	Lead-Lag Filter	11
2.4.2	Zero-Introducing Method	13
2.5	PLL Requirements/Constraints	14
2.6	PLL Design Procedure with the VCO as the Dominant Noise Source	18
2.7	Summary	20
3	Voltage Controlled Oscillator For Gigahertz Operation	22
3.1	VCO Architectures	22
3.2	Oscillator Phase Noise Theory	24
3.2.1	VCO Equivalent Model	26
3.2.2	VCO Design Equations	30
3.3	Two Independent VCO Designs	33
3.3.1	Design Of A 28 GHz LC-Tank VCO in 0.13 μ m CMOS Process	34
3.3.2	Design Of A 30 - 40 GHz LC-Tank VCO in 65 nm CMOS Process	41

3.4	A VCO Verilog-A Model	49
3.5	Summary	53
4	Phase Frequency Detector and Charge Pump Design	54
4.1	PFD Block	54
4.1.1	PFD Non-Ideal Behavior	54
4.1.2	PFD Circuit Level Implementation	58
4.2	Charge Pump Block	59
4.2.1	Charge Pump Non-Ideal Behaviour	59
4.2.2	Current-Voltage Variation	60
4.2.3	Current Mismatch	61
4.2.4	Phase Noise	63
4.2.5	High-Performance Charge Pump Topologies	64
4.3	Extracted Layout Results Tor The PFD and CP	75
4.3.1	PFD Simulation Results	75
4.3.2	Combined PFD and CP Simulation Results	77
4.4	PFD and CP Verilog-A Model Implementation	81
4.5	Summary	86
5	Loop Filter Design	87
5.1	Synthesis of The Loop Filter From The PLL Phase Noise Requirements	87
5.2	Loop-Filter Simulation Using A Verilog-A Hardware Description	95
5.3	Predicting The Output PLL Phase Noise Performance	97
5.4	Iterative Procedure For Loop Filter Selection	100
5.5	Passive/Active Loop Filter Realization	101
5.6	Summary	104
6	Programmable Divider For Fractional-N Frequency Synthesis	105
6.1	$\Delta\Sigma$ Modulators	106
6.2	A Fractional-N Modulus Divider	110
6.2.1	Verilog-A Implementation Of A Fractional-N Modulus Divider	113
6.3	$\Delta\Sigma$ Modulator Contribution To PLL Output Phase Noise	117
6.4	Implementation Details For A Programmable Frequency Divider Circuit	120
6.4.1	A Modulus Counter Approach	121

6.4.2	A Programmable Counter Approach	126
6.4.3	Programmable Frequency Divider Circuit For IC Implementation	130
6.5	Conclusion	137
7	Experimental Results	138
7.1	Fabricated Chip Results	138
7.2	65 nm 0-18 GHz Frequency Divider	144
7.3	Conclusion	145
8	Conclusion	146
8.1	Thesis Summary	147
8.2	Future Work	148
A	$\Delta\Sigma$ Modulators	149
A.1	$\Delta\Sigma$ Modulators	149
A.1.1	All Zero Pole Realization	149
A.1.2	Fixed Pole/Zero Realization	152
A.1.3	Arbitrary Pole-Zero Realization	153
A.2	$\Delta\Sigma$ Topology Comparison	156
B	Optimized Arbitrary Zero-Pole Placement $\Delta\Sigma$ Structure	161
C	Front-End Delta Sigma Frequency Synthesis	163
C.0.1	Bit-Stream Fractional-N Frequency Synthesis	163
C.0.2	Phase Noise Of Front-End Fractional-Synthesizer	167
	References	173

List of Figures

2.1	Phase Locked Loop Architecture	6
2.2	Phase Frequency Dectector	7
2.3	Ideal Charge Pump	8
2.4	Ideal Voltage Controlled Oscillator	8
2.5	Divider Operation	9
2.6	PLL Phase-Domain Representation	10
2.7	Comparing the Frequency and Phase Response of Overall Closed Loop PLL Response Using The Lead-Lag and Zero-Introducing Methods	14
2.8	Resulting Filters Transfer Function Frequency and Phase Responses Using the Lead-Lag and Zero-Introducing Methods	15
2.9	Phase Noise Spectrum of an Oscillator	17
2.10	Starting Point For Design Of PLL Can be A Mess	19
2.11	Proposed Design Method For a PLL	21
3.1	Two Different VCO Architectures: (a) Current-Starved Ring Oscillator (b) Distributed Oscillator	23
3.2	LC-Tank VCO	24
3.3	A Simplified LC-Tank VCO and Its Equivalent Passive Representation	27
3.4	A Symmetric LC-Tank Equivalent Model	28
3.5	A Small-Signal Equivalent Model Of A Pair OF Cross-Coupled NMOS Tran- sistors	29
3.6	Illustrating The Various Noise Sources In An LC-Tank VCO	32
3.7	28-GHz LC-Tank Circuit	34
3.8	Phase Noise 28GHz VCO Schematic	36
3.9	28-GHz LC-Tank VCO Schematic and Layout	37

3.10	28-GHz LC-Tank VCO Layout Zoomed	38
3.11	28-GHz LC-Tank VCO Extracted Layout Output Voltage At 50 ohm Load (Top) and VCO Node (Bottom)	39
3.12	28-GHz LC-Tank VCO Layout Phase Noise	39
3.13	28 GHz LC-Tank Layout Extracted Frequency vs Vctrl	40
3.14	28 GHz LC-Tank Layout Extracted K_{VCO} vs Vctrl	41
3.15	30-40GHz 65nm LC-Tank VCO Schematic	42
3.16	30-40GHz 65nm LC-Tank VCO Frequency Range vs Control Voltage	43
3.17	30-40GHz 65nm LC-Tank K_{VCO} vs Control Voltage	44
3.18	30-40GHz 65nm LC-Tank Schematic Circuit Phase Noise	45
3.19	31-40GHz LC-Tank VCO Schematic and Layout	46
3.20	31-40GHz LC-Tank VCO Extracted Layout K_{VCO} vs V_{ctrl}	47
3.21	31-40GHz LC-Tank VCO Extracted Layout Phase Noise	48
3.22	Modulus Integrator and Addition of Jitter Process	49
4.1	Phase Frequency Detector DC Output vs Input Phase Difference (a) Ideal (b)Dead-Zone (c)Blind-Zone	55
4.2	Illustrating The PFD Dead-Zone Non-Ideal Behavior	56
4.3	Illustrating The PFD Blind-Zone Non-Ideal Behavior	57
4.4	PFD Block Level Implementation	58
4.5	PFD Transistor Level Implementation	59
4.6	PFD Input-Output Transfer Characteristic	60
4.7	A Zoomed In View Of The PFD Blind-Zone Region Of Fig. 4.6.	61
4.8	Simple Charge Pump Circuit	62
4.9	CP Source and Sinking Currents vs Output DC Voltage	63
4.10	PFD Outputs And CP Mismatch Effects On Vout in Steady State	64
4.11	Charge Pump With Feedback Op-Amp Included	65
4.12	Simple Charge Pump With Feedback Op-Amp Current Mismatch	65
4.13	CP With 100 MHz BW Op-Amp With 15 k Ω Load <i>Top trace: PFD $\overline{UP}$ Signal, Second trace: V_{B2} Signal</i> <i>Third trace: Resistor Load Current, Bottom trace: $V_X - V_{OUT}$ Signal</i> . . .	66

4.14	CP With 50 MHz BW Op-Amp With 15 k Ω Load <i>Top trace: PFD $\overline{UP}$ Signal, Second trace: V_{B2} Signal</i> <i>Third trace: Resistor Load Current, Bottom trace: $V_X - V_{OUT}$ Signal</i>	67
4.15	CP With 100 MHz BW Op-Amp With 15 k Ω Load and Slew Rate of 200 MV/s <i>Top trace: PFD $\overline{UP}$ Signal, Middle trace: V_{B2} Signal</i> <i>Bottom trace: Resistor Load Current</i>	68
4.16	CP With 100 MHz BW Op-Amp With 15 k Ω Load and Slew Rate of 100 MV/s <i>Top trace: PFD $\overline{UP}$ Signal, Middle trace: V_{B2} Signal</i> <i>Bottom trace: Resistor Load Current</i>	69
4.17	CP With 100 MHz BW Op-Amp With 1 k Ω Load and Slew Rate of 1000 MV/s <i>Top trace: PFD $\overline{UP}$ Signal, Middle trace: V_{B2} Signal</i> <i>Bottom trace: Resistor Load Current</i>	70
4.18	Complementary CP Circuit From Reference [1]	71
4.20	Transient Analysis for the Complementary CP Circuit with a 1 k Ω Resistive Load <i>Top trace: Resistor Load Current, Second trace: PFD DN Signal</i> <i>Third trace: PFD $\overline{UP}$ Signal, Fourth trace: V_{o1} Signal, Bottom trace: V_{o2} Signal</i>	72
4.19	Complementary CP Circuit Sinking and Sourcing Currents vs Output Voltage	73
4.21	Transient Analysis for the Complementary CP Circuit with a 100 pF Capacitive Load <i>Top trace: PFD DN Signal, Second trace: PFD $\overline{UP}$ Signal,</i> <i>Third trace: V_{OUT}</i>	74
4.22	Layout of The PFD	75
4.23	Phase Frequency Detector Extrated Layout DC Average vs In Phase Difference	76
4.24	Phase Frequency Detector Extrated Layout Blind Zone Region	76
4.25	CP Layout Without Op-Amp Circuits	77
4.26	CP DC UP and DN Current vs Vout	78
4.27	Up Current Transient Simulation <i>Top trace: Output Load Current, Second trace: PFD $\overline{UP}$ Signal, Third trace: PFD DN Signal</i>	79
4.28	Down Current Transient Simulation <i>Top trace: Output Load Current, Second trace: PFD $\overline{UP}$ Signal, Third trace: PFD DN Signal</i>	80
4.29	Extracted Layout PFD and CP Input Referred Phase Noise vs Frequency .	80
4.30	Verilog-A PFD and CP UP and DN Currents vs Vout	84
4.31	Verilog-A PFD and CP DN Currents Transient	84

5.1	A Comparison Of The Magnitude/Phase Response of VCO Noise Transfer Response Under Maximum Loop Gain Conditions	90
5.2	A Comparison Of The Magnitude/Phase Response of VCO Noise Transfer Response Under Minimum Loop Gain Conditions	91
5.3	A Comparison Of The Magnitude/Phase Response of VCO Noise Transfer Response Under Average Loop Gain Conditions	92
5.4	A Comparison Of The Magnitude/Phase Response of VCO Noise Transfer Response Under Minimum Loop Gain Conditions And A 3-dB bandwidth of 4 MHz	93
5.5	(a) Magnitude and Phase Response Of The Input-Output Closed-Loop Response Of The PLL With A 4 MHz Bandwidth (b) PLL Step Response . .	94
5.6	Comparing The PLL Output Phase Noise Verilog-A Simulation Results With Those Predicted By Theory	97
5.7	Verilog-A Simulation Results For The PLL Control Voltage vs Time (a) Transient Result from 0 to 50 μ s (b) Variation In Steady-State Voltage. .	98
5.8	PLL Output Phase Noise With Nonlinear VCO Operating At 35 GHz . . .	99
5.9	PLL Output Phase Noise With Fixed-Gain VCO Operating At 35 GHz . .	100
5.10	Loop Filter High Level Realization Based on Modal Form	103
5.11	Loop Filter High Level Realization Based on Modal Form	104
6.1	Fractional-N Phase Locked Loop	106
6.2	A Simplified Representation of The Operation Of A Delta-Sigma Modulator	107
6.3	Single-Loop Feedback Representation Of A $\Delta\Sigma$ Modulator: (a) general block diagram (b) linear equivalent representation.	107
6.4	Unity-Gain SFT $\Delta\Sigma$ Topology	109
6.5	Noise Mapping $\Delta\Sigma$ Modulated Divider	111
6.6	Output Spectrum At Divider Output For SSMF-II $\Delta\Sigma$ Modulated Divider (Fs of 299.7186 MHz)	112
6.7	Output Spectrum At Divider Output For SSMF-II $\Delta\Sigma$ Modulated Divider (Fs of 444.444MHz)	112
6.8	Verilog-A Integration Operation Along With Change Of Period Crossing .	114
6.9	Output Phase Noise Standalone VCO With Embedded Fractional Divider .	115
6.10	Linear and Circular Integrator Functions Captured By A Verilog-A Model	116

6.11 Feedback Fractional-N $\Delta\Sigma$ Fractional Synthesizer Frequency Equivalent Model	119
6.12 Predicted Phase Noise At PLL Output Cause By The $\Sigma\Delta$ Modulator Quantization Noise	120
6.13 (a) Divide-By-2 Circuit (b) Divide-by-3 Circuit (c) Divide-by-3 Waveform .	121
6.14 Modulus 2/3 Counter	122
6.15 Modulus 4-to-7 Counter	123
6.16 A CML D-Type Flip-Flop	124
6.17 CML AND Flip-Flop	125
6.18 CML 4 To 7 Modulus Divider	126
6.19 6-Bit Asynchronous Ripple Counter	127
6.20 6-Bit Asynchronous Counter Logic Controller From Reference [2] and Adapted For Realization Using the TSMC Digital Logic Library	127
6.21 6-Bit Ripple Counter Layout Using TSPC D-Type Flip-Flops	128
6.22 Circuit Schematic Of A TSPC D-Type Flip-Flop	129
6.23 Layout Of TSPC D-Type Flip-Flop	129
6.24 D-Type Flip-Flop With Load Function Assembled From TSMC 65nm Components	130
6.25 Priority Encoder Circuit Built Using TSMC 65nm Digital Component Library	131
6.26 Overall Frequency Divider Schematic Programmable Counter Output For 10 GHz Input Signal and A Frequency Division of 210	134
6.27 Overall Frequency Divider Schematic Output For 10 GHz Input Signal and A Frequency Division of 210	134
6.28 Layout OF The Complete Programmable Frequency Divider Circuit	135
6.29 Overall Frequency Divider Layout Extracted Output For 10 GHz Input Signal and A Frequency Division of 210	136
6.30 Overall Frequency Divider Layout Extracted Output For 10 GHz Input Signal and A Frequency Division of 210 At Temperature of 75°C	136
7.1 Chip Layout Of The LC-Tank VCO (Two Different Variants Of the Same Design Are Shown) (1). <i>VCO With Power Amplifiers</i> , (2). <i>Stand-Alone VCO</i>	139
7.2 VCO Test Setup	140
7.3 IC Chip Bounded Directly To PCB	141

7.4	Output Tone For Vctrl of Zero Volts	142
7.5	Output Phase Noise At 300-KHz Offset	143
7.6	Measured Output Frequency vs Control Voltage For VCO	144
7.7	Microphotograph of A Portion Of The IC Containing the 4-to-7 Modulus Frequency Divider	145
A.1	Mash-III $\Delta\Sigma$ Modulator	150
A.2	Mash-III $\Delta\Sigma$ Modulator Equivalent Linear Model	151
A.3	MashI-II $\Delta\Sigma$ Modulator	152
A.4	SSMF-II $\Delta\Sigma$ Modulator	153
A.5	Power Density in NTF Pass Band and Quantization Band	154
A.6	$\Delta\Sigma$ Poles and Zeros Usual Location Region	156
A.7	NTF of $\Delta\Sigma$ Modulator Magnitude Response	157
A.8	In-Band Total Power Noise for Different $\Delta\Sigma$ Modulators	158
A.9	Spurs For DC Encoded Signal for SSMF-II $\Delta\Sigma$ s	159
A.10	SNR VS AC Input Power For Different $\Delta\Sigma$ s	160
B.1	Higher Order Synthesized $\Sigma\Delta$ Modulator Filter Structure By Xavier	162
B.2	Higher Order Synthesized $\Sigma\Delta$ Modulator Optimized Filter Structure	162
C.1	Bit-Stream Fractional Synthesis	164
C.2	Derived Front-End Fractional Synthesis	164
C.3	Front End Proposed Architecture Operation Principle	165
C.4	Front End Fractional-N $\Delta\Sigma$ Fractional Synthesizer Frequency Equivalent Model	167
C.5	Front End Fractional-N $\Delta\Sigma$ Fractional Synthesizer Frequency Topology . .	168
C.6	Front-Head $\Sigma\Delta$ Modulator Fractional Synthesis Output For Topology Shown In Figure C.3	169
C.7	Front-Head $\Sigma\Delta$ Modulator Fractional Synthesis Output For Topology Shown In Figure C.5	169
C.8	Front-Head $\Sigma\Delta$ Modulator Fractional Synthesis Outputs Near Main Signals For Both Topologies	170
C.9	Proposed Front-Head Fractional Synthesis (fig. C.5) With Synthesized 1-bit $\Sigma\Delta$ Modulator Phase Noise Output	171

C.10 Proposed Front-Head Fractional Synthesis (fig. C.5) With SSMF-II $\Sigma\Delta$	
Modulator Phase Noise Output	171
C.11 Front-Head Fractional Synthesis (fig. C.3) With SSMF-II $\Sigma\Delta$ Modulator	
Phase Noise Output	172

List of Tables

3.1	0.13 μ m 28 GHz VCO Components Parameters	36
3.2	65nm 31-40GHz VCO Components Parameters	43
4.1	65 nm Charge Pump Circuit Parameters	72
5.1	PLL Operating Parameters For Three Different Conditions	89
5.2	State Space Matrix Coefficients	102
5.3	State Space Matrix Coefficients for C Matrix	102
6.1	PLL Characteristics	119
6.2	CML DFF Values	124
6.3	CML AND-DFF Values	125
7.1	0.13 μ m 23 GHz VCO Phase Noise	144
A.1	NTF $\Delta\Sigma$ Poles and Zeros	156

Chapter 1

Introduction

1.1 Motivation

Today's communication systems are divided into various communication bands. These communication bands, defined in terms of their electromagnetic frequency range, are decomposed into many bins. Each of these bands are defined for specific applications. For instance, Global Positioning Satellite (GPS) band uses the L-band which ranges in frequency from 1 to 2 GHz. Another band, say 3 to 4 GHz is used for satellites communications, such as extracting weather information. This frequency range is also used for in-doors wireless communication systems such as Bluetooth, etc., making interference issues a challenge to solve. The other bands of interest are the C, X, K_u, K_a-bands that spans from 4 to 40 GHz. These band are mostly used for military telecommunication systems, radar systems or specific communications involved in scientific data collection.

All of the existing communication systems are based on Heterodyne or Homodyne receivers. These systems all require some form of mixing in which to translate the given information to higher frequency regions for long-distance communications, followed by more mixing to recover the original information. The operation of mixing is typical performed using a Phase Locked Loop, or more commonly referred to as a PLL.

Over the past few years, Field Programmable Gate Arrays (FPGA) have increased their computation power by embedding greater amounts of logic into a single package device, together with improvements to the software that compiles the hardware description of the logic circuits. In addition, advancements in die-stacking is also making great strides forward with the amount of logic available per package device. Logic speeds have also increased

as the physical dimensions of CMOS transistors are being reduced to 22 nm and smaller. FPGAs provide unheard of advantages for digital signal processing (DSP) on account of the amount of computation that can be performed in parallel with an FPGA.

Combining an FPGAs potential processing power together with a receiver/transceiver capable of tuning to each and every band from the L to K_a would provide a universal computational platform. Such a computational platform would improve the efficiency, cost and performance of collecting data for scientific, military or consumer applications. One of the many challenges faced with creating a device with such capability is the creation of a wide-band transceiver/receiver. Within this context, this thesis offers a study in the design of the frequency synthesizer required for the design of such wide-band transceiver/receiver circuit.

1.2 Thesis Contributions/Literature Review

The challenge in designing a frequency synthesizer for the L - K bands lies with the design of a wide-ranging low-phase noise PLL. In order to achieve this goal, a design methodology for such PLLs had to be developed. The method begins by first designing the phase-frequency detector/charge-pump, voltage-controlled oscillator and frequency divider circuit for realization in a specific CMOS technology. The method of choice mixes insight deemed from both a theoretical and simulation perspective. Next, the loop filter is derived based on the layout extracted behaviour of each component. Once complete, all components of the PLL are described using the high-level description language of Verilog-A available in the Cadence tool set over its full range of operating characteristics. Ideally, these components would be fabricated first and characterized afterward. The Verilog-A description of the PLL enables a fast and efficient simulation of the complete PLL in a closed-loop configuration. This latter steps allows further optimization of the overall design.

This thesis is based on the state-of-the-art in PLL technologies that is available to the public. In particular this thesis pays close attention to the voltage-controlled oscillator (VCO) phase noise optimization work of Hajimiri's in references [3] and [4]. In addition, the work in references [5], [6], [7] and [1] on phase-frequency detectors (PFD) and charge-pumps (CP) was extensively consulted. This in turn lead to the development of a new means to select the characteristics of the loop filter.

In addition to these components, a programmable fractional-N frequency divider circuit

[5] for use in a fractional-N PLL was designed. This design was performed from conception to final layout, where a chip was returned from fabrication and tested. This particular design involved the use of a $\Delta\Sigma$ modulator. An extensive investigation of the best topology for use as a component of a fractional-N frequency divider was undertaken. A $\Delta\Sigma$ modulator analysis was also carried out in this thesis. The theory of design for the $\Delta\Sigma$ modulator was taken from [8], [9], [10] as well as [11].

Finally, a mathematical model of the noise behaviour for the complete PLL system is derived using the work of [12]. A Verilog-A model of the PLL was constructed in order to simulate its closed-loop behaviour and tweak the design for optimal performance. Much of the Verilog-A modelling methods were extracted from [13].

1.3 Thesis Overview

Besides this introductory chapter, the next chapter, Chapter 2 will provide the background theory of a PLL together with some of its performance metrics. The general idea related to the design procedure will also be described.

Chapter 3 provides the design details for a wide-ranging VCO for our PLL application. Two separate designs of the VCO intended for fabrication in a 0.13 μm CMOS process from IBM and another in a 65 nm CMOS process from TSMC is provided. Simulation results at the schematic and layout level are provided and compared. In addition, a high-level description Verilog-A model of the VCO is provided. This model will be used later when the PLL is analyzed for its closed-loop behaviour.

In Chapter 4 the PFD and CP is designed for application in a 65 nm CMOS process from TSMC. Design issues and trade-offs are described. Simulation results at the schematic and layout level are provided and compared. A Verilog-A model for the PFD and CP is also provided. Close attention to the noise behaviour of these two elements is made.

In Chapter 5, the design of the loop filter is discussed, along with its short comings. This chapter identifies the need for an adaptive VCO or loop filter that automatically adjust for changes in the VCO nonlinear operating conditions. A Verilog-A model for the loop filter is also provided.

Chapter 6 provides a lengthy description of the design of the variable divider circuit for use in a fractional-N frequency synthesizer. Key to the development of a fractional-N divider is the concept of a $\Delta\Sigma$ modulator. This chapter provides a review of the principles

of $\Delta\Sigma$ modulation as it applies to a fractional-N divider network within a PLL circuit. It also studies which $\Delta\Sigma$ modulator topology is best suited for this application. Much of the analysis performed in this chapter is based on a Verilog-A description of the PLL, together with the fractional-N divider circuit in its feedback path. The chapter concludes with a discussion on the circuit implementation of the fractional-N synthesizer for fabrication in a 65 nm CMOS process from TSMC.

Chapter 7 provides some experimental results for the ICs that have successfully returned from the semiconductor foundry. While two chips were fabricated, one had errors due to incorrect use of the technology. The reasons for this will be provided in this chapter.

Finally, the thesis concludes in Chapter 8 with some suggestions for future work.

Chapter 2

Background Theory of Phase Locked Loops

In this chapter, the structure and components of the versatile Phase Locked Loop (PLL) will be described. A linear analysis of the behaviour of the PLL will be conducted and used to establish its input-output dynamic operation, as well as its output phase noise behaviour subject to various internal noise sources.

2.1 System Perspective of A Standard PLL

Many applications use a PLL, especially in the communication and integrated circuit field. PLLs are essential components in generating high frequencies from a low frequency source. Take, for instance, the situation where a high frequency signal is required to synchronize the operation of two separate ICs present on a printed circuit board (PCB). Running the high-frequency signal across the pads and die bonds of each IC experiences signal attenuation and large noise pick-up that usually renders the timing information incorrect. Furthermore, a higher frequency signal facilitates the emission of EM radiations and can potentially affect signals passing near by on the PCB. Instead, one signals each IC with a low frequency source and an on-chip PLL is used to increase its frequency while phase-locked with the reference signal. Of course, the improved signalling comes at the expense of greater circuits and power; nonetheless, it is often the only way forward. Other important applications of PLLs are found in analog-to-digital (ADC) applications, where the jitter on the sampling clocked is reduced to very small levels using the PLL as a time-domain filter circuit.

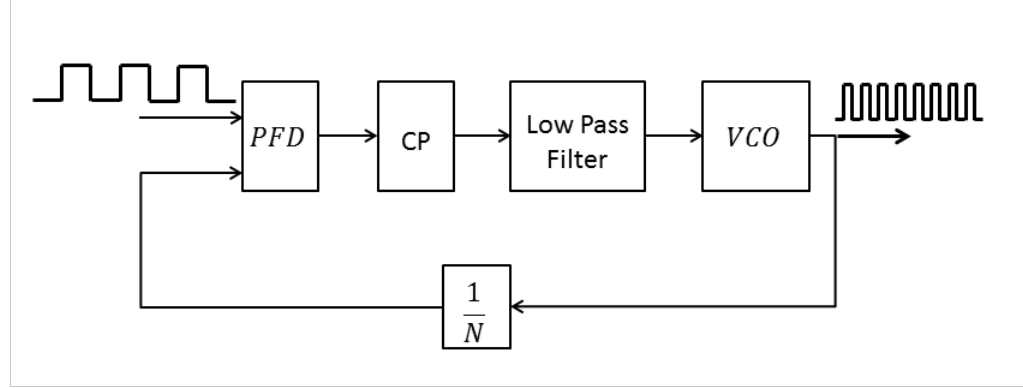


Fig. 2.1 Phase Locked Loop Architecture

2.2 PLL Components

A standard PLL architecture is shown in Fig. 2.1. As can be seen from this figure, the PLL consists of five primary components: a phase frequency detector, a charge pump, a low pass filter, a voltage controlled oscillator and a divider [14]. Each component will be described more fully below.

Phase Frequency Detector

The Phase Frequency Detector or PFD for short, usually has two outputs that transition high and low depending on which signal is triggered first. In Fig. 2.2, a flip-flop based PFD is shown along with its input-output timing characteristics. As the diagram illustrates, the PFD tracks the error between the rising edges of the reference clock and the feedback signal. Typically, the reference clock is fixed in frequency [15] and the output signal is forced to follow this signal in accordance to the divider ratio placed in its feedback path. For the specific case shown here, the divider ratio is one, so the reference and feedback signal have identical frequency. Depending on the phase relationship between the reference and the feedback signal, the UP and DN outputs are set differently. If the feedback signal lags the clock reference the UP signal will be set high for a duration determined by the next rising edge of the reference. In contrast, if the feedback signal leads the reference, the DN signal will be set high.

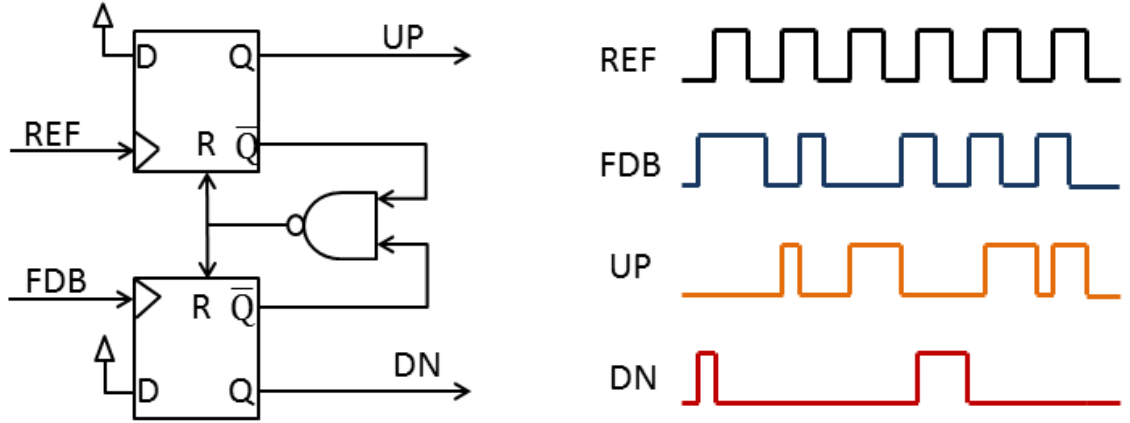


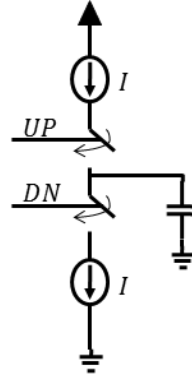
Fig. 2.2 Phase Frequency Dectector

Charge Pump

The output of the PFD detector is usually preceded by a charge pump [14]. In this case, this topology is referred as a Type II PLL [16]. This thesis will focus on a type II PLL, but the ideas presented in this thesis can be applied to type I as well. The charge pump, responds to the PFD signals by sourcing or sinking a fixed and equal amount of current [15] as seen in Fig. 2.3. The overall gain of the PFD and the charge pump is a function of the output voltage levels from the PFD and the current level used to bias the charge pump circuit. By varying the edge of the feedback signal with respect to the reference clock, the DC value of the PFD output changes in some linear manner. Ideally, linear operation extends over a phase range of 2π . Any phase difference greater than this results in a phase discontinuity.

Low Pass Filter

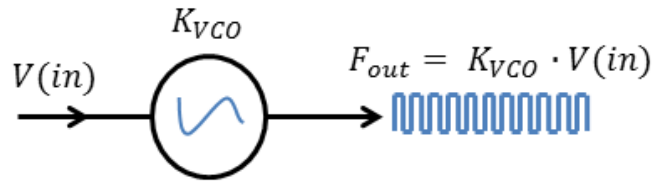
The next component in a PLL system is the low-pass loop filter. The filter has two primary functions: the first function is to extract the DC value from the output of the charge-pump circuit and the second is to produce a voltage signal to drive the following VCO circuitry. While the characteristics of the PFD and the CP, as well as the VCO, are generally established by the choice of circuit implementation, the loop-filter can be tuned to a specific type of transfer function whereby establishing the overall PLL dynamic behaviour, e.g., step

**Fig. 2.3** Ideal Charge Pump

response.

VCO

The output frequency of the voltage-controlled oscillator (VCO) is proportional to the input voltage signal; generally described with a proportionality constant K_{VCO} . The VCO can be seen as a frequency generator as shown in Figure 2.4 whose output frequency varies between f_{min} and f_{max} . Generally, K_{VCO} is determined by dividing the output frequency range by the maximum input voltage swing ΔV_c allowed, i.e., $K_{VCO} = \frac{f_{max} - f_{min}}{\Delta V_c}$ expressed in units of radians per second per volt.

**Fig. 2.4** Ideal Voltage Controlled Oscillator

Frequency Divider

The last component of the PLL is the divide-by- N frequency divider. It is used in the feedback path of the PLL. The operation of the divide-by- N frequency divider is illustrated in Figure 2.5. The purpose of the frequency divider is to set the frequency at the output of the PLL, denoted by f_{out} to be N times that of the reference frequency, f_{ref} . This is achieved by forcing the frequency of the feedback signal to that of the reference signal [15]. When the PLL is set with a unity-gain feedback configuration, i.e., $N = 1$, the output frequency will match that of the reference signal. In such situations, the PLL is used as a time-domain or phase-domain filter. Otherwise, when $N \neq 1$, the PLL is used to synthesize higher frequency signals. Seldom, if ever, does one configure the PLL with $N < 1$.

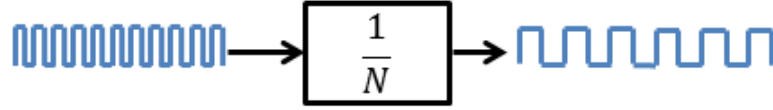


Fig. 2.5 Divider Operation

2.3 PLL Phase-Domain Model

The small-signal behaviour of the PLL can be analyzed using linear techniques in the phase domain [14], [15], [16]. In the phase domain, the PFD can be modeled as a differentiator, the charge pump as frequency independent gain stage with current gain I (that represents the sourcing/sinking current of the charge pump) and a loop filter with transfer function $F(s)$. As phase is the integral of frequency, the VCO is modeled as an integrator with a gain constant K_{VCO} . The divide-by- N frequency divider is simply modeled with a $1/N$ gain block. Figure 2.6 illustrates the equivalent circuit in the phase domain subject to an input reference phase-domain signal ϕ_{ref} .

2.3.1 PLL Phase-Domain Transfer Function

Common to all applications of a PLL, e.g., phase filter, clock recovery or frequency synthesis, the dynamic operation of the PLL is described by the input-output transfer function

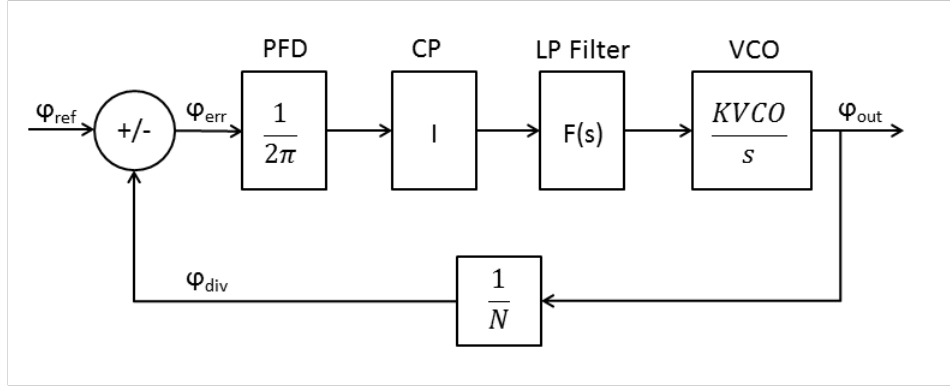


Fig. 2.6 PLL Phase-Domain Representation

given by:

$$\frac{\varphi_{out}}{\varphi_{ref}} = \frac{I \cdot \frac{KVCO}{s} \cdot F(s)}{1 + I \cdot \frac{KVCO}{Ns} \cdot F(s)} \quad (2.1)$$

Here φ_{ref} and φ_{out} represent the input reference and VCO output phase-domain signals. The transfer function from the input to the divider output φ_{div} is given by

$$\frac{\varphi_{div}}{\varphi_{ref}} = \frac{I \cdot \frac{KVCO}{Ns} \cdot F(s)}{1 + I \cdot \frac{KVCO}{Ns} \cdot F(s)} \quad (2.2)$$

The transfer functions captured by Eqn. 2.2 and Eqn. 2.1 are low pass in nature; hence the PLL acts as a low pass filter for signals appearing at the input of the PLL. In many frequency synthesis applications, the reference clock is of very high quality and the noise contribution from the VCO is dominant at the PLL output. The transfer function for a signal injected at the VCO output, denoted by signal $\varphi_{out.noise}$, to the PLL output is given by

$$\frac{\varphi_{out}}{\varphi_{VCO.noise}} = \frac{1}{1 + I \frac{KVCO}{Ns} F(s)} \quad (2.3)$$

This transfer function is characterized by a high-pass transfer function; complementary to the input-output transfer function.

2.4 Design For Specific PLL Transfer Function

Generally, the design of a PLL requires the selection of a transfer function, either from an input-to-output or a VCO-to-output perspective. For example, since a VCO injects phase noise into the PLL, based on the phase noise requirements at the PLL output, the noise transfer function must be selected so that this specification is met. Once the noise transfer function $\varphi_{out}/\varphi_{out.noise}$ is identified, the loop-filter transfer function $F(s)$ would be solved using Eqn. 2.3. While in principle this is quite straightforward, in practice, there is a realizability issue that arises.

To illustrate this, let us assume that the desired input-output PLL transfer function is to be a third-order low-pass Butterworth transfer function described in general terms as

$$H(s) = \frac{a}{s^3 + b_2s^2 + b_1s + b_0} \quad (2.4)$$

Substituting this back into Eqn. 2.2 and solving for $F(s)$ one finds

$$F(s) = \frac{Ns}{I K_{VCO}} \cdot \frac{H(s)}{1 - H(s)} = -\frac{N a}{I K_{VCO}} \cdot \frac{s}{s^3 + b_2s^2 + b_1s + b_0 - a} \quad (2.5)$$

As the rightmost term contains a zero at DC, rather than a pole, the requirement to be a Type II PLL cannot be met. Two methods can be used to alleviate this issue: multiplying the main desired transfer function by a lead-lag filter as done in [16] or by introducing a zero in the desired transfer function $H(s)$, named here as the zero-introducing method. This method appears in reference [17]. Both methods are further discussed below.

2.4.1 Lead-Lag Filter

A lead-lag filter transfer function containing a zero and a pole can be described as

$$C(s) = \frac{\tau_1s + 1}{\tau_2s + 1} \quad (2.6)$$

The multiplication of this lead-lag transfer function by the desired transfer function will increase the pole-zero count by one. While it enables a pole at DC to be introduced, it has the side effect of introducing frequency domain peaking in the PLL input-output transfer function. This peaking is proportional to the ratio between the lead-lag zero and the desired transfer function cutoff frequency. A high ratio yields more peaking, while a small ratio

results in a transfer function behavior that resembles more like the desired one. The ratio is usually constrained between 1/10 and 1/3 [16]. A higher ratio would yield a loop filter whose transfer function approaches an ideal low pass filter, while a lower ratio usually results in a loop filter with poles and zero very close and hence with a transfer function that would be more complicated to implement. On the other hand, the pole location has to be found explicitly. The method outline here, consists of multiplying the desired transfer function with this lead-lag transfer function. The multiplication of the desired Butterworth transfer function with this lead-lag yields a new modified desired transfer function as follows,

$$H_{new}(s) = \frac{a(\tau_1 s + 1)}{s^3(\tau_2 s + 1) + b_2 s^2(\tau_2 s + 1) + b_1 s(\tau_2 s + 1) + b_0(\tau_2 s + 1)} \quad (2.7)$$

Further simplification leads to

$$H_{new}(s) = \frac{a\tau_1 s + a}{\tau_2 s^4 + s^3(1 + \tau_2 \cdot b_2) + s^2(b_2 + \tau_2 \cdot b_1) + s(b_1 + \tau_2 \cdot b_0) + b_0} \quad (2.8)$$

As before, substituting Eqn. 2.8 back into Eqn. 2.2 with $H(s)$ replaced by $H_{new}(s)$ and solving for $F(s)$ one finds

$$\begin{aligned} F(s) &= \frac{Ns}{I K_{VCO}} \cdot \frac{H_{new}(s)}{1 - H_{new}(s)} \\ &= \frac{Ns}{I K_{VCO}} \cdot \frac{a\tau_1 s + a}{\tau_2 s^4 + s^3(1 + \tau_2 \cdot b_2) + s^2(b_2 + \tau_2 \cdot b_1) + s(b_1 + \tau_2 \cdot b_0) + b_0 - a\tau_1 s - a} \end{aligned} \quad (2.9)$$

If the following condition is met,

$$\tau_2 = (a \cdot \tau_1 - b_1)/b_0 \quad (2.10)$$

then Eqn. 2.9 reduces to

$$F(s) = \frac{N}{I K_{VCO}} \cdot \frac{a\tau_1 s + a}{\tau_2 s^3 + s^2(1 + \tau_2 \cdot b_2) + s(b_2 + \tau_2 \cdot b_1)} \quad (2.11)$$

As is clearly evident, $F(s)$ now has a pole at DC, enabling a type-II PLL to be realized. This method can be applied to any type of filter provided a and b_0 are equal or made equal and offers flexibility in placing, up to a limit, the zero of the overall closed loop PLL transfer function.

2.4.2 Zero-Introducing Method

The main disadvantage with the previous method is it increases the order of the loop filter. If power or area is considered, the filter should be kept to a minimum order. In the zero-introducing method, instead of pre-multiplying the main desired closed loop transfer function, the numerator of $H(s)$ is replaced with the values from its denominator having the power of s^1 and s^0 , i.e.,

$$H_{new}(s) = \frac{b_1s + b_0}{s^3 + b_2s^2 + b_1s + b_0} \quad (2.12)$$

Substituting Eqn. 2.12 back into Eqn. 2.2 with $H(s)$ replaced by $H_{new}(s)$ and solving for $F(s)$ one finds

$$F(s) = \frac{Ns}{I K_{VCO}} \frac{H_{new}(s)}{1 - H_{new}(s)} = -\frac{Ns}{I K_{VCO}} \frac{b_1s + b_0}{s^3 + b_2s^2 + b_1s + b_0 - b_1s - b_0} \quad (2.13)$$

which reduces further to

$$F(s) = -\frac{N}{I K_{VCO}} \frac{b_1s + b_0}{s^2 + b_2s^1} \quad (2.14)$$

Clearly $F(s)$ has a pole at DC and satisfies the type-II PLL condition. As the zero of the closed-loop PLL transfer function will depend on the initial pole positions, it is difficult to position *a priori* the zero for optimum frequency response behaviour. Figure 2.7 compares the result of the two synthesis methods of the PLL transfer function using a third-order Butterworth filter as its initial behavior. Here the ratio for the lead-lag zero was set to be 1/4 of its cutoff frequency. As is evident from this plot, the lead-lag method yields an overall closed loop transfer function quite different than the desired one. Whereas the zero-introducing method gives a response much closer to the desired transfer function.

Figure 2.8 shows the resulting filters frequency responses for both methods. The zero-

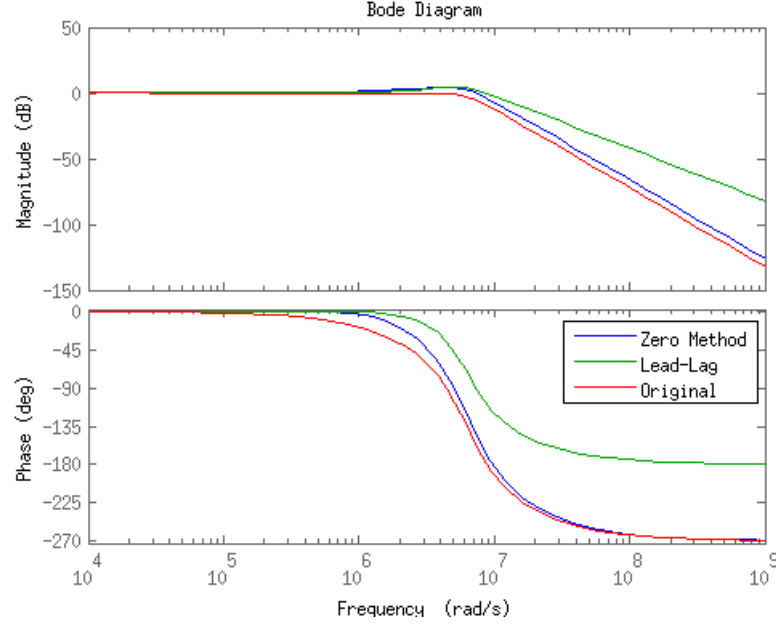


Fig. 2.7 Comparing the Frequency and Phase Response of Overall Closed Loop PLL Response Using The Lead-Lag and Zero-Introducing Methods

introducing method yields a filter with lower order, however, as can be seen, the pole and the zero are very tight together. On the other hand, the lead-lag method yields a better shaped filter, with poles farther spaced from the zeros.

2.5 PLL Requirements/Constraints

Any system using a PLL will impose performance and physical constraints on the PLL. The major types of requirements involve silicon area, power dissipation, frequency range, overall PLL closed-loop step response, jitter and phase noise, as well as output frequency spurs. These types of issues are discussed below:

Area

The integrated IC area required by the PLL is an important economical metric. While the cost of silicon is a fixed quantity, the area available for a given design affects the overall design choices. For example, a high frequency modulus divider requires more silicon

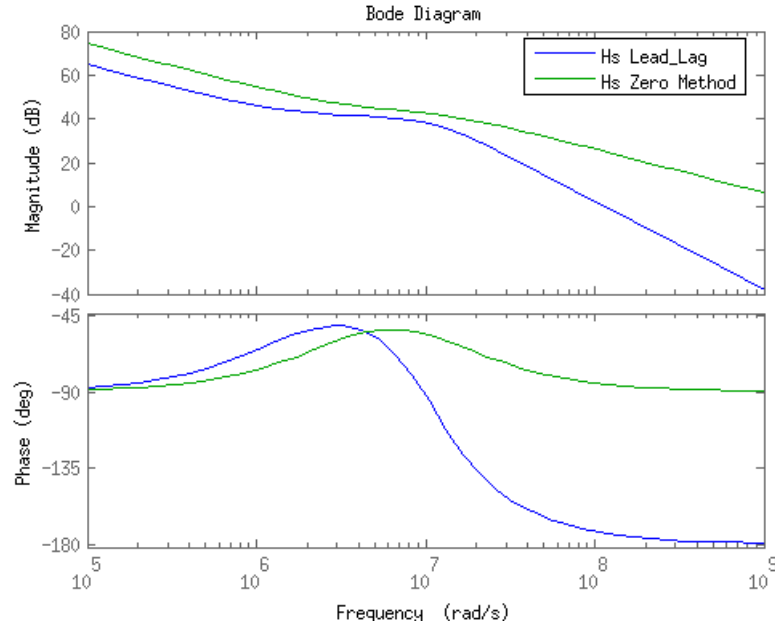


Fig. 2.8 Resulting Filters Transfer Function Frequency and Phase Responses Using the Lead-Lag and Zero-Introducing Methods

area than a fixed counter. If frequency synthesis is required whereby a modulus divider is necessary, then the silicon area can be reduced by selecting a higher input reference frequency as this requires a lower divider value.

Power

All PLL components dissipate power; some to a lesser extent than others. For instance, a high frequency PLL will require more power to operate than a low frequency one. As the loads internal to an IC are largely capacitive in nature (as opposed to resistive), higher frequency designs require larger current drive, hence, they require more power. It is therefore crucial at the start of any high frequency PLL application to minimize the power consumption of the critical high-frequency components, such as the VCO and modulus divider, all the while considering other performance metrics of the PLL like frequency range and phase noise, to name just a few. The choice of implementation of the loop filter can also have important impact on the power consumption of the PLL. While a passive loop-filter will help to keep power consumption to a minimum, it comes at the expense of reduce flexibility in

tuning the PLL dynamic response. While an active loop filter realization provides greater design choices at the expense of greater static power and, potentially, higher system noise.

Frequency Range

An important consideration of any PLL is the range in which it is capable of locking onto a reference signal. While incorporating a universal wide-ranging VCO into a design makes the design process simpler, it comes at the expense of a larger sensitivity or gain factor K_{VCO} . This, in turn, makes the VCO more sensitive to noise at its input control port and generally leads to reduced PLL phase noise performance. The characteristics of a VCO should be customized to the PLL application for optimum performance.

Step Response

When a PLL is expected to react to changes in the reference frequency, it must do so quickly. One metric used to quantify the PLL step response is its lock time. The lock time performance metric relates to the bandwidth of the overall closed loop PLL transfer function. The wider the bandwidth, the faster the lock time. However, the wider the system bandwidth the larger the noise that passes from the reference signal port to the PLL output. Finding the right compromise is an important PLL design consideration.

Phase Noise PSD

One of the critical specifications of a PLL is its output phase noise behavior. Phase noise is expressed as

$$\mathcal{L}\{\Delta\omega\} = 10 \cdot \log_{10} \frac{\text{noise power in a } 1 - \text{Hz bandwidth at frequency } \omega_c + \Delta\omega}{\text{carrier power at } \omega_c} \quad (2.15)$$

It is a ratio measure of the power of these side bands, considered as noise, to the main tone. Most often, the logarithmic scale is used, and the output is thus given in dBc/Hz. Where dBc implies with respect to the carrier. The latter equation is sometimes written in terms of $S_\phi(\Delta\omega)$

$$\mathcal{L}\{\Delta\omega\} = 10 \cdot \log_{10} \frac{S_\phi(\Delta\omega)}{2} \quad (2.16)$$

where $S_\phi(\Delta\omega)$ relates to the output voltage power spectrum, $S_v(\Delta\omega)$ as in

$$S_\phi(\Delta\omega) = \frac{S_v(\Delta\omega)}{\text{carrier power at } \omega_c} \quad (2.17)$$

The output phase noise spectrum of an oscillator is usually characterized by three different regions such as shown in Figure 7.5. In an oscillator, thermal noise internal to the oscillator gets up-converted by the integration process. This creates a noise in the $1/\omega^2$ region with a decay of 20 dBc/Hz. The $1/\omega^3$ region is the flicker noise superimposed on the up-converted thermal noise; this region has a decay of 30 dBc/Hz. The point of intersection between these two regions is referred to as the flicker-thermal noise corner frequency denoted by $\Delta\omega_{1/f^3}$. The frequency-independent noise region, $1/\omega^0$ is generally thermal noise that couples into the output signal path of the VCO. The point of intersection of the up-converted thermal noise with the thermal noise is ref denoted by $\Delta\omega_{1/f}$.

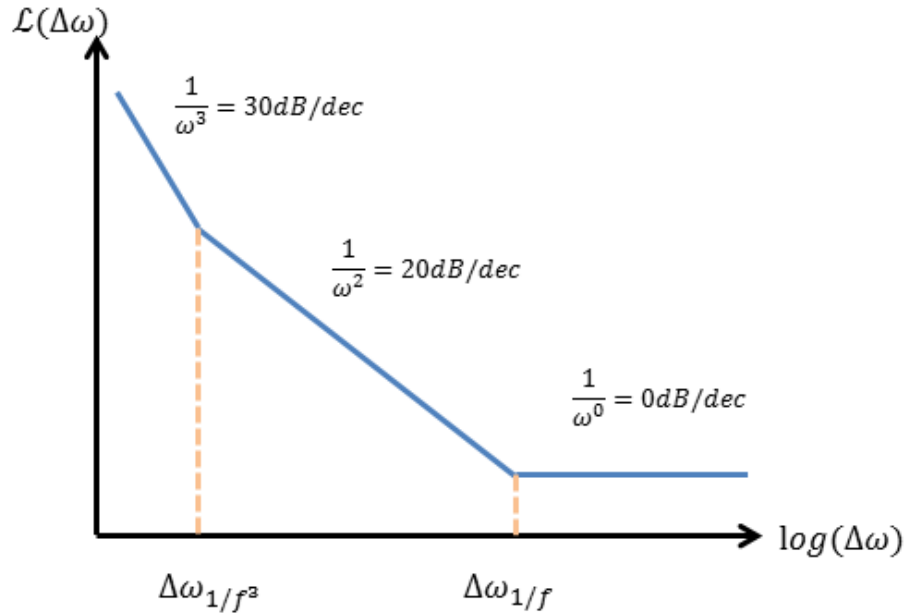


Fig. 2.9 Phase Noise Spectrum of an Oscillator

Phase noise contributes to variations in the period T of an oscillation signal produced by a PLL driven by a fixed-frequency reference signal. Such a variation is often referred to as periodic jitter and can be observed directly in the time-domain using an oscilloscope

of time-interval analyzer (TIA). Phase noise, on the other hand, is a frequency-domain quantity generally observed using a power spectrum analyzer. Assuming the PSD of a signal appearing at the output of a PLL is described by $S_\phi(f)$, then the total power or variance of the jitter at its output can be quantified as [18]

$$\sigma_A^2 = \frac{T^2}{(2\pi)^3} \int_{-\pi/T}^{\pi/T} S_\phi(\omega) d\omega. \quad (\text{second}^2) \quad (2.18)$$

Output Spurs

The output PSD of a PLL will often will contain spurs. These spurs are created by extraneous periodic noise sources generated anywhere in the system. Most often, these spurs occur when the divide-by-N divider is switched at some periodic frequency. These spurs could also come from nonlinearities associated with the PFD, which will be describe more fully in the next chapter.

2.6 PLL Design Procedure with the VCO as the Dominant Noise Source

Given a set of requirements, the designer is faced with numerous questions at the start of the design of the PLL. Some might be:

1. What is the order in which the PLL components should to be designed?
2. How much phase noise can each component contribute while meeting the noise requirements of the overall PLL?
3. What are the architectures choices available for a given PLL design requirement?

Such questions are captured in the illustration shown in Figure 2.10. This diagram is meant to capture the importance of one's choices in the design of the PLL for a given application. The impact of incorrect design decisions can lead to problems later in the design process such as missing performance specifications. Components might have to be re-evaluated or re-optimized at the end of the PLL design. Furthermore, the overall PLL topology might need to be changed, or components re-adapted and the filter re-evaluated

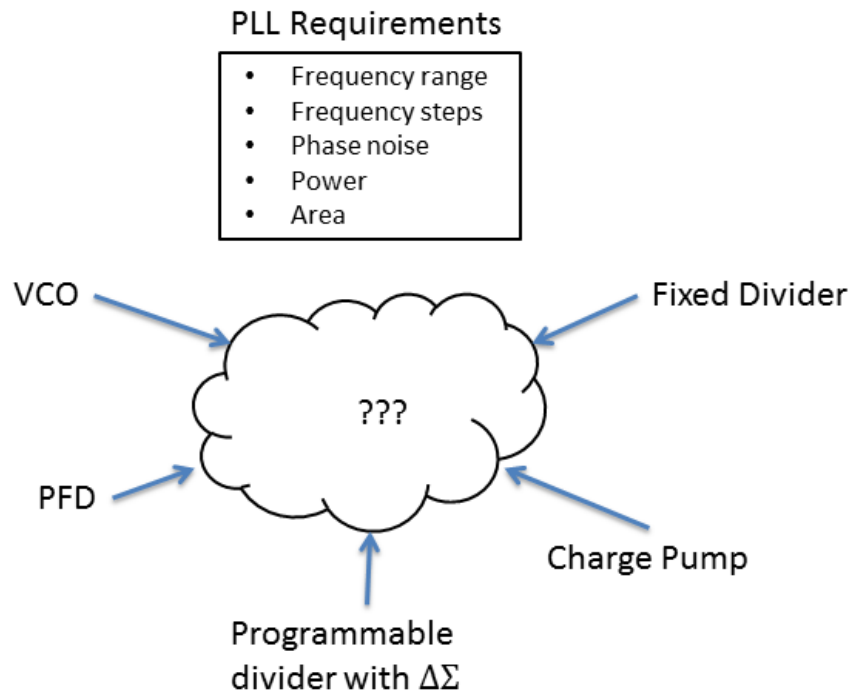


Fig. 2.10 Starting Point For Design Of PLL Can be A Mess

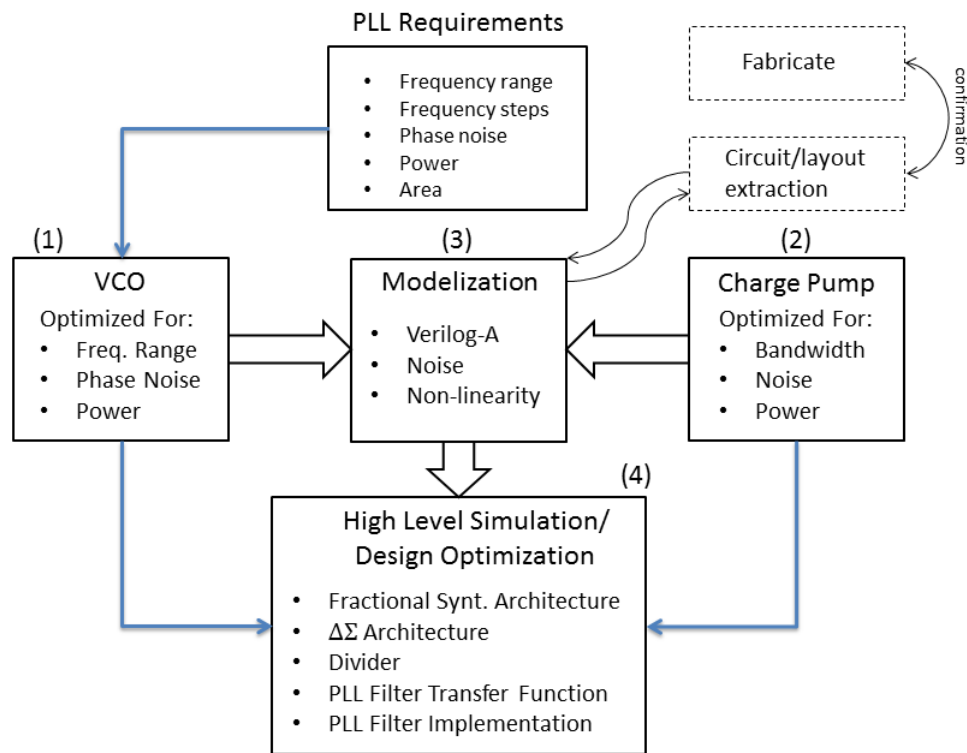
because of bad decisions or assumptions. All of these issues could be avoided if the proper technique of PLL design is employed.

The intention of this thesis is to propose design strategy for PLLs that is optimized over a set of performance criteria. Figure 2.11 outlines the main steps of the design method. Given a set of requirements for the PLL, the first component to be designed is the VCO. The VCO should be designed based on its frequency range of operation, phase noise requirement and power consumption. In some cases, multiple VCOs will be required to cover the full frequency range of operation; a choice that will have a significant impact on the power budget. The second component to be designed should be the PFD/CP combination. The PFD architecture should be carefully chosen. Many articles have appeared in the literature that cite the advantages and disadvantages of different topologies for clock recovery circuits and frequency synthesis [19],[20]. The optimization of the PFD/CP for a clock recovery application should focus on the bandwidth (if the reference frequency is set high), noise, power and, as well, the silicon footprint. In the case of a frequency synthesis application,

the reference frequency will be selected based on the design requirements associated with the variable divider network, generally realized with a $\Delta\Sigma$ modulator. The reference signal is again assumed to be generated from an external low-phase noise reference oscillator. Finally, the loop filter is designed using the performance data derived from the VCO, PFD/CP and divider components. However, unlike the previous components, the loop filter is designed using the data extracted from each component and collected into a high-level model description using a language such as Verilog-A. The model should capture the operational aspects of the design, as well any noise it produces. Furthermore, any non-linearity or sensitivity to power supply fluctuations should be captured as well. By doing this, a fast and efficient simulation of the PLL can be conducted and the loop filter with the best characteristics can be selected for the application at hand. Once the transfer function for the loop filter is identified, the model can be replaced with an actual circuit implementation and simulated. In this way, the overall behaviour of the PLL can be investigated in a reasonable amount of time.

2.7 Summary

In this chapter, the structure and components of the versatile PLL was described. Through a linear analysis of the PLL, the input-output behaviour of the PLL was described. This same analysis can also be used to predict the output phase noise behaviour when the PLL is subjected to internal noise sources. Various performance issues were described such as frequency range of operation, step response, phase noise and power requirements. Finally, the chapter ended with a description of a proposed design methodology for PLLs using a high-level description language such as Verilog-A. The following chapters will go into greater detail for the design of each component of the PLL.

**Fig. 2.11** Proposed Design Method For a PLL

Chapter 3

Voltage Controlled Oscillator For Gigahertz Operation

In the last chapter, the PLL design strategy was introduced together with a brief description of the main components of the PLL. In this chapter, the design of a wide ranging VCO for implementation in a 130 nm and 65 nm CMOS process from IBM will be described. Towards the end of this chapter a Verilog-A model of the VCO implemented in 65 nm CMOS will be given.

3.1 VCO Architectures

In this thesis, the PLL will be constructed from an analog VCO as opposed to a digital one. As such, an oscillation is created by placing the poles of the circuit on the $j\omega$ axis. Moving the poles along the $j\omega$ axis will change the oscillation frequency. A positive feedback system, such as an operational amplifier with its output connect to its positive input, could create those poles. The poles are aligned along the $j\omega$ axis and the distance with respect to the origin determines the frequency of oscillation. Another method, employed uses odd pairs of inverters is to create this positive feedback phenomenon. Such VCOs are termed ring oscillators or sometimes delay-based oscillators. A current-starved ring oscillator [21] is shown in Figure 3.1(a). The frequency of oscillation is altered by controlling the current through the inverters, which in turn changes the delay of each inverter. This results in a change in the period of the oscillation. Such oscillators are most commonly used in wide-band frequency synthesis applications, since they have a small silicon foot print, consume

little power and have a very wide frequency range. However, this type of oscillator is prone to large phase noise [22]. Another type of oscillator that is beginning to appear in the open literature is the distributed or mm-wave oscillator shown in Figure 3.1(b). The operation of this oscillator relies on the phase velocity of the propagation of a wave around the loop formed by the distributed elements [23]. Its frequency is tuned by changing the current or the amplification gain of each segment of the overall transmission line. The advantages of this oscillator is its very high frequency generation and good phase noise properties [24]. The main disadvantage of this circuit is its limited tuning range and its strong dependency on device parasitics.

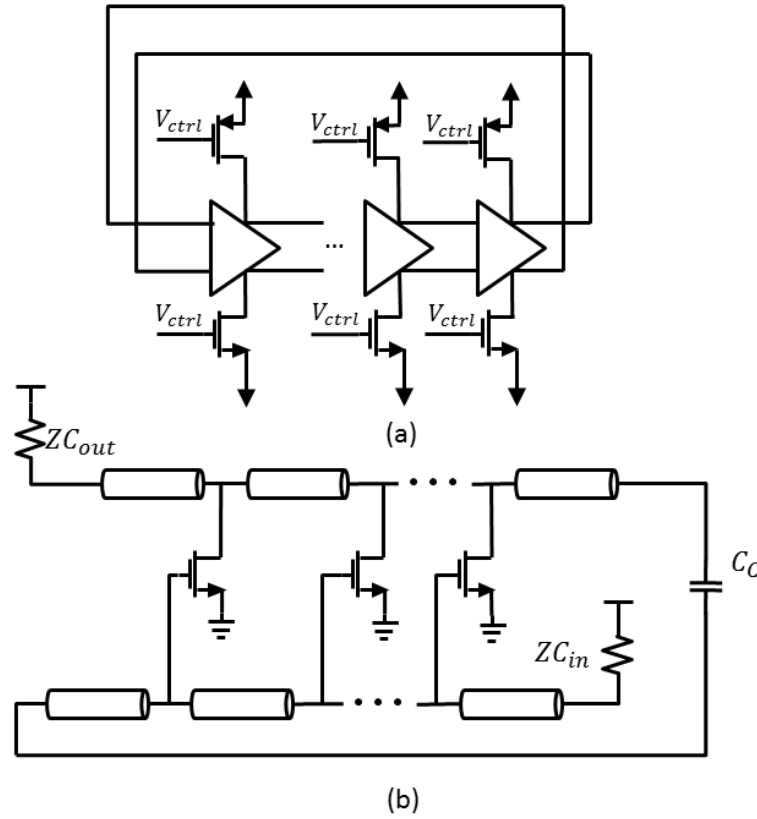


Fig. 3.1 Two Different VCO Architectures: (a) Current-Starved Ring Oscillator (b) Distributed Oscillator

The most commonly used IC oscillator for low phase noise application is the LC tank oscillator shown in Figure 3.2. The LC-tank oscillator consists of a parallel combination of an inductor and a capacitor driven by a pair of complementary cross-coupled NMOS

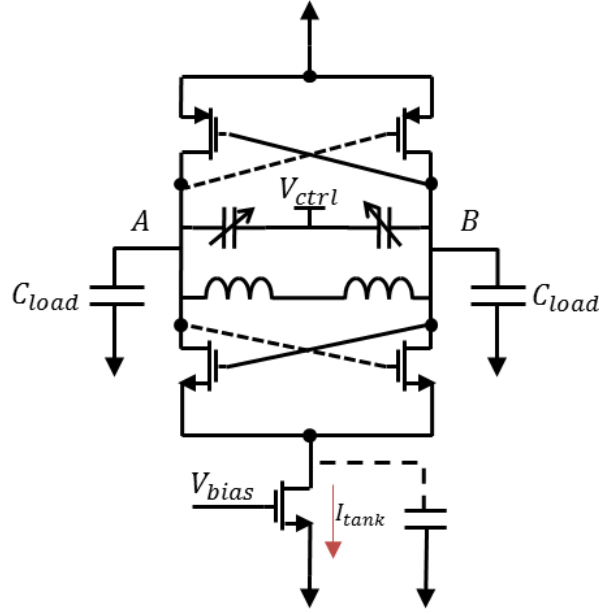


Fig. 3.2 LC-Tank VCO

and PMOS transistors (essentially a cascade of two CMOS logic inverters). The cascade of two inverters provides a positive feedback loop with a gain greater than unity that ensures oscillation and the LC-tank forces the frequency of oscillation ω_0 to be

$$\omega_0 = \frac{1}{\sqrt{LC}} \quad (3.1)$$

The two series capacitors are constructed from varactors, which are nothing more than NMOS or PMOS transistors with their source and drain terminals shorted together. The capacitance of each varactor is set by the bias voltage V_{ctrl} applied to the common node of the series capacitor connection.

3.2 Oscillator Phase Noise Theory

In most integer frequency synthesis systems, most of the phase noise is caused by the VCO. One general theory that predicts phase noise in oscillators is the Leeson-Culter formula [25]. The following formula by Leeson was derived to predict the phase noise of the LC-Tank

VCO [22],

$$L\{\Delta\omega\} = 10 \cdot \log \left\{ \frac{2FkT}{P_s} \cdot \left[1 + \left(\frac{\omega_0}{2Q_L\Delta\omega} \right)^2 \right] \cdot \left(1 + \frac{\Delta\omega_{1/f^3}}{|\Delta\omega|} \right) \right\} \quad (3.2)$$

In this formula, the parameter F represents the oscillator noise figure, or the amount of noise the oscillator adds into the system, P_s represents the signal power at the output of the VCO and the term $\Delta\omega_{1/f^3}$ represents the corner frequency of the $1/\omega^3$ and $1/\omega^2$ regions of the phase noise. The term $\Delta\omega_{1/f^3}$ must be extracted from empirical data. Finally, the term Q_L represents the LC-tank Q factor, given by

$$Q_L = \frac{\omega_0}{2\text{Bandwidth}} \quad (3.3)$$

The Q_L factor can also be written in terms of the LC-tank components according to

$$Q_L = \frac{\omega_0 L}{R_{tank}} \quad (3.4)$$

where the additional term R_{tank} represents the effective parallel/series resistance of the resonant LC tank circuit. Substituting Eqn. 3.4 into Eqn. 3.2 leads to

$$\mathcal{L}\{\Delta\omega\} = 10 \cdot \log \left\{ \frac{2FkT}{P_s} \cdot \left[1 + \left(\frac{R_{tank}}{2L\Delta\omega} \right)^2 \right] \cdot \left(1 + \frac{\Delta\omega_{1/f^3}}{|\Delta\omega|} \right) \right\} \quad (3.5)$$

The Leeson-Culter phase noise formula of Eqn. 3.6 states that in order to reduce the output phase noise of an oscillator, the tank load resistance R_{tank} needs to be decreased. The phase noise can also be reduced if the oscillation output power (i.e., P_s) is increased. Since the output power P_s is given as $I_{tank}^2 \cdot R_{tank}$, where I_{tank} is the tank bias current, the overall phase noise is re-written as:

$$\mathcal{L}\{\Delta\omega\} = 10 \cdot \log \left\{ \frac{2FkT}{I_{tank}^2} \cdot \left[\frac{1}{R_{tank}} + \frac{R_{tank}}{2L^2(\Delta\omega)^2} \right] \cdot \left(1 + \frac{\Delta\omega_{1/f^3}}{|\Delta\omega|} \right) \right\} \quad (3.6)$$

The term $\Delta\omega_{1/f^3}$ has to be obtained from data simulation or experimentally. This term depends on factors such as R_{tank} etc..., therefore, this formula does not provide any valuable insight in the design of low phase noise VCOs.

The issues with this simple model of the Leeson-Cutler formula for phase noise pre-

diction in LC-Tank VCO is that it does not take into account the periodically-varying or cyclostationary nature of noise inside the oscillator. As well, the model depends on parameters that must be extracted from measured data making it difficult to predict the noise behavior without a working prototype. In reference [29], Hajirmii proposes a new phase noise formula that takes into account these two issues. Specifically, the LC-tank VCO phase noise is stated as follows

$$L\{\Delta\omega\} = 10 \cdot \log_{10} \left(\Lambda(g_{m,n}, g_{m,p}, g_{m,tail}, \dots) \cdot \frac{S_{n,i,o}(\Delta\omega)}{2 \cdot \Delta\omega^2} \right) \quad (3.7)$$

where the function Λ represents the overall root mean square value of the so-called normalized Impulse Sensitivity Function (ISF) of the phase noise theory presented by Hajirmii in ref [22]. There is no simple linear form describing this function, however, it is a function of the transistor transconductances that make up the VCO. $S_{n,i,o}(\Delta\omega)$ corresponds to the PSD of the input current noise source at the corresponding VCO output node at a frequency offset from the carrier frequency ω_c . This simple equation expresses the dependence of the VCO phase noise on the individual components and the equivalent output current noise source. Hajirmii goes further and derives the following condition for minimum phase noise:

$$g_{m,n} = g_{m,p} \quad (3.8)$$

In essence, this requires the VCO to be designed for symmetrical operation and layout. This will be explored and defined more fully in the next subsection.

3.2.1 VCO Equivalent Model

A simple LC-tank VCO along with its simplified model is illustrated in Figure 3.3. Here the LC-tank circuit is modeled with two ideal reactive elements, L and C , and a parallel conductance g_{tank} . This conductance is used to represent the real losses associated with the actual LC-tank circuit. In addition, an amplifier with gain G is represented by a negative conductance of magnitude $g_{o,active}$ and a parasitic shunt capacitance C_p . A negative conductance implies that for a given voltage applied across it, a current is sourced to the surrounding circuit instead of being sunk into it. If the parallel combination of the two conductances goes to zero, the remaining circuit components consist of a parallel L and $C + C_p$ combination. For sake of any future discussion the effective C and L of the LC-tank

VCO will be denoted as C_{tank} and L_{tank} , respectively. The poles of the remaining circuit will therefore be located directly on the $j\omega$ axis, resulting in frequency of oscillation described by

$$\omega_o = \frac{1}{\sqrt{L_{tank}C_{tank}}} = \frac{1}{\sqrt{L(C + C_p)}} \quad (3.9)$$

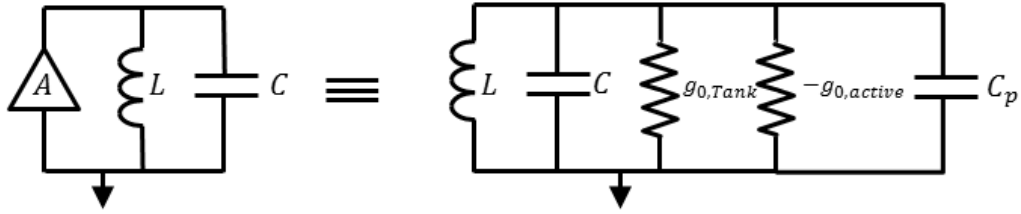


Fig. 3.3 A Simplified LC-Tank VCO and Its Equivalent Passive Representation

Returning to the LC-tank VCO circuit of Figure 3.2, rather than lump all the active elements into a single negative shunt conductance in a single step, one can model [3] the contribution of various active portions of the circuit as separate entities using the symmetric model shown in Figure 3.4. Here the cross-coupled NMOS transistors can be modeled as a series combination of two parallel conductances; one conductance represents the negative conductance associated with the transistor transconductance $-g_{m,n}$ and the other is related to the output conductance of the transistor $g_{o,n}$. A similar approach can be used to model the PMOS transistors, leading to another series combination of two parallel conductances, $-g_{m,p}$ and $g_{o,p}$. In Figure 3.4 these two sets of conductances are combined in some small way. Also included with the NMOS/PMOS transistor configuration is the effective capacitance C_{NMOS} and C_{PMOS} seen between the output terminals of the NMOS or PMOS transistor configuration. More on how this is extracted in a moment. The nonidealities associated with the capacitor varactor and the inductor are also included in this circuit model. More specifically, the inductor L is modelled with a series/parallel combination of a series resistance R_s and a shunt resistance of R_p . In addition, a shunt capacitor C_L is also included. The capacitor varactor C_v includes a series resistance R_v . Finally, the load is assumed to be a single capacitance of C_{load} .

In terms of the simplified model of Figure 3.3, we are now in a position to compute the

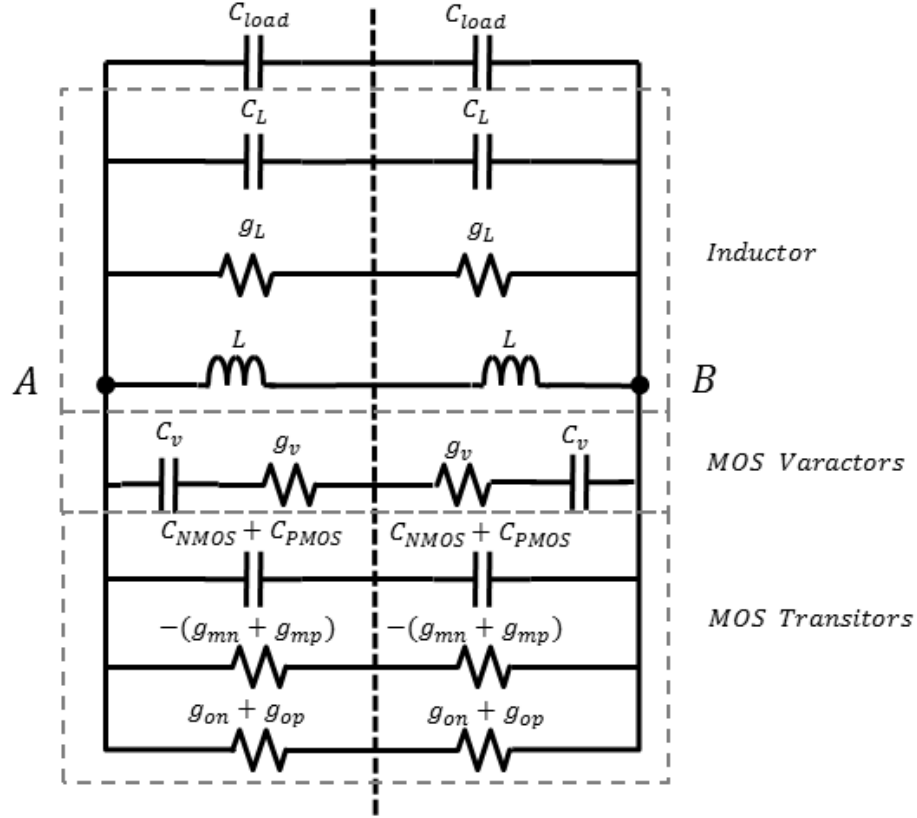


Fig. 3.4 A Symmetric LC-Tank Equivalent Model

effective terminal capacitance of the VCO as

$$C_{tank} = 2(C_{NMOS} + C_{PMOS} + C_L + C_v + C_{load}) \quad (3.10)$$

This equation includes all parasitic capacitances introduced by the transistors, inductor and the varactors, as well as the load capacitance. The transistor effective terminal capacitances C_{NMOS} and C_{PMOS} is found from the small-signal equivalent circuit shown in 3.5. Specifically, the terminal capacitance is found to be

$$C_{NMOS} = C_{gs} + 4 \cdot C_{gd} + C_{dB} \quad (3.11)$$

In a similar manner, for the PMOS transistor arrangement, we find

$$C_{PMOS} = C_{gs} + 4 \cdot C_{gd} + C_{dB} \quad (3.12)$$

In a similar fashion, the output conductance of the tank due to the lossiness of the reactive elements is given by

$$g_{o,tank} = \frac{g_{o,n} + g_{o,p} + g_C + g_L}{2} \quad (3.13)$$

Likewise, the active output conductance is given by

$$g_{o,active} = \frac{g_{m,n} + g_{m,p}}{2} \quad (3.14)$$

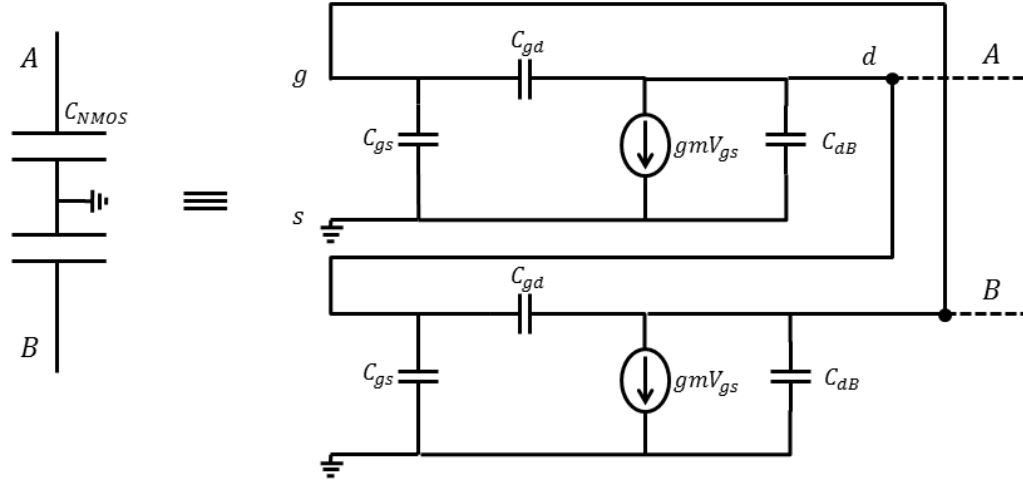


Fig. 3.5 A Small-Signal Equivalent Model Of A Pair Of Cross-Coupled NMOS Transistors

3.2.2 VCO Design Equations

When designing a VCO it must be made to satisfy certain constraints such as power consumption, output voltage signal amplitude, upper and lower frequency bounds, as well as phase noise and of course silicon area footprint. In this subsection, the design procedure proposed in ref [3] and [22] will be described.

Output Amplitude and Power

The output voltage amplitude for the VCO of Figure 3.2 is given simply as

$$V_{o,amp} = \frac{I_{tank}}{g_{o,tank}} \quad (3.15)$$

where I_{tank} is the current used to bias the LC-tank circuit and is the only parameter under direct design control, as $g_{o,tank}$ is either an intrinsic transistor parameter or associated with a parasitic element of a passive component. Generally, the output voltage will have to satisfy a minimum output voltage requirement, denoted by $V_{o,amp,min}$. In essence, this requirement sets the minimum bias current required by the VCO.

In [4], Hajimiri defines two regions of operation for this type of oscillator: the inductance limited and the voltage limited regimes, also called the current and voltage limited region in [4]. The current limited regime is the region in which the tail current in an LC-tank VCO (see figure 3.2) is purely sinusoidal. In this region, the voltage amplitude is below the supply limits and is described by 3.15. Whereas, in the voltage limited region, the tail current shows multiple harmonics and the output voltage swing is nearly rail-to-rail. In this regime, all transistors operate within their triode regions. Generally, this region of operation should be avoided.

Min/Max Frequency

The VCO must be designed to operate over a range of frequencies bounded between $\omega_{o,min}$ and $\omega_{o,max}$. Tuning is generally provided by varying the effective tank capacitor value over some range bounded between $C_{tank,min}$ and $C_{tank,max}$. Assuming the mid-capacitance value is given by $C_{tank,mid} = \frac{C_{tank,min} + C_{tank,max}}{2}$ resulting in a mid-frequency oscillation of $\omega_{o,mid} = \frac{\omega_{o,min} + \omega_{o,max}}{2}$, the inductor value $L = L_{tank}$ is selected according to

$$L = \frac{1}{\omega_{o,mid}^2 \cdot C_{tank,mid}} \quad (3.16)$$

Active Gain

In order to ensure VCO oscillation the following condition must be met

$$g_{o,active} \geq g_{o,tank} \quad (3.17)$$

Generally, $g_{o,active} = 3 \cdot g_{o,tank}$ to ensure proper operation at start-up [3].

Phase Noise

Phase noise at the output of the VCO is an important consideration for a VCO of any type. For an LC-tank VCO, the noise sources are generally thermal noises, apart from the noise generated by the tail current. The thermal noise current PSD for either the NMOS and PMOS transistors is given by

$$S_{n,i,m}(f) = 2kT\gamma(g_{o,n} + g_{o,p}) \quad (3.18)$$

where γ is 2.5 for short channel transistors and $g_{o,n}$ and $g_{o,p}$ are the NMOS and PMOS output conductances [3]. Similarly, the thermal noise current PSD of the tail current transistor is given by

$$S_{n,i,tail}(f) = 2kT\gamma g_{o,tail} \quad (3.19)$$

The other thermal noise current PSD comes from the parasitic conductance of the inductor g_L and is given by

$$S_{n,i,ind}(f) = 2kT\gamma g_L \quad (3.20)$$

Likewise, the varactors also inject thermal noise current into the circuit due to their parasitic series conductance g_v . The PSD for this thermal noise current is

$$S_{n,i,var}(f) = 2kT\gamma g_v \quad (3.21)$$

As the varactor conductance changes with biasing, the largest conductance value should be selected for this analysis to establish an upper noise limit.

From all these noise sources, the transistors thermal noise dominates. However, flick noise generated by the tail current transistor will influence the output phase noise. While the noise from the tail current transistor does not directly contribute to the VCO output, it is influenced by the signal path gain from the tail current to the VCO. The flicker noise of a transistor is characterized by

$$S_{1/f,i}(f) = \frac{KF \cdot I_D^{AF}}{(C_{OX})^2 LW} \cdot \frac{1}{f} \quad (3.22)$$

where AF is the flicker noise exponent ranging from 0.5 to 2 and KF is the flicker noise coefficient typically given as $10^{-28} \text{A}^{2-AF} (\text{F/m})^2$ [26]. These parameters can be extracted from circuit as described in the [26]. The current level I_D corresponds to the DC transistor drain current. In this specific case, I_D is equal to the tail current i_{tail} . Hence, from this equation, the tail current transistor should have a large W and L in order to decrease the impact of this flicker noise on the output phase noise.

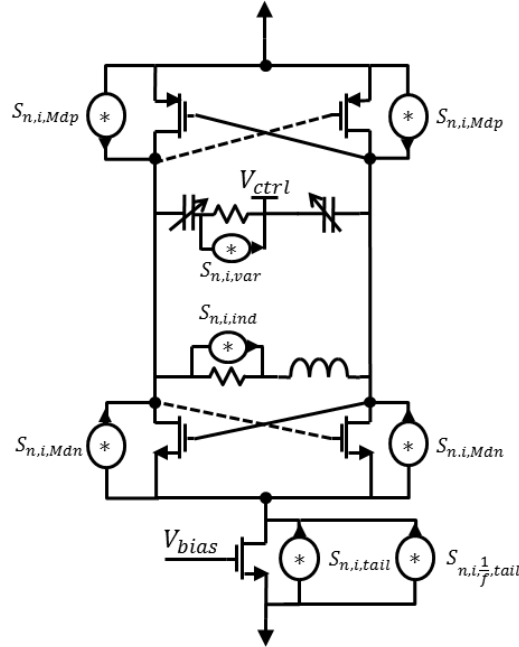


Fig. 3.6 Illustrating The Various Noise Sources In An LC-Tank VCO

The noise sources in the LC-Tank are illustrated in Figure 3.6. Here the thermal noise sources for the NMOS and PMOS transistors are shown, as well as the flicker noise

component for the tail current transistor. In [4], Hajirmiri shows that for a symmetric LC-tank VCO, within the inductor limited region the phase noise is a strong function of the inductor value L , its series conductance parasitic g_L and the tail current I_{tank} according to

$$\mathcal{L}\{\Delta_{offset}\} \propto \frac{L^2 g_L^2}{I_{tank}} \quad (3.23)$$

Clearly, to minimize the output phase noise, the product $L^2 g_L^2$ must be minimized.

Collectively, the above set of equations must be considered to optimize the performance of the VCO for a given set of specifications and loads. In [3] a graphical approach is used to define a feasibility region that yields the corresponding width for the transistors. The lengths of the NMOS and PMOS transistors are set to their minimum allowable lengths so that they introduce the least amount of parasitic capacitance. Furthermore, it assumes that the tank current is set at the highest level deemed possible to meet the power constraint. The inductance L is chosen such that its parasitic conductance g_L meets the output voltage swing condition, i.e.,

$$V_{o,amp} \approx \frac{I_{tank}}{g_L} \quad (3.24)$$

Here it is assumed that the tank output resistance $g_{tank} \gg g_L$. Generally, the inductor parasitic conductance decreases as the inductor value decreases. Finally, the varactors are sized such that they can cover the desired capacitance range.

3.3 Two Independent VCO Designs

In this section, the design of two independent LC-tank VCOs will be described. The first one will be targeted for a $0.13\mu\text{m}$ CMOS process made available from IBM for a single frequency of oscillation of 28 GHz. The second is to be targeted for a 65 nm CMOS process made available from TSMC. This second design should be capable of oscillating anywhere from 30 GHz to 40 GHz. Each design is intended to be implemented with minimum power dissipation and minimum phase noise.

3.3.1 Design Of A 28 GHz LC-Tank VCO in 0.13 μ m CMOS Process

The specifications of the first VCO can be summarized as follows:

- Phase Noise less than equal to: -80 dBc/Hz @ 10kHz, -80 dBc/Hz @100kHz and -80 dBc/Hz @1MHz
- Oscillation Frequency: 28 GHz
- Lowest Power Consumption

The overall architecture of the LC-tank VCO is shown in Figure 3.7. Here, resistor R_s is set to 2000 ohm so as not to load the Q factor of the overall LC-Tank with the 50 ohm load, and is placed outside the chip. Here a PMOS-based current source is used to create the tail current to the VCO. This was selected as a PMOS transistor is expected to inject less 1/f noise than a corresponding NMOS transistor. This will help minimise the output phase noise.

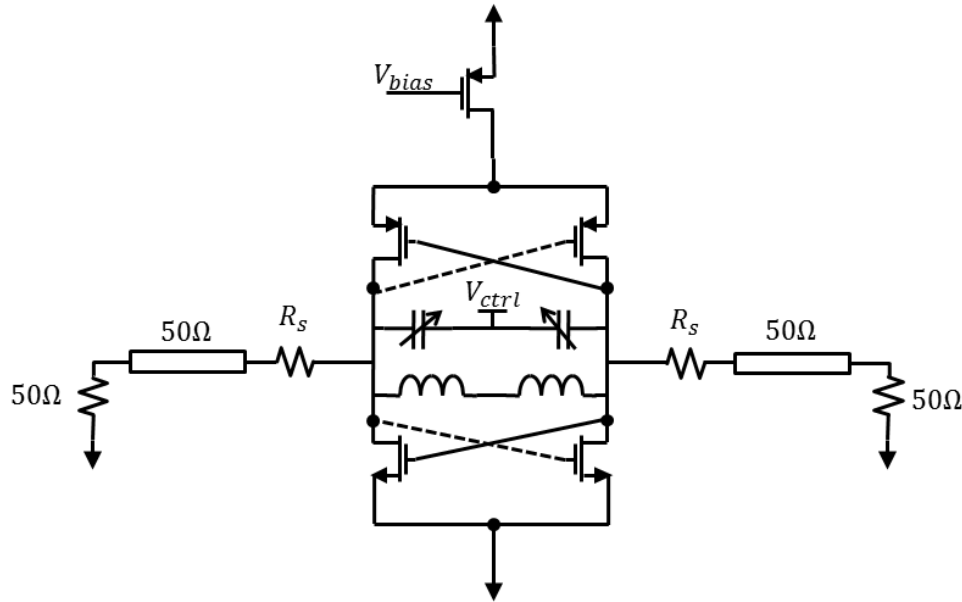


Fig. 3.7 28-GHz LC-Tank Circuit

The procedure used to design the VCO is based on Hajimri's optimization procedure [3]. However, in order to expedite the design process, some short cuts were identified that

help to shorten the design time.

LC-Tank VCO Design Procedure

The LC-tank VCO design procedure is as follows:

1. Set the W and L of the PMOS current source transistor for a desired current level
2. Set the length L of the NMOS/PMOS transistors to minimum size
3. Find W_{PMOS} and W_{NMOS} to yield equal g_m
4. Using inductor libraries, select inductor value with highest Q at desired frequency
5. Insert inductor with value found in previous step
6. With no varactors inserted, find the largest W for each NMOS/PMOS transistors that will yield the highest desired oscillation frequency
7. Choose slightly higher oscillation frequency, then pick the W for each transistor
8. Insert IBM inductor, verify frequency again and output voltage swing, increase/decrease current tail as needed
9. Use the IBM varactor libraries, select the varactor value with the highest Q at the desired frequency
10. Characterize the varactor's conductance at that frequency
11. Characterize transistors to determine the small signal parameters $g_{o,active}$
12. Compute $g_{o,active}$ and $g_{o,tank}$ using Eqns. 3.17 and 3.13
13. Make sure $g_{o,active}$ is at least three time larger than $g_{o,tank}$
14. Verify output frequency of VCO meets specifications, adjust varactor, then transistor Ws if frequency is lower or higher.
15. Increase bias current near voltage limited region if I_{tank} is non-sinusoidal, increase its DC bias level as the VCO is not in the voltage limiting region
16. If one cannot meet the frequency then one must either change the circuit topology or technology.

Following the above design procedure, the values for all components of the LC-tank VCO of Figure 3.7 are summarized in Table 3.3.1. With these components, an initial Spectre simulation was performed that revealed an output voltage swing of about 1.0 V centered on a DC level of 0.75 V. The output frequency is about 30 GHz; slightly larger

than desired, but this is meant to compensate for the additional losses that will appear at layout. The output phase noise can be seen from Figure 3.8. As is evident, the phase noise is remarkably good starting at 1 kHz offset (i.e., -132 dBc/Hz).

Table 3.1 0.13 μ m 28 GHz VCO Components Parameters

Component	Type	Value
Inductor (L)	Symmetric	139 pH Q_{peak} @ 23.64 GHz
PMOS Source	RF Transistor	nf = 25 w = 14 μ l = 300n
PMOS Drain	RF Transistor	nf = 10 w = 2 μ l = 120n
NMOS Drain	RF Transistor	nf = 7 w = 2 μ l = 120n
Varactor	Dgncap	w = 2.5 μ l = 4 μ
I_{tank}		4.11 mA

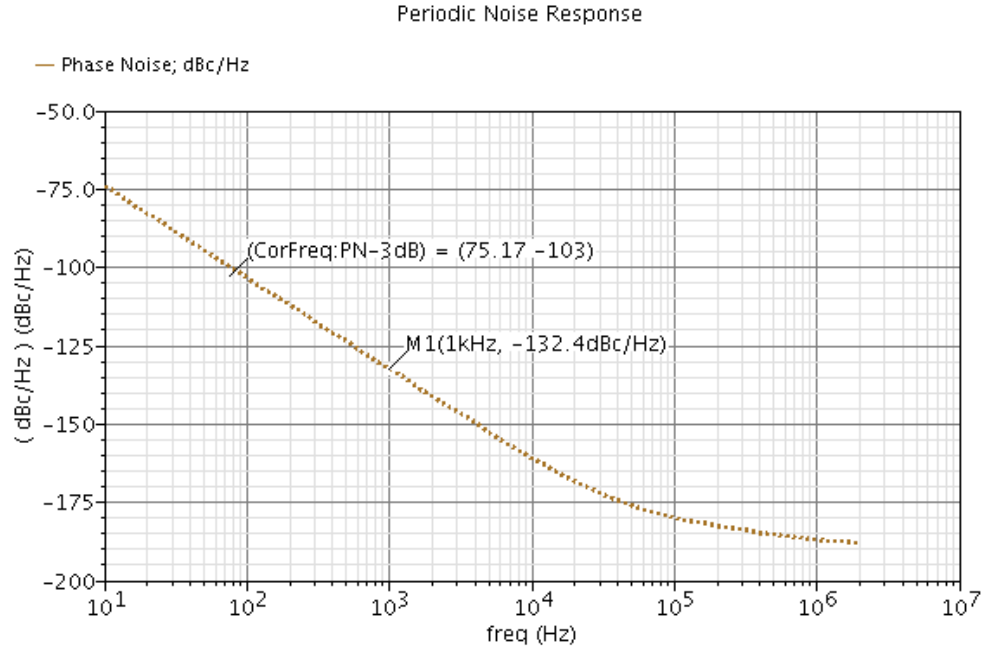


Fig. 3.8 Phase Noise 28GHz VCO Schematic

Extracted Layout Results

The layout of 28-GHz LC-tank VCO implemented in the $0.13\mu\text{m}$ IBM CMOS technology is shown in Figure 3.9. An expanded view of the layout is provided in Figure 3.10. The layout was made as symmetrical as possible about a vertical line drawn through the middle of the layout. The current direction through the layout was directed from top to bottom in all cases except through the inductor. The expanded view of the layout shows its compactness, which is essential for limiting any parasitic capacitances. Furthermore as seen in Figure 3.10, the PMOS and NMOS transistors are arranged in a common-centroid fashion as described in reference [27]. Common-centroid layout was performed on the complete transistor itself, instead of splitting each of them apart using various fingers. This was deemed necessary as the transistors were carefully characterized by IBM and we did not want to alter their expected behaviour in any possible way.

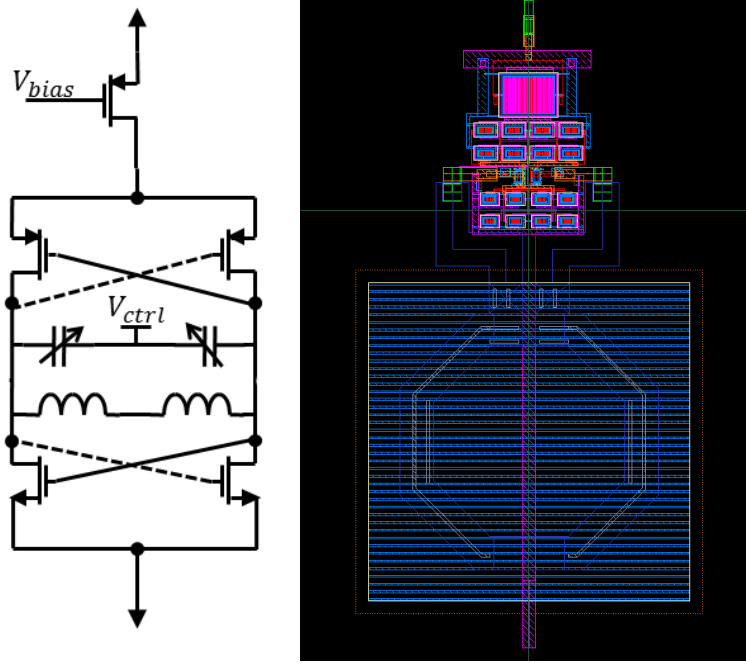


Fig. 3.9 28-GHz LC-Tank VCO Schematic and Layout

After the layout was completed, a Spectre simulation was performed on the extracted circuit. This view includes the effects of various parasitics associated with the various components. Often it is necessary to go back to the original design and re-size the transistors, increase the bias current level, lower the varactor capacitances, etc., until the layout ex-

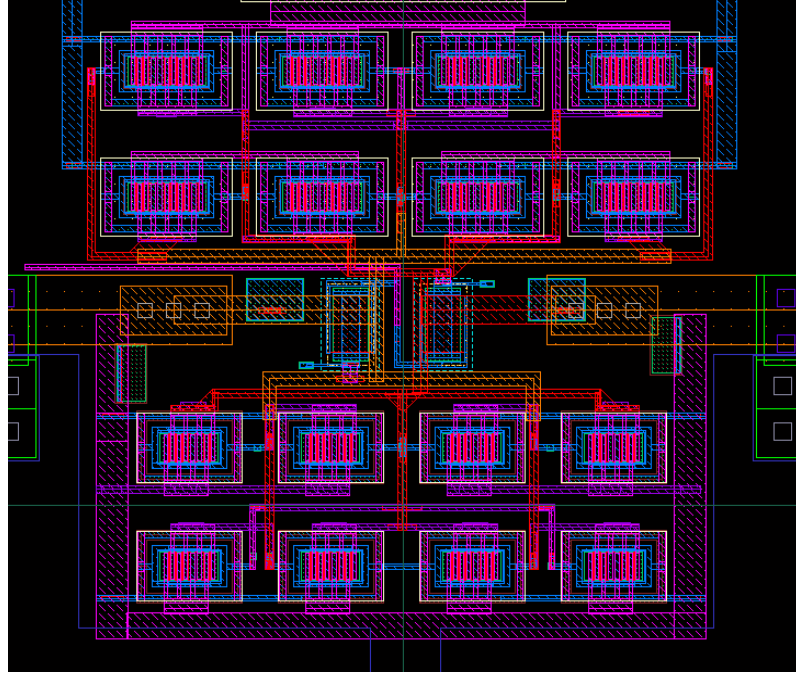


Fig. 3.10 28-GHz LC-Tank VCO Layout Zoomed

tracted view is within specification. After some iterations, Figure 3.11 displays the output voltage signals at the 50 ohm load and from the VCO output node. Prior to layout extraction, the circuit oscillates at a frequency of about 30 GHz with a signal amplitude of about 1.0 V. When the layout parasitics are included, we see the VCO oscillates at a frequency of 28.3 GHz with a single amplitude of about 1.2 V.

The output phase noise of the extracted VCO circuit is shown in Fig. 3.12. Compared with schematic simulation results shown previously in Figure 3.8, the phase noise is much worst. To the point, where no direct comparison can be properly made. This could be explained by the additional parasitic resistance included with the inductors and capacitors, the large-sized transistors and potential coupling between the input and output of the VCO.

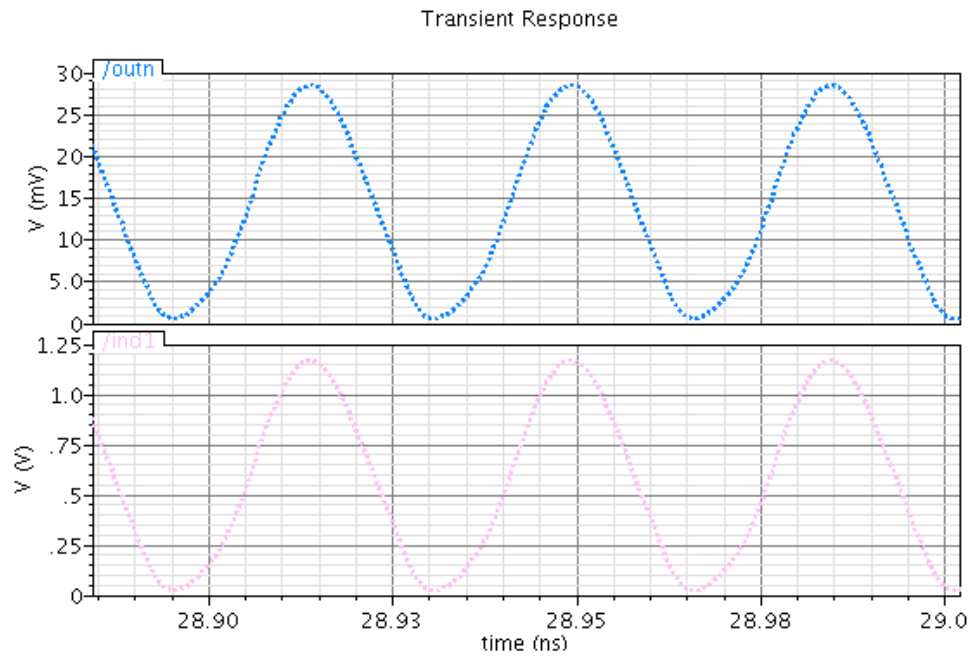


Fig. 3.11 28-GHz LC-Tank VCO Extracted Layout Output Voltage At 50 ohm Load (Top) and VCO Node (Bottom)

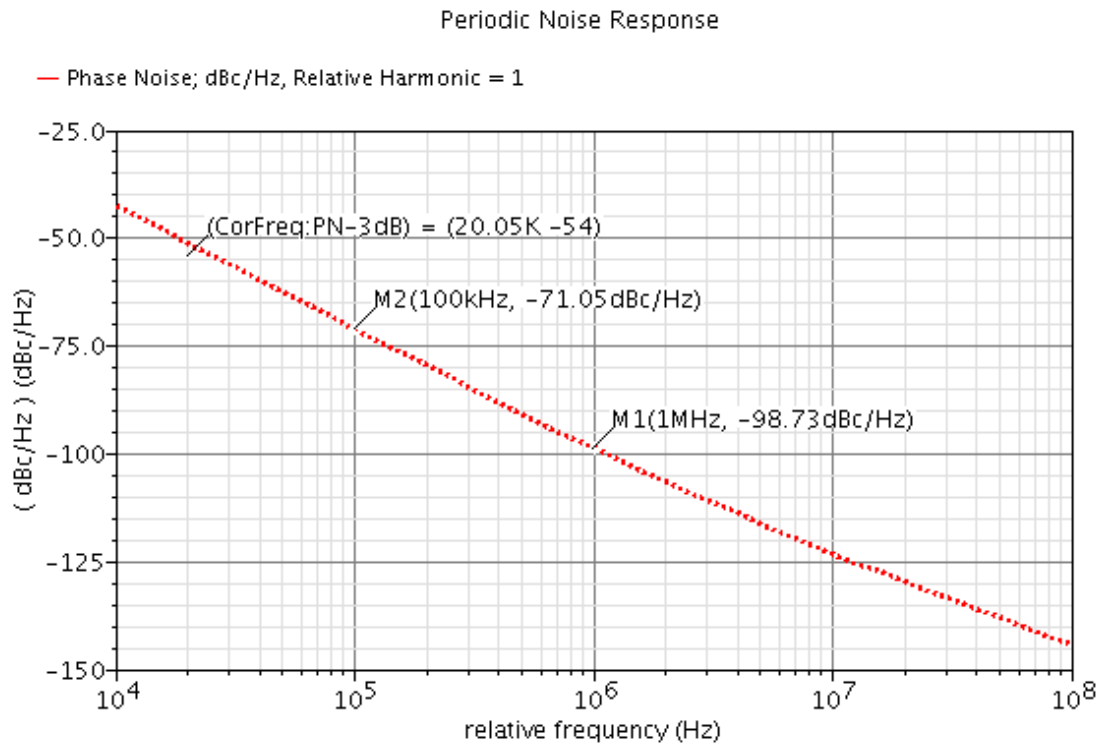


Fig. 3.12 28-GHz LC-Tank VCO Layout Phase Noise

The output frequency with respect to the control voltage is shown in Figure 3.13. Here the control voltage is sweep from 0 to 2 V. As can be seen, the oscillation behaviour is quite nonlinear. This is due to the nonlinear behavior of the varactor itself. The varactor capacitance is nothing more than C_{gs} and C_{gd} of a MOS transistor in parallel. These capacitors vary non-linearly with their source and drain voltage. If this VCO is to be used in a system such as a PLL, the region between 0.5 V up to 1.0 V could be used as it is much more linear in this region.

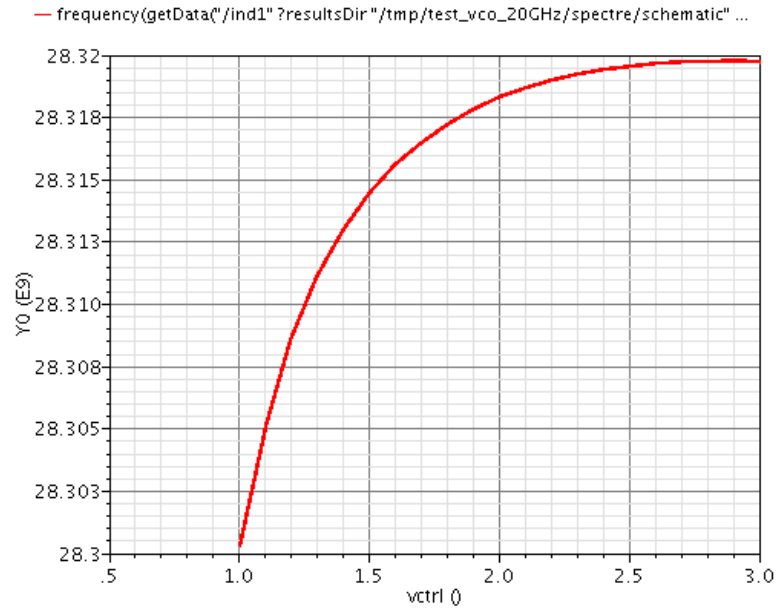


Fig. 3.13 28 GHz LC-Tank Layout Extracted Frequency vs Vctrl

Considering the transfer function of the PLL mentioned in Chapter 2, the overall closed loop response of a PLL is dependent on the gain of the VCO, K_{VCO} as previously defined. This gain as a function of the VCO control voltage can be seen from Figure 3.14. As is evident, this gain is not flat and can vary as much as 50 times over its extremes. The impact on the PLL can be as serious as instability or unable to meet its specifications.

A chip has been fabricated and the results will be provided in Chapter 7.

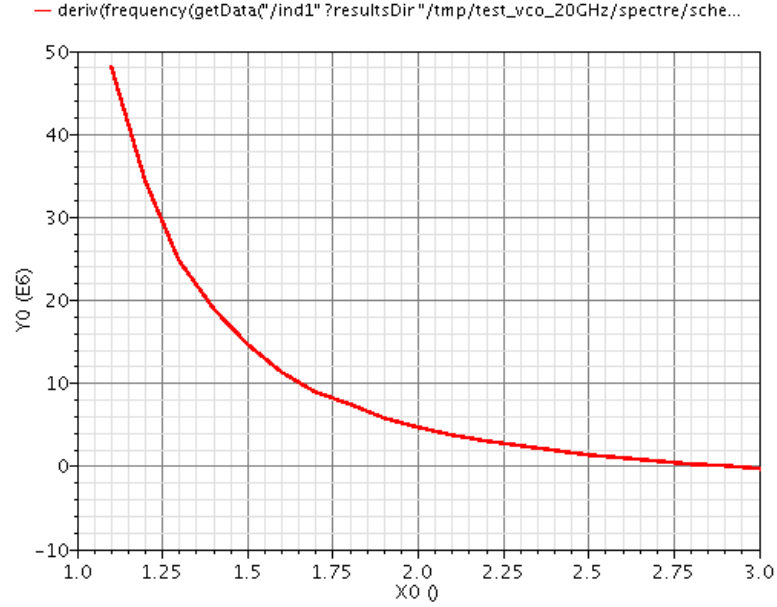


Fig. 3.14 28 GHz LC-Tank Layout Extracted K_{VCO} vs V_{ctrl}

3.3.2 Design Of A 30 - 40 GHz LC-Tank VCO in 65 nm CMOS Process

At one point the specifications of the PLL was increase to as high as 40 GHz. As this oscillation frequency is almost 50% higher than what could be achieved with the previous design. In fact, the complementary structure was reaching the limits of the technology for a 40 GHz sustained oscillation. The circuit was very sensitive to temperature and process variations due, in part, to the lack of g_{active} at this frequency. A higher g_{active} would have required transistors with larger widths which, in turn, would have added to the parasitic capacitance and hence lower the oscillation frequency. It was hence decided to switch technology and move to the TSMC 65 nm CMOS process. Within this technology, a 30-40 GHz LC-tank VCO was designed. The design of this VCO is presented next.

The 65 nm VCO was designed using the same procedure presented previously. The circuit diagram of the tuneable LC-tank VCO is shown in Figure 3.15. The circuit values can be seen listed in Table 3.3.2. The circuit includes varactors controlled by an analog voltage V_{CTRL} and a set of digitally-controlled varactors, D_1 , D_2 and D_3 . These digitally-controlled varactors permit the range of the VCO to be extended by adding fixed amounts of capacitance.

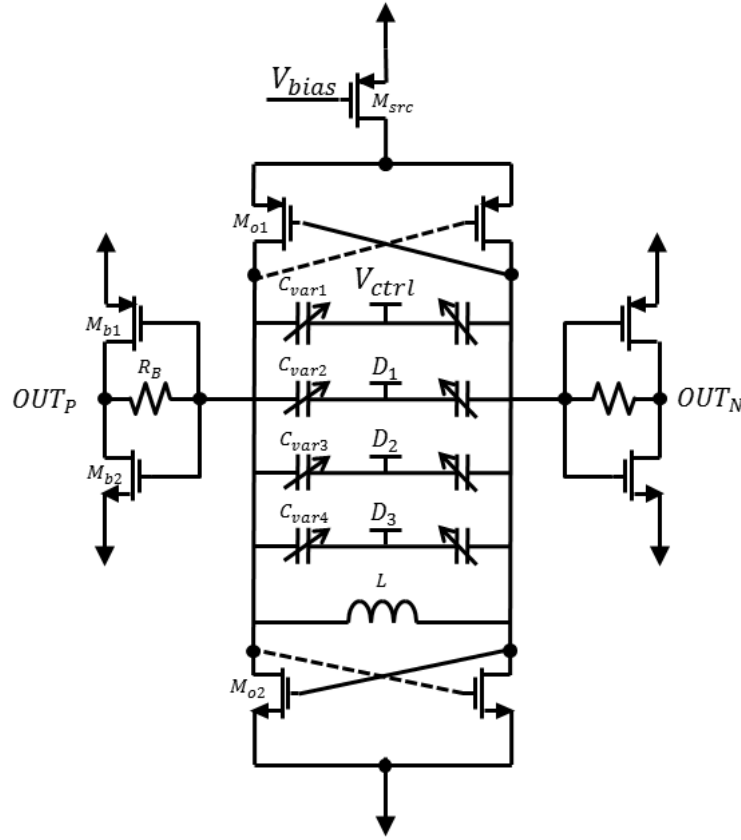
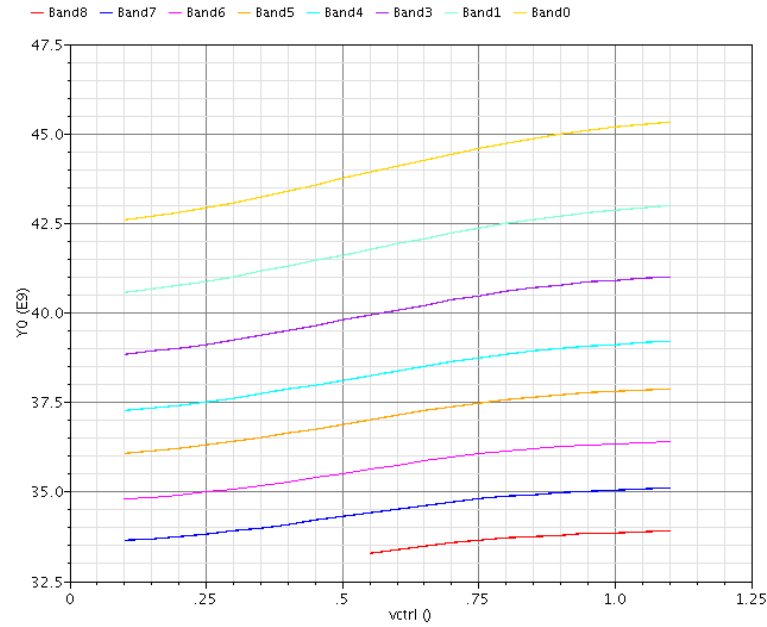


Fig. 3.15 30-40GHz 65nm LC-Tank VCO Schematic

The simulated frequency range of the VCO from the schematic level is shown in Fig. 3.16. The highest frequency corresponds to all digitally-controllable varactors grounded; thus setting the smallest shunt capacitance. Conversely, the smallest oscillating frequency is achieved when all varactors are turned on, at a voltage bias level of 1.5 V. As this graph shows, the frequency steps overlap, and the maximum frequency is about 46 GHz while the minimum is 32.5 GHz. When designing high frequency VCOs, layout plays a major role and the design requires back and forth iterations in order to succeed, hence the reason for selecting the higher oscillation frequency.

Table 3.2 65nm 31-40GHz VCO Components Parameters

Component	Type	Value
Inductor (L)	Symmetric	72.034 pH Q_{peak} @ 20.0 GHz
PMOS Source M_{src}	RF Transistor	nf = 32 w = 5.00μ l = 200n
PMOS Drain M_{o1}	RF Transistor	nf = 16 w = 4.11μ l = 60n
M_{b1}	RF Transistor	nf = 4 w = 2.40μ l = 60n
NMOS Drain M_{o2}	RF Transistor	nf = 8 w = 3.20μ l = 60n
M_{b2}	RF Transistor	nf = 4 w = 1.04μ l = 60n
Varactors C_{var1}	MOSCAP RF	w = 2.37μ l = 0.9μ
C_{var2}	MOSCAP RF	w = 1.75μ l = 0.9μ
C_{var3}	MOSCAP RF	w = 1.72μ l = 0.9μ
C_{var4}	MOSCAP RF	w = 1.84μ l = 0.9μ
Resistors R_B	Rpolywo RF	w = 1.00μ l = 10.0μ
I_{tank}		6.11 mA

**Fig. 3.16** 30-40GHz 65nm LC-Tank VCO Frequency Range vs Control Voltage

The next figure of interest shown in Fig. 3.17 is the gain of VCO as a function of the control voltage for each tuneable frequency. As clearly evident the VCO gain is not constant, instead it changes non-linearly with input control voltage. The gain varies as much as three times the nominal value. As a consequence, it affects the overall closed loop PLL transfer function and needs to be considered when doing a high-level design implementation.

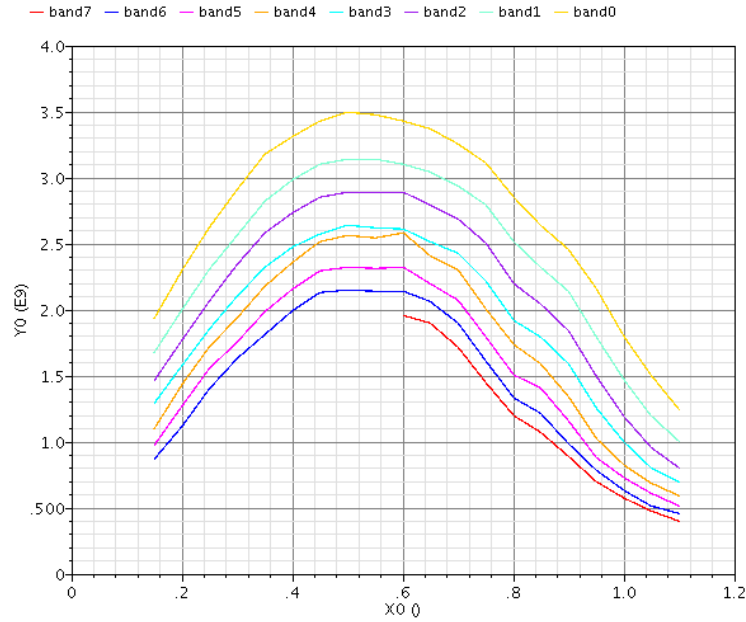


Fig. 3.17 30-40GHz 65nm LC-Tank K_{VCO} vs Control Voltage

The output phase noise associated with this LC-tank VCO is shown in Figure 3.18. The 3-dB of the $1/f^3$ corner frequency is at about 66.5 kHz. A lower value yields a better phase noise and relaxes the requirements on the loop filter. This phase noise plot for the schematic circuit is exceptionally good, however as will be seen in the next chapter, the layout dictates the behavior, and hence most of these results are kind of meaningless for these high frequency circuits. Furthermore, it is noted that transistor noise in 65 nm have only been characterized up to 10 GHz.

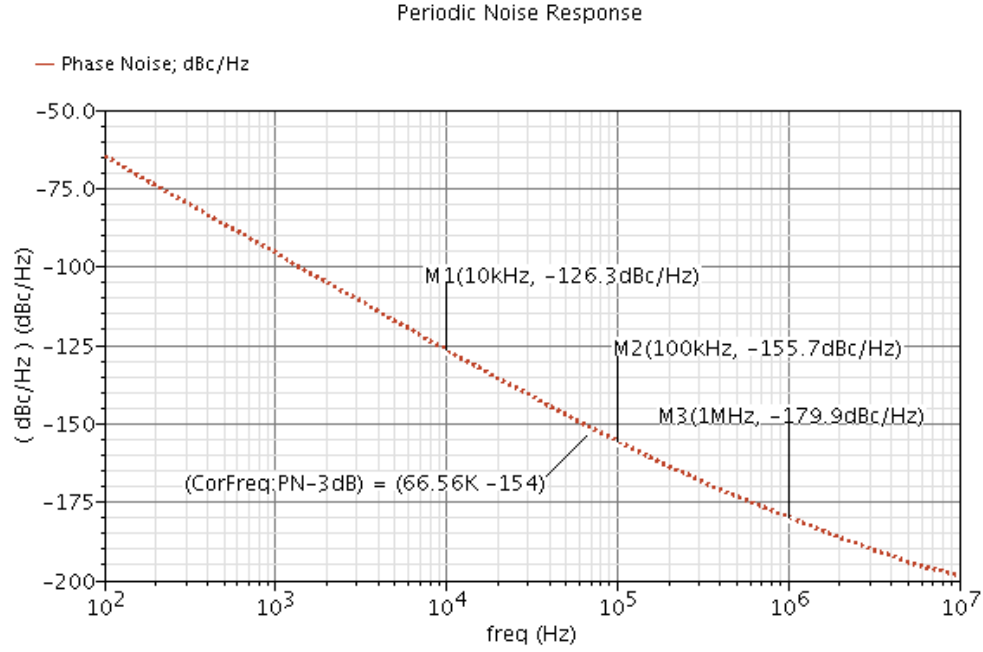


Fig. 3.18 30-40GHz 65nm LC-Tank Schematic Circuit Phase Noise

Extracted Layout Results

In this subsection, the extracted layout results for the 31-40 GHz LC-tank VCO are reported. The circuit along with the layout of this VCO are shown in Figure [?]. The layout is made compact and as symmetric as possible. In comparison to the previously designed VCO in $0.13\mu\text{m}$ technology, the VCO transistors were not interdigitated in a pattern, instead, they were layout directly as is. This permits a more compact design but at the expense of greater sensitivity with respect to wafer variations. In addition to the previous VCO, MOS capacitors are populated along empty spaces of the VCO cell, they serve as decoupling capacitors for the power, as well as fillers for the cell.

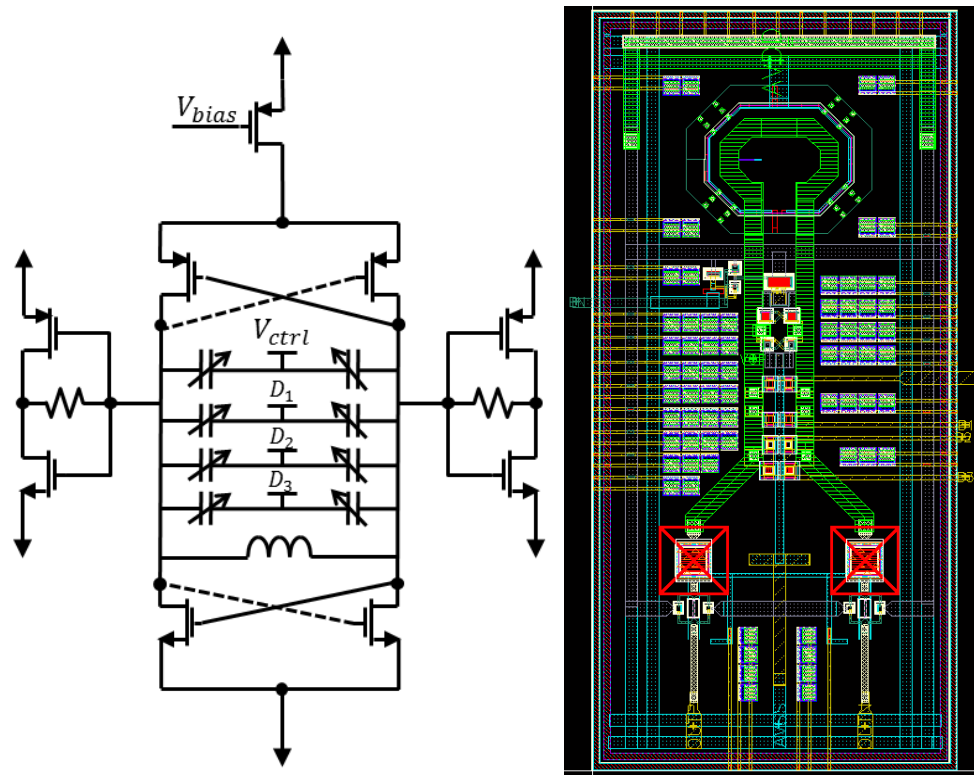


Fig. 3.19 31-40GHz LC-Tank VCO Schematic and Layout

The gain of the VCO with respect to the control voltage behaves in much the same way as the schematic extraction results as shown in Figure 3.20. Here the gain changes as much as 10 times over the full range of all possible digital control settings and about three times within a specific digital control setting.

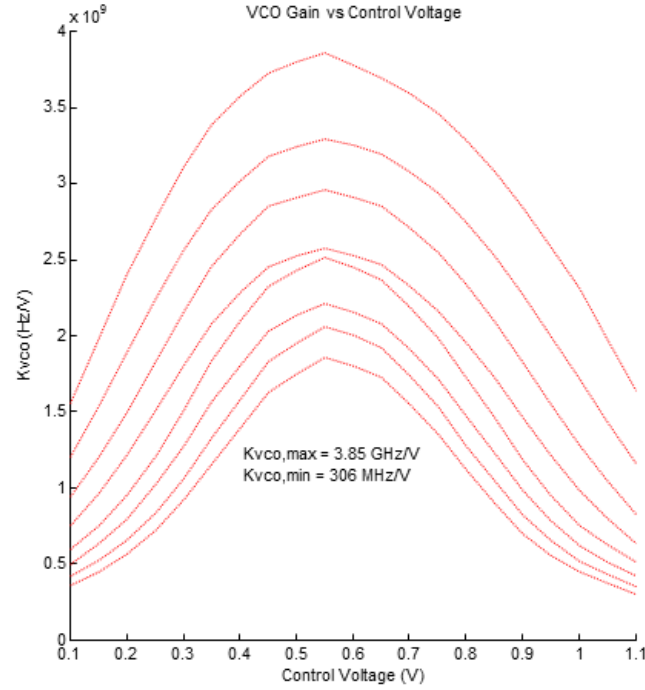


Fig. 3.20 31-40GHz LC-Tank VCO Extracted Layout K_{VCO} vs V_{ctrl}

The output phase noise for the extracted design is shown in Figure (3.21). The $1/f^3$ corner frequency is at about 1.47 MHz versus 66.56 kHz for the schematic equivalent. The difference can be attributed to additional resistance associated with nets within the layout. Furthermore, the phase noise at 1 MHz is -96 dBC/Hz or nearly twice as much as that derived from the schematic level result at -179.9 dBC/Hz. This is, of course, related to the $1/f^3$ corner frequency as well as extra noise generated by the interconnections (which are mostly resistive in nature).

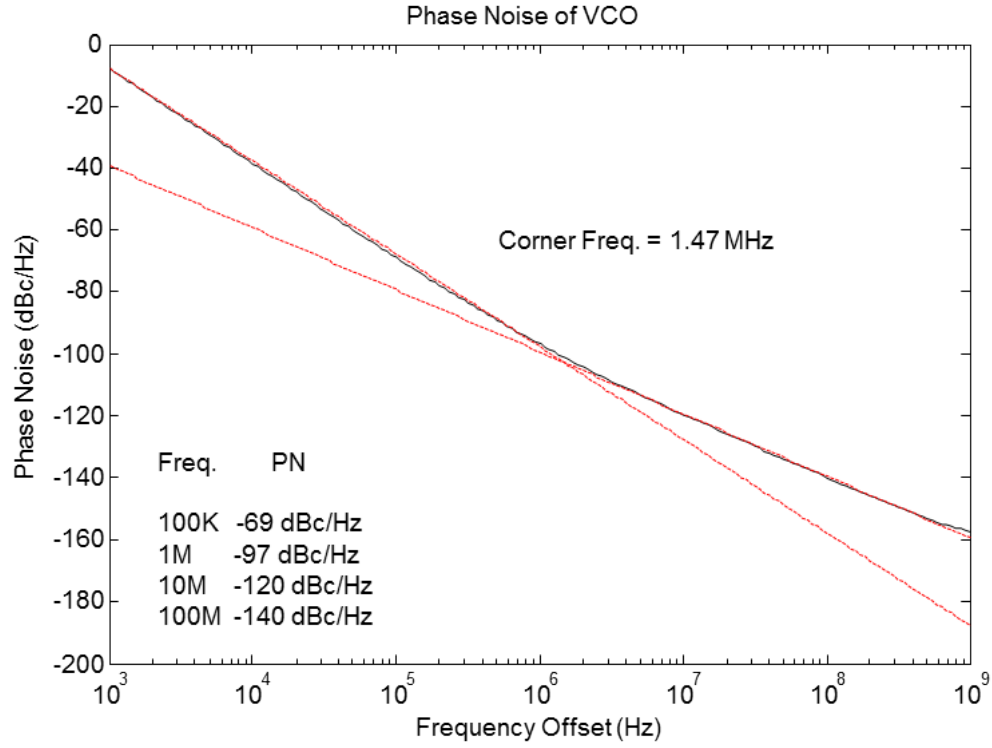


Fig. 3.21 31-40GHz LC-Tank VCO Extracted Layout Phase Noise

From the simulated results associated with the extracted layout of the VCO, it is imperative to note the importance of aiming for a higher operating frequency and a lower phase noise, and any other parameter dependent on layout, to pre-compensate for the losses associated with the interconnections that make up a circuit. These effects cannot be predicted a priori due mainly to the non-linear mappings that influence the noise.

3.4 A VCO Verilog-A Model

In this section, a Verilog-A model of a VCO that takes into account its K_{VCO} gain and phase noise is presented. The material is taken from the Designer's Guide application note [13]. In order to generate the VCO frequency, the modulus integration approach can be used in Verilog-A. The *idtmod* function takes as inputs the frequency, along with its initial phase condition, its upper frequency bound, as well as any frequency offset and absolute tolerances. The absolute tolerance need not be given as the default values set by Spectre can be used. At each period of the input, the function will hit unity and, if an offset is added then the function will march on the value one minus this offset. This mathematical procedure effectively implements a phase conversion. In order to generate a square wave, the *cross* function is used. When the expression inside the cross function evaluates to zero, within a tolerance given by *ttol*, the code that follows gets executed. Figure 3.22 shows the modulus integration along with two points that been chosen to trigger on a high or low value.

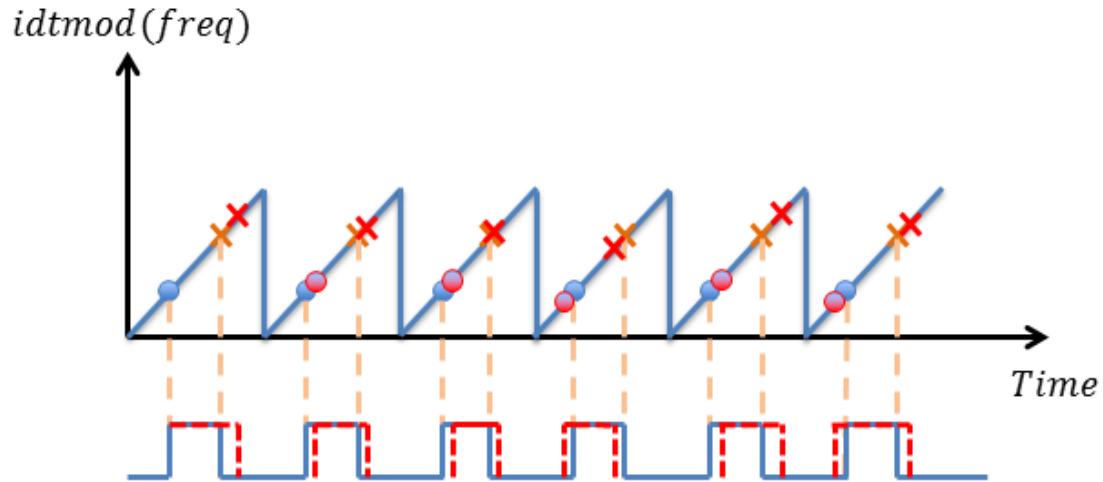


Fig. 3.22 Modulus Integrator and Addition of Jitter Process

In this figure, the red line shows how jitter could be added by simply changing the frequency of the input to the integration function. The jitter is injected by re-calculating the frequency as $\text{frequency}/(\text{frequency} \cdot \text{jitter} + 1)$. The VCO phase noise is mapped to

jitter using the procedure described in [13], and repeated here as

$$J = \sqrt{(\Delta f) \frac{\Delta f^2}{f_0^3}} \quad (3.25)$$

As explained in [13] this jitter quantity needs to be multiplied by the factor of $\sqrt{2}$ in order to account for two transitions per period. A Gaussian uniform distribution is used to create the random pattern between 0 and 1. The desired absolute jitter value is then multiplied by this Gaussian noise to generate the random effect. Flicker noise is not included in the model and usually is not necessary as the the corner $1/f^3$ in the phase noise plot of the VCO is well below the loop filter cutoff frequency. If this is not the case then it should be included.

Verilog-A code for the VCO with the divider function included is as follow:

```
'include "constants.vams"
'include "disciplines.vams"

module VCO_NON_IDEAL(in, bits_freq, out);

input in;
input [0:2] bits_freq;
output out;

electrical out, in, delta_in;
electrical [0:2] bits_freq;

parameter real vmin = 0;
parameter real vmax = 1 from(vmin:inf);
parameter real ratio = 392.05 from [2:inf); //fixed divider ratio N*2
parameter real Vlo = -1, Vhi = 1;
parameter real tt = 1e-13 from (0:inf);
parameter real jitter = 2.528e-15 ;// from [0:0.25*ratio/Fmax); //vco
    period jitter
//parameter real jitter = 0;
parameter real ttol = 0.1u/31e9;
parameter real outStart = 2e-3;

real Kvco, Kvmin, Kvmax, Fmin, Fmax;
real freq, phase, dT, delta, prev, Vout;
```

```

integer n,seed,fp;
real freq_range;
analog begin
    @(initial_step) begin
        seed = -561;
        delta = jitter*sqrt(2*ratio);
        fp = $fopen("~/PROC_65nm/High_Level_PLL/data/
            periods_vco_ds.m");
        prev = $abstime;
        Vout = Vlo;
    end

    $discontinuity(0);
    //check Frequency selected range
    freq_range = floor(V(bits_freq[0])+0.5) +2*(floor(V(bits_freq[1])
        +0.5))+4*(floor(V(bits_freq[2]+0.5)));

    //Create Look-up Table for Kvmin, Kvmax, Fmin based on KVCO graph
    case(freq_range)
        0: begin Kvmin = 01.560e9; Kvmax = 3.851e9; Fmin = 38.94
            e9; Fmax = 40.99e9; end
        1: begin Kvmin = 01.165e9; Kvmax = 3.290e9; Fmin = 37.40
            e9; Fmax = 39.24e9; end
        2: begin Kvmin = 824.50e6; Kvmax = 2.957e9; Fmin = 36.05
            e9; Fmax = 37.73e9; end
        3: begin Kvmin = 639.90e6; Kvmax = 2.582e9; Fmin = 34.79
            e9; Fmax = 36.31e9; end
        4: begin Kvmin = 515.60e6; Kvmax = 2.515e9; Fmin = 33.86
            e9; Fmax = 35.28e9; end
        5: begin Kvmin = 427.00e6; Kvmax = 2.219e9; Fmin = 32.80
            e9; Fmax = 34.09e9; end
        6: begin Kvmin = 355.70e6; Kvmax = 2.065e9; Fmin = 31.85
            e9; Fmax = 33.04e9; end
        7: begin Kvmin = 306.00e6; Kvmax = 1.857e9; Fmin = 30.96
            e9; Fmax = 32.05e9; end
        default: begin Kvmin = 1.560e90; Kvmax = 3.851e9; Fmin =
            38.94e9; Fmax = 40.99e9;
    end
endcase

```

```

//Approximate KVCO by Triangular wave
if (V(in) <= 0.6) Kvco = Kvmin + V(in)*(Kvmax - Kvmin)/0.6;
if (V(in) > 0.6) Kvco = Kvmax - (V(in)-0.6)*(Kvmax - Kvmin)/(vmax
    - 0.6);

// $bound_step(1e-10);
//compute frequency
freq = (V(in) - vmin)*Kvco + Fmin;

//bound frequency
if (freq>Fmax) freq = Fmax;
if (freq<Fmin) freq = Fmin;

//add phase noise based on periodic jitter
freq = (freq/ratio)/(1-dT*freq/ratio);

//compute phase
phase = idtmod(freq,0.0,1.0,-0.5);

@(cross(phase-0.25,+1,ttol)) begin
    dT = delta*$rdist_normal(seed,0,1);
    Vout = Vhi;
end

@(cross(phase+0.25,+1,ttol)) begin
    dT = delta*$rdist_normal(seed,0,1);
    Vout = Vlo;
    if($abstime >= outStart) $fstrobe(fp,"%0.20e",$abstime-
        prev);
    prev = $abstime;
end

//print max divide ratio
@(final_step) begin
    $strobe("input_vctrl_e_and_kvco_is_e", V(in), Kvco);
    $strobe("frequency_is_e", freq);
    $fclose(fp);
end

```

```
        V(out) <+ transition(Vout,0,tt);  
end  
endmodule
```

The VCO gain in this case is mapped non-linearly with respect to the control voltage in order to better reflect that found in practise. It is important to state that the resolution of the phase noise relies on the transitions step size along with the tolerances. The `bound_step` function might be included in order to limit the maximum step in the simulation engine. In reference [28] another Verilog-A model is proposed in which the ISF characteristic function of the VCO for any input node is used to create the jitter. This is very useful, say for instance, to predict the influence of power supply variations on the phase noise.

3.5 Summary

This chapter has described the procedure to design an LC-tank VCO for very high-frequency operation using a CMOS technology. The design procedure focuses on the frequency range optimization as well as the phase noise and power consumption for the VCO circuit. Schematic circuit and extracted layout results were reported for two separate VCO designs. Finally, a Verilog-A model that includes the non-linear VCO gain behaviour and phase noise was described. This model will play an important role in the design of a PLL in later chapters of this thesis.

Chapter 4

Phase Frequency Detector and Charge Pump Design

This chapter outlines the design of the phase-frequency detector (PFD) and the charge-pump (CP). The designs are meant for implementation in a 65 nm CMOS process from TSMC. Simulation results for both the schematic and extracted layout level circuits will be provided. Furthermore, a Verilog-A model of the PFD and CP that incorporates their phase noise and non-linear operation will be provided.

4.1 PFD Block

The first block to be discussed is the PFD implementation that was realized in a 65 nm CMOS technology from TSMC. First, the non-ideal theoretical behavior of a PFD is presented, followed by its circuit implementation.

4.1.1 PFD Non-Ideal Behavior

As previously described in Chapter 2, the phase frequency detector is an essential component of a PLL that generates an output that is proportional to the difference in phase between its two inputs. In that chapter, the tristate PFD was shown. This topology is best suited to frequency synthesis since it is able to detect phase difference up to 2π and beyond. In Figure 4.1(a), the ideal output DC voltage versus input phase difference is shown. As can be seen, the average or DC output of the PFD is linearly proportional to the phase

difference from -2π to 2π , the pattern is modulo 2π . In practice, due to time delays within the PFD, two main non-ideal behaviors are observed, these are referred to as dead-zone and blind-zone phenomena. These are shown in parts (b) and (c) of Figure 4.1.

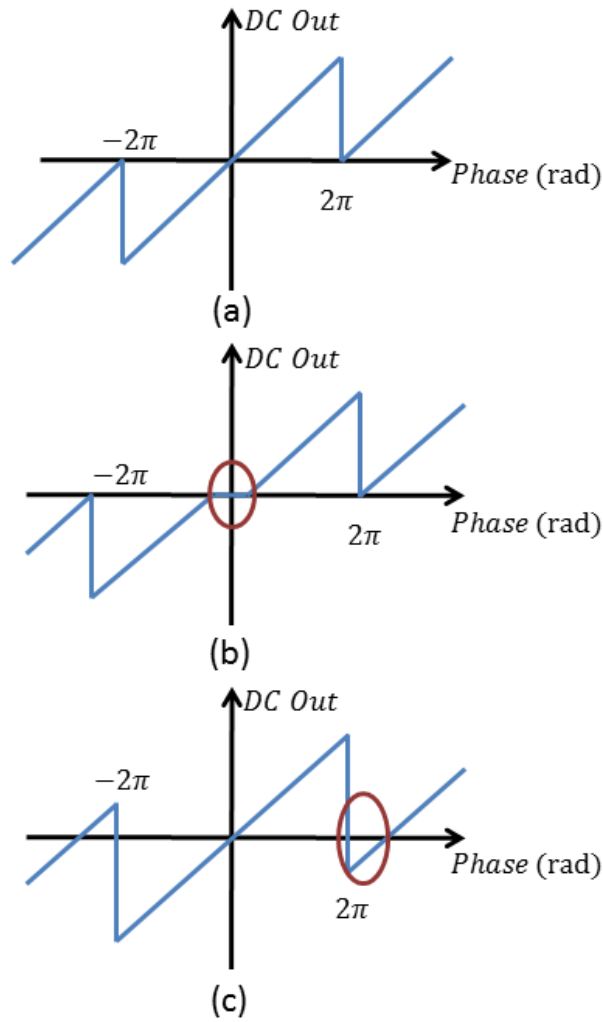


Fig. 4.1 Phase Frequency Detector DC Output vs Input Phase Difference
(a) Ideal (b)Dead-Zone (c)Blind-Zone

Dead-Zone Region

The dead zone is the region for which a phase difference smaller than a given Δt produces zero output. This is attributed to the time the internal blocks within the PFD need to reset its internal states. Figure 4.2 shows the effect of the dead zone on the UP1 and DN1 signals. As shown for a rising edge of the VCO output, the block resets its value, however, the reset function takes longer than the time difference for REF to rise up, and hence, this rising edge is being missed by the PFD. Dead-zone increases the phase noise at the output of the VCO as any small-signal phase noise cannot be filter by the PLL closed-loop phase response. One solution is to alleviate the dead-zone is to insert a time delay, denoted by t_{delay} , in the feedback path of the reset function. As shown in the same figure, the signals UP2 and DN2 signals are both one over the time duration of t_{delay} . This delay lets the PFD detect two close rising edge and hence eliminates the dead-zone region. However, since both outputs are on at the same time, any mismatch in the charge pump will inject a negative or positive extra charge pulse into the CP of the PLL resulting in output spurs [29]. Therefore careful selection of this time delay is essential for good PLL operation.

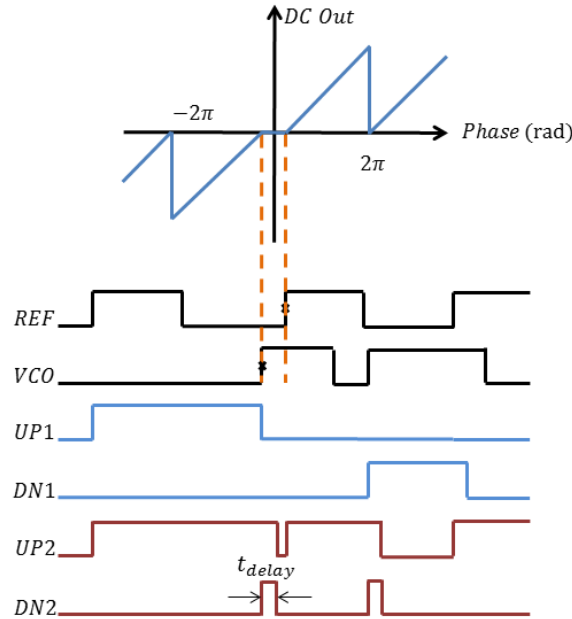


Fig. 4.2 Illustrating The PFD Dead-Zone Non-Ideal Behavior

Blind-Zone Region

The other non-ideal behavior associated with the PFD is its blind-zone. As previously shown, instead of returning to zero at each 2π phase difference increment, the output of the PFD undershoots or overshoots this value. To better understand the impact of this behaviour, the situation shown in Figure 4.3 for a PFD output that undershoots its expected value of zero. This scenario is similar to the one shown for the dead-zone region, but in this case the feedback delay is also included to eliminate the dead-zone region. For this particular case, the VCO input propagates and resets the UP1 output, which, in turn, deactivates the DN1 signal. Meanwhile, a rising edge at REF does not get detected, since VCO output still controls the internal logic of the PFD. The correct behavior of the circuit is illustrated by the UP2 and DN2 signals also shown in this figure. In order to realize the ideal behavior, the internal PFD signals must be delayed. This will be discussed next along with the circuit level implementation of a PFD.

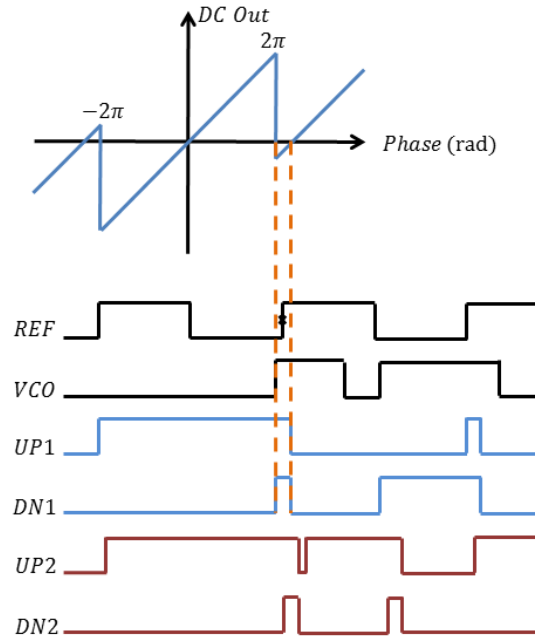


Fig. 4.3 Illustrating The PFD Blind-Zone Non-Ideal Behavior

4.1.2 PFD Circuit Level Implementation

The block level circuit implementation of the tristate PFD is illustrated in Figure 4.4. Here two blocks (each consisting of a D-type flip-flop as described in Chapter 1 with an AND gate and additional delay in the feedback path to eliminate the dead-zone region) are shown. The PFD is driven with two input signals (VCO and REF) and a reset control line. The transistor level implementation of the dynamic PFD is illustrated in 4.5. The circuit is taken from [6] with a slight modification of including a reset signal. The circuit needs to be reset before its operation, as the circuit might assume a random state on power up.

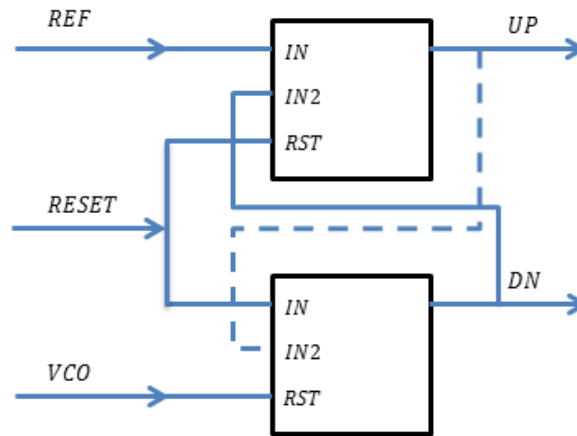


Fig. 4.4 PFD Block Level Implementation

The number of inverters at the input were determined by the delay required to eliminate the blind-zone. The dead-zone is alleviated by the feedback reset delay embedded into the logic. The behavior of the circuit when both UP and DN signals are high is as follows: the IN2 signal that corresponds to the output of the second block along with the OUT signal brings the logic to a zero level if the input is still high. The delay is determined by the propagation delay of the logic within the last three branches.

Figure 4.6 shows the PFD average DC output versus the input phase difference. In this case, a 10 ns period reference was applied as input along with a VCO pulse that was swept from 5 ns to 35 ns. The reset signal was applied at the beginning of the simulation. As is evident, the PFD exhibits no dead-zone region and nearly no blind zone region, implying the delay in the feedback path is enough to eliminate these two non-ideal behaviours. Figure 4.7 shows a zoom-in region of the blind-zone region. From this figure, the blind zone value



4.2 Charge Pump Block

4.2.1 Charge Pump Non-Ideal Behaviour

A simple charge pump circuit is shown in Figure 4.8. In this circuit, a fixed current I generated by a Bandgap voltage reference circuit is being reproduced using a current mirror. Transistors M_5 to M_8 carry a current i_{pn} and i_{dn} larger than I and hence are sized bigger than transistors M_1 to M_4 . The $\overline{UP}$ signal corresponds to the negated UP signal

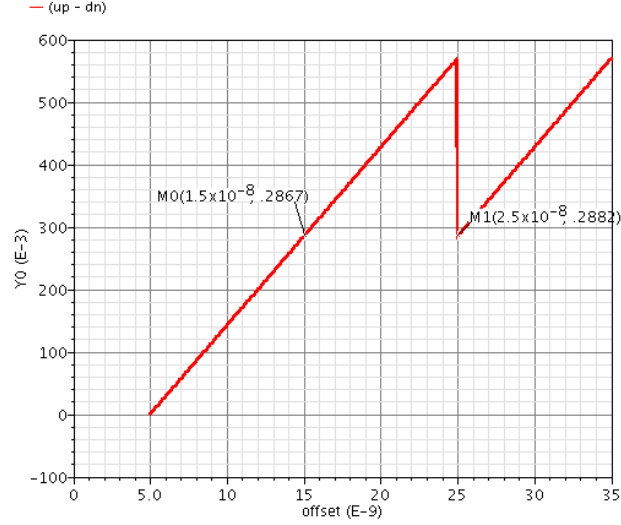


Fig. 4.6 PFD Input-Output Transfer Characteristic

from the PFD, hence, when the VCO lags the Reference signal at the PFD input, the UP signal will rise, and a current, i_{pn} will be sourced into the load. In contrast, when the Reference leads the VCO output, the PFD will trigger a DN signal, which in turn will sink a current i_{dn} from the load.

4.2.2 Current-Voltage Variation

As the current sinking and sourcing toggles over time, the output voltage of the charge pump changes. From the transistor square law behaviour, the output current depends on its V_{DS} value according to

$$I_{out} = \left(\frac{\mu C_{ox}}{2} \right) \left(\frac{W}{L} \right) (V_{GS} - V_{Th})^2 (1 + \lambda V_{DS}) \quad (4.1)$$

As V_{DS} fluctuates, the current will vary in direct proportion to the voltage variation. The output resistance of the transistor can be increased by increasing the length of the transistor, however this adds noise and increases the parasitic capacitance in the circuit, thereby decreasing the overall speed of operation of the CP. A small test was performed in Cadence in which the output voltage of the CP was varied and currents i_{pn} and i_{dn} were extracted over a some time interval. Figure 4.9 shows the results. As can be seen, the two

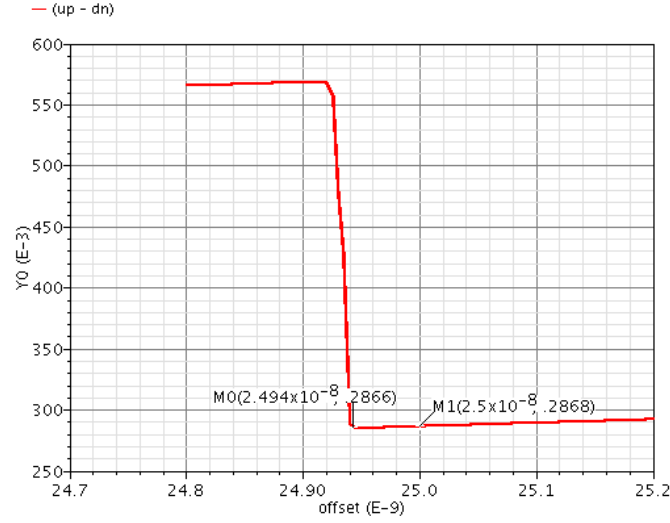


Fig. 4.7 A Zoomed In View Of The PFD Blind-Zone Region Of Fig. 4.6.

currents tend to vary a lot with the output voltage. In a PLL system, this would result in a gain change in the PFD and CP high level block, which results in dynamic movement of the poles positions. As a result, the PLL could become unstable or, the phase response behavior loses its noise rejection capabilities.

4.2.3 Current Mismatch

As explained previously with the PFD block, in order to cancel the dead-zone effect, a delay is incorporated into the feedback path of the PFD. In effect, the UP and DN signals from the PFD are set high for a small time interval corresponding to the reset delay time. In the CP circuit presented in Figure 4.8, voltages V_{B1} and V_{B2} mirror the exact same current I through the branch M_1 to M_4 , which in turn mirror the current to branch M_5 to M_6 . In essence, the i_{pn} and i_{dn} should be matched perfectly, however in practice, layout, technology process variations, and mismatches between transistors create currents that are unequal.

The effects of mismatches between the sourcing and sinking current is better illustrated with the timing diagram shown in Figure 4.10. In this diagram, the steady state operation of the PLL is assumed. The voltage at the output of the CP is denoted as V_{OUT} . Here the source current is assumed to be slightly larger than the sinking current. Because of this

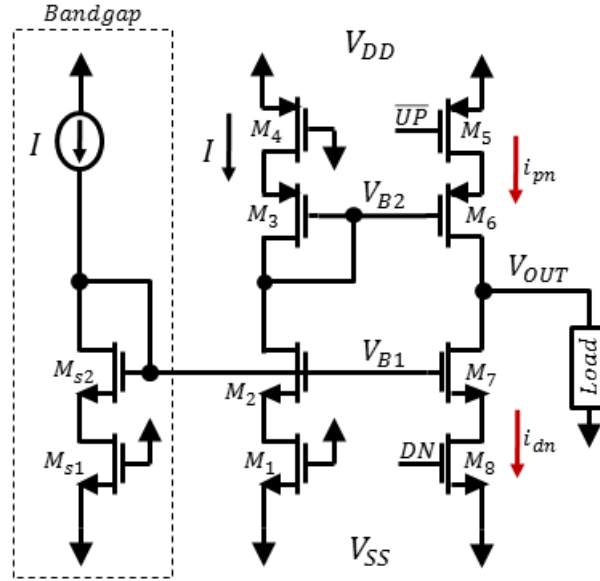


Fig. 4.8 Simple Charge Pump Circuit

mismatch, whenever the PLL is in steady state (equal number of UP and DN signals), a positive charge is injected into the output of the CP on every cycle of the reference. Since a type-2 PLLs contain an integrator in its loop filter, most often, the capacitor or equivalent integration constant will convert these pulses to a fix voltage level, thereby producing an offset. In steady state, the PLL loop will try to compensate for this offset by making the VCO lead the reference by an amount such that the on time of the DN signal is high enough to completely cancel this charge. As explained in reference [5], the extra output voltage for a fraction of period can be calculated by integrating this triangular wave shaped function, and taking its average. The injected mismatch period is effective for the reset delay of the PFD, while the time the PLL compensates for the DN signal is proportional to the time the sinking current takes to cancel this charge. Then, the total introduced voltage is given by:

$$V_{extra} = \frac{\Delta t_{PFD}^2 \cdot \Delta I}{C_{coeff_int} \cdot T_{ref}} \left(1 + \frac{\Delta I}{I_{CP}} \right) \quad (4.2)$$

In this equation, C_{coeff_int} corresponds to the integration coefficient of the loop filter, and Δt_{PFD} represents the PFD reset delay. Hence, a higher integration coefficient in the loop

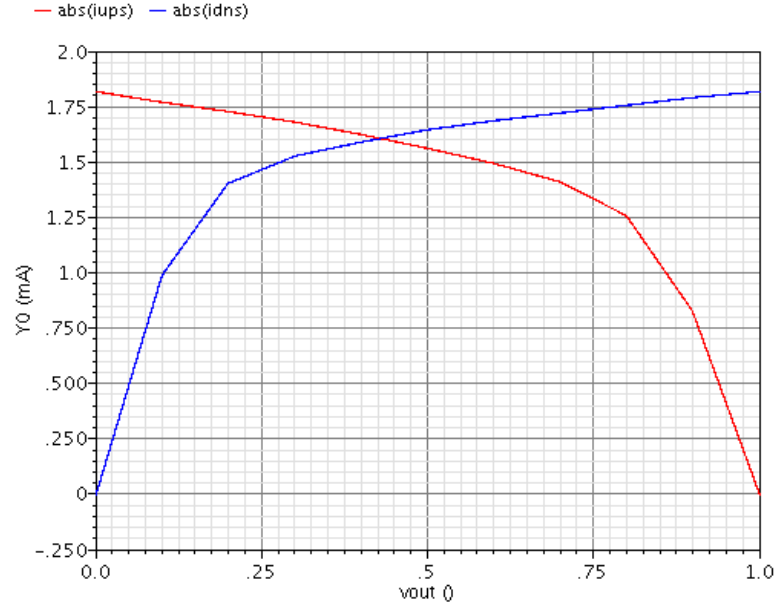


Fig. 4.9 CP Source and Sinking Currents vs Output DC Voltage

filter along with a slower clock reference and increased current charge pump, leads to a lower voltage induced by the current mismatches of order ΔI . In commercial products, it is often mentioned that a current mismatch of the order of 10% can be present. This voltage gets up converted to frequencies of $\omega_0 - \omega_{REF}$ and $\omega_0 + \omega_{REF}$ with amplitude of V_{extra} times the gain of the VCO (K_{VCO}), leading to spurs that can affect VCO performances.

4.2.4 Phase Noise

In addition to the spurs caused by mismatches, a CP also injects noise into the system. The majority of the noise generated by a CP is due to thermal noise. This noise (current), as seen previously for the VCO design, is proportional to the transistor length and inversely proportional to its width, hence advocating the use of large transistor widths to lower its noise. However, this would lead to slower CP operation. In addition to thermal noise, flicker noise also contributes to noise in CP circuits. Flicker noise for CMOS transistors was previously described in Section 3.2.2 using Eqn. 3.22. As was seen, flicker noise decreases with increased transistor width and length suggesting a higher length will decrease $1/f$ noise. Hence a treadoff exists between the thermal and flick noise reduction.

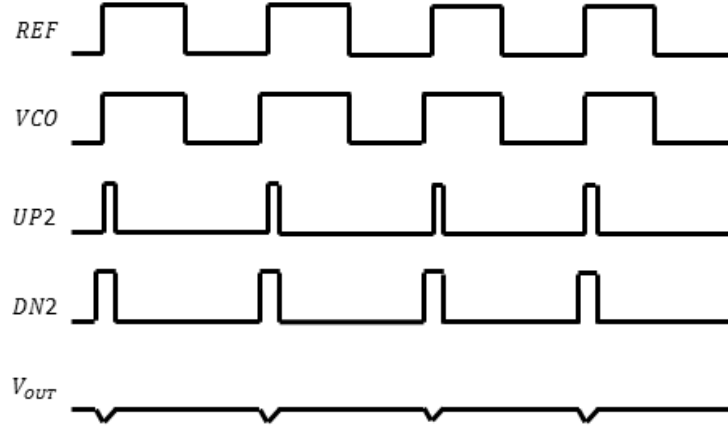


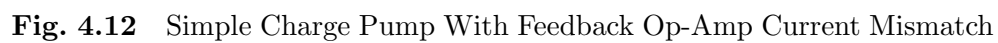
Fig. 4.10 PFD Outputs And CP Mismatch Effects On Vout in Steady State

The noise in a CP is cyclostationary, meaning it is periodic with period T_{REF} and it is injected into the PLL only when the UP or DN signals are on. A smaller reference frequency will result in a lower noise level. Or, from another point of view, the use of frequency hopping will result in higher noise levels. This noise is usually referred to the input of the PFD by integrating it first over the period of the reference frequency and dividing it by the gain of the PFD and CP block. Hence, the higher the current level used in the CP, the lower the noise level.

4.2.5 High-Performance Charge Pump Topologies

Charge Pump With Single Op-Amp

As previously seen, the design of the charge pump is crucial in lowering output spurs, injected noise and to maintain a desired closed-loop phase-noise transfer function for the PLL. In order to alleviate the dependence of the CP currents on the output voltage, another topology for the CP has been proposed in [7]. This topology is shown in Figure 4.11.



This topology is identical to the previous one shown in Figure 4.8 with the exception of the addition of an opamp that connects node V_X to node V_{B2} . This op-amp tries to force V_{OUT} to be equal V_X . This op-amp compensates the effects of an increase of V_{DS} by lowering the gate voltage on transistors M_3 and M_4 . This topology was implemented in Cadence using the TSMC 65nm CMOS technology. The op-amp was included in the simulation using a Verilog-A block generated by Spectre Verilog-A macro-module. The output voltage of the CP was varied and the sinking and sourcing currents extracted. Figure 4.12 shows the dependence on the source/sink current as a function of the output CP node voltage. Compared with the simpler topology, this circuit offers a greater voltage independence. The op-amp of the CP needs to be properly designed in order to be able to react to rising signals at the input of the CP.

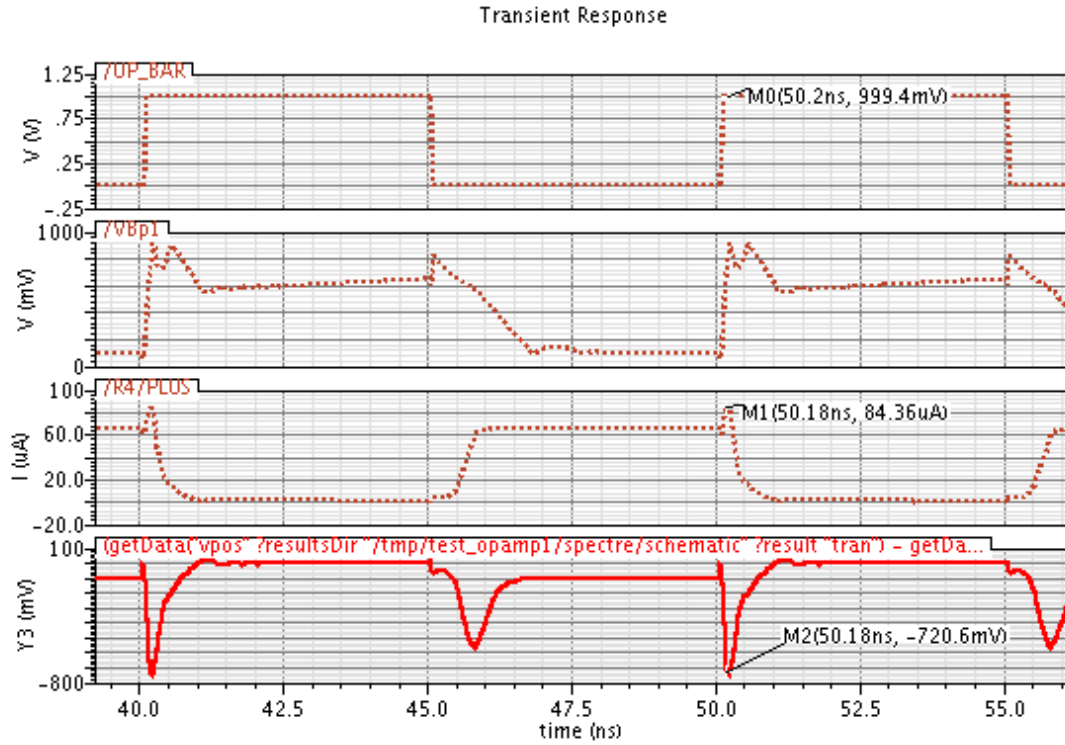


Fig. 4.13 CP With 100 MHz BW Op-Amp With 15 k Ω Load

Top trace: PFD $\overline{UP}$ Signal, Second trace: V_{B2} Signal

Third trace: Resistor Load Current, Bottom trace: $V_X - V_{OUT}$ Signal

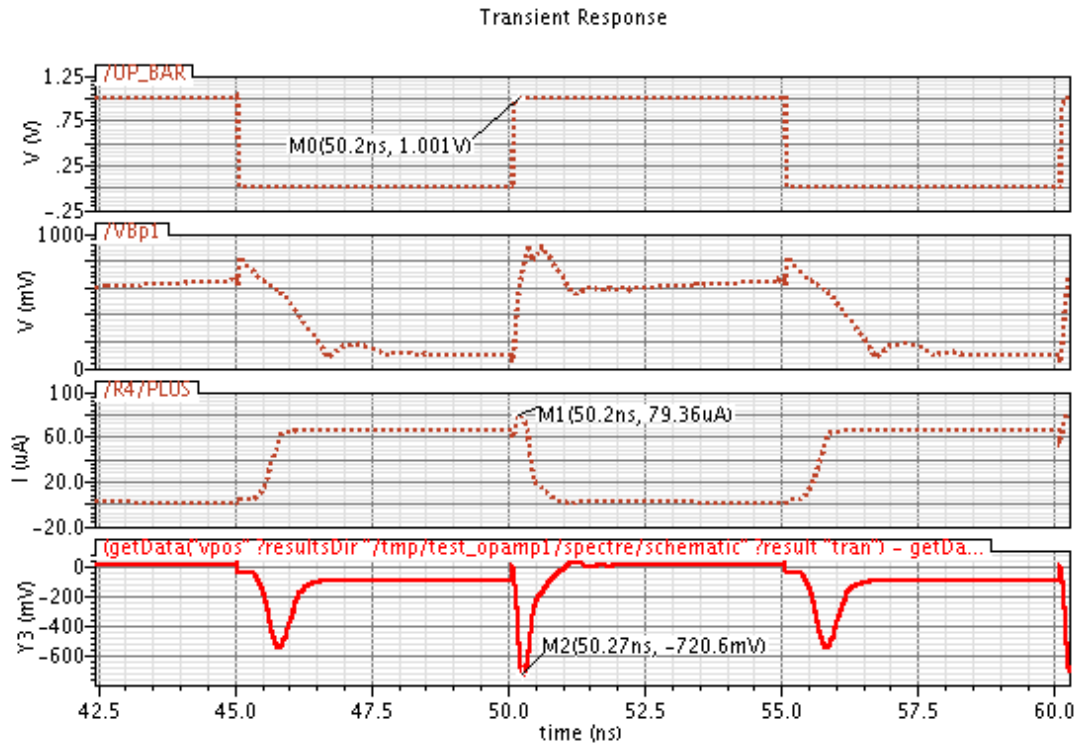


Fig. 4.14 CP With 50 MHz BW Op-Amp With 15 k Ω Load

Top trace: PFD $\overline{UP}$ Signal, Second trace: V_{B2} Signal

Third trace: Resistor Load Current, Bottom trace: $V_X - V_{OUT}$ Signal

In order to illustrate the effects that the op amp can have on the CP, a simulation was performed using an op amp with a DC gain of 60 dB, a bandwidth of 100 MHz and a slew rate set to 1000 MV/sec. The output resistance of the opamp was set to 10 k Ω and its input impedance set to 4 k Ω . Figure 4.13 shows the output response for a resistive load of 15 k Ω . This figure shows the $\overline{UP}$ from the PFD output, the opamp output, the current through the 15 k Ω load resistor as well as the input difference at opamp. As this plot shows, as the PMOS is turned on, some charges flow through the resistor, which in turn decreases the difference at the opamp input. The opamp reacts by turning on the PMOS current source. However, the turn on is slower and hence the potential at opamp's negative inputs increases. Slowly, the opamp catches up and a linear transition for the rising edge current occurs. When the UP signal goes low, or the $\overline{UP}$ goes high, charge sharing is most noticeable. During this transition extra current is injected from that PMOS capacitors. This in turn increases the current for a brief amount of time. For this particular case, the DN

signal should have been on for a slightly longer time after the UP signal went down. This would have provided a low impedance path to the charges.

A second simulation was performed, but this time with an op amp having a 50 MHz 3-dB bandwidth. All other op-amp parameters remain the same. The results are shown in Figure 4.14. As is evident, the op-amp output does not track the event changes correctly. Furthermore, a delay is noticeable between the time the maximum output peak current occurs and the peak opamp output voltage.

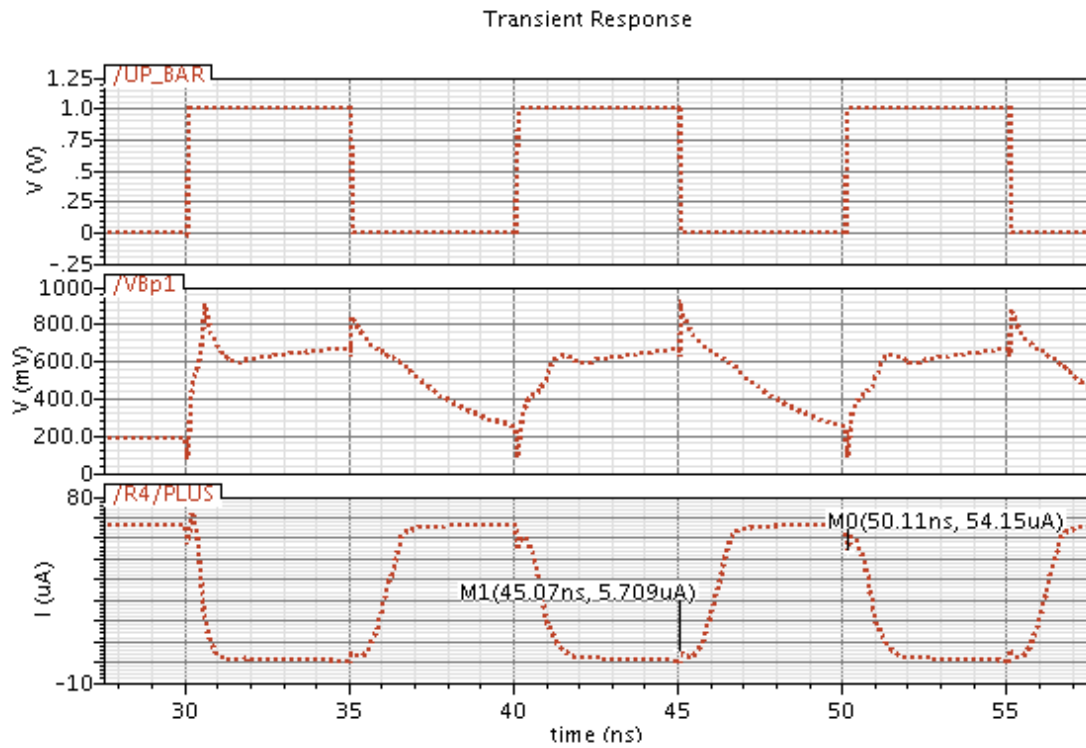


Fig. 4.15 CP With 100 MHz BW Op-Amp With 15 k Ω Load and Slew Rate of 200 MV/s Top trace: PFD $\overline{UP}$ Signal, Middle trace: V_{B2} Signal Bottom trace: Resistor Load Current

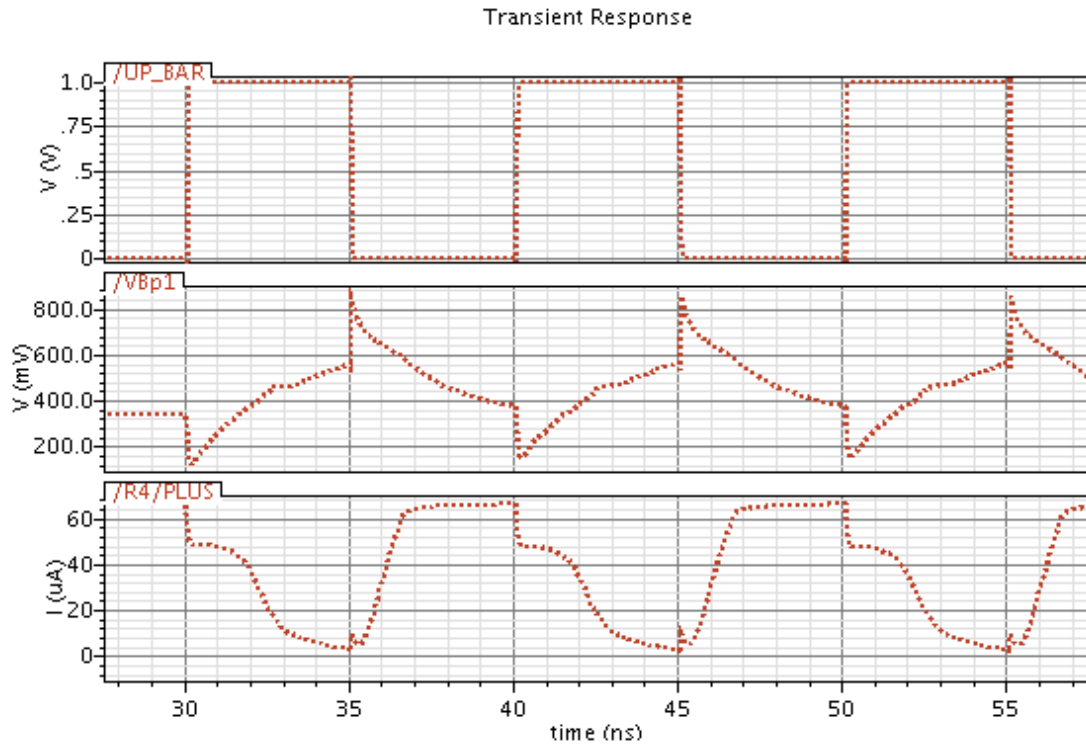


Fig. 4.16 CP With 100 MHz BW Op-Amp With 15 k Ω Load and Slew Rate of 100 MV/s *Top trace: PFD $\overline{UP}$ Signal, Middle trace: V_{B2} Signal*
Bottom trace: Resistor Load Current

The next important thing to consider for the opamp is the slew rate effect on the output current. Figure 4.15 shows the effect of an output slew rate of 200 MV/s. In this figure, the peaking of the output load current has decreased. This is because the op-amp output increases at slower rate, allowing the PMOS current to settle as the opamp voltage increases. However, lowering the slew rate to 100 MV/s results in a slower response and a distorted output load current as Figure 4.16.

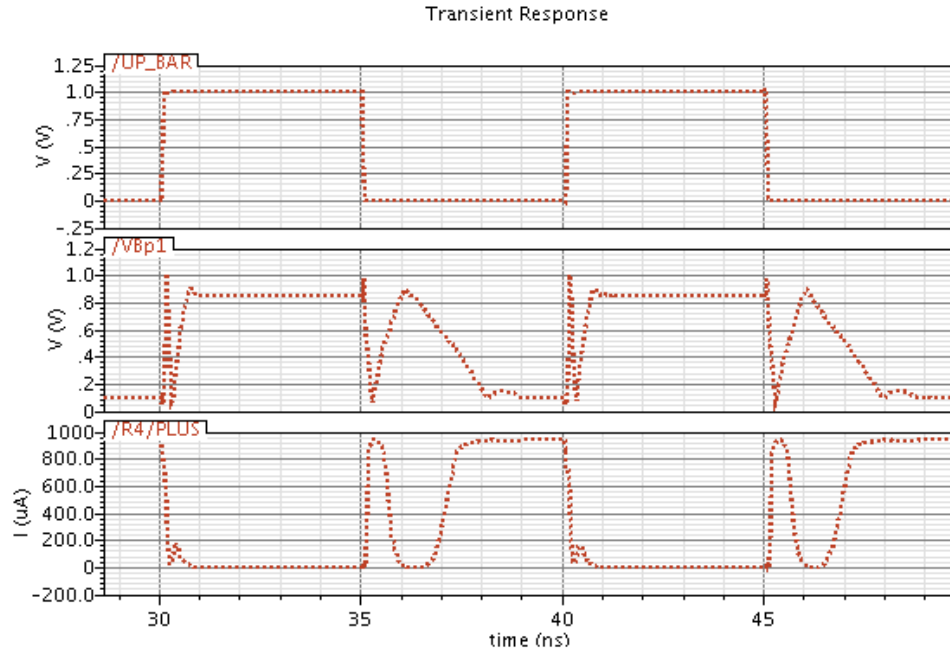


Fig. 4.17 CP With 100 MHz BW Op-Amp With 1 k Ω Load and Slew Rate of 1000 MV/s *Top trace: PFD $\overline{UP}$ Signal, Middle trace: V_{B2} Signal*
Bottom trace: Resistor Load Current

Finally, the output load also needs to be considered in the design of the charge pump. A smaller load represents a low impedance path to ground for the charges due to the charge sharing phenomenon. In Figure 4.17, the load was set to 1 k Ω . When signal $\overline{UP}$ rises, charges rapidly dissipate through the load resistor, hence the small reset delay for the PFD is enough to absorb those charges to ground. However, more charges flow through this path. Meanwhile, when signal $\overline{UP}$ falls, the DN signal being off, any extra charge rapidly flows through the low impedance, since op-amp output is low. The PMOS current source is completely turned on, and current builds up rapidly. This in turn, creates a negative potential difference at the op-amp's output that counteracts this increased in current by shutting off the PMOS transistor. Since the PMOS source is now turned off, no current flows, and the op-amp re-adjusts its output value accordingly.

Complementary Charge Pump With Two Op-Amps

In this subsection, a slightly more complex charge pump [1] is to be described. This architecture permits a wider operating output voltage and better linearity for the output

The circuit of Figure 4.18 was implemented in a 65 nm CMOS technology from TSMC. The power supply was set to 1 V. Table 4.2.5 reports the transistors width and length for this architecture. The transistors were sized using standard widths for all PMOS and two different widths for the NMOS transistors. This permits interdigitation of the circuit in layout and hence a better matching for critical transistors such as those involved in the current source mirroring or those sharing the same gate voltage. Common centroid geometries could also be employed but the difference in the number of fingers required additional of dummy transistors and, hence, would increase the complexity of layout. Figure 4.19 shows sourcing currents i_{pn1} together with i_{pn2} along with sinking currents i_{dn1} together with i_{dn2} as a function of the output voltage.

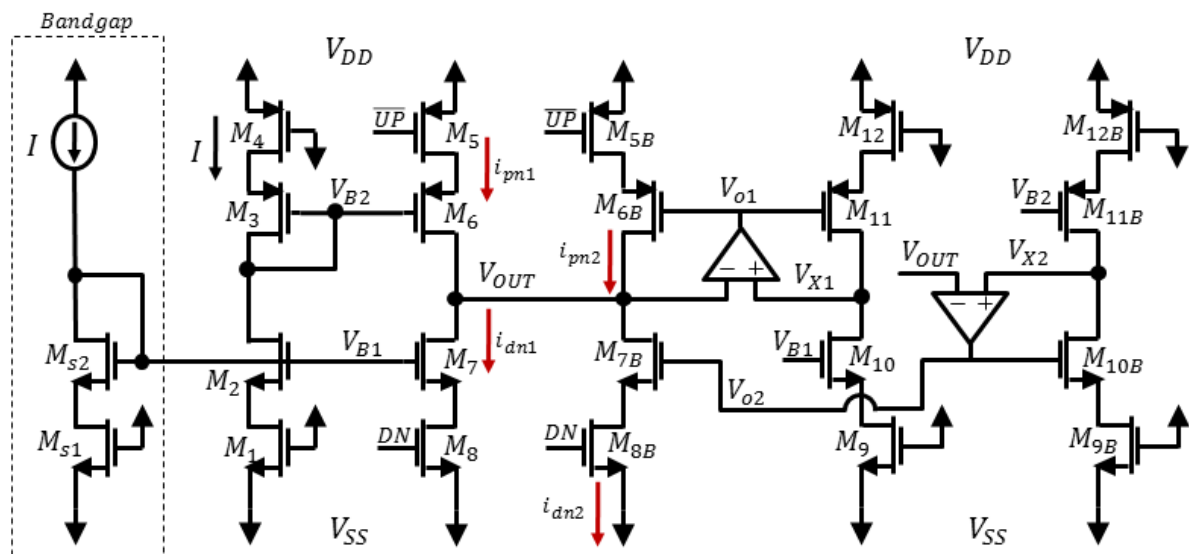


Fig. 4.18 Complementary CP Circuit From Reference [1]

Table 4.1 65 nm Charge Pump Circuit Parameters

Component	Width/Length	Nb. Fingers
M ₁	750n/ 60n	2
M ₂	625n/120n	5
M ₃	1.25u/120n	6
M ₄	1.25u/ 60n	4
M ₅ , M _{5B}	1.25u/ 60n	12
M ₆ , M _{6B}	1.25u/120n	18
M ₇ , M _{7B}	625n/120n	13
M ₈ , M _{8B}	750n/ 60n	8
M ₉ , M _{9B}	750n/ 60n	2
M ₁₀ , M _{10B}	625n/120n	5
M ₁₁ , M _{11B}	1.25u/120n	6
M ₁₂ , M _{12B}	1.25u/ 60n	4

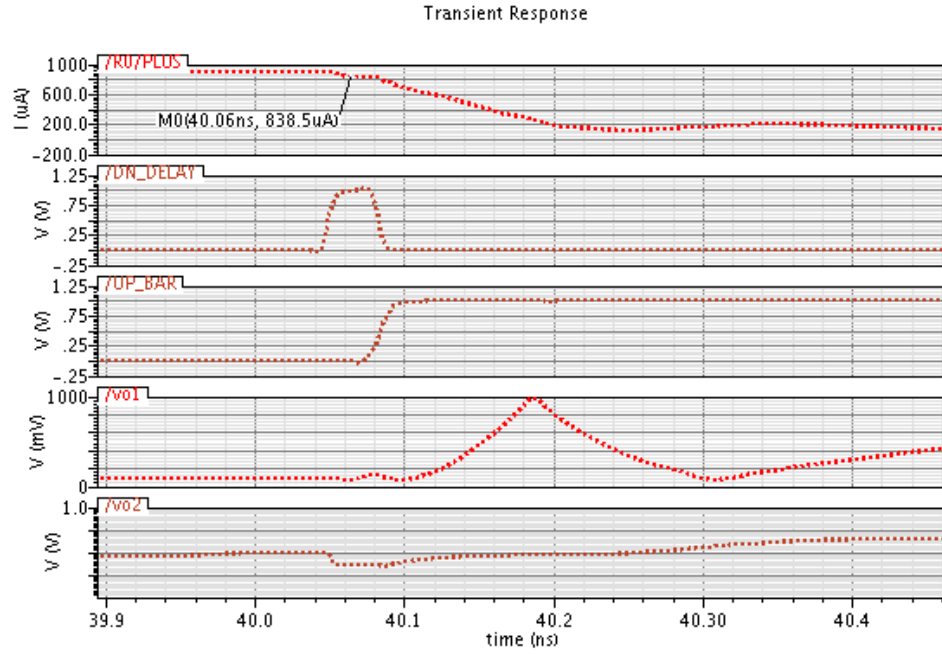


Fig. 4.20 Transient Analysis for the Complementary CP Circuit with a 1 k Ω Resistive Load *Top trace: Resistor Load Current, Second trace: PFD DN Signal Third trace: PFD $\overline{UP}$ Signal, Fourth trace: V_{o1} Signal, Bottom trace: V_{o2} Signal*

In comparison with the previous CP circuits, this architecture offers more stable output

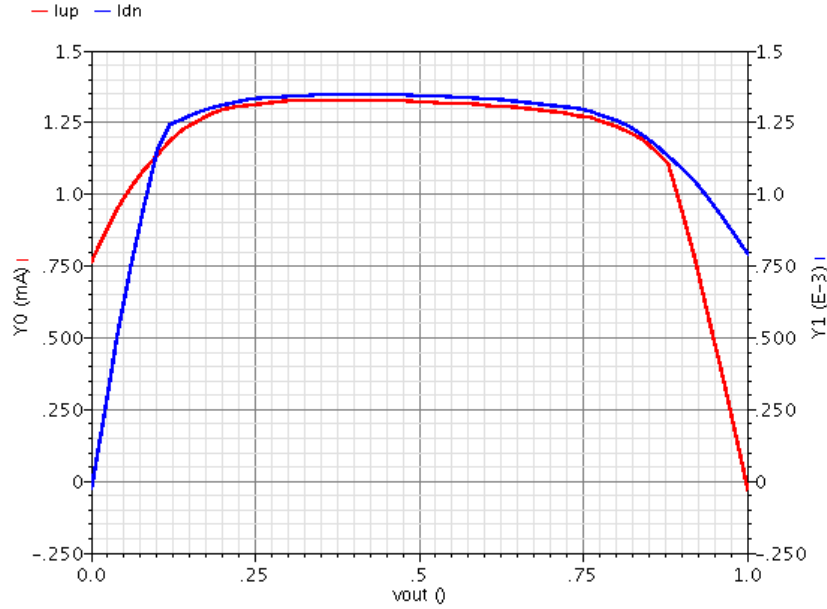


Fig. 4.19 Complementary CP Circuit Sinking and Sourcing Currents vs Output Voltage

currents. For this circuit, the difference in sinking and sourcing current is about 1.5%. As for previous architecture, a transient test was conducted for a resistive load. Figure 4.20 illustrates the waveforms for the current into the resistive load, the output signals DN and $\overline{UP}$, as well as the outputs from both opamps. Looking at the current waveform associated with the resistive load, a fix current flows when the signal UP is high. As both PFD outputs are high, a current still flows into the 1 k Ω resistive load. While the DC simulation showed currents matched, the impedance of the load needs to be consider. In this case, the resistive load has a lower impedance to ground than the sourcing current branch, and hence the current splits. In order to cancel this effect, another current sinking signal is produced and added to the load when both PFD outputs are set high. As $\overline{UP}$ signal rises, the current into the resistor decreases. However, as the voltage across the resistor decreases, V_{O1} tends towards its minimum resulting in the PMOS M_6 pair being completely turn on. This provides a low impedance path to ground for the charges trapped on these transistors and the transistor pair M_5 . This creates a small current flow into the load resistor. This phenomenon will results in an extra charge being dumped slowly into the load, which, in turn, will create output spurs, as previously mentioned.

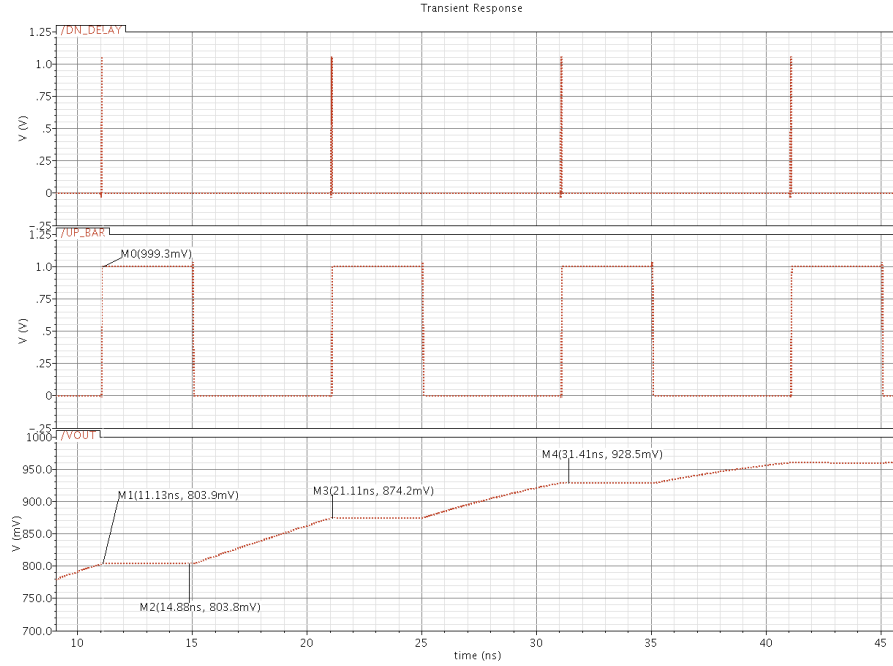


Fig. 4.21 Transient Analysis for the Complementary CP Circuit with a 100 pF Capacitive Load *Top trace: PFD DN Signal, Second trace: PFD $\overline{UP}$ Signal, Third trace: V_{OUT}*

In the next situation, the 1 k Ω resistive load was replaced by a 100 pF capacitive load. The results of a transient analysis are shown in Figure 4.21. As the plot shows, the ΔV being the results of the current integration decreases with increased output voltage at the capacitor node. From the dc characteristics, the output voltage is in the region of current mismatches, operating the CP within the PLL system in this range will results in poor phase noise, output spurs and even no locking conditions. From this plot, since the impedance of the capacitors is greater than the impedance of the sourcing branch, all current flows within the branch for the case of simultaneous high UP and DN signals. The input referred phase noise of the charge pump topologies using cascoded opamp topology for 1 ns input PFD phase offset was calculated to be about 0.6 ps after transistor sizing optimization both for the CP and the Opamp.

4.3 Extracted Layout Results Tor The PFD and CP

In the following, the simulation results for the PFD and CP of the previous section will be provided.

4.3.1 PFD Simulation Results

The layout for PFD implemented in the 65 nm CMOS process is shown in Figure 4.22. It is based on the PFD schematic of Figure 4.5. The layout is compact and measures $13.450\mu\text{m}$ by $10.10\mu\text{m}$. The reset delay for the extracted layout PFD is about 58.3837 ps; an increase of about 8 ps compared to that found from the schematic level simulation. The average DC output versus phase difference for the extracted layout circuit is shown in Figure 4.23. As shown, no dead-zone is present in the circuit. The behavior matches the schematic level simulation. The only difference seems to be that it has a lower output DC value.

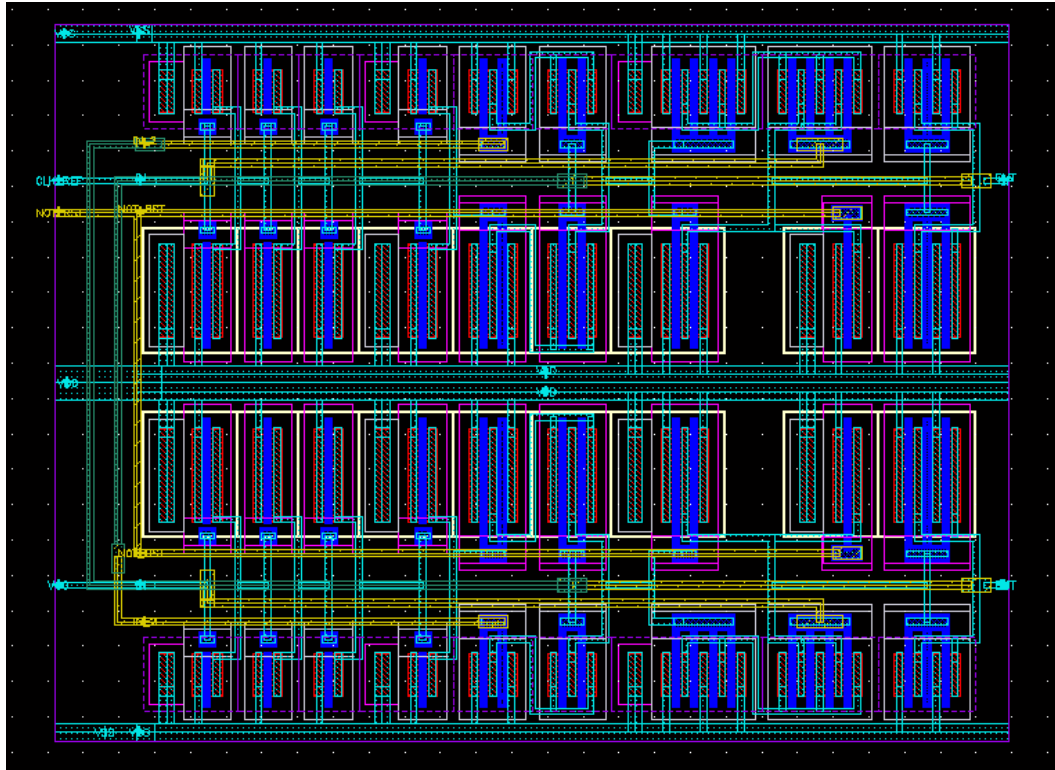


Fig. 4.22 Layout of The PFD

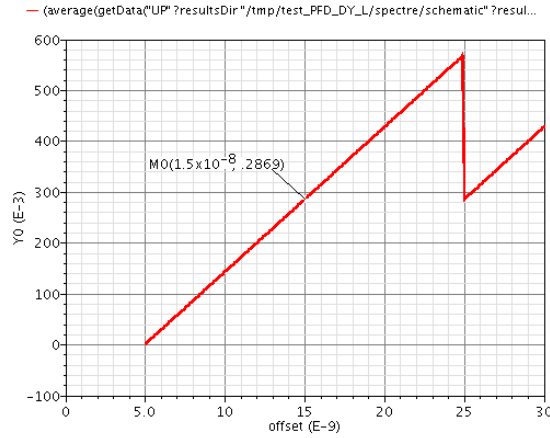


Fig. 4.23 Phase Frequency Detector Extrated Layout DC Average vs In Phase Difference

Figure 4.24 shows an expanded view of the region where the blind-zone region is expected. Compared with the schematic level simulation results, the blind-zone region is much smaller - a level of about 4 ps in duration. The reason for this is attributed to the extra delays caused by parasitic resistance and capacitance introduced by the layout.

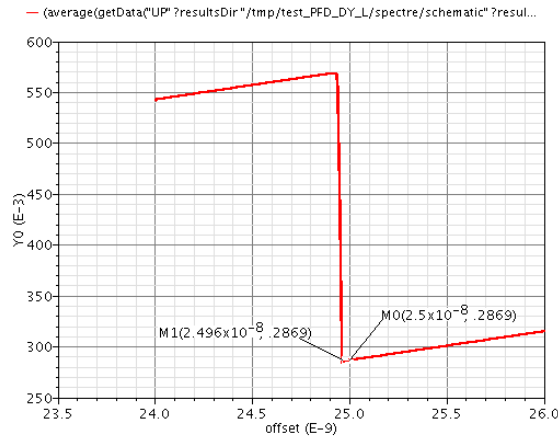


Fig. 4.24 Phase Frequency Detector Extrated Layout Blind Zone Region

4.3.2 Combined PFD and CP Simulation Results

The opamps in the CP circuit shown previously in Figure 4.18 were layout separately from the rest of the circuit in order to distribute the work. Because the opamp were not layout compact, they were not yet integrated into the whole PFD and CP component. Nonetheless, the opamps were extracted from layout in order to assert the overall extracted layout behavior of the PFD and CP circuit. Figure 4.25 shows the layout of the CP without the op-amps circuits included. The layout measures $28\mu\text{m}$ by $24\mu\text{m}$. The layout uses interdigitation in order to increase matching between critical transistors used to mirror the current within the different branches. Furthermore, the layout is made symmetric as possible and reserves the odd metal layers for horizontal connections and the even metal layers for vertical connections. This permits a more compact layout and easier overall integration.

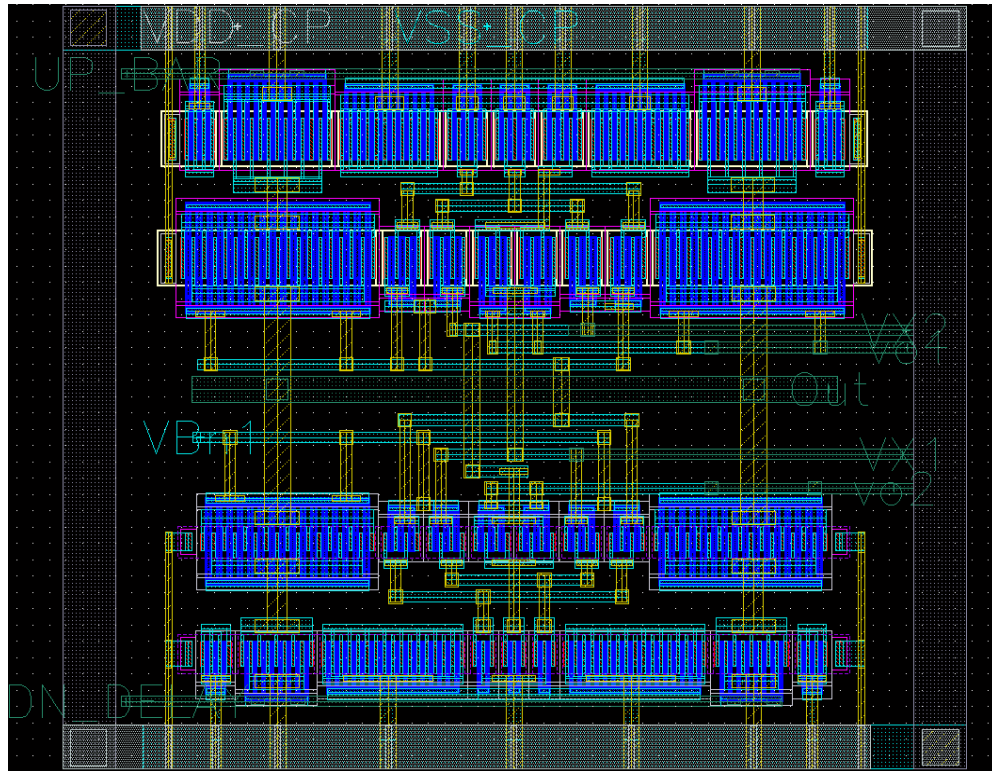


Fig. 4.25 CP Layout Without Op-Amp Circuits

The DC current characteristic versus the output CP node voltage is plotted in Figure

4.26. As shown, compared with the schematic level simulation result, the current matching is about 96.1% for the extracted layout simulation versus 98.5% for the schematic level simulation. The output voltage range for matched current within 3.9% is between 0.22 V and 0.8 V, slightly less than that which occurs for the schematic level simulation.

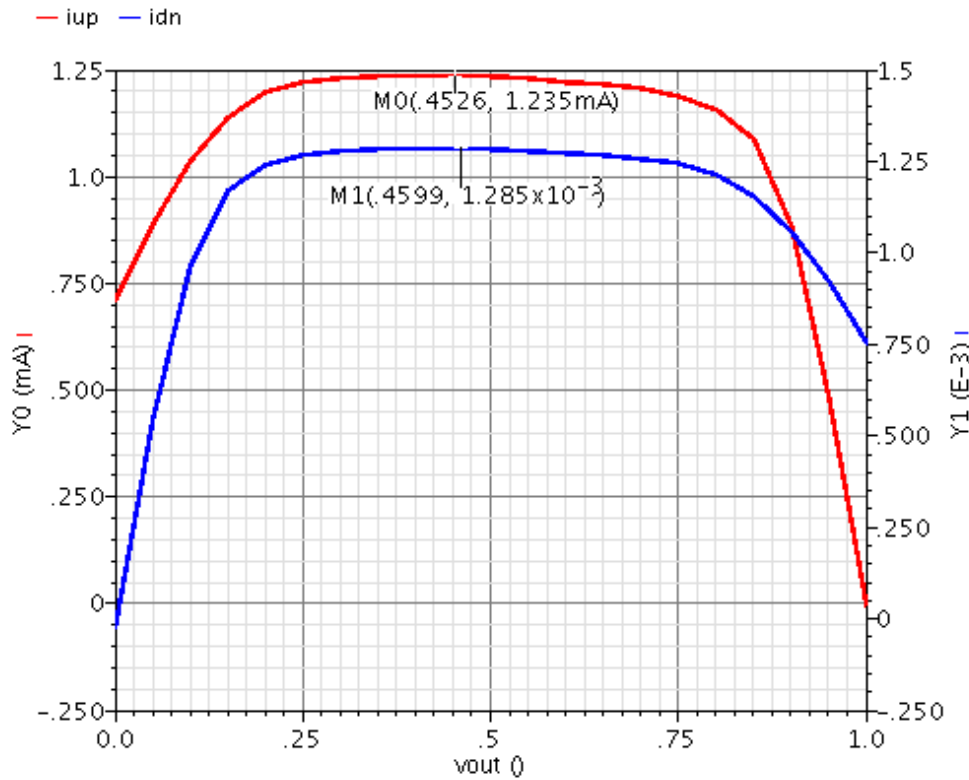


Fig. 4.26 CP DC UP and DN Current vs Vout

The transient behavior for the UP and DN currents for the extracted layout is shown in Figures 4.27 and 4.28. These results differ from that found at the schematic level. The test was performed using a DC source set to 600 mV with a 1 Ω resistance in series connected at the output node of the CP.

The input phase difference was set to 1 ns. The UP current rises more slowly than the DN current and varies between 1 mA up to 1.2525 mA, while the DN current varies from 1.2872 mA down to 1.2597 mA. The total current mismatch ranges from 0.57% and to 22%. It is also important to note the UP current drops below 0 A with the falling edge,

as the charge injection for the UP case is less (since the charges cancel) than for the DN case. For the DN case a current of magnitude 400 μA for about 26.8 ps is injected.

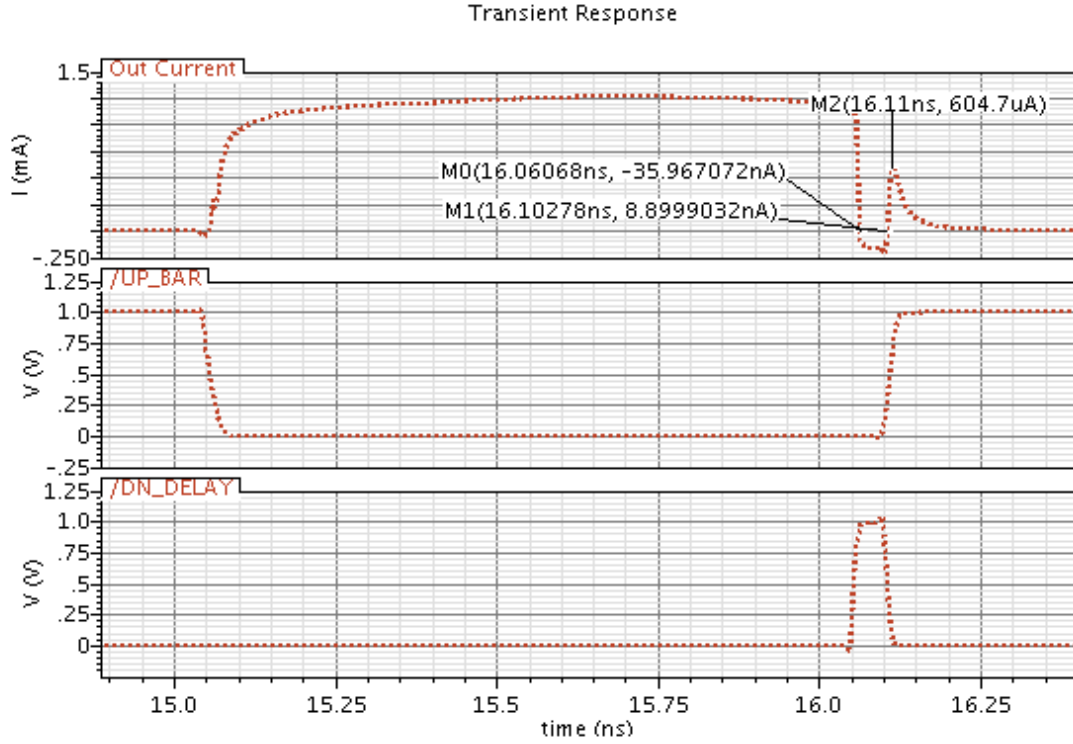


Fig. 4.27 Up Current Transient Simulation *Top trace: Output Load Current, Second trace: PFD $\overline{\text{UP}}$ Signal, Third trace: PFD DN Signal*

The input referred phase noise characteristics of the extracted layout PFD and CP is plotted in Figure 4.29. The test setup consisted of driving both PFD inputs with a phase offset of 1 ns and extracting the phase noise at the output of the CP. As seen, the phase noise is under -130 dBC/Hz starting at 10 kHz.

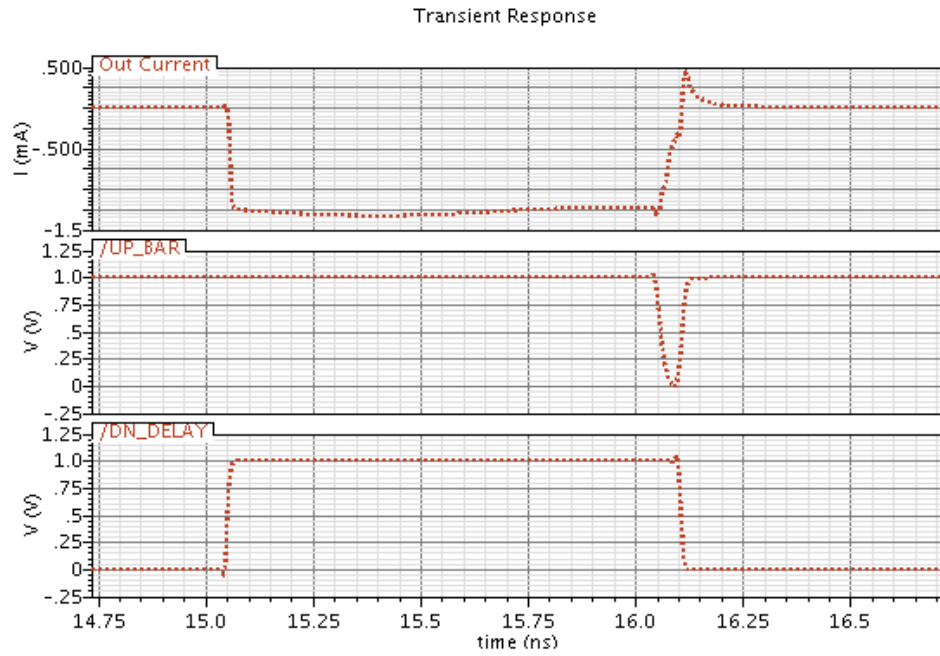


Fig. 4.28 Down Current Transient Simulation *Top trace: Output Load Current, Second trace: PFD $\overline{UP}$ Signal, Third trace: PFD DN Signal*

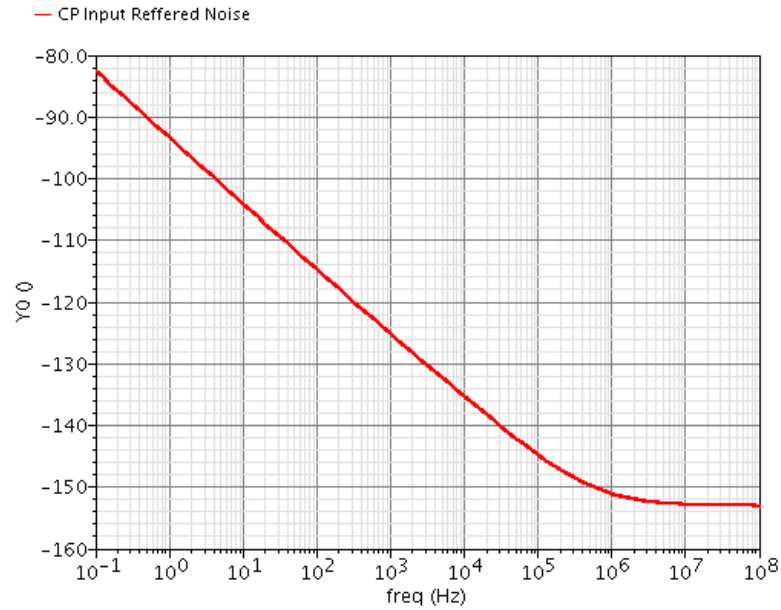


Fig. 4.29 Extracted Layout PFD and CP Input Referred Phase Noise vs Frequency

4.4 PFD and CP Verilog-A Model Implementation

The PFD and CP circuit was modeled using Verilog-A. The model is based on the one provided in [13] but with increased complexity that reflects the behavior of the current with respect to the CP output voltage as well as the extra injected charge. Furthermore, the way jitter is generated by the CP in [13] using a transition function with a varied delay is not valid here, as it introduces spikes in the output phase due to simulation artifacts. Instead, the phase noise generated by the CP is mapped to the input of the reference signal even though it should be present when the PLL is out of lock. However, since a CP contains mismatches, it is fair to assume it will always be active in the PLL even in locking state. The Verilog-A file of this PFD and CP is given below:

```
// VerilogA for High_Level_PLL, CP_Non_Ideal_2, veriloga

`include "constants.vams"
`include "disciplines.vams"

module CP_Non_Ideal_2(ref,vco,out);

input ref,vco; inout out; electrical ref,vco,out;

parameter real iup_max = 1.235e-3;
parameter real idn_max = 1.285e-3;

parameter real extra_current = 600e-6; //extra injected current
parameter real charge_time = 30p; //time extra injection charge
parameter real delay_charge = 25p;

parameter integer dir = +1 from [-1:1] exclude 0; //dir 1 for pos edge
trigger
//dir -1 for neg edge
trigger

parameter real tt = 1e-14 from(0:inf);
parameter real td = 0 from[0:inf);
```

```

//parameter real jitter = 137e-12; // from [0:td/5];           //edge to
    edge jitter
parameter real jitter = 0;
parameter real ttol = 1e-15 from (0:inf];

parameter real vcp_min = 0.22; //parameter CP max up
parameter real vcp_max = 0.8; //parameter CP max dn
parameter real idn_init = 0.75e-3; //init idn current
parameter real iup_init = 0.75e-3; //init iup current

real iout;
integer seed, charge, sign;
real state, dt;
real next_time;

real delta_iup, delta_idn;
analog begin
    @(initial_step)begin
        seed = 716;
        charge = 0;
        next_time = 0;
        delta_iup = (iup_max - iup_init)/(vcp_min);
        delta_idn = (idn_max - idn_init)/(vcp_max - 1);
    end

    @(cross(V(ref),dir,ttol)) begin
        if(state >-1) state = (floor(state + 0.5) -1);
        //dt = jitter*$rdist_normal(seed,0,1);
        if(state ==0) begin
            next_time = $abstime + charge_time;
            charge = 1;
            sign = -1;
        end
        if(V(out)<vcp_min) begin
            iout = delta_iup*V(out) + iup_init;
        end
        else if(V(out) > vcp_max) begin
            iout = iup_max/(vcp_max - 1)*(V(out)-1);
        end
    end
end

```

```

        else      iout = iup_max;

    end

    @(cross(V(vco), dir,ttol)) begin
        if(state<(1 )) state = (state+1);
        //dt = jitter*$rdist_normal(seed,0,1);
        if(state ==0) begin
            next_time = $abstime + charge_time;
            charge = 1;
            sign = -1;
        end
        if(V(out) > vcp_max) begin
            iout = delta_idn*(V(out)-1) +idn_init;
        end
        else if(V(out) < vcp_min) begin
            iout = idn_max/vcp_min*V(out);
        end
        else      iout = idn_max;
    end

    I(out) <+ transition(iout*state,td+dt,tt,tt,ttol);

    @(timer(next_time))begin
        charge = 0;
        sign = 0;
    end

    I(out) <+ transition(charge*extra_current*sign,delay_charge,tt,tt
    );
    $bound_step(1/(100*100e6));
end

endmodule

```

In order to test the Verilog-A model, the DC characteristic of the UP and DN currents were observed and plotted. These can be seen in Figure 4.30. When compared to the signals derived from the extracted layout, the result match quite well. Finally, the transient analysis of the DN current is shown next in Figure 4.31. This illustrates the extra charge injected that is included in the PFD and CP Verilog-A model.

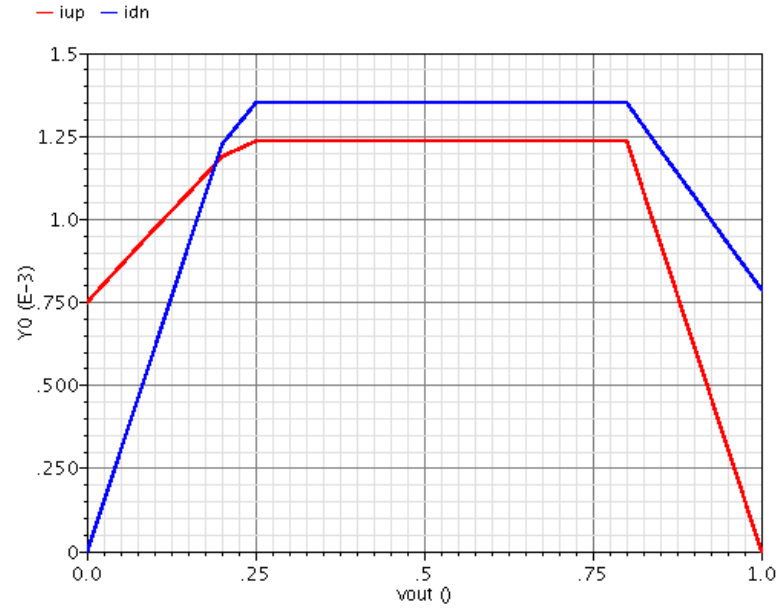


Fig. 4.30 Verilog-A PFD and CP UP and DN Currents vs Vout

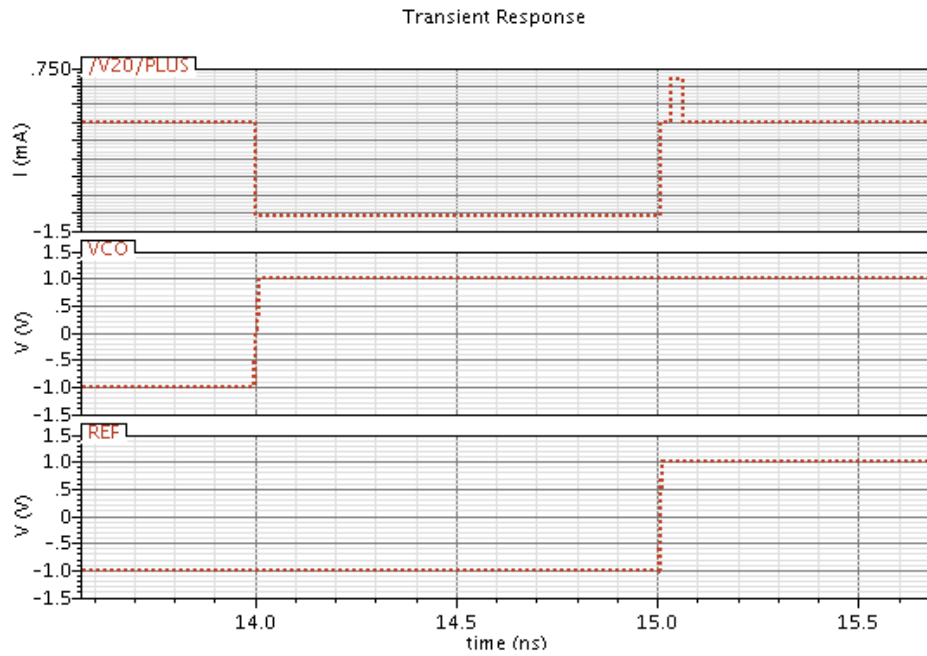


Fig. 4.31 Verilog-A PFD and CP DN Currents Transient

The jitter generation of the PFD and CP in ref. [13] in the Verilog-A model introduces numerical artifacts in the output phase noise. To eliminate this issue, the CP and PFD noise is added as synchronous jitter to the reference oscillator. This might not reflect the ideal case, since the PFD and CP only inject noise when active, however, since there is a mismatch present as well as a charge injection, it fair to assume the is active each cycle. From the previous extracted layout results of the phase noise, the synchronous jitter is about 1ps. The following code implements the fixed reference oscillator with the synchronous jitter of the PFD and CP embedded:

```
'include "constants.vams"
'include "disciplines.vams"

module Fixed_OSC2(out);

output out;
electrical out;

parameter real freq = 100e6 from (0:inf);
parameter real Vlo = -1, Vhi = 1;
parameter real tt = 1e-13 from (0:inf);
//parameter real jitter = 137p from [0:0.1/freq);
parameter real jitter = 1p;

integer n,seed;
real next,dt;

analog begin
    @(initial_step)begin
        seed = 286;
        next = 0.5/freq + $abstime;
    end
    @(timer(next+dt)) begin
        n = !n;
        dt = jitter*$rdist_normal(seed,0,1);
        next = next + 0.5/freq;
    end
    V(out) <+ transition(n ? Vhi: Vlo,0,tt);
end
endmodule
```

4.5 Summary

In this chapter, the design, layout and simulation of the PFD and CP was described. These circuit were designed for implementation in a 65 nm CMOS process made available through TSMC. Much thought went into reducing their output noise contributions. Finally, a Verilog-A model of the PFD and CP was provided.

Chapter 5

Loop Filter Design

In this chapter, the loop filter of the PLL is to be designed based on the phase noise requirements of the PLL. A Verilog-A model of the loop filter will be provided, so that all the PLL components can be combined and simulated in a fast and efficient manner.

5.1 Synthesis of The Loop Filter From The PLL Phase Noise Requirements

In this thesis, the PLL is required to operate over a frequency range of 30 - 40 GHz with a phase noise of at least -80 dBc/Hz from a 10 kHz offset. Moreover, any spurs at the output should be less than -70 dB. Of course, like all designs today, low power operation is essential. The reference frequency is assumed to be derived from an off-chip crystal oscillator with extremely low phase noise (much lower than that called for by the PLL). This implies that the dominant noise source will be that coming from the VCO. It should also be noted that the noise characteristics of the transistors made available from the 65 nm CMOS process from TSMC have only been characterized up to 10 GHz. This implies that some element of overdesign must be included in the design of the loop filter to ensure the noise uncertainty above 10 GHz is accounted for.

In Chapter 3, the $1/f^3$ phase noise corner of the extracted layout of the VCO implemented in the 65 nm CMOS is at about 1.47 MHz. At this offset frequency, the phase noise is about -97 dBc/Hz. Assuming an additional noise penalty of 10 dB in the frequency range of 30 - 40 GHz, it is reasonable to assume that the actual VCO will have a phase noise of

about -87 dBc/Hz at a 1.47 MHz offset. In a similar vein, it is reasonable to assume that the expected VCO phase noise at a 10 kHz offset will be about -50 dBc/Hz. Given that our PLL specification requires a phase noise of -80 dBc/Hz at a 10 kHz offset, the PLL must be capable of attenuating the noise from the VCO anywhere between 30 - 40 dB at a 10 kHz offset from the PLL output frequency.

From Chapter 2, the transfer function from the VCO output to the PLL output is described by the following high-pass function

$$\frac{\varphi_{out}}{\varphi_{VCO_noise}} = \frac{1}{1 + I \frac{K_{VCO}}{N s} F(s)} \quad (5.1)$$

It is clear from above discussion that the magnitude of the above HP transfer function must have a 3-dB bandwidth of about 2 MHz and provide at least 40 dB of attenuation at 10 kHz. For a type-2 PLL, the magnitude of $\frac{\varphi_{out}}{\varphi_{VCO_noise}}$ consists of a single zero at DC and a set of poles that cluster around the 3-dB frequency of 2 MHz. Hence there is enough attenuation to satisfy a 40 dB change in attenuation from 10 kHz to 2 MHz.

Returning to the PLL synthesis method described in Chapter 2, the first step is to select the transfer function of the PLL in closed-loop such that it has a 3-dB bandwidth of 2 MHz. For the problem at hand, we'll select a 5th-order Butterworth filter, together with the introduction of an additional zero and pole to satisfy the type-2 realizability constraint, leading to the following PLL transfer function:

$$H_{new}(s) = \frac{2.494e29s + 3.134e35}{5.383e-07s^6 + 22.89s^5 + 4.857e08s^4 + 6.42e15s^3 + 5.383e22s^2 + 2.494e29s + 3.134e35} \quad (5.2)$$

The loop filter $F(s)$ would then be computed using Eqn. 2.13 according to

$$F(s) = \frac{N s}{I K_{VCO}} \frac{H_{new}(s)}{H_{new}(s) - 1} \quad (5.3)$$

In this expression, I is regarded as a constant and independent of the PLL operating conditions; however, N and K_{VCO} depend on the frequency selection and the corresponding

control voltage. The challenge therefore is to find values for N and K_{VCO} such that over the range of possible values the loop filter $F(s)$ leads to a PLL that meets all of its closed-loop requirement, i.e., frequency selection, phase noise, settling time, etc.

Table 5.1 PLL Operating Parameters For Three Different Conditions

Mode of Operation	Component	Characteristic	Value
max	VCO	K_{VCO}	2.515 GHz/V
	f_{out}	F_{max}	35.28 GHz
	Divider	N	352
mid	VCO	K_{VCO}	1.5153 GHz/V
	f_{out}	F_{mid}	34.57 GHz
	Divider	N	343
min	VCO	K_{VCO}	515.6 MHz/V
	f_{out}	F_{min}	33.86 GHz
	Divider	N	339

As an example, Table 5.1 lists three different operating conditions for the PLL involving N , K_{VCO} and f_{out} assuming a reference frequency of about 100 MHz. The CP current bias level is assumed to be $I = 1.235$ mA. Let us first calculate the loop filter $F(s)$ according to Eqn. 5.3 using the maximum operating conditions. In this case, the loop-filter $F(s)$ becomes

$$F_{max}(s) = \frac{8.528e31s + 1.072e38}{1.672s^5 + 7.109e07s^4 + 1.509e15s^3 + 1.994e22s^2 + 1.672e29s} \quad (5.4)$$

Substituting back into Eqn. 5.1, the following VCO noise transfer function, $\frac{\varphi_{out}}{\varphi_{VCO_noise}}$, can be determined,

$$\frac{\varphi_{out}}{\varphi_{VCO_noise}}(s) = \frac{571.8s^6 + 2.431e10s^5 + 5.16e17s^4 + 6.819e24s^3 + 5.718e31s^2}{571.8s^6 + 2.431e10s^5 + 5.16e17s^4 + 6.819e24s^3 + 5.718e31s^2 + 2.649e38s + 3.329e44} \quad (5.5)$$

Figure 5.1 shows the magnitude response of $\varphi_{out}/\varphi_{out_noise}$ for the overall PLL under maximum operating conditions. It is interesting to compare the magnitude response of the VCO noise transfer function for the PLL when its operating conditions have dropped to those given by the minimum conditions in Table 5.1. This plot is also provided in Figure

5.1. As is evident, as the VCO gain changes, with the same loop filter synthesized under maximum K_{VCO} , the bandwidth of the VCO noise magnitude response decreases by about a factor 10 and the attenuation at 10 kHz increases by about 20 dB. In essence, the PLL would not satisfy its phase noise requirements.

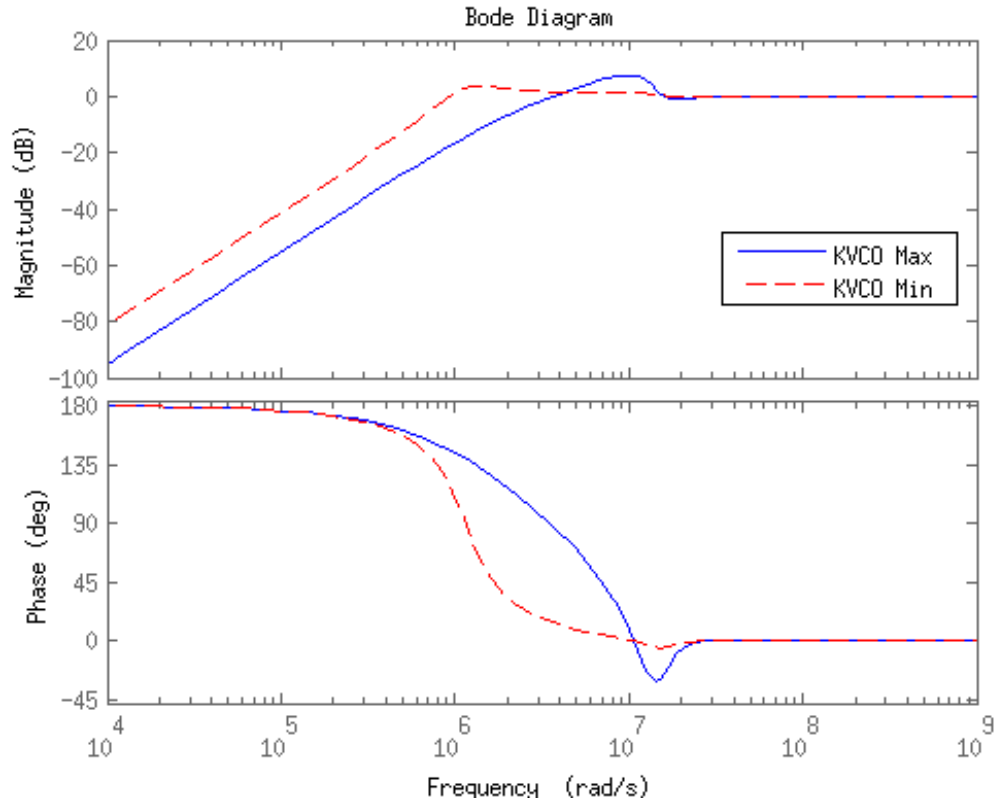


Fig. 5.1 A Comparison Of The Magnitude/Phase Response of VCO Noise Transfer Response Under Maximum Loop Gain Conditions

Reversing the situation and synthesizing the loop filter using the minimum operating conditions shown in Table 5.1 does not provide a viable solution either. This is evident from Figure 5.2 where the bandwidth of the VCO noise transfer function changes dramatically with the variation in the gain of the VCO.

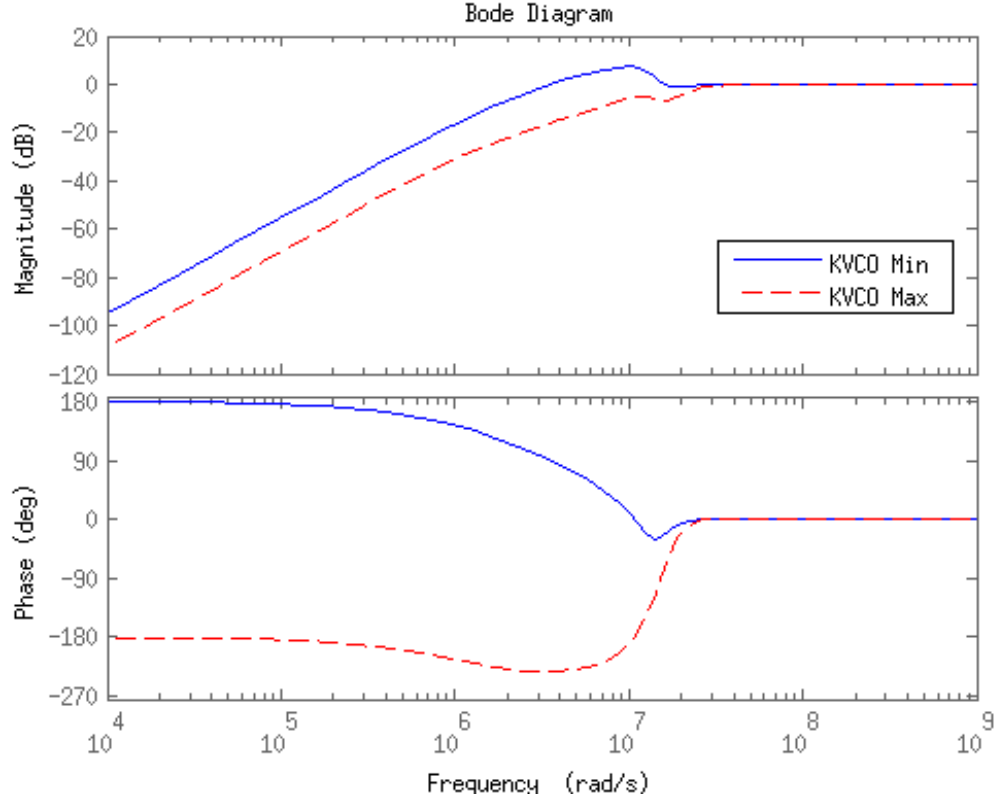


Fig. 5.2 A Comparison Of The Magnitude/Phase Response of VCO Noise Transfer Response Under Minimum Loop Gain Conditions

A third attempt was made, this time using the average or middle operating values seen list in Table 5.1 to synthesize the loop-filter $F(s)$. In this case, Figure 5.3 again shows variation in the VCO noise magnitude response over the full range of operating conditions within the PLL. Of particular interest is the fact that the 3-dB of the noise-transfer function varies widely, as well some resonance or frequency peaking is present in one of the responses.

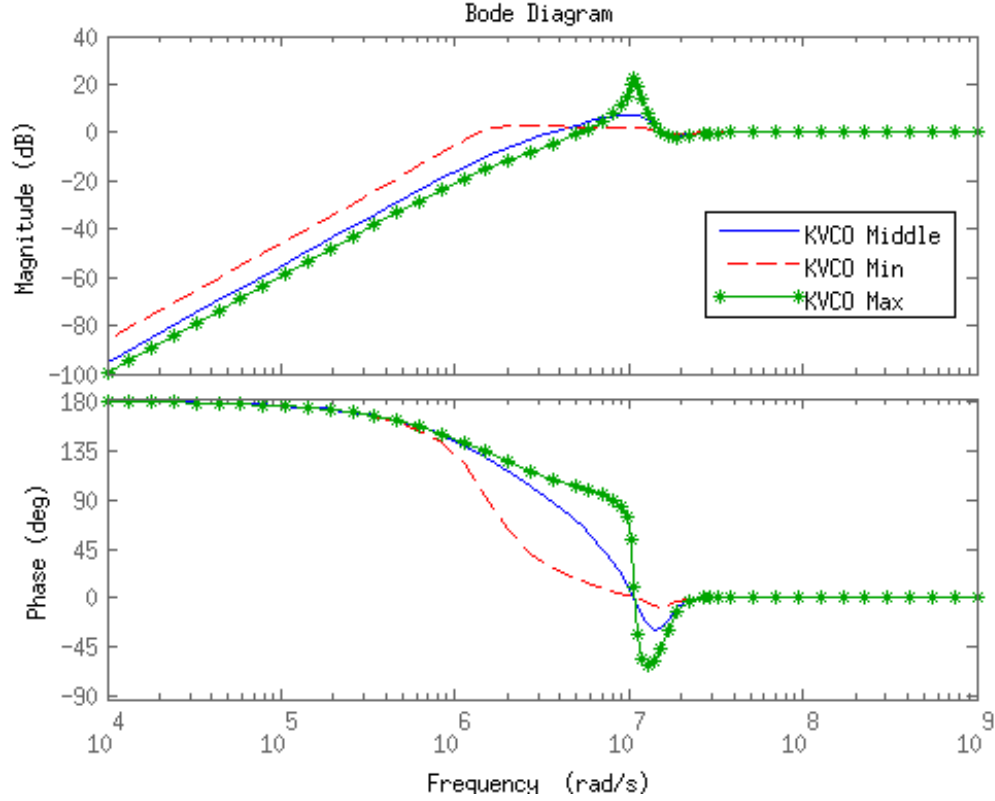


Fig. 5.3 A Comparison Of The Magnitude/Phase Response of VCO Noise Transfer Response Under Average Loop Gain Conditions

Since the noise transfer function requirements cannot be met using any particular instance of the loop-filter, it was decided to change the closed-loop transfer function of the PLL and set its 3-dB bandwidth to twice its initial value of 4 MHz. The synthesis procedure was repeated. The loop filter $F(s)$ assuming minimum VCO gain conditions was found to be

$$F(s) = \frac{1.365e33s + 3.429e39}{0.8359s^5 + 7.109e07s^4 + 3.017e15s^3 + 7.976e22s^2 + 1.337e30s} \quad (5.6)$$

The VCO noise transfer function evaluated over a range of frequencies under minimum and maximum VCO gain conditions can be seen plotted in Figure 5.4. As shown, the requirements of the PLL are met using this filter function since there is enough attenuation starting at 100 kHz offset and the peak in the response under 10 dB at 1 MHz offset.

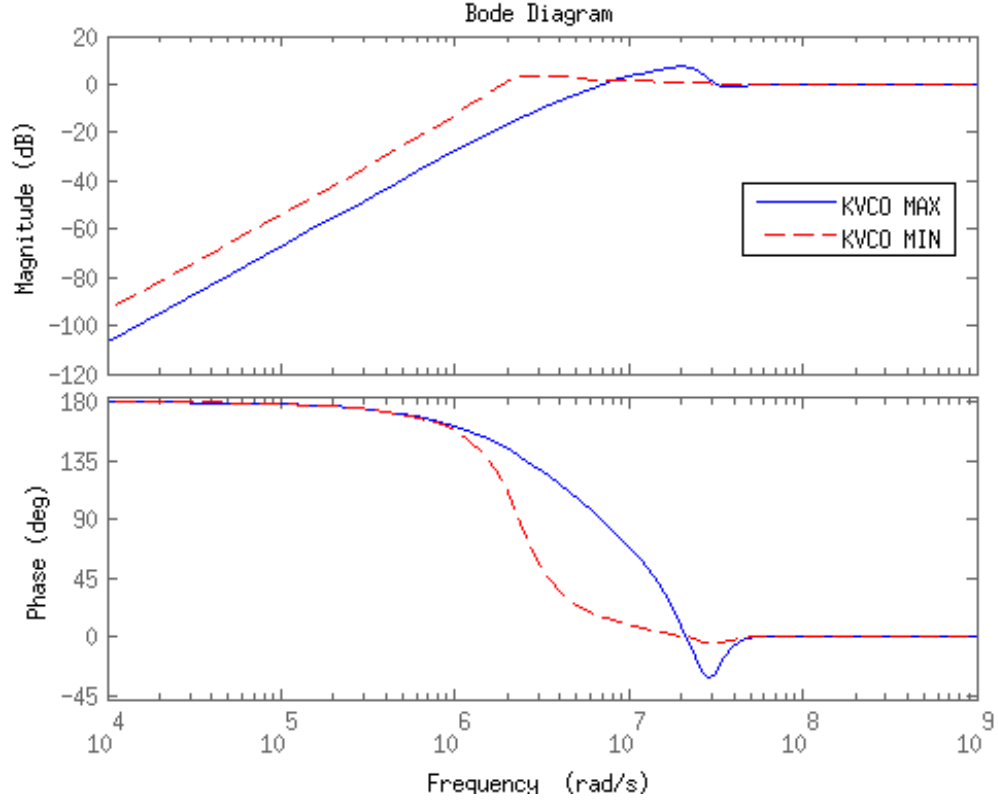


Fig. 5.4 A Comparison Of The Magnitude/Phase Response of VCO Noise Transfer Response Under Minimum Loop Gain Conditions And A 3-dB bandwidth of 4 MHz

The input-output PLL transfer function response is shown in Fig. 5.5. As seen, the magnitude response for the input reference noise, generate by the fixed oscillator, and the input referred phase noise of the PFD and CP does not get amplify to exceed the phase noise output requirements. The step response of the PLL is shown in the same Figure. The settling time is about $1.51 \mu\text{s}$. Since no requirements have been provided on the settling time, this metric will not be discuss further.

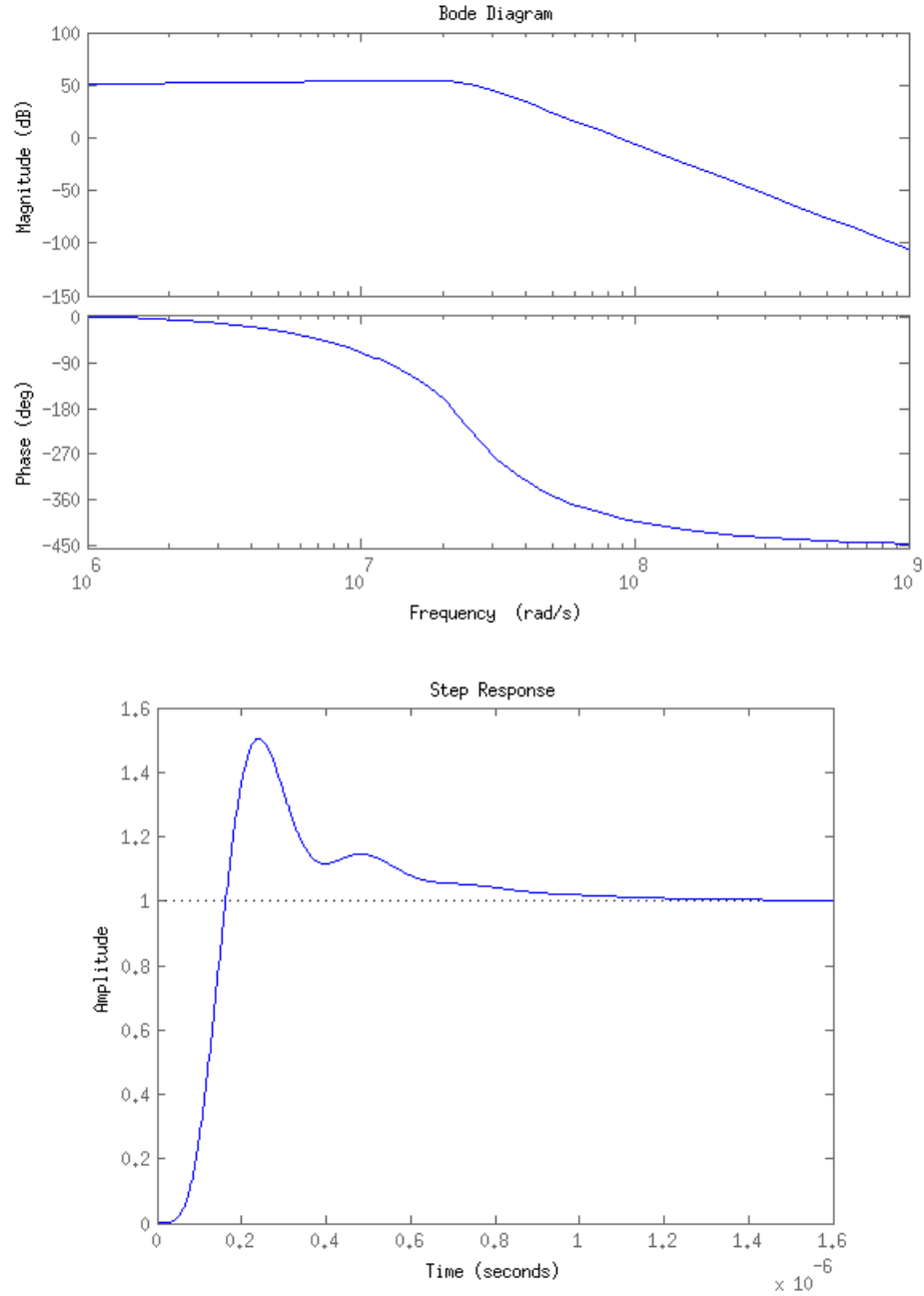


Fig. 5.5 (a) Magnitude and Phase Response Of The Input-Output Closed-Loop Response Of The PLL With A 4 MHz Bandwidth (b) PLL Step Response

With the designed filter, the PLL will be stable within the VCO operating range.

However, it is important to mention that this filter is only valid for a specific operating condition for the VCO. The problem introduced by the VCO various operating conditions suggest that some form of tuning or adjustment be made to the loop filter such that it depends on the VCO operating conditions and/or adjustments be made to the VCO to make its gain less variable. In this way, the variations in the PLL dynamics and noise behaviour can be minimized.

5.2 Loop-Filter Simulation Using A Verilog-A Hardware Description

Now that a loop filter transfer function $F(s)$ has been derived, a Verilog-A model of the loop-filter is required. This is necessary to combine with the rest of the PLL components. Assuming the input to the loop filter is a current signal biased at a voltage of 0.6 V, the Verilog-A model of a fifth-order loop-filter with transfer function

$$F(s) = \frac{a_1s + a_0}{b_5s^5 + b_4s^4 + b_3s^3 + b_2s^2 + b_1s} \quad (5.7)$$

is shown below:

```
'include "constants.vams"
'include "disciplines.vams"

module Filter_TF2(in,out);

  inout in;
  output out;

  electrical in,out;

  real num_hs[0:5];
  real den_hs[0:5];
  real min_cur, max_cur;

  analog begin
    @(initial_step)begin
      num_hs[0] = 3.429457336617084e+39;
      num_hs[1] = 1.364537718113437e+33;
```

```

    num_hs[2] = 0;
    num_hs[3] = 0;
    num_hs[4] = 0;
    num_hs[5] = 0;

    den_hs[0] = 0;
    den_hs[1] = 1.337482343260487e+30;
    den_hs[2] = 7.975752464916746e+22;
    den_hs[3] = 3.017325229022312e+15;
    den_hs[4] = 7.109238931851029e+07;
    den_hs[5] = 0.835919240523593;

    min_cur = 0;
    max_cur = 0;

    //V(out) <+ 0.5;
    end
    //fixed current
    V(in) <+ 0.6;
    V(out) <+ laplace_nd(I(in), num_hs, den_hs);
    //bound vout to zero min
    if (V(out) < 0) V(out) <+ 0;
    //bound vout to 1 max
    if (V(out) > 1) V(out) <+ 1;

    if(I(in) > max_cur) max_cur = I(in);
    if(I(in) < min_cur) min_cur = I(in);

    @(final_step) begin
        $strobe("Min_current_is_%e_and_max_is_%e", min_cur,
            max_cur);
    end
end

endmodule

```

5.3 Predicting The Output PLL Phase Noise Performance

The VCO jitter was also set to be 10 dB more than the extracted layout results, with these values a simulation was run in Verilog-A. The simulation results in Verilog-A and the theoretical linear model for a frequency of 34 GHz match as illustrates Figure 5.6.

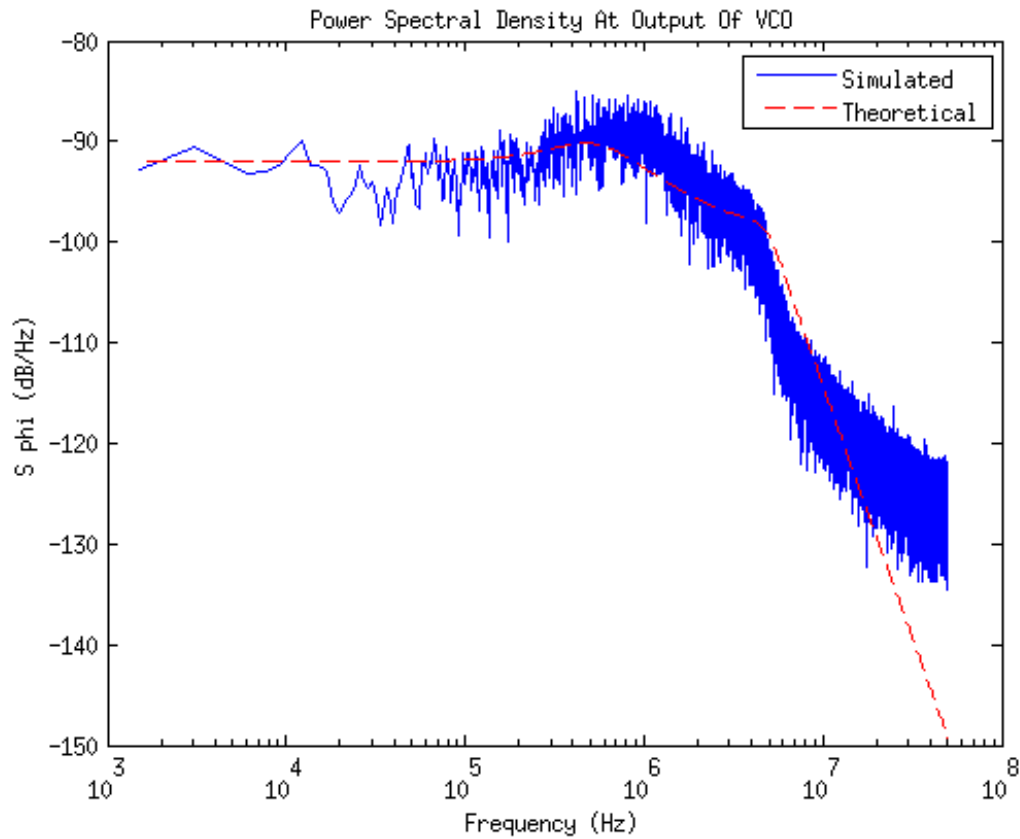


Fig. 5.6 Comparing The PLL Output Phase Noise Verilog-A Simulation Results With Those Predicted By Theory

The control voltage, before being capped at ground and 1 V supply, versus the time is shown in Figure 5.7. The behavior of the control voltage seem to indicate a cycle slip has occurred. This is well know to be caused by a high frequency step at the input of the PFD. In this simulation, the control voltage initial condition was set to zero. In the same Figure, in the steady state region, it is observed the control voltage noise varies as much as 10mV peak to peak.

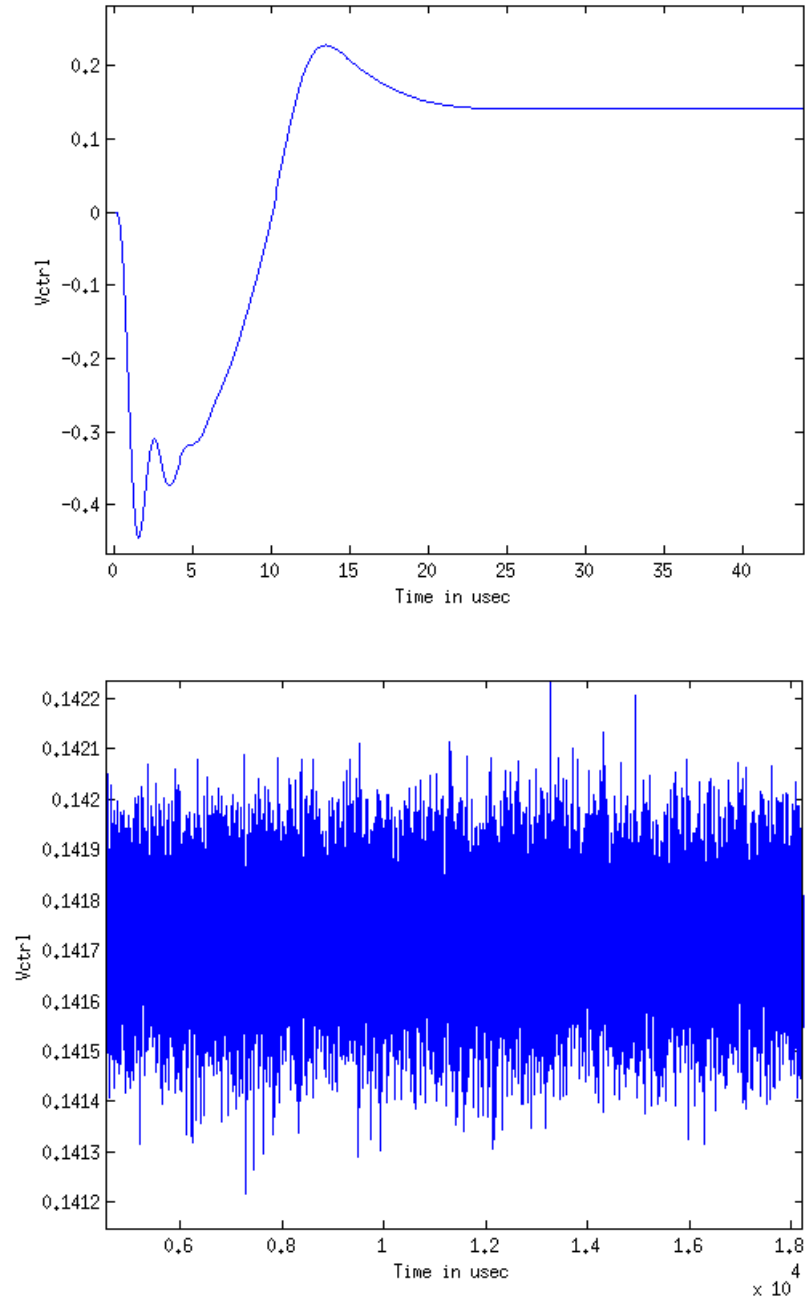


Fig. 5.7 Verilog-A Simulation Results For The PLL Control Voltage vs Time
 (a) Transient Result from 0 to 50 μs (b) Variation In Steady-State Voltage.

Finally, the PLL was tested at a frequency of 35 GHz. In this region, the VCO gain is

near its maximum slope. Figure 5.8 reports the phase noise at the output of the VCO. As seen, the behavior of this phase noise does not match the linear model derived earlier. In this case, the phase noise does even meet the requirements anymore.

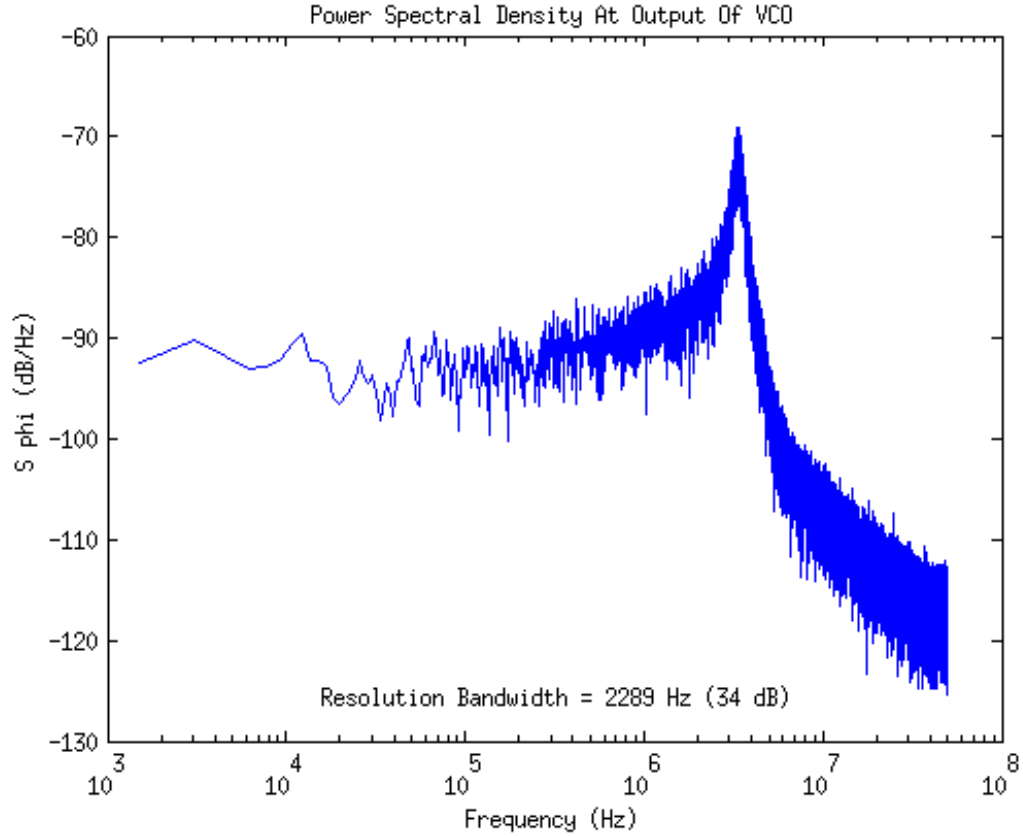


Fig. 5.8 PLL Output Phase Noise With Nonlinear VCO Operating At 35 GHz

In order to verify that this behavior is not caused by simulation artifacts, K_{VCO} was fixed to its corresponding value at 35 GHz and the simulation redone. From Figure 5.9 the phase noise behaves exactly like the linear model predicted, supporting the hypotheses the non-ideal effects of the VCO gain influence the ideal behavior.

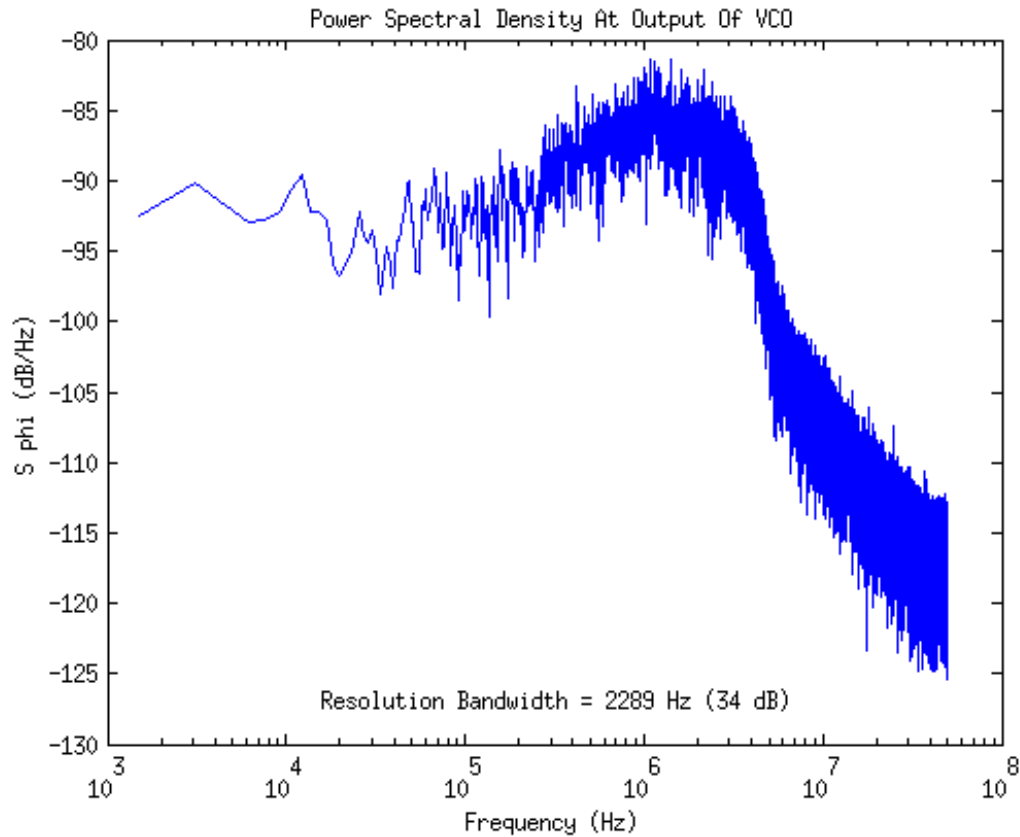


Fig. 5.9 PLL Output Phase Noise With Fixed-Gain VCO Operating At 35 GHz

From these results, it is imperative to outline the usefulness of the Verilog-A simulation in determining how non-ideal behaviors affect the phase noise. In this particular case, it is believed, the perturbation on the control voltage induce much more noise into system as the frequency steps is higher. This explains why at lower frequencies, for which the VOC gain was lowered, no such behavior non-ideal behavior was observed.

5.4 Iterative Procedure For Loop Filter Selection

When faced with these non-linear VCO behaviour, there is no easy way to derive the loop filter for all possible output frequencies. A lot of time was spend trying to figure out an optimal solution with no concrete results. It is therefore believed that for the present situation, a feedback system that compensates for the VCO nonlinear gain needs to be

found. One could always construct a PLL with multiple VCOs all tuned to a specific frequency, however, this would require a lot of silicon area and consume large amounts of power. Another option is to increase the reference frequency in order to lower the divider ratio. This increases the circuit complexity of the PFD and CP. The former will require more silicon area and more power as well. Moreover, there is no guaranteed that the noise behaviour of the PFD or CP will continue to be insignificant.

5.5 Passive/Active Loop Filter Realization

While a passive filter realization would be best from a power, area and noise perspective, the reality is that not all loop filter transfer functions are realizable. Instead, one opts for an active realization, e.g., one involving operational amplifiers, and one that generally can include a zero in the complex plane.

For this particular implementation of the filter given in 5.6, the partial fraction expansion yields zeros with imaginary parts. Because of this, the filter cannot be easily implemented with passive components. One possible filter realization using active components can be derived from the modal form of the state space equivalent model (transformed to yield a diagonal matrix with the eigenvalues in the entry as well as only real values). The matrices below map the derivatives of internal states and the input denoted by U .

$$\begin{bmatrix} \dot{X}_1 \\ \dot{X}_2 \\ \dot{X}_3 \\ \dot{X}_4 \\ \dot{X}_5 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_1 & \omega_1 & 0 & 0 \\ 0 & -\omega_1 & \sigma_1 & 0 & 0 \\ 0 & 0 & \sigma_2 & \omega_2 & 0 \\ 0 & 0 & -\omega_2 & \sigma_2 & 0 \end{bmatrix} \cdot \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ B_4 \\ B_5 \end{bmatrix} U$$

Here, the middle matrix represents the poles of the system, hence the zero row corresponds to extra integrator present in the loop filter. The table below gives the value for theses coefficients for the considered filter.

Table 5.2 State Space Matrix Coefficients

σ_1	ω_1	σ_2	ω_2	
-7.295e+06	3.131e+07	-3.523e+07	1.752e+07	
B ₁	B ₂	B ₃	B ₄	B ₅
202.8	-7.673e+06	-2.157e+07	-1.018e+07	5.67e+07

Finally, the output Y is related to the internal states X, as given by:

$$Y = \begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \end{bmatrix} \cdot \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \end{bmatrix}$$

The values for the C matrix coefficients are:

Table 5.3 State Space Matrix Coefficients for C Matrix

C ₁	C ₂	C ₃	C ₄	C ₅
1.264e+07	448.8	1199	642.9	587

Using these matrices, the general form of the filter can be created. Figure 5.10 shows the filter realization using ideal integrators from a high level perspective. The modal implementation in this case yields a parallel like structure for this filter, this has an advantage as errors in the signal path are not propagated from one stage to another as would be the case for other serialized implementations. For the current filter, it is not possible to separately implement the two major paired branches using only passive elements. Thus, this filter restricts its implementation using active elements, this adds a bit more noise into the system. In Figure 5.11 a possible implementation for the integrator (along with front adders) along with the adder is illustrated. Using these two active implementations, the total number of opamps needed for the overall filter structure would be about 10, including the inverting ones needed for the internal states etc...

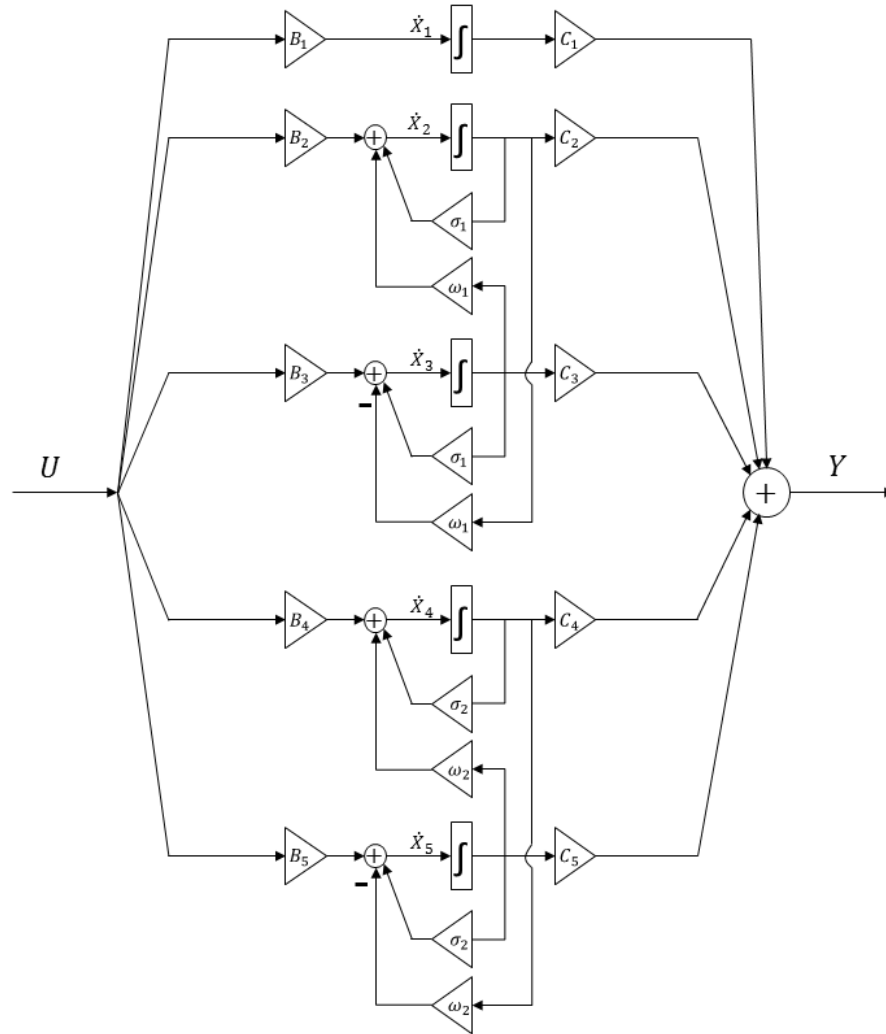


Fig. 5.10 Loop Filter High Level Realization Based on Modal Form

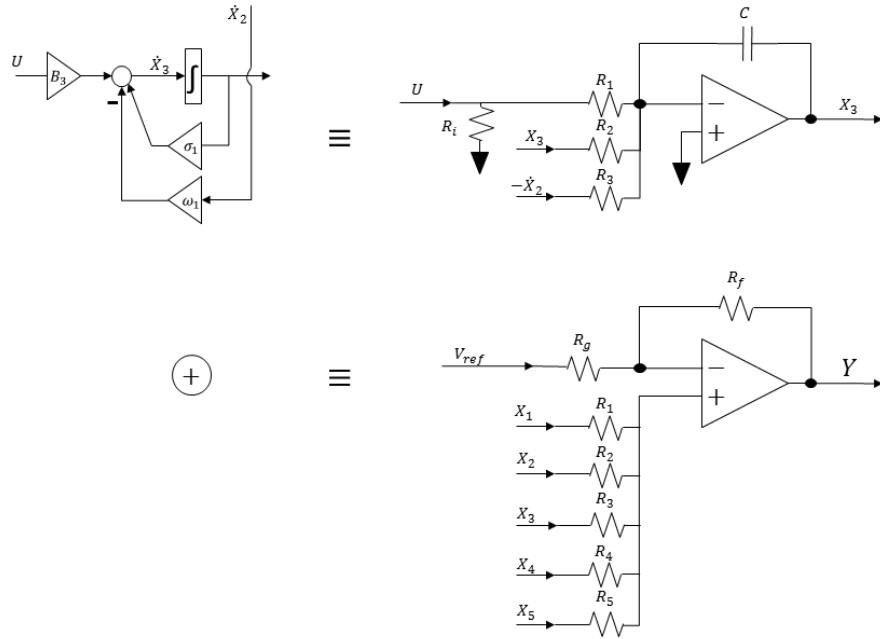


Fig. 5.11 Example of Active Filter Implementation for Integrator and Summer

Due to lack of time, the filter was not implemented fully in its active components. Different topologies for the integrators, etc, would need to be analyzed for their direct contribution to noise, immunity to offsets, power consumption and area. All these demand a lot of time in order to come up with a robust implementation.

5.6 Summary

In this chapter, the means to select the loop filter was described, together with a Verilog-A description model. Owing to the nonlinear variation in the gain of the VCO, the selection of the loop filter is not an easy task.

Chapter 6

Programmable Divider For Fractional-N Frequency Synthesis

Most often, a phase locked loop is used as a frequency synthesizer. This means that a single frequency is generated from another fixed reference frequency. In Chapter 2 the reader was introduced to a PLL that created an output frequency that is an integer multiple of the reference frequency. There they saw how the divide-by-N divider in the feedback path of the PLL set this multiplication factor. However, it is desired at times, to modulate the divider ratio between two or more values to produce an effective non-integer divider ratio. This, in turn, requires a fractional-N divider in the feedback loop of the PLL. Such an architecture is referred to as a fractional-N modulus divider as is shown in Figure 6.1.

In this chapter, the design of a fractional-N PLL together with its application for frequency synthesis will be described. The main component of the fractional-N modulus divider is the $\Delta\Sigma$ modulator. A brief introduction to this topic will be provided in this chapter; a more detailed discussion is provided in the appendix. The chapter will then continue with a discussion on the various topologies used for a fractional-N frequency synthesis together with a discussion of the tradeoff between phase noise injection and power consumption. The quantization-noise-to-phase-noise conversion in a fractional-N PLL is also derived. A Verilog-A model of a fractional-N frequency synthesizer will be used to simulate several implementations of the fractional-N PLL. This chapter will then conclude with a lengthy discussion on a 65 nm CMOS implementation of a multiple-modulus high frequency divider. Schematic level and extracted layout simulation results will be provided.

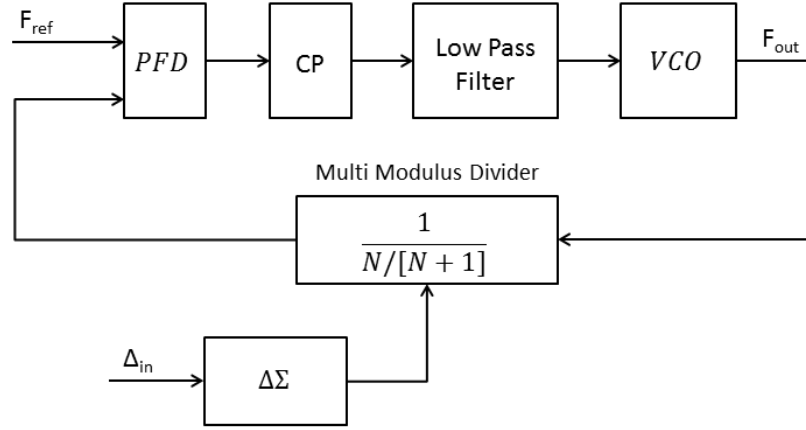


Fig. 6.1 Fractional-N Phase Locked Loop

6.1 $\Delta\Sigma$ Modulators

A delta sigma modulator encodes an input into a bit stream that can take different fixed values. There exists analog or digital $\Delta\Sigma$ modulators. An analog modulator encodes analog input signals into digital outputs, usually 1-bit, while a digital modulator encodes a digital input with fractional bits, into smaller integer bits [11]. Figure 6.2 shows the operation of a $\Delta\Sigma$ modulator. The input is converted into a bit stream that takes only two possible values. In this case, a low-pass filter $\Delta\Sigma$ was used. In order to achieve the former, such a modulator uses an integrator and a quantizer. The input is oversampled, meaning, it is sampled at a rate many times higher than its Nyquist frequency. Such a rate is usually defined as the oversampling ratio. The main input is encoded as a DC value or within a pass-band for a low pass $\Delta\Sigma$. Since the output only takes fixed integer values, quantization noise power is introduced in the system. As illustrated in Figure 6.2, in the frequency domain, the quantization noise is pushed into the high frequency band for a low-pass delta sigma modulator. The solid line is an overly simplified representation of the noise shaping of delta sigma modulators having arbitrary poles and zeros, while the dashed line represents the output spectrum for a special class of $\Delta\Sigma$ modulators involving a cascade of first order integrators.

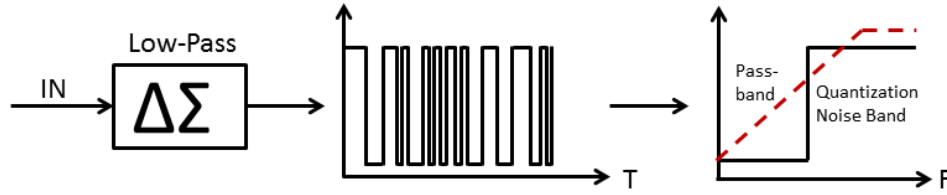


Fig. 6.2 A Simplified Representation of The Operation Of A Delta-Sigma Modulator

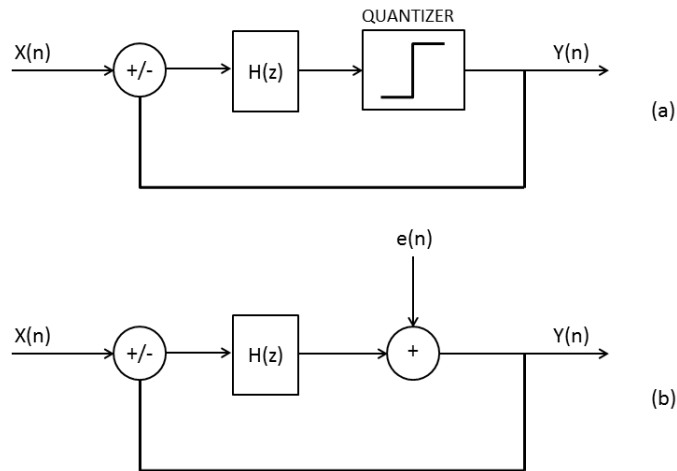


Fig. 6.3 Single-Loop Feedback Representation Of A $\Delta\Sigma$ Modulator: (a) general block diagram (b) linear equivalent representation.

A delta sigma modulator is composed of a discrete-time transfer function filter $H(s)$ and a single-bit quantizer. These two fundamental components are interconnected in a single feedback loop. One popular topology is shown in Figure 6.3. In Figure 6.3(a), the block containing $H(z)$ performs a filtering function. The filter can be low-pass, bandpass or high-pass. The quantizer block simply decides which region the input signal lies in. For instance, for a 1-bit quantizer the quantizer function is to decide whether its input is in the

lower or upper half of its input range. A linear representation of the modulator is provided in Figure 6.3(b). Here the quantizer is modeled as summation element with two inputs. One input is the input to the quantizer and the other is an error signal $e(n)$ that represents the difference between quantizer output and input. Since the instantaneous error can not be known a priori, it is generally assumed to have a uniform probability distribution function with the following standard deviation,

$$e_{rms} = \frac{\Delta^2}{12} \quad (6.1)$$

where Δ is

$$\Delta = \frac{2 \cdot FS}{(2^{bits} - 1)} \quad (6.2)$$

Here, FS represents the full range of the quantizer's input and the term $bits$ represents the number of bits used to model the quantizer operation. From a linear analysis of the block diagram of Figure 6.3(b) in the z -domain, the output $Y(z)$ can be written in terms of the input $X(z)$ and the quantization error $E(z)$ according to

$$Y(z) = X(z) \frac{H(z)}{1 + H(z)} + \frac{E(z)}{1 + H(z)} \quad (6.3)$$

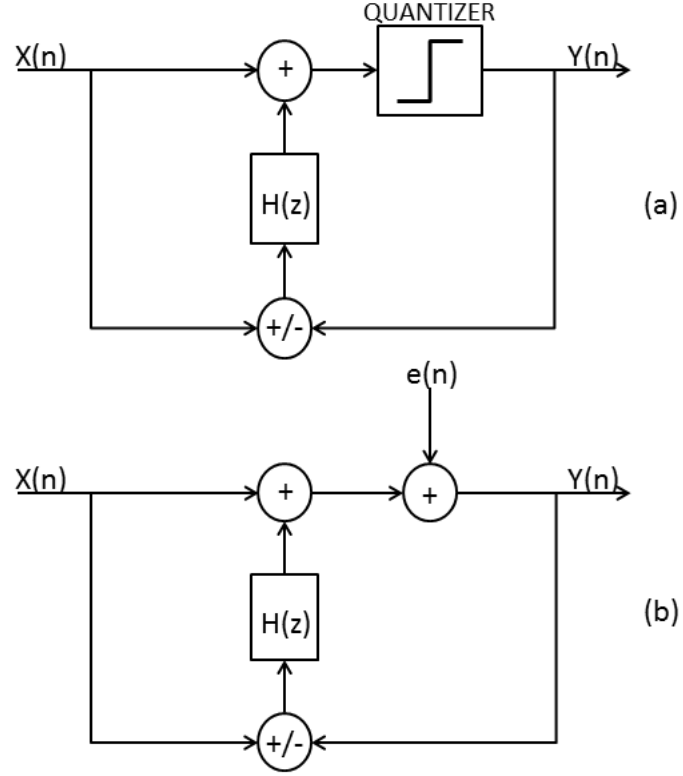
With the input $X(z)$ set to zero, the transfer function from the quantization error to the output is called the noise transfer function or NTF and is expressed as

$$NTF(z) = \frac{Y(z)}{E(z)} = \frac{1}{1 + H(z)} \quad (6.4)$$

Here the NTF is dependent on the filter function $H(z)$ and is used to shape the spectrum of the quantization noise. Conversely, with the quantization error $E(z)$ set to zero, the transfer function from the input to the output is called the signal transfer function or STF and is expressed as

$$STF(z) = \frac{Y(z)}{X(z)} = \frac{H(z)}{1 + H(z)} \quad (6.5)$$

A drawback to this topology lies in the fact that the STF varies with frequency. In contrast, the next topology provides the same NTF function but with a unity-gain STF [30]. This topology is shown in Figure 6.4.

**Fig. 6.4** Unity-Gain SFT $\Delta\Sigma$ Topology

(a): Non-Linear System, (b): Linear System For Noise Analysis

The discrete-time output for this topology is written as

$$y(z) = X(z) + \frac{e(z)}{1 + H(z)} \quad (6.6)$$

Clearly, the STF is unity and the NTF is given by Eqn. 6.4. The only drawback with this topology is the increase delay in the feedback loop. This acts to decrease its operational speed compared with the other topology.

At this juncture, we'll leave the discussion of $\Delta\Sigma$ modulation and return the central focus of this dissertation, that being, the fractional-N PLL. However, for those readers

interested in further details on $\Delta\Sigma$ modulation, they can refer to the appendix for a discussion of the various topologies used in a fractional-N PLL. Some of the topologies discussed are the MASH and SSMF realizations.

6.2 A Fractional-N Modulus Divider

The main architecture for the fractional-N PLL was previously shown in Figure 6.1. The general concept of the fraction-N PLL is about multiplexing the divider ratio, for instance between $N+1$ and N , in feedback of the PLL over a period of time, such as the average division ratio corresponds to an integer plus a fractional part.

In order to demonstrate the operation of the Fractional-N modulus divider concept, Figure 6.5 illustrates how the quantization noise from a single bit $\Delta\Sigma$ modulator is mapped to the output of the frequency divider. First, in (a) and (b) the output spectrum of the divider consists of the reference frequency divided by the division factor, N or M when the select bit is high or active. In (c), the delta output spectrum contains tones at the same frequency than the two previous cases, however, because of the modulation operation, their magnitude differs. As such, one tone will have a higher magnitude than the other depending on the dc encoded delta sigma value. In addition, the spectrum will contain the quantization noise near the encoded tone, here shown as dashed line. This tone will be not present in the output of the divider spectrum, instead, the closed loop PLL will lock onto it and filter all the noise around it.

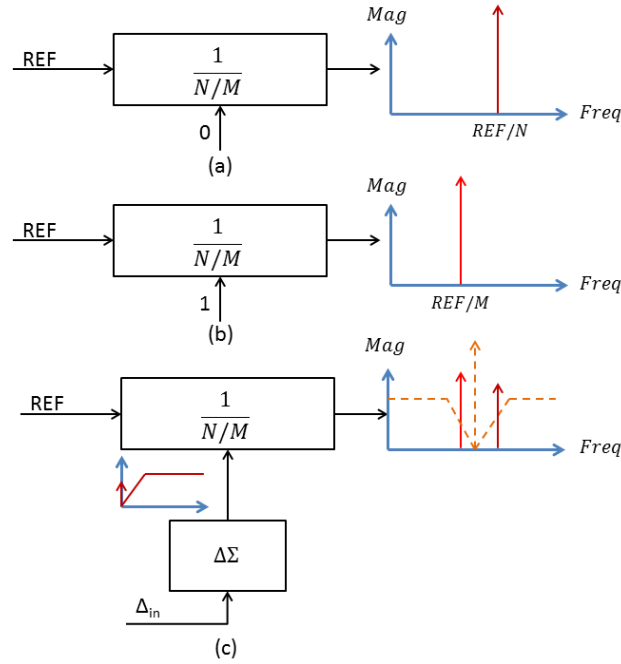


Fig. 6.5 Noise Mapping $\Delta\Sigma$ Modulated Divider

(a): Output Tone With N divisor, (b): Output Tone With M Divisor, (c): Output Tone With Delta Sigma Modulator Active

A test was performed in which an SSMF-II $\Delta\Sigma$ modulator (see appendix A) with two output bits, was used in the configuration of Figure 6.1. The four output frequencies were selected as 111.111 MHz, 100 MHz, 90.9092 MHz and 83.33335 MHz. The output power spectrum is shown in Figure 6.6 for a sampling frequency of 299.718 MHz and Figure 6.7 for a sampling frequency of 444.444 MHz. In the former figure, the sampling frequency yields an integer bin for the main encoded frequency, this was chosen in order to show the noise shaping, however due to leakage from other bins, the resolution is not very good, even for different windowing functions (in this case a Nutall fourth order window was used). In the former figure, the sampling frequency was chosen as an integer value of the greatest frequency at the output (111.111 MHz).

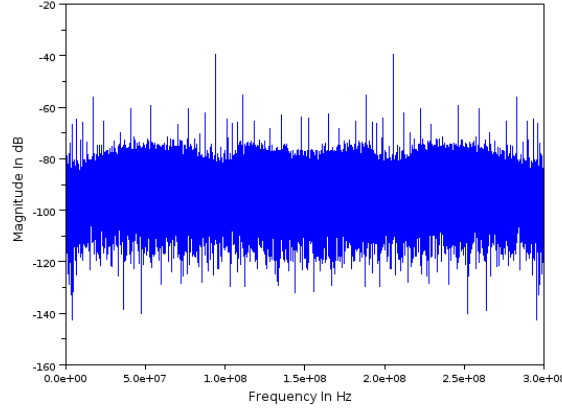


Fig. 6.6 Output Spectrum At Divider Output For SSMF-II $\Delta\Sigma$ Modulated Divider (Fs of 299.7186 MHz)

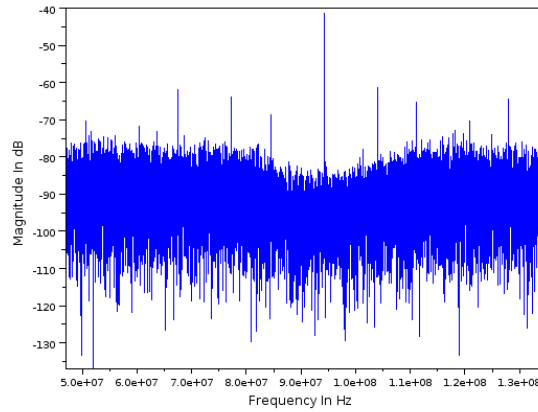


Fig. 6.7 Output Spectrum At Divider Output For SSMF-II $\Delta\Sigma$ Modulated Divider (Fs of 444.444MHz)

From Figure 6.6 it is evident that the sampling frequency yields an integer bin for the main encoded frequency, this was chosen in order to show the noise shaping, however due to leakage from other bins, the resolution is not quite good, even for different windowing functions (in this case a Nuttall fourth order window was used). From Figure 6.7 it is evident that the sampling frequency for the output sampling was chosen as an integer value of the greatest frequency at the output (111.111 MHz). As both figures illustrate, the quantization

noise mapping previously discussed, agrees with the power spectrum results obtain from a Verilog-A simulation.

6.2.1 Verilog-A Implementation Of A Fractional-N Modulus Divider

A model of the fractional-N modulus divider described with Verilog-A is introduced in this subsection.

Contrary to the integral function previously used in the VCO chapter, function `idt` is used instead for the integration. This function is not modulus bounded, and hence continues up to infinity as shown in Figure 6.8. As shown on the graph, the function generates an integer value starting from zero and by increments of one each time the period of the generated frequency elapsed. The crossing function is used to trigger a high output on the detection of an integer value and a low value on the detection of any integer plus 0.5, 0.5, corresponding to half of a period. For modulated divider, it is required to change the period dynamically. This could be done by changing the frequency input of the integration function `idt`, however, because of difference in steps, this results in very poor resolution and hence extra injected phase noise due to resolution issues. The other method is to implement the division as in reference [9]. This method changes the crossing event value such as to increase or decrease the overall time period. Figure 6.8 illustrates the concept. By computing the next crossing event to occur a fraction longer than the scheduled one, the time period can be extended. However, while doing so, the value of the falling edge crossing needs to be updated according to this new rising edge value.

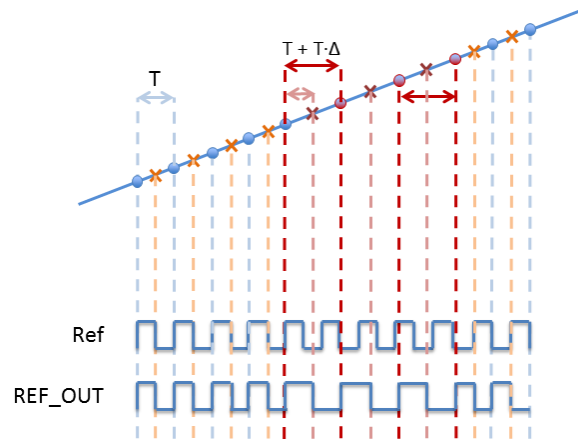


Fig. 6.8 Verilog-A Integration Operation Along With Change Of Period Crossing

This method is implemented inside the VCO with Verilog-A in order to reduce the amount of events generated and hence increase the speed of the simulation. The Verilog-A code fragment below describes a fractional-N modulus divider:

```

freq = (V(in)-Vmin)*(Fmax-Fmin)/(Vmax-Vmin) + Fmin;
if(freq > Fmax) freq = Fmax;
if(freq < Fmin) freq = Fmin;

freq = freq/N;

phase = idt(freq,0);

@(cross(V(delta_in),+1,ttol));
@(cross(V(delta_in)+0.5,-1,ttol));
@(cross(V(delta_in)-1,+1,ttol));
@(cross(V(delta_in) -2, +1, ttol));

@(cross(phase-nexth,+1,ttol)) begin
    nexth = nexth + (N+floor(V(delta_in)+0.5)*delta_f)/N;
    out_div = Vhigh;
end
@(cross(phase - nexth + (N + floor(V(delta_in)+0.5)*delta_f)/2/N, +1, ttol)) begin
    out_div = Vlow;
end
end

```

Delta_f in the code represents the step change in the fractional divider. In addition, the rounding function is implemented by adding 0.5 to the input and flooring the result. The crossing event is changed by adding an increment. This increment is nothing more than delta_f times divided by the original division factor, N. In this way, the the crossing period extends to a larger period of time. One of the problems with the unbounded integration is rounding errors. As explained in [31], as the integration continues in time, the numbers get larger and larger and the fractional quantities loose their precision. Thus, the crossing at specified point might not actually occur within the given tolerances. In order to check this effect, a test was performed in which the $\Delta\Sigma$ modulator was removed from the PLL and the PLL simulated with ideal components. The output phase noise generated by the PLL is shown in Figure 6.9.

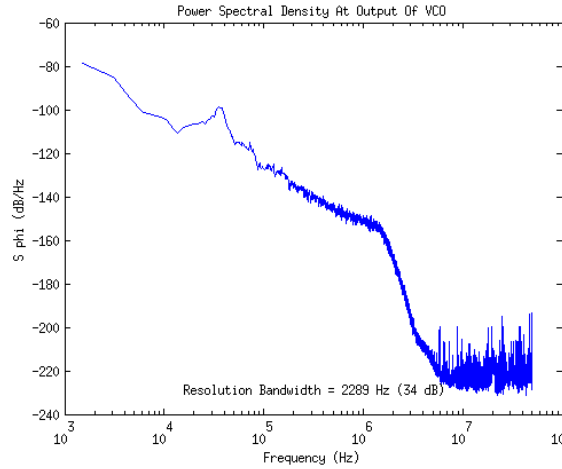


Fig. 6.9 Output Phase Noise Standalone VCO With Embedded Fractional Divider

In order to deal with this extra phase noise, an attempt was made to use a circular integrator in place of the unbounded linear integrator function. Figure 6.10 illustrates the mapping between the linear and circular integrators. The operation of the linear integrator as well as the process of extending the period was explained previously, now the same can be achieved using the circular integrator. As shown in Figure 6.10, the crossing event can be mapped to the circular integrator but special attention has to be made when one cycle has to be skipped.

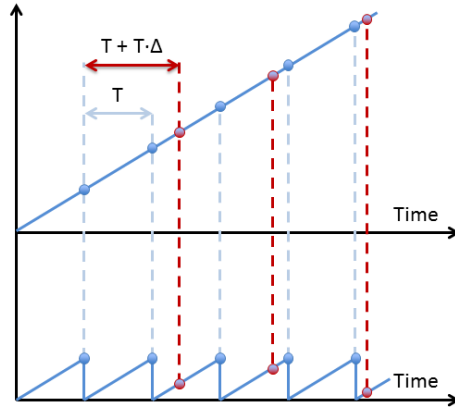


Fig. 6.10 Linear and Circular Integrator Functions Captured By A Verilog-A Model

The Verilog-A code shown below shows the implementation of the VCO embedded with a fractional-N modulus divider:

```
@(cross(phase - next, +1, ttol, 1e-15))begin
    if(skip ==0) begin
        delta_next = (floor(V(in_delta)+0.5)*delta_f)/N;
        next = (delta_next + next_prev/10)%1;
        next = 10*next;
        if(next_prev > next) skip = !skip;
        next_prev = next;
        out_div = vhi;
        nextl = (next/10 - (1 + delta_next)/2)%1;
        if (nextl <0) nextl = 1 + nextl;
        nextl = 10*nextl;
    end
    else begin
        skip = !skip;
    end
    if(next ==0) next = 0.000001;

    @(cross(phase - nextl, +1, ttol))begin
        out_div = vlow;
    end
end
```

This code unfortunately did not work when implemented in Verilog-A. The reason was found to be the crossing detection. When the signal is very small, near one of the edges (as the crossing events can take any possible values on the integration line), the crossing function has issues detecting a transition. Furthermore, extra noise would be introduced when both extremities would be crossed. One solution would be to detect these edges and re-map them to a lower or higher value instead. This, in turn, produces more phase noise. Reference [9] suggest a way of using the circular modular but it seems that the behaviour near the edge crossing are not taken care of. Furthermore, the output signal does not produce a 50% duty cycle. This, in turn, is not critical as most of the blocks will get updated on the rising edge. However, if the simulation includes more realistic models of the various components then unexpected time violations could occur.

6.3 $\Delta\Sigma$ Modulator Contribution To PLL Output Phase Noise

The noise contribution from the $\Sigma\Delta$ modulator to the PLL output will be described in this section.

In reference [12], a linear noise model of a fractional-N PLL is provided. In this subsection, the main idea behind this analysis along with some basic derivations are reported. An integer-N VCO like that seen previously in Chapter 2 has a nominal output frequency $f_{o,nom}$. The PLL will start at this nominal frequency, and the absolute phase at the output of the VCO will be give by:

$$\Phi_{vco}(t) = 2\pi f_{o,nom}t + \int 2\pi K_v V_{in}(t)dt \quad (6.7)$$

The integral part of Eqn. 6.7 will be referred as $\Phi_{out}(t)$ as first used in ref. [12]. The total phase at given time t , can also be conceived as a point on the linear integrator line previously described in the last section (see Figure C.3). The difference between these two phase points will be linearly proportional to the division sequence, expressed as

$$\Phi_{vco}(t_k + \Delta t_k) - \Phi_{vco}(t_{k-1} + \Delta t_{k-1}) = 2\pi N[k - 1] \quad (6.8)$$

After substituting Eqn. 6.7 into Eqn. 6.11, and after some algebraic manipulation, following

equation is obtained

$$2\pi f_{o,nom}(\Delta t_k - \Delta t_{k-1}) = 2\pi(N[k-1] - N_{nom}) - (\Phi_{out}(t_k + \Delta t_k) - \Phi_{out}(t_{k-1} + \Delta t_{k-1})) \quad (6.9)$$

This equations states that the absolute change in phase at the output of the VCO equals the change in the divider ratio from the nominal or encoded $\Delta\Sigma$ modulator dividing ratio. Carrying out the summation and assuming the difference between $\Phi_{out}(t_k - \Delta t_k)$ is small, the time difference in the rising edge between the main reference and the feedback output from the PLL is given as

$$\Delta t_k = \left(\frac{T}{2\pi}\right) \left(\frac{1}{N_{nom}}\right) \left(2\pi \sum_{m=1}^k n[m-1] - \Phi_{out}[k]\right) \quad (6.10)$$

This time difference can be rewritten in terms of the phase difference between the reference signal and the PLL feedback signal, which corresponds to the output of the divider, as

$$\Delta t_k = \frac{T}{2\pi}(\Phi_{ref}[k] - \Phi_{div}[k]) \quad (6.11)$$

Substituting this equation into Eqn. 6.10, the phase at the output of the divider can be isolated and the following derived

$$\Phi_{div}[k] = \frac{1}{N_{nom}} \left(-2\pi \sum_{m=1}^k (N[m-1] - N_{nom}) + \Phi_{out}[k]\right) \quad (6.12)$$

Equation 6.12 can be intuitively supported as well. The output phase without the divider is equal to the $\Phi_{out}[k]$ divided by the nominal division ratio (see Figure XXX of Chapter 1). The other argument is simply stating that any variation in the divider value which does not correspond to the nominal division factor will introduce a phase noise or a shift in Δt_k as seen in Eqn. 6.10. This variation is expressed in frequency hence it has to be integrated to yield a phase difference. Figure 6.11 shows the equivalent model in the frequency domain [12]. The summation in Eqn. 6.12 was replaced with a delay integrator in the z-domain. This is possible since any aliasing caused by this integrator in the frequency domain will be suppressed by the filter, whose cutoff frequency is well below the main reference frequency.

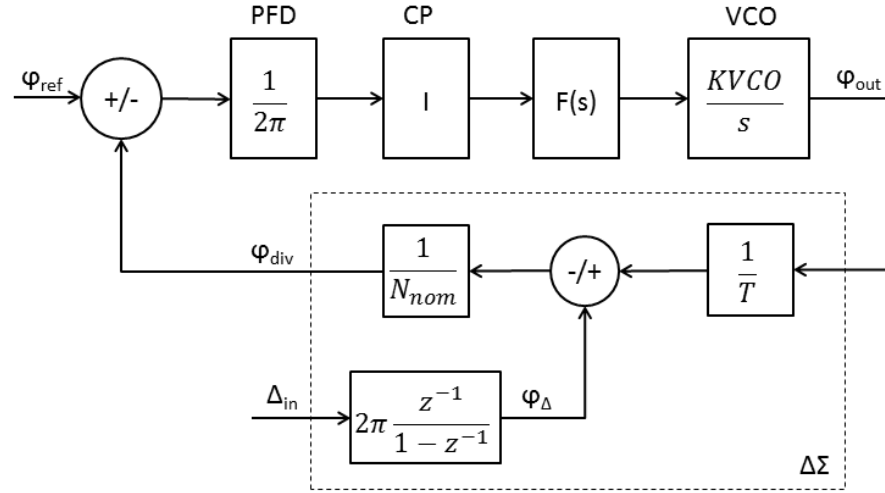


Fig. 6.11 Feedback Fractional-N $\Delta\Sigma$ Fractional Synthesizer Frequency Equivalent Model

The quantity $N[m-1] - N_{nom}$ in Eqn. 6.12 corresponds to the quantization noise in a $\Delta\Sigma$ modulator. This is so, since any value beside the DC encoded or perhaps ac signal, if modulation is desired, will produce on average a value other than the nominal division ratio.

In order to demonstrate the various phase noises injected by the different $\Sigma\Delta$ modulator topologies (such as those described in the appendix), a PLL with a closed-loop input-output transfer function initialized with a fifth order Butterworth filter function was implemented in Verilog-A and the phase noise computed. Table 6.3 lists the parameters of the various PLL components used in this example.

Table 6.1 PLL Characteristics

VCO	F_{min}	31e9	F_{max}	41e9	KVCO	10e9 Hz/V
PFD	Tristate					
CP	Current	1.3mA				
G(s)	5 th Order	Butterworth	w_O	1 Mhz		
Lead_Lag	Ratio	1/4				

Recall from Chapter 2 that the phase noise injected into node φ_{div} is altered by the

transfer function $\frac{\varphi_{div}}{\varphi_{ref}}$. Assuming a nominal division ratio of $N = 349.1036$, this transfer function would be found to be

$$\frac{\varphi_{div}}{\varphi_{ref}} = \frac{2.756e37s + 4.328e43}{551.8s^6 + 1.576e10s^5 + 2.063e17s^4 + 1.655e24s^3 + 8.677e30s^2 + 2.756e37s + 4.328e43} \quad (6.13)$$

Figure 6.12 illustrates the expected phase noise at the output of the VCO due to the quantization noise from the $\Delta\Sigma$ modulator. This plot is derived assuming a 3-bit output from the modulator (not true for the synthesized NTF though). The injected quantization noise is calculated from Eqn. A.1 as well as Eqn. A.2. As this figure suggest, a high-order NTF injects more phase noise into the system than a lower order one.

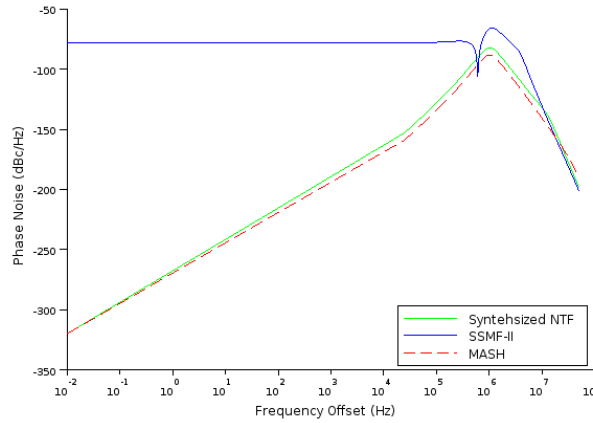


Fig. 6.12 Predicted Phase Noise At PLL Output Cause By The $\Sigma\Delta$ Modulator Quantization Noise

6.4 Implementation Details For A Programmable Frequency Divider Circuit

A fractional-N PLL requires a modulus divider in its feedback loop. A modulus divider changes its division ratio based on an input selection signal. In this section, the operation of a modulus counter is introduced and the circuit diagram for a high speed modulus 4-to-7 counter is presented. Furthermore, a modulus divider covering ranges from 0 to 10 GHz for implementation in a 65 nm CMOS process is to be described.

6.4.1 A Modulus Counter Approach

In Figure 6.13(a) and (b) show a digital circuit representation of a synchronous divide-by-2 circuit and a divide-by-3 circuit. The additional And gate in the circuit of part (b) introduces an extra period delay as compared with the divide-by-2 circuit. In doing so, the waveform at the output at Q_{out} no longer has a 50% duty cycle. Also, it should be noted that it is important to pay attention to the clock being negated in the divide-by-2 circuit.

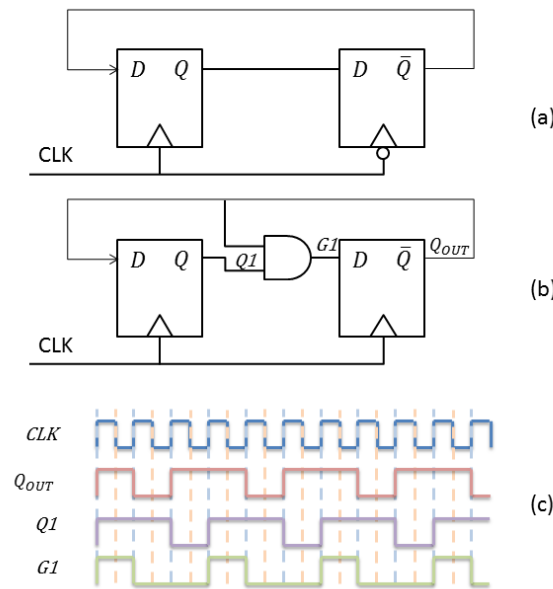


Fig. 6.13 (a) Divide-By-2 Circuit (b) Divide-by-3 Circuit (c) Divide-by-3 Waveform

A dual-modulus counter (one that has more than one divide function) could be synthesized by combining the two circuits of Figure 6.13 into one and whose function is selectable by an external select port. Figure 6.14 illustrates this concept. Assuming R is high, when MODIN is low, only the upper portion of the circuit is activated. It is equivalent to the divide-by-2 circuit introduced earlier at its output port F. When MODIN is high, the system is equivalent to the divide-by-3 circuit. If the input R is low, it disables the modulus operation, hence the circuit performs a fixed divide by two.

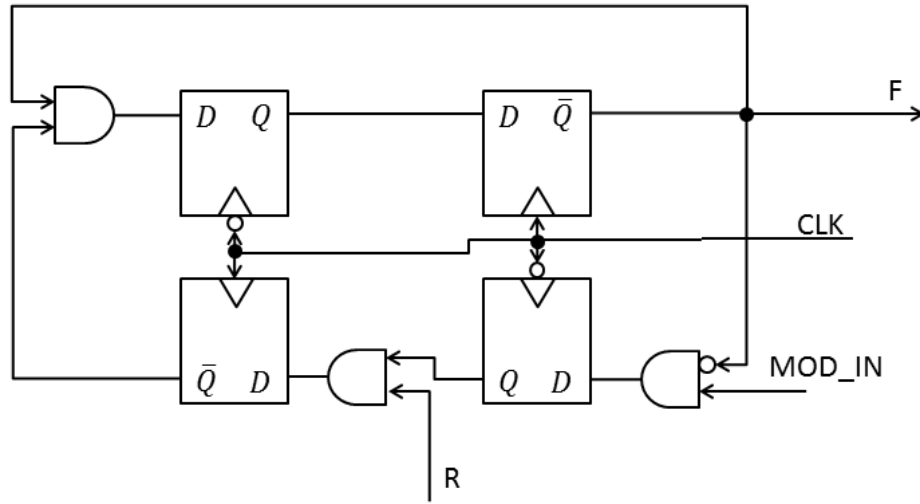


Fig. 6.14 Modulus 2/3 Counter

The divider output, F , can be further sub-divided by cascading multiple modulus blocks. The Figure 6.15) shows the implementation of a modulus divide by 4-to-7 created by cascading two dual 2/3 modulus dividers [5]. The inputs R are used to select the division factor, from 4 to 7. The operation of the circuit is as follows. For both inputs R_0, R_1 zero, the divide-by-3 operation is disable for both sub-circuits resulting in the upper circuits being activated (i.e., leaving the divide-by-2 circuit alone). When only R_0 is active, the divide-by-3 action is enable in the first block. The MODIN signal previously seen in Figure 6.14 is connected to the second block output. The first block triggers a high to the second block for two periods of the main CLK signal. The second block in turn is set to count up to two, and will generate a high output for twice the period of this latter signal. The MODIN signal will trigger a high logic. This in turn makes the first block to divide by three, hence the output of the first block will be low for 3 CLK cycles and high for two clock cycles. The same operation principles applies when R_0 low and R_1 are high but now two pulses will be added instead of one (since the input clock has already been divided by two in the first block). When both inputs are high, the whole circuit is activated. In this case, the first block adds a clock period while the second block adds two extra periods, in addition of the division by four, hence resulting in a division by seven.

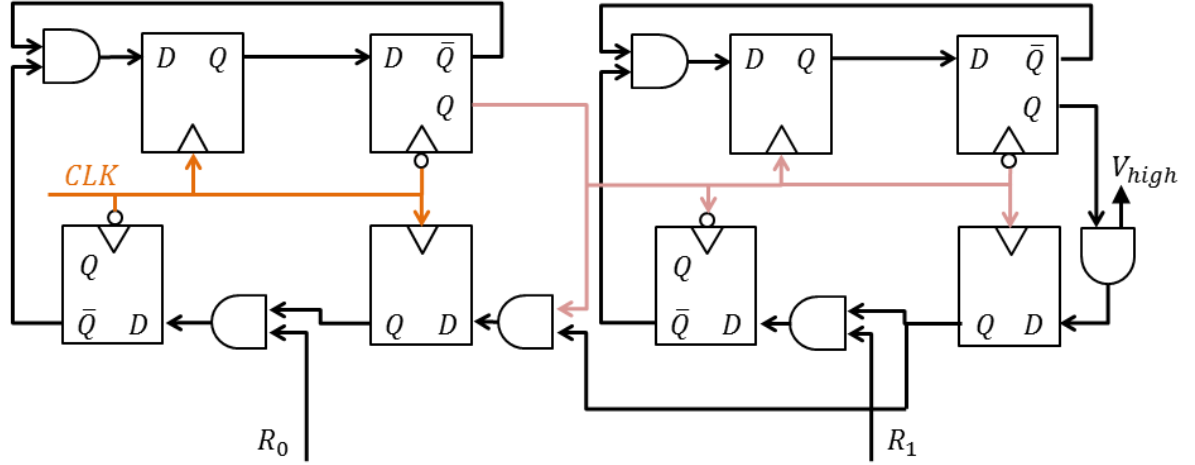
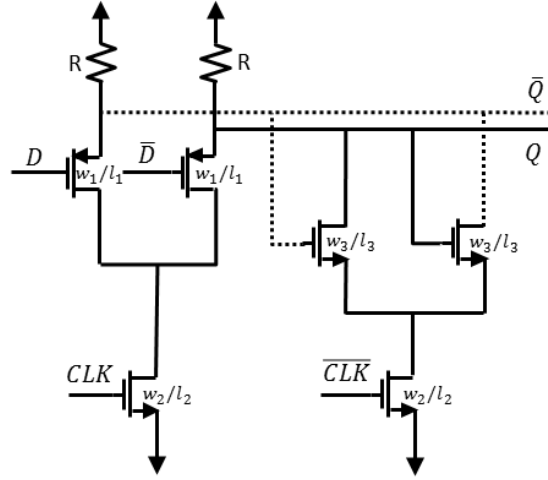


Fig. 6.15 Modulus 4-to-7 Counter

A 4-to-7 Modulus Divider Circuit Implementation

The 4-to-7 modulus divider was implemented in a 65 nm CMOS process using current-mode logic (CML). The choice of CML is due to its ability to operate over a wide range of frequencies, from zero to GHz, however, this comes at a cost of large power consumption and silicon area. Figure 6.16 shows the circuit of a D-flip-flop CML. This circuit is inspired from reference [32]. Compared with conventional CML, this structure contains no current source in its tail. This simplifies the design and increases its speed of operation. The optimization for the required values follows the strategy given for conventional CML in [5]. The main tradeoffs in CML logic as such is the current consumption, transistor sizing, resistor value required for a minimal required output voltage swing and frequency of operation.

**Fig. 6.16** A CML D-Type Flip-Flop

The flip-flop of Figure 6.16 was used to implement the dual modulus 2/3 counter previously described. Furthermore, in order to reduce power consumption, the second block was optimized by decreasing the current, since the main high frequency has been reduced by a factor of two. The optimized 65nm transistor values for the flip flop for both cases is summarized in Table 6.4.1. These values will yield an operational frequency from zero to 18 GHz with a current consumption of 2.333 mA and 1.555 mA for both flip flops.

Table 6.2 CML DFF Values

	First	Second
w_1/l_1	$6*2 \mu/80n$	$4*2\mu/80n$
w_2/l_2	$8*2 \mu/80n$	$6*2\mu/80n$
w_3/l_3	$4*2 \mu/80n$	$4*2\mu/80n$
R	382.039Ω	512.371Ω

In order to increase speed and achieve compactness, the AND gate was embedded inside the CML D-flip-flop as shown in Figure 6.17. Transistors M_4 are there to compensate for the branch of the CML by mimincking the on resistance and charge of M_1 . As before, two components were built to minimize the current consumption of the second modulus divider, as it runs at half the frequency of the primary one.



Table 6.3 CML AND-DFF Values

	First	Second
w_1/l_1	$8^*2 \mu/80n$	$6^*2\mu/80n$
w_2/l_2	$6^*2 \mu/80n$	$6^*2\mu/80n$
w_3/l_3	$3^*2 \mu/80n$	$2^*2\mu/80n$
w_4/l_4	$3^*2 \mu/80n$	$2^*2\mu/80n$
w_5/l_5	$8^*2 \mu/80n$	$6^*2\mu/80n$
w_6/l_6	$5^*2 \mu/80n$	$3^*2\mu/80n$
R	304.566Ω	445.356Ω

The layout of the circuit was very challenging as the RF transistors have more stringent layout and spacing rules than transistors operating at low frequencies. Furthermore, the

resistors also require additional spacing. The circuit was nonetheless laid out using RF layout techniques, e.g., compact and symmetric layout, wider wires for low resistance, and making the current circulate from top to bottom and not reverse. Figure 6.18 shows the final layout of the CML modulus divider.

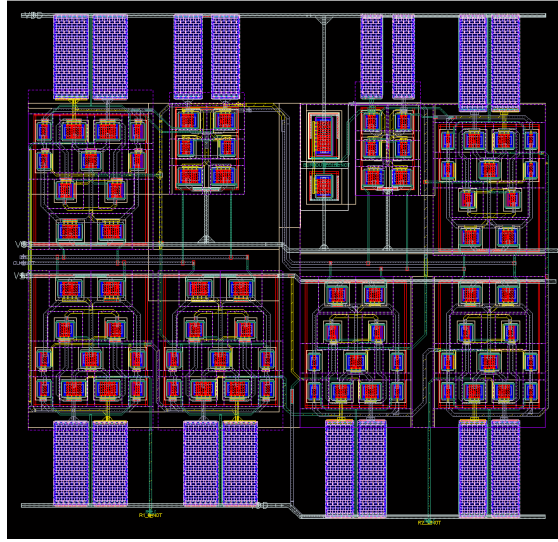


Fig. 6.18 CML 4 To 7 Modulus Divider

6.4.2 A Programmable Counter Approach

In a PLL, the division ratio depends on the reference frequency and, sometimes, huge divider factors are required. The cascading approach of dual modulus CML logic blocks might be too expensive in terms of area and current consumption. One way to increase the divider ratio is to include a programmable counter in the division block. If a programmable counter with N bits is used to precede the modulus counter, with modulation factor M and $M+1$, the resulting overall division ratio would be $(2^N-1)M$ and $(2^N-1)(M+1)$.

For high frequency operation, an asynchronous design generally work best. With such an approach in mind, a 6-bit asynchronous programmable counter can be implemented by cascading several D-type flip flops together such as that shown in Figure 6.19. This type of counter is also called ripple counter, as each preceding stage clocks the next stage. As shown in the figure, the counter is implemented with load capable flip-flops. In this case, the this type of counter is used to count down, as it is easier and faster to implement the logic when counting down. The ST1 input in Figure 6.19 is used to clear the value of the

first bit, and is specific to the way the logic is implemented (more on this next).

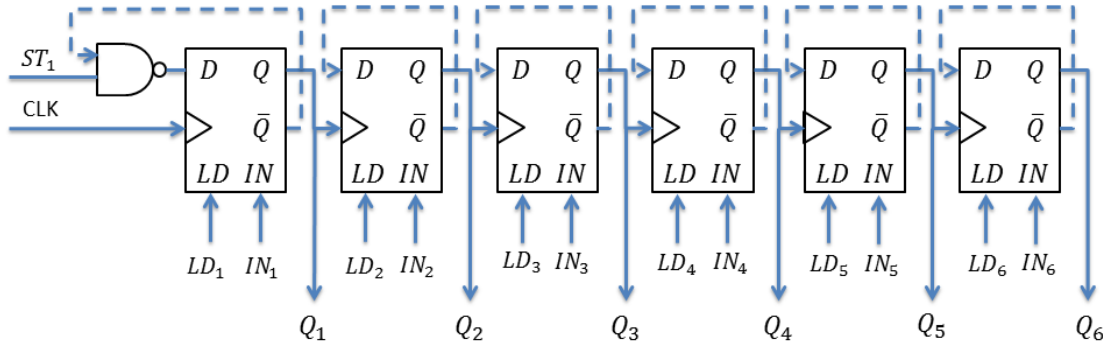


Fig. 6.19 6-Bit Asynchronous Ripple Counter

The speed of a counter is dictated by the speed capabilities of the flip-flop to load bits. In addition to the speed of the logic controller that resets the counter after counting down to zero. Instead of waiting till the counter reaches one to load the values, a clever implementation would load the values as the number of most significant bits are counted down. This is exactly what the circuit [2] does that is shown in Figure 6.20. This circuit has been slightly modified for implementation using the TSMC 65 nm digital logic components and to make use of the available flip-flops without reset capabilities.

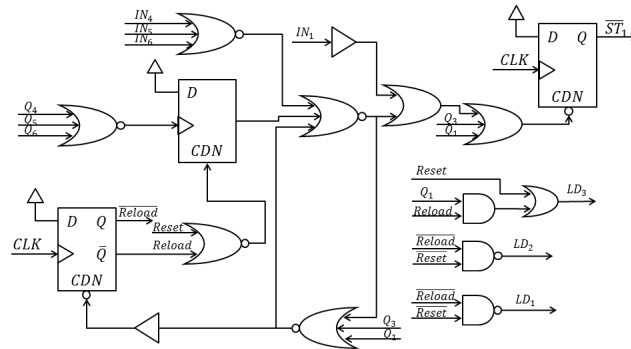


Fig. 6.20 6-Bit Asynchronous Counter Logic Controller From Reference [2] and Adapted For Realization Using the TSMC Digital Logic Library

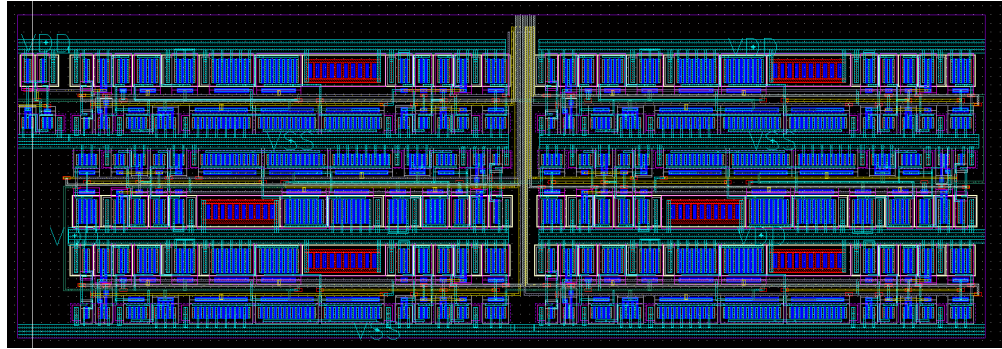


Fig. 6.21 6-Bit Ripple Counter Layout Using TSPC D-Type Flip-Flops

The operation of the controller shown in Figure 6.20 is easy to described. The reset signal is an external signal that can be used to reset the state of the counter. Signals LD1 and LD2 are the same and are just there to decrease the load on the components. These signals are routed to the load signals of the ripple counter corresponding to bits 6 to 4. These signals go high when the reload signal is high. The reload signal basically turns on when the output values Q_6 , Q_5 and Q_4 are low. The third bit is not set until the outputs Q_1 and Q_2 are high. The last bit is set by setting ST_1 low. Further details are giving in [2]. This counter topology along with the logic controlled was implemented in the 65 nm CMOS technology from TSMC. The layout of the counter is shown in Figure 6.21.

Circuit Implementation

The first component to be built for the counter was the flip-flop. In the first trial, a flip-flop with a load capability was designed. The topology was inspired from [33]. Figure 6.22 shows the TSPC-based flip-flop with load input and Figure 6.23 shows its corresponding layout. The transistors were sized according to the procedure for the TSPC logic provided in [34]. The frequency operation of this flip-flop was tested to be above 6.666 GHz in schematic and 3.333 GHz for the extracted layout. The reason for this large frequency reduction lies in the layout of this flip-flop. Owing to the bulkiness of the flip flop, it was impossible to use odd metal layers for horizontal tracing and even metal layers for vertical routing. This, in turn, introduced large delays in the circuit, hence decreasing the overall speed of operation.

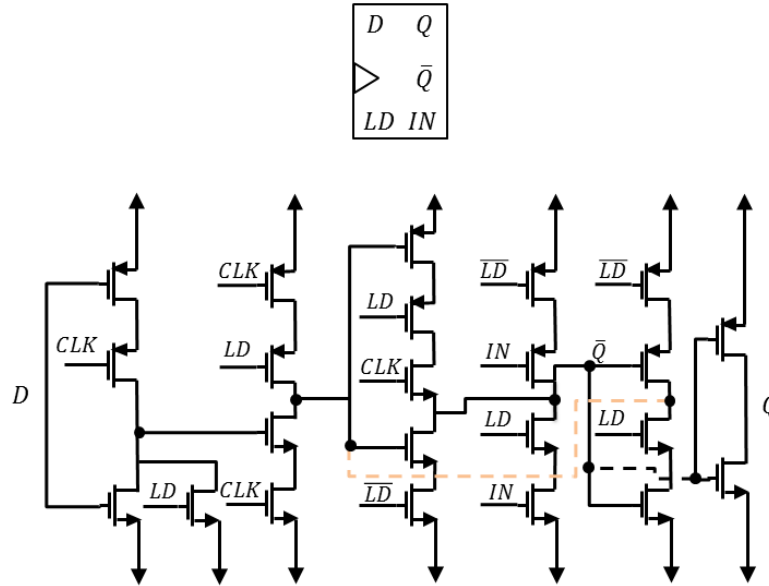


Fig. 6.22 Circuit Schematic Of A TSPC D-Type Flip-Flop

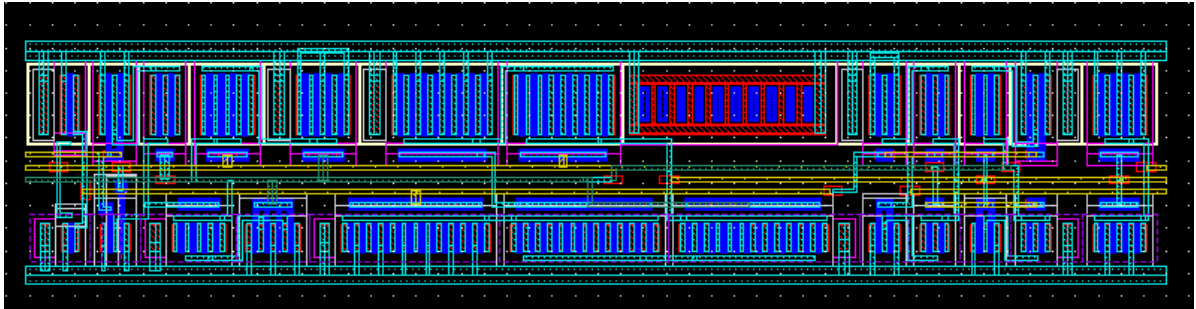


Fig. 6.23 Layout Of TSPC D-Type Flip-Flop

Owing to the poor high-frequency operation, a new flip-flop was created; one that uses components from the TSMC 65 nm digital library. The schematic of this cell is shown in Figure 6.24. The components in this library are optimized for power and area. The logic controller was also implemented using the components from the digital library.

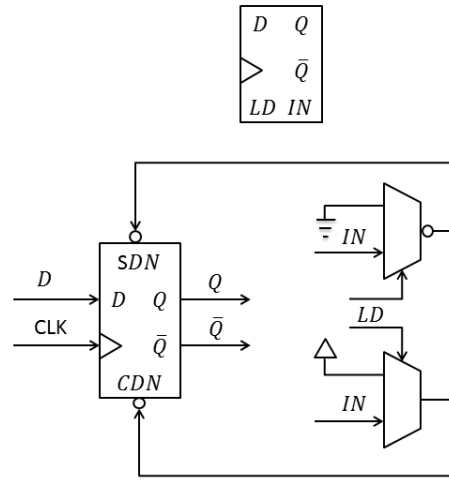


Fig. 6.24 D-Type Flip-Flop With Load Function Assembled From TSMC 65nm Components

6.4.3 Programmable Frequency Divider Circuit For IC Implementation

As the counter counts down only, a priority encoder was implemented as shown in Figure 6.25. This circuit enables the selection of the appropriate counter output. This insures the output remains high for a period of time large enough so that the precedent logic has time to react and perform the required actions. The priority encoder was designed using components from the TSMC 65 nm digital library.

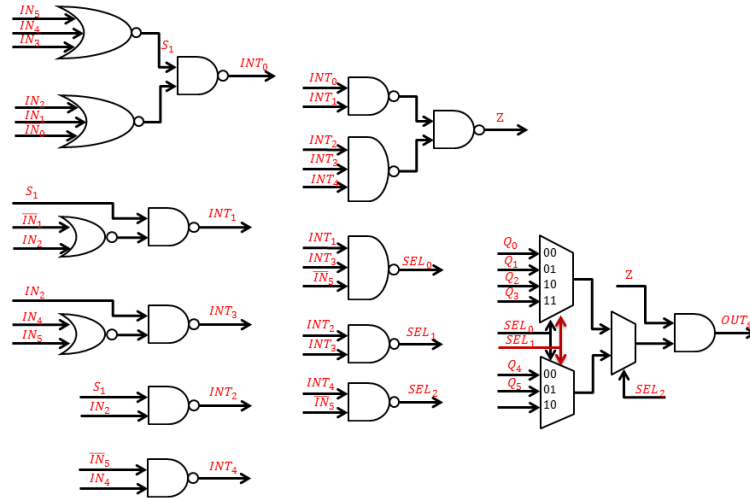


Fig. 6.25 Priority Encoder Circuit Built Using TSMC 65nm Digital Component Library

In order to test the overall circuit behavior, a state machine that compares the input value versus the number of rising edges were counted. When a mismatch occurs, it triggers a flag. As a means for validation, the circuit was modelled in Verilog and simulated. As long as the flag bit remain low, the circuit is deem to work correctly. The Verilog code for this state machine is provided below:

```
'timescale 100ps/10ps

module test_counter (not_reset,clk,in,q,ld3);

    output not_reset;
    reg not_reset;

    output [5:0] in;
    reg [5:0] in;

    input clk;
    input [5:0] q;

    reg [5:0] temp;
```

```
reg[1:0] state;

reg [5:0] count;
input ld3;
reg old_ld3;

parameter s1 = 2'b00;    parameter s2 = 2'b01;
parameter s3 = 2'b10;    parameter s4 = 2'b11;

initial begin
in <= 6'b010000;
count <= 6'b00000000;
state = 2'b00;
end

always @( posedge clk)
begin
case(state)
s1: begin
not_reset <= 1'b0;
#4
temp <= in;
state <= s2;
end
s2: begin
#0.8 not_reset <= 1'b1;
temp <= in;
$display("loaded_counter_data_%b/n", q);
state <= s3;
end
s3: begin
not_reset <= 1'b1;
if (temp == 6'b000011) begin
state <= s4; end else begin
state <= s3;
temp <= temp - 1'b1;
end
end
end
```

```

        s4: begin
            not_reset <= 1'b1;
            temp <= in;
            #0.5
            $display ("counter_output_should_be_XXXXX1_n");
            $display("counter_is_%b_and_IN_is_%b", q, in);
            state <= s2;
        end
    endcase
end

always @(posedge clk)
begin
    if(old_ld3 & ~ld3) begin
        count <= 6'b000000;
        $display("_LD3_to_LD3_Count_value_is_%b", count
            +1'b1);
    end else begin
        count <= count + 1'b1;
    end

    old_ld3 <= ld3;
end

endmodule

```

Figures 6.26 and Figure 6.27 show the simulation results at the outputs of the programmable counter as well as the output of the priority encoder when excited by a 10 GHz clock signal. The fixed division value for the programmable counter was set to 30 while the modulus divider was set to 7. The priority encoder output exhibits a period of 21 ns; exactly what was expected.

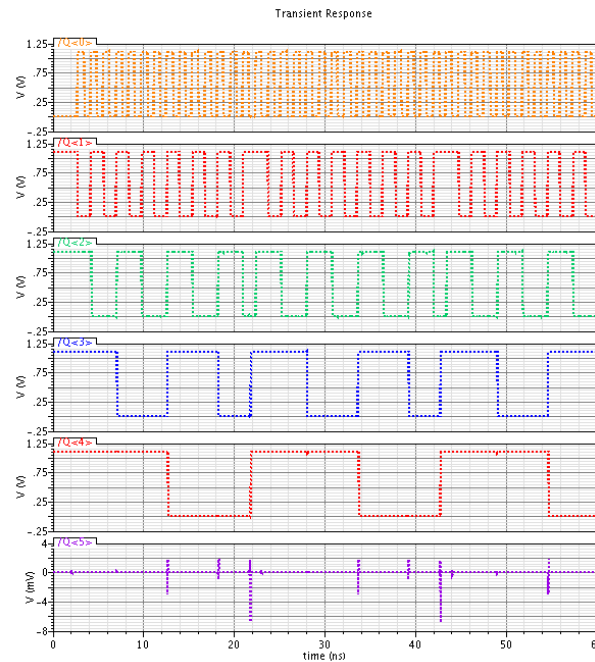


Fig. 6.26 Overall Frequency Divider Schematic Programmable Counter Output For 10 GHz Input Signal and A Frequency Division of 210

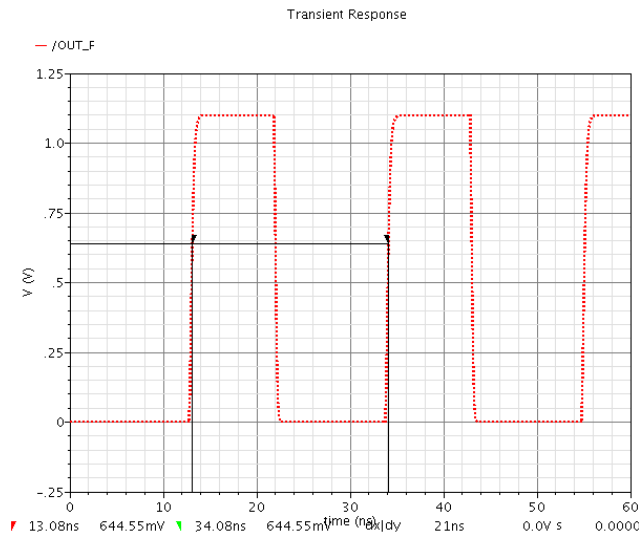


Fig. 6.27 Overall Frequency Divider Schematic Output For 10 GHz Input Signal and A Frequency Division of 210

The layout of the complete programmable frequency divider circuit is shown in Figure

6.28. The main components are: (1) a level shifter for the output of the divider (used for interaction with an FPGA board at 2.5 V); (2) additional level shifters and input buffers; (3) the programmable counter with logic counter and the priority encoder; and (4) a reference bias for the CML logic. Figure 6.29 shows the output of the frequency divider for the same setup as before, but this time including the layout parasitics. As seen both figures yield the same result at the output, suggesting the layout is not affecting the operation of the circuit.

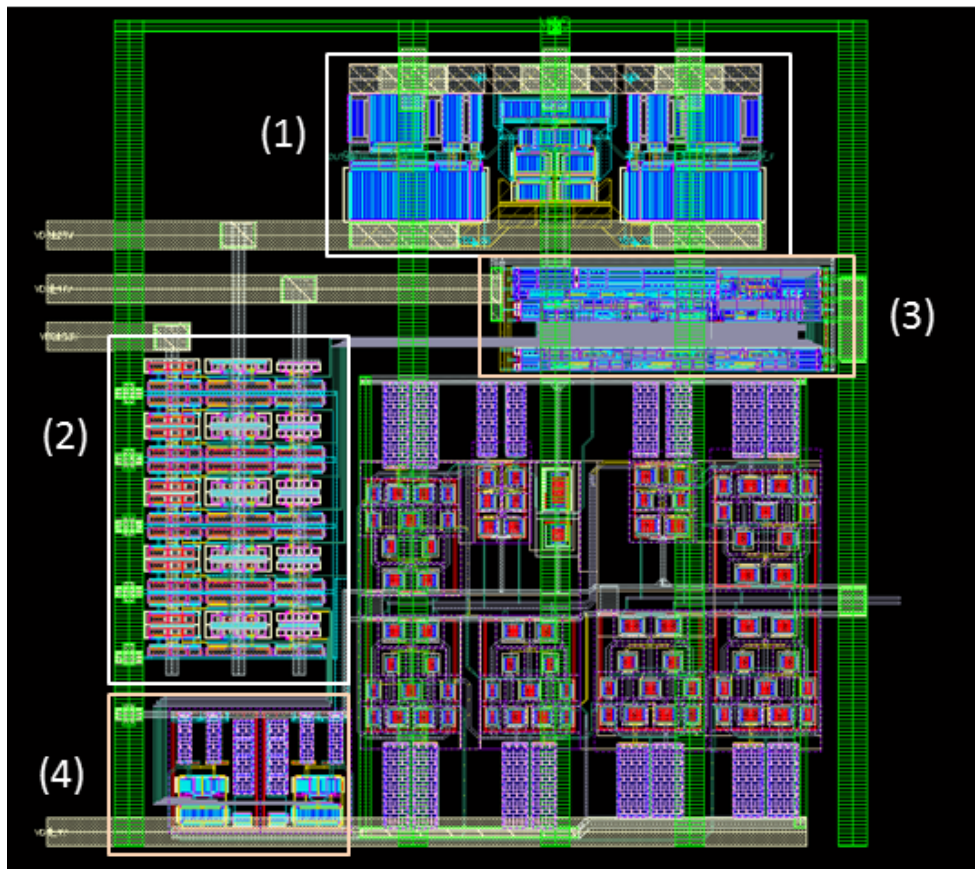


Fig. 6.28 Layout OF The Complete Programmable Frequency Divider Circuit

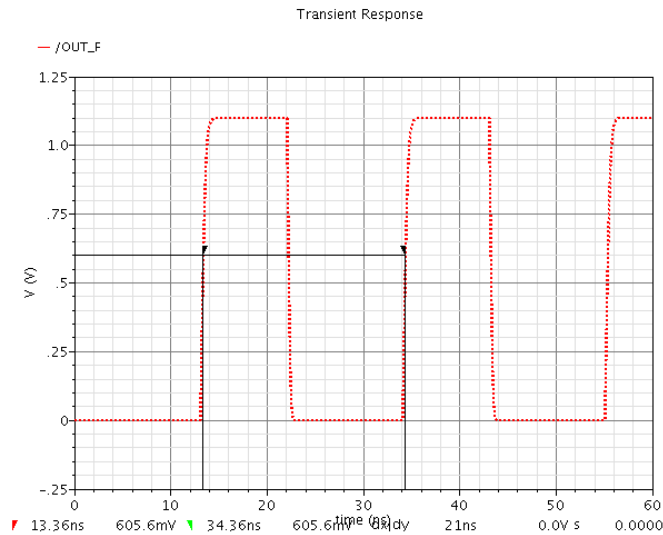


Fig. 6.29 Overall Frequency Divider Layout Extracted Output For 10 GHz Input Signal and A Frequency Division of 210

The temperature of the circuit was changed to 75°C under the exact same simulation conditions as before. Figure 6.30 shows the output of the divider circuit at a temperature of 75°C. As can be seen, the circuit reproduces the correct waveform with the exact same period as before.

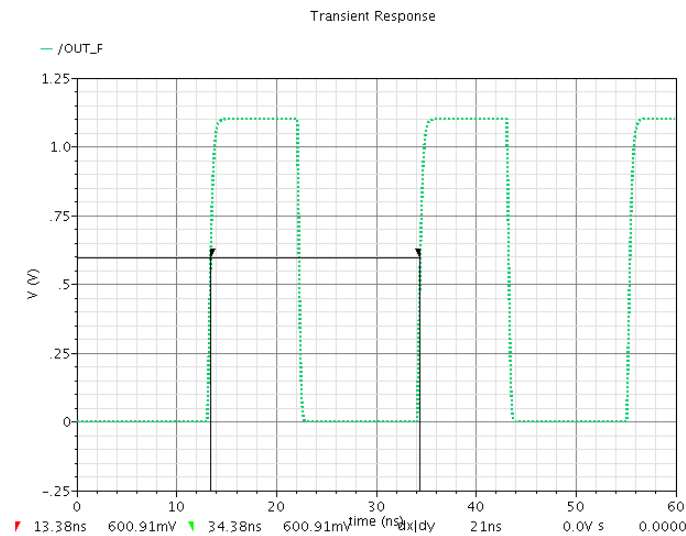


Fig. 6.30 Overall Frequency Divider Layout Extracted Output For 10 GHz Input Signal and A Frequency Division of 210 At Temperature of 75°C

The jitter of the overall extracted layout frequency divider was not able to be characterize. Issues with the PSS convergence and PNOISE analysis were encountered in Cadence. In addition, a great amount of time would be needed if simulation convergence would be guaranteed, since the division ratio is relatively high, lots harmonics are required to be computed.

6.5 Conclusion

The topology used to realize the fractional-N divider was extensively investigated. This chapter has shown the mathematical model for the phase noise injected by the delta sigma modulator and reported a means of simulating in Verilog-A, a fractional-N synthesizer by embedding the fractinal-N divider inside the VCO. Furthermore, a 0 to 18 GHz programmable frequency divider in CMOS 65 nm was discuss and results from layout reported. The next chapter talks about the experimental results of this programmable frequency divider as well as a 28 GHz VCO in IBM 0.13 μ m CMOS.

Chapter 7

Experimental Results

This chapter reports on the results obtained from measuring the 28-GHz VCO fabricated in a $0.13\mu\text{m}$ CMOS process from IBM. The chapter also reports on the failure of the variable frequency divider IC that was fabricated in the 65 nm CMOS process from TSMC.

7.1 Fabricated Chip Results

The layout of chip containing the LC-tank VCO fabricated the $0.13\mu\text{m}$ CMOS process is illustrated in 7.1. Two designs were included, one with and one without power amplifiers.

The power amplifiers were included in lieu of a failure of the normal VCO, however such was not the case. The power amplifiers would require extensive work on providing the DC power through a bond wire acting as a waveguide on the PCB; as always, line matching would also need to be considered. This is so because the output load of the last stage of the power amplifier, which consists of a common source, has to be connect outside the chip. This permits to tune the inductance value outside the chip to match for a higher gain. Furthermore, the matching circuit for the 50 ohm load of the power spectrum has to be implemented on the PCB as well.

The test setup for the VCO is shown in figure 7.2. The chip was not placed in a standard IC package, rather, the IC was directly wire-bonded to the PCB as shown in Figure 7.3. The idea was the package parasitics would be eliminated. Experimentally, this idea did not bore out in practice, as the bonding wires were quite long and introduces significant parasitics. A better option would have been to perform on chip probing or use a flip-chip pad.

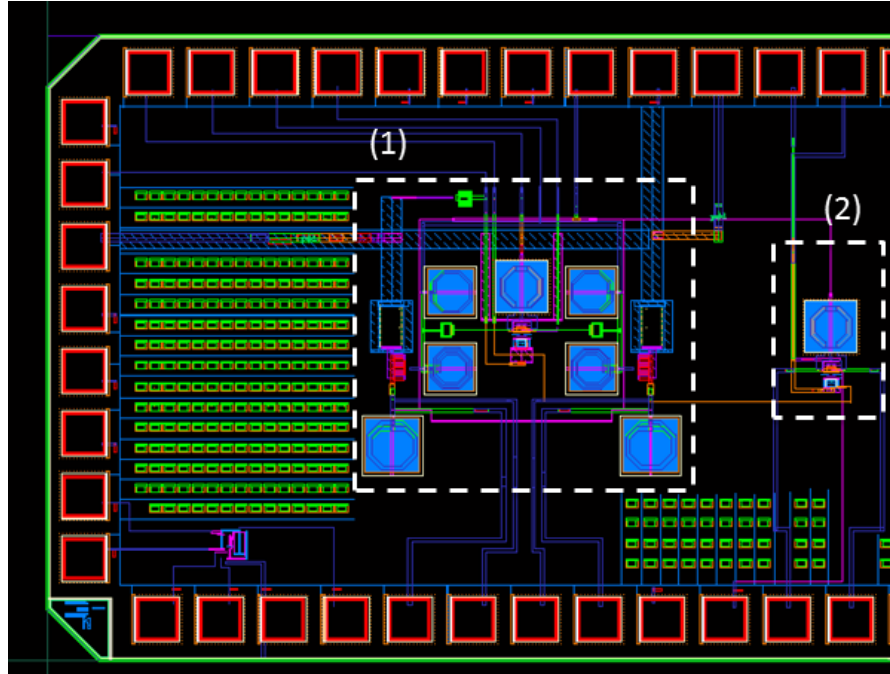


Fig. 7.1 Chip Layout Of The LC-Tank VCO (Two Different Variants Of the Same Design Are Shown) (1). *VCO With Power Amplifiers*, (2). *Stand-Alone VCO*

The PCB was designed by another teammate. Unfortunately, some fundamental high-frequency design principles were violated, as well as some routing errors introduced. First, the $2000\ \Omega$ resistor at the output of the VCO was not included and the power supply was accidentally wired to ground. In order to fix this, several vias had to be cut and a wire was re-routed to a region that allowed the insertion of some capacitors. While this procedure is quite normal for low-frequency applications, the additional wire length had a big impact on the performance of the VCO.

The PCB was constructed from Rogers dielectric material instead of FR4, as the Rogers material has a lower permittivity. The speed of propagation of wave is inversely proportional to the dielectric constant, hence higher frequency requires a lower dielectric value. Furthermore, the Rogers dielectric is more constant over a given surface area, hence maintaining its characteristics and the designed impedance of the micro-strip transmission line. One of the problems with the PCB is the lack of a properly terminated transmission line for the output of the VCO. Furthermore, the output is bent twice at 90° , causing unwanted reflections. A ground plane in the inner layer and a ground plane on the top layer should

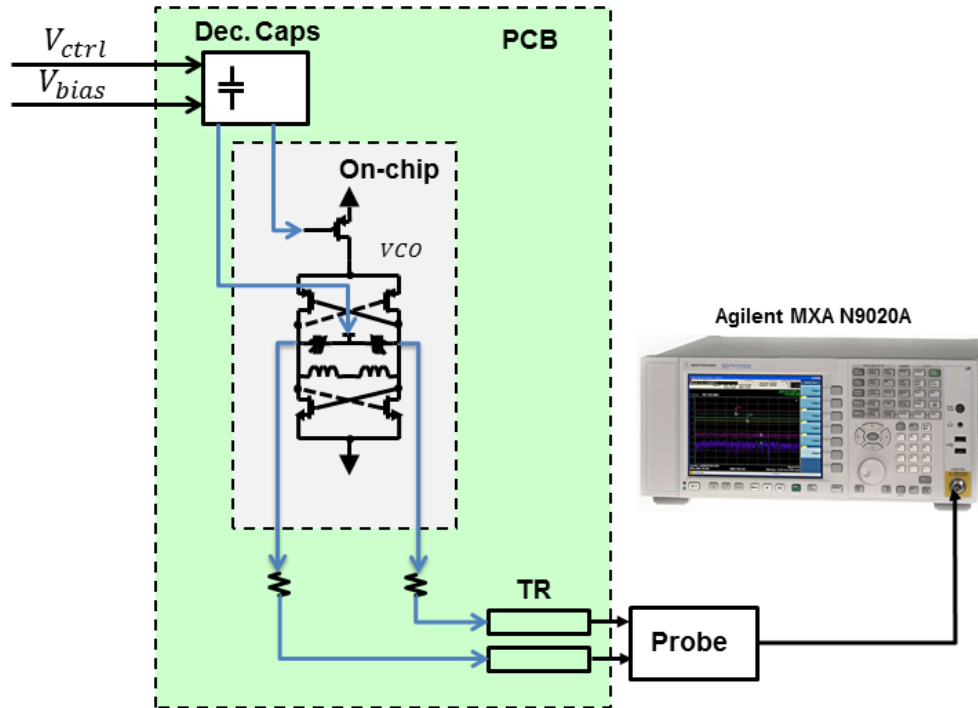


Fig. 7.2 VCO Test Setup

have been used. The inner ground plane is essential in maintaining a return path with low impedance for the return signal. Arrays of vias should have been placed at a distance of a quarter of a wavelength from the transmission line, and connected to ground, in order to decrease the inductance of the return path. This PCB provides a ground plane on the bottom layer, with very few vias connected to it. The ground plane is essential in sinking any outside interference signal with the board that could easily couple with the output signal. Finally, the capacitor banks are inter-connected by a small trace and are placed far apart from the high frequency power supply signal. This increases the series impedance of the capacitors and cancel their main effect - to provide a low impedance path to ground.

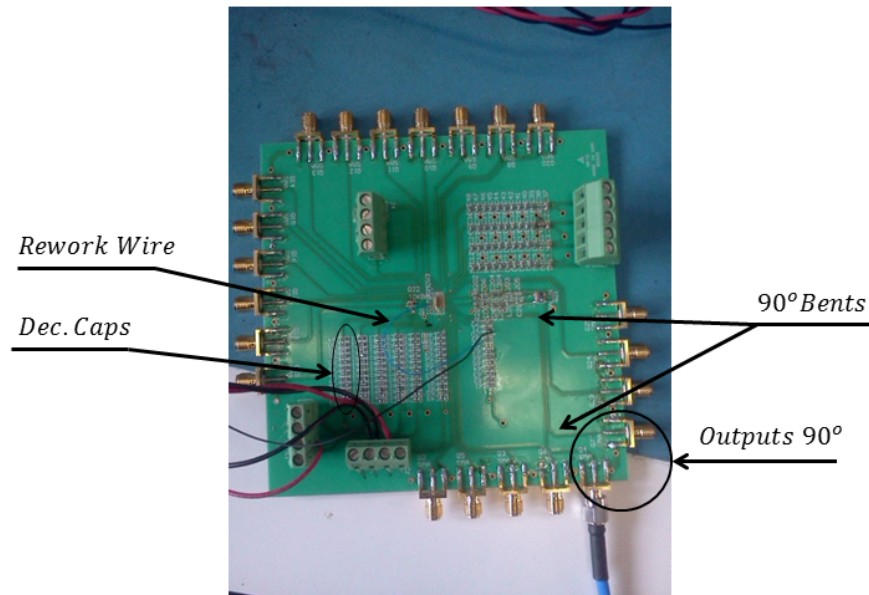


Fig. 7.3 IC Chip Bounded Directly To PCB

Figure 7.4 shows the output spectrum from the VCO. The main tone is located at a frequency of 22.384 GHz for a 0 V control voltage. The noise floor -60 dB while the signal is at -20 dB. For the reasons previously explained, the output power of the signal is small due to an improperly terminated transmission line and cable loss (a 2 feet cable was used to connect the output to the power spectrum).

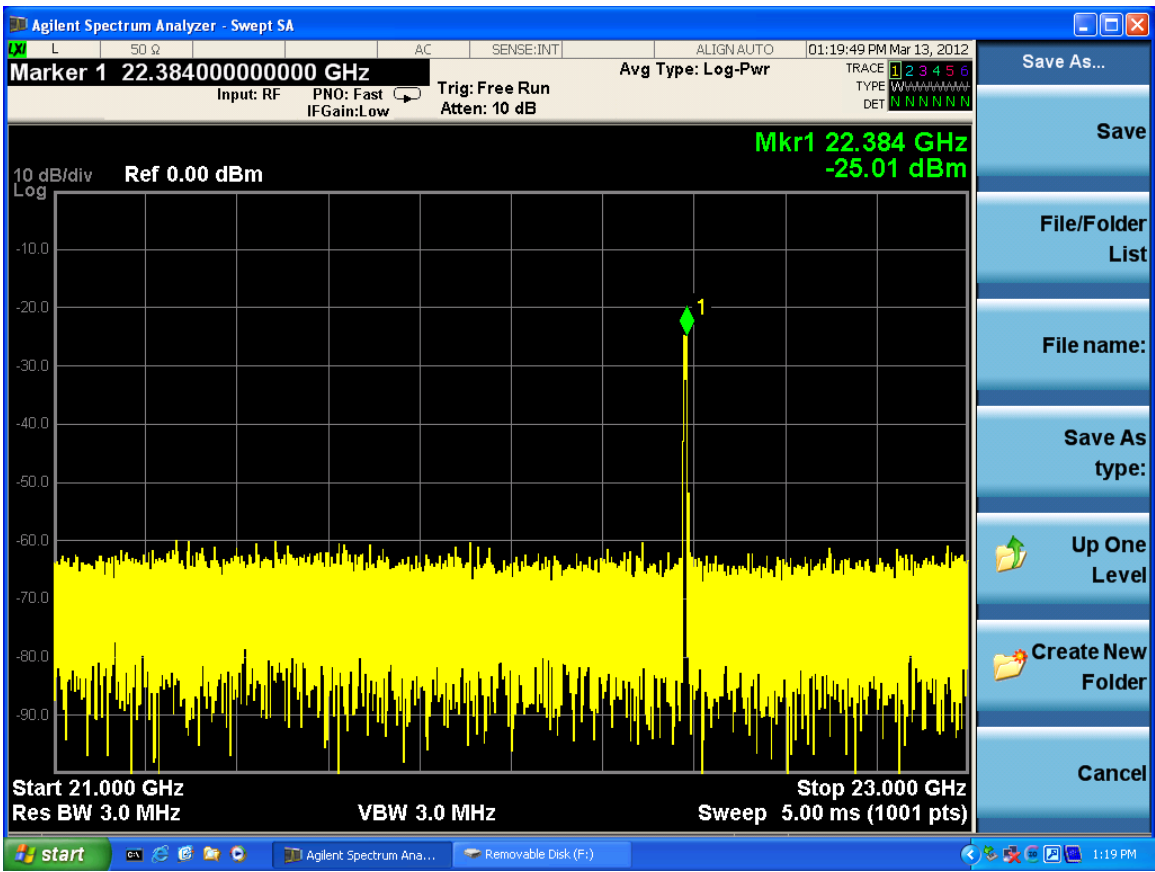


Fig. 7.4 Output Tone For Vctrl of Zero Volts

Figure 7.5 shows the output phase noise at a 300 kHz offset. The value changes a lot, and might be due to reflections present on the board, and interference with the other external signals. The average phase noise over a few samples is about -56 dBc/Hz. Table 7.1 shows the phase noise at various offset frequencies.

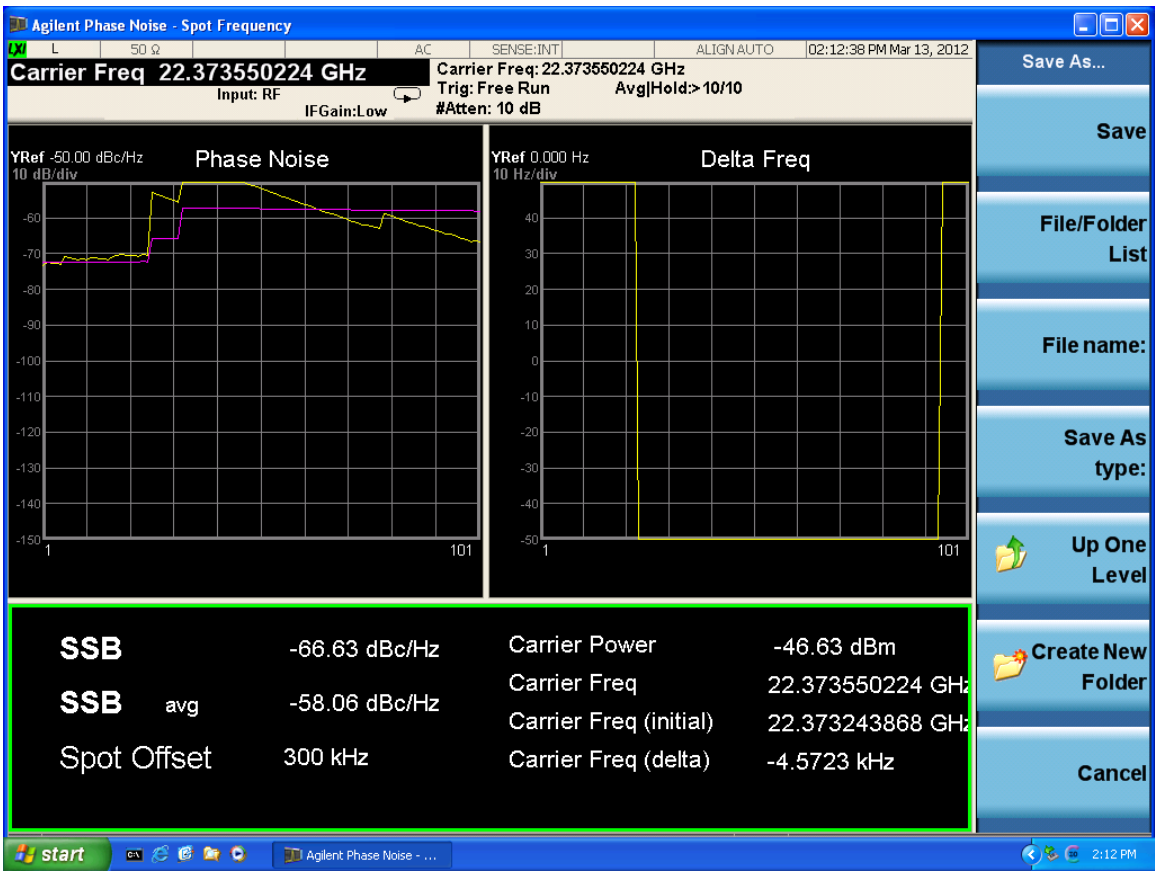
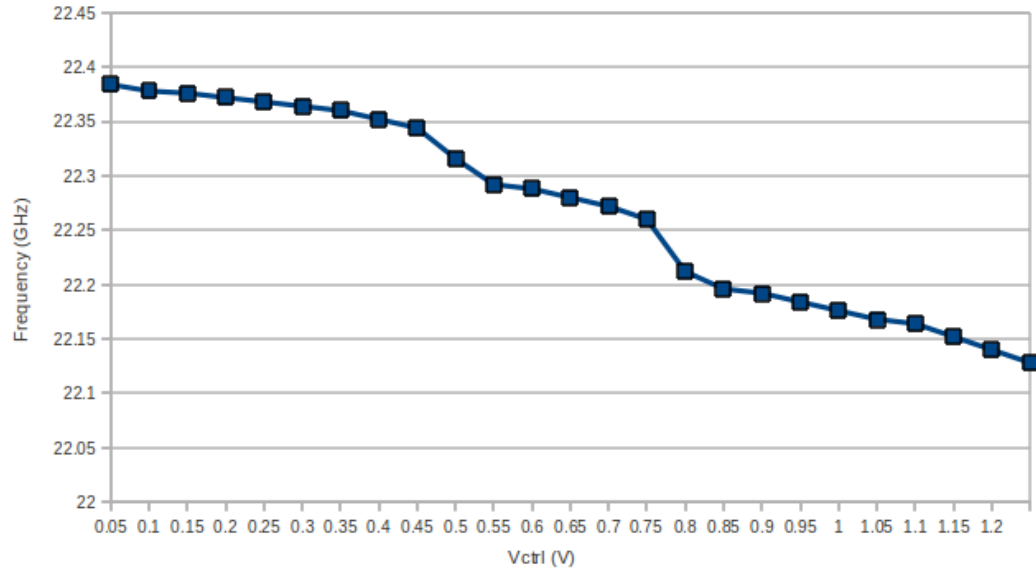


Fig. 7.5 Output Phase Noise At 300-KHz Offset

The VCO was also tested for its tuning range. The control voltage was swept from 0 to 1.2 volts by increments of 0.05 V. Figure 7.6 illustrates the results collected. Compared with the extracted layout results, this VCO oscillates at a frequency lower than expected. This we attribute to the additional capacitance associated with the output node. Also, the frequency span of the VCO is increased compared to extracted layout results. This latter behavior is not easily understood, and another better PCB should be fabricated in order to eliminate the possibilities of reflection and interference due to poor pcb layout.

Table 7.1 0.13 μm 23 GHz VCO Phase Noise

Offset Frequency	Phase Noise dBc/Hz
10 KHz	-40.4
100 KHz	-51.70
200 KHz	-53.02
300 KHz	-56.01
400 KHz	-66
500 KHz	-78
600 KHz	-78.54
700 KHz	-80
800 KHz	-80.21
900 KHz	-82.16

**Fig. 7.6** Measured Output Frequency vs Control Voltage For VCO

7.2 65 nm 0-18 GHz Frequency Divider

The 0-18 GHz, 4-to-7 modulus frequency divider preceded with 6 bits of programmable divider was fabricated in the 65 nm CMOS process. A microphotograph of the die showing the frequency divider portion is shown in Figure 7.7. Since a high frequency input clock

signal is needed, the pads were adjusted to enable a probe station to be used. CMC facilities in Manitoba are the only ones to offer the testing capabilities need to characterize this frequency divider. Through our contacts, we were able to have the chip tested in an RF facility in China. Unfortunately, however, the pads were shorted to the substrate and hence no data could be extracted from the IC.

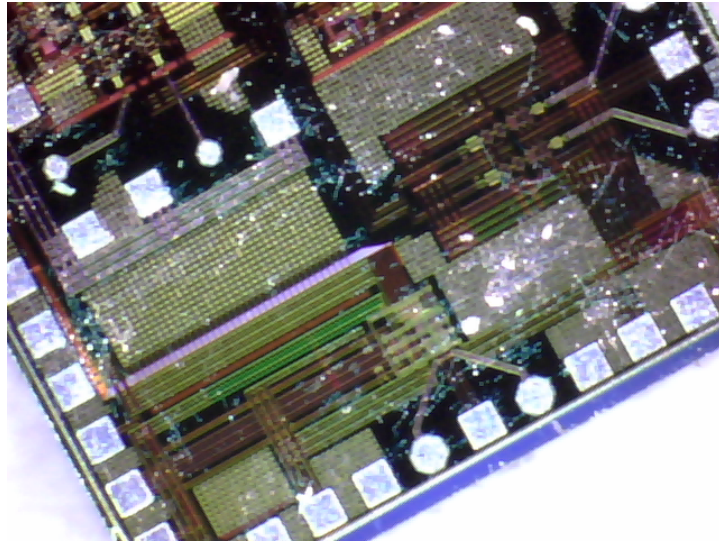


Fig. 7.7 Microphotograph of A Portion Of The IC Containing the 4-to-7 Modulus Frequency Divider

7.3 Conclusion

Bonding the IC directly to a PCB was not a good idea. The parasitics from the long bonding wires defeat any performance gain thought possible by avoiding a package. Rather the next chip should be bonded to the PCB through a flip-chip package, or on-chip probing be used, or on-chip probe be insert on the same silicon as the VCO. In addition, the distribution of the pads for the IC should be better controlled into to reduce inductance in the signal and power paths. GND pads should be placed near power supplies pins and control voltage pins.

Chapter 8

Conclusion

This thesis, through various demonstrations, has provided a series of design steps for constructing a wide-ranging high-frequency PLL in a 65 nm TSMC CMOS technology for application as a frequency synthesizer in the 30 - 40 GHz frequency range with a step resolution of 10 Hz. Design, layout and simulation of the phase-frequency detector, charge-pump, VCO, loop-filter and fractional-N divider was provided. In order to expedite the simulation of the PLL, a complete model of the PLL was created using the high-level description language Verilog-A available in the Cadence design tool. This model enabled the optimization of the individual components of the overall design in a timely manner. A prototype of this design was sent for fabrication. Unfortunately, due to a direct short between the bonding pads and the substrate (later identified to be a problem with the ESD structure) this prototype failed to operate.

Prior to the project moving into the TSMC 65 nm CMOS process, the original project had targeted a 0.13m CMOS process from IBM. However, from the onset of this project when developing the VCO, it became apparent that this technology was not capable of reaching the desired performance specifications. Silicon results were collected from a prototype of the VCO implemented in the IBM process. The IC was directly bonded to a printed circuit board that interfaced to a set of bench top equipment. Through experimentation the VCO was capable of oscillating at a frequency of 23.384 GHz with a phase noise of -51.7 dBc/Hz at a 100 kHz offset.

8.1 Thesis Summary

Beginning in Chapter 2, a mathematical model of the linear behaviour of a PLL, the key performance metrics of the PLL were described. This includes such measures as frequency response, step response and noise behaviour.

Chapter 3 provides a design procedure for developing a wide-ranging VCO for a high-frequency PLL application. Two separate designs of the VCO intended for fabrication in a 0.13 μm CMOS process from IBM and another in a 65 nm CMOS process from TSMC was provided. Simulation results at the schematic and layout level were provided and compared. In addition, a high-level description Verilog-A model of the VCO was outlined. This simulation model was used in Chapter 5 and 6 to simulate the operation of the PLL in closed-loop.

Chapter 4 discussed the design of the PFD and CP. A rather elaborate CP circuit was developed from a previous design found in the literature. This circuit was shown to be more robust and less sensitive to transistor non-idealities. Both the PFD and CP were laid out in a 65 nm CMOS process. The simulations results from both the schematic level and layout level were compared. A Verilog-A model for the PFD and CP is also provided. Close attention to the noise behaviour of these two elements is made.

Chapter 5 developed a design procedure in which to select the characteristics of the loop filter based on the PLL phase noise characteristics. The impact of the nonlinear behaviour of the VCO on the closed-loop operation of the PLL was explored, specifically from the perspective of the desired phase noise. It became quite clear from this study that a VCO with a linear transfer characteristic is important for a wide-ranging high-frequency PLL. The work in this chapter relied heavy on the Verilog-A models for the PFD/CP and VCO developed in the previous chapters, as well as the model for the loop-filter developed in this chapter.

Chapter 6 provides a description of the fractional-N frequency synthesis method that uses a $\Delta\Sigma$ modulator in the feedback path of the PLL. In this chapter the theory of delta sigma modulation and the mathematical model for the phase noise introduced by the $\Delta\Sigma$ modulator is described. The analysis includes two situations, one where the $\Delta\Sigma$ modulator is placed in front of the PLL and the other where the $\Delta\Sigma$ modulator is placed in the feedback path of the PLL. A working $\Delta\Sigma$ modulator along with a switching division factor VCO were introduced and modeled using the Verilog-A description language. Also provided

in this chapter was a description of the circuit and layout of a high-frequency 0-to-18 GHz, 4-to-7 modulus and 6-bits programmable divider in a 65nm CMOS process. The resolution of the counter is given as $\text{frequency}/(\text{modulus} \cdot 3843)$.

Finally, Chapter 7 provides some experimental results for the VCO that was fabricated and tested in 0.13m CMOS technology. The die consisting of the high-frequency fractional-N divider did not function on account of a direct short to ground. Reason for why this happened were provided.

Finally, the thesis concludes in Chapter 8 and provides some thought for future work.

8.2 Future Work

This thesis provided the base foundation that permit the simulation of PLLs in Verilog-A. This high level simulation is crucial in shorting the simulation time and predicting the output phase noise or transient responses of an overall PLL with a given loop filter. All the Verilog-A files were testes and the overall Verilog-A PLL with ideal components was compared with results derived using the Laplace transfer function in order to assert the overall Verilog-A setup. With this complete model, great innovation can be made by testing circuits that could compute the phase noise perhaps, or increase the robustness of the overall PLL by providing closed loop control for the VCO amplitude, counter balancing the effects of the non-linear VCO gain, etc. Furthermore, with this model, more complex PLL structures can be verified and the integration time reduced by providing a means to pin point different component characteristics that have to be met for a general overall requirement.

Appendix A

$\Delta\Sigma$ Modulators

A.1 $\Delta\Sigma$ Modulators

Different type of modulators exist. These modulators are categorized by their noise transfer function pole placement. The noise transfer function is the transfer function that the quantization noise power gets mapped with at the output of the modulator. This concept will be further explain along. The tree main categories of $\Delta\Sigma$ modulators are the:

1. All Zero Pole Realization
2. Fixed Pole/Zero Realization
3. Arbitrary Pole-Zero Realization

A.1.1 All Zero Pole Realization

This category of modulators consists of modulators composed uniquely of first order integrators, most often in a parallel like realization. The two most common ones are the Mash-3 and the Mash1-2 [5]. The Mash-3 and Mash1-2 are $\Delta\Sigma$ modulators are composed of series connected and/or paralleled first order integrators. The structure of a third order Mash3 delta sigma modulator is shown in figure A.1. The operation of the Mash-III is as follows: with zero initial conditions,the input $X(z)$ propagates to the input of the 1-bit quantizer. The quantizer's output assumes values of +1 and -1. The threshold crossing is usually from 0 to 1, and from -0 to -1. The output of the quantizer is fed-back and subtracted from the input. The integrator inside the loop continues to integrate the error. The

error between the quantizer output and its present input is fed as the input to another first order $\Delta\Sigma$ modulator. The same operation principle applies here. Figure A.2, illustrates the linear model of the MASH-III that permits the analysis in the discrete time domain of the transfer functions for the quantization noise, here denoted as e , and the input to output transfer functions. The non-linear comparators are replaced with their equivalent uniform noise injection signal that is calculated as below. Here, y represents the maximum spanning range of the quantizer.

$$\Delta = \frac{2 \cdot y}{(2^{bits} - 1)} \quad (A.1)$$

$$e_{rms} = \frac{\Delta^2}{12} \quad (A.2)$$

Since the quantizers are the same for each section of the MASH-III modulator, the in-

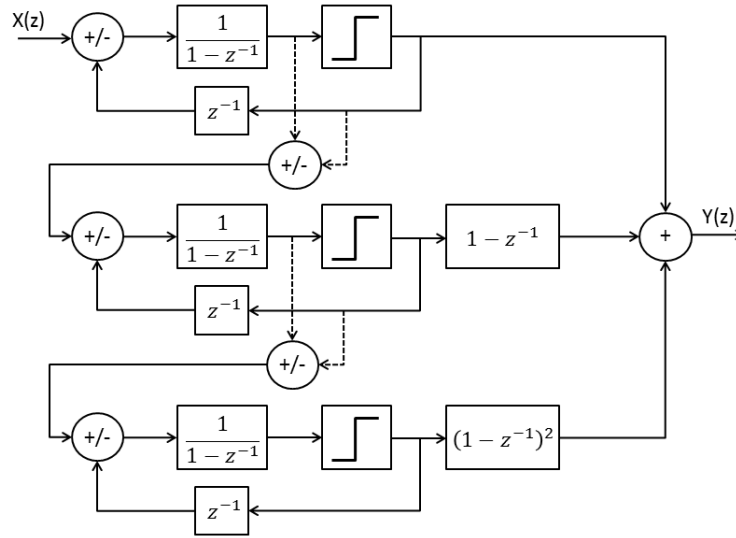


Fig. A.1 Mash-III $\Delta\Sigma$ Modulator

roduced quantization error equals the sum of the transfer functions from each of these injection nodes to the output. The discrete time Z domain transfer function for the input and overall quantization noise is reported in equation A.3 and was taken from [5]. From this equation, the input to output transfer function is an all-pass while the quantization

noise is a high-pass with +30dB rising slope. The MashI-II is discussed next.

$$Y(z) = X(z) + E(z)(1 - z^{-1})^3. \quad (\text{A.3})$$

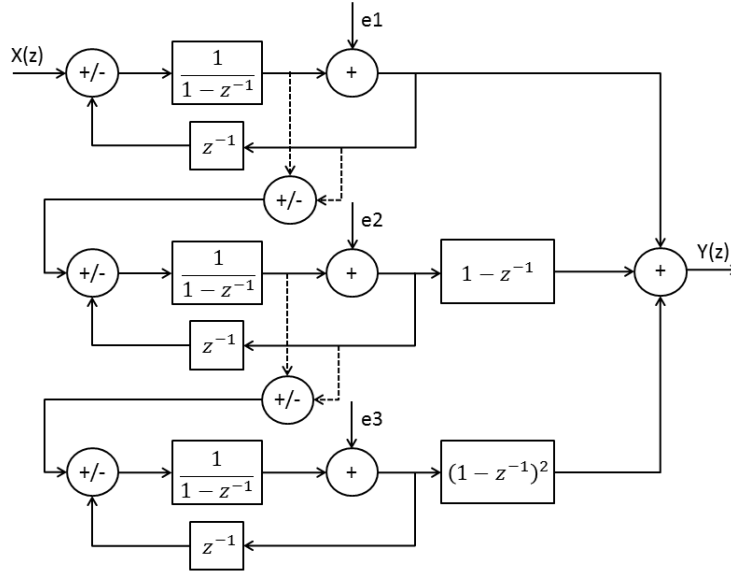


Fig. A.2 Mash-III $\Delta\Sigma$ Modulator Equivalent Linear Model

The third order, MashI-II topology resembles the MASH-III but instead of using only first order modulators, it incorporates a second order, made by cascading two first order modulators. The circuit is shown in figure A.3. The noise transfer function for the former is the same as the previous one as well as for the input to output. The advantages of the MASHI-II modulator is reduced hardware compared to MASH-III. Other benefits of this structure will be discuss after the next two structures are introduced.

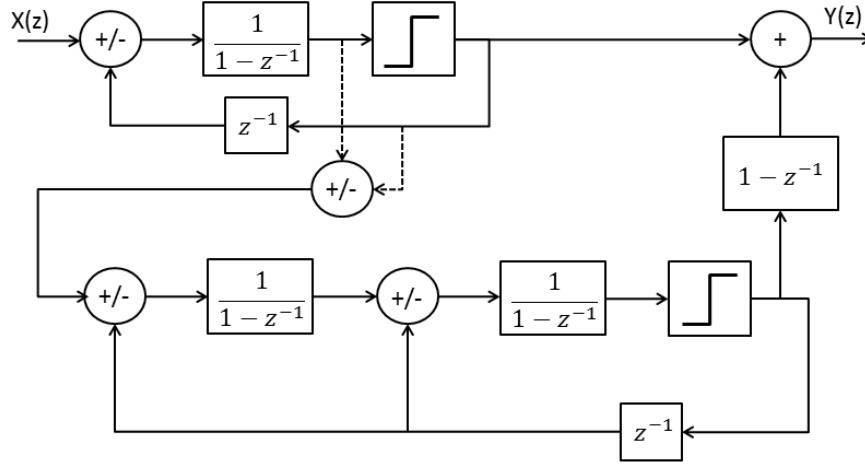


Fig. A.3 MashI-II $\Delta\Sigma$ Modulator

A.1.2 Fixed Pole/Zero Realization

The next $\Delta\Sigma$ modulator, contains more hardware, and is slightly more complex. The diagram of the former is illustrated in picture A.4. Compared to the two previous modulators, the SSMF-II contains coefficients and three cascaded integrators. The transfer function for the input and quantization noise is derived as:

$$Y(z) = X(z) \frac{2z^{-1} - 2.5z^{-2} + z^{-3}}{1 - 3z^{-1} + 0.5z^{-2}} + e(z) \frac{(1 - z^{-1})^3}{1 - 3z^{-1} + 0.5z^{-2}} \quad (\text{A.4})$$

The input is not an all pass anymore, instead, it has a gain shaped as high pass. Unity gain is constant up to fs/20000. This structure is henceforth not suitable for modulated fractional synthesis above the unity gain pass band. The noise transfer function compared with the previous structures, has Butter-worth poles that reduce the quantization noise power in the high-pass region and hence, decreases the total power available for the input in the pass band.

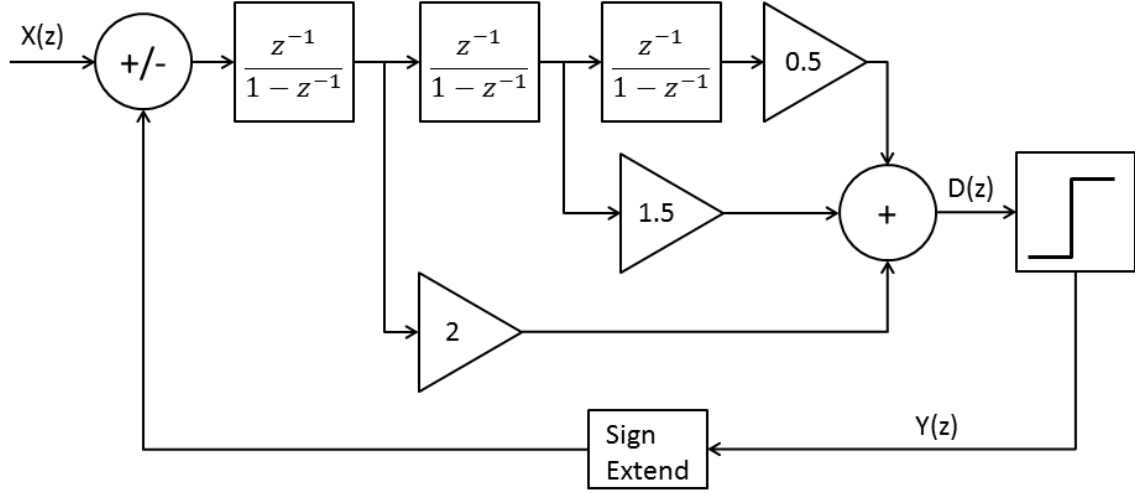


Fig. A.4 SSMF-II $\Delta\Sigma$ Modulator

A.1.3 Arbitrary Pole-Zero Realization

An arbitrary poles-zero realized $\Delta\Sigma$, shapes the noise transfer function by spreading poles near DC. The MASH modulators, has all its poles at zero and hence most of the quantization power is pushed in the high pass band. The following theory is taken from [11] and [10]. The total quantization noise power input in a delta sigma modulator comes from the quantizer and its total rms value is computed as given previously in equation A.2. This quantization error power will be multiplied by the NTF transfer function of the modulator, yielding the total noise power at the output of the $\Delta\Sigma$:

$$P_{e_total} = \frac{P_e}{2\pi} \int_{-\pi}^{\pi} |NTF(e^{j\theta})|^2 d\theta = A \cdot P_e \quad (\text{A.5})$$

The constant A depends on the NTF order, the higher the order, the higher the A constant and hence the higher the overall quantization error power. Assuming a low-pass NTF, the total in-band quantization error power, defined as the quantized error present in the pass band at the output of the modulator, corresponds to the quantization error times the magnitude square of the NTF in the pass band

$$P_{e_in} = \frac{P_e}{\pi} \int_0^{\pi/OSR} |NTF(e^{j\theta})|^2 d\theta. \quad (\text{A.6})$$

The pass band region is related to the over sampling ratio of the modulator by equation

$$f_{in.band} = \frac{0.5}{OSR} \cdot f_s \quad (A.7)$$

Hence, the higher the OSR, the smaller the allowed encoded input frequency bandwidth. From these equations, there seems to be a relationship between the in-band quantization power, OSR and total noise quantization power. This relationship is more easily described using the illustration in [8] and shown in figure A.5. In a modulator, as the OSR increases, the quantization power in the noise frequency band, increases. Since the overall power at the delta sigma output is constant and is given by the maximum quantizer output (δ), the power in the signal band, decreases. In the graph, this is illustrated with the arrows direction. Since the quantization noise in the signal band decreases, the SNR at the output also augments.

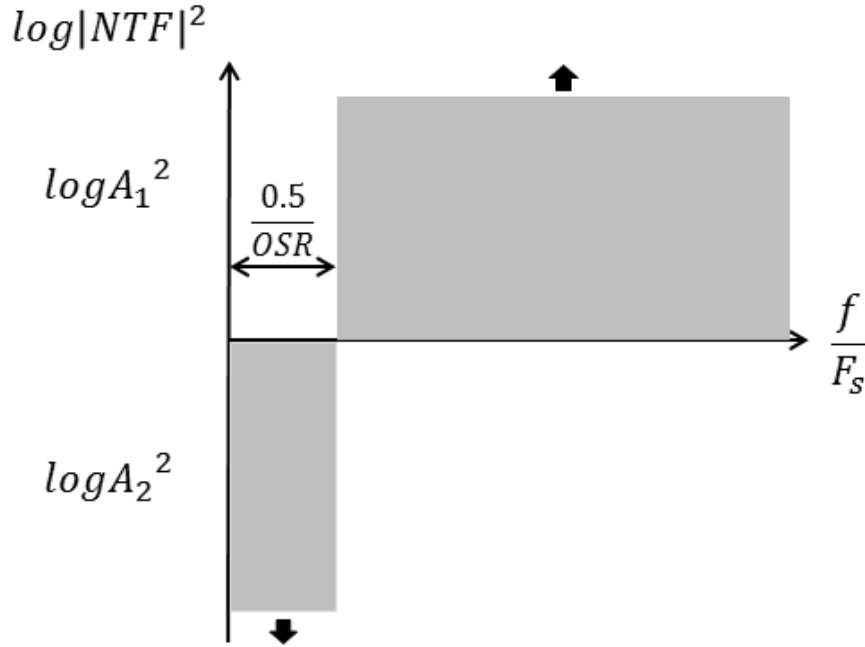


Fig. A.5 Power Density in NTF Pass Band and Quantization Band

Defining P_x as the signal power and δ as the total power available in the $\Delta\Sigma$, which is usually determined by the quantizer maximum output, the maximum SNR is re-written as

[10]:

$$SNR_{max} = \frac{\delta^2 - P_{e_total}}{P_{e_in}} \quad (\text{A.8})$$

This equation states that as the total noise quantization power, that depends on the constant A previously defined in A.5, increases, the SNR decreases. Furthermore, as the in-band quantization noise augments, the same effect is observed. A relationship between the maximum SNR and the quantization noise power, as well as the A constant using the graph A.5 is derived in [8] and reported below.

$$SNR_{max} = e \cdot \frac{OSR}{P_e} A^{-(R-1)} (\delta^2 - P_e \cdot A) \quad (\text{A.9})$$

As the NTF order increases, the P_{e_total} goes up, which in turn decreases the power available for the input or P_x (from equation A.8). The tradeoffs are apparent, a higher SNR will yield a lower dynamic range for the input signal. For the MASH modulators described previously, because of their NTF, most of the quantization noise is present in the higher frequency band, and hence the in-band power noise is very small, as an effect, these modulators are optimized for SNR only. However, since most of these modulators have multiple output bits (at least equal to their order), they still out-perform a synthesized equivalent order 1-bit $\Delta\Sigma$ modulator.

The zero locations in a synthesized NTF are located as described in reference [11], that is on the unit circle. These poles correspond to Chebyshev roots, since they yield the minimum quantization error in the pass band. The pole locations on the other hand, are usually optimized for highest SNR. These optimizations are carried away based on stability criterion and sometimes stability levels that involve optimized solutions (that have been searched for through extensive simulations). The results of such simulations are given in reference [10]. The next picture shows the poles and zero general placement region on the unit circle. Different topologies exist for the implementation of this type of modulator, such as the CIDF, CIDIDF and CIDIFF as given in references [10] and [11], however, the issues with these topologies lie in the quantization error affecting the input to output transfer function, hence in this thesis another structure given in [35] is used.

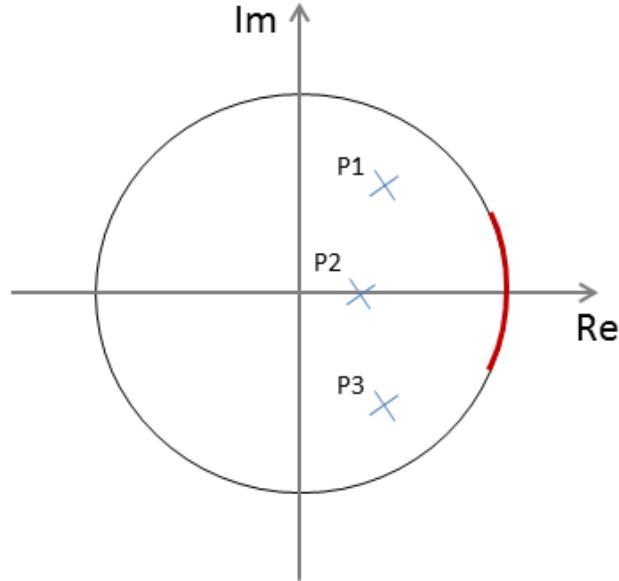


Fig. A.6 $\Delta\Sigma$ Poles and Zeros Usual Location Region

A.2 $\Delta\Sigma$ Topology Comparison

In this section, all previous architectures are compared. For the higher order synthesized NTF, the poles and zero location are given in table A.2. These poles and zeros were taken from page 94 of reference [10] for an OSR of 64. The magnitude frequency response of this NTF as well as the NTF of the previous architecture are obtained by running simulations in Scilab and Xcos environment, and are pictured in figure A.7.

Table A.1 NTF $\Delta\Sigma$ Poles and Zeros
 $\Delta\Sigma$ NTF Poles/Zeros

Zeros	1 , $0.9992772 \pm 0.0380139i$
Poles	-0.0285 , $0.9049 \pm 0.2073i$

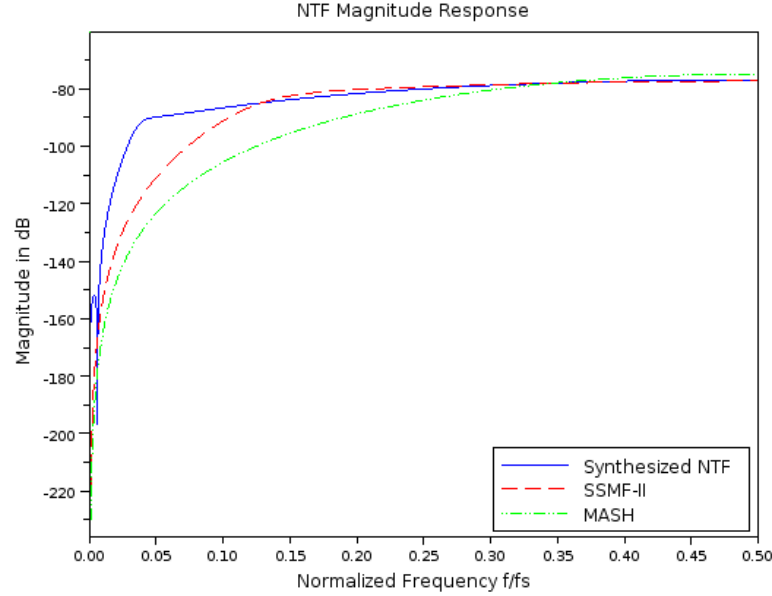


Fig. A.7 NTF of $\Delta\Sigma$ Modulator Magnitude Response

All these noise transfer functions are of third order. By looking at figure A.7, the MASH NTF and SSMF-II rises more quickly than the synthesized NTF. Furthermore, the in band error quantization power is less for these topologies. However, as previously mentioned, the NTF magnitude response increase at higher frequencies compared with the synthesized NTF, meaning, most noise power is concentrated in the high pass band (quantization error band), therefore increasing the SNR. However, this in turn can potentially lower the maximum input signal that can be encoded. In order to carry away the comparison process, the designed third order synthesized NTF was implemented with Xavier's topology previously introduced. This topology offers the unique all-pass transfer function for the input to output and permits the synthesis of any desired NTF characteristic function.

The previous $\Delta\Sigma$ structures were implemented in Xcos, Scilab Simulink's like editor and simulator and were characterized for in-band noise total quantization power, SNR and spurious content. The DC input characterization is an important metric, as most of the modulators in a fractional synthesizer are used for a fixed DC input to generate a fractional frequency. Furthermore, dc inputs is useful in analyzing the stability of the system. The next figure A.8 portrays the total in-band quantization noise for the different $\Delta\Sigma$ topologies.

As expected, the in-band noise of the synthesized NTF is the biggest, but, this topology along with the SSMF-II, in turn, present a more stable, less oscillatory behavior for the in-band noise power than the MASH structures. The oscillatory behavior and non linear response near the edges of the MASH, are explained by the next metric, which takes into consideration the spurs.

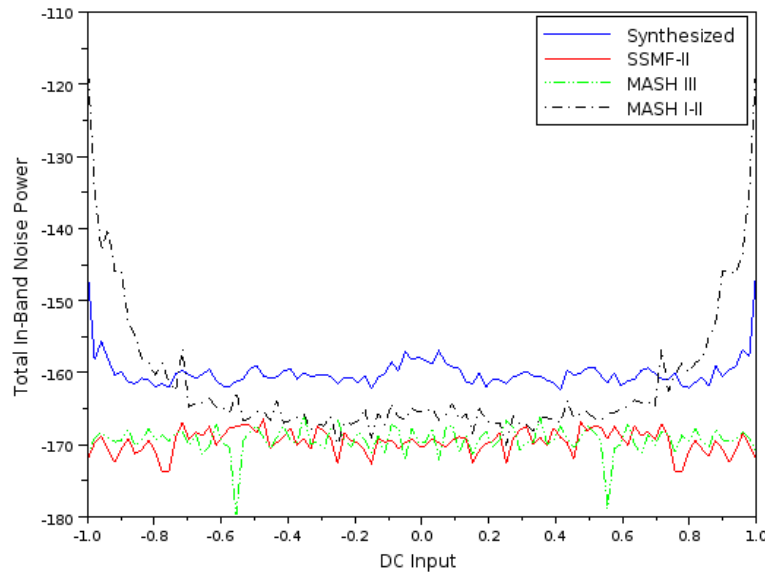


Fig. A.8 In-Band Total Power Noise for Different $\Delta\Sigma$ Modulators

When a $\Delta\Sigma$ modulator is presented with a DC input, the integrators, depending on the randomize efficiency, might see a periodic error repeating at given sub-harmonics of the main sampling frequency. This phenomena gives rise to spurs. Spurs are undesirable for fractional synthesis since they increase power consumption, and might feed-through the PFD and filter, and find themselves influencing the jitter at the output of the PLL. Furthermore, spurs may influence the SNR. Figure A.9, shows the spurs phenomenon for a SSMF-II topology.

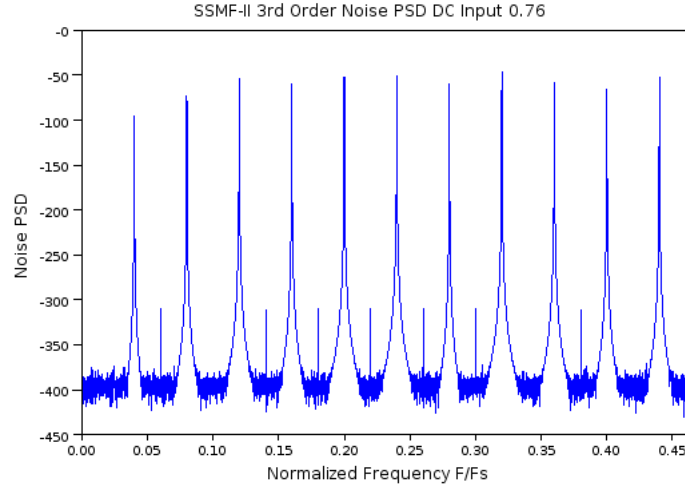


Fig. A.9 Spurs For DC Encoded Signal for SSMF-II $\Delta\Sigma$ s

All topologies exhibit spurs, however for the synthesized NTF, the spurs only begin at a dc higher than 0.65, and the number of spurs is somewhat less than the other topologies. The SSMF-II exhibits spurs for dc values, but to a lesser degree than MASH-III or MASH-II topologies. Hence, for PLL with lowest output spurs, it seems higher order synthesized NTF are a good option. In [5] as well as [11] spurs reduction techniques through dithering are discussed. This dithering action is about injecting at the integrator node or input of the quantizer, random gaussian noise in low amplitude, such as to randomize the error, and hence lower the spurs level. The next metric measures the signal to noise ratio for a sine-wave encoded signal. While this metric might not seem necessary for fractional synthesis, in some cases, frequency modulation is desired. Graph A.10 illustrates the SNR versus the input power of a sine-wave within the in-band frequency of the synthesized NTF. The results suggest higher SNR can be achieved with MASH modulators or SSMF-II. Recall equation A.4, the SSMF-II, amplifies the encoded input signal, hence, the SNR seems to be higher than the rest of the architectures, while in fact it is slightly lower than the MASH modulator. For highest SNR, the MASH modulators should be used, or the order of the synthesized NTF increased, in order to achieve equivalent SNR as explained in [10].

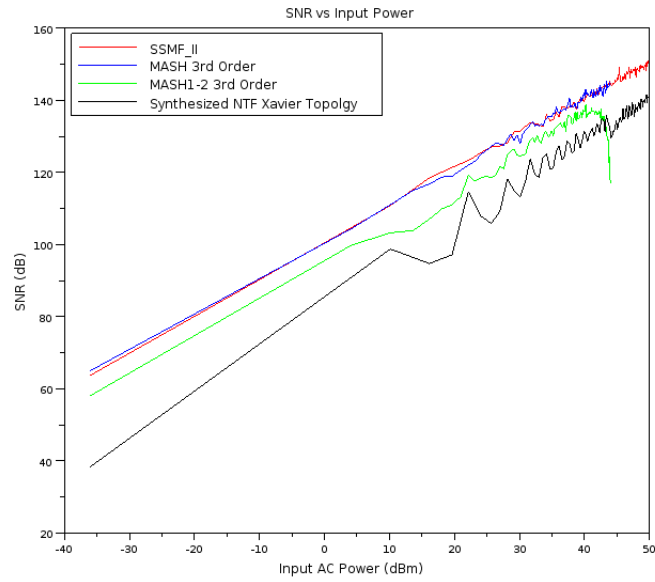


Fig. A.10 SNR VS AC Input Power For Different $\Delta\Sigma$ s

Appendix B

Optimized Arbitrary Zero-Pole Placement $\Delta\Sigma$ Structure

The digital implementation of higher order synthesized modulators can be optimized for the number of bits required versus the maximum error desired. In Xavier's topology [35], the structure is very slow, and in order to improve its speed, the minimization of the numbers of shift and add operations required in the filter can improve the former. In Xavier's thesis, the proposed structure looks like in figure B.1. The numbers on each node corresponds to the number of bits required for the integer part, in red, and the fractional part in blue. This bits were optimized using the algorithm in [36] and with help of the author of the latter. In order to use this algorithm, the transfer functions from each internal node to the output and the like were derived using the symbolic tool in pythong, "sympy".

To lower the number of bits required to even lesser degree, the previous structure was modified slightly. The modifications were done based on the results from the optimization in the number of bits for the former topology. The B coefficients were tried to be moved away and included in the feedback in order to lower the amounts of bits required, since in a feedback, they would be less sensitive to variations hence, less bits would be required to yield the same amount of tolerable error. Figure B.2 illustrates the new proposed structure along with the optimized number of bits required to yield the same amount of error as the previous one. The gain in the number of bits is apparent when comparing both results. Furthermore, both structures were synthesized in ISE of Xilinx. A gain in frequency of 6 MHz was observed from the reports.

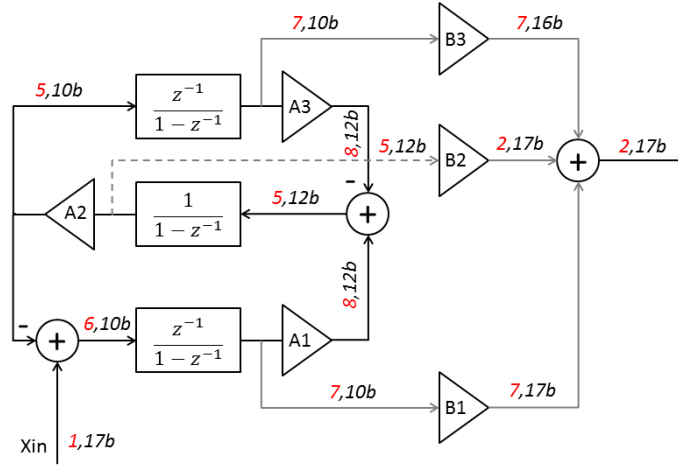


Fig. B.1 Higher Order Synthesized $\Sigma\Delta$ Modulator Filter Structure By Xavier

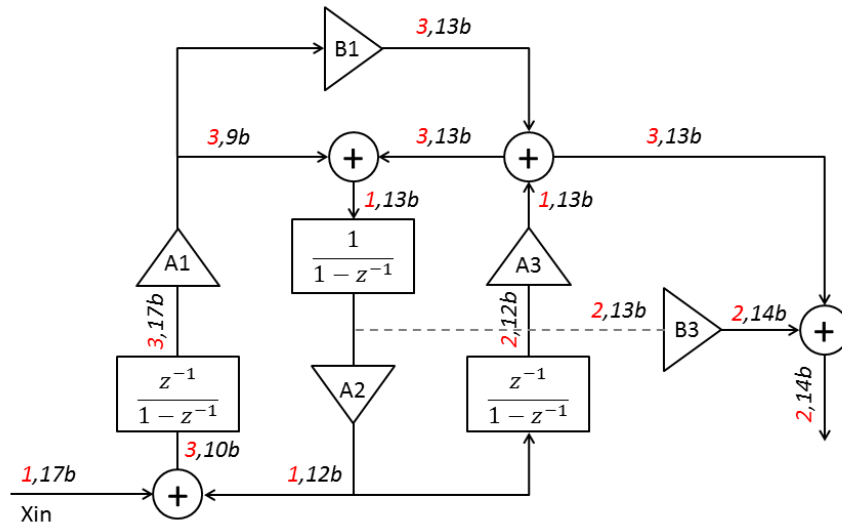


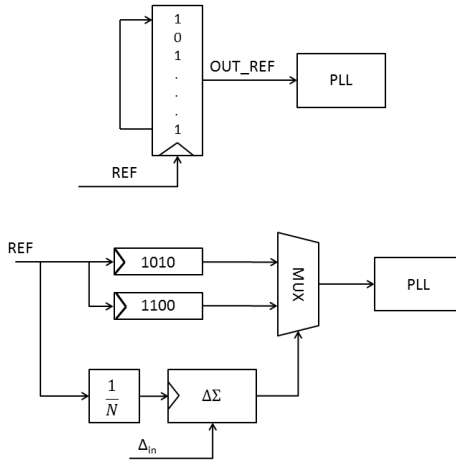
Fig. B.2 Higher Order Synthesized $\Sigma\Delta$ Modulator Optimized Filter Structure

Appendix C

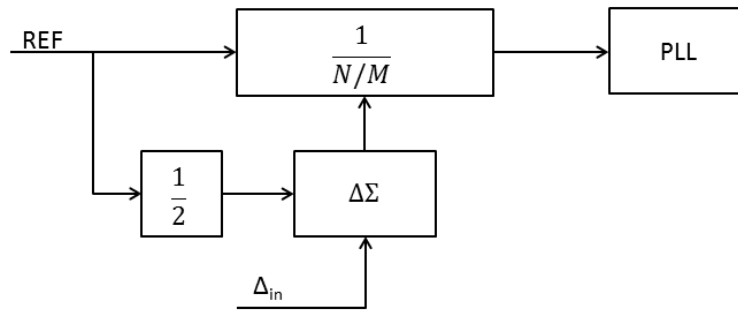
Front-End Delta Sigma Frequency Synthesis

C.0.1 Bit-Stream Fractional-N Frequency Synthesis

Recently, as explained in reference [37] another way of performing frequency synthesis is reported. This method is shown in figure C.1 along with the encoding principle. In this method, fractional synthesis is performed at the input of the PLL. A value is encoded using a delta-sigma and for each output of the $\Delta\Sigma$ bit stream, different encoded frequency bits are chosen using the select bit on the multiplexer. The $\Delta\Sigma$ clock needs to be slower by the amount that equals the number of bits needed for the frequency encoding. In this example, the frequency is $\text{REF}/2$ and $\text{REF}/4$. In the original paper, this topology is not used in a stand-alone mode, rather, the bits are placed in a ROM. The problem with that is the appearance of periodic spurs over-time due to the repetition of the same bit pattern. If a delta sigma is used instead as shown below, this pattern is no longer present (depending on the used $\Delta\Sigma$). The other issue with this method, is the need of a high frequency for the reference as the frequency spacing between the inputs of the multiplexer decreases.

**Fig. C.1** Bit-Stream Fractional Synthesis

In figure C.2, a derivation of the latter architecture is pictured. In this structure, a modulus divider instead of a multiplexer is used for the frequency modulation. The delta sigma needs to be clocked at least half the frequency of the highest output frequency. For instance, if N is unity, then the $\Delta\Sigma$ will be clocked by $\text{REF}/2$. M can assume any other division factor. This architecture is useful in implementing the front end fractional frequency synthesis in Verilog-A.

**Fig. C.2** Derived Front-End Fractional Synthesis

The implementation of the front-end divider uses the integral function in Verilog-A. The integral function takes as input a fixed frequency. A one is cross exactly in time at the period the frequency. The cross function then triggers an interrupt when the simulation

arrives at a specific point. In figure C.3, the integration as well as the interrupts are shown on the diagonal line. In addition, the Reference clock, along with its divide by two are illustrated. In this implementation, the delta sigma modulator reacts on the rising edge of the REF/2 signal. When a change in the $\Delta\Sigma$ output occurs, such as shown, the next interrupt interval is recomputed by changing the crossing point. This changes the time by a Δt . It is important to note that the system changes its next interval on the rising edge of its trigger. As illustrated, when the output of the modulator does not coincide with the rising edge of the output reference, the interval for the interrupt remains the same until the next change.

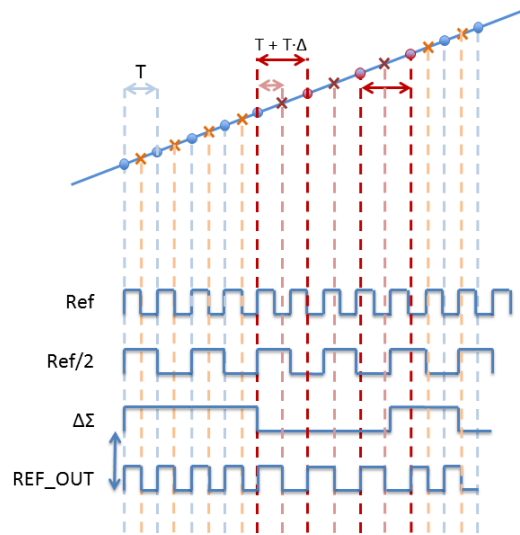


Fig. C.3 Front End Proposed Architecture Operation Principle

Below is the Verilog-A code that implements this fraction front end. The Delta-Sigma modulator is not shown.

```

phase = idt(freq,0);
@(cross(phase - next, +1, ttol)) begin
    next = next + 1 + floor(V(select)+0.5)*delta_freq;
    out = vhi;
end
@(cross(phase - next + (1 + floor(V(select) + 0.5)*delta_freq)/2, +1, ttol))begin
    out = vlow;
end

```

In order to alienate the waiting until the next rising edge, another implementation is proposed. In this method, a Δ value that represents the change in the next time period from a previous one (captured on the rising edge) and a new one, captured in the falling edge is included. This makes sure no output from the modulator is skipped while processing the phase difference. Furthermore, an if statement is provided in order to select the extension in the period of the present frequency. In order to use the code with a modulator that swings negative, the output of the latter needs to be offset to zero. To improve accuracy, a "bound_step" statement, that limits the maximum step size the simulator takes for the block was added, this greatly influenced the resolution. A fragment of the overall module in VerilogA is shown next.

```

phase = idt(freq,0);

if (floor(V(select)+0.5) == 3) k = 0.333333;
else if (floor(V(select)+0.5) == 2) k = 0.222222;
else if (floor(V(select)+0.5) == 1) k = 0.111111;
else k = 0;

$discontinuity(0);
$bound_step(1/(100*freq));

@(cross(phase - next - delta, +1, ttol)) begin
    next = next + delta;
    next = next + 1 + k;
    prev_k = k;
    out = vhi;
end

@(cross(phase - next + (1 + k)/2, +1, ttol)) begin
    delta = (k - prev_k);
    out = vlow;
end

```

The structure in figure C.2 may not be practical in real situation, due primarily to the integer division only. A multiplexer could be used, and two referenced frequency instead. The only thing that needs to be made sure about is the no smaller frequency will be produced while switching from one input to the other. Most of the FPGAs have clock multiplexer that only switch when the actual input has been low and a high to low transition

has been detected on the second switched input. This kind of multiplexer could be used for this architecture. In contrast, if the frequencies are farther apart, some bits from the modulator might get ignored. In the next section, the quantization power noise from $\Delta\Sigma$ modulators into phase noise process is discussed.

C.0.2 Phase Noise Of Front-End Fractional-Synthesizer

In this subsection, a model is derived for the fractional front end synthesizer. Previously, in figure C.3, a front end delta sigma modulator topology was shown. The derivation of the phase noise injected by the $\Sigma\Delta$ modulator can be deducted using the same concepts as the author of the the previous fractional synthesizer topology used. Without re-deriving the same equations again, the results will be remapped to the front end topology. First, when the delta sigma is placed up-front to synthesize a frequency, the reference frequency is encoded withing an average instead of the division factor. Using this idea, one can argue that the any other reference frequency not corresponding with the main encoded one, name it nominal frequency, a deviation in t_k as previously defined in equation 6.11 will occur. This time deviation will be translated into a phase noise as previously stated in equation 6.12. The figure below shows the equivalent phase model for the front end delta sigma modulator fractional-N synthesizer.

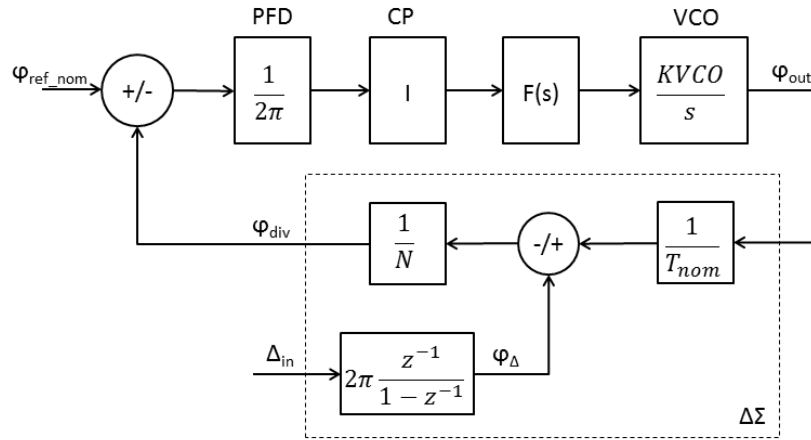


Fig. C.4 Front End Fractional-N $\Delta\Sigma$ Fractional Synthesizer Frequency Equivalent Model

Using this model, the only thing that needs to be properly map is the amount of

quantization noise power delivered by the modulators. For instance, the topology shown in C.3, injects more noise than the one with the modulator in a feedback loop. The reason for this discrepancy, lies in the necessity of the clock for the modulator to be at least half in frequency of the maximum output reference clock. This creates an over-sample of its spectrum and hence injects more phase noise. Before continuing, another structure for front-head delta sigma fractional frequency synthesis is proposed below.

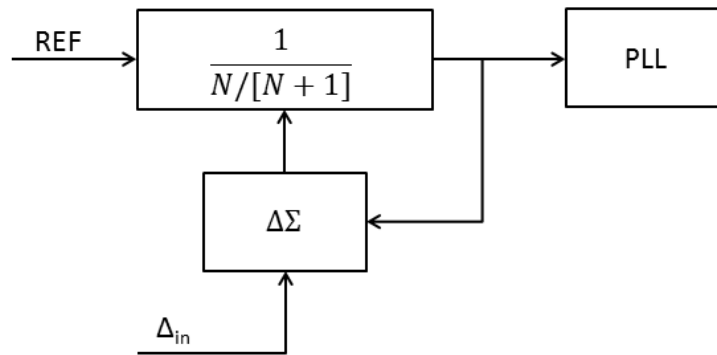


Fig. C.5 Front End Fractional-N $\Delta\Sigma$ Fractional Synthesizer Frequency Topology

This topology mimics the topology used for feedback fractional synthesis. The N and M values can be integers or fractional. For instance, a multiplexer can be used and two reference input frequencies selected or an integer divider. The important thing is that edges need to be aligned with respect to the rising edge. In order to compare this topology with the one in C.3, the output that feeds the PLL reference for both topologies is extracted and its spectrum computed. The output spectrum for the first front head method is pictured in figure C.6, followed by that of the former method in figure C.7.

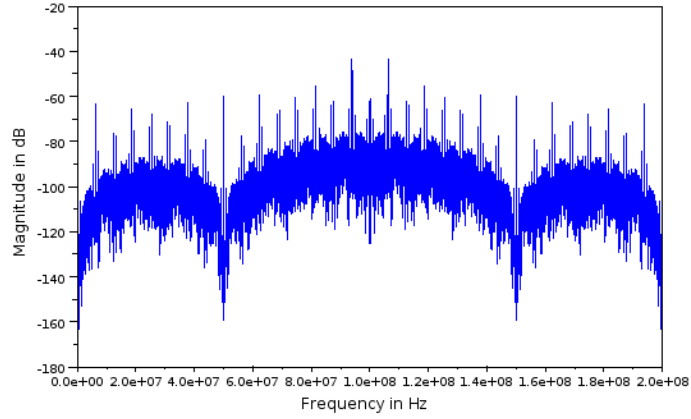


Fig. C.6 Front-Head $\Sigma\Delta$ Modulator Fractional Synthesis Output For Topology Shown In Figure C.3

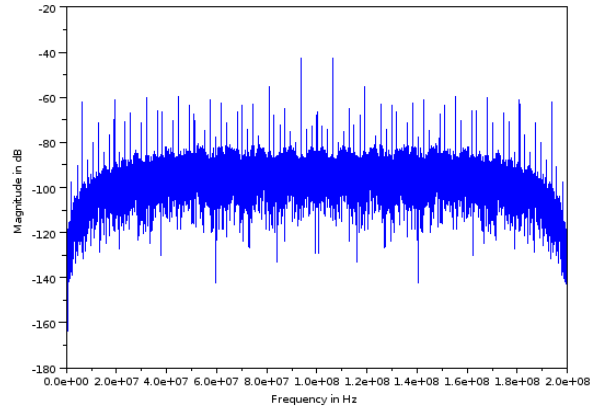


Fig. C.7 Front-Head $\Sigma\Delta$ Modulator Fractional Synthesis Output For Topology Shown In Figure C.5

Analyzing both figures, there is evidence that there are more harmonics in the topology using the delta sigma clocked at half the clock period than the proposed topology. Furthermore, since the system reference is at higher frequency (in this case 93.62127Mhz), the output spectrum of the quantization noise is oversampled, hence more noise enters the PLL system compare to the other topologies. In addition, because of the half clock period, harmonics appear near by the the two encoded tones, this further adds phase noise. This

phenomena is pictured below, where method 2 defines the new proposed method.

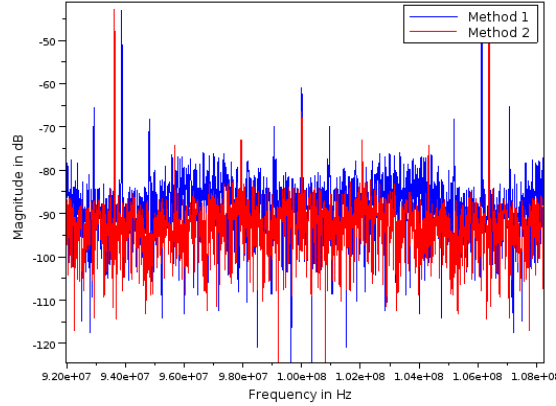


Fig. C.8 Front-Head $\Sigma\Delta$ Modulator Fractional Synthesis Outputs Near Main Signals For Both Topologies

In pursuance of testing the theory, a verilog-A simulation along with Scilab scripts were used to concur the theory and simulation matches. For this example, the same PLL characteristics as previously described were used, but, now the input selected frequency was picked between 100 MHz and 90 MHz for the single bit $\Delta\Sigma$ and between 11.111 MHz to 83.3333 MHz for the SS MF-II topology. Figures C.9 and C.11 illustrate the theoretical as well as the simulation results for the fractional synthesis PLL using the front head delta sigma modulator topology from figure C.5. As seen, these results perfectly match the results obtained from the modified feedback fractional synthesis linear model. The results prove that: 1. the modified frequency model for front head fractional synthesis works, and 2. both the front head and feedback methods are equivalent in terms of phase noise injection and filtering action. Furthermore, it is important to notice that the amount of quantization noise added is still depended on the feedback divider. The Δ_{rms} of the input power noise in equation A.1 is still computed the same way, but the division factor in this case is found by taking the ratio of the highest to lowest input selected reference frequency minus one, and multiplying it by the divider ratio. Hence, bigger the step difference in the references, more noise injected into the PLL.

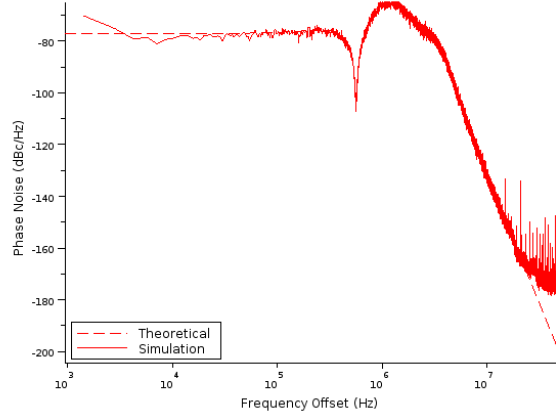


Fig. C.9 Proposed Front-Head Fractional Synthesis (fig. C.5) With Synthesized 1-bit $\Sigma\Delta$ Modulator Phase Noise Output

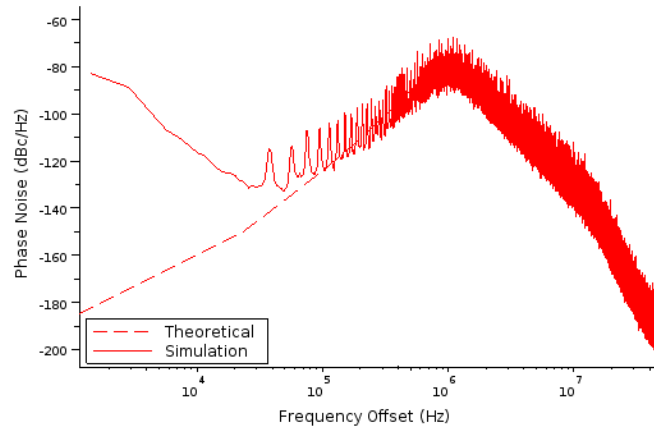


Fig. C.10 Proposed Front-Head Fractional Synthesis (fig. C.5) With SSF-II $\Sigma\Delta$ Modulator Phase Noise Output

In carrying out the proof that the proposed front-head deltas sigma modulator outperforms the topology in figure C.3, a simulation is carried away for the structure in figure C.3. A synthesized delta sigma modulator is used, and the frequency selection is set to 111.111MHz and 100MHz. The results are reported in figure C.11. In this figure, the theoretical lie corresponds to the phase noise injected by the proposed front head method, while the simulation represents the output phase noise of the former method. As seen in

the figure, the proposed method injects 40dB less phase noise into the system. This is because the spectrum in the former method, is actually sampled twice in the PLL and aliasing occurs, hence the output phase noise of such a topology is much higher than the proposed one in figure C.5.

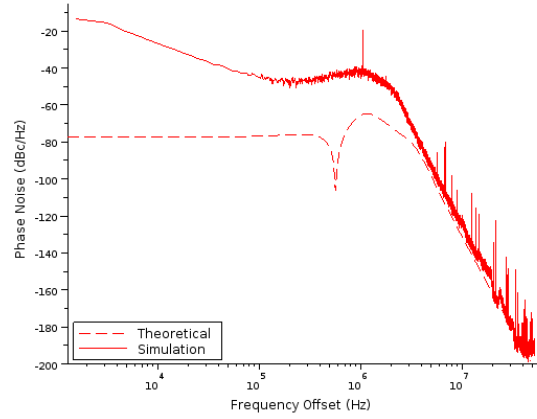


Fig. C.11 Front-Head Fractional Synthesis (fig. C.3) With SSMF-II $\Sigma\Delta$ Modulator Phase Noise Output

Compared with the previous simulations results for the feedback topology, the resolution errors for the front-head fractional synthesis in Verilog-A simulations are less to some extent. The reason of this higher resolution is the possibility of using a lower frequency for the integration function previously mentioned. Therefore, more precision is left for the fractional representation of the output signal. The next section proposes an optimization strategy for the filter implementation of high order synthesized deltas sigma modulators.

References

- [1] M.-S. Hwang, J. Kim, and D.-K. Jeong, "Reduction of pump current mismatch in charge-pump pll," *Electronics Letters*, vol. 45, pp. 135 –136, 29 2009. [ix](#), [2](#), [70](#), [71](#)
- [2] S.-H. Lee and H. J. Park, "A cmos high-speed wide-range programmable counter," *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on*, vol. 49, pp. 638 – 642, sep 2002. [xi](#), [127](#), [128](#)
- [3] D. Ham and A. Hajimiri, "Concepts and methods in optimization of integrated lc vcoss," *Solid-State Circuits, IEEE Journal of*, vol. 36, pp. 896 –909, jun 2001. [2](#), [27](#), [30](#), [31](#), [33](#), [34](#)
- [4] A. Hajimiri and T. Lee, "Design issues in cmos differential lc oscillators," *Solid-State Circuits, IEEE Journal of*, vol. 34, pp. 717 –724, may 1999. [2](#), [30](#), [33](#)
- [5] J. Rogers, C. Plett, and F. Dai, *Integrated Circuit Design For High-Speed Frequency Synthesis*. Artech House, first ed., 2006. [2](#), [3](#), [62](#), [122](#), [123](#), [149](#), [150](#), [159](#)
- [6] C. Zhang and M. Syrzycki, "Modifications of a dynamic-logic phase frequency detector for extended detection range," in *Circuits and Systems (MWSCAS), 2010 53rd IEEE International Midwest Symposium on*, pp. 105 –108, aug. 2010. [2](#), [58](#)
- [7] J. S. Lee, W. K. Jin, D. M. Choi, G. S. Lee, and S. Kim, "A wide range pll for 64x speed cd-roms amp; 10x speed dvd-roms," *Consumer Electronics, IEEE Transactions on*, vol. 46, pp. 487 –493, aug 2000. [2](#), [64](#)
- [8] A. G. P. G. I. Bourdopoulos and T. L. Deliyannis, "Optimal ntfs for single-bit $\sigma\delta$ modulators," *Digital Signal Processing*, vol. 2, July 2002. [3](#), [154](#), [155](#)
- [9] X. Mao, H. Yang, and H. Wang, "Behavioral modeling and simulation of jitter and phase noise in fractional-n pll frequency synthesizer," in *Behavioral Modeling and Simulation Conference, 2004. BMAS 2004. Proceedings of the 2004 IEEE International*, pp. 25 – 30, oct. 2004. [3](#), [113](#), [117](#)

- [10] A. V. Bourdopoulos George, Pnevmatikakis Aristodemos and D. T. L., *Delta-Sigma Modulators Modeling, Design and Applications*. Imperial College Press, first ed., 2003. [3](#), [153](#), [155](#), [156](#), [159](#)
- [11] S. R. Norsworthy Steven R. and T. G. C., *Delta-Sigma Data Converters Theory, Design and Simulation*. IEEE Press, first ed., 1997. [3](#), [106](#), [153](#), [155](#), [159](#)
- [12] M. Perrott, M. Trott, and C. Sodini, "A modeling approach for sigma;- delta; fractional-n frequency synthesizers allowing straightforward noise analysis," *Solid-State Circuits, IEEE Journal of*, vol. 37, pp. 1028 – 1038, aug 2002. [3](#), [117](#), [118](#)
- [13] K. Kundert, "Modelling Jitter in PLL-based Frequency Synthesizers," tech. rep., The Designer's Guide Community, August 2006. Available from www.designers-guide.org. [3](#), [49](#), [50](#), [81](#), [85](#)
- [14] D. Abramovitch, "Phase-locked loops: a control centric tutorial," in *American Control Conference, 2002. Proceedings of the 2002*, vol. 1, pp. 1 – 15 vol.1, 2002. [6](#), [7](#), [9](#)
- [15] B. Razavi, "Monolithic phase-locked loops and clock recovery circuits: Theory and designs," [6](#), [7](#), [9](#)
- [16] C. Lau and M. Perrott, "Fractional-n frequency synthesizer design at the transfer function level using a direct closed loop realization algorithm," in *Design Automation Conference, 2003. Proceedings*, pp. 526 – 531, june 2003. [7](#), [9](#), [11](#), [12](#)
- [17] S. Aouini, K. Chuai, and G. Roberts, "Anti-imaging time-mode filter design using a pll structure with transfer function dft," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 59, pp. 66 – 79, jan. 2012. [11](#)
- [18] A. Demir, "Computing timing jitter from phase noise spectra for oscillators and phase-locked loops with white and noise," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 53, pp. 1869 – 1884, sept. 2006. [18](#)
- [19] G. K. Hikmet Sari, Herv Houeix, "Phase and frequency detectors for clock synchronization in high-speed optical transmission systems," *European Transactions on Telecommunications*, vol. 5, pp. 635–642, September 1994. [19](#)
- [20] K.-H. Cheng, W.-B. Yang, and C.-M. Ying, "A dual-slope phase frequency detector and charge pump architecture to achieve fast locking of phase-locked loop," *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on*, vol. 50, pp. 892 – 896, nov. 2003. [19](#)
- [21] N. Rashidi, A. Nabavi, and M. Mehrabian, "7ghz voltage control ring oscillator for uwb applications," in *Microelectronics, 2008. ICM 2008. International Conference on*, pp. 429 – 432, dec. 2008. [22](#)

- [22] A. Hajimiri and T. Lee, "A general theory of phase noise in electrical oscillators," *Solid-State Circuits, IEEE Journal of*, vol. 33, pp. 179–194, feb 1998. [23](#), [25](#), [26](#), [30](#)
- [23] N. Seller, E. Chataigner, H. Lapuyade, and J.-B. Begueret, "A distributed voltage controlled oscillator in a standard 65nm cmos process," in *Research in Microelectronics and Electronics 2006, Ph. D.*, pp. 93–96, 0-0 2006. [23](#)
- [24] A. Hajimiri, "Distributed integrated circuits: an alternative approach to high-frequency design," *Communications Magazine, IEEE*, vol. 40, pp. 168–173, feb 2002. [23](#)
- [25] D. Leeson, "A simple model of feedback oscillator noise spectrum," *Proceedings of the IEEE*, vol. 54, pp. 329–330, feb. 1966. [24](#)
- [26] R. J. Baker, *Circuit Design, Layout and Simulation*. IEEE Series on Microelectronics Systems, third ed., 2010. [32](#)
- [27] A. Hastings, *The Art of Analog Layout*. Prentice Hall, first ed., 2000. [37](#)
- [28] S. Vamvakos, V. Stojanovic, and B. Nikolic, "Discrete-time, linear periodically time-variant phase-locked loop model for jitter analysis," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 58, pp. 1211–1224, june 2011. [53](#)
- [29] C. Charles and D. Allstot, "A calibrated phase/frequency detector for reference spur reduction in charge-pump plls," *Circuits and Systems II: Express Briefs, IEEE Transactions on*, vol. 53, pp. 822–826, sept. 2006. [56](#)
- [30] X. Haurie and G. Roberts, "A multiplier-free structure for 1-bit high-order digital delta-sigma modulators," in *Circuits and Systems, 1995., Proceedings., Proceedings of the 38th Midwest Symposium on*, vol. 2, pp. 889–892 vol.2, aug 1995. [108](#)
- [31] K. S. Kundert and Z. Olaf, *The Designer's Guide to Verilog-AMS*. Kluwer Academic Publishers, first ed., 2004. [115](#)
- [32] S. Yan, Y. Chen, T. Wang, and H. Wang, "A 40-gb/s quarter rate cdr with 1 x2236;4 demultiplexer in 90-nm cmos technology," in *Communication Technology (ICCT), 2010 12th IEEE International Conference on*, pp. 673–676, nov. 2010. [123](#)
- [33] H. Zhang, H.-G. Yang, J. Zhang, and F. Liu, "High-speed programmable counter design for pll based on a delay partition technique," in *Radio-Frequency Integration Technology, 2009. RFIT 2009. IEEE International Symposium on*, pp. 100–103, 9 2009-dec. 11 2009. [128](#)
- [34] J. M. Rabaey, A. Chandrakasan, and B. Nikolic, *Digital Integrated Circuits: A Design Perspective*. Prentice Hall Electronics and VLSI Series, second ed., 2003. [128](#)

-
- [35] X. Haurie and G. Roberts, “A multiplier-free structure for 1-bit high-order digital delta-sigma modulators,” *Proceedings of the IEEE Midwest Symposium on Circuits and Systems*, pp. 889–892, August 1995. [155](#), [161](#)
 - [36] O. Sarbishei, K. Radecka, and Z. Zilic, “Analytical optimization of bit-widths in fixed-point lti systems,” *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 31, pp. 343–355, march 2012. [161](#)
 - [37] S. Aouini and G. Roberts, “Frequency synthesis using digital-to-frequency conversion and filtering,” *Electronic Letters*, vol. 46, pp. 979–980, July 2010. [163](#)