2016

A Wide Band Adaptive All Digital Phase Locked Loop With Self Jitter Measurement And Calibration

Bo Jiang
University of Vermont

A WIDE BAND ADAPTIVE ALL DIGITAL PHASE LOCKED LOOP WITH SELF JITTER MEASUREMENT AND CALIBRATION

A Dissertation Presented

by

Bo Jiang

to

The Faculty of the Graduate College

of

The University of Vermont

In Partial Fullfillment of the Requirements
for the Degree of Doctor of Philosophy
Specializing in Electrical Engineering

May, 2016

Defense Date: Feberary 18, 2016
Dissertation Examination Committee:

Tian Xia, Ph.D., Advisor
Joseph E. Brayden, Ph.D., Chairperson
Stephen Titcomb. Ph.D.
Walter J. Varhue, Ph.D.
Cynthia J. Forehand, Ph.D., Dean of the Graduate College

Abstract

The expanding growth of mobile products and services has led to various wireless communication standards that employ different spectrum bands and protocols to provide data, voice or video communication services. Software defined radio and cognitive radio are emerging techniques that can dynamically integrate various standards to provide seamless global coverage, including global roaming across geographical regions, and interfacing with different wireless networks. In software defined radio and cognitive radio, one of the most critical RF blocks that need to exhibit frequency agility is the phase lock loop (PLL) frequency synthesizer. In order to access various standards, the frequency synthesizer needs to have wide frequency tuning range, fast tuning speed, and low phase noise and frequency spur. The traditional analog charge pump frequency synthesizer circuit design is becoming difficult due to the continuous down-scalings of transistor feature size and power supply voltage. The goal of this project was to develop an all digital phase locked loop (ADPLL) as the alternative solution technique in RF transceivers by taking advantage of digital circuitry's characteristic features of good scalability, robustness against process variation and high noise margin. The targeted frequency bands for our ADPLL design included 880MHz-960MHz, 1.92GHz-2.17GHz, 2.3GHz-2.7GHz, 3.3GHz-3.8GHz and 5.15GHz-5.85GHz that are used by wireless communication standards such as GSM, UMTS, bluetooth, WiMAX and Wi-Fi etc.

This project started with the system level model development for characterizing ADPLL phase noise, fractional spur and locking speed. Then an on-chip jitter detector and parameter adapter was designed for ADPLL to perform self-tuning and self-calibration to accomplish high frequency purity and fast frequency locking in each frequency band. A novel wide band DCO is presented for multi-band wireless application. The proposed wide band adaptive ADPLL was implemented in the IBM 0.13μm CMOS technology. The phase noise performance, the frequency locking speed as well as the tuning range of the digitally controlled oscillator was assessed and agrees well with the theoretical analysis.

Acknowledgements

It has been an honor to study at the University of Vermont. There are many people to thank. They have provided inspiration guidance and friendship. This dissertation would not have been completed without the help of many people.

My foremost appreciation must go to my advisor, Dr. Tian Xia who pointed me in the right direction at the beginning of my research. His extensive vision and creative thinking have provided the source of inspiration for me all through my graduate study. His academic research experiences have encouraged me for continuing exploring the integrated circuits in the future. And his invaluable assistance in conducting research and writing this dissertation is greatly appreciated.

I am especially grateful to the members of committee, Dr. Varhu, Dr. Titcomb, and Dr. Brayden for their valuable suggestions and numerous help. Through their instruction, I learned how to pursue research.

It has been a great pleasure to work in the Electrical Engineering of the University of Vermont. I would like to acknowledge all of my colleagues and friends for their suggestions and help.

My sincere appreciation goes to my parents who have always given me unconditional love, guidance, and support. Without them, none of these could have accomplished. I would like to thank my sister for being so supportive and encouraging throughout years of my study.

Dedication

Dedicate to my parents

Table of Contents

Acknowledgements . . . ii
Dedication . . . iii
List of Figures . . . x

1 Introduction . . . 1
1.1 PLL Fundamental . . . 1
1.2 Motivation . . . 5

2 Multi-band ADPLL Design . . . 11
2.1 Introduction . . . 11
2.2 Digitally Controlled Oscillator Design . . . 13
2.2.1 Quad-mode DCO Design Review . . . 16
2.2.2 Proposed Multi-mode DCO Design . . . 27
2.2.3 Experimental Results . . . 38
2.3 Time to Digital Converter based Phase Frequency Detector Design . . . 46
2.3.1 Basic delay line based TDC . . . 52
2.3.2 Vernier TDC . . . 53
2.3.3 Proposed Time to Digital Converter based Phase Frequency Detector Design . . . 55
2.4 Digital Loop Filter . . . 58
2.4.1 FIR Filter . . . 58
2.4.2 IIR Filter . . . 59
2.4.3 Proposed Digital Loop Filter . . . 60
2.5 Fractional Divider with Delta Sigma Modulator . . . 63
2.5.1 Delta Sigma Modulator . . . 64
2.5.2 Proposed Frequency divider with Delta Sigma Modulator . . . 67
2.6 The Proposed Multi-band ADPLL Simulation Results . . . 71
2.6.1 Frequency Response . . . 72
2.6.2 Phase Noise Performance . . . 74
2.6.3 Power Consumption . . . 76
2.7 Conclusion . . . 77

3 ADPLL Design Parameters Determinations through Noise Modeling . . . 78
3.1 Definition of Phase Noise . . . 79
3.2 Phase Noise Model . . . 79

3.2.1 Noise Sources in An ADPLL . . . 81
3.2.2 ADPLL Noise Transfer Function . . . 82
3.3 Output Phase Noise of ADPLL . . . 85
3.3.1 Phase Noise due to Input Reference Signal . . . 85
3.3.2 Phase Noise due to DCO . . . 88
3.3.3 Phase Noise due to Delta Sigma Modulator . . . 88
3.3.4 Phase Noise due to TDC-PFD . . . 89
3.4 ADPLL Fractional Spur . . . 90
3.5 ADPLL Design with Adjustable Loop Parameters . . . 93
3.6 Experimental Results . . . 100
3.6.1 Architecture-I ADPLL . . . 100
3.6.2 Architecture-II ADPLL . . . 102
3.7 Conclusions . . . 105

4 ADPLL On Chip Jitter Measurement . . . 106
4.1 Relationship among Jitter, Phase Noise and DCO Control Code . . . 109
4.1.1 Relationship between Period Jitter and Phase Noise . . . 109
4.1.2 Relationship between ADPLL Jitter and DCO Control Code . . . 114
4.2 On Chip Jitter Measurement Circuit Design . . . 116
4.2.1 Behavior Model . . . 116
4.2.2 Matlab Simulation . . . 117
4.2.3 Circuit Topology . . . 121
4.2.4 Experimental Results . . . 126
4.3 Conclusion . . . 127

5 Wide band ADPLL with Self Jitter Measurement and Calibration Design . . . 128
5.1 Digital Loop Filter with Self Jitter Calibration Design . . . 129
5.1.1 Coefficients adjustable PPI filter . . . 129
5.1.2 Circuit topology . . . 132
5.2 Wide band ADPLL with Self Jitter Measurement and Calibration Circuit Topology . . . 134
5.2.1 Phase noise and jitter performance . . . 138
5.2.2 Power consumption . . . 141
5.2.3 Measurement of ADPLL . . . 143
5.3 Conclusions . . . 143

6 Conclusion . . . 145

A Matlab Codes 147
A.1 Matlab Codes for Case Studies in Chapter 3 147
A.2 Matlab Code for Phase Noise Calculation 156
A.3 Matlab Code for Phase Noise to Jitter Conversion 157
A.4 Matlab Code for Jitter Calculation from DCO Control Code 158

List of Figures

1.1 PLL applications in wireless transceiver. (a) A simplified direct-conversion transmitter; (b) A simplified direct-conversion receiver. 2
1.2 The block diagram of charge pump PLL. 3
1.3 The block diagram of ADPLL. 4
1.4 Frequency range performances of recent wide band VCOs/DCOs. . . 6
1.5 Phase noise performances of recent VCOs/DCOs. 9
1.6 The block diagram of proposed wide band ADPLL with self jitter calibration. 10

2.1 The second order negative feedback ADPLL structure. 12
2.2 LC tank DCO with varactor array. 14
2.3 DCO models for (a) Quad-mode DCO; (b) Structure-I, (c) Structure-II. 17
2.4 DCO output frequency f_{osc} (a) Mode-I; (b) Mode-II. 19
2.5 DCO output frequency f_{osc} (a) Mode-III; (b) Mode-IV. 21
2.6 Frequency output under different L_2 to L_1 ratio. $L_1 = 2.5nH$; $L_2 = 0.625nH; 1.25nH; 2.5nH$. (a) Mode-I, (b) Mode-II, (c) Mode-III, (d)Mode-IV. 24
2.7 Detailed schematic of quad-mode DCO. 25
2.8 Transient signal of DCO output, (a) 5.6GHz; (b) 2.4GHz. 26
2.9 (a) Output frequency with control code in Mode-I, (b) Phase noise at frequency 5.6GHz and 2.4GHz. 26
2.10 DCO models for (a) Multi-mode DCO; (b) Structure-I, (c) Structure-II, (d) Structure-III 27
2.11 2-bit DCO structure select circuit 32
2.12 Simplified circuits of (a) Inverter (b) 2-input OR gate. 33
2.13 Transistor level circuits of (a) Inverter (b) 2-input OR gate. 34
2.14 Simplified circuits of (a) 5-bit coarse tune (b) 6-bit coarse tune (c) 6-bit fine tune. 35
2.15 Transistor level circuits of (a) 5-bit coarse tune (b) 6-bit coarse tune (c) 6-bit fine tune. 35
2.16 Transistor level circuits of the 3-bit fine tune. 36
2.17 3:1 multiplexer for DCO output selection. 37
2.18 Detailed schematic of proposed DCO. 38
2.19 Detailed transistor level circuit of proposed DCO in Cadence. 40
2.20 Transient signal and phase noise of DCO output at 5.6GHz. 41
2.21 Phase noise at 1MHz offset of DCO output at 5-6GHz. 42
2.22 Transient signal and phase noise of DCO output at 3.6GHz. 43

2.23 Phase noise at 1MHz offset of DCO output at 3.2-4GHz. 43
2.24 Transient signal and phase noise of DCO output at 2.4GHz. 44
2.25 Phase noise at 1MHz offset of DCO output at 1.9-2.8GHz. 44
2.26 Transient signal and phase noise of DCO output at 925MHz. 45
2.27 Phase noise at 1MHz offset of DCO output at 0.8-1.1GHz. 45
2.28 PFD with 2-D flip flops. 46
2.29 PFD dead zone. 47
2.30 Gate level phase frequency detector. 48
2.31 The simplified circuits of NAND gate (a) 2-input (b) 3-input, (c) 4-input. 49
2.32 The transistor level circuits of NAND gate (a) 2-input (b) 3-input, (c) 4-input. 49
2.33 PFD simulation results when phase of reference and feedback signal is same. 50
2.34 PFD simulation results when reference leads feedback signal. 50
2.35 PFD simulation results when reference lags feedback signal. 51
2.36 Implementation of a basic delay-line based TDC. 52
2.37 Implementation of 3 bit inverter based TDC. 52
2.38 The linearity of 3 bit inverter based TDC. 53
2.39 Implementation of a vernier TDC. 54
2.40 Implementation of a 3 bit vernier TDC. 54
2.41 The linearity of 3 bit vernier delay chain TDC. 55
2.42 Implementation of a 6 bit vernier TDC. 56
2.43 The detailed TDC schematic of vernier TDC and fat tree encoder. . . 57
2.44 The linearity of 6 bit vernier TDC. 57
2.45 The basic architecture of FIR filter. 58
2.46 The basic architecture of IIR filter. 60
2.47 The basic architecture of PPI filter. 61
2.48 The basic architecture of proportional path. 62
2.49 The basic architecture of integral path. 62
2.50 The behavior model of proposed digital loop filter. 63
2.51 Half adder. 63
2.52 Full adder. 64
2.53 Details of the second order MASH $\Delta\Sigma$ modulator. 66
2.54 High frequency divide by 2. 66
2.55 DCO output signal divider by 2. 67
2.56 Behavior model of proposed programmable divider. 68
2.57 D flip flop implement. 69
2.58 Frequency divider. 70
2.59 8-bit counter. 70
2.60 Proposed wide band ADPLL. 71

2.61 Frequency response of (a) 1GHz; (b) 2.4GHz; (c) 3.6GHz; (d) 5.6GHz. 73
2.62 ADPLL output phase noise at 920MHz (a) Projected; (b) simulated. . 74
2.63 ADPLL output phase noise at 2.4GHz (a) Projected; (b) simulated. . 75
2.64 ADPLL output phase noise at 3.6GHz (a) Projected; (b) simulated. . 76
2.65 ADPLL output phase noise at 5.6GHz (a) Projected; (b) simulated. . 76

3.1 Phase noise definition. 80
3.2 The s-domain phase noise model of phase-domain ADPLL [35]. 81
3.3 The z-domain phase noise model of phase-domain ADPLL. (a) Architecture-I ADPLL; (b) Architecture-II ADPLL. 82
3.4 Prototype of oscillator phase noise spectrum density. 86
3.5 Phase noise of 13MHz crystal oscillator. (a)Measured, (b)Modeled. . 87
3.6 Traditional N-bit TDC in ADPLL. 89
3.7 Fractional spur of ADPLL in [46]. 92
3.8 ADPLL locking time with different DLF coefficients. 97
3.9 DLF coefficients determination. 98
3.10 Output phase noise with different DLF coefficients. 100
3.11 Simulated phase noise of each noise source and total phase noise at 1.5GHz output frequency. 102
3.12 Simulated phase noise of each noise source and total phase noise at 5.376021831GHz output frequency. 104

4.1 Period jitter.[15] 107
4.2 Dead zone method [15]. 108
4.3 Variance metric jitter estimation [15]. 109
4.4 Traditional phase noise measurement [54]. 111
4.5 Behavior model of ADPLL on chip jitter measurement block. 113
4.6 Behavior model of ADPLL on chip jitter measurement block. 116
4.7 Detail model of ADPLL on chip jitter measurement block. 117
4.8 ADPLL simulation model in MATLAB. 118
4.9 PFD model in MATLAB. 118
4.10 TDC model in MATLAB. 119
4.11 LPF model in MATLAB. 119
4.12 Divider with delta sigma modulato model in MATLAB. 120
4.13 Schematic of the proposed on-chip jitter measurement. 121
4.14 Schematic of mean value calculation block. 122
4.15 8-bit digital comparator. 123
4.16 Histogram of 256 DCO control code when output frequency is 920MHz. 124
4.17 Histogram of 256 DCO control code when output frequency is 2.4GHz. 124
4.18 Histogram of 256 DCO control code when output frequency is 3.6GHz. 125

4.19 Histogram of 256 DCO control code when output frequency is 5.65GHz. 125
4.20 Jitter measurement simulation at 1GHz. 126

5.1 The architecture of adjustable proportional path. 130
5.2 The architecture of adjustable integral path. 131
5.3 Top level model of adjustable low pass filter. 131
5.4 The adjustable low pass filter. 132
5.5 The adjustable DLF coefficients. 133
5.6 Wide band ADPLL with on chip jitter measurement and self jitter calibration. 134
5.7 Wide band adaptive ADPLL with external noise model. 135
5.8 Histogram of Code<7:5> with noise in 920MHz (a) without jitter calibration (b) with jitter calibration. 136
5.9 Histogram of Code<7:5> with noise in 2.4GHz (a) without jitter calibration (b) with jitter calibration. 136
5.10 Histogram of Code<7:5> with noise in 3.6GHz (a) without jitter calibration (b) with jitter calibration. 137
5.11 Histogram of Code<7:5> with noise in 5.6GHz (a) without jitter calibration (b) with jitter calibration. 137
5.12 Phase noise performance at 920MHz output (a) without calibration block; (b) with calibration block. 138
5.13 Phase noise performance at 2.4GHz output (a) without calibration block; (b) with calibration block. 138
5.14 Phase noise performance at 3.6GHz output (a) without calibration block; (b) with calibration block. 139
5.15 Phase noise performance at 5.6GHz output (a) without calibration block; (b) with calibration block. 139
5.16 The minimum current of the ADPLL. 142
5.17 The maximum current of the ADPLL. 142

Chapter 1

Introduction

1.1 PLL Fundamental

A Phase Locked Loop (PLL) is one of the most important blocks in wireless communication system. It is widely used for clock and data recovery or frequency synthesizer. The PLL based frequency synthesizer is deployed as local oscillator (LO) to perform frequency translation between baseband (BB) and radio frequency (RF) in wireless transceivers as shown in Fig. 1.1.

In a direct-conversion transmitter, the digital I/Q signals are converted into analog signals via the digital-to-analog converter (DAC). Then the low pass filter (LPF) filters out the DAC noise caused by DAC in frequency domain. After that, the baseband analog signals are up-converted to RF frequency by a single-sideband modulator. A power amplifier (PA) which is connected to an antenna is the last stage in the transmitter path to provide enough output power. In a direct-conversion receiver, the signal received from the antenna goes through a low noise amplifier (LNA) first. Then it is down-converted to baseband signal via a quadrature mixer. The following

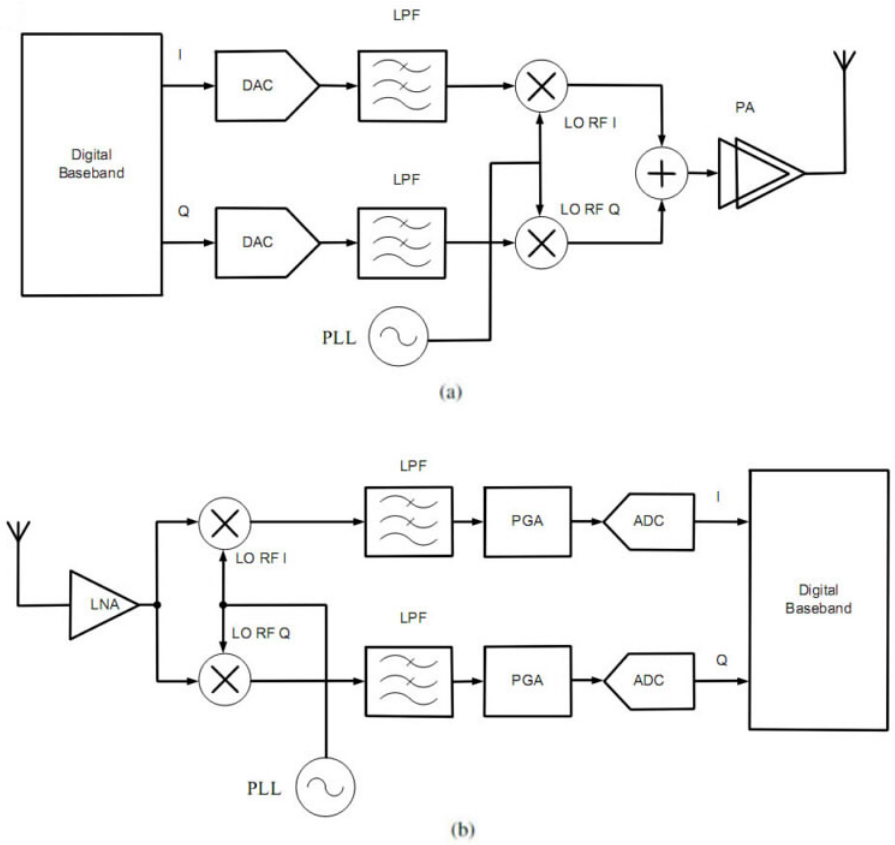


Figure 1.1: PLL applications in wireless transceiver. (a) A simplified direct-conversion transmitter; (b) A simplified direct-conversion receiver.

LPF filters frequency noise and the programmable gain amplifier (PGA) can alternate signals to the required level for the analog-to-digital converter (ADC). At last, the I/Q signals are fed into digital signal processing block.

Fig. 1.2 illustrates a basic block diagram of charge pump PLL (CPPLL). A traditional analog charge pump PLL typically consists of five main blocks, which are phase frequency detector (PFD), charge pump (CP), loop filter (LF), voltage controlled oscillator (VCO) and frequency divider (DIV). PFD block compares the phase difference between the input reference clock (REF) and the feedback signal (FB) and

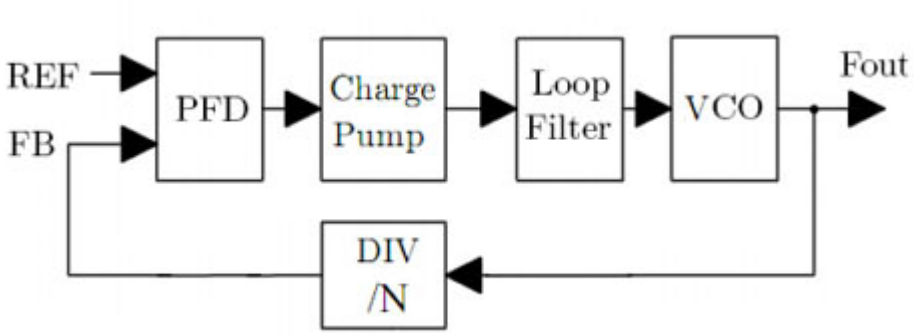


Figure 1.2: The block diagram of charge pump PLL.

generates up and down signals according to phase difference as its outputs. Charge pump converts up and down signals to current by two switches. After filtering out the low frequency noise through the loop filter, the VCO generates the output signal with the targeted frequency. The frequency divider divides the VCO output frequency by a programmable number N and generates the feedback signal. It is clear that, PLL operates as a negative feedback control system. In the locking condition, the relationship between PLL output signal frequency and the reference signal frequency is:

$$f_{out} = N \cdot f_{ref} \tag{1.1}$$

In modern wireless applications, a fine frequency resolution is desired in PLL design. In an integer PLL, the divider value N is an integer number, which means the PLL frequency resolution equals f_{ref}. Also, the bandwidth of PLL is usually no more than one tenth of f_{ref} for stability consideration. In order to get a better fine frequency resolution in the integer PLL, it requires a reference signal with a lower frequency. The lower reference frequency results in a longer settling time. Besides, a very narrow bandwidth will result in a loop filter that takes large chip area.

Compared to the integer PLL, the fractional-N PLL can improve fine frequency resolution without reducing the PLL bandwidth. The effective bits of fractional part of divider value N determines the PLL fine frequency resolution. In a fractional-N PLL, a multi-modulus divider (MMD) with delta sigma modulator (DSM) is employed. The DSM generates the command word for MMD to produce a fractional division ratio. The divider value is changed between different integer values and the average results in a fractional value.

All digital PLL (ADPLL) frequency synthesizers are recently emerging to replace CPPLL in many RF transceivers and computer chips on account of the superior features that digital circuits can provide, such as robustness, scalability, small area and power dissipation etc. As shown in Fig. 1.3, an ADPLL typically consists of four main blocks, which are phase frequency detector (PFD), digital loop filter (DLF), digitally controlled oscillator (DCO), frequency divider (DIV).

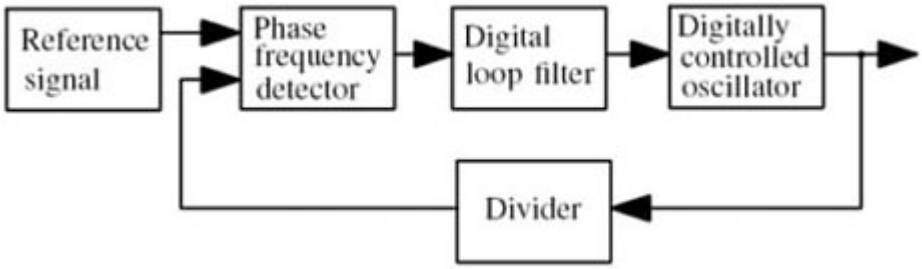


Figure 1.3: The block diagram of ADPLL.

Comparing with the analog charge pump PLLs, each block in ADPLLs is digital. Time to digital converter (TDC) based digital phase frequency detector replaces the conventional phase frequency detector. The TDC is applied to quantize the phase-frequency difference between the feedback clock and the input reference clock. The charge pump and analog loop filter are replaced by a digital loop filter. The VCO is

replaced by a DCO, whose output frequency is controlled by a digital control code. The digital frequency divider is used to control the generation and tuning of PLL output frequency.

In wireless communication, frequency tuning range, phase noise, jitter and locking time are key characteristics of the ADPLL frequency synthesizer. Frequency tuning range determines the band applicability. ADPLL frequency synthesizer's phase noise and jitter cause significant degradation in wireless communication systems performance. ADPLL with low quality phase noise performance will reduce the effective signal to noise ratio, limit bit error and data rate. Therefore, it is important to investigate the relationship between ADPLL variables and characteristics.

1.2 Motivation

With the expanding growth of mobile products and services, the wireless communication standards employ different spectrum bands and protocols to provide data, voice or video communication services. Smart phone is just a particular example that relies on PLLs to modulate or demodulate multi wireless communication standards such as GSM, GMTS, bluetooth, Wi-Fi and WiMAX. In order to save chip area, power consumption and cost, it is necessary for a single PLL to modulate or demodulate wireless signals with different wireless standards. In order to access various standards such as GSM, GMTS, WiMAX, bluetooth and Wi-Fi, the PLL frequency synthesizer needs to have wide frequency tuning range, fast tuning speed, and low phase noise and frequency spur. Recently, software defined radio and cognitive radio are emerging techniques that can dynamically integrate various standards to provide seamless

global coverage, including global roaming across geographical regions, and interfacing with different wireless networks.

One of the most critical components in wide band ADPLL is digitally controlled oscillator (DCO). The DCO must cover wide frequency range and have low phase noise performance. There are two types of DCO in ADPLL: LC-tank DCO and ring oscillator DCO. In wireless communication applications, LC-tank DCO is widely adopted in ADPLL frequency synthesizer. Comparing to the ring oscillator, LC-tank DCO has superior phase noise performance. However, LC resonator typically has limited frequency tuning range due to limited varactor tuning capability. In order to increase PLL frequency tuning range, there are some researches focusing on wide band VCOs/DCOs development [1-5]. Fig. 1.4 summarizes recent wide band VCOs/DCOs operating frequency range performances.

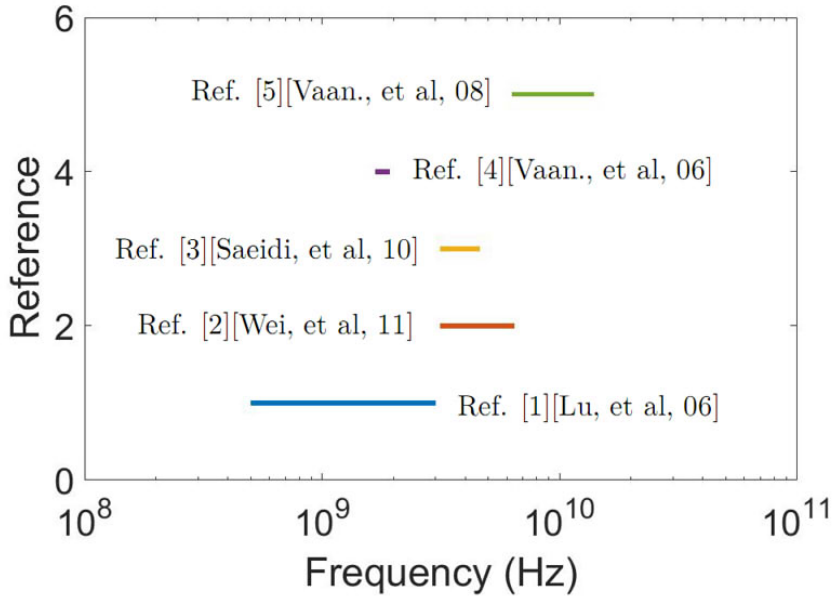


Figure 1.4: Frequency range performances of recent wide band VCOs/DCOs.

Also, there are many researches focusing on multi-band PLL frequency synthesizer [6-11] recently. Table 1.1 summarizes those PLLs' frequency tuning ranges, phase noise, jitter, power consumptions and etc. In reference [6], the core VCO is operating in frequency of 3.2GHz-4GHz. The VCO output is divided by 2 to generate the 1.6GHz-2GHz signal and mixed up to form the 4.8GHz-6GHz signal. The 2.4GHz-5GHz and 0.8GHz-1GHz signals are generated by adding divide-by-two circuits after the 4.8GHz-6GHz and 1.6GHz-2GHz signals. Extra power is consumed since multiplier and divider are applied in this method. In reference [7], the frequency tuning range covers 0.38GHz-6GHz and 9GHz-12GHz. A divide-by-two circuit is designed to accomplish the frequency band of 0.3GHz-13.7GHz. Reference [8-11] covers multi-band frequency range whose central frequency is closed to each other.

Table 1.1: Recent multi-band PLLs characteristics.

Ref.	Freq. Range GHz	Band	Phase Noise @1MHz freq. offset	Power mW	Tech.
[6]	0.8-6	802.11abg/PCS/ DCS/Cellular band	-110dBc/Hz @3.24GHz	43.2	0.18μm
[7]	0.3-13.7		-114.6dBc/Hz @4.85GHz	24	65nm
[8]	0.8-2	GSM/WCDMA	-135dBc/Hz	N/A	0.18μm
[9]	2.39-3.28 4.79-6.55	FMCW Rada system	-103dBc/Hz @2.4GHz	N/A	0.18μm
[10]	9.1-11.6	X band	-102dBc/Hz @9.61GHz	32.5	0.13μm
[11]	1.8-3	802.15.4, BLE 5Mbps proprietary	N/A	20	65nm

This research presents a multi-band LC-tank DCO. The frequency range covers multiple frequencies including 800MHz to 1.1GHz, 1.8GHz to 2.8GHz, 3GHz to 4GHz and 5GHz to 6GHz. The proposed ADPLL can cover GSM, UMTS, WiMAX,

bluetooth and Wi-Fi wireless communication frequency bands. Moveover, as the proposed DCO eliminates the need of switches in the LC-tank, the resistive loss and phase noise effects are greatly alleviated.

Besides frequency tuning range, phase noise and jitter are key parasitics in ADPLL. ADPLL frequency synthesizer's phase noise and jitter cause significant degradation in wireless communication systems performance. ADPLL with low quality phase noise and jitter performance will reduce the effective signal to noise ratio, limit bit error and data rate. Designing a low phase noise, low jitter PLL becomes a challenge for RF circuit designers. Fig. 1.5 summarizes recent wide band VCOs/DCOs phase noise performances. In our previous research [12], ADPLL's phase noise performance can be further improved by adjusting circuit parameters including PFD resolution, loop filter coefficients and DCO gain. Also, there are some researches focus on PLL design with adaptive bandwidth [13-14]. In reference [13], authors have presented an adaptive bandwidth PLL according to the locking status and the phase error amount. In reference [14], authors have proposed an adaptive bandwidth PLL which is relied on the small-signal conductance tracking the large-signal conductance of the VCO. There is no research on adjusting ADPLL parameter according to output phase noise or jitter performance. In order to accomplish adaptive ADPLL, the on chip jitter measurement is an important function block. In reference [15], on chip jitter measurement is based on an all digital frequency discriminator. In reference [16], on chip jitter measurement is analyzed and designed based on the deadzone method. Reference [17] uses a low-noise voltage-controlled delay-line and mixer-based frequency discriminator to extract the phase-noise fluctuations at baseband. Those researches only measure the jitter or phase noise, but not adjust the PLL loop parameters to

improve PLL performance. This research also focuses on both on chip jitter measurement and jitter calibration through adjustable filter coefficients. The research analyzes the relationship between ADPLL circuit parameter and jitter performance. The on chip jitter measurement is presented according to the analysis of the relationship between PLL output signal's jitter and DCO control code. By adjusting the loop filter's coefficients according to the jitter measurement results, the ADPLL output jitter performance can be further improved to meet requirement.

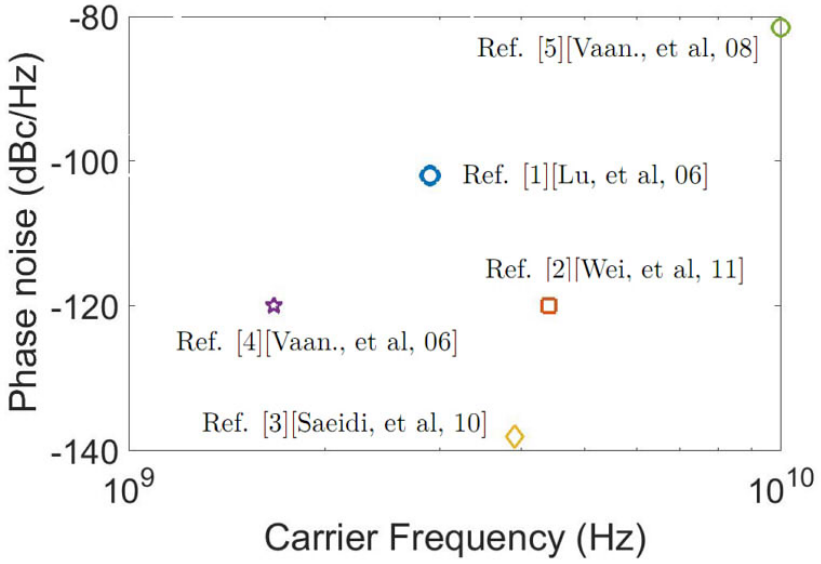


Figure 1.5: Phase noise performances of recent VCOs/DCOs.

Fig. 1.6 shows the model of the proposed wide band ADPLL with self jitter calibration. Phase frequency detector compares the reference signal and feedback signal to generate corresponding up and down signals. Up and down signal are fed into time to digital converter (TDC) to produce digital code. After passive proportional integral (PPI) filter, the digital code is used to control DCO output. On the feedback path, a multi-modulus divider with a second order delta sigma modulator divides the

DCO output signal to produce the feedback signal. Once the ADPLL is locked, on chip jitter measurement function block starts to collect the information of DCO control code and calculate DCO control code variance. The variance of DCO control code can be translated to jitter performance by given output frequency, loop bandwidth, divider value, DCO gain and PFD resolution. Then, the result from jitter measurement block is compared with the preset threshold to determine to turn on or turn off the jitter calibration block. In jitter calibration process, digital low pass filter coefficients are preset to a large value to reduce PLL settling time, and then its value will be adjusted according to the jitter measurement result. Based on the theoretical analysis, the design schemes for measuring and adjusting ADPLL jitter performance are presented and verified by simulation.

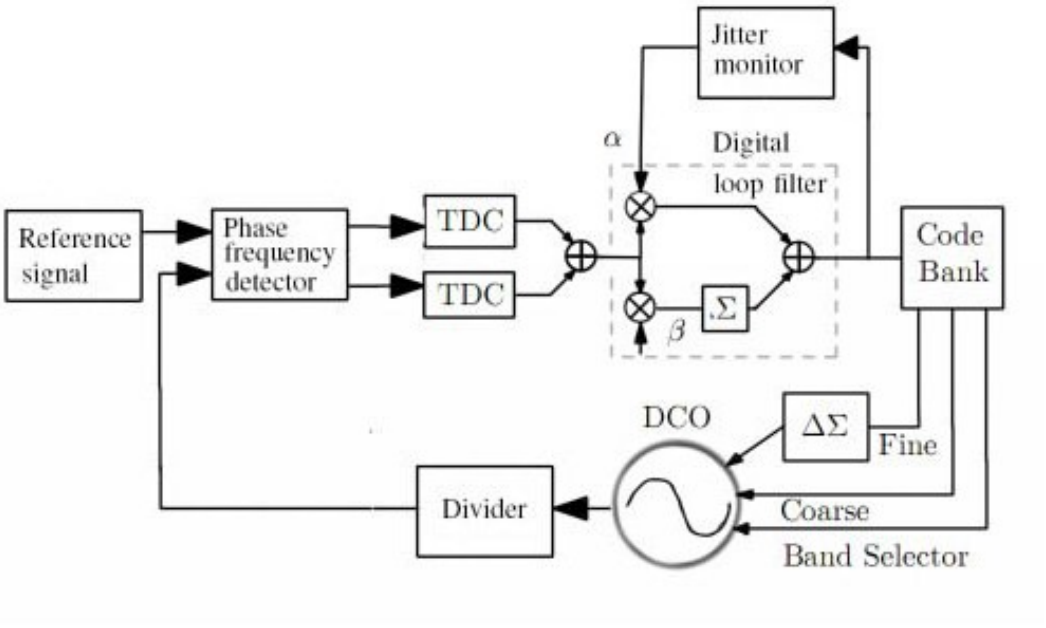


Figure 1.6: The block diagram of proposed wide band ADPLL with self jitter calibration.

Chapter 2

Multi-band ADPLL Design

2.1 Introduction

Nowadays, the ADPLL has become more attractive due to the increasing performance requirement and decreasing cost of VLSI technology. The ADPLL provides a faster locking time and better programmability, testability, stability, and smaller chip size over different processes [69-71]. The ADPLL avoids analog components such as charge pump and analog loop filter and takes the advantage of nanometer-scale CMOS process. Since all signals in the ADPLL are digital, the ADPLL is immune to the digital switching noise in a system-on-chip (SOC) environment. Table 2.1 presents the comparisons between the ADPLL and the traditional charge pump analog PLL.

The ADPLL is comprised of time to digital converter (TDC) based phase frequency detector (PFD), digital loop filter (DLF), digitally controlled oscillator (DCO) and frequency divider (DIV). In the ADPLL design, there are two major issues needed to be carefully considered. One is how to design a digitally-controlled oscillator (DCO) with wide operating range and high resolution. The other one is how to speed up

Table 2.1: Comparison between the performance of ADPLLs and charge pump analog PLL

	ADPLL	Analog PLL
Stability	Good	Poor
Programmability	Good	Poor
Scalability	Good	Poor
Frequency Resolution	Limited	Unlimited
Frequency Tuning	Discrete	Continuous
Frequency Control	Digital Code	Voltage
Locking Range	Limited	Wide
Phase Noise/Jitter	Predictive	Sensitive
Simplicity	Good	Poor
Immune to variations in PVT variations	Good	Poor

the locking process, and reduce the clock jitter and phase noise. As shown in Fig. 2.1, the second order negative feedback system which has a fast locking time and a limited locking range is widely used in the ADPLL.

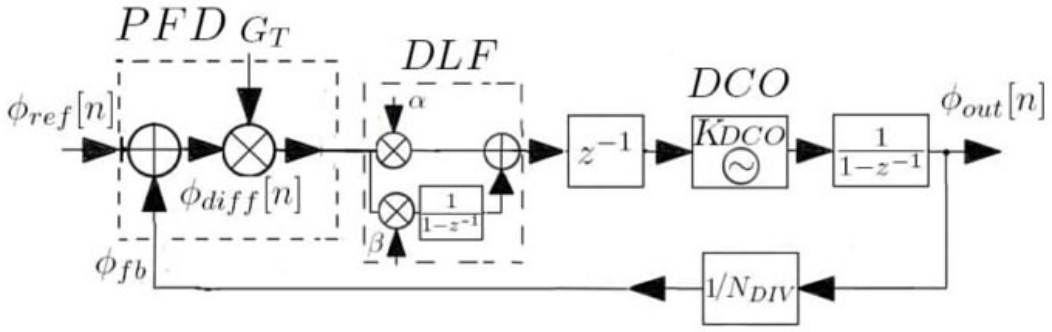


Figure 2.1: The second order negative feedback ADPLL structure.

In this chapter, we present a novel multi-band LC-tank DCO with wide frequency tuning range. Six different resonants are obtained by tuning corresponding varactor values. Such approach increases the frequency tuning range of the LC-tank DCO. The proposed DCO can cover GSM, UMTS, WiMAX, bluetooth and Wi-Fi frequency

bands. Then, we describe architectures and circuit implements of the key blocks such as digitally controlled oscillator (DCO), time-to-digital converter (TDC) based phase frequency detector (PFD), digital loop filter (DLF) and frequency divider with $\Delta\Sigma$ modulator. At last, the proposed ADPLL is implemented in IBM 0.13μm CMOS technology and simulated in Cadence Spectre. Simulation results show the proposed ADPLL meet all requirements including frequency range, frequency resolution and phase noise requirements.

2.2 Digitally Controlled Oscillator Design

Digitally controlled oscillator (DCO) is a key function block in the ADPLL. The function of DCO is to generate an oscillator signal whose frequency is determined by digital control code. It is more flexible and more robust than voltage controlled oscillator (VCO) in the conventional charge pump PLL. Frequency tuning range, frequency resolution, phase noise performance and jitter performance are important characteristics in DCO design. Traditionally, there are two main kinds of DCOs, ring oscillator and LC-tank DCO. Compared to ring oscillator [55-58], LC-tank DCO has superior phase noise performance [72-73]. Therefore, LC-tank DCO is more suitable for wireless communication applications. In order to obtain fine tuning resolution, a main technique in LC-tank DCO design is adopting varactor array to tune capacitance loading as shown in Fig. 2.2 [1-2].

In LC-tank DCO design, the ratio of the maximum output frequency to the min-

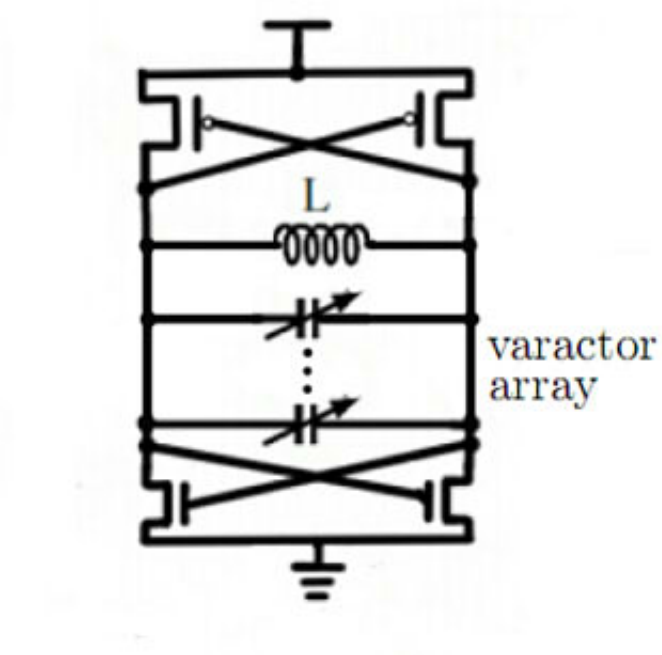


Figure 2.2: LC tank DCO with varactor array.

imum output frequency is:

$$R_f = \frac{f_{max}}{f_{min}} = \frac{\frac{1}{2\pi\sqrt{LC_{min}}}}{\frac{1}{2\pi\sqrt{LC_{max}}}} = \sqrt{\frac{C_{max}}{C_{min}}} \tag{2.1}$$

The frequency tuning range is:

$$TR_f = \frac{f_{max} - f_{min}}{\frac{f_{max}+f_{min}}{2}} \times 100\% = 2\frac{R_f f_{min} - f_{min}}{R_f f_{min} + f_{min}} \times 100\% = 2\frac{R_f - 1}{R_f + 1} \times 100\% \tag{2.2}$$

Practically, the varactor has limited tuning range. For example, the varactor tuning range is around 5 in IBM 8rf technology. The minimum capacitance of varactor and MOSFETs' parasitic capacitances can not be ignored. So, there is a trade off between the maximum output frequency and the operating frequency range. The increasing of the operating frequency range by adding more varactors will lower the maximum

frequency.

The single LC resonator has limited frequency tuning range due to the limited varactor tuning range. The frequency tuning ranges of single LC resonator DCOs in reference [10, 18-22] are less than 30%. Recently, there are some researches focused on increasing the operating frequency range of LC-tank DCO. In reference [1], authors present a tunable active inductor to obtain a wide frequency coverage. However, the active inductor usually doesn't result in low phase noise performance. In reference [2], authors present a dual-mode LC-tank oscillator, where the frequency ranging from 3.14GHz to 6.44GHz is achieved from the core LC-tank oscillator? while the lower frequency band, from 0.25MHz to 3.22GHz, is achieved by a frequency divider chain. Although such design can achieve a wide frequency range, it is not power efficient. In [23], a switched inductor LC oscillator has been proposed to increase frequency tuning capability. However, switch devices employed in the LC-tank degrade the phase noise performance. In addition, even though large size switches can be utilized to reduce their resistive loss, the DCO frequency tuning range will be decreased due to their large parasitic capacitances. In reference [24], authors present a triple-mode oscillator using three coupled inductors. Each inductor has vertical dimensions to save chip area. However, vertical dimensions result in extra resistance loss and the quality factor for inductors is degraded [2].

In this research, we extend our previous research of quad-mode LC-tank DCO and propose a multi-mode LC-tank DCO. The multi-mode LC-tank DCO has five inductors and three varactor banks. Six different resonators are obtained to increase the frequency tuning range. The proposed DCO covers GSM, UMTS, WiMAX, bluetooth and Wi-Fi frequency bands, whose frequency tuning range, channel bandwidth

and phase noise performance requirements are listed in Table 2.2. Then we apply the proposed DCO in the ADPLL for muilti-band wireless communication application. In this section, we first review the mathematical analysis and circuit topology of the quad-mode LC-tank DCO. Then, we present the multi-mode LC-tank DCO which has wide output frequency tuning range. At last, we provide the detailed circuit topology and simulation results by using Cadence Spectre simulator and IBM 0.13μm CMOS process design kit (PDK).

Table 2.2: Output frequency and phase noise performance requirements of multi-band DCO.

Standard	Frequency range	Channel	Phase noise requirements
GSM	880MHz-960MHz	4MHz	-105dBc/Hz@1MHz
UMTS	1.92GHz-2.17GHz	4MHz	-100dBc/Hz@5MHz
WiMAX	2.3GHz-2.7GHz	20MHz	-100dBc/Hz@1MHz
	3.3GHz-3.8GHz	20MHz	-105dBc/Hz@1MHz
Bluetooth	2.4GHz-2.48GHz	20MHz	-109dBc/Hz@1MHz
Wi-Fi	2.412GHz-2.472GHz	20MHz	-102dBc/Hz@1MHz
	5.15GHz-5.35GHz	20MHz	-102dBc/Hz@1MHz
	5.65GHz-5.85GHz	20MHz	-102dBc/Hz@1MHz

2.2.1 Quad-mode DCO Design Review

In our previous work, a quad-mode DCO is presented in reference [25]. The DCO employs three inductors and two varactor arrays (Var_1, Var_2) as shown in Fig. 2.3. By alternatively turning on two pairs of switches SW_{1N} and SW_{1P} , or SW_{2N} and SW_{2P}, the circuit can be converted into two different structures as shown in Fig. 2.3 (b) and (c), respectively. In structure-I, L_2, Var_2 branch features capacitive or inductive through Var_2 tuning. In structure-II, L_1, Var_1 shunt shows capacitive or inductive by setting Var_1 values. Thus, there are two different modes in each structure

producing two different frequency bands. Combining these two structures and four operation modes, the output operating frequency range covers GPS, bluetooth, Wi-Fi 802.11a/b/g frequency bands. Moveover, as the DCO eliminates the need of switches in the LC-tank, the resistive loss and phase noise effects are greatly alleviated.

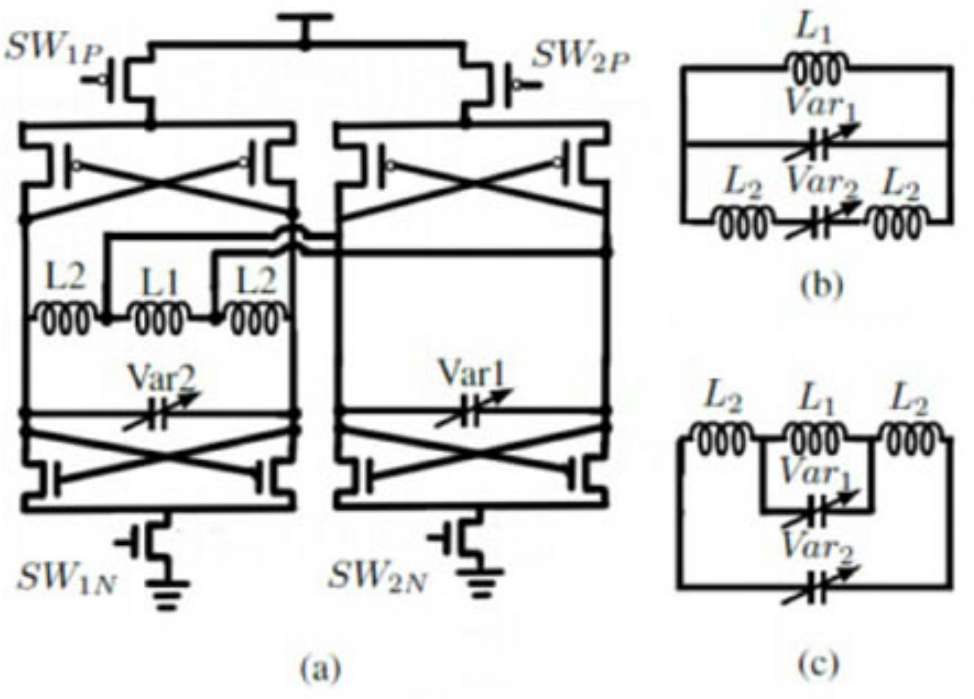


Figure 2.3: DCO models for (a) Quad-mode DCO; (b) Structure-I, (c) Structure-II.

The DCO structure-I model is shown in Fig. 2.3 (b) when SW_1 (SW_{1N} and SW_{1P}) is off and SW_2 (SW_{2N} and SW_{2P}) is on. Assuming the capacitances of two varactors Var_1 and Var_2 are C_{v1} and C_{v2} respectively. The corresponding tank resonant frequency is f_{osc}. The resonant frequency f_s for series connected L_1 and Var_1 sub-branch is:

$$f_s = \frac{1}{2\pi\sqrt{L_1 C_{v1}}} \tag{2.3}$$

By tuning the varactor Var_2, f_{osc} can be made lower than or higher than the DCO

output frequency f_s respectively. The L_2, Var_2 branch is inductive when $f_{osc} > f_s$ (Mode-I). The equivalent inductance is:

$$L_2' = 2L_2 + \frac{\frac{1}{j\omega_{osc}C_{v2}}}{j\omega_{osc}} = 2L_2 - \frac{1}{4\pi^2 f_{osc}^2 C_{v2}} \tag{2.4}$$

where ω_{osc} is DCO radial frequency. In mode-I, the DCO output frequency f_{osc} equals:

$$f_{osc1} = \frac{1}{2\pi\sqrt{(L_1||L_2') \cdot C_{v1}}} \tag{2.5}$$

The L_2, Var_2 branch is capacitive when $f_{osc} < f_s$ (Mode-II). The equivalent capacitance is:

$$C_2' = \frac{1/j\omega_{osc}}{(2j\omega_{osc}L_2 + \frac{1}{j\omega_{osc}C_{v2}})} = \frac{C_{v2}}{1 - 8\pi^2 f_{osc}^2 L_2 C_{v2}} \tag{2.6}$$

In this mode, the DCO output frequency f_{osc} is:

$$f_{osc2} = \frac{1}{2\pi\sqrt{L_1 \cdot (C_{v1} + C_2')}} \tag{2.7}$$

By manipulating equations (2.5) and (2.7), equations (2.8) and (2.9) can be obtained, which characterize the DCO output frequency in each mode. Apparent to see that the DCO can produce higher output frequency in mode-I than in mode-II.

$$f_{osc1,(3)} = \sqrt{\frac{L_1C_{v1} + L_1C_{v2} + 2L_2C_{v2} + \sqrt{(L_1C_{v1} + L_1C_{v2} + 2L_2C_{v2})^2 - 8L_1L_2C_{v1}C_{v2}}}{16\pi^2 L_1L_2C_{v1}C_{v2}}} \tag{2.8}$$

$$f_{osc2,(4)} = \sqrt{\frac{L_1C_{v1} + L_1C_{v2} + 2L_2C_{v2} - \sqrt{(L_1C_{v1} + L_1C_{v2} + 2L_2C_{v2})^2 - 8L_1L_2C_{v1}C_{v2}}}{16\pi^2 L_1L_2C_{v1}C_{v2}}} \tag{2.9}$$

Fig. 2.4 shows DCO output frequency in mode-I and mode-II when C_{var1} is changed from $0.5pF$ to $2.5pF$. Other circuit parameters are: $L_1 = 2.5nH$; $L_2 = 1.25nH$; C_{var2} equals $5pF$ in mode-I, and $0.5pF$ in mode-II. From Fig. 2.4, it can be observed that two different frequency bands are produced in these two operating modes. In mode-I, the DCO frequency band spans from 3.5GHz to 6.5GHz, while in mode-II, the DCO frequency band spans from 1.8GHz to 2.8GHz.

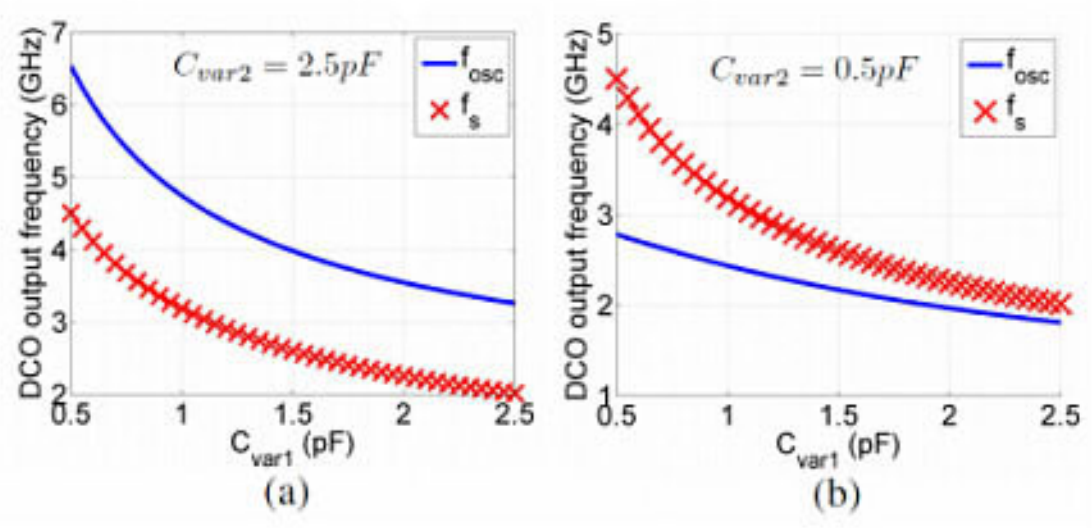


Figure 2.4: DCO output frequency f_{osc} (a) Mode-I; (b) Mode-II.

The DCO structure-II model is shown in Fig. 2.3 (c) when SW_1 is on and SW_2 is off. L_2 and Var_2 sub-branch resonant frequency equals:

$$f_s' = \frac{1}{2\pi\sqrt{2L_2C_{var2}}} \tag{2.10}$$

By tuning the capacitance of Var_1, f_s' can be made lower than or higher than the DCO output frequency f_{osc} respectively. L_1 and Var_1 branch impedance is capacitive when $f_{osc} > f_s'$ (Mode-III). The equivalent capacitance is:

$$C_1' = \frac{1/j\omega_{osc}}{j\omega_{osc}L_1 || \frac{1}{j\omega_{osc}C_{v1}}} = C_{v1} - \frac{1}{4\pi^2 f_{osc}^2 L_1} \tag{2.11}$$

The resulting DCO output frequency f_{osc} is:

$$f_{osc3} = \frac{1}{2\pi\sqrt{(2L_2 - \frac{L_1}{4\pi^2 f_{osc}^2 L_1 C_{v1} - 1}) \cdot C_{v2}}} \tag{2.12}$$

L_1 and Var_1 branch impedance becomes inductive when $f_{osc} < f_s'$ through Var_1 tuning. The equivalent inductance is:

$$L_1' = \frac{j\omega_{osc}L_1 || \frac{1}{j\omega_{osc}C_{v1}}}{j\omega_{osc}} = \frac{L1}{1 - 4\pi^2 f_{osc}^2 L_1 C_{v1}} \tag{2.13}$$

The corresponding DCO output frequency is:

$$f_{osc4} = \frac{1}{2\pi\sqrt{(2L_2 + \frac{L_1}{1 - 4\pi^2 f_{osc}^2 L_1 C_{v1}}) \cdot C_{v2}}} \tag{2.14}$$

By manipulating equations (2.13) and (2.14), the same equations (2.8) and (2.9) are obtained, which imply that the DCO output frequency is higher in mode-III than in mode-IV.

Fig. 2.5 shows the DCO output frequency in mode-III and mode-IV when C_{var2} is changed from $0.5pF$ to $2.5pF$. Other circuit parameters are $L_1 = 2.5nH$, $L2 = 1.25nH$, C_{var1} equals $2.5pF$ in mode-III and $0.5pF$ in mode-IV. As shown in Fig. 2.5,

in mode-III, when C_{var2} is tuned from $0.5pF$ to $2.5pF$, the DCO output frequency is produced from 4.5GHz to 5.5GHz, while in mode-IV, the frequency is from 1.5GHz to 3GHz.

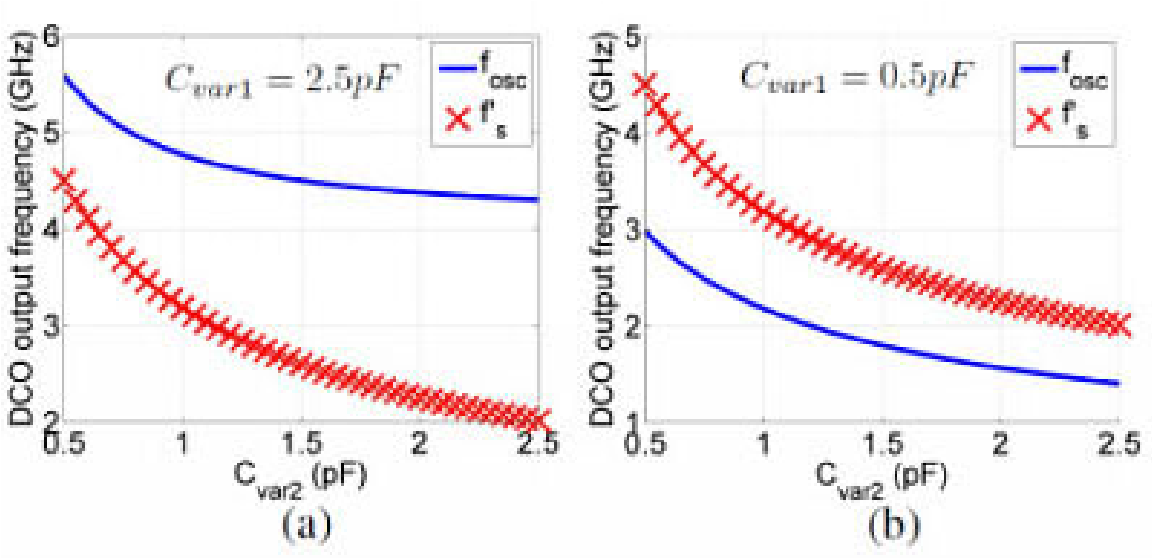


Figure 2.5: DCO output frequency f_{osc} (a) Mode-III; (b) Mode-IV.

The DCO output frequency and the frequency tuning range in each mode are also dependent on inductor ratio $R_L = L_2/L_1$. Fig. 2.6 shows DCO output frequencies versus different R_L in each mode by applying different L_2 values. The DCO output frequency is decreased with the raising of R_L value in all operation modes. In mode-II and mode-III, the DCO output frequency tuning range is reduced with the increasing of R_L value. From Fig. 2.6, when $R_L = 0.5$, the DCO output frequency from 3.5GHz-6.5GHz, 1.8GHz-2.8GHz, 4.5GHz-5.5GHz and 1.5GHz-3GHz can be obtained in mode-I, II, III and IV, respectively.

The quad-mode DCO circuit is implemented and simulated in IBM $0.13\mu m$ CMOS technology. The simulation tool is Cadence SpectreRF. The circuit operates under a

1.5V supply voltage.

Fig. 2.7 shows the detailed schematic of the proposed multi-band DCO. Circuit parameters are: $L_1 = 2.5nH$, $L_2 = 1.25nH$, Var_1 is tuned from $0.4pF$-$2pF$ and Var_2 tuning range is $0.66pF$-$3pF$. Transistors' sizes are listed in Table. 2.3. The structure selection is achieved by controlling power supply switches SW_1 and SW_2. The structure-I is utilized to generate higher frequency that covers Wi-Fi 802.11a frequency band, whose bandwidth is $200MHz$ spanning from $5.15GHz$ to $5.35GHz$ and $5.65GHz$ to $5.85GHz$. To meet such requirement, Var_1 consists of a 4-bit coarse tuning and an 11-bit fine tuning including a 5-bit digital to analog converter (DAC) controlled tuning. The coarse tuning varactor value is $0.38pF$-$1.85pF$, which makes coarse tuning resolution to be $200MHz$ per step when output frequency is in $5GHz-6GHz$ range. The fine tuning varactor value is $60fF$-$180fF$ results in $120KHz$ frequency resolution in higher frequency band. The structure-II is applied to generate lower frequency that covers GPS, bluetooth and Wi-Fi 802.11b and Wi-Fi 802.11g, whose frequency range is from $1.56GHz$ to $2.48GHz$. The maximum bandwidth among those standards is $80MHz$. Therefore, Var_2 is configured with a 5-bit coarse tuning and an 11-bit fine tuning including a 5-bit DAC controlled tuning. The coarse tuning varactor value is $0.66pF$-$3pF$, which makes coarse tuning resolution to be $25MHz$. The fine tuning varactor value is $60fF$-$180fF$ which produces $15KHz$ resolution in lower frequency band.

Fig. 2.8 shows the transient signal of the DCO output at 5.6GHz (Mode-I) and 2.4GHz (Mode-II). Fig. 2.9 shows the output frequency with the control code and phase noise performance. Circuit parameters, output frequency tuning range, DCO gain K_{DCO} and phase noise performances of each mode are summarized in Table. 2.3.

Table 2.3: Proposed DCO circuit parameters.

L_1/L_2	Var. pF	Mode	Freq. GHz	Phase Noise dBc/Hz@1MHz	Q	K_{DCO} Hz
2.5nH/ 1.25nH	Var_1 =0.4-2 Var_2 0.66-3	I	4.3-6.24	-112@5.6GHz	3.3	120K
		II	2.21-2.56	-117@2.4GHz	2.4	35K
		III	4.91-5.48	-114@5GHz	6.2	50K
		IV	1.49-2.52	-124@1.5GHz	8.1	15K
MOS. (W/L)	P1/P2/N3/N4: $100\mu m/0.13\mu m$; P3/P4/P5/P6: $70\mu m/0.13\mu m$; N1/N2: $40\mu m/0.13\mu m$; N5/N6: $30\mu m/0.13\mu m$					

FOM_T is calculated to evaluate frequency tuning range along with phase noise [26].

$$FOM_T = \mathcal{L}(f_{offset}) - 20log(\frac{f_{osc} \cdot FTR}{10 f_{offset}}) + 10log(\frac{P}{1mW}) \tag{2.15}$$

where $\mathcal{L}(f_{offset})$ is phase noise at offset frequency f_{offset}. P is power consumption. FTR is frequency tuning range in percentage. Table. 2.4 summarizes the comparison against other published wideband VCOs/DCOs. In addition, the simulation shows that the DCO power consumption is within the range from 7.6 mW to 11.5 mW.

Table 2.4: Comparison between wideband VCOs/DCOs

Ref. NO.	Tech. μm	Power mW	Phase Noise dBc/Hz@1MHz	DCO Freq. Range GHz	DCO Freq. Range %	FOM_T dBc/Hz
[1]	0.18	6-28	-102@2.9GHz	0.5-3	143	-180
[2]	0.18	7.1-16.3	-120@4.4GHz	3.14-6.44	69	-197
[24]	0.13	4.4-9.4	-114@5.6GHz	1.3-6	128	-201
[27]	0.18	4.6-6	-132@1.5GHz	2.4-2.52	4.9	-187
			-125@5.0GHz	4.65-5.12	9.6	-191
This work	0.13	7.6-11.5	-124@1.5GHz	1.49-2.56	53	-197
			-112@5.6GHz	4.3-6.24	37	-187

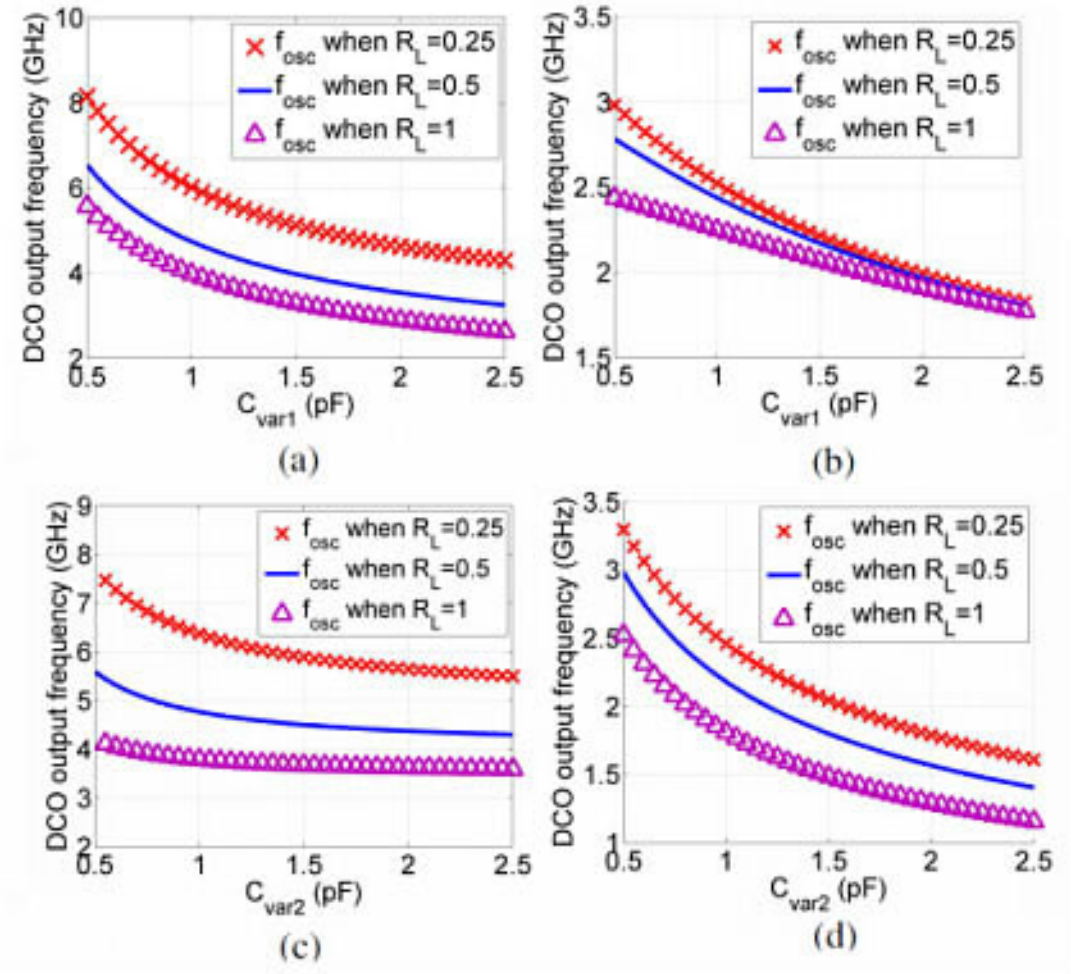


Figure 2.6: Frequency output under different L_2 to L_1 ratio. $L_1 = 2.5nH$; $L_2 = 0.625nH; 1.25nH; 2.5nH$. (a) Mode-I, (b) Mode-II, (c) Mode-III, (d)Mode-IV.

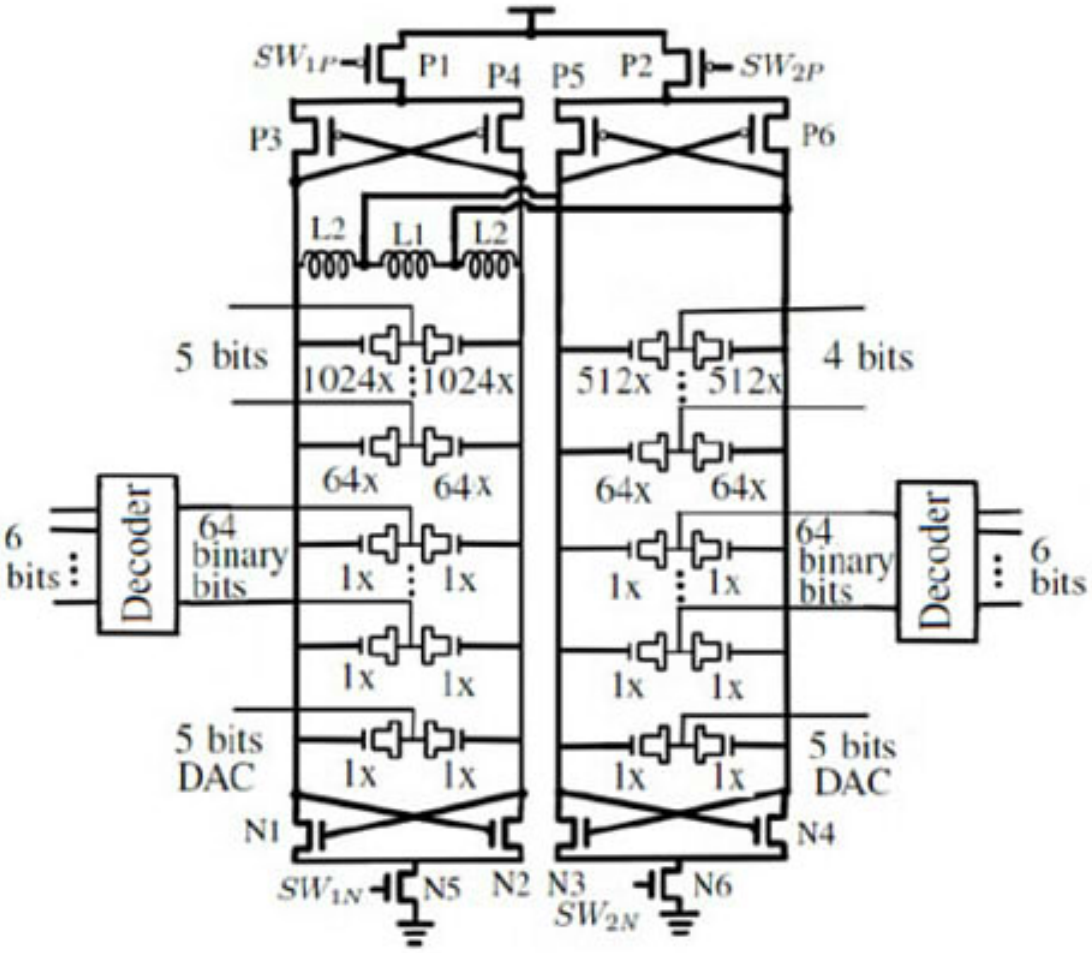


Figure 2.7: Detailed schematic of quad-mode DCO.

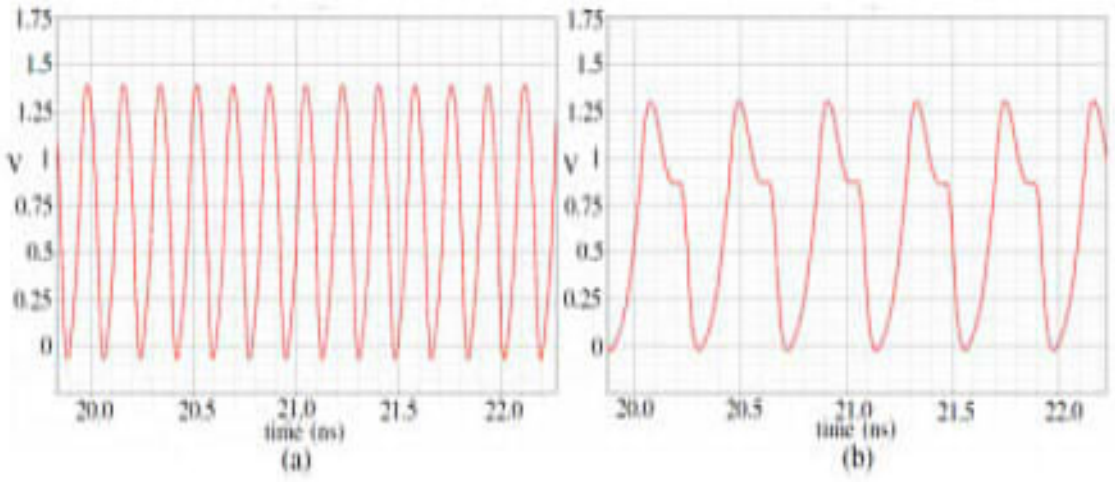


Figure 2.8: Transient signal of DCO output, (a) 5.6GHz; (b) 2.4GHz.

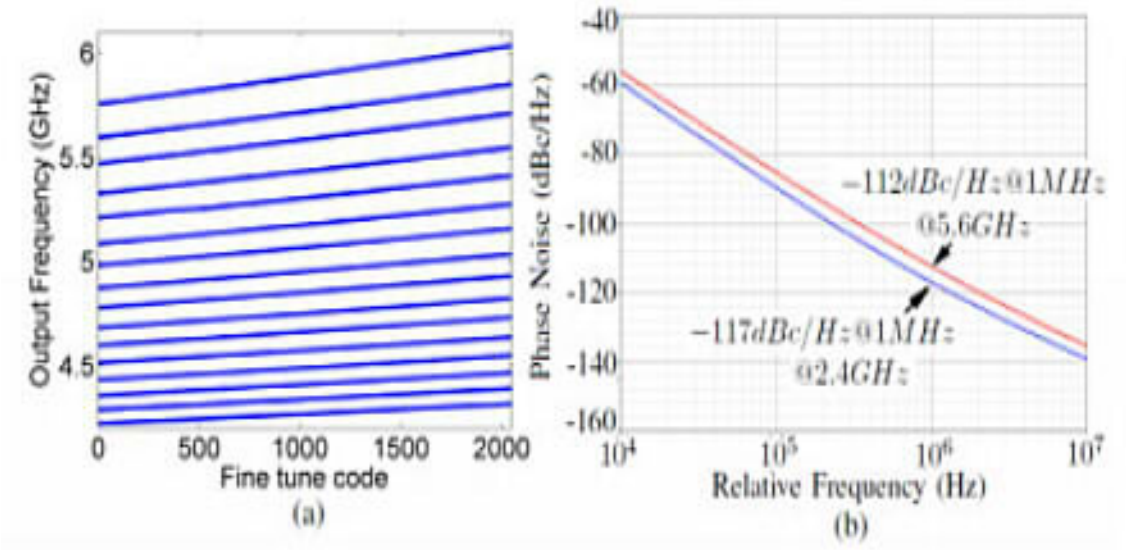


Figure 2.9: (a) Output frequency with control code in Mode-I, (b) Phase noise at frequency 5.6GHz and 2.4GHz.

2.2.2 Proposed Multi-mode DCO Design

In order to further increase the DCO frequency tuning range, we employ five inductors and three varactor arrays to achieve the multi-mode DCO to cover more frequency bands. As shown in Fig. 2.10, by alternatively turning on three pairs of switches SW_{1N} and SW_{1P}, or SW_{2N} and SW_{2P}, or SW_{3N} and SW_{3P} the circuit can be converted into three different structures as shown in Fig. 2.10 (b), (c) and (d) respectively.

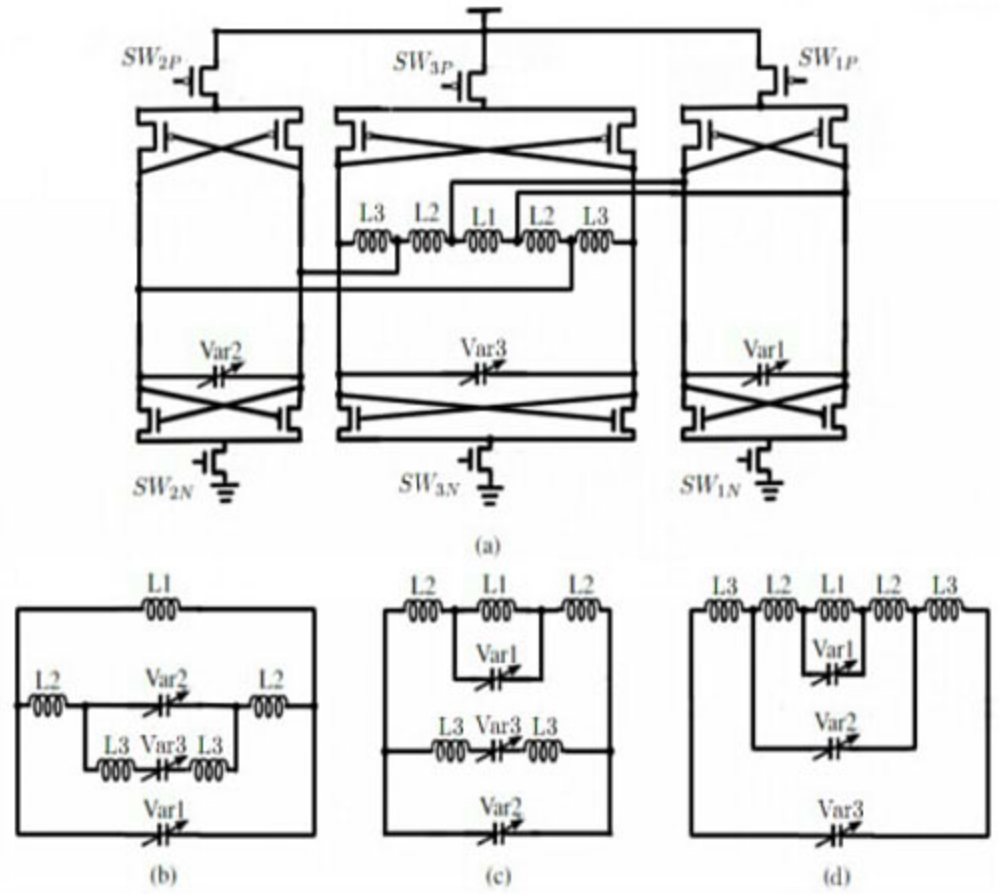


Figure 2.10: DCO models for (a) Multi-mode DCO; (b) Structure-I, (c) Structure-II, (d) Structure-III

Structure-I

The DCO structure-I model is shown in Fig. 2.10 (b) when SW_1 is on and SW_2, SW_3 are off. Assuming the capacitances of varactors Var_1, Var_2 and Var_3 are C_{v1}, C_{v2} and C_{v3} respectively. The corresponding tank resonant frequency is f_{osc}. The impedance I_1 for complex connected L_2, L_3, Var_2 and Var_3 sub-branch is:

$$I_1 = 2j\omega L_2 + \frac{1}{j\omega C_{v2}} || (2j\omega L_3 + \frac{1}{j\omega C_{v3}}) \tag{2.16}$$

The resonant frequency f_{sI} for L_1 and Var_1 is:

$$f_{sI} = 1/(2\pi\sqrt{L_1 C_{v1}}) \tag{2.17}$$

By tuning the varactor Var_2, f_{osc} can be made lower than or higher than frequency f_s. I_1 is inductive when $f_{osc} > f_{sI}$ (Mode-I). The equivalent inductance is:

$$L'_{I_1} = 2L_2 + \frac{1 - 2\omega_{osc}^2 C_{v3} L_3}{2\omega_{osc}^4 C_{v2} C_{v3} L_3 - \omega_{osc}^2 (C_{v2} + C_{v3})} \tag{2.18}$$

where $\omega_{osc} = 2\pi f_{osc}$ is DCO radial frequency. In mode-I, the DCO output frequency f_{oscI} equals:

$$f_{oscI} = 1/[2\pi\sqrt{(L_1 || L'_{I_1}) \cdot C_{v1}}] \tag{2.19}$$

By manipulating equations (2.18) and (2.19), the Mode-I DCO output frequency

$\omega_{oscI} = 2\pi f_{oscI}$ can be obtained.

$$\begin{aligned} &4C_{v1}C_{v2}C_{v3}L_1L_2L_3\omega_{oscI}^6 - 2(C_{v1}C_{v2}L_1L_2 + C_{v1}C_{v3}L_1L_2 + C_{v1}C_{v3}L_1L_3 \\ &+ C_{v2}C_{v3}L_1L_3 + 2C_{v2}C_{v3}L_2L_3)\omega_{oscI}^4 + (C_{v1}L_1 + C_{v2}L_1 + C_{v3}L_1 + 2C_{v2}L_2 \\ &+ 2C_{v3}L_2 + 2C_{v3}L_3)\omega_{oscI}^2 - 1 = 0 \end{aligned} \tag{2.20}$$

When $f_{osc} \leq f_{sI}$ (Mode-II), I_1 is capacitive.The equivalent capacitance is:

$$C'_{I_1} = \frac{C_{v2} + C_{v3} - \omega_{osc}^2 C_{v2}C_{v3}L_3}{2\omega_{osc}^2 L_2(\omega_{osc}^2 C_{v2}C_{v3}L_3 - C_{v2} - C_{v3})} \tag{2.21}$$

In mode-II, the DCO output frequency f_{oscII} equals:

$$f_{oscII} = 1/[2\pi\sqrt{(L_1 \cdot (C_{v1} + C'_{I_1})}] \tag{2.22}$$

By manipulating equations (2.21) and (2.22), the Mode-II DCO output frequency $\omega_{oscII} = 2\pi f_{oscII}$ equals:

$$\begin{aligned} &2C_{v1}C_{v2}C_{v3}L_1L_2L_3\omega_{oscII}^6 - 2(C_{v1}C_{v2}L_1L_2 + C_{v1}C_{v3}L_1L_2 + C_{v2}C_{v3}L_1L_3 \\ &+ C_{v2}C_{v3}L_2L_3)\omega_{oscII}^4 + (C_{v2}L_1 + C_{v3}L_1 + 2C_{v2}L_2 + 2C_{v3}L_2)\omega_{oscII}^2 = 0 \end{aligned} \tag{2.23}$$

Structure-II

The DCO structure-II model is shown in Fig. 2.10 (c) when SW_2 is on and SW_1, SW_3 are off. The impedance I_2 for series connected L_3 and Var_3 sub-branch is:

$$I_2 = 2j\omega L_3 + \frac{1}{j\omega C_{v3}} \tag{2.24}$$

The equivalent inductor of L_2, L_1 and Var_1 is

$$L_2' = 2L_2 + \frac{L_1}{1 - \omega^2 L_1 C_{v1}} \tag{2.25}$$

The resonant frequency f_{sII} for L_2' and Var_2 is:

$$f_{sII} = 1/(2\pi\sqrt{L_2' C_{v2}}) \tag{2.26}$$

By tuning the varactor Var_3, f_{osc} can be made lower than or higher than frequency f_{sII}. I_2 is inductive when $f_{osc} > f_{sII}$ (Mode-III). The equivalent inductance is:

$$L_{I_2}' = 2L_3 - \frac{1}{\omega^2 C_{v3}} \tag{2.27}$$

In mode-III, the DCO output frequency f_{oscIII} equals:

$$f_{oscIII} = 1/[2\pi\sqrt{(L_2' || L_{I_2}') \cdot C_{v2}}] \tag{2.28}$$

By manipulating equations (2.27) and (2.28), the Mode-III DCO output frequency $\omega_{oscIII} = 2\pi f_{oscIII}$ is:

$$\begin{aligned} &4C_{v1}C_{v2}C_{v3}L_1L_2L_3\omega_{oscIII}^6 - 2(C_{v1}C_{v2}L_1L_2 + C_{v1}C_{v3}L_1L_2 + C_{v1}C_{v3}L_1L_3 \\ &+ C_{v2}C_{v3}L_1L_3 + 2C_{v2}C_{v3}L_2L_3)\omega_{oscIII}^4 + (C_{v1}L_1 + C_{v2}L_1 + C_{v3}L_1 + 2C_{v2}L_2 \\ &+ 2C_{v3}L_2 + 2C_{v3}L_3)\omega_{oscIII}^2 - 1 = 0 \end{aligned} \tag{2.29}$$

When $f_{osc} \leq f_{sII}$ (Mode-IV), I_2 is capacitive.The equivalent capacitance is:

$$C'_{I_2} = \frac{C_{v3}}{1 - 2\omega^2 C_{v3} L_3} \tag{2.30}$$

In mode-IV, the DCO output frequency f_{oscIV} equals:

$$f_{oscIV} = 1/[2\pi\sqrt{(L'_2 \cdot (C_{v2} + C'_{I_2})}] \tag{2.31}$$

By manipulating equations (2.30) and (2.31), the Mode-IV DCO output frequency $\omega_{oscIV} = 2\pi f_{oscIV}$ equals:

$$\begin{aligned} &2C_{v1}C_{v2}C_{v3}L_1L_2L_3\omega_{oscIV}^6 - 2(C_{v1}C_{v2}L_1L_2 + C_{v1}C_{v3}L_1L_2 + C_{v2}C_{v3}L_1L_3 \\ &+ C_{v2}C_{v3}L_2L_3)\omega_{oscIV}^4 + (C_{v2}L_1 + C_{v3}L_1 + 2C_{v2}L_2 + 2C_{v3}L_2)\omega_{oscIV}^2 = 0 \end{aligned} \tag{2.32}$$

Structure-III

The DCO structure-III model is shown in Fig. 2.10 (d) when SW_3 is on and SW_1, SW_2 are off. The equivalent inductance is:

$$L'_3 = \frac{L_1 + 2(1 - 4\pi^2 f_{osc}^2 L_1 C_{v1})L_2}{(1 - 8\pi^2 f_{osc}^2 L_2 C_{v2})(1 - 4\pi^2 f_{osc}^2 L_1 C_{v1}) - 4\pi^2 f_{osc}^2 L_1 C_{v2}} + 2L_3 \tag{2.33}$$

The DCO output frequency f_{oscV} equals:

$$f_{oscV} = \frac{1}{2\pi\sqrt{L'_3 \cdot C_{v3}}} \tag{2.34}$$

The resulting DCO output frequency f_{oscV} is:

$$f_{oscV} = \frac{1}{2\pi\sqrt{[\frac{L_1+2(1-4\pi^2 f_{osc}^2 L_1 C_{v1})L_2}{(1-8\pi^2 f_{osc}^2 L_2 C_{v2})(1-4\pi^2 f_{osc}^2 L_1 C_{v1})-4\pi^2 f_{osc}^2 L_1 C_{v2}} + 2L_3]C_{v3}}} \tag{2.35}$$

Structure select block

Fig. 2.11 shows the circuit for DCO structure selecting. The 2-bit structure select code MS<1:0> has four different codes. Once MS<1:0> is "11" or "10", SW_1 turns on, SW_2 and SW_3 are turned off. While MS<1:0> is "01", SW_2 turns on, SW_1 and SW_3 are turned off. While MS<1:0> is "00", SW_3 turns on, SW_1 and SW_2 are turn off. The simplified circuits of inverter and 2-input OR gate are shown in Fig. 2.12.

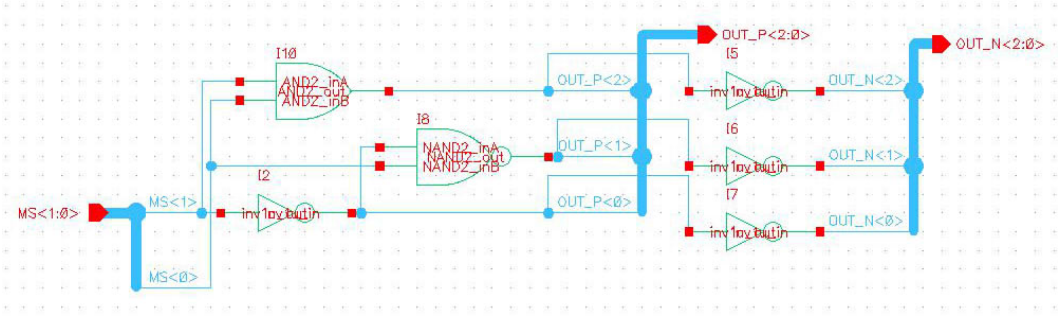


Figure 2.11: 2-bit DCO structure select circuit

The corresponding detailed transistor level circuits are shown in Fig. 2.13. Detail transistors' sizes are listed in Table 2.5

The true table of structure selecting block is shown as Table 2.6.

Varactor array

In the IBM 0.13μm CMOS technology, the minimum size of varactor is 1μm/240nm (W/L). We use varactor array as capacitance tuning block. We apply 5-bit coarse

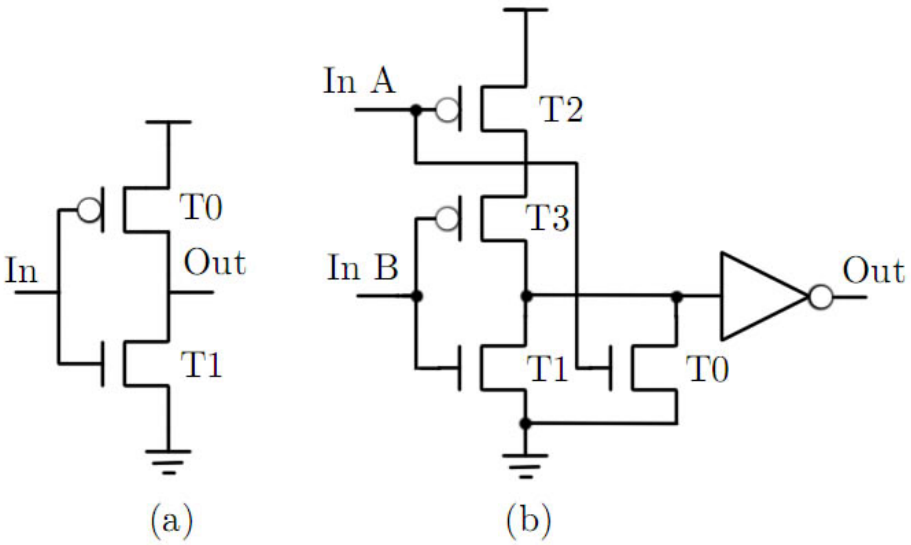


Figure 2.12: Simplified circuits of (a) Inverter (b) 2-input OR gate.

Table 2.5: Transistors' sizes in structure selector

			W	L
Inverter	T0	PMOS	$10\mu m$	$130nm$
INV	T1	NMOS	$5\mu m$	$130nm$
2-input OR gate	T0-T1	NMOS	$5\mu m$	$130nm$
OR2	T2-T3	PMOS	$10\mu m$	$130nm$

tuning blocks in both structure-I and structure-II and a 6-bit coarse tuning block in structure-III. For fine tune, we have a 6-bit fine tuning block. The simplified circuit of 5-bit coarse tune, 6-bit coarse tune and 6-bit fine tune are shown in Fig. 2.14 Detailed transistor level circuits of varactor arrays are shown in Fig. 2.15. The varactors' sizes are listed in Table 2.7.

As shown in Table 2.7, the capacitance tuning ranges of 5-bit coarse tuning, 6-bit coarse tuning and 6-bit fine tuning are $0.86pF \sim 5.48pF$, $1.73pF \sim 11.1pF$ and $29fF \sim 173fF$, respectively. The 3-bit fine tune controlled by the third order delta

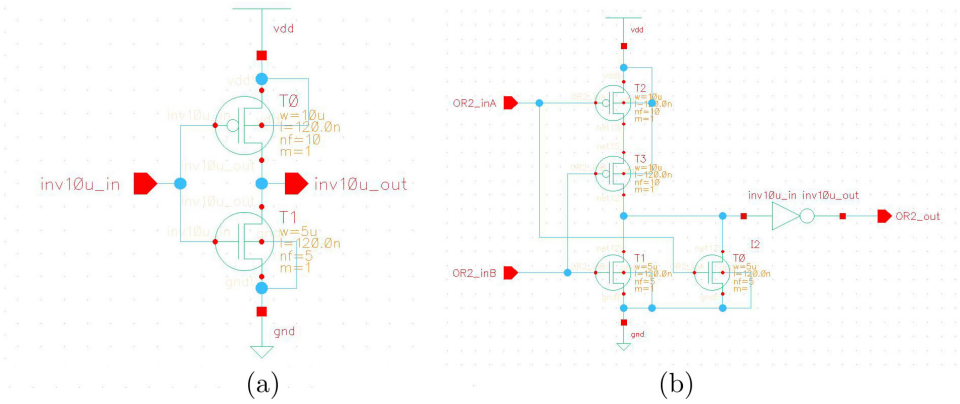


Figure 2.13: Transistor level circuits of (a) Inverter (b) 2-input OR gate.

Table 2.6: DCO structure select true table

Structure select code		Structure select switch						Frequency
MS<1>	MS<0>	$S1_P$	$S1_N$	$S2_P$	$S2_N$	$S3_P$	$S3_N$	(GHz)
1	1	0	1	1	0	1	0	5.0-6.0
1	0	0	1	1	0	1	0	3.2-4.0
0	1	1	0	0	1	1	0	1.9-2.8
0	0	1	0	1	0	0	1	0.8-1.1

sigma modulator is shown in Fig.2.16. The transistors' sizes and capacitances are listed in Table 2.8.

Fig. 2.17 shows a 3:1 multiplexer to chose the DCO output.

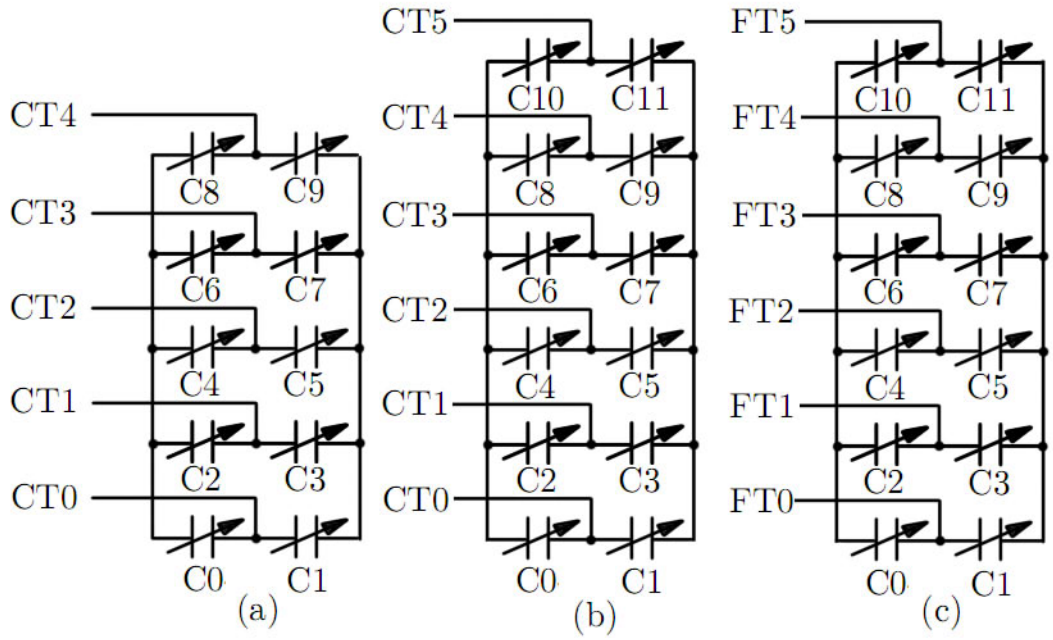


Figure 2.14: Simplified circuits of (a) 5-bit coarse tune (b) 6-bit coarse tune (c) 6-bit fine tune.

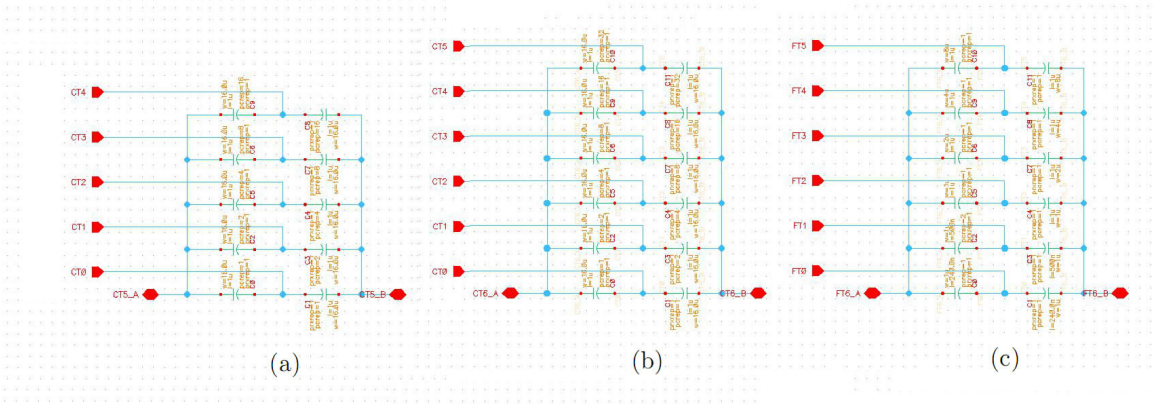


Figure 2.15: Transistor level circuits of (a) 5-bit coarse tune (b) 6-bit coarse tune (c) 6-bit fine tune.

Table 2.7: Varactors' sizes in varactor banks

		W (μm)	L (μm)	No. of Gates	Min. Cap. Datasheet (fF)	Max. Cap. Datasheet (fF)	Min.(1.2V) Cap. Measured (fF)	Max.(0V) Cap. Measured (fF)
5-bit Coarse Tune	C0-C1	16	1	1	37.7	176.2	27.5	176.6
	C2-C3	16	1	2	75.3	352.3	57.0	353.9
	C4-C5	16	1	4	150.7	704.6	112.4	708.2
	C6-C7	16	1	8	301.3	1409	237.5	1415
	C8-C9	16	1	16	602.7	2819	426.5	2829
6-bit Coarse Tune	C0-C1	16	1	1	37.7	176.2	27.5	176.6
	C2-C3	16	1	2	75.3	352.3	57.0	353.9
	C4-C5	16	1	4	150.7	704.6	112.4	708.2
	C6-C7	16	1	8	301.3	1409	237.5	1415
	C8-C9	16	1	16	602.7	2819	426.5	2829
	C10-C11	16	1	32	1205	5637	869	5657
6-bit Fine Tune	C0-C1	1	0.24	1	0.89	2.70	0.77	2.69
	C2-C3	1	0.5	1	1.49	5.52	1.21	5.53
	C4-C5	1	1	1	2.65	10.95	2.08	10.98
	C6-C7	2	1	1	4.99	21.96	3.75	22.08
	C8-C9	4	1	1	9.66	43.99	7.34	44.24
	C10-C11	8	1	1	19.0	88.0	13.8	88.2

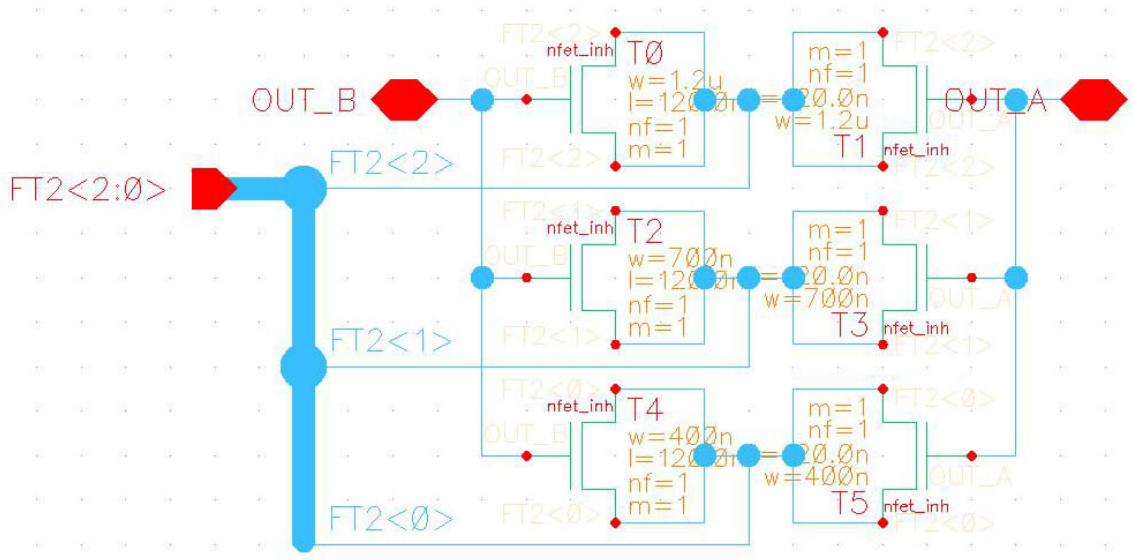


Figure 2.16: Transistor level circuits of the 3-bit fine tune.

Table 2.8: Transistors' sizes in 3-bit fine tune

		W	L	Min. Cap. (1.2V)	Max. Cap. (0V)
		(nm)	(nm)	(fF)	(fF)
3-bit Fine tune	T0-T1	1200	130	0.85	1.53
	T2-T3	700	130	0.49	0.89
	T4-T5	400	130	0.27	0.50

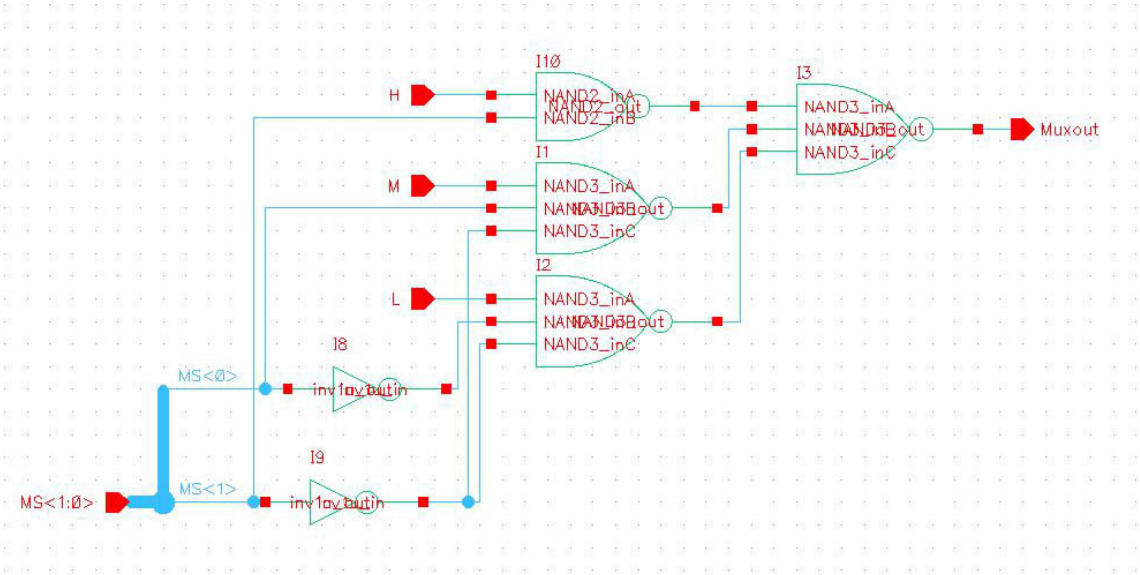


Figure 2.17: 3:1 multiplexer for DCO output selection.

2.2.3 Experimental Results

The proposed multi-mode DCO is implemented and simulated in the IBM 0.13μm CMOS technology. The simulation tool is Cadence SpectreRF. The circuit operates under a 1.2V supply voltage. Fig. 2.18 shows the detailed schematic of the proposed multi-band DCO.

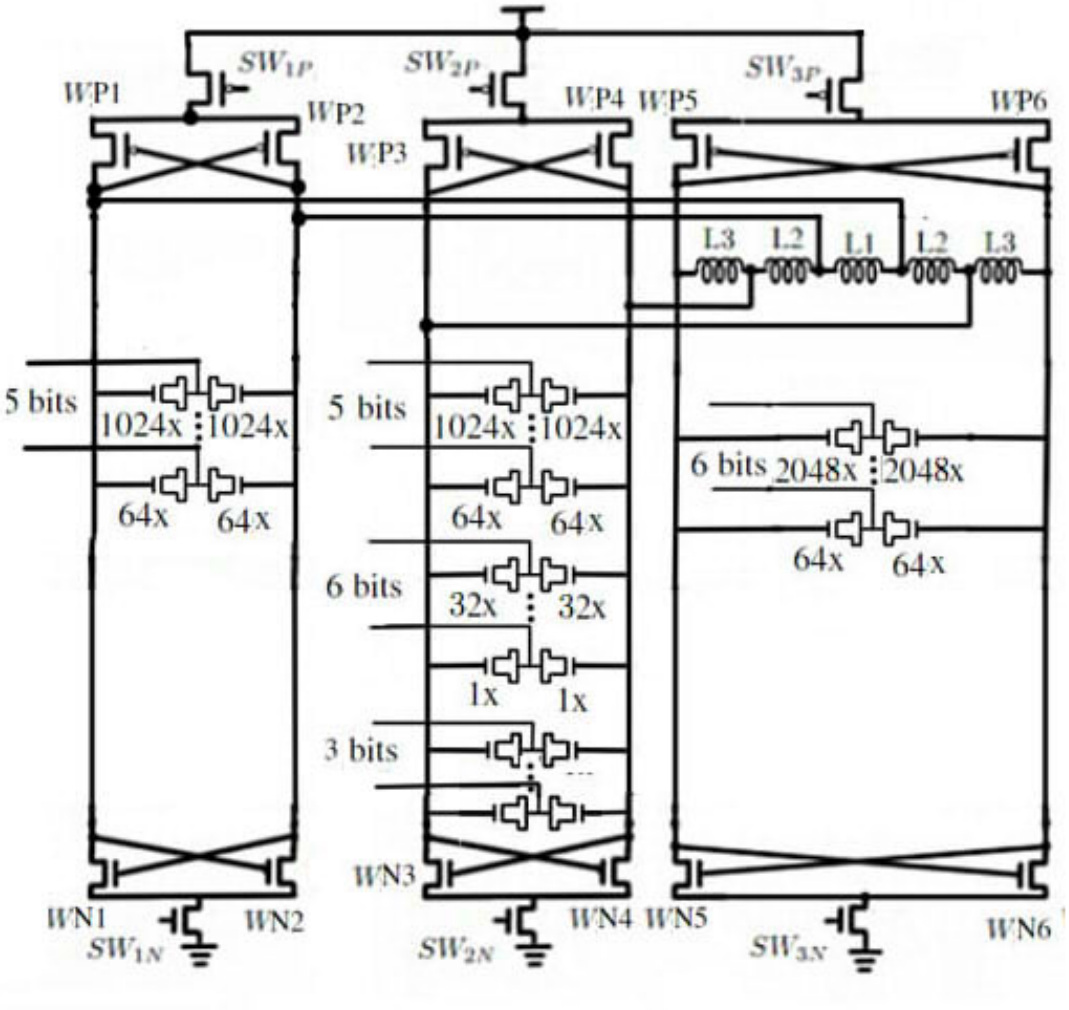


Figure 2.18: Detailed schematic of proposed DCO.

Circuit parameters including inductances, varactor arrays' capacitances and transistors' sizes are listed in Table. 2.9

Table 2.9: Proposed LC tank DCO parameters

Str.	L (nH)	Varactor (pF)	WN (W/L) ($\mu m/\mu m$)	WP (W/L) ($\mu m/\mu m$)	WNS (W/L) ($\mu m/\mu m$)	WPS (W/L) ($\mu m/\mu m$)
1	1.6	1.1-5.4	WN1-2: 75/0.13	WP1-2:75/0.13	50/0.13	100/0.13
2	1.4	1.2-5.6	WN3-4: 75/0.13	WP3-4:75/0.13	75/0.13	100/0.13
3	3.4	2.4-11	WN5-6: 35/0.13	WP5-6:50/0.13	35/0.13	100/0.13

The detailed transistor level circuit schematic of the proposed DCO is shown in Fig. 2.19.

The structure selection is achieved by controlling power supply switches SW_1, SW_2 and SW_3. The mode-I of structure-I is utilized to generate higher frequency that covers Wi-Fi 802.11a frequency band, whose bandwidth is $200MHz$ spanning from $5.15GHz$ to $5.35GHz$ and $5.65GHz$ to $5.85GHz$. To meet such requirement, Var_1 consists of a 5-bit coarse tuning and Var_2 consists of a 5-bit coarse tuning and an 11-bit fine tuning including the third order delta sigma modulator (5-bit input/3-bit output) controlled tuning. The coarse tuning resolution is $35MHz$ per step when output frequency is in $5GHz - 6GHz$. The fine tuning results in $30KHz$ frequency resolution in higher frequency band. Fig. 2.20 shows the $5.6GHz$ DCO output transient signal and its phase noise performance. Fig. 2.21 shows the phase noise performance at $1MHz$ offset of DCO in mode-I. The worst case of phase noise at $1MHz$ offset is $-106.6dBc/Hz$ when the output frequency is $6GHz$.

The mode-II of structure-I generates frequency that covers WiMAX frequency band, whose bandwidth is $50MHz$ spanning from $3.3GHz$ to $3.8GHz$. The coarse

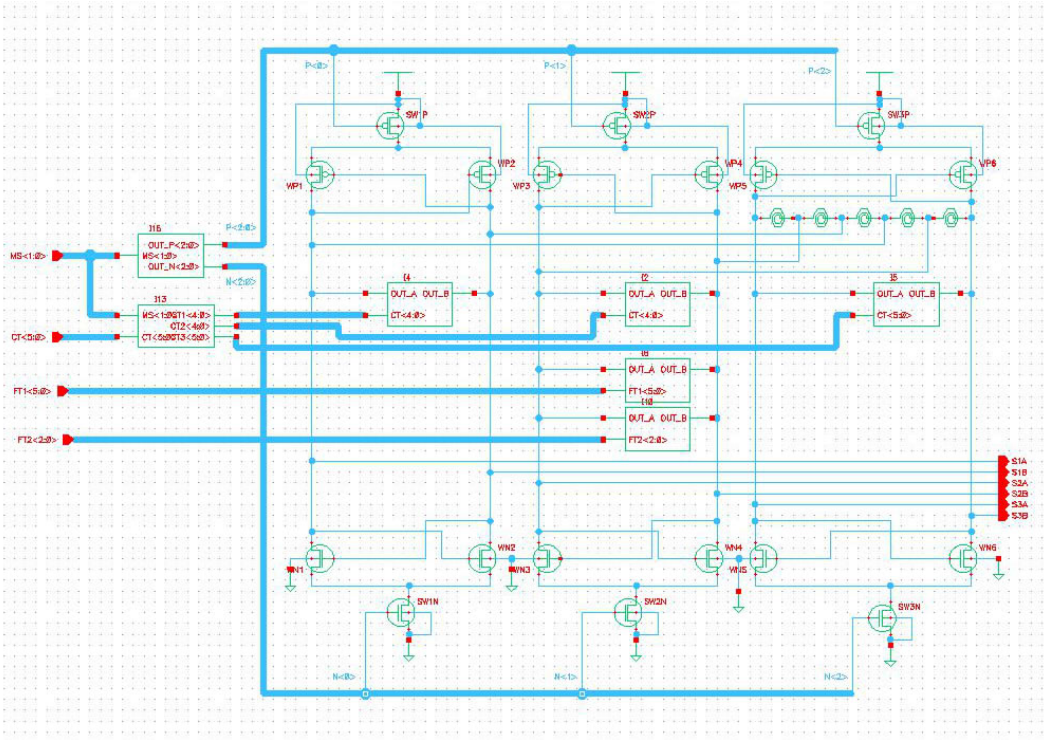


Figure 2.19: Detailed transistor level circuit of proposed DCO in Cadence.

tuning resolution is $30MHz$ per step when output frequency is in $3.2GHz$-$4GHz$. The fine tuning results in $15KHz$ frequency resolution in such frequency band. Fig. 2.22 shows the $3.6GHz$ DCO output transient signal and its phase noise performance. Fig. 2.23 shows the phase noise performance at $1MHz$ offset of DCO in mode-II. The worst case of phase noise at $1MHz$ offset is $-109.6dBc/Hz$ when the output frequency is $4GHz$.

The structure-II is applied to generate frequency that covers UMTS, WiMAX $2.3GHz - 2.7GHz$, bluetooth, Wi-Fi 802.11b and Wi-Fi 802.11g, whose frequency range is from $1.92GHz$ to $2.7GHz$. The maximum bandwidth among those standards is $80MHz$. Therefore, Var_2 is configured with a 5-bit and an 11-bit fine tuning including the third order delta sigma modulator controlled tuning. The output fre-

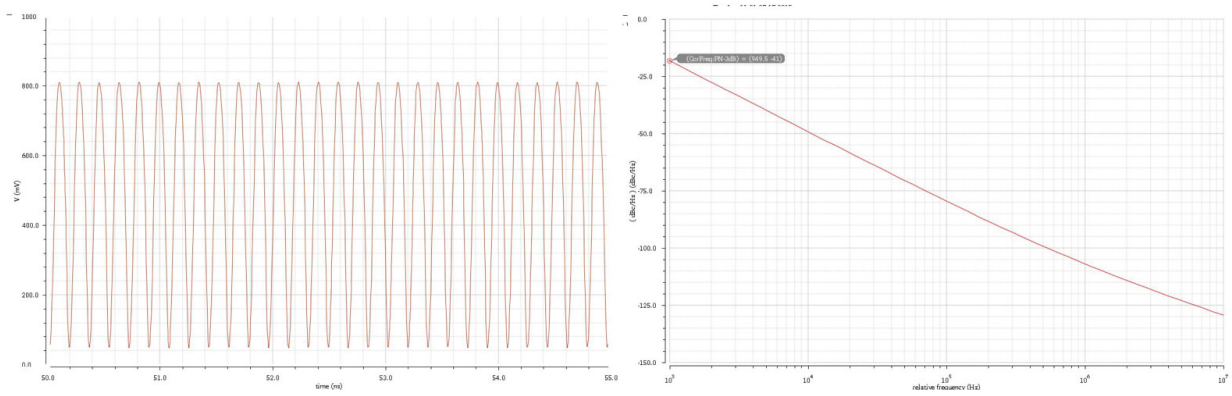


Figure 2.20: Transient signal and phase noise of DCO output at 5.6GHz.

quency range is $1.9GHz - 2.8GHz$. The coarse tuning resolution is $30MHz$. The fine tuning resolution is $15KHz$. Fig. 2.24 shows the $2.4GHz$ DCO output transient signal and its phase noise performance. Fig. 2.25 shows the phase noise performance at $1MHz$ offset of DCO in structure-II. The worst case of phase noise at $1MHz$ offset is $-110.3dBc/Hz$ when the output frequency is $2.8GHz$.

The structure-III is applied to generate lower frequency that covers GSM, whose frequency range is from $880MHz$ to $960MHz$. The bandwidth is $4MHz$. Therefore, Var_3 is configured with 6-bit. The fine tuning is accomplished by Var_2. The coarse tuning resolution is $5MHz$. The fine tuning resolution is $3KHz$. Fig. 2.26 shows $925MHz$ DCO output transient signal and its phase noise performance. Fig. 2.27 shows the phase noise performance at $1MHz$ offset of DCO in structure-III. The worst case of phase noise at $1MHz$ offset is $-118.5dBc/Hz$ when the output frequency is $1.1GHz$.

Since the DCO output frequency is varied according to the digital control code. The linear characteristic of digital control code to frequency of the DCO is essential for the ADPLL. The linearity of the proposed DCO is presented in Figure.

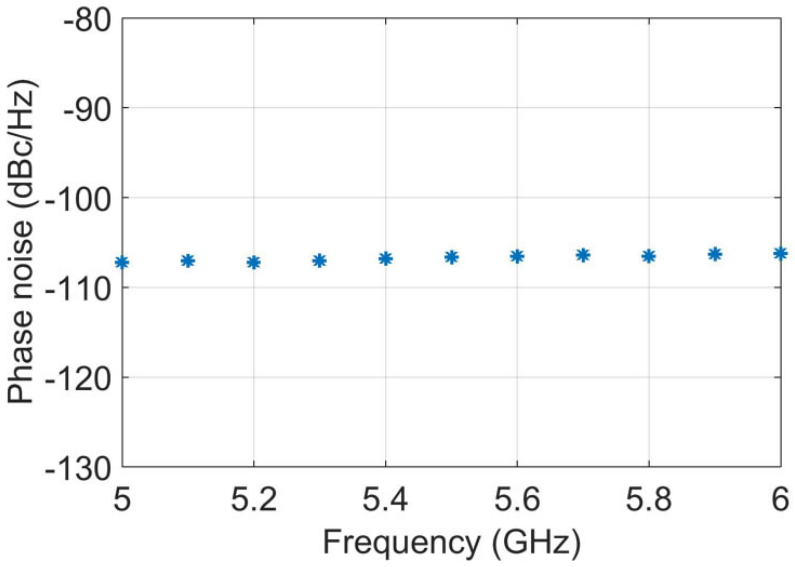


Figure 2.21: Phase noise at 1MHz offset of DCO output at 5-6GHz.

The power consumption is very important in VLSI systems. In the ADPLL, the DCO power consumption is one of the most important issues. The simulation shows that the proposed DCO power consumption ranges from 8.2 mW to 12.3 mW. This is a reasonable power consumption for the multi-band ADPLL. In the worst case: 12.3 mW when DCO has the highest frequency (6GHz) and the best case: 8.2 mW when DCO has the lowest frequency (850MHz). The proposed DCO is more power efficient than the conventional wide band DCOs in many research papers.

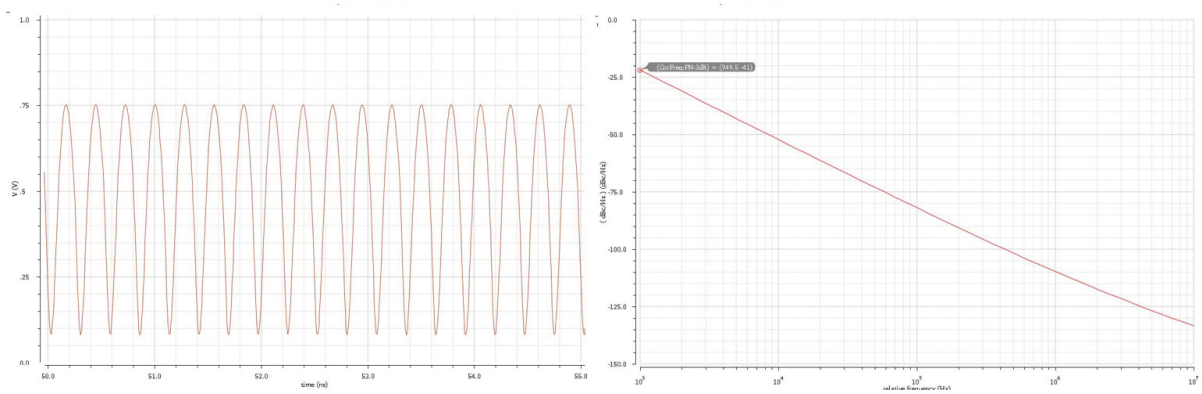


Figure 2.22: Transient signal and phase noise of DCO output at 3.6GHz.

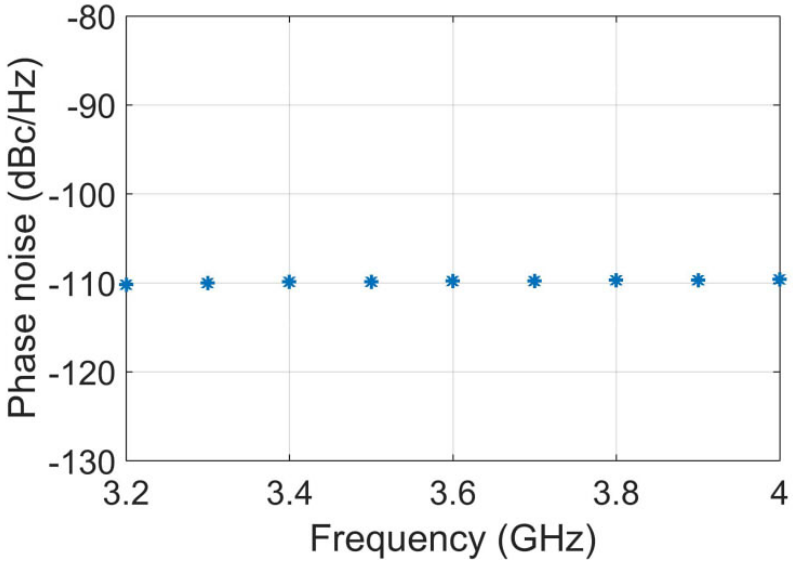


Figure 2.23: Phase noise at 1MHz offset of DCO output at 3.2-4GHz.

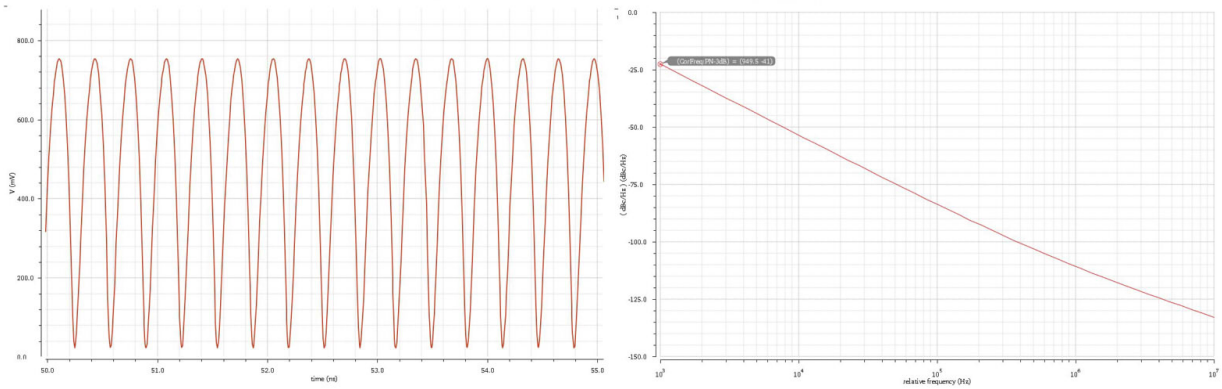


Figure 2.24: Transient signal and phase noise of DCO output at 2.4GHz.

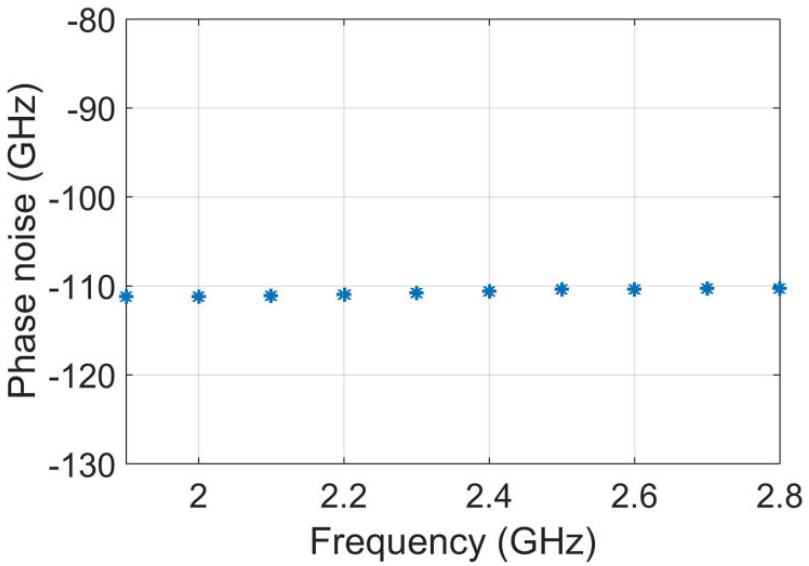


Figure 2.25: Phase noise at 1MHz offset of DCO output at 1.9-2.8GHz.

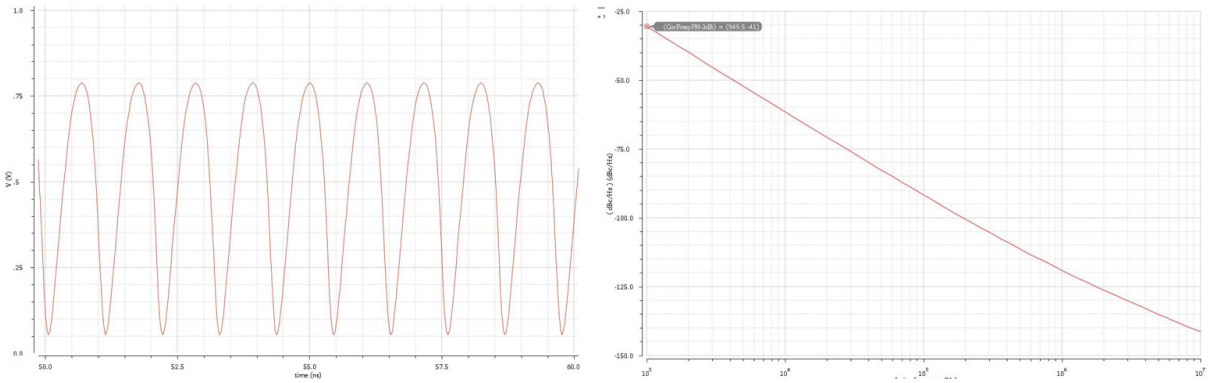


Figure 2.26: Transient signal and phase noise of DCO output at 925MHz.

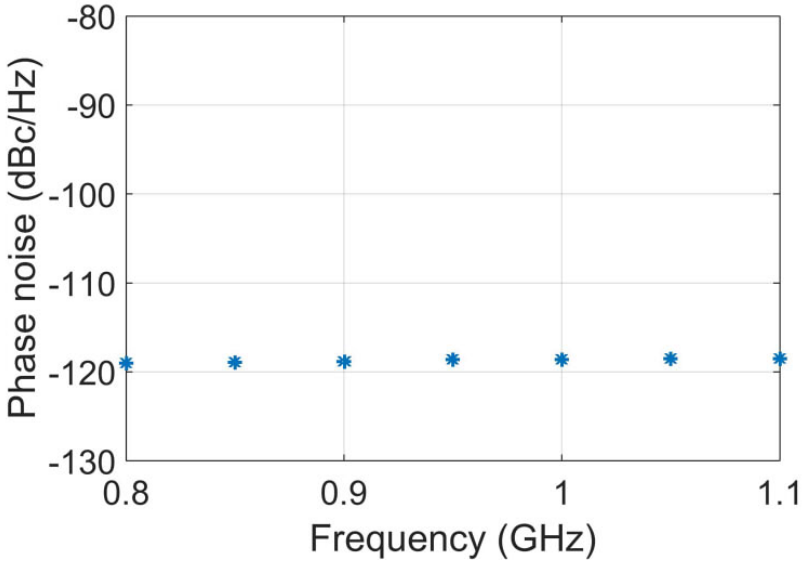


Figure 2.27: Phase noise at 1MHz offset of DCO output at 0.8-1.1GHz.

2.3 Time to Digital Converter based Phase Frequency Detector Design

A phase frequency detector (PFD) is a function block which compares the phase of reference signal and feedback signal. Fig. 2.28 shows a traditional implementation of PFD. A PFD is basically consists of two D-type flip flops. It has two outputs including UP and DOWN signals. One Q output enables the UP signal, and the other Q output enables the DOWN signal.

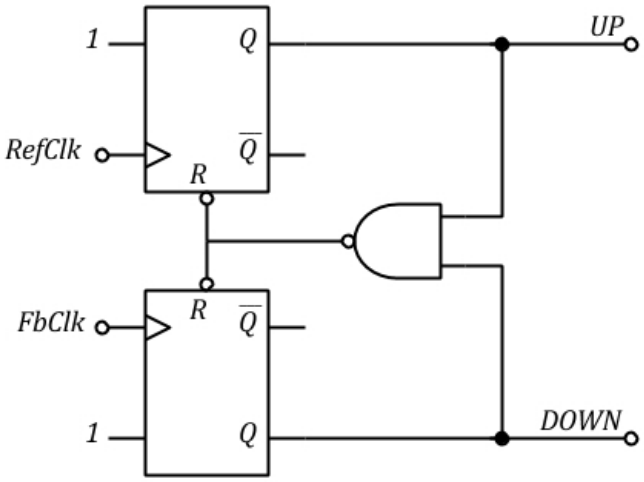


Figure 2.28: PFD with 2-D flip flops.

The minimum pulse-width of the PFD output called dead zone as shown in Fig. 2.29 is the most important problem of PFD. In order to mitigate the dead zone problem, the reset signal should be designed as the trigger pulses with a constant width at the PFD outputs. However, the PFD has the blind zone during the reset process, where the PFD can not work any transitions on the input signals. If the

phase difference is in the blind zone during the frequency acquisition, the PFD delivers wrong phase difference information.

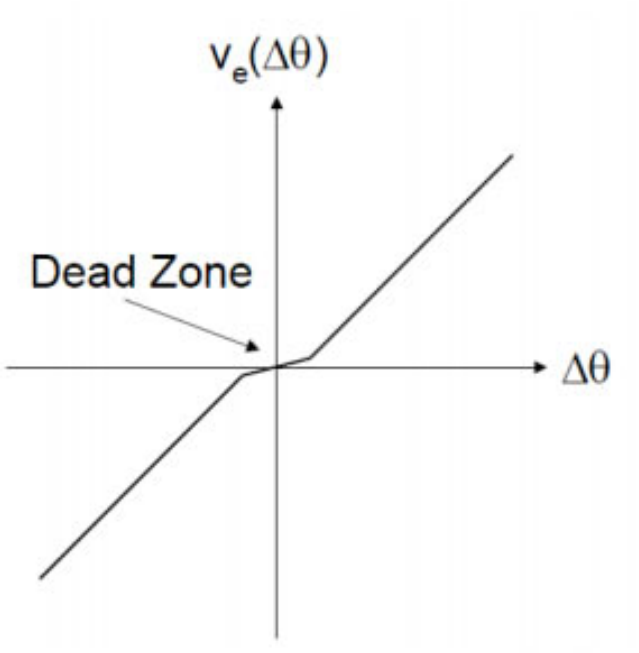


Figure 2.29: PFD dead zone.

Due to the existence of blind zone, the chance of cycle for comparisons the phase and frequency differences and PLL frequency acquisition time are increased. In order to reduce the blind zone, an extra delay cell is added in most designs. Our approach reduces the blind zone close to the theoretical limit imposed by PVT variations.

Fig. 2.30 shows the phase frequency detector (PFD) employed in our design. The PFD is composed of four inverters, four 2-input NAND gates, three 3-input NAND gates and one 4-input NAND gate.

The simplified circuits and transistor level circuits of 2-input , 3-input, and 4-input NAND gates are shown in following. The detailed transistors' sizes are listed in Table. 2.10.

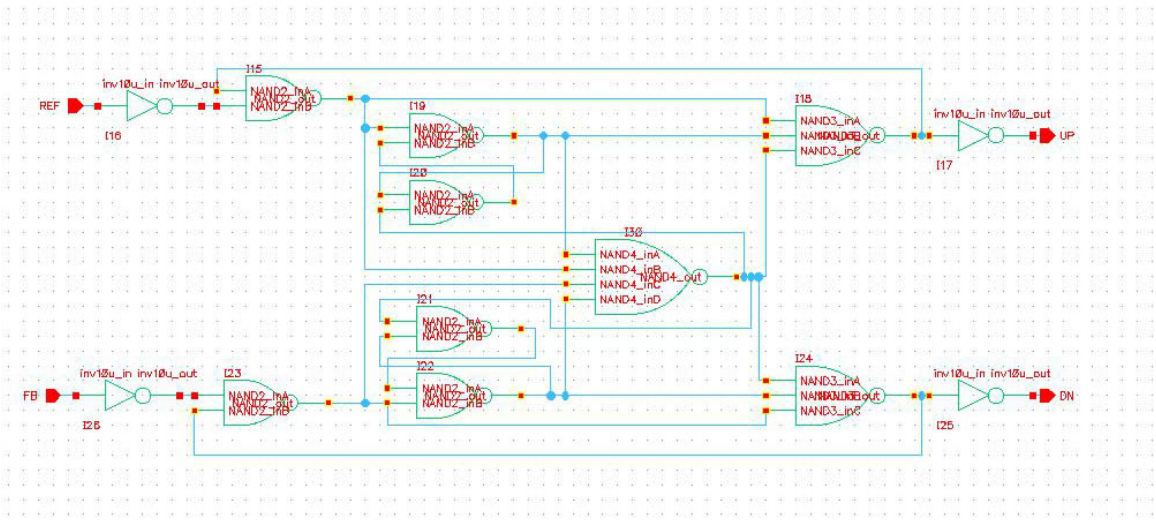


Figure 2.30: Gate level phase frequency detector.

Table 2.10: Transistors' sizes in NAND gates

			W	L
2-input NAND gate	T0-T1	NMOS	$10\mu m$	$130nm$
	T2-T3	PMOS	$10\mu m$	$130nm$
3-input NAND gate	T0-T2	NMOS	$15\mu m$	$130nm$
	T3-T5	PMOS	$10\mu m$	$130nm$
4-input NAND gate	T0-T3	NMOS	$20\mu m$	$130nm$
	T4-T7	PMOS	$10\mu m$	$130nm$

When the phase of reference signal equals the phase of feedback signal, the up (UP) and down (DN) signals are zero as shown in Fig. 2.33. While the reference leads feedback signal, UP signal is high and DN signal is low as shown in Fig. 2.34. While the reference lags feedback signal, UP signal is low and DN signal is high as shown in Fig. 2.35.

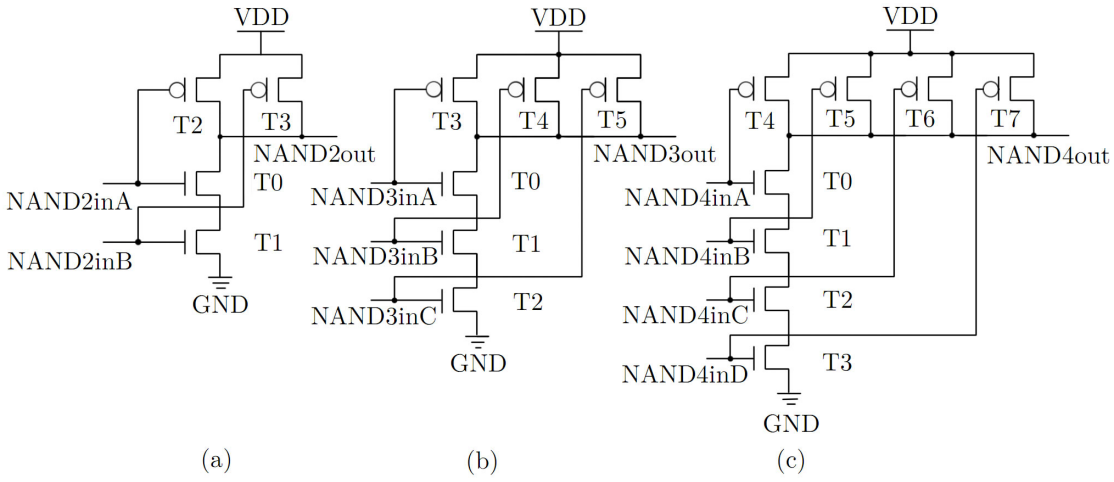


Figure 2.31: The simplified circuits of NAND gate (a) 2-input (b) 3-input, (c) 4-input.

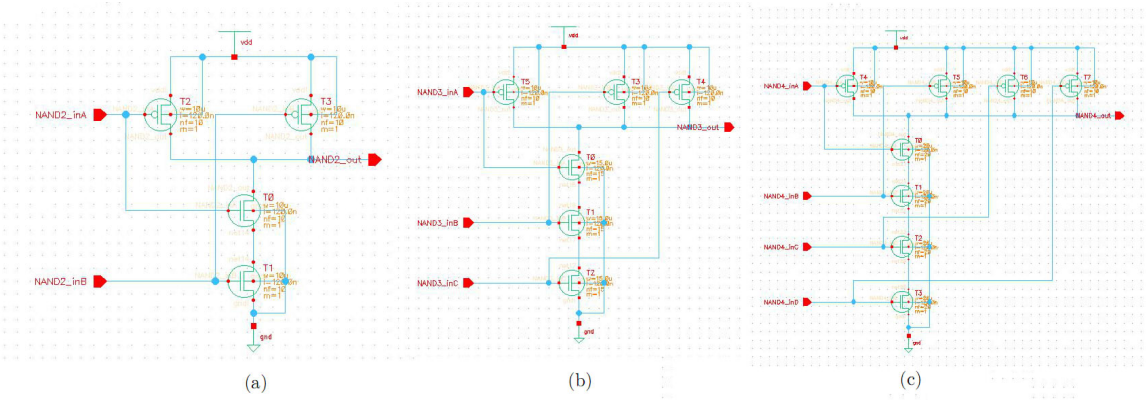


Figure 2.32: The transistor level circuits of NAND gate (a) 2-input (b) 3-input, (c) 4-input.

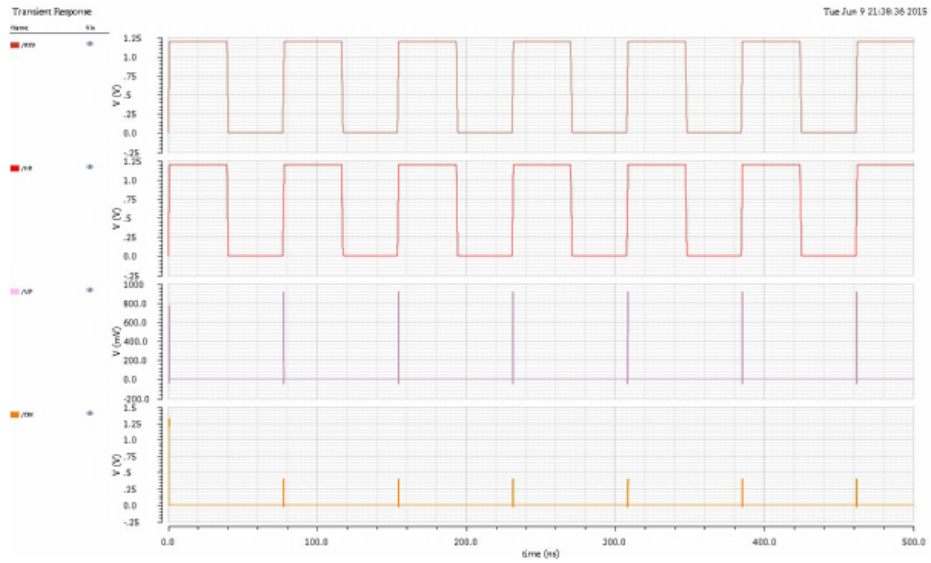


Figure 2.33: PFD simulation results when phase of reference and feedback signal is same.

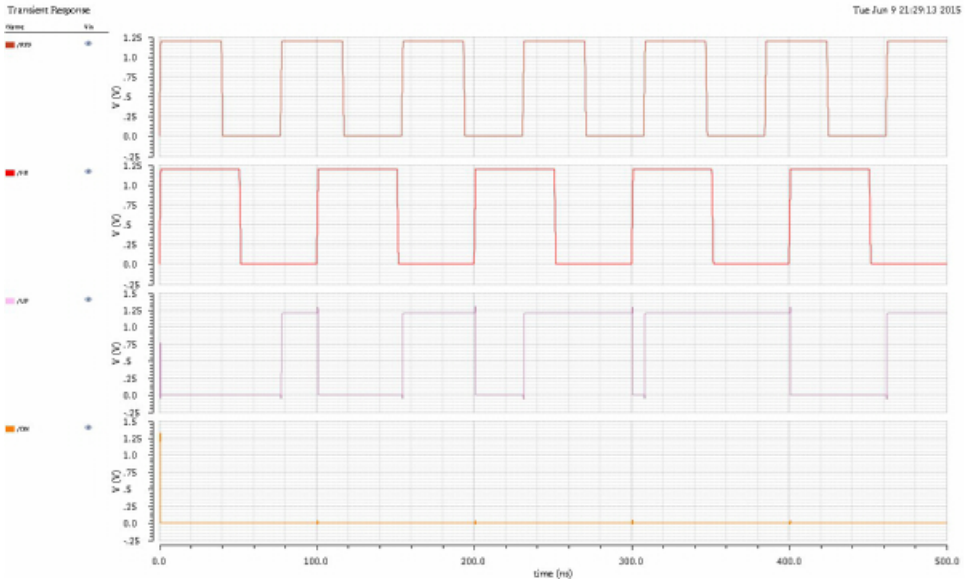


Figure 2.34: PFD simulation results when reference leads feedback signal.

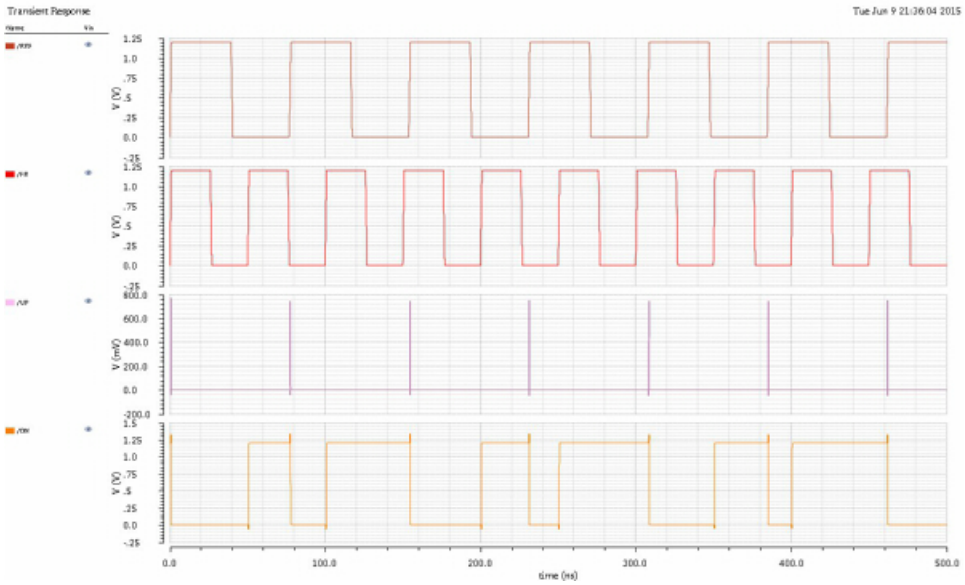


Figure 2.35: PFD simulation results when reference lags feedback signal.

2.3.1 Basic delay line based TDC

Fig. 2.36 shows an implementation of the basic delay-line based TDC.

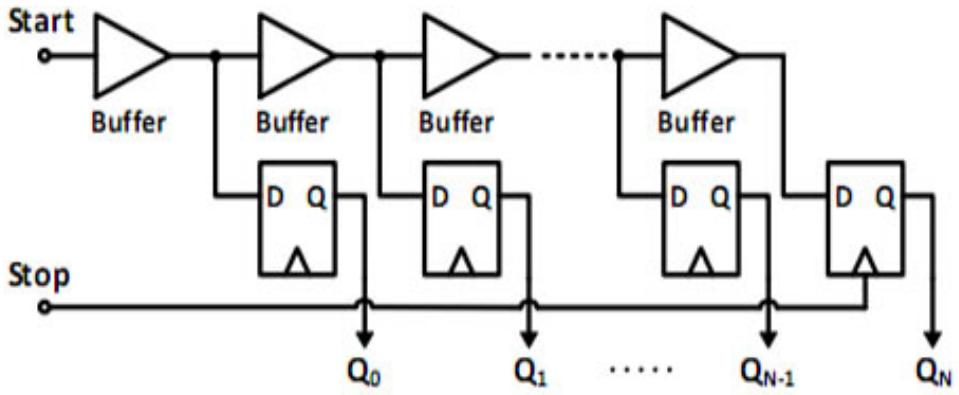


Figure 2.36: Implementation of a basic delay-line based TDC.

Fig. 2.37 shows a 3-bit inverter chain based TDC schematic by using IBM 8rf technology.

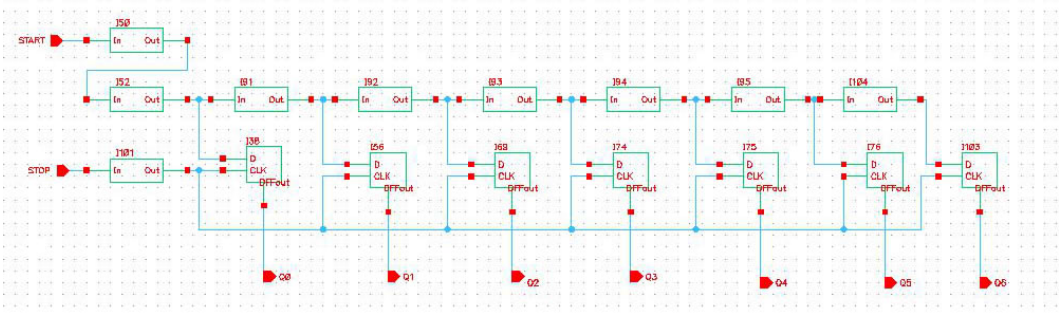


Figure 2.37: Implementation of 3 bit inverter based TDC.

It is hard to further reduce the delay value in inverter based delay cell. The TDC resolution is $\frac{400ps}{2^3} = 50ps$. Fig. 2.38 shows the linearity of such 3-bit inverter chain based TDC.

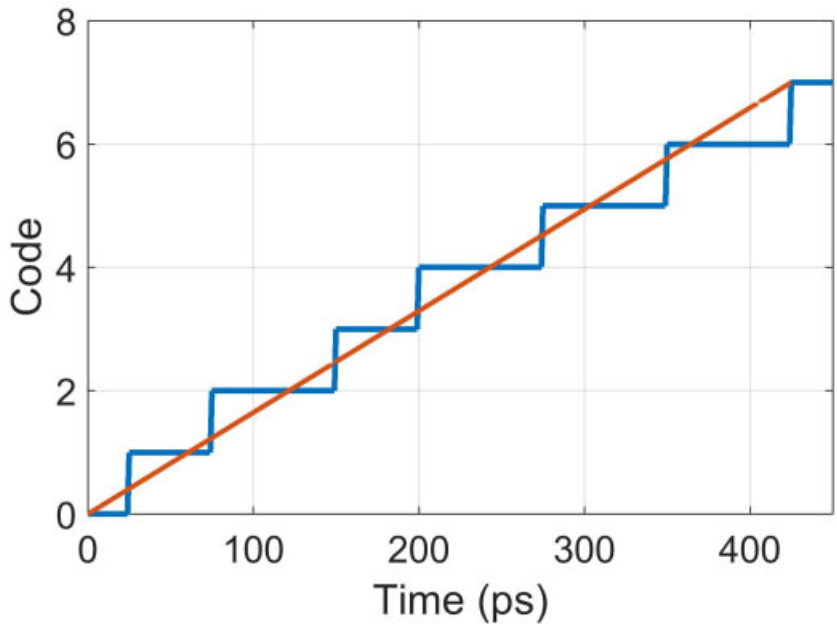


Figure 2.38: The linearity of 3 bit inverter based TDC.

2.3.2 Vernier TDC

Fig. 2.39 shows a vernier delay line TDC. The basic concept of the vernier delay chain technique is that the timing resolution is determined by the difference between two propagation delay values. A vernier delay chain structure consists of a pair of delay lines with a D-flip flop at each corresponding pair of delay cell. A stop signal propagates through the faster delay chain, while the start signal propagates through the other chain, clocking the flip flop at each stage. The difference between the stop and start propagation delays calculates the timing between adjacent stages.

The dynamic range of the TDC based on vernier delay chain is limited to

$$t_{DR} = n \cdot (\tau_1 - \tau_2) \tag{2.36}$$

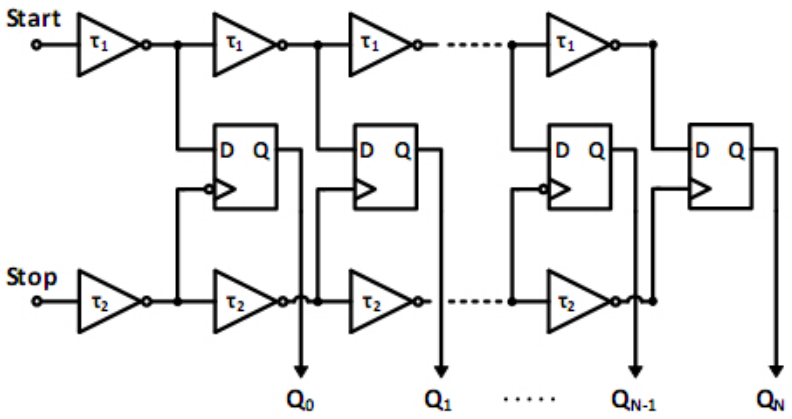


Figure 2.39: Implementation of a vernier TDC.

where n is the number of delay cells of the delay line.

Fig. 2.40 shows a 3-bit vernier delay line TDC schematic by using IBM 8rf technology. The resolution equals $4ps$ and the dynamic range is $28ps$. Fig. 2.41 shows its

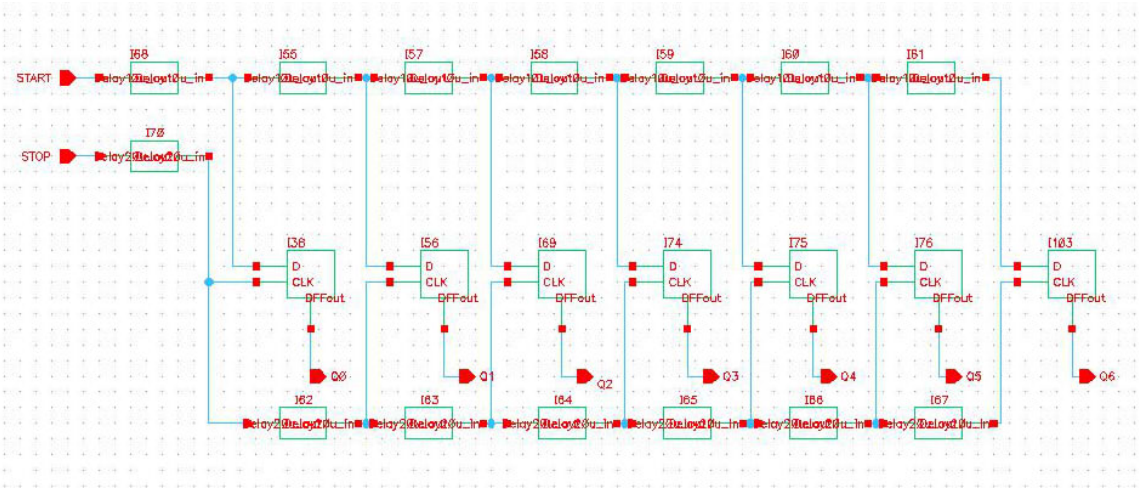


Figure 2.40: Implementation of a 3 bit vernier TDC.

linearity.

The variation is an important problem of the performance and behavior of TDCs

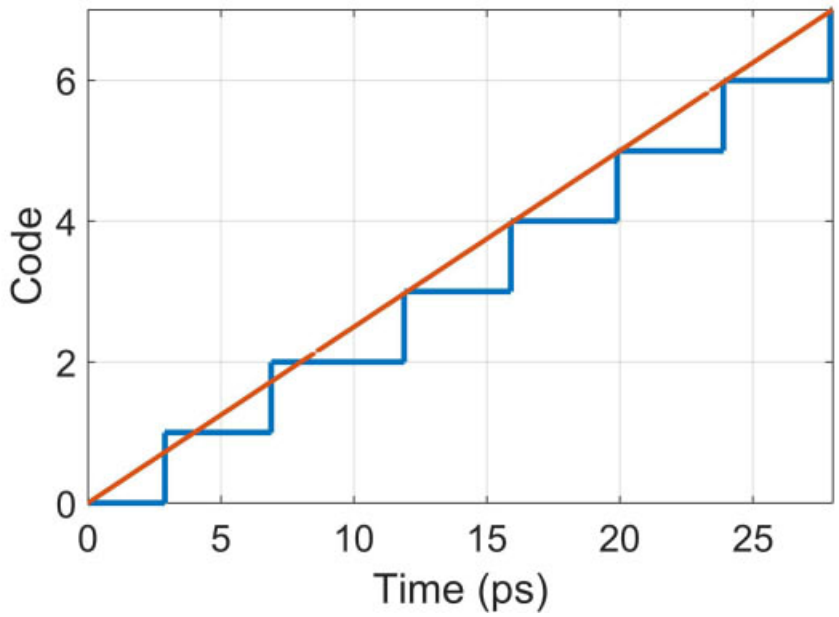


Figure 2.41: The linearity of 3 bit vernier delay chain TDC.

due to the process variation and environmental noise. In the TDC, the gate delays in the delay cell are changed by variations. Therefore, the variation of TDC should be considered.

2.3.3 Proposed Time to Digital Converter based Phase Frequency Detector Design

In order to increase the dynamic range of vernier TDC. A 6-bit vernier TDC is presented as shown in Fig. 2.42. It is composed of 63 pairs of delay cells and 63 D-flip flops.

The Fig. 2.43 shows the detailed TDC schematic of vernier TDC and fat tree encoder [74]. The resolution equals $4ps$ and the dynamic range is $252ps$. The linearity

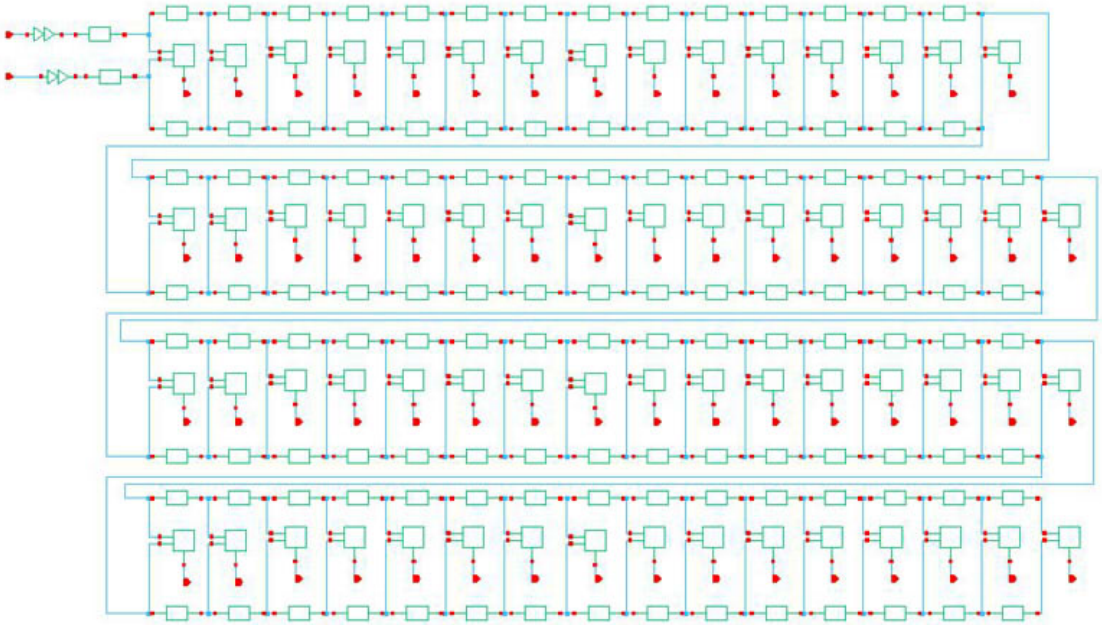


Figure 2.42: Implementation of a 6 bit vernier TDC.

of proposed vernier TDC is shown in Fig. 2.44

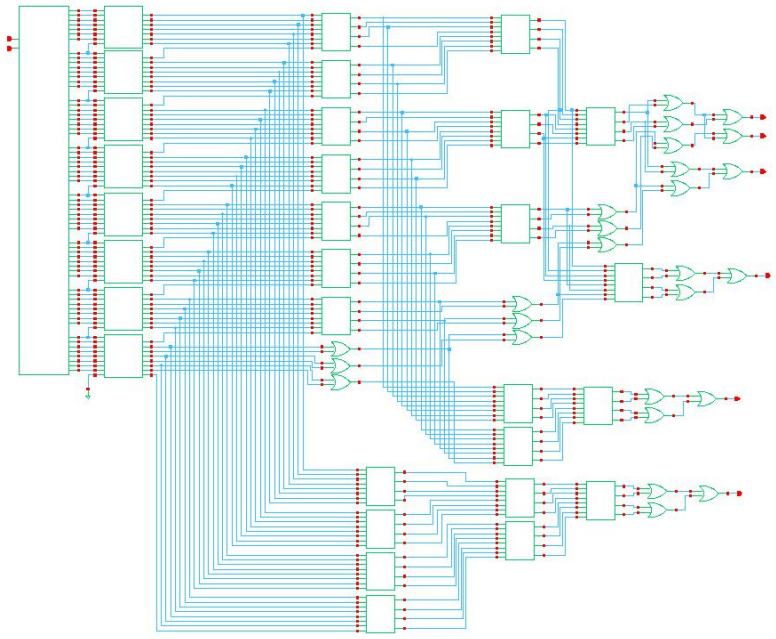


Figure 2.43: The detailed TDC schematic of vernier TDC and fat tree encoder.

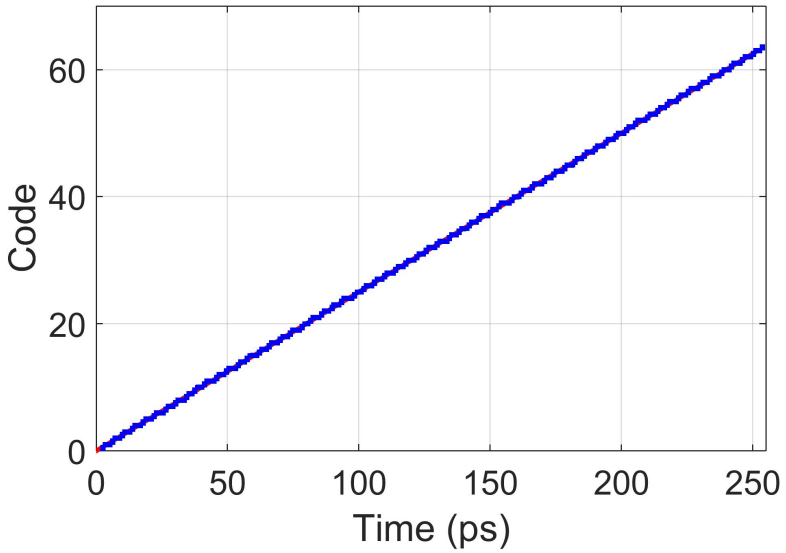


Figure 2.44: The linearity of 6 bit vernier TDC.

2.4 Digital Loop Filter

The traditional loop filter (analog loop filter) is consisted of resistors and capacitors. It has large size area and its output is quite noisy. In this design, we replace the bulky passive loop filter by a more flexible digital loop filter. As a basic building block in digital systems, digital loop filter has advantages including higher programmability, less size area and lower power consumption. Digital filter frequency response depends on the value of its coefficients. The values of the coefficients can be obtained based on the desired frequency response or phase noise, locking time requirements [12]. Typically, digital filters are categorized as finite impulse response (FIR) filters and infinite impulse response (IIR) filters.

2.4.1 FIR Filter

A finite impulse response (FIR) filter whose impulse response is of finite duration. Fig. 2.45 shows the basic architecture of FIR filter. For a causal discrete-time FIR

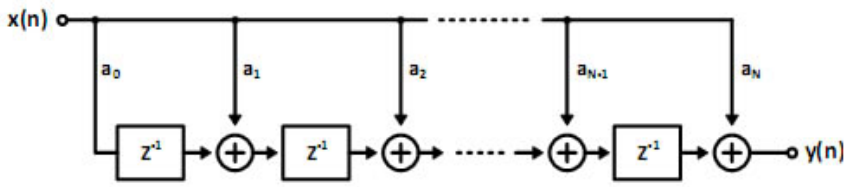


Figure 2.45: The basic architecture of FIR filter.

filter of order N, each value of the output is a summation of the most recent input

values:

$$y[n] = a_0x[n] + a_1x[n-1] + \cdots + a_Nx[n-N] = \Sigma_{i=0}^{N} a_i \cdot x[n-i] \tag{2.37}$$

where $x[n]$ is the input signal, y[n] is the output signal, N is filter order and a_i is the value of the impulse response at the corresponding ith instant for $0 \leq i \leq N$ of an Nth order FIR. The transform function of a typical FIR filter can be expressed as a polynomial of z^{-1}. All the poles of FIR transfer function are located at origins so that FIR filter always stable. In FIR filter, the output depends only on the previous inputs.

The advantages of FIR filter including linear phase response, simply design, bounded input bounded output (BIBO) stability and low sensitivity to filter coefficient quantization errors.

2.4.2 IIR Filter

IIR filters are digital filters with infinite impulse response. IIR filters have the feedback and are known as recursive digital filters. Fig. 2.46 shows the basic architecture of IIR filter. IIR filters are often described and implemented as following:

$$y[n] = \frac{1}{a_0}(b_0x[n]+b_1x[n-1]+\cdots+b_Px[n-P]-a_1y[n-1]-\cdots-a_Qy[n-Q]) \tag{2.38}$$

where P, Q are the filter order of feed forward and feedback, respectively. b_i are feed forward filter coefficients and a_i are feedback filter coefficients. The transfer function

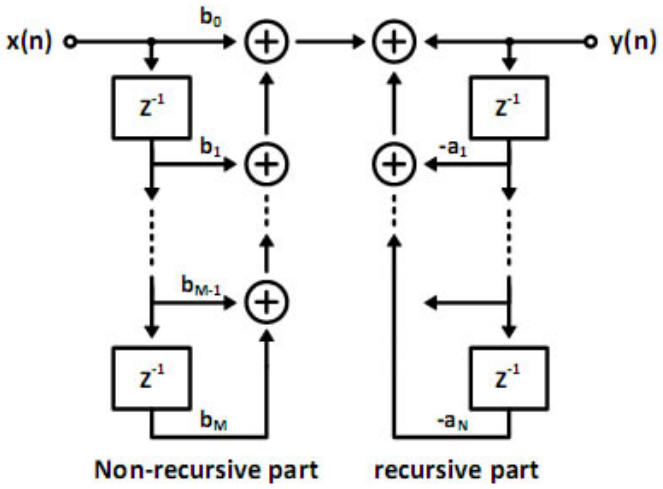


Figure 2.46: The basic architecture of IIR filter.

of IIR filter is generally expressed as following equation:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\Sigma_{i=0}^{P} b_i z^{-i}}{1 + \Sigma_{j=1}^{Q} a_j z^{-j}} \tag{2.39}$$

where a_0 equals 1 in most IIR filter designs.

The main advantage of IIR filters is efficiency in implementation. In order to meet specifications such as passband, stop band and ripple, IIR filter can have lower order than FIR filter. It implies that IIR filter occupies less chip area.

2.4.3 Proposed Digital Loop Filter

There are many different digital low pass filters used in different ADPLL designs. The passive proportional integral (PPI) filter is widely used in the ADPLL. Fig. 2.47 shows the basic architecture of PPI filter. The z-domain transfer function of the PPI

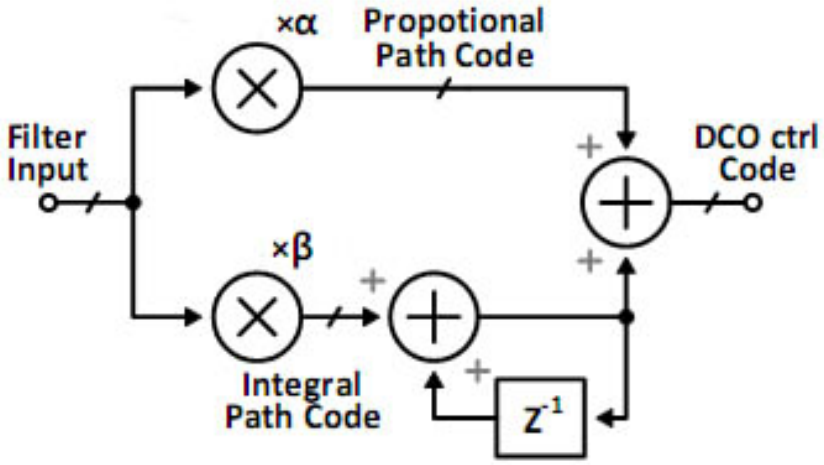


Figure 2.47: The basic architecture of PPI filter.

filter is:

$$H_{DLF}(z) = \alpha + \frac{\beta}{1 - z^{-1}} \tag{2.40}$$

In ref. [28], the author gives a method to calculate coefficients α and β. For the selected damping ratio (ζ), natural frequency (ω_n), the coefficients α and β are:

$$\alpha = \frac{1 - e^{-2\zeta\omega_n T_{ref}}}{G_L} \tag{2.41}$$

$$\beta = \frac{e^{-2\zeta\omega_n T_{ref}} - 2e^{-2\zeta\omega_n T_{ref}} \cos(\omega_n T_{ref}\sqrt{1-\zeta^2}) + 1}{G_L} \tag{2.42}$$

where T_{ref} is reference clock period and G_L is ADPLL close loop gain. In digital system, the limited bit number of coefficient generates the error between the calculated coefficient value and real coefficient used in digital loop filter. However, the coefficient noise can be neglected because the accuracy of loop bandwidth is not highly restricted. Hence, in the approximation method, the DLF coefficients are changed to simple expression in a binary mode. The approximation method uses a shift register

instead of a multiplier. The coefficients for both proportional and integral paths are calculated as $\alpha = 2^{-3}$ and $\beta = 2^{-5}$ by the equations of the relation loop parameters. The proportional and integral block for proposed DLF are presented in Fig. 2.48 and Fig. 2.49.

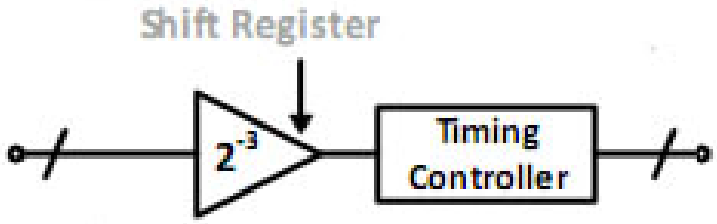


Figure 2.48: The basic architecture of proportional path.

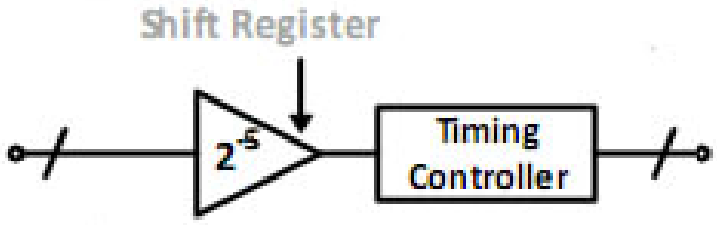


Figure 2.49: The basic architecture of integral path.

Fig. 2.50 shows the detailed schematic of the proposed digital loop filter. The detailed half adder and full adder circuits are shown in Fig. 2.51 and Fig. 2.52.

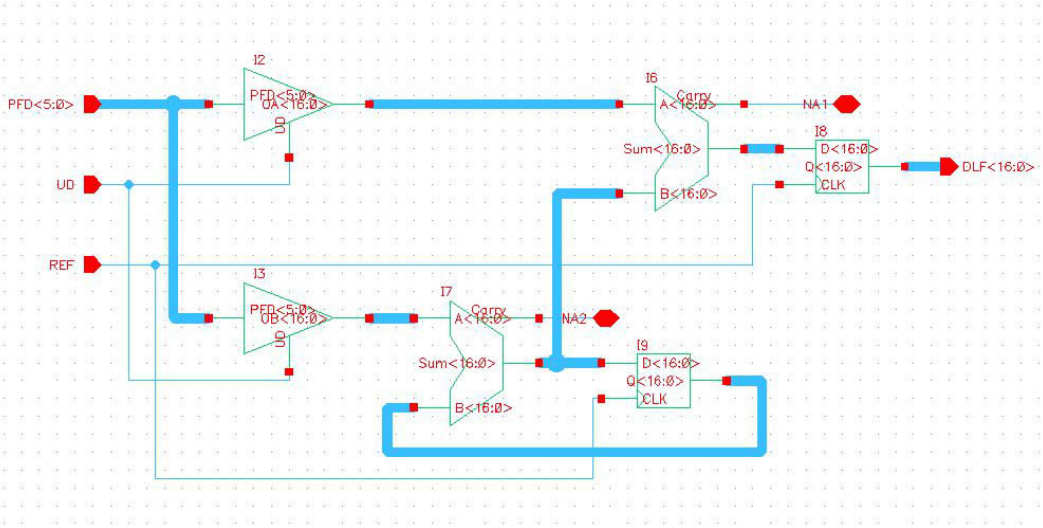


Figure 2.50: The behavior model of proposed digital loop filter.

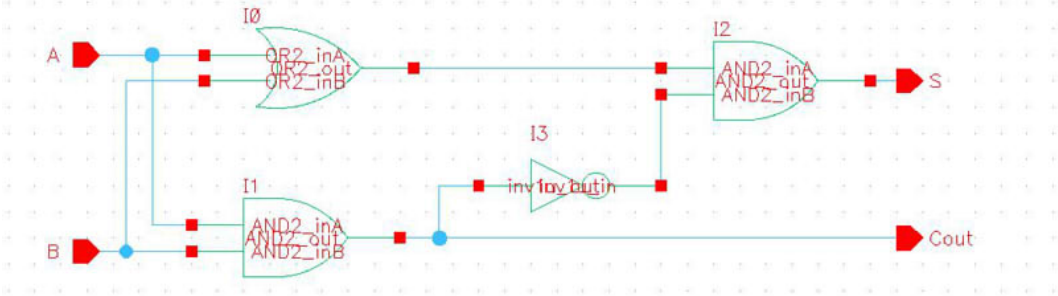


Figure 2.51: Half adder.

2.5 Fractional Divider with Delta Sigma Modulator

In modern wireless transceivers operating at radio frequencies, the frequency divider chain of the PLL frequency synthesizer is one of the most critical function blocks. The issue of actual implementation of the frequency divider at several GHz is a non trivial one. Traditionally, different approaches of divider can be used depending on

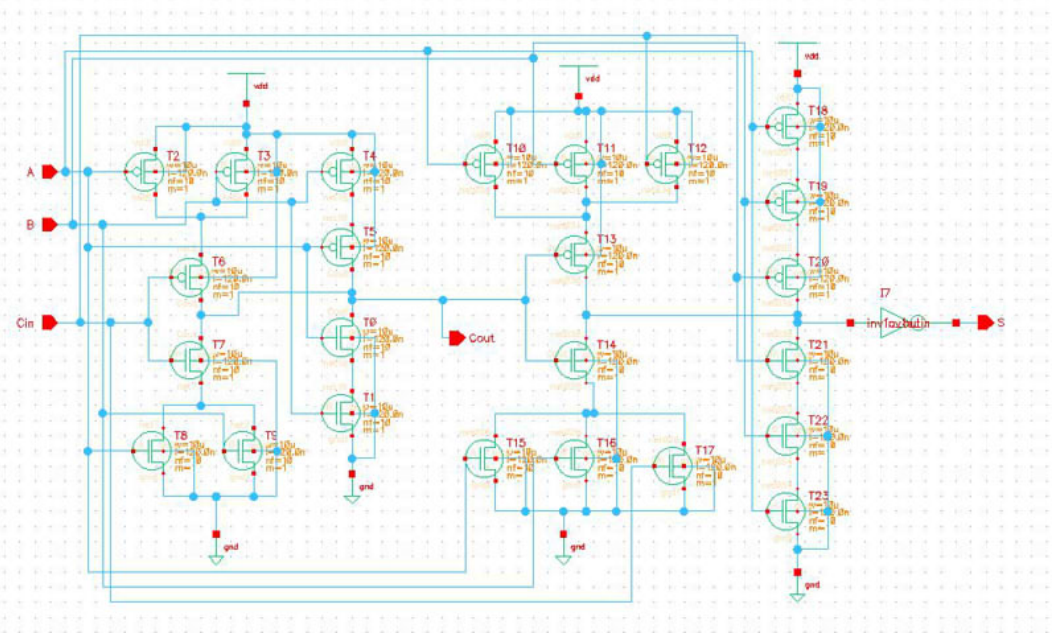


Figure 2.52: Full adder.

the PLL output frequency. The simplest way to implement a clock frequency division is to design a digital counter, with a digital logic resetting the counter after a number of input clock cycles equal to the division ratio have been counted.

2.5.1 Delta Sigma Modulator

In the North American wireless system, frequency resolution/step size is 30 kHz. In China, Japan and the Far East it is 25 kHz. In Europe, the system requires a 200 kHz step. In our project, the step size is designed less than 5KHz.

In traditional integer PLL, the frequency resolution FR equals the desired output frequency, f_{out}, divided by integer divider number N_I.

$$FR = \frac{f_{out}}{N_I} = f_{ref} \tag{2.43}$$

In such a system, in order to meet frequency resolution requirement, the reference signal should be $25KHz$ and divider number N_I equals 2000000 when output frequency is $5GHz$. An unavoidable occurrence in such integer PLL synthesis is that frequency multiplication, raises the signal's phase noise by $20\log(N_I) = 20\log(2000000) = 126dB$. It would seem that we could radically reduce the close-in phase noise by reducing the value of N_I but unfortunately the channel spacing of an integer PLL is dependent on the value of N_I. Due to this dependence, the phase detectors typically operate at a frequency equal to the channel spacing of the communication system. Compare to integer PLLs, fractional PLLs allow better resolution and performance by allowing the integer counter to support fractional values. In our design, the reference clock signal equals $13MHz$. The integer part range is $67 \sim 450$ which raises the signal's phase noise approximately $36 \sim 53dB$. The bit number of the fractional part of divider number is 12 which results in the frequency resolution equals

$$FR = \frac{f_{ref}}{2^{12}} = \frac{13MHz}{2^{12}} \approx 3.2KHz \tag{2.44}$$

Delta sigma modulator is widely used in the fractional PLL. Table 2.11 shows examples of delta sigma modulator in different orders.

Table 2.11: Delta sigma modulator examples in different orders

Modulator order	Range	Sample divider number sequence
First	0, 1	20, 21
Second	-1, 0, 1, 2	19, 20, 21, 22
Third	-3, -2, ..., 3, 4	17, 18, 19, 20, 21, 22, 23, 24
Fourth	-7, -6, ..., 7, 8	13, ..., 19, 20, 21, 22, ..., 28

In our proposed ADPLL, the second order MASH delta sigma modulator is applied

in the divider block. The detail of the second order MASH ΔΣ modulator is shown in Fig. 2.53. The noise caused by delta sigma modulator and fractional spur is analyzed in Chapter 3.

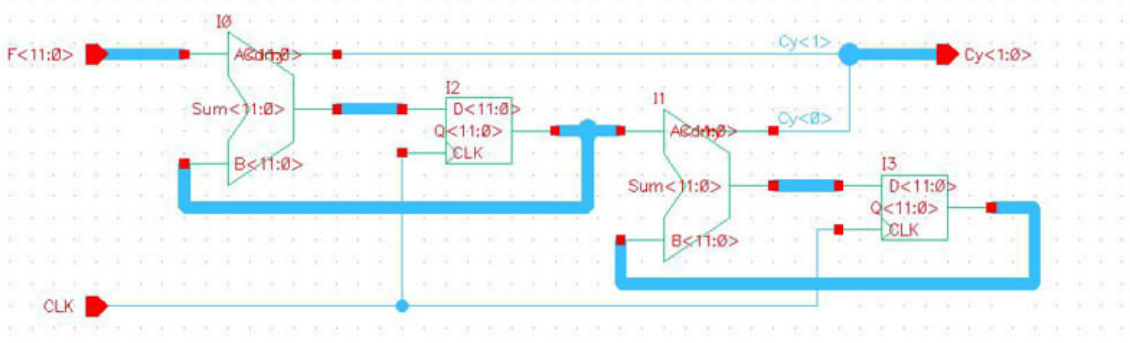


Figure 2.53: Details of the second order MASH ΔΣ modulator.

The high frequency divider by 2 is composed of a single D-flip flop as shown in Fig. 2.54. Fig. 2.55 shows the simulation results of divider by 2 signal. The top

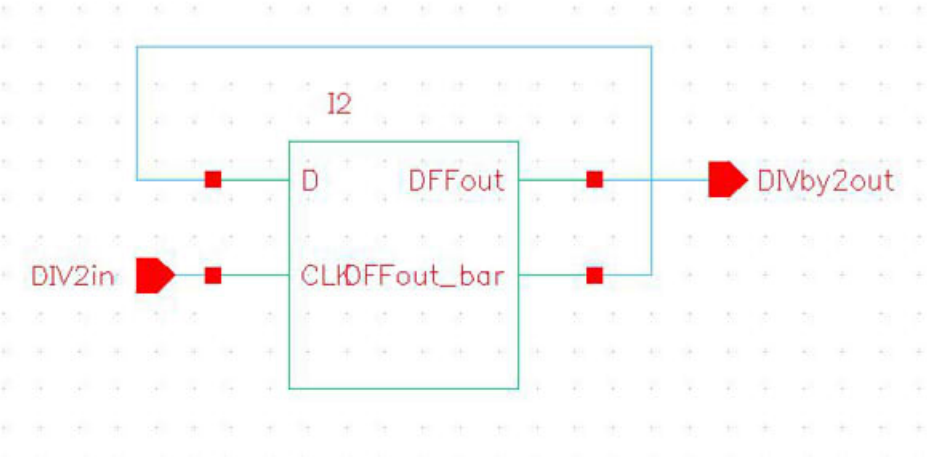


Figure 2.54: High frequency divide by 2.

curve is DCO output with frequency equals 6GHz. The middle curve is output after 3:1 multiplexer. The bottom signal is divider by 2 output signal. It is clear that, a single D-flip flop is applicable for divide by 2 function.

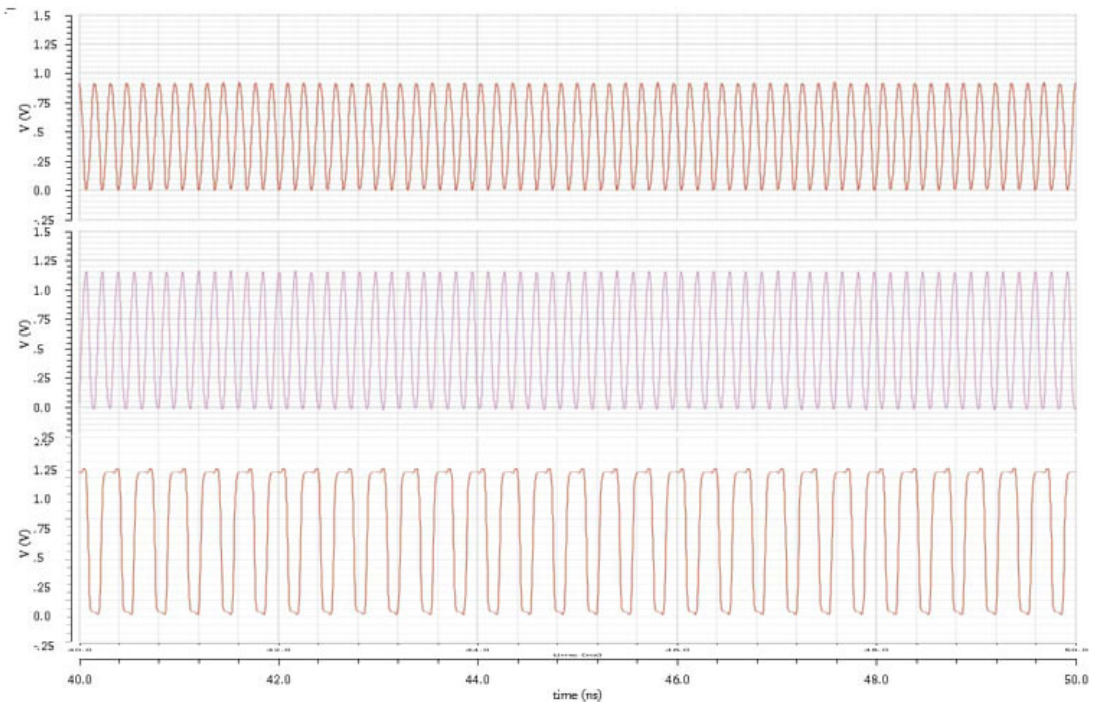


Figure 2.55: DCO output signal divider by 2.

2.5.2 Proposed Frequency divider with Delta Sigma Modulator

Since the reference clock is 13MHz, we can calculate the divider value of PLL. Table 2.12 shows the integer part of divider value of each wireless standard in proposed PLL design.

Fig. 2.60 shows behavior model of proposed programmable fractional divider.

The high frequency divider is a divide-by-2 function block. It is implemented using a simple D-Flip flop with the negative output fed back to the data input. Its maximum operating speed will set the maximum frequency that the D-FF can divide. In order to achieve the maximum possible operating frequency (up to 6GHz

Table 2.12: Integer part of Divider value in proposed PLL.

Standard	Frequency range	Divider Number
GSM	880MHz-960MHz	67-74
UMTS	1.92GHz-2.17GHz	147-167
WiMAX	2.3GHz-2.7GHz	176-208
	3.3GHz-3.8GHz	253-293(126-147)
Bluetooth	2.4GHz-2.48GHz	184-191
WiFi	2.412GHz-2.472GHz	185-191
	5.15GHz-5.35GHz	396-412(198-206)
	5.65GHz-5.85GHz	434-450(217-225)

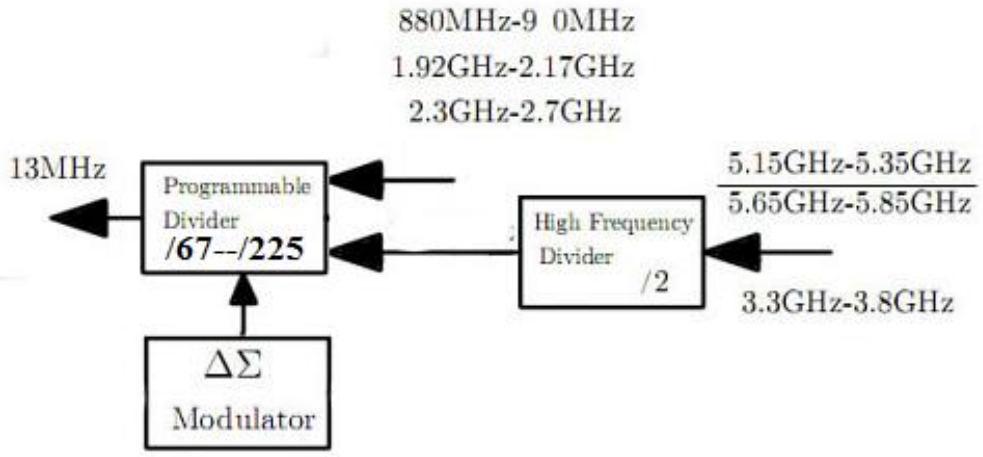


Figure 2.56: Behavior model of proposed programmable divider.

in proposed PLL), it is necessary to select the appropriate implementation for the D-FF.

Compare to static D-FF, dynamic D-FF provides very high speed of operation. For the purposes of this high frequency divider, the selected D-FF topology is shown in Fig. 2.57. The detailed transistors' sizes are listed in Table .

Fig. 2.58 shows the detailed schematic of frequency divider with the second order delta sigma modulator. Fig. 2.59 shows the 8-bit counter applied in the proposed

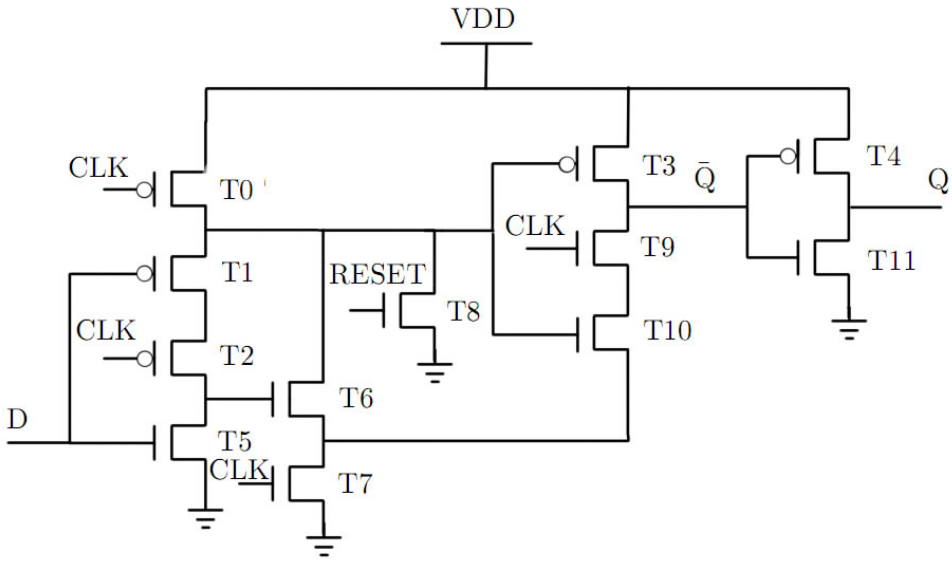


Figure 2.57: D flip flop implement.

frequency divider.

Table 2.13: Transistors' sizes of D flip flop in divide-by-2 block

Transistor	Type	W/L($\mu m/\mu m$)	Transistor	Type	W/L($\mu m/\mu m$)
T0	PMOS	10/0.13	T6	NMOS	5/0.13
T1		10/0.13	T7		5/0.13
T2		10/0.13	T8		10/0.13
T3		10/0.13	T9		5/0.13
T4		10/0.13	T10		5/0.13
T5	NMOS	5/0.13	T11		5/0.13

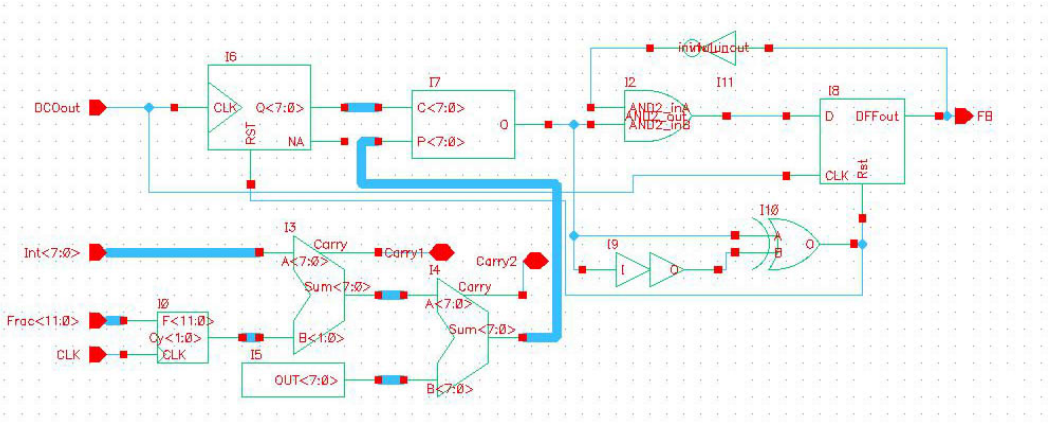


Figure 2.58: Frequency divider.

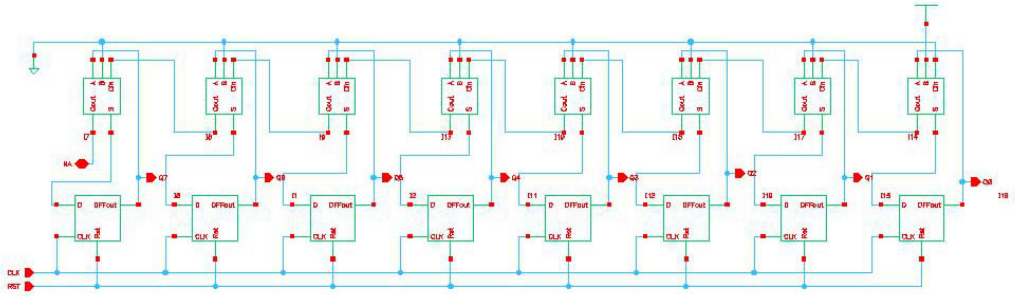


Figure 2.59: 8-bit counter.

2.6 The Proposed Multi-band ADPLL Simulation Results

In previous sections, we have presented each block of proposed ADPLL. In this section, we connect each blocks and provide coder, decoder and connection functional blocks as shown in following. The reference clock signal frequency is 13MHz.

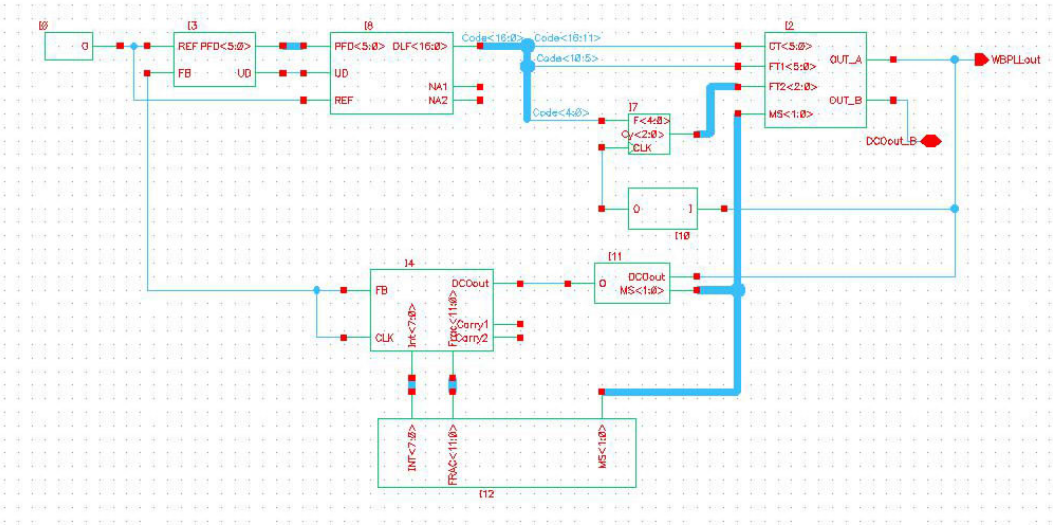


Figure 2.60: Proposed wide band ADPLL.

To further confirm ADPLL behavior, the proposed ADPLL is implemented in IBM8rf 0.13μm Model with 1.2V supply voltage and simulated using Cadence spectre. The simulation results including frequency response, phase noise performance and power consumption are presented at the end of this section which proves the advantages of the proposed ADPLL.

The low pass filter coefficients values are determined by phase noise performance, locking time and damping ratio, which is analyzed in Chapter 3.5. $\alpha = 2^0$ and $\beta = 2^{-5}$

is selected and corresponding bandwidth, damping ratio and locking time values in different output frequencies are listed in Table. 2.14

Table 2.14: Loop bandwidth and damping ratio

Frequency	Bandwidth	Damping ratio	Locking Time
920MHz	63KHz	0.72	$0.6\mu s$
2.4GHz	41KHz	0.99	$0.6\mu s$
3.6GHz	54KHz	0.81	$0.6\mu s$
5.6GHz	46KHz	0.92	$0.6\mu s$

2.6.1 Frequency Response

Fig. 2.61 shows the frequency responses under 1GHz, 2.4GHz, 3.6GHz and 5.6GHz. The lock-in time is less than 13 cycles of the reference signal, equivalently less than $1\mu s$. When ADPLL is locking, the frequency variation is less than 0.05%.

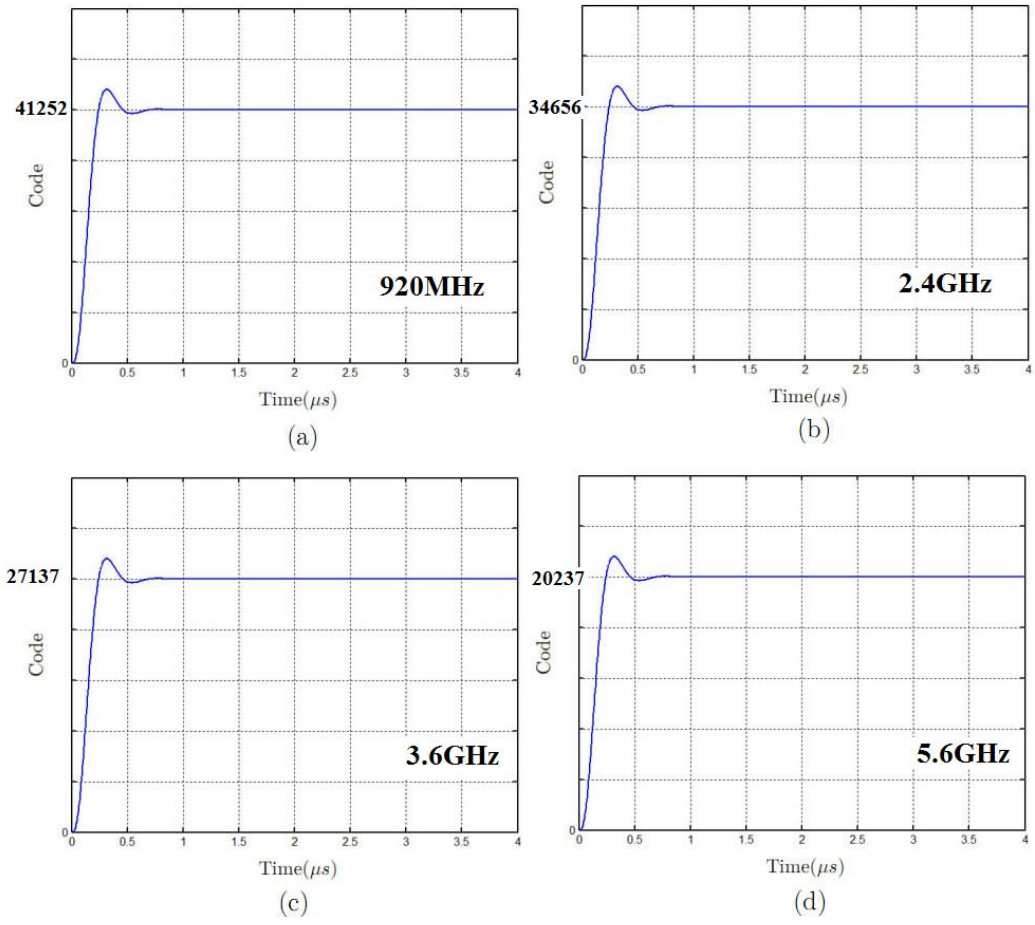


Figure 2.61: Frequency response of (a) 1GHz; (b) 2.4GHz; (c) 3.6GHz; (d) 5.6GHz.

2.6.2 Phase Noise Performance

Phase noise is a specification that characterizes spectral purity. The detailed analysis of ADPLL phase noise performance and noise characterization are presented in next chapter. In this section, the phase noise performance of the proposed wide band ADPLL in each frequency range is analyzed. Fig. 2.62 (a) shows output phase noise performance projected by the method in Chapter 3. Fig. 2.62 (b) shows the transistor level simulation result of ADPLL output phase noise when output frequency is $920MHz$. The phase noise at 1MHz frequency offset is $-117dBc/Hz$. The estimated fractional spur by using equation (3.24) is $-99dBc$ at $10MHz$. The simulation result of fractional spur is $-94dBc$ at $10MHz$. When output frequency ranges from

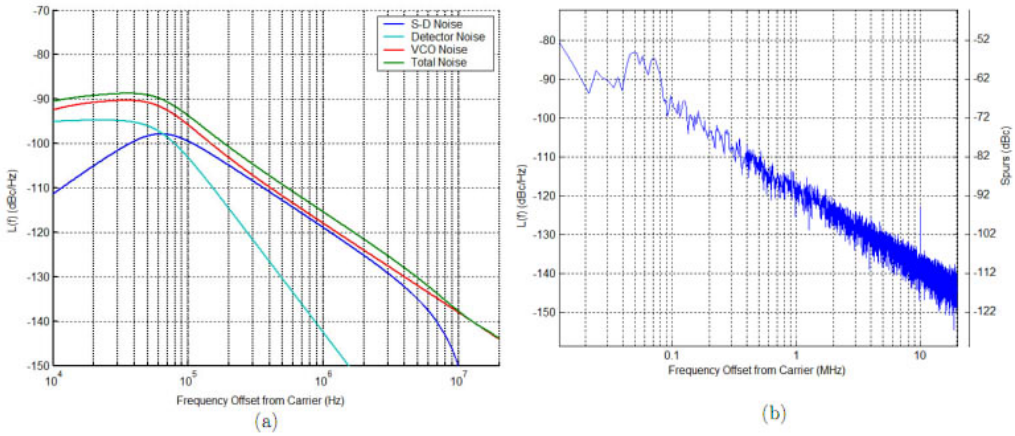


Figure 2.62: ADPLL output phase noise at 920MHz (a) Projected; (b) simulated.

$0.8GHz$ to $1.1GHz$, the loop bandwidth is $50KHz$. The output phase noise at $1MHz$ frequency offset is less than $-116dBc/Hz$.

Fig. 2.63 (a) shows the projected output phase noise performance when output frequency is $2.4GHz$. Fig. 2.63 (a) shows the transistor level simulation result. The phase noise at $1MHz$ frequency offset is $-110.2dBc/Hz$. The estimated fractional

spur is $-102dBc$ at $8MHz$. The simulation result of fractional spur is $-104dBc$ at $8MHz$. When output frequency is from $1.9GHz$ to $2.8GHz$, the loop bandwidth is $50KHz$. The output phase noise at $1MHz$ frequency offset is less than $-110dBc/Hz$.

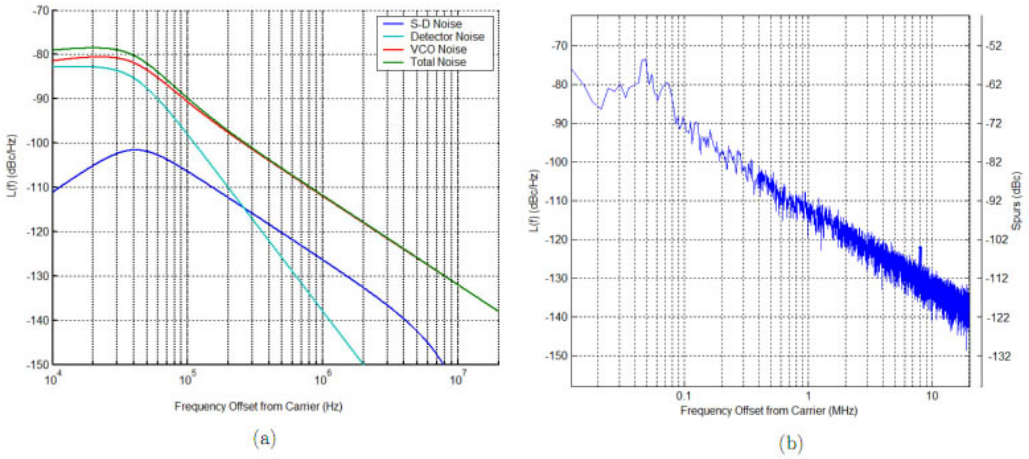


Figure 2.63: ADPLL output phase noise at 2.4GHz (a) Projected; (b) simulated.

Fig. 2.64 (a) and (b) show the projected and simulated output phase noise performance when output frequency is $3.6GHz$. The phase noise at $1MHz$ frequency offset is $-110dBc/Hz$. The estimated fractional spur is $-113dBc$ at $12MHz$. The simulation result of fractional spur is $-115dBc$ at $12MHz$. Once output frequency is from $3.2GHz$ to $4GHz$, the loop bandwidth is $50KHz$. The output phase noise at $1MHz$ frequency offset is less than $-105dBc/Hz$.

Fig. 2.65 (a) and (b) show the projected and simulated output phase noise performance when output frequency is $5.6GHz$. The phase noise at 1MHz frequency offset is $-105.6dBc/Hz$. The estimated fractional spur is $-113dBc$ at $10MHz$. The simulation result of fractional spur is $-115dBc$ at $10MHz$. Once output frequency is from $5GHz$ to $6GHz$, the loop bandwidth is $50KHz$. The output phase noise at

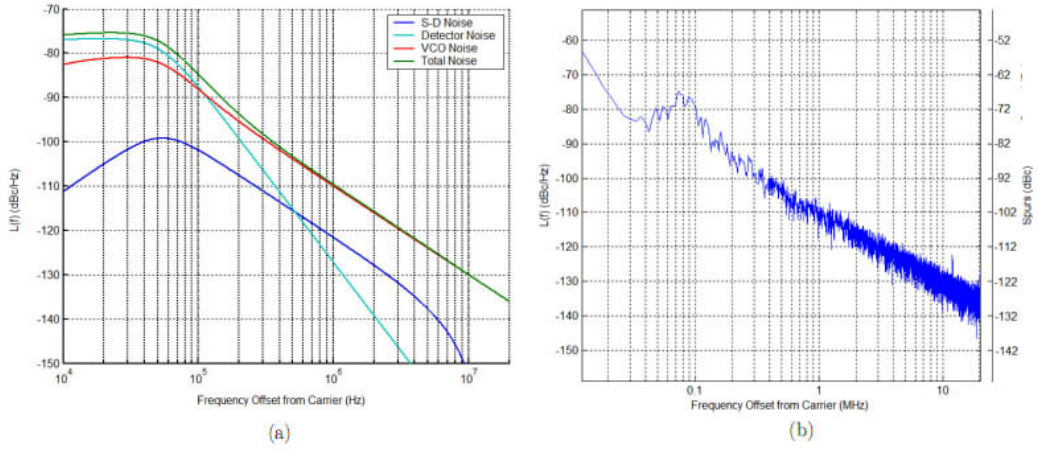


Figure 2.64: ADPLL output phase noise at 3.6GHz (a) Projected; (b) simulated.

$1MHz$ frequency offset is less than $-104dBc/Hz$.

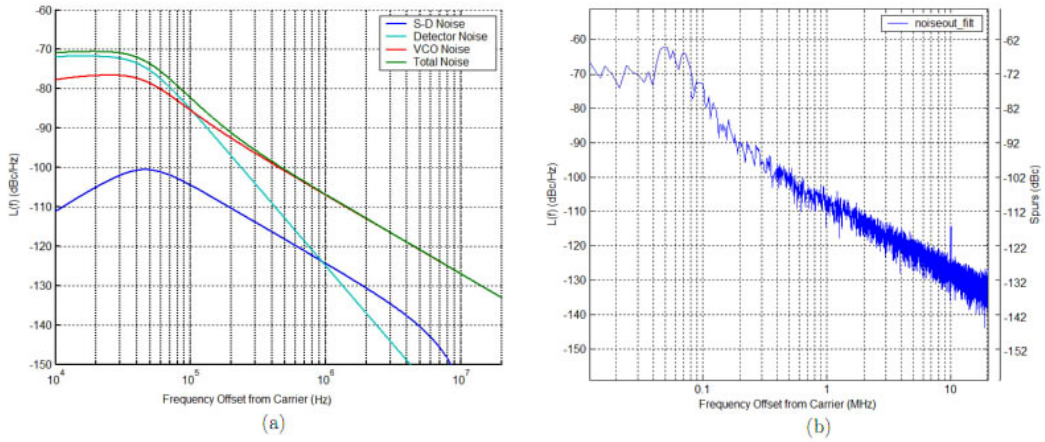


Figure 2.65: ADPLL output phase noise at 5.6GHz (a) Projected; (b) simulated.

2.6.3 Power Consumption

The power consumption is one of the most important issues for VLSI systems. Here we list the power consumption of each block and total power consumption of ADPLL

under different wireless communication standards.

Table 2.15: Multi-band ADPLL power consumption

Standard	Frequency Hz	PFD mW	LPF mW	DIV mW	DCO mW	Total mW
GSM	1G	0.01	4.8	7.3	8.3	20.4
GMTS	2G	0.01	4.9	8	8.9	21.8
WiMAX	2.4G	0.01	4.9	8.4	9.1	22.4
WiMAX	3.6G	0.01	5.1	8.8	10.3	24.2
Bluetooth	2.4G	0.01	4.9	8.4	9.1	22.4
Wi-Fi	2.45G	0.01	4.9	8.4	9.1	22.4
Wi-Fi	5.7G	0.01	5.3	9.6	12.2	27.1

2.7 Conclusion

In this chapter, we first review quad-mode digitally controlled oscillator based on switched frequency resonant LC-tank DCO. Then, we extend our previous design and present a multi-mode digitally controlled LC-tank DCO. Based on the proposed DCO, a multi-band ADPLL is developed which is able to operate from $800MHz$ to $1.1GHz$, $1.9GHz$ to $2.8GHz$, $3.2GHz$ to $4GHz$ and $5GHz$ to $6GHz$. Next, we analyze different kinds of time to digital converter and propose a vernier TDC based phase frequency detector. Then, PLL function blocks including digital low pass filter and delta sigma modulator fractional frequency divider are developed. Finally, the proposed wide band ADPLL is built and simulated in Cadence SpectreRF simulation program. The results show that the proposed ADPLL is applicable for wireless communication including GSM, UMTS, GPS, WiMAX and Wi-Fi.

Chapter 3

ADPLL Design Parameters Determinations through Noise Modeling

Phase noise is one of the key characteristics of ADPLL frequency synthesizer. In the literature, there are numerous studies analyzing analog PLLs [29-34], [68]. However, due to its unique structure, there are new variables in ADPLL that must be carefully configured in order to meet performance requirements. These variables include TDC resolution, bit-width of each digital unit, DLF coefficients and digitally controlled oscillator (DCO) resolution. In ADPLL, besides oscillator noise, quantization noise is the other major noise source that may deteriorate performance on account of the limited bit-width of different digital units. Therefore, the traditional analog PLL's analytic model cannot be applied directly for ADPLL design characterizations. Currently, there are limited researches dedicated to analyzing the relationship between performance and variables that are applicable to both architectures of ADPLLs. In [35], TDC noise and DCO noise are analyzed for architecture-II ADPLL. In [36], authors present a z-domain model of architecture-II ADPLL without noise analy-

sis. In [37], authors develop a z-domain model for analyzing DCO noise and TDC noise for architecture-II ADPLL. In [38]-[40], authors develop time domain models for ADPLL. In [41], authors propose a linear discrete time multi-rate model for ADPLL. However, the study of circuit variables and performance that are applicable for both architecture-I and architecture-II ADPLLs is lacking. Moreover, those analytical models study ADPLL noise only and do not determine variables based on performance specifications.

3.1 Definition of Phase Noise

Phase noise is a specification that characterizes spectral purity. For example, in the frequency domain, an oscillator output should ideally be a pure sinusoid represented as a vertical line. However, in reality, there are noise sources in the oscillator that can cause the output frequency to deviate from its ideal position. As shown in Fig. 3.1, phase noise is usually specified as the ratio of a noise power at an offset frequency away from the carrier to the carrier power, in a 1Hz bandwidth.

3.2 Phase Noise Model

In order to evaluate phase noise, fractional spur and locking time, an ADPLL behavior model is developed. The s-domain model as shown in Fig. 3.2 and the z-domain of architecture-II ADPLL are discussed in [35] and [37], respectively.

In this paper, comprehensive analytical models of both architecture-I and architecture-II ADPLL are presented as shown in Fig. 3.3. Type-I ADPLL as shown in Fig. 3.3

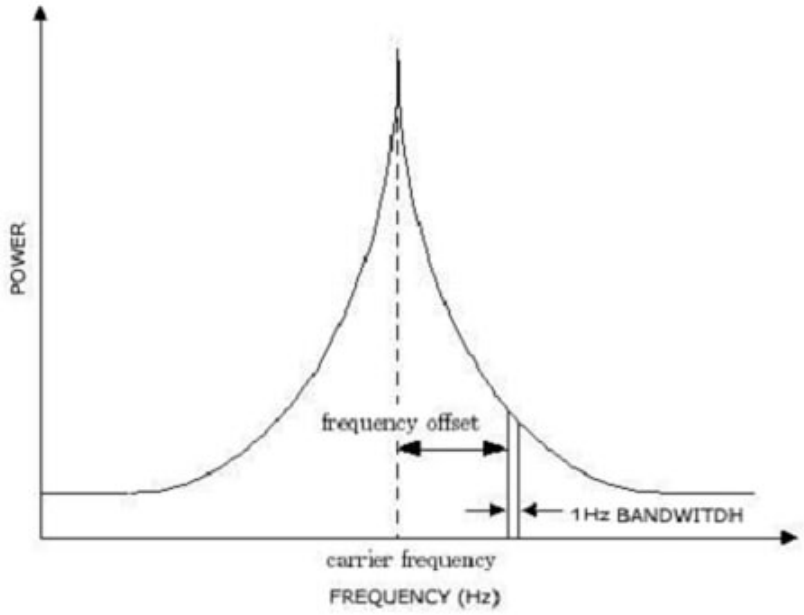


Figure 3.1: Phase noise definition.

(a) replaces the standard phase frequency detector (PFD), charge pump with time-to-digital converter (TDC) which quantizes the phase difference between reference clock and the feedback signal. The resulted phase difference is fed into the digital loop filter (DLF) instead of the analog filter to eliminate high frequency noise. And a programmable divider with $\Delta\Sigma$ modulator is used in the feedback path to set up the fractional divide ratio. In the type-II ADPLL, as shown in Fig. Fig. 3.3 (b), the phase detection (PD) is performed by the combination of a counter and the TDC which compares the difference between reference clock frequency control word and the feedback frequency control word without a frequency divider in the feedback loop.

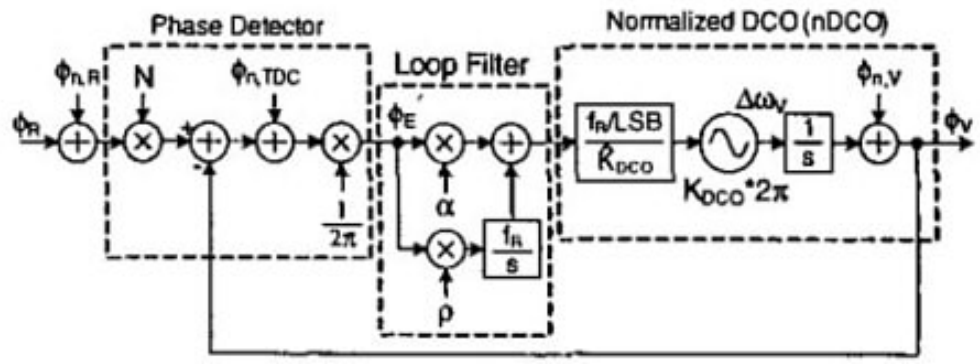


Figure 3.2: The s-domain phase noise model of phase-domain ADPLL [35].

3.2.1 Noise Sources in An ADPLL

For both architectures of ADPLLs, there are five dominant noise sources, which are input reference clock noise η_{ref} , TDC noise η_{TDC}, DCO oscillator noise η_{DCO}, DCO quantization noise η_{DQN} and delta sigma modulator noise η_{DSM}. Generally, dominant noise of ADPLLs can be categorized as oscillator noise or quantization noise. Reference signal noise is oscillator noise. DCO noise is composed of oscillator noise and quantization noise. TDC quantization noise caused by TDC resolution is the dominant noise source in PFD block. DCO quantization noise is caused by the limited bit number of DCO control number. $\Delta\Sigma$ modulator also generates quantization noise due to limited output bit number. Table 3.1 summarizes the noise sources and their types.

ADPLL phase noise spectrum density equals the summation of noise spectrum density caused by each noise source at the output node.

$$S_{out}(f) = \Sigma S_i(f) \cdot |H_i(z)|^2 \tag{3.1}$$

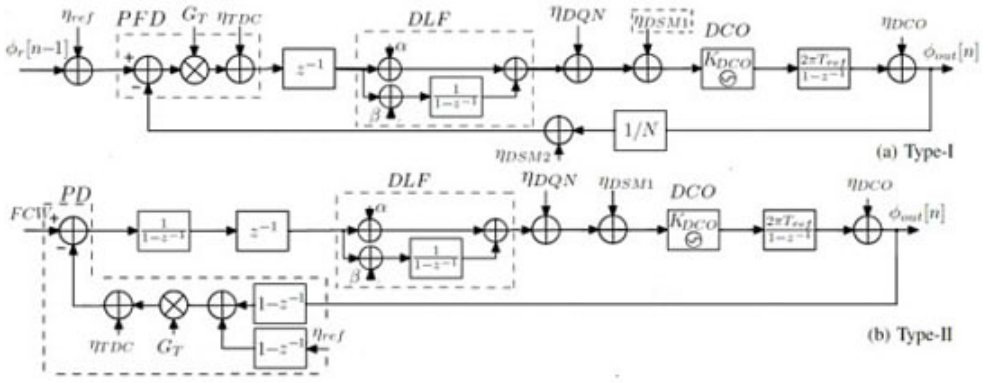


Figure 3.3: The z-domain phase noise model of phase-domain ADPLL. (a) Architecture-I ADPLL; (b) Architecture-II ADPLL.

Table 3.1: Noise sources of ADPLLs

Noise Source	Noise Type
Reference signal	Oscillator noise
PFD-TDC	Quantization noise
DQN	Quantization noise
DCO	Oscillator noise
$\Delta\Sigma$ modulator	Quantization noise

where $S_i(f)$ and $H_i(z)$ represent the noise spectrum density and the corresponding closed loop transfer function of each noise source in ADPLL.

3.2.2 ADPLL Noise Transfer Function

The second-order block diagram of an ADPLL is shown in Fig. 3.3. The transfer function of the reference signal features a low pass filter characteristic in ADPLL. For noise frequencies below the loop bandwidth, reference signal noise causes a significant effect on the total phase noise. For noise frequencies above the loop bandwidth, the

effect of reference signal noise is attenuated. The phase domain transfer function of the reference signal noise is:

$$H_{\eta_{REF}}(z)_I = \frac{NG_I(\alpha z + \beta z - \alpha)}{(z-1)^2 + G_I(\alpha z + \beta z - \alpha)} \tag{3.2}$$

$$H_{\eta_{REF}}(z)_{II} = \frac{NG_{II}(\alpha z + \beta z - \alpha)}{(z-1)^2 + G_{II}(\alpha z + \beta z - \alpha)} \tag{3.3}$$

where $G_I = 2\pi G_T K_{DCO} T_{ref}/N$, $G_{II} = 2\pi G_T K_{DCO} T_{ref}/FCW$ and T_{ref} is reference clock period. G_T is TDC gain.

Fig. 3.3 shows that the transfer function of the DCO features a high pass filter characteristic. The phase domain transfer function of the DCO noise of each architecture ADPLL can be expressed as:

$$H_{\eta_{DCO}}(z)_I = \frac{(z-1)^2}{(z-1)^2 + G_I(\alpha z + \beta z - \alpha)} \tag{3.4}$$

$$H_{\eta_{DCO}}(z)_{II} = \frac{(z-1)^2}{(z-1)^2 + G_{II}(\alpha z + \beta z - \alpha)} \tag{3.5}$$

In architecture-I ADPLL, the transfer function of the DCO quantization noise in phase domain equals:

$$H_{\eta_{DQN}}(z)_I = \frac{NG_I(z-1)/G_T}{(z-1)^2 + G_I(\alpha z + \beta z - \alpha)} \tag{3.6}$$

In architecture-II ADPLL, it can be deduced as:

$$H_{\eta_{DQN}}(z)_{II} = \frac{NG_{II}(z-1)/G_T}{(z-1)^2 + G_{II}(\alpha z + \beta z - \alpha)} \tag{3.7}$$

Here, the phase domain transfer function of the TDC noise in architecture-I ADPLL is derived as:

$$H_{\eta_{TDC}}(z)_I = \frac{NG_I(\alpha z + \beta z - \alpha)/G_T}{(z-1)^2 + G_I(\alpha z + \beta z - \alpha)} \tag{3.8}$$

In architecture-II ADPLL, it is:

$$H_{\eta_{TDC}}(z)_{II} = -\frac{G_{II}(\alpha z + \beta z - \alpha)/G_T}{(z-1)^2 + G_{II}(\alpha z + \beta z - \alpha)} \tag{3.9}$$

$\Delta\Sigma$ modulator can be deployed not only in divider function block but also in DCO block. When used in divider block, it generates fractional part of divider value. When used in DCO input, it is to dither fine tuning of DCO. The transfer functions of digital $\Delta\Sigma$ modulator in DCO block $H_{\eta_{DSM_1}}(z)_I$ and in divider block $H_{\eta_{DSM_2}}(z)_I$ in architecture-I ADPLL are shown below:

$$H_{\eta_{DSM_1}}(z)_I = \frac{2^{-N_{\Delta\Sigma}} \cdot N \cdot G_I(z-1)/G_T}{(z-1)^2 + G_I(\alpha z + \beta z - \alpha)} \tag{3.10}$$

$$H_{\eta_{DSM_2}}(z)_I = \frac{2^{-N_{\Delta\Sigma}}/N}{(z-1)^2 + G_I(\alpha z + \beta z - \alpha)} \tag{3.11}$$

Since there is no divider function block in architecture-II ADPLL, the transfer function of $\Delta\Sigma$ modulator in DCO block is:

$$H_{\eta_{DSM_1}}(z)_{II} = \frac{2^{-N_{\Delta\Sigma}} K_{DCO}(z-1)}{(z-1)^2 + G_{II}(\alpha z + \beta z - \alpha)} \tag{3.12}$$

where $N_{\Delta\Sigma}$ is the input bit number of $\Delta\Sigma$ modulator.

3.3 Output Phase Noise of ADPLL

3.3.1 Phase Noise due to Input Reference Signal

Reference clock noise is categorized as oscillator noise. The phase noise spectrum density of the reference clock $S_{REF}(f)$ can be described using oscillator phase noise model. According to the prototype of phase noise spectrum density as shown in Fig. 3.4, the oscillator phase noise spectrum density can be expressed as [59]:

$$S_{OSC}(f) = k_0 + \frac{k_1}{f} + \frac{k_2}{f^2} + \frac{k_3}{f^3} \tag{3.13}$$

where k_0, $\frac{k_1}{f}$, $\frac{k_2}{f^2}$ and $\frac{k_3}{f^3}$ represent thermal noise, flicker noise, carrier noise and intermodulation of both carrier and transistor noise, respectively.

As an example, a typical 13MHz crystal oscillator phase noise curve is shown in Fig. 3.5. The noise floor of such reference signal is -150dBc/Hz. The flicker corner of buffer stage is around 5KHz and flicker corner of oscillator transistor is around 12Hz. According to Eq. (3.13), the phase noise of such reference signal can be expressed by Eq. (3.14) by performing curve fitting.

$$S_{REF}(f) = 10^{-15.0} + \frac{10^{-11.2}}{f} + \frac{10^{-9.4}}{f^2} + \frac{10^{-7.0}}{f^3} \tag{3.14}$$

Another example of phase noise spectrum of $12.3MHz$ reference signal can be

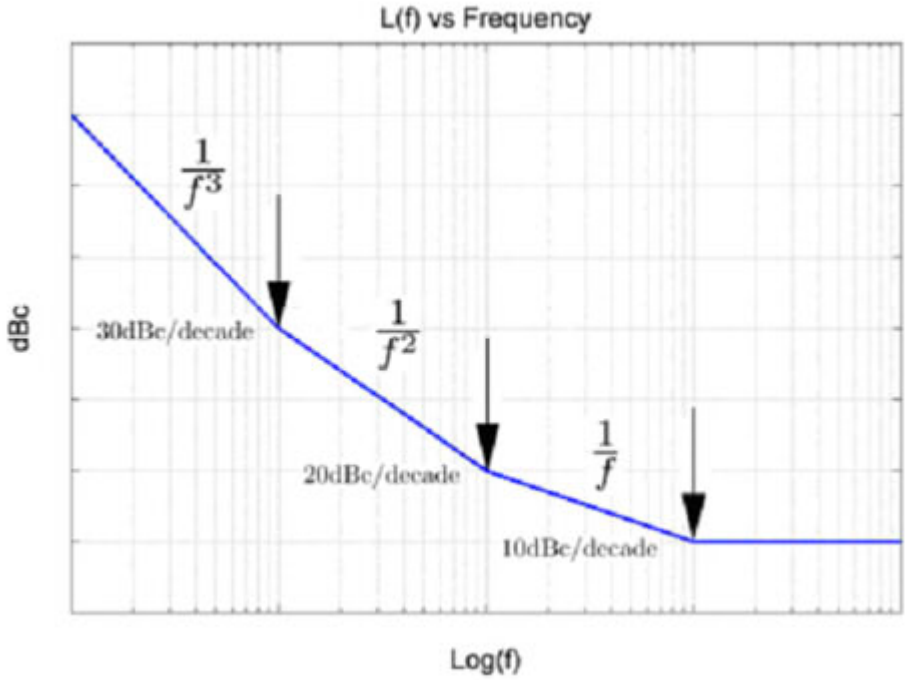
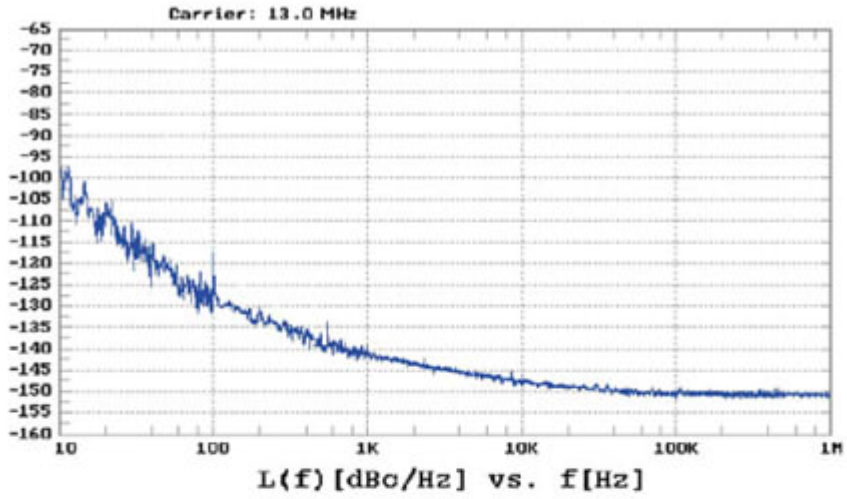


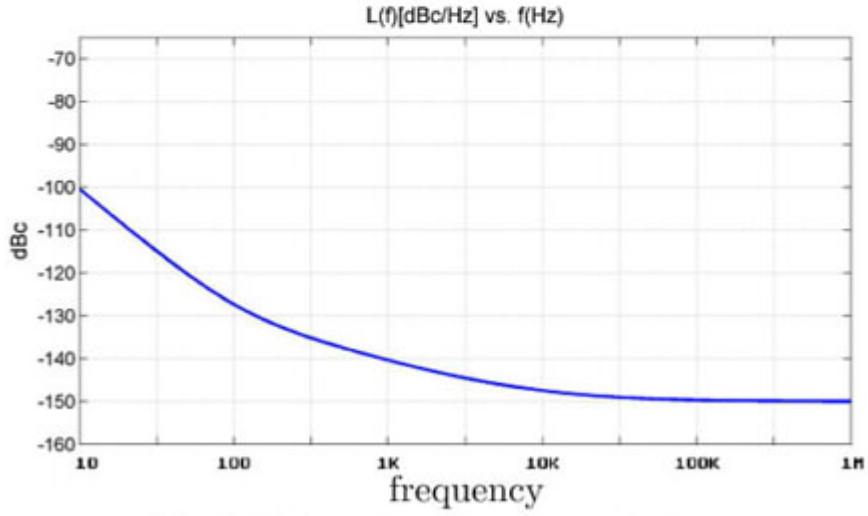
Figure 3.4: Prototype of oscillator phase noise spectrum density.

derived as [66]:

$$S_{REF}(f) = 10^{-15.0} + \frac{10^{-11.1}}{f} + \frac{10^{-9.8}}{f^2} + \frac{10^{-7.1}}{f^3} \tag{3.15}$$



(a) Measured 13MHz reference signal phase noise



(b) Modeled 13MHz reference signal phase noise

Figure 3.5: Phase noise of 13MHz crystal oscillator. (a)Measured, (b)Modeled.

3.3.2 Phase Noise due to DCO

DCO noise is composed of oscillator and quantization noise [42]-[43]. The DCO oscillator noise $S_{DCO}(f)$ can be described using Eq. (3.13). An example of phase noise spectrum of $2.45GHz$ DCO noise can be derived as [67]:

$$S_{DCO}(f) = 10^{-15} + \frac{10^{-0.7}}{f^2} + \frac{10^{4.7}}{f^3} \tag{3.16}$$

The DCO quantization noise is determined by the bit number of DCO control code. When DCO dithering resolution bit is N_{DCO}, DCO quantization noise σ^2_{DQN} equals $\frac{(2^{-NDCO})^2}{12}$. The DCO quantization noise power spectrum density(PSD) equals:

$$S_{DQN} = \frac{1}{12 \cdot (2^{N_{DCO}})^2 \cdot f_r} \tag{3.17}$$

where f_r is reference signal frequency and K_{DCO} is DCO gain.

3.3.3 Phase Noise due to Delta Sigma Modulator

The multi-stage noise shaping (MASH) $\Delta\Sigma$ modulator is widely used in fractional-N ADPLLs. The quantization noise contributed by $\Delta\Sigma$ modulator is treated as an additive noise source [44]. For a $\Delta\Sigma$ modulator with transfer function $H_{\Delta\Sigma}(z)$, the power spectrum density (PSD) of phase fluctuations equals $\frac{\pi^2|H_{\Delta\Sigma}|}{3f_{s\Delta}|1-z^{-1}|^2}$. $f_{s\Delta}$ represents the sampling frequency of $\Delta\Sigma$ modulator. The z-domain transfer function of the m^{th}-order MASH $\Delta\Sigma$ modulator is $H_{\Delta\Sigma}(z) = (1-z^{-1})^m$. The quantization noise spectrum

density of an m^{th}-order MASH $\Delta\Sigma$ modulator is deduced as [44]:

$$S_{DSM} = \frac{\pi^2[2sin(\frac{\pi f}{f_{s\Delta}})]^{2(m-1)}}{3f_{s\Delta}} \tag{3.18}$$

3.3.4 Phase Noise due to TDC-PFD

Time to digital converter (TDC) quantization noise is the dominant noise source in phase detector. Fig. 3.6 shows a widely used N-bit TDC, where the input phase error goes through a delay chain. Flip-flops are connected to the outputs of inverters and sample the state of the delay chain. By using an edge detector and an encode function block, the phase difference is encoded to digital codes.

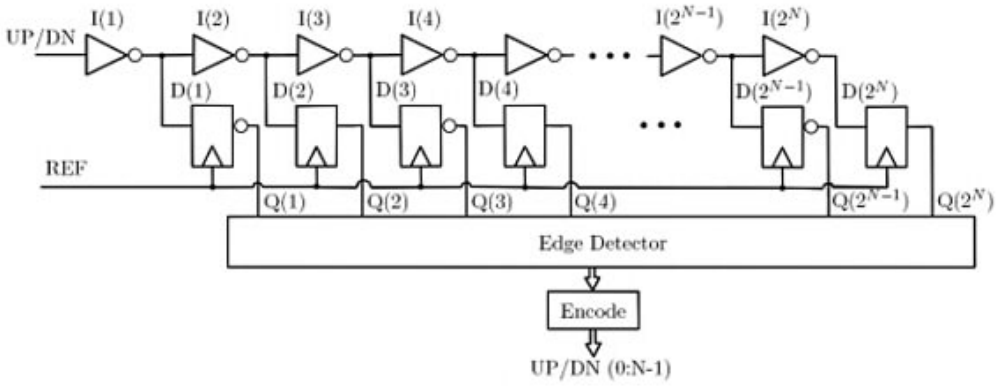


Figure 3.6: Traditional N-bit TDC in ADPLL.

TDC's resolution equals single inverter delay τ_T . For an ADPLL, the gain of TDC G_T is the ratio of TDC output code and input phase difference between the reference clock and the feedback signal.

$$G_T = \frac{1}{2\pi\tau_T f_r} \tag{3.19}$$

where f_r is the reference signal frequency. For a B_T-bit TDC, the maximum phase detection range is:

$$P_{max} = \frac{2^{B_T} - 1}{G_T} \tag{3.20}$$

The standard deviation of TDC quantization noise σ^2_{TDC} equals $\frac{\tau_T^2}{12}$. Normalizing τ_{TDC} by ADPLL output signal period T_{out} and converting it to phase in radians results in $\sigma_{TDC_\phi} = \frac{2\pi\sigma_{TDC}}{T_{out}}$. Thus the TDC quantization noise PSD equals [35]:

$$S_{TDC} = \frac{\sigma^2_{TDC_\phi}}{f_r} = \frac{(\pi\tau_T f_{out})^2}{3f_r} \tag{3.21}$$

where f_{out} is ADPLL output signal frequency. From this equation, it is clear that S_{TDC} can be reduced by increasing TDC resolution with a smaller τ_T. For instance, for given f_r and f_{out} to f_r ratio N, reducing τ_T by half will reduce S_{TDC} by 3dB.

3.4 ADPLL Fractional Spur

Spurious level of ADPLL output is another important parameter. In fractional-N ADPLL, one important spur specification is the fractional spur. The digital $\Delta\Sigma$ modulator is the fundamental source of fractional spurious tones in fractional-N ADPLL while the limited TDC output bit-width can further increase fractional spur level. For an m^{th}-order MASH DSM applied in ADPLL, in locking state, the phase difference varies from $-\frac{(2^m-2)\pi}{N}$ to $\frac{2^m\pi}{N}$. In order to minimize the fractional spur level, TDC output should cover the phase difference range. Thus, we need to ensure:

$$P_{max} \geq \frac{(2^{m+1}-2)\pi}{N} \Rightarrow 2^{B_T} - 1 \geq \frac{2^m - 1}{N \cdot f_r \cdot \tau_T} \tag{3.22}$$

The primary frequency of the fractional spur is $f_{spur} = F_{frac} \cdot f_{ref}$. F_{frac} represents the fractional part of divider value N. The PSD of fractional spur in dB can be deduced as [45]:

$$S_{spur} = 20log(\frac{\Delta f_{max}}{2f_{spur}}) + rolloff(f_{spur}) = 20log(\frac{\Delta C_{max} \cdot K_{DCO}}{2f_{spur}}) + rolloff(f_{spur}) \tag{3.23}$$

where ΔC_{max} is the maximum control code variation and $rolloff(f)$ is defined as the magnitude of the closed loop transfer equation of reference signal subtracting the factor of $20 \log N$.

$$rolloff(f) = 20 \log(|H_{cl}(z)|_{z=e^{j2\pi f T_r}}) - 20 \log N \tag{3.24}$$

It is clear that $\Delta Cmax = 2m - 1$ when $\Delta\Sigma$ modulator is in DCO function block. For ADPLL with $\Delta\Sigma$ modulator only existing in divider block, the maximum phase variation ϕ_{dr} in locking state equals $\frac{2\pi(2^m-1)f_r}{f_{out}}$. Thus ΔC_{max} equals:

$$\Delta C_{max} = \phi_{dr} G_T(\alpha + \beta) = \frac{2\pi f_r G_T(\alpha + \beta)(2^m - 1)}{f_{out}} \tag{3.25}$$

In order to verify equation (3.23), we compare the fractional spur PSD calculated from equation (3.23) and from measurement in [46]. The ADPLL in [46] is an architecture-I APDLL with a 1^{st} order and a 2^{nd} order modulators in divider and DCO block, respectively. The parameters are listed as follows: reference signal frequency equals 40MHz, TDC resolution $\tau_T = 1ps$, DCO gain $K_{DCO} = KHz/LSB$, divider value $N = 90.27195$, DLF coefficients $\alpha = 2^{-3}$ and $\beta = 2^{-6}$. The bandwidth of the PLL is 500KHz. Fractional spur measurement from [46] is shown in Fig. 3.7.

In this paper, we only focus on the primary fractional spur. The primary fractional spur occurs at frequency $f_{spur} = F_{frac} \cdot f_r = 0.27195 \times 40MHz = 10.88MHz$. The fractional spur PSD from measurement equals -118dBc while from direct calculation equals -118.7dBc. The difference is 0.7dB.

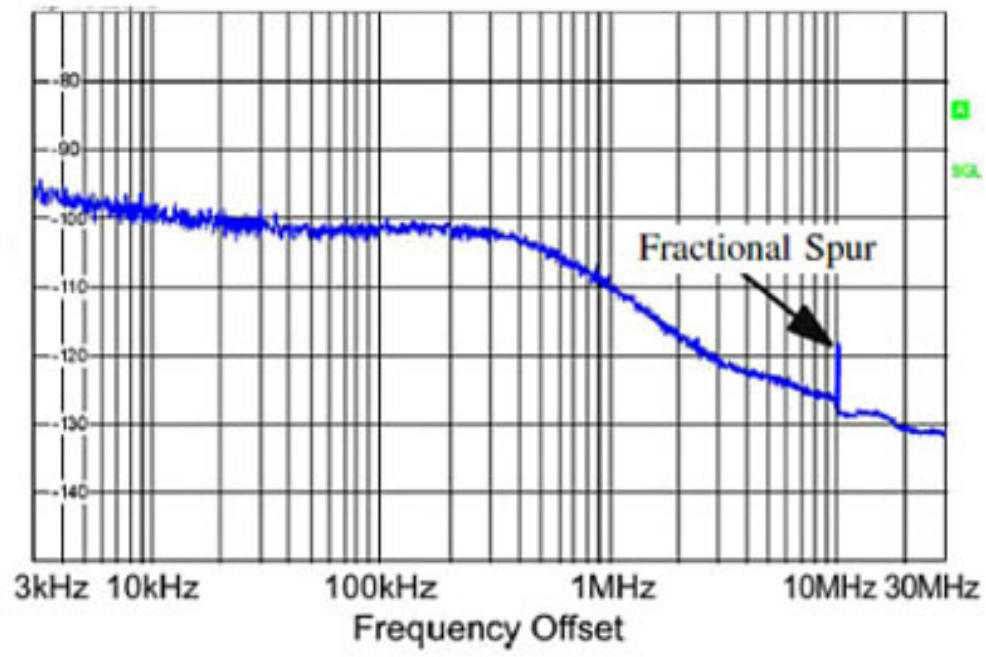


Figure 3.7: Fractional spur of ADPLL in [46].

To make further validations, we also choose ADPLL circuits from [47] and [48]. In [47], an architecture-I ADPLL with the 1^{st} order $\Delta\Sigma$ modulator in DCO is designed. In [48], an architecture-II ADPLL with the order 2^{nd} modulator in DCO is presented. The simulation and measurement results are summarized in Table 3.2. The maximum fractional spur PSD difference between simulation projection and measurement is less than 7.7dB.

Table 3.2: Output phase noise differences between simulation and measurement at some typical frequencies.

		Fractional Spur		Output Frequency (GHz)	ADPLL Arch. Arch.
		Prim. Freq. (Hz)	Prim. PSD (dBc)		
REF [46]	Simu.	10.88M	-118.7	3.610878	I
	Meas.	10M	-118		
REF [47]	Simu.	15M	-145.8	1.5	I
	Meas.	15M	-42.5		
REF [48]	Simu.	21.8K	-52.7	5.37602183	II
	Meas.	20K	-45		

3.5 ADPLL Design with Adjustable Loop Parameters

DLF coefficients are important parameters in ADPLL design. In the literature, there have been considerable researches investigating ADPLL DLF designs [49]-[50]. Nearly all of them focus on accomplishing bandwidth and stability. None have included phase noise into design considerations. In reference [64], author presents a method to calculate the DLF variables in architecture-II ADPLL according to ADPLL bandwidth and phase margin. In reference [65], ADPLL DLF variables are determined by given loop bandwidth and damping ratio. However, no design procedure for determining ADPLL variables based on given required phase noise performance and locking time is provided in these publications. In this paper, an analytic approach will be developed to determine DLF coefficients by considering more comprehensive performance constraints, basically adding phase noise into design consideration in conjunction with bandwidth and loop stability.

Equation (3.2) and equation (3.3) are z-domain closed loop transfer functions of architecture-I and architecture-II ADPLLs. Although a discrete-time system is naturally described by z-transform, it is common to approximate it with a linear continuous-time system and describe it in the s-domain when PLL bandwidth is much lower than the sampling frequency. In ADPLL design, the sampling frequency equals the reference signal frequency. ADPLL bandwidth f_{BW} is usually 10 times lower than the reference signal frequency to ensure system stability with a sufficient phase margin. Therefore, within the bandwidth when $f_{BW} \ll f_r$, the bilinear transform can be applied [37]:

$$z = e^{sT_r} \approx 1 + sT_r \tag{3.26}$$

Thus, s-domain closed loop transfer functions of reference signal can be approximated as:

$$H_{cl}(s) = N \cdot \frac{(\alpha' + \beta')f_r s + \beta' f_r^2}{s^2 + (\alpha' + \beta')f_r s + \beta' f_r^2} \tag{3.27}$$

where $\alpha' = G_L\alpha$, $\beta' = G_L\beta$. In architecture-I ADPLL $G_L = G_I$, and in architecture-II ADPLL $G_L = G_{II}$. Comparing Eq. (3.27) with a classic 2^{nd} order s-domain transfer function:

$$H(s) = N \cdot \frac{2\zeta\omega_n s + \omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{3.28}$$

where ω_n is natural frequency and ζ is damping ratio, we get:

$$(\alpha' + \beta')f_r = 2\zeta\omega_n; \quad \beta' f_r^2 = \omega_n^2 \tag{3.29}$$

The natural frequency and damping ratio can be calculated from Eq. (3.30) and

(3.31):

$$f_n = \frac{\omega_n}{2\pi} = \frac{\sqrt{G_L \beta \cdot f_r}}{2\pi} \tag{3.30}$$

$$\zeta = \frac{\sqrt{G_L \beta(\alpha + \beta)}}{2\beta} \tag{3.31}$$

Locking time T_S and loop bandwidth f_{BW} for both architecture-I and architecture-II ADPLLs can be calculated by the following equations [50]:

$$T_S = \frac{4}{\omega_n \zeta} = \frac{8T_r}{G_L(\alpha + \beta)} \tag{3.32}$$

$$f_{BW} = f_n \sqrt{(1 - 2\zeta^2) + \sqrt{4\zeta^4 - 4\zeta^2 + 2}} \tag{3.33}$$

Assuming the locking time and phase noise constraints are:

$$T_S \leq T_L \tag{3.34}$$

$$0.45 \leq \zeta \leq 1.5 \tag{3.35}$$

where T_L is locking time constrains.

Regarding the phase noise, it is usually evaluated at a selected frequency offset (i.e. $1MHz$) from the central frequency. Additionally, for an under damped ($\zeta < 1$) PLL system, its closed loop transfer function has a peak value at the natural frequency, which will lead to an increased phase noise at this particular frequency point. Therefore, in the design analysis, we choose to analyze ADPLL output phase noise at frequency points f_n and $1MHz$ offset respectively for phase noise constraint characterizations. The output phase noise constraints are: phase noise at natural

frequency $S_{out}(f_n)$ and $1MHz$ $S_{out}(1_M)$ not exceeding S_{f_n} and S_{1_M}, respectively.

$$S_{out}(f_n) \leq S_{f_n} \tag{3.36}$$

$$S_{out}(1_M) \leq S_{1_M} \tag{3.37}$$

By plugging Eq. (3.30) and (3.31) into formulas (3.34) and (3.35) upon manipulations, we have:

$$\frac{8T_r}{G_L T_L} \leq \alpha + \beta \tag{3.38}$$

$$0.45 \leq \frac{\sqrt{G_L \beta(\alpha+\beta)}}{2\beta} \leq 1.5 \tag{3.39}$$

To demonstrate the model, an architecture-I ADPLL with the following parameters has been simulated: reference clock frequency $f_r = 13MHz$, feedback divider $N = 100.25$, TDC resolution $\tau_T = 50ps$, the 2^{nd} order MASH DSM with sampling frequency $f_{s\Delta}$ is in divider block, dithering resolution bit $N_{DCO} = 11$, $K_{DCO} = 100KHz/LSB$. The performance constraints are $T_L \leq 200\mu s$, $S_{f_n} = -75dBc/Hz$, $S_{1_M} \leq -110dBc/Hz$. Table 3.3 shows phase noise of each functional block. Accordingly, the total ADPLL output phase noise can be computed using Eq. (3.1).

Table 3.3: Phase noise of each functional unit.

REF	$S_{REF}(f) = 10^{-15} + \frac{10^{-11.2}}{f} + \frac{10^{-9.4}}{f^2} + \frac{10^{-7.0}}{f^3}$	Eq. (3.14)
PFD	$S_T = 1.075 \times 10^{-9}$	Eq. (3.21)
DCO	$S_{DQN} = 1.53 \times 10^{-}15$	Eq. (3.17)
	$S_{DCO}(f) = \frac{10^{0.33}}{f^2} + \frac{10^6}{f^3}$	Ref.[46]
DSM	$S_{DSM} = \pi^2[2\sin(\frac{\pi f}{f_{s\Delta}})]^2/(3f_{s\Delta})$	Eq. (3.18)

Fig. 3.8 illustrates the relationship between ADPLL locking time and DLF coefficients. According to formula (3.38), for the locking time to be less than a design

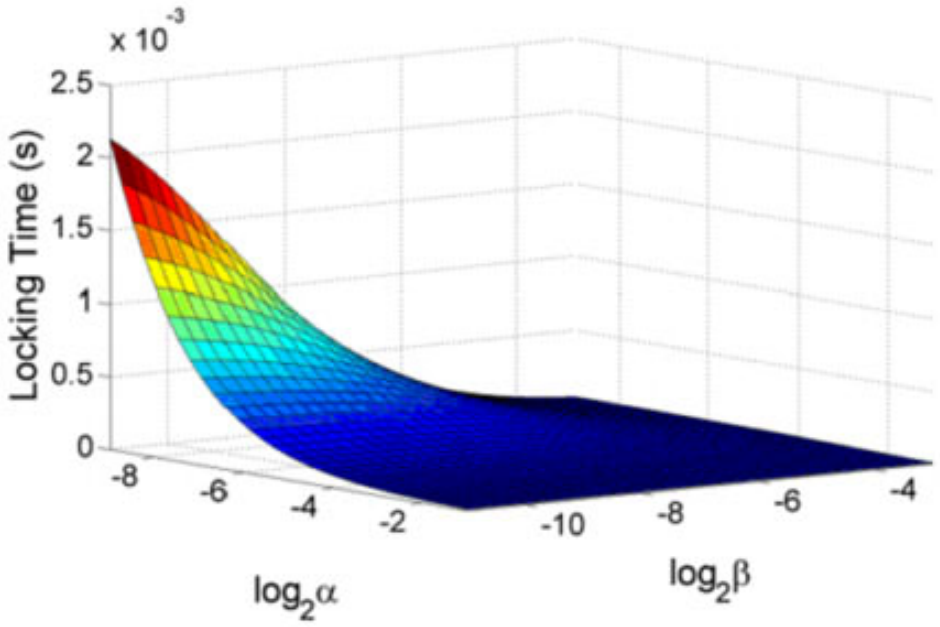


Figure 3.8: ADPLL locking time with different DLF coefficients.

specification, for instance, the DLF coefficients should meet the requirement specified by following formula:

$$\frac{8T_r}{G_L(\alpha+\beta)} \leq 200\mu s \Rightarrow \alpha+\beta \geq 0.026 \tag{3.40}$$

After substituting ADPLL parameters into formula (3.39) we have:

$$2.64\sqrt{\beta}-\beta \leq \alpha \leq 8.82\sqrt{\beta}-\beta \tag{3.41}$$

By plugging phase noise performance constraints, phase noise from each noise

source equations in Table 3.3 and close loop transfer functions of each noise source into formulas (3.36) and (3.37) respectively, we can derive:

$$\frac{1.7\beta^{0}.5 + 1.1 + 87\beta^{1}.5[0.12\beta(\beta+2\alpha)+\beta+\alpha]}{0.34\sqrt{\beta}[0.12(\alpha+\beta)+2\cos(0.34\sqrt{\beta})-2]^{2}} \leq 3.16\times10^{5} \tag{3.42}$$

$$3.25\alpha^{2} + 14.12\beta^{2} + 0.24\alpha + 2.06\beta + 0.07\alpha\beta \leq 1 \tag{3.43}$$

Based on formulas (3.40)-(3.43), we can plot the boundary condition lines as shown in Fig. 3.9. The shadow area specifies the possible DLF coefficient values for α and β that can meet all the requirements including locking time, phase noise and system stability.

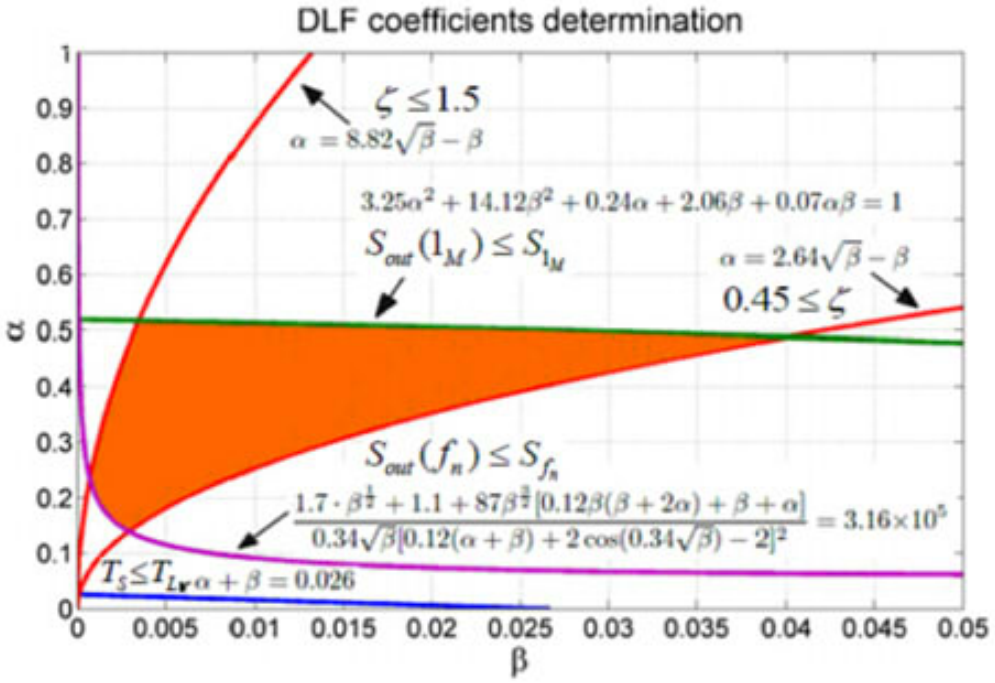


Figure 3.9: DLF coefficients determination.

Fig. 3.10 plots ADPLL output phase noise with different DLF coefficients. As

illustrated, when $\alpha = 2^{-1}$ and $\beta = 2^{-5}$, the maximum in band phase noise is -89dBc/Hz, the phase noise at $1MHz$ frequency offset is -110dBc/Hz. Accordingly, the locking time is calculated to be $9.8\mu s$. When $\alpha = 2^{-2}$ and $\beta = 2^{-7}$, the maximum in band phase noise is $-82dBc/Hz$, the phase noise at $1MHz$ frequency offset is $-111dBc/Hz$ and the locking time is $20.2\mu s$. Apparently, there is a trade off among ADPLL locking time, maximum in band phase noise and phase noise at the specific frequency offset (i.e. frequency offset at $1MHz$). Lower phase noise requires smaller DLF coefficients, while faster locking speed needs larger DLF coefficients. Therefore, it is important to set proper DLF coefficient values to achieve performance balance. In another word, to obtain lower phase noise, DLF coefficients should be selected from the left lower corner of the shadow area in Fig. 3.9. While to achieve a faster locking speed, DLF coefficients should be selected from the right upper corner of the shadow area in Fig. 3.9. Compared to the infinite frequency resolution in VCO, DCO resolution is determined by the last significant bit of DCO input control code. The limited frequency resolution causes DCO quantization noise. We have to consider the DCO resolution when DCO quantization noise is comparable to DCO oscillator noise. We use $N_{DCO} = 6$, $K_{DCO} = 100KHz/LSB$ in previous case study. The quantization noise of DCO is $-151dBc$ which is at least $12dB$ smaller than DCO oscillator noise at frequency offset range from $1KHz$ to $5MHz$. Hence, the quantization noise of DCO in this case study can be neglected.

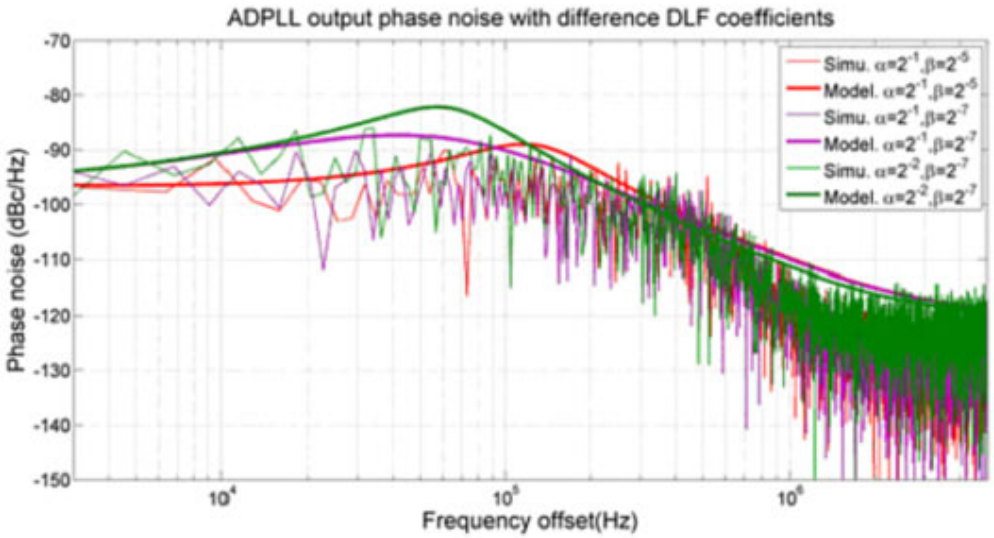


Figure 3.10: Output phase noise with different DLF coefficients.

3.6 Experimental Results

In order to verify the developed ADPLL analytical model, we select ADPLLs in references [47] and [48] as study cases. We develop models based on ADPLL topology presented in these papers and perform noise simulations. By comparing simulation results and measurement data provided in the original manuscripts, we are able to evaluate the proposed phase noise model effectiveness.

3.6.1 Architecture-I ADPLL

In [47], a 1.5GHz ADPLL is developed. The loop bandwidth is 20KHz. Other parameters are listed as follows: reference signal frequency f_r is $45MHz$. Output frequency is $1.5GHz$. Divider ratio N is 33.33. K_{DCO} equals $577Hz/LSB$. DCO dithering

resolution bit number is 5. TDC resolution equals $10ps$. The order number and bit number of $\Delta\Sigma$ modulator in DCO block is 1. The sampling frequency of $\Delta\Sigma$ modulator is $f_{S_\Delta} = f_r$. The DLF coefficients $\alpha = 2^2$ and $\beta = 2^{-6}$ are computed from loop bandwidth and phase margin. A $45MHz$ crystal oscillator [51] phase noise equals:

$$S_r(f) = 10^{-16.1} + \frac{10^{-10.2}}{f} + \frac{10^{-7.6}}{f^2} + \frac{10^{-6.1}}{f^3} \tag{3.44}$$

The free running DCO oscillator phase noise characteristic at 1.5GHz in [47, 60] is approximated as:

$$S_{DCO}(f) = 10^{-15.5} + \frac{10^{-8.2}}{f} + \frac{10^{-4.2}}{f^2} + \frac{10^{0.9}}{f^3} \tag{3.45}$$

By using the analytical modeling method developed in proceeding sections, we are able to characterize the noise contributions of different noise sources as shown in Fig. 3.11. As can be seen clearly, the in band phase noise is mainly contributed by the reference clock and PFD function block. In out-of-band frequency range, the total phase noise is mainly affected by PFD, DCO and $\Delta\Sigma$ modulator. The primary fractional spur occurs at the frequency $f_{spur} = F_{frac} \cdot f_r = 15MHz$. From Eq. (3.24), $rolloff(f_{spur})$ equals $-48.5dB$. The primary fractional spur PSD equals:

$$S_{spur} = 20\log[\frac{(2^1 - 1) \cdot 577}{2 \times 15MHz}] - 48.5dB = -142.8dBc \tag{3.46}$$

For result validation, we obtain ADPLL circuit hardware measurement results. The difference between the simulated and measured total phase noise at different frequency points are listed in Table 3.4. It shows that our model simulation is in a good

agreement with hardware results.

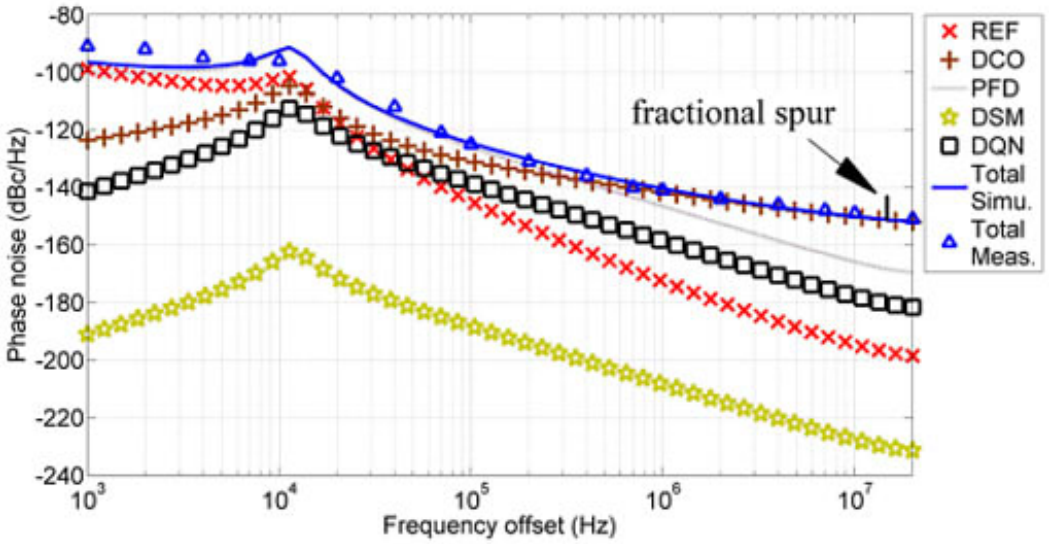


Figure 3.11: Simulated phase noise of each noise source and total phase noise at 1.5GHz output frequency.

Table 3.4: Output phase noise differences between simulation and measurement at some typical frequencies.

	Phase Noise (dBc/Hz) at frequency offset				Spur	
	10KHz	100KHz	1MHz	10MHz	Frequency	PSD
Simulation	-92.4	-124.7	-140.4	-150.1	15MHz	-142.8dBc
Measurement	-96.3	-124.8	-141.5	-149.5	15MHz	-142.5dBc
Error	3.9dB	0.1dB	1.1dB	-0.6dB	0	0.3dB

3.6.2 Architecture-II ADPLL

In [48], a 4.9GHz-6.9GHz fractional-N ADPLL for radio telecommunication is presented. The loop bandwidth of ADPLL is 200KHz. Other important parameters

are: reference clock frequency $f_r = 48MHz$, output frequency is $5.376021831GHz$, FCW is 112.0004548125. $K_{DCO} = 26KHz$. TDC-PD resolution $\tau_T = 15ps$. TDC output bit-width is 9 and the 2^{nd} order MASH $\Delta\Sigma$ modulator in DCO block with sampling frequency $f_{S_\Delta} = f_r$ is employed. There is an extra 9-bit fine tuning code in DCO block. The DLF coefficients are $\alpha = 2^{-5}$ and $\beta = 2^{-10}$. Phase noise PSD of the 48MHz reference clock and LC-tank DCO at 5.3GH output are approximated as follows, which are in agreements with the similar oscillators in references [51] and [52].

$$S_r(f) = 10^{-14.7} + \frac{10^{-10.3}}{f} + \frac{10^{-8.3}}{f^2} + \frac{10^{-4.8}}{f^3} \tag{3.47}$$

$$S_{DCO}(f) = 10^{-13.8} + \frac{10^{-0.2}}{f^2} \tag{3.48}$$

From our model analysis, the contribution of each noise source to the total output phase noise is shown in Fig. 3.12. The in band phase noise is mainly contributed by the reference clock noise and TDC-PD noise. In high noise frequency range, the total noise is affected by the TDC-PD, DCO and $\Delta\Sigma$ modulator. The primary fractional spur occurs at the frequency $f_{spur} = F_{frac} \cdot f_r = 21.8KHz$. Since the fractional spur locates within loop bandwidth, the $rolloff(f_{spur})$ equals 0dB. The primary fractional spur PSD equals

$$S_{spur} = 20\log[\frac{(2^2-1)\cdot 26KHz/2^9}{2\times 21.8KHz}] + 0dB = -49.1dBc \tag{3.49}$$

The difference between the simulated and measured total phase noise at different frequency points are listed in Table 3.5. Also, the simulation results from proposed model for architecture-II ADPLL show good agreement with hardware results.

In the presented phase noise model, other noise sources such as power supply

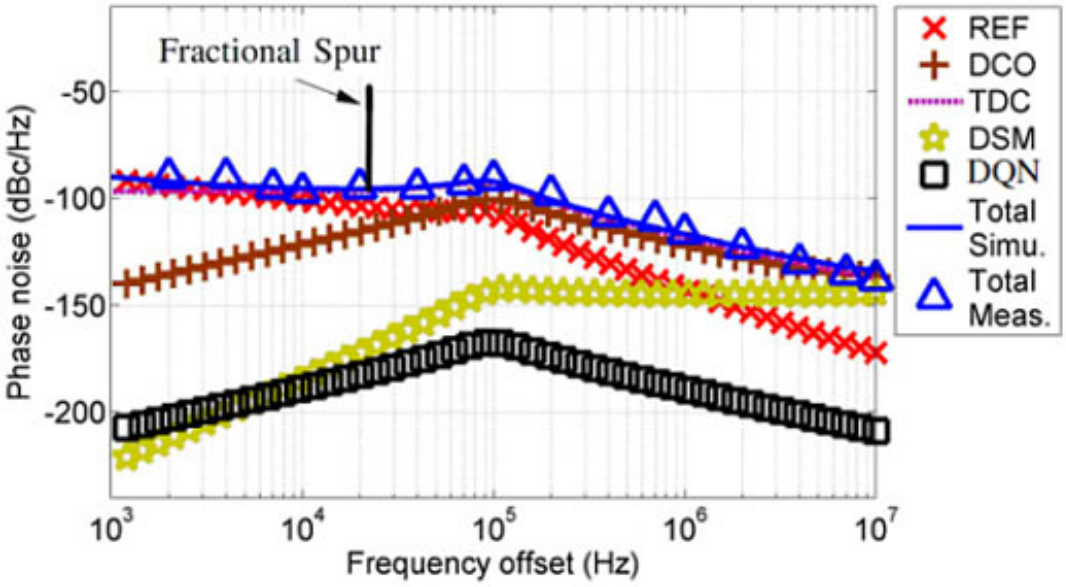


Figure 3.12: Simulated phase noise of each noise source and total phase noise at 5.376021831GHz output frequency.

noise and thermal noise are not considered, which is the major cause of the deviation between the simulation results and the measurement results. Those noises heavily depend on the operating conditions of the system that ADPLL is embedded in, such as the on-chip power grid signal integrity etc, whose noise characterizations are not available in the work presented in this paper. Actually, if that noise data is provided, its effect on ADPLL output phase noise can also be projected. Some other comparisons at typical frequency offsets of ADPLLs in references [61-63] are listed in Table 3.6.

Table 3.5: Output phase noise differences between simulation and measurement at some typical frequencies.

	Phase Noise (dBc/Hz) at frequency offset				Spur	
	10KHz	100KHz	1MHz	10MHz	Frequency	PSD
Simulation	-95.2	-92.1	-116.7	-134	21.8KHz	-49.1dBc
Measurement	-96.9	-90.7	-116.4	-137	20KHz	-45dBc
Error	1.7dB	-1.4dB	-0.3dB	3dB	1.8KHz	-4.1dB

Table 3.6: Simulation and measurement of output phase noise and fractional spur .

Ref.		Phase Noise (dBc/Hz) at frequency offset				Spur	
		10KHz	100KHz	1MHz	10MHz	Frequency	PSD
[61]	Simu.	-98	-103	-111	-120	10.88MHz	-117dBc
	Meas.	-99	-101	-111	-121	10MHz	-118.5dBc
[62]	Simu.	-98	-108	-109	-110	Integer Divider	
	Meas.	-99	-106	-107	-111		
[63]	Simu.	-87	-98	-123	NA	Integer Divider	
	Meas.	-85	-97	-123	NA		

3.7 Conclusions

In this chapter, an analytical noise model for both architecture-I and architecture-II ADPLLs is developed. By analyzing noise contribution and transfer function of each function block in ADPLL, the total phase noise can be projected. The total phase noise and primary fractional spur between simulation results and measurement data in the literature show reasonable agreements. Furthermore, we plot boundary condition lines according to phase noise, locking time and damping ratio constrains in order to determine proper DLF coefficients. This method can be used to project ADPLL phase noise and fractional spur and to guide ADPLL variables design.

Chapter 4

ADPLL On Chip Jitter Measurement

In the application such as time recovery, jitter performance is very important in the ADPLL. The data set-up and hold time are crucial issues. Jitter is the timing variation of a set of signal edges from their ideal values. It is typically caused by noise or other disturbances. In this research, we focus on period jitter which is time difference between an ideal cycle period and measured cycle period.

$$Jitter_{PER} = T_{mearsured} - T_{ideal} \tag{4.1}$$

Figure 4.1 shows a graphical definition of period jitter. The period jitter is measured as peak to peak jitter by the root of mean square (RMS).

In PLL, jitter performance can be improved by adjusting loop parameters including loop bandwidth and damping ratio. Generally, better jitter performance requires a small loop bandwidth. However, faster locking time needs a large bandwidth. In

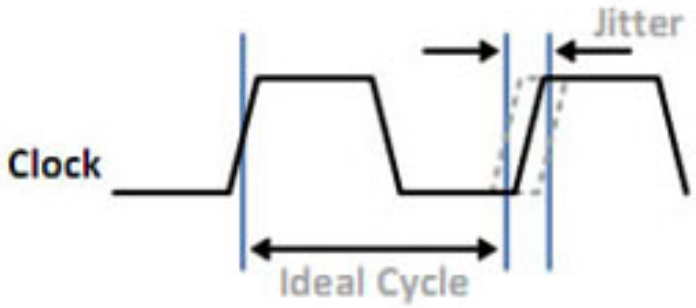


Figure 4.1: Period jitter.[15]

traditional ADPLL design, it is a trade off among loop bandwidth, jitter or phase noise performance and locking time. In our design, the circuit is applied a larger loop bandwidth to obtain a fast locking time. Then on-chip jitter measurement and self calibration block is used to make sure ADPLL jitter performance meet the requirement.

Traditional jitter measurement requires a spectrum analyzer. The previous jitter estimation methods include dead-zone algorithm [15] and variance metric algorithm. In the dead-zone algorithm shown in Fig. 4.2, it finds boundaries of jitter distribution for tail probability P. In variance metric jitter estimation as shown in Fig. 4.3, it combines measurements along jitter histogram to compute variance metric. However, the results from those methods are not accurate enough. Moreover, a spectrum analyzer is also needed in those methods which increases the measurement cost.

In this chapter, we first review the relationship among period jitter, phase noise and DCO control code. Then, we derive the equation to calculate jitter performance from DCO control code. We develop a mathematical model to calculate the PLL jitter performance by obtaining standard deviation of DCO control code, loop band-

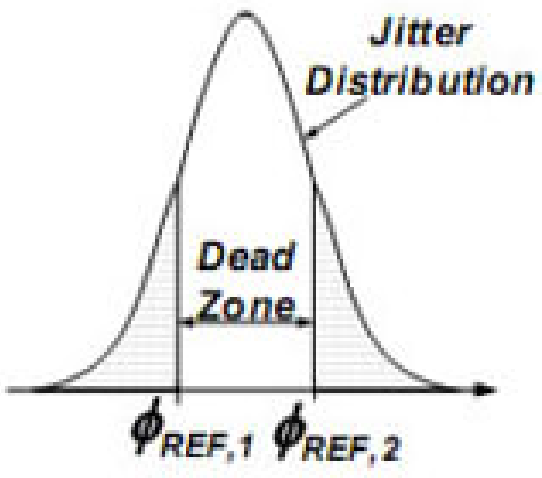


Figure 4.2: Dead zone method [15].

width, DCO gain, damping ratio and TDC resolution. According to the calculated jitter value, if jitter performance does not meet requirement, the loop bandwidth is adjusted by tuning loop filter coefficients value. At last, we propose the on chip jitter measurement circuit implemented at the transistor level. Both the behavior level simulation in Matlab and transistor level simulation in Cadence show good agreements with the mathematical analysis.

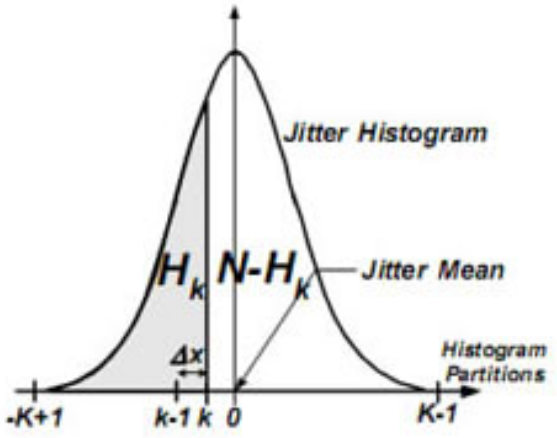


Figure 4.3: Variance metric jitter estimation [15].

4.1 Relationship among Jitter, Phase Noise and DCO Control Code

4.1.1 Relationship between Period Jitter and Phase Noise

We first review the period jitter calculation as follows. A periodic square wave is expressed as:

$$f(x) = \begin{cases} 0 & \text{if } -\pi \leq x < 0 \\ 1 & \text{if } 0 \leq x < \pi \end{cases} \tag{4.2}$$

and

$$f(x + 2\pi) = f(x) \tag{4.3}$$

The Fourier series of $f(x)$ is therefore:

$$f(x) = \frac{1}{2} + \frac{2}{\pi}\sin x + \frac{2}{3\pi}\sin 3x + \frac{2}{5\pi}\sin 5x + \cdots \tag{4.4}$$

It shows that the square-wave clock signal has the same jitter behavior as baseband harmonic sinusoid signal. Hence, PLL output signal with phase noise can be written as:

$$f_{out}(t) = A\sin[2\pi f_0 t + \theta(t)] = A\sin\{2\pi f_0[t + \frac{\theta(t)}{2\pi f_0}]\} \tag{4.5}$$

where f_0 is the central frequency and $\theta(t)$ denotes the phase jitter. The period jitter is:

$$J_{PER} = \frac{\theta(t)}{2\pi f_0} \tag{4.6}$$

Since the phase jitter magnitude is usually very small, we have

$$\begin{aligned} f_{out}(t) &= A\sin[2\pi f_0 t + \theta(t)] \\ &= A\sin(\omega_0 t)\cos\theta(t) + A\cos(\omega_0 t)\sin\theta(t) \\ &\simeq A\sin(\omega_0 t) + A\theta(t)\cos(\omega_0 t) \end{aligned} \tag{4.7}$$

where $\cos\theta(t) \simeq 1$ and $\sin\theta(t) \simeq \theta(t)$.

The spectrum of $f_{out}(t)$ is:

$$S_f(f) = \frac{A^2}{4}[\delta(f - f_0) + \delta(f + f_0)] + \frac{A^2}{4}[S_\theta(f - f_0) + S_\theta(f + f_0)] \tag{4.8}$$

where $S_\theta(f)$ is the spectrum of phase $\theta(t)$. We have:

$$\begin{aligned} L(f-f_0) &= 10\log[\frac{S_f(f)}{S_f(f_0)}] \\ &= 10\log[S_\theta(f-f_0)] \end{aligned} \tag{4.9}$$

The traditional phase noise measurement is shown as Fig. 4.4 [54]. The signal

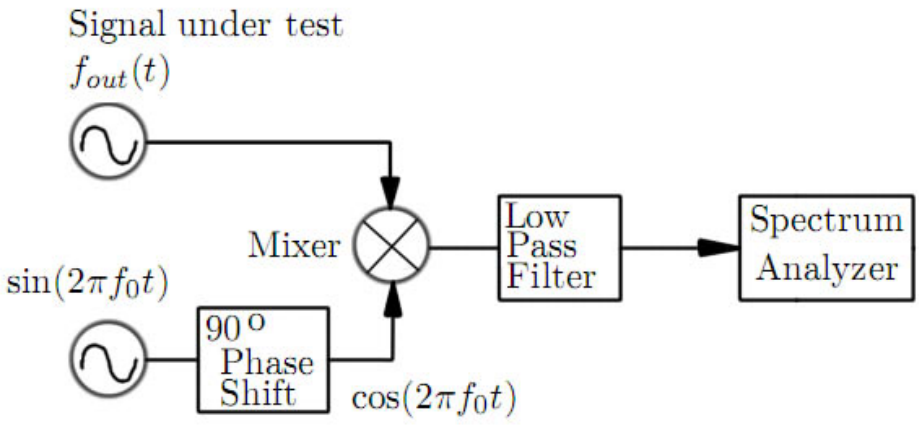


Figure 4.4: Traditional phase noise measurement [54].

$f_{out}(t)$ is mixed with $\cos(2\pi f_0 t)$ and filtered by the low pass filter. Thus, we can express the signal $n(t)$ at the input of the spectrum analyzer as:

$$n(t) = \frac{A}{2}\theta(t) \tag{4.10}$$

The signal spectrum on the spectrum analyzer is:

$$\begin{aligned} S_{sa}(f) &= \int_{-\infty}^{\infty} n(t)e^{-2\pi ft}dt \\ &= \frac{A^2}{4}S_\theta(f) \end{aligned} \tag{4.11}$$

After being scaled down by $\frac{A^2}{4}$, the $L(f)$ can be read in dBc directly from the spectrum analyzer. The relationship between phase spectrum $S_\theta(t)$ and phase noise $L(f)$ is:

$$\begin{aligned} S_\theta(f) &= \frac{4}{A^2}\int_{-\infty}^{\infty} n(t)e^{-2\pi ft}dt \\ &= \frac{4}{A^2}S_n(f) = 10^{\frac{L(f)}{10}} \end{aligned} \tag{4.12}$$

Therefore, the mean square value of $\theta(t)$ can be calculated by:

$$\begin{aligned} \langle\theta^2(t)\rangle &= 2\int_0^{\infty} S_\theta(f)df = 2\int_0^{\infty}\frac{4}{A^2}S_n(f)df \\ &= 2\int_0^{\infty} 10^{\frac{L(f)}{10}}df \end{aligned} \tag{4.13}$$

Above all, we can derive the relationship between the period jitter, J_{PER}, and the phase noise spectrum, $L(f)$ as [75]:

$$RMS\ J_{PER} = \frac{\sqrt{\langle\theta^2(t)\rangle}}{2\pi f_0} = \frac{\sqrt{2\int_0^{\infty} 10^{\frac{L(f)}{10}}df}}{2\pi f_0} \tag{4.14}$$

In chapter 2, Table 2.2 lists the phase noise requirements of each wireless standards. Once PLL phase noise sloped is projected, the jitter requirement can be calculated by given loop bandwidth (BW), output signal frequency (f_{out}) and phase noise requirement including phase noise (pn_{dBc}) and frequency offset (fre_{os}) [53]. Fig. 4.5 shows an example of phase noise slope of an ADPLL.

From Fig. 4.5, it is clear that $PN2 = PN1 - 20dB$ and $PN3 = PN1 - 30dB$.

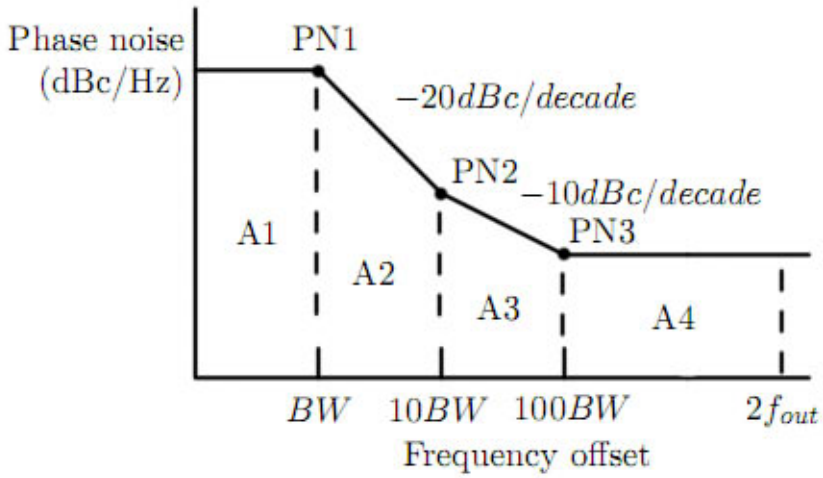


Figure 4.5: Behavior model of ADPLL on chip jitter measurement block.

$PN1$ can be calculated by given BW, pn_{dBc} and fre_{os}.

$$PN1 = \begin{cases} pn_{dBc} & \text{if } fre_{os} < BW \\ pn_{dBC} + 20 \times \log_{10}(\frac{fre_{os}}{BW}) & \text{if } BW < fre_{os} < 10BW \\ pn_{dBC} + 20dB + 10 \times \log_{10}(\frac{fre_{os}}{10BW}) & \text{if } 10BW < fre_{os} < 100BW \\ pn_{dBC} + 30dB & \text{if } 100BW < fre_{os} \end{cases} \tag{4.15}$$

The noise $A1$, $A2$, $A3$, $A4$ can be calculated by:

$$A1 = PN1 + 10\log_{10}(BW - 0) \tag{4.16}$$

$$A2 = \frac{PN1 + PN2}{2} + 10\log_{10}(10BW - BW) \tag{4.17}$$

$$A3 = \frac{PN2 + PN3}{2} + 10\log_{10}(100BW - 10BW) \tag{4.18}$$

$$A4 = PN3 + 10\log_{10}(2f_{out} - 100BW) \tag{4.19}$$

By using Eqn. (4.14), the RMS jitter is:

$$RMS\ J_{PER} = \frac{\sqrt{2 \cdot 10^{\frac{A1}{10}}} + \sqrt{2 \cdot 10^{\frac{A2}{10}}} + \sqrt{2 \cdot 10^{\frac{A3}{10}}} + \sqrt{2 \cdot 10^{\frac{A4}{10}}}}{2\pi f_{out}} \tag{4.20}$$

Tabel 4.1 lists the jitter requirements of each wireless communication standard by using Matlab function [A.1.3].

Table 4.1: Jitter requirements of each wireless communication standard

Standards	Output frequency	Phase noise requirement	Bandwidth	Jitter requirement
GSM	920MHz	-105dBc/Hz@1MHz	50KHz	4.4ps
UMTS	2GHz	-100dBc/Hz@5MHz	50KHz	90.9ps
WiMAX	2.4GHz	-90dBc/Hz@100KHz	50KHz	16.3ps
WiMAX	3.5GHz	-95dBc/Hz@100KHz	50KHz	7.3ps
Bluetooth	2.4GHz	-109dBc/Hz@1MHz	50KHz	12.9ps
Wi-Fi	2.45GHz	-102dBc/Hz@1MHz	50KHz	28.6ps
Wi-Fi	5.2GHz	-102dBc/Hz@1MHz	50KHz	13.8ps
Wi-Fi	5.7GHz	-102dBc/Hz@1MHz	50KHz	13.0ps

4.1.2 Relationship between ADPLL Jitter and DCO Control Code

In the ADPLL, the DCO control code $code[n]$ is updated every reference clock time T_{ref}. The mean value of DCO control code (M_C) is calculated as:

$$M_C = \frac{\Sigma_{n=1}^{N_c} code[n]}{N_c} \tag{4.21}$$

Therefore, the DCO control code can be expressed as:

$$code[n] = M_C + \eta_c[n] \tag{4.22}$$

where $\eta_c[n]$ is the difference between mean value of DCO control code (M_C) and DCO control code $code[n]$. Assuming DCO gain (K_{DCO}) remains the constant when DCO output is in a small tuning range. We have:

$$f_{out}(t) = A\sin[2\pi M_C K_{DCO} t + 2\pi\eta_c[n]K_{DCO}t] \tag{4.23}$$

Hence, the RMS jitter can be calculated:

$$RMS\ J_{PER} = \frac{\sqrt{\langle code[n]^2\rangle}K_{DCO}}{2\pi f_0^2} \tag{4.24}$$

where $\sqrt{\langle code[n]^2\rangle}$ is the standard deviation of DCO control code.

Above all, it is clear that, the ADPLL RMS period jitter can be estimated with the value of the standard deviation of DCO control code, DCO gain and central frequency of output signal.

4.2 On Chip Jitter Measurement Circuit Design

4.2.1 Behavior Model

Fig. 4.6 shows the behavior model of the proposed ADPLL on-chip jitter measurement circuit which only measures and calcultes the standard deviation of DCO control code $\sqrt{\langle code[n]^2 \rangle}$. For a given DCO gain and the central frequency of output signal, the jitter can be calculated using equation (4.24).

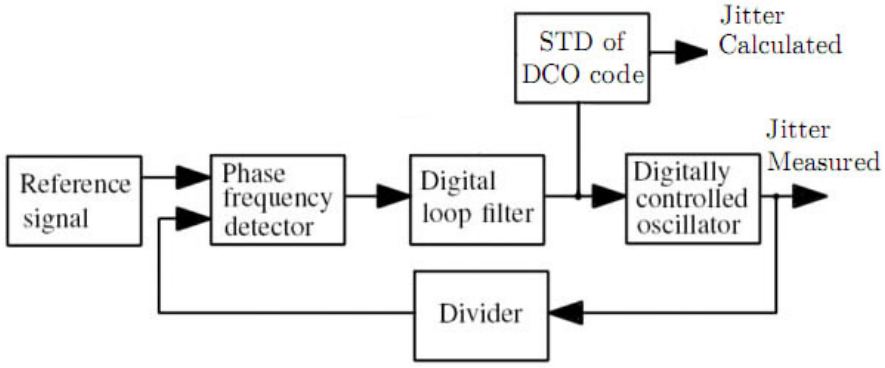


Figure 4.6: Behavior model of ADPLL on chip jitter measurement block.

The detail model of on-chip jitter measurement block is shown in Fig. 4.7.

In our ADPLL, the output frequency variation $\Delta f / f_{out} \cdot 100\%$ is less than 1% when the circuit is locked. The output frequency variation (FV) can be estimated by:

$$FV = \frac{code[n] \cdot res}{f_{out}} \cdot 100\% \tag{4.25}$$

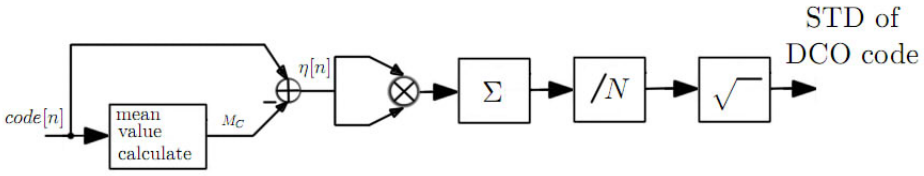


Figure 4.7: Detail model of ADPLL on chip jitter measurement block.

where res means the corresponding resolution. Table 4.2 summarizes the relationship between bit number of code[n] and frequency variation of the ADPLL output in presented ADPLL design. From table 4.2, it is clear that the bit number of $code[n]$

Table 4.2: DCO control code variation vs DCO output frequency variation

Bit number of jitter measurement input	Frequency Variation		
	Structure I	Structure II	Structure III
4 bits	$\approx 0.26\%$	$\approx 0.22\%$	$\approx 0.2\%$
3 bits	$\approx 0.13\%$	$\approx 0.11\%$	$\approx 0.1\%$
2 bits	$\approx 0.07\%$	$\approx 0.06\%$	$\approx 0.05\%$

should be 3 to minimized the circuit complexity.

4.2.2 Matlab Simulation

In this subsection, a Matlab model is developed to compare the simulation result obtained from the jitter measurement block and the results obtained from the direct measurement. Also, the accuracy influenced by the bit length of jitter measurement input is analyzed. Fig. 4.8 shows the ADPLL simulation model in MATLAB. In Fig. 4.8, the frequency of reference signal is $13MHz$. The PFD compares the phase difference of reference signal and feedback signal. The detailed PFD and TDC are

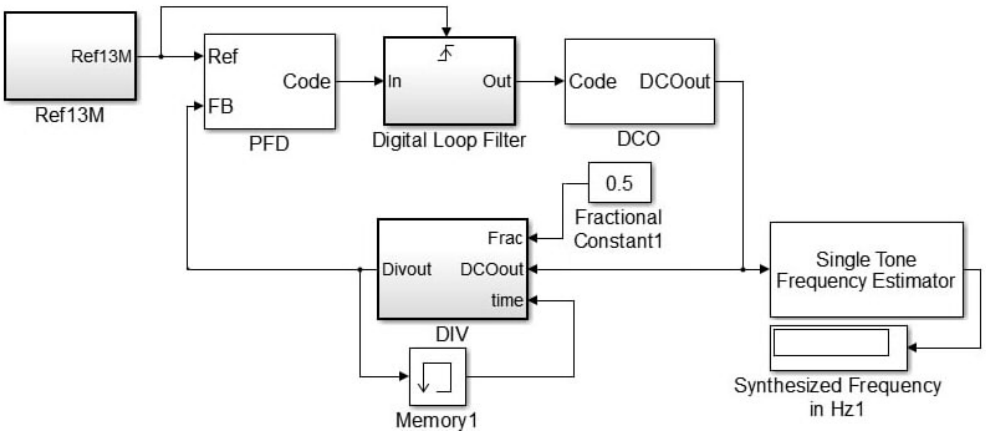


Figure 4.8: ADPLL simulation model in MATLAB.

shown in Fig. 4.9 and Fig. 4.10. The embedded functions in PFD and TDC are $y = u \cdot v$ and $y = u$, respectively. The phase difference signal goes through digital

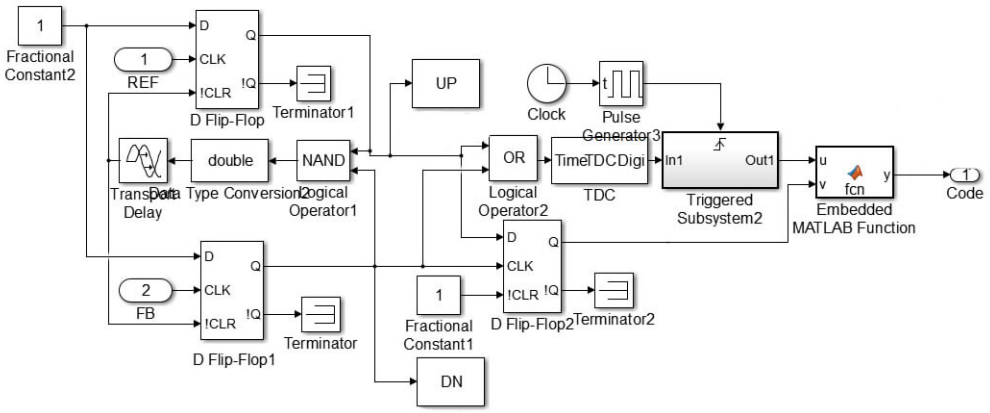


Figure 4.9: PFD model in MATLAB.

low pass filter and then generates DCO control code signal. The detailed Matlab model of LPF is shown in Fig. 4.11. There is a single tone frequency estimator to calculate the DCO output signal's frequency. In the feedback path, there is a fractional frequency divider with a delta sigma modulator as shown in Fig. 4.12. We use 256 reference cycles to calculate the standard deviation of DCO control code and

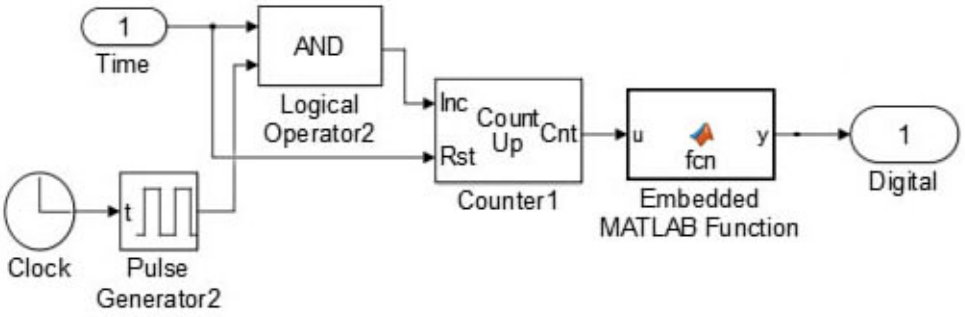


Figure 4.10: TDC model in MATLAB.

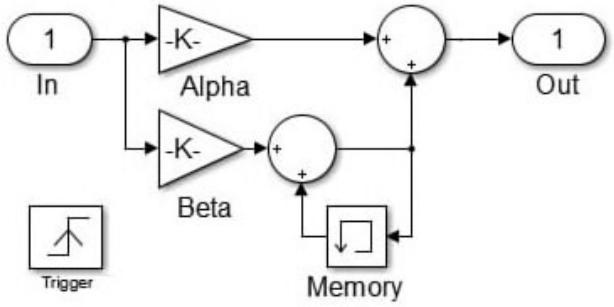


Figure 4.11: LPF model in MATLAB.

the ADPLL jitter performance [76].

Table 4.3 summarizes the calculated STD of DCO control code and both calculated and simulated jitter performances under different output frequencies. The K_{DCO} represents the resolution of 6-bit fine tune.

According to Table 4.1, the preset STD of DCO control code values can be determined by equation (4.24). The threshold values are the first 8-bit of STD values of DCO control code and listed in Table 4.4.

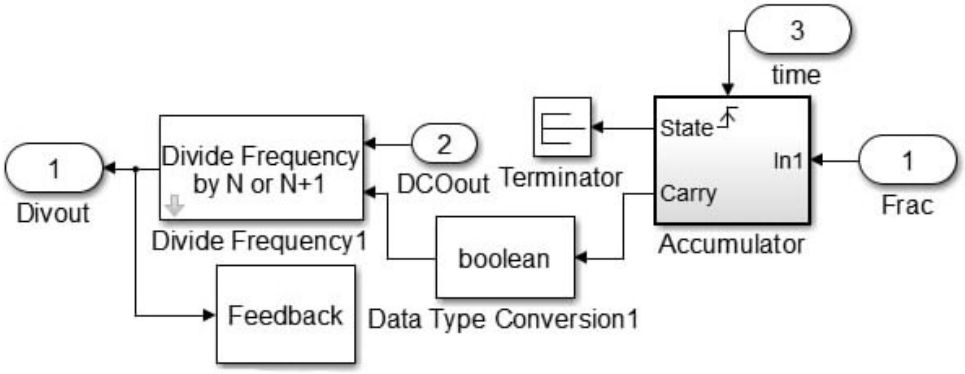


Figure 4.12: Divider with delta sigma modulato model in MATLAB.

Table 4.3: Simulated standard deviation of DCO control code and measured jitter results

Standard	Output frequency	K_{DCO} KHz	STD of DCO code	Jitter	
				Calculated	Simulated
GSM	880MHz	96	162	3.2ps	2.7ps
GMTS	2GHz	480	272	5.2ps	2.1ps
WiMAX	2.4GHz	480	279	3.7ps	1.8ps
WiMAX	3.6GHz	480	593	3.5ps	1.3ps
Bluetooth	2.4GHz	480	279	3.7ps	1.8ps
Wi-Fi	2.45GHz	480	283	3.6ps	1.7ps
Wi-Fi	5.7GHz	960	595	2.8ps	0.9ps

Table 4.4: Preset threshold value of STD of DCO control code

Frequency band	Jitter requirement	STD of DCO code	Threshold value
800MHz-1.1GHz	4.4ps	223	66
1.9GHz-2.8GHz	12.9ps	609	152
3.2GHz-4GHz	7.3ps	978	244
5GHz-6GHz	130ps	276	69

4.2.3 Circuit Topology

Fig. 4.13 shows the detailed schematic of the proposed on-chip jitter measurement behavior model. The mean value calculate block is shown in Fig. 4.14.

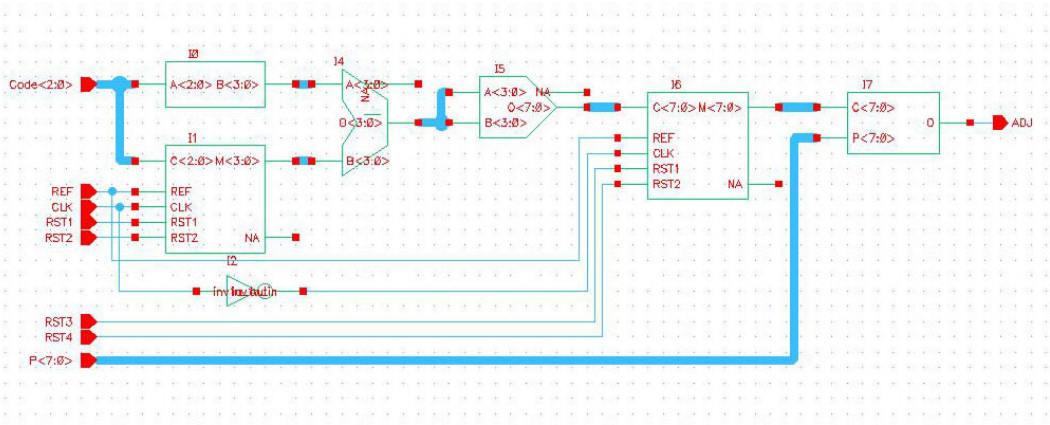


Figure 4.13: Schematic of the proposed on-chip jitter measurement.

Comparing to Fig. 4.7, the square root block is removed to reduce operation time and chip size. In the presented design, one reference cycle time could be saved by removing the square root block. The jitter measurement output is compared against a preset threshold code. If the output is larger than the threshold code, it means jitter performance does not meet the requirement. Then the comparison block generates DLF adjustment signal '1' to tune the DLF coefficients. If the output is smaller than the threshold code, the comparison block generates signal '0' to stop the adjustment, and DLF values remain unchanged.

Fig. 4.15 shows the detailed circuits of 8-bit comparator in the on-chip jitter measurement block.

Fig. 4.16-Fig. 4.19 show the histograms of 256 number of DCO control codes when output frequency is 920MHz, 2.4GHz, 3.6GHz and 5.6GHz, respectively.

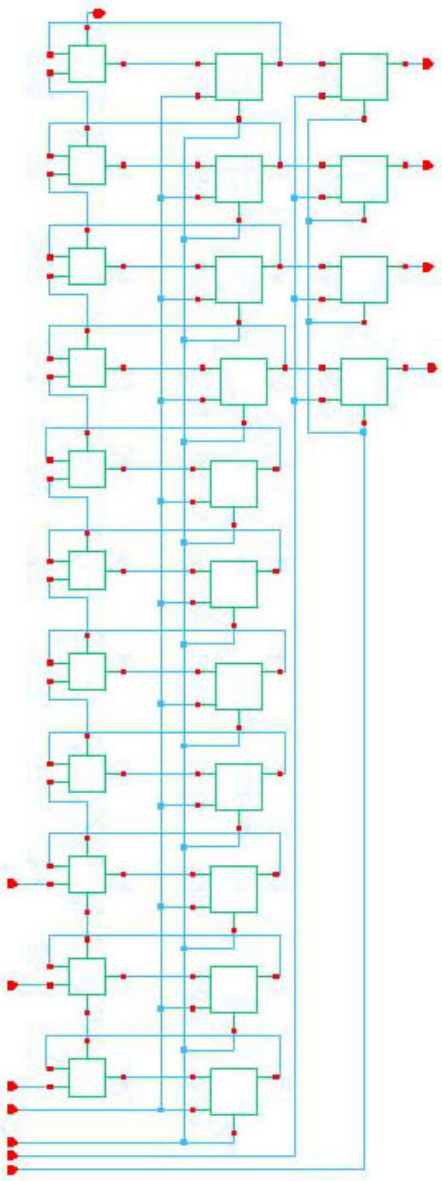


Figure 4.14: Schematic of mean value calculation block.

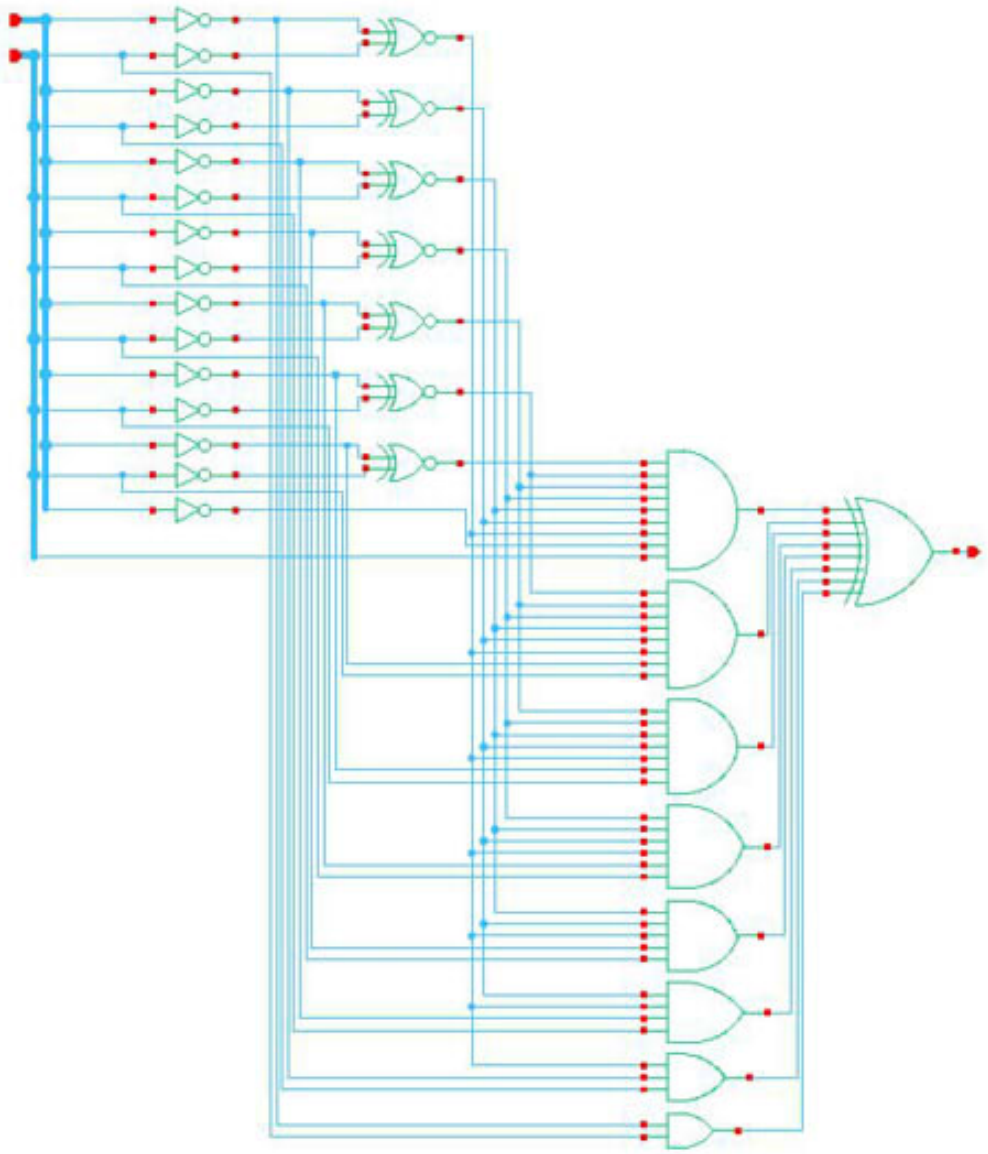


Figure 4.15: 8-bit digital comparator.

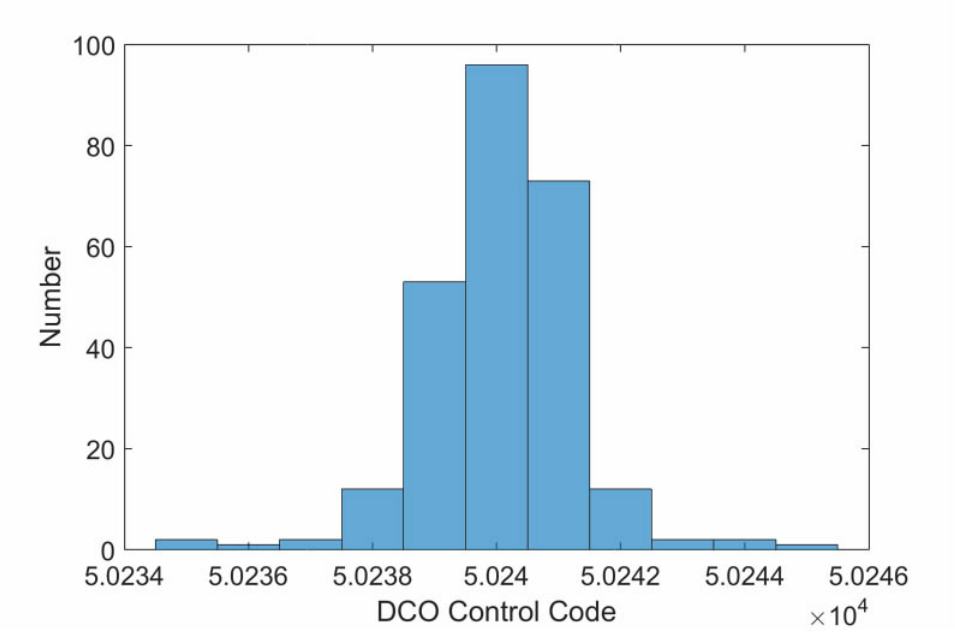


Figure 4.16: Histogram of 256 DCO control code when output frequency is 920MHz.

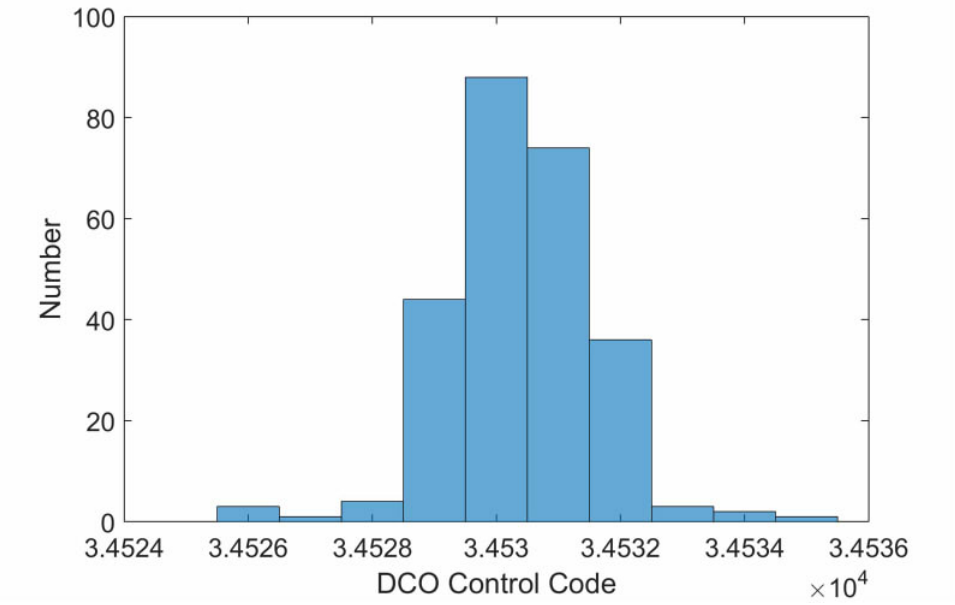


Figure 4.17: Histogram of 256 DCO control code when output frequency is 2.4GHz.

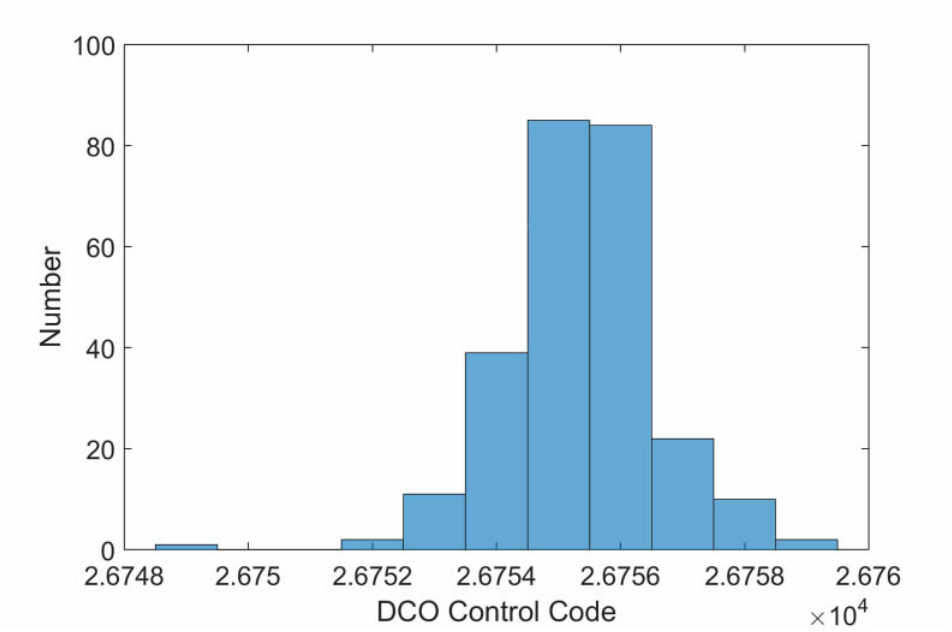


Figure 4.18: Histogram of 256 DCO control code when output frequency is 3.6GHz.

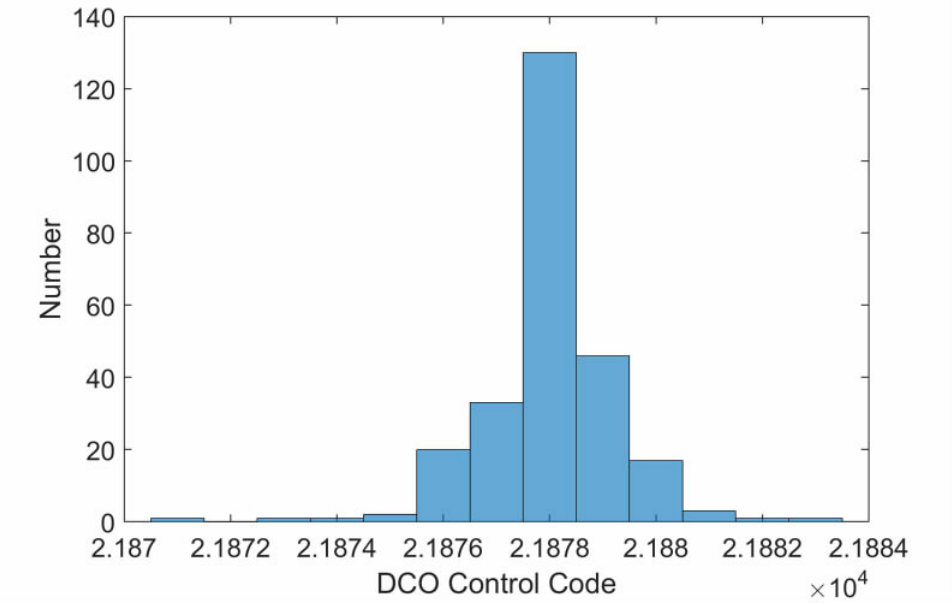


Figure 4.19: Histogram of 256 DCO control code when output frequency is 5.65GHz.

4.2.4 Experimental Results

In order to verify the presented ADPLL on-chip jitter measurement block. The proposed ADPLL with on-chip jitter measurement is implemented and simulated in the IBM 0.13μm CMOS technology. The simulation tool is Cadence SpectreRF. The calculated jitter performance is calculated from on-chip jitter measurement block. The simulated jitter performance is obtained directly from ADPLL output. The jitter requirement is converted by phase noise performance requirement. Table 4.5 summarizes all jitter performance results under different wireless standards.

Fig. 4.20 shows the example of simulated jitter measurement. The jitter is simulated in Cadence of the actual ADPLL output. The RMS jitter is 3.6ps when output signal frequency is 920MHz.

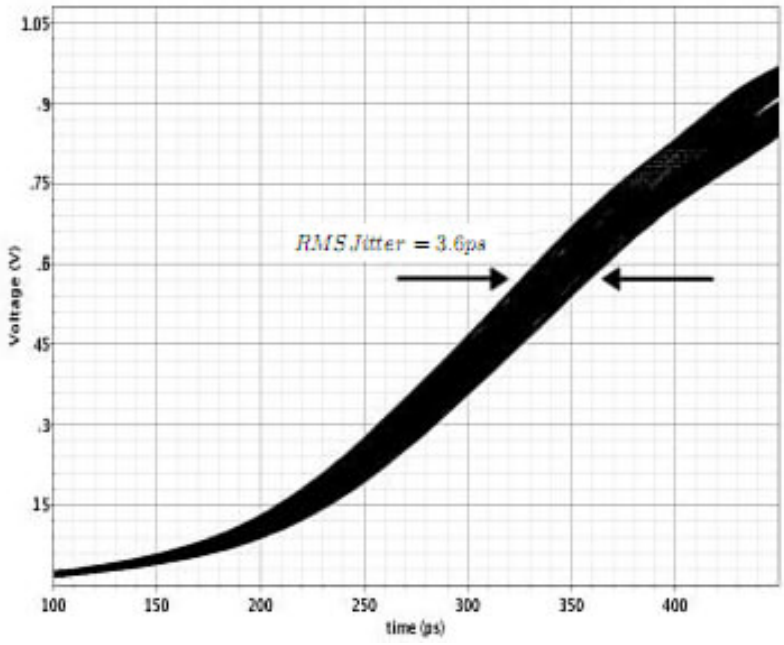


Figure 4.20: Jitter measurement simulation at 1GHz.

Table 4.5: Simulated jitter resultes VS. measured jitter results

Standard	Output frequency	Jitter		
		Calculated	Simulated	Requirement
GSM	920MHz	3.8ps	3.6ps	4.4ps
GMTS	2GHz	4.8ps	2.1ps	90.9ps
WiMAX	2.4GHz	4.2ps	1.8ps	16.3ps
WiMAX	3.6GHz	3.7ps	1.3ps	7ps
Bluetooth	2.4GHz	4.3ps	1.8ps	12.9ps
Wi-Fi	2.45GHz	3.9ps	1.7ps	29ps
Wi-Fi	5.7GHz	2.6ps	0.9ps	13ps

4.3 Conclusion

In this chapter, we analyze the relationship among ADPLL jitter, phase noise performance and DCO control code and present the behavior model of jitter measurement to calculate the jitter by using DCO control code. According to the mathematical analysis, the jitter can be estimated by calculating the standard deviation of DCO control code. At last, we develop the on-chip jitter measurement circuit. The simulation results show the good agreement with analytical results.

Chapter 5

Wide band ADPLL with Self Jitter Measurement and Calibration Design

Based on the multi-band ADPLL design in chapter 2 and the on-chip jitter measurement design in chapter 4, the wide band ADPLL with self-jitter calibration is proposed. In this chapter, we first present an adjustable digtal low pass filter. The low pass filter coefficients can be adjusted based on DLF adjust signal. So that, the loop bandwidth is adjusted and the ADPLL jitter performance is improved to meet requirements. Then, the whole ADPLL circuit topology of wide band ADPLL with self-jitter measurement and calibration is presented. The ADPLL structure is designed and simulated using 0.13μm CMOS technology in Cadence. At last, We adopt an external noise to test jitter calibration function. The simulation results show that, jitter measurement block works after PLL is locking. Then jitter calibration function block works once jitter performance does not meet requirements. The

jitter performance is significantly improved with jitter calibration block.

5.1 Digital Loop Filter with Self Jitter Calibration Design

5.1.1 Coefficients adjustable PPI filter

Once the jitter performance does not meet the jitter requirement, jitter measurement block will generate the control signal to the digital loop filter to adjust DLF coefficients. Hence, the DLF circuit should be adaptive. We improved the DLF design in Chapter 2. The adjustable proportional and integral circuit blocks for the proposed DLF are presented in Fig. 5.1 and Fig. 5.2. In Fig. 5.1, the proportional path coefficient α is controlled by a 3-bit control code. The range of α is from 2^{-3} to $2^{0} - 2^{-3}$. The corresponding resolution is 2^{-3}. In Fig. 5.2, the integral path coefficient β is also controlled by a 3-bit control code. β ranges from 2^{-7} to $2^{-4} - 2^{-7}$. The corresponding resolution is 2^{-7}.

The top level model of the adjustable low pass filter is shown as Fig. 5.3. The low pass filter coefficients can be adjusted based on LPF adjustment signal. Once jitter measurement block generates DLF control signal '1', the DLF coefficients are reduced by the minimum step.

DLF coefficients affect the loop band-width and phase margin of the designed ADPLL. The ADPLL bandwidth f_{BW} is usually 10 times lower than the reference signal frequency to ensure system stability with a sufficient damping ratio. Table 5.1 summarizes ADPLL loop bandwidths and damping ratio for different output

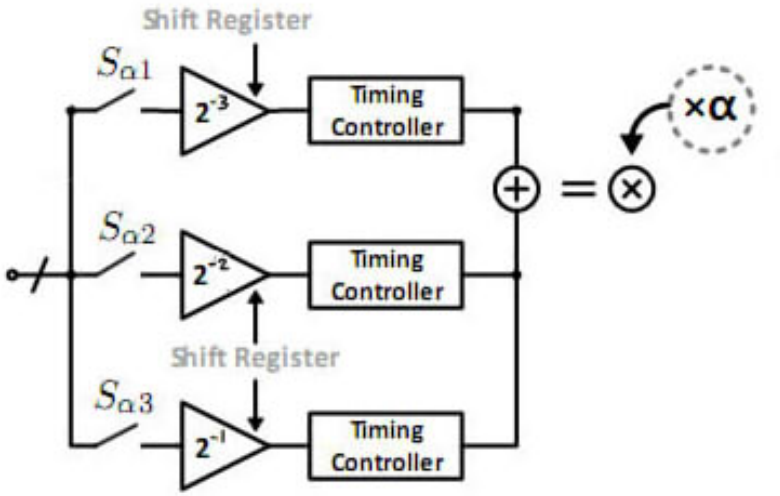


Figure 5.1: The architecture of adjustable proportional path.

frequencies by tuning the DLF coefficients.

Table 5.1: Loop band-width and phase margin in different frequency with different pair of DLF coefficients

Frequency (Hz)	α	β	Bandwidth (Hz)	Damping ratio
1G	$2^{-3} \sim 2^{0} - 2^{-3}$	$2^{-7} \sim 2^{-4} - 2^{-7}$	8K∼157K	0.52∼1.3
2.4G	$2^{-3} \sim 2^{0} - 2^{-3}$	$2^{-7} \sim 2^{-4} - 2^{-7}$	5.4K∼124K	0.7∼1.5
3.6G	$2^{-3} \sim 2^{0} - 2^{-3}$	$2^{-7} \sim 2^{-4} - 2^{-7}$	6.9K∼147K	0.5∼1.5
5.6G	$2^{-3} \sim 2^{0} - 2^{-3}$	$2^{-7} \sim 2^{-4} - 2^{-7}$	5.9K∼133K	0.7∼1.5

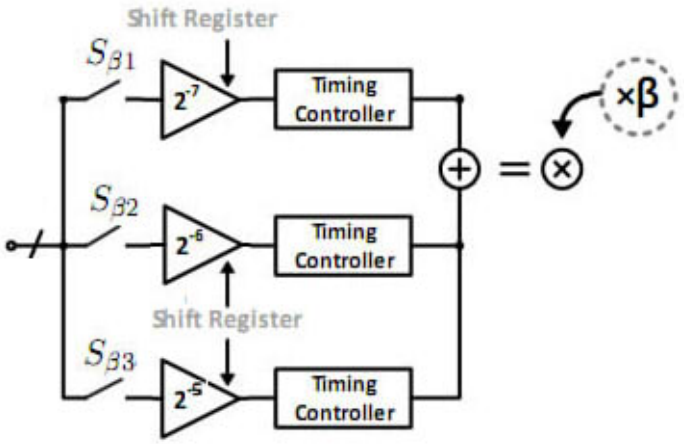


Figure 5.2: The architecture of adjustable integral path.

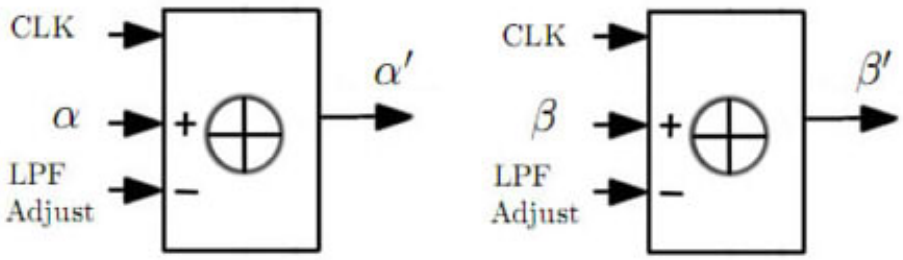


Figure 5.3: Top level model of adjustable low pass filter.

5.1.2 Circuit topology

Fig. 5.4 shows the detailed circuit topology of adjustable digital low pass filter.

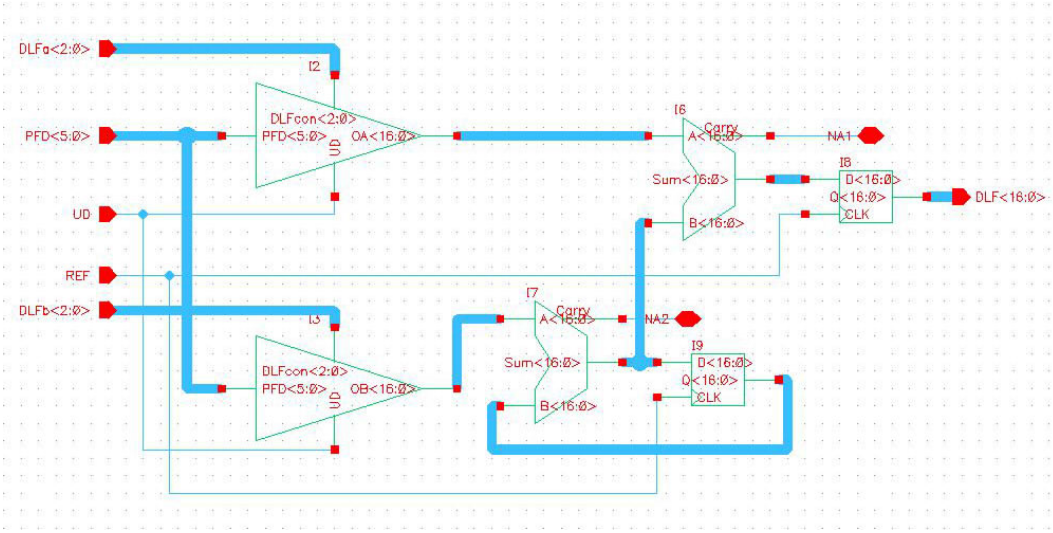


Figure 5.4: The adjustable low pass filter.

The adjustable function is controlled by a delay cell block. After 256 cycles of reference signal, the values of DLF coefficients can be adjusted. Once the alpha or beta control code equals "001", the adjust function is not available and generates a warning signal. The detailed schematic of DLF coefficients adjustment is shown in Fig. 5.5.

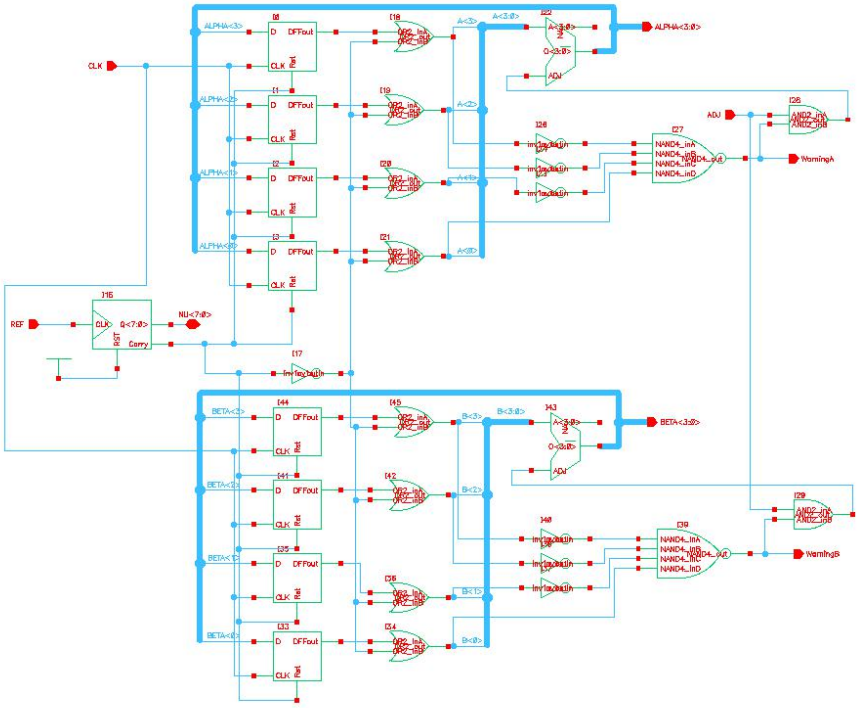


Figure 5.5: The adjustable DLF coefficients.

5.2 Wide band ADPLL with Self Jitter Measurement and Calibration Circuit Topology

In chapter 2 and chapter 4, we have presented each block of the proposed wide band ADPLL and on-chip jitter measurement block. In previous section, we have also presented an adjustable digital low pass filter. In this section, we connect the on-chip measurement block and the adjustable digital low pass filter block to the wide band ADPLL as shown in the following.

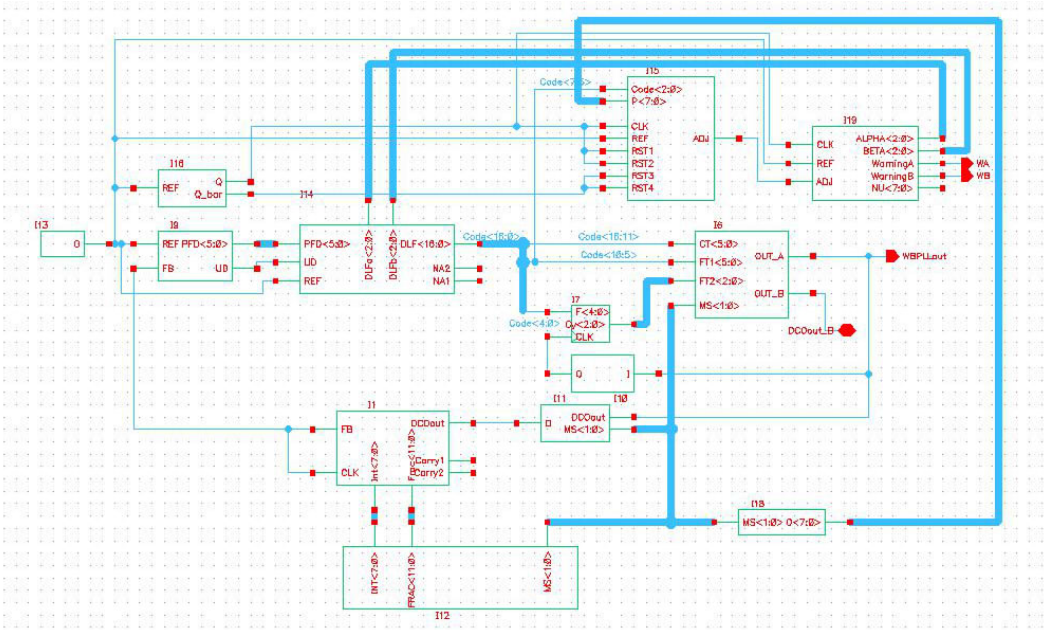


Figure 5.6: Wide band ADPLL with on chip jitter measurement and self jitter calibration.

The overall measurement results such as frequency response and phase noise performance of the proposed ADPLL are shown in chapter 2. In this section, we list

the comparisons between ADPLL phase noise and jitter performance with and without self-jitter calibration block. External noise is added to the proposed ADPLL as shown in Fig. 5.7. The external Gaussian noise is generated by using MATLAB Gaussian noise generate function. The mean values of the noises are all '0' and the variances of the noises are 1.5, 5.6, 6.5 and 1.2 in 920MHz, 2.4GHz, 3.6GHz and 5.6GHz, respectively.

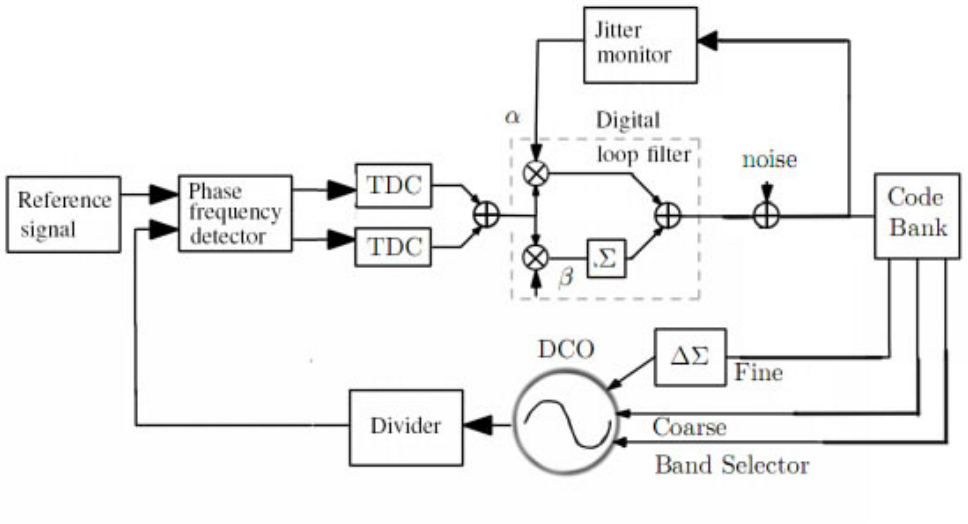


Figure 5.7: Wide band adaptive ADPLL with external noise model.

Fig. 5.8-Fig.5.11 show the histograms of DCO control code (Code<7:5>) without and with jitter calibration block when output frequency are 920MHz, 2.4GHz, 3.6GHz and 5.6GHz, respectively. It is clear that, the standard deviations of Code<7:5> are reduced with the self jitter calibration block.

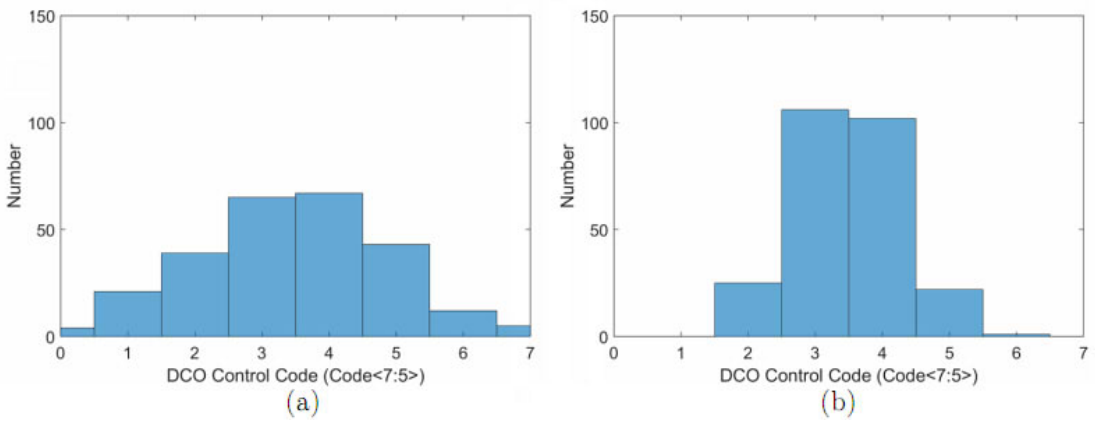


Figure 5.8: Histogram of Code<7:5> with noise in 920MHz (a) without jitter calibration (b) with jitter calibration.

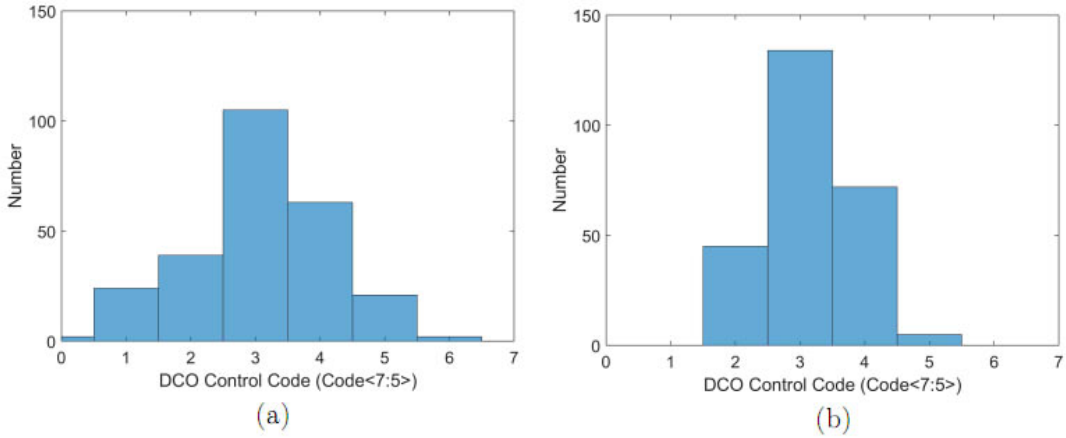


Figure 5.9: Histogram of Code<7:5> with noise in 2.4GHz (a) without jitter calibration (b) with jitter calibration.

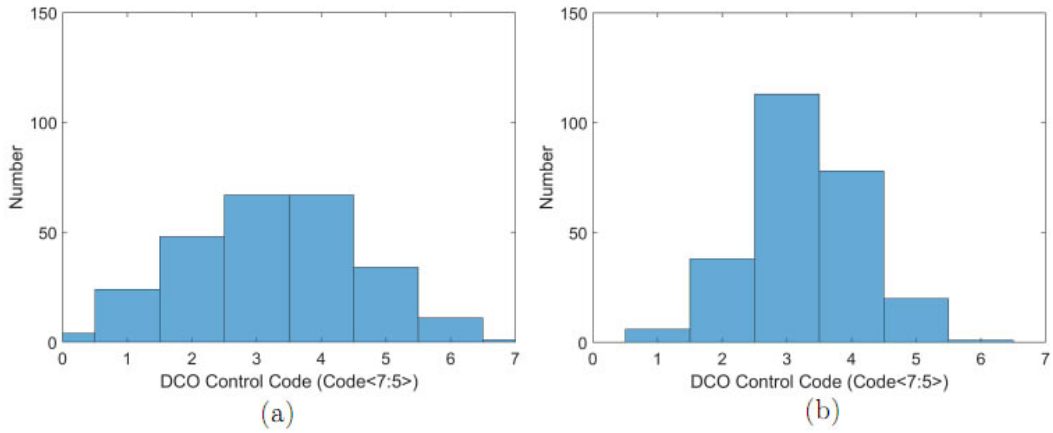


Figure 5.10: Histogram of Code<7:5> with noise in 3.6GHz (a) without jitter calibration (b) with jitter calibration.

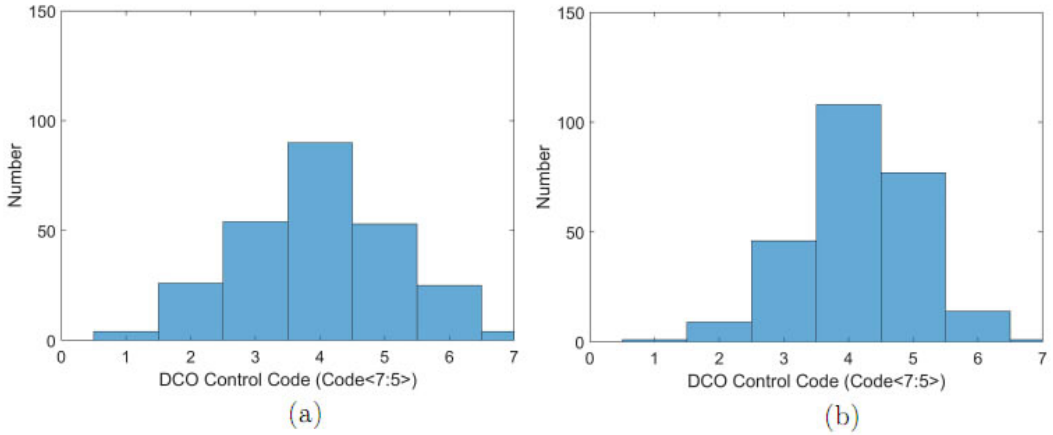


Figure 5.11: Histogram of Code<7:5> with noise in 5.6GHz (a) without jitter calibration (b) with jitter calibration.

5.2.1 Phase noise and jitter performance

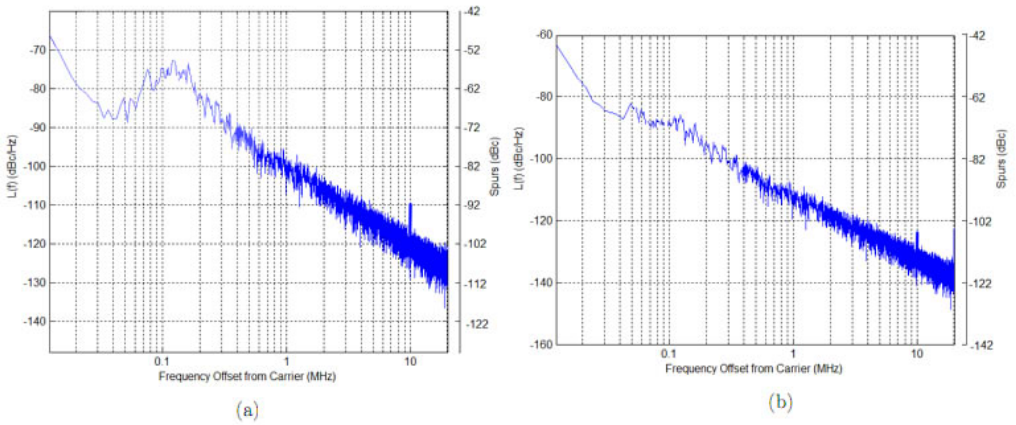


Figure 5.12: Phase noise performance at 920MHz output (a) without calibration block; (b) with calibration block.

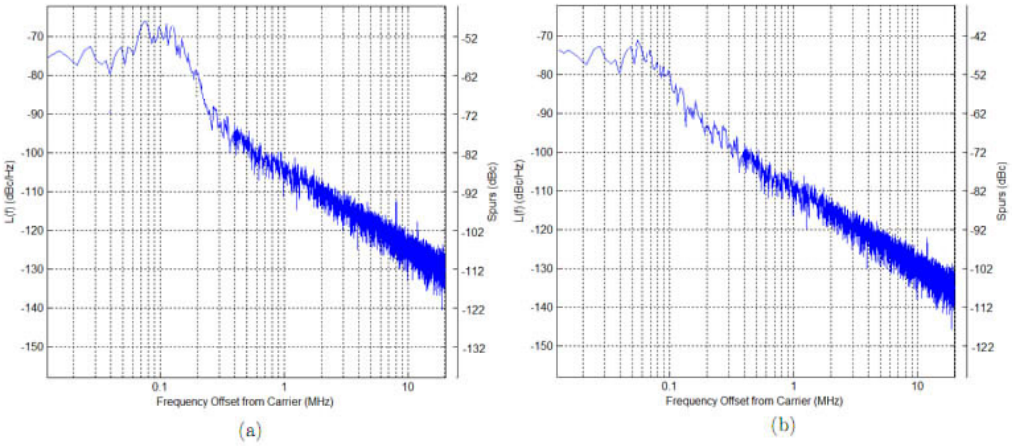


Figure 5.13: Phase noise performance at 2.4GHz output (a) without calibration block; (b) with calibration block.

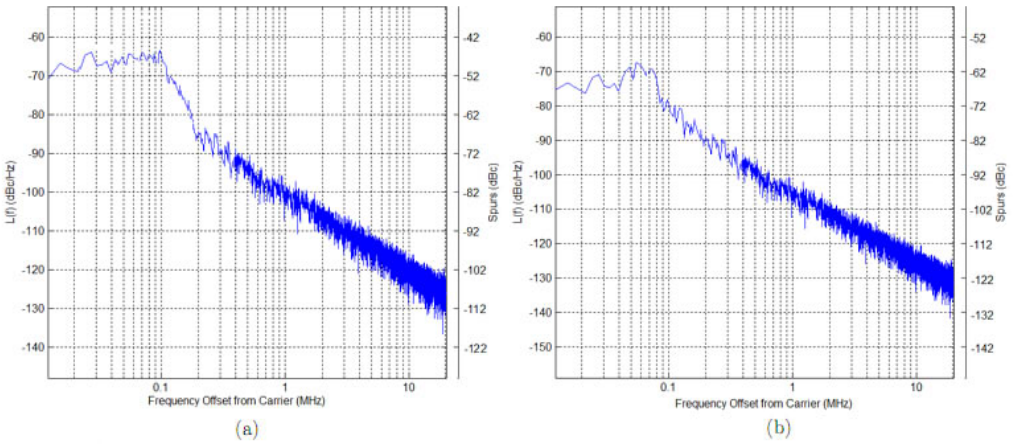


Figure 5.14: Phase noise performance at 3.6GHz output (a) without calibration block; (b) with calibration block.

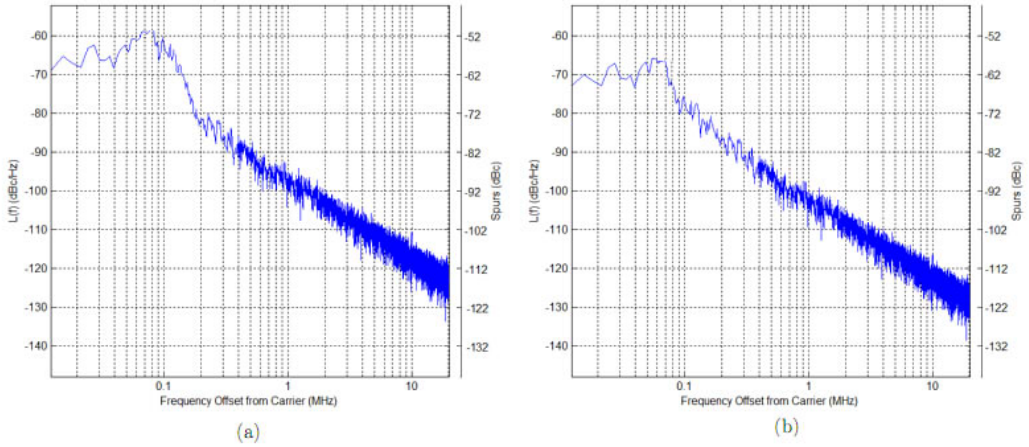


Figure 5.15: Phase noise performance at 5.6GHz output (a) without calibration block; (b) with calibration block.

Fig. 5.12- 5.15 show phase noise performance with and without jitter calibration block. In Fig. 5.12, the fractional spur is $-91dBc$ without jitter calibration and is $-104dBc$ with jitter calibration. In Fig. 5.13, the fractional spur is $-94dBc$ without jitter calibration and is $-97dBc$ with jitter calibration. In Fig. 5.14, the fractional spur is $-100dBc$ without jitter calibration and is $-113dBc$ with jitter calibration. In Fig. 5.15, the fractional spur is $-94dBc$ without jitter calibration and is $-97dBc$ with jitter calibration.

The corresponding jitter performance and loop bandwidth are listed in Table 5.2.

Table 5.2: Phase noise and jitter performance w/o jitter calibration block

Frequency (GHz)	Without jitter calibration			With jitter calibration		
	Phase noise @1MHz (dBc/Hz)	Jitter (ps)	Bandwidth KHz	Phase noise @1MHz (dBc/Hz)	Jitter (ps)	Bandwidth KHz
920MHz	-100	6.9	157	-112	3.8	49.5
2.4GHz	-104	18.9	124	-109	11.6	51.5
3.6GHz	-100	9.8	147	-105	6.5	67.0
5.6GHz	-97	14.9	133	-102	11.5	81.6

It is clear that, the self-jitter calibration block can effectively detect and adjust ADPLL when the jitter performance does not meet the requirement. With the self-jitter calibration block, the loop bandwidth is reduced and phase noise and jitter performance are improved.

5.2.2 Power consumption

The power consumption is a very important issue for on-chip micro systems. We have listed power consumption value of each block of wide band ADPLL in chapter 2. In this section, we focus on the power consumptions of adjustable low pass filter, on-chip jitter measurement and the overall ADPLL system.

The power consumption values of all function blocks of the wide band ADPLL with self-jitter calibration are summarized in Table 5.3.

Table 5.3: Multi-band ADPLL with self jitter calibration power consumption

Standard	Frequency Hz	PFD mW	LPF mW	DIV mW	DCO mW	Jitter Meas. mW	Total mW
GSM	1G	0.01	7.0	7.3	8.3	2.3	24.9
GMTS	2G	0.01	7.1	8	8.9	2.3	26.3
WiMAX	2.4G	0.01	7.1	8.4	9.1	2.3	26.9
WiMAX	3.6G	0.01	7.3	8.8	10.3	2.3	28.7
Bluetooth	2.4G	0.01	7.1	8.4	9.1	2.3	26.9
Wi-Fi	2.45G	0.01	7.1	8.4	9.1	2.3	26.9
Wi-Fi	5.7G	0.01	7.5	9.6	12.2	2.3	31.6

The ADPLL has the lowest power $24.9mW$ when output frequency is 880MHz and the highest power $31.6mW$ when output frequency is 5.8GHz. Fig. 5.16 and Fig. 5.17 show the current of the ADPLL when the system is locked. Fig. 5.16 shows the minimum current whose average value is 20.75mA and Fig. 5.17 shows the maximum current whose average value is 26.33mA.

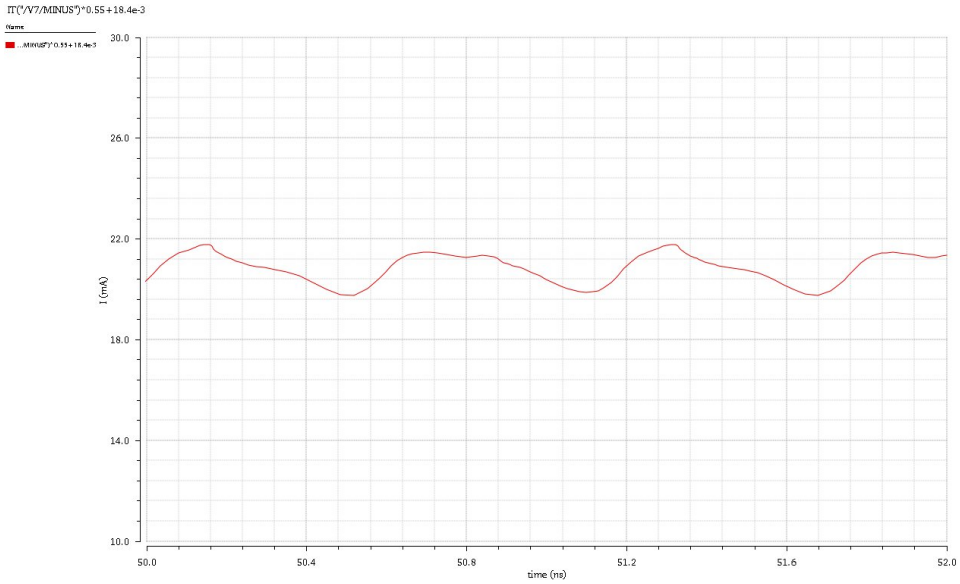


Figure 5.16: The minimum current of the ADPLL.

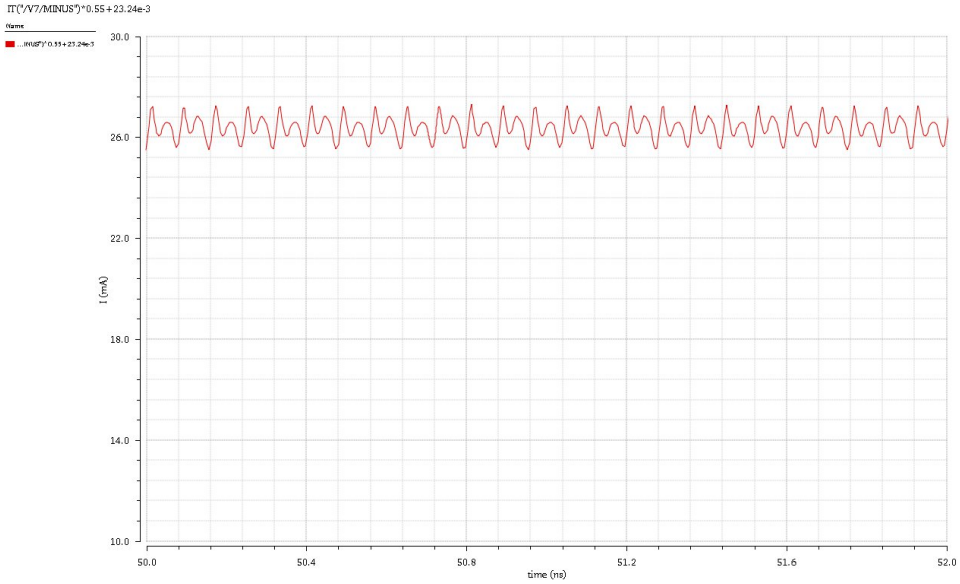


Figure 5.17: The maximum current of the ADPLL.

5.2.3 Measurement of ADPLL

The overall measurements of the proposed ADPLL are shown in Table 5.4. The proposed ADPLL is designed and simulated using Cadence IC design tools. The proposed ADPLL with a novel multi-band LC tank DCO provides wide frequency tuning range, less power consumption, and lower jitter values in various frequency ranges.

The range of frequency of the proposed ADPLL is from 850MHz to 1.1GHz, 1.8GHz to 2.8GHz, 3.2GHz to 3.8GHz and 5GHz to 6GHz. The linearity of output frequency of the proposed ADPLL is presented in the section of DCO. The detailed resolution is analyzed in chapter 2. The jitter is measured through the output of digitally controlled oscillator (DCO). The phase noise performance is calculated through the DCO output by developing a MATLAB script which is included in the appendix section (A.1.2).

5.3 Conclusions

This chapter focuses on the wide band ADPLL with self-jitter calibration design. A compact multi-band fractional frequency synthesizer covering GSM, GMTS, bluetooth, WiMAX and Wi-Fi is presented in this work. An on-chip jitter measurement technique is proposed to monitor PLL jitter performance. By calculating standard deviation of DCO control code, the RMS jitter value can be obtained with loop bandwidth, output frequency and DCO gain parameter values. Also, an adjustable low pass filter (LPF) is presented. The LPF coefficients are controlled with the codes

Table 5.4: Overall measurements for the proposed ADPLL

	Proposed ADPLL
Process	$0.13\mu m$
Locking Time	$< 1\mu s$
Frequency Range	0.8GHz-1.1GHz 1.9GHz-2.8GHz 3.2GHz-4GHz 5GHz-6GHz
Resolution	3KHz@0.8-1.1GHz 15KHz@1.9-2.8GHz 15KHz@3.2-4GHz 30KHz@5-6GHz
Phase Noise Performance	-116.6@1MHz offset@1GHz output -112.4@1MHz offset@2GHz output -110.2@1MHz offset@2.4GHz output -105.8@1MHz offset@3.6GHz output -105.6@1MHz offset@5.6GHz output
Power Consumption	24.9mW@1GHz 26.3mW@2GHz 26.9mW@2.4GHz 28.7mW@3.6GHz 31.6mW@5.7GHz

generated from the on-chip jitter measurement block. It also has the advantage of reducing ADPLL locking time when LPF coefficients are pre-setted with large numbers. We provide the whole circuit topology and transistor level simulation of proposed ADPLL. At last, we summarize the comparisons of proposed ADPLL with conventional wide band ADPLLs.

Chapter 6

Conclusion

This research focused on the analysis and design of a wide tuning range, low noise and self noise calibration all digital phase-locked loop. Using the theory and developed circuits, an all-digital phase-locked loop with multi-band frequency range based on a multi-mode LC tank DCO designed. Efforts concentrated on the design of the new multi-band digitally controlled oscillator, phase noise analysis, on-chip jitter measurement circuit, adjustable low pass filter and self jitter calibration circuit. The key research contributions are highlighted as below:

1. The proposed multi-band ADPLL is able to operate in frequency of 800MHz-1.1GHz, 1.9GHz-2.8GHz, 3.2GHz-4GHz and 5GHz-6GHz. A unique multi-mode digitally controlled oscillator based on multi-resonant LC tank is presented. The proposed ADPLL is able to cover multiple wireless communication standards including GSM, UMTS, bluetooth, WiMAX and Wi-Fi.

2. An analytic noise model for ADPLLs is developed. A z-domain model of ADPLL and the transfer function of each function block are derived. Utilizing the analytical model, the total phase noise can be projected by analyzing the noise con-

tributions and the transfer function of each function block. Also, the fractional spur in the fraction ADPLL is analyzed. Furthermore, the boundary condition lines are derived accordance with phase noise, locking time and damping ratio constrains in order to determine proper DLF coefficients. Finally, ADPLL parameter including TDC resolution, LPF coefficients can be pre-determined by phase noise, locking time and damping ratio requirements.

3: The relationship between ADPLL jitter performance and DCO control code is analyzed. The jitter can be estimated by calculating the standard deviation of DCO control code. Based on this mathematical analysis, an on-chip jitter measurement function block is developed, which is accomplished by using standard deviation calculation of DCO control code block.

4: A self-jitter calibration block is presented to adjust the ADPLL jitter performance. We develop an adjustable low pass filter in this research. Once the on-chip jitter measurement detects that the jitter performance does not meet the requirement, the loop filter coefficients will be adjusted to improve the jitter of ADPLL.

5: A multi-band ADPLL with self-jitter calibration is presented based on the proposed wide band ADPLL, on-chip jitter measurement and self-jitter calibration block. The circuit is simulated by using Cadence Spectre and IBM8rf 0.13μm CMOS process develop kits. Comparing the proposed ADPLL with conventional ADPLLs, the advantages of this work include: wider tuning range, better phase noise/jitter performance, lower power consumption and adjustable loop bandwidth.

Appendix A

Matlab Codes

A.1 Matlab Codes for Case Studies in Chapter 3

Phase noise caused by reference signal

Matlab file: "phasenoise_ref_cal.m"

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%This function is used to estimate PLL phase noise caused by
%reference signal
%please provide following parameters:
%k0ref,k1ref,k2ref,k3ref,loop gain (GL); Divider value or FCW (N);
%low pass filter coeffients (alpha, beta)
%The output is reference phase noise in dB
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function phasenoise_ref=phasenoise_ref_cal(k0ref,k1ref,k2ref,k3ref,N,GL,alpha,
    beta)
%%%Z transform%%%
Zres=50;                    %z transform resolution
f=logspace(2,7.7,Zres);     %z transform frequency
omega=2*pi*f;               %radius frequency
z=tf('z',ts);

```

```
sref=k0ref+k1ref./f+k2ref./(f.^2)+k3ref./(f.^3);
%z domain reference noise signal
Href=N*GL*(alpha*(z-1)+beta*z)/((z-1)^2+GL*(alpha*(z-1)+beta*z));
%Transferfunction of reference noise signal

[magref,phaseref]=bode(Href,omega); %bode function

for i=1:1:Zres
phirefx(i)=magref(:,:,i);
end;

srefout=sref.*(phirefx.^2);            %phase noise caused by reference signal
phasenoise_ref=10.*log10(srefout);     %reference phase noise in dB
```

Phase noise caused by DCO

Matlab file: "phasenoise_dco_cal.m"

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%This function is used to estimate PLL phase noise caused by DCO
%please provide following parameters:
%k0dco,k1dco,k2dco,k3dco,loop gain (GL);
%low pass filter coeffients (alpha, beta)
%The output is DCO phase noise in dB
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function phasenoise_dco=phasenoise_dco_cal(k0dco,k1dco,k2dco,k3dco,GL,alpha,beta
    )
%%%Z transform%%%
Zres=50;                    %z transform resolution
f=logspace(2,7.7,Zres);     %z transform frequency
omega=2*pi*f;               %radius frequency
z=tf('z',ts);

sdco=k0dco+k1dco./f+k2dco./(f.^2)+k3dco./(f.^3);
%z domain DCO noise signal
Hdco=(z-1)^2/((z-1)^2+GL*(alpha*(z-1)+beta*z));
%Transferfunction of DCO noise signal

[magdco,phasedco]=bode(Hdco,omega); %bode function

for i=1:1:Zres
phidcox(i)=magdco(:,:,i);
end;

sdcoout=sdco.*(phidcox.^2);            %phase noise caused by DCO
```

```
phasenoise_dco=10.*log10(sdcoout);     %DCO phase noise in dB
```

Phase noise caused by DCO quantization noise

Matlab file: "phasenoise_dqn_cal.m"

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%This function is used to estimate PLL phase noise caused by
%DCO quantization noise
%please provide following parameters:
%Divider value (N),loop gain (GL); TDC gain (GT)
%low pass filter coeffients (alpha, beta)
%The output is DQN phase noise in dB
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function phasenoise_dqn=phasenoise_dqn_cal(Ndco,N,GL,GT,alpha,beta)
%%%Z transform%%%
Zres=50;                    %z transform resolution
f=logspace(2,7.7,Zres);     %z transform frequency
omega=2*pi*f;               %radius frequency
z=tf('z',ts);

sdqn=1/(12*(2^Ndco)^2*fref); %z domain DQN signal
Hdqn=(N*GL/GT)*(z-1)/((z-1)^2+GL*(alpha*(z-1)+beta*z));
%Transferfunction of DQN signal

[magdqn,phasedqn]=bode(Hdqn,omega); %bode function

for i=1:1:Zres
phidqnx(i)=magdqn(:,:,i);
end;

sdqnout=sdqn.*(phidqnx.^2);             %phase noise caused by DQN
phasenoise_dqn=10.*log10(sdqnout);      %DQN phase noise in dB
```

Phase noise caused by TDC-PFD architecture-I

Matlab file: "phasenoise_pfdI_cal.m"

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%This function is used to estimate PLL phase noise caused by
%TDC-PFD noise Architecture-I
%please provide following parameters:
%TDC resolution (tauT), reference freuqncy (fref);
```

```
%Output signal freuqency (fout);
%Divider value (N),loop gain (GL); TDC gain (GT)
%low pass filter coeffients (alpha, beta)
%The output is PFD architecture-I phase noise in dB
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function phasenoise_pfdI=phasenoise_pfdI_cal(tauT,fref,fout,N,GL,GT,alpha,beta)
%%%Z transform%%%
Zres=50;                    %z transform resolution
f=logspace(2,7.7,Zres);     %z transform frequency
omega=2*pi*f;               %radius frequency
z=tf('z',ts);

spfdI=(pi*tauT*fout)^2/(3*fref); %z domain PFD architecture-I
HpfdI=(N*GL/GT*(alpha*(z-1)+beta*z))/((z-1)^2+GL*(alpha*(z-1)+beta*z));
%Transferfunction of PFD architecture-I signal

[magpfdI,phasepfdI]=bode(HpfdI,omega); %bode function

for i=1:1:Zres
phipfdIx(i)=magpfdI(:,:,i);
end;

spfdIout=spfdI.*(phipfdIx.^2);            %phase noise caused by PFD
    architecture-I
phasenoise_pfdI=10.*log10(spfdIout);      %PFD architecture-I phase noise in dB
```

Phase noise caused by TDC-PFD architecture-II

Matlab file: "phasenoise_pfdII_cal.m"

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%This function is used to estimate PLL phase noise caused by
%TDC-PFD noise Architecture-II
%please provide following parameters:
%TDC resolution (tauT), reference freuqncy (fref);
%Output signal freuqency (fout);
%loop gain (GL); TDC gain (GT)
%low pass filter coeffients (alpha, beta)
%The output is PFD architecture-II phase noise in dB
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function phasenoise_pfdII=phasenoise_pfdII_cal(tauT,fref,fout,GL,GT,alpha,beta)
%%%Z transform%%%
Zres=50;                    %z transform resolution
f=logspace(2,7.7,Zres);     %z transform frequency
```

```
omega=2*pi*f;              %radius frequency
z=tf('z',ts);

spfdII=(pi*tauT*fout)^2/(3*fref); %z domain PFD architecture-II
HpfdII=-(GL/GT*(alpha*(z-1)+beta*z))/((z-1)^2+GL*(alpha*(z-1)+beta*z));
%Transferfunction of PFD architecture-II signal

[magpfdII,phasepfdII]=bode(HpfdII,omega); %bode function
for i=1:1:Zres
phipfdIIx(i)=magpfdII(:,:,i);
end;
spfdIIout=spfdII.*(phipfdIIx.^2);          %phase noise caused by PFD
    architecture-II
phasenoise_pfdII=10.*log10(spfdIIout);     %PFD architecture-II phase noise in
    dB
```

Phase noise caused by DSM in DCO block architecture-I

Matlab file: "phasenoise_dsm1I_cal.m"

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%This function is used to estimate PLL phase noise caused by
%DSM noise Architecture-I in DCO block
%please provide following parameters:
%DSM order (m), DSM sampling freuqncy (fdsm); DSM bit number (Ndsm)
%Divider value (N), loop gain (GL); TDC gain (GT)
%low pass filter coeffients (alpha, beta)
%The output is DSM Architecture-I in DCO block phase noise in dB
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function phasenoise_dsm1I=phasenoise_dsm1I_cal(m,fdsm,Ndsm,N,GL,GT,alpha,beta)
%%%Z transform%%%
Zres=50;                   %z transform resolution
f=logspace(2,7.7,Zres);    %z transform frequency
omega=2*pi*f;              %radius frequency
z=tf('z',ts);

sdsm1I=(pi^2/(3*fdsm))*((2*sin(pi*f./(fdsm))).^(2*(m-1)));
%z domain DSM Architecture-I in DCO block
Hdsm1I=(2^(-Ndsm)*N*GL*(z-1)/GT)/((z-1)^2+GL*(alpha*(z-1)+beta*z));
%Transferfunction of DSM Architecture-I in DCO block

[magdsm1I,phasedsm1I]=bode(Hdsm1I,omega); %bode function
for i=1:1:Zres
phidsm1Ix(i)=magdsm1I(:,:,i);
end;
```

```
sdsm1Iout=sdsm1I.*(phidsm1Ix.^2); %phase noise caused by DSM Architecture-I in DCO block
phasenoise_dsm1I=10.*log10(sdsm1Iout); %DSM Architecture-I in DCO block phase noise in dB
```

Phase noise caused by DSM in DIV block architecture-I

Matlab file: "phasenoise_dsm2I_cal.m"

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%This function is used to estimate PLL phase noise caused by
%DSM noise Architecture-I in divider block
%please provide following parameters:
%DSM order (m), DSM sampling freuqncy (fdsm); DSM bit number (Ndsm)
%Divider value (N), loop gain (GL);
%low pass filter coeffients (alpha, beta)
%The output is DSM Architecture-I in DIV block phase noise in dB
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function phasenoise_dsm2I=phasenoise_dsm2I_cal(m,fdsm,Ndsm,N,GL,alpha,beta)
%%%Z transform%%%
Zres=50;                    %z transform resolution
f=logspace(2,7.7,Zres);     %z transform frequency
omega=2*pi*f;               %radius frequency
z=tf('z',ts);

sdsm2I=(pi^2/(3*fdsm))*((2*sin(pi*f./(fdsm))).^(2*(m-1)));
%z domain DSM Architecture-I in DIV block
Hdsm2I=(2^(-Ndsm)/N)/((z-1)^2+GL*(alpha*(z-1)+beta*z));
%Transferfunction of DSM Architecture-I in DIV block

[magdsm2I,phasedsm2I]=bode(Hdsm2I,omega); %bode function
for i=1:1:Zres
phidsm2Ix(i)=magdsm2I(:,:,i);
end;
sdsm2Iout=sdsm2I.*(phidsm2Ix.^2); %phase noise caused by DSM Architecture-I in DIV block
phasenoise_dsm2I=10.*log10(sdsm2Iout); %DSM Architecture-I in DIV block phase noise in dB
```

Phase noise caused by DSM in DCO block architecture-II

Matlab file: "phasenoise_dsm1II_cal.m"

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%This function is used to estimate PLL phase noise caused by
%DSM noise Architecture-II in DCO block
%please provide following parameters:
%DSM order (m), DSM sampling freuqncy (fdsm); DSM bit number (Ndsm)
%Divider value (N), loop gain (GL); DCO gain (kdco)
%low pass filter coeffients (alpha, beta)
%The output is DSM Architecture-II in DCO block phase noise in dB
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function phasenoise_dsm1II=phasenoise_dsm1II_cal(m,fdsm,Ndsm,GL,kdco,alpha,beta)
%%%Z transform%%%
Zres=50;                    %z transform resolution
f=logspace(2,7.7,Zres);     %z transform frequency
omega=2*pi*f;               %radius frequency
z=tf('z',ts);

sdsm1II=(pi^2/(3*fdsm))*((2*sin(pi*f./(fdsm))).^(2*(m-1)));
%z domain DSM Architecture-II in DCO block
Hdsm1II=(2^(-Ndsm)*kdco*(z-1))/((z-1)^2+GL*(alpha*(z-1)+beta*z));
%Transferfunction of DSM Architecture-II in DCO block

[magdsm1II,phasedsm1II]=bode(Hdsm1II,omega); %bode function
for i=1:1:Zres
phidsm1IIx(i)=magdsm1II(:,:,i);
end;
sdsm1IIout=sdsm1II.*(phidsm1IIx.^2); %phase noise caused by DSM Architecture-II
    in DCO
phasenoise_dsm1II=10.*log10(sdsm1IIout); %DSM Architecture-II in DCO phase noise
     in dB
```

Matlab example for Case 1

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%This program is used as an example to calculate phase noise of the
%architecture-I ADPLL in Chapter 3
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear;
clc;
%%%Please provide following parameters%%%
fref=45e6;      %reference frequency 45MHz
fout=1.5e9;     %output frequency 1.5GHz
kdco=577;       %DCO gain 577Hz/LSB
tauT=10e-12;    %TDC resolution
fdsm=45e6;      %DSM sampling frequency equals fref
```

```
m=1;           %DSM in DCO, the order is 1
Ndsm=5;        %DSM in DCO, the bit number is 5
alpha=2^2;     %DLF coefficient alpha
beta=2^(-6);   %DLF coefficient beta
k0ref=10^(-16.1); %Reference noise coefficient
k1ref=10^(-10.2); %Reference noise coefficient
k2ref=10^(-7.6);  %Reference noise coefficient
k3ref=10^(-6.1);  %Reference noise coefficient
k0dco=10^(-15.5); %DCO noise coefficient
k1dco=10^(-8.2);  %DCO noise coefficient
k2dco=10^(-4.2);  %DCO noise coefficient
k3dco=10^(0.9);   %DCO noise coefficient
Ndco=5;           %DCO dithering bit number
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

N=fout/fref;             %Divider value 33.33
GT=1/(2*pi*tauT*fref);   %TDC gain
GL=2*pi*GT*kdco/(N*fref);%Loop gain

%%%Z transform%%%
Zres=50;                  %z transform resolution
f=logspace(2,7.7,Zres);   %z transform frequency

%%%Call functions%%%
phasenoise_ref_cal(k0ref,k1ref,k2ref,k3ref,N,GL,alpha,beta);
phasenoise_dco_cal(k0dco,k1dco,k2dco,k3dco,GL,alpha,beta);
phasenoise_dqn_cal(Ndco,GL,GT,alpha,beta);
phasenoise_pfdI_cal(tauT,fref,fout,N,GL,GT,alpha,beta);
phasenoise_dsm1I_cal(m,fdsm,Ndsm,N,GL,GT,alpha,beta);

%%%Calculate total phase nosie%%%
stotal=10.^(phasenoise_ref/10)+10.^(phasenoise_dco/10) +10.^(phasenoise_dqn/10)
    +10.^(phasenoise_pfdI/10) +10.^(phasenoise_dsm1I/10);
logstotal=10.*log10(stotal);  %Total phase noise in dB

%%%Figure plot%%%
semilogx(f,phasenoise_ref,'r');
hold on;
semilogx(f,phasenoise_dco);
semilogx(f,phasenoise_dqn,'k');
semilogx(f,phasenoise_pfdI,'g');
semilogx(f,phasenoise_dsm1I,'p');
semilogx(f,logstotal);

%%%Plot measured phase noise%%%
Frequency=[1e3 2e3 4e3 7e3 1e4 2e4 4e4 7e4 1e5 2e5 4e5 7e5 1e6 2e6 4e6 7e6 1e7 2
    e7];
```

```
PhaseNoise_measured=[-92 -93 -95 -96 -96.3 -101 -110 -120 -124.8 -132 -138 -139
    -141.5 -143 -146 -147.5 -149.5 -152];
semilogx(Frequency,PhaseNoise_measured,'o');
grid;
```

Matlab example for Case 2

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%This program is used as an example to calculate phase noise of the
%architecture-II ADPLL in Chapter 3
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear;
clc;
%%%Please provide following parameters%%%
fref=48e6;          %reference frequency 48MHz
fout=5.376021831e9; %output frequency 5.376021831GHz
kdco=26e3;          %DCO gain 26KHz/LSB
tauT=15e-12;        %TDC resolution 15ps
fdsm=48e6;          %DSM sampling frequency equals fref
m=2;                %DSM in DCO, the order is 2
Ndsm=9;             %DSM in DCO, the bit number is 9
alpha=2^(-5);       %DLF coefficient alpha
beta=2^(-10);       %DLF coefficient beta
k0ref=10^(-14.7);   %Reference noise coefficient
k1ref=10^(-10.3);   %Reference noise coefficient
k2ref=10^(-8.3);    %Reference noise coefficient
k3ref=10^(-4.8);    %Reference noise coefficient
k0dco=10^(-13.8);   %DCO noise coefficient
k1dco=0;            %DCO noise coefficient
k2dco=10^(-0.2);    %DCO noise coefficient
k3dco=0;            %DCO noise coefficient
Ndco=9;             %DCO dithering bit number
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

N=fout/fref;            %FCW value
GT=1/(2*pi*tauT*fref);  %TDC gain
GL=2*pi*GT*kdco/(N*fref);%Loop gain

%%%Z transform%%%
Zres=50;                    %z transform resolution
f=logspace(2,7.7,Zres);     %z transform frequency

%%%Call functions%%%
phasenoise_ref_cal(k0ref,k1ref,k2ref,k3ref,N,GL,alpha,beta);
phasenoise_dco_cal(k0dco,k1dco,k2dco,k3dco,GL,alpha,beta);
phasenoise_dqn_cal(Ndco,GL,GT,alpha,beta);
```

```
phasenoise_pfdII_cal(tauT,fref,fout,N,GL,GT,alpha,beta);
phasenoise_dsm1II_cal(m,fdsm,Ndsm,N,GL,GT,alpha,beta);

%%%Calculate total phase nosie%%%
stotal=10.^(phasenoise_ref/10)+10.^(phasenoise_dco/10) +10.^(phasenoise_dqn/10)
    +10.^(phasenoise_pfdII/10) +10.^(phasenoise_dsm1II/10);
logstotal=10.*log10(stotal);  %Total phase noise in dB

%%%Figure plot%%%
semilogx(f,phasenoise_ref,'r');
hold on;
semilogx(f,phasenoise_dco);
semilogx(f,phasenoise_dqn,'k');
semilogx(f,phasenoise_pfdII,'g');
semilogx(f,phasenoise_dsm1II,'p');
semilogx(f,logstotal);

%%%Plot measured phase noise%%%
Frequency=[2e3 4e3 7e3 1e4 2e4 4e4 7e4 1e5 2e5 4e5 7e5 1e6 2e6 4e6 7e6 1e7];
PhaseNoise_measured=[-95 -95.5 -96 -96.9 -96.5 -96 -93 -90.7 -97 -104 -110
    -116.4 -122 -128 -134 -137];
semilogx(Frequency,PhaseNoise_measured,'o');
grid;
```

A.2 Matlab Code for Phase Noise Calculation

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%This function is used to calculate PLL phase noise.
%Please provide following parameters:
%DCO output signal(DCOout);
%DCO frequency (fdco);Sampling frequency (fs); Settling time (tset);
%The output of this function is phase noise.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function phasenoise=phasenoise_cal(DCOout,fdco,fs,tset)
DCOout_length=length(DCOout);                          %DCOout length;
DCOout_settle=DCOout(ceil(tset*fs+1):DCOout_length);   %DCO data when it is
    locked
DCOout_settle_length=length(DCOout_settle);            %length of DCOout_settle
DCOout_calcul=DCOout_settle(1:2^(floor(log2(DCOout_settle_length))));
```

```
%2^N data is used to calculte phase noise

%%%Fourier Transform%%%
Per=1/fs;                          %Sampling period;
Ndco=length(DCOout_calcul);        %length of DCO data in calculation
Tend=(Ndco-1)*Per;                 %end Time
fftsig=fft(DCOout_calcul);         %fourier transform
mag_fftsig=abs(fftshift(fftsig)); %magnitude
fre=(0:Ndco-1)*fs/Ndco-fs/2;       %frequency

phasenoise=20*log10(mag_fftsig/max(mag_fftsig))+20*log10(Tend);
%phase noise in log scale

%%%Figure plot%%%
semilogx(fre-fdco,phasenoise);
grid;
```

A.3 Matlab Code for Phase Noise to Jitter Conversion

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%This program is used to calculate RMS jitter requirement based on
%phase noise requirement
%Please provide parameters: loop bandwidth (BW);
%Output frequency (fout);offset freuqncy (fre_os);
%Phase noise at offset frequency (pn_dBc) in dBc/Hz.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function jitter_conv=phasenoise_to_jitter(BW,fout,fre_os,pn_dBc)
%Calculate PN1
if fre_os<=BW
PN1=pn_dBc;
elseif fre_os<=10*BW
PN1=pn_dBc+20*log10(fre_os/BW);
elseif fre_os<=100*BW
PN1=pn_dBc+20+10*log10(fre_os/(10*BW));
else
PN1=pn_dBc+30;
end;
PN2=PN1-20;   %Calculate PN2
PN3=PN1-30;   %Calculate PN3
```

```
22
23 A1=PN1+10*log10(BW-0);                    %Calculate A1
24 A2=(PN1+PN2)/2+10*log10(10*BW-BW);       %Calculate A2
25 A3=(PN2+PN3)/2+10*log10(100*BW-10*BW);   %Calculate A3
26 A4=PN3+10*log10(2*fout-100*BW);          %Calculate A4
27
28 jitter_conv=(sqrt(2*10^(A1/10))+sqrt(2*10^(A2/10))+sqrt(2*10^(A3/10))+sqrt
       (2*10^(A4/10)))/(2*pi*fout);  %Calculate JMS jitter
```

A.4 Matlab Code for Jitter Calculation from DCO Control Code

```
1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  %This program is used to calculate RMS jitter based on
3  %DCO control code
4  %Please provide parameters:
5  %DCO control code (Dcode),Sampling frequency (fs);
6  %Number of code in one calculation period (Np), locking time (TL).
7  %DCO gain (Kdco) and output frequency (fout).
8  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
9
10 function RMS_jitter=RMS_jitter_cal(Dcode, fs, Np, TL, Kdco,fout)
11 Dcode_length=length(Dcode);                    %DCO control code length;
12 Dcode_settle=Dcode(ceil(TL*fs+1):Dcode_length); %DCO control code when it is
       locked
13
14 if Np>=length(Dcode_settle)
15 %Make sure the number of DCO control code is larger or equal to Np
16 Dcode_cal=Dcode_settle(1:Np); %DCO control code for calculation
17 STD_Dcode=std(Dcode_cal);     %Standard deviation of DCO control code for
       calculation
18 RMS_jitter=STD_Dcode*Kdco/(2*pi*fout^2); %RMS jitter result
19 else disp(' Please provide more DCO control codes.');
20 end;
```

Bibliography

[1] L. H. Lu, et al., "A wide tuning-range CMOS VCO with a differential tunable active inductor", *in IEEE Trans. Microwave Theory Tech.*, vol. 54, no. 9, pp. 3462-3468, Sep. 2006.

[2] Wei Deng, Kenichi Okada, and Akira Matsuzawa., "A 25MHz-6.44GHz LC-VCO Using a 5-port Inductor for Multiband Frequency Generation", *in 2011 IEEE RFIC Virtual Journal Radio Frequency Integrated Circuits Symposium (RFIC)*, pp. 1-4, Jun. 2011.

[3] B. Saeidi, Aliso Viejo, J. Cho, G. Taskov, A. Paff., "A wide-range VCO with optimum temperature adaptive tuning", *in 2010 IEEE Radio Frequency Integrated Circuits Symposium (RFIC)*, pp. 337-340, May. 2010.

[4] P. Vaananen, N. Mikkola, P. Helio., "VCO Design With On-Chip Calibration System", *in 2006 IEEE Transactions on Circuits and Systems I: Regular Papers*, pp. 2157-2166, Oct. 2006.

[5] P. Vaananen, N. Mikkola, P. Helio., "Low-power high-tuning range CMOS ring oscillator VCOs", *in 2008 IEEE International Conference on Semiconductor Electronics, 2008. ICSE 2008.*, pp. 40-44, Nov. 2008.

[6] Heng-Yu Jian, Zhiwei Xu, Yi-Cheng Wu, and Mau-Chung Frank Chang., "A Fractional-N PLL for Multiband (0.8-6GHz) Communications Using Binary-Weighted D/A Differentiator and Offset-Frequency Δ-Σ Modulator", *in IEEE Journal of Solid-State Circuits*, vol. 45, no. 4, pp. 768-780, Apr. 2010.

[7] Xiaodong Liu, Wenfeng Lou, Liyuan Liu and Nanjian Wu., "A multi-band fully differential fractional-N PLL for wideband reconfigurable wireless communication", *IEEE International Conference of Electron Devices and Solid-State Circuits (EDSSC)*, pp. 1-2, Jun. 2013.

[8] K. Ueda, T. Uozumi, R. Endo, T. Nakamura, et al., "A digital PLL with two-step closed-locking for multi-mode/multi-band SAW-less transmitter", *in IEEE Custom Integrated Circuits Conference (CICC)*, pp. 1-4, Sept. 2012.

[9] N. Joram, B. Lindner, J. Wagner and F. Ellinger., "Integrated multi-band fractional-N PLL for FMCW radar systems at 2.4 and 5.8GHz", *in 10th Conference on Ph.D. Research in Microelectronics and Electronics (PRIME)*, pp. 1-4, July, 2014.

[10] Ja-Yol Lee, Kwidong Kim, Seung-Chul Lee, Jong-Kee Kwon, et al., "A 9.1-to-11.5-GHz Four-Band PLL for X-Band Satellite & Optical Communication Applications", *in IEEE Radio Frequency Integrated Circuits (RFIC) Symposium*, pp. 233-236, June. 2007.

[11] S. Chakraborty, V. Parikh, S. Sankaran, T. Motos, et al., "An ultra low power, reconfigurable, multi-standard transceiver using fully digital PLL", *in 2013 Symposium on VLSI Circuits (VLSIC)*, pp. C148-C149, June 2013.

[12] B. Jiang, T. Xia, "ADPLL Variables Determinations based on Phase Noise, Spur and Locking Time", *Proceedings of IEEE System-on-Chip Conference (SOCC)*, 2012.

[13] Joonsuk Lee and Beomsup Kim, "A low-noise fast-lock phase-locked loop with adaptive bandwidth control",*IEEE Journal of Solid-State Circuits*, pp. 1137-1145. Aug. 2002.

[14] M. Ouda, E. Hegazi, H. F. Ragai., "Digital on-chip phase noise measurement", *in 4th International Design and Test Workshop (IDT)*, pp. 1-5, Nov. 2009.

[15] Socrates D. Vamvakos, Vladimir Stojanović, Jared L. Zerbe, Carl W. Werner, Don Draper, Borivoje Nikolić., "PLL On-Chip Jitter Measurement: Analysis and Design", *in IEEE Symposium on VLSI Circuits Digest of Technical*

Papers, 2006.

[16] Khalil, W. ; Bakkaloglu, B. ; Kiaei, S. "A Self-Calibrated On-Chip Phase-Noise Measurement Circuit With -75 dBc Single-Tone Sensitivity at 100 kHz Offset", *in IEEE Journal of Solid-State Circuits*, vol. 42, issue. 12, pp. 2758-2765, Dec. 2007.

[17] Staszewski, R.B., Leipold, D., Chih-Ming Hung and Balsara, P.T., "TDC-based frequency synthesizer for wireless applications", *IEEE Radio Frequency Integrated Circuits (RFIC) Symposium*, Aug. 2004, pp. 215-218.

[18] Ponton, D., Knoblinger, G., Roithmeier, A., Cernoia, F., Tiebout, M., Fulde, M. and Palestri, P., "LC-VCO in the 3.3- to 4-GHz Band Implemented in 32-nm Low-Power CMOS Technology", *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 58, issue. 8, pp. 467-471, Jul. 2011.

[19] Rout, P.K., Nanda, U.K., Acharya, D.P. and Panda, G., "Design of LC VCO for optimal figure of merit performance using CMODE", *2012 1st International Conference on Recent Advances in Information Technology (RAIT)*, pp. 761-764, Mar. 2012.

[20] Hyunwon Moon, Sungweon Kang, Youn Tae Kim and Kwyro Lee, "A fully differential LC-VCO using a new varactor control structure", *IEEE Microwave and Wireless Components Letters*, pp. 410-412, Sept. 2004.

[21] Liu, P., Upadhyaya, P., Jung, J., Heo, D. et., "Dynamically switched low-phase-noise LC VCO with harmonic filtering", *Electronics Letters*, vol. 47, issue. 14, pp. 792-793, Jul. 2011.

[22] Kim, D.D., Jonghae Kim and Choongyeun Cho, "The process variability of a V-band LC-VCO in 65nm SOI CMOS", *IEEE Radio Frequency Integrated Circuits Symposium, 2008. RFIC 2008.*, pp.127-130, Jun. 2008.

[23] Seong-Mo Yim and Kenneth K. O., "Switched resonators and their applications in a dual-band monolithic CMOS LC-tuned VCO", *in IEEE Transactions on Microwave Theory and Techniques*, pp. 74-81, 2006.

[24] Z. Safarian and H. Hashemi, “A 1.3-6 GHz triple-mode CMOS VCO using coupled inductors", *in IEEE Custom Integrated Circuits Conference*, pp. 69-72, Sep. 2008.

[25] Bo Jiang and Tian Xia, “A quad-mode DCO for multi-standard communication application", *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 24-27, May 2015.

[26] J. Kim, et al., “A 44 GHz differentially tuned VCO with 4 GHz tuning range in 0.12 m SOI CMOS", *in IEEE International Solid-State Circuits Conference Digest of Technical Papers*, pp. 416-417, Feb. 2005.

[27] Z. Li and K. K. O, “A 1-V low phase noise multi-band CMOS voltage controlled oscillator with switching inductors and capacitors", *in RFIC Symp., Fort Worth, TX,* pp. 467-470. Jun. 2004.

[28] Kratyuk, V., Hanumolu, P.K., Un-Ku Moon, Mayaram, K., “A Design Procedure for All-Digital Phase-Locked Loops Based on a Charge-Pump Phase-Locked-Loop Analogy", *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 54 pp. 247-251, Mar. 2007.

[29] M. H. Perrott, M. D. Trott, C. G. Sodini, “A modeling approach for fractional-N frequency synthesizers allowing straightforward noise analysis", *in IEEE Journal of Solid-State Circuits*, vol. 37, pp. 1028-1038, Nov. 2002.

[30] B. Jiang, T. Xia, G. Wang, “PLL low pass filter design considering unified specification constraints", *in Analog Integrated Circuits and Signal Processing*, pp. 1-8, 2014.

[31] F. Herzel, S. A. Osmany J. C. Scheytt, “Analytical Phase-Noise Modeling and Charge Pump Optimization for Fractional-N PLLs", *in IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, pp. 1914-1924, Aug. 2010.

[32] A. Hajimiri, S. Limotyrakis, T. H. Lee, “Jitter and Phase Noise in Ring Oscillators", *in IEEE Journal of Solid-State Circuits,* vol. 34, no. 6, pp. 790-804, Jun. 1999.

[33] T. Xia, “Analytical Model for Spread-Spectrum Clock Generator Circuit Characterization",*in International Journal of Circuit Theory and Applications*, 2010.

[34] Nicola Da Dalt, “A Design-Oriented Study of the Nonlinear Dynamics of Digital Bang-Bang PLLs", *in IEEE Transactions on Circuits and Systems-I: Regular Papers*, vol. 52, no. 1, pp. 21-31, Jan. 2005.

[35] Staszewski, R.B., Leipold, D., Chih-Ming Hung and Balsara, P.T., “TDC-based frequency synthesizer for wireless applications", *IEEE Radio Frequency Integrated Circuits (RFIC) Symposium*, pp. 215-218, Aug. 2004.

[36] R. B. Staszewski, P. T. Balsara, “Phase-Domain All-Digital Phase-Locked Loop", *in IEEE Transactions on Circuits and Systems-II: Express Briefs*, vol. 52, no. 3, pp. 159-163, Mar. 2005.

[37] S. Mendel C. Vogel, “A z-domain model and analysis of phase-domain all-digital phase-locked loops", Norchip, pp. 1-6, Nov. 2007.

[38] Ioannis L. Syllaios, Poras T. Balsara, R. B. Staszewski, “Time-Domain Modeling of a Phase-Domain All-Digital Phase-Locked Loop for RF Applications", *in IEEE Custom Intergrated Circuits Conference (CICC)*, pp. 861-864, 2007.

[39] Ioannis L. Syllaios, Robert Bogdan Staszewski, Poras T. Balsara, “Time-Domain Modeling of an RF All-Digital PLL", *in IEEE Transactions on Circuits and Systems-II: Express Briefs*, vol. 55, no. 6, pp. 601-605, Jun. 2008.

[40] Ioannis L. Syllaios, Poras T. Balsara, “Multi-Clock Domain analysis and Modeling of All-Digital Frequency Synthesizers", *in IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 153-156, May. 2011.

[41] Ioannis L. Syllaios, Poras T. Balsara, “Linear Time-Variant Modeling and Analysis of All-Digital Phase-Locked Loops", *in IEEE Transactions on Circuits and Systems-I: Regular Papers*, vol. 59, no. 11, pp. 2495-2506, Nov. 2012.

[42] Paolo Madoglio, Marco Zanuso, Salvatore Levantino, Carlo Samori, Andrea L. Lacaita, “Quantization Effects in All-Digital Phase-Locked Loops", *in IEEE*

Transactions on Circuits and Systems-II: Express Briefs, vol. 54, no. 12, pp. 1120-1124, Dec. 2007.

[43] R. B. Staszewski, C. Hung, N. Barton, M. Lee, D. Leipold, "A Digitally Controlled Oscillator in a 90 nm Digital CMOS Process for Mobile Phones", *in IEEE Journal of Solid-state Circuits*, vol. 40, no. 11, pp. 2203-2211, Nov. 2005.

[44] Dram De Muer, Michiel S. J. Steyaert, "On The Analysis of $\Delta\Sigma$ Fractional-N Frequency Synthesizers for High-Spectral Purity", *in IEEE Transactions on Cricuits and Systems-II: Analog and Digital Signal Processing*, vol. 50, pp. 784-793, Nov. 2003.

[45] Steve Williams and Tony Caviglia, Simulating PLL reference spurs, in www.rfdesing.com, Mar, 2006.

[46] Tasca, D., Zanuso, M., Marzin, G., Levantino, S., Samori, C., Lacaita, A. L, "A 2.9-4.0GHz Fractional-N Digital PLL With Bang-Bang Phase Detector and 560-fsrms Integrated Jitter at 4.5-mW Power", *in IEEE Journal of Solid-State Circuits*, vol. 46, no. 12, pp. 2745-2758, Dec. 2011.

[47] Julie R. Hu, Richard C. Rubyy, Brian P. Otis, "A 1.5GHz 0.2psRMS Jitter 1.5mW Divider-less FBAR ADPLL in 65nm CMOS", *in IEEE Custom Integrated Circuits Conference (CICC)*, pp. 1-4, Sep. 2012.

[48] N. Pavlovic, J. Bergervoet., "A 5.3GHz digital-to-time-converter-based fractional-N all-digital PLL", *in 2011 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, pp. 54-56, 2011.

[49] V. Kratyuk, P. K. Hanumolu, Un-Ku Moon K. Mayaram, "A Design Procedure for All-Digital Phase-Locked Loops Based on a Charge-Pump Phase-Locked-Loop Analogy", *in IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 54, pp. 1-6, Mar. 2007.

[50] Wen Li, Jason Meiners, "Introduction to phase-locked loop system modeling", *in Texas Instruments Incorporated Analog Applications Journal*, pp. 5-11, May. 2000.

[51] WF5028 series 0.8V operation Crystal Oscillator Mudule ICs, Seiko NPC Corporation. Jan. 2008.

[52] Yuan-Kai Wu, Cheng-Chen Liu, Jhin-Fang Huang, "A Dual-Band CMOS Voltage-Controlled Oscillator Implemented With Dual-Resonance LC Tank", *in IEEE Microwave and Wireless Components Letters*, vol. 19, pp. 816-818, Dec. 2009.

[53] Walt Kester, "Converting Oscillator Phase Noise to Time Jitter", *in MT-008 TUTORIAL*, Analog Device, pp. 1-10.

[54] "Agilent's Phase Noise Measurement Solutions", Agilent Technologies, Inc. 2013.

[55] Kuendiger, T., Fang Chen, MacEachern, L. and Mahmoud, S., "A novel digitally controlled low noise ring oscillator", *in IEEE International Symposium on Circuits and Systems*, May 2008, pp. 1016-1019.

[56] Alsharef, A. A., Marvast, M.J.T., Ali, M.A.M. and Sanusi, H., "A high speed and low power voltage controlled ring oscillator for phase locked loop circuits", *in 2011 IEEE Regional Symposium on Micro and Nanoelectronics (RSM)*, pp. 132-134, Sept. 2011.

[57] Patro, B.S., Panigrahi, J.K. and Mandal, S.K., "A 6-17 GHz linear wide tuning range and low power ring oscillator in 45nm CMOS process for electronic warfare", *in 2012 International Conference on Communication, Information & Computing Technology (ICCICT)*, pp. 1-4, Oct. 2012.

[58] Arakali, A., Gondi, S. and Hanumolu, P.K., "Analysis and Design Techniques for Supply-Noise Mitigation in Phase-Locked Loops", *IEEE Transaction on Circuit and System I: regular papers*, Nov 2010, vol. 57, pp. 2880-2889.

[59] Sethapong Limkumnerd and Duangrat Eungdamrong., "Mathematical Phase Noise Model For A Phase-locked-loop", ICCAS, 2005.

[60] Hu, J.R., Ruby, R.c. and Otis, B.P, "A 1.56GHz wide-tuning all digital FBAR-based PLL in 0.13μm CMOS", *IEEE 2010 Custom Intergrated Circuits*

Conference, pp. 1-4, Sep. 2010.

[61] Tasca, D., Zanuso, M., Marzin, G., Levantino, S., Samori, C. and Lacaita, A. L., "A 2.9-4.0-GHz Fractional-N Digital PLL With Bang-Bang Phase Detector and 560-fsrms Integrated Jitter at 4.5-mW Power", *IEEE Journal of Solid-State Circuits*, Aug. 2011.

[62] Jingcheng Zhuang, Qingjin Du, Kwasniewski, T., "A 4GHz Low Complexity ADPLL-based Frequency Synthesizer in 90nm CMOS", *IEEE Custom Integrated Circuits Conference*, pp.543-546, Sept. 2007.

[63] Liangge Xu, Saska Lindfors, Kari Stadius and Ryynänen, "A 2.4-GHz Low-Power All-Digital Phase-Locked Loop", *IEEE 2009 Custom Intergrated Circuits Conference*, pp. 331-334, 2009.

[64] Kratyuk, V., Hanumolu, P.K., Un-Ku Moon, Mayaram, K., "A Design Procedure for All-Digital Phase-Locked Loops Based on a Charge-Pump Phase-Locked-Loop Analogy?, *in IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 54 pp. 247-251, Mar. 2007.

[65] Wen Li and Jason Meiners, "Introduction to phase locked loop system modeling", *Analog Applications Journal*, May, 2000.

[66] "Phase Noise / Jitter In Crystal Oscillators", Rakon, 2009.

[67] Lin, T.Y., Yu, T.Y., Ke, L.W. and Dehng, G.K., "A low-noise VCO with a constant KVCO for GSM/GPRS/EDGE applications", *in IEEE Radio Frequency Integrated Circuits Symposium*, pp. 387-390, Apr. 2008.

[68] Mehrotra, A., "Noise analysis of phase-locked loops", *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 49, pp. 1309-1316, Nov. 2002.

[69] Staszewski, R.B. et al.,"All-digital PLL and transmitter for mobile phones", *in IEEE Journal of Solid-State Circuits*, vol. 40, no. 12, pp. 2469-2482, Dec. 2005.

[70] Minjae Lee, Heidari, M.E., Abidi, A.A., "A Low-Noise Wideband Digital Phase-Locked Loop Based on a Coarse Fine Time-to-Digital Converter With Subpicosecond Resolution", *in IEEE Journal of Solid-State Circuits*, vol. 44, no. 10, pp. 2809-2816, Oct. 2009.

[71] M. E. s. Elrabaa, "A portable high-frequency digitally controlled oscillator (DCO)", *in Integration, the VLSI Journal*, Nov. 2013.

[72] Liangge Xu. and Lindfors, S., "A digitally controlled 2.4-GHz oscillator in 65-nm CMOS", Norchip, Nov. 2007, pp. 1-6.

[73] Fanori, L., Liscidini, A., Castello, R., "3.3GHz DCO with a Frequency Resolution of 150Hz for All-Digital PLL", *in 2010 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, pp. 48-49, Feb. 2010.

[74] Daegyu Lee, Jincheol Yoo, Kyusun Choi and Jahan Ghaznavi, "Fat tree encoder design for ultra-high speed flash A/D converters", *The 2002 45th Midwest Symposium on in Circuits and Systems, 2002. MWSCAS-2002.*, Aug. 2002.

[75] "Estimating Period Jitter from Phase Noise", *Silicon Labs*, AN279, pp. 1-7.

[76] "Fractional-N Frequency Synthesis", *Mathworks*.