A Low Complexity Digital Phase-Locked Loop Based Frequency Synthesizer

By

Jingcheng Zhuang

A thesis submitted to

The Faculty of Graduate Studies and Research

In partial fulfillment of

The requirements for the degree of

Doctor of Philosophy

Ottawa-Carleton Institute for Electrical and Computer Engineering

Department of Electronics

Carleton University

Ottawa, Ontario

Canada

December 2007

Your file Votre référence
ISBN: 978-0-494-36801-5
Our file Notre référence
ISBN: 978-0-494-36801-5

The undersigned recommend to

the Faculty of Graduate Studies and Research

acceptance of this thesis

"A Low Complexity Digital Phase-Locked Loop Based Frequency Synthesizer"

Submitted by Jingcheng Zhuang

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Thesis Supervisor

Prof. Tad Kwasniewski

Chair, Department of Electronics

Carleton University

December 2007

The information used in this thesis comes in part from the research program of Dr. Tad Kwasniewski and his associates in the VLSI in Communications group. The research results appearing in this thesis represent an integral part of the ongoing research program. All research results in this thesis including tables, graphs and figures but excluding the narrative portions of the thesis are effectively incorporated into the reach program and can be used by Dr. Kwasniewski and his associates for educational and research purpose, including publication in open literature with appropriate credits. The matters intellectual property may be pursued cooperatively with Carleton University and Dr. Kwasniewski and where and when appropriate.

Abstract

This dissertation presents a proposed low-complexity digital PLL and a digitally-controlled oscillator with an enhanced frequency resolution for frequency synthesis applications. The basic operation of the conventional PLL-based frequency synthesizers is first briefly reviewed, followed by the literature review of some reported digital PLLs. A low-complexity digital PLL is thus proposed, including the system architecture and implementations of its sub-blocks. In the proposed digital PLL, the complex digital loop filter used in many reported digital PLLs is replaced by a simple logical decision circuit, and digital encoders are avoided to significantly reduce the hardware complexity. A novel digital tuning scheme for the digitally-controlled oscillator is proposed to achieve small frequency tuning steps and to improve the matching accuracy between LSB/MSB tuning banks so that the overlaps between LSB/MSB banks can be eliminated.

The loop behavior of the proposed DPLL is analyzed theoretically and its unique noise behavior, automatic adaptation to different reference phase noise levels, is investigated, together with the determination of the design parameters. To confirm the theoretical analysis, behavioral simulations using SimuLink and the event-driven technique were done to observe the locking process and obtain the phase noise performance.

To further prove the feasibility of proposed digital PLL, two test chips, the proposed high-resolution digitally controlled oscillator and the low-complexity digital PLL, were implemented and fabricated. The phase noise performance and the frequency tuning characteristic of the digitally controlled oscillator were measured. The phase noise tracking and compensation characteristics were observed from the digital PLL measurements, which well agree with both the theoretical analysis and the behavioral simulations.

Table of Contents

Chapter 1: Introduction **1**

1.1 Introduction 1

1.2 Contributions 4

1.3 Document organization 6

Chapter 2: Review of PLL-based frequency synthesizers **7**

2.1 Introduction 7

2.2 Basic operation of a PLL-based frequency synthesizer 7

2.2.1 Operation principles of PLL 7

2.2.2 PLL frequency synthesis 10

2.3 Performance parameters of a frequency synthesizer 12

2.3.1 Phase noise 12

2.3.2 Spurious power level 15

2.3.3 Output period jitter 16

2.3.4 Operating frequency range 16

2.3.5 Frequency step 17

2.3.6 Channel switching time 17

2.3.7 Power consumption 18

2.4 Analog and digital PLL design trade-offs 18

2.5 Review of digital PLL-based frequency synthesizers 20

2.5.1 A novel all-digital PLL with software adaptive filter 21

2.5.2 A compact, low power low jitter digital PLL 23

2.5.3 An all-digital phase-locked loop for high-speed clock generation 24

2.5.4 A fast lock digital phase-locked loop 27

2.5.5 A fast lock all-digital PLL for frequency multiplication by 4 to 1022 29
2.5.6 An all-digital PLL clock multiplier 30
2.5.7 A PVT tolerant, self-calibrated digital PLL.....31
2.5.8 A phase-domain all-digital phase-locked loop 34
2.6 Summary of the literature review 35
Chapter 3: A proposed digital PLL-based frequency synthesizer 38
3.1 System architecture of the proposed digital PLL.....38
3.2 The proposed digitally-controlled oscillator.....42
3.3 The proposed digital PFD 45
3.4 The proposed decision circuit 47
3.5 Summary 49
Chapter 4: Behavioral analysis and simulation of the proposed DPLL 50
4.1 Introduction.....50
4.2 Mathematical analysis of the loop behavior 51
4.2.1 The loop bandwidth of the proposed DPLL 51
4.2.2 Phase noise created by the proportional path quantization. 55
4.2.3 DCO phase noise attenuation.....61
4.2.4 Optimum loop bandwidth of the digital PLL.....62
4.2.5 The large signal loop response and the loop stability 66
4.3 The determination of the design parameters.....68
4.3.1 The proportional path frequency step size 69
4.3.2 The updating period and the integration step size 71
4.4 Simulink modeling of the DPLL 73
4.4.1 The model of the non-linear digital PFD 73
4.4.2 The model of the frequency divider.....74

4.4.3 The model of the DCO 75
4.4.4 The model of the decision circuit 76
4.4.5 The top-level Simulink model of the DPLL 78
4.4.6 Simulink simulation results 78
4.5 Event-Driven simulation technique 79
4.5.1 The concept of event-driven technique 80
4.5.2 Event-drive simulation engine 81
4.5.3 Event dispatcher 82
4.5.4 Event starter 83
4.5.5 Modeling of building blocks 83
4.6 Event-driven modeling of the proposed DPLL 84
4.6.1 The model of the reference clock source 84
4.6.2 The model of the digital phase and frequency detector 85
4.6.3 The model of the frequency divider 86
4.6.4 The model of the decision circuit 86
4.6.5 The model of the digitally-controlled oscillator 87
4.7 Event-driven simulation results 87
4.8 Phase noise performance simulation 91
4.8.1 The simulation configurations 91
4.8.2 Simulated phase noise performance of the proposed DPLL 91
4.9 Summary 96
Chapter 5: CMOS implementation of the proposed DPLL 98
5.1 Introduction 98
5.2 The DCO implementation 99
5.2.1 The implementation of the oscillator 99
5.2.2 The digital to capacitance converter 99

5.2.2.1 High resolution varactor array 100

5.2.2.2 The mismatch of the varactor 106

5.2.3 The varactor driver 108

5.2.4 The varactor tuning circuit 109

5.2.4.1 The multi-input D flip-flop 109

5.2.4.2 The one-hot coded tuning circuit 110

5.2.4.3 The thermometer coded tuning circuit 112

5.2.4.4 The phase tuning path 112

5.3 The implementation of the digital phase detector 113

5.4 Implementation of the frequency decision circuit 116

5.5 Implementation of the frequency divider 118

5.5.1 The divide-by-2, 3 frequency divider 118

5.5.2 The DCO frequency divider 119

5.5.3 Frequency divider for frequency decision clock generation 120

5.6 The digital serial data interface 122

5.7 Summary 123

Chapter 6: Performance evaluation 124

6.1 Introduction 124

6.2 The test bench for the DCO measurement 124

6.3 The measurement results of the DCO 126

6.4 The test bench for the DPLL measurement 130

6.5 The measurement results of the DPLL 132

6.5.1 The phase noise performance 134

6.5.2 The spur performance 137

6.6 The phase noise of the DPLL with the low-noise DCO 138

6.7 Conclusions 140

Chapter 7: Conclusions 142

7.1 Summary 142

References 144

List of Abbreviations

AM	Amplitude modulation
BPF	Band-pass filter
LF	Loop filter
CMOS	Complementary metal oxide semiconductor
CP	Charge pump
DCO	Digitally controlled oscillator
DDS	Direct digital synthesis
DLF	Digital loop filter
DLL	Delay-locked loop
DNL	Differential non-linearity
DPFD	Digital phase/frequency detector
DPLL	Digital phase locked loop
FM	Frequency modulation
LF	Loop filter
LPF	Low-pass filter
LSB	Least significant bit
MSB	Most significant bit
PD	Phase detector
PFD	Phase/frequency detector
PLL	Phase-locked loop
PM	Phase modulation

TDC	Time to digital converter
T2D	Time to digital converter
VCDL	Voltage-controlled delay line
VCO	Voltage-controlled oscillator

List of Symbols


Adder


Current source


Voltage-controlled oscillator

Gain stage

Frequency divider

Unit Delay


Capacitor


Varactor


Crystal


Ground

Resistor


Inductor

Integrator


Inverter


Delay cell or output buffer

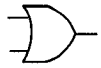

VDD

NMOS transistor

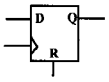
PMOS transistor

Quantizer

2-input NAND gate



2-input OR gate



D flip-flop

2-input AND gate

Switch

Two-level Quantizer

List of Tables

TABLE 2.1: A mathematical representations of analog/digital DPLLs 19

TABLE 4.1: The minimum frequency step in the fractional path 71

TABLE 4.2: The Simulink simulation parameters of the DPLL 73

TABLE 4.3: The cases of the noise performance simulation 92

TABLE 5.1: The frequency division ratio of the DCO frequency divider 120

TABLE 5.2: The division ratio of the frequency divider for the frequency decision... 121

TABLE 6.1: Performance comparison with recently reported DCOs 130

TABLE 6.2: The loop parameters 134

TABLE 6.3: The loop parameters for the DPLL with low noise DCO 139

List of Figures

Figure 2.1: The basic structure of a phase-locked loop 8

Figure 2.2: The linear model of a phase-locked loop 9

Figure 2.3 A PLL-based frequency synthesizer 10

Figure 2.4 A fractional-N PLL-based frequency synthesizer 11

Figure 2.5 The spectrum of an ideal signal and a practical signal 12

Figure 2.6 The spectrum of a sinusoid FM signal 14

Figure 2.7 The spectrum of a FM signal 14

Figure 2.8 The power spectrum of a frequency source with spurs 15

Figure 2.9 Illustration of the period jitter 16

Figure 2.10 The principle of a PLL-based frequency synthesizer 18

Figure 2.11 The block diagram of a digital PLL 20

Figure 2.12 An all-digital PLL with software adaptive filter 21

Figure 2.13 The DCO for the all-digital PLL with software adaptive filter 23

Figure 2.14 A compact, low power low-jitter digital PLL 23

Figure 2.15 An all-digital phase-locked loop for high-speed clock generation 25

Figure 2.16 The DCO of an ADPLL for high-speed clock generation 26

Figure 2.17 The fine tuning stage of the DCO 26

Figure 2.18 An all-digital PLL with small DCO hardware and fast phase lock 27

Figure 2.19 A digitally controlled delay cell for the DCO 28

Figure 2.20 An DPLL with matched operating resolution of PFD and DCO 29

Figure 2.21 An all-digital PLL clock multiplier 30

Figure 2.22 The PFD for an all-digital PLL clock multiplier 31

Figure 2.23 A PVT tolerant self-calibrated digital PLL 32

Figure 2.24 The DPFD of the PVT tolerant self-calibrated digital PLL 33
Figure 2.25 Phase-domain ADPLL-based frequency synthesizer 34
Figure 2.26 Classification of DCOs 35
Figure 3.1 A typical digital PLL-based frequency synthesizer 38
Figure 3.2 A proposed digital PLL-based frequency synthesizer 39
Figure 3.3 The basic principle of the varactor tuning circuit 40
Figure 3.4 The architecture of the proposed DCO 42
Figure 3.5 Tuning characteristics of a PMOS transistor 43
Figure 3.6 The frequency tuning of the proposed DCO 44
Figure 3.7 The phase tuning of the proposed DCO 45
Figure 3.8 The quantization of a linear T2D 46
Figure 3.9 The proposed DPFD 46
Figure 3.10 An optional T2D in the proposed DPFD 47
Figure 4.1 Mathematical model of the proposed digital PLL 51
Figure 4.2 Full phase attenuation of the proposed digital PLL 53
Figure 4.3 partial phase attenuation of the proposed digital PLL 54
Figure 4.4 False phase attenuation of the proposed digital PLL 54
Figure 4.5 The noise model for analysis of the proportional path quantization 56
Figure 4.6 Phase noise caused by the proportional path activities 58
Figure 4.7 The DCO phase noise and the quantization phase noise 61
Figure 4.8 The phase noise profile of an oscillator 62
Figure 4.9 The optimum loop bandwidth of a conventional PLL 63
Figure 4.10 The noise model of the PLL with DCO/reference phase noise 64
Figure 4.11 The linear combination of the DCO and reference phase noise 65
Figure 4.12 The PLL phase noise for different reference phase noise profiles 65

Figure 4.13 The phase of the open-loop PLL transfer function 68
Figure 4.14 The phase of the open-loop digital PLL 69
Figure 4.15 The noise model of the DPLL for time-domain analysis 70
Figure 4.16 Determination of the decision period of the DPLL 72
Figure 4.17 The Simulink model of the digital PFD 74
Figure 4.18 The Simulink model of the frequency divider 75
Figure 4.19 The Simulink model of the DCO 76
Figure 4.20 The Simulink model of the up/down counter 76
Figure 4.21 The Simulink model of the frequency decision circuit 77
Figure 4.22 The Simulink model of the phase/frequency decision circuit 77
Figure 4.23 The Simulink model of the digital PLL 78
Figure 4.24 The Simulink simulation result of the digital PLL 79
Figure 4.25 The concept of the event driven program 80
Figure 4.26 The principle of the event driven simulation 81
Figure 4.27 The digital phase and frequency detector 85
Figure 4.28 The simulated locking behavior of the DPLL 88
Figure 4.29 Correct jitter tracking of the digital PLL 89
Figure 4.30 Jitter tracking failure of the digital PLL 90
Figure 4.31 The reference phase noise transfer characteristic 90
Figure 4.32 The simulated phase noise performance of case 1 93
Figure 4.33 The simulated phase noise performance of case 2 93
Figure 4.34 The simulated phase noise performance of case 3 94
Figure 4.35 The simulated phase noise performance of case 4 95
Figure 4.36 The simulated phase noise performance of case 5 95
Figure 4.37 The simulated phase noise performance of case 6 96

Figure 5.1 Block diagram of the DPLL 98
Figure 5.2 The proposed digitally controlled oscillator 99
Figure 5.3 The schematic of the oscillator 100
Figure 5.4 The characteristic of the capacitance tuning unit 101
Figure 5.5 Simulated capacitance tuning characteristic of a PMOS transistor 101
Figure 5.6 Continuous coverage of multiple tuning banks 102
Figure 5.7 The proposed fine capacitance tuning bank 102
Figure 5.8 The proposed coarse capacitance tuning bank 104
Figure 5.9 The layout floor plan of varactor banks 105
Figure 5.10 N+/NWell MOSVAR mismatch performance[25] 107
Figure 5.11 The schematic of the varactor driver circuit 108
Figure 5.12 The varactor tuning circuit 109
Figure 5.13 The multi-input D flip-flop 110
Figure 5.14 The schematic of the multi-inputs D flip-flop 111
Figure 5.15 The schematic of the fine varactor tuning circuit 111
Figure 5.16 The schematic of the coarse varactor tuning circuit 112
Figure 5.17 The schematic of the phase tuning circuit 113
Figure 5.18 The principle of the UP/DN sensor 114
Figure 5.19 The characteristics of sign detection 114
Figure 5.20 The schematic of the up/dn sensor 115
Figure 5.21 The proposed cycle slip detection circuit 115
Figure 5.22 The principle of the proposed cycle slip detection circuit 116
Figure 5.23 The logical circuit for frequency decision 117
Figure 5.24 The decision clock generator 118
Figure 5.25 The schematic of the divide-by-2, 3 frequency divider 118

Figure 5.26 The schematic of the DCO frequency divider 119
Figure 5.27 The schematic of the frequency divider for the frequency decision 122
Figure 5.28 The data interface 123
Figure 5.29 The three-wire interface 123
Figure 6.1 Micrograph of the chip 125
Figure 6.2 The test bench for the DCO 125
Figure 6.3 The screen-shot DCO graphic control panel 126
Figure 6.4 The measured band tuning characteristic of the DCO 127
Figure 6.5 The measured coarse tuning characteristic of the DCO 127
Figure 6.6 The measured fine tuning characteristic of the DCO 128
Figure 6.7 The measured fine/coarse tuning characteristic of the DCO 129
Figure 6.8 The measured phase noise of the DCO 129
Figure 6.9 The measured analog gain of the DCO 130
Figure 6.10 The chip micrograph of the proposed DPLL 131
Figure 6.11 Top-level inputs/outputs of the chip 131
Figure 6.12 The screen-shot of the control panel for the DPLL measurement 132
Figure 6.13 The test bench for the DPLL 133
Figure 6.14 The photo of the test setup of the DPLL measurement 133
Figure 6.15 The measured open loop phase noise of the digital PLL 134
Figure 6.16 The measured reference phase noise 135
Figure 6.17 The measured phase noise of the low-complexity digital PLL 136
Figure 6.18 The measured output frequency spectrum of closed-loop DPLL 136
Figure 6.19 The measured reference spur performance of the closed-loop DPLL 137
Figure 6.20 The measured spur from the digital CDR test chip 138
Figure 6.21 The phase noise of the DPLL with the low noise DCO 140

CHAPTER 1 *Introduction*

1.1 Introduction

Phase-locked loops (PLLs) have been widely used in frequency synthesis and timing recovery fields for many years. To date, most PLLs in the literature are in analog forms, where analog phase and frequency detectors, charge pumps, loop filters (LFs) and voltage controlled oscillators (VCOs) are employed. With the increasing demand for the system-on-chip (SoC) solutions, more and more system components are integrated into a single chip, demanding high noise immunity and portability of those building blocks. Analog intensive circuits are difficult to be integrated in such a noisy environment because they are usually sensitive to circuit noise, and are not portable for different processes. To overcome those difficulties and enable more desired features that are not achievable in an analog implementation, many analog circuit components have been digitized, including analog PLLs. Recently, the deep-submicron CMOS processes make possible to extensively digitize conventional analog circuits using high-speed digital circuits. In fact, all digital PLLs (ADPLLs) gain increasing interests in both frequency synthesis and the data recovery fields in recent years [1]-[24].

Most reported digital PLLs so far are basically digital versions of their analog counterparts, in that each blocks are digitized separately. The phase/frequency detector is replaced by a digital phase/frequency detector, whose output is the digital representation

of the phase/frequency error. The analog loop filter is replaced by a digital loop filter with a similar frequency response. The VCO is replaced by a digitally-controlled oscillator (DCO), whose frequency is controlled by its digital frequency control word. Although those DPLLs can be designed and analyzed based on existing design methodology of analog PLLs, they normally require high resolution/linearity/complexity so that the DPLL behaves similarly as an analog PLL. For example, a popular type-II digital loop filter, consisting of multipliers, adders, and so on, may require thousands of gates and is a large digital component comparing with the remaining components of the DPLL. Because the digital loop filter usually works with binary weighted signals, the output of the digital phase and frequency detector usually needs to be encoded to a binary weighted signal and the output of the digital loop filter is usually coded/decoded to some kind of linear coded signal before it is connected to the digital tuning units of the DCO to achieve better linearity. In this research work, a low complexity digital PLL architecture, including a non-linear digital PFD (DPFD), a phase/frequency decision circuit, an enhanced frequency resolution DCO and a frequency divider is proposed, analyzed, simulated and measured. The proposed low-complexity digital PLL can easily be implemented and employed in some integer-N frequency multiplication applications, where the in-locked performance is of concern.

A VCO is an important building block of an analog PLL. The output frequency of a VCO is controlled by its analog control voltage, and the main parameters of a VCO are the centre frequency, the frequency tuning range, the VCO gain, the phase noise and so on. Because of the frequency tuning nature, a VCO usually has a high gain to meet the frequency tuning range requirement. However, for a high-gain VCO, any analog noise at the control line significantly degrades the phase noise performance. This effect can be reduced by reducing the VCO gain, but the frequency tuning range is also reduced because of the limited voltage swing of the VCO control voltage. The combination of the coarse frequency tuning and the fine frequency tuning is employed in some analog PLLs to

enlarge the frequency tuning range and to keep a low VCO gain. Although the gain of the fine tuning is reduced to make a VCO insensitive to the noise from the fine tuning control signal, the gain of the coarse tuning is still very large, thus the VCO is still highly sensitive to the noise in the coarse tuning signal.

In a digital PLL, the analog VCO is replaced with a DCO to operate with a digital loop filter. This replacement makes it possible to achieve both small analog gain and large frequency tuning range if all tuning units are biased in their low-gain regions. This requires the tuning unit have two flat regions in its capacitance tuning characteristic curve. In this case, the frequency resolution is normally limited by the difference (i.e. the capacitance difference) between the two digital tuning levels of the smallest tuning unit for a given technology. In [13], a sigma-delta modulator was used to enhance the frequency resolution by dithering the DCO frequency, and it increases the complexity of the DCO interface and the power consumption. Additionally, it may result in some undesired quantization noise. In this work, a capacitance tuning scheme for an LC-tank based DCO with incrementally sized varactors and matched varactor banks (across LSB/MSB banks) is proposed. It can achieve both a high frequency resolution (small frequency steps) and a large linear frequency tuning range with a small differential nonlinearity.

The digital PFD detects and quantizes the phase/frequency error so that it can be processed by the digital loop filter. For a fractional-N frequency synthesizer, the phase error quantization requires a high linearity and high resolution to reduce the output spur and the in-band noise, to minimize the noise folding (if the sigma-delta noise shaping is used). However, for an integer-N frequency synthesizer, the phase error in the locked condition is usually small with some boundaries. In [12], a time to digital converter (T2D) with exponential quantization steps is employed to reduce the complexity of the T2D without degrading the noise performance in the locked condition, and a look-up table is used to linearize the T2D output before it is processed by the digital loop filter. A further reduction of the complexity of the T2D is done in this work by employing only few neces-

sary phase quantization levels. In addition, the proposed DPFD, operating with a phase/frequency decision circuit, does not require the encoder or the look-up table.

Most all-digital PLLs reported so far employ arithmetic linear digital loop filters, in that the loop filter performs mathematical multiplication, integration and summation operations so that its frequency response can be comparable to an analog loop filter. The output of such digital loop filters is normally a binary weighted signal, while the linearly coded signals, such as the thermometer-coded and the one-hot coded signals are desired for the digital tuning units in a DCO to achieve a small differential-nonlinearity of the frequency tuning. In such case, the digital loop filter output needs to be encoded/coded to the desired format before it is connected to the digital tuning units. Clearly, those approaches require a high-complexity digital circuitry. In this work, a phase/frequency decision circuit, providing the phase/frequency up/down signals to the DCO, is proposed in this work. Not only the proposed circuit has a low complexity comparing with other popular arithmetic digital loop filters, but it can significantly simplify the interface of the DCO.

Although this dissertation mainly focuses on one phase/frequency decision algorithm, other phase/frequency decision algorithms may be used to achieve different loop behaviors if necessary.

1.2 Contributions

The major contributions made in this work are listed as follows:

1. The architecture of a low-complexity integer-N digital PLL frequency synthesizer was proposed.
   - The digital PFD was simplified to consist of a conventional PFD with a UP/DN sensor.
   - The commonly used linear digital loop filter was shrunk to a low-complexity phase and frequency decision circuit.

- The phase error integration operation was implemented inherently in the one-hot code or thermometer code generator in the DCO, and the arithmetic integrator and the encoder/decoder in the DCO interface were eliminated.

2. A digitally-controlled oscillator with an enhanced frequency resolution, whose frequency step is much smaller than the step corresponding to a single smallest varactor, was proposed.
   - The LSB varactor bank consists of incrementally sized varactors controlled by one-hot coded control bits.
   - The MSB varactor bank consists of unit-sized varactors controlled by thermometer-coded control bits, and each unit is matched with the tuning range of the LSB bank to achieve the monotonic frequency tuning across MSB and LSB banks without overlaps.
3. Novel theoretical analysis methodologies were developed and used to analyze the loop behavior of the proposed low-complexity DPLL.
   - The close-loop phase noise performance was analyzed.
   - The large signal behavior and the loop stability were investigated.
   - The determination of the design parameters was discussed.
4. The matlab implementation of an event-driven behavioral simulation technique was developed and used to simulate the proposed low-complexity digital PLL, and confirm the theoretical analysis.
   - The matlab implementation of the event-driven behavioral simulation technique was first discussed in general, and the modeling method of each building blocks of the digital PLL was provided.
   - The phase and frequency acquisition behavior of the proposed digital PLL was simulated using the event-driven simulation technique and compared with the

result obtained with the Simulink modeling simulation to confirm the feasibility of the event-driven simulation technique.

- With the developed behavioral simulation technique, the phase noise performance of the proposed digital PLL for different loop parameters and different reference and DCO phase noise profiles were simulated and compared with the theoretical analysis.

5. Some novel circuit implementation techniques were proposed to be used in the proposed DCO and DPLL. Some of them were applied to two test chips, implemented in CMOS 90nm technology, to confirm the flexibility of the proposed DCO and the DPLL.

   - The method of minimize the hysteresis of the UP/DN sensor was proposed.
   - A circuit implementation of the cycle-slip detector was proposed.
   - A low-complexity implementation of the frequency decision circuit, using only four flip-flops, was proposed.

1.3 Document organization

The operation principle of a PLL-based frequency synthesizer is briefly reviewed in Chapter 2, followed by the review of some typical digital PLLs in the literature. The proposed digital PLL, including the architectures of the DCO, Digital PFD and the phase/frequency decision circuit, is described in Chapter 3. In Chapter 4, the theoretical analysis of the proposed DPLL is given, and the determination of the design parameters is discussed followed by some behavioral simulations, which confirm the theoretical analysis. Chapter 5 describes the CMOS implementation of the proposed digital PLL, while the performance evaluation of two test chips, the DCO chip and the digital PLL chip, is described in Chapter 6. Finally, chapter 7 concludes the whole document.

CHAPTER 2

Review of PLL-based frequency synthesizers

2.1 Introduction

Frequency synthesizers are key building blocks of the data communication systems, and their performance, including the phase noise, jitter, frequency range, power consumption, and so on, are crucial for those systems. Phase-locked loops have been widely used in frequency synthesis field for many years. In recent years, digital phase-locked loops (DPLLs) have been under exploration and some frequency synthesizers based on DPLLs appear in the literature[1]-[14]. This chapter first reviews the basic operation of a PLL-based frequency synthesizer and some typical analog implementations, then reviews some DPLLs reported in the recent years.

2.2 Basic operation of a PLL-based frequency synthesizer

2.2.1 Operation principles of PLL

The concept of phase-locking has come out for more than half century. Nowadays phase-locked loop is widely used in many applications, among which the most common are frequency synthesis, phase modulation/demodulation, tracking filter, carrier recovery, and clock recovery.

A phase-lock loop is essentially a negative feedback loop in which the phase of the frequency controlled oscillator is obliged to follow that of the input signal. One measure

of a PLL's performance is the phase error, which is the phase difference between the input signal and output signal. A basic block diagram of PLL is shown in Figure 2.1. When the

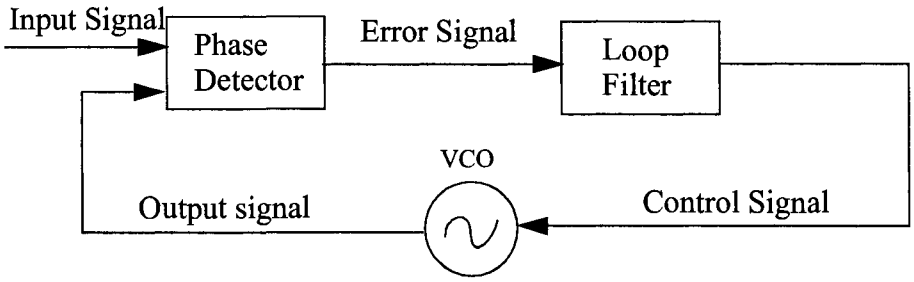


Figure 2.1: The basic structure of a phase-locked loop

loop is in lock, the VCO frequency equals to the input frequency, and in order to make the two signals in the same phase, a control voltage is generated with the phase detector and the loop filter based on the phase error to tune the frequency of the VCO.

When the loop is unlocked, suppose the input signal is

$$y_i(t) = A\sin[\omega t + \phi_i(t)] \tag{2.1}$$

and output signal is

$$y_o(t) = B\cos[\omega t + \phi_o(t)] \tag{2.2}$$

Suppose the phase detector has a sinusoidal characteristic, so the error signal u_1 is

$$u_1 = K_d\sin[\phi_i(t) - \phi_o(t)] \tag{2.3}$$

K_d represents the phase detector sensitivity (V/rad). Name the transfer function of the loop filter as $f(t)$, and the loop filter gain as K_f, then the command signal u_2 is $K_f u(t) * f(t)$, where symbol $*$ represents the convolution product. The angular frequency unit is radian per second, which is the time derivative of the phase. A time domain nonlinear differential equation comes out as,

The phase-locked loop is a nonlinear system and it is very complicated to analyze

$$\frac{d\phi_o}{dt} = K_d K_f K_{vco}\{\sin[\phi_i(t) - \phi_o(t)] * f(t)\} \quad (2.4)$$

its performance. Fortunately, when a PLL is in the steady state or in a very slow transient process, its dynamic response to the input-signal phase and frequency changes can be well approximated by linear model or small signal model as shown in Figure 2.2. In the figure,

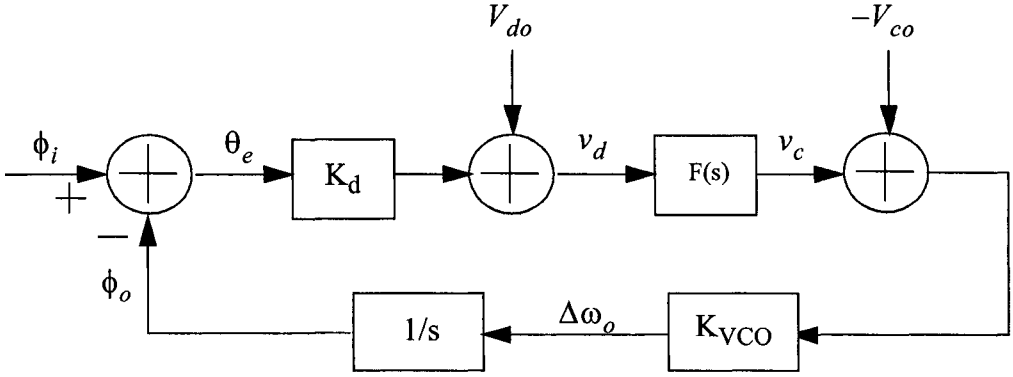


Figure 2.2: The linear model of a phase-locked loop

V_{do} is the offset voltage of phase detector, i.e. the output voltage of phase detector when the phase difference is *0*. In some applications, it is desirable for V_{do} to be *0*. Similarly, V_{co} is the control voltage that brings the output frequency of the voltage controlled oscillator in phase with the input when in lock, so different from V_{do} which depends on the phase detector, V_{co} is not a property of VCO instead that it also depends on the input frequency in some applications.

In the linear model, the phase error between the input and the output, i.e. $|\phi_i(t) - \phi_o(t)|$ is small enough. Usually when $|\phi_i(t) - \phi_o(t)| < \frac{1}{2} rad$, equation (2.6) can be written as,

$$\frac{d\phi_o}{dt} = K_d K_f K_{vco}\{[\phi_i(t) - \phi_o(t)] * f(t)\} \quad (2.5)$$

Normally it is preferred to express in frequency domain rather than time domain in

terms of the PLL's response to some external disturbances, especially the noise performance. After Fourier transform, with $\Phi_i(j\omega)$, $\Phi_o(j\omega)$ and $F(j\omega)$, as the Fourier transforms of $\phi_i(t)$, $\phi_o(t)$ and $f(t)$, a linearized transfer function of the PLL can be expressed as,

$$H(j\omega)=\frac{\Phi_o(j\omega)}{\Phi_i(j\omega)}=\frac{KF(j\omega)}{j\omega+KF(j\omega)} \qquad (2.6)$$

Where K is the product of $K_dK_fK_{VCO}$, the open loop gain, and usually K_f is 1.With this equation, PLL's dynamic response to some external disturbances can be achieved. Some typical disturbances are frequency step, phase step and frequency ramp.

2.2.2 PLL frequency synthesis

A typical structure of a PLL-based frequency synthesizer is shown in Figure 2.3.

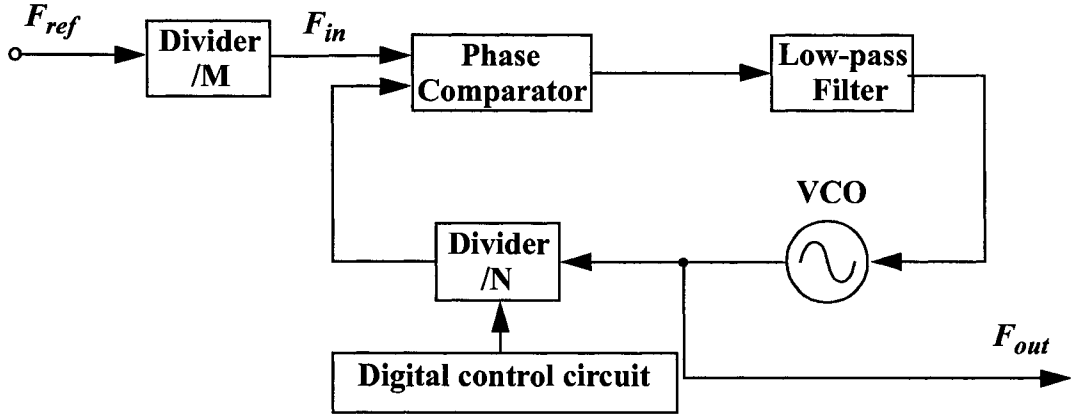


Figure 2.3 A PLL-based frequency synthesizer

The basic function of a PLL is to tune the phase of the VCO to make the frequencies of two inputs of the phase comparator identical and the phase difference between them constant. With a divider (/N), the actual output frequency of the VCO is,

$$F_{out} = N \cdot F_{in} \qquad (2.7)$$

Normally N is an integer. By changing the value of N, the output frequency can be

changed with the increment of F_{in}. To obtain a smaller increment, another 'divider' is employed to divide down the reference signal F_{ref} by M, thus the increment is also divided by M. The actual relation between the output (F_{out}) and the reference signal (F_{ref}) is thus,

$$F_{out} = N \cdot \frac{F_{ref}}{M} \qquad (2.8)$$

However, with this method, the frequency increment can not be reduced infinitely because the value of M cannot be too large due to the noise performance issue.

Not only can the phase-locked loop be employed to generate an output frequency with the ratio of an integer, but a ratio which is the sum of an integer and a fraction of an integer can be obtained, this is often referred as fractional-N frequency synthesis. To achieve the fractional division ratio, a programmable divider can be employed in the PLL-based frequency synthesizer. Figure 2.4 shows a simplified diagram of a fractional-N

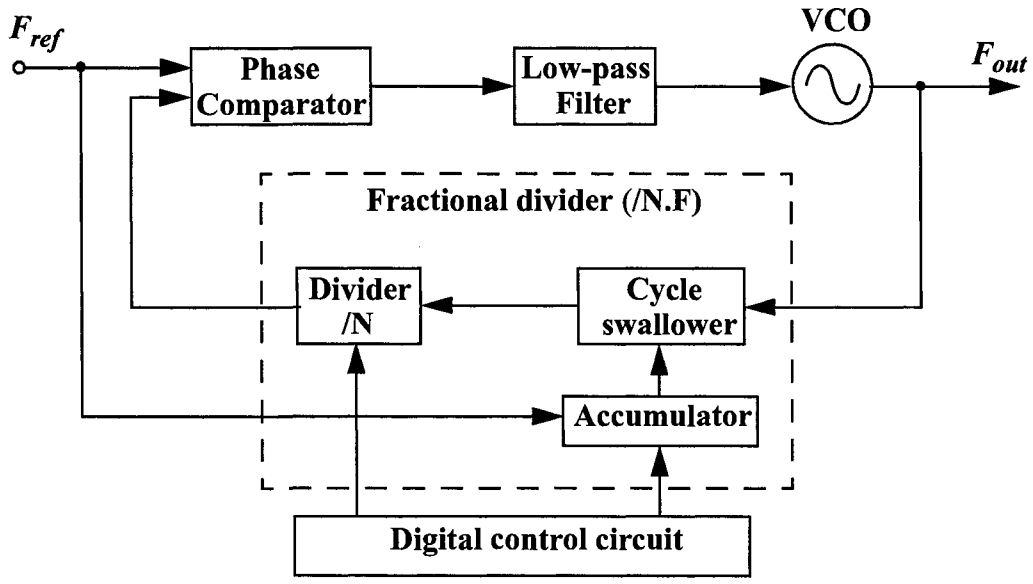


Figure 2.4 A fractional-N PLL-based frequency synthesizer

PLL-based frequency synthesizer with a divider including a cycle swallower to obtain a fractional division ratio. With this fractional division ratio, the ratio between the output

frequency and the input frequency can be fractional since the output of the divider is locked at the same frequency as the input.

2.3 Performance parameters of a frequency synthesizer

2.3.1 Phase noise

The spectral purity of a periodic signal can be easily characterized and visualized in the frequency domain. The impulse spectrum of an ideal frequency source is shown in

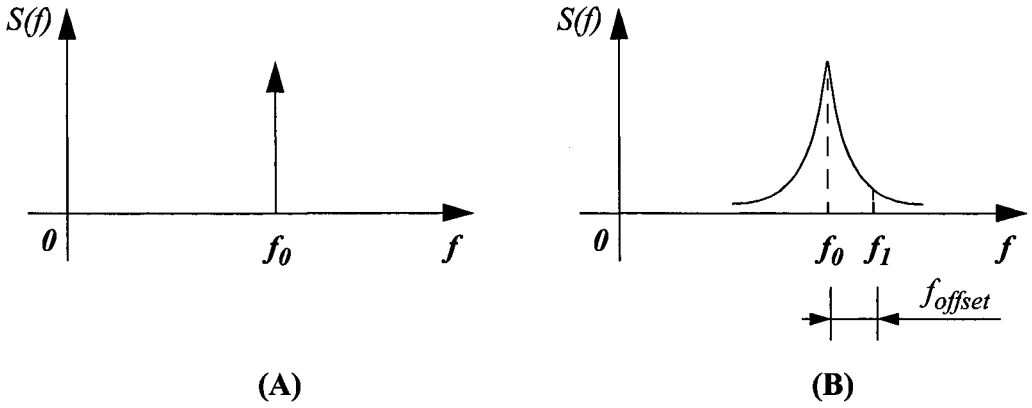


Figure 2.5 The spectrum of an ideal signal and a practical signal

Figure 2.5-A. Figure 2.5-B shows the spectrum of its practical counterpart having the same nominal output frequency. The ratio between the power at a frequency offset (f_{offset}) within a certain bandwidth to the power of the center frequency (f_0), is defined as phase noise. It is a measure of the relative level of the undesired noise components,

$$\text{Phase noise}(f_{offset}) = \frac{\text{Noise power at } (f_0 + f_{offset})}{\text{Power at } f_0} \qquad (2.9)$$

Usually, the measurement bandwidth of the noise power is 1Hz and the phase noise is expressed in logarithmic scale, resulting in an unit of dBc/Hz. It can be seen that the phase noise is essentially a measure of the instantaneous uncertainty in a carrier's period. This uncertainty can be considered as the result of frequency modulating the car-

rier by some other signals and therefore phase noise can be better understood by examining the frequency modulation process.

In frequency modulation (FM), the instantaneous frequency of a sinusoidal carrier is a function of the amplitude of the modulating signal. Suppose this modulating signal is a sinusoid,

$$v_m(t) = A_m \cos(\omega_m t) \qquad (2.10)$$

This assumption does not result in a loss of generality, because any physically-realizable signal can be decomposed into a set of sinusoidal signals by Fourier transform. The signal resulting from the frequency modulation is thus written as,

$$v(t) = A_c \cos\left(\omega_c t + k_f \cdot \frac{A_m}{\omega_m} \sin(\omega_m t)\right) \qquad (2.11)$$

Where k_f is a constant, A_c the amplitude, and ω_m is the angular frequency of the modulating sinusoid. When $k_f \cdot \frac{A_m}{\omega_m} \ll \frac{\pi}{2}$, Equation (2 . 11) may be written as,

$$v(t) \cong A_c\left[\cos(\omega_c t) + \frac{m_f}{2}\cos((\omega_c + \omega_m)t) - \frac{m_f}{2}\cos((\omega_c - \omega_m)t)\right] \qquad (2.12)$$

Where $m_f = k_f \cdot \frac{A_m}{\omega_m}$ is the modulation index. Equation (2 . 12) shows that at low modulation levels, frequency modulation of the carrier by a single low-level noise sinusoid results in the appearance of two sidebands at $\omega_c \pm \omega_m$ as shown in Figure 2.6.

For the output of a practical frequency synthesizer, the modulating signal is not a simple sinusoidal signal as above, but a complex noise, which contains many frequency components. Figure 2.7 shows the power spectrum produced when a sinusoidal carrier is modulated by a more general signal (noise). The relationship of the device output spectrum to the spectrum of the modulating noise can be observed in this diagram.

Equation (2 . 9) is a definition of phase noise in terms of power levels. A more

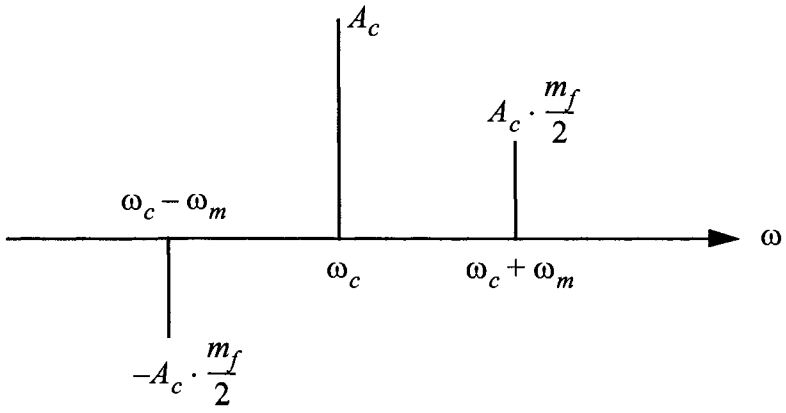


Figure 2.6 The spectrum of a sinusoid FM signal

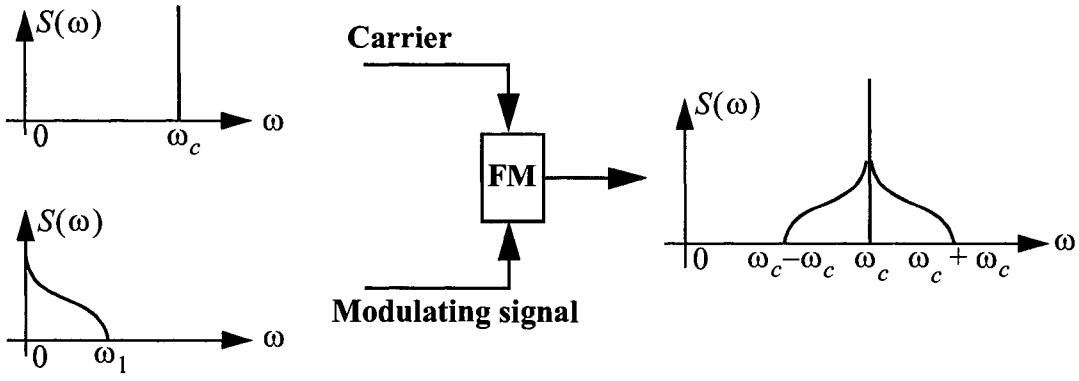


Figure 2.7 The spectrum of a FM signal

practical definition of phase noise (at the frequency offset f_m) in terms of frequency deviation from a central value (carrier) can also be given as,

$$\text{Noise-to-carrier ratio(dB)} = 20\log\left(\frac{\Delta f_{rms}}{\sqrt{2} f_m}\right) \qquad (2.13)$$

Where f_m is the frequency offset from carrier and the Δf_{rms} is the RMS value of the frequency variation with respect to the carrier frequency within a certain bandwidth (normally 1Hz). Particularly, if the bandwidth is 1Hz, the result of Equation (2 . 13) is in unit of dBc/Hz.

2.3.2 *Spurious power level*

The undesired frequency components which appear in the operating band of a device at some discrete frequency points are called spurs. Their power levels are described in a way similar to the phase noise power levels, namely by comparison with the carrier power. Figure 2.8 shows an example of a frequency source corrupted by both phase noise

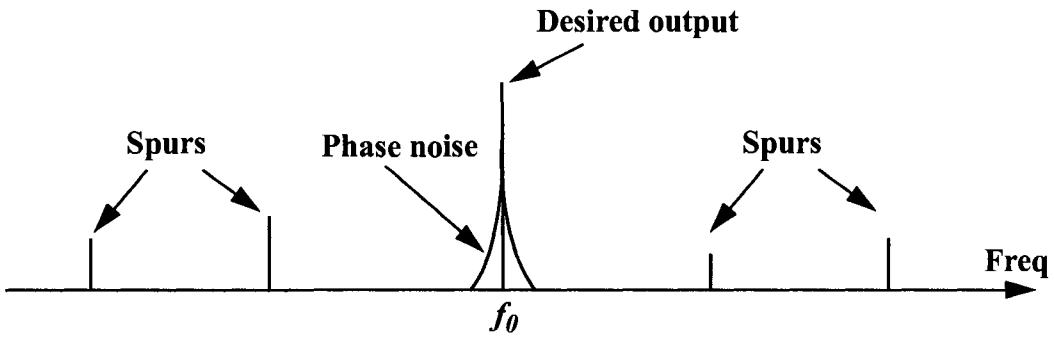


Figure 2.8 The power spectrum of a frequency source with spurs

and spurs.

In theory, the phase noise can be defined to measure the phase variation over a wide band centered on the desired output frequency. But such a wide band measurement would give a misleading impression because spurs only appear at discrete frequencies.

Spurious power level is measured in a way that is analogous to phase noise measurements,

$$\text{spurious power level(dBc)} = 10\log\left(\frac{\text{Spurious power}}{\text{Carrier power}}\right) \qquad (2.14)$$

Spurious power level is related to carrier power but is independent of measurement bandwidth, so it is reasonable to express it in unit of dBc (decibels relative to carrier). Among those spurs shown in Figure 2.8, the two spurs closest to carrier are of most concern because they may affect the following systems significantly. So the spurious power level normally refers to the spur with relative higher power.

2.3.3 Output period jitter

In data communication systems, the period jitter of the clock signal are of great interest. The period jitter is the time difference between a measured cycle period and the ideal cycle period. The jitter can be measured as peak to peak jitter or by Root of Mean Square (RMS). The period jitter is illustrated in Figure 2.9A, in which the dashed curve is

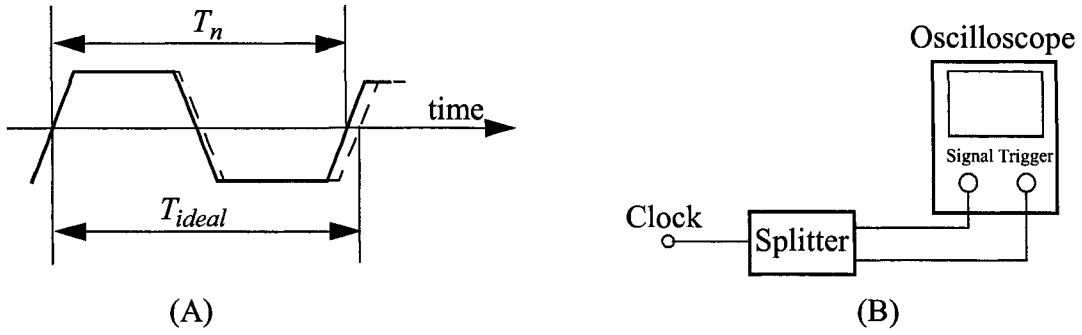


Figure 2.9 Illustration of the period jitter

the ideal signal, having a period of T_{ideal}, while the solid curve is the actual clock signal, whose period is T_n. The period jitter in this case is calculated as:

$$J_{period} = T_n - T_{ideal} \qquad (2.15)$$

2.3.4 Operating frequency range

Normally, the frequency synthesizer produces an output signal whose frequency is the reference frequency multiplied by some factor. This factor may be fixed, determined by the circuit design itself, or it may be programmable. But in either case the output frequency changes if the reference frequency does.

The operating frequency range of the PLL denotes the maximum input frequency range over which the desired frequency ratio between input and output can be maintained with acceptable output phase noise, spurious power level and so on. It is important that such operating frequency range is usually determined by the frequency range of the VCO

or DCO, and the frequency multiplication ratio.

This range is dictated by the target application, but the designer should realize that it also depends on unpredictable effects on devices caused by such things as temperature variation, supply voltage fluctuation and more, and 'over-design' the synthesizer accordingly.

2.3.5 Frequency step

The frequency increment of a tunable frequency synthesizer is defined to be the minimum possible frequency shift of its output when the reference frequency is fixed. The integer-N synthesizer, the most common one, has a constant frequency increment equal to the reference frequency over the full tuning range.

For some applications, tunable frequency multiplying synthesizers, which generate output frequencies which are N times the input, do not have adequate frequency resolution (the frequency increment is too large) and more sophisticated technologies, such as fractional-N synthesis, are employed.

2.3.6 Channel switching time

Channel switching time refers to the time needed for a tunable frequency synthesizer to settle to the new output frequency after the reference frequency or the frequency multiplication ratio is changed. Most indirect frequency synthesizers incorporate some kind of filtering in their feedback loops, and the filter bandwidth normally affects the channel switching time.

A well-known relation between a filter's 3 dB bandwidth (B_{3dB}) and the rise time (t_r) is shown in Equation (2 . 16), where k is a value between 0.3 to 0.45. The rise time t_r

$$t_r \cong \frac{k}{B_{3dB}} \qquad (2.16)$$

is the time needed for a voltage 'step' to move from 10% to 90% of its full swing. The equation can be used to estimate the channel switching time during initial design stages. It should however be kept in mind that Equation (2 . 16) does not involve phenomena such as overshoot and ringing, and a more refined model should be used in later design stages.

2.3.7 Power consumption

The concept of power consumption, which refers to how much energy a device consumes within a unit time, is applicable for all electronic devices. Power consumption is always an important consideration for the designers. This is especially true for portable devices such as mobile phones since it determines the battery life. Consequently, low power consumption design aspect is always a big issue in the design stages of those devices.

2.4 Analog and digital PLL design trade-offs

Figure 2.10 shows the principle of a PLL-based frequency synthesizer. The input

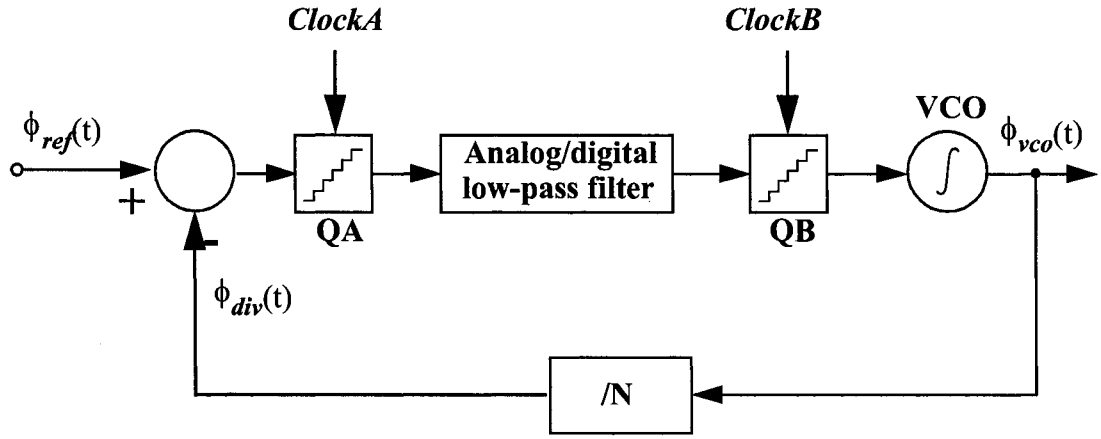


Figure 2.10 The principle of a PLL-based frequency synthesizer

and output are represented by their phase information. The actual time-domain reference signal, VCO output, and divider output are $\cos(\phi_{ref}(t))$, $\cos(\phi_{vco}(t))$ and $\cos(\phi_{div}(t))$ respectively. The phase detector is essentially a subtracter whose output is $\phi_{ref}(t)-\phi_{div}(t)$.

This phase error information is filtered to tune the VCO, which also performs an integration operation to convert the filtered phase error to VCO phase ($\phi_{vco}(t)$).

Two quantizers in the figure are optional depending on types of PLLs. For an analog PLL employing a continuous time phase detector, no quantization operation is involved. However, almost all analog PLLs employ a discrete time phase detector in that the phase error sample is only produced once per reference or divider cycle. In such PLLs, the quantizer A (QA) has infinite small quantization steps and its sampling frequency is the same as the reference frequency in the locked condition. For a digital PLL, the phase error has to be quantized with finite steps by (QA) so the phase error is represented in discrete time and discrete value, which is processed by some digital circuit (in most cases, a digital loop filter) so that the tuning signal for the oscillator can be obtained. Usually, such digital circuit operates in the same clock rate as the QA (*ClockB* = *ClockA*). Although the output of the digital circuit may have a very high resolution, the DCO in a digital PLL normally has a limitation on the tuning resolution, which is represented as the finite steps of quantizer B (QB). Table 2.1 summarizes the mathematical representations of analog/digital PLLs.

TABLE 2.1: A mathematical representations of analog/digital DPLLs

Parameters	Continuous-time analog PLLs	Discrete-time analog PLLs	Digital PLLs
Frequency of ClockA	Infinite high	Reference frequency	Reference frequency
Quantization step of QA	Infinite small	Infinite small	finite step sizes
Frequency of ClockB	Infinite high	Infinite high	Reference frequency
Quantization step of QB	Infinite small	Infinite small	finite step sizes
Filter	Analog filter	Analog filter	Digital loop filter

In an analog PLL, noise in PFD, charge pump, loop filter and so on is transferred to the VCO output, increasing the output phase noise. In addition, the design flow and circuit design techniques of an analog PLL are analog intensive and they are difficult to be re configured, adaptive, scaled with new technologies.

Digitizing blocks of an analog PLL gains increasing interest with the increase of

the operating speed of digital circuits. From description above, the digitizing process is essentially inserting two quantizers in the loop so that the analog filter can be replaced by digital circuit. The quantizer A is usually implemented in the phase detector, resulting in a digital phase detector. Although the quantization introduces some additional quantization noise, the digital circuit itself does not contributes any additional noise as an analog loop filter does. By using digital circuits to perform the filtering function, various features can be achieved, such as dynamic bandwidth, fast acquisition, auto-calibration, and so on. In addition, the digital circuit is highly portable for different technologies. However, direct digitization of an analog PLL requires some mathematical operations in high precision, some complex code conversion operation, and high precision A/D conversion, resulting a complex circuitry.

2.5 *Review of digital PLL-based frequency synthesizers*

Figure 2.11 shows a block diagram of a typical digital PLL. The basic operating

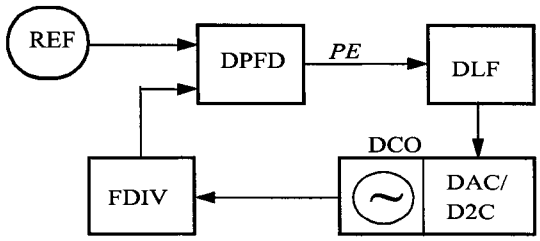


Figure 2.11 The block diagram of a digital PLL

principle of digital PLLs is still the same as that of their analog counterparts, in that they normally consist of a tunable oscillator, frequency divider, phase/frequency detector, and loop filter. In an all digital PLL, the DCO usually relies on some analog oscillating circuit although the oscillator tuning is achieved digitally. The frequency divider, in many cases, can be the same as a conventional frequency divider used in an analog PLL. The digital phase and frequency detector, which is usually a combination of a analog phase/frequency

detector and a time to digital converter (T2D), compares the reference frequency with the divider output to produce a phase error information represented in digital signals. In some digital PLLs, the frequency divider and the DPFD are implemented together to realize more features. The digital loop filter, however, has both digital inputs and outputs. As we can see, digitizing an analog PLL is essentially replacing the analog loop filter with a digital loop filter together with some ADC and DAC components as its interface to other analog blocks. Some reported digital PLLs are reviewed in this section as follows.

2.5.1 A novel all-digital PLL with software adaptive filter

In [1], an all-digital PLL with a software adaptive filter for video applications is reported. This PLL needs to lock to the horizontal synchronization (HS) signal and generate the pixel clock for ADCs, and other clocks for various digital function blocks. The system block diagram is shown in Figure 2.12. It consists of a DCO, a counter/divider, a phase detector and a low-pass filter. To simplify the T2D in the phase detector, the fre-

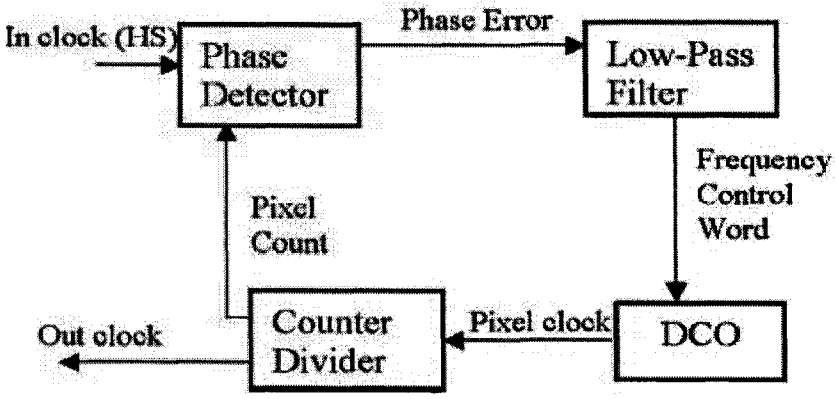


Figure 2.12 An all-digital PLL with software adaptive filter

quency divider is replaced by a period counter and its output is sampled at the edge of the reference clock (HS) so that the phase error can be determined. This phase detection approach results in a time resolution equal to one DCO period. The counter in this PLL, operating at the same frequency as the pixel clock, counts up until it reaches max_pix-1, and then it rolls over and starts from zero again. The latched counter output (pix_cnt) is

converted to phase error using following formula:

If pix_cnt<max_pix/2,

phase_error = pix_cnt+1,

otherwise,

phase_error=pix_cnt-max_pix

This conversion deliberately avoids the case where phase error is zero so that the deadzone of one DCO period can be avoided. Thus the minimum phase error is either +1 or -1. By this way, the dead zone is essentially limited to the performance of the sampling register.

The low-pass loop filter is implemented in software that is executed on an on-chip microprocessor shared with other functions. The microprocessor computes the frequency control word based on the transfer function:

$$H(z) = G_1 + \frac{G_2}{1 - z^{-1}} \qquad (2.17)$$

The proportional gain (G_1) and the integral gain (G_2) determine the filter response. They can be adjusted dynamically on the fly, with greater gain in the start-up process for fast locking and smaller gain in the steady-state for better stability.

The DCO used in this reported design is shown in Figure 2.13, which is essentially a combination of an analog PLL-based frequency multiplier and a direct frequency and phase synthesizer. The analog frequency multiplier generates high frequency multiple phase clock signals and the direct frequency and phase synthesizer picks up a correct phase on the fly, based on the frequency and phase control word.

This ADPLL was implemented in a 0.6μm CMOS process with 3.3-V supply. It is responsible for generating all pixel clocks within 10-80MHz for various VESA video modes. The ADPLL consumes approximately 180mW by itself, with an area of approximately 1.8mm^2. A long-term peak-to-peak jitter of 1.5ns was observed for a 78MHz out-

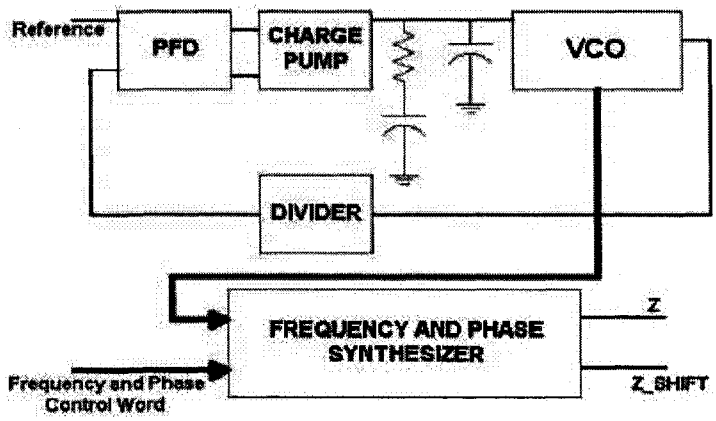


Figure 2.13 The DCO for the all-digital PLL with software adaptive filter

put with the reference of 75Hz.

2.5.2 A compact, low power low jitter digital PLL

A compact digital PLL with variable loop gain for fast acquisition is reported in [2] and its architecture is shown in Figure 2.14. The PFD is reduced to a one-bit compara-

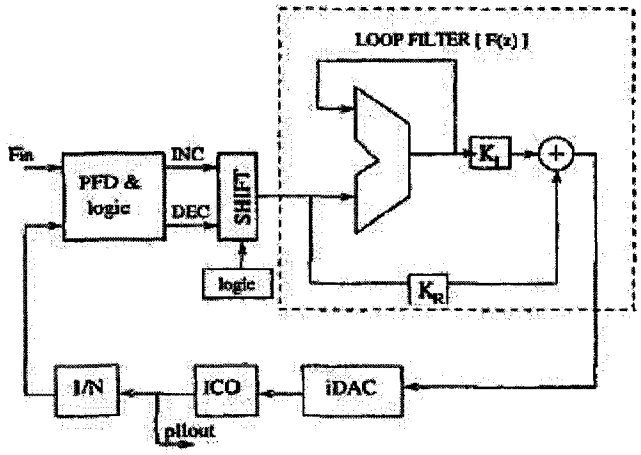


Figure 2.14 A compact, low power low-jitter digital PLL

tor, which simply outputs the sign of the phase error. A gain factor is applied to the output of the PFD by the "SHIFT" controlled by a logic circuit. Initially, the error is assumed to be equal to one-half the size of the MSB. Once the sign of the error is changed, the amplitude of the error is also reduced by half. This allows a fast binary tuning of the frequency.

The loop filter is updated only when the PFD outputs are updated. When the system is locked, the updating frequency is the same as the reference frequency. The output of the digital loop filter is sent to a DAC, which produces a current to tune a current controlled oscillator (ICO). The transfer function of the digital loop filter is shown in Equation (2 . 18), which is a conventional type-II loop filter.

$$F(z) = K_R + \frac{K_I}{1 - z^{-1}} \qquad (2.18)$$

An adaptive loop bandwidth is used in this DPLL by adjusting two parameters K_R and K_I based on the jitter estimated by counting the number of consecutive INC and DEC pulses.

This DPLL was fabricated in 0.25μm CMOS technology and the measured close-in phase noise is less than -87dBc/Hz and the closed-loop bandwidth is around 2MHz. The VCO frequency range is 30MHz - 160 MHz, and the power consumption is 3.12mW @ 144MHz with 2.6V power supply. The measured cycle jitter is 130ps and the RMS jitter is 60ps.

2.5.3 An all-digital phase-locked loop for high-speed clock generation

An all-digital phase-locked loop for high-speed clock generation is presented in [3]. This ADPLL employs a novel fine-tuning delay cell, which can reduce both cost and design time for building a high-resolution cell-based DCO. There are two identical DCOs in the ADPLL, one (Inner DCO) is used for tracking the reference clock and the other one (Output DCO) is used for generating the output clock.

The output clock of the inner DCO is divided by M then compared with the reference signal (REF_CLK). The PFD detects the frequency/phase difference between the reference signal and the divided output clock and generates an up (P_UP) signal or down (P_DOWN) signal to indicate whether the inner DCO should be sped up or slowed down.

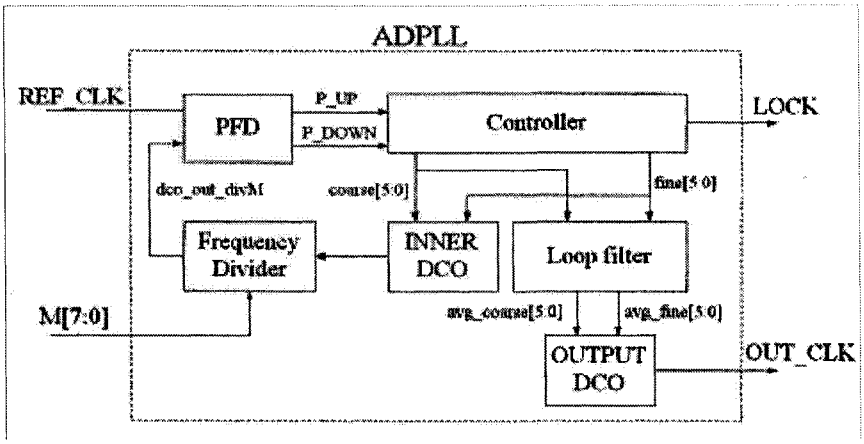


Figure 2.15 An all-digital phase-locked loop for high-speed clock generation

The ADPLL controller updates the control word of the inner DCO according to the output of the PFD. The control word of the inner DCO is also sent to an low-pass filter to generate the control word for the output DCO.

This ADPLL uses an adaptive search step when it searches for the target frequency. The frequency acquisition starts from the middle frequency band of the DCO and the initial frequency step is one quarter of the frequency tuning range. This frequency step is divided by two whenever the output of the PFD changes its polarity until it is reduced to the fine-tuning step of the inner DCO and the frequency acquisition is done. If the DCO's control word is updated once every *m* reference cycles, the worst case lock-in time for this frequency acquisition, in term of reference clock cycles, can be calculated as

$$T(n) = m \cdot \left(1 + 2 \cdot \log_2\left(\frac{n}{2}\right)\right) = m \cdot (2\log_2 n - 1) \qquad (2.19)$$

where n is the total number of different frequency levels of the inner DCO.

The DCO in this ADPLL has a frequency tuning range from 41MHz to 545MHz from HSPICE simulation. As shown in Figure 2.16, the frequency tuning is achieved by two stages: a coarse-tuning stage and a fine-tuning stage. In the coarse-tuning stage, the coarse-tuning delay chain with 64-to-1 select-path architecture is used to provide different delays for coarse tuning with a 300ps step on the DCO period. To increase the frequency

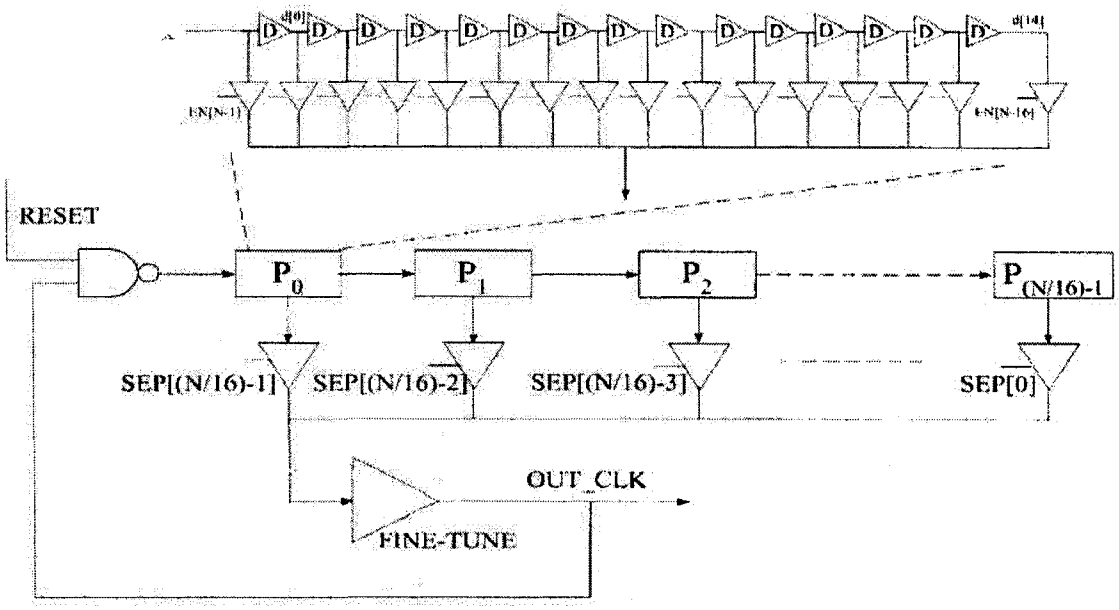


Figure 2.16 The DCO of an ADPLL for high-speed clock generation

resolution, a fine-tuning delay cell is added after the coarse tuning stage. The fine-tuning

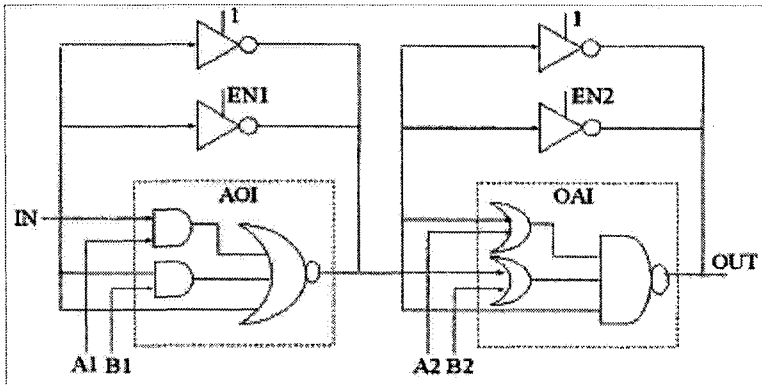


Figure 2.17 The fine tuning stage of the DCO

delay cell consists of an AND-OR-INV (AOI) cell and an OR-AND-INV (OAI) cell. Both the AOI cell and the OAI cell are shunted with two tri-state buffers. Shunted tri-state buffers can increase the controllable range of the fine-tuning delay cell. The controllable range of the fine-tuning delay cell should cover one coarse-tuning step (i.e., 300 ps). In the fine-tuning delay cell, six bits (EN1, A1, B1, EN2, A2, B2) can be controlled. Thus, in total 64 different delays can be used. After HSPICE simulation, the lookup table for mapping the fine-tuning control code can be created. The DCO resolution can be improved to about 5 ps by

adding a fine-tuning delay cell. Because the PFD of this ADPLL only needs to detect the polarity of the phase error, digital pulse amplifiers are used to minimize the dead zone of the PFD.

The measured RMS jitter and the peak-to-peak jitter are 7ps and 20ps respectively at 45MHz and are 22ps and 70ps at 450MHz.

2.5.4 A fast lock digital phase-locked loop

One of advantages of the digital PLL is that some advanced locking behaviors can be achieved by changing the loop parameters and behaviors on the fly. A fast phase lock DPLL is reported in [4] (similar approaches are found in [5] and [6]) and its block diagram is shown in Figure 2.18. This DPLL employs both the frequency comparator and phase

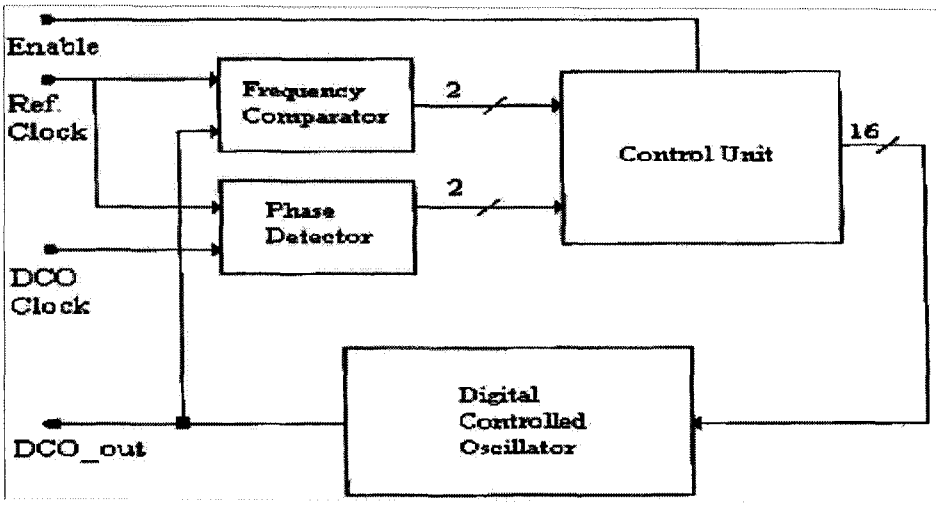


Figure 2.18 An all-digital PLL with small DCO hardware and fast phase lock

detector. The frequency comparison is done by initially aligning the DCO output to the reference edge and comparing the *N*th DCO transition edge with the next reference edge. A complete frequency comparison needs two reference clock cycles. The frequency comparator produces signals, "slow" or "fast" to indicate if the DCO frequency is lower or higher than the desired frequency.

Like many other DCOs, the DCO in this ADPLL is based on a ring oscillator and

the tuning is achieved by changing both propagation delay of one signal stage and total number of inverters. By changing the number of inverters, four modes (frequency bands) are obtained. The propagation delay tuning is achieved by tuning the current of the inverters as shown in Figure 2.19. Each inverter is cascaded with 14-bit control MOS devices. The size ratios of the control devices are two times as shown in the figure. The most significant control bit, bit 13, corresponds to the largest control devices. According to the simulation, the least significant bit resolution is 177ps when the most significant control bit (bit 13) is asserted. As a result, the DCO has four frequency bands and 2^{14} frequency

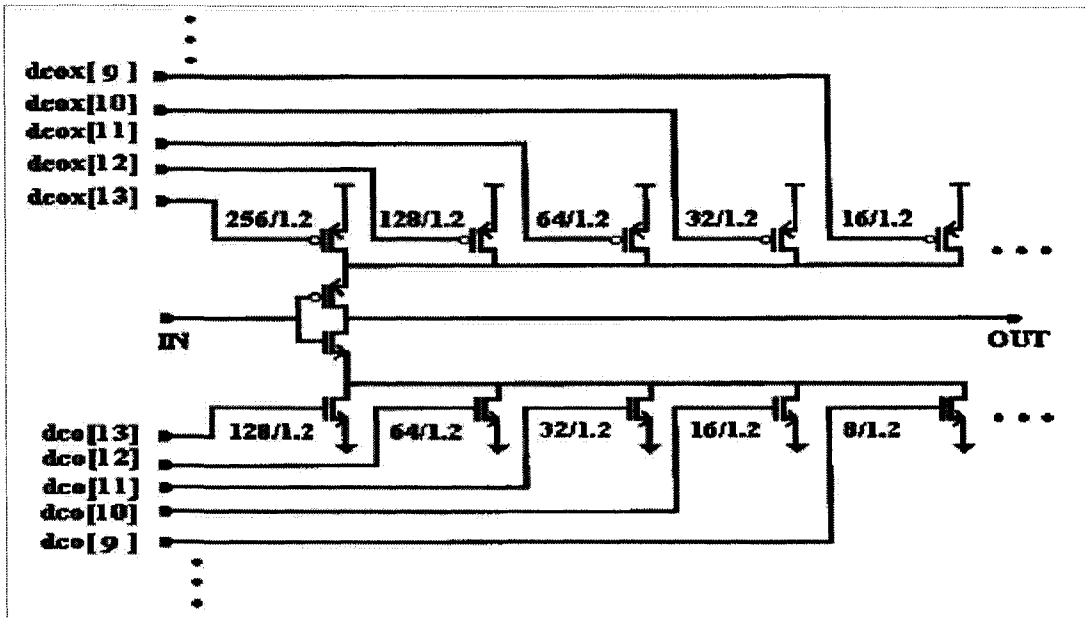


Figure 2.19 A digitally controlled delay cell for the DCO

levels within each band.

A fast frequency/phase acquisition approach is used in this ADPLL and it can finish both frequency/phase acquisition within 47 reference cycles: at most 8 cycles are needed to find the correct mode/band, and at most 28 reference cycles are need to acquire the frequency with a binary search method, further more, at most 11 reference cycles are need to achieve the phase acquisition.

After both the frequency and phase acquisitions are done, the digital PLL still dynamically adjusts the phase-gain value to reduce the phase error. When the phase polar-

ity is changed, the phase-gain value is divided by two, and the phase-gain is multiplied by two whenever the ADPLL detects eight consecutive incremental changes in the same direction.

The implementation of this ADPLL requires 4026 transistors and with 923μm X 921 μm layout area in a TSMC 0.6μm SPDM CMOS process. The ADPLL runs up to 400MHz with a supply voltage of 3.3V.

2.5.5 A fast lock all-digital PLL for frequency multiplication by 4 to 1022

In most DPLLs, the phase and frequency detector and the DCO are independent and their operating resolution do not coincide with each other. A DPL, which combines the PFD and DCO, is reported in [7], and its system diagram is shown in Figure 2.20. The time to digital conversion circuit, which is usually in the DPFD, is implemented together

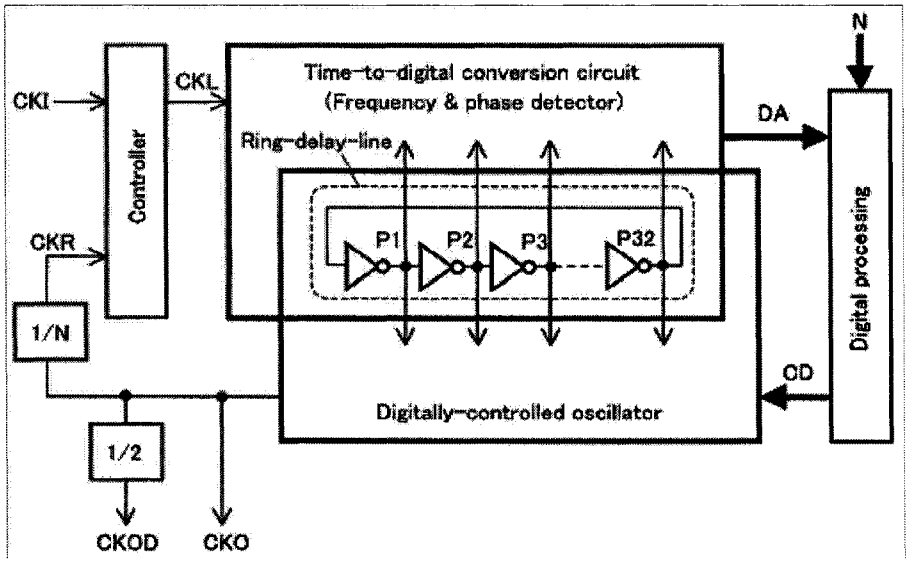


Figure 2.20 An DPLL with matched operating resolution of PFD and DCO

with the multiple-stage ring type DCO. Because the time difference between the two consecutive signals of the DCO is proportional to the period of the DCO, the time resolution of the T2D is proportional to the DCO period. This characteristic is desirable for the

DPLL because the loop behavior can be independent to the operation frequency. The frequency acquisition of this DPLL is achieved by directly measuring the DCO frequency in terms of the reference frequency, and calculating and updating the DCO control word.

This DPLL is fabricated with a 0.65μm CMOS, and the core size is 1.08x1.08mm^2. It has an output frequency range from 44.9kHz ~ 61.3MHz with the reference clock frequency range from 11.2kHz ~ 339.7kHz. THe output clock peak to peak jitter is 1.59ns and the standard deviation is 234ps.

2.5.6 An all-digital PLL clock multiplier

Another ADPLL clock multiplier, implemented in 0.35μm standard CMOS process, operating from 152MHz to 366MHz, is reported in [8]. Figure 2.21 shows the block

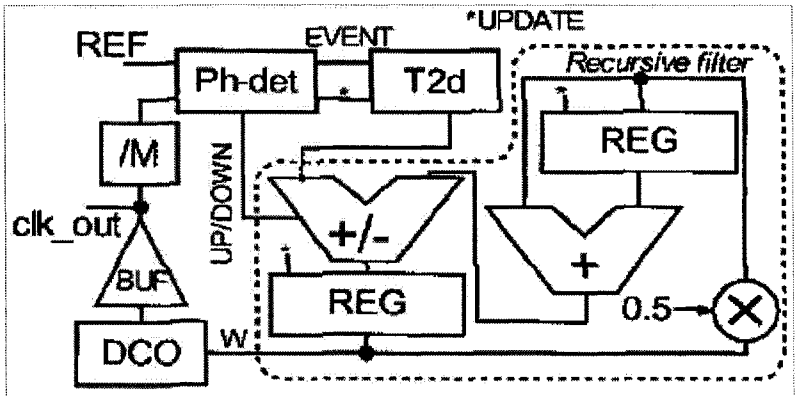


Figure 2.21 An all-digital PLL clock multiplier

structure of the all-digital PLL. The oscillator for this digital PLL is a 7 state ring oscillator with one inverter replaced by a NAND-gate for shutting down the ring oscillator during idle mode. To change the frequency of the ring oscillator, a set of 21 inverting tri-state gates are connected in parallel with each inverter. When the tri-state gates are enabled, additional current drive is added to each inverter state. The 126 (21x6) tri-state gates are controlled by 126 bit vector, which is decoded from a 7 bit control word. This tuning method guarantees the monotonic frequency tuning because adding current drive by

enabling a tri-state gate speeds up the oscillator. An advantage of this DCO is that it can be made from all-standard cells. However, this DCO results relatively high power consumption and nonlinearity.

The phase detector is a modified tri-state PFD and it produces one signal "DIRECTION", showing the polarity of the phase error and one signal "EVENT" showing the length of the phase error as shown in Figure 2.22. Another signal "UPDATE" is a short

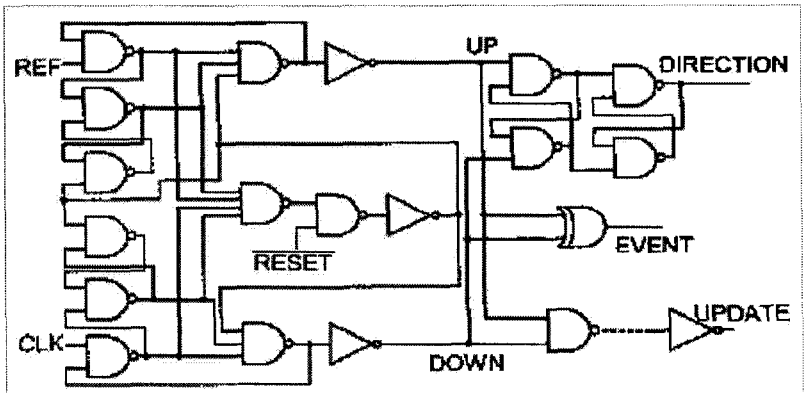


Figure 2.22 The PFD for an all-digital PLL clock multiplier

pulse at the end of each phase error, and it works as a clock signal for the updating all registers in the T2D and the loop filter, which is a digital recursive filter (as shown in Figure 2.21).

2.5.7 A PVT tolerant, self-calibrated digital PLL

A self-calibrated digital PLL implemented in 90nm CMOS process is reported in [12]. This digital PLL has wide reference frequency range (0.03M-6MHz), and the output frequency range is 0.18M-600MHz with near constant damping factor and loop bandwidth to reference frequency ratio, independent of the reference frequency, the divider ratio, and the PVT variations. The low jitter and wide operation range are achieved through (1) a logarithmic time to digital converter (T2D), (2) a start-up auto-calibration and normalization algorithm, (3) a novel digitally controlled oscillator (DCO) with low voltage opera-

tion and high PSRR.

The block diagram of the PLL is shown in Figure 2.23. The input clock divides

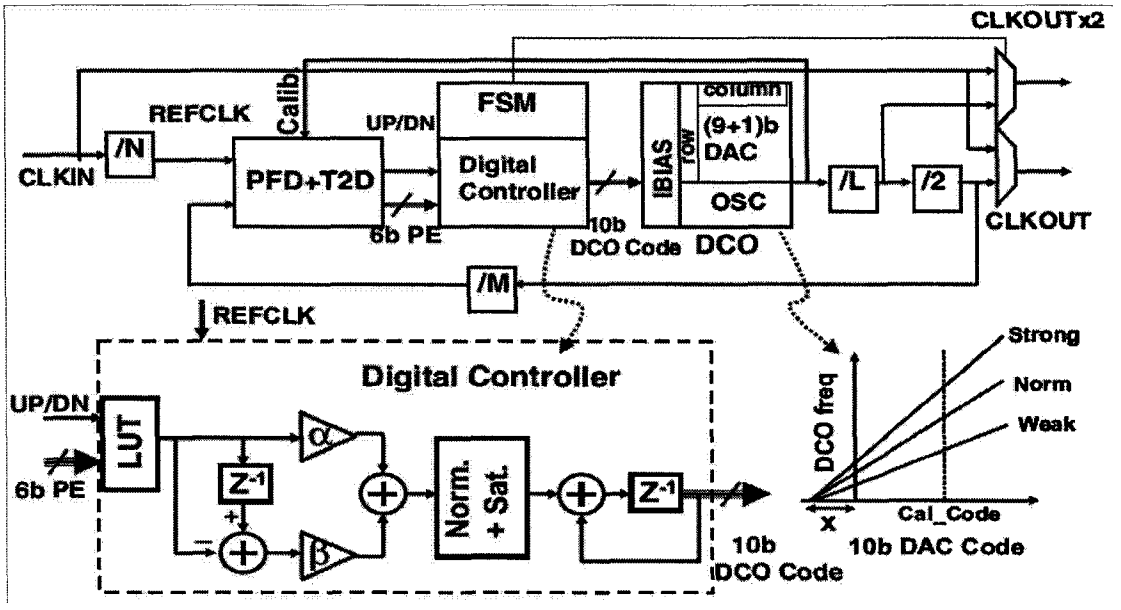


Figure 2.23 A PVT tolerant self-calibrated digital PLL

into the REFCLK. PFD and T2D blocks compare the feedback clock (FBCLK) and REFCLK to generate the signed, logarithmically compressed 6b digital representation of the phase error in terms of the number of inverter delays. Digital controller then linearizes the 6b phase error through Look-Up-Table (LUT), and calculates the required DCO code with the digital filter, normalization block, and accumulator. The DCO frequency vs. code transfer function is almost linear. The wide output frequency (>300x) is achieved through the "smart" L divider, with the ratio determined during the start-up calibration phase by setting the DCO at a pre-determined code and comparing the frequency difference between the divide by M clock and REFCLK.

To maintain the constant loop bandwidth to REFCLK frequency ratio and constant damping factor, this DPLL performs a start-up calibration and normalization by calibrating the DCO period with respect to the T2D delay.

The wide input frequency range is achieved through the logarithmically compressed time measurement (T2D) and a linearization LUT in the digital domain. The digital PFD is shown in Figure 2.24, in which the phase/frequency error is first detected by a

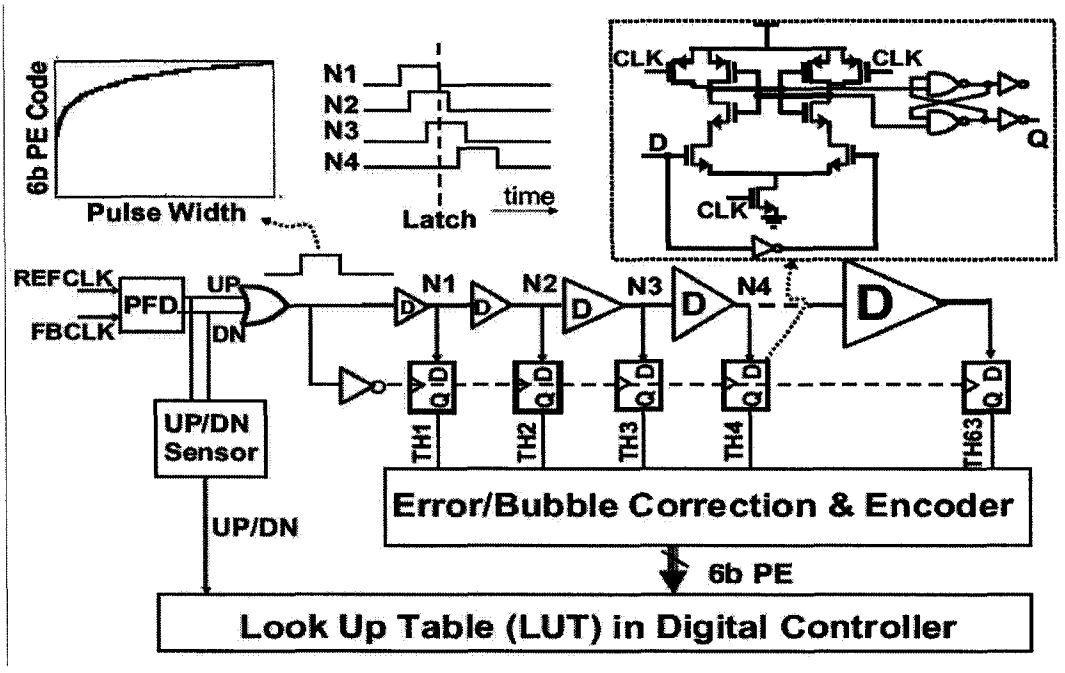


Figure 2.24 The DPFD of the PVT tolerant self-calibrated digital PLL

tri-state PFD and UP/DN pulses are generated. The sign of the phase error is determined by a UP/DN sensor and the absolute value of the phase error is represented by the pulse width of the output of a XOR gate. The phase error (PE) pulse width is converted to 6b PE code through the self-terminating exponentially increased delay chain. This compressed PE code is linearized by the LUT. The combination of logarithmic T2D and LUT significantly increases the reference frequency range with very little area and power penalty. Although quantization error increases with the delay chain, it does not affect the output jitter percentage in the locked state.

The DCO in this PLL first converts the digital tuning word to a current with a (9+1)b segmented current steering DAC. This resulting current is to tune a differential ring oscillator.

2.5.8 *A phase-domain all-digital phase-locked loop*

Another all-digital phase-locked loop is reported in [13], in which the DCO phase is first digitalized and compared with a reference phase ($R_R[k]$) which is also represented with digital signal. The bock diagram of this all digital PLL is shown in Figure 2.25. The

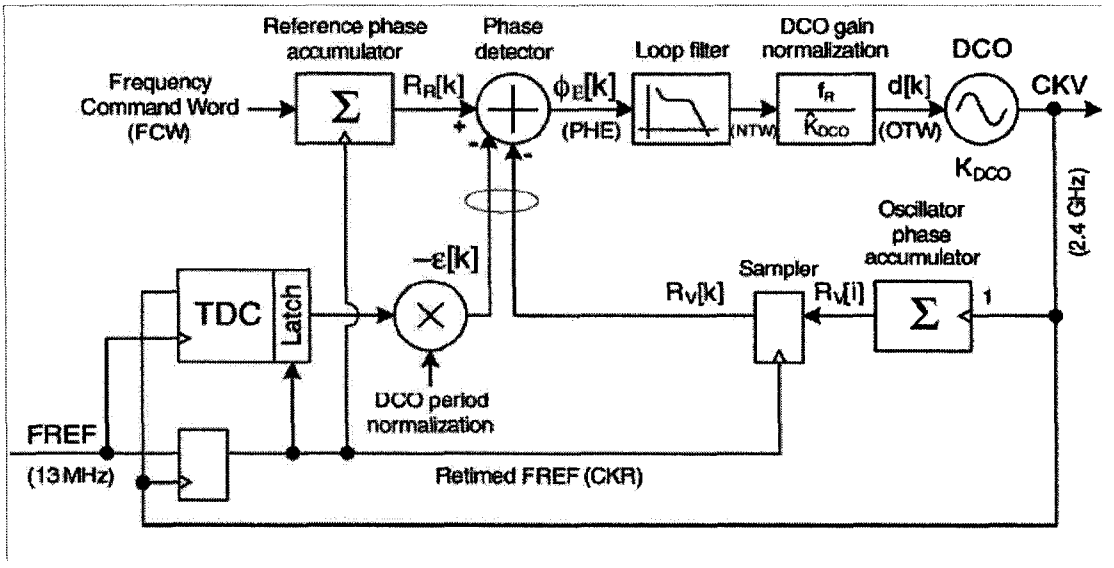


Figure 2.25 Phase-domain ADPLL-based frequency synthesizer

digital reference phase is provided by a accumulator. The DCO phase is digitized at each reference edge. An oscillator phase accumulator is used to count the DCO cycles to provide the integral part of the DCO phase. A time to digital converter (TDC) [22] and a DCO period normalization circuit are used to obtain the fractional part of the DCO phase. The integral part and the fractional part are summed together and compared with the reference phase with the phase detector, which is essentially a subtracter. The output of the phase detector is thus filtered by the digital loop filter and controls the DCO after the DCO gain normalization.

The output frequency of this frequency synthesizer can be changed by changing the frequency control word (FCW), thus the reference phase ($R_R[k]$)

2.6 *Summary of the literature review*

The review of digital PLL-based frequency synthesizers shows that all reported digital loop filters to date are implemented mathematically based on their desired z-domain transfer functions, operating with binary weighted signals, using multipliers, adders, and memories and so on. Although the design of such filters is straight-forward, and the loop analysis can be easily done by using widely available techniques for analog PLL, such filters and the input/output interface from/to the digital PFD/DCO normally require a complex hardware, thus large area and high power consumption. The DCOs, digital PFDs, and the frequency dividers of the digital PLL are summarized separately in the following paragraphs.

Based on the mechanism of the oscillation, DCOs, similar to VCOs, can be classified into three categories, direct-frequency-synthesis-based DCOs [1][15][18], ring-based DCOs [2][3][4][8][9][16][17][21][21] and LC-based DCOs [10][11][19] as illustrated in Figure 2.26. Direct-frequency-synthesis-based DCOs require a high-frequency or multi-

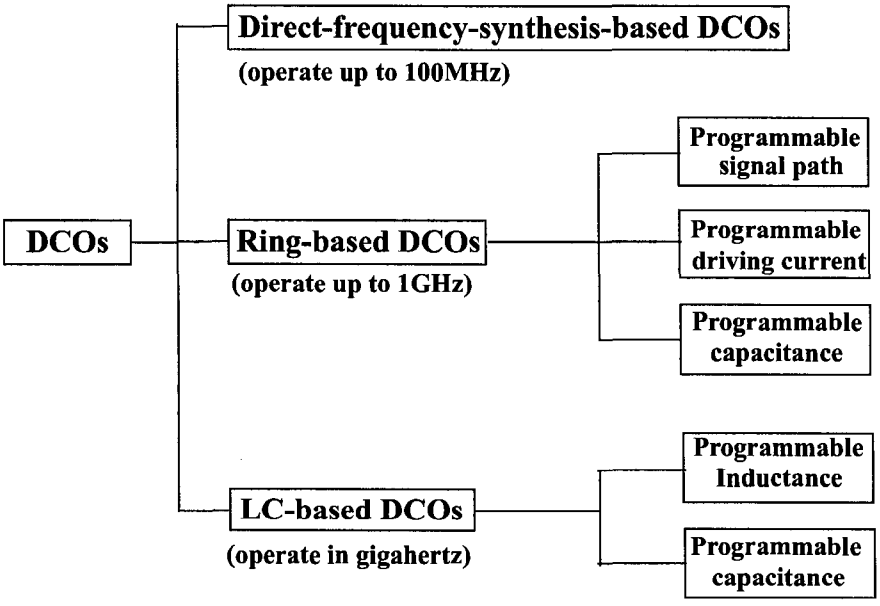


Figure 2.26 Classification of DCOs

ple-phase clock and the output is generated by dynamically dividing the high frequency

clock or selecting one phase from multiple phases. Because of the nature of the frequency generation, those DCOs normally operate below 100MHz. Although ring VCO can easily operate up to gigahertz, ring DCOs hardly operate in gigahertz because the digital frequency tuning units normally slow down the oscillation significantly. A DCO, which is a combination of a DAC and a ring-based VCO[2][12], may operate in gigahertz, but the analog control signal between the DAC and the VCO is a high noise sensitive node, which should be avoided to benefit the advantage of a DCO on the phase noise performance. The frequency tuning of a ring-based DCO is achieved by changing the stage delay, such as programmable signal path [3][17], programmable current[2][4][9][12], programmable capacitance [3]. The first multi-gigahertz DCO for wireless applications, based on a LC tank oscillator, was reported in [10][13]. Another LC based DCO was reported in [11]. The conventional varactor of a LC-VCO is replaced by a digitally-controlled capacitor (D2C). A LC DCO with both programmable capacitance and inductance is reported in [19] Similar as VCOs, LC-based DCOs normally provides better noise performance than ring-based DCOs. If all frequency tuning units are biased in their low-gain regions, a DCO may achieve better noise performance than its analog counterpart. In the proposed digital PLL described in the next chapter, a LC-based DCO, with a novel D2C scheme is used to operate at approximately 3GHz.

Because the reference signal is intrinsically analog in nature, the digital phase/frequency detector acts as an analog to digital converter, which converts an analog value (phase error) to its digital representation. In [1], this conversion is done by counting the DCO output cycles and examining the result at the edge of the reference. The time resolution in this case equals to one DCO period. A simplified digital PFD was reported in [2], where only the sign of the phase error is determined by the PFD, and the amplitude of the phase error is dynamically estimated by a shift register controlled by a digital state machine along the locking process. A combination of a PFD and a T2D, employed as a DPFD, is reported in [12]. The T2D actually quantizes the pulse width, representing the

phase/timing error. A phase-domain DPLL is reported in [14], whose DPFD is a combination of TDC and some digital signal processing circuits. The TDC has a time resolution corresponding to a single inverter delay, and its output is normalized to the DCO period to maintain a consistent loop behavior for different operating frequencies. The high time-resolution and linearity makes it suitable for RF applications, where the frequency ratio between the output frequency and the input frequency is fractional and a direct frequency modulation is needed.

A conventional frequency divider is still used for many digital PLLs[2]-[8][12] in the same manner as analog PLLs. For some Digital PLLs [1][14], the frequency division function is implicitly implemented together with the phase/frequency detection circuit.

CHAPTER 3

A proposed digital PLL-based frequency synthesizer

3.1 System architecture of the proposed digital PLL

Based on the literature review in Chapter 2, to date, most reported DPLL-based frequency synthesizers are digital versions of their analog PLL counterparts, in that each blocks are digitized separately, resulting in a high-complexity digital circuit. Figure 3.1

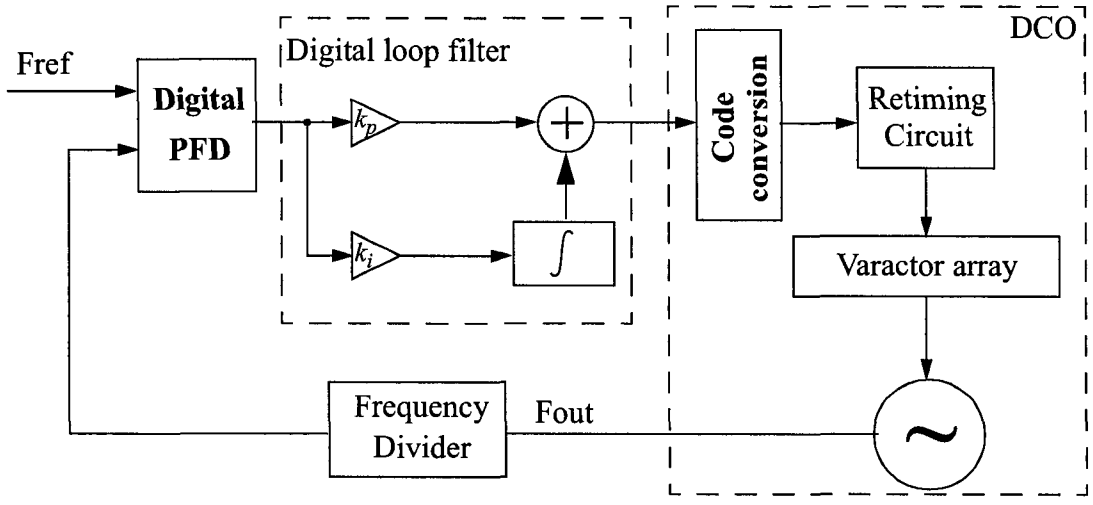


Figure 3.1 A typical digital PLL-based frequency synthesizer

shows the block diagram of a typical digital PLL-based frequency synthesizer using an LC-based DCO. A digital PFD detects and digitizes the phase/frequency difference between the divided oscillator output signal and the reference signal, then a commonly used type-II digital loop filter, which consists of a proportional path and an integral path,

is used to filter the output of the digital PFD. The output of the digital loop filter controls the frequency of the digitally controlled oscillator. Because binary weighted signals are normally employed in the digital loop filter, while the digital frequency tuning units (i.e. the varactor array) inside the DCO normally prefers linearly coded signals such as the thermometer code, the 'one-hot' code and so on to achieve better linearity, the output signals of the digital loop filter need to be converted and re-timed before they are connected to the digital tuning units.

A low-complexity digital PLL architecture is proposed in this work as shown in Figure 3.2. It eliminates the need for the code conversion and significantly simplifies the

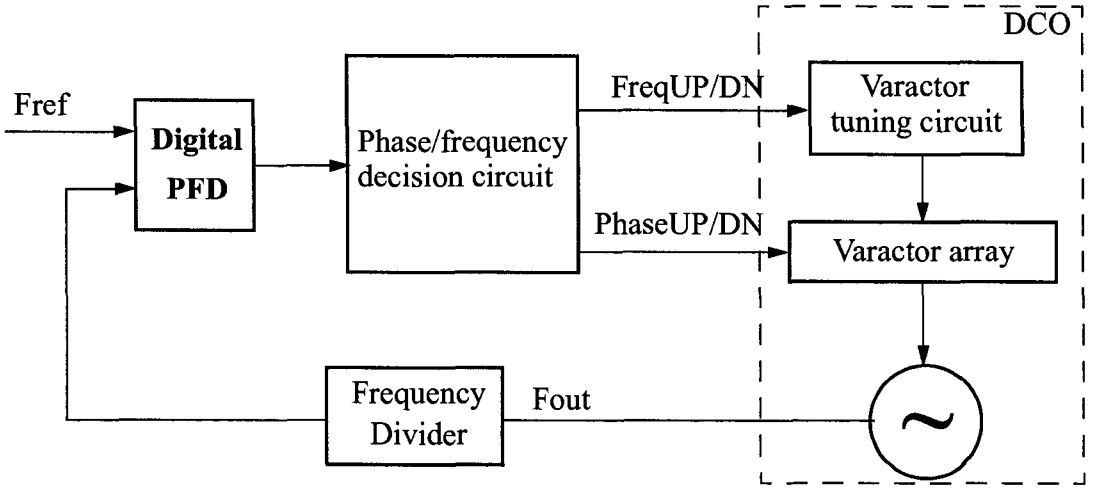


Figure 3.2 A proposed digital PLL-based frequency synthesizer

digital phase/frequency detector, the digital loop filter, and the DCO, while it still provides similar performance in the locked condition. In addition, the proposed DPLL is able to adapt to different reference phase noise and maintain an optimum closed-loop phase noise performance. The proposed architecture, including a DPFD, a decision circuit, a DCO and a frequency divider, has a very low complexity and a low power consumption. It is important that the proposed digital PLL is dedicated for integer-N frequency synthesis applications where the noise performance in the locked condition is of concern.

In a digital PLL, the locked condition is achieved and maintained by dynamically tuning the DCO, which includes frequency tuning (increasing or decreasing the frequency) and phase tuning (a positive/negative phase shift). In the PLL shown in Figure 3.1, the phase tuning is achieved by the proportional path with the gain of K_i, and the frequency tuning is achieved by an integrator with the gain of K_i. The results are added together to tune the DCO.

In the proposed DCO, the retiming circuit (a set of samplers) is configured as a varactor tuning circuit which accepts inputs of "*FreqUP*" and "*FreqDN*" and produces one-hot coded signals (for the LSB bank) or thermometer-coded signals (for the MSB bank). The outputs of the varactor tuning circuit are held until the signal *FreqUP* or *FreqDN* comes.

Figure 3.3 illustrates the basic principle of such a varactor tuning circuit. The out-

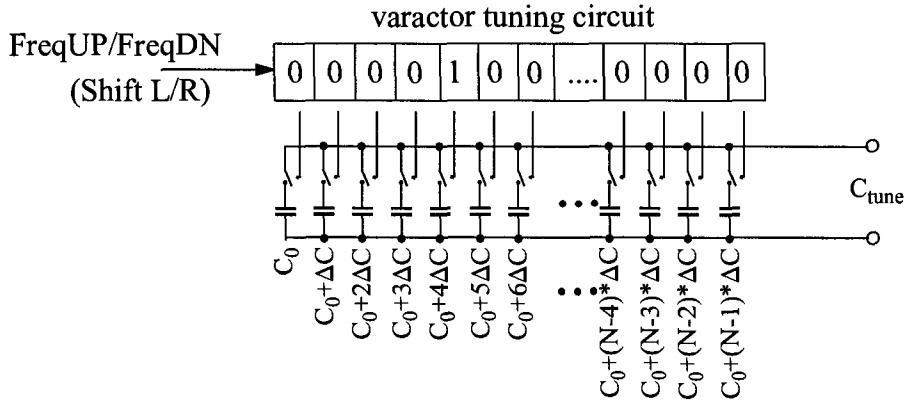


Figure 3.3 The basic principle of the varactor tuning circuit

puts of the varactor tuning circuit control a bank of incrementally sized varactors respectively. These bits are pre-loaded with the one hot coded bits, "0...010...0". A right shift of those control bits results in an capacitance increase of ΔC, while a left shift results in an capacitance decrease of ΔC. Consequently, the frequency tuning is achieved by shifting to the left or to the right operations.

To achieve the phase tuning feature of the DCO, separate varactors are controlled by *PhaseUP*/*PhaseDN* directly. As we will see in the next chapter, the phase noise performance of the proposed digital PLL is mainly determined by the phase tuning path. In the proposed architecture, the varactors related to the phase tuning path are not dynamically changed like most other digital PLLs. Because only a small number of varactors are for the phase tuning path, they can be laid out separately from other varactors related to the frequency tuning path to achieve a better matching accuracy. Actually, the phase tuning path has only two states, *PhaseUP* and *PhaseDN*, in the normal operation, so the mismatch of those varactors in the path tuning path affects nothing but the gain of the phase path, which corresponds to the capacitance difference between the *PhaseUP* and *PhaseDN*. By making the gain of the phase tuning path programmable, such mismatch can be calibrated and compensated easily.

Implementing the frequency tuning mechanism in the DCO simplifies the DCO interface and eliminates the need for the integrator of the digital loop filter in Figure 3.1. In the proposed digital PLL, the complex digital loop filter is reduced to a simple phase/frequency decision circuit. By employing different phase/frequency decision algorithms in the decision circuit, the digital PLL can exhibit different loop behaviors.

Two terminologies, "frequency tuning" and "phase tuning" were introduced in this proposed low-complexity digital PLL. Although both of them actually change the frequency of the DCO, the frequency tuning changes the DCO frequency permanently while the frequency change (Δf_p) of the phase tuning only exists within a given period of time(Δt), after which the frequency goes back and the resulting phase change ($2\pi\Delta f_p \cdot \Delta t$) stays permanently. From mathematical point of view, the frequency tuning and the phase tuning are equivalent to the integral path and the proportional path of digital loop filters in most reported digital PLLs. In this proposed DPLL, the terminologies of "frequency tuning" and the "phase tuning" are employed for readers to better understand the fact that the frequency tuning and the phase tuning are integrated seamlessly into the

DCO, instead of the digital loop filter.

3.2 The proposed digitally-controlled oscillator

A DCO is very similar to a VCO except that the DCO frequency is tuned digitally while the VCO's frequency is tuned by an analog voltage signal. In this work, an LC-tank based oscillator is used and a digitally-tuned capacitor is used as the tuning unit, as shown in Figure 3.4. The analog part of this DCO is a capacitance-controlled oscillator (CCO),

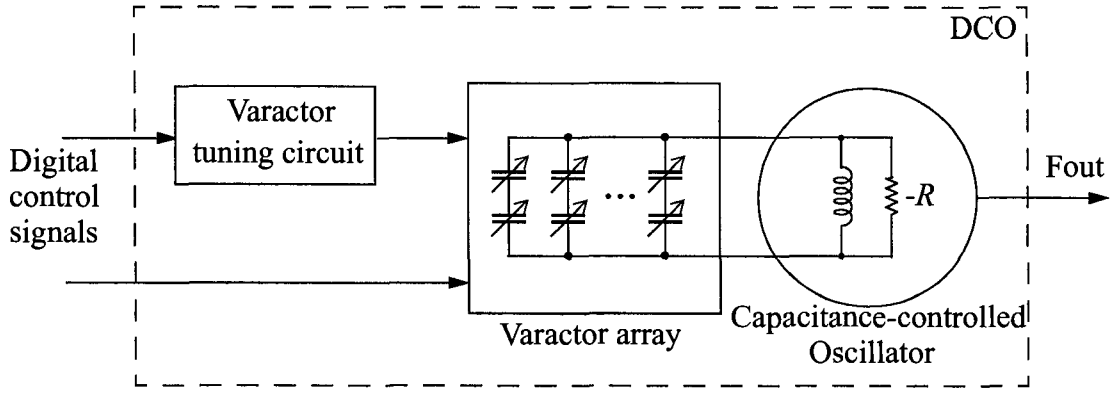


Figure 3.4 The architecture of the proposed DCO

which can be the same as the corresponding part of a VCO. This section focuses on the digital capacitance tuning block.

The digital capacitance tuning is normally realized by engaging or disengaging some unit-sized varactors, so the smallest frequency step size, corresponding to the minimum capacitance step size, is usually limited by the smallest varactor size for a given technology. In [13], the required frequency resolution is achieved by frequency dithering using a sigma-delta modulator, which increases both the hardware complexity and the power consumption. In this work, a capacitance tuning scheme for an LC-tank based DCO with incrementally sized varactors and matched varactor banks is proposed. It can achieve both the high frequency resolution (small frequency step, which is approximately 5 kHz in the DCO test chip) and the large frequency tuning range with a small differential nonlin-

earity (DNL, which is approximately $\pm 0.5LSB$ based on the measured results).

A MOS varactor normally has two flat regions in its V-C characteristic curve as illustrated in Figure 3.5 and they can be used as two levels of digital switch to minimize

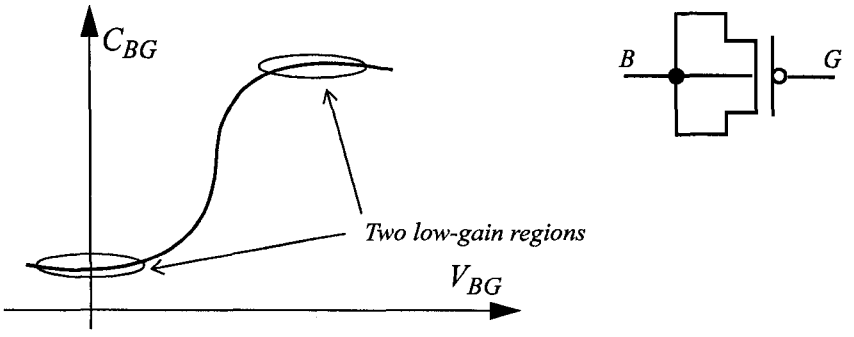


Figure 3.5 Tuning characteristics of a PMOS transistor

the analog VCO gain. However, simply using a set of such varactors results in that the minimum capacitance step is the same as the capacitance difference between two flat regions of a MOS varactor with the minimum size determined by a given technology. In the proposed DCO, a set of incrementally sized MOS varactors is employed as the LSB bank to enhance the capacitance resolution, and a set of unit sized MOS varactors is used in the MSB bank. They are physically matched to minimize the differential non-linearity, specifically for the transition between LSB bank to the MSB bank. Both banks employ linearly coded signals: the one-hot coded signals for the LSB bank and the thermometer coded signals for MSB as shown in Figure 3.6. To operate with the proposed varactor banks, the varactor tuning circuit (essentially, the bi-directional shift register) is split into two parts, LSB part and MSB part. The LSB part operates with the one-hot code and the MSB part operates with the thermometer code, which is updated by the overflow flags (borrow or carry) of the LSB part.

It is clear that the step size of the MSB bank should be matched with the tuning range of the LSB bank to minimize the differential non-linearity and to guarantee the monotonic frequency tuning. To achieve a better matching accuracy between the MSB

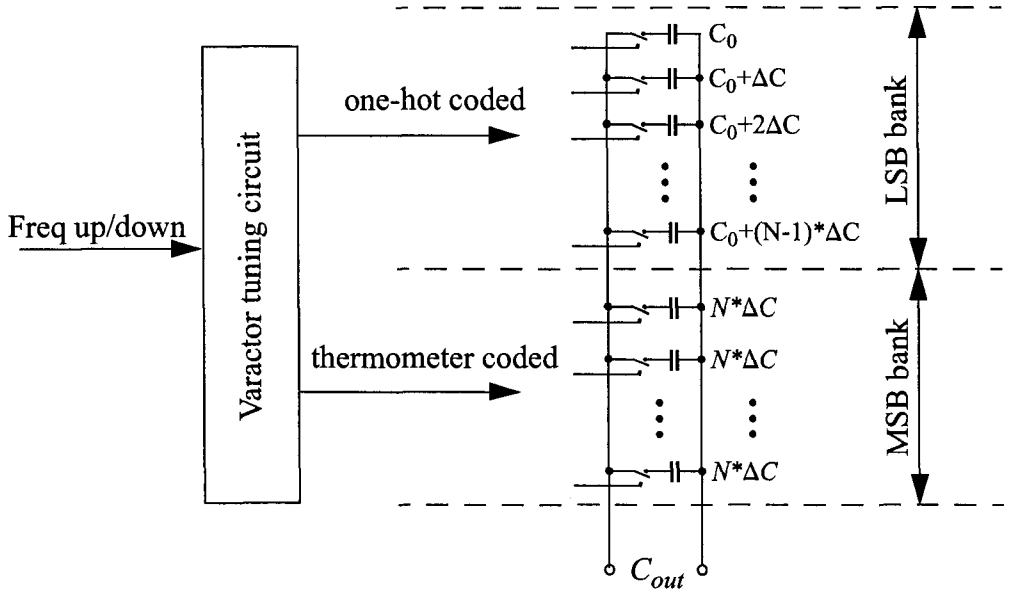


Figure 3.6 The frequency tuning of the proposed DCO

bank and the LSB bank, each unit of the MSB bank employs two varactors (C_0 and $C_0 + N \cdot \Delta C$), which are easier to be matched with the LSB bank than one single varactor ($N \cdot \Delta C$) is. The capacitance step of the MSB bank is achieved by taking the difference of those two varactors: $N \cdot \Delta C = (C_0 + N \cdot \Delta C) - C_0$. Using the incremental sized varactors in the LSB bank not only enhances the frequency resolution, but also helps improve the matching accuracy between the LSB bank and the MSB bank. More details can be found in the section "High resolution varactor array" on page 100.

Besides the LSB/MSB banks shown in Figure 3.6, a coarse frequency tuning bank (the band tuning bank), which employs a set of binary sized varactors and is controlled by binary coded signals is implemented to further enlarge the frequency tuning range. A certain frequency overlap has to be maintained among different bands to guarantee the continuous frequency coverage.

The phase tracking of the DPLL is achieved by implementing a separate phase tuning bank in the DCO as shown in Figure 3.7. The phase up/down is achieved by reducing/ increasing the total capacitance for a certain period of time. The phase tuning bank and the

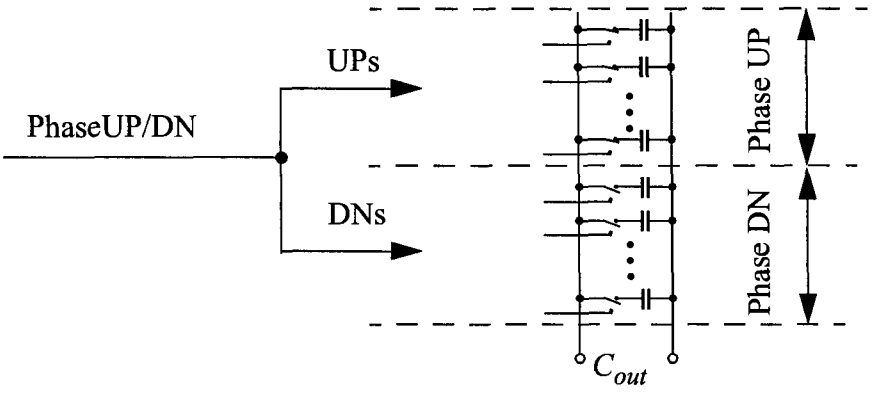


Figure 3.7 The phase tuning of the proposed DCO

frequency tuning bank are added together inherently by connecting them in parallel, resulting in both phase/frequency tuning in the proposed DCO.

3.3 The proposed digital PFD

The digital PFD compares the reference signal with the divider output, and produces digital representation of the phase/timing error. A digital PFD is normally implemented as (or mathematically equivalent to) as a combination of an analog PFD and a time to digital converter (T2D). In a fractional-N frequency synthesizer, the PFD phase error is usually non-zero in the ideal locked condition, so high resolution/linearity of the phase error quantization is desired. However, in an integer-N frequency synthesizer, the ideal phase error is zero in the locked condition and the quantization of a small phase error is of concern for the noise performance in the locked condition.

For a linear T2D having the transfer characteristic as shown in Figure 3.8, the timing resolution of the T2D doesn't affect the locked condition as long as the timing error is bounded between -Δ and Δ. In this case, the digital PFD is essentially a bang-bang PD, producing only signals *UP* or *DN* according to the polarity of the phase error.

Based on the analysis above, a low-complexity digital PFD is proposed as shown in Figure 3.9. It consists of a tri-state PFD with a cycle slip detection circuit and an *UP/*

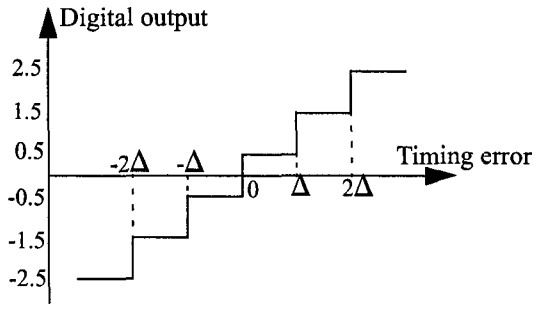


Figure 3.8 The quantization of a linear T2D

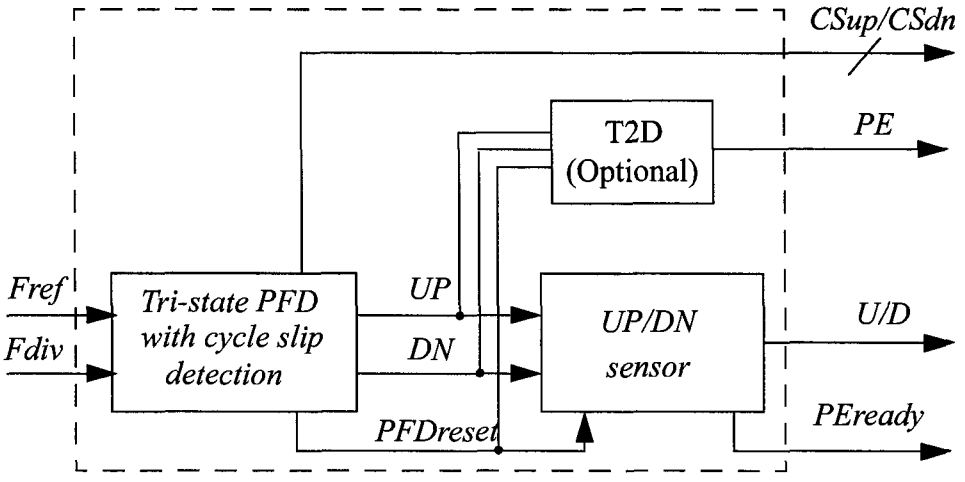


Figure 3.9 The proposed DPFD

DN sensor. The PFD compares the phase/frequency of its inputs (*Fref*, *Fdiv*) and produces two pulses, *UP*/*DN*, whose width difference represents the phase/frequency error. The *UP*/*DN* sensor compares *UP*/*DN* to determine the polarity of the phase/frequency error while a T2D converts the absolute phase error to a digital value. In the DPFD, the *UP*/*DN* sensor must have a very small hysteresis comparing with the jitter requirement of the whole digital PLL.

With the polarity information of the phase error (U/D), the decision circuit produces phase/frequency up/down, adjusting the DCO to achieve, as well as to maintain, the locked condition of the digital PLL. However, the frequency/phase acquisition speed is usually low without the information of the phase error amplitude. A cycle-slip detection is

proposed in the PFD so that *CSup/CSdn* signals are generated whenever a cycle-slip happens or the phase error is above one cycle (2π). The decision circuit responds to the signals *CSup/CSdn* by fast adjusting the DCO frequency. To further reduce the frequency/phase acquisition time, more phase error levels can be implemented, such as $\pi/8$, $\pi/4$, with a T2D converter. Figure 3.10 shows an implementation of the T2D converter, which per-

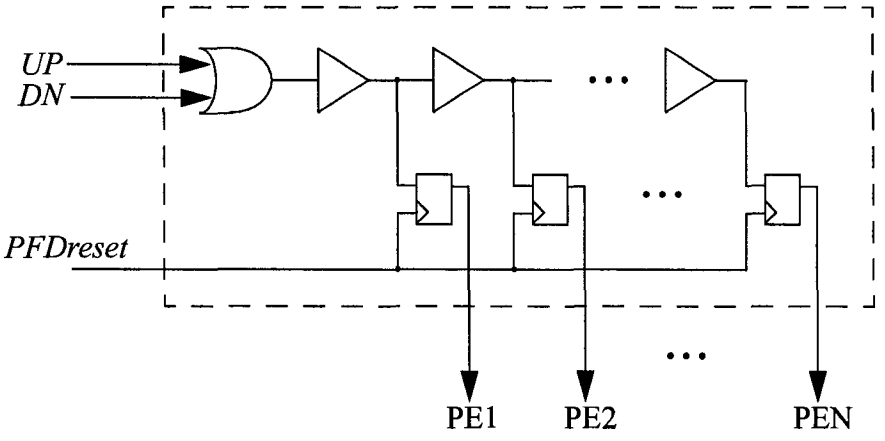


Figure 3.10 An optional T2D in the proposed DPFD

forms the logical operation OR to its input signals *UP/DN*, and the result is delayed by a serial of delay cells. Those delayed signals are sampled on the rising edge of *PFDreset* to produce the output signals, *PE1*, *PE2*... *PEN*, indicating whether the amplitude of the phase error is larger than the pre-defined threshold values (the accumulated delays of those delay cells) respectively.

3.4 The proposed decision circuit

The phase/frequency error between the reference clock and the divided clock is obtained from the digital PFD and the result is used to adjust the frequency or the phase of the DCO with some kind of decision algorithms. A simple algorithm is proposed to be used in the decision circuit to achieve a low complexity implementation.

When the digital PLL is in locked condition, the phase/timing error appearing at the DPFD inputs is typically small compared with the quantization step size of the T2D

converter. Consequently, the decision circuit only needs to react to the outputs of the *UP/DN* sensor (*U/D*) to achieve the desired phase noise performance in the locked condition.

In the phase decision circuit, the outputs of phase *up/down* can be generated directly from *U/D*. A positive phase error results in a pulse of '*phase down*' while a negative phase error results in a pulse of '*phase up*'. The amount of actual phase change ($2\pi\Delta f_p \cdot \Delta t$) is proportional to the width of output pulse (Δt), which is usually one period of the reference signal.

The decision on the frequency *up/down* is made after each updating period, *M* reference cycles or divider cycles, based on the signs of phase error during that updating period. If there are only positive phase errors in the updating period, '*frequency down*' is generated. Similarly, if there are only negative phase errors in the updating period, '*frequency up*' is generated. Otherwise, the frequency keeps unchanged. In this decision algorithm, the frequency updating speed can be changed by changing the updating period. The updating period can even be determined adaptively to achieve some advanced features.

To implement frequency decision algorithm above, two latches are needed to indicate whether the positive/negative phase errors ever appeared during the updating period, and both latches are reset at the beginning of each updating period once the decision is made.

To achieve a faster frequency acquisition, more quantization levels of phase/frequency error can be generated by the digital PFD. Similar decision method can be applied to each level individually and the outputs of phase and frequency decision circuits for a larger phase/frequency error would result in a larger step size of the phase/frequency adjustment of the DCO.

Unlike many other reported digital PLLs, which are designed to be as linear as possible, the proposed digital PLL is deliberately designed as a nonlinear system to achieve low complexity, thus the conventional frequency domain analysis cannot be

directly applied to such systems, resulting some difficulties in the system analysis.

3.5 Summary

A low-complexity non-linear digital PLL-based integer-N frequency synthesizer is proposed, and the system level operation is explained in this chapter. In the next chapter, the loop behavior and the phase noise performance of the proposed DPLL is analyzed theoretically and some behavioral modeling simulations are done to confirm the operation of the proposed digital PLL and the theoretical analysis.

CHAPTER 4

Behavioral analysis and simulation of the proposed DPLL

4.1 Introduction

A novel low-complexity digital PLL architecture was described in the previous chapter. It emphasized the operation in the locked condition when the phase error is small compared with the time resolution of a T2D converter. To better understand the behavior of the proposed DPLL, the mathematical analysis and system-level behavioral simulations of the proposed digital PLL are done and presented in this chapter.

First, the phase noise performance of the proposed DPLL is analyzed mathematically, including the loop bandwidth dependence on the noise amplitude, the phase noise created by the proportional path quantization, the DCO phase noise attenuation behavior, the effective loop bandwidth, and the large signal loop behavior as well as the loop stability, followed by the determination of the loop design parameters.

Second, the proposed PLL is modeled using Simulink to confirm the basic operation and to observe the loop acquisition behavior. After that, an event-driven simulation technique is implemented using Matlab to speed up the simulation and to improve the simulation accuracy. With this technique, the phase noise of the DCO and the reference source are modeled in time domain and the phase noise performance of the proposed PLL is simulated and compared with the theoretical analysis to confirm the feasibility of the theoretical analysis.

4.2 Mathematical analysis of the loop behavior

Because of the non-linear operation, the performance of the proposed digital PLL is significantly different from that of other linear digital or analog PLLs. The phase noise behavior and the large signal behavior of the proposed digital PLL are analyzed mathematically in this section.

4.2.1 The loop bandwidth of the proposed DPLL

Figure 4.1 shows the mathematical model of the proposed digital PLL. To simplify

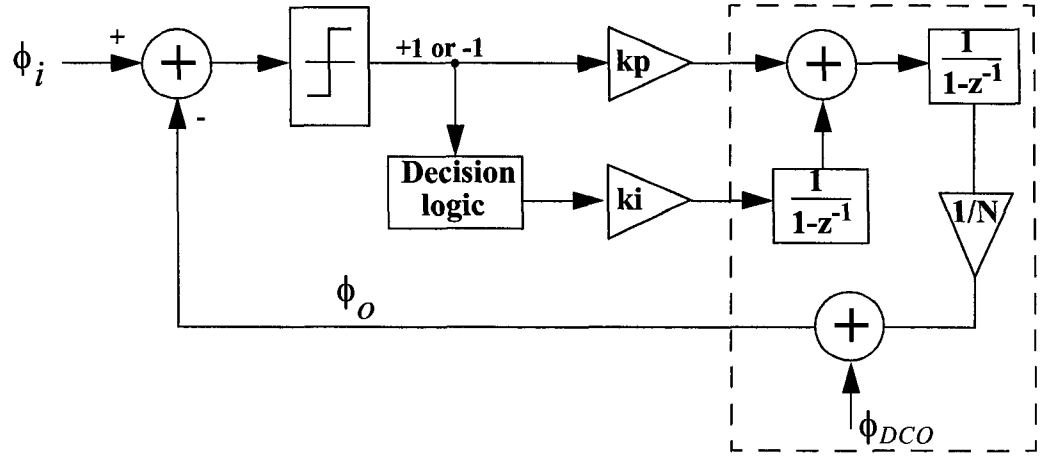


Figure 4.1 Mathematical model of the proposed digital PLL

the analysis, the DCO and the frequency divider are modeled together as a new DCO with a lower gain equal to K_{DCO}/N, as shown in the dashed box in the figure, and the DCO phase (ϕ_o) is defined to be the phase of the frequency divider output.

In the locked condition, depending on the noise power and the loop parameters, the model of the proposed digital PLL can be simplified in different ways. If the decision period of the frequency decision circuit is large enough, the phase error polarity always changes within each decision period and the output of the frequency decision circuit is zero, so the frequency tuning path does not affect the phase noise performance of the PLL. Otherwise, the digital PLL needs to be modeled as a second order loop with both the inte-

gral path (frequency tuning path) and the proportional path (the phase tuning path). In this section, the polarity of the phase error is assumed to always change within each updating period for simplification, so the output of the integral path is constant and the PLL is solely governed by the proportional path. This assumption is usually true after the frequency and phase acquisition is achieved when the loop is attenuating the DCO phase noise and tracking the reference phase noise. More details about this assumption can be found in the following subsections.

With the proportional path phase adjustment, the DCO phase can be increased or decreased by the amount determined by the gain of the proportional path. If the polarity of the phase error is constant, the DCO phase is always increased or decreased, resulting in the maximum phase tracking speed of P_T as given in Equation (4 . 1).

$$P_T = 2\pi\Delta f_p \qquad \text{(rad/sec)} \qquad (4.1)$$

where Δf_p denotes the gain of the proportional path with respect to the new DCO (the combination of DCO and the frequency divider), and the actual proportional path gain is $\Delta f_p N$. By assuming the reference signal is noise free, the digital PLL can completely attenuate the DCO phase noise if its maximum speed of the phase error variation is no more than P_T, or its maximum instantaneous frequency variation is no more than Δf_p. For simplification, the DCO phase noise is assumed to be a single tone sinusoidal wave as,

$$\phi_{DCO}(t) = \phi_p \cos(\omega_\phi t) \qquad (4.2)$$

where ϕ_p is the amplitude of the phase variation and ω_ϕ is the frequency of the phase variation. The maximum speed of the phase variation happens at the zero-crossing and its speed is calculated to be $\phi_p\omega_\phi$ by differentiating Equation (4 . 2). Considering Equation (4 . 1), for a certain phase amplitude (ϕ_p), the maximum frequency ($\omega_{\phi,\,max}$)

that the DPLL can completely attenuate can be derived as,

$$\phi_p \omega_{\phi,\,max} = 2\pi\Delta f_p \Rightarrow \omega_{\phi,\,max} = 2\pi\frac{\Delta f_p}{\phi_p} \quad \text{rad/sec} \qquad (4.3)$$

Any phase variation whose frequency is no more than $\omega_{\phi,\,max}$ is detected by the phase detector, and a negative phase variation with the same amplitude is generated by the proportional path (PP phase) to fully cancel the DCO phase variation as illustrated in Figure 4.2.

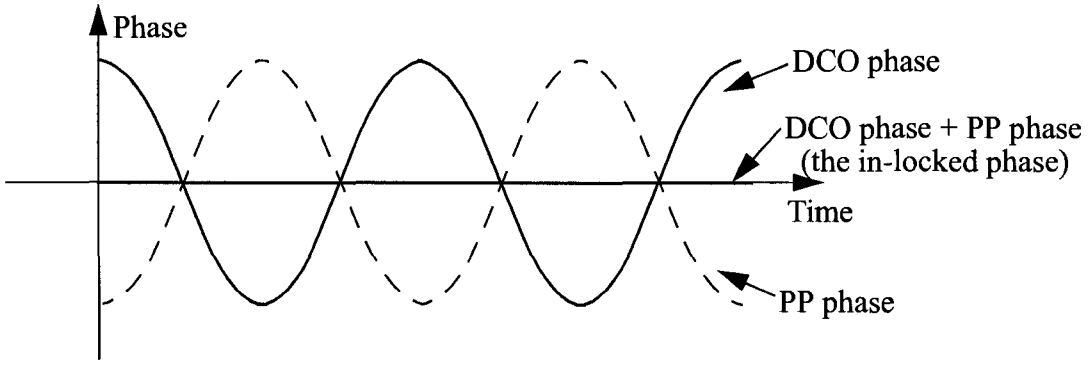


Figure 4.2 Full phase attenuation of the proposed digital PLL

The in-locked phase, the sum of the DCO phase and the phase created by the proportional path, is kept zero by increasing or decreasing the DCO frequency by f_p based on the output of the phase detector, which is a train of $\pm 1s$ whose average value [-1,+1] corresponds to the average proportional path frequency from $-\Delta f_p$ to $+\Delta f_p$. Please note that the in-locked phase is not absolutely zero, but a small value fluctuating around zero, which is analyzed in the section "Phase noise created by the proportional path quantization." on page 55.

Any DCO phase variation with the frequency larger than the maximum frequency obtained from Equation (4 . 3) can only be attenuated partially by the proportional path as illustrated in Figure 4.3. One period of the phase attenuation, which is shown as the curve *abcde,* is explained as follows.

During the period of *ab*, the proportional path attenuates the phase change effi-

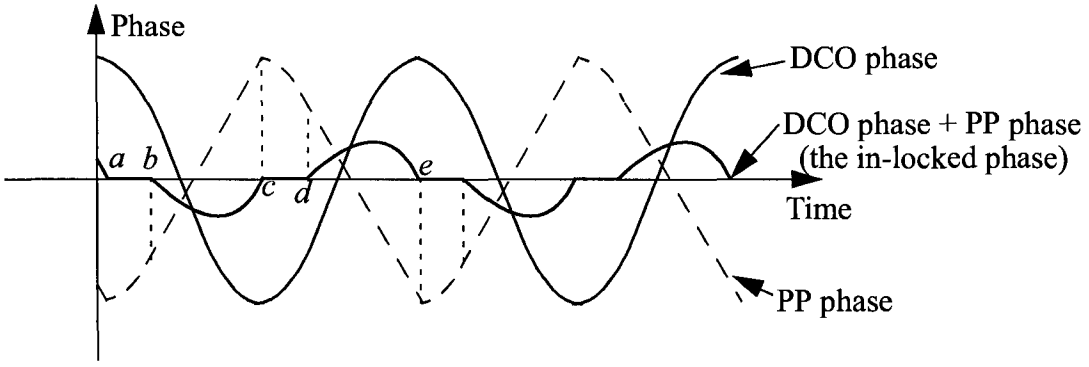


Figure 4.3 partial phase attenuation of the proposed digital PLL

ciently because the phase variation speed is below the maximum phase tracking speed (given by Equation (4 . 1)). After point *b*, the phase change speed is larger than the maximum phase tracking speed and the PP phase can no longer fully attenuate the DCO phase until the point *c*, resulting in non-zero phase output during the time period of *bc*. The similar behavior appears in the region of *cde*.

With the increase of the phase variation frequency, the percentage of the portion of *bc* or *de* increases and the percentage of the portion of *ab* or *cd* decreases until the proportional of *ab* or *cd* disappears as shown in Figure 4.4.

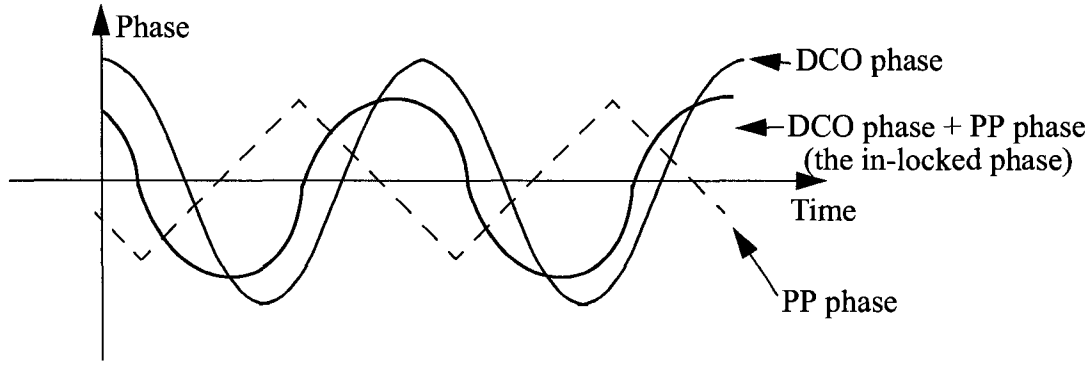


Figure 4.4 False phase attenuation of the proposed digital PLL

To provide an intuitive understanding of the jitter/phase compensation capability of the proposed digital PLL, the concept of the corner frequency or the loop bandwidth, which is widely used in a linear PLL system, is employed here. However, because of the non-linear operation of the proposed digital PLL, the corner frequency or the corner fre-

quency is no longer a single frequency, but dependent on a certain phase noise amplitude. Consequently, we define the corner frequency to be, for a given phase noise amplitude, the frequency where the DCO phase variation is attenuated by half in power (3dB). It is clear that the corner frequency is slightly larger than the maximum frequency given in Equation (4 . 3).

Based on the first-order assumption (with a 20dB/dec. roll-off slope), the output power decreases by 3dB when the frequency is increased by a factor of $\sqrt{2}$. Consequently, the corner frequency is defined as the $\omega_{\phi,\,max}$ multiplied by $\sqrt{2}$,

$$\omega_{\phi,\,corner} = 2\sqrt{2}\pi\frac{\Delta f_p}{\phi_p} \qquad (4.4)$$

Obviously the loop bandwidth (or the corner frequency) is inversely proportional to the amplitude of the phase variation. Please note that the analysis above is based on the assumption that the output of the integral path is kept constant. Based on the principle of the decision circuit, such an assumption is true as long as the noise frequency is larger than half of the inverse of the updating period.

The analysis above is based on the single-tone noise assumption. For linear systems, single-tone analysis can be easily extended to the real noise environment using the superposition principle. However, for the proposed digital PLL, although the single-tone analysis helps us understand the loop behavior, the frequency domain transfer function is not accurate to characterize the noise behavior in a real noise environment due to the non-linear operation and the principle of superposition does not hold in this case. In this work, a time-domain analysis is used to determine some loop parameters as described later in this chapter.

4.2.2 Phase noise created by the proportional path quantization.

Based on the analysis in previous section, the phase noise of the DCO is mainly

attenuated by the proportional path of the digital PLL. Even if the phase noise is well below the loop bandwidth and the proportional path is able to effectively track and attenuate the phase noise, the output phase of the PLL (ϕ_{PLL}) is not absolutely zero in the ideal locked condition because of the proportional path quantization (the UP/DN sensor in the digital PFD). In this section, both the reference phase noise and the DCO phase noise are assumed to be zero and the output phase noise of the PLL, solely caused by the non-linear effect of proportional path quantization is analyzed.

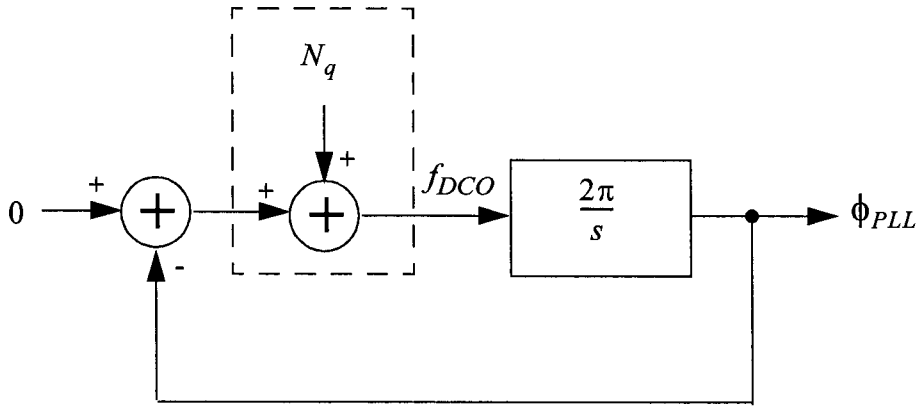


Figure 4.5 The noise model for analysis of the proportional path quantization

Figure 4.5 shows the simplified mathematical model of the digital PLL with only the proportional path, in which N_q denotes the quantization noise of the quantizer (PD) multiplied by the gain the phase tuning path, so N_q can be any value between $\pm 2\Delta f_p$. The DCO is modeled as $\frac{2\pi}{s}$. This simplified model can actually be treated as a first-order sigma-delta modulator and the frequency noise (f_{DCO}) can be calculated as the white quantization noise shaped by a high-pass filter with the transfer function of

$$H_n(z) = 1 - z^{-1} \qquad (4.5)$$

With the quantization step size of $2\Delta f_p$, the quantization noise without the noise shaping has a total noise power of

$$E_{total} = (2\Delta f_p)^2/12 \qquad (4.6)$$

Because the sampling rate in this system is the same as the reference frequency (f_{ref}), the quantization noise above, assumed to be white, is evenly distributed over the frequency range of $\pm f_{ref}/2$, so the power spectrum density of the quantization noise without noise shaping is,

$$E_q(f) = \frac{E_{total}}{f_{ref}} = \frac{(2\Delta f_p)^2}{12 f_{ref}} \quad \text{(/Hz)} \qquad (4.7)$$

The noise spectrum density of f_{DCO} can be calculated as the above noise power spectrum shaped by a high-pass filter with the transfer function of Equation (4 . 5),

$$E_f(f) = E_q(f)\left|1 - e^{j2\pi f/f_{ref}}\right|^2 \quad \text{(/Hz)} \qquad (4.8)$$

The noise power spectrum density of the DCO phase (ϕ_{PLL}), $E_\phi(f)$ can be expressed as,

$$E_\phi(f) = E_f(f) \cdot \left|\frac{2\pi}{2\pi f}\right|^2 \cdot \left|\mathrm{sinc}\left(\frac{f}{fref}\right)\right|^2 \quad \text{(/Hz)} \qquad (4.9)$$

where *sinc*() function denotes the transfer function of the zero-order hold due to the fact that the DCO frequency is kept within each reference period. Consequently, the phase

noise resulting from the proportional path quantization (quantization phase noise) is,

$$PN(f) = 10\log[E_{\phi}(f)] \quad \text{(dBc/Hz)}$$

$$=10\log\left[\left(\frac{(2\Delta f_p)^2}{12f_{ref}}\right)\cdot\left|1-e^{j2\pi f}\right|^2\cdot\left|\mathrm{sinc}\left(\frac{f}{fref}\right)\right|^2\cdot\left|\frac{2\pi}{2\pi f}\right|^2\right]$$

$$=10\log\left[\left(\frac{\Delta f_p^{\,2}}{3f_{ref}}\right)\cdot\left|\frac{1-e^{j2\pi f/f_{ref}}}{f}\cdot\mathrm{sinc}\left(\frac{f}{fref}\right)\right|^2\right] \quad \text{(dBc/Hz)} \qquad (4.10)$$

The quantization phase noise above reduces with the increase of the frequency offset for offset frequency no more than the reference frequency. For the replicas above the reference frequency, the noise energy drops dramatically because of the integration of the DCO and the zero-order hold effect. As an example, assume the reference frequency is 100MHz, and Δf_p is 20kHz, the quantization phase noise is plotted in Figure 4.6.

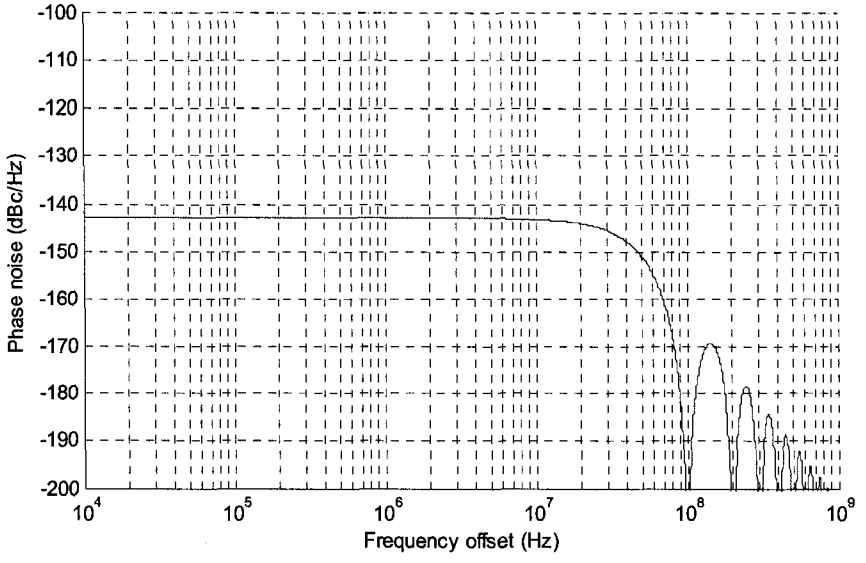


Figure 4.6 Phase noise caused by the proportional path activities

Obviously, such quantization phase noise should not dominate the phase noise performance of the PLL so that it doesn't significantly degrade the overall phase noise performance. Because the two main phase noise contributors of a PLL are the reference phase

noise and the DCO phase noise, they are compared with the quantization phase noise separately.

To compare the quantization phase noise with the phase noise contributed by the reference source, the low-frequency quantization phase noise (PN_1) is calculated based on Equation (4 . 10) by assuming that the frequency offset is close to zero,

$$PN_1 \approx 10\log\left(\frac{4\pi^2\Delta f_p^2}{3f_{ref}^3}\right) \quad \text{(dBc/Hz)} \qquad (4.11)$$

with the PLL multiplication ratio of N, the input referred phase noise of the PN_1 can expressed as,

$$PN_{input-refered} \approx 10\log\left(\frac{4\pi^2\Delta f_p^2}{3f_{ref}^3N^2}\right) \quad \text{(dBc/Hz)} \qquad (4.12)$$

Equation (4 . 11) shows that the low-frequency quantization phase noise increases (30dB/dec.) with the decrease with the reference frequency. On the other hand, with the same DCO frequency, lower reference frequency means higher multiplication ratio of the PLL, resulting lower input referred quantization phase noise of the PN_1 (20dB/dec.) for the same reference phase noise. As a result, as shown in Equation (4 . 12), the input-referred quantization phase noise PN_1 increases with a slope of 10dB/dec. when the reference frequency is reduced for a constant the PLL output frequency (f_{ref}*N).

As an example, for a reference frequency of 20MHz, and Δf_p of 20kHz, Equation (4 . 11) results in a quantization phase noise of -122dBc/Hz at low frequency offset. If the DCO frequency is 3GHz, meaning the frequency multiplication ratio is 150 (43.5dB), the input referred quantization phase noise of PN_1 is 165dBc/Hz (-122-43.5). As long as the reference phase noise is much higher than -165dBc/Hz, the quantization phase noise doesn't dominate the PLL noise performance. With the increase of the reference frequency, such input referred quantization phase noise is decreased, resulting in less noise

contribution to the overall phase noise performance.

To investigate the relation between the DCO phase noise and the quantization phase noise, the DCO phase noise is assumed to have phase noise profile with a slope of -20dB/dec. Because the tangent point of the quantization phase noise shown in Equation (4 . 10) and a straight line with a slope of -20dB/dec. is approximately located at $f_{ref}/4$. We consider the phase noise at the frequency offset of $f_{ref}/4$ as the worst case.

From Equation (4 . 10), the quantization phase noise at the frequency offset of $f_{ref}/4$ can be calculated as,

$$PN\left(\frac{f_{ref}}{4}\right) = 10\log\left[8.65 \cdot \left(\frac{\Delta f_p^{\,2}}{f_{ref}^{\,3}}\right)\right] \quad \text{(dBc/Hz)} \qquad (4.13)$$

The quantization phase noise is larger for a larger Δf_p, and a larger Δf_p means higher DCO phase noise (refer to "The proportional path frequency step size" on page 69). As a result, the relative quantization phase noise (related to the DCO phase noise profile keeps unchanged. To confirm this statement, the Δf_p is assumed to be (according to "The proportional path frequency step size" on page 69),

$$\Delta f_p = \sqrt{2}\Delta f \cdot \sqrt{f_{ref}} \cdot 10^{P_{-20dB}/20} \qquad (4.14)$$

Substitute Equation (4 . 14) into Equation (4 . 13) and assume the offset frequency Δf to be $f_{ref}/4$, relative phase noise can be obtained as,

$$PN\left(\frac{f_{ref}}{4}\right) \approx P_{-20dB}\left(\frac{f_{ref}}{4}\right) + 1 \quad \text{(dBc/Hz)} \qquad (4.15)$$

This shows that the quantization phase noise may be slightly higher than the DCO phase noise at certain frequency offset as illustrated in Figure 4.7. However, in the proposed digital PLL, such quantization phase noise does not exist independently, and it is actually reshaped by the DCO phase noise. This noise re-shaping tends to change the shape of the quantization phase noise profile toward the DCO phase noise profile, resulting in a reduc-

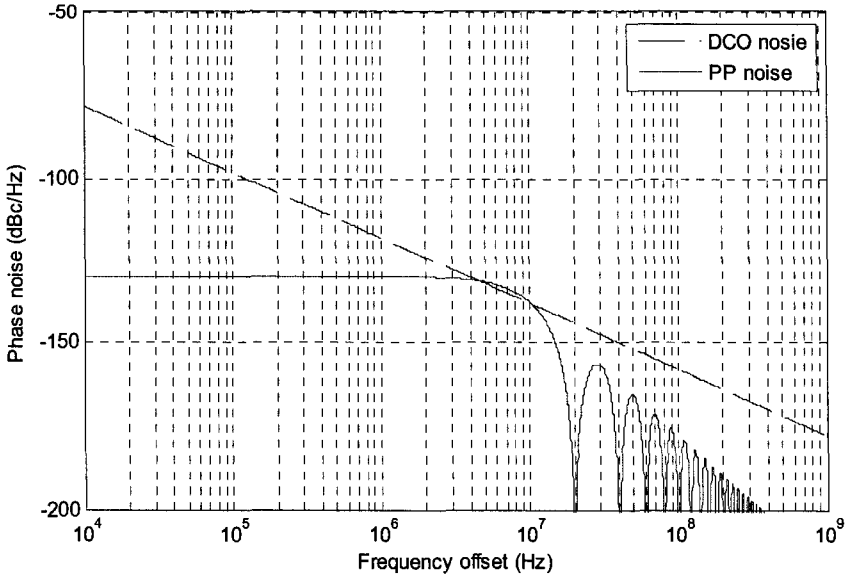


Figure 4.7 The DCO phase noise and the quantization phase noise

tion of the energy around the frequency offset of $f_{ref}/4$. Consequently, the quantization phase noise is far below the DCO phase noise for any frequency offset, and does not degrade the overall phase noise performance.

In fact, even if there is no quantization phase noise re-shaping, the combination of the DCO phase noise and the quantization phase noise is still far below the original DCO phase noise, even if the frequency offset is around $f_{ref}/4$. This is due to the fact that the DCO phase noise and the quantization phase noise do not be added in power or added linearly, but they are subtracted from each other linearly as analyzed in the previous section. For instance, the linear subtraction of the DCO phase noise from a quantization phase noise, which is 1dB higher than the DCO phase noise, results in a phase noise of 18dB below the DCO phase noise. Actually, because of such linear subtraction, the quantization phase noise has to be 6dB above the DCO phase noise before it can introduce extra phase noise, which does not happen in the proposed digital PLL.

4.2.3 DCO phase noise attenuation

An oscillator normally exhibits a phase noise profile as shown in Figure 4.8, which

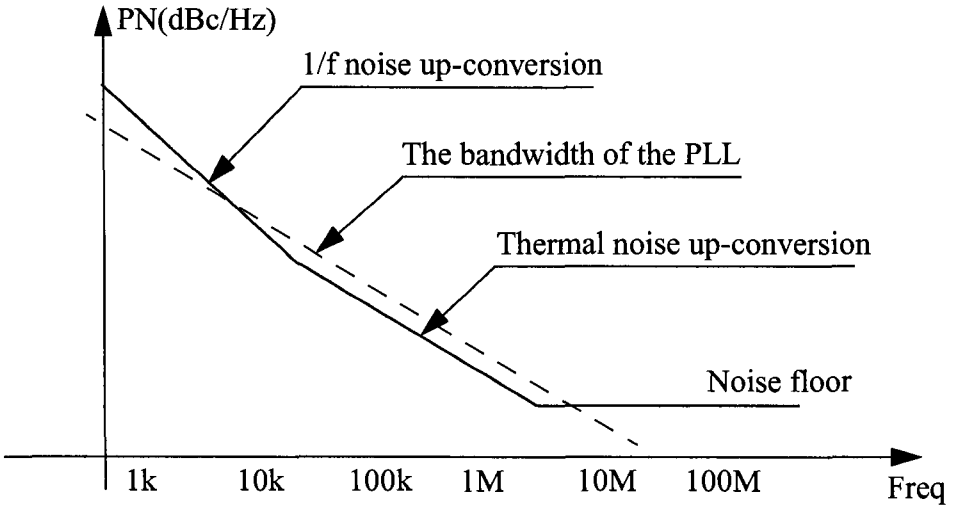


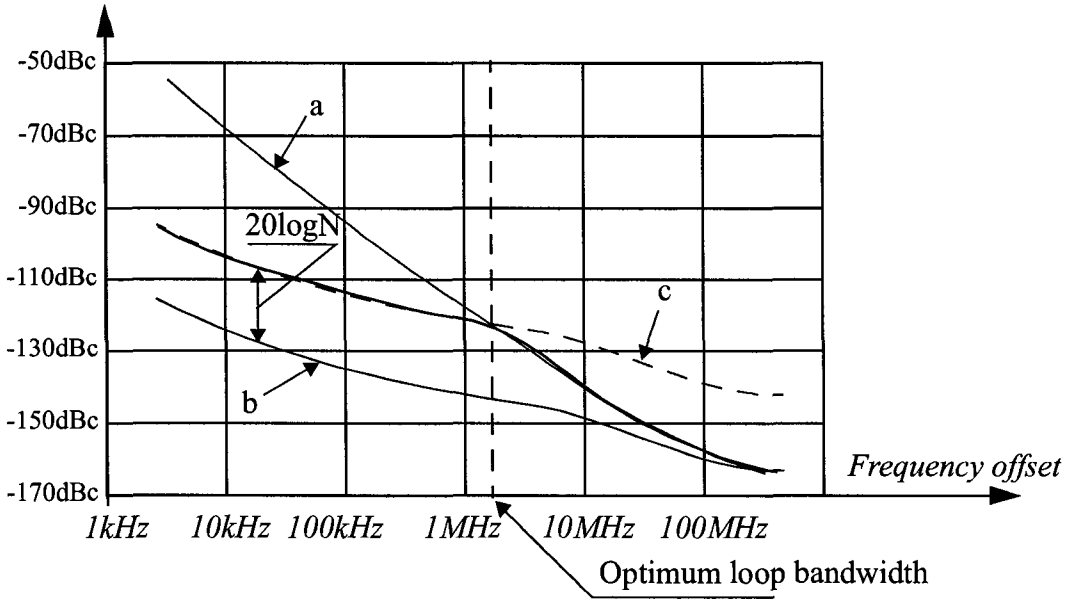
Figure 4.8 The phase noise profile of an oscillator

consists of three regions, the *1/f* noise up-conversion region with a slope of -30dB/dec., thermal noise up-conversion region with a slope of -20dB/dec., and the noise floor with a flat spectrum. Assuming no noise is added by the reference signal of the PLL, the gain of the proportional path (Δf_p) can be set to be a certain value so that the bandwidth (inversely proportional to the noise amplitude) of the DPLL is set slightly above the thermal noise up-conversion region as illustrated in Figure 4.8. In this case all DCO phase noise components below the dashed line can be attenuated by the DPLL with only the proportional path. A small portion of the noise at the low frequencies can be attenuated by the combination of the proportional path and the integral path.

4.2.4 Optimum loop bandwidth of the digital PLL

Two main noise sources of the PLL are the reference phase noise and the DCO phase noise. In a conventional PLL, the loop bandwidth, which is mainly determined by the loop filter, plays a key role on the noise performance of the PLL-based frequency synthesizer. The low-frequency, in-band, noise components generated in the VCO or DCO are attenuated by the PLL, and any such noise appearing in the PLL output is mainly due to

the reference phase noise. The output noise components whose frequencies lie outside the loop bandwidth originate mainly from the VCO or the DCO. In such PLLs, the loop filter is normally parameterized so that the PLL has an optimum loop bandwidth for the best noise performance. Figure 4.9 shows how to choose the synthesizer loop bandwidth so



a. Phase noise of VCO or DCO
b. Phase noise of the reference signal
c. Phase noise of the PLL

Figure 4.9 The optimum loop bandwidth of a conventional PLL

that the output phase noise can be minimum.

The phase noise of the reference signal and its contribution to the output phase noise (curve (c), which is 20log$N$ higher than that of the reference phase noise due to the frequency multiplication ratio of N) are also plotted in the figure, and the optimum loop bandwidth is the frequency at the interception point of curve (a) and curve (c).

In a digital PLL, the DCO, having a consistent phase noise performance, is normally integrated with other loop components, while different kinds of reference sources might be connected externally to the PLL input. The loop parameters usually need to be adjusted to adapt to different reference sources and to obtain the optimum noise perfor-

mance. Even if a fixed reference source is used, the optimum loop bandwidth still varies with different multiplication ratios.

In the proposed low complexity digital PLL, the loop bandwidth is related to the noise amplitude by Equation (4 . 4), and the DCO phase noise can be attenuated if the loop bandwidth is set to slightly larger than the DCO phase noise, as illustrated in Figure 4.8 for the case where a noiseless reference source is used. In the cases where the reference source contributes to the phase noise, the proposed digital PLL is able to automatically adapt to different reference phase noise levels and to different multiplication ratios so that the PLL noise always operates in the optimum loop bandwidth. To better understand this point, the mathematical model of the proposed digital PLL is redrawn as shown in Figure 4.10, in which the input noise (called the reference phase noise for the

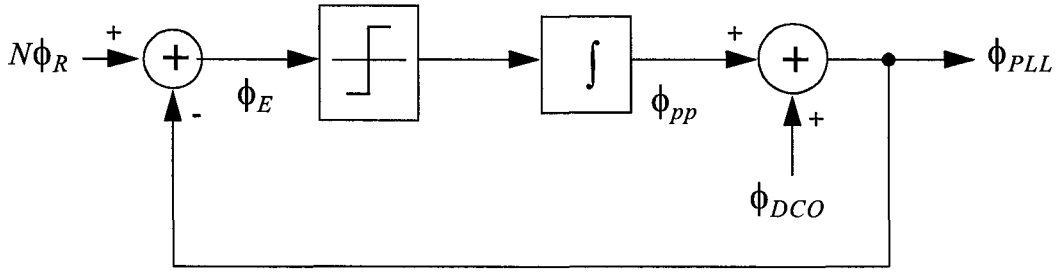


Figure 4.10 The noise model of the PLL with DCO/reference phase noise

remaining part of this section for simplification) is defined by actual reference phase noise (ϕ_R) times the multiplication ratio(N), while the frequency divider is eliminated from the model. We first assume that the connection between the integrator and the adder for the DCO phase noise is open. The phase error (ϕ_E) is the linear combination of the reference phase noise and the DCO phase noise as the bold curve in Figure 4.11. Because only the noise below the dashed line (the loop bandwidth, which is inversely proportional to the noise amplitude as shown in Equation (4 . 4)) may pass the nonlinear proportional path (the combination of the quantizer and the integrator in Figure 4.10) to appear at the input of the DCO (ϕ_{pp}).

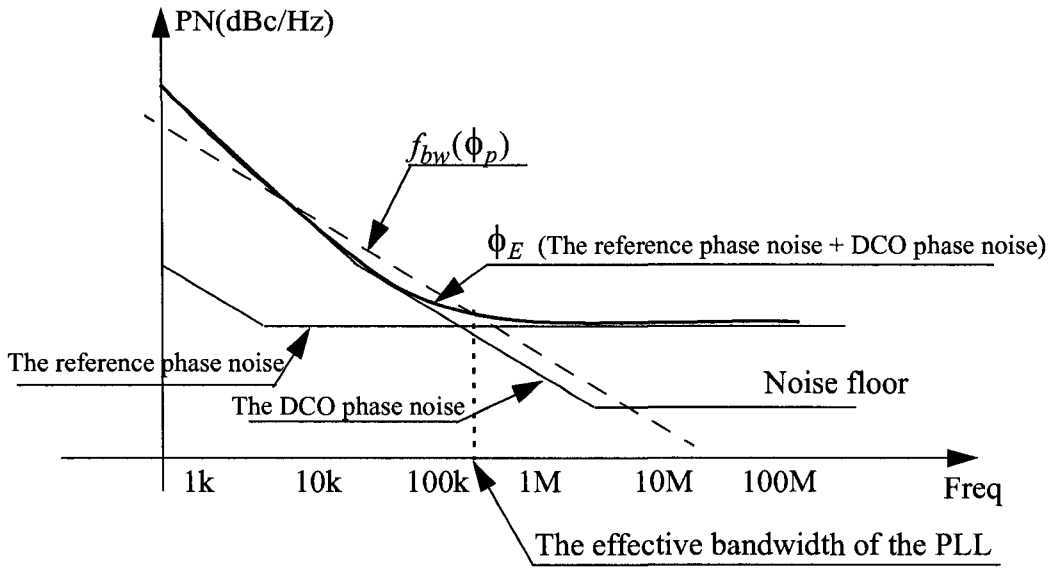


Figure 4.11 The linear combination of the DCO and reference phase noise

Here the frequency at the interception point of the dashed curve of $(f_{bw}(\phi_p))$ and the curve of ϕ_E, is defined as the effective loop bandwidth as shown in Figure 4.11. One can see that this effective bandwidth is always very close to the optimum loop bandwidth no matter what the reference phase noise profile is. Figure 4.12 shows some possible ref-

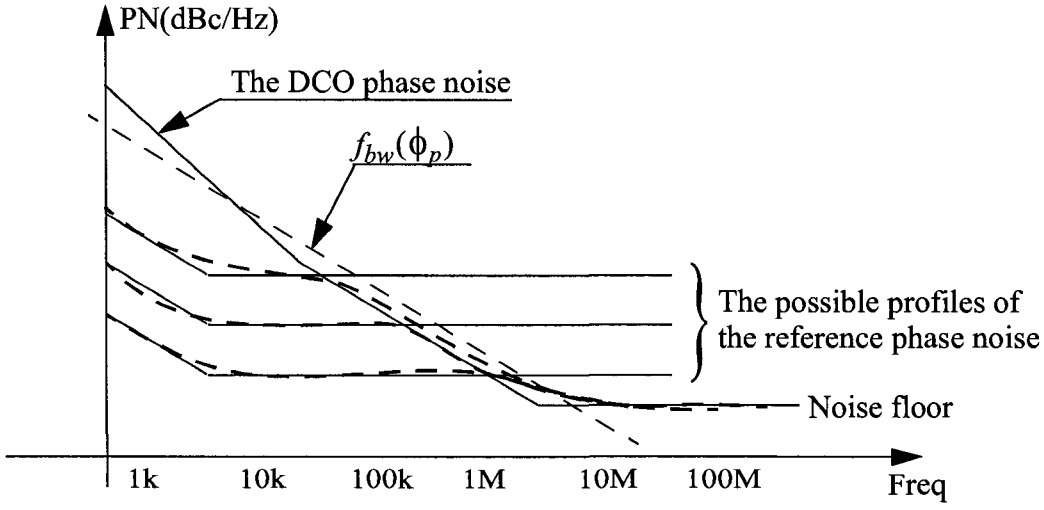


Figure 4.12 The PLL phase noise for different reference phase noise profiles

erence phase noise profiles and the PLL phase noise (the long bold dashed curves). The

effective loop bandwidth decreases with the increase of the reference phase noise and the PLL automatically adapts to different reference phase noise profiles once the loop parameters are set based on the DCO phase noise.

The loop bandwidth of the proposed DPLL ($f_{bw}(\phi_p)$) without the integral path has a slope of -20dB/dec., corresponding to the thermal noise up-conversion region. Any reference phase noise above the loop bandwidth (the dashed line in Figure 4.11) is filtered out by the PLL, while noise below that curve can pass the 'loop filter' and appear at the PLL output.

For a conventional PLL, the loop bandwidth is independent of the noise amplitude. Such PLLs track the reference phase noise within the loop bandwidth no matter how large the noise power or amplitude is, so the loop parameters are required to be re-adjusted to different noise levels to achieve the optimum noise performance.

4.2.5 The large signal loop response and the loop stability

Without the integral path, the proposed DPLL is actually a first-order PLL, which is unconditionally stable. However, when the integral path is included, the loop stability has to be considered. Based on the decision method of the proposed DPLL, the integral path takes effect only for the noise frequencies less than twice of the decision frequency (the inverse of the decision period). For low-frequency phase noise (frequency offset far below the decision frequency), the phase/frequency decision circuit together with its phase/frequency step sizes are approximately modeled as a linear second-order type-II loop filter, having an integral path and proportional path with the gains of K_p and K_i respectively. Using the same model as shown in Figure 4.1 on page 51, the open-loop

transfer function can be written as,

$$H_o(z) = \frac{K_{PFD} \cdot K_{DCO}}{(z-1) \cdot F_{ref} \cdot N_{div}} \cdot \left(K_p + \frac{K_i}{1 - z^{-1}} \right)$$

$$= \frac{K_{PFD} \cdot K_{DCO} \cdot K_p}{(z-1) \cdot F_{ref} \cdot N_{div}} \cdot \left(1 + \frac{K_i / K_p}{1 - z^{-1}} \right) \qquad (4.16)$$

Based on above transfer function, the natural frequency (ω_n) and the damping constant (ζ) of the loop can be derived as,

$$\omega_n = \sqrt{\frac{K_{PFD} \cdot K_{DCO} \cdot K_i}{F_{ref} \cdot N_{div}}} \qquad (4.17)$$

$$\zeta = \frac{1}{2} \cdot \sqrt{\frac{K_{PFD} \cdot K_{DCO} \cdot K_p}{F_{ref} \cdot N_{div} \cdot K_i / K_p}} \qquad (4.18)$$

Because the K_{PFD} is dependent on the noise amplitude, and it increases during the process of acquisition with the decrease of the phase error, both the natural frequency (ω_n) and the damping constant (ζ) of the loop increase while the loop is approaching the locked state. However, once the loop is close to be in lock, the high frequency noise, larger than one half of the decision frequency (f_{dec}), dominates and the output of the integral path keeps constant, resulting in an unconditionally stable first-order system.

The phase of the open-loop transfer function shown in Equation (4 . 16) is given in Equation (4 . 19) and plotted as in Figure 4.13.

$$P = \text{Angle}\left(\frac{1}{z-1} \cdot \left(1 + \frac{K_i / K_p}{1 - z^{-1}} \right) \right) \qquad (4.19)$$

If the unit gain frequency is smaller than $f_{dec}/2$, the difference between -180 degree and phase shift at the unit gain frequency is defined as the phase margin. However, the unit gain frequency is normally larger than $f_{dec}/2$, where K_i equals to zero (because the output of the frequency decision circuit is zero) and the phase shift is always -90 degree

(90 degree phase margin). Consequently, the actual phase of the open loop digital PLL is obtained as shown in Figure 4.14. From the figure, in the ideal case, the digital PLL is always stable because the phase margin is always positive. However, because of the latency of the digital PLL, especially in the paths of the DCO and the frequency divider, a certain phase margin needs to be maintained. To characterize the stability of the proposed DPLL, the phase margin is calculated based on the phase at the frequency of $f_{dec}/2$. As

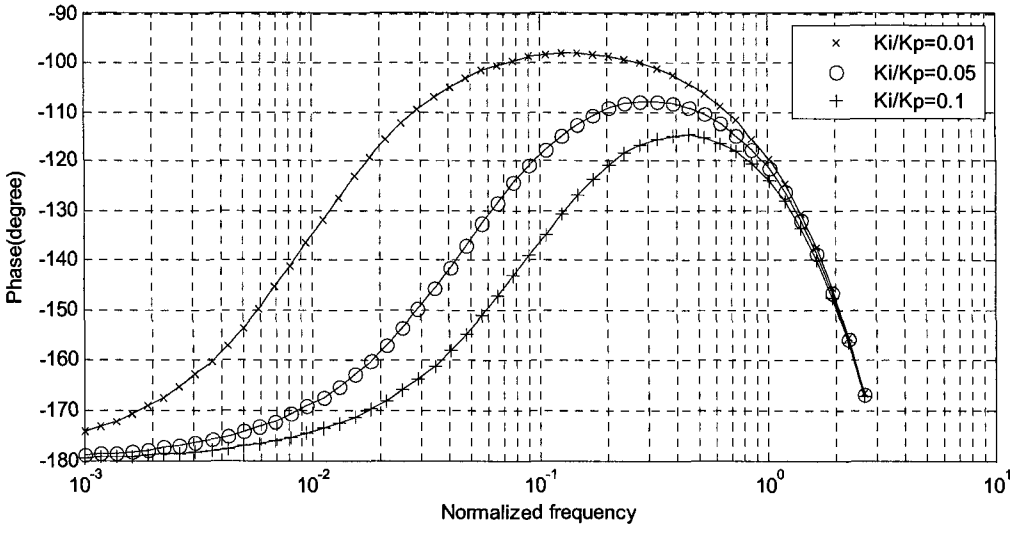


Figure 4.13 The phase of the open-loop PLL transfer function

shown in Figure 4.13, the phase margin is larger for a larger ratio of K_p and K_i. With the decision frequency of 0.2, the phase margin of more than 80 degree can be achieved with K_i/K_p of 0.01. In this case, it seems the phase margin is decreased with the decrease of decision frequency. However, if the actual integration and proportional frequency step sizes are kept constant, the lower decision frequency tends to reduce the gain of the integral path and to reduce the ratio of K_i/K_p, resulting in larger phase margin.

4.3 The determination of the design parameters

Once the DCO phase noise profile is determined, some loop parameters need to be selected to obtain the optimum phase noise performance as analyzed in the previous sections.

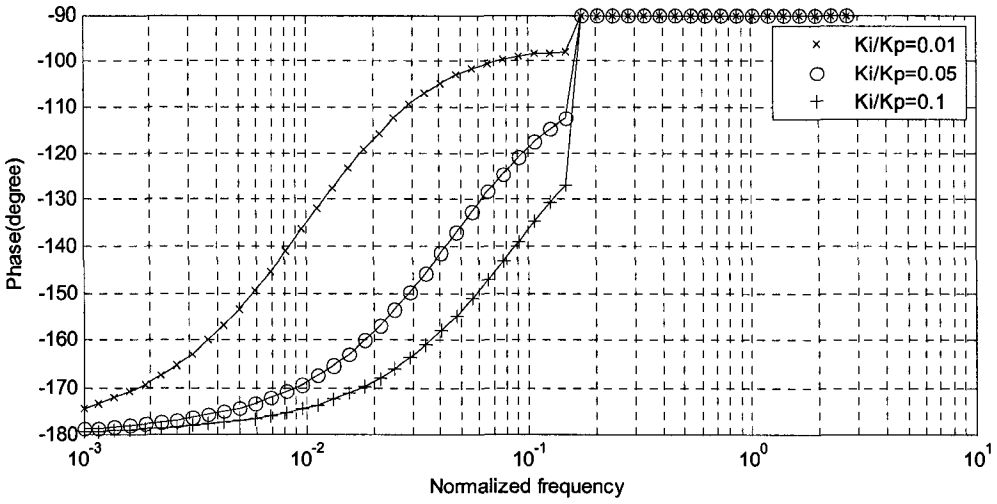


Figure 4.14 The phase of the open-loop digital PLL

4.3.1 The proportional path frequency step size

Based on the analysis above, the loop parameters are independent of the reference phase noise, and the frequency step of the proportional path (Δf_p) is determined by the thermal noise up-conversion region of the DCO phase noise with a slope of -20dB/decade. In this section, the DCO is assumed to have only the up-converted thermal noise of P_{-20dB} (dBc/Hz) at the frequency offset of Δf, and the minimum required Δf_p for the digital PLL to attenuate such DCO phase noise is determined based on the time-domain analysis, which is more accurate than the frequency domain analysis for non-linear systems.

Figure 4.15 shows the mathematical noise model of the DPLL for the time-domain analysis, in which the DCO represents the combination of the actual DCO and the frequency divider, so the DCO gain is $1/N$ of the original gain. The DCO edge jitter (J_E) is modeled as the integration of the period jitter (J_P), having a flat spectrum density. It is clear that when the DCO phase noise/jitter is attenuated by the feedback of the PLL,

$$J_E = -J_{pp} \qquad (4.20)$$

To make it possible for such attenuation, the proportional path frequency step

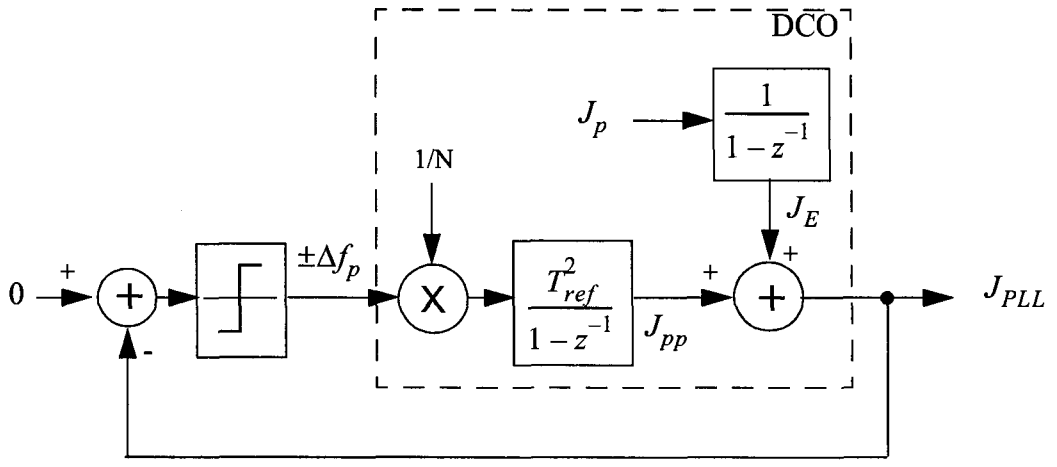


Figure 4.15 The noise model of the DPLL for time-domain analysis

(Δf_p) must be large enough so that the change speed (or differentiation) of the J_{pp} is larger than the variation speed (or differentiation) of J_E, which means $(\Delta f_p T_{ref}^2)/N$ (the differentiation of J_{pp}) should be larger than J_P(the differentiation of J_E). Since J_P is a random signal having a zero mean and a flat spectrum, its statistic peak value, which is $\sqrt{2}$ times its RMS value(σ_{JP}), is considered here and expressed as,

$$(\Delta f_p T_{ref}^2)/N > \sqrt{2}\sigma_{JP} \qquad (4.21)$$

For a DCO having an up-converted thermal noise of P_{-20dB} (dBc/Hz) at the frequency offset of Δf, the RMS value of its period jitter is[23],

$$\sigma_{\Delta t} = \frac{\Delta f}{f_{DCO}} \cdot \sqrt{T_{DCO}} \cdot 10^{P_{-20dB}/20} \qquad (4.22)$$

Considering the effect of the frequency divider (/N), the σ_{JP} can be calculated as,

$$\sigma_{JP} = \frac{\Delta f}{f_{ref}} \cdot \sqrt{T_{ref}} \cdot \frac{10^{P_{-20dB}/20}}{N} \qquad (4.23)$$

Consequently, the minimum required Δf can be obtained by substituting the above

equation into Equation (4 . 21),

$$(\Delta f_p T_{ref}^2)/N > \sqrt{2}\left(\frac{\Delta f}{f_{ref}} \cdot \sqrt{T_{ref}} \cdot \frac{10^{p-20dB/20}}{N}\right)$$

$$\Rightarrow \Delta f_p > \sqrt{2}\Delta f \cdot \sqrt{f_{ref}} \cdot 10^{p-20dB/20} \qquad (4.24)$$

As an example, the minimum required Δf_p for different phase noise and reference frequencies (for the digital PLL to be able to attenuate the DCO phase noise) are listed in Table 4.1.

TABLE 4.1: The minimum frequency step in the fractional path

DCO Phase noise (dBc/Hz)	Reference frequency	$\Delta f_{p,min}$
-118dBc/Hz at 1MHz offset	100MHz	18kHz
-100dBc/Hz at 1MHz offset	100MHz	141kHz
-95dBc/Hz at 1MHz offset	140MHz	298kHz

Because the increase of Δf_p allows more reference phase noise to go through the PLL and appear at the PLL output, Δf_p should normally be set to its minimum value for best noise performance especially for a noisy reference source.

4.3.2 The updating period and the integration step size

While the phase noise performance is mainly determined by the proportional path frequency step, the loop acquisition behavior is mainly determined by the parameters of the integral path, the updating period (T_{DEC}) and the integration step size (Δf_i). Such acquisition includes the frequency/phase acquisition together with the attenuation of some strong low-frequency phase noise originating from the DCO, which is beyond the capability of the proportional path.

First, the required updating period is related the largest frequency offset of DCO phase noise which is beyond the attenuation range of the proportional path. As illustrated

in Figure 4.16, the frequency at the interception point of the *1/f* DCO phase noise curve

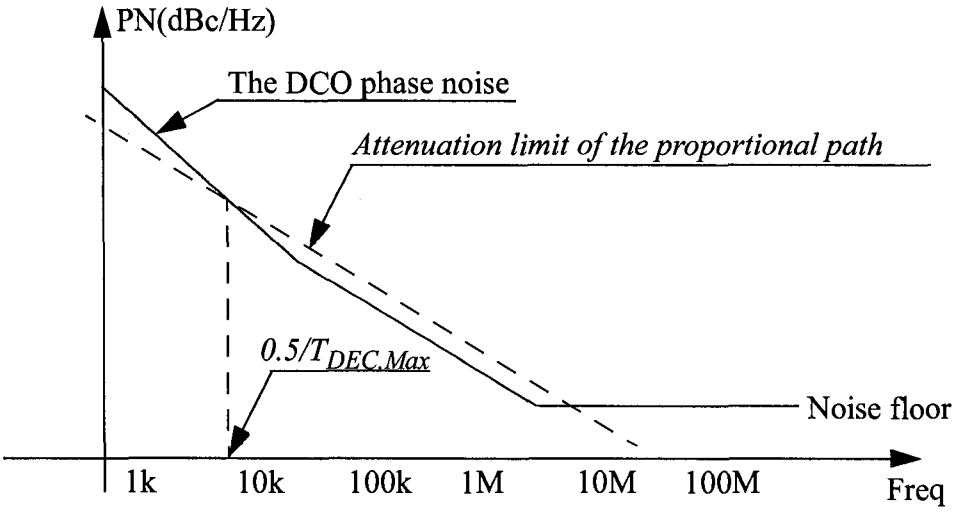


Figure 4.16 Determination of the decision period of the DPLL

and the attenuation limit of the proportional path determines the maximum decision period, by which the phase noise above the attenuation limit of the proportional path can be attenuated by the integral path. Normally the decision period can be set to one half (or even smaller) of the maximum value determined above. Because the flicker noise normally dominates the frequencies below 1MHz, so the decision period can be as large as 1/500kHz. If a decision period of 1/1MHz is needed for a reference frequency of 100MHz, a frequency divider, having division ratio of 100 can be used to obtain the required decision period.

Once the decision period is determined, the integration step size can be determined by two factors. First, the integration step size has to be much smaller than the proportional path step size to enable the proper operation of the proportional path. Second, the step size, together with the decision period and the proportional path step size determines the loop stability and the acquisition speed. Normally larger integration step size means faster loop acquisition and the loop is less stable. When the acquisition speed is not of concern, the integration step size can be set to the lowest possible step size of the DCO to improve

the operation of the proportional path. Normally, the integration step size could be approximately *1/5* of the proportional step size or less. For example, a proportional step size of 20kHz normally corresponds to the integration step size of approximately 4kHz.

4.4 Simulink modeling of the DPLL

To confirm its operation, the proposed DPLL was modeled using Simulink and the locking behavior was observed. All blocks are assumed to be noiseless in this modeling process. Although Simulink modeling simulation is not suitable for accurate noise/jitter performance analysis, it enables intuitive modeling of each components and helps confirm the basic operation of the proposed DPLL.

To simplify the modeling process, the band switch mechanism is eliminated from the proposed system and the digital PFD is a combination of a tri-state PFD and a one-bit T2D (the UP/DN sensor). The phase/frequency decision logic circuit produces phase/frequency up/down signals. The DCO has the inputs of phase/frequency up/down and generates a square wave output. A frequency divider is used to divide the oscillator output. A DCO with a linear frequency tuning characteristic is modeled without the band switch. The system parameters for the Simulink modeling simulation are given in Table 4.2.

TABLE 4.2: The Simulink simulation parameters of the DPLL

Components	Parameters	Values
DPFD	Edges to PEReady delay	100ps
Phase/frequency decision circuit	Decision period	8 reference cycles
DCO	proportion path step size	+/-200kHz
	Number of frequency levels	2048
	Frequency tuning step size	50kHz
	Center frequency	2GHz
Frequency divider	Frequency division ratio	20
DPLL	Reference frequency	100MHz

4.4.1 The model of the non-linear digital PFD

The non-linear digital PFD is modeled as a commonly used tri-state PFD, consist-

ing of two resetable D flip-flops, together with the reset circuit as shown in Figure 4.17.

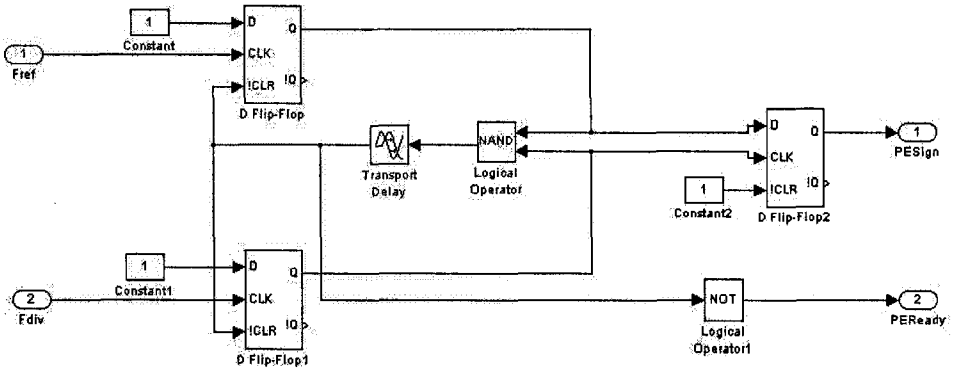


Figure 4.17 The Simulink model of the digital PFD

An ideal D flip-flop is used to determine the sign of the phase error for modeling purpose only. In the actual circuit implementation, such a simple D flip-flop is normally not sufficient for the sign detection due to the hysteresis of the D flip-flop[26]. In the DPLL, the hysteresis of the *UP/DN* sensor is linearly added to the accumulated edge jitter of the DCO in the closed-loop operation. Therefore, careful design must be done to minimize the hysteresis of the sign detector. As shown in Figure 4.17, the reset signal is used to generate the signal *PEReady.*

4.4.2 The model of the frequency divider

The frequency divider normally consists of D flip-flops and logical gates, but modeling the frequency divider with flip-flops significantly slows down the simulation and increases the possibility of simulation errors. A z-domain integrator, configured as a cycle counter, is used to model the frequency divider in this work as shown in Figure 4.18. The division ratio can be programmed by changing the parameter '*Ratio*' in the model. The model is triggered on the rising edge of the input clock and the integrator output is increased by one at each trigger event. If the integrator output equals to the value of (*Ratio*-1), the next trigger event resets the integrator to zero, and then it starts to increase again. The output of the integrator, a saw wave, is converted to a square wave by compar-

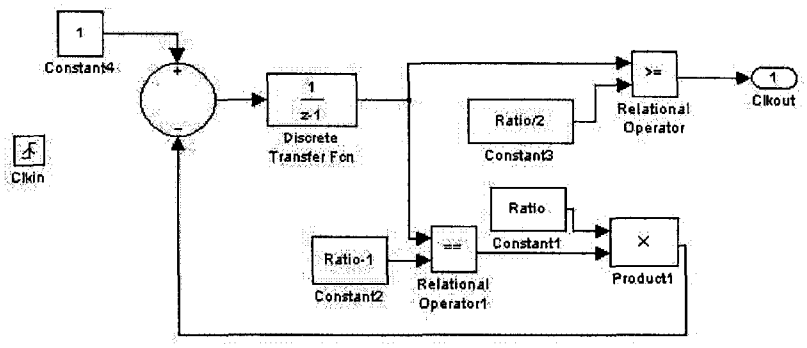


Figure 4.18 The Simulink model of the frequency divider

ing it with a constant (*Ratio*/2).

The model simulation consumes less CPU time and memory because only mathematical functions are used in the model above. Since all components are triggered on the same clock edges, the model itself does not contribute additional noise caused by the oversampling process of the simulation, thus more accurate simulation results are expected.

4.4.3 The model of the DCO

A DCO can be modeled in two steps, determining the frequency based on the inputs and generating a clock with the resulting frequency as shown in Figure 4.19. The frequency up/down signals control an *UP/DN* counter, representing the bi-directional shift register of the DCO, and a step signal is used to initialize the *UP/DN* counter to its middle value at the beginning of the simulation. The counting value and the phase up/down signals are then converted to the frequency. To generate a time-domain signal for a given frequency, a straight-forward method is used to integrate the frequency to phase and to convert the phase to a time-domain signal with a sinusoidal function, and if necessary, a comparator is used to convert the sinusoidal signal to a square wave signal so that it can be recognized by the following logic circuits. In this work, however, the conversion from the phase to the time-domain signal is achieved by a rounding function and a relational operation to reduce the load of the CPU. Besides the desired output (*Vout*), the model also outputs the frequency for monitoring purpose.

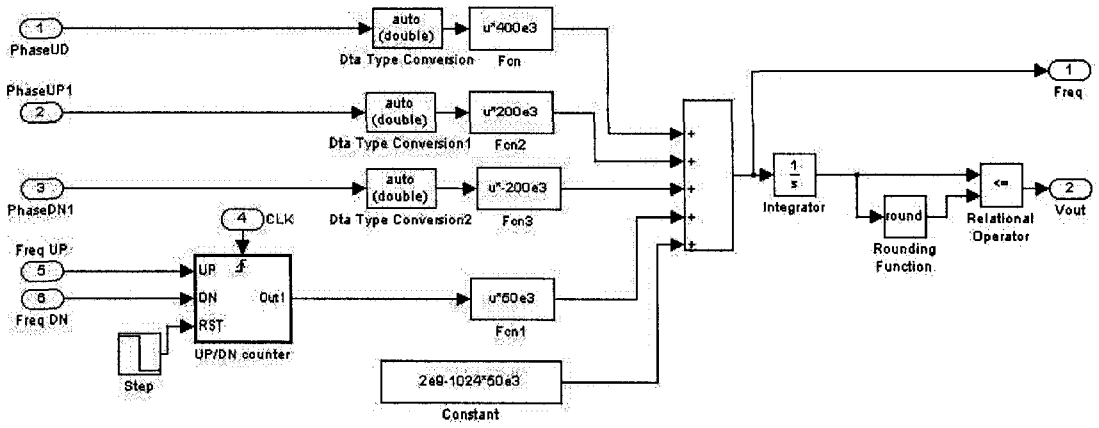


Figure 4.19 The Simulink model of the DCO

Figure 4.20 shows the model of the up/down counter with up/down limits and

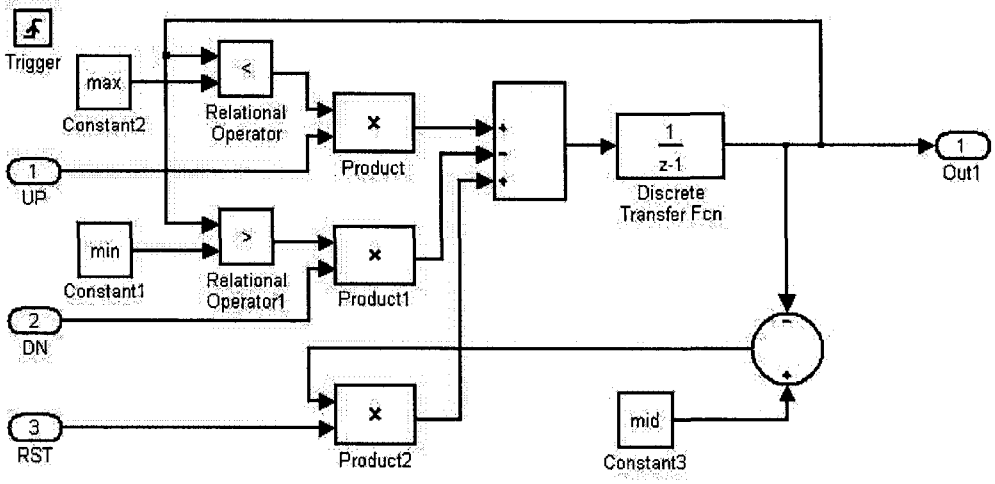


Figure 4.20 The Simulink model of the up/down counter

reset. It increases or decreases by one if *UP* or *DN* is high and the counted value does not reach the up/down limits defined by the two parameters, *max* and *min*. If *Reset* is high, the counter is reset to a value specified by the parameter '*mid*'.

4.4.4 The model of the decision circuit

The decision circuit consists of the phase decision circuit and the frequency decision circuit. The phase decision is normally made directly based on the phase error, and

the frequency decision is made based on a certain algorithm. A simple algorithm is modeled in this section. The frequency up/down decision logical circuit produces '*FreqUP*' if all previous *M PESigns* are ones or produces '*FreqDN*' if all previous *M PDSigns* are zeros at the end of each updating period. The model of the frequency decision circuit is shown in Figure 4.21. The signal *PESign* is shifted into the delay line (a shift register)

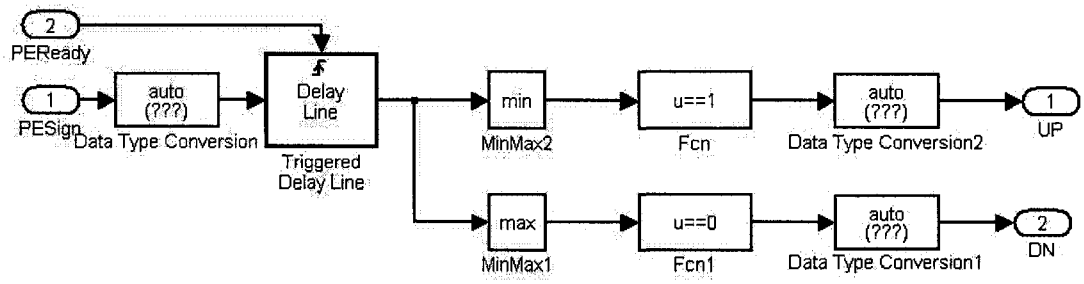


Figure 4.21 The Simulink model of the frequency decision circuit

whose length is the number of *PEReady* periods within one decision period. The output vector (having elements of zeros and ones) of the delay line contains previous *N PESigns*. If the minimum value of its elements is one, the previous *N PESigns* are all ones and *UP* is asserted. Similarly, if the maximum value of its elements is zero, the previous *N PESigns* are all zeros and *DN* is asserted.

The model of the whole phase/frequency decision circuit is a combination of the phase decision and the frequency decision circuit as shown in Figure 4.22. A decision

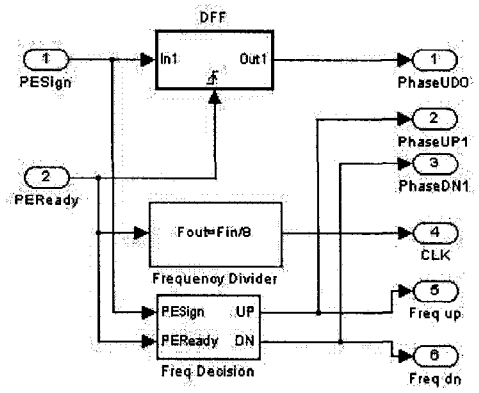


Figure 4.22 The Simulink model of the phase/frequency decision circuit

clock signal (CLK) is generated in the model and is to used as the clock of the shift registers in the frequency tuning path of the DCO.

4.4.5 The top-level Simulink model of the DPLL

With the models of the building blocks described above, the top-level model of the digital PLL is formed as shown in Figure 4.23. A Simulink component, scope, is used to

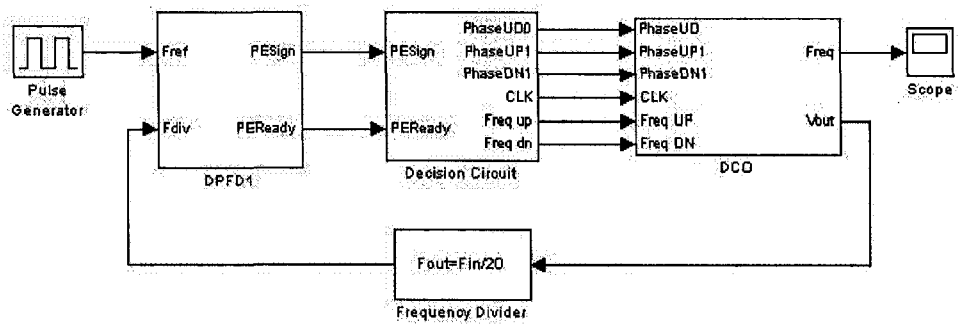


Figure 4.23 The Simulink model of the digital PLL

monitor the frequency change of the DCO.

4.4.6 Simulink simulation results

With the reference frequency of 100MHz and frequency divider ratio of 20, a simulation was done and the locking process was observed by plotting the frequency of the DCO. Figure 4.24 shows the simulation result. To better observe the frequency quantization, the result was linearly converted to an integer value by (*Freq*-2GHz)/50kHz+1024. The frequency 2GHz corresponds to the integer value of 1024. In this simulation, it takes less than 43μs for the loop to acquire lock. The frequency of DCO control word change increases with the decrease of the phase error during the process of the frequency and phase acquisition as expected in "The large signal loop response and the loop stability" on page 66.

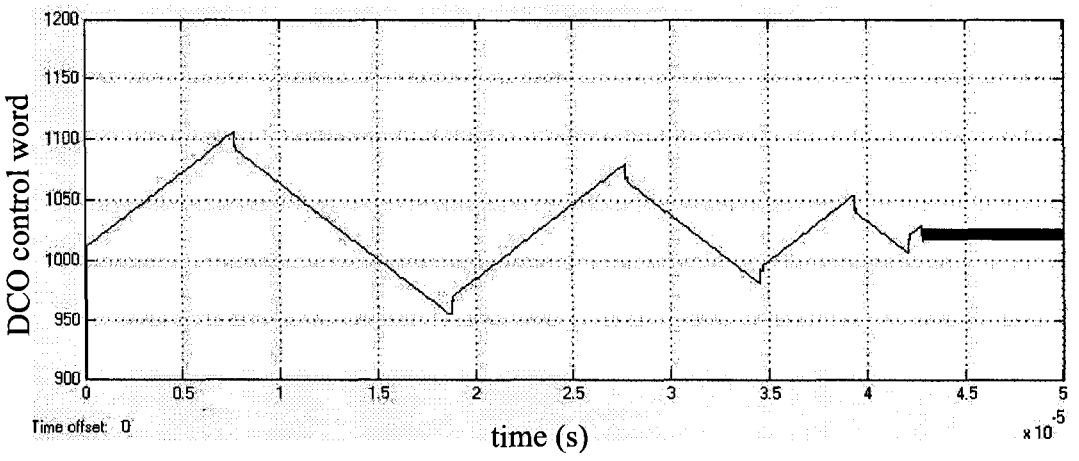


Figure 4.24 The Simulink simulation result of the digital PLL

4.5 Event-Driven simulation technique

To simulate the noise performance of the entire DPLL in time domain, the simulation has to be able to provide precise edge positions of the output clock. In this case, time-driven type (over-sampling type) simulators, such as Simulink, Hspice, and so on, are not efficient because the accuracy is limited by the simulation step size, and using a small step size, corresponding to the required accuracy, normally results in an excessively long simulation time. In fact, lots of time steps calculated by a time-driven simulator are not useful for the system level analysis, instead only some specific time points of interest, such as the transition time points of a clock signal, the sampling outputs right after each sampling operation, and so on are needed for loop analysis. In [24], the simulation and modeling of the phase noise in an RF oscillator in time-domain uses an event driven VHDL simulator. However, the VHDL simulator normally has a limited timing resolution and very limited number of mathematical data processing functions. Simulating a DPLL using VHDL simulator requires to create VHDL models for all blocks, and to create additional models to generate various testing noise sources.

In this research work, an event-driven modeling and simulation technique was employed to simulate and to analyze the behavior of the whole proposed digital PLL. It

calculates only the time points of interest, and provides precise edge position information, corresponding to the computer calculation accuracy. This modeling and simulation technique can be implemented in various computer languages. In this work, Matlab script was selected to implement this technique because it provides various data processing functions and simplifies modeling of each function block and post-processing of the simulation output data. For a 2GHz digital PLL as given in the following sections, the simulation speed is approximately 2μs per second of CPU time on a P4-2G PC. The simulation speed can be further significantly increased by using pre-compiled computer languages such as C language. Since this technique allows high-level modeling of the function blocks, it is especially helpful for the system level design and the system parameters optimization.

4.5.1 The concept of event-driven technique

Conventional oversampling simulation tools do a time-sweep with a certain time step, and calculate signals at each time point. To achieve a good time resolution, the time step has to be sufficiently small, which means the simulation needs to calculate a large number of time points. In an event-driven simulation, however, only time points of interest are calculated. The concept of event driven has been widely used for many years. An event-driven program basically consists of three parts, the event generators, the event dispatcher and the event handlers as shown in Figure 4.25. Event generators generate the

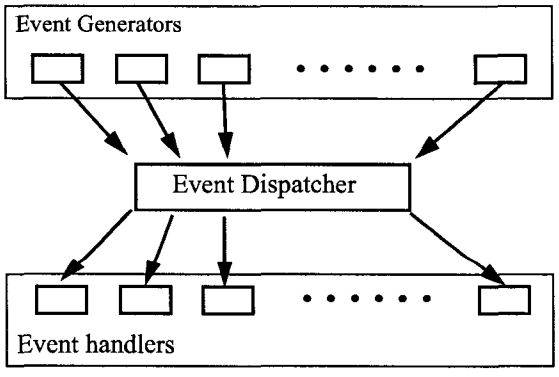


Figure 4.25 The concept of the event driven program

events. For example, in a computer system, an event can be generated by a mouse click, a key press, a timer and so on. The event dispatcher receives and stores all events and calls corresponding event handlers based on a certain order. This is normally achieved by maintaining an event queue inside the event dispatcher. The program always returns to the event dispatcher after a call to an event handler is completed. In some cases, the event handlers also generate some events, like an event generator, when they are executed and those events are processed in the same manner by the event dispatcher.

4.5.2 Event-drive simulation engine

The principle of the event-driven modeling is shown in Figure 4.26, in which each

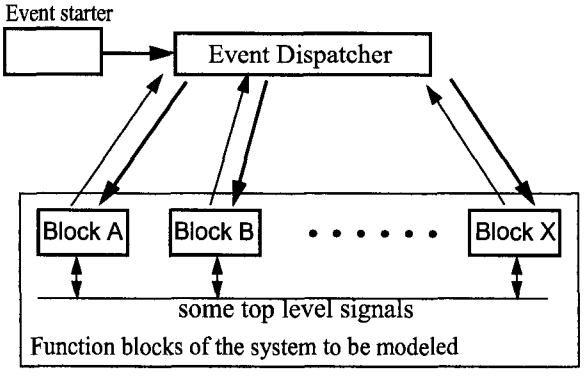


Figure 4.26 The principle of the event driven simulation

building block (Block A, B...) of the system to be simulated is represented as an event handler. The event starter generates some initialization events for each building block before the event dispatcher calls each function block according to the event queue, maintained inside the event dispatcher. Those event handlers are responsible for generating all subsequent events to be added to the event queue by the event dispatcher. Each event includes four fields: the event ID, the event handler, the event time and the event parameter. The later two are to be passed to the corresponding event handler so that the event handler can process the event accordingly. In addition, some top-level signals or variables are

shared among building blocks as shown in Figure 4.26.

4.5.3 Event dispatcher

As shown in Listing 4.1, the event dispatcher is basically a loop, which always

Listing 4.1. Matlab code for the event dispatcher.

```
%start the event-driven engine
h=waitbar(0,'Simulation is running, please wait...');
Proc=0; EventCount=0; t=cputime;
Result=event('RUN');
while (Result<=EndTime)
   EventCount=EventCount+1;
   if floor(Result/EndTime*50)>Proc
        Proc=Proc+1;
        waitbar(Result/EndTime);
   end
   Result=event('RUN');
end
close(h);
disp(sprintf('Simulation is done!\n  Total number of events: %i',EventCount));
disp(sprintf('  Total CPU time used: %f (s)',cputime-t));
```

picks up the earliest event in the event queue and calls the corresponding event handler. The sub-function "event" defined as "Result=event(Task, Func, Time, Para)" is responsible for managing the event queue, depending on the parameter of *Task*, it performs one of the following tasks:

1. Initialize (INIT) the event queue.

2. Insert (INS) an event to the event queue and return an event ID. This task is called whenever an event is generated by the event starter or an event handler. The event is inserted to the event queue and sorted by event time.

3. Remove (DEL) an event specified by the event ID from the event queue.

4. Run (RUN) the earliest event in the event queue and remove it from the event queue. Part of the Matlab code, responsible for this task, is given in Listing 4.2. It returns the event time so that the main program can determine the progress of the simulation. It is important that the earliest event has to be temporally stored and removed from the event

queue before the corresponding event handler is called to avoid possible errors because the event handler may call the function "event" recursively to add or remove events.

Listing 4.2. Matlab code to run&remove the earliest event.

```
CurrentEvent=Events(1);
Events=Events(2:EventLength);
EventLength=EventLength-1;
feval(CurrentEvent.FuncHandle, CurrentEvent.Time, CurrentEvent.Para);
Result=CurrentEvent.Time;
```

4.5.4 Event starter

In the event-driven simulation, almost all events are generated by those building blocks, except for some events at the beginning of the simulation. Those events, generated by the event starter, are to initialize all building blocks so that more subsequent events can be generated by those blocks. The event starter first requests the initialization of the event queue, and then generate initialization events for each building block. The Matlab code given in Listing 4.3 is the event starter for the model of the digital phase locked loop as shown in Figure 4.25. It generates the initialization events for REF, DCO, FDIV, DPFD

Listing 4.3. Matlab code of the event starter.

```
event('INIT');
%Reset all blocks by passing a parameter of '0' at time of 0 to those blocks
event('INS', @REF, 0, 0);
event('INS', @DCO, 0, 0);
event('INS', @FDIV, 0, 0);
event('INS', @DEC, 0, 0);
event('INS', @DLF, 0, 0);
```

and DLF.

4.5.5 Modeling of building blocks

The modeling of each building block is actually done by writing a Matlab function for each block. As given in Listing 4.4, this function processes different events specified by *para*. In the process of an event, some new events might be generated if necessary. The event '0' is assumed to be always the initialization event. Other events are pre-defined for each building block. The models of each building block of a DPLL are presented in the

Listing 4.4. Matlab code of a build block

```
function Block_Name(time,para)
persistent Var1 Var2 ...;
switch para
  case 0 %Initialize the block
    <Initialization>;
  case 1
    <Process of event type 1>;
  case 2
    <Process of event type 2>;
    <...>;
  case N
    <Process of event type N>;
end
```

following section.

4.6 Event-driven modeling of the proposed DPLL

4.6.1 The model of the reference clock source

The PFD compares the edges (the rising edge is considered here) of the reference clock and the edges of the divided clock, so only the time of each rising edge of the reference clock is of interest. The reference clock source is modeled as follows: In the initialization part of the model, the first rising edge event is generated, and during the process of each rising edge, the event of next rising edge is generated so that the clock signal can be continuously provided to the loop. Obviously, all rising edge events are sent to the PFD so that the PFD can compare them with the divided edges. Consequently, the model of the reference source is written as given in Listing 4.5, where the reference period is defined as

Listing 4.5. Matlab code of the model of the reference clock.

```
function REF(time,para)
persistent Tref;
switch para
  case 0 %initialization
    event('INS', @REF, 0 ,1);
    Tref=10e-9; %reference period
  case 1 %process of edges
    %Add next reference edge
    event('INS', @REF, time+Tref,1);
    %Send edges to DPFD
    event('INS', @DPFD, time, 1);
end
```

Tref, the time of the next rising edge is always calculated as the "*time+Tref*". Here, it is easy to add some additional timing jitter/noise to the reference clock source by replacing "*time+Tref*" with "*time+Tref+Tj*", where *Tj* is the period jitter to be added.

4.6.2 The model of the digital phase and frequency detector

In the DPLL, the phase and frequency detector compares the reference clock edges and the divided clock edges to get the phase error, which is quantized to form the output of the DPFD. The structure of the DPFD to be modeled, is shown in Figure 4.27. A conven-

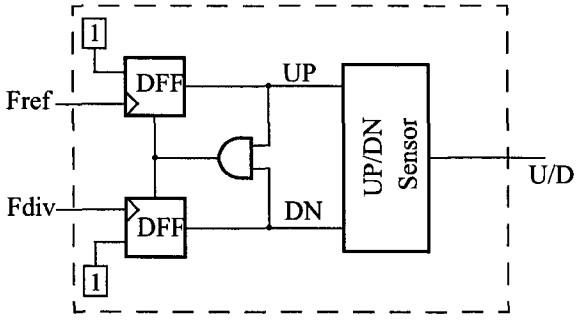


Figure 4.27 The digital phase and frequency detector

tional tri-state PFD is used to obtain the phase error (the width difference between *UP/DN*) and this error is converted to a digital signal by a time to digital converter (T2D) or a sign detector. The sign of the phase error is carried by a global variable *PFDSign*, which is accessible to the digital loop filter.

Besides the initialization, the DPFD processes three events: the rising edge of the reference clock, the rising edge of the divided clock from the frequency divider, and the internal reset event. Listing 4.6 shows the Matlab code for the PFD model. At the event of reference rising edge, if *UP* is zero, it changes *UP* to one and stores the current time to *RefTime*. It also checks if *DN* is already one to determined if it is necessary to generate a reset event. If *DN* is one, a reset event is generated with a delay of 50ps in the reset path. Similar process is done at the event of the divider edge. In the event of PFD reset (case 3), the time error is calculated and the sign of the phase/frequency error, is obtained and

Listing 4.6. Matlab code of the model of the PFD

```
function DPFD(time,para)
global PFDSign;
persistent UP DN RefTime DivTime;
switch para
 case 0  %Initialization
  UP=0; DN=0;
 case 1  %Referece edge
  if UP==0
   RefTime=time;
   UP=1;
   if DN==1
    event('INS', @DPFD, time+50e-12,3);
   end
  end
 case 2  %Divider edge
  if DN==0
   DivTime=time;
   DN=1;
   if UP==1
    event('INS', @DPFD, time+50e-12,3);
   end
  end
 case 3  %PFD reset
  UP=0; DN=0;
  PFDSign=(DivTime>RefTime); %Determines the sign of the phase/frequency error
  event('INS', @DEC, time+20e-12,1);
end
```

stored in the global variable *PFDsign*. A new event, with another 20ps delay, is inserted to the event queue so that the phase/frequency decision circuit can process this phase error.

4.6.3 The model of the frequency divider

The frequency divider accepts the rising edge events generated by the DCO and generates an divider edge event for the DPFD once every *N* DCO edge events. The matlab code for the frequency divider model is given in Listing 4.7. The time of the DPFD event is calculated as "time+100e-12" so that a 100ps time delay of the frequency divider is modeled. Again, some additional noise can still be added here if necessary.

4.6.4 The model of the decision circuit

With the similar method as other blocks, the phase/frequency decision circuit and the DCO can be created easily. The key parts of the decision circuit model, the frequency decision, is given in Listing 4.8, in which the variable *Delta* can be zero, +/-1, indicating

Listing 4.7. Model of the frequency divider.

```
function FDIV(time,para)
global Div_Edges
persistent COUNT N;
switch para
 case 0
  COUNT=0; N=20; %Initialize the counter
 case 1  %DCO edge event
  COUNT=COUNT+1;
  if (COUNT==N)
   COUNT=0;
   event('INS', @DPFD, time+100e-12, 2);
  end
end
```

Listing 4.8. The part of the frequency decision

```
    if PFDSign==0 UP=0; end
    if PFDSign==1 DN=0; end

    Count=Count+1;
    if(Count==8)  %updating period=8
       Count=0;
       Delta=(UP-DN);
       UP=1; DN=1;
    end
```

no frequency change, frequency up, frequency down respectively.

4.6.5 The model of the digitally-controlled oscillator

The model of the digitally controlled oscillator first calculates the output frequency and the output period, then generates oscillator edge events. It is important that the frequency may change after the next edge event is generated. In this case, the old edge event has to be replaced with the new edge event calculated based on the new frequency. The script to conduct such task is given in Listing 4.9.

Listing 4.9. The frequency update of the DCO

```
if(NewPeriod~=Tvco)
       event('DEL', @DCO, 0, EventID); %Remove old next edge
       NextEdge=(NextEdge-time)/Tvco*NewPeriod+time;
       event('INS', @DCO, NextEdge ,1);
       Tvco=NewPeriod;
end
```

4.7 Event-driven simulation results

With the models and the event-driven engine created above, various types of anal-

ysis can be performed to exam the performance of the digital PLL. With the same parameters (Table 4.2) as used in the Simulink model, the locking behavior can be observed by plotting the control words of the DCO and the output of the PFD. To further speed up the simulation, the DCO gain as well as its center frequency is divided by 10 and the division ratio of the frequency divider is set to 2 instead of 20. The DPLL modeled above is simulated for 50μs within 12 seconds CPU time on a 2GHz P4 PC, while the simulation using simulink may take hours to achieve a reasonable time resolution for the noise analysis. Figure 4.28 shows the simulation result from the event-driven simulation. The loop

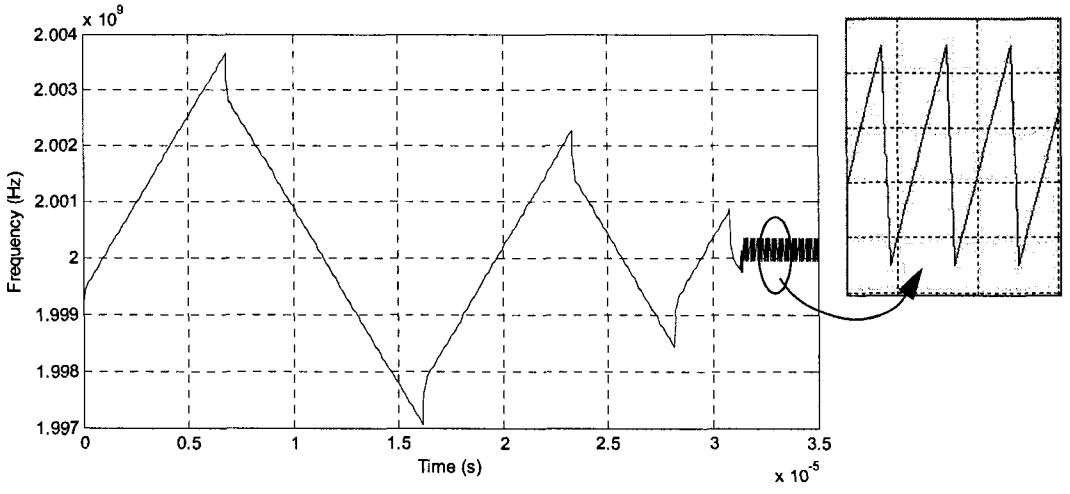


Figure 4.28 The simulated locking behavior of the DPLL

acquires locking within 32μs, which is different from the result in Figure 4.24 on page 79 due to the different initial condition. Because of the phase/frequency quantization and the bang-bang operation, the frequency is not constant after the loop is locked, but jumping among several discrete frequency levels resulting an average frequency equal to the desired frequency. Because no random noise was modeled in the DCO or the reference source, the simulation shows a periodical variation of the DCO frequency. This periodicity should disappear once the reference or the DCO phase noise was included. More details can be found in the section "Phase noise created by the proportional path quantization." on page 55. The simulation result matches with the results obtained by Simulink simulation

exactly, confirming the feasibility of the modeling and simulation technique.

The main advantages of the event-driven simulation technique are the simulation speed and accuracy, which make various kinds of analysis possible. The jitter tracking capability and noise transfer characteristic of the proposed digital PLL is analyzed using the event-driven simulation technique described above.

First, a sinusoidal jitter denoting the edge variation is added to the reference clock and the edge jitter of the frequency divider output is monitored to observe the tracking capability of the proposed digital PLL. Figure 4.29 shows the input and output jitter with the reference clock corrupted by 1MHz sinusoidal jitter whose amplitude is 2ps. The digi-

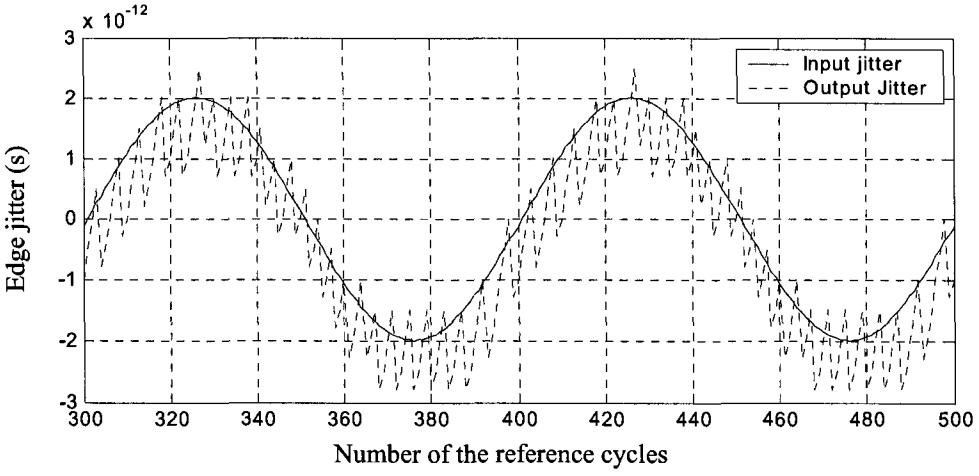


Figure 4.29 Correct jitter tracking of the digital PLL

tal PLL tracks the input jitter correctly and the high frequency jitter is due to the bang-bang operation of the digital PLL. However, when the input jitter amplitude is increased by a certain amount, the output can no longer track the input as shown in an example in Figure 4.30, where the jitter amplitude is 70ps. Although the frequency domain transfer function does not apply to a non-linear system such as the proposed digital PLL, an approximate frequency domain jitter transfer characteristic is obtained by the simulation with different noise amplitudes for better understanding the loop tracking behavior.

A single-tone phase variation or noise with certain frequency and amplitude is

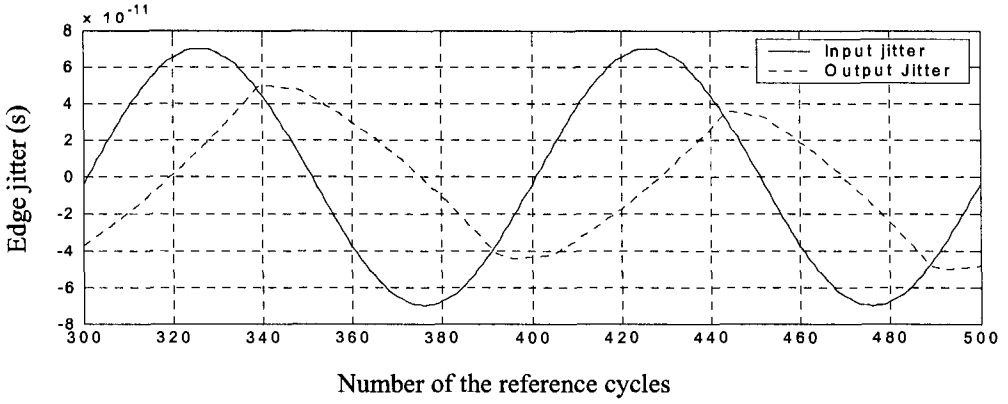


Figure 4.30 Jitter tracking failure of the digital PLL

added in the reference source and the edge timing jitter of the divider output is examined. A simulation with two-dimension parameter sweep, the noise amplitude and noise frequency, is done and the jitter at the output of the frequency divider is calculated. The power ratio of the output noise power to the input noise power for different noise amplitudes (*An*) is plotted as shown in Figure 4.31. It shows that the DPLL exhibits different

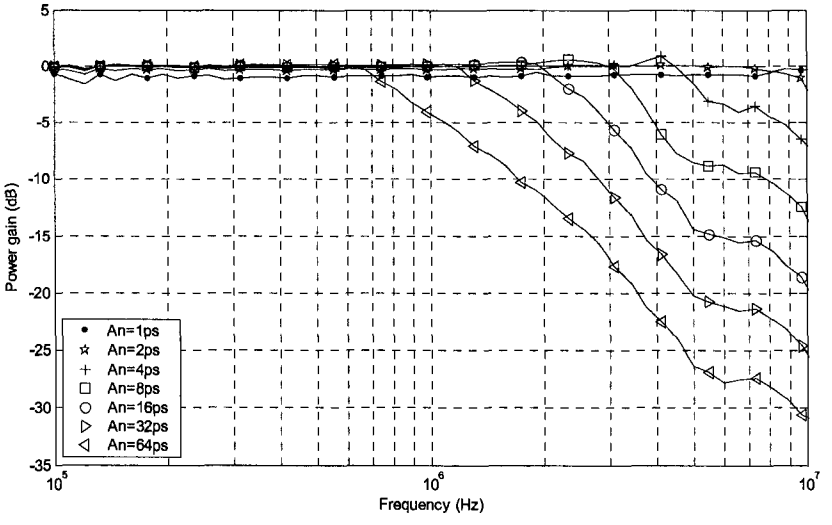


Figure 4.31 The reference phase noise transfer characteristic

loop bandwidths for different noise amplitudes, which is due to the non-linear and quantization operation of the loop as the analysis given in "The loop bandwidth of the proposed DPLL" on page 51.

4.8 Phase noise performance simulation

The noise transfer characteristics obtained in the last section are based on a single sinusoidal phase noise assumption. As analyzed earlier in this chapter, the proposed DPLL exhibits some unique features comparing with most conventional PLLs. For instance, the design parameters are independent of the reference phase noise level for optimum noise performance. However, the noise performance is difficult to analyze in the frequency domain due to the non-linear operation. The design parameters are mainly determined by the DCO phase noise and they are derived by using time-domain analysis. To confirm the theoretical analysis, the noise performance of the proposed DPLL was simulated using the event driven technique described above.

4.8.1 The simulation configurations

In the simulation, the noise profile of the free-running DCO is assumed and included in the DCO model after it is converted to time domain noise [23][24]. Different DCO and reference phase noise profiles are assumed in this simulation to examine the theoretical analysis and to confirm the noise performance of the proposed DPLL. The simulation results of several typical cases are provided and explained in the following paragraphs.

The cases to be simulated are listed in Table 4.3, in which the phase noise N_1, N_2, N_3 refer to the flicker noise up-conversion region, the thermal noise up-conversion region and the noise floor respectively.

4.8.2 Simulated phase noise performance of the proposed DPLL

In case 1, it is assumed that the reference is noiseless and the DCO has a noise profile of -98dBc/Hz at 100kHz in the up-converted flicker phase noise region, -118dBc/Hz at 1MHz offset in the up-converted thermal noise region with the noise floor of -160dBc/Hz. The calculated proportional path frequency step based on Equation (4 . 24) is used in the

TABLE 4.3: The cases of the noise performance simulation

<table>
<tr><th rowspan="2">No.</th><th colspan="2">DCO and the reference source</th><th colspan="3">The loop parameters</th></tr>
<tr><th>DCO frequency and noise</th><th>Reference frequency and noise</th><th>PP. step</th><th>Dec. period</th><th>INT. step</th></tr>
<tr><td>1</td><td rowspan="4">Frequency: 3.9GHz
N_1: -98dBc/Hz@100kHz
N_2: -118dBc/Hz @ 1MHz
N_3: -160dBc/Hz</td><td>Frequency: 100MHz
N_1: N/A
N_2: N/A
N_3: N/A</td><td rowspan="3">18kHz</td><td rowspan="4">100*Tref</td><td rowspan="4">4kHz</td></tr>
<tr><td>2</td><td>Frequency: 100MHz
N_1: N/A
N_2: N/A
N_3: -133dBc/Hz</td></tr>
<tr><td>3</td><td>Frequency: 100MHz
N_1: N/A
N_2: N/A
N_3: -150dBc/Hz</td></tr>
<tr><td>4</td><td>Frequency: 100MHz
N_1: N/A
N_2: N/A
N_3: -150dBc/Hz</td><td>100kHz</td></tr>
<tr><td>5</td><td rowspan="2">Frequency: 3.915GHz
N_1: N/A
N_2: -94dBc/Hz @ 1MHz
N_3: -120dBc/Hz</td><td>Frequency: 145MHz
N_1: N/A
N_2: N/A
N_3: -133dBc/Hz</td><td>340kHz</td><td>100*Tref</td><td>5kHz</td></tr>
<tr><td>6</td><td>Frequency: 145MHz
N_1: N/A
N_2: N/A
N_3: -123dBc/Hz</td><td>340kHz</td><td>100*Tref</td><td>5kHz</td></tr>
</table>

simulation. With the reference frequency of 100MHz and the frequency multiplication ratio of 39, the output phase noise and the DCO free-running phase noise are plotted as shown in Figure 4.32.

From the figure, it can be observed that the DCO phase noise is attenuated efficiently and the effective loop bandwidth is approximately 10MHz.

In case 2, the reference phase noise is assumed to have a flat spectrum at -133dBc/Hz. With the same loop parameters as in case 1 (because the DCO phase noise is the same as in case 1), the DPLL is simulated and the phase noise performance is given in Figure 4.33.

As expected, the effective loop bandwidth is observed to be narrower than that of case 1 and the DPLL still has an optimum loop bandwidth. See "Optimum loop bandwidth

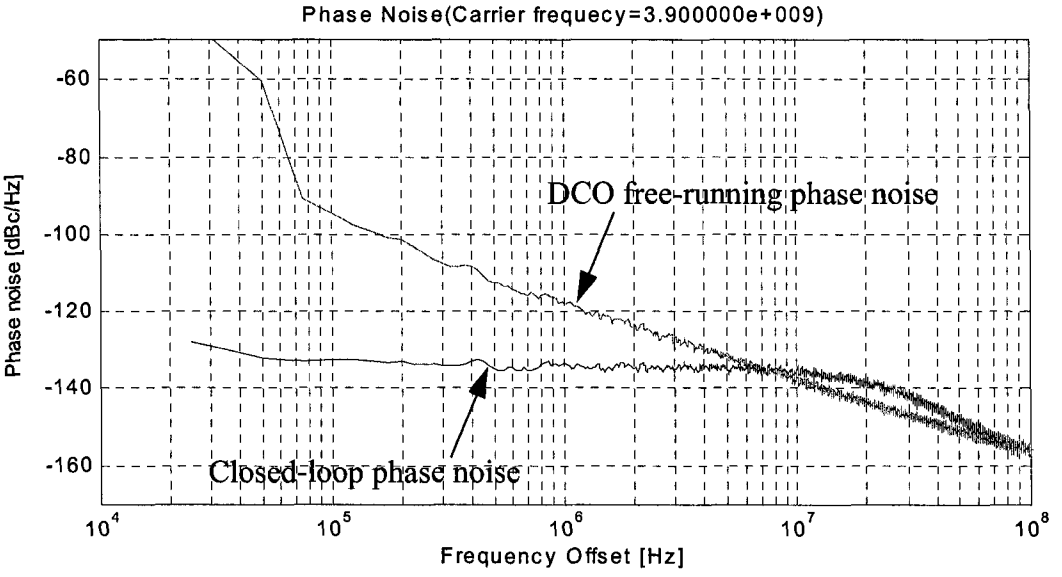


Figure 4.32 The simulated phase noise performance of case 1

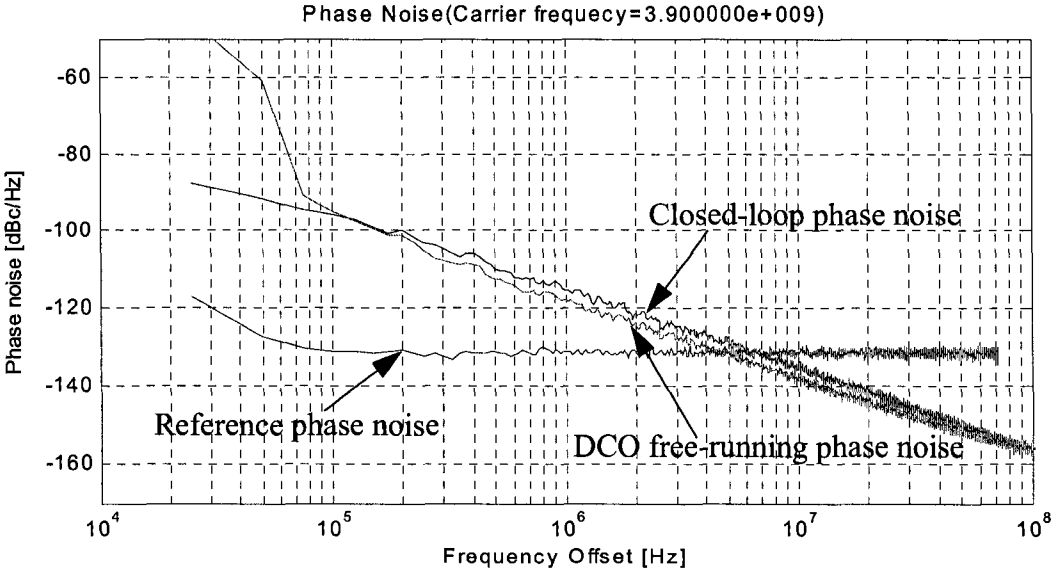


Figure 4.33 The simulated phase noise performance of case 2

of the digital PLL" on page 62 for related analysis.

In case 3, lower reference phase noise than that of case 2 is assumed and the simulated results are plotted in Figure 4.34. The reference phase noise has a flat spectrum at -150dBc/Hz, and the effective loop bandwidth is wider than that of case 2 because the reference phase noise is lower. The PLL behavior of the reference phase noise adaptation is

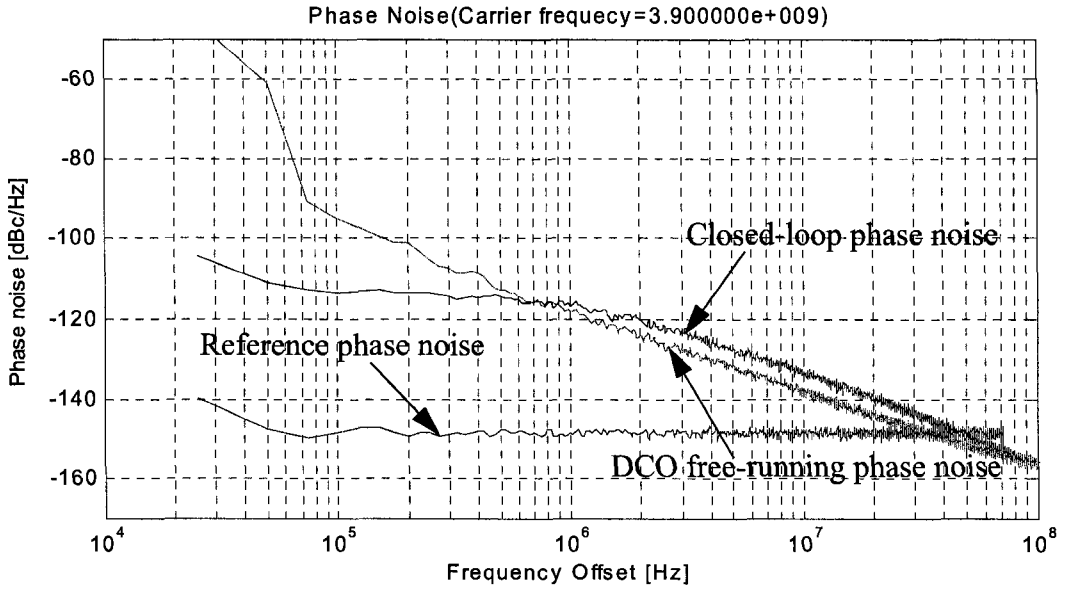


Figure 4.34 The simulated phase noise performance of case 3

clearly observed from Case 1, 2 and 3, which agrees well with the theoretical analysis presented earlier in this chapter.

Case 4 is the case where the proportional path frequency step is set to 100kHz, far larger than the calculated optimum value (18kHz) based on Equation (4 . 24). Based on the analysis before, in this case, the DPLL allows more reference phase noise to pass and appear at the output, resulting in increased output phase noise. The actual simulation results are given in Figure 4.35, which shows an effective loop bandwidth far larger than the optimum one as expected.

A different DCO phase noise profile is assumed in case 5 and case 6. The DCO now has a noise profile of -94dBc at 1MHz offset in the up-converted thermal noise region with the noise floor of -120dBc/Hz. Based on the DCO phase noise, the required proportional path frequency step size is calculated to be 340kHz for a reference frequency of 145MHz.

In case 5, the reference phase noise is assumed to have a flat phase noise of -133dBc/Hz and the frequency multiplication ratio is 27. The simulated phase noise perfor-

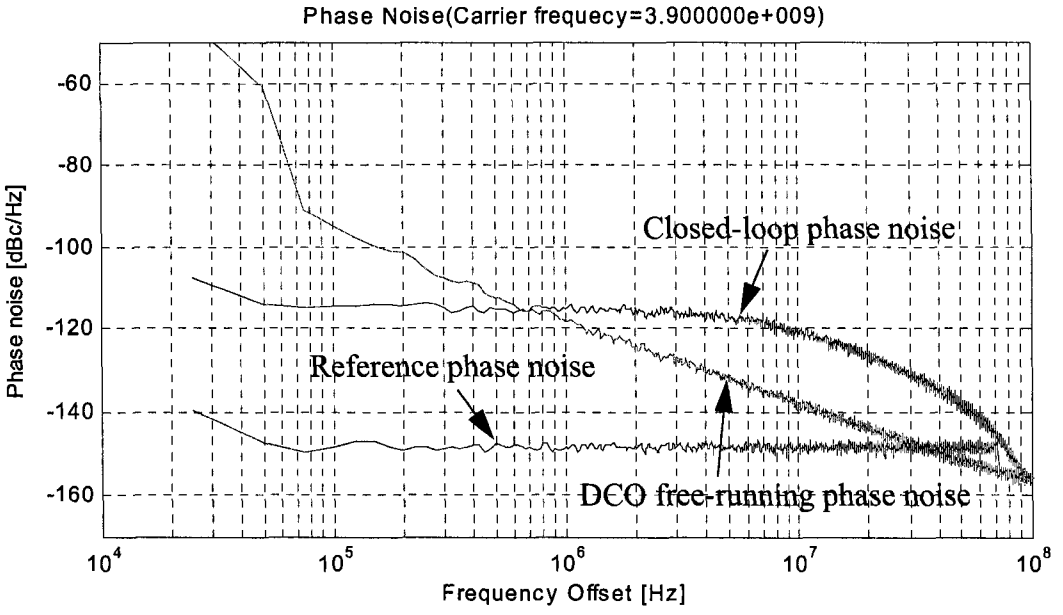


Figure 4.35 The simulated phase noise performance of case 4

mance is given in Figure 4.36, from which the DCO phase noise is attenuated and the ref-

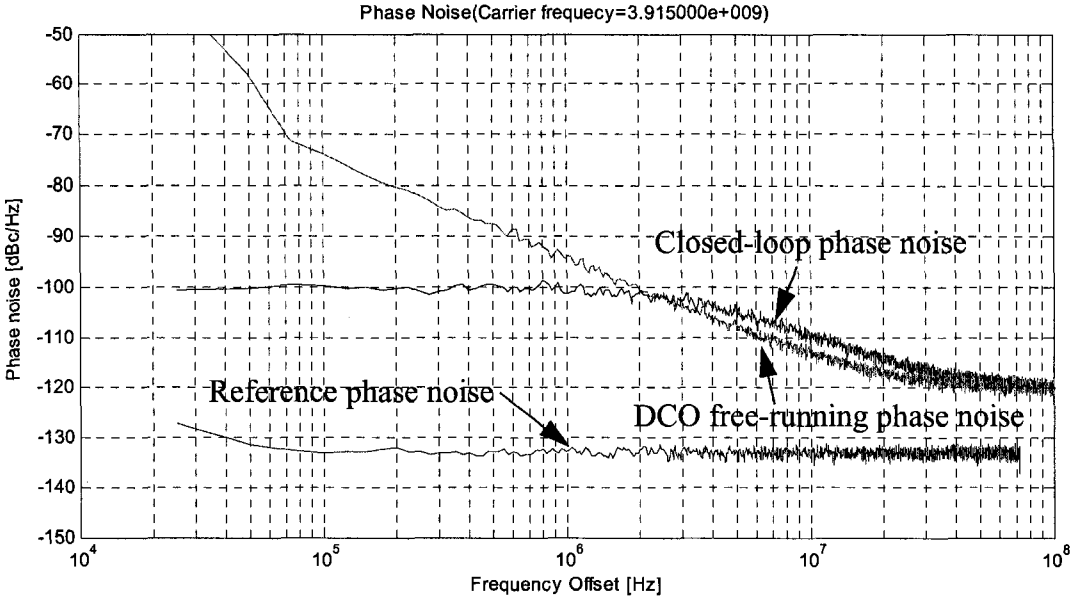


Figure 4.36 The simulated phase noise performance of case 5

erence phase noise contributes to the phase noise of the low-frequency offset portion. Similarly, in case 6, when the reference phase noise is increased, the effective loop bandwidth is reduced to keep the optimum phase noise performance as shown in Figure 4.37.

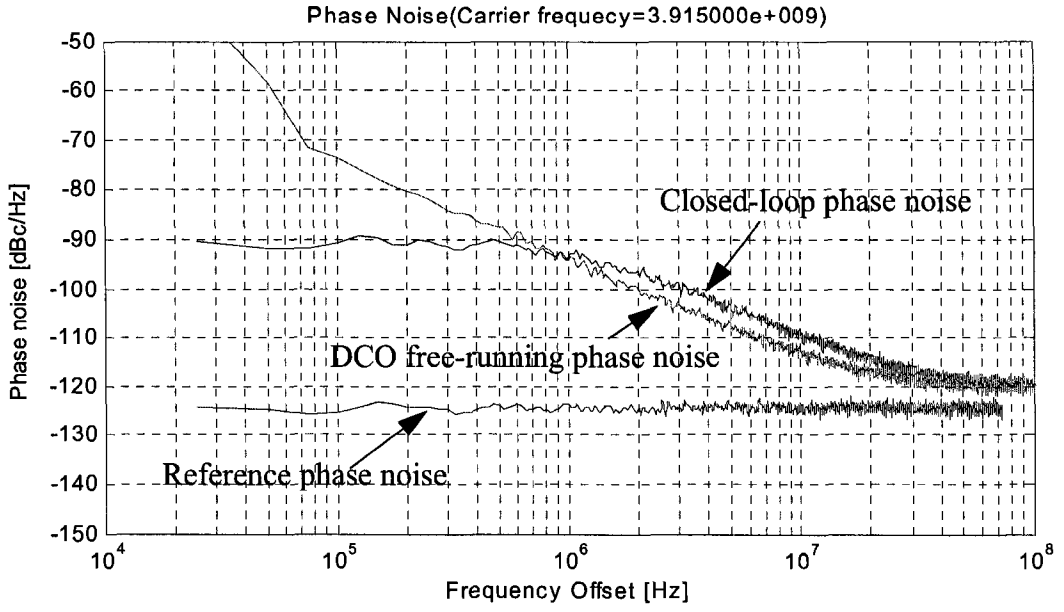


Figure 4.37 The simulated phase noise performance of case 6

In this case, the reference phase noise is assumed to be flat at -123dBc/Hz and the effective loop bandwidth is approximately 1MHz.

It can be concluded that the noise performances of all different simulation cases agree well with the theoretical analysis. To further confirm the feasibility of the proposed DPLL and the theoretical analysis, one of the cases (case 5) is compared with the measurements presented in chapter 6, in section "The measurement results of the DPLL" on page 132.

4.9 Summary

This chapter first gives the theoretical analysis of the noise behavior of the proposed DPLL, followed by modeling simulations using Simulink and the event-driven technique. The digital PLL was simulated with Simulink and the locking behavior was obtained. To efficiently simulate the loop, an event-driven modeling and simulation technique using Matlab, is proposed. The proposed technique achieves a super fast and accurate time-domain simulation, and it allows various noise sources to be easily considered

and analyzed. For example, the jitter tracking capability and the jitter transfer characteristics are obtained. With this technique, the noise performance of the proposed DPLL was simulated with different parameters and the results were compared with the theoretical analysis to verify the theoretical analysis.

CHAPTER 5

CMOS implementation of the proposed DPLL

5.1 Introduction

The proposed DPLL was analyzed theoretically and simulated in the previous chapter, and the behavioral simulation results well agree with the theoretical analysis. To further confirm the loop behavior and the feasibility of the proposed DPLL and the DCO, the proposed DPLL and DCO were implemented in ST 90nm CMOS technology for performance evaluation. This chapter describes the CMOS implementation of the DCO and the DPLL.

Figure 5.1 shows the block diagram of the proposed DPLL with four levels of

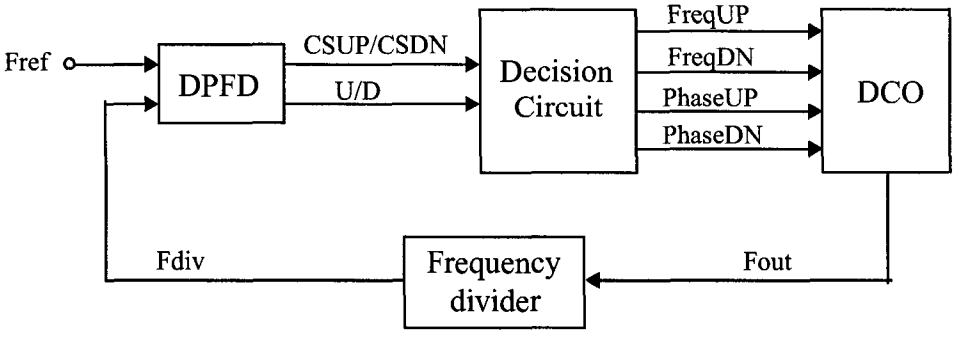


Figure 5.1 Block diagram of the DPLL

phase error quantization. More quantization levels of the DPFD and the decision circuit can be added to reduce phase/frequency acquisition time if the acquisition speed is of con-

cern. Because the phase noise performance at the locked condition is to be evaluated with the test chip, the cycle-slip detection circuit, which does not affect the phase noise performance in the locked condition, was eliminated from the test chip while its implementation is still given in this chapter. The following subsections describe the building blocks separately.

5.2 The DCO implementation

The DCO in this digital PLL, as shown in Figure 5.2, consists of four parts: the

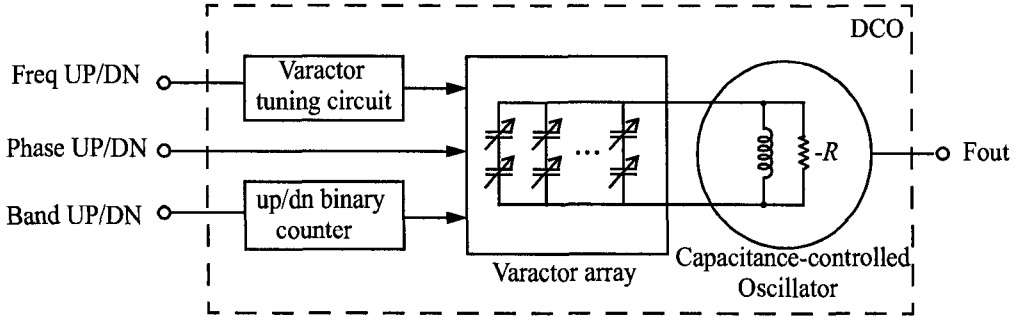


Figure 5.2 The proposed digitally controlled oscillator

capacitance-controlled oscillator, the varactor array, the one-hot or thermometer code generator, consisting of bidirectional shift registers, and the phase tuning path. Each part is described in the following sub-sections.

5.2.1 The implementation of the oscillator

The core oscillating circuit used in the proposed DCO is similar to that of a conventional VCO. Figure 5.3 shows the schematic of the oscillator, which is an LC tank based oscillating circuit and the transistors M_1 to M_4 form a positive feedback to inject energy to the LC tank and to maintain the oscillation. The bias current is provided by mirroring the reference current I_{ref} with a simple current mirror consisting of M_5 and M_6.

5.2.2 The digital to capacitance converter

The oscillation frequency of an LC-based oscillator can be tuned by tuning either

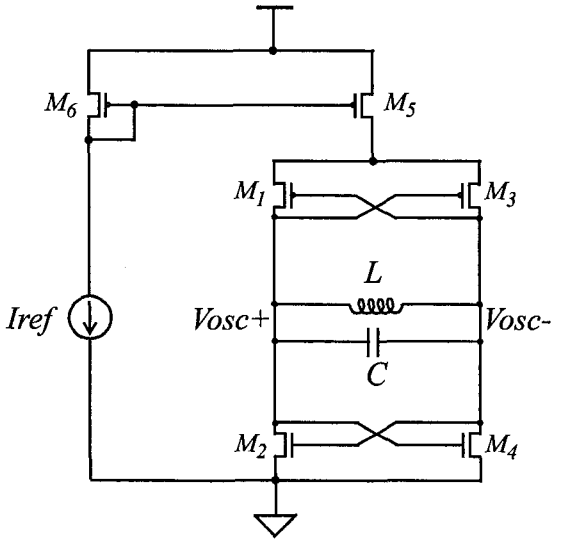


Figure 5.3 The schematic of the oscillator

the inductance (L) or the capacitance (C). Because switching or tuning the inductance normally degrades the quality factor of the LC tank, the capacitance tuning became a common approach. In this work, the frequency tuning of the DCO is achieved by replacing the capacitor (C) of the LC tank with a high-resolution digital tunable captor as described in the following section.

5.2.2.1 High resolution varactor array

Different types of varactors or transistors may be used as the capacitance tuning unit. To minimize the effect of the analog noise coming from the control line and to achieve better phase noise performance than its analog counterpart, the desired capacitance tuning unit should have two distinguishable low-gain regions as illustrated in Figure 5.4, in which two low-gain regions are used as the two digital tuning levels of the varactor[13]. A PMOS transistor with its drain/source/bulk tied together exhibits a similar characteristic when the capacitance between the bulk and the gate (C_{BG}) is measured against the voltage across them (V_{BG}), and it is employed in this work. Figure 5.5 shows the simulated capacitance tuning characteristic of such a PMOS transistor. For each single unit, the capacitance step size is the capacitance difference between the two levels (C_{ON} and

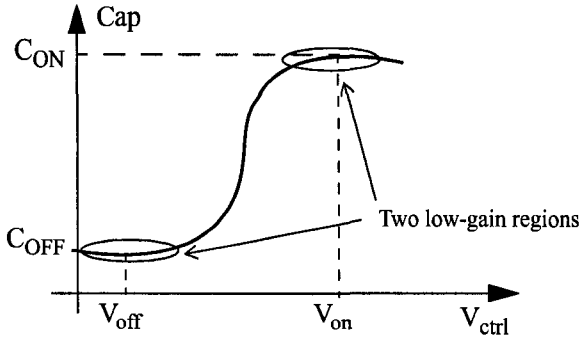


Figure 5.4 The characteristic of the capacitance tuning unit

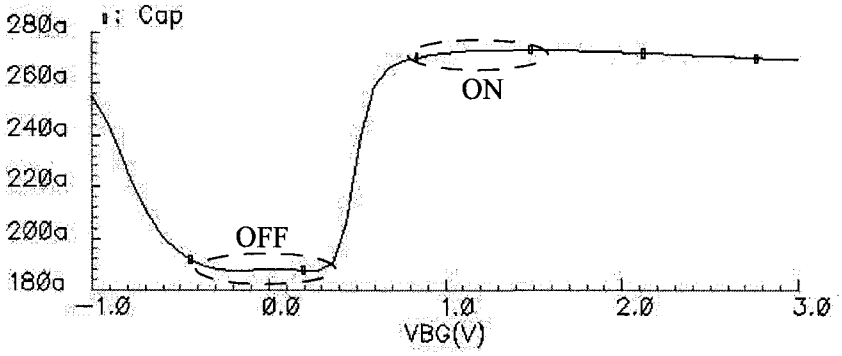


Figure 5.5 Simulated capacitance tuning characteristic of a PMOS transistor

C_{OFF}) shown in Figure 5.5. By digitally controlling a set of parallel connected tuning units, a capacitance tuning circuit can be built and its output capacitance can be tuned with multiple levels. Assuming k tuning units are used, the tuning range, C_{max}-C_{min}, can be written as,

$$C_{max} - C_{min} = \sum_{i=1}^{k} (\Delta C_i D_i) \qquad (5.1)$$

where D_i is the ith bit of a k-bit thermometer-coded digital control word, ΔC_i is the step size of the ith tuning unit. With equally-sized tuning units, such a DCC can achieve a linear capacitance tuning with its step size equal to the step size of one single tuning unit (ΔC_i), and the number of steps is equal to the word length (k). To realize a large tuning range with a small tuning step, multiple tuning banks with different step sizes are needed. Figure 5.6 shows the capacitance tuning of a DCC with two banks of tuning

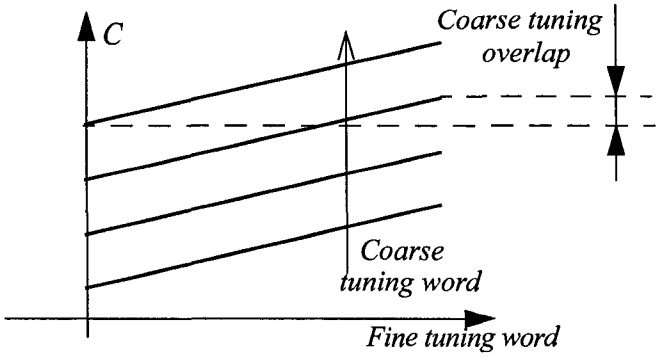


Figure 5.6 Continuous coverage of multiple tuning banks

units, the coarse tuning bank and the fine tuning bank. If the two banks are matched perfectly, that is, the tuning range of the fine bank is matched with the step size of the coarse bank, the overlap can be reduced to zero to achieve a monotonic capacitance tuning across two banks. However, without a specific technique, it is normally difficult to match the fine and coarse banks and the certain overlaps are required to guarantee the continuous capacitance coverage. Consequently, to drive such a DCC, the digital circuitry, such as a digital loop filter in a digital PLL, has to be implemented to deal with the bank overlap.

In this work, the capacitance resolution is enhanced by incrementally sizing the tuning units and the overlap is avoided by matching fine and coarse tuning banks. The fine capacitance tuning bank is shown in Figure 5.7, in which only the effective capacitances,

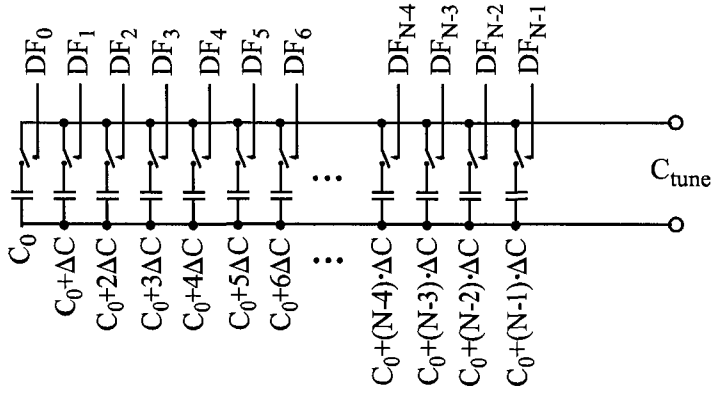


Figure 5.7 The proposed fine capacitance tuning bank

(the differences between C_{ON} and C_{OFF}), are shown. The digital control word is one-hot

coded, among which only one bit is one and its position represents the digital control value. Consequently, when the control word equals to *i* (*i=0, 1,... N-1*), DF_i is one and the output capacitance C_{tune} is $C_0 + i \cdot \Delta C$ with a tuning step size of ΔC.

To implement the tuning bank in Figure 5.7, using tuning units with similar characteristic as shown in Figure 5.4, a set of such units is sized incrementally as $W_i = W_0 + \Delta W \cdot i$, where *i=0, 1, 2 ... N*-1, W_0 is the size of the first unit, and ΔW is the step size of the tuning unit. Because the C_{ON} and C_{OFF} are scaled with the size of a tuning unit, the effective tuning capacitance, is also linearly related to the size of W_i, resulting in a fine capacitance bank with capacitance step of ΔC, which is expressed as,

$$\Delta C = (C_{ON,0} - C_{OFF,0})(\Delta W / W_0) \qquad (5.2)$$

where $C_{ON,0}$ and $C_{OFF,0}$ denote the C_{ON} and C_{OFF} of the first tuning unit respectively. Because normally ΔW can be much smaller than the minimum size of W_0 for a given CMOS technology, the capacitance step is reduced by the factor of $\Delta W/W_{min}$ with the proposed capacitance tuning scheme, comparing with the thermometer-coded fine tuning scheme, where the capacitance step is ($C_{ON,0}$ - $C_{OFF,0}$). In 90nm CMOS, the minimum transistor width is 0.12um and the capacitance resolution is enhanced by a factor of six if a width increment (ΔW) of 0.02um is implemented.

In addition to the proposed fine capacitance tuning bank, a coarse capacitance tuning bank, matched with the fine bank, is proposed to extend the capacitance tuning range without overlaps. Figure 5.8 shows the proposed coarse tuning bank. A thermometer-code digital word is used to switch a set of equally-sized capacitance tuning units, and each of them consists of two sub-units (C_0 and $C_0 + N \cdot \Delta C$) controlled by two inverted control signals. When the main control signal is high, C_0 is off while $C_0 + N \cdot \Delta C$ is on, and when it is low, C_0 is off while $C_0 + N \cdot \Delta C$ is on. The capacitance step of such coarse bank is thus $N \cdot \Delta C$, the difference between two sub-units. Two PMOS transistors are employed as two sub-units in this design. The advantage of such configuration is that two tuning

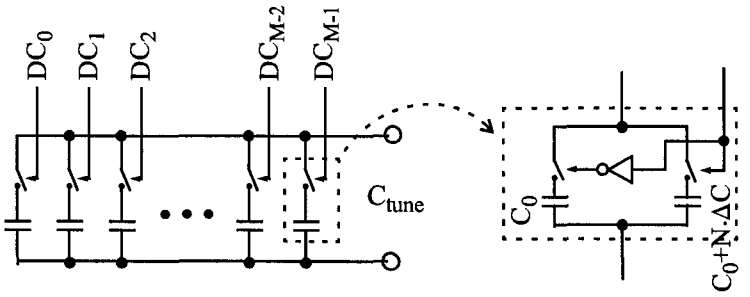


Figure 5.8 The proposed coarse capacitance tuning bank

units can be matched with the fine bank to achieve monotonic capacitance tuning across the two banks without overlaps.

To further extend the tuning range, a band tuning bank was implemented. Consequently, the DCC in the proposed DCO consists of 3 tuning bands, the band switching bank, the coarse tuning bank and the fine tuning bank. The fine tuning bank has 32 tuning levels, and the coarse tuning bank has 64 levels, resulting in 2048 (32x64) linear capacitance tuning levels in total with the matched coarse and fine banks. The fine and coarse banks are controlled by 11 binary bits, five of which are converted to a one-hot coded word to control the fine bank and six of which are converted to a thermometer-coded word to control the coarse bank. In addition, 64 bands were implemented with binary-weighted tuning units to extend the capacitance range. The resulting DCC has a capacitance range from 2pF to 3pF with the capacitance step of 10aF. Operating with a 1.8nH ($0.9nH \cdot 2$) inductor, the DCO has a frequency tuning range of 3GHz to 3.7GHz, and the frequency step size is approximately from 3kHz to 6kHz.

Similar to other DCOs, the layout of the varactor arrays are crucial for the DCO performance. Figure 5.9 shows the layout floor plan of the fine/coarse banks. The fine bank consists of 32 incrementally-sized units (*F0* to *F31*). *F0* to *F15* and *F16* to *F31* are placed in reversed order with two dummy units beside *F15/F16*. The first unit (C0, consisting of two subunits, the larger one marked as L0 and the smaller one marked as S0) of the coarse bank are placed adjacently to *F0* and *F31*.

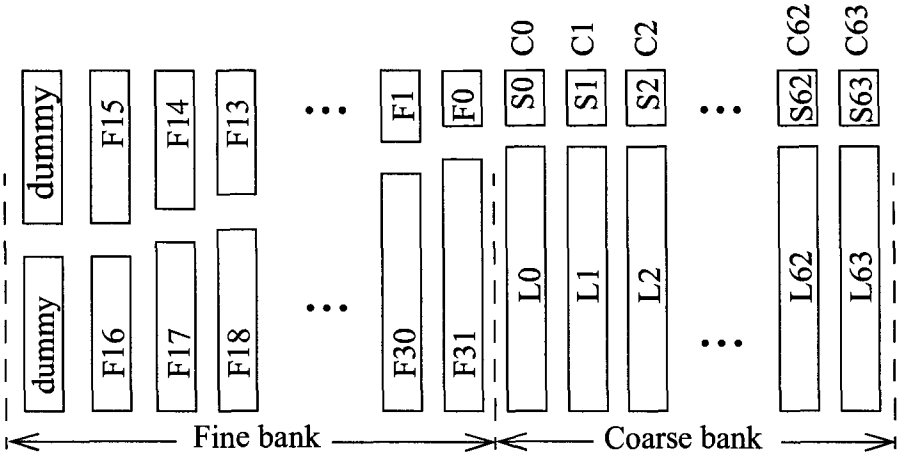


Figure 5.9 The layout floor plan of varactor banks

A special case, where the control word is changed from "000000,11111" to "000001,00000", is detailed here. There is a carry operation from fine bank to coarse bank in the transition and the desired capacitance increment is the same as the step size of the fine bank or the difference between Fn and Fn+1 (n=0, 1...30). Before the transition, in the fine bank, only F31 is on and in the coarse bank, the small subunit of C0 (S0) is on while the large subunit of C0 (L0) is off. After the transition, F31 is off and F0 is on. In the mean time, S0 is turned off while L0 is turned on. Consequently, the overall capacitance change is the capacitance difference between F31 and L0 plus the difference between S0 and F0. Because S0 and F0 are physically matched in the same transistor size and the difference between F31 and L0 is the same as the fine tuning step (the difference between Fn and Fn+1), the overall capacitance step is approximately the same as the fine tuning step size.

Assuming the tuning range of the capacitance is from C_{min} to C_{max} with the step size of C_{step}, the frequency tuning characteristic of the corresponding DCO can be derived as follows. First, the frequency tuning range corresponding to the capacitance tuning range is expressed as follows,

$$f_{DCO} = \left(\frac{1}{2\pi\sqrt{LC_{max}}}, \frac{1}{2\pi\sqrt{LC_{min}}} \right) \quad (5.3)$$

And the frequency step size is calculated as,

$$\Delta f = \left| \frac{\partial f}{\partial C} \times C_{step} \right| = \frac{f}{2C} \cdot \Delta C \qquad (5.4)$$

For a given capacitance step, the larger the capacitance, the smaller the frequency step. Consequently, for a given capacitance step and a desired oscillating frequency, it is possible to reduce the frequency step by reducing the inductance (L) and increasing the capacitance (C).

5.2.2.2 *The mismatch of the varactor*

According to the relation between the resonating frequency and the product of the inductance and the capacitance of an LC-tank based DCO, the relation between the frequency step size and the capacitance step size can be derived as Equation (5 . 4),

For a capacitance of 3pF and an inductance of 0.9nH (f= 3GHz), Equation (5 . 4) results in the frequency step as calculated below,

$$\Delta f = \frac{f}{2C} \cdot \Delta C = \frac{3 \times 10^{9}}{2 \times 3 \times 10^{-12}} \cdot \Delta C = \Delta C \times 5 \times 10^{20} \qquad (5.5)$$

Transistor-level simulation showed that the PMOS transistors at 2.5V with the length of 0.28μm, can achieve a capacitance step of 40aF, corresponding to a frequency step of approximately 20kHz at 3GHz, if the width increment is 0.05μm. With the same width increment, PMOS transistors at 1.2V with the length of 0.1μm can achieve a capacitance step of 25aF, corresponding to a frequency step of approximately 12.5kHz at 3GHz.

With a width increment of 0.02μm, a capacitance step of 10aF, corresponding to a frequency step of approximately 5kHz at 3GHz, can be achieved with such transistors. For a minimum allowed transistor width of 0.2μm, the proposed method enhances the frequency resolution by a factor of 10. However, to guarantee the monotonic frequency tuning of the DCO, the mismatch between the transistors' capacitance has to be less than

10%, for a frequency step of 10kHz or 25% for a frequency step of 25kHz.

A measured MOS varactor mismatch performance in 90nm CMOS is studied as shown in Figure 5.10 [25]. An estimation of the mismatch of the PMOS varactor is done

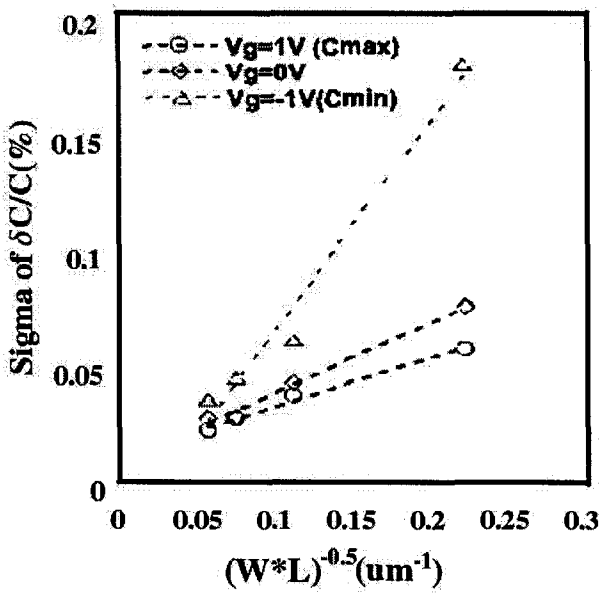


Figure 5.10 N+/NWell MOSVAR mismatch performance[25]

based on this data. From the Figure, the varactor has a superior capacitance mismatch of 0.04% (for $(W \cdot L)^{-0.5} = 0.2\mu m^{-1}$) and 0.02% (for $(W \cdot L)^{-0.5} = 0.05\mu m^{-1}$) when it operates in the accumulation mode (*Cmax*). For the smallest transistor (*W/L*=0.2μm/0.1μm), $(W \cdot L)^{-0.5}$ is $7\mu m^{-1}$ and the capacitance mismatch can be estimated[1] to be 1%, which may be significantly reduced if specific match techniques are used in the layout of the varactor array. The estimated mismatch, far smaller than the required 10% or 25%, makes it possible to use the proposed incrementally sized varactor array to enhance the frequency resolution.

1. This is estimated by extrapolating the graph in Figure 5.10 to $(W \cdot L)^{-0.5}=7\mu m^{-1}$.

A possible implementation of the fine frequency tuning bank can be a set of 1.2V PMOS transistors with the gate connected to the oscillating node and the drain, source, bulk connected to the digital control signal. All transistors have the same length of 0.1μm and the widths are 0.2μm, 0.22μm, 0.24μm ..., resulting a capacitance step of approximately 10aF, and frequency step of approximately 5kHz in a 3GHz DCO.

5.2.3 The varactor driver

To bias varactors to one of two distinct flat regions, the varactor control bits need to switch between two voltages, *VL* and *VH*, which are determined by the varactor characteristic. The varactor driver [13] is to convert the two standard digital logic output levels, 0V and 1.2V (VDD), to *VL* and *VH* respectively.

The schematic of the varactor driver circuit is shown in Figure 5.11, it is essen-

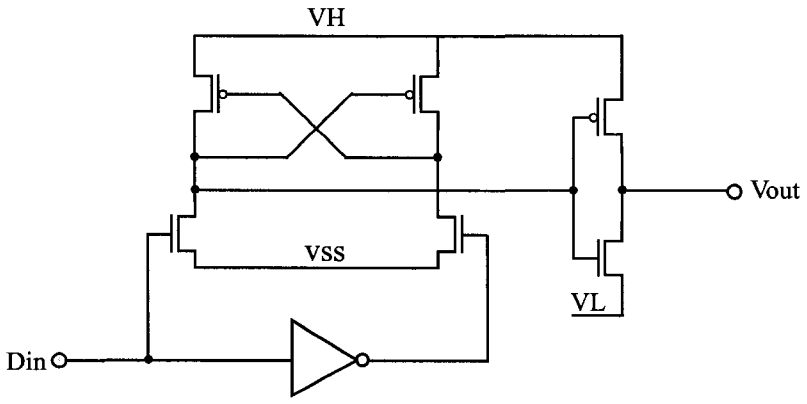


Figure 5.11 The schematic of the varactor driver circuit

tially a voltage comparator to compare *Din* and $\overline{Din}$, and the output is buffered with an inverter. The inverter for *Din* is supplied by the same supply of the digital circuit while the output buffer is supplied by *VH* and *VL*, two distinct voltages for the varactor tuning unit.

5.2.4 The varactor tuning circuit

The varactor tuning circuit is to control the varactor arrays, based on the input signal *FreqUp* or *FreqDN*. Whenever a *FreqUP* or *FreqDN* asserts, it updates its outputs (varactor control bits) so that capacitance is decreased or increased and the DCO frequency is increased or decreased.

Figure 5.12 shows schematic of the varactor tuning circuit, which consists two

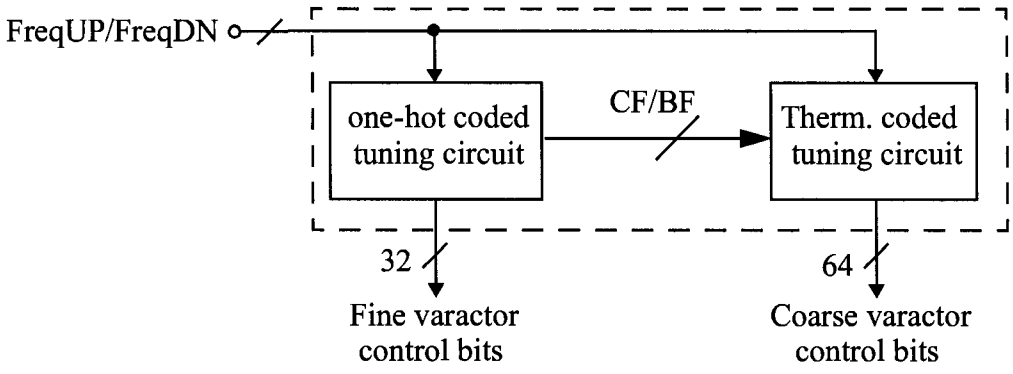


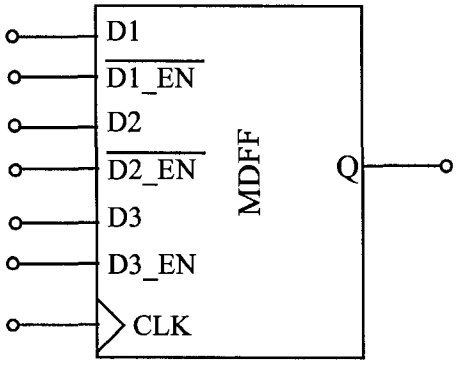
Figure 5.12 The varactor tuning circuit

parts, the one-hot coded tuning circuit to tune the fine varactor bank, the thermometer coded tuning circuit which tunes the coarse varactor bank. When the fine bank reaches its maximum or minimum value, the flag of "carry" (*CF*) or "borrow" (*BF*) is set and a further *FreqUP* or *FreqDN* causes a roll-over of the fine varactor control bits from its maximum value to its minimum or vice versa, resulting in an increase or decrease of the coarse varactor control value, which further causes the same increment or decrement of the fine tuning step as described in last section.

5.2.4.1 The multi-input D flip-flop

As described above, the basic operation of the varactor tuning circuit is to update the varactor control bits so that the capacitance can be increased or decreased by one unit step size. For a one-hot or a thermometer coded signal, such operation can be achieved by bit shifting operation. The basic unit circuit to realize the shift operation is a multi-inputs

D flip-flop (MDFF) as shown in Figure 5.13.



No:	Control bits	Operation
1	$\overline{D1\_EN}$=0 $\overline{D2\_EN}$=1 D3_EN=0	D1->Q @ CLK
2	$\overline{D1\_EN}$=1 $\overline{D2\_EN}$=0 D3_EN=0	D2->Q @ CLK
3	$\overline{D1\_EN}$=X $\overline{D2\_EN}$=X D3_EN=1	D3->Q @ CLK
4	$\overline{D1\_EN}$=1 $\overline{D2\_EN}$=1 D3_EN=X	CLK is disabled

Figure 5.13 The multi-input D flip-flop

There are three data inputs and three control bits in this D flip-flop. Four basic operations determined by the three control bits are listed in the right side of Figure 5.13. Operation No.1 and No.2 are used to perform the bi-directional operation, while the operation No.3 is used to load the value of the tuning circuit. If no shift operation is needed, the *CLK* path is disabled as in the operation No. 4.

The schematic of the multi-inputs D flip-flop is shown in Figure 5.14. Operations No. 1,2,3 are achieved by gating three inputs based on the control bits respectively, while the operation No. 4 is achieved by gating the path of the *CLK* signal. The control signal *D3_EN*, has higher priority than others to ensure the load operation. The resulting data and clock are fed to a regular D flip-flop to generate the output.

5.2.4.2 The one-hot coded tuning circuit

The one-hot coded tuning circuit is used to control the fine varactor bank. 32 MDFFs are used to hold the 32 control bits, one of which is initialized to "1", while others are initialized to "0". To realize the roll-over whenever carry or borrow operation is required, the last and the first MDFFs are connected together to form a cycle, in which the

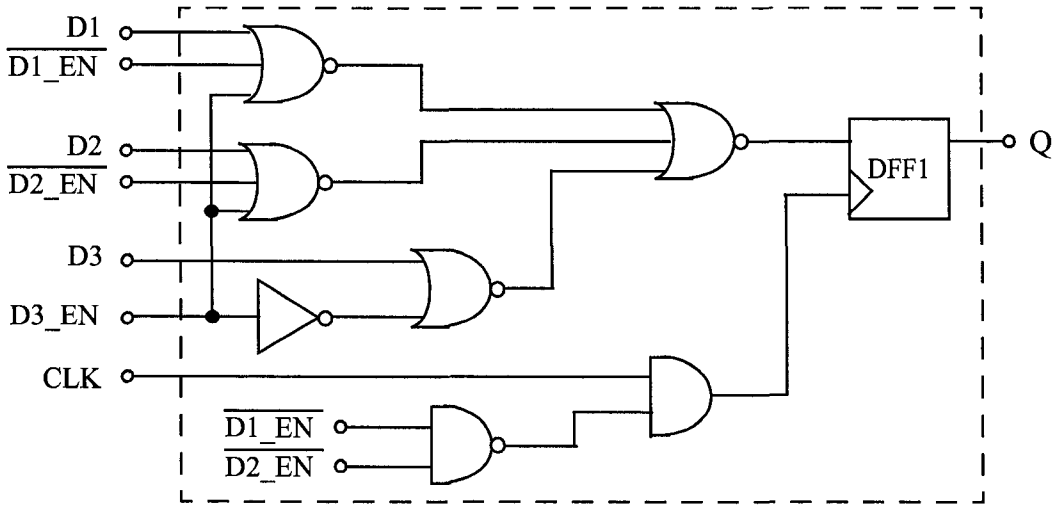


Figure 5.14 The schematic of the multi-inputs D flip-flop

"1" rotates according to the inputs *FreqUP* and *FreqDN*. The schematic of the fine varactor tuning circuit is shown in Figure 5.15. Some input and output buffers are not shown in

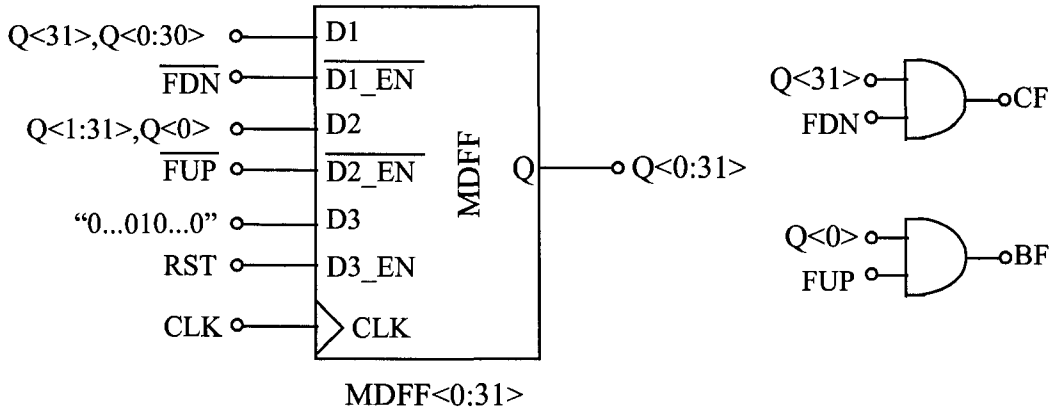


Figure 5.15 The schematic of the fine varactor tuning circuit

the figure for simplification. Two inverting buffers are inserted between the *FreqUP/ FreqDN* and *FUP/FDN*, while some non-inverting buffers are connected to the outputs of MDFFs and to the signals *CF/BF* to guarantee enough drive ability.

To initialize the output bits to a one-hot coded word, the *D3_EN* is set high and a

rising edge of the *CLK* is applied so that the input word at *D3* ("0...010...0") is loaded into the MDFFs.

When *FreqUP* is asserted, *FUP* is low and the path of *D2* is selected. The control bits Q<1:31,0> are shifted to Q<0:31>, so the capacitance is reduced if "1" is not at the location of Q<0>. Otherwise, *BF* is asserted and "1" is shifted to Q<31> while the decrement by one is performed by the coarse varactor tuning circuit.

5.2.4.3 The thermometer coded tuning circuit

The coarse varactor tuning circuit is similar to the fine varactor tuning circuit, but it has thermometer-coded output bits and its shift is determined by the *CF/BF* coming from the fine varactor tuning circuit. Figure 5.16 shows the schematic of the coarse varac-

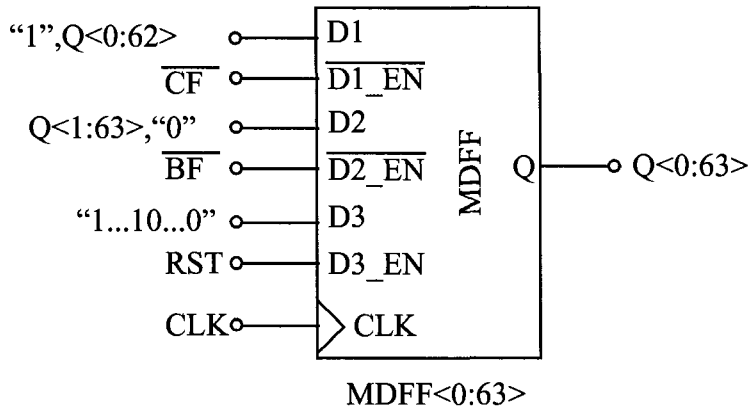


Figure 5.16 The schematic of the coarse varactor tuning circuit

tor tuning circuit. Instead of forming a cycle, 64 MDFFs form a bidirectional shift register, with the first one and the last one are always connected to "1" and "0" respectively.

The initialization is done by loading a thermometer-coded word, "1...10...0", and signal *CF/BF* causes a right/left shift so that the varactor is increased/decreased.

5.2.4.4 The phase tuning path

While the frequency of the DCO is tuned by the varactor tuning circuit, a phase

tuning is achieved by directly controlling the varactors with the outputs of the *UP/DN* sensor. To achieve a programmable phase tuning gain, two sets of binary-weighted varactors are employed to react to *PhaseUP/PhaseDN* respectively as shown in Figure 5.17. The

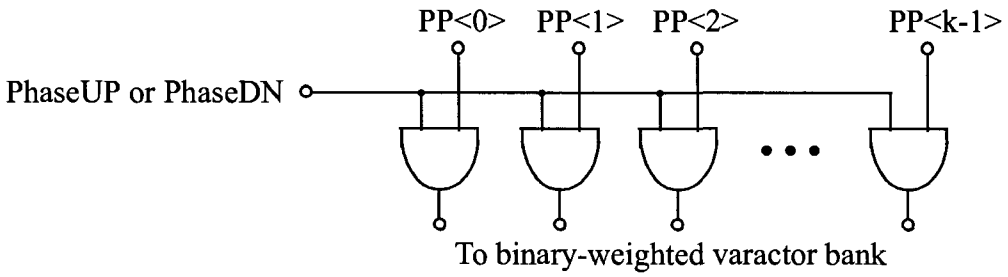


Figure 5.17 The schematic of the phase tuning circuit

binary phase gain control bits *PP*<0:k-1> switch on and off the paths of the corresponding varactors so that the *PhaseUP/PhaseDN* results in a given phase change. Five bits are used for path of *PhaseUP* or *PhaseDN* to control the phase gain, resulting in 2 x 2^5=64 levels of phase gain. Because the phase path has only two states, *PhaseUP* or *PhaseDN*, different gains (or different control words) can be applied to the paths of *PhaseUP* and *PhaseDN*.

5.3 The implementation of the digital phase detector

The proposed digital phase/frequency detector (shown in Figure 3.9 on page 46) is a combination of a tri-state PFD and an UP/DN sensor. A tri-state PFD, operating in a conventional charge-pump PLL, has no phase dead zone or hysteresis as long as there is enough delay in its reset path. However, for the detection of the polarity of the phase error using an UP/DN sensor, an improper design may result in a large dead-zone or hysteresis, which degrades the noise/jitter performance of the proposed digital PLL significantly.

In the Simulink model of the PFD given in Figure 4.17 on page 74, a single D flip-flop is employed as the UP/DN sensor. The principle of the polarity detection is illustrated in Figure 5.18, in which the signals *RST*, *UP* and *DN* are from the tri-state PFD and the signal *Q* is the sample of the signal *UP* at the rising edge of the signal *DN*. Obviously, for

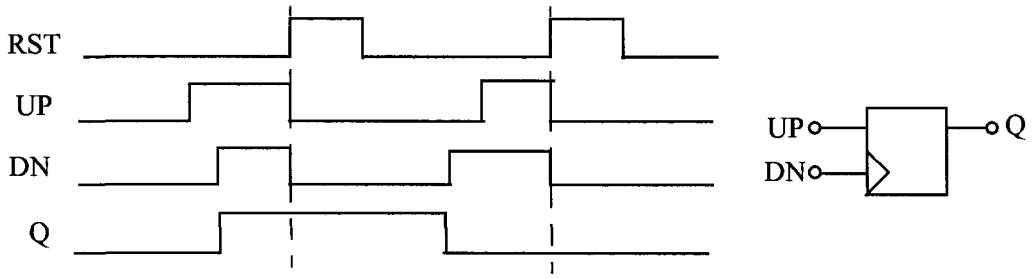
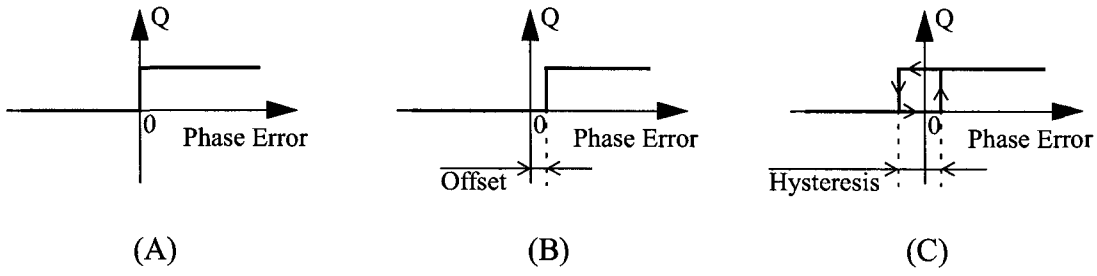


Figure 5.18 The principle of the UP/DN sensor

an ideal sampler, the sampling result Q indicates exactly the polarity of the phase error as shown in Figure 5.19(A). However, a real D flip-flop usually exhibits some offset and



(A). The characteristic of an ideal sign detector (B). The characteristic of a sign detector with non-zero phase offset (C). The characteristic of a sign detector with non-zero hysteresis

Figure 5.19 The characteristics of sign detection

some hysteresis as shown in Figure 5.19(B) and Figure 5.19(C) respectively. The phase error offset is equivalent to a delay in the reference clock path, and a constant phase error offset does not degrade the performance of the digital PLL. The hysteresis, however, directly affects noise/jitter performance [26] by linearly adding itself to the accumulated edge jitter of the DCO output. Consequently, the hysteresis in the UP/DN sensor has to be minimized.

A static D flip-flop may exhibit a ±20ps hysteresis, which can be significantly reduced (i.e. ±1.5 ps) by using a dynamic flip-flop as reported in [26]. However, the ±1.5 ps hysteresis is still not negligible comparing with the noise amplitude at the PFD input.

In this work, such hysteresis is avoided by using a resetable D flip-flop as shown in

Figure 5.20. After each *UP/DN* sensing operation, the DFF1 is reset while its output is

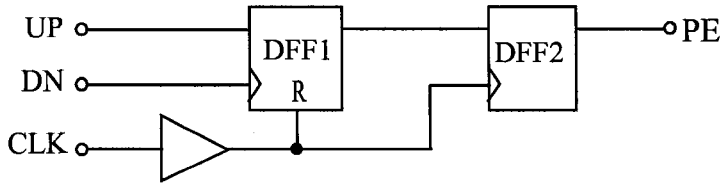


Figure 5.20 The schematic of the up/dn sensor

held by DFF2 as the output of the UP/DN sensor. Enough time should be allocated before the reset to allow the DFF1 to reach its stable states and the result is held by DFF2. Because the state change from "1" to "0" is avoided in the *UP/DN* sensing operation, this UP/DN sensor is free of hysteresis and even conventional static D flip-flops can be used. Although a non-zero conversion offset may exist, it does not impact the loop performance as long as it is constant.

To reduce the acquisition time, as described in "The proposed digital PFD" on page 45, a cycle-slip detector was implemented together with the tri-state PFD to provide fast frequency tuning. A cycle-slip detection circuit was proposed to produce an *CSup/ CSdn* pulse whenever a cycle slip happens. It is based on that fact that, when a cycle slip happens, the rising edge of *Fref/Fdiv* does not result in a rising edge of the tri-state PFD output, *UP/DN*. Figure 5.21 shows the schematic of the proposed cycle slip detector,

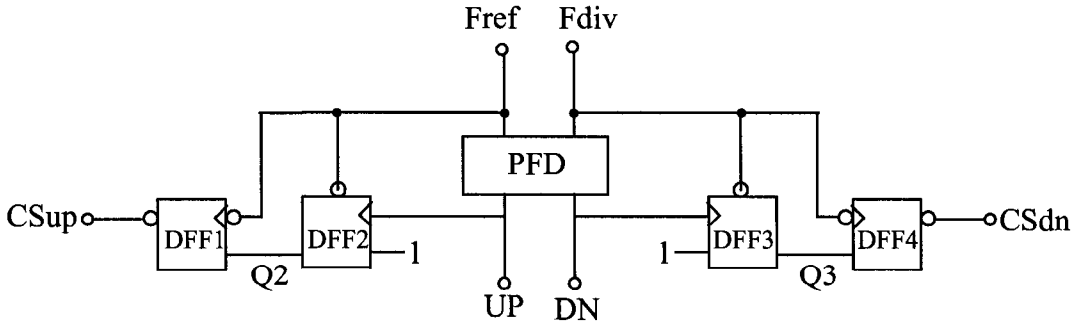


Figure 5.21 The proposed cycle slip detection circuit

together with a tri-state PFD. DFF2/DFF3 is set to be 1 at the rising edge of *UP/DN* as

long as *Fref/Fdiv* is high. DFF1/DFF4 samples and holds Q2/Q3 at the falling edge of *Fref/Fdiv*. The results are inverted to indicate the cycle slip (*CSup/CSdn*).

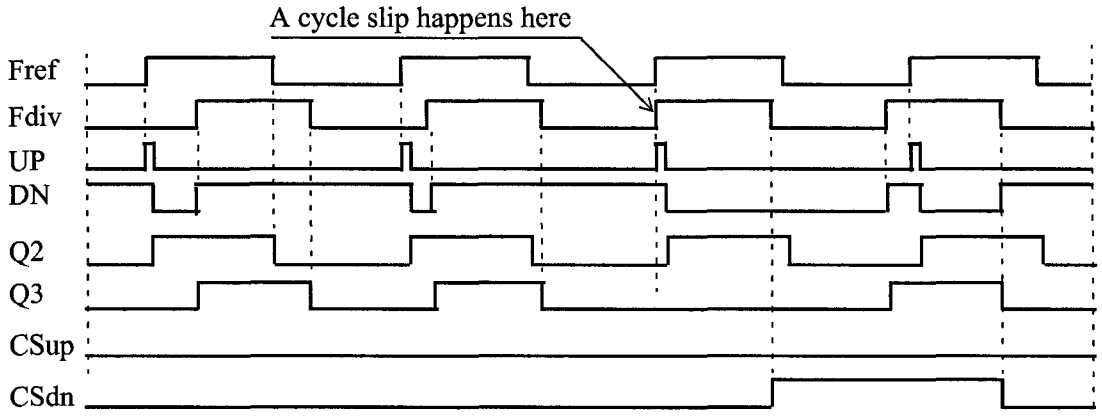


Figure 5.22 The principle of the proposed cycle slip detection circuit

Figure 5.22 shows the operating principle of the cycle slip detection, where *Fdiv* is larger than *Fref*. As illustrated in the figure, one of the rising edges of the *Fdiv* does not cause a rising edge of the signal *DN*, so Q3 is zero at the following falling edge, resulting in positive pulse at *CSdn*. The saturation detection is capable to produce a *CSup/CSdn* pulse whenever the phase error reaches $\pm 2\pi$, $\pm 4\pi$, $\pm 6\pi$...

5.4 Implementation of the frequency decision circuit

A simple frequency decision algorithm is proposed and described in "The proposed decision circuit" on page 47. It is repeated here for convenience. The decision on the frequency up/down is made after each updating period, *M* reference cycles or *M* divider cycles, based on the output of the UP/DN sensor during that updating period. If there are only positive phase errors in the updating period, *FreqDN* is generated. Similarly, if there are only negative phase errors in the updating period, *FreqUP* is generated. Otherwise, the frequency keeps unchanged. It is clear that the decision circuit consists of two parts, a circuit to generate the decision clock (*CLKdec*) to define the decision period (*M*), and a logical circuit to make decision at each rising edge of the decision clock. The deci-

sion clock can be generated by dividing the signal *PEReady* by *M*. The logical circuit has four inputs, *U, D, PEReady, CLKdec*, and two outputs, *FreqUP* and *FreqDN*. The Simulink model shown in Figure 4.21 on page 77 is for mathematical modeling purpose, and it is not a good solution for the circuit implementation because it requires memorization of all values of the input signals, *U/D,* during each updating period to make the decision. A simple implementing of the proposed algorithm is shown in Figure 5.23, where the signal

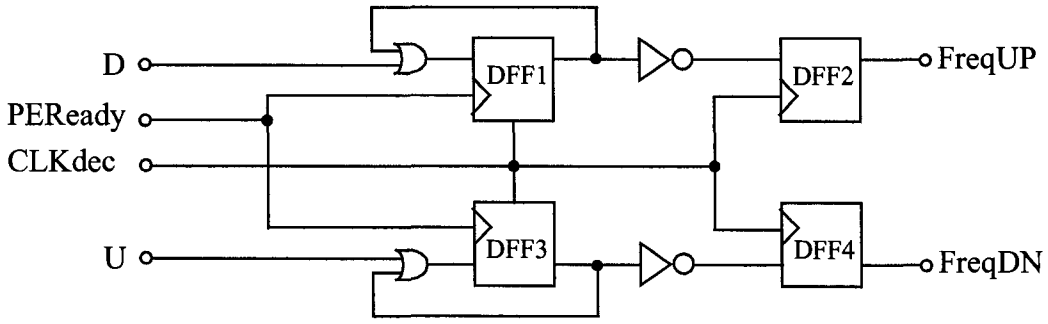


Figure 5.23 The logical circuit for frequency decision

CLKdec is a narrow positive pulse at the end of each updating period. The DFF1 and DFF3 are reset by the signal *CLKdec* after each decision, so that their outputs are always zeros at the beginning of each updating period. If all phase errors are positive in the following updating period (*U* is always zero), the output of DFF3 keeps zero and the output of the DFF4 is one (*FreqDN* is asserted) at the end of the updating period when the rising edge of *CLKdec* comes. Similarly, if all phase errors are negative during one updating period (*D* is always zero), the signal *FreqUP* is asserted. This low-complexity implementation can work with any updating period (*M*). The updating period of the proposed frequency decision circuit can be changed by programming the frequency division ratio of the circuit which generates the signal *CLKdec* as shown in Figure 5.24. Proper timing alignment between signals *PEReady* and *CLKdec* are required to guarantee the correct operation. In this work, as shown in the figure, the frequency divider output is sampled by the *PEReady* and the sampled results are passed to a pulse generator to generate *CLKdec*.

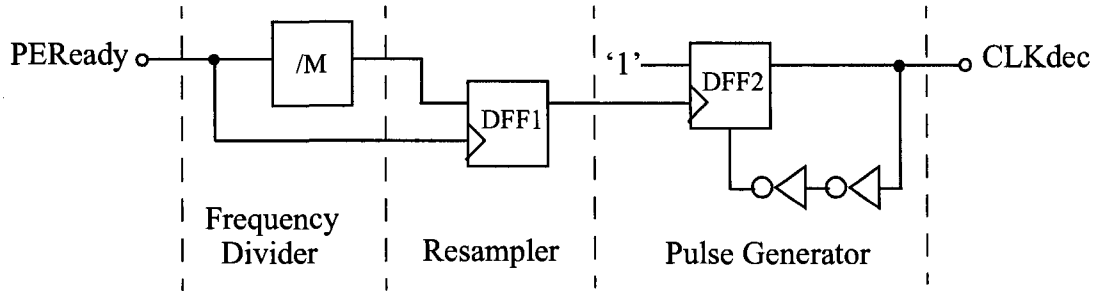


Figure 5.24 The decision clock generator

The design of the frequency divider is given in the following section.

5.5 Implementation of the frequency divider

Two frequency dividers are employed in this proposed DPLL, one is used to divide down the DCO frequency before it is compared with the reference signal and another one is used in the frequency decision circuit to generate the decision clock. Both frequency dividers are built based on the frequency divider with a division ratio of 2 or 3.

5.5.1 The divide-by-2, 3 frequency divider

The divide-by-2, 3 frequency divider divides the input clock by 2 or 3 depending

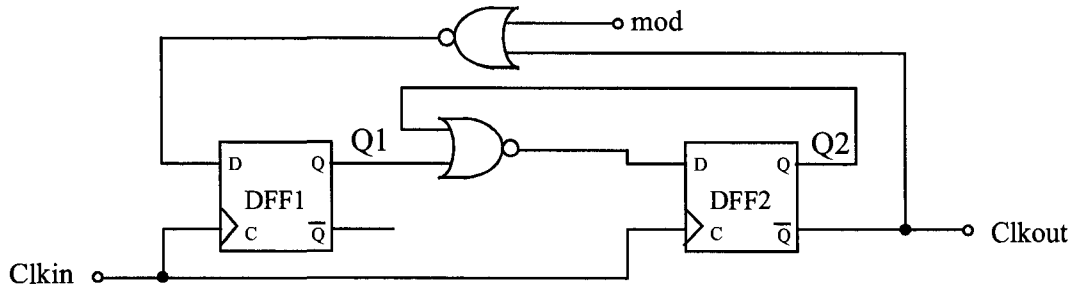


Figure 5.25 The schematic of the divide-by-2, 3 frequency divider

on the mode selection signal *mod*. Figure 5.25 shows the schematic of the frequency divider.

When *mod* is "1", the data input and output of DFF1 are always zero, and the output data of the DFF2 is always inverted and fed into DFF2, so the DFF2 is configured to be a simple divide-by-2 divider and the frequency division ratio of the whole divide-by-2, 3 frequency divider is two.

When *mod* is "0", two outputs of two flip-flips (*Q1* and *Q2*) are changed on each rising edge of the input clock *Clkin* as, "00" -> "01" -> "10" -> "00" -> "01" ..., so the frequency division ratio of the whole divide-by-2, 3 frequency divider is 3.

With the divide-by-2, 3 frequency divider, different frequency dividers, with more programmable frequency division ratios can be implemented. The DCO frequency divider and the frequency divider for frequency decision clock generation are described in the following sub-sections respectively.

5.5.2 The DCO frequency divider

The DCO frequency divider is to divide down the DCO output before it is compared with the reference signal. In this work, 8 different division ratios are achieved using three divide-by-2, 3 dividers (Div 2, 3) as shown in Figure 5.26. A three-bit mode control

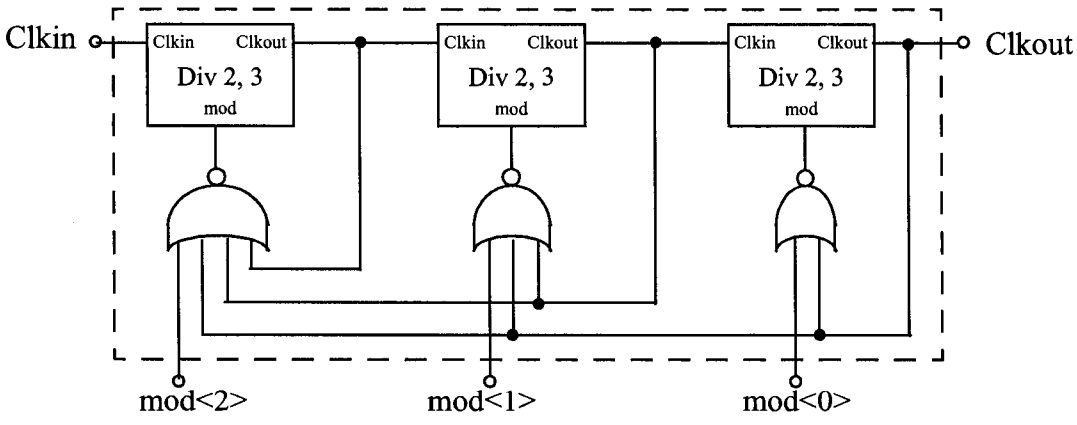


Figure 5.26 The schematic of the DCO frequency divider

word (*mod*<2:0>) is used to control the frequency divider ratio as listed in Table 5.1.

TABLE 5.1: **The frequency division ratio of the DCO frequency divider**

mod<2>	mod<1>	mod<0>	Frequency division ratio
0	0	0	14
1	0	0	15
0	1	0	17
1	1	0	18
0	0	1	23
1	0	1	24
0	1	1	26
1	1	1	27

5.5.3 Frequency divider for frequency decision clock generation

As shown in Figure 5.24, the frequency decision clock is generated using a frequency divider with the division ratio of *M*, which determines the decision period. To achieve a large range of decision period, the frequency divider for the decision clock generation provides a wide range of the division ratio (from 2 to 315), controlled by a 10-bit

control word (mod<9:0>) as listed in Table 5.2.

TABLE 5.2: The division ratio of the frequency divider for the frequency decision

mod<9:0>	Division ratio	mod<9:0>	Division ratio	mod<9:0>	Division ratio
0000100010	2	1001001100	44	1111010000	112
0000100011	3	1101000100	45	1011011000	120
0000101100	4	1111001000	48	1101010000	126
0000100100	5	1011000100	50	0101010100	130
0000101000	6	0101001100	52	1001011000	132
0000100000	7	1101001000	54	1101010101	135
1110000000	8	1001000100	55	1011010000	140
1100000000	9	1111000000	56	0111011000	144
1010000000	10	1011001000	60	0001010100	150
1000000000	11	1101000000	63	1001010000	154
0110000000	12	1111011100	64	0101011000	156
0100000000	13	0101000100	65	1101011001	162
0010000000	14	1001001000	66	1001010101	165
0000000000	15	1011000000	70	0111010000	168
1111000010	16	0111001000	72	0001011000	180
1101000010	18	0001000100	75	0101010000	182
1011000010	20	1001000000	77	1101010001	189
0000110001	21	0101001000	78	0101010101	195
1001000010	22	1111010100	80	0011010000	196
0111000010	24	0111000000	84	1001011001	198
0101000010	26	1001011100	88	0001010000	210
1101000011	27	0001001000	90	0111011001	216
0011000010	28	0101000000	91	0001010101	225
0001000010	30	1111011000	96	1001010001	231
1111001100	32	0011000000	98	0101011001	234
1001000011	33	1011010100	100	0111010001	252
1101001100	36	0101011100	104	0001011001	270
0101000011	39	0001000000	105	0101010001	273
1111000100	40	1101011000	108	0011010001	294
0011000011	42	1001010100	110	0001010001	315

The frequency divider in the decision circuit consists of three sub frequency dividers, having division ratios of 2~3, 4~7, 8~16 respectively, and four 2-to-1 MUXes. Each division ratio is achieved by a certain combination of the three sub dividers while four MUXes are employed to configure those sub-frequency dividers to different combinations. The schematic of the frequency divider is shown in Figure 5.27.

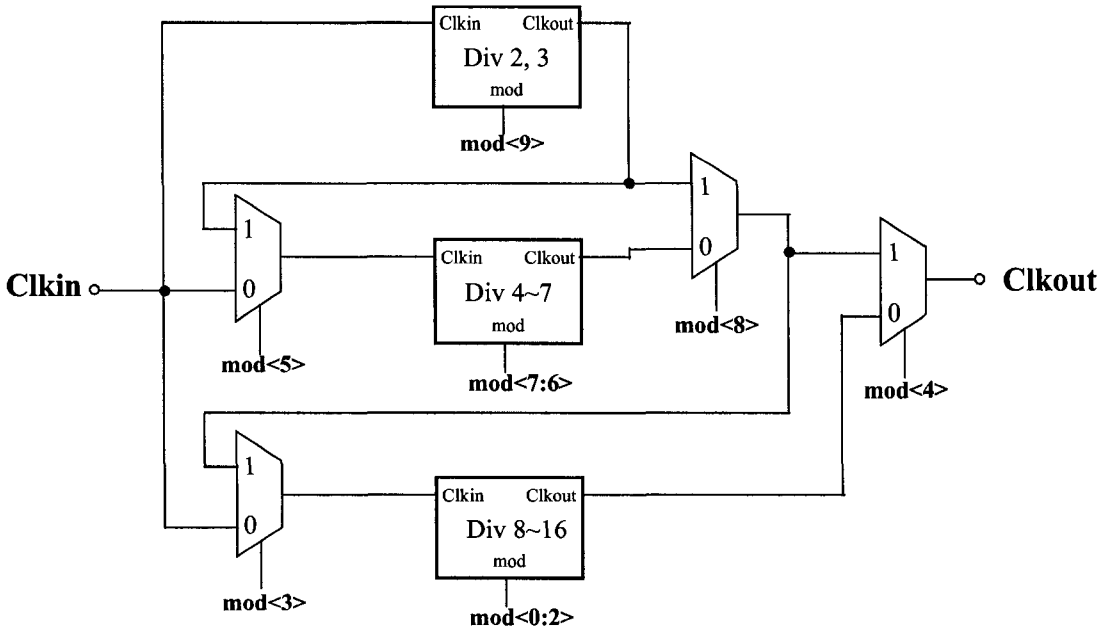


Figure 5.27 The schematic of the frequency divider for the frequency decision

The divide-by-2,3 divider in this frequency divider is the same as the one shown in Figure 5.25, except for that the NOR gates are replaced by NAND gates. Three such divide-by-2, 3 dividers are used to form a frequency divider with division ratios from 8 to 16, in the same way as shown in Figure 5.26. The frequency divider having a ratio from 4~7 is implemented in this similar way but with only two such divide-by-2, 3 dividers.

5.6 The digital serial data interface

Many control bits/words are needed to be supplied externally to configure the digital PLL or the DCO. In this work, all control bits/words are supplied into the test chips using a three-wire serial data interface and stored in a memory with 16(bits) x 16(words) = 256 bits as shown in Figure 5.28. The three-wire serial interface converts the serial input data (*D_data*) to parallel data, 16 bits data, 4 bits address, and a signal "*Write*". The serial input data is provided together with the clock signal (*D_clk*), and the conversion from the serial data to the parallel data is achieved using shift registers as shown in Figure 5.29.

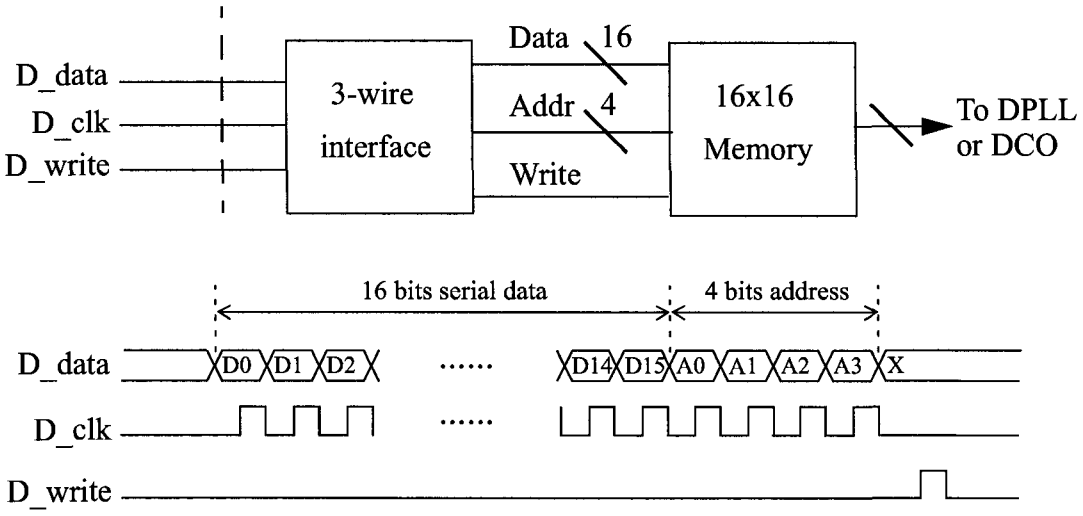


Figure 5.28 The data interface

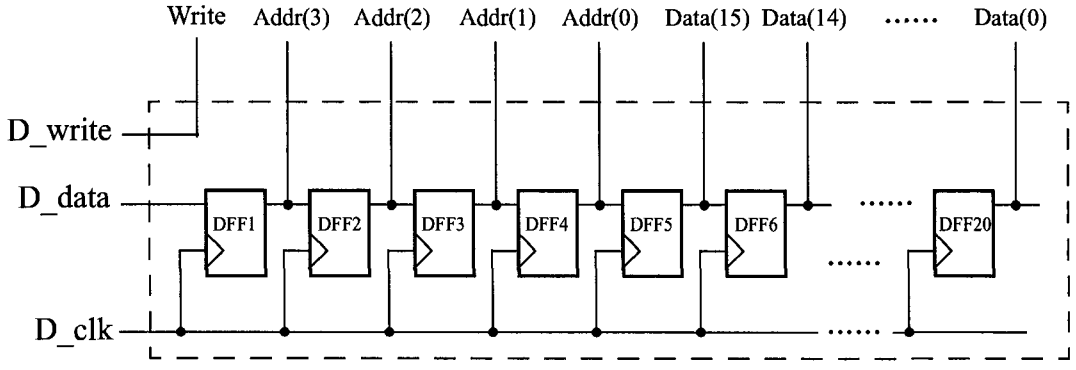


Figure 5.29 The three-wire interface

5.7 Summary

This chapter describes CMOS implementation of the proposed digital PLL, including the DCO, the digital PFD, the frequency decision circuit, and the frequency dividers. The implementation of the serial data interface, used to supply the control words/bits is also briefly described.

CHAPTER 6

Performance evaluation

6.1 Introduction

The proposed DCO and DPLL were described, analyzed and simulated in the previous chapters. Based on the circuit implementation given in the previous chapter, two test chips were fabricated in CMOS 90nm technology and measured to further confirm their feasibility. First, the proposed DCO was fabricated, together with the serial digital data interface. The frequency tuning characteristics and the phase noise performance were measured. Second, the proposed low-complexity digital PLL, together with the proposed DCO, was fabricated. The phase noise tracking and attenuation characteristics of the digital PLL were measured and compared with the theoretical analysis and the behavioral simulation given in Chapter 4. All measurements were done by probing the loose dice, using a wentworth probe station.

6.2 The test bench for the DCO measurement

The proposed DCO was fabricated in CMOS 90nm technology and measurements were done to confirm its feasibility. The inductor was implemented using metal 6 and 7 in parallel to reduce the resistive load, thus increasing the quality factor. Figure 6.1 shows the micro-graph of the DCO test chip. To measure the frequency tuning characteristic and the phase noise performance of the DCO, besides some DC power supplies, a data genera-

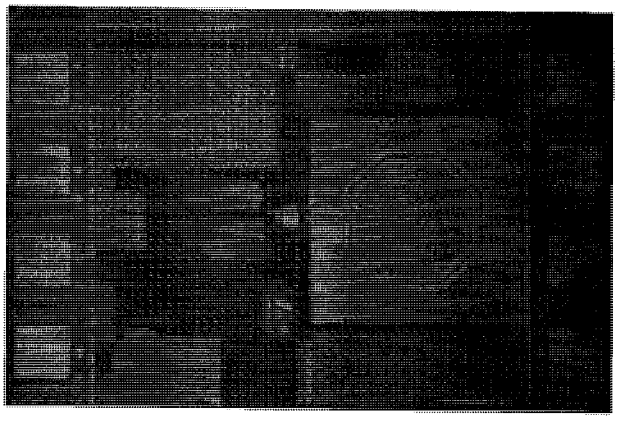


Figure 6.1 Micrograph of the chip

tor is used to supply the digital control bits to the DCO using the 3-wire serial data interface, and a spectrum analyzer is used to measure the output frequency and the phase noise as shown in Figure 6.2. In this work, all control data are generated using a PC with a par-

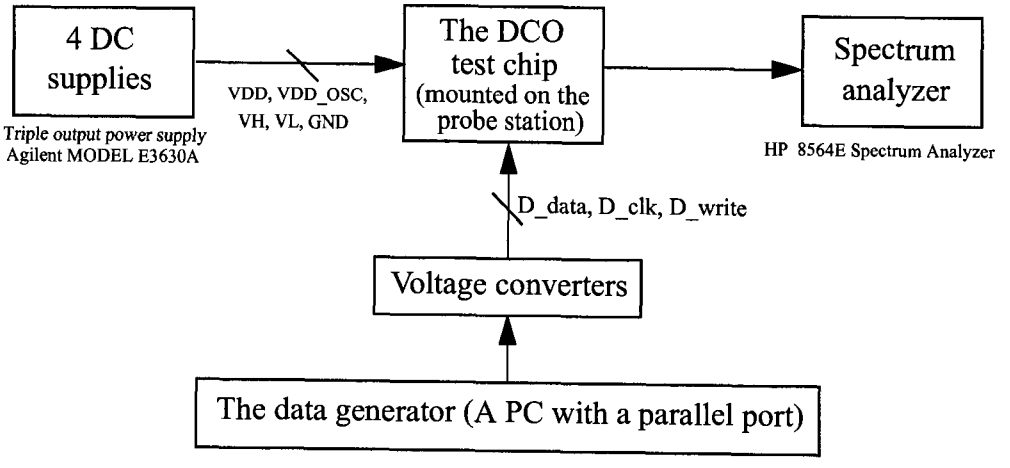


Figure 6.2 The test bench for the DCO

allel port, and a computer program with a graphic interface was developed to convert all DCO parameters to a certain format and to send all control words/bits to the 3-wire serial data interface automatically. Figure 6.3 shows a screen-shot of the DCO control panel running on a PC. Normally, when the DCO control parameters are changed, the program automatically compares the new parameters with the buffered parameters, and sends only the updated control words to the DCO chip. There is an option to send out all words, which is required at the initialization stage after the chip is powered on.

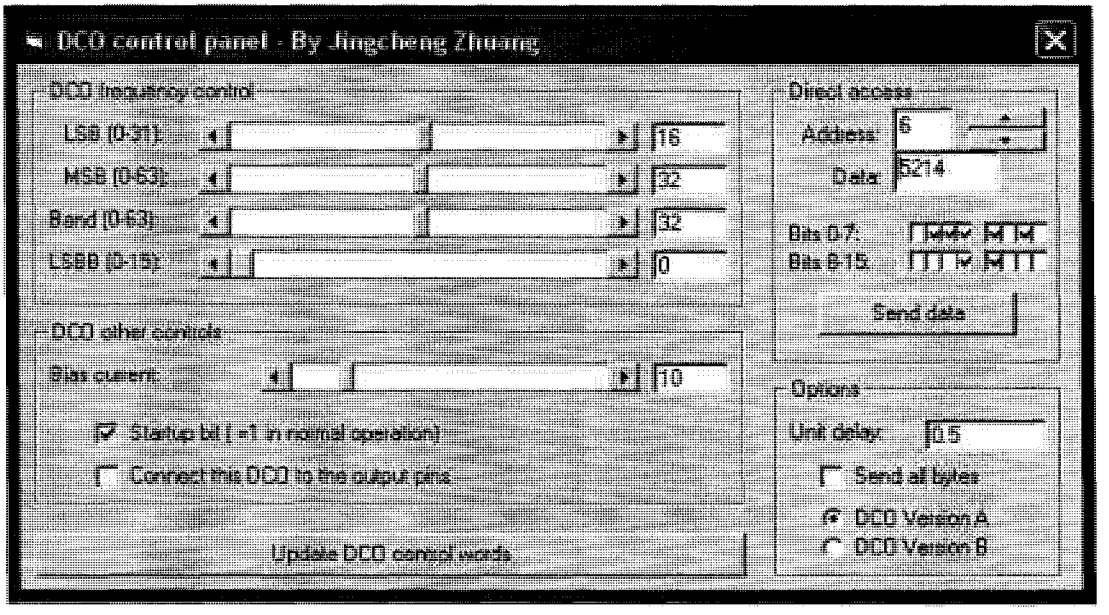


Figure 6.3 The screen-shot DCO graphic control panel

6.3 The measurement results of the DCO

For convenience, all digital control values are sorted so that larger values always correspond to higher frequencies. First, the band control word is changed from 0 to 63 and the frequencies for each band are measured with the coarse band word of 0 and 63 respectively. The results are plotted in Figure 6.4, which shows that the frequency tuning range of the DCO is from 3.06GHz to 3.7GHz.

There are 64 coarse tuning levels for each band, frequencies corresponding to each coarse tuning levels, together with the step sizes, are plotted for band 0 as shown in Figure 6.5. Because of the frequency drift of the free running DCO, measurement errors within a few kilohertz are expected in the results.

Because approximately 5kHz frequency steps are expected in the fine frequency tuning bank, it is impossible to characterize the fine bank by measuring the frequencies for different control words independently due to the natural frequency drift of the DCO. In this work, the fine frequency tuning characteristic is measured indirectly by measuring all

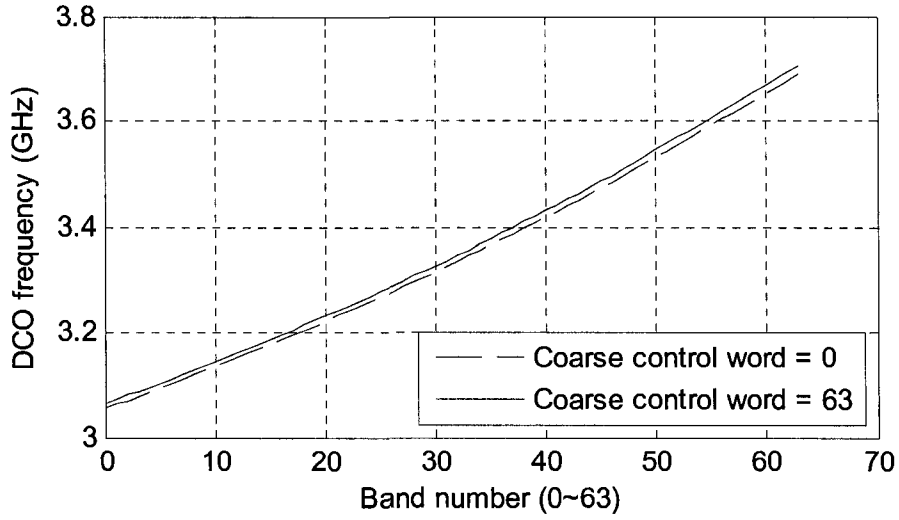


Figure 6.4 The measured band tuning characteristic of the DCO

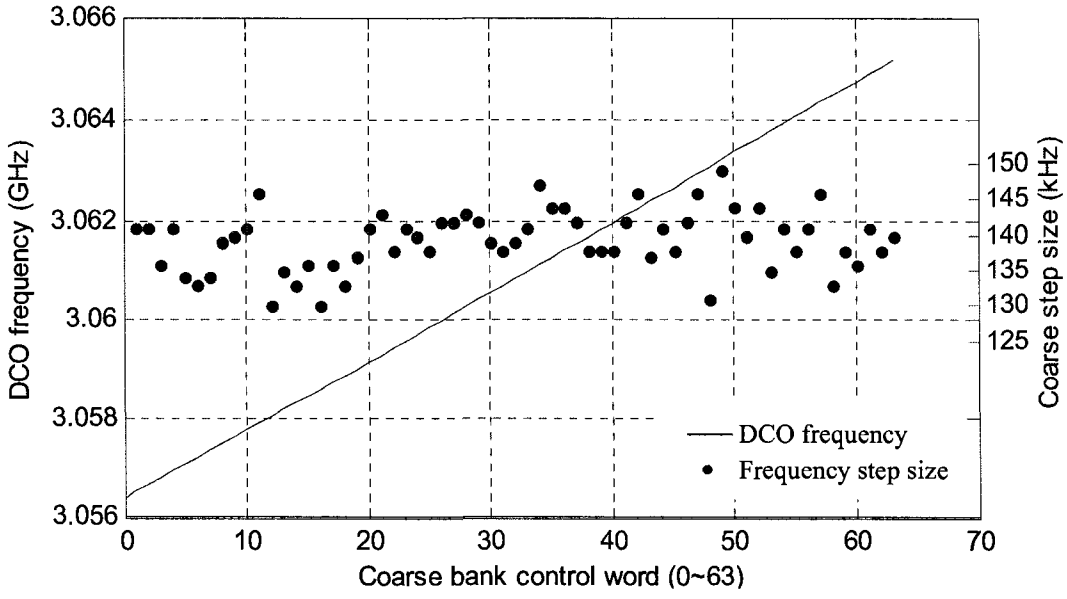


Figure 6.5 The measured coarse tuning characteristic of the DCO

fine frequency steps. The fine control word is quickly switched back and forth between two adjacent values for certain times and the frequency increments/decrements are measured and averaged to eliminate the scatter. Once the frequency step sizes for every two adjacent control word are obtained, the actual frequency tuning curve is constructed by accumulating the measured frequency steps and scaling the result to the measured fine fre-

quency tuning range. Figure 6.6 shows the obtained results, in which the fine control word

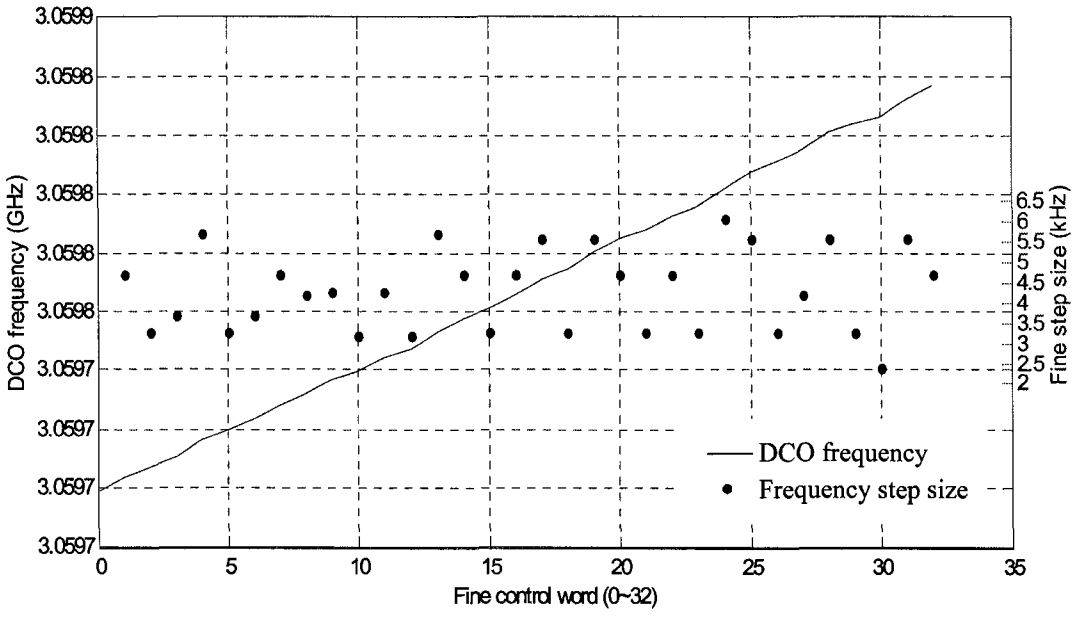


Figure 6.6 The measured fine tuning characteristic of the DCO

changes from 0 to 32. The change from 0 to 31 represents the actual fine bank tuning, and the change from 31 to 32 represents the case of transition between the fine/coarse banks (the fine bank changes from 31 to 0 while the coarse bank increases by one). The average step size is 4.5kHz, which increases with the increase of the band number.

Figure 6.7 shows the tuning characteristic of the combined fine/coarse tuning banks. 2048 frequency tuning levels are achieved in each band and the average frequency step size is calculated as 4.36kHz for band 0, and the differential non-linearity (DNL) is calculated as $\pm 0.5LSB$. The DNL is approximately the same for different bands, and the average frequency step size increases with the increase of the band number, resulting in an overall average frequency step of approximately 7.3kHz.

The measured DCO phase noise is shown in Figure 6.8. For the carrier frequency of 3.058GHz, the phase noise is measured as -118dBc/Hz at 1MHz frequency offset.

The ideal analog gain of the DCO is zero to prevent the analog noise in the control

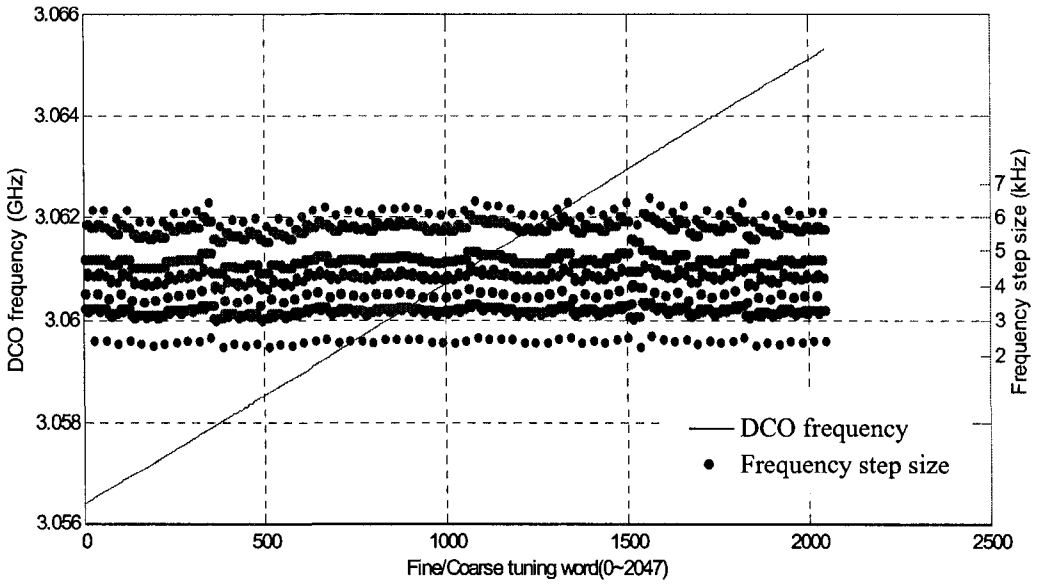


Figure 6.7 The measured fine/coarse tuning characteristic of the DCO

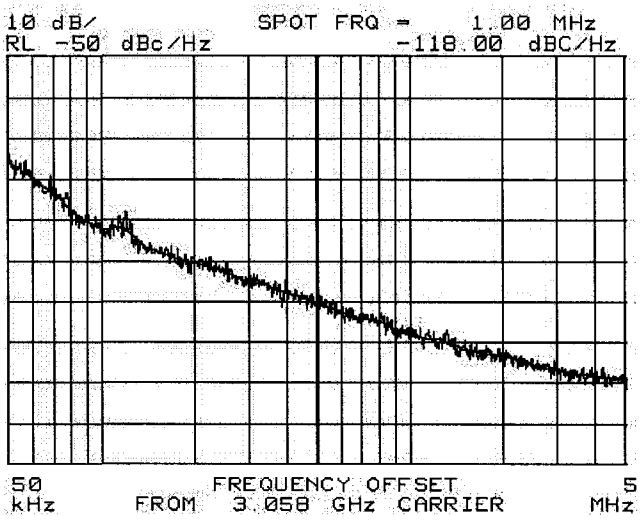


Figure 6.8 The measured phase noise of the DCO

line from being up converted to the DCO output. The analog gain was measured in this work by varying the voltage bias (VH) of the capacitance tuning circuit with all varactors biased to the high voltage level (the lowest output frequency point). The results are plotted in Figure 6.8, which shows the analog gains are no larger than 20MHz/V in the region from 1.35V~ 1.5V.

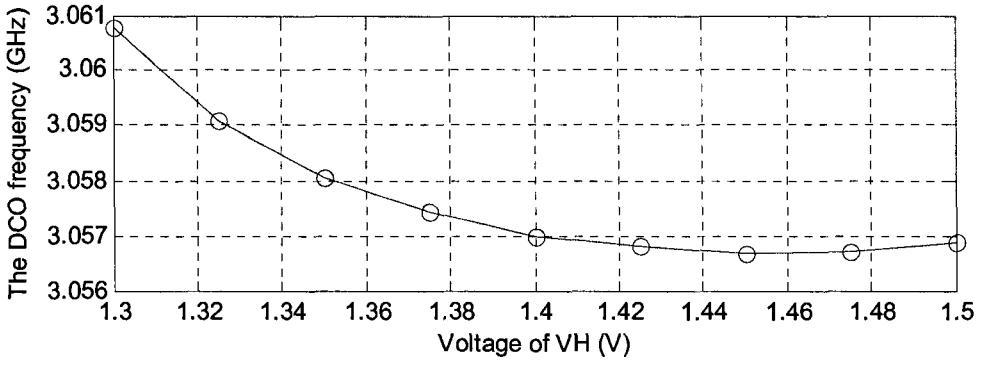


Figure 6.9 The measured analog gain of the DCO

Performance comparison with several recently reported DCOs is given in Table 6.1. The finest frequency resolution and the largest number of linear frequency tuning levels to date is achieved in this work with a comparable phase noise performance and a small current consumption.

TABLE 6.1: Performance comparison with recently reported DCOs

Reference No.	Tech.	Frequency range	Freq. resolution	Number of linear steps	Phase noise (dBc/Hz)	Core current (mA)
[10]	130nm CMOS	2.4GHz ±23%	23kHz[b]	64	-112@500kHz	2.3
[11][a]	0.13um CMOS	1.634GHz~1.766GHz	150kHz	32	-109@1MHz	0.37
[27]	90nm CMOS	3.2GHz~4GHz	24kHz[b]	64	-165[c]@20MHz	18
[28]	130nm CMOS	3.45GHz~4.45GHz	200kHz	1024	-118[d]@1MHz	2.3
[29]	65nm CMOS	9.87GHz~10.92GHz	1.03MHz	1024	-102@1MHz	3.0
This Work	90nm CMOS	3.05GHz~3.65GHz	5kHz	2048	-118@1MHz	2.0

a.The integrated one; b. Physical frequency resolution only; c. After the output is divided by 4. d. Phase noise measured at 2GHz

6.4 The test bench for the DPLL measurement

The proposed DPLL-based frequency synthesizer was fabricated in ST 90nm CMOS technology and the measurements were done to confirm its feasibility. Figure 6.10

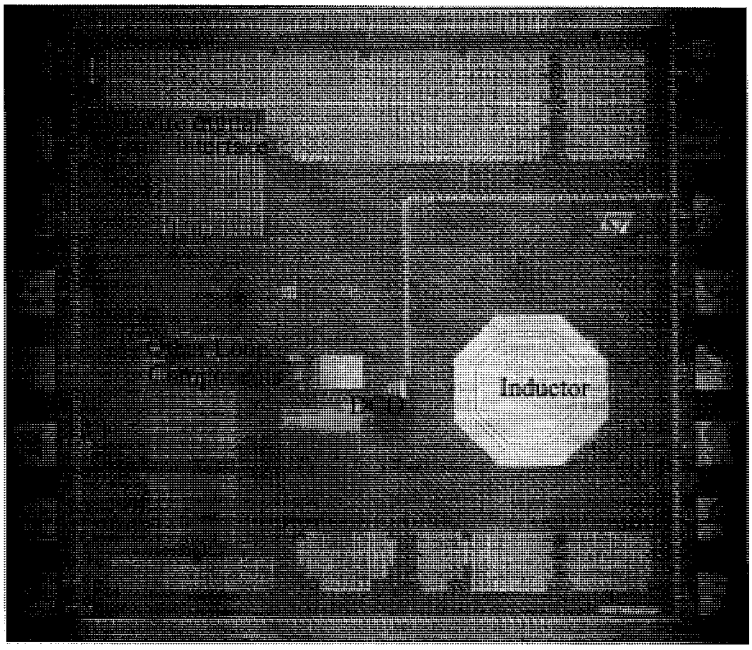


Figure 6.10 The chip micrograph of the proposed DPLL

shows the chip micrograph of the DPLL test chip. A total of 16 pads are employed and their arrangements are based on the available multi-contact probes. Besides the six signal pads, there are six ground pads and four DC power supply pads. A custom three-wire serial interface is integrated together with the digital PLL to enable the programmability. The signal arrangement is shown in Figure 6.11. *Fref* is the reference signal, while the

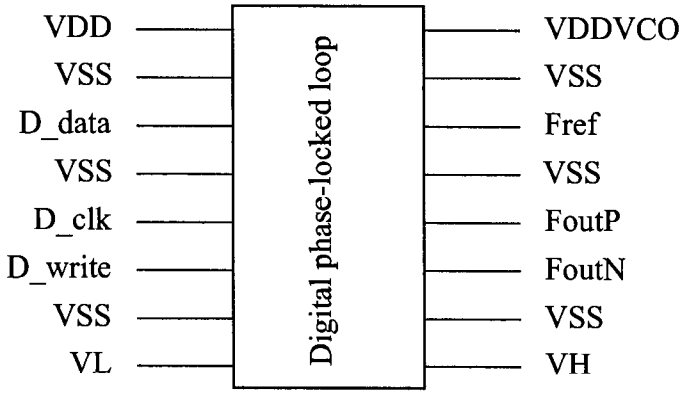


Figure 6.11 Top-level inputs/outputs of the chip

FoutN/FoutP are the output signals. The three signals for the three-wire interface are

D_data, *D_clk* and *D_write*. Similar to the DCO measurement, the serial inputs were provided by running a graphic software interface in a PC. This software interface provides options to change all parameters of the DPLL and automatically sends the selected parameters to the test chip. The screen-shot of the software interface is shown in Figure 6.12.

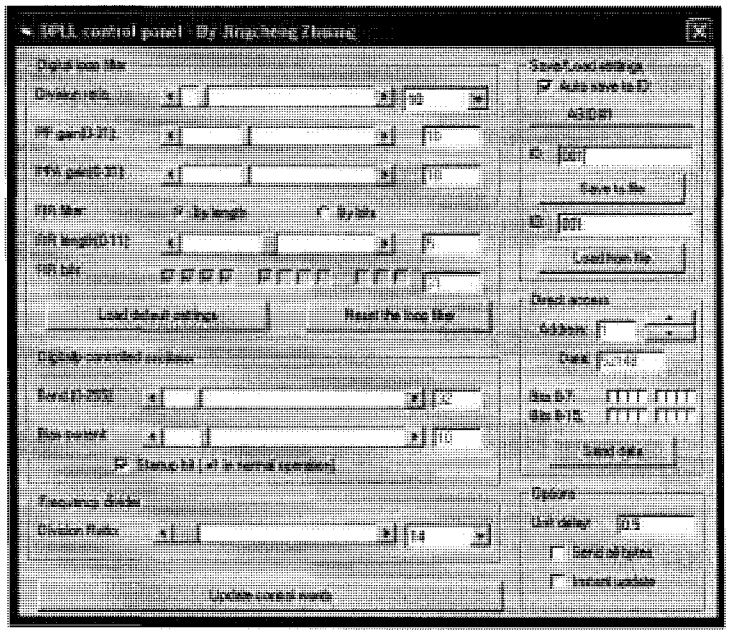


Figure 6.12 The screen-shot of the control panel for the DPLL measurement

The DPLL test bench is almost the same as the DCO test bench, except for that a reference signal is needed for the DPLL measurement. As the DPLL test bench shown in Figure 6.13, the reference signal is generated by an RF signal generator (Rohde & Schwarz Signal Generator SME06) with a frequency range of from 5kHz to 6.06Ghz. A DC bias circuit is used to provide a DC level of VDD/2, the required level for the DPLL. The test setup for the DPLL measurement is shown in Figure 6.14.

6.5 The measurement results of the DPLL

The digital PLL test chip was measured with the test bench described above to examine the phase noise performance and the spur performance. Ideally, the free running DCO in the digital PLL should have the same performance, including the frequency tun-

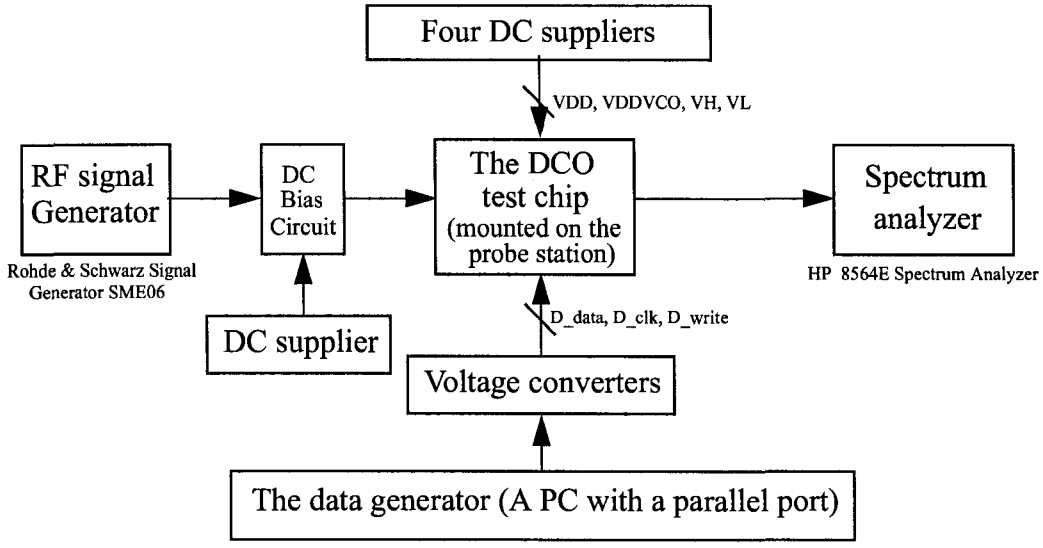


Figure 6.13 The test bench for the DPLL

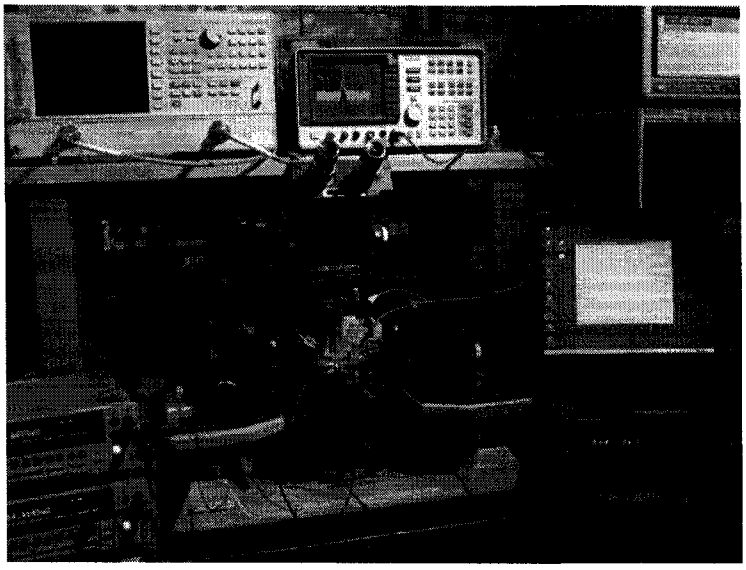


Figure 6.14 The photo of the test setup of the DPLL measurement

ing characteristic and the phase noise performance, as the DCO test chip given earlier in this chapter. However, because of mis-function of the DCO bias tuning/control circuitry, the DCO in the digital PLL exhibits higher free running phase noise than the DCO test chip does. Fortunately, the DCO phase noise attenuation and reference phase noise tracking behaviors of the proposed DPLL can still be clearly observed and they agree well with both the theoretical analysis and the behavioral simulation.

6.5.1 The phase noise performance

The phase noise of the free-running DCO, the reference signal and the in-lock DCO, were measured to observe the phase noise tracking and attenuation of the DPLL. With the multiplication ratio of 27 and a reference clock of 145MHz from the RF signal generator, the output frequency of the PLL is 3.916GHz. Figure 6.15 shows the measured

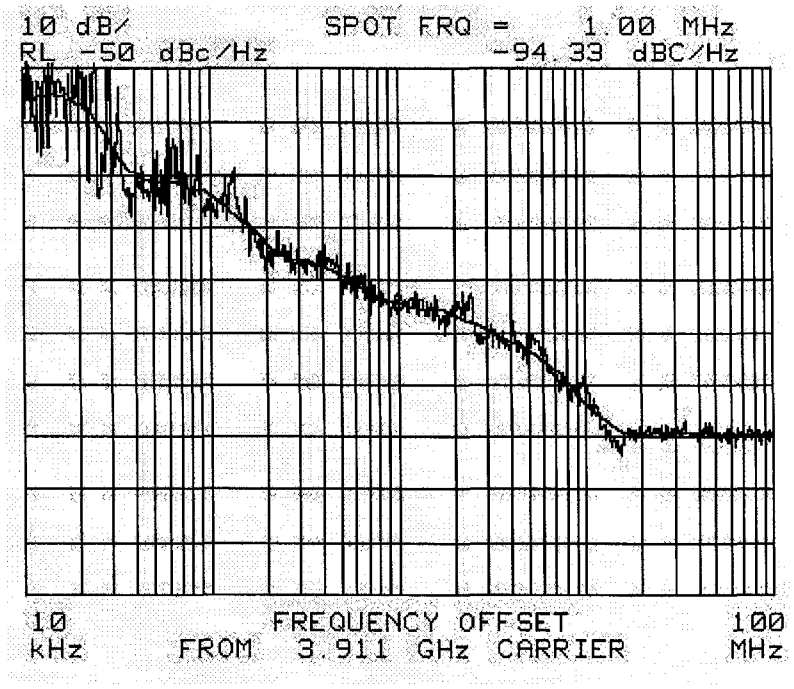


Figure 6.15 The measured open loop phase noise of the digital PLL

open-loop phase noise at the center frequency of 3.911GHz, and the measured phase noise is -94.33dBc/Hz at 1MHz frequency offset, which is much worse than the phase noise performance (Figure 6.8) of the DCO test chip. The measured reference phase noise at the frequency of 145.6MHz is shown in Figure 6.16. The reference source exhibits a phase noise of -133.17dBc/Hz at 1MHz frequency offset.

With the loop settings listed in Table 6.2., the closed loop phase noise of the PLL

TABLE 6.2: The loop parameters

parameters	Values
Proportional path frequency step	+/-340kHz

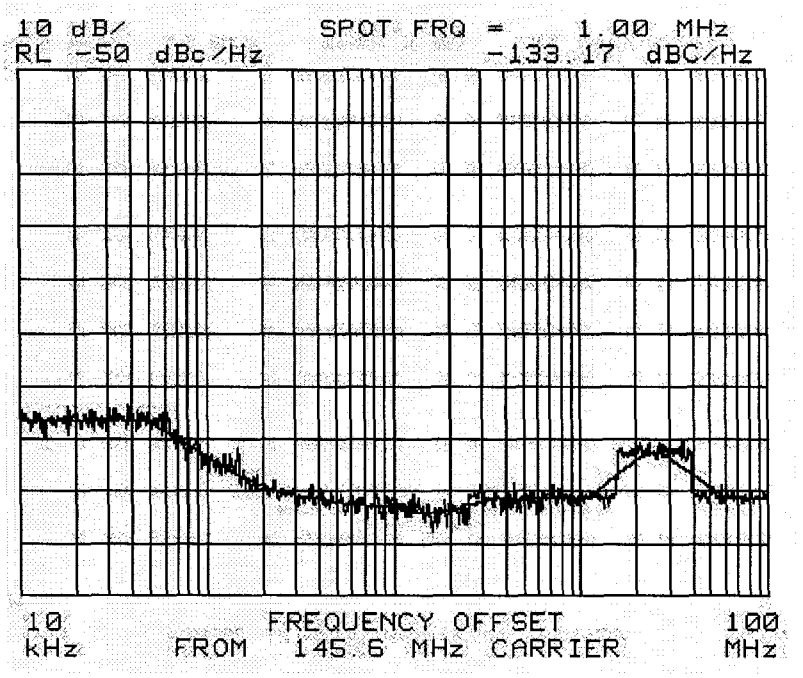


Figure 6.16 The measured reference phase noise

TABLE 6.2: The loop parameters

parameters	Values
integral path frequency step	5kHz
Frequency decision period	100*Tref

output was measured as shown in Figure 6.17, in which the reference phase noise and the DCO free-running phase noise are also illustrated for convenience. The PLL exhibits an effective loop bandwidth of approximately 5MHz, and any phase noise within 5MHz frequency offset is corrected by the PLL based on the reference clock. The theoretical in-lock phase noise at the frequency offset of 1MHz is approximately -105dBc/Hz (-133+20log(27)) in an ideal phase tracking condition, which agrees with the measured result of -106dBc/Hz and confirms the effective noise tracking and attenuation of the proposed low-complexity digital PLL.

By reducing the reference frequency to 134MHz, the output frequency is reduced to 3.63MHz, and the measured frequency spectrum of the closed-loop DPLL is shown in Figure 6.18. Similarly, the phase noise of the reference source as well as the phase noise of

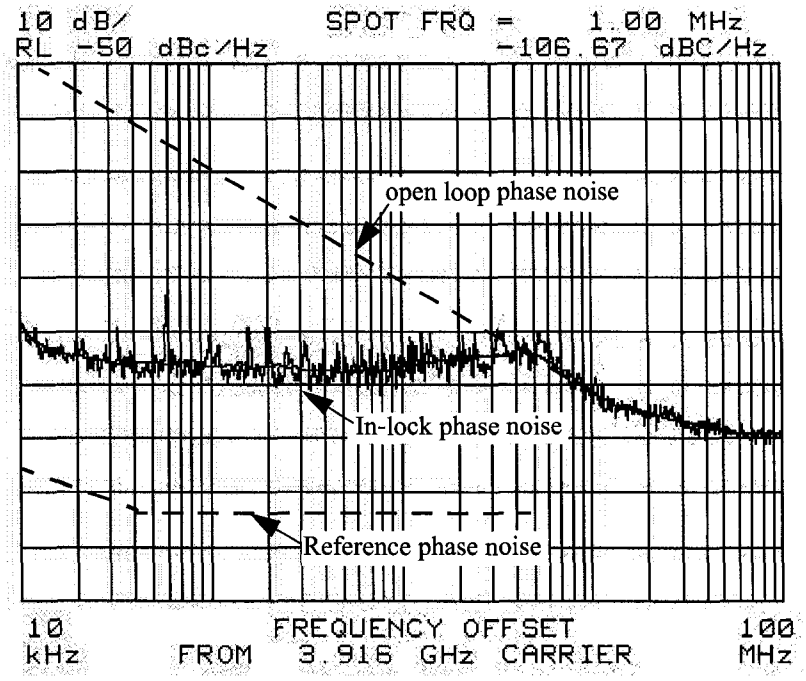


Figure 6.17 The measured phase noise of the low-complexity digital PLL

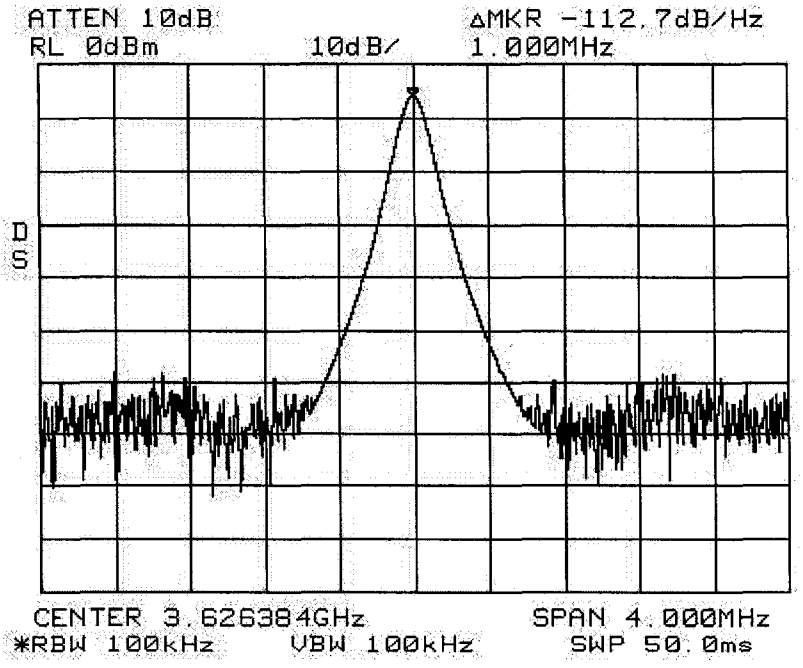


Figure 6.18 The measured output frequency spectrum of closed-loop DPLL

the open/closed loop DPLL are measured at this frequency (and other frequencies) and the similar noise attenuation and tracking behaviors were observed. Those measured results are eliminated here to avoid redundancy.

6.5.2 *The spur performance*

Another important performance of a PLL, the spur performance was measured by

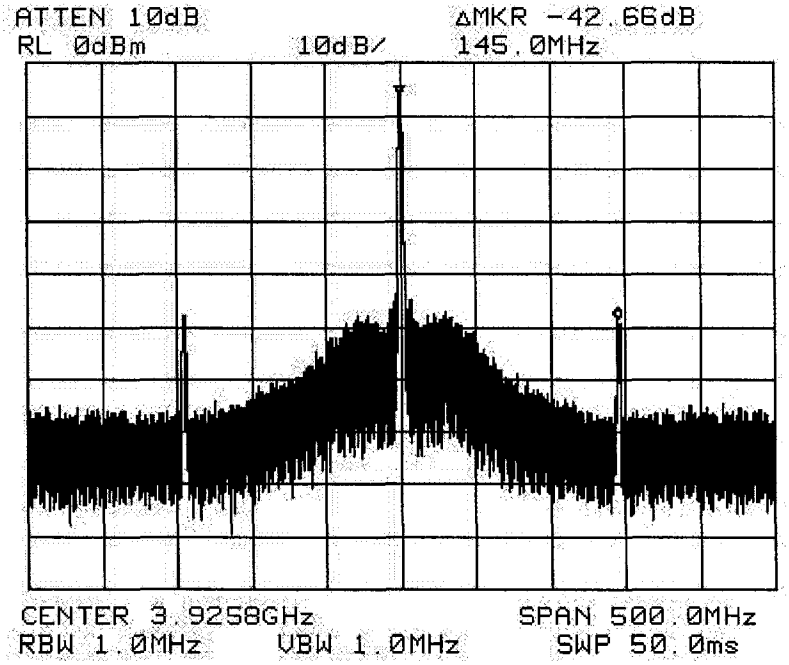


Figure 6.19 The measured reference spur performance of the closed-loop DPLL

capturing the output spectrum with a wide span as shown in Figure 6.19. Although no reference spur was expected from the theoretical analysis and the behavioral simulation in chapter 4, a reference spur of approximately -43dB at 145MHz frequency offset was observed from the measurement.

To investigate the sources of the reference spur, several designs were reviewed and the spur performances were compared. In the reduced in-locked DLL frequency synthesizer [30], a -46.17dB reference spur was measured, and in the DLL frequency multiplier with period error compensation[31], a -46.5dB reference spur was observed. A digital CDR test chip was tested by supplying two clock signals with different frequencies (3GHz and 150MHz respectively). In the CDR test chip, two clock signals go through the test chip by two separate paths and two separate output buffers without any deliberate cou-

pling between them. The power spectrum of the signal with frequency of 3GHz was measured as shown in Figure 6.20, in which a spur level of -45dB at 150MHz frequency offset was observed.

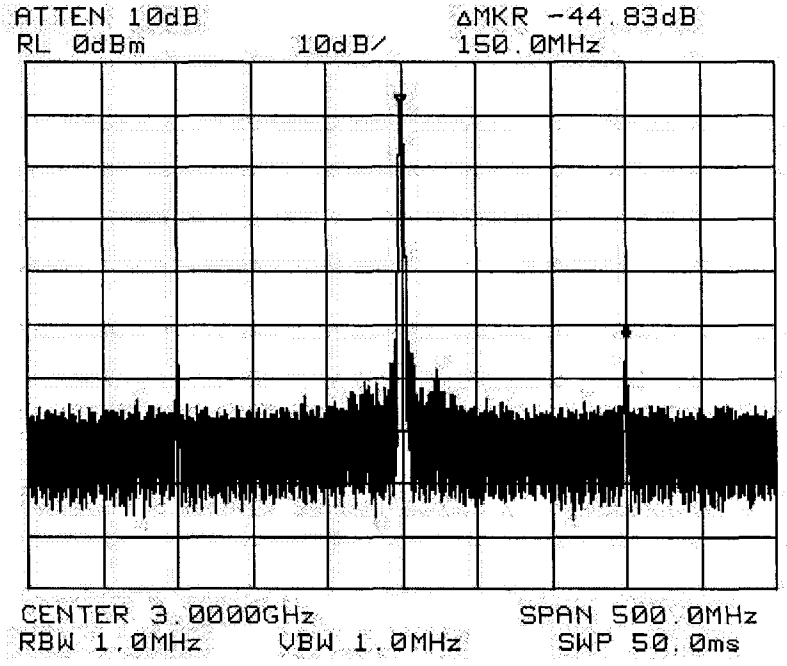


Figure 6.20 The measured spur from the digital CDR test chip

Approximately the same spur levels ($-44 \pm 2dB$) were observed in three different designs listed above as well as the digital PLL in this work, and it is reasonable to believe that those spurs are dominated by a common factor among those designs, and it is that different parts of the circuit in each chip share the same power supply. Consequently, those spurs are dominated by the strong AC coupling through the power supply and the ground in those designs and they can be reduced by employing different power supplies (or using separate voltage regulators) in different parts of the circuit to reduce the strength of the AC coupling.

6.6 The phase noise of the DPLL with the low-noise DCO

By modeling the measured phase noise of the reference source (Figure 6.16) and of the free running DCO (Figure 6.15), and by modeling the DPLL with the parameters

listed in Table 6.2 (which were determined according to the analysis in Chapter 4), the closed loop phase noise can be simulated using event-driven technique described in "Event-Driven simulation technique" on page 79. In fact, this simulation case was already included in "Phase noise performance simulation" on page 91 and the simulated result was given in Figure 4.36 on page 95, in which the simulated closed-loop phase noise performance is almost identical to the measurement (Figure 6.17 on page 136), confirming the accuracy of the theoretical analysis and the behavioral simulation, as well as the feasibility of the proposed digital PLL. Consequently, the phase noise performance of the DPLL, if the low noise DCO (the one in the DCO test chip) was included, can be predicted using the same simulation technique.

By assuming the DCO performance is the same as the DCO test chip with the phase noise shown in Figure 6.8 on page 129, a phase noise simulation was done. The loop parameters were determined by the DCO phase noise as listed in Table 6.3. (refer to

TABLE 6.3: The loop parameters for the DPLL with low noise DCO

parameters	Values
Proportional path frequency step	+/-19kHz
integral path frequency step	4.5kHz
Frequency decision period	100*Tref
The reference frequency	113MHz
The multiplication ratio	27
The output frequency	3.051GHz

"The determination of the design parameters" on page 68).

Assuming the reference frequency is 113MHz with a flat phase noise at -145dBc/Hz and the multiplication ratio is 27, the phase noise was simulated as shown in Figure 6.21. The DPLL exhibits an effective loop bandwidth of 1MHz and the DCO phase noise is attenuated effectively and the in-band phase noise is approximately 29dB $(20 \cdot \log(27))$ higher than the reference phase noise as expected.

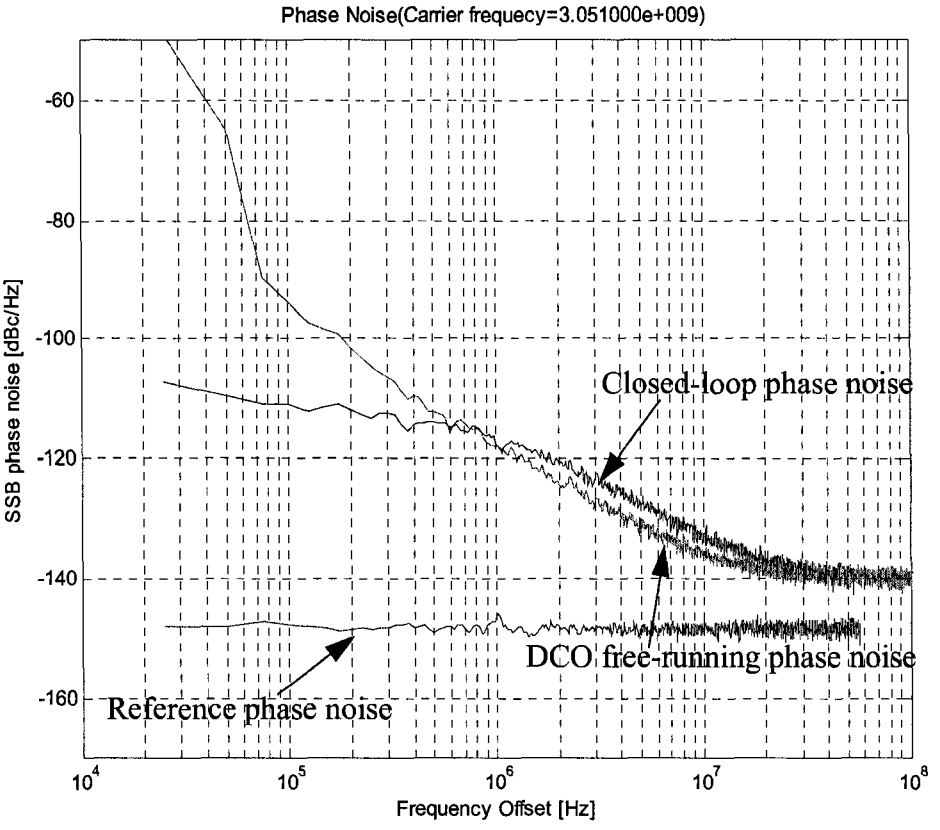


Figure 6.21 The phase noise of the DPLL with the low noise DCO

6.7 Conclusions

The performance evaluation of the two test chips, the DCO and the DPLL, was described in this chapter. The measurement of the DCO shows a frequency tuning range from 3.05GHz to 3.65GHz. with a monotonic frequency tuning characteristic of the fine bank and the transition between the fine and coarse banks, confirming the feasibility of the frequency resolution enhancement technique for the DCO. The phase noise was measured to be -118dBc at 1MHz frequency offset with the carrier frequency of 3.06GHz.

For the DPLL, the noise tracking and attenuation behaviors were clearly observed from the measurement by measuring the phase noise of the reference signal, the open-loop DCO and the closed loop PLL, which agree well with the theoretical analysis and the behavioral simulation. The reference spurs were measured and investigated followed by

the predicted phase noise performance of the DPLL with the low-noise DCO (the one in the DCO test chip) using the theoretical analysis and the event-driven simulation technique described in Chapter 4.

CHAPTER 7 *Conclusions*

7.1 Summary

This dissertation first briefly reviewed some concepts of the PLL-based frequency synthesizers, followed by the review of all-digital PLLs. The architecture of a low-complexity digital PLL, consisting of a non-linear digital PFD, phase/frequency decision circuit, a digitally-controlled oscillator with an enhanced frequency resolution and the frequency divider, was proposed for the integer-N frequency synthesis applications, where the noise performance in the locked condition is of concern.

The proposed PLL was modeled using Simulink and the locking behavior was confirmed by the modeling simulation. The loop behavior, including the phase noise performance and the loop stability, was analyzed theoretically, followed by the determination of the loop parameters.

To confirm the system analysis, which requires fast and accurate simulation in the system level, an event-driven modeling and simulation technique was implemented in Matlab, and some event-driven simulations, such as jitter tracking, jitter transfer, and the closed loop phase noise performance were done to confirm the operation of the proposed digital PLL in more details and to show the efficiency of the event-driven simulation technique.

Transistor-level verification was performed by implementing the DPLL and simu-

lating with Spectre in transistor-level. The measurements, taken from two test chips, further verified the feasibility of the proposed DPLL and the theoretical analysis.

References

1. L. Xiu, W. Li, J. Meiners, R. Padakanti, "A novel all-Digital PLL with software adaptive filter", IEEE Journal of Solid-State Circuits, Vol. 39, No. 3, March 2004, pp: 476~483.
2. A. M. Fahim, "A compact, low-power low-jitter digital PLL", The 29th European Solid-State Circuits Conference, 2003. pp: 101 - 104.
3. C. C. Chung, C. Y. Lee, "An all-digital phase-locked loop for high-speed clock generation", IEEE Journal of Solid-State Circuits, vol. 38. No. 2, Feb. 2003, pp: 347 - 351.
4. J. S. Chiang, K. Y. Chen, "The design of an all-digital phase-locked loop with small DCO hardware and fast phase lock", IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, Vol. 46, Issue 7, July 1999, pp: 945 - 950.
5. J. Dunning, G. Garcia, J. Lundberg, Ed Nuckolls, "An all-digital phase-locked loop with 50 cycle lock time suitable for high-performance microprocessors", IEEE Journal of Solid-State Circuits, Vol. 40, No. 4, April 1995, pp: 412 - 422.
6. S. J. Jou, Y. L. Tsao, I. Y. Yang, "An all digital phase-locked loop with modified binary search of frequency acquisition", IEEE International Conference on Electronics, Circuits and Systems, 1998, Vol. 2, pp: 195 - 198.
7. T. Watanabe, S. Yamauchi, "An all-digital PLL for frequency multiplication by 4 to 1022 with seven cycle lock time", IEEE Journal of Solid-State Circuits, Vol. 38, No. 2, February 2003, pp: 198 - 204.
8. T. Olsson, P. Nilsson, "An all-digital PLL clock multiplier", Proceedings of IEEE Asia-pacific Conference on ASIC 2002. pp: 275 - 278.

9. S. M. Rezaul Hasan, "A 5GHz CMOS digitally controlled oscillator with a 3GHz tuning range for PLL applications", IEEE International Conference on Electronics, Circuits and Systems, 2003. Vol 1, pp: 208-211.

10. R. B. Staszewski, D. Leipold and P. T. Balsara, "A first multi gigahertz digitally controlled oscillator for wireless applications", IEEE Transactions on Microwave theory and techniques, vol. 51, Nov. 2005, pp: 2154 - 2164.

11. N. M. Pletcher and J. M. Rabacy, "a 100μw, 1.8GHz oscillator with fully digital frequency tuning", Proceeding of ESSCIRC 2005, pp: 387 - 390.

12. J. Lin, B. Haroun, T. Foo, J. S. Wang, B. Helmick, S. Randall, T. Mayhugh, C. Barr, J. Kirkpatrick, "A PVT tolerant 0.18MHz to 600MHz self-calibrated digital PLL in 90nm CMOS process", Digest of Technical papers, IEEE International Solid-State Circuits Conference 2004, pp: 488 - 541.

13. R. B. Staszewski, D. Leipold, K. Muhammad and P.T. Balsara, "Digitally Controlled Oscillator (DCO)-Based Architecture for RF Frequency Synthesis in a Deep-Submicrometer CMOS Process", IEEE Transactions on Circuits and System-II, Vol. 50, No 11. November 2003. pp: 815 - 828.

14. R. B. Staszewski and P. T. Balsara, "Phase-domain all-digital phase-locked loop", IEEE Transactions on Circuits and Systems-II: Express briefs, vol. 52, No. 3, March 2005. pp: 195 - 163.

15. G. Donzellini, D. D. Caviglia, G. Parodi, D. Ponta, and P. Repetto, "A digital controlled oscillator based on controlled phase shifting", IEEE Transactions on Circuits and Systems, Vol 36, August 1989, pp: 1101 - 1105.

16. S. R. Abdollahi, S. Kiaei, B. Bakkaloglu, S. M. fakhraie, R. Anvari and S.E. Abdollahi, "An all-digital programmable digitally-controlled-oscillator (DCO) for digital wireless applications", IEEE International Symposium on Circuits and Systems, 2002. pp: IV-101-IV-104.

17. Martin Saint-Laurent, Gabriel Patrick Muyshondt, "A digitally controlled oscillator constructed using adjustable resisters", Southwest Symposium on Mixed-Signal Design, 2001, pp: 80 -82.

18. D. E. Calbaza, I. Cordos, N. Seth-Smith and Y. Savaria, "An ADPLL circuit using a DDPS for genlock applications", International Symposium on Circuits and Systems. 2004, Vol 4. pp: IV-569-72.

19. N. D. Dalt, E. Thaller, P. Gregorius, L. Gazsi, "A low jitter triple-band digital LC PLL in 130nm CMOS", The 30th European Solid-State Circuit Conference, 2004, pp: 371 - 374.

20. P. L. Chen, C. C Chung, and C. Y. Lee, "A portable digitally controlled oscillator using novel varactors", IEEE Transactions on Circuits and Systems-II, Vol 52. No. 5, May 2005, pp: 233 - 237.

21. J. J. Jong, C. Y. Lee, "A novel structure for portable digitally controlled oscillator", IEEE International Symposium on Circuits and Systems 2001. vol. 1, pp: 272-275.

22. R. B. Staszewski, D. Leipold, C. M. Hung, P.T. Balsara, "TDC-based frequency synthesizer for wireless applications", IEEE Radio Frequency Integrated Circuits Symposium, 2004, pp 215 - 218.

23. T. C. Weigandt, B. Kim, P. R. Gray, "Analysis of timing jitter in CMOS ring oscillators," Proc. of IEEE Symp. of Circuits and Systems, vol. 4, pp: 27-30, June 1994

24. R. B. Staszewski, C. Fernando, P. T. Balsara, "Event-driven Simulation and modeling of phase noise of an RF oscillator", IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, Vol. 52, Issue 4, April 2005, pp: 723 - 733.

25. C. F. Huang, C. C. Wu, C. H. Chen, C.C. Ho, Y. J. Chan, C. S. Chang, C. P. Chao and G. J. Chern, "A comprehensive varactor study for advance CMOS RFIC design", 2005 IEEE VLSI-TSA International Symposium on 25-27 April 2005 pp: 66 - 67.

26. T. Okayasu, M. Suda, K. Yamamoto, S. Kantake, S. Sudou, D. Watanabe, "1.83ps-resolution CMOS dynamic arbitrary timing generator for >4GHz ATE application", 2006 IEEE International Conference, Feb. 2006, pp: 2122-2131.

27. R. B. Staszewski, C. M. Hung, N. Barton, M. C. Lee, D. Leipold, "A digital controlled oscillator in a 90nm digital CMOS process for mobile phones", IEEE. Journal of Solid-State Circuits, Vol. 40. Nov. 2005, pp 2203-2211.

28. T. Pittorino, Y. Chen, V. Neubauer, T. Mayer, L. Maurer, "A UMTS-Compliant Fully Digitally Controlled Oscillator with 100Mhz Fine-Tuning Range in 0.13um CMOS", Digest of Technical papers, IEEE International Solid-State Circuits Conference 2006, pp: 770-779.

29. N. D. Dalt, C. Kropf, M. Burian, T. Hartig, H. Eul. "A 10b 10GHz Digitally Controlled LC Oscillator in 65nm CMOS", Digest of Technical papers, IEEE International Solid-State Circuits Conference 2006, pp: 669 - 678.

30. J. Zhuang, Q. Du, T. Kwasniewski, "A -107dBc, 10kHz carrier offset 2-GHz DLL-based frequency synthesizer", Proc. of IEEE Custom Integrated Circuits Conference 2003.

31. Q. Du, J. Zhuang, T. Kwasniewski, "A Low-Phase Noise, Anti-Harmonic Programmable DLL Frequency Multiplier With Period Error Compensation for Spur Reduction", IEEE Transaction on Circuits and System II, Vol 53, Issue 11, Nov, 2006, pp: 1205-1209.

32. J. Zhuang, Q. Du, T. Kwasniewski, "A 4GHz Low Complexity ADPLL-based Frequency Synthesizer in 90nm CMOS", Proc. of IEEE Custom Integrated Circuits Circuits conference, 2007.

33. J. Zhuang, Q. Du, T. Kwasniewski, "A 3.3 GHz LC-Based Digitally Controlled Oscillator with 5kHz Frequency Resolution", Proc. of IEEE Asian Solid-State Circuits Conference, 2007.