In compliance with the
Canadian Privacy Legislation
some supporting forms
may have been removed from
this dissertation.

While these forms may be included
in the document page count,
their removal does not represent
any loss of content from the dissertation.

A DLL-Based Frequency Synthesizer

By

Qingjin Du

A thesis submitted to the Faculty of Graduate Studies and Research

in partial fulfillment of the requirements for the degree of

Master of Applied Science

Department of Electronics

Carleton University

Ottawa, Ontario, Canada

June 2003

© 2003 Qingjin Du

National Library
of Canada

Bibliothèque nationale
du Canada

Acquisitions and
Bibliographic Services

Acquisitons et
services bibliographiques

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

ISBN: 0-612-88870-3

Our file Notre référence

ISBN: 0-612-88870-3

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

Canada

The information used in this thesis comes in part from the research program of Dr. Tad Kwsniewski 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 research 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 thesis presents a theoretical analysis, the design and a monolithic implementation in submicron CMOS technology of a DLL-based frequency synthesizer. A comprehensive review shows that the conventional as well as one of the most popular techniques employed in the synthesizers is PLL based. The first goal was to evaluate any significant differences in performance of the two device classes. To this end, mathematical models of two ‘equivalent’ loop circuits, one from each class, were developed, analyzed and simulated using Matlab software. The simulation results were consistent with theoretical predictions that given same delay stage structures employed in the voltage controlled ring oscillator in the PLL and the voltage controlled delay line in the DLL, the DLL has better phase noise performance than the PLL. A DLL-based frequency synthesizer is thus built in CMOS 0.18 μm technology. New circuit ideas were tried in this process, ideas for circuits showing reduced output phase noise, for improving the performances of the VCDL, the lock-detector and the edge-combiner. The device was measured with a phase noise of -105.5dBc/Hz at 10kHz offset with the carrier frequency of 2.07GHz, which once again supports the theoretical results, as does the model.

Acknowledgements

I would like to thank my supervisor Dr. Tad Kwasniewski for his overall support, encouragement, patience and guidance during the development of this research as well as my course work in this program.

I would also like to thank my husband Jingcheng Zhuang for his support, encouragement in my life, suggestion for my research, and my parents, for their emotional support.

Table of Contents

Chapter 1: Introduction.....	1
1.1 Thesis motivation.....	1
1.2 Contributions	3
1.3 Thesis outline	3
Chapter 2: Overview of Frequency Synthesis.....	5
2.1 Techniques of frequency synthesis	5
2.2 Direct frequency synthesis.....	5
2.2.1 Table look-up synthesis.....	5
2.2.2 Analog direct frequency synthesis	8
2.3 Indirect phase-locked loop frequency synthesis	9
2.3.1 PLL operation principles	9
2.3.2 Integer N division PLL frequency synthesis.....	12
2.4 Fractional-N division frequency synthesis	13
2.5 Hybrid techniques of frequency synthesis	17
2.5.1 Dual-loop PLL frequency synthesis	17
2.5.2 Direct digital frequency synthesis (DDFS).....	18
2.6 Characteristics of PLL clock and frequency synthesis	19
2.6.1 Considerations in Integer-N PLL FS design.....	20
2.6.2 Phase noise.....	20
2.6.3 Timing jitter	22
2.6.4 Period jitter	22
2.7 Summary	23
Chapter 3: Comparison of PLL and DLL systems.....	24

3.1	Introduction.....	24
3.2	Operating principles of PLL and DLL.....	24
3.3	Applications of PLL and DLL	25
3.4	Phase noise in PLL/DLL based frequency synthesizers	26
3.4.1	Phase noise in the VCO of PLLs	26
3.4.2	Phase noise in the VCDL of DLLs	29
3.5	PLL and DLL system output timing jitter	30
3.6	PLL and DLL's ability to suppress phase noise and jitter.....	31
3.6.1	PLL-based system noise performance analysis	31
3.6.2	DLL-based system noise performance analysis.....	35
3.6.3	Noise powers at the outputs of PLL and DLL	37
3.7	Behavior simulation of PLL and DLL	40
3.7.1	PLL model and simulation.....	40
3.7.2	DLL model and simulation	43
3.8	Loop components requirements of PLL and DLL-based frequency synthesizers.....	45
3.9	Summary	46
Chapter 4: Proposed Architecture of the Frequency Synthesizer		47
4.1	Introduction.....	47
4.2	Survey of relevant publications of DLL-based frequency synthesizers	47
4.3	System design of the proposed architecture	48
4.4	Behavioral simulation of DLL-based frequency synthesizer with MatLab.....	49
4.4.1	Phase detector model	50
4.4.2	Charge pump and loop filter models.....	51
4.4.3	Voltage-controlled delay line and pulse combiner models	52
4.4.4	Lock detector model	54

4.4.5	Top model of the frequency synthesizer	55
4.5	Matlab simulation results.....	56
4.6	Conclusion	60
Chapter 5: CMOS Design of the DLL-Based Frequency Synthesizer.....		61
5.1	Introduction.....	61
5.2	Techniques of phase detector	61
5.3	Charge pump and loop filter functions	68
5.3.1	Charge pump.....	68
5.3.2	Design considerations	71
5.4	Loop filter	73
5.5	Operation and design of voltage-controlled delay line	75
5.5.1	Capacitive model of CMOS transistors	75
5.5.2	Delay cell design.....	76
5.5.3	Control signal circuit (driver) design.....	80
5.5.4	Voltage-controlled delay line with drive circuit.....	82
5.6	Design of lock detection circuit	85
5.7	Edge combining circuits	88
5.8	Top level post-layout simulation.....	90
5.8.1	The locking process	91
5.8.2	Synthesizer output of the locked loop.....	94
5.9	Summary	95
Chapter 6: Test Setup and Performance Evaluation		96
6.1	Introduction.....	96
6.2	Layout considerations	96

6.3	Floorplan and package pinout.....	98
6.4	Testbench	99
6.5	Measured noise performance	102
6.6	Operating frequency range.....	106
6.7	Power consumption.....	107
6.8	Summary	108
Chapter 7: Conclusion.....		109
7.1	Summary	109
7.2	Suggestion for Further Research.....	110
Appendix A: Schematics of the Frequency Synthesizer		111
Appendix B: Layout of the Frequency Synthesizer		122
References.....		132

List of Tables

TABLE 4.1:	Summary of recent publications on the DLL-based frequency synthesizer	48
TABLE 4.2:	Parameters and specifications of Matlab simulation	58
TABLE 6.1:	Test equipment	102
TABLE 6.2:	Measured chip performance summary	108

List of Figures

Figure 1.1	Block diagram of a typical transceiver	2
Figure 2.1	Basic diagram of a standard DDS	6
Figure 2.2	Illustration of phase to amplitude conversion.....	7
Figure 2.3	Example of analog direct frequency synthesis	8
Figure 2.4	Basic structure of phase locked loop	9
Figure 2.5	Linear model of phase-locked loop	11
Figure 2.6	General block diagram of N division frequency synthesizer.....	13
Figure 2.7	General block diagram of fractional N PLL frequency synthesizer	14
Figure 2.8	An example of digital phase accumulator (DPA)	15
Figure 2.9	Quantization phase error diagram.....	16
Figure 2.10	Dual-loop PLL frequency synthesizer	17
Figure 2.11	Diagram of direct digital frequency synthesizer.....	18
Figure 2.12	Power spectrum for ideal output and practical output frequency	21
Figure 2.13	Principle of timing jitter measurement	22
Figure 2.14	Principle of Period jitter measurement	23
Figure 3.1	Operating principles of PLL and DLL.....	25
Figure 3.2	An n-stage ring oscillator.....	26
Figure 3.3	Effect of impules injected at transition edges and saturated peaks.....	28
Figure 3.4	An n-stage delay line	29
Figure 3.5	Accumulation of timing jitter over cycles	32
Figure 3.6	Jitter suppressed by a PLL	32
Figure 3.7	Simplified PLL model	33
Figure 3.8	Simplified PLL discrete-time model for phase noise analysis	34

Figure 3.9	Simplified DLL discrete-time model	35
Figure 3.10	Simplified DLL discrete-time model for phase noise analysis	36
Figure 3.11	PLL/DLL power ratio and transfer functions	39
Figure 3.12	VCO model for jitter simulation	41
Figure 3.13	PLL model for jitter simulation	42
Figure 3.14	Input and output jitter of PLL simulation	42
Figure 3.15	VCDL model for jitter simulation	43
Figure 3.16	DLL model for jitter simulation	44
Figure 3.17	Input and output jitter of DLL simulation	44
Figure 4.1	System diagram of the proposed frequency synthesizer	49
Figure 4.2	Model of the phase detector	50
Figure 4.3	Model of charge pump and loop filter	51
Figure 4.4	VCDL & pulse combiner models	53
Figure 4.5	Lock detector model	54
Figure 4.6	Top model of the frequency synthesizer	55
Figure 4.7	Test bench of the frequency synthesizer	56
Figure 4.8	Wave form of Vcontrol during the locking process	56
Figure 4.9	Input/output waveforms when the lock is not in lock	57
Figure 4.10	Input/Output when the loop is in lock	59
Figure 5.1	Characteristic of phase detector	62
Figure 5.2	Gilbert multiplier circuit	63
Figure 5.3	Characteristic of differential pair	63
Figure 5.4	Dual flip-flop phase detector	65
Figure 5.5	State diagram for dual DFF phase detector	65
Figure 5.6	Diagram of dead zone example	66

Figure 5.7	State diagram for PD without dead-zone	68
Figure 5.8	Operation principle of phase detector	69
Figure 5.9	Post-layout simulation for phase difference vs. width difference between UP & DN.....	69
Figure 5.10	Simple diagram of charge pump and loop filter	70
Figure 5.11	Typical current mirrors	71
Figure 5.12	Schematic of charge pump	72
Figure 5.13	Post layout simulation for current pump up.....	74
Figure 5.14	Post layout simulation for current pump down.....	74
Figure 5.15	MOSFET parasitic capacitances	75
Figure 5.16	Schematic of typical delay cells.....	77
Figure 5.17	Behavior of a simple inverter.....	78
Figure 5.18	Schematic of the delay stage.....	79
Figure 5.19	Post layout simulation for delay cell.....	80
Figure 5.20	Schematic of driving circuit.....	81
Figure 5.21	Post layout simulation for drive circuits	81
Figure 5.22	Schematic of voltage controlled delay line.....	83
Figure 5.23	VCDL post-layout simulation.....	84
Figure 5.24	In-lock timing relationship.....	86
Figure 5.25	Schematic of lock detection circuit.....	87
Figure 5.26	Schematic of sample & hold.....	87
Figure 5.27	Post layout simulation for operation of lock-detecting circuit	88
Figure 5.28	VCDL pulse generation	89
Figure 5.29	Pulse-generation circuit for combining.....	89
Figure 5.30	Pulse combining circuit	90

Figure 5.31	Post-layout simulation of the edge combiner	91
Figure 5.32	Schematic of top level synthesizer.....	92
Figure 5.33	Post-layout locking process with initial $V_{control} = 0$	93
Figure 5.34	Post-layout locking process with initial $V_{control}=V_{dd}$	93
Figure 5.35	Time-domain reference signal/output signal.....	94
Figure 5.36	In-lock timing error between VCDL input/output.....	94
Figure 5.37	Post-layout simulation of the output period jitter	95
Figure 6.1	Structure of the capacitor layout	97
Figure 6.2	Floorplan of the frequency synthesizer.....	98
Figure 6.3	Pad arrangement	99
Figure 6.4	Bonding diagram of the frequency synthesizer	100
Figure 6.5	PCB connection illustration diagram.....	101
Figure 6.6	Top level test setup.....	101
Figure 6.7	Measured phase noise of the reference signal and the output signal	103
Figure 6.8	Measured phase noise at carrier frequency of 1.71GHz.....	103
Figure 6.9	Measured phase noise at carrier frequency of 2.07GHz.....	104
Figure 6.10	Output phase noise at output frequency of 2.16GHz.....	104
Figure 6.11	Output spectrum at output frequencies of 1.386 and 1.575GHz	105
Figure 6.12	Output waveform with input frequency of 50 MHz and 100 MHz	105
Figure 6.13	Operating frequency range.....	106
Figure 6.14	Consumption current vs. the reference frequency	107

List of Symbols

Antenna
Mixer
Crystal Oscillator
Oscillator
Inverter
Sum
Unit Delay
Low pass filter
Output buffer
Capacitor
Ground
NAND logic gate
logical AND gate
Inverter
Logival NOR gate
Delay cell

TimeScope

Resistor

NMOS transistor

PMOS transistor

Current source

VDD

2-input NAND gate

4-input NOR gate

D flip-flop

2-input AND gate

Switch

List of Abbreviations

AND	Logical AND gate
BW	Band Width
C	Capacitor
CMC	Canadian Microelectronics Corporation
CMOS	Complementary MOS
CP	Charge pump
DAC	Digital to Analog Converter
DC	Direct Current
DLL	Delay-Locked Loop
DPA	Digital Phase Accumulator
IC	Integrated Circuit
LD	Lock Detector
LP	Loop Filter
LPF	Low Pass Filter
MOS	Metal Oxide Semiconductor
MOSFET	Metal Oxide Semiconductor Field Effect Transistor
NAND	Logical NAND gate
NMOS	Negative Metal Oxide Semiconductor
NOT	Inverter
NOR	Logical NOR gate
OR	Logical OR gate

PCB	Printed Circuit Board
PD	Phase Detector
PFD	Phase Frequency Detector
PLL	Phase-Locked Loop
PMOS	Positive Metal Oxide Semiconductor
R	Resistor
RF	Radio Frequency
RMS	Root Mean Square
SSB	Single Side Band
VCDL	Voltage-Controlled Delay Line
VCO	Voltage-Controlled Oscillator
VLSI	Very Large Scale Integrated circuits
VTC	Voltage Transform Characteristic
XOR	Exclusive OR gate

1.1 Thesis motivation

Wireless telecommunication industry has been experiencing rapid growth recently as the demand for telecommunication systems explodes and the rapid development of VLSI technology is part of that growth. The basic thing required of a telecommunication system is that it should transmit signals of acceptable quality from one location to another at an acceptable cost.

The quality of signal transmission is often characterized by the transmit rate and the bit error rate (BER) - high transmission speed and low bit error rate being desirable. Design work is aimed at a reduced noise level which will improve speed and BER figures while keeping its cost as low as possible.

The block diagram of the radio part of a typical transceiver is shown in Figure 1.1, which includes the receiver path from the antenna to the A/D convertor, and the transmitter path from D/A convertor to the antenna. The quality of the local oscillator (feeding the mixer, and is normally a frequency synthesizer) is critical to the quality of the transceiver itself. Characteristics of low phase noise and high frequency stability are always desired.

Wireless communication is not the only place where frequency synthesizers find use. They serve in applications such as video, audio, graphics processors and DSP apparatus, and as clock generators in microprocessor systems.

Figure 1.1 Block diagram of a typical transceiver

Phase-locked loops (PLLs) are a class of devices which are widely employed in the most contemporary frequency synthesizer designs. New techniques are always being explored to enhance the output phase noise performance, among which recently delay-locked loop (DLL) has been employed in frequency synthesizer design as a new technique. Compared with PLL-based frequency synthesizers, those based on the DLL display reduced output phase noise level due to the lack of phase noise accumulation in the voltage-controlled delay line (VCDL), the defining component of the DLL, whereas in PLL-type synthesizers, phase noise can accumulate indefinitely in the device's voltage-controlled oscillator (VCO) [1].

So in theory at least, a DLL-based frequency synthesizer can achieve lower phase noise than PLL-based systems. In recent years, several DLL-based frequency synthesizers/clock generator have been reported, and most of the design effort was aimed at reducing phase noise with the measured noise performance that can be further improved with more efficient circuit structures. The objective of this thesis work is aimed at modelling, simulation, and actual integrated circuit device design with some new circuit ideas to fur-

ther reduce the output phase noise. The predicted noise-reduction efficacies of new circuit architectures are tested using measurements taken of a new IC fabricated using CMOS $0.18\mu m$ technology. The circuit operates up to 2GHz with a supply voltage of 1.8V, and a phase noise of -105.5 dBc/Hz at the offset frequency 10KHz. The active area is only about 0.06 mm^2 , and the maximum power consumption is about 7.2mW.

1.2 Contributions

The major contributions of this thesis are:

1. Complete linear models for basic PLLs and DLLs are developed and their noise performance are compared using theoretical analyses and Matlab simulation.

2. A DLL-based frequency synthesizer is proposed and its functionality verified by Matlab model simulation. It employs a lock detector to eliminate the need for control voltage initiation and enable it to recover automatically after false locking. Novel type of delay line and edge combiner designs make the synthesizer insensitive to the reference signal duty cycle variation. This proposed structure can be employed to implement an integer-N frequency synthesizer by changing the number of delay stages and modifying the lock detector.

3. The proposed DLL synthesizer is subjected to detailed analysis, first of the components individually, and then of the complete assembly of these components. A 9-time DLL-based frequency synthesizer is fabricated in $0.18\mu m$ CMOS technology, and its performance quantified by the actual measurement results.

1.3 Thesis outline

In this thesis, Chapter 2 reviews the typical different techniques of frequency synthesis, including direct and indirect methods. Emphasis is placed on the indirect techniques because among these are numbered the two methods of interest in the present work,

namely the PLL and the DLL. Chapter 3 describes a theoretical analysis of phase performance in PLLs and DLLs, with Matlab simulation being used to evaluate the predictions. Chapter 4 follows up with an analog model of a DLL-based frequency synthesizer and verification of its functionality, and then the proposed structure of each block - the phase detector, the lock detector and the edge combining circuit - is also described. The complete frequency synthesizer assembly is proved functional. Chapter 5 presents the CMOS design of a DLL-based frequency synthesizer which produces an output frequency 9 times that of the reference. In Chapter 5, the working principles of the phase detector employed in the design are analyzed in detail. Just as in Chapter 4, the study is done first at the component level and then at the system level. In Chapter 6, layout considerations, test setup and the measurement results are described and discussed. Chapter 7 summarizes the previous chapter results, and some suggestion is presented for future research directions. In the appendix are such things as schematics and layout diagrams, and following this material is a bibliography of referenced papers and books.

Overview of Frequency Synthesis

2.1 Techniques of frequency synthesis

A frequency synthesizer is a combination of system elements used to derive one or many frequencies from one or a few reference sources [1], [4]. Early devices took the form of crystal-controlled oscillators with banks of crystals switched manually. But rapid development in communication electronics has led to a need for more sophisticated frequency generation methods, and in the last 60 years, a number of frequency synthesis techniques have been developed. Generally, two methods of frequency synthesis can be distinguished as direct synthesis and indirect synthesis. Among these, indirect synthesis methods based on the phase-locked loop (PLL) are widely used in the field of electronic applications. For example, it is employed in a clock generator to drive some DSP devices, audio, video processors and so on.

2.2 Direct frequency synthesis

2.2.1 Table look-up synthesis

Table look-up synthesis is also called direct digital synthesis (DDS). A simplified block diagram of a device which does this is shown in Figure 2.1. It consists of four main components: a phase accumulator, a phase to amplitude converter (sine look-up table), and a digital to analogue converter (DAC) [1], [3], [5], [6].

Figure 2.1 Basic diagram of a standard DDS

A sequence of addresses are used to select stored samples of a sinusoid, samples which are then processed by the DAC and LPF to produce a clean analog waveform. The addresses is obtained by adding an (M-1)-bit integer corresponding to a phase increment ($\Delta\theta$) to the current address at each clock pulse. The register output corresponds to the stored phase of the signal and is used to address the look-up table that contains numbers proportional to the sines of the addresses as shown in Equation (2.1).

$$L_{out} = \sin\left(2\pi \frac{S_{\Sigma}}{2^M}\right) \quad (2.1)$$

where S_{Σ} is the adder output, and the table output is converted to an analog voltage by the DAC and filtered by LPF to remove the high frequency spur. Assuming F_d is the digital value of $\Delta\theta$, and the adder with a capacity of M bits, then the synthesized frequency is [6]

$$F_{out} = \frac{\Delta\theta}{2\pi} F_{ref} = \frac{F_d F_{ref}}{2^M} \quad (2.2)$$

Figure 2.2 illustrates the procedure of phase to amplitude conversion. The angle θ is the accumulated M-bit value of $\Delta\theta$, and once the accumulator ‘overflows’, the contents

in the register back to zero as shown in the figure, and the sum representing the accumulated phase θ is fed into the sine look-up table to generate amplitude which equals to $\sin(\theta)$.

Figure 2.2 Illustration of phase to amplitude conversion

This type of frequency synthesizer has the advantage of being able to switch from one frequency to another quickly. The output resolution depends upon the capacity of the look-up table and can be very fine. The frequency of the synthesized signal is determined by $\Delta\theta$ with a theoretical upper bound to $\Delta\theta$, and the maximum output frequency is always lower than the reference frequency, typically half of the clock frequency because mandated by Nyquist condition, which places an upper bound on $\Delta\theta$, namely 2^{M-1} . Another distinguished drawback is its excessive spurious content in the output frequency produced by the system components. The phase accumulator can be viewed as a fractional divider of the system signal [3]. From Equation (2.2), a fractional division can be achieved, resulting in the phase modulation at the output, which can cause non-harmonic spurs at the output and the phase is too high in some applications. This type of frequency synthesis also suffers from huge power consumption [4]. As reported in [7], the power consumption current is 685mA. Because of these limitations, this method is not usually considered for applications such as mobile radio and on-chip clock synthesis.

2.2.2 Analog direct frequency synthesis

Another direct synthesis technique makes use of multiplication, mixing and division to generate a prescribed frequency from a single reference [17]. Since it employs analog components, it is called as analog direct synthesis (ADS).

Its operation is best described by means of an example shown in Figure 2.3 [1]. In the figure, suppose switch1 selects a frequency as high as 39 MHz, and if 'a' equals to 3MHz, the output of the first mixer will be $4+39 = 43\text{MHz}$. The output of the divider is 4.3 MHz ($N_1 = 10$), then suppose the second switch selects 40 MHz ('b'= 4 MHz), and the output of the second divider is 4.43 MHz, which is $4b.a/10$ ($N_2 = 10$). Similarly, suppose switch3 selects 40 MHz ($c= 4$ MHz), then the output of the last mixer is 44.43 MHz. By subtracting the output from the last component by 40 MHz produces the final output of the synthesizer $c.ba (= 4.43 \text{ MHz})$. Advantages of this technique are that the frequency can be changed rapidly. But it is difficult to lay out this circuit, a large number of component are needed, and this results in a synthesizer that is bulky and power consuming. So it is not suitable for IC implementation. Another disadvantage is that the presence of spurious sidebands due to the non-linear elements used in the circuits.

Figure 2.3 Example of analog direct frequency synthesis

2.3 Indirect phase-locked loop frequency synthesis

2.3.1 PLL operation principles

The concept of phase-locking is known for more than half a century and nowadays the phase-locked loop finds many applications, the most common of which are frequency synthesis, phase modulation/demodulation, tracking filters, carrier recovery, and clock recovery [11]. The PLL has become popular because of its excellent performance, high speed and ease of IC implementation.

Figure 2.4 Basic structure of phase locked loop

The PLL is a negative feedback loop to force the phase of a frequency modulated oscillator to follow that of the reference signal. A basic block diagram of a PLL is shown in Figure 2.2 [11]. A measure of a PLL's performance is the phase error - the phase difference between the input signal and output signal. The loop is kept locked by tuning the VCO through adjusting the control signal to keep the VCO output frequency identical to the input frequency. The control signal is proportional to the phase error detected by the loop filter. In addition to its phase error, a PLL is also characterized by some other main factors: a frequency range over which it can acquire lock, its lock acquisition time, its tolerance of modulation without losing lock, the loop stability, output phase noise, and so forth.

Its principles of operation are now briefly particularized:

When the loop is unlocked, assume the input signal is and output signals are,

$$y_i(t) = A \sin[\omega t + \phi_i(t)] \quad (2.3)$$

$$y_o(t) = B \cos[\omega t + \phi_o(t)] \quad (2.4)$$

It is assumed that the phase detector has a sinusoidal transfer function, so that for the unlocked loop, the general expression for its output, the error signal u_1 , is written

$$u_1(t) = K_d \sin[\phi_i(t) - \phi_o(t)] \quad (2.5)$$

K_d represents the phase detector sensitivity in units of V/rad . The LP transfer function in the time domain is taken to be $f(t)$, and its gain K_f . Then the control signal can be written as the convolution $u_2(t) = K_f u_1(t) * f(t)$. The angular frequency unit is radians per second, and it means frequency is the time derivative of phase, and from that the instantaneous angular frequency is a function of control signal. A time domain nonlinear differential equation comes out as,

$$\frac{d\phi_o}{dt} = K_d K_f K_{vco} \{\sin[\phi_i(t) - \phi_o(t)] * f(t)\} \quad (2.6)$$

The phase-locked loop is a nonlinear system, and it is very complicate to analyze its performance. Fortunately, when a PLL is near lock, its dynamic response to the input changes (both phase and frequency) can be well approximated by linear model or small signal model. 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,

$$\frac{d\phi_o}{dt} = K_d K_f K_{vco} \{[\phi_i(t) - \phi_o(t)] * f(t)\} \quad (2.7)$$

A small signal model as shown in Figure 2.5 provides an alternative means of rendering the analysis tractable [1], [13]. In the figure, V_{do} is the offset voltage of phase

Figure 2.5 Linear model of phase-locked loop

detector, and it is the output voltage of the phase detector when the phase difference is θ . In some applications, it is desirable for V_{do} to be 0. Similarly V_{co} is the control voltage that keeps the output frequency of the voltage controlled oscillator in phase with the input when the loop is in lock, so unlike 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.

Normally it is easier to analyze in the frequency domain rather than in the time domain with respect to the PLL's response to some external disturbance, especially to the noise performance. Therefore $\Phi_i(j\omega)$, $\Phi_o(j\omega)$ and $F(j\omega)$ are denoted as the Fourier transforms of $\phi_i(t)$, $\phi_o(t)$ and $f(t)$, and a linearized transfer function of the PLL can be expressed as,

$$H(j\omega) = \frac{\Phi_o(j\omega)}{\Phi_i(j\omega)} = \frac{KF(j\omega)}{j\omega + KF(j\omega)} \quad (2.8)$$

where K is the product of $K_d K_f K_{VCO}$ (the open loop gain), and usually K_f is 1. With this equation, PLL's dynamic response to some external disturbance can be easily calculated. Usually the disturbance is modeled as a frequency step, a phase step and a frequency ramp.

2.3.2 Integer N division PLL frequency synthesis

Since digital frequency synthesis cannot be used for high output frequency and the direct analog synthesis hardware is too bulky for silicon integration, indirect phase locking frequency synthesis methods that use the principle of feedback in generating frequencies have become most widely used, which differs from direct frequency synthesis in important ways. In the phase-locked loop frequency synthesis, the output frequency can be made an integer multiple of the reference frequency by dividing the output before sending it to the phase detector. A simplified block diagram is shown in Figure 2.6.

The voltage-controlled oscillator (VCO) output frequency is divided by a integer number N . This divided frequency is compared with the reference frequency by the phase detector (PD), which outputs a signal proportional to the phase difference between the two inputs. This error signal is processed by the low pass loop filter to produce the control voltage, and when the control voltage changes, the output frequency of the VCO is tuned in such a way as to reduce the phase difference between the frequencies. When the loop is in lock, the reference and divided output frequencies are identical and the phase detector has a constant output.

Therefore, the output frequency of the locked loop is,

$$F_{out} = N \times F_{ref} \quad (2.9)$$

The loop filter of a PLL has a significant effect on the loop circuit's performance. It suppresses the undesired high frequency components in the PD output, and the phase delay of the loop filter has an important effect on the output noise, as well as on the acquisition time, loop stability and so on, and makes this kind of frequency synthesizer inherently slower than direct frequency ones. Increasing the divider modulus slows the response of the VCO to changes in the reference frequency and increases the acquisition time. There is thus an upper limit to the rate at which the output frequency can be changed.

Figure 2.6 General block diagram of N division frequency synthesizer

So the loop components introduce low frequency noise that can modulate the VCO control voltage and so corrupt the output frequency spectrum. The noise produced by the loop components, including the VCO, must be kept to a minimum. Thus the phase detector should be very accurate; the VCO must be of high quality; the frequency divider must be of high speed, the loop filter must reach an optimum compromise with respect to switching time, frequency settling, output noise and so forth.

2.4 Fractional-N division frequency synthesis

There is also need to synthesize a frequency which is fractional multiple of the reference frequency. The technique of fractional-N division frequency synthesis refers to a PLL-based frequency synthesizer which can produce an output frequency which is non-integer multiple of the reference frequency with fractional frequency step without decreasing the reference frequency. Usually the fractional frequency can be expressed as follows [11], [17],

$$f_{VCO} = f_{ref} \times \left(N + \frac{k}{M} \right) \quad (2.10)$$

where N in turn is the integer multiple of the reference frequency, M the minimum frequency step, and k is an integer value anywhere between 0 and M . Usually $(N+k/M)$ is written as $N.F$ with F representing fractional part.

A simple block diagram of fractional-N division synthesis is shown in Figure 2.7 [13]. In the figure, the modulus controller is used to select between divide-by- N or divide-by- $(N+1)$. By switching between the two moduli such that, on the average, the divisor lies somewhere between them, the effect of multiplying the input frequency by a fraction can be obtained. That is the so-called *division ratio averaging* as discussed in the following.

Figure 2.7 General block diagram of fractional N PLL frequency synthesizer

Suppose the frequency division is chosen to be N for p output cycles and to be $N+1$ for q cycles, then the time-averaged output will be [17],

$$F_{out} = \frac{p \times N + q \times (N + 1)}{p + q} \times F_{ref}$$

$$F = \left(N + \frac{q}{p + q} \right) \times F_{ref} \quad (2.11)$$

The 'F' part of the $N.F$ mentioned earlier is seen to be,

$$F = \frac{q}{p + q} \quad (2.12)$$

By properly choosing the number of cycles p and q , different fractional ratios can be achieved as desired. For example, $p = 3$, and $q = 2$ gives the fractional part $F = .4$, and with $N = 2$, an output frequency corresponding to the frequency division ratio 2.4 is

obtained. That means a rising edge occurs at the output of the VCO every 2.4 cycles of the reference signal, so the average period of the output is $1/2.4$ times of that of the reference signal.

An example of the modulus controller is illustrated in Figure 2.8 [14], [15]. In the

Figure 2.8 An example of digital phase accumulator (DPA)

figure, It can be seen to consist of an adder and a register and is similar in function to the phase accumulator of the DDS device discussed in the first section of this chapter. As the register is clocked by the input frequency, its output increases by amounts equal to the n -bit word. The carry-out of the adder is the output of DPA. It is a 1-bit quantization of the tuning input. Suppose the tuning input $K = 2$ and $n = 3$, so whenever the value of the adder reaches 8, it is reset to 0 and outputs 1, i.e. the carry out is 1, and at this point, suppose that the DPA controls the value of the divider to be $(N+1)$, so the period of the divider output is $(N+1)$ times the reference signal every $2^n/K (=4)$ cycles of the reference signal, and the fractional part is $K/2^n (=0.25)$ [16]. In this way, by changing the tuning input, different non-integer times the reference frequency can be achieved.

But this type of frequency synthesis has a dramatic drawback. Since the frequency divider switches between N and $N+1$, the pulse at the output is not divided by equal time interval, an error in positioning the edges of the pulses will be easily produced and the divider can produce quantization phase error at the output [4], thus degrading the phase noise performance of the loop. To make it clear, suppose at some cycle of the reference

signal, the division ratio is N_i , and the divider output is $N_i \times T_o$, so compared with the ideal divider output $N \times F \times T_o$, there is a phase error as of [17],

$$\phi = \sum_i \frac{2\pi}{N.F \times T_o} (N.F - N_i) \times T_o \quad (2.13)$$

The equation shows that the phase error accumulates over the time and its value is independent of the ideal period of the output signal. And the phase error is periodic with a period of $0.F$. That is because every $\frac{1}{0.F}$ cycles of the reference signal, the adder in the digital phase accumulator outputs a high level 1. Figure 2.9 shows how the quantization

Figure 2.9 Quantization phase error diagram

phase error is produced. As illustrated in the figure, $N = 3$ and $F = 0.25$, so the output frequency is 3.25 times the F_{ref} , and every $\frac{1}{0.25}$ cycles of the F_{ref} , the quantization error returns to 0, so in this way, the quantization phase error modulates the control signal which tunes the output of the VCO and degrades its spectral purity.

2.5 Hybrid techniques of frequency synthesis

2.5.1 Dual-loop PLL frequency synthesis

The dual loop version of PLL synthesis is another possibility. A typical device for doing this is shown in Figure 2.10:

Figure 2.10 Dual-loop PLL frequency synthesizer

In this type of frequency synthesis technique, two loops are employed with different division ratios N and M respectively, and the mixer is inserted inside one of them. The two output frequency signals from the two loops are combined by the single-side band (SSB) mixer to produce the output frequency:

$$F_{out} = N \times F_{ref1} - M \times F_{ref2} \quad (2.14)$$

Similar with the DDFS, this technique is characterized by fast settling time. With fixed reference frequencies, the output frequency can be adjusted to the desired value by changing the division ratio N and M . For example, if N and M are both increased by 1, the output frequency increment is the product of $(F_{ref1} - F_{ref2})$. It has been estimated that in this case, the settling time is inversely proportional to the reference frequency [7], so for the same input frequency increment, the settling time of a dual-loop frequency synthesizer will be significantly smaller than that of the single loop frequency synthesis [1], and to

meet the requirement of fast settling time, the two reference frequencies are usually kept relatively large.

A single-sideband mixer is employed to mix the two frequencies. The unwanted sidebands resulting from mismatch and harmonics created by mixing are suppressed by the loop filter in the first loop.

2.5.2 Direct digital frequency synthesis (DDFS)

The direct digital frequency synthesis (DDFS) [1] can be used to achieve high output frequency and fast frequency switching. Besides, it can be integrated relatively easily. Figure 2.11 gives one example of DDFS - a combination of table look-up synthesis and

Figure 2.11 Diagram of direct digital frequency synthesizer

fixed N division PLL frequency synthesis techniques already described. With table look-up synthesis, a low frequency F_{low} can be generated, and it is up-converted to RF frequency with a fixed frequency oscillator signal F_{IF} by using a mixer and the output frequency is:

$$F_{out} = F_{low} + F_{IF} \quad (2.15)$$

By changing the loop bandwidth, the noise from PLL synthesizer can be optimized, and small frequency steps can be achieved by changing the frequency of F_{low} . Another advantage is that since the reference frequency can be chosen freely and the loop bandwidth could be optimized, the frequency can be changed rapidly. But F_{low} is limited by the speed of DAC, so the type of frequency synthesis suffers from a limited frequency range. And for the desired pure spectrum, DAC needs more complex structure resulting in higher power consumption. Therefore this technique is useful for a system requiring a fast frequency tuning speed [4].

2.6 Characteristics of PLL clock and frequency synthesis

In most applications, the frequency synthesizer is designed for use as an integral component in a larger circuit or system. Thus the operational requirements regarding the frequency synthesizer will vary for different applications. To define the various requirements, some important parameters are introduced as follows [19],

- Input frequency
- Frequency tuning range
- Frequency tuning step
- Frequency switching speed
- Phase noise
- Discrete spurious levels
- Power consumption
- Cost

The above parameters are not completely independent of one another.

In the following subsections, some considerations in the integer-N frequency synthesizer design are discussed.

2.6.1 Considerations in Integer-N PLL FS design

Frequency step size and frequency switching speed

As mentioned previously, in integer-N synthesizers, the step size is tied to the reference frequency. A smaller step size requires a lower reference frequency to keep the instantaneous period error low when switching divisions. On the other hand, the loop bandwidth is limited by the reference frequency. The loop bandwidth should be sufficiently smaller than reference frequency in order to keep the reference feed-through low. As a rule, the loop bandwidth usually is at least 10 times lower than the reference frequency to avoid the sampling effect of the PFD [21], [22]. Thus, in this respect, the smaller the step size, the narrower the loop bandwidth. However loop bandwidth is one important factor in determining frequency switching speed. Generally speaking, the wider the loop bandwidth, the faster the frequency switching speed. Therefore small step size conflicts with fast loop response unless high reference feed-through spurs can be tolerated.

Frequency step size and phase noise

Phase noise at a given offset frequency is determined by the in-band phase noise floor and the loop bandwidth. At a fixed offset frequency, phase noise can be lowered by narrowing loop bandwidth, resulting in a slower frequency-switching speed.

2.6.2 Phase noise

One of the important characteristics of frequency synthesizers is the phase noise, in particular, the output phase noise. Phase noise is more easily understood and analyzed in frequency domain. No matter whether an output waveform is that of an oscillator or of an edge combiner, it can be expressed as follows:

$$V_{out}(t) = A \cdot \sin(\omega_o t + \theta) \quad (2.16)$$

Where A is the amplitude, ω_o is the operating frequency, and θ is a fixed phase value. But that is the ideal case. In practical clock/frequency devices, both the amplitude and the phase of the output frequency are unavoidably modulated, so that a more realistic version of Equation (2.16) is,

$$V_{out}(t) = A(t) \cdot \sin[\omega_o t + \theta(t)] \quad (2.17)$$

The time-domain amplitude and phase fluctuations lead to an output frequency spectrum having sidebands close to the operating frequency. As shown in Figure 2.12 [23], these harmonics have an observable power spread around the center frequency. To

Figure 2.12 Power spectrum for ideal output and practical output

quantify those fluctuations, one usually considers noise power within a unit bandwidth, a noise density [20], [34]. With the noise power within that band divided by the operating frequency power, the single sided spectral noise density relative to the carrier power per Herz (dBc/Hz) at some offset frequency $\Delta\omega$ is calculated as shown in equation below,

$$L(\Delta\omega) = 10\log\left[\frac{P_{noise}(\omega_o + \Delta\omega)}{P_o}\right] \quad (2.18)$$

where P_{noise} is the noise power in unit bandwidth at the frequency of $(\omega_o + \Delta\omega)$ and P_o is the carrier frequency power.

2.6.3 Timing jitter

Timing jitter is a measure of the uncertainty in the edge positions of a square wave or the zero-crossing point uncertainty of any other periodic signals. In the case of frequency synthesizer, the timing jitter refers to the uncertainty in the transition edges of its

Figure 2.13 Principle of timing jitter measurement

output signal as shown in Figure 2.13 [10]. If the zero-crossing time of a signal is denoted by $t(n)$, where n is the cycle index, one has in the ideal case of zero timing jitter, $t(n) = nT_o$, where T_o is the period.

However, a real periodic signal always has some timing jitter, so that $t(n)$ is not generally equal to nT_o . Instead, it can be written as $t(n) = nT_o + J_t(n)$, where the $J_t(n)$ is the timing deviation from the ideal case. The timing jitter of a signal is normally defined to be the RMS value or the peak-to-peak value of the variation $J_t(n)$.

2.6.4 Period jitter

Besides the timing jitter, there is a related parameter which quantifies the uncertainty in the period of a periodic signal. For an ideal periodic signal, the period T_o is constant over time, but this is again not true of real periodic signals. Strictly speaking, there is no such thing as a periodic signal in the real life.

If the period of a real signal is denoted by $T(n)$, where n is the index of the cycle. $T(n)$ contains the effect of jitter, and it can be written as $T(n) = T_o + J_p(n)$, where the $J_p(n)$

Figure 2.14 Principle of Period jitter measurement

is the period variation of the real signal. Period jitter is expressed in the same units as is timing jitter, namely the RMS value or the peak-to-peak value of $J_p(n)$.

It is clear that the relation between $J_t(n)$ and $J_p(n)$ can be written as,

$$J_t(n) = \sum_{m=-\infty}^n J_p(m) \quad (2.19)$$

As in the case of timing jitter, period jitter is customarily represented by the RMS or peak-to-peak value of $J_p(n)$.

2.7 Summary

This chapter has reviewed the techniques and types of frequency synthesis and clock synthesis developed in the past, among those techniques, PLL-based methods has been emphasized. To better understand the PLL-based frequency synthesis, the principle of PLLs has been reviewed in detail to enable understanding of their operation and of important design considerations. Among the different frequency synthesizers, the fractional-N division synthesis was described in greater detail because of its importance and popular application.

Comparison of PLL and DLL systems

3.1 Introduction

This chapter focuses on the performance of phase-locked loops and delay-locked loops (PLLs and DLLs) systems. The advantages and disadvantages of both are discussed in detail. For the sake of discussion, it will be supposed that the PLL and DLL systems being compared are identical in all respects except for VCO and VCDL elements (see Figure 3.1); in particular, the two types will have the phase detector, charge pump and loop filter. Also, the PLL's VCO and the DLL's VCDL employ the same delay structure (a ring oscillator in the former case).

3.2 Operating principles of PLL and DLL

One of the main components of a PLL is a voltage-controlled oscillator (VCO), whose output frequency is a function of its control voltage. The phase difference between the VCO output and that of the reference signal is detected by a phase detector (PD) transformed to the output voltage of a phase detector (PD). This is used as input to a charge pump (CP), whose output is in turn sent to the loop filter (LF). The LF output voltage is then sent to the VCO to tune the output frequency in such a way as to keep the phase error between the VCO output and the reference signal a constant value, normally zero, but it

doesn't have to be exactly zero in many applications. The operating principle of a PLL can be deduced by studying Figure 3.1-A.

In the DLL, the PLL's VCO is replaced by a voltage controlled delay line (VCDL), a device which outputs a delayed version of its input, with the delay being a function of a control voltage. A DLL operates in a way which is analogous to the operation of a PLL. The output of the VCDL is compared to the input, or reference signal, by a phase detector.

Figure 3.1 Operating principles of PLL and DLL

The output of the charge pump and loop filter is used to adjust and hold the delay generation of the VCDL to a constant value, which is normally equal to one period of the reference signal. In other words, the loop tunes the control voltage of the VCDL to keep its delay a constant value, which is normally a period of the reference signal. The appropriate block diagram is shown in Figure 3.1-B.

3.3 Applications of PLL and DLL

Despite differences between PLLs and DLLs, which may make one of the two types more suitable for some applications than the other, they represent options for uses in the applications such as frequency synthesis, phase demodulation, and so on. An application comparison is given as follows. The common PLL applications which were given in previous chapter and repeated here are tracking filter, frequency demodulation, phase

demodulation, frequency synthesis, carrier recovery, clock recovery, and so on. And the common uses for DLLs are phase adjustment, frequency synthesis, clock recovery, etc.

3.4 Phase noise in PLL/DLL based frequency synthesizers

Recently ring oscillators have been finding more uses in integrated circuits because they are easily integrated, and exhibit phase noise performance which is normally competitive with that of LC oscillators. In this section, the phase noise in the ring oscillator used as the VCO in a PLL will be compared to the phase noise of the VCDL of a DLL. To facilitate the comparison, ring oscillators and delay lines will be taken to consist of identical numbers of identical inverters.

3.4.1 Phase noise in the VCO of PLLs

Assume in a PLL-based systems the VCO is voltage-controlled ring oscillator, and the phase noise of a ring oscillator type of VCO can be analyzed as follows. Figure 3.2

Figure 3.2 An n-stage ring oscillator

shows an n-stage ring oscillator. A CMOS inverter is used for each stage of the ring VCO. In theory, the outputs of the stages are identical except for the phase shift. The output of this ring oscillator at a certain stage can be represented as

$$V_{out}(t) = A(t) \cdot W(\omega_0 t + \theta(t)) \quad (3.1)$$

where function W is a periodic function with a period of 2π , which defines the form of the output and normally is a square wave. $A(t)$ is the amplitude of the output wave.

For an ideal oscillator without phase noise, $\theta(t)$ keeps constant. However in real circuits, due to the noise sources caused by capacitive and resistive paracitics, the oscillator has a phase which is a function of t ($\theta(t)$), and therefore exhibits output phase noise.

To model the phase variation and analyze its effect, a single source of current noise $i(t)$ is placed at one of the nodes shown in Figure 3.2. During a time interval dt , the voltage change dV caused by the noise source $i(t)$ can be represented,

$$dV = \frac{i(t)dt}{C_n} \quad (3.2)$$

where C_n is the total capacitance at node n . As mentioned earlier, the stages are identical and theoretically $C_1 = C_2 = \dots C_n$. The node voltage variation can affect the phase of the signal during propagation through the ring. The phase variation $d\phi$ caused by dV in Equation (3.2) depends upon the value of the voltage swing V_{swing} at that node and the time at which the noise is injected as well as the output wave shape. The larger the voltage swing, the smaller the value of $d\phi$. A voltage change with the value of dV can cause larger phase variation $d\phi$ at the transition edges than at any other time. At the transition edges, the phase shift $d\phi$ caused by the voltage variation can be written as

$$d\phi = \frac{\omega_0 dV}{k} \quad (3.3)$$

Where $k = \frac{dV_{out}}{dt}$, and the equation shows the phase variation is inversely proportional to the slope of the signal at transition edges. In other words, the phase shift caused by dV gets larger at the larger rising time and falling time. However, Equation (3.3) is invalid only at the transition time. For example, when $V_{out}(t)$ is high at a time well removed from a transition edge, as shown in Figure 3.3, the voltage jump dV_1 does not cause a phase shift. Generally speaking, the phase variation is proportional to the voltage jump at some time and it can be represented as

$$d\phi = \frac{\Gamma(\omega_0 t)}{V_{swing}} dV \quad (3.4)$$

where V_{swing} is the amplitude of the square wave shown in Figure 3.3. The function $\Gamma(\omega_0 t)$ which is periodic in 2π shows the effect of the voltage jump on the phase shift. For a given voltage jump, the phase shift is proportional to $\Gamma(\omega_0 t)$. Since $\Gamma(\omega_0 t)$ represents the variation of sensitivity to the voltage disturbance, it is called the impulse sensitivity function [9].

Figure 3.3 Effect of impules injected at transition edges and saturated peaks

The ring oscillator is a closed loop circuit, and any phase shift will therefore circulate for an infinite time. The phase response to a voltage jump occurring at t_0 can be expressed as a function of the time,

$$\phi(t, t_0) = \frac{\Gamma(\omega_0 t_0)}{V_{swing}} dV(t_0) u(t - t_0) \quad (3.5)$$

where $u(t)$ is a unit step function, equal to 1 for $t \geq 0$ and 0 otherwise. If one is given the phase response at an initial time t_0 , the phase of the output signal for all subsequent time $\theta(t)$ can be obtained by summing the responses to all voltage jumps occurring before t_1 ,

$$\theta(t_1) = \int_{-\infty}^{t_1} \phi(t_1, t_0) dt_0 \quad (3.6)$$

If noise currents at all stages have identical amplitudes and are un-correlated, the total phase noise power is N (total number of stages) times of the phase noise power at each stage.

3.4.2 Phase noise in the VCDL of DLLs

Phase noise analysis in delay lines of a DLL can be studied by considering the differences between delay lines and ring oscillators with identical structures. An n -stage delay line is shown in Figure 3.4.

Figure 3.4 An n -stage delay line

The phase jumps caused by the current noise source at each input stage of the model delay line are identical to those seen in the ring oscillator just studied, except that this noise does not persist for an infinite time because the output is not fed back to the input. In other words, the phase jump at one stage only affects the following delay stages and it is not fed back from the output to the input as in the case of a ring oscillator. For the delay line of Figure 3.4, the actual phase noise of each stage of the delay line is not exactly the same as that of the other stages. The phase noise increases as the reference signal propagates from one delay stage to the next.

It is clear that because of the absence of output-input feedback, the noise performance of a delay line will be superior to that of a ring oscillator, given a low phase-noise reference signal and that both the oscillator and delay line are made up of the same number of inverters.

However, as the number of stages increases, the delay line loses its noise advantage over the ring oscillator because the noise will accumulate stage by stage and the noise accumulation causes almost the same noise performance.

3.5 PLL and DLL system output timing jitter

For the ideal output of oscillators or frequency synthesizers, the spacing between transitions is the same. However, in practical circuits, it varies cycle by cycle. This variance is represented in the time-domain as a timing jitter. Generally speaking, timing jitter is the time-domain representation of period variation while the phase noise normally only involves random components of phase variation in frequency domain. Some periodic phase variations will appear in the spectrum as spurs.

As does the phase noise, timing jitter also can accumulate over cycles of the output of some frequency synthesizers which use feedback loop. At each rising or falling edge of the reference signal, the integer-N division PLL-based synthesizer tends to reduce the timing jitter accumulated over previous cycles of the output signal. In this respect the behavior of a DLL-based frequency synthesizer is better than that of an 'equivalent' PLL device, in that it not just reduces by tuning the VCDL control voltage but actually resets the timing jitter of the output to zero every n -th cycle (in an n -stage device).

DLL-based synthesizers have, however, a source of jitter not found in PLL devices - even if the edge of the delayed signal exactly coincides with the reference edge, If the total delay of the VCDL is exactly equal to one period of the reference signal, the output will still have some jitter which is the result of mismatching among delay stages. Although there is no such problem for PLL-based frequency synthesizers, the ability of a DLL-based frequency synthesizer to compensate the phase noise or timing jitter by the loop remains on the whole better than a PLL-based one. This will be discussed in the following sections.

3.6 PLL and DLL's ability to suppress phase noise and jitter

In the previous sections, the phase noise and the timing jitter in PLL ring oscillators and DLL delay lines are examined. The ring oscillator and delay line can employ identical delay stages and differs only in that ring oscillator's output is fed back to its input. Theoretically each delay stage contributes the same amount of noise or timing jitter. However, the noise performance of the synthesizer output are different due to the different structure of PLL and DLL.

3.6.1 PLL-based system noise performance analysis

This analysis is based on the PLL-based frequency synthesizer with ring oscillator described in the previous sections. The timing jitter in the ring oscillator depends on the interaction of noise in the oscillator and the dynamics of the phase-locked loop. For an open loop, the timing jitter of the ring oscillator will be the sum of noise from each stage and it is accumulated cycle by cycle. If the jitter introduced by each delay stage is $\overline{\Delta t_n^2}$, the final output jitter will be $k\overline{\Delta t_n^2}$, where k is the number of stages through which the transition edge of the input signal has propagated since the starting time. Thus the timing jitter increases with time ΔT as shown in Figure 3.5 [9], [10] because any phase shift will persist infinitely in the oscillator loop.

However, once the PLL loop itself is closed, the jitter at the output will be detected by the phase detector and the loop will dynamically compensate for the phase shift by adjusting the control voltage of the ring oscillator. Thus, as shown in Figure 3.6, the timing jitter can never increase infinitely without limit. It is bound to a certain steady-state level as long as the loop is in lock.

The adjustment or compensation behavior of the loop usually requires more than one cycle time, and can not be too fast in most applications, so a certain amount of phase jump can not be compensated within just one cycle. Therefore the jitter in PLL will accu-

Figure 3.5 Accumulation of timing jitter over cycles

Figure 3.6 Jitter suppressed by a PLL

multate over many cycles. From this point, obviously the fewer the cycle times required for the phase error correction, the better the output jitter performance will be. The only way one can increase the compensation speed is to increase the loop bandwidth, which, however will allow more phase noise from reference signal to feed through to the output of the loop. Thus it is very important to find a compromise between the accumulated timing jitter and the loop bandwidth of clock synthesizers.

The reference frequency of most PLL-based synthesizers commonly comes from a low jitter signal source such as a crystal oscillator, and in those cases, the jitter in the output frequency originates mainly in the voltage-controlled oscillator in the PLL, and a large bandwidth should then be employed to reduce locking time and therefore achieve lower

output jitter. On the other hand, if the reference signal is not ‘clean’ enough comparing with the ring oscillator, for example, when PLL is used in clock recovery circuits, the designer must optimize the bandwidth for the lowest feed-through or the lowest jitter performance and a balance should be obtained among these competing factors.

A simplified discrete-time domain model for a PLL-based synthesizer incorporating a ring-oscillator is presented in Figure 3.7. The internal noise caused by the ring oscil-

Figure 3.7 Simplified PLL model

lator is made up of all the noise contributions produced at each delay stage as explained earlier. For the purpose of mathematical analysis, the internal noise can be modelled as an equivalent voltage noise source ϕ_n , which is added to the control signal of the VCO. In Figure 3.7, ϕ_{on} denotes the output jitter due to both internal and external jitter sources. It should be emphasized that ϕ_{on} does not denote the actual output of VCO, it refers only to the noise component in the VCO output ϕ_o instead. In this section, the input signal is supposed to be an ideal signal and only the output jitter caused by internal equivalent noise ϕ_n is analyzed, so for the following analysis, ϕ_{on} only denotes the jitter due to internal equivalent noise ϕ_n .

In clock synthesizer, the timing jitter is accumulated from one period after another and most PDs work on a discrete time basis, so it is more convenient to analyze the noise performance in discrete time domain than in s-domain or frequency domain. Another

advantage is that functions can be converted to z-domain from discrete time domain conveniently.

To analyze the output jitter caused by the internal jitter of a VCO, a simplified discrete time model for the PLL as a whole was developed as shown in Figure 3.8. The inter-

Figure 3.8 Simplified PLL discrete-time model for phase noise analysis

nal jitter of the VCO, $\phi_n(n)$ is accumulated over cycles. This would carry on indefinitely without the compensation of the loop. The relation between $\phi_n(n)$ and $\phi_{on}(n)$ can be written in z-domain as

$$\Phi_{on}(z) = (-\Phi_{on}(z) \cdot K_{pd} \cdot F(z) + \Phi_n(z)) \cdot \frac{z^{-1} K_{vco}}{1 - z^{-1}} \quad (3.7)$$

where K_{pd} is the gain of the phase detector, $F(z)$ is the transfer function of the loop filter in z-domain and K_{vco} is the gain of VCO in the discrete time domain. Equation (3.7) can be rearranged to define a jitter transfer function for PLL as

$$H(z) = \frac{\Phi_{on}(z)}{\Phi_n(z)} = \frac{z^{-1} K_{vco}}{1 - z^{-1} + z^{-1} K_{vco} \cdot K_{pd} \cdot F(z)} \quad (3.8)$$

From the equation above, The presence of K_{vco} in the numerator suggests that an increase in output jitter is at least possible when this parameter increases, but further

examination reveals that this will not happen. The internal jitter of the VCO is modelled as an equivalent noise presented to the control signal of the VCO, and the relation between the actual internal jitter and the equivalent jitter is,

$$\phi_{novco}(n) = K_{VCO} \cdot \phi_n(n) \quad (3.9)$$

where $\phi_{novco}(n)$ is the actual jitter of the VCO, which normally has constant power spectrum for the same VCO and under the same working environment.

So in fact, the output jitter will decrease while the K_{VCO} increases since it is at the position of denominator in the Equation (3.8).

3.6.2 DLL-based system noise performance analysis

The main difference between the DLL and the PLL with a ring oscillator output noise performance is that the noise at the output of the VCDL is not fed back to its input, so there is no jitter accumulation in the VCDL and only the jitter produced in the present cycle is ever seen. The simplified time-domain DLL model is shown in Figure 3.9.

Figure 3.9 Simplified DLL discrete-time model

The output of the VCDL, ϕ_o , is the reference signal ϕ_{ref} itself after having been delayed with the delay value tuned by the control signal. As in the previous PLL analysis,

the internal jitter of the VCDL will be modelled by an equivalent noise ϕ_n applied to the control input.

Using the superposition principle, the discrete model for noise analysis can be developed as shown in Figure 3.10.

Figure 3.10 Simplified DLL discrete-time model for phase noise analysis

The output of the VCDL is the sum of the previous phase (which is zero for noise analysis) and the delay which is the product of K_{VCDL} and the control voltage. For the sake of the noise analysis, the reference is always 0, the transfer function of the VCDL can be written as

$$H_{VCDL}(z) = z^{-1} K_{VCDL} \quad (3.10)$$

Consequently the relation between the output jitter $\phi_{on}(n)$ and the internal equivalent noise $\phi_n(n)$ can be written in z-domain as,

$$\Phi_{on}(z) = (-\Phi_{on}(z) \cdot K_{pd} \cdot F(z) + \Phi_n(z)) \cdot z^{-1} K_{VCDL} \quad (3.11)$$

Where K_{pd} is the gain of phase detector, $F(z)$ is the z-domain transfer function of the loop filter and K_{VCDL} is the gain of the VCDL in the discrete time domain. The z-domain jitter transfer function for DLL can be derived from the Equation (3.7),

$$H(z) = \frac{\Phi_{on}(z)}{\Phi_n(z)} = \frac{z^{-1} K_{VCDL}}{1 + z^{-1} K_{VCDL} \cdot K_{pd} \cdot F(z)} \quad (3.12)$$

On comparison with Equation (3.8), the output jitter of a DLL is seen to be smaller than the output jitter of a PLL using the same loop filter and phase detector when $K_{VCDL} = K_{VCO}$. Furthermore, if K_{VCDL} or K_{VCO} is sufficiently large, the PLL can lock quickly and it can compensate the accumulated jitter very quickly, thus the jitter doesn't accumulate excessively for a PLL in this case. This means that, the output jitter of a PLL is close to that of a 'similar' DLL. Similar conclusions can be drawn regarding K_{pd} or $F(z)$.

3.6.3 Noise powers at the outputs of PLL and DLL

In this section, the transfer functions of the output noise due to the internal noise of a VCO or a VCDL are derived. To compare the abilities of a PLL and a DLL to suppress internal noise, the output noise power due to equivalent internal jitters in the VCO and the VCDL components are also calculated.

The ratio of PLL and DLL noise power can be expressed as,

$$R = \frac{\sigma_{PLL}^2}{\sigma_{DLL}^2} = \frac{\int_0^\pi |H_{PLL}(\omega)|^2 d\omega}{\int_0^\pi |H_{DLL}(\omega)|^2 d\omega} \quad (3.13)$$

To compute this, expressions for the transfer functions will be needed. The necessary expressions for these were given earlier (Equations 3.8 and 3.12) - all that must be done is the substitution into them of loop filter expressions. As indicated earlier, both of

the PLL and DLL models are to use the same loop filter - by way of example, the low-pass LF transfer function is as follows,

$$F(z) = 0.05 \cdot \frac{z+1}{z-0.9} \quad (3.14)$$

which is a low pass filter with a pole at $z = 0.9$ and a zero at $z = -1$. Thus in this particular case of the PLL, Equation (3.8) which gives the z-domain transfer function becomes,

$$H_{PLL}(z) = \frac{z^{-1} K_{vco}}{1 - z^{-1} + z^{-1} K_{vco} \cdot K_{pd} \cdot 0.05 \cdot \frac{z+1}{z-0.9}} \quad (3.15)$$

Similarly, for the DLL with Equation (3.12), the z-domain transfer function can be expressed as,

$$H_{DLL}(z) = \frac{z^{-1} K_{VCDL}}{1 + z^{-1} K_{VCDL} \cdot K_{pd} \cdot 0.05 \cdot \frac{z+1}{z-0.9}} \quad (3.16)$$

These last two equations can easily be converted into the corresponding frequency-response expressions by using the relation between a z-domain transfer function and its frequency-domain equivalent as

$$H(\omega) = H(z); \quad z = e^{j\omega} \quad (3.17)$$

So the frequency responses of the PLL/DLL can be derived from Equation (3.15) and Equation (3.16) respectively, thus the power ratio between the outputs of PLL and DLL can be calculated. An example is specialized by putting $K_{vco} = K_{VCDL} = 1$, and by changing the value of K_{pd} from 0.001 to 0.3, the power ratio is plotted as shown in Figure 3.11-A. The transfer functions from PLL and DLL when K_{pd} is equal to 0.2 are also plotted as Figure 3.11-B.

(A)

(B)

Figure 3.11 PLL/DLL power ratio and transfer functions

It can be seen that for very small K_{pd} , the noise power of the PLL is far larger than that of DLL. However, the difference decreases with increasing K_{pd} because the PLL can compensate the jitter accumulation more and more quickly, resulting in the smaller difference between PLL and DLL.

Turning away from the previous example, it is remarked that in general the DLL is not as good as the PLL at rejecting the jitter coming from external sources, such as from

the reference signal. The PLL is superior here, in that it can suppress the input jitter greatly while DLL lacks this characteristic. However, this does not turn out to be a major difficulty in frequency synthesizer applications since in these cases the reference signal usually comes from a low-jitter signal source such as a crystal oscillator, so that most of output jitter of the DLL comes from the internal jitter of VCO or VCDL, and not from the external sources, so the DLL has its special advantage and represents a better choice in the application of frequency synthesizers. However, the DLL has operational problems of its own, ones not shared with PLL, and the most serious of these is for a DLL-based frequency synthesizer with a pulse combiner and is called ‘mismatching’, which produces some jitter, while a PLL-based frequency synthesizer has no such problem. This will be discussed in the following chapters.

3.7 Behavior simulation of PLL and DLL

To analyze the different jitter performance of PLLs and DLLs, mathematical models of both were set up with Matlab simulink with a noiseless input and the output jitter was observed for each. Because only the jitter performance is to be observed, some simplified models were employed. In the mathematical models, the input and output are modelled as the phase which increases linearly for an ideal oscillator and the input jitter is the phase variation over the linear increase. In the case where input was kept at a constant value of zero, the superposition assures only the jitter will appear as the output signal. When there is no jitter, the output signal will be 0.

3.7.1 PLL model and simulation

The phase of the VCO is determined by the integrated value of its control signal which in the discrete time domain can be written as

$$Out(n) = Out(n - 1) + In(n - 1) \quad (3.18)$$

Where $Out(n)$ is the output phase of the VCO, the $In(n)$ denotes the normalized frequency which is the sum of the free running frequency (f_0) when $V_{control}(n)$ is 0, and the product of the gain of the VCO and $V_{control}(n)$. Because the internal jitter is an essential part of this model, a jitter signal denoted as $Jitter(n)$ should also be added to the $In(n)$. This will be accumulated over cycles. The complete model of VCO for the purpose of jitter simulation is shown in Figure 3.12.

Figure 3.12 VCO model for jitter simulation

Since both the input and the output are represented as the values of their respective phases, the model of the phase detector can be represented as the product of K_{pd} and the phase error between its input phases, and the loop filter is a low pass filter having the same transfer function used previously (Equation (3.14)). Consequently, the entire PLL model for jitter simulation can be set up as shown in Figure 3.13.

The jitter generator provides a random signal which simulates the internal jitter of the VCO at every cycle. The jitter accumulation that occurs is thus observed at the output of the VCO despite the loop's compensation of it. Simulation results displayed in a time scope with two channels show the effect. The simulation results for $K_{pd} = 0.2$ and $K_{VCO} = 1$ are shown in Figure 3.14. The input power of the jitter is as about eight times input noise power because of jitter accumulation.

Figure 3.13 PLL model for jitter simulation **Figure 3.14 Input and output jitter of PLL simulation**

3.7.2 DLL model and simulation

The next step in the PLL/DLL comparison was the analogous modelling of the DLL. A similar experiment was set up and run to show the noise advantage of a DLL over a PLL of the example (Equation (3.18)).

The VCDL is a component which outputs a delayed version of the input signal and the delay is a function of the control voltage. If the input and output signals are both represented with their phases, the output of the VCDL is just the sum of the input of VCDL and a delay time determined by its control signal. From mathematical point of view, this can be expressed as,

$$Out(n) = In(n-1) + Delay(n-1) \quad (3.19)$$

Where the $In(n)$ is the reference signal to the DLL, and $Delay(n)$ the length of delay in the following cycle. As in the VCO case, the value of $Delay(n)$ should include a free running delay time $Delay_o$, which shows the delay value when the control signal of VCDL is zero and is added to the product of K_{VCDL} and the control signal. For the experimental purpose of jitter simulation, $Delay(n)$ should also include the $Jitter(n)$, which represents the internal jitter of the VCDL during each cycle. The complete VCDL model for jitter simulation is shown in Figure 3.15.

Figure 3.15 VCDL model for jitter simulation

Figure 3.16 DLL model for jitter simulation **Figure 3.17 Input and output jitter of DLL simulation**

A DLL model for jitter simulation which used the same phase detector and loop filter as did the PLL model of Figure 3.13 was set up as shown in Figure 3.16. And a two-channel time scope is also used to display the input jitter and the output jitter.

The simulation results for K_{PD} of 0.2 and K_{VCDL} of 1 are shown in Figure 3.17. This time, the output jitter power is nearly the same as the input jitter power. Comparing this results with the PLL results, the jitter performance of the DLL is seen to be far better than that of the PLL as far as internal jitter of the VCO or the VCDL is concerned.

3.8 Loop components requirements of PLL and DLL-based frequency synthesizers

Because they use different operating principles, DLL-based frequency synthesizer place different demands on the components that make them up than do their PLL-based counterparts.

The delay difference between the stages of a PLL's ring oscillator is not important since the output of the synthesizer is obtained from only one of these stages. However, for delay line of the DLL, such stage-to-stage differences will introduce timing jitter at the output since the output is obtained by combining the outputs of all delay stages.

Since it is adequate for the PLL just to keep the phase difference of the two inputs of phase detector at some constant value, not necessarily zero, and some systematic, time-independent phase errors in the fed-back path, including those of the phase detector, the charge pump and the loop filter, will not affect the performance of the PLL-based synthesizers. But it is not true for the DLL-based synthesizer. The VCDL must be tuned to exactly one period of the reference So that the aforementioned errors are not permissible. In particular, the phase difference between two inputs of phase detector should be minimized, ideally to zero. These performance requirements on the loop components mean that one may have to resort to specialized technology in their design.

Finally, the loop filter of the a PLL plays a very important role in its performance. Changing the loop filter can bring about dramatic changes to some important loop characteristics. The loop filter of a DLL, on the other hand normally only affects the locking speed, so that the device will often have a simpler design than would a PLL loop filter, usually a first order loop filter [25], [26].

3.9 Summary

The relative advantages and disadvantages of PLL and DLL-based frequency synthesizers were explored in this chapter. Simulation was used to verify the predicting theory, namely that the DLL type has compelling advantages over the PLL type where phase noise/jitter performance is concerned. The following chapters will therefore concentrate on the DLL-based synthesizer.

Proposed Architecture of the Frequency Synthesizer

4.1 Introduction

In the previous chapters, some representative techniques of frequency synthesis are reviewed and a new means of to synthesize multiple frequency, the DLL-based one, is analyzed and compared with a typical PLL-based system. In this chapter a new way to decrease the phase noise in the output clock is proposed and analyzed using behavioral simulation. In addition, this chapter presents a survey of reported DLL-based frequency synthesizers. Target specifications are also listed.

4.2 Survey of relevant publications of DLL-based frequency synthesizers

Some papers on the topic of DLL-Based clock synthesizer have been published [8], [24], [26], [27]. Each paper employs the DLL which provides multiple phases that are combined to produce the desired output frequency of the synthesizer. In [8], the circuit employs a differential delay stage structure which helps to minimize the supply-induced and substrate-induced jitter noise. One of the disadvantages is a large inter-stage delay variation and a large inter-period jitter (100ps) as shown in the paper. A unique feature of [26] is its use of LC tanks at the output to enhance the load impedance at the resonance frequency. But since the quality factor of the inductor is low, the LC-tanks do not help to suppress the phase noise of the output signal, but do help to achieve a bigger voltage

swing due to large tail current. As for [27], there are only a few changes in the circuit. For example, in PD a new reset circuit is attached to the conventional PD, and the edge combining circuit also has a different structure from others.

TABLE 4.1: Summary of recent publications on the DLL-based frequency synthesizer

Ref	Process	Structure & Area (mm ²)	Reference & output frequency	Supply voltage (V)	Tuning range (MHz)	Phase Noise (dBc/Hz) (ps)	Duty cycle ratio / loop bandwidth	Power (mW)
[8]	0.5μm CMOS	Differential. 9 delay-stages / 0.6	180MHz / 1.62GHz	1.8~3.3	@2V, 600~1000 @3.3V 900~1600	@ 1GHz edge jitter -P-P <20 ps, -rms = 3.2ps -inter-period jitter= 100ps	N/A / 1MHz	40 @2V
[24]	0.25μm CMOS	PLL with realignment /0.22	19.68MHz (from crystal frequency /4-time the reference	2.7~3.3	N/A	-102.5@20kHz	N/A 25kHz	180
[26]	0.35μm CMOS	LC tanks 9 stages / 1.2 x 1.0	100MHz / 900MHz	3.3	N/A	-120@30KHz -123@60KHz -127@330K	N/A	130
[27]	0.35μm CMOS	4 delay stages / 0.07	250MHz /1GHz	3.3	120~1100	Cycle to cycle -rms jitter 2ps -p-p 14.56ps	50+-0.5% / N/A	55

4.3 System design of the proposed architecture

The analysis of Chapter 3 indicates that the DLL's noise performance makes it a good design choice when considering a new system architecture. Figure 4.1 shows the architecture of clock synthesizer being proposed in this design. It consists of a phase detector, a charge pump, a loop filter, a voltage controlled delay line, a lock detect circuits and an edge combiner. The loop filter is of first order since the DLL does not require suppressing the sideband noise to the same extent as does a PLL-based frequency synthesizer. The new idea for decreasing output phase noise appears in the design of the voltage controlled delay line as well as that of the edge combiner. The edge combiner uses only the rising edge of the incoming external clock. By way of contrast, in the reported clock syn-

thesizers, both the rising and the falling edges of the reference clock are used in the edge combiner. In this design, the use of the rising edge of the reference clock makes the output clock insensitive to the duty cycle of the reference clock, thus decreasing the output jitter to some extent because it helps to mitigate the inter-period timing jitter caused by changes of duty cycle in the delay line and by any duty cycle error in the reference signal. Although both rising and falling edges are used in the lock detector, the duty cycle doesn't affect the loop when it is in lock as long as the variance of the duty cycle of the reference is in a certain range.

Figure 4.1 System diagram of the proposed frequency synthesizer

4.4 Behavioral simulation of DLL-based frequency synthesizer with MatLab

To verify the proposed architecture's functionality, a complete model of a DLL-based frequency synthesizer with an output frequency nine times the input frequency was

modelled using Simulink of Matlab and the simulation results show that the proposed architecture of the frequency synthesizer works as expected.

4.4.1 Phase detector model

In the architecture presented in Figure 4.1, the phase detector works linearly with four inputs, *Clk_Ref*, *Clk_Delay*, *!Under*, *!Over* and two outputs, *Up*, *Down*. The width difference between *Up* and *Down* pulses encodes the phase difference between *Clk_Ref*, *Clk_Delay*. Since this frequency synthesizer employs a lock detector to prevent the loop from losing lock, other two inputs *!Under*, *!Over* are included in the phase detector. For the phase detector, *!Under* and *!Over* have higher priority than other two inputs.

To make the further work of phase detector easier, only logical gates are employed in the model of phase detector as shown in Figure 4.2, the detailed operating principle is described in the following chapter.

Figure 4.2 Model of the phase detector

4.4.2 Charge pump and loop filter models

The two outputs of phase detector represent the phase difference of its two inputs Clk_Ref , Clk_Delay , and the charge pump is employed to sink or source a current from or to a capacitor according to the width difference between Up and $Down$ pulses coming from the phase detector. Mathematically, the response of the loop filter to the charge pump inputs can be written as,

$$V_{control}(t) = \int_{-\infty}^t K(Up(t) - Down(t))dt \quad (4.1)$$

According to the equation above, the model of the charge pump and loop filter can be set up as shown in Figure 4.3. The limiter is employed to constrain the $V_{control}$ to a cer-

Figure 4.3 Model of charge pump and loop filter

tain range, as a real circuit does. In the model shown in Figure 4.3, a manual switch is used only to simulate the disturbance and it does not appear in the design of an actual charge pump or loop filter. The main parameter of this charge pump is the gain K_{cp} (V/s), which determines the increment of the output (control voltage) for a given width difference between Up and $Down$.

4.4.3 Voltage-controlled delay line and pulse combiner models

The voltage-controlled delay line is one of the key components in this frequency synthesizer; it produces multiple-phase signals which are combined to form the output signal by the edge combiner. A control signal coming from the charge pump adjusts the delay of each stage equally and tunes the total delay of all stages toward one period of the reference signal. The detailed circuit implementation and postlayout simulation is in Page 75~85.

In the Simulink library in Matlab, the *Variable Transport Delay*, which has 3 terminals, *In*, *Out*, *Delay*, is employed to construct a voltage-controlled delay line. The signal named *Delay* is the value of the desired delay time.

Since the total delay time of 9 delay stages should be equal to one period of the reference signal when the loop is in lock. A gain stage and an offset value should be introduced to match $V_{control}$ coming from the loop filter to the input requirement of the voltage-controlled delay line as shown in Equation (4.2). A VCDL model can thus be set up.

$$D_{control} = K_D \left(V_{control} - \frac{V_{max} + V_{min}}{2} \right) + Delay_0 \quad (4.2)$$

where V_{max} , V_{min} is the maximum and minimum values of $V_{control}$; $D_{control}$ is the total delay of the delay line and $Delay_0$ is its average value. Ideally $Delay_0$ is equal to the period of the reference signal. According to Equation (4.2), the tuning range of each delay stage in the delay line can be obtained by substituting V_{max} and V_{min} which is expressed as $Delay_0 \pm \frac{K_D(V_{max} - V_{min})}{2}$. The VCDL model is shown in Figure 4.4. So the main parameters for the VCDL are $Delay_0$, the average delay of the delay line, and K_D , the gain of the delay line. To get the delay of each delay stage, $D_{control}$ in Equation (4.2) is divided by the number of delay stages included in the delay line.

The output frequency of this frequency synthesizer is to be nine times that of the input signal. To make the circuit insensitive to the duty cycle of the reference signal, only the rising edges of the reference signal are used by the combination circuit. And it is achieved by combining pulses coming from each delay stage. For each stage, the pulse width should be equal to half of its delay time. Therefore each delay stage in the VCDL must be split into two delay cells and the pulse is obtained by a logical operation of the two delay cells outputs as shown in the models. The detailed analysis and explanation is given in the chapter followed.

With an “OR” logical operator, all pulses can be combined to generate an output signal whose frequency is nine times the frequency of the reference signal propagating in the delay stages. The model of the pulse combiner is also shown in Figure 4.4.

Figure 4.4 VCDL & pulse combiner models

4.4.4 Lock detector model

The lock detector in the frequency synthesizer serves to prevent the loop from locking to zero period or multiple periods of the reference signal. It is necessary because the phase detector compares the phase difference between two signals, but it is insensitive to the cycle differences. For example, if the phase difference between two signals is $2k\pi$, where k is an integer, the phase detector may treat as the case of zero phase error over one cycle. Therefore in this frequency synthesizer, lock detector is employed to solve this problem. It takes control of phase detector when the phase difference between the two inputs of the phase detector exceeds a certain range, namely the total delay of delay line, which is supposed to be the period of the reference signal when the loop is in lock. The working range of the lock detector is smaller than $0.9T$ or greater than $1.125T$, where T is the period of the reference signal.

Figure 4.5 Lock detector model

The implementation of the lock detector is shown in Figure 4.5, the detailed operating principle is discussed in the next chapter.

4.4.5 Top model of the frequency synthesizer

After all sub models were defined, they were interconnected as a complete frequency synthesizer, its block diagram shown in Figure 4.6. Since the phase detector acts

Figure 4.6 Top model of the frequency synthesizer

on the falling edges of its inputs while the rising edges of the signal are employed in the pulse combiner, to eliminate the effect of the variation of the duty cycle, two inverters are placed at the inputs of the phase detector.

4.5 Matlab simulation results

The complete matlab simulink model of a frequency synthesizer was set up in the previous section. A series of simulations were performed to verify their co-operative functionality.

With the test bench shown in Figure 4.7, the input and the output signals can be

Figure 4.7 Test bench of the frequency synthesizer

Figure 4.8 Wave form of $V_{control}$ during the locking process

observed with a simulink model *Time Scope*. To gain more control of the data of the input and output signals obtained from simulation, the signals exhibited by *Time Scope* were

also saved to the Matlab work space for subsequent plot. So that the locking process by viewing $V_{control}$ was also plotted.

Figure 4.9 Input/output waveforms when the lock is not in lock

The simulation conditions and the specifications are listed in Table 4.2. When the simulation began, $V_{control}$ was about 1.6V (one can choose other values as well) as shown in Figure 4.8. The total delay of the voltage-controlled delay line is far less than the desired value (one period of the reference) at the time. This can be observed on the input and output wave forms shown in Figure 4.9. At this time, the lock detector works to pull the $V_{control}$ down and increase the delay of the delay line.

TABLE 4.2: Parameters and specifications of Matlab simulation

Components	Parameters/Specifications	Values
Phase Detector	Type of phase detector	Linear PD
Lock Detector	!Over enabled range	Delay > 1.125T
	!Under enabled range	Delay < 0.9T
Charge Pump Loop Filter	Output (V_{control}) range	0 -- 1.8V
	Gain, K_{cp} (V_{control} /Timing error)	2E7/1.8(V/s)
Voltage-controlled Delay Line	Number of delay cells (2 cells for each delay stage)	9 x 2=18
	Maximum delay of each delay cell	0.6ns
	Gain, K_{D} (Delay/ V_{control})	-5.5556ns/V
	Average delay time, Delay_0	5ns
	Tuning range of the total delay	0ns--10ns
Frequency Synthe- sizer	Frequency of input signal (square wave with duty cycle of 50%)	200MHz
	High/low voltage of the input signal (square wave)	1.8V/0V
	Frequency of output signal	1.8GHz
	High/low voltage of the input signal (square wave)	1.8V/0V
Matlab Simulation	Step control method	Fixed step
	Time step of simulation	0.5 ps

With the decrease of $V_{control}$, the delay time increases. Once the total delay reaches a certain range, the lock detector returns control to the phase detector, and the fine adjustment of the $V_{control}$ is accomplished by the phase detector.

Ideally, the $V_{control}$ will be constant once the loop is in lock and 9 pulses will be evenly aligned within one period of the reference signal as shown in Figure 4.10.

Figure 4.10 Input/Output when the loop is in lock

4.6 Conclusion

A proposed architecture of DLL-based frequency synthesizer was introduced and a complete matlab simulink model of a 9-time DLL-Based Frequency Synthesizer was set up in this chapter. Matlab simulation shows that it works as expected. The next chapter describes the design and simulation of the DLL-based frequency synthesizer in transistor level.

CMOS Design of the DLL-Based Frequency Synthesizer

5.1 Introduction

A new structure of DLL-based frequency synthesizer has been described, one whose performance is demonstrably superior to that of an equivalent PLL-based one using a ring oscillator. This chapter describes the design, to the target specifications of Table 4.2, of an actual 9-time frequency multiplier to be implemented with TSMC CMOS 0.18 μ m technology. Each of the functional blocks is individually analyzed and simulated, and then a full post layout simulation is used to verify the functionality of the entire device.

5.2 Techniques of phase detector

Phase detectors (PDs), frequency detectors (FDs) and phase and frequency detectors (PFDs) are devices which generate an output signal proportional to the phase or frequency difference of two input signals. To compare the phase difference between the reference signal and the delayed signal, a phase detector is employed in this design.

The linear model for a phase detector is as shown in Figure 5.1,

$$V_d = k_d \theta_e + V_{do} \quad (5.1)$$

where k_d is the gain of the PD, and θ_e is the phase error between its two input signals, and V_{do} is the offset voltage or free-running voltage. Usually the ideal offset volt-

Figure 5.1 Characteristic of phase detector

age is 0 volt, especially for DLL-based systems. Normally, the response linearity described by Equation 5.1 fails outside of certain phase-difference bounds, the values of which specify the PD's operation range.

Phase (frequency) detectors are characterized by parameters such as operation range, gain, dependence of duty cycle, the maximum operation frequency, noise performance as well as power dissipation. Dead-zone behavior must be also considered when choosing a PD (see Figure 5.6). Over the years, many analog and digital PD options available have been established, making it possible to select a circuit which is 'ideal' for a given application.

All PDs are in a sense equivalent in that they may be considered multipliers. Among all the types of PDs, based on the characterizations, they can be broadly categorized into three types, combinational PDs, XOR PDs and edge-triggered PDs [2], [28].

1. Combinational PD

A common implementation of combinational PDs is the so-called Gilbert PD, which is essentially a typical implementation of four-quadrant multipliers as shown in Figure 5.2 [28]. The average output DC signal $V_d = (i_4 + i_6)R_1 - (i_3 + i_5)R_2$, where i_3 , i_4 , i_5 and i_6 is the channel current of M_3 , M_4 , M_5 and M_6 respectively. The two input voltages v_i and v_o are assumed to lie in the transistors' linear range ($|v| < \alpha$ millivolts) and

Figure 5.2 Gilbert multiplier circuit

keeps each differential pairs working in the linear region as shown in Figure 5.3. In this circumstance, the PD output signal is,

$$V_d = \frac{RI}{\alpha^2} \times v_i v_o \quad (5.2)$$

Where it is supposed that $R_1 = R_2 = R$. A transistor mismatch can offset the inputs v_i and v_o and ultimately V_d , as will a mismatch of R_1 and R_2 offset V_d . These facts militate against the use of Gilbert Multiplier PDs in DLLs, where zero output offset is usually highly desirable. The PD gain depends on the amplitudes of the input signals and is not solely a property of the circuits. Substantial static power consumption is another drawback. But it does have the advantages of high-speed operation and simple implementation.

Figure 5.3 Characteristic of differential pair

2. The XOR phase detector

When the amplitudes of input signals fed into Gilbert multiplier is larger enough that the multiplier is over-driven, the multiplier output will be saturated at either a positive value, corresponding to a logic high, or a negative value, corresponding to a logic low. At this time, the PD's linear operation range is larger than that of the Gilbert multiplier described before. The disadvantage is that the output DC signal V_d depends on the duty cycle of the input signals. In other words, when the phase error between the two input signals as well as the duty cycle of either two input signal changes, the output DC signal will increase or decrease correspondingly.

3. Dual D flip-flop PFD

A widely used phase detector is the edge-triggered phase detector. A simple edge-triggered PD can be implemented with a single reset-set (RS) flip flop. The output signal has only two states, one for each of the positive or negative phase error and this type of PD is often called a two-state phase detector. Since it is edge-triggered, its behavior is independent of the duty cycle of the input signals.

A similar device, the dual D-flip-flop PD, is built as shown in Figure 5.4 [3], where $\overline{V_d}$ is the average accumulated value of the difference between Q_U and Q_D . Its output behavior is independent of input duty cycle. Its operation is illustrated in the state diagram of Figure 5.5.

This PD acts on the rising edges of its inputs. Assuming the initial state is in state-2 ($Q_U = 0, Q_D = 0$), the state of PD changes upon the rising edge of f_i or f_o . Depending upon which of the rising edge of f_i or f_o occurs first, the state will change to state-3 or to state-1 and hence the PD indicates whether the phase difference between f_i and f_o is positive or negative by outputting a pulse at either of the Q_U or the Q_D . In the other words, if the rising edge of f_o comes first, the state changes to state-1 ($Q_U = 0, Q_D = 1$), then it comes back to state-2 ($Q_U = 0, Q_D = 0$) upon the rising edge of f_i . Consequently the state-

Figure 5.4 Dual flip-flop phase detector

3 shows that f_i leads f_o and state 1 shows that f_i lags f_o . So it can detect the phase difference between the two input signals.

According to the analysis above, since this PD acts on the rising edges only, there is no special requirement on the input duty cycle. Furthermore, the range of phase error under which the PD can work properly is $\pm 2\pi$, a range which is larger than that of most other PD types. Because of these advantages, the dual D flip-flop PFD is widely used.

Figure 5.5 State diagram for dual DFF phase detector

However, it has a disadvantage existing where the output of the PFD is employed to switch the source and sink current in a conventional charge pump, potentially the circuit can suffer from dead-zone problem which occurs when the f_i and f_o is nearly in phase, as shown in Figure 5.6.

Figure 5.6 Diagram of dead zone example

The problem originates with the non-zero rise and fall times of the Q_U and Q_D signals. If f_i leads f_o by time t_l , and the delay between Q_U and Q_D is also t_l . Suppose t_l is sufficiently small, then the source or sink transistor, as appropriate, may not be fully turned on, or not turned on at all. In other words, Q_U even does not fully reach the high level voltage, and Q_D reaches the threshold voltage of the AND gate, thus the reset in the PFD is activated, resetting Q_U and Q_D to 0. Therefore when t_l is very small, both the current source and the current sink in charge pump are not fully switched on, i.e. the output voltage $\overline{V_d}$ is neither increased nor decreased, and thus a dead zone is generated.

The problem is dealt with by holding Q_U and Q_D high simultaneously for a minimum amount of time during which both source and sink switches are on. To force Q_U and Q_D both high for the minimum time, a delay usually is inserted in the reset path to retard the reset action. Such a PD has four states instead of the three of Figure 5.5. It is described in the following section.

4. Other types of PD and design considerations

The Dual-DFF PD is not the only edge-sensitive one available. Other types of PD or PFD ever developed are mainly as follows, two XOR PD, Sample and Hold PD, Half-Rate PD, Bang-Bang PD [28], [31], [32], [33]. These edge-triggered devices are often used in clock and data recovery circuits. As analyzed in the last chapter, any PD used in a DLL must be such that its offset voltage is exactly zero; otherwise the offset voltage in PD when circuit is in lock condition can cause steady state phase error between the input signal and the delayed signal, which propagates through the delay stages, and creates output jitter and cannot be compensated. This requirement imposes restrictions on the types of PDs acceptable for use in DLL synthesizers.

5. Phase detector design and simulation

In a PLL-based frequency synthesizer, non-zero PD offset voltages can be tolerated, but as was explained above, this is not true when one is working with DLL-based synthesizers. A PD output offset of exactly zero is not just desirable, but necessary, and this in turn implies that the PD used must not have the dead-zone problem described earlier.

As shown in Figure 4.2 on page 50, the phase detector used in this design of this thesis is a dual D Flip-flop device which responds to falling input edges, and the flip-flop can equally well be made of NOR or NAND gates [29]. The later option was chosen for this work and it is therefore falling-edge triggered.

In a DLL-based structure, when the loop is locked, the PD output signal UP and DN should have exactly the same width. So it is required to make the path delays of the two input signals (Clk-ref and Clk-delay instead of f_i and f_o) of PD the same with the same delay and load, and the paths of two signals are exactly symmetrical. The PD senses the phase difference by detecting the falling edges of Clk-ref and Clk-delay, and generates two negative pulses whose width difference is proportional to the phase difference of the

two input signals. From the schematic, the starting points of UP and DN are decided by the two falling edges of two input signals independently. Once the two input signals are both high, the reset path will reset the UP and DN signals to '0' after a small delay as shown in the state diagram in Figure 5.7.

Figure 5.7 State diagram for PD without dead-zone

Suppose the two inputs (Clk-ref and Clk-delay) have a small frequency difference, acting on their falling edges, the phase detector produces the outputs UP and DN with width difference proportional to the phase difference of its 2 inputs as shown in Figure 5.8

According to the post layout simulation results, the width difference between UP and DN varies when the phase difference changes. This relation is shown in Figure 5.9.

5.3 Charge pump and loop filter functions

5.3.1 Charge pump

The charge pump is a necessary component in DLL-based applications. The PD provides two output signals (UP and DN) to indicate the sign of the phase difference between its inputs. This information is used by the charge pump, which is connected as shown in Figure 5.10. The CP works as an integrator and converts the currents switched by signals UP and DN to the control voltage used by the VCDL. When the loop is locked,

Figure 5.8 Operation principle of phase detector

Figure 5.9 Post-layout simulation for phase difference vs. width difference between UP & DN

this control voltage will be constant and such that the sum of the VCDL stage-delays is equal to one period of the reference signal.

The charge pump and the loop filter work together to convert the “digital” phase difference time into an analog voltage to tune the VCO in a PLL-based synthesizer or the VC DL in a DLL-based synthesizer [27]. The advantage in the CP/LF combination not only lies in its elimination of the risk of false lock [34], but also its ability to make the capture range of a synthesizer dependent only upon the VCO output frequency in the case of a PLL, or to make the output frequency range depend only upon on the dynamic range of the VC DL in the case of a DLL.

Figure 5.10 shows a simplified diagram of charge pump/loop filter combination. The charge pump operation can be described simply as follows. Assertion of the UP signal

Figure 5.10 Simple diagram of charge pump and loop filter

causes charge to be injected into the LPF output capacitance, raising the output voltage, while the assertion of the DN signal brings about withdrawal of charge from the same capacitance and thereby lowers the output voltage. The voltage increase when a current is injected into a single capacitor C for a period of time Δt is

$$\Delta v = \Delta t(I/C) \quad (5.3)$$

In this equation, I may be I_{cp1} (causing voltage to increase) or I_{cp2} (causing voltage to decrease), or a sum of the two. The equation is also linear in time.

5.3.2 Design considerations

The dead-zone problem which occurs when the two inputs are in phase, or nearly so, may be eliminated by the simultaneous assertion of UP and DN for the (short) time needed to bring the S_1 and S_2 switch gates up to threshold (assuming that they are equal). Ideally there will be no capacitor voltage change when this is done and the loop state will be maintained. However, there are further problems for the practical circuits.

Even in a monolithic circuit, one can expect device mismatches which lead to non-negligible mismatches in the two currents I_{cp1} and I_{cp2} , so that the output voltage of the LF will not correspond exactly to the difference of input phases. This will bring about phase noise in the DLL synthesizer output in the way previously described [30]. The present design deals with this difficulty by using a charge pump which employs multiple current mirrors. Design options for the component mirrors include the three of Figure 5.11.

The simple current mirror of Figure 5.11-A requires few transistors for its implementation, but it has low output impedance (undesirable in a current source) and poor matching of input and output currents [34].

Figure 5.11 Typical current mirrors

Figure 5.11-B is one of a cascode mirror which uses ‘active’ degeneration. Unlike the previous device, it has a high output impedance and good input/output current match-

ing, but its structure, two current mirrors in series, causes it to suffer from a limited output voltage swing. Hence the range of VCDL tuning voltage will be small, and the VCDL designed for small control-voltage swings tends to have poor control-voltage signal-to-noise ratios, or if a VCDL having a high SNR is used, the DLL dynamic (tuning) range will be limited.

The small voltage-swing problem can be alleviated if resistive degeneration is used, as in Figure 5.11-C, a source degeneration current mirror. Both high output impedance and large output swing are thus obtained, making this structure the most favorable of the three for this design (The resistors shown are actually physically implemented with transistors).

The schematic of the charge pump employed in this design is shown in Figure 5.12.

Figure 5.12 Schematic of charge pump

M₁₁ produces the bias voltage and current I_{bias} for the entire circuit. M₉, M₁₀, M₁, M₂ form the current mirror which supplies the current I_{cp1} (which is equal to I_{bias}). The currents through M₅ and M₆ are held equal to I_{bias} by the mirror comprised of M₉, M₁₀, M₅, and M₆. Since the transistors M₅, M₆, M₇, and M₈ are in series, the current passing

through them are equal. I_{cp2} (again equal to I_{bias}) is produced by the M_7 , M_8 , M_3 , M_4 mirror.

M_4 , M_8 functions as the degeneration resistors mentioned in connection with Figure 5.11-C, so do M_1 , M_5 and M_9 . When output voltage changes, the current through M_1 and M_2 , or M_3 and M_4 changes only very little, and the less the current changes, the less the steady state phase error caused by charge pump is.

M_1 and M_4 in this circuit are used as the current switches controlled by signals UP and DN coming from the phase detector, but since for phase detector, the UP is enabled when it is at high level, while for charge pump the M_1 is turned on when UP is low voltage, so an inverter is inserted in the UP path between the PD and the CP so that the signal will be active-high.

The post layout simulation results of the charge pump are shown in Figure 5.13 for current pump up with the width of UP 52ps and that of DN 50ps. Figure 5.14 shows simulation results for current pump down with UP of 50ps and DN of 52ps. The charge pump is seen to work properly with no more than 2ps width difference, meaning that the charge pump can cause no more than 2ps steady-state error in the loop.

5.4 Loop filter

The performance of most PLL applications depends in a critical way upon the loop filter, to a much greater extent than does that of a DLL application. This is especially true of DLL frequency synthesizers, since LF usually affects only the locking speed and the channel-switching speed; the channel switching speed of a DLL synthesizer can in fact be modified by placing an off-chip capacitor at the CP output in this design.

Figure 5.13 Post layout simulation for current pump up

$(t_{UP} = 52 \text{ ps}; t_{DN} = 50 \text{ ps})$

Figure 5.14 Post layout simulation for current pump down

$(t_{UP} = 50 \text{ ps}; t_{DN} = 52 \text{ ps})$

5.5 Operation and design of voltage-controlled delay line

The voltage-controlled delay line and the combiner are the most important components in the In DLL-based frequency synthesizer, their performance being the main determinant of phase noise and jitter behavior. It is constructed to propagate the reference signal with an identical desired value of each delay and the total delay depends on how many stages there are.

To minimize the effect of pattern repeating caused by the mismatching among the delay stages, the VCDL and the combination circuit are required to be designed as symmetrical as possible to minimize the difference among stages. In particular, great care must be taken to ensure this translational symmetry in the string of VCDL delay cells, and careful design is imperative in the case of the combiner as well - for example, a perfect VCDL is useless if the combiner input thresholds or setup times vary in an unpredictable way.

5.5.1 Capacitive model of CMOS transistors

Figure 5.15 shows the capacitive model of a MOS transistor (MOSFET). There are four terminals: S (source), D (drain), (G) gate and (B) substrate. Parasitic capacitances

Figure 5.15 MOSFET parasitic capacitances

between the various device elements exist: C_{GB} (between gate and substrate), C_{GD} (gate, drain), C_{GS} (gate, source), C_{BS} (substrate, source), and C_{DB} (drain, substrate). These

capacitances lead to signal propagation delays in transistor circuits which are usually undesirable, but occasionally necessary, as in the case of the delay stage (delay cell).

5.5.2 Delay cell design

Many types of delay cell have been developed over the years and they are widely used in a number of applications, most importantly VCDLs of DLLs, ring oscillator of PLLs and so on (The delay stage will be studied in the context of the VCDL below). Poor-quality delay cells, or mismatching between cells caused by non-uniform characteristics in an array of them, will lead to duty cycle problems and a resulting degradation of synthesizer performance.

Figure 5.16 shows a few types of common delay cells [34], [35], [36], [37], among which (a), (b), (c) are inverter-based delay cells, and (d), (e), (f) are differential delay cells.

One important advantage of the differential delay cell is its ability to reject power supply noise. However, this type of delay cell normally has smaller voltage swing than inverter-based type, resulting in it having higher sensitivity to the noise coming from its load than that of inverter-based types. These considerations have led to the choice of inverter-based delay cells for the present work.

Figure 5.16-A is just an ordinary inverter - an operation model is a pair of switches ('on' when $V_{GS} > V_T$ and 'off' when $V_{GS} < V_T$, V_T being the threshold voltage). It is readily understood that transistor can be modelled as switches with infinite off-resistance when $V_{GS} < V_T$ and a finite resistance when $V_{GS} > V_T$. When V_{in} is high and equal to VDD, the N [P]-transistor is turned on [off], the situation of Figure 5.17-A1, and the capacitor voltage behaves as shown in Figure 5.17-B1. The opposite situation is presented in Figure 5.17-A2 and B2. When V_{in} is low and at the same voltage level with ground, the N-transistor is off while P-transistor is on (Figure 5.17-A2), and the output is charged to low (Figure 5.17-B2). This output is fed into another inverter which may have same character-

Figure 5.16 Schematic of typical delay cells

istics and its VTC is shown in Figure 5.17-C, and further delay is thus introduced. This procedure is repeated for each inverter within the delay string, resulting in a large delay.

According to the VTC of an inverter, both the inverter input and output voltage swing are equal to the supply voltage if the interval between the transition edges of the signal is big enough. Since the logic level is independent of transistor sizes, so they can be minimized.

In DLLs, the delay line is controlled by a voltage tuned by the phase difference between input signal and output signal. Two ways of voltage-controlling the delay of the inverter delay cell just discussed are illustrated in Figure 5.16-B and Figure 5.16-C. The topology of Figure 5.16-B is called an output-switch scheme, and that of Figure 5.16-C a

Figure 5.17 Behavior of a simple inverter

power-switch scheme [9]. More delay is introduced in both cases simply because the capacitor charging or discharging current passes through two MOS channels instead of just one, and the RC time constant of the delay element is at least doubled, and the maximum output frequency is smaller than the inverter shown in Figure 5.16-A for the same transistor size. In addition, The resistance of the extra channels, and therefore the delay cell RC time constant, can be further increased by adjusting V_c . However, the behavior of C is more ‘predictable’ than that of B because the control voltages are with reference to stable VDD and ground levels (see paragraph following Figure 5.16).

Compared with structure in Figure 5.16-B, the maximum frequency of structure in Figure 5.16-C is closer to that of the single inverter since the two control transistors neighboring to VDD and ground respectively, consequently reducing the impact of the control signal on the inverter during charging and discharging and less likely to be modulated by the control voltage itself. The output swing of Figure 5.16-C will also be greater than that of

Figure 5.16-B, so it is the latter circuit topology that was chosen for delay elements in this project. The attention paid to output swing is justified by the fact that CMOS18 VDD is only 1.8 volts. Finally, the delay of an single inverter is insufficient for the 9-stage VCDL/ 180MHz input frequency of this project, so multiple inverters are used for each delay cell.

As in some DLL examples reported thus far, the 2-stage inverter-based delay cell, consisting of two of the structures of Figure 5.16-C, is also used in this design as shown in Figure 5.18.

Figure 5.18 Schematic of the delay stage

According to analysis in the Chapter 3, the edge transition time should be as small as possible to minimize the effect of the noise coming from the output node of each delay stage. This is achieved by inserting an output buffer after each of the cell's two voltage-controlled delay structures, as shown in the figure. Actually two buffers can be seen in the figure - the purpose of these buffers is to minimize the effect of the load of each delay cell, and the use of two inverters in each buffer instead of just one is to compensate duty-cycle variation. This compensation for duty-cycle variation is important in DLLs, since a phase error introduced at one delay stage is propagated unchanged through the remainder, and

Figure 5.19 Post layout simulation for delay cell

multiple phase error accumulated in the circuit, resulting in a very big variation of the duty cycle.

Figure 5.19 shows the post layout simulation results of the diagrammed delay cell. It can be seen that transition times have been minimized by the output buffer and there is almost no duty-cycle variation.

5.5.3 Control signal circuit (driver) design

It is observed that a delay cell needs two control signals in the voltage controlled delay line, while the loop filter only provides one output. A circuit is therefore needed to generate two signals from the one output.

Figure 5.20 shows the schematic of this circuit. The input control voltage is converted to a current first with M_3 and the a diode connected PMOS transistor M_1 converts

Figure 5.20 Schematic of driving circuit

Figure 5.21 Post layout simulation for drive circuits

the current to the control voltage $V_{\text{control}+}$. At the same time, the current flowing through M_1 is mirrored to M_2 and passed to another diode-connected NMOS transistor M_4 , where it is converted to the control voltage $V_{\text{control}-}$. A resistor is placed in parallel with M_3 to keep the current flowing through M_1 non-zero while the input voltage is below the threshold voltage of NMOS transistor M_3 .

The relation between the output voltages ($V_{\text{control}+}$ and $V_{\text{control}-}$) and the input voltage (V_{control}) of the drive circuit can be observed from the post layout simulation shown in Figure 5.21.

5.5.4 Voltage-controlled delay line with drive circuit

In the previous sections, a voltage-controlled delay cell and its drive circuit were designed. To multiply the reference frequency by nine, there must be nine delay stages, the sum of whose delays is locked to one period of the reference signal. The schematic of the voltage-controlled delay line with its drive circuit is shown in Figure 5.22. Additional two delay stages are used as an input buffer and another delay stage at the end of the delay chain acting as a dummy load to balance the load besides nine delay stages mentioned above, so the delay difference among nine delay stages can be minimized effectively. Since the output frequency is to be nine times the input frequency which is used to be 180MHz in this design, the ‘free-running’ delay time for each of the nine delay cells must be 1/9 the reference period which is about 617 ps. Delay-cell and drive-circuit transistors as well as the resistor were sized to provide the widest-possible tuning range for the specified DLL.

The dependence of VCDL stage-delay upon the control voltage is easily deduced: when the V_{control} is less than the threshold voltage V_{GS} of the transistor M_3 , the driver-generated voltage $V_{\text{control}+}$ ($V_{\text{control}-}$) is very high (low), resulting a very large delay because of the high channel resistances of the transistors. An increase of the V_{control} , leads to a $V_{\text{control}+}$ ($V_{\text{control}-}$) decreases (increases), thus the delay decreases. The delay attains its minimum value when V_{control} equals to vdd.

Since VCDL delays are very sensitive to transistors parameters process variations, VCDL simulations should be carried out at the parameter corner values, FF, FS, SS and SF in addition to typical parameter values. FF means fast PMOS and fast NMOS transistors. FS is fast PMOS and slow NMOS, and so on. Figure 5.23 shows the results of post-layout

Figure 5.22 Schematic of voltage controlled delay line

simulation using the typical one of these corners (TT) with the VCDL transistors carefully sized. The typical stage delay for the project VCDL is expected to be about 617ps (the

Figure 5.23 VCDL post-layout simulation

input frequency of 180MHz is used for simulation), so that the VCDL and drive-circuit transistors are sized in such a way that the chip-to-chip design lies in the range 450-

1000ps per stage to accommodate most cases caused by the parameter variations as shown in Figure 5.27.

5.6 Design of lock detection circuit

The function of the lock detection circuit is to prevent the loop from losing lock and falsely locking to zero period or multiple periods instead of one period of the reference. It also eliminates the need for control-voltage initialisation. In the model described in the last chapter, there are two lock-detector outputs, *Over*, which indicates that the VCDL delay exceeds some value comparing with the period of the reference and the control voltage should be increased, and *Under*, which shows a VCDL delay that is too small and V_{control} should be decreased. The delay range within which *Over* and *Under* functions is analyzed in the following paragraphs.

The delay-stage output phase alignment when the loop is in correct lock is diagrammed in Figure 5.24. Signal p_0 is the output to the first delay cell, and P_9 is the buffered output of the 9th delay stage. It can be seen from the figure that when the 9-stage DLL is locked, outputs P_1 , P_2 , P_3 and P_4 is ‘low’, and P_5 ‘high’ if the delay between P_0 and P_9 is one period of the reference signal.

More generally, if the duty cycle of the signals propagating through this VCDL is 50%, the condition under which the outputs P_1 , P_2 , P_3 and P_4 are ‘low’ is that the total delay of 4 delay stages is smaller than $T/2$, where T is the period of the reference signal, or the total delay of the delay line should be smaller than $9T/8$. Similarly, the condition under which the P_1 , P_2 , P_3 , P_4 and P_5 are ‘low’ is that the total delay of 5 delay stages is smaller than $T/2$, or the VCDL delay is smaller than $9T/10$.

According to the conclusion above, the output signal *Over*, which means the delay is too much can easily can be expressed,

$$Over = \overline{P_{s1} + P_{s2} + P_{s3} + P_{s4}} \quad (5.4)$$

Figure 5.24 In-lock timing relationship

where P_{s1} , P_{s2} , P_{s3} , P_{s4} are values of P_1 , P_2 , P_3 , P_4 sampled at the rising edge of P_0 . If the delay is neither equal to one period nor it is too much as expressed in Equation (5.4), the delay is considered as too small. Consequently, the signal *Under* can be written as,

$$Under = \overline{\overline{OVER} \bullet P_{s5} \bullet OVER} \quad (5.5)$$

As a result, *Under* flags delays less than $8T/9$, and *Over* those above $10T/9$. Neither of them being enabled indicates a VCDL delay should be in the range $8T/9$ to $10T/9$.

If the duty cycle of the signals propagating through the delay line is not exactly 50%, the switching threshold of *Under* or *Over* will not be exactly equal to $8T/9$ or $10T/9$ respectively. In this case, proper loop operation is ensured by defining the switching threshold of *Under* to be a value less than T , and that of *Over* to be larger than T . This is equivalent to requiring that the input signal duty cycles lie in the range from 44% to 56%.

Figure 5.25 Schematic of lock detection circuit

Figure 5.26 Schematic of sample & hold

Figure 5.25 is the schematic of the lock detection circuit corresponding to Equation (5.4) and Equation (5.5). Figure 5.26 is the schematic of one of the sample-and-hold (S/H) circuits indicated in Figure 5.25. Five of these S/H circuits are used to sample the delay stage outputs at the positive transitions of the reference clock, and these samples are combined in accordance with Equation (5.4) and Equation (5.5) to produce the outputs *Over* and *Under*. A working test bench for the simulation was assembled by adding circuit elements of a VCDL to generate the delayed signals $P_1 - P_5$ as shown in Figure 5.27. From

Figure 5.27 Post layout simulation for operation of lock-detecting circuit

the figure, when the control voltage is very low (large), the delay of each delay stage is too large (small), so the lock detector output signal *Over* (*Under*) to pump up (down) the control voltage until the delay is within the range for phase detector, and both *Over* and *Under* are 'low' enabled.

5.7 Edge combining circuits

The edge-combining circuit (edger combiner) uses VCDL delay stage outputs to create an output signal having a frequency that is a multiple of the reference signal's frequency. This circuit can be considered as having two components, one which generates a succession of multiple phase pulses from the signals in the delay line, and the other one is the actual combination circuit combining those phase pulses.

Figure 5.28 VCDL pulse generation

The pulse width from the pulse combination circuit should be half of a VCDL delay stage time. The delay-stage schematic of Figure 5.18 shows that each delay stage is made of two identical inverting delay cells connected in series, and it turns out to be convenient to produce the required pulse via combination of the delay cells outputs with a logical operation summarized in Figure 5.28, and its implementation is illustrated by the following Figure 5.29. Each delay stage (stage $n+1$) has two outputs: p_n , to carry the

Figure 5.29 Pulse-generation circuit for combining

VCDL signal to the next stage, and P_{outn} , the actual delay-cell output, pulse for combining. The delay-cell output pulse is only determined by the rising edges of the signals propagating through delay stages, so it is insensitive to the variation of the duty cycle.

After having multiple phase pulses, the final step is to generate the high-frequency output by adding all pulses together. A ‘wire-OR’ configuration is used for this purpose because its short signal path and symmetry characteristics result in a minimal phase-noise contribution as shown in Figure 5.30.

Figure 5.30 Pulse combining circuit

The post layout simulation results for the edge combiner are shown in Figure 5.31.

5.8 Top level post-layout simulation

The top level schematic of synthesizer is presented in Figure 5.32. The inverters shown in the diagram are used for level-shifting roles. There are two inverters at the inputs of phase detector to convert rising edges to falling edges. Simulations of this structure were done to produce plots of the parameter changes with the overall frequency synthesizer. With the reference frequency of 180MHz and a power supply of 1.8V, the locking process and the output signal are observed as follows.

Figure 5.31 Post-layout simulation of the edge combiner

5.8.1 The locking process

The lock detector enables the loop to acquire lock no matter what the value of the initial VCDL control voltage is. In this section, two simulations were done, one with an initial V_{control} of zero and another which started with an initial V_{control} of 1.8V (V_{dd}).

Figure 5.32 Schematic of top level synthesizer

Figure 5.33 and Figure 5.34 show the respective simulation results. In the first case, the VCDL delay time starts out at a value that is far greater than one reference signal period. As expected, the lock detector outputs *Over* and takes over the control of phase detector (which could itself cause the loop to lock onto a non-integral reference signal period at this point), and $V_{control}$ is increased under the lock detector's supervision until the VCDL delay has fallen below $9T/8$. At this point, the lock detector will return loop control to the PD which makes the fine $V_{control}$ adjustments necessary to hold the DLL in lock, and then *Over* keeps in high level (See Figure 5.33).

The results of testing the opposite situation, wherein the initial value of $V_{control}$ is maximally high, are recorded in Figure 5.34. Analogous to that of the previous simulation, the lock detector assuming the PD's duties by negating *Under* until the VCDL delay has risen to $9T_{ref}/10$, and *Under* keeps in high level.

Figure 5.33 Post-layout locking process with initial $V_{\text{control}} = 0$

Figure 5.34 Post-layout locking process with initial $V_{\text{control}} = V_{\text{dd}}$

5.8.2 Synthesizer output of the locked loop

When the loop is locked, the output signal and the reference signal in time domain from post layout simulation are shown in Figure 5.35. As expected, each nine pulses are evenly aligned in each period of the reference signal. Thus the output frequency

Figure 5.35 Time-domain reference signal/output signal

Figure 5.36 In-lock timing error between VCDL input/output

(1.62GHz) is nine times the reference frequency (180MHz). Based on the synthesizer output from post layout simulation, the in-lock timing jitter and period jitter were calculated and plotted with Matlab as shown in Figure 5.36 and Figure 5.37 respectively.

The phase difference between VCDL input and output should be zero in theory, but in practical circuits there is always some error present between them. From Figure 5.36, the maximum error being about 5ps.

The period jitter of the output signal is shown in Figure 5.37. It is very evident that the period jitter is repeated for each 9 cycles with a peak-peak value of about 7ps.

Figure 5.37 Post-layout simulation of the output period jitter

5.9 Summary

In the chapter, a complete DLL-based frequency synthesizer was designed at the transistor level, and the post-layout simulations done to verify the function of each component, then of the complete frequency synthesizer. In addition to being simulated, this design was fabricated. The next chapter is taken up with the actual chip test details.

Test Setup and Performance Evaluation

6.1 Introduction

The 9-time frequency synthesizer described in the previous chapters was fabricated in CMOS 0.18 μ m technology and was packaged in 24-leaded CFP (ceramic flat pack). Some measurements have been made to verify its performance:

- Phase noise performance
- Operating frequency range
- Power characteristics

The detailed test setup, test method, and test results are described in this chapter.

6.2 Layout considerations

Careful considerations must be taken in layout design. Layout was begun with the phase detector. To minimize the in-lock error of the PD, the two paths from *Clkref* to *Up* and from *Clkdly* to *Down* should be identical as far as possible, so in the layout as shown in Appendix B, the placement of logic gates was done symmetrically to ensure that the metal paths connecting gates are of equal length. Design rules of MOS transistor matching and placement must be adhered to. For example, the metal paths are routed in a way that they do not cross the active gate areas in case of forming insymmetric parasitic capacitance among matched transistor [38].

Next came the charge pump, where as much careful considerations must be taken as in the case of PD. In addition to constructing the two charging and discharging paths so that they are symmetrical, the circuit is surrounded with double guardrings to reduce the effects of the voltage fluctuation caused by the electromagnetic coupling in the wafer.

Symmetrical layout is as important for the operation of the delay line and of edge combiner (shown in the appendix) as for the above two devices described.

A guardring is placed around each component block as well as around the entire frequency synthesizer circuit to suppress electromagnetic interaction of the component blocks which is possibly produced from the dynamic current and voltage. To eliminate the coupling between signals, multiple ground pads are aligned between critical signal lines, i.e. *Clkin*, *Clkout* and *Cap*. For more flexibility, the VCDL has independent power supply connections, but in actual measurements reported in this thesis, the VCDL used the same power supply as the other components.

To mitigate the effect caused by supply noise, capacitors was placed on the chip with the capacitance as large as possible. Six metal layers were used to form the capacitor and to maximize the capacitance. The capacitor structure is shown in Figure 6.1, and the estimated total capacitance is about 20pF with a area of about 0.05mm².

Figure 6.1 Structure of the capacitor layout

6.3 Floorplan and package pinout

Figure 6.2 Floorplan of the frequency synthesizer

The entire DLL-based frequency synthesizer described in the previous chapters was fabricated on an $800\mu m \times 800\mu m$ silicon wafer with the active area of about $300\mu m \times 200\mu m$. An output buffer made up of CMOS inverters was included to enhance the driving capability of the circuit. The ‘floor plan’ of the design is shown in Figure 6.2. It includes all the DLL components described in this thesis: the phase detector, the lock detector, the charge pump, the voltage-controlled delay line, the edge combining circuit and the output buffer. Note the large area is covered by the power-supply filter capacitor. The pad arrangement is shown in Figure 6.3. This DLL-based frequency synthesizer described in this thesis is ‘Circuit Design 1’ as shown in Figure 6.3. The ‘Circuit Design 2’ section is occupied by a circuit that was designed by another member in our research group.

There are no ESD protection circuits on any pad, so extreme care is necessary when handling and testing the device. This is especially true of the pad connected to the

Figure 6.3 Pad arrangement

input signal *Clkin*, since it is directly connected to the gate of a CMOS inverter without any discharging path. The very low capacitance of this structure makes it susceptible to destruction by even a small static charge.

The design is packaged in a 24-leaded CFP (ceramic flat pack) which has a small cavity (3.5mm x 5.5mm) and is suitable for high frequency (up to 2GHz) designs. The bonding diagram is shown in Figure 6.4.

6.4 Testbench

The packaged chip is mounted on a Printed Circuit Board Test Fixture (PCB-TF) developed by Canadian Microelectronics Corporation (CMC) for integrated circuit testing at clock speeds up to approximately 2 GHz in a 50 ohm controlled impedance environment. The fixture accepts a 24-pin CFP, twelve signal lines from which are routed to SMA connectors, eight lines directly to ground, and four selectable lines to power sources or ground.

Figure 6.4 Bonding diagram of the frequency synthesizer

However the package doesn't fit the PCB-TF above exactly. Inexperience with custom design operations involving the PCB-TF led to a small degree of incompatibility between it and the CFP developing during the process of layout and package design. Fortunately, the only problem was with power-supply connection *vdd2* - it was taken to a pin connected to ground by the PCB-TF instead. Two simple remedies were possible: pin *vdd2* not being soldered to the PCB-TF and instead connected to pin *vdd4* outside the package, or *vdd2* being left open-circuited since there is a connection between *vdd2* and *vdd4* inside the chip. The PCB provided by CMC is thus applicable in this design. The PCB-TF and the connections between package and PCB are shown in Figure 6.5. All 4 *vdd* pins are wired together by the PCB and some additional wires. The CFB terminals brought out to the PCB connectors are:

- *VDD1*-- the DC power supply(1.8V) for the circuit
- *Clkin1*-- input port of the reference signal
- *Clkout1*-- port of the output signal.

Figure 6.5 PCB connection illustration diagram

In the test process, a DC power supply is used to provide the 1.8V DC supply voltage. The reference signal was produced by an RF signal generator, and the output is measured by a spectrum analyzer. Equipment designations are listed in Table 6.1. The top level test setup is shown in Figure 6.6. There is an adjustment at the output of signal gen-

Figure 6.6 Top level test setup

erator to match the DLL DC level because the DC level of the signal generator is zero while the required DC level for the tested chip is about $V_{DD}/2$ (0.9V). Changing this bias DC voltage can actually change the duty cycle of the signal inside the VCDL since the output signal of the signal generator is sinusoidal wave not a ideal square wave. An oscilloscope is used to observe the input and output waveforms. To measure the current consumption, a multimeter was also used in the test process.

TABLE 6.1: Test equipment

No.	Name and function	Quantity
1	Rohde&Schwarz Signal Generator with output 5K~6GHz	1
2	HP 3630A DC Voltage Source, 0--6V	2
4	HP8564E Spectrum Analyzer	1
5	ZFBT-6GW DC level adjustment circuit	1
6	0~1GHz Oscilloscope	1
7	Multimeter for current measurement	1

6.5 Measured noise performance

The main advantage of the DLL-based frequency synthesizer over the PLL-based frequency synthesizer is its phase noise performance. In this part, phase noise was measured with the reference signal supplied by an RF generator. To better evaluate the output phase noise performance of this frequency synthesizer, the phase noise of the input signal (Figure 6.7-A) was measured as was the output phase noise shown in Figure 6.7-B. It shows that the output phase noise is only 8dBc/Hz higher than the input phase noise at an offset frequency of 100kHz. Figure 6.8 and Figure 6.9 show that the phase noise measurements of -104.67dBc/Hz and -105.5dBc/Hz are achieved corresponding to the input reference signal of 190MHz and 230MHz respectively at the offset 10kHz. For the carrier frequency of 1.71GHz, the phase noise is -108.67dBc/Hz at the offset 25kHz.

Figure 6.10 shows the output phase noise when the output frequency is 2.16GHz. At a frequency offset of 10kHz, the measured phase noise is -104.5 dBc/Hz. Due to the

Figure 6.7 Measured phase noise of the reference signal and the output signal

Figure 6.8 Measured phase noise at carrier frequency of 1.71GHz

nature of the synthesizer structure, the DLL-based frequency synthesizer shows much better phase noise performance than a PLL-based one. But in DLL-based ones, the sideband spur at the output is of concern. As shown in Figure 6.11, the sideband spur at output frequencies of 1.385GHz and 1.575GHz is measured as -27.33 dBc/Hz and -31.33 dBc/Hz respectively. So compared with the PLL-based ones with about usually less than -60 dBc/Hz, improvement effort is called for in this regard.

Due to the lack of advanced equipment, the output waveform is measured using an oscilloscope having a maximum input frequency of 1 GHz. Consequently the maximum

Figure 6.9 Measured phase noise at carrier frequency of 2.07GHz

Figure 6.10 Output phase noise at output frequency of 2.16GHz

input frequency can be 111MHz, a value that is right at the boundary of the synthesizer's operation range, so poor output phase noise can be expected as shown in Figure 6.12. When the input frequency is about 100MHz, the in-lock error is relatively large. Firstly it is because being at the boundary of the operation range, the maximum delay of the delay line can just hardly approximate one period of the reference signal. Secondly, when the delay of the delay line is near the maximum value, the control voltage is very low, which means the voltage at the output of the charge pump is quite low and the difference between the sink current and source current is rather large, and the larger in-lock error is produced.

Figure 6.11 Output spectrum at output frequencies of 1.386 and 1.575GHz

Figure 6.12 Output waveform with input frequency of 50 MHz and 100 MHz

As for the graph on the left in the Figure 6.12-A, it shows the output waveform when the reference frequency is only about 50MHz, a value that lies completely outside of the operation range, and the waveform shows the delay is not enough at all. Figure 6.12-B shows the output waveform when the input frequency is 100MHz, which is just at the boundary of the operating range of this frequency synthesizer.

6.6 Operating frequency range

The operating frequency is mainly determined by the tuning range of the voltage controlled delay line, and this tuning range depends on its supply voltage. So changing the supply voltage can greatly affect the operating frequency range of this frequency synthesizer.

Figure 6.13 Operating frequency range

To completely analyze the operating frequency range, the maximum and minimum operating frequency were obtained by tuning the input frequency and observing the spectrum of the output signal when the supply voltage was changed. The frequency synthesizer should work so long as the power of desired frequency component is higher than that of its side-band spurs with frequency offset equal to the reference frequency. The relation between the supply voltage and the operating frequency range was measured and plotted as shown in Figure 6.13. With increasing supply voltage, the VCDL delay decreases and the average operating frequency increases. The average frequency when the supply voltage is 1.8V is about 180MHz which is the original aim of this design.

6.7 Power consumption

The consumption power is obtained by measuring the consumption current as a function of the reference frequency when the supply voltage is 1.8V. The result is a rather coarse depiction of the power characteristic, but one which shows a trend that increasing the reference frequency, the consumption current increases too. The reason for this is that both the VCDL and the output buffer are inverter based, so the larger the frequency, the larger the timing-average current needed to charge and discharge gate capacitances, and the greater the power dissipation. But the maximum current is 3.7mA, so this DLL-based frequency synthesizer actually consumes little power with a maximum value of 6.67mW.

Figure 6.14 Consumption current vs. the reference frequency

A summary of chip performance is listed in Table 6.2.

Comparing the results in the table with the reported performance listed in Table 4.1 on page 48, the measured phase noise performance in this design is improved and the side-band spur power is comparable with the most recently reported one which is -34 dBc at

the carrier of 96MHz (shown in [24]). The active area is very small and the power dissipation is less than 6.67 mW when the reference signal is no more than 180MHz.

TABLE 6.2: Measured chip performance summary

Input frequency range	65MHz~270MHz
Output frequency range (9x)	580MHz~2.43GHz
Power supply	1.5~1.9 V
Phase noise performance	
190MHz @ 10kHz offset	-104.67dBc
@ 25kHz offset	-108.67dBc
230MHz @ 10kHz offset	-105.5dBc
Sideband Spur @ 1.575GHz carrier	-31.33dBc
Power dissipation (100~180MHz)	< 6.67mW
Active die area	0.06 mm ²
Process	0.18 μ m CMOS

6.8 Summary

This chapter presents the layout considerations, test setups and measurement results. Partial description of the performance analysis and explanation of the results are also given.

7.1 Summary

Contemporary methods of frequency synthesis are predominantly derivatives of the Phase-Locked Loop (PLL). The increase of the operation frequency pushes engineers to explore more efficient techniques to achieve the ideal circuit performance. Two of the most important of these goals are the reduction of circuit output phase noise and of jitter, and its promise for accomplishing those ends has led to the Delay-Locked Loop (DLL) attracting much attention in recent times.

In this thesis, recent PLL and DLL designs are studied and then analyzed by simulation using the applied mathematics toolkit Matlab. The noise performance and phase-noise suppression capabilities of both system types are investigated and the outcomes presented. It is demonstrated that a DLL-based system shows behavior that is superior to that of an ‘equivalent’ PLL one.

A DLL frequency synthesizer chip was physically built using $0.18\mu m$ CMOS technology. New circuit structures, such as a ‘dual’ delay cell, having improved phase-noise and jitter characteristics were devised and evaluated in this device (The test VCDL conceptually has nine delay stages, but actually eighteen are used when the new delay cell is used). New designs also led to a reduction of the effects of circuit mismatches between functional blocks. Measurements were made with results showing that this DLL-based

frequency synthesizer has better phase noise performance than PLL-based ones. In a word, this thesis has succeeded both in analyzing the noise performance of PLLs and DLLs and in designing as well as implementing a DLL-based frequency synthesizer.

7.2 Suggestion for Further Research

This research work was one of the first activities of a program of DLL synthesizer research being undertaken by our research group. In the immediate future, group efforts will be directed at component-level improvements in DLL functioning.

Output timing jitter measurements have not been made because of the absence of adequate equipment in this Department, but spur measurements show that inter-period jitter and in-lock jitter are of serious concern. Effort should be applied to the reduction of systematic phase errors, ones mainly due to mismatch between delay cells and asymmetric conduction paths in the charge-pump phase detector. Performance of the charge pump of the present design performance can be improved if more-efficient structures are used. The encouraging z-domain models of PLL systems and DLL systems developed in this work can also be developed into tools for experimentation with frequency synthesizers in general, as well as with other application types.

Appendix A Schematics of the Frequency Synthesizer

Appendix A.1 Schematic of the 2-NAND gate

Appendix A.2 Schematic of the inverter

Appendix A.5 Schematic of the 3-AND gate

Appendix A.6 Schematic of the 4-NAND gate

Appendix A.7 Schematic of the 4-AND gate

Appendix A.8 Schematic of the 4-NOR gate

Appendix A.9 Schematic of the sample and hold circuit

Appendix A.10 Schematic of the lock detector

Appendix A.12 Schematic of the phase detector

Appendix A.13 Schematic of the delay stage

Appendix A.14 Schematic of the output buffer

Appendix A.15 Schematic of the voltage-controlled delay line

Appendix A.16 Top level schematic of the DLL-based frequency synthesizer

Appendix B *Layout of the Frequency Synthesizer*

Appendix B.1 Layout diagram for the 2-NAND gate

Appendix B.2 Layout diagram for the inverter gate

Appendix B.3 Layout diagram for the 2-NAND gate

Appendix B.4 Layout diagram for the 3-NAND gate

Appendix B.5 Layout diagram for the 3-AND gate

Appendix B.6 Layout diagram for the 4-NAND gate

Appendix B.7 Layout diagram for the 4-AND gate

Appendix B.8 Layout diagram for the 4-NOR gate

Appendix B.9 Layout diagram for the sample and hold circuit

Appendix B.10 Layout diagram for the lock detector

Appendix B.11 Layout diagram for the charge pump

Appendix B.12 Layout diagram for the phase detector

Appendix B.13 Layout diagram for the delay stage and the buffer

Appendix B.14 Layout diagram for the frequency synthesizer output buffer

Appendix B.15 Layout of diagram for the voltage-controlled delay line

Appendix B.16 Layout diagram for the frequency synthesizer

Appendix B.17 Die Diagram

References

1. B.Kim, T.C. Weingandt, P.R. Gray, “*PLL/DLL System Noise Analysis for Low Jitter Clock Synthesizer Design*”, Proceeding of ISCAS, June 1994.
2. J.Craninckx, and M.Steyaert, “*Wireless CMOS Frequency Synthesizer Design*”, Kluwer Academic Publications, 1998,
3. W.J.Jones, H.P.Benn; J.G. Gardiner; “*A Fast Hopping Frequency for Agile Radar*”, Circuits and Systems, 1988., IEEE International Symposium on, 1988.
4. Noel, D.P.; Kwasniewski, T.A.; “*Frequency Synthesis: A Comparison of Techniques*”, Electrical and Computer Engineering, 1994 Canadian Conference on, 1994.
5. Kuleshov, V.N.; Liu, H.Y.’ “*Fundamental Noise in Direct Digital Frequency Synthesizers*”, Proceedings of the 1995 IEEE International, 1995.
6. Cercas, F.A.B.; Tomlinson, M.; Albuquerque, A.A.; “*Direct digital frequency synthesizer for a FH spread spectrum system: analysis and design*”, ITS '90 Symposium Record., SBT/IEEE International, 3-6 Sep 1990.
7. Kenichi Tajima, H. Tsuru; H. Ikematsu; K. Itoh; Y. Isota; O. Ishida; “*Phase Decrement Type Direct Frequency Synthesizer Driven By A DDS*”, IEEE MTT-S Digest, 2001.
8. David J.Foley, “*CMOS DLL-Based 2-V 3.2-ps Jitter 1-GHz Clock Synthesizer and Temperature-Compensated Tunable Oscillator*”, IEEE Solid-State Circuits, 2001.
9. Ali Hajimiri, Sotirios Limotyrakis, and Thomas H. Lee, “*Jitter and Phase Noise in Ring Oscillators*”, IEEE Solid-State Circuits, 1999
10. John A. McNeil, “*Jitter in Ring Oscillator*”, IEEE Solide-state Circuits, June, 1997.

11. Alain Blanchard, "*Phase-locked Loops*", John Wiley&Sons, Inc., 1976
12. Shu, K.; Sanchez-Sinencio, E.; Maloberti, F.; Eduri, U.; "*A Comparative Study of Digital Sigma-delta Modulators for Fractional-N Synthesis*", The 8th IEEE International Conference on, Volume: 3, 2001.
13. Fangxin Wei, "*Submicron CMOS Building Blocks for High Speed Frequency Synthesis and Clock Recovery Applications*", M.Eng, Carleton University, 1997.
14. Adam Dong, "*Incorporation of digital angle-modulation into a fractional division synthesizer*", M.Eng, Carleton University, 1992.
15. Sun, L.; Lepley, T; Nozahic, F; Bellissant, A; Kwasniewski, T.; Heim, B.; "*Reduced complexity, high performance digital delta-sigma modulator for fractional-N frequency synthesis*", Proceedings of the 1999 IEEE International Symposium on Circuits and Systems, Volume: 2, Jul 1999.
16. Foroudi, N.; Kwasniewski, T.A.; "*CMOS high-speed dual-modulus frequency divider for RF frequency synthesis*", IEEE J. of Solid-State Circuits, Volume: 30 Issue: 2, Feb 1995.
17. Jaquet, Isabelle, "*A Comparison of phase noise control techniques used in high resolution high-speed frequency synthesizers for mobile radio transceivers*", M.Eng, Carleton University, 1990.
18. Kan, T.K.K.; Luong, H.C.; "*A 2-V 1.8-GHz Fully-integrated CMOS Dual-loop Frequency Synthesizer*", Digest of Technical Papers. 2000 Symposium on, 20.
19. James A. Crawford, "*Frequency Synthesizer Design Handbook*", Artech House Publishers, 1994.
20. Lizhong.Sun, "*High Speed Submicron CMOS Oscillators and PLL Clock Generators*", Ph.D, 1999.
21. T.A.D. Riley and M.A.Copeland, and T.A.Kwasniewski, "*Sigma Delta Modulation in Fractional-N Frequency Synthesis*," IEEE J.of Solid-State Circuits, Vol. 28, pp.553-559, May 1993.
22. Qinghong, Du, "*Delta-Sigma Fractional-N Synthesizers Enable Low-Cost, Low-Power, Frequency-Agile Software Radio*", Planet Analog, 2000.
23. Syed Irfan Ahmed, "*Submicron CMOS Components for PLL-based Frequency Synthesis*", M.A.Sc, Carleton University, 2002.

24. Sheng Ye, Lars Jansson, Lan Galton, "A Multiple-Crystal Interface PLL with VCO Realignment to Reduce Phase Noise", IEEE Solid-state Circuit Conference, Feb, 2002.
25. Sung-Sik Hwang; Ki-Mo Joo; Ho-Jin Park; Jae-Whui Kim; Chung, P.; "A DLL based 10-320 MHz clock synchronizer", The 2000 IEEE International Symposium on Circuits and Systems, 2000.
26. George Chien, Paul R. Gray, "A 900-MHz Local Oscillator Using a DLL-Based Frequency Multiplier Techniques for PCS Applications", IEEE Journal of Solid-State Circuits, 2000.
27. Chulwoo Kim, In-Chul Hwang; Sung-Mo Kang; "Low-Power Small-Area ± 7.28 ps Jitter 1GHz DLL-Based Clock Generator", ISSCC 2002, Nov, 2002.
28. S.Solaman, "An Overview of Techniques for CMOS Phase Detector", IEEE J. of Solid-State Circuits, 2001.
29. Mijuskovic, D.; Bayer, M.; Chomicz, T.; Garg, N.; James, F.; McEntarfer, P.; Porter, J.; "Cell-Based Fully Integrated CMOS Frequency Synthesizers", IEEE JSSC. vp;-SC29, pp. 271-279, March 1999.
30. M.Bayer; T.Chomicz; F.James; McEntarfer, P.; Mijuskovic, D.; Porter, J.; "A Low Noise CMOS Frequency Synthesizer with Dynamic Bandwidth Control", Poceedings of CICC, 1994.
31. J.Savoj and B.Ravazi, "A 10-Gb/s CMOS Clock and Data Recovery Circuit with a Half-Rate Linear Phase Detector", IEEE J. Solid-State Circuits, vol. 36. NO.5, May 2001.
32. M.Ramezani and C. Salama, "An Improved Bang-Bang Phase Detector for Clock and Data Recovery Applications", ISSCC Dig. Tech. Papers, vol.1, 2001,
33. J.Kang and D.Kim, "A CMOS Clock and Data Recovery with two-XOR Phase-Frequency Detector Circuit", ISSCC Dig. Tech. Papers, vol.IV, 2001.
34. David A. Johns; Ken Martin, "Analog Integrated Circuits Design", John Wiley & Sons, Inc., 1997.
35. Chen, Oscar T.-C.; Sheen, R.R.-B ; "A Power-Efficient Wide-Range Phase-Locked Loop", IEEE Journal of Solid-State Circuits, January, 2002.
36. Xavier Maillard; F., Devisch; M., Kujik; "A 900-Mb/s CMOS Date Recovery DLL Using Half-Frequency Clock", IEEE Journal of Solid-State Circuits, June, 2002.

37. Garlepp W., Bruno; Donnelly, K.S.; Jun Kim; Chau, P.S.; Zerbe, J.L.; Huang, C.; Tran, C.V.; Portmann, C.L.; Stark, D.; Yiu-Fai Chan; Lee, T.H.; Horowitz, M.A.; “*A Portable Digital DLL for High-Speed CMOS Interface Circuits*”, IEEE Journal of Solid-State Circuits, May 1999.
38. Alan Hastings, “*The Art of Analog Layout*”, Tom Robbins, 2001.