Low Spur Content DLL-Based Frequency Synthesizer Design

By

Jingcheng Zhuang

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

January 2004

© 2004 Jingcheng Zhuang

National Library
of Canada

Bibliothèque nationale
du Canada

Acquisitions and
Bibliographic Services

Acquisitions et
services bibliographiques

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

ISBN: 0-612-89879-2

Our file Notre référence

ISBN: 0-612-89879-2

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.

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

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de ce manuscrit.

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

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

The information used in this thesis comes in part from the research program of Dr. Tad Kwasniewski and his associates in the VLSI in Communications group. The research results appearing in this thesis represent an integral part of the ongoing research program. All research results in this thesis including tables, graphs and figures but excluding the narrative portions of the thesis are effectively incorporated into the 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

The frequency synthesizer is an important building block in most communication systems, whose performance is greatly affected by the phase variation of the frequency synthesizer output, including the random phase noise and spur-associated variation.

This thesis firstly reviews the characteristics of frequency synthesizers and a number of frequency synthesis technologies to date. Then noise performance of two typical frequency synthesis technologies, the phase-locked loop (PLL) based one and the delay-locked loop (DLL) based one, is analyzed theoretically in detail. The circuit components critical for the synthesizer performance are also discussed respectively. To suppress the spurious power level in the synthesizer output, a new architecture of a DLL-based frequency synthesizer is proposed, in which a novel phase comparison circuit, including a charge pump phase comparator and a lock detector, is employed. Compared to the conventional non-dead zone phase comparator, the novel charge pump phase comparator provides a high phase resolution and a small timing offset. With the lock detector, the phase comparison circuit can operate in an extended phase range. This DLL-based frequency synthesizer is thus expected to exhibit low spur content, low phase noise at its output and operate in a large frequency range. This frequency synthesizer is implemented in $0.18\ \mu\text{m}$ CMOS process. Its input frequency range is from 110MHz to 230MHz and its output frequency is nine times the input. The measured output spur power is -46.17dBc and the phase noise is -107.17dBc/Hz at 10kHz offset with the reference frequency of 173MHz from an RF signal generator, which is better than all other reported results to date and verifies the successful implementation of the design.

Acknowledgements

First of all, I would like to acknowledge my thesis supervisor, professor Tad Kwasniewski, for his guidance, kindness and support during the past two years.

I would also like to thank all staffs of our laboratory group for providing me an excellent environment.

Finally, I would like to thank my family, especially my wife, for their support, understanding and encouragement.

Table of Contents

Chapter 1: Introduction.....	1
1.1 Introduction.....	1
1.2 Contributions	3
1.3 Thesis organization	3
Chapter 2: Overview of frequency synthesis	5
2.1 Introduction.....	5
2.2 Characteristics of frequency synthesizers	6
2.2.1 Phase noise.....	6
2.2.2 Spurious power level	9
2.2.3 Operating frequency range.....	10
2.2.4 Frequency increment.....	10
2.2.5 Channel switching time	11
2.2.6 Power consumption.....	12
2.3 Some typical frequency synthesis techniques.....	12
2.3.1 Incoherent synthesis.....	12
2.3.2 Direct digital synthesis.....	13
2.3.3 Phase-locked loops	14
2.3.4 Delay-locked loops	15
2.3.5 Comparison of synthesis techniques.....	17
2.4 Summary	17
Chapter 3: PLL-based integer-N frequency synthesis technology.....	19
3.1 Introduction.....	19
3.2 Fundamental characteristics of PLLs.....	19

3.2.1	Operating principle of PLLs	19
3.2.2	Stability of PLLs	21
3.2.3	Noise performance of PLLs	25
3.2.3.1	External noise coming from reference signal	25
3.2.3.2	Internal noise coming from the VCO	27
3.2.3.3	Noise coming from the control signal of the VCO	28
3.3	Operating principle of PLL-based frequency synthesizers	29
3.4	Basic components of PLL-based frequency synthesizers	30
3.4.1	Voltage-controlled oscillator	30
3.4.2	Digital frequency dividers	32
3.4.3	Phase detector	35
3.5	Noise performance of the PLL-based frequency synthesizers	35
3.6	Low noise frequency synthesizer design	37
3.7	Summary	38
Chapter 4: DLL-based integer-N frequency synthesis technology		39
4.1	Introduction	39
4.2	Fundamental characteristics of DLLs	39
4.2.1	Basic operating principle of DLLs	40
4.2.2	Stability of DLLs	44
4.2.3	Noise performance of DLLs	44
4.2.3.1	External noise coming from reference signal	44
4.2.3.2	VCDL Internal noise	45
4.2.3.3	Internal noise coming from the control line	47
4.3	Operating principle of DLL-based frequency synthesizers	47

4.4	Basic components of DLL-based frequency synthesizers	49
4.4.1	Voltage-controlled delay line	49
4.4.2	Edge combining circuit.....	50
4.4.3	Charge pump phase comparator.....	50
4.4.4	Loop filter	50
4.5	Spur performance of the DLL-Based frequency synthesizers	51
4.6	Low noise frequency synthesizer design	61
4.7	Summary	61

Chapter 5: Proposed architecture of the DLL-based frequency synthesizer..62

5.1	Introduction.....	62
5.2	Comparison of PLL/DLL based frequency synthesis.....	62
5.3	System specifications.....	64
5.4	System block diagram of the DLL-based frequency synthesizer	64
5.5	Specifications of each block	65
5.5.1	Voltage-controlled delay line	65
5.5.2	Pulse combining circuit	66
5.5.3	Input buffer	66
5.5.4	Output buffer.....	67
5.5.5	Phase comparison circuit	67
5.5.6	Loop filter	68
5.6	Summary	68

Chapter 6: Voltage-controlled delay line and edge combiner69

6.1	Introduction.....	69
6.2	Structure of voltage-controlled delay line.....	69
6.3	Design of the voltage-controlled delay stage.....	70

6.3.1	Differential delay cells	71
6.3.2	Inverter-based delay cells	75
6.3.3	Design of delay stage	78
6.4	Design of driver circuit	79
6.5	Design of edge combining circuit	80
6.6	Simulation of the VCDL and edge combiner	81
6.6.1	Simulation of the driver circuit	82
6.6.2	Simulation of the delay stage	82
6.6.3	Simulation of edge combining circuit	82
6.7	Summary	83
Chapter 7: Charge pump phase detector and lock detector		85
7.1	Introduction	85
7.2	Some typical phase detectors	86
7.3	A widely used phase detector	87
7.4	A typical charge pump	88
7.5	Problems involved in the conventional circuitry	89
7.6	Some measures to overcome those problems	92
7.6.1	Reduction of control voltage glitches	92
7.6.2	Reduction of the current mismatch	93
7.6.3	Reduction of charge sharing	94
7.7	A novel charge pump phase comparator	94
7.8	Basics of the novel charge pump phase comparator	95
7.9	Lock detection and window generation circuit	96
7.10	Detailed design of the novel charge pump phase comparator	99
7.10.1	Pulse generator	99

7.10.2	Pulse integrator	102
7.10.3	Voltage comparator	104
7.10.4	Pulse control feedback circuit.....	104
7.11	Characteristics of the charge pump phase comparator	105
7.12	Simulation of the charge pump phase comparator.....	107
7.13	Summary	110
Chapter 8: Top-level simulation and layout consideration		112
8.1	Introduction.....	112
8.2	Top-level simulation of the frequency synthesizer	113
8.2.1	Locking process	113
8.2.2	In-lock error and output jitter.....	114
8.3	Simulation with pad and package models.....	116
8.4	Important layout issues	116
8.4.1	Matching issues of layout	116
8.4.2	Minimization of interference among blocks.....	117
8.4.3	On-chip power supply filter capacitors.....	117
8.4.4	Pad arrangement to minimize coupling noise.....	117
8.5	Post-layout simulation	117
8.6	Summary	118
Chapter 9: Test setup and performance evaluation		119
9.1	Introduction.....	119
9.2	Test setup	119
9.3	Measurement results	120
9.3.1	Phase noise performance	120
9.3.2	Spurious power level measurement	122

9.3.3	Summary of the measurement results	123
9.4	Summary	124
Chapter 10: Conclusions		125
10.1	Conclusion	125
10.2	Suggestions for future research.....	126
References		127
Appendix A: Schematics of the frequency synthesizer.....		130
Appendix B: Layouts of the frequency synthesizer		137
Appendix C: Package and PCB diagrams		142

List of Tables

TABLE 2.1:	Comparison of frequency synthesis techniques	17
TABLE 5.1:	Comparison of PLL/DLL for the frequency synthesis application.....	63
TABLE 5.2:	Specifications of the VCDL.....	66
TABLE 9.1:	Characteristics of the frequency synthesizer	124

List of Abbreviations

AC	Alternating current
AM	Amplitude modulation
BPF	Band-pass filter
CFP	Ceramic flat pack
CP	Charge pump
CPPC	Charge pump phase comparator
CMC	Canadian microelectronics corporation
CMOS	Complementary MOS
DC	Direct current
DDS	Direct digital synthesis
DFF	D flip-flop
DLL	Delay-locked loop
FM	Frequency modulation
LF	Loop filter
LPF	Low-pass filter
PCB	Printed circuit board
PCFB	Pulse control feed back
PD	Phase detector
PFD	Phase frequency detector
PG	Pulse generator
PLL	Phase-locked loop

PM	Phase modulation
RF	Radio frequency
MOS	Metal-oxide semiconductor
VC	Voltage comparator
VCDC	Voltage-controlled delay cell
VCDL	Voltage-controlled delay line
VCO	Voltage-controlled oscillator
VTC	Voltage-transfer characteristic

List of Symbols

Band-pass filter
Capacitor
Crystal
Ground
Resistor
Inductor
Low-pass filter
Mixer
Adder
Voltage-controlled oscillator
VDD
NMOS transistor
PMOS transistor
Current source
Inverter
Delay cell or output buffer

NAND gate

NOR gate

D Flip-flop

Switch

Voltage-controlled switch

List of Figures

Figure 2.1	Spectrum of an ideal signal and a practice signal	6
Figure 2.2	Spectrum of a narrow-band FM signal	8
Figure 2.3	Power spectrum of a FM signal	8
Figure 2.4	Power spectrum of a frequency source with spurs	9
Figure 2.5	An example of incoherent synthesis	12
Figure 2.6	A direct digital synthesizer	13
Figure 2.7	A PLL-based frequency synthesizer	14
Figure 2.8	A fractional-N PLL-based frequency synthesizer	16
Figure 2.9	Block diagram of a DLL-based frequency synthesizer	16
Figure 3.1	Block diagram of a PLL.....	20
Figure 3.2	Bode diagrams	23
Figure 3.3	Loop filters for PLLs	24
Figure 3.4	PLL model for VCO noise analysis	27
Figure 3.5	Block diagram of a PLL-based frequency synthesizer	30
Figure 3.6	A ring oscillator	31
Figure 3.7	Effect of AM-to-PM conversion in a frequency divider.....	32
Figure 3.8	Change of the power spectrum by a 10 divider	34
Figure 3.9	Phase noise of the input and output of a frequency divider.....	34
Figure 3.10	Noise performance of PLL-based frequency synthesizer	36
Figure 3.11	Optimum loop bandwidth	38
Figure 4.1	Block diagram of a basic DLL.....	40
Figure 4.2	Transfer functions of the DLL	43
Figure 4.3	Power gain for external noise versus the value of K	45

Figure 4.4	DLL model for VCDL internal noise analysis.....	46
Figure 4.5	Transfer functions of the DLL for internal noise.....	46
Figure 4.6	Multiple-stage VCDL and pulse combiner.....	47
Figure 4.7	Delay variation.....	51
Figure 4.8	Relative power spectrum of a frequency output ($N=10$)	52
Figure 4.9	Phase alignment error for case I	54
Figure 4.10	Relative power spectrum for Case I.....	55
Figure 4.11	Relation of spurious power level to the in-lock error	56
Figure 4.12	Phase alignment error for case II	57
Figure 4.13	Illustration of the spurious output.....	58
Figure 4.14	The relation of N , k_r vs. relative spurious level.....	59
Figure 5.1	System block diagram of the DLL-based frequency synthesizer	65
Figure 6.1	Structure of VCDL.....	70
Figure 6.2	A differential delay cell	71
Figure 6.3	A differential delay cell model	73
Figure 6.4	An inverter-based delay cell	75
Figure 6.5	Waveform of the output of an inverter delay cell	76
Figure 6.6	An inverter-based voltage-controlled delay cell.....	77
Figure 6.7	Schematic of the voltage-controlled delay stage	79
Figure 6.8	Schematic of the driver circuit.....	79
Figure 6.9	Pulse generator and pulse combiner	81
Figure 6.10	Schematic of the pulse combiner	81
Figure 6.11	Simulation result of the driver circuit.....	82
Figure 6.12	Simulation result of the delay stage.....	83
Figure 6.13	Simulation of edge combining circuit.....	83

Figure 7.1	A dual D flip-flop phase detector	88
Figure 7.2	Waveforms of the input and output of the phase detector	88
Figure 7.3	The principle of a typical charge pump	88
Figure 7.4	Glitch of the control voltage	91
Figure 7.5	The schematic of a typical charge pump.....	92
Figure 7.6	A measure to reduce the glitch	93
Figure 7.7	Charge pump to alleviate the mismatch between PMOS and NMOS	93
Figure 7.8	Charge pump to alleviate the charge sharing	94
Figure 7.9	Block diagram of the novel charge pump phase comparator.....	95
Figure 7.10	Operating principle of the lock detection circuit (1).....	97
Figure 7.11	Operating principle of the lock detection circuit (2).....	98
Figure 7.12	Simulation result of the lock detection circuit	99
Figure 7.13	Outputs of the lock detector and window generator	100
Figure 7.14	Schematic of the pulse generator	100
Figure 7.15	Operating principle of the pulse generator	102
Figure 7.16	Schematic of the integrators	102
Figure 7.17	Characteristics of the integrator	103
Figure 7.18	Schematic of the voltage comparator.....	104
Figure 7.19	Schematic of the pulse control feedback circuit	105
Figure 7.20	The relation between timing error and area error	106
Figure 7.21	Characteristic of the charge pump phase comparator	107
Figure 7.22	Test bench for charge pump phase comparator.....	108
Figure 7.23	Startup process of the charge pump phase comparator	109
Figure 7.24	Detailed operation of the charge pump phase comparator.....	109
Figure 7.25	Waveform of the output of the charge pump phase comparator	110

Figure 7.26	Transfer characteristic of the charge pump phase comparator	111
Figure 8.1	Top-level schematic of the frequency synthesizer	112
Figure 8.2	Simulation results of the locking process	114
Figure 8.3	Input and output of the frequency synthesizer	114
Figure 8.4	In-lock error calculated from the transistor-level simulation	115
Figure 8.5	The output periodic jitter	115
Figure 8.6	Pad arrangement of the chip	118
Figure 9.1	Test setup for frequency purity measurement.....	120
Figure 9.2	Power spectrum of the output signal.....	121
Figure 9.3	Phase noise measurement result	122
Figure 9.4	Spurious power level measurement	123

1.1 Introduction

The technique of generating one or a set of frequencies from one or more reference frequencies is called frequency synthesis, and the devices which carry out the process are called frequency synthesizers. High spectral purity in a synthesizer's output is always desirable but in communication applications the matter becomes critical. For example, the spectral purity of the local oscillator output has a strong effect on the channel spacing (density) which may be achieved. This is true at both the receiving and transmitting ends of an RF communication system. New methods of improving synthesizer spectral noise are consequently worth pursuing, especially in the case of DLL synthesizers, since the performance of these devices are not always superior to those which employ other methods.

PLL-based frequency synthesis has been a popular method for many years -- many related publications are found in the literature [1][2][3][4][5]. A rather different approach to the frequency synthesis problem, based on the DLL, has attracted more and more attention in recent years because of certain unique characteristics associated with the devices, important ones of which are unconditional stability and an absence of jitter accumulation

[7][8][10]. Actually, some designs [7][8][9][10] employing DLLs have been reported. In [10], a DLL-based frequency synthesizer has been detailed and its phase noise performance was confirmed by the actual measurement results to be better than that of many PLL-based ones. However, since it employs the conventional phase detector and charge pump which cause large in-lock error, its output exhibits relatively large spurious power level (-32dBc) and large output jitter. Similarly, the internal noise of the VCDL can not be effectively suppressed due to the limit of the loop bandwidth. Although it is designed to be insensitive to the duty cycle distortion, the duty cycle of its input has to be in a range from 44% to 56% due to the limitation of the phase comparison circuit.

In this thesis, the structure, principle as well as the performance of the DLL-based frequency synthesizer employing an edge combining circuit are analyzed in detail with emphasis on the power spectrum spur generation. A novel charge pump and phase comparator block is designed to effectively reduce the in-lock error while a large loop bandwidth is kept to reduce the spurious power level and enhance the phase noise performance [11]. This novel charge pump phase comparator working with a new lock detector results in a large operating range and a relaxed duty-cycle distortion requirement of its input (approximately 15% to 85%). A DLL-based frequency synthesizer employing those novel components was thus implemented and fabricated in CMOS 0.18 μ m technology. And its measurement shows the significant improvements on both spurious power level and phase noise performance [12] (better than any other reported designs to date) and it is able to operate under a very large duty-cycle distortion.

1.2 Contributions

The contributions of this work are as follows:

- An integer-N DLL-based synthesizer which employs an edge combining circuit is analyzed in theory and the key issues of its design are discussed. The predicted noise performance is compared to that of a ‘typical’ PLL-based structure.
- The effects of in-lock error and stage delay variation on the side-band spurs are analyzed and an approximate computation is done.
- A novel charge pump phase comparator is designed in this work to reduce the in-lock error and to obtain high loop bandwidth without introducing large output spurs.
- A lock detector used as a coarse phase detector to guarantee correct lock works together with the new charge pump phase comparator to provide a wide operating range and a high phase resolution.
- A DLL-based frequency synthesizer with an output frequency nine times the input frequency is proposed and implemented in CMOS 0.18 μ m technology. And the measurements taken from an operating chip shows a low phase noise and low spurious power level performance which is better than the ever reported results.

1.3 Thesis organization

In this thesis, there are ten chapters and the organization is as follows.

Chapter 2 reviews the typical frequency synthesis technologies, and the most important synthesizer specifications have been described. Then Chapter 3 and Chapter 4 analyze the detailed technologies of PLL-based and DLL-based frequency synthesis respectively and the key issues for a low-noise synthesizer design are also discussed.

Chapter 5 firstly compares PLL-based and DLL-based synthesis methods briefly,

then the architecture of a DLL-based frequency synthesizer is proposed and the specifications of each functional block are given.

In Chapter 6, 7, two key components, the voltage-controlled delay line (VCDL) and the new phase comparison circuit, including the novel charge pump phase comparator and the lock detector, are analyzed in detail. The detailed designs and simulation results of both are presented.

In Chapter 8, transistor-level simulations of the proposed DLL-based frequency synthesizer are described and the key issues involved in the layout of the circuit are discussed.

Chapter 9 describes the physical measurement setup and the analysis of the measurement results. And Chapter 10 outlines the conclusions drawn from this work and proposes some suggestions for future investigation.

Appendices contains information mainly about the component schematics, chip layouts, PCB photo and so on. At the end of this thesis document is the references cited herein.

Overview of frequency synthesis

2.1 Introduction

The frequency synthesizer converts one (or more, depending on the technique) 'reference' frequency to one or more frequencies, which may be higher or lower than the reference, again depending on the methodology. Since noise in the reference frequency may appear at the output of a synthesizer, the reference frequency is normally derived from a low-noise source such as a crystal oscillator.

Ideally, the output of a frequency synthesizer in frequency domain is an infinitely thin Dirac-like impulse at the desired frequency, one not accompanied by any 'skirts' due to phase noise. This corresponds in the time domain to an output signal which has an unvarying period. But the output of a practical synthesizer always has a non-zero bandwidth because of unwanted phase modulation introduced by the circuit or the reference signal. Since it is very difficult to remove the undesired frequency components close to the center frequency by filtering, methods to prevent their occurrence deserve attention in the first place.

This chapter starts by introducing several important parameters such as phase noise, spurious power level, and others which characterize the performance of a frequency

synthesizer, then some typical frequency synthesis technologies are reviewed and their performance is briefly compared.

2.2 Characteristics of frequency synthesizers

2.2.1 Phase noise

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

Figure 2.1 Spectrum of an ideal signal and a practice signal

Figure 2.1-A. Figure 2.1-B shows the spectrum of its practical counterpart having the same nominal output frequency. The ratio between the power at a frequency which is offset from a center frequency (f_0) by f_{offset} , to the power of the center frequency, is defined as phase noise. It is a measure of the relative level of the undesired noise components,

$$\text{Phase noise}(f_{offset}) = \frac{\text{Power at } (f_0 + f_{offset})}{\text{Power at } f_0} \quad (2.1)$$

Phase noise is essentially a measure of the uncertainty in a carrier's period. This uncertainty can be considered as the result of frequency modulating the carrier by some other signals and therefore phase noise can be better understood by examining the fre-

quency modulation process.

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

$$v_m(t) = A_m \cos(\omega_m t) \quad (2.2)$$

This assumption does not result in a loss of generality, because any physically-realizable signal can be decomposed into a set of sinusoidal signals by Fourier transform.

The signal resulting from the frequency modulation is thus written as,

$$v(t) = A_c \cos\left(\omega_c t + k_f \cdot \frac{A_m}{\omega_m} \sin(\omega_m t)\right) \quad (2.3)$$

Where k_f is a constant, A_c the amplitude, and ω_m is the angular frequency of the modulating sinusoid. When $k_f \cdot \frac{A_m}{\omega_m} \ll \frac{\pi}{2}$, Equation (2.3) may be written as,

$$v(t) \equiv A_c \left[\cos(\omega_c t) + \frac{m_f}{2} \cos((\omega_c + \omega_m)t) - \frac{m_f}{2} \cos((\omega_c - \omega_m)t) \right] \quad (2.4)$$

Where $m_f = k_f \cdot \frac{A_m}{\omega_m}$ is the modulation index. Equation (2.4) shows that at low modulation levels, frequency modulation of the carrier by a single low-level noise sinusoid results in the appearance of two sidebands at $\omega_c \pm \omega_m$ as shown in Figure 2.2.

For the output of a practical frequency synthesizer, the modulating signal is not a simple sinusoidal signal as above, but a complex noise, which contains many frequency components. Figure 2.3 shows the power spectrum produced when a sinusoidal carrier is modulated by a more general signal (noise). The relationship of the device output spectrum to the spectrum of the modulating noise can be observed in this diagram.

Figure 2.2 Spectrum of a narrow-band FM signal

Figure 2.3 Power spectrum of a FM signal

Equation (2.1) is a definition of phase noise in terms of power levels. A more practical definition of phase noise (at the frequency offset f_m) in terms of frequency deviation from a central value (carrier) can also be given as,

$$\text{Noise-to-carrier ratio(dB)} = 20 \log \left(\frac{\Delta f_{rms}}{\sqrt{2} f_m} \right) \quad (2.5)$$

Where f_m is the frequency offset from carrier and the Δf_{rms} is the RMS value of the frequency variation with respect to the carrier frequency within a certain bandwidth (normally 1Hz). Particularly, if the bandwidth is 1Hz, the result of Equation (2.5) is in unit

of dBc/Hz.

2.2.2 Spurious power level

The undesired frequency components which appear in the operating band of a device at some discrete frequency points are called spurs. Their power levels are described in a way similar to the phase noise power levels, namely by comparison with the carrier power. Figure 2.4 shows an example of a frequency source corrupted by both phase noise

Figure 2.4 Power spectrum of a frequency source with spurs

and spurs.

In theory, the phase noise can be defined as a measure of the phase variation over a wide range of frequencies centered on the desired output frequency. But such a wide band measurement would give a misleading impression of the extent of phase noise because spurs only appear at discrete frequencies. The assumption that the noise measured at large offset is phase noise will lead to one overlooking the various mechanisms (phase noise is normally generated from noise in device while spurs normally result from the actual system operation or electromagnetic radiation).

Spurious power level is measured in a way that is analogous to phase noise mea-

surements,

$$\text{spurious power level(dBc)} = 10\log\left(\frac{\text{Spurious power}}{\text{Carrier power}}\right) \quad (2.6)$$

Spurious power level is related to carrier power but is independent of measurement bandwidth, so it is reasonable to express it in unit of dBc (decibels relative to carrier). Among those spurs shown in Figure 2.4, the two spurs closest to carrier are of most concern because they may affect the following systems significantly. So the spurious power level normally reflects to the spur of higher power of the two.

2.2.3 Operating frequency range

Normally, the frequency synthesizer produces an output signal whose frequency is the reference frequency multiplied by some factor. This factor may be fixed, determined by the circuit design itself, or it may be a variable which is supplied by an external source. But in either case the output frequency will change if the reference frequency does.

The operating frequency range denotes the maximum input frequency range over which the desired frequency ratio between input and output can be maintained with acceptable output phase noise, spurious power level and so on.

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

2.2.4 Frequency increment

The frequency increment of a tunable frequency synthesizer is defined to be the

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

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

2.2.5 Channel switching time

Channel switching time refers to the time needed for a tunable frequency synthesizer to settle to the new output frequency after the reference frequency is changed. Most indirect frequency synthesizers (such as “Phase-locked loops” on page 14, and “Delay-locked loops” on page 15) incorporate some kinds of filtering in their feedback loops, and the filter bandwidth normally affects the channel switching time (this time increases as the loop bandwidth decreases).

A well-known relation between a filter’s 3 dB bandwidth (B_{3dB}) and the rise time (t_r) is shown in Equation (2.7), where k is a value between 0.3 to 0.45. The rise time t_r is

$$t_r \cong \frac{k}{B_{3dB}} \quad (2.7)$$

the time needed for a voltage ‘step’ to move from 10% to 90% of its full swing. The equation can be used to estimate the channel switching time during initial design stages. It should however be kept in mind that Equation (2.7) does not involve phenomena such as overshoot and ringing, and a more refined model should be used in later design stages.

2.2.6 Power consumption

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

2.3 Some typical frequency synthesis techniques

2.3.1 Incoherent synthesis

Incoherent synthesis refers to a way of generating output frequencies from a set of input frequencies by using frequency summation and subtraction operations. The block diagram of a typical incoherent synthesizer is shown in Figure 2.5 [6].

Figure 2.5 An example of incoherent synthesis

The frequency synthesis is carried out in a sequence of mixing and filtering operations involving selected stable reference sources. This is presented as a structural example and a more detailed treatment of its operation can be found in [6]. Undesired spurious

components are an unavoidable result of the mixing processes and some of them may fall in the operating band. Removal of these by filtering or others in a synthesizer designed for a range of outputs is a very difficult matter.

2.3.2 Direct digital synthesis

Direct digital synthesis (DDS) is a technique to produce an output frequency that is lower than the reference frequency with a completely digital approach. Figure 2.6 shows

Figure 2.6 A direct digital synthesizer

the block diagram of a DDS. The main component is a phase accumulator with the tunable increment F_{ctrl} . The value of the register is increasing by the value of F_{ctrl} at each period of F_{ref} and it appears as a saw wave due to the overflow of the register.

From the analysis, the frequency of this saw wave (a sequence of addresses for the ROM) can be obtained as,

$$F_{out} = F_{ref} \cdot \frac{F_{ctrl}}{2^m} \quad (2.8)$$

There should be at least 2 sampling points within each output period, thus the maximum value of F_{ctrl} is 2^{m-1} . Consequently, the theoretical maximum output frequency is

$F_{ref}/2$, and the minimum output frequency is $F_{ref}/2^m$.

Normally, the desired output is a sinusoid instead of a saw waveform, a look-up table is employed to map the saw waveform to a sinusoidal waveform. The actual output is obtained by a low-pass filter at the output of a D/A converter. The dot line in the Figure 2.6 denotes a possible control line to the low-pass filter to tune the bandwidth of the filter according to the actual output frequency determined by F_{ctrl} .

2.3.3 Phase-locked loops

Besides two typical direct synthesis techniques, an indirect synthesis technique, which utilizes the PLL as a feedback to generate output frequency, is widely used over many years.

A typical structure of a PLL-based frequency synthesizer is shown in Figure 2.7.

Figure 2.7 A PLL-based frequency synthesizer

The basic function of a PLL is to tune the phase of the VCO to make the frequencies of two inputs of the phase comparator identical and the phase difference between them con-

stant. With a divider (N), the actual output frequency of the VCO is,

$$F_{out} = N \cdot F_{in} \quad (2.9)$$

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

$$F_{out} = N \cdot \frac{F_{ref}}{M} \quad (2.10)$$

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

Not only can the phase-locked loop be employed to generate an output frequency with the ratio of an integer, but a ratio which is the sum of an integer and a fraction of an integer can be obtained, this is often referred as fractional- N frequency synthesis. To achieve the fractional division ratio, a programmable divider can be employed in the PLL-based frequency synthesizer. Figure 2.8 shows a simplified diagram of a fractional- N PLL-based frequency synthesizer with a divider including a cycle swallower to obtain a fractional division ratio. With this fractional division ratio, the ratio between the output frequency and the input frequency can be fractional since the output of the divider is locked at the same frequency as the input.

2.3.4 Delay-locked loops

In recent years, delay-locked loops, which have their incomparable advantages,

Figure 2.8 A fractional-N PLL-based frequency synthesizer

such as low noise and high stability, have been employed to design frequency synthesizers. In DLLs, multiple identical delay stages are included in a voltage-control delay line (VCDL) whose delay is locked to one period of the reference signal, the output signal is obtained by combining multiple-phase signals coming from multiple delay stages of the VCDL. A simplified diagram of a DLL-based frequency synthesizer is shown in Figure 2.9.

Figure 2.9 Block diagram of a DLL-based frequency synthesizer

2.3.5 Comparison of synthesis techniques

Some typical frequency synthesis techniques are compared briefly as listed in Table 2.1. The operating frequency ranges provided in the table are for reference only, the

TABLE 2.1: Comparison of frequency synthesis techniques

Techniques	Incoherent Frequency Synthesis	Direct Digital Frequency Synthesis	PLL-based Frequency Synthesis	DLL-based Frequency Synthesis
Number of sources needed	multiple	1	1	1
Operating frequency range	Large (Up to 10GHz)	Small (Up to 1GHz)	Medium (Up to 5GHz)	Medium (Up to 5GHz)
Frequency ratio range	Small	Small	Large	Large
Phase noise performance	Medium	Medium	Low	High
Spur performance	Medium	Medium	High	Low
Complexity	Low	Medium	High	High

actual range may vary with the development of techniques. As we can see, no single solution of frequency synthesis is perfect for all applications, actual technique should be selected based on the special application and requirement. In some special cases, one may even consider the combination of those techniques.

2.4 Summary

This chapter firstly describes the most important parameters to evaluate the performance of a frequency synthesizer. Then some typical frequency synthesis techniques are

briefly reviewed and comparison of these conventional frequency synthesis techniques is given. Over those techniques, PLL-based frequency synthesis is the most popular one and being widely used. The DLL-based technique, which has some incomparable advantages over the PLL-based technique, has found its uses in recent years. However, the PLL-based technique still cannot be replaced by the DLL-based one due to the limitation of the DLL-based technique. To further the analysis of frequency technologies, chapter 3, 4 discuss PLL-based frequency synthesis and DLL-based frequency synthesis technologies in detail respectively.

PLL-based integer-N frequency synthesis technology

3.1 Introduction

The PLL-based frequency synthesis has many attractive advantages over other techniques and has been widely used over many years. High PLL performance is however strongly dependent on careful design. In this chapter, the main characteristics of PLL-based frequency synthesizers, including phase noise performance and spur performance, are analyzed. This chapter mainly focuses on the most commonly used integer-N PLL-based frequency synthesizers; the analysis which is developed can be applied with minor modifications to sigma-delta ($\Delta - \Sigma$) fractional-N frequency synthesizers.

3.2 Fundamental characteristics of PLLs

3.2.1 Operating principle of PLLs

A simplified PLL block diagram is shown in Figure 3.1. It consists of a voltage-controlled oscillator (VCO), a phase detector (PD) and a loop filter (LF). The VCO provides a sinusoidal signal (y_o) which is applied to the PD to generate a voltage (u_1) proportional to the phase difference between the VCO output (y_o) and the reference (y_i). This is thus low-pass filtered and sent back to the VCO as a control voltage (u_2) to tune its fre-

Figure 3.1 Block diagram of a PLL

quency. Although practical PLLs are often more complicated, the structure in Figure 3.1 includes all of the fundamental features of the PLL.

For the purpose of mathematical analysis, an ideal linear model is set up, which includes a linear phase detector, a 2nd-order low pass filter, and an ideal VCO.

Suppose that the input reference signal and the output signal (output of the VCO) are given by,

$$y_i(t) = A \sin[\omega t + \phi_i(t)] \quad (3.1)$$

$$y_o(t) = B \sin[\omega t + \phi_o(t)] \quad (3.2)$$

The transfer function of the PLL can be developed in the following steps:

If a linear phase detector is assumed, the output of the phase detector can be written as,

$$u_1 = K_{PD}(\phi_i(t) - \phi_o(t)) \quad (3.3)$$

Where K_{PD} is the gain of the phase detector. An ideal voltage-controlled oscillator

behaves as,

$$\frac{d\phi_o(t)}{dt} = K_{VCO} \cdot u_2 \quad (3.4)$$

Where K_{VCO} is the gain of the VCO and u_2 is the control voltage. If the impulse response of the loop filter is denoted as $f(t)$ (transfer function $F(j\omega)$), one can obtain,

$$\frac{d\phi_o(t)}{dt} = K_{VCO} \cdot [K_{PD}(\phi_i(t) - \phi_o(t)) * f(t)] \quad (3.5)$$

And the transfer function of a PLL can be written as,

$$H(j\omega) = \frac{\Phi_o(j\omega)}{\Phi_i(j\omega)} = \frac{K_{VCO}K_{PD}F(j\omega)}{j\omega + K_{VCO}K_{PD}F(j\omega)} \quad (3.6)$$

Equation (3.6) shows that the PLL itself is a low-pass filter with respect to the phase variation of the reference signal even without a loop filter, which means that it can track slow phase variation and reject rapid phase variation.

The characteristics of a PLL are dependent strongly on its loop filter. This can range in complexity from no filter at all, to a common low-pass filter, to an integrator, or to something else.

3.2.2 Stability of PLLs

Stability is a criterion of great importance in almost all feedback systems since such systems may oscillate under certain conditions. Such behavior may be intentional in some circumstances, such as with oscillators, but for most other applications (PLLs in particular), it is undesirable and must be prevented.

One could ensure the stability of a feedback system by making certain that there is

absolutely no frequency component in the input signal which can be both amplified and phase shifted 360° in the feedback loop. Unfortunately the spectrum of an input signal in a practical application always occupies the full operating frequency range due to the presence of noise, so one should design a feedback system in such a way that it simply can not amplify any frequency component which is phase shifted by 360° .

A common tool which is used to verify the stability of a feedback system is the Bode plot, a graph in which the amplitude response and phase response of a system are displayed as functions of the frequency. If the phase shift over the frequency range in which the amplitude response is larger than unity is always less than 360° , the system will be stable; otherwise it will be unstable.

The PLL shown in Figure 3.1 on page 20 is a negative feedback loop in that it always acts to drive the phase detector output to a certain value. Thus, the loop itself introduces a 180° phase shift between the PD input and output. So if there is another 180° phase shift in the open loop (introduced by the loop filter and the VCO) the actual phase shift for the closed loop is 360° . According to this characteristic, the stability of the PLLs can be found by examining the open loop frequency response.

According to equations (3.3), (3.4) and (3.5), it is easily to find the open loop transfer function as,

$$H_{open}(j\omega) = \frac{K_{PD}K_{VCO}F(j\omega)}{j\omega} \quad (3.7)$$

Since PLL characteristics are largely determined by loop filter behavior, it would do well to examine several commonly used filters now.

For the case of no filter at all ($F(j\omega) = 1$), the open loop acts as an integrator which has a constant phase shift of 90° because the VCO output (phase) is actually the integral value of its input multiplied by the VCO gain, so the loop will be unconditionally stable. In practice, since most phase detectors work only at each edge of input signal (discontinuously), a filter is always employed to reduce the spurious power and thus to attenuate the phase detector's noise output. Such a loop (called a *first order* loop) is not practical for most uses and higher order loops are often used.

If an ideal integrator is employed as the loop filter ($F(j\omega) = \frac{1}{j\omega}$), the open loop transfer function will be,

$$H_{open}(j\omega) = \frac{K_{PD}K_{VCO}}{(j\omega)^2} \quad (3.8)$$

In this case, all frequency components are phase-shifted 180° as shown in Figure 3.2-A, and the system is always unstable.

Figure 3.2 Bode diagrams

When an RC low-pass filter (Figure 3.3-A) is used as loop filter

($F(j\omega) = \frac{1}{j\omega RC + 1}$), the open loop transfer function is,

$$H_{open}(j\omega) = \frac{K_{VCO} \cdot K_{PD}}{j\omega(j\omega RC + 1)} \quad (3.9)$$

The Bode plot of Figure 3.2-B indicates that if the frequency is very high, the phase shift would be 180° and it may result in an unstable system if the gains of the VCO and the PD are large enough.

The most common way to design a stable PLL is to use the filter above together with the phase-lead correction, which drags the phase shift of the loop filter frequency response from -90° back to 0° when the frequency is high and thus the phase shift of the open-loop transfer function is always above -180° as the dash line shown in Figure 3.2-B. A simple way to achieve this effect is to add a phase-lead correction resistor to RC low pass filter as shown in Figure 3.3-B.

Figure 3.3 Loop filters for PLLs

The transfer function of the modified loop filter is,

$$F(s) = \frac{1 + \tau_2 s}{\tau_1 s} \quad (3.10)$$

Where $\tau_1 = R_1 C$ and $\tau_2 = (R_1 + R_2)C$, the phase shift of this filter vanishes at high frequency and so the phase shift of the PLL's forward path will return to 90° as the

frequency is increased.

In addition to the phase shift discussed above, phase shift is introduced by all components in the loop of a practical PLL, and these extra sources must be also accounted for by the designer. Normally, a certain value of phase margin should be maintained for this reason. So for a loop employing a low pass filter with phase lead correction as its loop filter, careful design should also be done to achieve a stable PLL.

3.2.3 Noise performance of PLLs

A PLL's ability to deal with the noise that it will encounter in its working environment is an important issue for almost all applications. This section analyzes the PLLs' capacity to suppress noise in the reference, noise generated by the VCO and noise in the control voltage. The relation between the output noise power and the power of the original noise sources will be determined.

To simplify the analysis, all original noise sources are assumed white, that is to have constant power over the operating frequency band.

3.2.3.1 External noise coming from reference signal

We first consider the behavior of the basic PLL circuit shown in Figure 3.1 on page 20 under ideal conditions, namely those of no noise, either in the input signal (y_i) or generated by the PLL circuitry itself. In such a case the output of the phase detector (u_f) will be constant when the loop is locked and a steady DC level will exist at the output of the loop filter. The output of the VCO (y_o) will then be free of phase noise. If however the reference signal is contaminated by some phase noise, it will appear at the output of the phase detector (u_f), and a portion will remain in the output of the loop filter, meaning that

there will be a noisy signal at the control voltage of VCO (u_2) and a consequent noisy VCO output (y_o).

In this section, the noise component of the reference signal (not the reference signal itself) is denoted as N_{exn} , and that of the VCO output N_{out} . With the superimposition principle, one can find the relation between N_{exn} and N_{out} directly from the transfer function as Equation (3.6),

$$\frac{N_{out}(j\omega)}{N_{exn}(j\omega)} = \frac{K_{VCO}K_{PD}F(j\omega)}{j\omega + K_{VCO}K_{PD}F(j\omega)} = H_{ex}(j\omega) \quad (3.11)$$

This shows that the PLL has a low-pass filtering effect on the input noise content.

Suppose that all components of the PLL are noiseless and only the noise coming from the reference signal is considered, the output noise power can be easily calculated according to the transfer function as Equation (3.11) by integrating the noise component in the frequency domain.

$$\sigma_{noex}^2 = \int_0^{+\infty} (H_{ex}(j\omega) \cdot N_{exn}(j\omega))^2 d\omega \quad (3.12)$$

To simplify the analysis, all noise sources in this section are supposed to be white noise which has constant power over the operating frequency range, so Equation (3.12) can be written as,

$$\sigma_{noex}^2 = N_0^2 \cdot \int_0^{+\infty} |H_{ex}(j\omega)|^2 d\omega \quad (3.13)$$

Where N_0^2 denotes the external noise power within the operating frequency range.

3.2.3.2 Internal noise coming from the VCO

Positive feedback within a VCO leads to a cycle-by-cycle accumulation of the noise generated in the VCO.

Internal feedback of the VCO causes noise to build up over time, and the integrated noise appears at the VCO output. When the VCO is not operating in a PLL, this behavior is equivalent to a system having the transfer function of $H_v(j\omega) = \frac{1}{j\omega}$. The closed-loop behavior can be determined using a model shown in Figure 3.4.

Figure 3.4 PLL model for VCO noise analysis

An analysis in the section “Operating principle of PLLs” on page 19 is used to find the noise transfer function for this model,

$$N_{out}(j\omega)(-K_{PD}) \cdot F(j\omega) \frac{K_{VCO}}{j\omega} + N_{in}(j\omega) = N_{out}(j\omega) \quad (3.14)$$

$$H_{novco}(j\omega) = \frac{N_{out}(j\omega)}{N_{in}(j\omega)} = \frac{j\omega}{j\omega + K_{PD}K_{VCO}F(j\omega)} \quad (3.15)$$

Unlike its response in the case of external noise, the PLL acts as a high pass filter with respect to the noise coming from the VCO. As a result, only low-frequency VCO noise components can be suppressed by the PLL. The actual effect of the VCO’s internal

noise on the output of the PLL can be written,

$$H_{inovco}(j\omega) = H_{novco}(j\omega) \cdot \frac{1}{j\omega} \quad (3.16)$$

Here the internal noise of the VCO means the phase variation generated within only one cycle. The noise seen at the VCO's output will be a number of such contributions, integrated over operating cycles.

The output noise power due to VCO's internal noise can be specified as,

$$\sigma_{novco}^2 = N_0^2 \cdot \int_0^{+\infty} |H_{inovco}(j\omega)|^2 d\omega \quad (3.17)$$

Where N_0^2 denotes the VCO's internal noise power within the operating frequency range.

3.2.3.3 Noise coming from the control signal of the VCO

The effect of the noise in the VCO control signal can be equivalently treated as the internal noise of the VCO by multiplying the amplitude of the noise by the gain of the VCO (K_{VCO}). Equation (3.16) can be employed to calculate this effect by,

$$H_{novcl}(j\omega) = H_{inovco}(j\omega) \cdot K_{VCO} \quad (3.18)$$

From this equation, the PLL acts as a high pass filter with respect to the noise in the VCO control signal. The output noise power caused by this noise can be written as,

$$\sigma_{novcl}^2 = N_0^2 \cdot \int_0^{+\infty} |H_{novcl}(j\omega)|^2 d\omega \quad (3.19)$$

Where N_0^2 denotes the power of noise coming from the control signal of the VCO

within the operating frequency range.

To calculate the actual noise power caused by all noise sources analyzed above, one just adds them together as,

$$\sigma_n^2 = \sigma_{noex}^2 + \sigma_{novco}^2 + \sigma_{novcl}^2 \quad (3.20)$$

As we can see from Equation (3.20), minimization of the final output noise involves the optimization of phase noise from three main sources. Loop bandwidth is an important issue in this process - larger loop bandwidth means that more internal noise can be suppressed, but less external noise can be rejected.

3.3 Operating principle of PLL-based frequency synthesizers

According to the basic operating principle of the PLL, the output frequency of a PLL's VCO is equal to the input frequency when the loop is in lock. If the PLL is to be used to build up a frequency synthesizer, a frequency divider is inserted between the VCO and the PD so that the VCO output can be a multiple of the reference frequency. Multiplication of the reference by a fraction can also be achieved by using an appropriate divider.

The block diagram of a multiply-by-N PLL frequency synthesizer is shown in Figure 3.5.

The VCO and the frequency divider can be considered as parts of a single more complex VCO, one having two output frequencies differing by a factor of N. The lower of the two is locked to the same value as the input frequency, with the higher one being N times the input frequency. In many applications, the value N is variable so that the frequency synthesizer can derive more than one output frequency from a fixed input fre-

Figure 3.5 Block diagram of a PLL-based frequency synthesizer

quency. Each block of an integer-N PLL synthesizer will be analyzed.

3.4 Basic components of PLL-based frequency synthesizers

3.4.1 Voltage-controlled oscillator

The voltage-controlled oscillator (VCO) is normally the source of a synthesizer's output signal - the frequency stability and spectral purity of the VCO's output are therefore of paramount importance. Although a properly designed PLL can itself control the noise output of a VCO to some extent (the lower-frequency, in-band components), a low-noise VCO is still desired for low-noise frequency synthesizers.

There are two main types of VCO noise sources, the internal one accumulated over cycles, and the non-accumulating external one as N_{in} shown in Figure 3.4 on page 27. These origins of VCO phase noise must be studied separately.

The first step to build a VCO is to implement the oscillator itself. Then a structure to tune its frequency with a control signal is added. One of the most popular VCO realizations makes use of a ring oscillator because its resonatorless design makes for easy inte-

gration. Such an oscillator consists of odd number of inverting amplifiers connected in a feedback loop as shown in Figure 3.6.

Figure 3.6 A ring oscillator

If each inverting amplifier has a delay of τ , and there are N inverting amplifiers in the loop, the oscillating frequency can be expressed as,

$$f_{osc} = \frac{1}{2N\tau} \quad (3.21)$$

Advantages of this architecture, apart from ease of implementation, include ease of tuning (using voltage or current control of τ), and the simple derivation of quadrature or other multi-phase outputs. The oscillator's output may be taken from any of the equivalent inverters in the ring.

The phase noise exhibited by this type of oscillator will build up with each pass of the circulating signal through the loop. The phase variation at any time is the accumulation of inverter delay variations over many cycles. There are also non-accumulating noise components, but the output phase noise caused by accumulated variation is normally dominant. The delay variation can be caused by the variation in power supply or control voltage, and by device-level noise, such as flicker noise, thermal noise, and so on.

The term 'external noise' as applied to a ring-type VCO normally refers to the

noise introduced at the output buffer stage of the VCO. By careful design, this noise source can be made insignificant compared to the internal sources described above.

3.4.2 Digital frequency dividers

Of the two kinds of modulation likely to cause difficulties in the PLL synthesizer, AM and PM, AM is expected to create the least trouble because an ideal digital frequency divider will be unaffected by amplitude modulation of its input, and no effects will appear at the output. This is simply because the digital circuit responds to voltage transitions at its input, and not to the voltages themselves. However, this is not always true in the case of a practical circuit because of switching threshold voltage deviation. In this case, the AM is converted to PM finally transmitted to the output of the divider as shown in Figure 3.7. To

Figure 3.7 Effect of AM-to-PM conversion in a frequency divider

minimize this effect, the amplitude of the divider's input should be kept constant as possible, and the input transitions should be made as fast as possible.

The noise transmission discussion can now be confined to PM signals. In a synchronous divider each output transition of the frequency divider coincides with an input

transition with a constant delay, so any timing uncertainty in the input transition shows up in the output with the same uncertainty,

$$\Delta t_o = \Delta t_i \quad (3.22)$$

But the corresponding phase variation is proportional to the frequency,

$$\Delta\phi = \Delta t \cdot \omega \quad (3.23)$$

Where ω is the angular frequency of the carrier. Since the output frequency of a divider is its input frequency divided by N , the combination of Equation (3.22) and Equation (3.23) yields a relation between the phase variation of the output and that of the input as,

$$\Delta\phi_o = \frac{\Delta\phi_i}{N} \quad (3.24)$$

Consequently, the amplitude of modulation sidebands at the input of a frequency divider is divided by N at its output. And their powers are divided by N^2 . Or the power (in decibel) will be down by $20\log(N)$. This relation can be written as,

$$N_{out}(f_{offset}) = N_{in}(f_{offset}) - 20\log N \quad (3.25)$$

Where f_{offset} is the frequency offset from carrier frequency. Figure 3.8 shows an example of this effect for a frequency divider with the division ratio of 10 and the sidebands are down by $20\log(10)=20\text{dB}$. However, the phase noise can not go down infinitely as the division ratio increases, due to the noise floor limit.

In practice, the phase noise does not just occur at a few cleanly-separated discrete frequencies as shown in Figure 3.8, but extends over the operating band. Actual phase noise measurements appear as a 2-D curve which shows the phase noise power as a func-

Figure 3.8 Change of the power spectrum by a 10 divider

tion of frequency offset. The curve approaches the noise floor at frequencies far from the carrier's. If the division ratio is very large, the output phase noise goes to the noise floor even when the frequency offset is very small, and it will be larger than the value obtained from Equation (3.25). Figure 3.9 shows the phase noise of the input and the output of a

Figure 3.9 Phase noise of the input and output of a frequency divider

divider with the division ratio of 10 and 20 respectively. When the ratio is 10, the output phase noise at a 100Hz offset is 20dB down from the phase noise of the input signal, but when the ratio is 20, the phase noise is only 15dB down from that of the input because of the noise floor limitation. At the very large offset of 1MHz, the phase noise of the input is almost the same as that of the output for both cases.

When the output edges of a digital frequency divider are synchronized to the input edges, there will be a delay T_D between them, and thus a phase shift of $T_D \cdot \omega$. This phase shift may affect the PLL's stability at higher frequencies.

3.4.3 Phase detector

The phase detector is the source of the error signal which tunes the VCO and keeps the PLL locked. Its noise performance, which greatly depends on its actual circuit design, affects the loop performance directly. If a PLL with ideal components except for the PD is working with a noiseless frequency source, the phase noise components of the PD within the loop's bandwidth will be transferred to the VCO output.

A commonly-used phase detector has two inputs and one or two outputs. In the one-output case, the PD output is normally a voltage that is proportional to the timing error (or phase error) between its two inputs. When there are two outputs, the timing error between the two inputs is normally presented at the output in a differential form - for example as a difference in pulse widths.

3.5 Noise performance of the PLL-based frequency synthesizers

All components involved in PLL frequency synthesizers - the VCO, the frequency divider, the phase detector, the loop filter - affect the noise performance of the synthesizer.

This section analyzes their effects on the noise performance of the frequency synthesizer.

The loop bandwidth, which is mainly determined by the loop filter, plays a key role on the performance of the PLL-based frequency synthesizer. Low-frequency, in-band, noise components generated in the synthesizer are compensated by the action of the loop, and any such noise appearing in the PLL output will be mainly due to the noise contained in the reference signal. The output noise components whose frequencies lie outside the loop bandwidth originate mainly with the VCO as mentioned earlier. Figure 3.10 illus-

Figure 3.10 Noise performance of PLL-based frequency synthesizer

trates this effect for a PLL-based frequency synthesizer without a frequency divider (The output frequency is the same as the input frequency).

Normally, an integer-N frequency divider is inserted into the PLL so that the output frequency is multiplied by N. Since the phase noise at the output of the divider is

related to that at the input by Equation (3.25), the power (in decibel) of output phase noise components at offsets less than the loop bandwidth is related to that of the reference by,

$$N_{out}(f) = N_{ref}(f) + N_a(f) + 20\log N \quad (3.26)$$

Where $N_a(f)$ is the phase noise added by the loop itself and does not indicate the effect of frequency division. It is mainly determined by the characteristics of phase detector and the loop filter. Equation (3.26) shows that phase noise increases with the increase of N . The phase noise components outside the loop bandwidth are still determined by the characteristic of the VCO.

3.6 Low noise frequency synthesizer design

When designing a low noise PLL-based frequency synthesizer, one should of course use low noise devices for the VCO, the phase detector and the loop filter. For those applications where low noise at small frequency offsets is critical, designers should devote most effort to the phase detector and the loop filter, and for those applications where low noise at large offsets is especially important, attention should be concentrated on the design of a low-noise VCO. Both objectives are of course important and the loop bandwidth should be optimized to minimize the overall phase noise over the operating frequency band.

Figure 3.11 shows how to choose the synthesizer loop bandwidth for which the output phase noise can be a minimum. The phase noise of the reference signal and the estimate of the output phase noise ($20\log N$ higher than that of the reference signal) are plotted. The optimum loop bandwidth is the frequency at which the VCO phase noise and the output phase noise curves intersect (curve (a) and dotted curve). In frequency synthesis

- a. Phase noise of VCO
- b. Phase noise of the reference signal

Figure 3.11 Optimum loop bandwidth

applications, the input signal normally comes from a low-noise source (the dotted curve will be lower), so the intersection (optimum) frequency will be greater than that for other applications, such as clock/data recovery (CDR). This is why the large loop bandwidth is normally desired for the frequency synthesizer applications.

3.7 Summary

In this chapter, basic PLL characteristics have been reviewed, and the operating principle of PLL-based frequency synthesizers using frequency dividers is analyzed. Then the noise performance of the PLL synthesizer is discussed and key issues which occur in the course of low-noise design are given.

DLL-based integer- N frequency synthesis technology

4.1 Introduction

PLL-based frequency synthesis technology has been widely used in many years and its theory has been almost fully developed over years. For most applications, PLL designers may resort to available tools and established design rules in their work.

Besides the PLL-based frequency synthesis, DLL-based ones have gained more and more interest in recent years due to the nature of absence of noise accumulation in voltage-controlled delay line. Although some DLL-based frequency synthesizers have been actually designed up to date [7][8][9][10][13], and their performance shows some improvement, the theoretical treatment of this topic is very limited in the literature. This chapter firstly deals with the operating principle of the DLL and its noise performance, then the operating principle of a DLL-based frequency synthesizer is introduced and its output phase noise and spurs are analyzed theoretically in the later part of this chapter.

4.2 Fundamental characteristics of DLLs

Different from PLLs, which tune the phase of the VCO by a feedback, DLLs are those feed-back systems that tune the delay of voltage-controlled delay lines. The DLLs

can be used in many applications, such as phase correction, frequency synthesis, clock recovery, and so on. To analyze the performance of a DLL-based frequency synthesizer, the characteristics of its basic structure, a DLL itself, are analyzed in the following subsections.

4.2.1 Basic operating principle of DLLs

Basically, a DLL is a system employing a feedback loop to tune the delay of a voltage-controlled delay line (VCDL) to a certain value, normally one period of the reference signal. Figure 4.1 shows the diagram of a basic DLL. The periodic input signal y_i is

Figure 4.1 Block diagram of a basic DLL

delayed by the VCDL, whose output is sent to a phase detector (PD) where it is compared to the input signal. The PD output is thus to tune the delay of VCDL. The control voltage (u_2) is created by a charge pump (CP) and a loop filter (LF). Ideally, the output of the phase detector is zero and the control voltage of the VCDL is constant when the loop is in lock, which means the phase difference of two inputs of a phase detector is zero or $2n\pi$, ($n=0, 1, 2, \dots$). The total delay of the VCDL is therefore nT , where T is the period of the input signal. In most applications, the delay of VCDL is fixed at one period of the reference signal ($n=1$), and measures are taken to ensure this, such as control voltage initialization, lock detection, and so on. For the theoretical analysis in this section, the value of n is

supposed to be 1. So the total delay of the VCDL is always supposed to be locked to one period of reference signal. The output of the DLL, signal y_o in the Figure 4.1, will differ from the input signal by one-cycle delay. Although final outputs of an actual DLL-based system may not be the signal y_o , they are all greatly related to the signal y_o . So the analysis of the output signal y_o is always important for the analysis of the performance of a DLL-based system.

The output of the DLL is delayed version of the input signal, so its frequency is the same as the input. Normally, the input and output are supposed to be square waveforms and the input reference signal and the output signal (output of the VCDL) are represented respectively as,

$$y_i(t) = A \sin(\omega t + \phi_i(t)) \quad (4.1)$$

$$y_o(t) = B \sin(\omega t + \phi_o(t)) \quad (4.2)$$

Where $\phi_i(t)$ and $\phi_o(t)$ are the phases of the input and the output respectively and the desired phase error between them $(\phi_i(t) - \phi_o(t))$ is 2π when the loop is in lock.

In frequency synthesis applications, the positive zero crossings of the input signal and output signal are normally of concern. In an ideal synthesizer, the input and the output zero crossings are evenly aligned on the time axis. By ignoring the initial phase, those crossing points are expressed as,

$$t(n) = nT \quad (4.3)$$

Where n is an integer and T is the period of the ideal reference or the ideal output signal. But in practice, there is always some variation and the Equation (4.3) must be rewritten as,

$$t(n) = nT + t_n(n) \quad (4.4)$$

The value of $t_n(n)$ should of course be as small as possible for both reference and the output signals of a frequency synthesizer. As for the reference, the only answer is to employ a low-noise frequency source. The output $t_n(n)$ is determined by both of the noise behavior of the DLL and the reference signal.

Firstly, the effect of input noise will be examined by assuming that all internal components of the DLL are noiseless, and the only noise is from the input signal. To simplify the analysis, both $t_{nr}(n)$ (variation of the reference) and $t_{no}(n)$ (variation of the output) are assumed to be small comparing to the period T . Furthermore, the values of u_1 and u_2 in Figure 4.1 are assumed to change only at the rising edges of the reference signal, and that their deviations are functions of n : $u_1(n)$ and $u_2(n)$.

A linear phase detector is assumed, one whose output is proportional to the timing error between $t_{nr}(n)$ and $t_{no}(n)$,

$$u_1(n) = K_{PD}(t_{no}(n) - t_{nr}(n)) \quad (4.5)$$

Where K_{PD} is the gain of the phase detector. The charge pump and loop filter combination will be assumed to act as an ideal integrator, with its input and output related by,

$$u_2(n) = u_2(n-1) + K_{CP}u_1(n) \quad (4.6)$$

The gain of the charge pump is denoted as K_{CP} . The output of the VCDL is a delayed version of the input with the delay controlled by u_2 as shown in Figure 4.1, so its input and output timing difference is related to the deviation of u_2 by,

$$t_{no}(n) = t_{nr}(n-1) + K_{VCDL}u_2(n-1) \quad (4.7)$$

Where K_{VCDL} denotes the gain of the VCDL. By substituting Equation (4.5) and Equation (4.6) into Equation (4.7), and transferring to z-domain, the relation between the

Figure 4.2 Transfer functions of the DLL

input and output can be expressed in z-domain as,

$$H_z(z) = \frac{z^{-2} - z^{-1} - K}{z^{-1} - (1 + K)} \quad (4.8)$$

Where K is the product of K_{PD} , K_{CP} and K_{VCDL} . The corresponding frequency domain expression can be derived with the relation $z = e^{j\omega}$,

$$H_f(f) = H_{nf}\left(\frac{2\pi f}{f_{ref}}\right) = H_{nf}(\omega) = \frac{e^{-2j\omega} - e^{-j\omega} - K}{e^{-j\omega} - 1 - K} \quad (4.9)$$

Figure 4.2 shows the transfer function for different values of K . The frequency in the figure is the normalized frequency ($\omega_{nor} = \frac{2\pi f}{f_{ref}}$). As we can see, the gain at the zero frequency is always unity, independent of K , that is because the output is a delayed version of the input and they have same values of timing error if the input timing error is constant (the frequency is zero). In the low-frequency band ($0 < f < \frac{f_{ref}}{4}$), the gain is always larger than unity, we can see that the DLL has no ability to suppress the external noise, instead it amplifies the input noise for large loop gain (K). That is because the output of

the DLL is the delayed version of its input with some additional phase variation, which is in the same direction as the input phase variation and is introduced by the loop itself.

4.2.2 *Stability of DLLs*

Since the DLL is a feedback system, its stability should be analyzed. In the case where only an integrator is used as the loop filter, which provides a constant open-loop phase shift of 90° , the DLL will be unconditionally stable. In fact as long as the loop filter is of the first order, the loop can always be stable.

4.2.3 *Noise performance of DLLs*

Similar to PLLs, the noise performance is an important characteristic of DLLs in most applications. This section analyzes the DLLs' capability to suppress the various types of noise.

4.2.3.1 *External noise coming from reference signal*

The effect of external noise accompanying the reference signal can easily be derived from the DLL transfer function (Equation (4.9)). If a flat spectrum of noise is assumed, the ratio of output noise power to the input noise power can be calculated by integrating the power computed with the transfer function over a frequency range. In most applications, it is the noise in the low-frequency band that is of greatest interest, so the noise analysis in this section will be limited to frequency range from 0 to $f_{ref}/4$. The output noise power within this range is calculated as,

$$\sigma_n^2 = N_{ex}^2 \cdot \int_0^{\frac{\pi}{2}} |H(j\omega)|^2 d\omega \quad (4.10)$$

From the transfer functions shown in Figure 4.2, the output noise power (σ_n^2) is always larger than the input noise power (N_{ex}^2) since the $|H(j\omega)|$ is always larger than unity in the low frequency range. Figure 4.3 shows the ratio of the output power to the

Figure 4.3 Power gain for external noise versus the value of K

input power for different values of K . This ratio reaches a maximum at K approximately equal to 0.7. One can see that the maximum ratio is less than 1.24 and the value of K affects the output noise power only in a very limited range if only the external noise is considered.

4.2.3.2 VCDL Internal noise

The internal noise originating in the VCDL behaves in the way similar to the internal noise of the VCO in a PLL, but it does not accumulate over cycles as VCO noise does. Thus the DLL noise can be considered as the noise added at the output of the VCDL as shown in Figure 4.4.

With this model, the relation between output noise and input noise can be found in

Figure 4.4 DLL model for VCDL internal noise analysis z-domain with the similar method as before,

$$H_z(z) = \frac{N_{out}(z)}{N_{in}(z)} = \frac{z^{-1}(1 - z^{-1})}{1 + (K - 1)z^{-1}} \quad (4.11)$$

And the transfer function in frequency domain can be obtained as,

$$H_f(f) = H_{nf}\left(\frac{2\pi f}{f_{ref}}\right) = H_{nf}(\omega) = \frac{e^{-j\omega}(1 - e^{-j\omega})}{1 + (K - 1)e^{-j\omega}} \quad (4.12)$$

DLL transfer functions with different values of K are shown in Figure 4.5. The fre-

Figure 4.5 Transfer functions of the DLL for internal noise

quency is also normalized. In the figure, it is seen that the output noise power decreases with the increase of K , from which it can be concluded that large value of K should be used in DLLs to suppress the internal noise coming from the VCDL.

4.2.3.3 Internal noise coming from the control line

Due to the nature of the VCDL, the internal noise originating from the control line circuitry can be treated in the same manner as previously, by taking it to be added at the output. The relation between them is written as,

$$H_{novcl}(j\omega) = H_{nf}(j\omega) \cdot K_{VCDL} \quad (4.13)$$

It is seen that the power of the output noise caused by the noise coming from the control line is similar to that caused by the noise coming from the VCDL, with the ratio of K_{VCDL}^2 between them.

4.3 Operating principle of DLL-based frequency synthesizers

Based on a basic DLL, a frequency synthesizer can be built up by combining the multiple-phase signals produced by a multiple-stage VCDL shown in Figure 4.6, where

Figure 4.6 Multiple-stage VCDL and pulse combiner

each delay stage produces a pulse (P_{un}) whose width is one half its delay on the arrival of an input rising edge. If the stage delays are correctly tuned, the phase difference between Out and In equals to exactly one cycle. The output signal (f_{out}) is obtained by combining all pulses (P_{u0} to $P_{u(N-1)}$) coming from those multiple delay stages.

Since the fundamental frequency of the pulse coming from a delay stage is same as the reference frequency, the Fourier transform of the n th pulse can be expressed as $X_{pun}(kf_{ref})$, where k is an integer, then we have,

$$X_{pun}(kf_{ref}) = X_{pu0}(kf_{ref}) \cdot e^{j2\pi f_{ref} t_d(n)k} \quad (4.14)$$

Where $t_d(n)$ is the delay of the pulse generated by the n th delay stage relative to the pulse generated by the first delay stage ($t_d(0) = 0$). Ideally, the pulse combiner acts as an adder which combines all pulses together. The Fourier transform of f_{out} can be written as,

$$X_{fout}(kf_{ref}) = \sum_{n=0}^{N-1} X_{pun}(kf_{ref}) \quad (4.15)$$

By substituting the Equation (4.14) into Equation (4.15), one obtains,

$$X_{fout}(kf_{ref}) = X_{pu0}(kf_{ref}) \sum_{n=0}^{N-1} e^{j2\pi f_{ref} t_d(n)k} \quad (4.16)$$

Although the term $X_{pu0}(kf_{ref})$ usually has non-zero values for all values of k , it is possible for the output spectrum to be zero if the summation is zero for some values of k . In an ideal case, all delay stages are identical and the total delay of those delay stages is exactly one period of the reference signal (T), we have

$$t_d(n) = \frac{T}{N} \cdot n \quad (4.17)$$

Based on the assumption above, the summation term (denoted as $S(k)$) on the right of Equation (4.16) can be written as,

$$S(k) = \sum_{n=0}^{N-1} e^{j2\pi \frac{1}{N} \cdot nk} = \begin{cases} \frac{1 - e^{j2\pi k}}{1 - e^{j2\pi \frac{1}{N} k}} & \text{if } k/N \text{ is not an integer} \\ N & \text{if } k/N \text{ is an integer} \end{cases} \quad (4.18)$$

It is clear that $S(k)$ is zero for all values of k except for $k=0, N, 2N, \dots$, and it is these values of k for which $X_{f_{out}}(kf_{ref})$ has non-zero values. So the fundamental frequency of f_{out} is N times the reference frequency.

This is the objective in frequency synthesizer design and most of design work is directed to make a real circuit to satisfy Equation (4.17) as closely as possible. This means the total delay of all delay stages should be exactly one period of the reference and each delay stage should have identical delay. Also, the pulse combiner should be as symmetric as possible to avoid introducing further spurious output.

4.4 Basic components of DLL-based frequency synthesizers

Now that the operating principle of a DLL has been introduced, the components of a basic DLL can be discussed individually. Although a real synthesizer may have additional components, the basic principle remains the same.

4.4.1 Voltage-controlled delay line

Figure 4.6 shows a VCDL with multiple delay stages. Ideally, delays of all delay stages are identical and controlled by a control voltage coming from the loop filter. The

number of delay stages is determined by the ratio of the output frequency to the input frequency, and the total delay of all delay stages determines the operating range of the frequency synthesizer.

4.4.2 Edge combining circuit

The edge combining circuit consists of pulse generators and a pulse combiner. One pulse generator per delay stage produces a pulse whose width is approximately one half of the stage delay (being triggered by rising edges of its input). Pulses from all of the generators are thus added together by the combiner. To minimize the output frequency spurious power, the design of the edge combining circuit should be as symmetric as possible.

4.4.3 Charge pump phase comparator

The charge pump phase comparator supervises the loop by tuning the delay of the VCDL to exactly one period of the reference signal. Its output is a charge current which either charges or discharges a capacitor (the loop filter) based on the phase difference between its two inputs. The characteristic of a linear charge pump phase comparator can be expressed as,

$$I_{out} = K_{cp} \cdot (\phi_{ref} - \phi_d) + I_0 \quad (4.19)$$

Where I_0 is the offset term which is undesired in DLLs because it results in an in-lock phase error of I_0/K_{cp} and this causes Equation (4.17) not to be satisfied exactly and there is a resultant spurious output. One must try to minimize I_0 as much as possible.

4.4.4 Loop filter

To avoid introducing additional poles, the first order loop filter is normally

employed for DLL-based systems. Normally, it consists of a capacitor used as an integrator, by changing the capacitance, one changes the locking speed, loop bandwidth, and so on.

4.5 Spur performance of the DLL-Based frequency synthesizers

The noise performance of the basic DLL has been analyzed in section “Noise performance of DLLs” on page 44. The following section analyzes the effect of the edge combination on the performance of a DLL-based frequency synthesizer.

As shown in Figure 4.6 on page 47, the final output of the frequency synthesizer is the sum of a set of multiple-phase pulses, whose phase variation mainly depends on the delay variation of each delay stage. With proper design and layout, those delay variations may be nearly identical over all delay stages and the total delay variation(ΔD) of the VCDL being N times of that of a single delay stage(ΔD_s),

$$\Delta D = N \cdot \Delta D_s \quad (4.20)$$

This variation is normally caused by the variation of the control voltage and it changes on each cycle of the reference signal. Figure 4.7 shows an example of the ΔD over cycles of the reference. The static error, average value of ΔD , is caused by the con-

Figure 4.7 Delay variation

stant in-lock error, which is normally a result of asymmetric design of the charge pump phase comparator.

The timing variation of the pulse generated by the n th delay stage can be expressed as,

$$\Delta T_n = \Delta T_{ref} + n \cdot \Delta D_s \quad (4.21)$$

Where ΔT_{ref} denotes the timing variation of the reference edge. By considering the timing variation shown in the equation above, Equation (4.17) is no longer satisfied, and it should be written as,

$$t_d(n) = \frac{T}{N} \cdot n + (\Delta T_{ref} + n \cdot \Delta D_s) + \sum_{i=0}^n \Delta D(i) \quad (4.22)$$

Where $\Delta D(n)$, which has a null mean, denotes the delay variation among multiple delay stages, which is mainly caused by the mismatch among delay stages. Corrupted by this timing error, the output spectrum is noisy, unlike the ideal spectrum described in “Operating principle of DLL-based frequency synthesizers” on page 47. It may look like that of Figure 4.8 (where the actual frequency is k^*f_{ref}), in which, there are many other

Figure 4.8 Relative power spectrum of a frequency output (N=10)

undesired frequency components at kf_{ref} besides the carrier frequency component (Nf_{ref}).

All such undesired components should be suppressed as much as possible, especially the two appearing at $(N-1)f_{ref}$ and $(N+1)f_{ref}$ since they are very close to the centre frequency. A parameter which gives the power ratio between such undesired components and the centre frequency would be a good figure of merit to use in the characterization of frequency synthesizer. Of course, this parameter should be minimized.

This section analyzes the power ratio of the component at $(N-1)f_{ref}$ (or at $(N+1)f_{ref}$ if it is larger) to the centre component at Nf_{ref} . To simplify the analysis, the term $X_{pu0}(kf_{ref})$ in Equation (4.16) is assumed to be constant for $k=N-1, N, N+1$, so that the summation term shown in Equation (4.23) can be analyzed separately to obtain an estimate of the spurious power level.

$$S(k) = \sum_{n=0}^{N-1} e^{j2\pi f_{ref} t_d(n)k} \quad (4.23)$$

Consequently, the relative spurious power can be approximately expressed as,

$$R_{spur} = \max\left(\left|\frac{S(N-1)}{S(N)}\right|^2, \left|\frac{S(N+1)}{S(N)}\right|^2\right) \quad (4.24)$$

In most cases, the two sideband spurs are in almost the same power level and a further simplification is possible,

$$R'_{spur} = \left|\frac{S(N-1)}{S(N)}\right|^2 \quad (4.25)$$

Using the equations above, the relative spurious power level at the output of a DLL-based frequency synthesizer can be analyzed theoretically as follows.

Case I. $\Delta T_{ref} = 0$ and ΔD_s is constant, and all delay stages and edge combining

circuit are assumed to be ideal and perfectly matched ($\Delta D(i) = 0$).

This is a very common case that the total delay of those multiple delay stages is not exactly equal to one period of the reference, and the in-lock error, the difference between the total delay and the reference period, is evenly distributed over all delay stages. In this case, Equation (4.22) can be expressed as,

$$t_d(n) = \left(\frac{T}{N} + \Delta D_s \right) \cdot n \quad (4.26)$$

The phase alignment error for the case where N equals to 9 is shown in Figure 4.9.

Figure 4.9 Phase alignment error for case I

Substituting Equation (4.26) into Equation (4.23), the spurious power can be calculated by squaring the value of $S(k)$ shown in the following equation.

$$S(k) = \sum_{n=0}^{N-1} e^{j2\pi f_{ref} \left(\frac{T}{N} + \Delta D_s \right) \cdot nk} = \frac{1 - e^{j2\pi k j2\pi f_{ref} \Delta D_s \cdot Nk}}{j2\pi f_{ref} \left(\frac{T}{N} + \Delta D_s \right) k} \quad (4.27)$$

A sketch of the relative output power spectrum may be produced by calculating the value of $S^2(k)$ for different k . As an example, Figure 4.10 shows a calculated power spectrum for $N=10$ and $\Delta D/T_{ref} = 0.01$.

Figure 4.10 Relative power spectrum for Case I

From the Figure, we can see that the power at $(N-1)f_{ref}$ is larger than $(N+1)f_{ref}$. More generally, according to the Equation (4.25), the relative sideband spurious power level should be calculated as,

$$R_{spur} = \frac{|S(N-1)|^2}{|S(N)|^2} \quad (4.28)$$

Using ΔD_r as a symbol for the ratio $\Delta D/T_{ref}$, the output spurious power level is plotted as a function of number of delay stages (N) in Figure 4.11. One can see that the spurious power level increases along with the relative variation and for a given relative variation, the use of larger number of delay stage results in larger spurious output level. If the variation is so large that one period of the reference is closer to $(N-1)$ stage delays than to N delays, the power at $(N-1)f_{ref}$ may be larger than the power at Nf_{ref} , showing as

Figure 4.11 Relation of spurious power level to the in-lock error positive (larger than unity) relative power level in Figure 4.8.

In the case when $\Delta D_r < 0$, a similar phenomenon appears, but the power at $(N+1)f_{ref}$ is larger than the power at $(N-1)f_{ref}$ and the relative spurious power level should be calculated as,

$$R_{spur} = \frac{|S(N+1)|^2}{|S(N)|^2} \quad (4.29)$$

Furthermore, the relative spurious power level increases with the increase of the absolute value of ΔD_r .

Case II. $\Delta T_{ref} = 0$ and $\Delta D_s = 0$, but $\Delta D(n) \neq 0$. There are some variations among delay stages while the in-lock error is zero.

To simplify the analysis, the stage variation $\Delta D(n)$ is supposed to be linear and can be written as,

$$\Delta D(n) = nT_{ref}k_r + \Delta D_0 \quad (4.30)$$

Where k_r is the relative slope of the delay variation, or the ratio of the slope to the reference period. Since the average value of $\Delta D(n)$ ($0 \leq n \leq N-1$) is zero, the term ΔD_0 in the Equation should be equal to $-(N-1)T_{ref}k_r/2$. In this case, the phase variation of the output signal is still periodic with the frequency equal to the reference frequency, so it also produces some spurs as in the Case I.

By substituting Equation (4.30) into Equation (4.22) with $\Delta T_{ref} = 0$ and $\Delta D_s = 0$, one can obtain,

$$t_d(n) = \frac{T}{N} \cdot n + \sum_{i=0}^n \Delta D(i) \quad (4.31)$$

The phase alignment error of those multiple-phase pulses is shown in Figure 4.12.

Figure 4.12 Phase alignment error for case II

Although the stage delay variation is linear, the largest phase alignment error occurs at the middle of the delay line if the in-lock error is zero. This phenomenon also occurs in the case where the stage-to-stage uncertainties are random [16].

By substituting Equation (4.30) into Equation (4.31), and considering Equation (4.23), the relative power spectrum of the output can be calculated approximately. Figure 4.13 shows a relative power spectrum with $N=11$ and $k_r=0.001$. The desired frequency component ($f_{out}=11*f_{ref}$) is at $k=11$ (46.9dB). Two sidebands appear at

Figure 4.13 Illustration of the spurious output

$k=10$ (15.0dB) and $k=12$ (18.4dB), and the relative spurious level is -28.5dB. This value depends on the value of N and k_r . It can be observed that the power at $(N+1)f_{ref}$ is larger than the power at $(N-1)f_{ref}$ in this case. According to Equation (4.24), the spurious power level could be expressed as,

$$R_{spur} = \left| \frac{S(N+1)}{S(N)} \right|^2 \quad (4.32)$$

To show more general relation between the spurious power level and the delay variation (denoted as the slope k_r), the relation of N , k_r versus the relative spurious level is shown in Figure 4.14.

When the slope k_r is very small, the sideband spurious power level increases quickly with the increase of the slope (k_r) of the delay variation. But it does not continue to increase when slope k_r is large as that in Case I, where spurious power level keeps increasing with the increase of D_r . The reason is that in Case II, the undesired energy is distributed over a number of frequency components while in Case I, it only goes to a single sideband spur.

Figure 4.14 The relation of N , k_r vs. relative spurious level

Case III. $\Delta T_{ref} \neq 0$ and $\Delta D_s = 0$. Here, ΔT_{ref} is assumed to change randomly from one reference cycle to the next.

Due to the nature of VCDL, this timing variation propagates through multiple delay stages with almost same timing deviation for each N cycles of the output signal. Because this type of variation does not repeat with the reference frequency, it shows up in the output spectrum as white phase noise instead of some spurs.

ΔT_{ref} is related to the phase variation of the reference by,

$$\Delta \phi_{ref} = 2\pi f_{ref} \cdot \Delta T_{ref} \quad (4.33)$$

If there is no additional noise introduced in the delay stages and the edge combining circuit, the timing variation of the output edges should be the same as that of the refer-

ence. The phase variation of the output can be written as Equation (4.34), which is N times

$$\Delta\phi_o = 2\pi N f_{ref} \cdot \Delta T_{ref} \quad (4.34)$$

the phase variation of the reference, so the phase noise power can be N^2 times reference noise power and the output phase noise can be expressed as,

$$\text{Output Phase Noise}(f)[dB] = \text{Input Phase Noise}(f)[dB] + 20\log N \quad (4.35)$$

This relation is very similar to that of a PLL-based frequency synthesizer which employs a digital frequency divider. But in the case of the PLL, the output noise will be greater because the timing variation of the output edges normally is larger than that of reference edges.

Three representative cases are analyzed separately above. However, it must be kept in mind that the effects always affect the output simultaneously in a real circuit. There are also some other noise sources such as some random variation (with respect to time) of each delay stage and noisy power supply and so on, but the three cases analyzed above cover the most important circumstances arising in DLL-based frequency synthesizer design. They are several unique characteristics for this class of frequency synthesizers.

In the frequency synthesis application, the reference signal is normally derived from a low-noise source and the timing variation of the reference edges is small compared to those having origins in the frequency synthesizer itself. So the phase noise analyzed in Case III can be quite small in comparison to others.

The phenomena considered in Case I, II lead only to spur production. The stage

mismatch among delay stages, considered in Case II, can be minimized by careful buffer design and layout. The delay deviation considered in Case I can be minimized by using high performance phase detector and charge pump.

4.6 Low noise frequency synthesizer design

According to the analysis above, to achieve a low-noise DLL-based frequency synthesizer employing an edge combining circuit, including low phase noise and low spurious power level, besides the low-noise design issues similar to the design of PLL-based ones, it is very important issue to minimize the mismatch among delay stages and the in-lock error. As for the loop filter, since the second order loop filter offer no benefit for performance of DLL [17], a first order loop filter is normally employed. Consequently, the stability of a DLL is normally guaranteed.

4.7 Summary

In this chapter, the fundamental characteristics of a DLL are analyzed, and factors influencing its noise performance, including the spurious power level, are discussed. The relations of spurious power level, phase noise, stage mismatch, in-lock error and so on are analyzed theoretically. Several important issues affecting DLL-based frequency synthesizer design are introduced.

A DLL-based frequency synthesizer whose design takes advantage of this chapter's findings will be proposed and its implementation described in the following chapters.

Proposed architecture of the DLL-based frequency synthesizer

5.1 Introduction

In this chapter, the previous analyses of PLL/DLL frequency synthesis technologies are used to compare the two technologies briefly, after which the basic architecture of a DLL-based synthesizer is proposed. Then the requirements of each of its blocks are discussed.

5.2 Comparison of PLL/DLL based frequency synthesis

As the analyses of the previous two chapters show, PLLs and DLLs have a number of features in common. They also differ in important ways - those most relevant to frequency synthesis applications are described below.

Because a DLL normally has fewer poles in its forward path than does a PLL and most DLLs have first-order structures while PLLs are second-order loops, these DLLs are unconditionally stable, and very careful design is needed to ensure the stability of a PLL. So DLL design is on the whole more easily done than PLL design.

There are advantages and disadvantages if various kinds of noise sources are concerned. To external noise originating from the reference signal, the PLL acts as a low pass

filter and a properly-designed loop filter can lead to effective suppression of this noise. But a DLL may amplify external noise no matter what kind of loop filter is used. This characteristic enables the PLL to function in noisy environments, such as clock data recovery (CDR). But it is not very important in a frequency synthesis application since the reference signal is usually taken from a low-noise source.

In the case of the internal noise originating from the VCO (PLL) or from the VCDL (DLL), DLLs have a unique advantage in that no cycle-to-cycle noise accumulation takes place during operation since they do not employ an internal feedback loop, as does a VCO. This makes the DLL a very good choice for frequency synthesis applications.

There are some other considerations when comparing the two methods, such as output spurious power level, operating frequency range, power consumption and others. all of which are summarized in Table 5.1.

TABLE 5.1: Comparison of PLL/DLL for the frequency synthesis application

Characters	PLL	DLL
Stability	The stability is a big issue for the PLL design and the loop bandwidth is normally limited by the stability.	Can be unconditionally stable and the loop bandwidth can be very large to suppress internal noise coming from the VCDL.
Ability to suppress external noise	Can effectively suppress external noise with small loop bandwidth.	No ability to suppress the external noise.
Ability to suppress internal noise	Internal noise is normally accumulated by the VCO to a relative large level, and can't be suppressed very effectively due to the limit of the loop bandwidth.	Can effectively suppress internal noise by enlarge the loop bandwidth and the internal noise doesn't accumulate.
Spurious power level	The spurious power level is normally lower than that of DLLs.	The spurious power level is normally higher than that of PLLs.
Output phase noise	The phase noise is normally higher than that of DLLs.	The phase noise is normally lower than that of PLLs.
Main design challenges	Stability consideration, loop filter design, loop bandwidth optimization.	Mismatch problem within the VCDL, lock error originated from the charge pump phase comparator.

It is the DLL which makes a superior choice in frequency synthesis applications, especially for the noise performance. For this reason, a DLL was chosen as the core of the new frequency synthesizer.

5.3 System specifications

The proposed DLL-based frequency synthesizer will be required to operate under conditions typically found in a multichannel radio communications system. This means that its input frequency should have a relatively wide range in the vicinity of 180MHz. The ability to multiply the input frequency will be demonstrated by requiring that the DLL's output frequency be nine times the input frequency. The design will be directed toward minimizing the phase noise and spurious power level in the output.

The circuit is to operate with a 1.8V power supply. And it is to be fabricated using a 0.18 μ m CMOS process.

5.4 System block diagram of the DLL-based frequency synthesizer

Based on the principle of the DLL-based frequency synthesizer and the specifications, the block diagram of the DLL-based frequency synthesizer are proposed as shown in Figure 5.1. The block at the bottom is an important block of this frequency synthesizer which generates the output signal (f_{out}) by combining multiple-phase signals coming from multiple delay stages. It also includes the input buffer to obtain uniform signal shape within delay stages and the output buffer to gain more drive ability so that the signal can pass through the output pads, bonding wire, package pin, PCB connection and drive the actual load of this frequency synthesizer. All other blocks are used to tune the control voltage of VCDL so that the total delay of 9 delay stages are close to one period of input sig-

Figure 5.1 System block diagram of the DLL-based frequency synthesizer

nal (f_{in}) as possible. As a result, the performance of the frequency synthesizer greatly depends on the performance of the phase comparison circuit because it supervises the loop and guarantee the delay to be locked to the correct value. According to the system specifications, the specifications for each block can be given as follows.

5.5 Specifications of each block

5.5.1 Voltage-controlled delay line

Since the average input frequency of this frequency synthesizer is 180MHz, and the output frequency is 9 times the input frequency, the main part of the VCDL includes nine identical delay stages having the average total delay of one input period (5.56ns). The nominal delay of each of these stages is one-ninth the total, or about 617pS. This average delay should coincide with the control voltage of 0.9V (the average control voltage, one half the supply voltage), and the output pulse width of each stage is one half of its delay.

Due to the threshold voltage of the transistor, to guarantee the proper work of the

charge pump, the range of the control voltage can not be too large since the supply voltage is 1.8V. In this design, this range is about 0.4V (from 0.7V to 1.1V). Within this range, the corresponding delay range of the delay stage directly affects the operating frequency range of this frequency synthesizer. The detailed specifications for the VCDL are listed in Table 5.2.

TABLE 5.2: Specifications of the VCDL

Supply voltage	1.8V
Average control voltage	0.9V
Range of control voltage	0.7V to 1.1V
Average delay of each delay stage	617ps
Delay range of each delay stage	350ps to 900ps
Number of stages for combining	9
Width of output pulse for each stage	About 1/2 of its delay

5.5.2 Pulse combining circuit

There are nine pulses, whose width is one half the stage delay, coming from 9 delay stages separately which are combined together to generate the output of the frequency synthesizer by the pulse combining circuit. So there are nine inputs and one output for the pulse combining circuit which should combine nine pulses as symmetrical as possible to avoid introducing more spurious output, and the supply voltage is the same as other parts, which is 1.8V.

5.5.3 Input buffer

The purpose of the input buffer (of the VCDL) is to precondition the input signal

so that all of the delay stages can work in identical environments. Without the input buffer, the in-lock error would be difficult to control and different wave shapes at the delay stage outputs would result in a high spurious power level. So the input buffer should be designed as close to the actual delay stages as possible.

5.5.4 Output buffer

The output of the synthesizer is the output of the pulse combining circuit, which has a low drive capability and can not drive the load presented by its external connections, which include not only the actual electronic load, but the capacitance of a PCB connection and a wire connecting it to the load. The output buffer is included in this design to enhance the drive ability.

5.5.5 Phase comparison circuit

The phase comparison circuit, including the charge pump phase comparator and lock detector and so on, is of key importance in this design because of its influence on the synthesizer's output jitter and the operating frequency range. A poor design of the charge pump phase comparator could be responsible for a large in-lock error, which in turn would lead to large output jitter and high spurious power level. Please refer to "Spur performance of the DLL-Based frequency synthesizers" on page 51 for more information about these matters.

The PLL synthesizer doesn't have this problem, because the phase of the VCO is not necessarily the same as that of the input signal, and the resulting in-lock 'error' does not affect loop operation as long as the phase difference remains constant.

But in a DLL, the in-lock error must be as small as possible. In the proposed syn-

thesizer, a novel approach, one intended to achieve small in-lock error, is employed in the charge pump phase comparator.

5.5.6 Loop filter

The loop filter of a first-order DLL should be designed in such a way that additional poles are not introduced, as these could result in a unstable loop. A capacitor attached to the output of the charge pump phase comparator is an adequate and 'safe' filter.

5.6 Summary

This chapter begins with a comparison of two frequency synthesis technologies - the PLL-base one and the DLL-based one. It was found that each has its merits, and the DLL has unique advantages for frequency synthesis application. Since it has been decided to pursue the DLL idea further in this thesis, to improve its performance in general, a proposed structure of a DLL frequency synthesizer has been described and the specifications for each functional block of that structure discussed. The components most important to the DLL's noise performance, the voltage-controlled delay line and the phase comparison circuit which will be studied in detail at the circuit schematic level in the following chapters.

Voltage-controlled delay line and edge combiner

6.1 Introduction

The voltage-controlled delay line and the edge combiner are important components of the DLL-based frequency synthesizer. Their designs affect the performance and some specifications of the frequency synthesizer, such as the operating frequency range, the inter-period jitter of the output, the frequency ratio of the output frequency to the input frequency and so on. This chapter focuses on these two parts in the frequency synthesizer design.

6.2 Structure of voltage-controlled delay line

As described in the previous chapter, the function of the VCDL is to provide multiple-phase signals to the edge combining circuit which uses them to generate the synthesizer output. These multiple-phase signals are also applied to the lock detection circuit to guarantee correct locking. Also the total delay of the VCDL is locked to one period of the reference signal by the phase comparison circuit and the loop filter. Consequently, the VCDL should include multiple voltage-controlled delay stages with identical delay of each.

Ideally, a single-ended tuning signal of the VCDL, coming from the loop filter, would be linearly related to the delay of the VCDL, but the voltage controlled delay cells (VCDCs) normally require more than one control signal and the relation between its control signals and the delay may not be always linear one. A driver circuit is thus employed to convert one single control signal to one or two signals to tune the delay of the VCDC. In addition, the driver circuit can be built to help linearize the relation between control signal and the delay. Figure 6.1 shows the structure of a VCDL.

Figure 6.1 Structure of VCDL

As we can see, the VCDL design can be split into two parts, design of the delay stage and design of the driver circuit. Both will be described in the following sections.

6.3 Design of the voltage-controlled delay stage

The VCDL contains multiple identical delay stages, each of which is made up of one or more VCDCs. The design of the VCDC is therefore a key issue in VCDL design. The VCDCs of a VCDL should have a wide tuning range and should respond in exactly the same way to a common tuning voltage, in other words, the variation in delay from one delay stage to another must be as small as possible. Of course, some other types of noise, such as flicker noise, must be also considered by the designer.

A VCDC is the basic delay element of a VCDL and it outputs a delayed version of its input signal with the delay tuned by control signals. In general, most of the delay of a VCDC is due to the charging and discharging of capacitors. There are essentially two ways of tuning the delay, one by adjusting the charging (or discharging) current and the other by adjusting the capacitance itself. The former adjustment is often achieved by tuning a “resistor”, which is normally an appropriately connected transistor.

Many types of delay cells operating on various principles have been reported in the literature. A number of typical ones are reviewed in the following sub-sections.

6.3.1 Differential delay cells

A simplified structure of a differential delay cell is shown in Figure 6.2-A. Both its

Figure 6.2 A differential delay cell

input and output operate in a differential mode with the voltage transfer characteristic shown in Figure 6.2-B.

When V_{in+} is equal to V_{in-} , the current I_{bias} is separated equally into two branches,

$$V_{out+} = V_{out-} = V_{dd} - \frac{R_L \cdot I_{bias}}{2} \quad (6.1)$$

This kind of differential structure finds many applications and most of them operate in a small-signal mode. The voltage gain from its input to output is related to the parameters of the transistors and load resistors, which is approximately,

$$A_v = \frac{V_{out}}{V_{in}} = g_m \cdot R_L \quad (6.2)$$

Where $V_{in} \equiv V_{in+} - V_{in-}$ and $V_{out} \equiv V_{out+} - V_{out-}$. However, this small-signal approximation is not very important in DLL applications since the signal propagating through a delay cell is normally of large amplitude and the transistors switched deeply. In such a case, Equation (6.2) no longer holds.

To simplify the analysis, the input is assumed to have a very short transition time, with the transistors being switched rapidly at the input edges. Under this assumption, most delay of the circuit will be due to the charge and discharge time of those parasitic capacitors of active circuit elements. A differential model including dominant parasitic capacitances is shown in Figure 6.3-A. Although there are many parasitic capacitors in an actual device, all of them can be replaced with two equivalent capacitors C_{eq1} and C_{eq2} for the purpose of the delay analysis presented here.

When V_{in+} is high and V_{in-} is low, M_1 is on and M_2 is off, so the current transiting R_{L1} will be equal to I_{bias} while the current passing through R_{L2} will be zero. Consequently, the outputs are $V_{out+} = V_{dd}$ and $V_{out-} = V_{dd} - R_L \cdot I_{bias}$.

A negative (positive) transition of the input signal, V_{in+} (V_{in-}) now causes the tran-

Figure 6.3 A differential delay cell model

sistor M_1 to be turned off and M_2 on, resulting in that the I_{bias} passes through M_2 instead of M_1 . So the output V_{out+} will be pulled down while V_{out-} goes up, as shown in Figure 6.3-B. The delay is defined to be the duration from the transition edge to the time when V_{out+} and V_{out-} are equal (or when the $V_{out} \equiv V_{in+} - V_{in-}$ is zero). V_{out} can be expressed as,

$$V_{out}(t) = V_{in+}(t) - V_{in-}(t) = I_{bias} R_L \cdot (1 - 2 \cdot e^{-t/\tau}) \quad (6.3)$$

Where the time constant $\tau = R_L \cdot C_{eq}$. On solving the equation $V_{out}(t)=0$, the delay of the differential structure, T_{delay} is found to be,

$$T_{delay} = \ln(2) \cdot R_L \cdot C_{eq} \quad (6.4)$$

In practice, the delay of the structure of Figure 6.3 may be slightly larger than the result above because the actual rise and fall time of the input signals can not be infinitely short.

Equation (6.4) shows that the delay time is directly related to the load resistor R_L and to the equivalent capacitance C_{eq} . Since C_{eq} is mainly related to the parameters of the

transistor M1 or M2 and its value is normally constant, a common way of delay tuning in this case is making the value of R_L controlled by a control voltage $V_{control}$. In practice, this is achieved by replacing the R_L with a transistor-based structure acting as a voltage-controlled resistor.

From Equation (6.1), the common-mode voltage of the output is related to R_L linearly, so a change in R_L will cause a change to both the common-mode voltage and the voltage swing of the differential signal. This is a disadvantage of the differential structure because it may affect the operation of other parts of the frequency synthesizer such as the edge combining circuit. Since the input signal is normally single-ended, a special circuit must be employed to convert it to a differential signal. This will increase the chance of introducing additional noise. The common-mode voltage of the converter output is normally fixed and it can not follow the common-mode voltage of the differential delay cell as it changes with the control voltage. This mismatch increases the mismatch of the delay among multiple delay cells and finally results in a high spur frequency content in the frequency synthesizer output. Another disadvantage of this type of VCDL is that its power dissipation is normally large. A constant current I_{bias} flows in each delay cell all the time, meaning that each delay cell consumes a power of $V_{dd} \cdot I_{bias}$. This results in a large power dissipation of VCDL constructed with this kind of delay cell.

Although the differential structure has its inherent ability to eliminate the common-mode noise coming from the power supply, it is not always an ideal choice for the application of DLL-based frequency synthesizers because of the disadvantages discussed above.

Another class of delay cell, the inverter-based delay one, is analyzed in the follow-

ing section.

6.3.2 Inverter-based delay cells

An inverter-based delay cell takes use of the inherent delay of a widely used CMOS inverter. Although there are some additional elements to tune the delay in an actual inverter-based VCDC, analysis of an inverter, the fundamental element of the VCDC, will be helpful.

The analysis of an inverter begins by replacing the parasitic capacitors shown in

Figure 6.4 An inverter-based delay cell

Figure 6.4-A with an equivalent set of 3 capacitors, C_i , C_{io} , C_o as shown in Figure 6.4-B. Since a rising edge of the output corresponds to the charging of the output capacitors C_o and C_L , by the current flowing from R_p , and a similar falling output involves the discharge of these same capacitors through the resistor R_n . The rising delay and the falling delay (the 'rising' and 'falling' refer to the output of the inverter) are normally different because the R_p and R_n are not exactly equal.

If the input V_{in} is provided by a low-impedance source, the effect of C_i can be ignored in approximate calculations. Now suppose that the input signal V_{in} is rapidly switched from 0 to V_{dd} . At the time right after the transition edge, the output $V_{out}(0)$ is calculated as,

$$V_{out}(0) = V_{dd} \left(1 + \frac{C_{io}}{C_{io} + C_o + C_L} \right) \quad (6.5)$$

This value is slightly higher than V_{dd} , a phenomenon called overshoot. Then, 3 capacitors C_i , C_{io} , C_o are discharged through R_n and the output voltage falls from its initial value $V_{out}(0)$. By denoting $C_{total} \equiv C_{io} + C_o + C_L$, $V_{out}(t)$ can be expressed as,

$$V_{out}(t) = V_{out}(0) \cdot e^{\frac{-t}{R_n C_{total}}} \quad (6.6)$$

Where the product $R_n C_{total}$ is defined to be the time constant for the falling delay.

Figure 6.5 Waveform of the output of an inverter delay cell

The output is shown in Figure 6.5-A and the time needed for the output voltage to drop to one half of V_{dd} is defined as falling delay (T_{fd}). With Equation (6.5) and Equation (6.6),

the fall delay can be obtained as,

$$T_{fd} = R_n C_{total} \cdot \left[-\ln \frac{1}{2} \left(1 + \frac{C_{io}}{C_{total}} \right) \right] \quad (6.7)$$

The falling delay can be tuned by changing the value of R_n and the rising delay (as shown in Figure 6.5-B) can be tuned by changing the value of R_p . Thus an inverter-based voltage-controlled delay cell can be built as shown in Figure 6.6, where the resistances

Figure 6.6 An inverter-based voltage-controlled delay cell

have been partly replaced by two transistors M_3 and M_4 . The gate of M_3 is controlled by a control voltage (V_{ctrl-}), which is a part of R_n shown in Figure 6.4. An analogous replacement of R_p has been made with a PMOS transistor M_4 controlled by V_{ctrl+} .

As shown above, the rising delay and falling delay of this type of delay cell depend on the V_{ctrl+} and V_{ctrl-} respectively, and the duty cycle of the delayed signal (V_{out}) may be different from that of the input (V_{in}) even if the V_{ctrl+} and V_{ctrl-} are balanced by a driver circuit. This duty cycle distortion should be kept in mind and effort should be made to compensate it when the actual delay stage is designed.

6.3.3 Design of delay stage

The VCDL delay stage detailed in this section is derived from the basic VCDC shown in Figure 6.6. From the Equation (6.7), the load capacitor C_L , an important part of C_{total} , affects the delay greatly. If the load is just another identical delay cell, all delay cells will have identical loads and thus the same delays, which is desired for the edge combining operation to generate a low-spur output. However, each output of the delay chain may be connected to different load (outside of a VCDL core) in an actual design. It is necessary to add a buffer at the output of each delay cell to alleviate the effect of an unbalanced load. For convenience, the buffer is also considered as a part of the delay cell in this design.

To compensate the duty cycle deviation mentioned in the previous sections, a couple of inverters are used as one buffer. By this way, the output of each delay cell, an inverted and delayed version of the input, is applied to the next identical delay cell, in which the previous rising edge and falling edge are switched. So for each two consecutive delay cells, the rising delay is the sum of the falling delay of the first one and the rising delay of the second one, while the falling delay is the sum of the rising delay of the first one and the falling delay of the second one. This means that the rising delay and falling delay of a delay stage, including two identical delay cells connected in series, are the same and the duty cycle remains unchanged.

For the purpose of edge combining, the delay stage in this design consists of two identical delay cells to obtain half of the stage delay, which is the width of the pulse generated from each delay stage (Refer to “Voltage-controlled delay line” on page 65 for details). As a result, a delay stage is implemented as shown in Figure 6.7. This delay stage structure itself has ability to compensate the duty cycle distortion as discussed above.

Figure 6.7 Schematic of the voltage-controlled delay stage

Two extra buffers also introduce some additional delay. However, most of the stage delay is due to two voltage-controlled inverters. All transistors involved in the delay stage should be sized according to the required delay range, which is specified in Table 5.2 on page 66.

6.4 Design of driver circuit

As shown in Figure 6.7, two control voltages (V_{ctrl+} and V_{ctrl-}) are needed by the delay stages. The control voltage from the loop filter is however single-ended. It is the purpose of the control voltage driver circuit to perform the conversion.

Figure 6.8 Schematic of the driver circuit

The schematic of the driver circuit used in this design is shown in Figure 6.8. The input $V_{control}$ is connected to the gate of M_1 and it controls the current through that transistor. A diode-connected PMOS transistor M_2 converts the current to a voltage (V_{ctrl+}). The current is mirrored to M_3 and then converted to another voltage (V_{ctrl-}) by another diode-connected NMOS transistor M_4 .

If the $V_{control}$ is below the threshold voltage of the transistor M_1 , there will be no current flowing through M_1 , and the outputs V_{ctrl+} and V_{ctrl-} are floating, and the delay stage wouldn't work properly. This situation is avoided by placing a resistor R_1 in parallel with M_1 to maintain a minimum current if the transistor M_1 is completely off.

6.5 Design of edge combining circuit

The edge combining circuit is employed to generate the final output of the frequency synthesizer, a frequency which is N times the input frequency, by combining the multiple-phase signals for delay stages. To minimize the output timing jitter, a highly symmetrical structure is used in this design.

The circuit can be considered as having two parts, one, called pulse generator, generates one pulse from each delay stage and the other one, called pulse combiner, combines all pulses together.

The pulse generator can be implemented as part of each delay stage. A logical operation (AND) is used to produce a pulse whose width is half of the stage delay, as shown in Figure 6.9-A. These pulses are combined using a logical operation (OR) as shown in Figure 6.9-B. To maintain the symmetry of the edge combining circuit, a wire-OR logical gate is employed instead of a standard one, and the paths from each pulse (P_{un})

Figure 6.9 Pulse generator and pulse combiner

to the pulse combiner output (f_{out}) are identical.

Certainly, a buffer is necessary at the output of the wire-OR logical gate to ensure enough driving capability and output voltage swing. The schematic of the pulse combiner is shown in Figure 6.10.

Figure 6.10 Schematic of the pulse combiner

6.6 Simulation of the VCDL and edge combiner

The most important VCDL parameter is the delay, which depends strongly on the parasitic capacitance. VCDL requirements are listed in Table 5.2 on page 66. All transistors should be sized so that the specifications can be satisfied as closely as possible.

In order to verify the characteristics of the VCDL and edge combiner described above, simulations were done at the transistor level with the simulator spectreS and

0.18 μm CMOS models. It was confirmed that the requirements were met.

6.6.1 Simulation of the driver circuit

The driver circuit performs conversion from the one single-ended signal ($V_{control}$) to two control signals ($V_{ctrl+/-}$). Its voltage transfer characteristic (VTC) has a significant effect on the relation between $V_{control}$ and VCDL delay. However the sizing of the driver circuit can not be performed separately because it depends on the characteristics of the delay cell and the final goal is to obtain the desired relation between the $V_{control}$ and the delay. The initial simulation of the driver circuit (shown in Figure 6.8 on page 79) is done by sweeping the DC voltage of its input ($V_{control}$) and the result is shown in Figure 6.11.

Figure 6.11 Simulation result of the driver circuit

6.6.2 Simulation of the delay stage

By combining the driver circuit and the delay stage, the relation between the $V_{control}$ and the delay of each delay stage can be simulated and the result is shown in Figure 6.12.

6.6.3 Simulation of edge combining circuit

Simulation of the edge combining circuit was done by setting the $V_{control}$ to a con-

Figure 6.12 Simulation result of the delay stage

stant value and observing the waveforms of those delayed signals, the outputs of those pulse generators and the output of the pulse combiner. Figure 6.13 shows the output of

Figure 6.13 Simulation of edge combining circuit

pulse combiner when the $V_{control}$ is equal to 1.0V. The combining operation is clearly shown in this figure.

6.7 Summary

This chapter firstly reviews several typical delay cells. After comparison of those delay cells, an inverter-based voltage-controlled delay cell, which is suitable for the proposed frequency synthesizer, is employed. The detailed design of the whole delay line and

the edge combining circuit is given. Finally, some transistor-level simulations are done to confirm the functionality of the voltage-controlled delay line and edge combiner.

Charge pump phase detector and lock detector

7.1 Introduction

The key operation in a DLL-based frequency synthesizer is to tune the VCDL delay. The tuning voltage is generated by a phase comparison circuit and the loop filter. The performance of the charge pump phase comparator, a key component of the phase comparison circuit, is critical because it directly affects the in-lock error and thus the output phase noise and spurious power level.

Most phase detectors reported in the literature are designed for PLL applications, where a constant in-lock error has no effect on the loop operation as long as it is within a reasonable range. However, as analyzed previously, this is not true in the case of the DLL.

This chapter deals with the design of the phase comparison circuit, which includes the a charge pump phase detector and a lock detector. Firstly, some typical phase detectors and charge pumps are reviewed, then their disadvantages are discussed. A novel structure is then proposed, one designed to overcome those disadvantages. A theoretical analysis is done, followed by a detailed design, and the performance is verified by simulating the design with the simulator spectreS and 0.18 μ m CMOS models.

7.2 Some typical phase detectors

The phase detector (PD) is a component with its output proportional to the phase difference between its two inputs. Mathematically, if two input signals represented by Equation (3.1) and Equation (3.2) are presented to a phase detector, its output is the signal given by Equation (3.3), which is a continuous function of the time. However, in practice, most phase detectors operate in a “discrete-time” mode, which means that the phase detector compares the phases of its two inputs only at the transition edges (for example, rising edges) of the input signals, and its output represents the timing difference between two edges.

The most common PD types are reviewed as follows,

Multiplier If both inputs of the phase detector are sinusoidal signals presented by Equation (3.1) and Equation (3.2) on page 20. The product of two inputs signals can be expressed as,

$$v_{pd} = AB[\sin(\phi_i - \phi_o) + \sin(2\omega t + \phi_i + \phi_o)] \quad (7.1)$$

The high frequency term can be removed by a low-pass filter, leaving the slowly-varying $AB\sin(\phi_i - \phi_o)$. When the loop is in or near lock, the approximation $\sin(x) = x$ can be used, and the output of the multiplying PD is directly proportional to the phase difference between input signals. Multiplying PDs are especially useful in high frequency applications.

XOR gate In the case when inputs are square waveforms, an XOR gate can be used as a phase detector - the average value of the output is proportional to the phase difference between its two inputs. The average output ranges from zero when the phase dif-

ference is zero to a maximum when the phase difference is 180° . The ‘operating point’ of such a phase detector is chosen to be a phase difference of 90° .

Flip-Flop: An RS flip-flop can be used as a phase detector suitable for square wave signals where a transition at one input (A, “SET”) sets the output to “1” and a transition at (B, “RST”) sets the output to “0”. The output of the flip-flop is thus a train of pulses whose widths are proportional to the timing difference between two inputs. The average value of the output pulse stream is proportional to the timing difference. This type of the phase detector has a linear range of one cycle.

Phase-frequency detector This type of PD is called the 3-state PD (sometimes, 4-state PD). It has two outputs, and contains a “memory”. A charge pump is normally necessary for this type of phase-frequency detector. A popular structure is implemented with two Flip-Flops and is detailed in the following section.

7.3 *A widely used phase detector*

Of many types of phase detectors in the literature, the commonly used ones are based on a typical structure, dual D flip-flop structure, which outputs two signals, UP and DN , two pulses whose width difference corresponds to the phase difference between its two inputs, P_{in1} and P_{in2} . Figure 7.1 shows the structure of a dual DFF phase detector. The rising edges of UP occur at the rising edges of the input P_{in1} , and the rising edges of DN occur at the rising edges of the input P_{in2} . When UP and DN are both high, both will be reset after a propagation delay involved in the AND gate and the DFFs, as shown in Figure 7.2.

Figure 7.1 A dual D flip-flop phase detector

Figure 7.2 Waveforms of the input and output of the phase detector

7.4 A typical charge pump

The outputs UP and DN drive a charge pump connected to a capacitor: UP charges the capacitor, and DN discharges it. Figure 7.3 shows a simplified representation of the

Figure 7.3 The principle of a typical charge pump

process. Two identical current sources switched by the two pulses (UP and DN) determine the gain of the charge pump. In the ideal case, the net charge transferred to the capacitor is

equal to the product of the current and the width difference between its two inputs (UP and DN).

Mathematically, the net charge can be calculated as,

$$Q_n = I_u t_{up} - I_d t_{dn} \quad (7.2)$$

where t_{up} and t_{dn} denote the pulse widths of UP and DN respectively. When $I_u = I_d$, Equation (7.2) can be rewritten as,

$$Q_n = I_u (t_{up} - t_{dn}) = I_d (t_{up} - t_{dn}) \quad (7.3)$$

In an ideal circuit, two current sources I_u and I_d are kept equal using current mirrors. And the switches are a PMOS transistor (for I_u) and a NMOS transistor (for I_d). Since a PMOS transistor is turned on when the gate voltage is low, an inverter should be employed to convert the high-enabled signal UP coming from the phase detector to the low-enabled signal UP' as the input of the charge pump.

7.5 Problems involved in the conventional circuitry

There are some problems involved in the conventional charge pump phase detector described above, especially for the DLL applications. The main problems are analyzed in this section.

Current mismatch An ideal charge pump phase detector requires that two currents I_u and I_d be equal so that Equation (7.3) can be satisfied. But all current mirrors have a mismatch problem, and also have finite output impedance, so it is impossible to guarantee that the two currents are equal.

Suppose that there is an error (I_e) between I_u and I_d (that is, $I_u = I_d + I_e$) and let the

relative current error (ϵ_r) be,

$$\epsilon_r = \frac{I_e}{I_d} \quad (7.4)$$

Equation (7.2) is then written as,

$$Q_n = I_d((1 + \epsilon_r)t_{up} - t_{dn}) \quad (7.5)$$

If there is no other sources of error in the circuit, and the DLL employing this type of charge pump phase detector is in lock (constant control voltage, or net charge is zero), the difference (t_e) between t_{up} and t_{dn} according to Equation (7.5) is,

$$t_e = t_{up} - t_{dn} = -t_{dn}\epsilon_r \quad (7.6)$$

This means that the in-lock timing error (please refer to “Spur performance of the DLL-Based frequency synthesizers” on page 51 for details about the effects of the in-lock timing error) is proportional to the current mismatch. But for a given current mismatch, the in-lock error is proportional to the PD output pulse width. The smaller the width of *UP* or *DN* is, the smaller the in-lock timing error is. On the other hand, the widths of *UP* and *DN* are increased by the delay of the reset circuit in the phase detector, a delay which is necessary to overcome the dead-zone problem. Therefore, the widths cannot be too small, which means that the in-lock timing error (t_e) is relative large because of the large value of t_{dn} . Suppose that the width of *DN* is 100ps, and there is a relative current mismatch of 5%, an in-lock error of 5ps is caused only by the current mismatch.

Pulse width variation of the inverter The switches for I_u and I_d are realized by PMOS and NMOS transistors, but both outputs of the PD have the same polarity (high-enabled). So an inverter has to be placed between the high-enabled output (*UP*) and the low-enabled PMOS gate. The presence of this inverter may distort the ON-time of I_u since

the rising delay and falling delay of an inverter are not always the same. For example, a typical inverter delay in CMOS 0.18 μ m technology is about 50ps, so a 10% relative error between the rising delay and falling delay results in an in-lock timing error of 5ps.

Control-voltage glitches The inverter inserted between the PD and the charge pump may cause another problem, namely glitches in the control voltage, even in the absence of the aforementioned pulse-width variation. This is a result of the phase shift between the signals *DN* and inverted *UP* due to the inverter delay. This effect is illustrated in Figure 7.4. As a result, the control voltage can not be ideally DC signal even if the loop

Figure 7.4 Glitch of the control voltage

is in lock and all components are noiseless. These control-voltage glitches will affect the delay of the VCDL stages and result in some spurious output and timing jitter.

Charge sharing problem Normally the two currents (I_u and I_d) are provided by two current mirrors constructed with PMOS transistors and NMOS transistors respectively. Figure 7.5 shows the schematic of the charge pump. When the UP switch is off, the drain of M_1 is pulled up to V_{DD} , and when UP switch is on, it is pulled down to $V_{control}$. A current larger than the I_{bias} flows at this time because of discharge of parasitic capacitance. An analogous situation exists in the case of the drain of the M_2 when the DN switch is tuned on. This effect is called “charge sharing” because the drains of M_1 and M_2 and the

Figure 7.5 The schematic of a typical charge pump

output capacitor (C_L) share the charge current I_u and I_d . The results of the charge sharing are some static in-lock phase error, and large fluctuation on the signal $V_{control}$ which may lead to large power spectrum spurs at the output of the frequency synthesizer.

All four problems mentioned are important in the case of DLLs, but only the last two affect the performance of PLLs because there is no need for the phase difference between the reference and the VCO output to be exactly zero. That is the reason for the wide use of this kind of charge pump phase detector.

7.6 Some measures to overcome those problems

By taking advantage of some new circuit techniques, designers have indeed made some headway in solving the above problems.

7.6.1 Reduction of control voltage glitches

To compensate the inverter delay responsible for $V_{control}$ glitches, 4 inverters are inserted into the DN path and 3 inverters into the UP path, as shown in Figure 7.6. Careful sizing of the inverters leads to a cancellation of the delay difference between UP and DN , thus the reduction of $V_{control}$ glitches.

The effectiveness of delay matching is dependent on parameter variations in the

Figure 7.6 A measure to reduce the glitch

chip process, and its feasibility is limited by the capabilities of existing technology. On the other hand, using this technique, it is more difficult to control the pulse width deviation since many inverters are connected in series. Furthermore, it does help to alleviate the I_d/I_u mismatch problem.

7.6.2 Reduction of the current mismatch

Figure 7.7 shows the schematic of another type of charge pump, in which the

Figure 7.7 Charge pump to alleviate the mismatch between PMOS and NMOS

inherent mismatch between NMOS and PMOS devices is avoided by using only NMOS switches [18]. In this circuit, two currents (I_u and I_d) are more easily matched, but the performance is still limited by the behavior of the PMOS current mirror.

7.6.3 Reduction of charge sharing

The charge sharing problem causes large static phase errors and output spurs. Figure 7.8 shows the schematic of the charge pump, which was designed to alleviate the

Figure 7.8 Charge pump to alleviate the charge sharing

charge sharing problem by employing an active amplifier [19][20]. The active amplifier holds the drains of the PMOS and NMOS switches at the same level as $V_{control}$ when UP and DN are both zero, and there is no need for charging the drains when either the UP or DN switch is turned on.

Although many techniques to overcome individual problems are available, none of them can solve all problems simultaneously, and some measures may even cause some new problems in the course of solving existing ones. Obviously, a new type of charge pump phase comparator is needed, one which has the solution inherent in its architecture to the problems experienced by conventional PDs.

7.7 A novel charge pump phase comparator

A new charge pump phase comparator design is put forward in this thesis, one by which existing conventional PD problems are solved to a large extent. Its operating precision mainly depends on the match of transistor characteristics instead of the actual transistor sizes, so it is insensitive to the parameter variations in the chip manufacturing process.

This novel charge pump phase comparator can achieve a very low in-lock error (at least 10 times lower than that of conventional techniques) and minimize the glitch on the control voltage.

7.8 Basics of the novel charge pump phase comparator

In this design, the charge pump phase comparator uses a new technique, in which the paths from P_0 and P_N to I_{out} (the output of the charge pump) are short, symmetrical and can be easily matched. The resolution of this circuit can thus be much higher than that of conventional circuits. The block diagram of the novel charge pump phase comparator is shown in Figure 7.9. The circuit acts on rising edges of P_0 and P_N . A pulse generator (PG) provides two negative pulses at these rising edges. The phase error can be obtained by

Figure 7.9 Block diagram of the novel charge pump phase comparator

provides two negative pulses at these rising edges. The phase error can be obtained by comparing the output of two integrators, IntU and IntD, placed at the output of the pulse generator. The integrators are reset by the rising edge of V_{win} and kept at zero until V_{win} returns to a low voltage in the next cycle. There is no need for making the widths of P_u and P_d very large to solve the dead-zone problem as it is necessary in conventional circuits. The width ratio of pulses P_u to P_d is therefore larger, easing the requirements for the phase comparator. A pulse control feedback (PCFB) adjusts the amplitudes and widths of P_u and P_d to ensure two integrators and the voltage comparator work correctly. It makes

the circuit insensitive to process parameter variations, temperature variation, and so on.

There are extra two inputs for the voltage comparator, V_{over} and V_{under} , coming from the lock detection circuit (described in the following section). If either of them is zero, the output of voltage comparator is controlled by the two signals, instead of V_u and V_d .

7.9 Lock detection and window generation circuit

Almost all phase comparators have an output proportional to the cycle-by-cycle phase difference between inputs, normally in the range of $\pm\pi$, and all other values of phase difference are actually shifted into this range by adding $\pm 2n\pi$ ($n > 0$). This characteristic could result in the DLL attempting to lock to zero delay or to a delay equivalent to multiple reference cycles if the initial phase error is not in the range of $\pm\pi$ or if there is interference due to a sudden power supply variation, a sudden reference signal phase shift, and so on. To solve this problem, more delayed signals ($P_0, P_1, \dots, P_N$) are used to enlarge the operating range of the whole phase comparison circuit. The lock detection circuit works as a complement to the phase comparator to give it an extended operating range, one actually larger than the tuning range of the VCDL, so that the loop can always lock correctly and there is no need for control voltage initialization when the system is powered on. An auxiliary function of the lock detector is to provide the signal V_{win} to the charge pump phase comparator.

The operating principle of the lock detection circuit is shown in Figure 7.10. Two signals, S_1 and S_2 , shown in Equation (7.7) and Equation (7.8), are sampled at the rising

$$S_1 = \sum_{n=0}^{N-2} P_n \overline{P_{n+1}} \quad (7.7)$$

$$S_2 = P_{N-1} \overline{P_{N+1}} \quad (7.8)$$

Figure 7.10 Operating principle of the lock detection circuit (1)

edges of signal P_0 to produce two signals S_{s1} and S_{s2} . By considering S_{s1} and S_{s2} , the total delay of the VCDL can be roughly compared to one period of the reference.

If the delay of the VCDL (T_{delay}) is in the range from $T_{ref}\frac{N}{(N+1)}$ to $T_{ref}\frac{N}{(N-1)}$, which means the T_{delay} is in lock range, S_{s1} is "0" and S_{s2} is "1". When T_{delay} is in the range from $T_{ref}\frac{N}{(N-1)}$ to NT_{ref} , which means that the delay is too large, S_{s1} is "1". When the T_{delay} is in the range from 0 to $T_{ref}\frac{N}{(N+1)}$, which means the delay is too small, both of S_{s1} and S_{s2} are "0".

Based on this principle, two signals (V_{over} , V_{under}) can be generated to indicate whether the loop is in lock ($V_{over}=1$ and $V_{under}=1$). Delays that are too large are indicated by $V_{over}=0$, and delays that are too small by $V_{under}=0$. Figure 7.11 shows how V_{over} , V_{under} are derived from the signals S_{s1} and S_{s2} .

The operating range of this kind of lock detection circuit is from 0 to NT_{ref} (with respect to the total delay of the VCDL), a range far larger than that of a conventional phase

Figure 7.11 Operating principle of the lock detection circuit (2)

comparator, and likely larger than the tuning range of the VCDL.

Another output of this circuit is V_{win} , which is used in the charge pump phase comparator shown in Figure 7.9. When neither V_{under} nor V_{over} is enabled, V_{win} is a negative window signal around each of the rising edges of P_0 to drive the charge pump phase comparator. Otherwise, V_{win} keeps high to disable the charge pump phase comparator and the voltage comparator in the charge pump phase comparator is controlled by V_{under} and V_{over} .

Combining the lock detection circuit and the phase comparator extends the locking range without a reduction of the locking resolution. The lock detection circuit acts to force the delay into the range from $T_{ref} \frac{N}{N+1}$ to $T_{ref} \frac{N}{N-1}$, and then passes the control to the charge pump phase comparator to finely tune the delay.

Since the tuning range of the VCDL is normally not as large as the operating range of the lock detection circuit (0 to NT_{ref}), Equation (7.7) can be simplified to Equation (7.9) when N is an odd number,

This allows simplification of the circuit. The operating range would become

$$S_1 = \sum_{n=0}^{\frac{(N-1)}{2}} P_{2n} \overline{P_{2n+2}} \quad (7.9)$$

smaller, but would remain large enough to cover the tuning range of the VCDL. The transistor-level simulation result, when the period of the reference is 5ns and N is equal to 9, is shown in Figure 7.12,

Figure 7.12 Simulation result of the lock detection circuit

where T_d is the delay of each delay stage. It can be seen that both V_{under} and V_{over} are high when the delay is in the correct range. If the delay is too small (large), V_{under} (V_{over}) is zero. When the loop is in lock, the outputs of the lock detector and the window generator are shown in Figure 7.13.

7.10 Detailed design of the novel charge pump phase comparator

7.10.1 Pulse generator

As shown in Figure 7.9, the new charge pump phase comparator consists of a pulse generator, two integrators, a voltage comparator, and a pulse control feedback circuit. At the rising edges of the two inputs, two pulses are generated, with the amplitudes deter-

Figure 7.13 Outputs of the lock detector and window generator

mined by a control voltage. The width and amplitude difference of the two pulses carry input phase alignment information. Figure 7.14 shows the schematic of the pulse genera-

Figure 7.14 Schematic of the pulse generator

tor. The inputs P_0 and P_N come from the VCDL and the amplitude tuning voltage V_{ctl} comes from the pulse control feedback circuit. When V_{win} is high, M_{17} is off and M_6 , M_9 are on, thus there is no current passing through M_{19} and both M_{13} and M_{15} are off. It is clear that P_u and P_d are always high (VDD) and there is no current in parts (B, C) as long as V_{win} is high. Under this condition, the pulse generator is in a “idle” state.

Once V_{win} goes low, which can only happen when both P_0 and P_N are low, both M_{17} and M_{18} are turned on and there is a bias current, controlled by V_{ctl} , flowing through diode-connected M_{19} . Two transistors M_{13} and M_{15} attempt to mirror the current, but M_{12} and M_{14} are off because P_0 and P_N are low. At this point, the pulse generator is in a “wait” state, where it remains until either P_0 or P_N goes to high.

If P_0 or P_N goes to high, transistor M_{12} or M_{14} will be turned on, thus P_u or P_d will be pulled down to produce a negative edge. The current through M_{13} or M_{15} is a mirrored current of M_{19} , which in turn is tuned by V_{ctl} , a change in V_{ctl} can change the slope of the falling edge of P_u or of P_d and the amplitude of the pulse. Once both P_0 and P_N are high, part (A) of the circuit sets P_u and P_d to high simultaneously. This is the “output” stage of the pulse generator.

After a certain period (about the delay of one delay stage), V_{win} returns high and the circuit returns to “idle” state again. A state diagram summarizing the whole process is illustrated in Figure 7.15. In one point of view, the pulse generator is similar to a conventional Dual D flip-flop phase detector, but it produces two narrow tunable pulses instead of two very wide pulses, and it has relative short paths from P_0 (or P_N) to P_u (or P_d). This enables it to maintain a relative high phase resolution and accuracy compared to the conventional phase detector.

Figure 7.15 Operating principle of the pulse generator

7.10.2 Pulse integrator

The two pulses (P_u and P_d) are integrated separately by two integrators (IntU and IntD) as shown in Figure 7.16. Integrators are also under the control of V_{win} . When V_{win} is

Figure 7.16 Schematic of the integrators

high, there is a constant current flowing through M_6 , so M_4 and M_5 are on, M_0 and M_1 are off, V_u and V_d are both zero. At this time, the integrators are in "idle" state, and the current flowing through M_6 is called idle current. Once V_{win} goes low, the idle current drops to

zero but V_u and V_d are still zero due to the capacitors C_u and C_d . At this time, the integrators are activated. The actual integration is done by switching on the transistor M_0 or M_1 with the negative pulse P_u or P_d , and the outputs (V_u and V_d) remain constant after two pulses end until the V_{win} returns high and the two integrators are reset to zero.

From the schematic, one can see that the integrators are not ideal and the output voltage is not always proportional to the area of the negative pulse.

First of all, since the amplitude of P_u or P_d must be lower than $(V_{DD}-V_{TH})$ to turn on M_0 or M_1 , the effective pulse amplitudes are smaller than the actual ones as shown in Figure 7.17-A. Also, if the P_u or P_d stays low for a period of Δt , V_u or V_d is not proportional to Δt , instead, the relation will be exponential as shown in Figure 7.17-B. Since the

Figure 7.17 Characteristics of the integrator

information of phase difference is carried in Δt , V_u and V_d should be in a range where the slope is relatively large as the “Desired operating range” shown in Figure 7.17-B, so that a high phase resolution can be obtained when the loop is in lock or about in lock. This is guaranteed by the pulse control feedback circuit.

7.10.3 Voltage comparator

As shown in Figure 7.9, the integrator output voltages (V_u and V_d), carrying the phase difference information of the two inputs of the phase comparator, are applied to a voltage comparator to produce an error signal, indexing the phase error. The schematic of the voltage comparator is shown in Figure 7.18. From the figure, if V_{over} or V_{under} is zero,

Figure 7.18 Schematic of the voltage comparator

M_0 or M_1 is turned on and drives V_d or V_u high, a situation in which the voltage comparator is under the control of the lock detection circuit. To guarantee the correct operation of the voltage comparator, voltages V_u and V_d should be in a range slightly higher than the threshold voltage (V_{th}) of the NMOS transistor. This range matches the desired operating range of the integrators shown in Figure 7.17-B.

7.10.4 Pulse control feedback circuit

To guarantee that the outputs of the pulse generator have amplitudes suitable to place their integral values (V_u and V_d) into the desired operating range, a pulse control feedback circuit shown in Figure 7.19 is employed in the charge pump phase comparator. When V_{win} is high, M_3 and M_4 are off, so its output voltage (V_{cll}) stays constant and there

Figure 7.19 Schematic of the pulse control feedback circuit

is no current flowing through this feedback circuit. If V_u and V_d are too high when V_{win} is low, the current flowing through M_1 is larger than the current flowing through M_9 . As a result, the capacitor (C) is charged and V_{ctl} will increase to reduce the amplitudes of two pulses P_u and P_d (please refer to Figure 7.14, “Schematic of the pulse generator,” on page 100). Similarly, if either V_u or V_d is too low, the capacitor is discharged and V_{ctl} is reduced to obtain larger amplitudes of P_u and P_d . After a certain time, one of V_u and V_d , or both are in the desired range and no more than one of them is larger than the desired range, whether the DLL is in lock or not. Once the loop is nearly locked, both voltages are confined in the range where the charge pump phase comparator displays its best resolution.

7.11 Characteristics of the charge pump phase comparator

To obtain the transfer characteristic of the charge pump phase comparator, a serial analysis is done for each parts of the circuit separately. Firstly, if there is no timing error between its two inputs, the areas of both pulses (P_u , P_d) are identical (seen as FCDE shown in Figure 7.20-A). If there is a timing error (τ), one pulse may look like ABDE in that diagram, and there is an effective error area GBCH shown as the shadow area in the

Figure 7.20 The relation between timing error and area error figure.

This area can be expressed as,

$$A(\tau) = h\tau \quad (7.10)$$

Because of the limit of the signal V_{win} , the sensitivity of $A(\tau)$ to (τ) is lost when the timing error (τ) is very large. This characteristic is shown in Figure 7.20-B.

As shown in Figure 7.17-B, the characteristic of the integrators are not an ideal one, so voltage difference between V_u and V_d are not always proportional to the effective error area ($A(\tau)$) of P_u and P_d . As a result, the transfer characteristic of the charge pump phase comparator will be nonlinear, and will have the appearance of Figure 7.21. ΔV in this diagram denotes the increment of the $V_{control}$, which appears across a capacitor connected to the output of the charge pump phase comparator, on detecting a certain value of the timing error. From the figure, it can be seen that the charge pump phase comparator has the greatest sensitivity and thus the highest resolution in the low-timing-error region. Even though the linear range is not as good as that of conventional charge pump phase

Figure 7.21 Characteristic of the charge pump phase comparator

detectors, this charge pump phase comparator is preferred for DLLs since it provides very low in-locking error and low control voltage glitches. This is confirmed by the simulation results in the following section.

7.12 Simulation of the charge pump phase comparator

To verify the feasibility of the new charge pump phase comparator, simulations were performed using the test bench shown in Figure 7.22. To simulate the charge pump phase comparator in the absence of the VCDL, lock detector and window generator, two frequency sources are employed in the test bench to provide the signals, *Clk* and *Clk_delay*, respectively, and another pulse source is used to supply the window signal (V_{win}). Since the output of the charge pump phase comparator is actually a current, a capacitor was attached at the output port and the increment of the output voltage is measured for different timing error between the two inputs of the phase comparator. To obtain the transfer characteristic of the charge pump phase comparator, the capacitor is discharged before the rising edges of the inputs come.

Figure 7.22 Test bench for charge pump phase comparator

Because the charge pump phase comparator utilizes a pulse control feedback circuit to guarantee that the output amplitudes of the pulse generator are in the correct range, there is a startup process during which the feedback is established when the charge pump phase comparator starts. To illustrate this process, two inputs (*Clk* and *Clk_delay*) are connected to two identical frequency sources and the output of the feedback circuit (V_{ctl}) is shown in Figure 7.23. One can see that the output of charge pump phase comparator does not correspond to the timing error between its two inputs until the signal V_{ctl} reaches a certain value.

Once the startup process is completed, the charge pump phase comparator is ready to determine the timing error between its two inputs. To illustrate the detailed operation of the charge pump phase comparator (please refer to “Basics of the novel charge pump phase comparator” on page 95), the simulated internal signals P_u , P_d , V_u , V_d , are shown in

Figure 7.23 Startup process of the charge pump phase comparator

Figure 7.24. $V_{control}$ keeps increasing cycle by cycle in the real working environment, but

Figure 7.24 Detailed operation of the charge pump phase comparator

in the figure, it goes down to 900mV periodically because of the reset circuit (which is

employed only for the test). We can see that the actual circuit operates exactly as expected.

To obtain the transfer characteristic of the charge pump phase comparator, two signals whose edge timing error is a function of the time are applied to the charge pump phase comparator after the startup process is completed. The simulation result is shown in Figure 7.25. The relation between the voltage increment (ΔV) and the timing error is plot-

Figure 7.25 Waveform of the output of the charge pump phase comparator

ted with Matlab as shown in Figure 7.26. Although the linearity is not perfect, ΔV is almost zero when the timing error is zero, which means the timing offset of the charge pump phase comparator is almost zero. This is the main advantage of this design because the voltage offset is very critical and should be minimized in most DLL applications.

7.13 Summary

In this chapter, the problems involved in the conventional phase detector and

Figure 7.26 Transfer characteristic of the charge pump phase comparator

charge pump are analyzed and a novel charge pump phase comparator is designed to overcome those problems. The charge pump phase comparator working with a lock detector and a window generator provides a very large dynamic range, which is larger than the tuning range of the VCDL. As a result, a DLL which uses this new phase comparison circuit can lock correctly no matter what the initial control voltage is. Transistor-level simulation results show that the lock detector and window generator work properly and the new charge pump phase comparator has a very small timing offset. Consequently, a DLL which uses this charge pump phase comparator should have a very low in-lock error.

Top-level simulation and layout consideration

8.1 Introduction

The designs of the DLL functional blocks were investigated in previous chapters. This chapter will describe their interconnections, and the influence of each on the performance of the DLL-based frequency synthesizer as a whole. An interconnection block diagram is shown in Figure 8.1

Figure 8.1 Top-level schematic of the frequency synthesizer

Simulations were done with the ultimate working environment - the electrical effects of fabrication, packaging, and PCB mounting were modelled using the pad model,

the package model and the PCB model.

8.2 Top-level simulation of the frequency synthesizer

There is a trade-off between the desired simulation results which should be as close to the real ones as possible and the simulation time in the top-level simulation which involves the entire circuit because a ‘real’ run would be time-consuming. So simulation time was reduced in ways that would not affect the validity of the outcome - for example, the capacitance of the loop filter was reduced to enable fast locking.

In order to verify the earlier predictions about the locking range of the loop, simulations with two extreme initial $V_{control}$ values 0 or V_{DD} were carried out. Furthermore, the in-lock error as well as the output jitter were measured to verify if the expected improvements had been realized. Finally, the operating range of the synthesizer is also estimated from specially designed simulations.

8.2.1 Locking process

The new frequency synthesizer was designed in such a way as to obviate the need for $V_{control}$ initialization. To confirm it, two simulations involving initial values of zero and V_{DD} volts were done. Successful locking in these two simulations demonstrates a correct DLL functionality over the whole possible range of $V_{control}$. The simulation results for a reference frequency of 200 MHz are shown in Figure 8.2.

When the $V_{control}$ is very large or very small at the beginning of the simulation, the delay is very small or very large, and the lock detector works to decrease or increase the control voltage so that the delay can be tuned toward to the desired value. Once the phase error is in a certain range (from $9T/10$ to $9T/8$), two outputs of the lock detector (V_{under}

Figure 8.2 Simulation results of the locking process

and V_{over}) are both high and the loop is under the control of the phase comparator. The $V_{control}$ is then finely tuned and finally reaches a constant value. The startup process of the charge pump phase comparator can be observed clearly in Figure 8.2-A. When the loop is in lock, the output waveform is shown in Figure 8.3.

Figure 8.3 Input and output of the frequency synthesizer

8.2.2 In-lock error and output jitter

The in-lock error and the output jitter are obtained from the simulation results of

the locking process. By measuring the half-swing crossing points for the rising edges of the P_O and P_N during the period from 600ns to 900ns, the in-lock error can be calculated and plotted with MatLab as shown in Figure 8.4. The RMS value of the in-lock error is

Figure 8.4 In-lock error calculated from the transistor-level simulation

calculated to be 0.119ps.

Similarly, the output periodic jitter can be determined from successive measurements of the output signal's period and comparing each value to the ideal one of 5ns/9. The results is shown in Figure 8.5, and the RMS value of the output timing jitter was cal-

Figure 8.5 The output periodic jitter

culated to be 0.766ps. It is clear that the periodic jitter is mainly caused by the mismatch

among the delay stages.

8.3 Simulation with pad and package models

The effects of fabrication, packaging and PCB layout on the synthesizer operation should be checked in order to make sure the correct operation in the non-laboratory environments. And these effects are mainly due to pad capacitances, bonding wire inductances and so on. A simulation with the pad and package models (please refer to Appendix A for the details of the models) are done and the simulation results show that this synthesizer can indeed tolerate the effects of the parasitics, although that the output swing becomes smaller than before.

8.4 Important layout issues

8.4.1 Matching issues of layout

A study of the charge pump phase comparator circuit diagram shows that high resolution and a low V_{offset} depend upon the symmetries of the circuit characteristics such as the two paths in the pulse generator, the two integrators, and the two paths in the pulse control feedback circuit. The layout of the device should be also made as symmetrical as possible.

In the VCDL, the stage-to-stage variations of the delay intervals (which should all be identical at any time) are very important, as the study of the operating principle of the device has revealed, the layout of all delay stages should be precisely the same. Similarly, the paths from the delay-stage outputs to the combiner should be 'alike' as closely as possible, and the paths should not cross or pass by other components such as resistors in order to reduce the effects of the coupling noise.

8.4.2 Minimization of interference among blocks

To minimize the interference among blocks including the VCDL, the charge pump phase comparator, the lock detector and the window generator, several guard rings are placed as shown in the layout diagram in Appendix B.

8.4.3 On-chip power supply filter capacitors

Even if the packaged chips is powered by a stable DC source, the inductances of power bonding wires combined with the varying current demands would cause the DLL circuit blocks to experience supply voltage fluctuations. Therefore capacitances working with these parasitic inductances to form low-pass filters were placed on the chip. These on-chip power supply filter capacitors were made as large as possible (approximately 20pF in this work), and they took up most of the unused die area (See C_{par} in Figure 8.6).

8.4.4 Pad arrangement to minimize coupling noise

Substantial capacitive coupling exists between pads because normally the pad area is very large ($80\mu\text{m} \times 80\mu\text{m}$ in this design). To reduce this coupling, each of the signal pads was placed between two grounded pads. Figure 8.6 shows the pad arrangement in this design. Please note that Figure 8.6 only shows half of the whole chip and the upper part is another design which is not included in this thesis.

8.5 Post-layout simulation

Inherently, there are parasitic capacitances, inductances and resistances in the actual CMOS layout, and the effects of these parasitics could not be fully taken into account in the transistor-level simulations. It is particularly important in the VCDL where

Figure 8.6 Pad arrangement of the chip

these parasitics would affect the performance of the circuit since the operation of the device depends on the parasitic capacitance. So simulations of the frequency synthesizer at the post-layout level were done, and the results show the operating frequency range of the circuit different from that obtained at the transistor level, and the relative error is about 10% because the delay of the VCDL is increased by about 10% when those parasitic components are considered. The effect of parasitic elements on lock detector and charge pump phase comparator functioning was observed to be negligibly small.

8.6 Summary

By combining all circuit blocks together and doing the transistor-level simulations, the functionality of the new DLL-based frequency synthesizer was verified.

A number of important issues involved in the layout of the frequency synthesizer were discussed and a post-layout simulation was done. In the next chapter, the actual measurement results and analysis will be given.

Test setup and performance evaluation

9.1 Introduction

The design of the DLL-based frequency synthesizer was fabricated in CMOS 0.18 μ m technology and packaged in a 24-leaded CFP (ceramic flat package). To evaluate the performance of the frequency synthesizer, a number of measurements including phase noise, spurious power level, operating frequency range, power consumption, have been done with the PCB board provided by CMC (Canadian Microelectronics Corporation).

9.2 Test setup

Since all the components in this frequency synthesizer were integrated in a single chip, the test setup is relatively straightforward and mainly includes an RF generator to provide a reference frequency, a DC power supply and an instrument to measure the output signal.

The input source is Rohde&Schwarz signal generator with the frequency range of 5K~6GHz, and the frequency can be easily adjusted. A DC offset of $V_{DD}/2$ has to be added to its output because the frequency synthesizer requires an input swing in the range of 0~1.8 volts (V_{DD}) - this was done with a DC offset adjustment circuit (ZFBT-6GW DC

level adjustment circuit). It was observed that the duty cycle was changed as the offset was adjusted, which was used to test the frequency synthesizer's sensitivity to duty cycle distortion.

An adjustable power supply (HP 3630A DC voltage source, output voltage range 0 to 6V) was used to provide power so that the effect of changing the supply voltage can be observed.

Since the spectral purity of the output signal was of greatest interest, the output phase noise and spurious power level were measured by applying the output to a spectrum analyzer (HP8564E spectrum analyzer). The block diagram of the test setup is shown in

Figure 9.1 Test setup for frequency purity measurement

Figure 9.1. The power consumption was obtained by measuring the consumption current with a multi-meter. The following paragraphs give the detailed measurement results.

9.3 Measurement results

9.3.1 Phase noise performance

The phase noise performance is one of the most important characteristics of a frequency synthesizer, and the potential for achieving a performance improvement over that

of the PLL was the main reason for investigating the DLL in this thesis. Since the centre frequency of this DLL-based frequency synthesizer output is highly stable, the phase noise measurement can be easily done with a spectrum analyzer instead of a specific phase noise measurement instrument.

With the input (reference) frequency of 173MHz, the power spectrum of the output signal (the reference frequency multiplied by nine) is shown in Figure 9.2. And phase

Figure 9.2 Power spectrum of the output signal

noise at different frequency offsets was measured as shown in Figure 9.3. From the figure, it can be observed at an offset frequency of 10kHz, a phase noise of -107.17dBc/Hz was obtained, and it decreases rapidly as the frequency offset increases. The output phase noise for different input frequencies was also measured and a variation below 3dB was observed. As was mentioned previously, the DLL-based frequency synthesizer has no ability to suppress the noise coming from its reference source, which means the use of a relatively unstable source such as an RF signal generator would bring about worse phase

Figure 9.3 Phase noise measurement result

noise than the use of a very stable source such as a crystal oscillator would do. But even with the input signal taken from an RF signal generator, the phase noise performance of the present design is superior to that of all the other devices reported so far.

9.3.2 Spurious power level measurement

DLLs do not compare well with PLLs as far as the spurious noise production is concerned. And there are some factors inherent in the DLL design which has to be considered to improve the performance (See “Spur performance of the DLL-Based frequency synthesizers” on page 51). With the new circuit ideas in this design, a reduction of spurious noise is expected.

The spurious power level was also measured using the spectrum analyzer. A spurious power level of -46.17dBc is obtained when the input frequency is 173MHz as shown in Figure 9.4. Although the spurious power level of the new synthesizer is not so good as those of PLL-based ones, it is far better than that of all the DLL-based ones reported to

Figure 9.4 Spurious power level measurement

date

9.3.3 Summary of the measurement results

The phase noise performance and spurious power level performance are given for the input frequency of 173MHz, a frequency that is about halfway along the operating range of 110MHz to 230MHz. Actual measurements show slight variations on the output phase noise (within about 3dB) over this frequency range.

The input duty cycle can be adjusted by changing the DC offset in the DC offset adjustment block shown in Figure 9.1 on page 120. Actual measurement shows that this design can work with the DC offset in the range about from 0.2V to 1.6V, which means that the design can tolerate large duty-cycle variation (approximately in the range from 15% to 85%).

The power dissipation was determined from power supply current measurement. This current is 3.7mA when the input frequency is 173MHz, which corresponds to a

6.7mW power dissipation. This power dissipation is not constant over the operating frequency range, instead, it increases with the increasing frequency as expected.

The measurements are summarized in Table 9.1.

TABLE 9.1: Characteristics of the frequency synthesizer

Supply voltage	1.8V
Reference frequency	173MHz
Output frequency	1.557GHz
Current consumption	3.7mA
Phase noise @10kHz offset	-107.17dBc/Hz
Spur power	-46.17dBc
Input signal duty-cycle range	15% to 85%
Input frequency range	110MHz to 230MHz
Output frequency range	0.99GHz to 2.07GHz
Active area	0.05mm ²

9.4 Summary

In this chapter, the test setup is described and the measurement results are given. The detailed package diagram, PCB diagram and connections are given in Appendix A. And the actual measurement results prove both phase noise and spur performance of this design to be better than these of all the other circuits reported to date [7][8][9][10].

10.1 Conclusion

In this thesis, a theoretical analysis, design and implementation of a DLL-based integer-N frequency synthesizer with low spur content has been presented. It starts with a review of the typical frequency synthesis techniques and the PLL and DLL-based frequency synthesizer techniques were analyzed in detail. Due to the potential superior performance of the DLL-based devices, analysis concentrates on this class of synthesizers, and the performance of the DLL-based circuits was analyzed and important issues which would arise in an attempt to implement a low noise, low spur frequency synthesizer were identified and discussed.

Although there are some reported designs of the DLL-based frequency synthesizers in the literature [7][8][9][10], they suffer from large in-lock error, large stage-to-stage error caused by the mismatch among the delay stages in the VCDL, and their performance regarding to both phase noise and spurious power level is not very good. The DLL-based frequency synthesizer presented in this thesis document employs a number of new circuit ideas which greatly contributes to the phase noise and spurious power level reduction, especially the novel charge pump and phase comparator. The favorable performance

results were observed on simulating this new charge pump phase comparator - its timing offset is below 0.02ps and it provides a very low in-lock error with an RMS value of about 0.1ps, which makes it possible to greatly reduce the spurious power level.

This frequency synthesizer is fabricated in 0.18 μm CMOS process with the active area of only 0.05mm², and a power consumption of 6.66 mW from a 1.8 volts supply at an input frequency of 173MHz. At the same input frequency, the measured phase noise is -107.17 dBc/Hz at 10kHz offset and the measured spurious power level is -46.17dB.

Both the simulation results and actual physical measurement results show that this design represents a significant improvement over all other reported designs to date, and verify the feasibility of the new circuit concepts.

10.2 Suggestions for future research

A continuance of the present work, directed at obtaining further improvement of DLL-based frequency synthesizer performance can use the following ideas,

- Increase of tuning range of the VCDL,
- Optimization of the VCDL driver circuit so that the effect of the noise coming from the transistors could be reduced. For example, the switched bias circuit can be employed to reset flicker noise periodically [17],
- Optimization of the voltage comparator in the charge pump phase comparator.

References

1. Robin R. -B. Sheen, Oscar T. -C. Chen, Chia-Yi, "A CMOS PLL-based frequency synthesizer for wireless communication systems at 0.9, 1.8, 1.9 and 2.4 GHz" Circuits and Systems, 2001. ISCAS 2001. The 2001 IEEE International Symposium on , Volume: 4 , 6-9 May 2001 Page(s): 722 -725 vol. 4
2. Haiyong Wang, Min Lin, Yongming Li, Hongyi Chen, "A novel architecture for integrated CMOS wideband PLL-based frequency synthesizer" Circuits and Systems, 2002. MWSCAS-2002. The 2002 45th Midwest Symposium on , Volume: 3 , 4-7 Aug. 2002 Page(s): III -509-11 vol.3
3. Konstantinos A. Efstathion, George Papadopoulos, Grigorios A. Kalivas, "High speed frequency synthesizer based on PLL" Electronics, Circuits, and Systems, 1996. ICECS '96., Proceedings of the Third IEEE International Conference on , Volume: 2 , 13-16 Oct. 1996 Page(s): 627 -630 vol.2
4. Sadeka Ali, Faquir Jain, "A low jitter 5.3-GHz 0.18- μ m CMOS PLL based frequency synthesizer" Radio Frequency Integrated Circuits (RFIC) Symposium, 2002 IEEE , 2-4 June 2002 Page(s): 173 -176
5. Mohd S. Sulaimain, "A 2.4-GHz, 3.-74-mW CMOS PLL-based frequency synthesizer" Semiconductor Electronics, 2002. Proceedings. ICSE 2002. IEEE International Conference on , Dec. 19 - 21, 2002 Page(s): 72 -76
6. Manassewitsch, Vadim, *Frequency synthesizers, theory and design* ISBN 0-471-01116-9 Page 2-7
7. David J. Foley and Michael P. Flynn, "A 3.3V, 1.6GHz, Low-Jitter, Self-Correcting

-
- DLL Based Clock Synthesizer in 0.5 μ m CMOS” Circuits and Systems, 2000. Proceedings. ISCAS 2000 Geneva. The 2000 IEEE International Symposium on , Volume: 2 , 28-31 May 2000 Page(s): 249 -252 vol.2
8. David J. Foley, Michael P. Flynn, “CMOS DLL-Based 2-V 3.2-ps Jitter 1-GHz Clock Synthesizer and Temperature-Compensated Tunable Oscillator” Custom Integrated Circuits Conference, 2000. CICC. Proceedings of the IEEE 2000 , 21-24 May 2000 Page(s): 371 -374
 9. Chua-Chin Wang, Hsien-Chih She, and Ron Hu, “A 1.2GHz Programmable DLL-Based Frequency Multiplier for Wireless Applications” Electronics, Circuits and Systems, 2002. 9th International Conference on , Volume: 1 , 15-18 Sept. 2002 Page(s): 69 -72 vol.1
 10. Qingjin Du “A DLL-based frequency synthesizer” Master thesis 2003, Carleton University
 11. Jingcheng Zhuang, Qingjing Du, Tad Kwasniewski, “Reduced in-lock error DLL-based clock synthesizer with a novel charge pump phase comparator”, Electronics Letters , Volume: 39 Issue: 1 , 9 Jan. 2003 Page (s): 48 -49
 12. Jingcheng Zhuang, Qingjing Du, Tad Kwasniewski, “A -107dBc, 10kHz Carrier Offset 2-GHz DLL-Based Frequency Synthesizer”, Proceeding of CICC 2003, 13-3.
 13. Dagnachew Birru, “A Novel Delay-Locked Loop Based CMOS Clock Multiplier” Consumer Electronics, IEEE Transactions on , Volume: 44 Issue: 4 , Nov. 1998 Page(s): 1319 -1322
 14. Mathieu Renaud, Yvon Savaria, “A Linear Phase Detector for Arbitrary Clock Signals” Circuits and Systems, 2002. ISCAS 2002. IEEE International Symposium on, Volume: 4 , 26-29 May 2002 Page(s): IV-775 -IV-778 vol.4
 15. Beomsup Kim, Todd C. Weigandt, Paul R. Gray, “PLL/DLL System Noise Analysis for Low Jitter Clock Synthesizer Design” Circuits and Systems, 1994. ISCAS '94., 1994 IEEE International Symposium on , Volume: 4 , 30 May-2 June 1994 Page(s): 31 -34 vol.4
 16. R.C.H. van de Beek, E.A.M. Klumperink, C.S. Vaucher and B. Nauta, “On Jitter Due to Delay Cell Mismatch In DLL-Based Clock Multipliers” Circuits and Systems, 2002. ISCAS 2002. IEEE International Symposium on , Volume: 2 , 26-29 May 2002

Page(s): II-396 -II-399 vol.2

17. Sheng Ye, Lars Jansson and Ian Galton, "Techniques for In-band Phase Noise Suppression in Re-circulating DLLs", Proceeding of CICC 2003, 13-2
18. J. Maneatis, "Low-Jitter and Process-Independent DLL and PLL Based on Self-Biased Techniques," ISSCC Digest of Technical papers, 1996.
19. M. Johnson and E. Hudson, "A variable delay line PLL for CPU-coprocessor synchronization," IEEE journal of Solid-State Circuits, vol. 23, pp. 1218-1223, Oct. 1988.
20. I. A. Young, J. K. Greason, and K. L. Wong, "A PLL Clock Generator with 5 to 110MHz of Lock Range for Microprocessors," IEEE Journal of Solid-State Circuits, vol. 27, pp. 1599-1607, Nov. 1992.

Appendix A Schematics of the frequency synthesizer

Appendix A.1 Schematic of the chip with package model

Appendix A.2 Schematic of the frequency synthesizer with pad models

Appendix A.3 Schematic of the frequency synthesizer with output buffer

Appendix A.4 Top-level schematic of the frequency synthesizer

Appendix A.5 Schematic of the lock detector and window generator

Non-deadzone Phase Detector

High-resolution Charge Pump

Appendix A.6 Schematic of the charge pump phase comparator

Appendix A.7 Schematic of the VCDL and edge combiner

Appendix B Layouts of the frequency synthesizer

Appendix B.1 Die diagram of the frequency synthesizer

Appendix B.2 Top-level layout of the frequency synthesizer

Appendix B.3 Layout of the frequency synthesizer

Appendix B.4 Layout of the charge pump phase comparator

Appendix B.5 Layout of the lock detector and window generator

Appendix C Package and PCB diagrams

Appendix C.1 Photograph of the die

Appendix C.2 Photograph of the PCB