Jitter in Oscillators with 1/f Noise Sources and Application to True RNG for Cryptography

by

Chengxin Liu

A Dissertation
Submitted to the Faculty
of the

WORCESTER POLYTECHNIC INSTITUTE

in partial fulfillment of the requirements for the
Degree of Doctor of Philosophy
in
Electrical Engineering

January 2006

Approved:

Professor John A. McNeill
ECE Department
Thesis Advisor

Professor Berk Sunar
ECE Department
Thesis Committee

Professor Donald R. Brown
ECE Department
Thesis Committee

Professor William J. Martin
Math. Department
Thesis Committee

To my wife

Abstract

In the design of voltage-controlled oscillators (VCOs) for communication systems, timing jitter is of major concern since it is the largest contributor to the bit-error rate. The latest deep submicron processes provide the possibility of higher oscillator speed at the cost of increased device noise and a higher 1/f noise corner. Therefore it is crucial to characterize the upconverted 1/f noise for practical applications.

This dissertation presents a simple model to relate the time domain jitter and frequency domain phase noise in the presence of non-negligible 1/f noise sources. It will simplify the design, simulation, and testing of the PLL, since with this technique only the open loop VCO needs to be considered. Design methodologies for white noise dominated ring oscillators and PLLs are also developed by analyzing the upconverted thermal noise in time domain using a LTI model. The trade-off and relationship between jitter, speed, power dissipation and VCO geometry are evaluated for different applications. This model is supported by the measured data from 24 ring oscillators with different geometry fabricated in TSMC 0.18μm process.

The theory developed in this dissertation is applied to the design of PLL- and DLL- based true random number generators (TRNG) for application in the area of "smart cards". New architectures of dual-oscillator sampling and delay-line sampling are proposed for random number generation, which has the advantage of lower power dissipation and lower cost over traditional approaches. Both structures are implemented in test chips fabricated in AMI 1.5μm process. The PLL-based TRNG passed the NIST SP800-22 statistical test suite and the DLL-based TRNG passed both the NIST SP800-22 statistical test suite and the Diehard battery of tests.

Acknowledgements

I would like to acknowledge the following people for helping to make this thesis possible:

My advisor, Professor John McNeill, for his continuous guidance, support, and encouragement;

Professors Berk Sunar, Donald Brown, and William Martin for serving on my thesis committee and insightful comments;

Bob Brown for his help to keeping Cadence running;

Carlos, Ping, Tony, and Chris for their fellowship and all the good memories of the Analog Lab and ISSCC;

The staff in the ECE Department for their help throughout my stay at WPI;

Most importantly, I thank my family – especially my wife, Ying - for their love, patience, and unconditional support.

TABLE OF CONTENTS

Abstract iv

Acknowledgments v

Table of Contents vi

List of Tables x

List of Figures xi

1 Introduction 1

1.1 Motivation 2

1.2 Organizations 3

2 Jitter and Phase Noise Concepts 4

2.1 Definition 4

2.2 Jitter Classification 6

2.3 Current and Noise Models for MOSFETs 7

2.3.1 Current Model 7

2.3.2 Thermal Noise 12

2.3.3 1/f Noise 15

3 Technique to Relate Freq. and Time Domain Oscillator Jitter Performance 18

3.1 Introduction 18

3.2 VCO Jitter and Phase Noise Performance Characterization 22

3.2.1 Frequency Domain Using Spectrum Analyzers 22

3.2.2 Time Domain: Equivalent Time Oscilloscope Method 24

3.2.3 Time Domain: Real Time Oscilloscope Method 27

3.3 Theoretical Development 30

3.3.1 Near Carrier Oscillator Spectrum 30

3.3.2 Relationship between Jitter and Phase Noise 32
3.3.3 Jitter due to White Noise 33
3.3.4 Jitter due to 1/f Noise 34
3.3.5 1/f Transition Time t_c 37
3.4 Experimental Verification 38

4 Technique to Relate Freq. and Time Domain PLL Jitter Performance 40
4.1 Loop and Noise Transfer Function 40
4.2 Frequency Domain Using Spectrum Analyzers 43
4.3 Time Domain, PLL Clock Referenced 45
4.3.1 Case I: White Noise Dominated... 46
4.3.2 Case II: In the Presence of Non-negligible 1/f Noise......................... 47
4.4 Time Domain, PLL Self Referenced 49
4.4.1 Case I: White Noise Dominated... 50
4.4.2 Case II: In the Presence of Non-negligible 1/f Noise......................... 51

5 Jitter and the Geometry of Ring Oscillators 53
5.1 Introduction.. 53
5.2 VCO Design in Time Domain with κ 55
5.3 κ of the CMOS Inverter... 57
5.3.1 Propagation Delay of the CMOS Inverter 58
5.3.2 Drain Thermal Noise in the Switching MOSFET 61
5.3.3 KTC Noise due to the Loading Transistor.. 64
5.3.4 κ_{total} of the CMOS Inverter 66
5.4 Jitter and VCO Geometry.. 67
5.4.1 κ and VCO Geometry.. 67
5.4.2 Normalized RMS Jitter and VCO Geometry...................................... 79
5.4.3 Normalized Cycle-to-Cycle Jitter and VCO Geometry 83
5.4.4 Summary... 86

5.5 Jitter and VCO Tuning 87
5.5.1 Tuning Transistors in Saturation 88
5.5.2 Tuning Transistors in Triode 91
5.6 Experimental Results 92
5.6.1 Test Chip Design 92
5.6.2 Inverter Delay and Geometry 95
5.6.3 κ and the VCO Geometry 98
5.6.4 Normalized RMS Jitter and VCO Geometry 104
5.6.5 Normalized Cycle-to-cycle Jitter and VCO Geometry 107
5.6.6 κ and VCO Tuning 110
5.7 Summary: Design of Low Jitter Ring Oscillators 112
5.7.1 Design for κ 112
5.7.2 Design for Normalized RMS Jitter 114
5.7.3 Design for Normalized Cycle-to-Cycle Jitter 115

6 Design of Digital PLL-based True Random Number Generator 116
6.1 Introduction 116
6.2 Review of TRNG Design Techniques 118
6.3 System Architecture 121
6.4 Time Domain Analysis and System Simulation 124
6.5 DAC-controlled Ring Oscillator 131
6.5.1 Sizing Transistors 132
6.5.2 DAC Control 134
6.6 Low Metastability D Flip-flop 135
6.7 Up/Down Counters 141
6.8 Digital Post-processing 143
6.9 Experimental Results 146
6.9.1 Test Chip Design 146

6.9.2 Measurement Results 148

7 Design of DLL-based True Random Number Generator 153

7.1 Delay-locked Loops 153

7.2 System Architecture 155

7.3 Voltage-controlled Delay Line 160

7.4 Improved Low Metastability D Flip-flop 162

7.5 Experimental Results 165

7.5.1 Test Chip Design 165

7.5.2 Measurement Results 167

8 Conclusion 178

8.1 Future Work 180

Appendix A. Kappa Plot Extraction from TIE Data 181

Appendix B. RNG Simulation by Matlab 182

Appendix C. Diehard Test Results for DLL-based RNG with the Classic Von Neumann Corrector 186

Appendix D. Diehard Test Results for DLL-based RNG with the Modified Von Neumann Corrector 204

References 212

LIST OF TABLES

Table 3.1 Measured and predicted ζ for implemented ring oscillators 39

Table 5.1 Predicted and simulated κ_{total} vs. L for 7-stage ring oscillators 76

Table 5.2 Relationship between jitter, VCO geometry, and ring configuration..... 86

Table 5.3 Relationship between jitter, power dissipation and die area 86

Table 5.4 Geometry range of the implemented oscillators..................................... 93

Table 5.5 Ring configuration of the implemented oscillators 93

Table 5.6 Measured min. inverter delay for implemented oscillators..................... 96

Table 5.7 Measured κ versus VCO geometry 99

Table 5.8 Measured κ-to-T_d ratio and VCO geometry 105

Table 5.9 The extracted κ-to-sqrt(T_d) ratio and VCO geometry 108

Table 6.1 NIST SP800-22 statistical test results for the PLL-based RNG (Post-processed by the classic von Neumann corrector)....................... 150

Table 6.2 NIST SP800-22 statistical test results for the PLL-based RNG (Post-processed by the modified von Neumann corrector).................. 151

Table 6.3 Performance summary for the PLL-based RNG 152

Table 7.1 NIST SP800-22 statistical test results for the DLL-based RNG (Post-processed by the classic von Neumann corrector)....................... 175

Table 7.2 NIST SP800-22 statistical test results for the DLL-based RNG (Post-processed by the modified von Neumann corrector).................. 176

Table 6.3 Summary and comparison of the PLL- and DLL-based RNG 177

LIST OF FIGURES

Figure 2.1 Clock jitter in time and frequency domains 4

Figure 2.2 Evaluation of the current model in equation (2.9) 10

Figure 2.3 Simulated drain current versus L... 11

Figure 2.4 Channel cross section of a MOSFET in the saturation region 13

Figure 2.5 Higher 1/f noise corner for MOSFETs with shorter channel length 17

Figure 3.1 Typical oscillator phase noise spectrum... 19

Figure 3.2 Phase noise p.s.d. of a 155MHz VCO in a 5GHz f_T bipolar process ... 20

Figure 3.3 Typical phase noise p.s.d. of VCOs in a 0.18μm process.................... 20

Figure 3.4 Free-running oscillators measurement in frequency domain 22

Figure 3.5 Frequency domain measurement result: VCO open loop.................... 22

Figure 3.6 Jitter measurement over time delay ΔT... 24

Figure 3.7 Measurement result: Time domain, open loop.................................... 25

Figure 3.8 Time interval error 27

Figure 3.9 TIE data post-processing 28

Figure 3.10 TIE and post-processed kappa plot for a 150MHz VCO..................... 29

Figure 3.11 Phase noise p.s.d. with close-in corner frequencies............................. 30

Figure 3.12 Evaluation of α... 37

Figure 3.13 The current-starved inverter 38

Figure 4.1 Basic block diagram of the PLL... 40

Figure 4.2 Bode plots of the PLL loop transfer functions.................................... 42

Figure 4.3 Phase noise shaping by the PLL... 42

Figure 4.4 Typical PLL phase noise spectrum 43

Figure 4.5 Measurement technique: Time domain, PLL clock referenced............ 45

Figure 4.6 Measurement result: Time domain, PLL clock referenced 45

Figure 4.7 Phase noise shaping by the PLL with different loop bandwidth f_L...... 48

Figure 4.8 Measurement technique: Time domain, self referenced 49
Figure 4.9 Measurement result: Time domain, self referenced 49
Figure 4.10 Numerical evaluation of the integral in (4.27) 52
Figure 5.1 A three-stage ring oscillator and resulting waveform 55
Figure 5.2 The CMOS inverter 57
Figure 5.3 Modeling of propagation delay for the CMOS inverter 58
Figure 5.4 Drain thermal noise of the switching MOSFET 61
Figure 5.5 Ideal output and actual output due to the drain thermal noise 62
Figure 5.6 KTC noise due to the loading transistor 64
Figure 5.7 The MOSFET layout under SCMOS design rules 71
Figure 5.8 Threshold voltage versus channel length for MOSFETs 74
Figure 5.9 Inverter configuration for 7-stage oscillators 76
Figure 5.10 Plot of predicted and simulated κ for 7-stage ring oscillators 77
Figure 5.11 Simulated input referred noise versus channel length 78
Figure 5.12 Simulated roll-up of the threshold voltage 78
Figure 5.13 The normalized rms jitter 79
Figure 5.14 The current-starved inverter 87
Figure 5.15 Die micrograph 94
Figure 5.16 Micrograph of a 7-stage ring oscillator on chip 94
Figure 5.17 Measured oscillator output waveforms 95
Figure 5.18 Measured minimum inverter delay versus VCO geometry 97
Figure 5.19 Measured duty cycle for the ring oscillator with M_1 size of
600μm/0.18μm 98
Figure 5.20 Measured relationship between κ and channel width 99
Figure 5.21 Measured relationship between κ and channel length 100
Figure 5.22 Simulated V_T roll-up for TSMC 0.18μm process 101
Figure 5.23 Measured γ_s for TSMC 0.18μm process in [95] 102
Figure 5.24 Predicted and measured κ for the VCOs with M_1 width of 20μm 103

Figure 5.25 Measured κ-to-T_d ratio versus W 106
Figure 5.26 Measured κ-to-T_d ratio versus L 106
Figure 5.27 Measured κ-to-sqrt(T_d) ratio versus W 108
Figure 5.28 Measured κ-to-sqrt(T_d) ratio versus L 109
Figure 5.29 Measured tuning characteristic for the ring oscillator
with M_1 size of 600μm/0.18μm 110
Figure 5.30 Measured relationship between κ and inverter during tuning 111
Figure 5.31 Contour plot for measured κ and minimum inverter delay 112
Figure 6.1 RNG implemented by direct amplification 118
Figure 6.2 RNG implemented by oscillator-sampling 119
Figure 6.3 Architecture of the digital PLL-based RNG 121
Figure 6.4 Autocorrelation and p.s.d. 122
Figure 6.5 Definition of random processes for clock with jitter 124
Figure 6.6 Data sampling by the D flip-flop 126
Figure 6.7 Sampling a Gaussian random variable 126
Figure 6.8 System behavior simulated by Matlab 128
Figure 6.9 RNG output simulated by Matlab 129
Figure 6.10 RNG output after dividing by 20 simulated by Matlab 130
Figure 6.11 DAC-controlled ring oscillator 131
Figure 6.12 Capacitor array with dummy transistors 134
Figure 6.13 Rising edge triggered master-slave D flip-flop 135
Figure 6.14 D flip-flop sampling of closely positioned edges 136
Figure 6.15 Simulation results for the master-slave D flip-flop in Figure 6.13 136
Figure 6.16 The low metastability D flip-flop 137
Figure 6.17 The pre-amplifier in the designed D flip-flop 138
Figure 6.18 The D latch in the designed D flip-flop 138
Figure 6.19 Simulation results for the designed D flip-flop 140
Figure 6.20 Analog loop filter 141

Figure 6.21 Bandwidth versus configuration of the counter p 141
Figure 6.22 System simulation without the stabilizing zero 142
Figure 6.23 The digital post-processing circuits 143
Figure 6.24 The classic von Neumann corrector 143
Figure 6.25 The modified von Neumann corrector 144
Figure 6.26 Chip micrograph of the PLL-based RNG 147
Figure 6.27 Jitter performance of the free-running ring oscillator at 30MHz 148
Figure 6.28 Spectrum and autocorrelation coefficient of post-processed data 149
Figure 7.1 Basic block diagram of the DLL 153
Figure 7.2 Architecture of the DLL-based RNG 155
Figure 7.3 Block diagram of the DLL-based RNG 156
Figure 7.4 Loop filter 157
Figure 7.5 Voltage-controlled delay line 160
Figure 7.6 Improved low metastability D flip-flop 162
Figure 7.7 Simulation results for the improved D flip-flop 164
Figure 7.8 Chip micrograph for the DLL-based RNG 166
Figure 7.9 Measured tuning characteristic of the implemented VCDL 167
Figure 7.10 Measured jitter versus tuning for the implemented VCDL 168
Figure 7.11 Measured κ versus tuning for the implemented VCDL 168
Figure 7.12 Rising edges of CLK locked to falling edges of DATA 169
Figure 7.13 Spectrum and autocorrelation of the raw data at the RNG output 170
Figure 7.14 Spectrum and autocorrelation of the post-processed data 171
Figure 7.15 Experiment to justify the noise source 172
Figure 7.16 Measured CDF and PDF of the jitter sampling process from the Experiment in Figure 7.15 173

Chapter 1: Introduction

1.1 Motivation

Voltage-controlled oscillators (VCOs) are widely used in modern communication systems and play a critical role in applications such as clock generation and recovery [1]-[3]. VCOs are also applicable in the area of cryptography; the method of oscillator-sampling is the most popular approach by far to generate high quality random numbers in system-on-chip designs for data encryption [4]-[6], due to the advantages of less die area, improved power efficiency, and high speed.

In most applications, the dominant sources of jitter in oscillators are 1/f and white noise. Previous research in this area was concentrated on white noise and the theory was well developed [7]-[11]. Since the $1/f^3$ phase noise corner was usually inside the loop bandwidth of the phase-locked loops (PLL), the 1/f noise contribution was "tracked out" by the PLL so that it was negligible.

The latest deep submicron processes provide the possibility of higher VCO speed. But they also introduce the problem of a higher $1/f^3$ phase noise corner. In previous work [7] and [8], the VCO design process was simplified by assuming the $1/f^3$ phase noise corner to lie below the PLL loop bandwidth. In a deep submicron process this assumption is no longer valid.

Several analytical models for the phase noise upconverted from 1/f device noise have been proposed recently [12]-[14]. The data from ring oscillators in [15] suggest that the jitter introduced by the 1/f noise over time delay ΔT is of the form $\zeta\Delta T$, but no analytical model was provided for calculating ζ.

This dissertation will extend the work in [8] to account for the 1/f noise contribution, discuss the relationship between jitter and the geometry of ring oscillators, and apply the developed theory to the design of true random number generators for cryptography applications.

The key contributions of this work are:

1. A technique to relate frequency and time domain oscillator and PLL performance in the presence of non-negligible 1/f noise sources. This technique will give designers the flexibility of working in whatever domain is easiest while still being able to accurately predict performance when measured by the end user. It also simplifies the design, simulation, and testing of the PLL, since with this technique only the open loop VCO needs to be considered.

2. Design methodologies for white noise dominated ring oscillators and PLLs. Thermal noise upconversion in ring oscillators is analyzed in time domain using a LTI model. This developed theory is supported by the measured relationship of jitter and VCO geometry from a test chip fabricated in TSMC 0.18µm process. As feature sizes scale down, wider PLL loop bandwidth is necessary to minimize the higher 1/f noise corner effect. This work also shows that jitter caused by white noise sources can be reduced by increasing the VCO's channel width and carefully choosing channel length and number of stages.

3. The design and implementation of PLL- and DLL- based true random number generators based on the theory developed in this dissertation for application in the area of "smart cards". New architectures of dual-oscillator sampling and delay-line sampling are proposed for random number generation. The main advantage over the traditional approach is the capability of achieving the same data rate using slower clocks, thus enabling cheaper process, lower power, and lower cost.

1.2 Organization

This dissertation is divided into eight chapters. Chapter 2 introduces the basic concepts of jitter and phase noise. Chapter 3 presents the developed technique to relate frequency and time domain oscillator jitter performance. VCO jitter and phase noise measurement techniques are first reviewed in Section 3.2. Detailed theory derivations are covered in Section 3.3.

In Chapter 4, the relationship between jitter and phase noise are analyzed for closed-loop PLLs. It starts with reviews of PLL loop and jitter transfer functions and frequency domain measurement techniques. Time domain measurement techniques are then discussed with theory development for each case. Two important results in this chapter are the close-loop jitter prediction in Section 4.3.2 and the upper bound for the total self-referenced jitter in Section 4.4.2.

Chapter 5 proposes ring oscillator and PLL design methodologies for different applications based on the developed relationship between jitter, speed, power dissipation and VCO geometry. The jitter upconverted by thermal noise is modeled in time domain for CMOS inverters in Section 5.3. This developed model is discussed in Section 5.4 and 5.5 for different applications and supported by experimental results in Section 5.6. Proposed design methodologies are then summarized in Section 5.7.

The model developed in Chapter 5 is applied to the design of PLL- and DLL-based true random number generators (RNGs) in Chapter 6 and Chapter 7. There are detailed time domain analysis, Matlab simulation, and circuit description for the PLL-based RNG in Chapter 6, while only a quick analysis is presented for the DLL-based RNG in Chapter 7 since these two architectures are similar.

Chapter 8 summarizes the contributions of the thesis and pointing out possible areas for future work.

Chapter 2: Jitter and Phase Noise Concepts

2.1 Definition

Phase noise and jitter are different ways of quantifying the same phenomenon. Jitter is the time domain manifestation of the noise sources in oscillators, and is defined as the short-term variations of a digital signal's significant instants from their ideal positions in time [16].

As illustrated in Figure 2.1, for an ideal noiseless sinusoidal oscillator, the zero crossing times of the oscillation waveform are evenly spaced at intervals of half of the period T_0. In frequency domain, the entire power of the oscillator will be located at the fundamental frequency, $f_0=1/T_0$, and the spectrum is an ideal impulse.

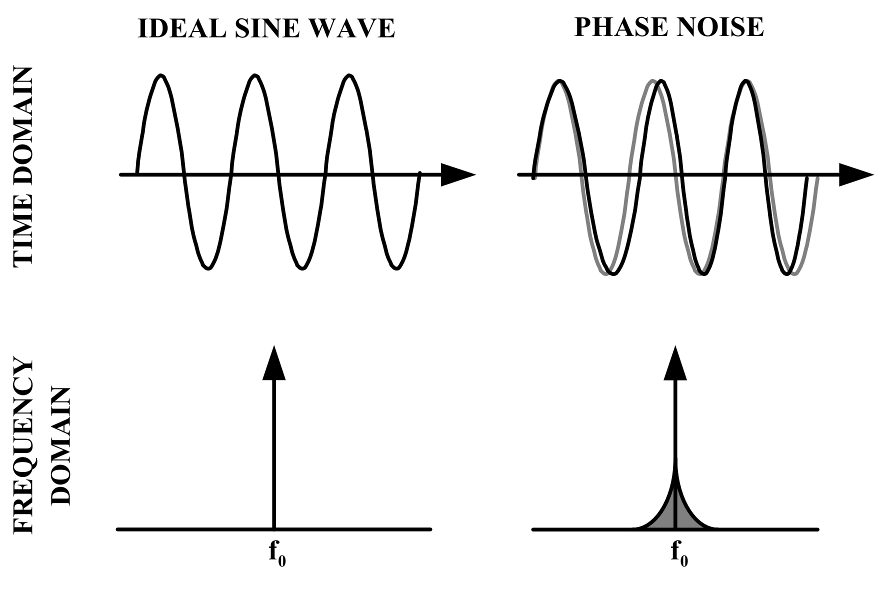


Figure 2.1 Clock jitter in time and frequency domains.

However, the noise adds uncertainty to the zero crossing times and introduces jitter. This results in variations in oscillating frequency from the ideal constant.

Since phase is the integral of frequency, the output phase will not increase uniformly, but executes a "random walk" around the ideal phase. This rapid, short-term, random fluctuation in the phase caused by time domain instability is defined as phase noise. In the frequency domain, the phase noise appears as "close in" sidebands around the carrier frequency as shown in Figure 1.1. Phase noise is usually specified in dBc/Hz at a given offset frequency, where dBc is the level in dB relative to the carrier.

By convention, timing variations are split into two categories, jitter and wander [17]. Wander is timing variations that occur slowly. According to ITU specifications [18], the threshold between wander and jitter is defined to be 10Hz.

2.2 Jitter Classification

There are two main types of jitter: deterministic jitter (DJ) and random jitter (RJ). DJ and RJ are also referred to as systematic and non systematic jitter respectively [19].

DJ is timing jitter that is repeatable and predictable. It is always bounded in amplitude and the bounds can usually be observed or predicted with high confidence. DJ comprises data dependent jitter (DDJ) and jitter which is bounded and uncorrelated to the data (BUJ) [19]. DDJ is the jitter that is added when the transmission pattern is changed from a clock like to a non-clock like pattern. The main sources of BUJ are duty cycle distortion, crosstalk, EMI radiation, and noise from power supply and substrate.

Random jitter (RJ) is the timing jitter due to random fluctuations and noise sources, and cannot be predicted. The main sources of RJ are 1/f noise (flicker noise) and white noise (shot noise and thermal noise). The jitter due to white noise is Gaussian random processes since the sum of a large number of statistically independent events will approach a Gaussian distribution by the central limit theorem.

This work will concentrate on RJ since RJ is not bounded in amplitude, and there is no way to eliminate RJ from a system since it is caused by fundamental noise.

2.3 Current and Noise Models for MOSFETs

2.3.1 Current Model

For long-channel MOS transistors in the saturation region, the drain current is I_D given by the square-law model [20]

$$I_D = \frac{1}{2}\mu_{eff} C_{ox} \frac{W}{L} V_{eff}^2 \tag{2.1}$$

where μ_{eff} is the effective mobility, C_{ox} is the gate-oxide capacitance per unit area, W and L are the channel width and length, and V_{eff} is the transistor gate over-drive voltage defined by

$$V_{eff} = V_{GS} - V_T \tag{2.2}$$

where V_{GS} is the gate-to-source voltage and V_T is the threshold voltage.

The transconductance g_m is

$$g_m = \frac{\partial I_D}{\partial V_{eff}} = \mu_{eff} C_{ox} \frac{W}{L} V_{eff} \tag{2.3}$$

Mobility depends on bias conditions and many process parameters such as gate oxide thickness, substrate doping concentration, threshold voltage, gate and substrate voltages, etc [21]. Reference [22] proposed an empirical unified mobility model as

$$\mu_{eff} = \frac{\mu_0}{1 + \left(\dfrac{E_{eff}}{E_0}\right)^n} \tag{2.4}$$

where μ_0 is the zero field mobility, E_0 and n are empirical constants, and E_{eff} is the average electrical field experienced by the carriers in the inversion layer which can be approximated by [21]

$$E_{eff} \approx \frac{V_{GS} + V_T}{6T_{ox}} \tag{2.5}$$

where T_{ox} is the gate oxide thickness.

By taking the Taylor approximation of (2.4), the effective mobility is modeled in the BSIM3 model as [21]

$$\mu_{eff} = \frac{\mu_0}{1 + U_a\left(\frac{V_{GS} + V_T}{T_{ox}}\right) + U_b\left(\frac{V_{GS} + V_T}{T_{ox}}\right)^2} \tag{2.6}$$

where U_a and U_b are first and second order mobility degradation coefficients, which can be determined by fitting (2.6) to the unified formulation of (2.4).

For short-channel MOS transistors in the saturation region, the drain current will deviate from the value predicted by the square-law model of (2.1) due to the short channel effects [20]. When the electric field under the gate of MOSFETs reaches a critical value E_c, the velocity of the carriers will saturate at a value v_{sat}. The μ_{eff} is no longer a constant and is a function of the transverse field in the inversion layer. The empirical expression for the velocity of the carriers is [22]

$$v_d = \begin{cases} \dfrac{\mu_{eff} E}{1 + \dfrac{E}{E_c}} & for\ E \le E_c \\ v_{sat} & for\ E \ge E_c \end{cases} \tag{2.7}$$

where the critical electric field E_c is

$$E_c = \frac{2v_{sat}}{\mu_{eff}} \tag{2.8}$$

By taking into account the velocity saturation, the drain current I_D can be modeled by [23]

$$I_D = WC_{ox}v_{sat}\frac{V_{eff}^2}{V_{eff} + E_c L_{eff}} \tag{2.9}$$

where L_{eff} is the effective channel length due to the short channel modulation.

Equation (2.9) will approach the classic square-law of (2.1) for long-channel MOS transistors since the V_{eff} in the denominator of (2.9) is negligible when $L \gg V_{eff}/E_c$. Therefore the current model of (2.9) is applicable to both long- and short-channel MOSFETs operating in the saturation region.

By defining the critical channel length L_c as

$$L_c = \frac{V_{eff}}{E_c} \tag{2.10}$$

the drain current can also be modeled by

$$I_D = \frac{1}{2}\mu_{eff}C_{ox}\frac{W}{L_{eff} + L_c}V_{eff}^2 \tag{2.11}$$

When $L \ll L_c$, (2.9) and (2.11) will approach

$$I_D = WC_{ox}v_{sat}V_{eff} \quad \text{when } L \ll L_c \tag{2.12}$$

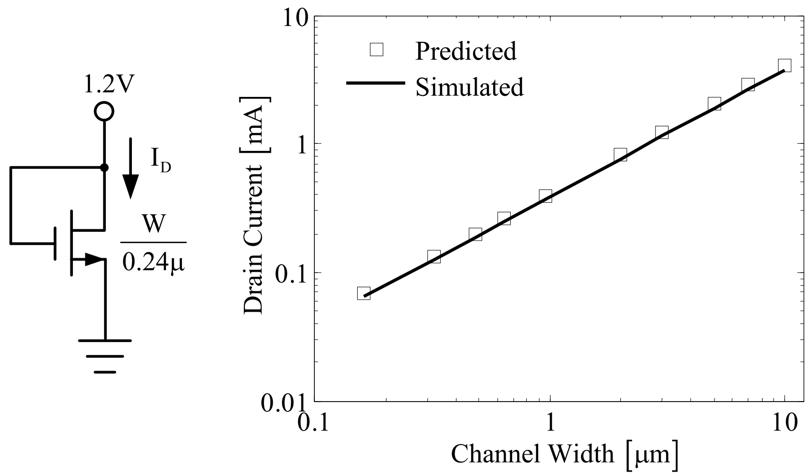
and the transconductance g_m is [22]

$$g_m = WC_{ox}v_{sat} = \frac{I_D}{V_{eff}} \quad \text{when } L \ll L_c \tag{2.13}$$

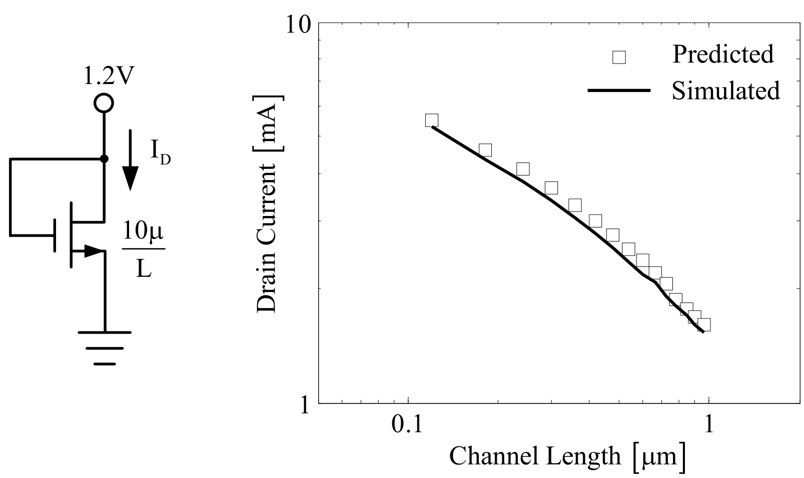
Therefore for a MOSFET operating in the saturation region with extremely short channel, the drain current increases linearly with V_{eff} and the only way to get a larger transconductance is to increase the channel width since decreasing channel length will not help any more.

Figure 2.2 shows the comparison of the drain current I_D calculated by (2.9) and simulated by Spectre using the BSIM3 model [21] for a NMOS transistor in

IBM 0.13μm process which is biased in the saturation region. The agreement between the predicted and simulated results is within 10%.



(a) Drain current versus the channel width.



(b) Drain current versus the channel length.

Figure 2.2 Evaluation of the current model in equation (2.9).

The relationships between the drain current I_D and the transistor geometry observed from Figure 2.2 are as follows:

1. I_D is proportional to W for both long- and short-channel MOSFETs;

2. The current model of (2.9) predicts that I_D is inversely proportional to L for long-channel MOSFETs and has no dependency on L for short-channel MOSFETs. From Figure 2.3 which shows only the simulated I_D in Figure 2.2b on a log-log scale, I_D is observed with a slope of -1 for L around 1μm. When L gets shorter, I_D shows less dependency on L. For L in the range of [0.18μm, 0.3μm], the observed slope of I_D has changed from -1 to -0.5, indicating I_D is inversely proportional to square root of L in this region.

3. From Figure 2.2b, the drain current is not entirely saturated even at the minimum channel length of 0.12μm.

For MOSFETs in the triode region, it is straightforward that the drain current is given by

$$I_D = \mu C_{ox} \frac{W}{L_{eff} + L_c} \left(V_{GS} - V_T - \frac{1}{2} V_{DS} \right) V_{DS} \tag{2.14}$$

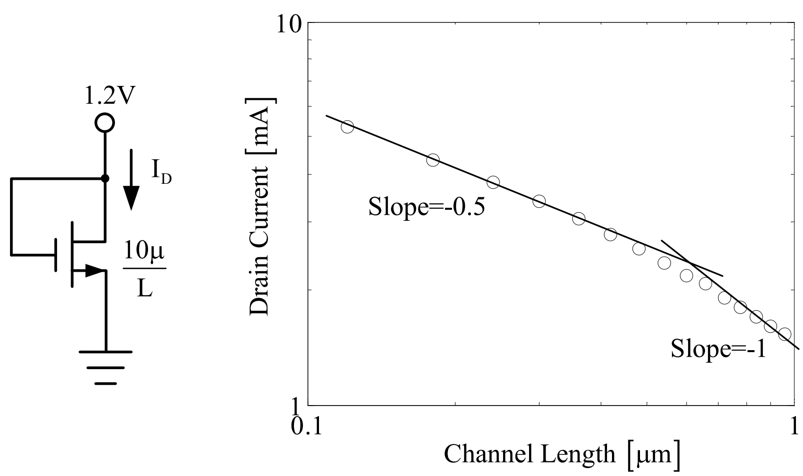


Figure 2.3 Simulated drain current versus L.

2.3.2 Thermal Noise

Thermal noise is generated by the effective channel resistance R_{ch} and is modeled by [24]

$$\frac{\overline{i_n^2}}{\Delta f} = \frac{4kT}{R_{ch}} \quad \left[\frac{\mathrm{A}^2}{\mathrm{H_z}}\right] \tag{2.15}$$

where k is the Boltzmann's constant and T is the temperature in Kelvin.

For long channel CMOS transistors, the thermal noise current spectral density is given by [20]

$$\frac{\overline{i_n^2}}{\Delta f} = \frac{4kT}{L^2}\mu_{eff}Q_{inv} \tag{2.16}$$

with

$$Q_{inv} = C_{ox}WLV_{eff}\frac{1-\eta+\dfrac{\eta^2}{3}}{1-\dfrac{\eta}{2}} = C_{ox}WLV_{eff}\gamma \tag{2.17}$$

$$\eta = \frac{A_{bulk}\left(\psi_{SL}-\psi_{S0}\right)}{V_{eff}} \tag{2.18}$$

where A_{bulk} is the correction factor for the linearized bulk charge, ψ_{S0} is the surface potential at source and ψ_{SL} is the surface potential at the drain.

By defining the zero-bias drain source conductance g_{d0} as

$$g_{d0} = \mu_{eff}C_{ox}\frac{W}{L}V_{eff} \tag{2.19}$$

(2.16) can be written as

$$\frac{\overline{i_n^2}}{\Delta f} = 4kT\gamma g_{d0} \quad \left[\frac{\mathrm{A}^2}{\mathrm{H_z}}\right] \tag{2.20}$$

with γ=2/3 for MOSFETs in the saturation region and γ varies from 2/3 to 1 as the drain-to source voltage V_{DS} varies from zero to the onset of the saturation [20].

Substituting the expression of g_{d0} in (2.19) into (2.20), the thermal noise current spectral density of long-channel MOSFETs in the saturation region is

$$\frac{\overline{i_n^2}}{\Delta f} = \frac{8}{3} kT \mu_{eff} C_{ox} \frac{W}{L} V_{eff} \quad \left[\frac{\mathrm{A}^2}{\mathrm{H_z}}\right] \tag{2.21}$$

For short-channel MOSFETs in the saturation region, equation (2.21) will not be valid since the mobility of the carriers is degraded due to the lateral electric field and the infinitesimal channel segment cannot be regarded as a linear resistor any more [26]. Reference [25] reported large excess channel noise in short-channel MOS transistors. The representation of (2.20) can still be used, with the parameter γ observed to be two or three times larger than that of long-channel MOS transistors in saturation [26], [27], [31].

The thermal noise behavior of short-channel MOSFETs in the saturation region is not well understood yet and even controversial. Many approaches are based on a two-section channel model in which the channel of the MOSFET is divided into two regions: a gradual channel region of length L_{elec} and a velocity saturation region of length ΔL [27], as shown in Figure 2.4.

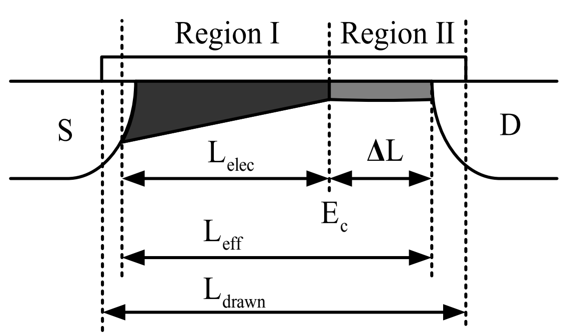


Figure 2.4 Channel cross section of a MOSFET in the saturation region.

References [25] and [28]-[30] attempted to explain the excess factor γ by introducing the hot electron effect in velocity saturation region. But the reported thermal noise in MOSFETS fabricated in 0.18μm processes shows that the excess factor γ does not exceed more than two at strong inversion [31], [32]. And some experimental works even address that the excess noise due to hot electrons in velocity saturation region is negligible [31], [33]. Reference [27] states that there is no noise current (or current fluctuation) generated in the velocity saturation region, because Ohm's law is not valid in the velocity saturation region and the carriers which travel at their saturation velocity will not respond to the local change of the electric field caused by the noise voltage fluctuation.

The latest work in [26] developed a model which takes into account both velocity saturation effect and carrier heating effect, while ignoring the noise in the velocity saturation region. It shows that the well-known formula (2.16), which is valid for the long-channel MOS devices, can be extended into the short channel. This model has been verified by measurement results from MOSFETs in a 0.18μm process, and the extracted excess factor γ increased steadily with the drain bias in the saturation region due to the channel length modulation effect.

In order to simplify the analysis, this work will still use the compact noise model in (2.20), and let the factors γ_s and γ_t vary to account for the excess noise in short-channel MOSFETs operating in the saturation and triode region respectively. Therefore for short-channel MOSFETs in saturation, the thermal noise current spectral density is modeled by

$$\frac{\overline{i_n^2}}{\Delta f} = 4KT\gamma_s \cdot \mu_{eff} C_{ox} \frac{W}{L} V_{eff} \quad \left[\frac{\mathrm{A}^2}{\mathrm{H_z}}\right] \tag{2.22}$$

2.3.3 1/f Noise

First observed by Johnson [34] in early amplifiers, the 1/f (flicker) noise prevails in many semiconductor devices. There are two theories to explain the physical origins of 1/f noise. Originally proposed by McWhorter [35], the carrier number fluctuation theory [36]-[39] believes that the 1/f noise is attributed to the random trapping and detrapping processes of charges in the oxide traps associated with contamination and crystal defects near the Si-SiO_2 interface. The charge fluctuation results in fluctuation of the surface potential and thus modulates the channel carrier density. Since the carrier lifetime in silicon is on the order of tens of microseconds, the resulting current fluctuations are concentrated at lower frequencies [40]. Typically PMOS transistors have less 1/f noise than NMOS transistors since their majority carriers (holes) are less likely to be trapped [41].

The mobility fluctuation theory [42]-[45] explains the 1/f noise as a result of the fluctuations in the mobility based on the Hooge's empirical equation [42]. Reference [46] and [47] found that both carrier number fluctuations and correlated mobility fluctuations result in 1/f noise of MOS devices and unified 1/f noise models were proposed [46]-[49].

The BSIM3v3 noise model in [48] is available in SPICE and shows good fittings with experimental 1/f noise results. It models the 1/f noise of MOSFETs in saturation as

$$\begin{aligned}\frac{\overline{i_n^2}}{\Delta f} &= \frac{kTq^2 I_D \mu_{eff}}{10^8 C_{ox} L^2 f}\left[AIn\frac{N_0+N^*}{N_L+N^*} + B\left(N_0-N_L\right) + \frac{C}{2}\left(N_0^2-N_L^2\right)\right] \\ &\quad + \Delta L \frac{kTI_D^2}{10^8 WL^2 f}\frac{A+BN_L+CN_L^2}{\left(N_L+N^*\right)^2}\end{aligned} \tag{2.23}$$

with

$$qN_0 = qN\left(0\right) = C_{ox}\left(V_{GS}-V_T\right) \tag{2.24}$$

$$qN_L = qN(L) = C_{ox}(V_{GS} - V_T - V_{DS}) \tag{2.25}$$

$$N^* = \frac{kT(C_{ox} + C_d + C_{it})}{q^2} \tag{2.26}$$

where A, B and C are three noise fitting parameters, N_0 is the charge density at the source side, N_L is the charge density at the drain side, C_d is the depletion layer capacitance and C_{it} is the interface trap capacitance, and ΔL is the electrical channel length reduction due to channel length modulation.

Substituting the expression of I_D in (2.1) and (2.12) into (2.23), the 1/f noise current spectral density is related to the channel length L by

$$\frac{\overline{i_n^2}}{\Delta f} \propto \begin{cases} \dfrac{1}{L^3} & when \;\; L \gg L_c \\ \dfrac{1}{L^2} & when \;\; L \ll L_c \end{cases} \tag{2.27}$$

Reference [50] reported that the 1/f noise spectral density is proportional to $1/L^3$ from the experimental results of four MOSFETs with L from 0.8μm to 3.8μm, which supports the relationship of (2.27) for the long-channel case.

From (2.27), as the channel length scales down in the same semiconductor process, the 1/f noise current spectral density will increase at least quadratically, while the thermal noise current spectral density only increases linearly with a slightly increased excess factor γ according to (2.22). This results in that the 1/f noise corner, at which the 1/f noise and white noise have the same level of spectral density, increases at least linearly when the channel length scales down as illustrated in Figure 2.5.

For the scaling across different deep submicron processes, the experimental results in [51] show that with technology scaling from 0.35μm to 0.13μm at a constant drain current of 10mA, the 1/f noise current spectra density is proportional to $1/L^3$ for thin gate oxide NMOS transistors with minimum channel lengths. These results showed a much stronger dependence of L than the model of (2.23) which predicts that the 1/f noise current spectra density is proportional to $1/L^2$ when the drain current is kept constant. The authors of [51] explained their results with the process-dependent parameters in (2.23) and the use of nitrided gate oxide for the processes with feature sizes of 250nm and below.

The use of nitrided oxides has become attractive in deep submicron processes for a number of reasons [52-54]. It provides resistance to interface state generation during hot-carrier degradation, it is robust in a radiation environment, it is essential to suppress impurity diffusion into the gate oxide, and it is able to provide increased gate dielectric capacitance density without compromising the gate leakage current [55]. However, the use of nitrided oxides significantly increases the 1/f noise in MOSFETs through the introduction of interface traps [53], [56].

Summarizing the above, downscaling CMOS technologies in general raises the average 1/f noise level and introduces higher 1/f noise corner. Since the 1/f noise will be upconverted in oscillators and giving rise to a $1/f^3$ sideband around the carrier frequency, the characterization of 1/f noise becomes critical for oscillators designed in deep submicron processes.

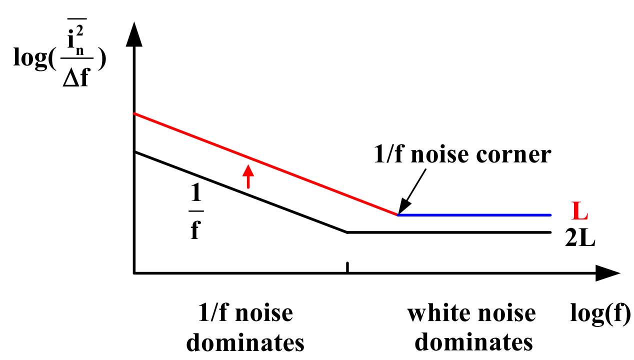


Figure 2.5 Higher 1/f noise corner for MOSFETs with shorter channel length.

Chapter 3: Technique to Relate Frequency and Time Domain Oscillator Jitter Performance

3.1 Introduction

The oscillation frequency of a VCO is specifically designed to be controlled by a voltage input V_{ctl}, which is given by

$$\omega_{out} = \omega_0 + K_{VCO} \cdot V_{ctl} \tag{3.1}$$

where ω_0 is the center frequency and K_{VCO} is the VCO gain factor in the unit of [rad/V·sec].

Phase is the integral of frequency. Assuming the initial phase to be zero, the phase at the VCO output is

$$\phi(t) = \int_0^t \omega_{out}(t) \cdot dt = \omega_0 t + K_{VCO} \int_0^t V_{ctl} \cdot dt \tag{3.2}$$

Therefore the transfer function of the VCO is [57]

$$H(s) = \frac{K_{VCO}}{s} \tag{3.3}$$

The noise sources within the VCO and at the VCO input terminal directly modulate the oscillation frequency, which results in the upconverted close-in phase noise. The typical oscillator phase noise spectrum is shown in Figure 3.1. Since the VCO is a perfect integrator, the integrated white noise is sort of random walk and is not stationary, showing a spectrum proportional to $1/f^2$, where f is the offset frequency from the carrier. The $1/f^3$ region is due to 1/f noise, showing a slope of -30dB/dec in the phase noise spectrum on a log-log scale. The corner between the $1/f^2$ and $1/f^3$ regions is called the $1/f^3$ phase noise corner, denoted by f_c in Figure 3.1, and is smaller than 1/f noise corner of the oscillator's components by a factor determined by the symmetry properties of the oscillation waveform [10].

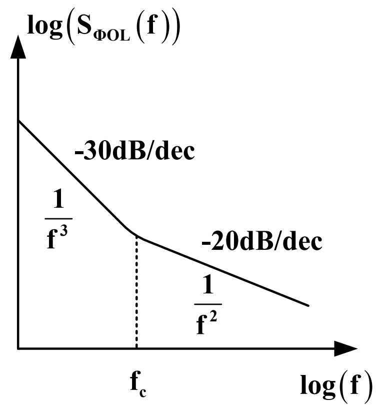


Figure 3.1 Typical oscillator phase noise spectrum.

Jitter can also be characterized in time domain by measuring the standard deviation of the jitter process using oscilloscopes. Therefore a technique for relating jitter measures from either domain is desirable so that the designer can work in whichever domain is easiest while still being able to accurately predict performance when measured by the end user.

In [8], a technique was developed to relate frequency and time domain oscillator performance for oscillators dominated by white noise sources. The VCO design process is simplified by assuming the $1/f^3$ phase noise corner to lie below the PLL loop bandwidth and the 1/f noise contribution will be filtered out by the loop. Figure 3.2 shows the measured phase noise p.s.d. from a 155MHz ring oscillator in a 5GHz f_T bipolar process [8]. The $1/ f^3$ corner frequency is so low that only white noise upconverted phase noise is observed.

Although the use of a deep submicron process allows the possibility of higher VCO frequency, it also introduces the problem of a higher 1/f noise corner. Figure 3.3 shows a typical phase noise spectrum from the single-ended ring oscillators implemented in this work in a 0.18μm process. The $1/f^3$ phase noise corner is located at an offset frequency of approximately 1MHz for a 60MHz carrier, locked by a PLL with 30 kHz loop bandwidth. Therefore the assumption of white noise domination is no longer valid, and the theory in [8] needs to be extended.

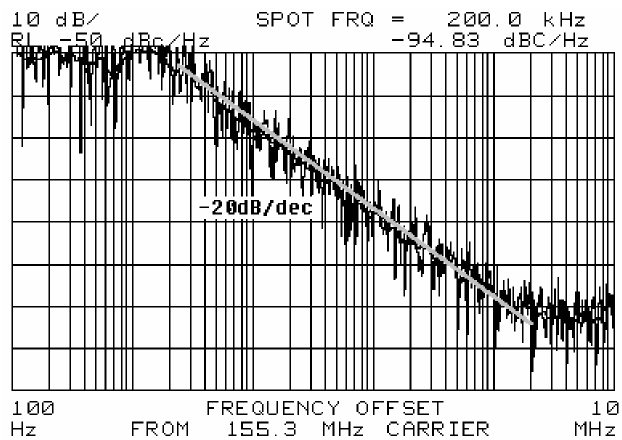


Figure 3.2 Phase noise p.s.d. of a 155MHz VCO in a 5GHz f_T bipolar process.

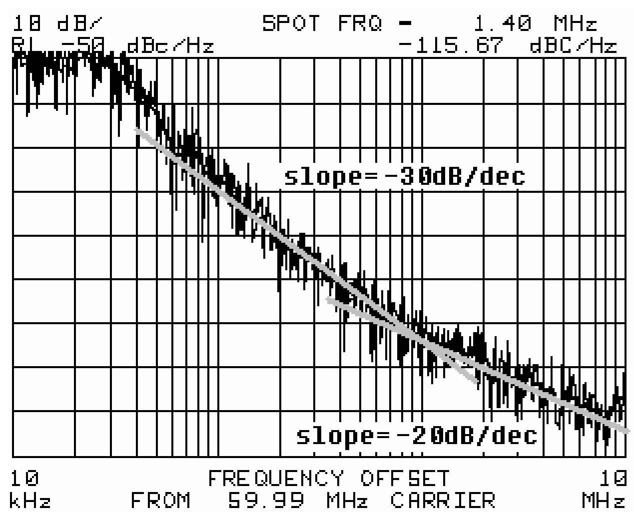


Figure 3.3 Typical phase noise p.s.d. of VCOs in a 0.18μm process.

The high $1/f^3$ phase noise corner is due to the poor 1/f device noise in the deep submicron process. Even though the $1/f^3$ phase noise corner can be significantly lowered by improving waveform symmetry [10], the applicability is limited for ring oscillators since it is impossible to get symmetric rising and falling edges. Differential ring oscillators do not have symmetry advantage over single-ended peers since it is the symmetry of the half circuits that matters [10].

The goal of this chapter is to find an easy link between time domain and frequency domain figures of merit without getting into the details of where the $1/f^3$ phase noise corner comes from. Section 3.2 will briefly review the characterization of VCO jitter and phase noise performance. The development of the technique to relate frequency and time domain oscillator performance in the presence of non-negligible 1/f noise sources is presented in Section 3.3. Section 3.4 gives supporting experimental results.

3.2 VCO Jitter and Phase Noise Performance Characterization

3.2.1 Frequency Domain Using Spectrum Analyzers

The free-running VCO can be characterized in the frequency domain using a spectrum analyzer. The spectrum analyzer can measure the spectral density of phase fluctuations per unit bandwidth. As shown in Figure 3.4, the output of the free-running VCO is directly applied to the spectrum analyzer. The resulting spectrum, normalized to the carrier power, is $S_{\Phi OL}(f)$, as shown in Figure 3.5a. Figure 3.5b shows the typical phase noise p.s.d. measured by spectrum analyzers on a log-log scale.

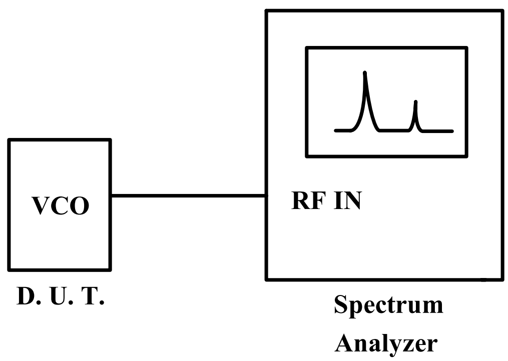
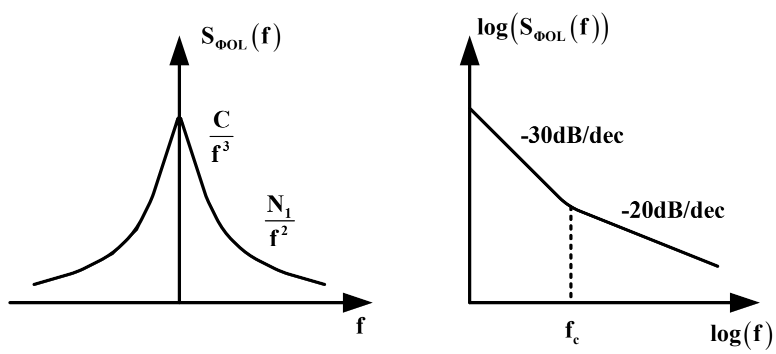


Figure 3.4 Free-running oscillators measurement in frequency domain.



(a) VCO output spectrum. (b) Typical phase noise p.s.d.

Figure 3.5 Frequency domain measurement result: VCO open loop.

White noise upconverted phase noise dominates at higher offset frequencies. Its phase noise p.s.d. can be fitted to a characteristic [7]

$$S_W = \frac{N_1}{f^2} \quad \left[\mathrm{rad}^2/\mathrm{Hz}\right] \tag{3.4}$$

where N_1 is the frequency domain white noise figure of merit in the unit of [$\mathrm{rad}^2 \cdot \mathrm{Hz}$]. This spectrum shows a slope of -20dB/dec in Figure 3.5b.

The -30dB/dec region is upconverted from the 1/f noise. Its p.s.d. has the form of

$$S_{1/f}(f) = \frac{C}{|f|^3} \tag{3.5}$$

where C is the frequency domain 1/f noise figure of merit in the unit of [$\mathrm{rad}^2 \cdot \mathrm{Hz}^2$].

Since at $1/f^3$ phase noise corner frequency f_c, the phase noise due to white noise and the 1/f noise are equal, the figure of merit C is calculated as

$$C = N_1 f_c \tag{3.6}$$

In order to ensure that only phase noise power is present in the waveform, some form of limiting is usually used to remove the amplitude noise in practice [58]. And to get accurate results, the noise floor of the spectrum analyzer must be lower than the phase noise of the oscillator under test.

This is a simple, quick test to obtain frequency domain figures of merit N_1 and C. But in general, the free-running VCO drifts randomly with typical rate of 10 ppm/min [59], while it usually takes 2 to 5 minutes for the spectrum analyzer to perform a single test, which introduces measurement errors. To stabilize the frequency of the device-under-test, the VCO is usually locked to an ultra low noise reference clock in the measurement, rather than free-running, which will be discussed in Chapter 4.

3.2.2 Time Domain: Equivalent Time Oscilloscope Method

Most types of jitter can be characterized in the time domain by equivalent time sampling oscilloscopes using the method of two-sample standard deviation [8].

The idea is to construct a histogram of threshold crossings from the VCO output waveform during a user defined time window. As illustrated in Figure 3.6, the output of the free-running VCO is directly applied to the sampling oscilloscope as both the trigger and the input. The sampling oscilloscope compares the phase difference between transitions in the clock waveform, separated by a delay ΔT derived from the internal time base of the sampling oscilloscope. Then a distribution of the threshold crossing times is observed, and the standard deviation of this distribution, $\sigma_{\Delta T}$, is the rms jitter accumulated in the time delay ΔT.

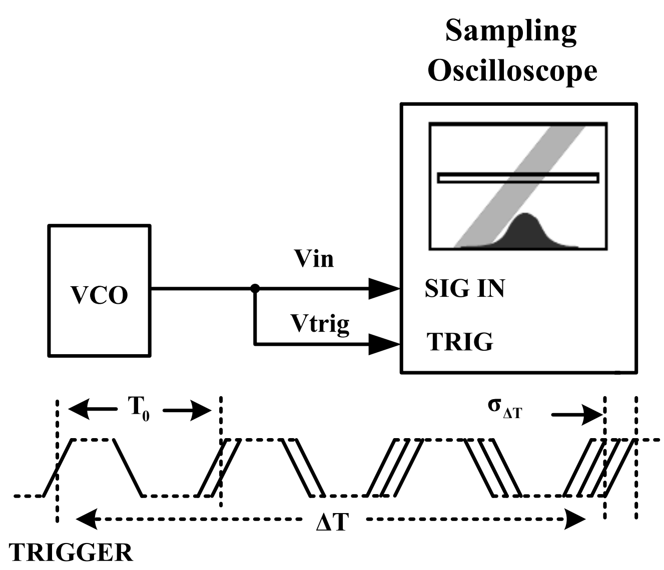


Figure 3.6 Jitter measurement over time delay ΔT.

The amount of the accumulated jitter depends on the delay ΔT. As shown in Figure 3.7, by varying the time delay ΔT and repeating the measurement procedure, the measured standard deviation $\sigma_{\Delta T}$ can be plotted as a function of delay ΔT on a log-log scale, which is called a "kappa plot".

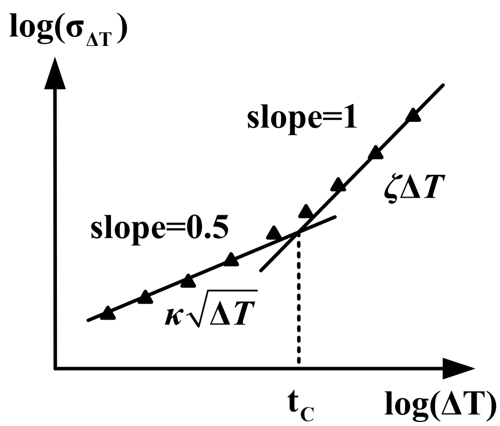


Figure 3.7 Measurement result: Time domain, open loop.

For short delays the VCO is dominated by white noise. The rms jitter after time delay ΔT is [7]

$$\sigma_W(\Delta T) = \kappa\sqrt{\Delta T} \quad [\text{sec rms}] \tag{3.7}$$

where κ is the time domain white noise figure of merit. Therefore it shows a slope of 0.5 in the kappa plot on a log-log scale. κ is related to the frequency domain white noise figure of merit N_1 by [7]

$$\kappa = \frac{\sqrt{N_1}}{f_0} \quad \left[\sqrt{\mathbf{sec}}\right] \tag{3.8}$$

where f_0 is the VCO's oscillating frequency.

For longer delays over which the VCO is dominated by the 1/f noise, (3.7) changes to [15]

$$\sigma_{1/f}\left(\Delta T\right) = \zeta \Delta T \tag{3.9}$$

where ζ is the time domain 1/f noise figure of merit, and it is a dimensionless constant.

The time delay at the "corner" between these two regions is the 1/f transition time, denoted by t_c in Fig. 3.7.

In this measurement, the internal time base of the sampling oscilloscope is the reference which defines the delay interval ΔT. Therefore the jitter floor of the sampling oscilloscope must be better than the clock under test.

3.2.3 Time Domain: Real-Time Oscilloscope Method

The equivalent time sampling oscilloscopes can only acquire statistics for a clock edge at a single delay [60]. Recently introduced digital oscilloscopes [61] can acquire jitter information on thousands of clock edges in a single shot and store results as time interval error (TIE) data, as illustrated in Figure 3.8.

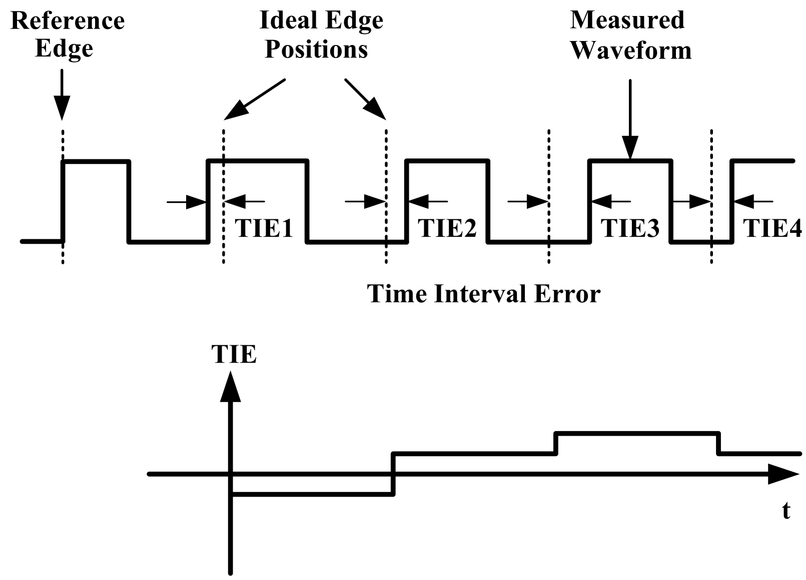


Figure 3.8 Time interval error.

TIE is the timing variation for each active clock edge from the ideal position [61], given by:

$$TIE_{clock}\left[n\right] = T_{clock}[n] - n \cdot T_0 \tag{3.10}$$

where $TIE_{clock}[n]$ is the clock time interval error after the n_{th} clock cycle, $T_{clock}[n]$ is the specified clock edge, and T_0 is the calculated ideal clock period.

TIE actually consists of samples of the continuous time phase noise process. The time domain figures of merit κ and ζ can be obtained by post-processing the TIE data. The algorithm is as follows:

1. Construct the distributions of threshold crossing times by calculating the distribution matrix

$$Distribution[n,\ j] = TIE_{clock}[j+n] - TIE_{clock}[j] \quad n, j = 1, 2, \ldots \tag{3.11}$$

The n_{th} row of the distribution matrix forms the histogram of threshold crossing times with jitter accumulated in n clock cycles.

2. Calculate the standard deviation of each row in the distribution matrix to obtain the rms jitter accumulated in different cycles as illustrated in Figure 3.9:

$$\sigma(nT_0) = stdev\left(TIE_{clock}[j+n] - TIE_{clock}[j]\right) \quad n, j = 1, 2, \ldots \tag{3.12}$$

3. Plot the rms jitter vector $\sigma(nT_0)$ versus its time stamp, nT_0, to get "Kappa" plot. Then the time domain figures of merit κ and ζ can be extracted.

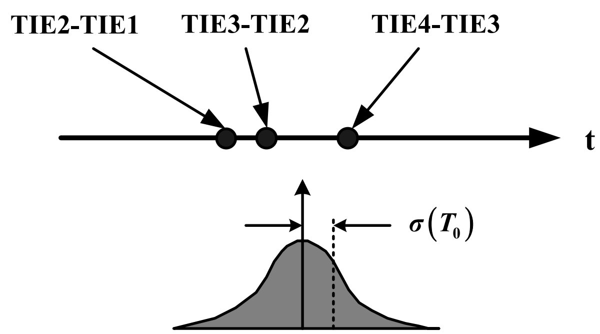
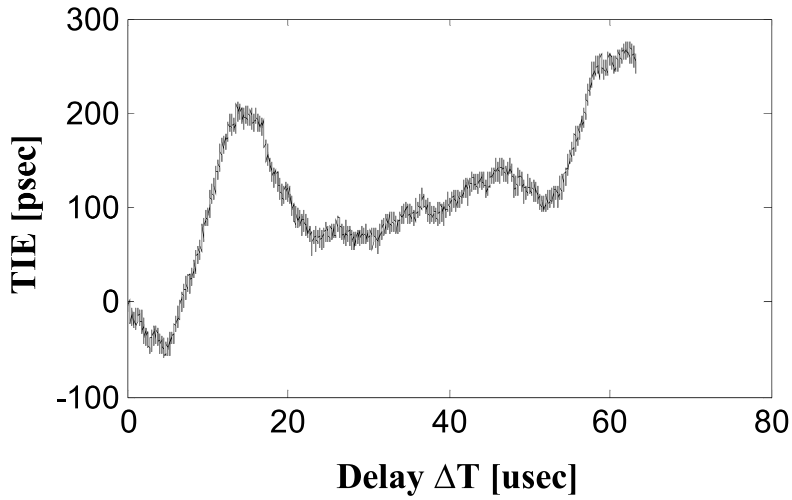
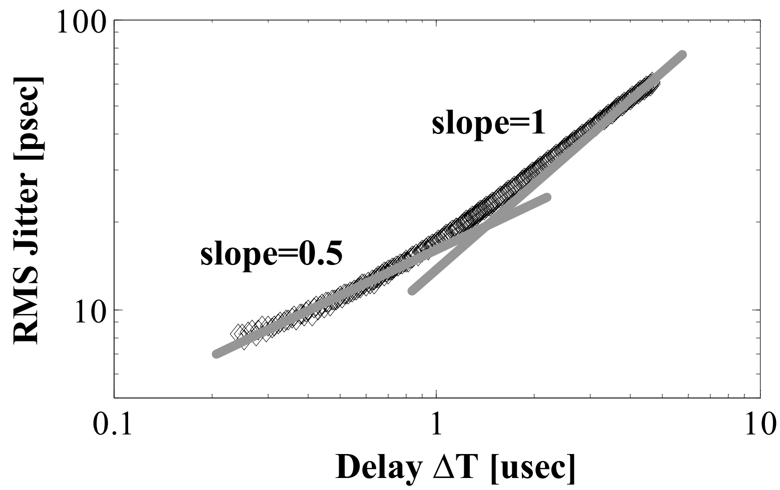


Figure 3.9 TIE data post-processing.

Appendix A shows a Matlab implementation of this proposed algorithm. Fig. 3.10 shows the TIE and post-processed kappa plot for a 150MHz VCO designed in this work.



(a) Time Interval Error for a 150MHz VCO in this work.



(b) Extracted kappa plot from the TIE data in (a).

Fig. 3.10 TIE and post-processed kappa plot for a 150MHz VCO.

3.3 Theoretical Development

3.3.1 Near-carrier Oscillator Spectrum

As discussed in Section 3.2.1, the typical phase noise spectrum for an open-loop VCO is as shown in Fig. 3.11a.

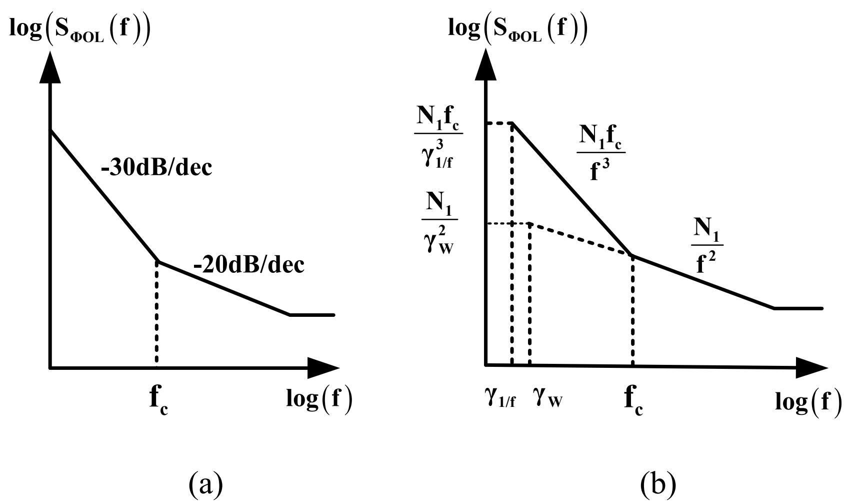


Fig. 3.11 Phase noise p.s.d with close-in corner frequencies.

The white noise integrated phase noise has the p.s.d. in the form of

$$S_W = \frac{N_1}{f^2} \tag{3.13}$$

The 1/f noise upconverted phase noise has the p.s.d. in the form of

$$S_{1/f}(f) = \frac{N_1 f_c}{f^3} \tag{3.14}$$

However both equation (3.13) and (3.14) indicate that the phase noise power goes to infinity at the carrier frequency which is obviously contradictory to reality. In [12] and [13] mathematical models are developed for the near-carrier phase noise spectrum, but the theory is still incomplete. For practical purposes, cutoff frequencies $\gamma_{1/f}$ and γ_w are assumed to exist at very small offset frequencies (several hertz or below) in this work so that the phase noise power at the carrier is finite, as shown in Fig. 3.11b. Therefore, the double sideband p.s.d. of the white noise and 1/f noise upconverted phase noise are modeled as

$$S_W(f) = \frac{\frac{N_1}{\gamma_w^2}}{1+\left(\frac{f}{\gamma_w}\right)^2} \tag{3.15}$$

$$S_{1/f}(f) = \frac{\frac{N_1 f_c}{\gamma_{1/f}^3}}{1+\left(\frac{|f|}{\gamma_{1/f}}\right)^3} \tag{3.16}$$

3.3.2 Relationship between Jitter and Phase Noise

The time domain jitter measurement using the method of two-sample standard deviation in Section 3.2.2 is actually a sampling process of the phase difference between the threshold crossings of the reference and the observed transitions. The testing equipment, such as the sampling oscilloscope, can be modeled as a linear time invariant (LTI) sampling system with the impulse response of

$$h(t) = \delta\left(t + \Delta T\right) - \delta\left(t\right) \tag{3.17}$$

The transfer function of this LTI system is

$$H(j\omega) = e^{j\omega\Delta T} - 1 \tag{3.18}$$

If the upconverted phase noise is wide sense stationary (WSS), the variance of the jitter process at the output of this system can be obtained by

$$\sigma^2\left(\Delta T\right) = \int_{-\infty}^{\infty} S_\Phi(f)\left|H\left(j2\pi f\right)\right|^2 df \quad [\text{rad}^2] \tag{3.19}$$

Therefore the relationship between jitter and phase noise is

$$\sigma^2\left(\Delta T\right) = 8\int_0^{\infty} S_\Phi(f)\sin^2\left(\pi\Delta T f\right)df \quad [\text{rad}^2] \tag{3.20}$$

or

$$\sigma^2\left(\Delta T\right) = \frac{2}{\pi^2 f_0^2}\int_0^{\infty} S_\Phi(f)\sin^2\left(\pi\Delta T f\right)df \quad [\text{sec}^2] \tag{3.21}$$

3.3.3 Jitter Due to White Noise

Since the jitter process due to white noise over a finite time interval is WSS [8], its variance can be calculated by applying (3.21)

$$\sigma_W^2(\Delta T) = \frac{2}{\pi^2 f_0^2} \int_0^\infty \frac{N_1/\gamma_w^2}{1+\left(f/\gamma_w\right)^2} \sin^2\left(\pi \Delta T f\right) df \tag{3.22}$$

The result is

$$\sigma_W^2(\Delta T) = \frac{N_1}{2\pi\gamma_w f_0^2}\left(1-\exp\left(-2\pi\gamma_w \Delta T\right)\right) \tag{3.23}$$

Since the time delay ΔT usually cannot exceed several milliseconds in the measurements due to the record length limitation of the sampling or the digital oscilloscope, it is reasonable to assume

$$2\pi\gamma_w \Delta T << 1 \tag{3.24}$$

Using the Taylor approximation,

$$\exp\left(-2\pi\gamma_w \Delta T\right) \approx 1 - 2\pi\gamma_w \Delta T \tag{3.25}$$

equation (3.23) is approximated by

$$\sigma_W^2(\Delta T) = \frac{N_1}{f_0^2}\Delta T \tag{3.26}$$

Therefore the rms jitter due to white noise is

$$\sigma_W(\Delta T) = \frac{\sqrt{N_1}}{f_0}\sqrt{\Delta T} = \kappa\sqrt{\Delta T} \quad [\text{sec rms}] \tag{3.27}$$

Result of (3.27) matches the result in [7] for white noise integrated phase noise.

3.3.4 Jitter due to the 1/f Noise

The 1/f noise is a non-stationary process, and there is still controversy about its modeling. But for practical purposes it can be modeled as a colored stationary process [14]. After the upconversion, the integrated jitter process with p.s.d. modeled by (3.16) is well-behaved and upper bounded. For simplicity in analysis, it is also modeled as a WSS process. Under this assumption, the jitter due to the 1/f noise can be computed by applying (3.21)

$$\sigma_{1/f}^2\left(\Delta T\right)=\frac{2}{\pi^2 f_0^2}\int_0^\infty \frac{N_1 f_c/\gamma_{1/f}^3}{1+\left(f/\gamma_{1/f}\right)^3}\sin^2\left(\pi\Delta T f\right)df \tag{3.28}$$

The integral in (3.28) is too complicated to be computed analytically. For simplification, (3.28) is approximated as the summation of the integral of two parts, the plateau region and the -30dB/dec region in Fig. 3.11.

The jitter due to the plateau region of the phase noise p.s.d. is

$$\sigma_P^2\left(\Delta T\right)=\frac{2}{\pi^2 f_0^2}\int_0^{\gamma_{1/f}} \frac{N_1 f_c}{\gamma_{1/f}^3}\sin^2\left(\pi\Delta T f\right)df \tag{3.29}$$

which equals

$$\sigma_P^2\left(\Delta T\right)=\frac{N_1 f_c}{\pi^2 f_0^2\gamma_{1/f}^2}\left(1-\frac{\sin\left(2\pi\gamma_{1/f}\Delta T\right)}{2\pi\gamma_{1/f}\Delta T}\right) \tag{3.30}$$

Under the condition of (3.24), and using the Taylor approximation,

$$\sin\left(x\right)\approx x-\frac{1}{6}x^3 \tag{3.31}$$

(3.30) can be simplified to

$$\sigma_P^2\left(\Delta T\right)=\frac{2N_1 f_c}{3f_0^2}\Delta T^2 \tag{3.32}$$

Result of (3.32) does not show any dependency on the cut-off frequency $\gamma_{1/f}$. The reason is that the shaping function

$$\sin^2\left(\pi\Delta Tf\right) \approx \left(\pi\Delta Tf\right)^2 \quad \text{when } \pi\Delta Tf \ll 1 \tag{3.33}$$

and the integral of this shaping function over the frequency in $[0, \gamma_{1/f}]$ is proportional to $\gamma^3_{1/f}$, while the p.s.d. in the plateau region is proportional to $1/\gamma^3_{1/f}$, with the results that the product of both is independent of $\gamma_{1/f}$.

The jitter due to the -30dB/dec region is computed by

$$\sigma_F^2\left(\Delta T\right) = \frac{2}{\pi^2 f_0^2}\int_{\gamma_{1/f}}^{\infty} \frac{N_1 f_c}{f^3}\sin^2\left(\pi\Delta Tf\right)df \tag{3.34}$$

The result is

$$\sigma_F^2\left(\Delta T\right) = \frac{N_1 f_c \beta}{f_0^2}\Delta T^2 \tag{3.35}$$

where

$$\beta = \left(\frac{\sin\left(\pi\gamma_{1/f}\Delta T\right)}{\pi\gamma_{1/f}\Delta T}\right)^2 + \frac{\sin\left(2\pi\gamma_{1/f}\Delta T\right)}{\pi\gamma_{1/f}\Delta T} - 2\cos\text{int}\left(2\pi\gamma_{1/f}\Delta T\right) \tag{3.36}$$

The function cosint(x) in (3.36) is the cosine integral function defined by

$$\cos\text{int}(x) = \gamma + In(x) + \int_0^x \frac{\cos t - 1}{t}dt \tag{3.37}$$

where γ is Euler's constant 0.577215664… Under the condition of (3.24),

$$\frac{\sin\left(2\pi\gamma_{1/f}\Delta T\right)}{2\pi\gamma_{1/f}\Delta T} \approx 1 \tag{3.38}$$

Therefore,

$$\beta \approx 3 - 2\cos\text{int}\left(2\pi\gamma_{1/f}\Delta T\right) \tag{3.39}$$

The total jitter due to the 1/f noise is obtained by the summation of (3.32) and (3.35), which is

$$\sigma_{1/f}^2\left(\Delta T\right)=\frac{N_1 f_c}{f_0^2}\alpha^2\Delta T^2 \tag{3.40}$$

where

$$\alpha^2=\frac{11}{3}-2\cos\mathrm{int}\left(2\pi\gamma_{1/f}\Delta T\right) \tag{3.41}$$

So the rms jitter due to the 1/f noise is

$$\sigma_{1/f}\left(\Delta T\right)=\frac{\sqrt{N_1 f_c}\alpha}{f_0}\Delta T \quad \left[\text{sec rms}\right] \tag{3.42}$$

or

$$\sigma_{1/f}\left(\Delta T\right)=\kappa\sqrt{f_c}\alpha\Delta T \quad \left[\text{sec rms}\right] \tag{3.43}$$

Thus the analytical expression of the figure of merit ζ is

$$\zeta=\kappa\sqrt{f_c}\alpha \tag{3.44}$$

Fig. 3.12 shows that α varies very slowly versus $2\pi\gamma_{1/f}\Delta T$ through an 11 decade range of arguments. In the measurement of just 2 or 3 decades, α is almost a constant. Therefore from (3.43) we are able to approximate that the rms jitter due to the 1/f noise is proportional to the measurement time delay ΔT.

Since 1/f noise dominates at longer delays and the jitter due to 1/f noise usually appears in the time delay interval of [1E-7, 1E-5] second, α can be taken to be approximately as 5. Hence equation (3.43) can be simplified to

$$\sigma_{1/f}\left(\Delta T\right)\approx 5\kappa\sqrt{f_c}\Delta T \quad \left[\text{sec rms}\right] \tag{3.45}$$

And the frequency domain and time domain 1/f noise figures of merit are related as

$$\zeta \approx \frac{5\sqrt{C}}{f_0} \tag{3.46}$$

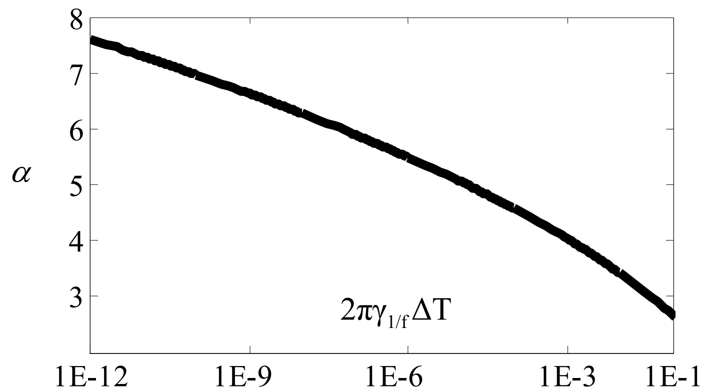


Fig. 3.12 Evaluation of α.

3.3.5 1/f Transition Time t_c

The total rms jitter over time delay ΔT is

$$\sigma_{total} = \kappa\sqrt{\Delta T + f_c \alpha^2 \Delta T^2} \quad [\text{sec rms}] \tag{3.47}$$

By solving for the time at which equation (3.27) and (3.43) are equal, the 1/f transition time t_c is related to the $1/f^3$ phase noise corner f_c by

$$t_c = \frac{1}{\alpha^2 f_c} \quad [\text{sec}] \tag{3.48}$$

3.4 Experimental Verification

To verify the model proposed in Section 3.3, jitter and phase noise measurements were made for single-ended CMOS ring oscillators fabricated in TSMC 0.18μm process. The measurements were performed using a LeCroy Wavemaster 8600A digital oscilloscope [61] and an Agilent Technologies E4440A spectrum analyzer [63].

The ring oscillators under test were implemented with the so-called current-starved inverters [40] as shown in Figure 3.13. M_1 and M_2 form the CMOS inverter. M_3 and M_4 are the control transistors which determine the current available for switching and thus control the speed of the VCO. The PMOS transistors are sized relative to NMOS transistors to provide rise and fall times that as symmetric as possible, and the control transistors are twice the size of the switching transistors.

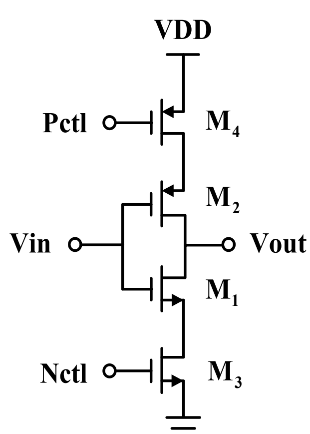


Figure 3.13 The current-starved inverter.

Table 3.1 gives information on the VCO configurations and measurement results, as well as predicted ζ based on the measured κ and f_c by equation (3.45). The agreement between the predicted and measured results is within 10%.

TABLE 3.1 Measured and predicted ζ for implemented ring oscillators.

Index	N	f_0 [Hz]	W/L of M1 [μm/μm]	κ [$\sqrt{\text{sec}}$]	f_c [Hz]	Pred. ζ	Meas. ζ	Error
VCO set I	25	250M	10/0.18	4.49E-09	1.14M	2.40E-05	2.62E-05	8.51%
			20/0.18	3.13E-09	950k	1.53E-05	1.60E-05	4.66%
			60/0.18	1.80E-09	1.13M	9.57E-06	9.12E-06	4.90%
			100/0.18	1.45E-09	805k	6.50E-06	7.07E-06	8.06%
VCO set II	7	250M	10/0.6	4.48E-09	921k	2.15E-05	2.09E-05	2.86%
			20/0.6	3.64E-09	1.16M	1.96E-05	1.82E-05	7.70%
			60/0.6	2.81E-09	884k	1.32E-05	1.29E-05	2.40%
			100/0.6	2.49E-09	878k	1.17E-05	1.07E-05	9.03%
VCO set III	3	90M	10/1.8	9.76E-09	766k	4.27E-05	4.13E-05	3.42%
			20/1.8	7.13E-09	740k	3.07E-05	3.01E-05	1.88%
			60/1.8	5.68E-09	699k	2.37E-05	2.20E-05	7.93%
			100/1.8	4.95E-09	689k	2.05E-05	2.18E-05	5.76%

Chapter 4: Technique to Relate Frequency and Time Domain PLL Jitter Performance

4.1 Loop and Noise Transfer Function

A phase-locked loop is basically and oscillator whose frequency is locked onto a clock reference by a feedback control loop [64]. Figure 4.1 shows a block diagram of the PLL consisting of a phase detector (PD), a loop filter (LP), and a VCO. θ_i is the input phase that PLL is trying to track, and θ_o is the phase of the VCO output. θ_n represents the phase noise of the VCO referred to its output. K_d is the phase-detector gain factor and is measured in unit of [V/rad]. K_{VCO} is the VCO gain factor and has the unit of [rad/V·sec]. Any of the PLL components shown in Figure 4.1 can contribute to jitter. But usually jitter from the VCO dominates [8].

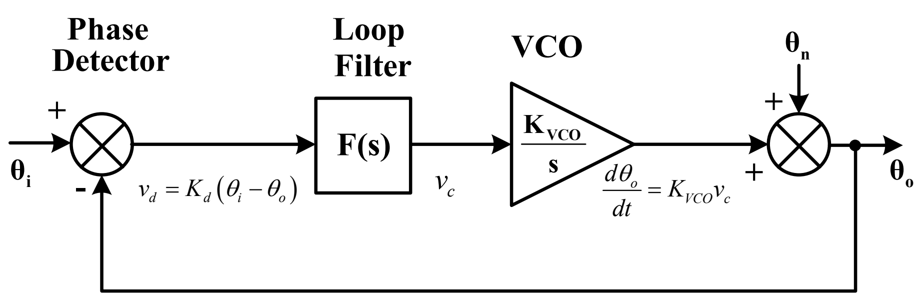


Figure 4.1 Basic block diagram of the PLL.

The signal transfer function $H_s(s)$ from θ_i to θ_o is

$$H_s(s) = \frac{\theta_o}{\theta_i} = \frac{K_d K_{VCO} F(s)}{s + K_d K_{VCO} F(s)} \tag{4.1}$$

The VCO output-referred phase noise transfer function $H_n(s)$ from θ_n to θ_o is

$$H_n(s) = \frac{\theta_o}{\theta_n} = \frac{s}{s + K_d K_{VCO} F(s)} = 1 - H_s(s) \tag{4.2}$$

When lag-lead compensation is used, the PLL is a second-order system [57]. In clock recovery PLLs, however, it is common to overdamp the loop to avoid peaking in the jitter transfer function [8], and the loop transfer function can be approximated as a first-order system by

$$H_s(s) = \frac{2\pi f_L}{s + 2\pi f} \tag{4.3}$$

$$H_n(s) = \frac{s}{s + 2\pi f} \tag{4.4}$$

where f_L is the loop bandwidth.

Equation (4.3) shows that the PLL acts as a low-pass filter for the input phase. The output phase of the PLL is only able to follow input phase fluctuations that occur at frequencies below the loop bandwidth f_L while the input phase fluctuations at frequencies above f_L are attenuated at the output. Equation (4.4) indicates that the PLL acts as a high-pass filter for the VCO phase noise, and is able to attenuate VCO phase noise that occurs at frequencies below f_L. Figure 4.2 shows Bode plots of the PLL loop transfer functions, and Figure 4.3 shows the phase noise shaping by the PLL.

Although the open loop VCO noise process is nonstationary, the process at the output of the closed loop VCO is stationary, due to shaping of the noise by the feedback loop [8]. This means transform techniques can be used when the PLL loop is closed.

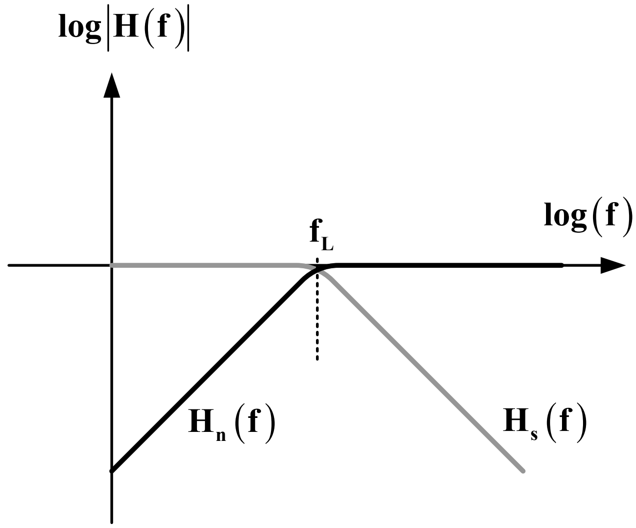


Figure 4.2 Bode plots of the PLL loop transfer functions.

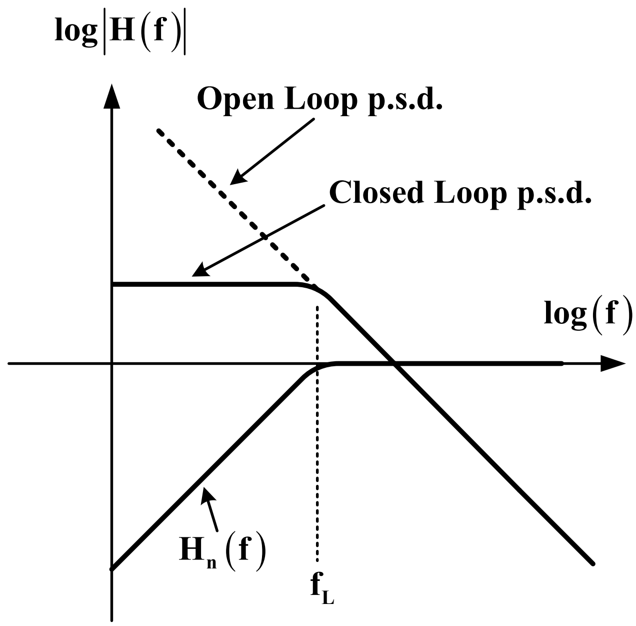
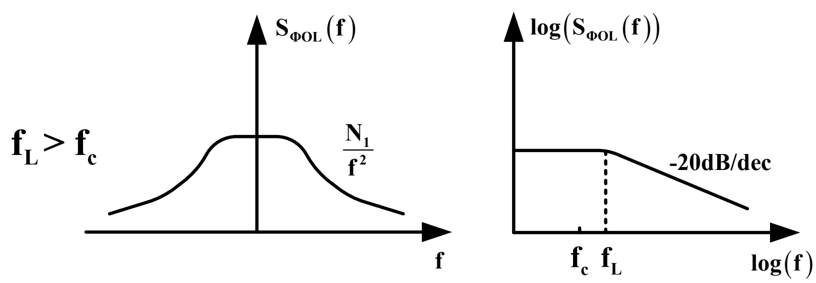


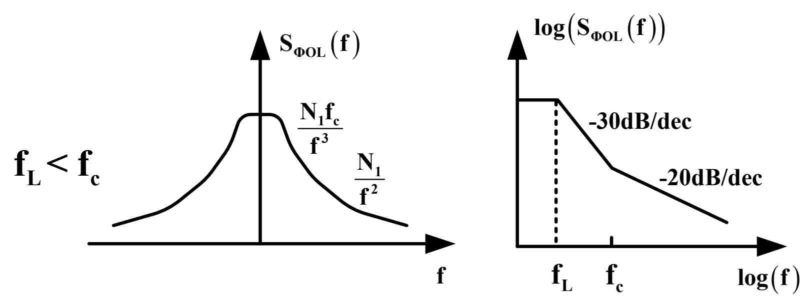
Figure 4.3 Phase noise shaping by the PLL.

4.2 Frequency Domain Using Spectrum Analyzers

Depending on whether the PLL loop bandwidth is able to cover the $1/f^3$ phase noise corner f_c, two different types of phase noise spectrum will be observed by spectrum analyzers as illustrated in Figure 4.4. Since the VCO is locked to a reference clock, the measured phase noise power is the sum of the jitter contributions from both the reference clock and the VCO. As long as the jitter of the reference clock is much smaller than that of VCO, the VCO will be the dominant contributor of the phase noise at all offset frequencies.



(a) PLL phase noise spectrum when $f_L > f_c$.



(b) PLL phase noise spectrum when $f_L < f_c$.

Figure 4.4 Typical PLL phase noise spectrum.

Equation (4.4) shows that the noise transfer function of PLL corresponds to the first-order high-pass transfer function. When the PLL loop bandwidth f_L is able to cover the corner frequency f_c, only the -20dB/dec region is observed in the phase noise spectrum, as shown in Figure 4.4a. Most of the 1/f noise contribution is filtered out by the loop and negligible. The closed loop phase noise p.s.d. has the form of [8]

$$S_W\left(f\right)=\frac{N_1/f_L^2}{1+\left(f/f_L\right)^2} \tag{4.5}$$

When the PLL loop bandwidth f_L is lower than the corner frequency f_c, the phase noise due to 1/f noise dominates at the offset frequencies lower than f_c. After the shaping by the first order loop, the slope of the 1/f noise upconverted phase noise should theoretically change from -30dB/dec to -10dB/dec at the offset frequencies below f_L in the measured spectrum on a log-log scale. But in reality, there are always extra poles at lower frequency which will further shape the loop [8]. Thus it is reasonable to assume the phase noise power approaches a constant at offset frequencies lower than f_L, as illustrated in Figure 4.4b. The introduced error by this assumption is negligible since the variance (average power) of the PLL jitter process is the integral of the phase noise p.s.d. over all frequencies [8], or the area under the phase noise p.s.d. in Figure 4.4. Under this approximation, the closed loop p.s.d. can be expressed as

$$S_{\Phi CL}(f)=\underbrace{\frac{N_1 f_c/f_L^3}{1+(|f|/f_L)^3}}_{S_{1/f}(f)}+\underbrace{\frac{N_1/f_L^2}{1+(f/f_L)^2}}_{S_W(f)} \tag{4.6}$$

4.3 Time Domain, PLL Clock Referenced

In this measurement technique, the setup is as shown in Figure 4.5. The reference clock is used as the trigger while the PLL output is applied to the input of the sampling oscilloscope. In the presence of jitter, the distribution of threshold crossing times is observed. The standard deviation of this distribution is the output rms jitter of this PLL system, σ_x, as shown in Figure 4.6.

As discussed in Section 4.1, with the PLL loop closed, the phase noise process is stationary. According to the Wiener-Khinchin theorem, the variance of the jitter process can be obtained by integrating the phase noise p.s.d. over all frequencies.

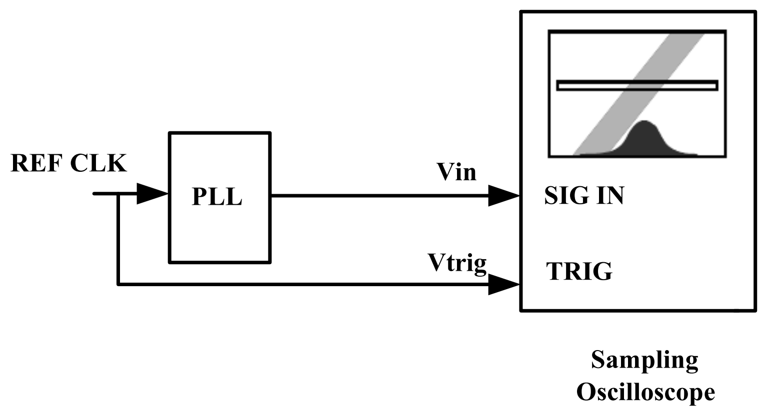


Figure 4.5 Measurement technique: Time domain, PLL clock referenced.

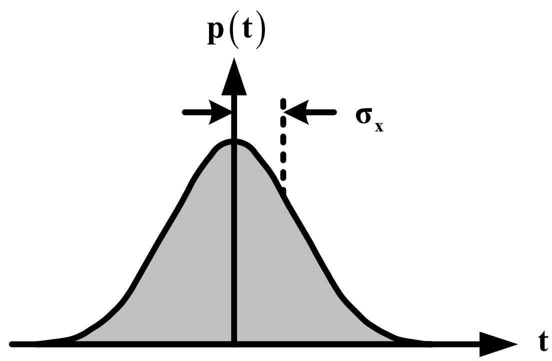


Figure 4.6 Measurement result: Time domain, PLL clock reference.

4.3.1 Case I: White Noise Dominated ($f_L > f_c$)

In the case that the $1/f^3$ phase noise corner f_c is located inside the PLL loop bandwidth f_L, white noise dominates. The end user's measure of jitter performance, rms jitter σ_x, is obtained by taking the integral [8]

$$\left(\sigma_x^2\right)_W = \int_{-\infty}^{+\infty} S_W\left(f\right) df = \int_{-\infty}^{+\infty} \frac{N_1 / f_L^2}{1+\left(f / f_L\right)^2} df = \frac{N_1 \pi}{f_L} \tag{4.7}$$

$$\left(\sigma_x\right)_W = \sqrt{\frac{N_1 \pi}{f_L}} \quad \left[\text{rad rms}\right] \tag{4.8}$$

By normalizing to the carrier frequency f_0, σ_x can be expressed in seconds rms as

$$\left(\sigma_x\right)_W = \frac{1}{f_0}\sqrt{\frac{N_1}{4\pi f_L}} = \kappa\sqrt{\frac{1}{4\pi f_L}} \quad \left[\text{sec rms}\right] \tag{4.9}$$

Therefore, for PLL dominated by white noise, only the open loop VCO white noise figure of merit κ and the PLL loop bandwidth f_L are needed for closed loop jitter prediction. And increasing loop bandwidth f_L will help to reduce the PLL rms jitter since from (4.9),

$$\left(\sigma_x\right)_W \propto \frac{1}{\sqrt{f_L}} \tag{4.10}$$

4.3.2 Case II: In the Presence of Non-negligible 1/f Noise ($f_L < f_c$)

In the case that the $1/f^3$ phase noise corner f_c is located outside the PLL loop bandwidth f_L, the end user's measure of jitter performance σ_x is obtained by taking the integral

$$\left(\sigma_x^2\right)_{Total} = \int_{-\infty}^{+\infty} S_{\Phi CL}\left(f\right) df = \underbrace{\int_{-\infty}^{+\infty}\left(\frac{N_1 f_c / f_L^{\ 3}}{1+(\left|f\right|/f_L)^3}\right) df}_{\left(\sigma_x^2\right)_{1/f}} + \underbrace{\int_{-\infty}^{+\infty}\left(\frac{N_1 / f_L^{\ 2}}{1+(f/f_L)^2}\right) df}_{\left(\sigma_x^2\right)_W} \quad (4.11)$$

The result is

$$\left(\sigma_x^2\right)_{Total} = \underbrace{\frac{N_1\pi}{f_L}\cdot\frac{4f_c}{3\sqrt{3}f_L}}_{\left(\sigma_x^2\right)_{1/f}} + \underbrace{\frac{N_1\pi}{f_L}}_{\left(\sigma_x^2\right)_W} \quad \left[\text{rad}^2\right] \quad (4.12)$$

$$\left(\sigma_x\right)_{Total} = \sqrt{\frac{N_1\pi}{f_L}\left(1+\frac{4f_c}{3\sqrt{3}f_L}\right)} \quad \left[\text{rad rms}\right] \quad (4.13)$$

By normalizing to the carrier frequency f_0, σ_x can be expressed in seconds rms as

$$\left(\sigma_x\right)_{Total} = \frac{1}{f_0}\sqrt{\frac{N_1}{4\pi f_L}\left(1+\frac{4f_c}{3\sqrt{3}f_L}\right)} \quad \left[\text{sec rms}\right] \quad (4.14)$$

or

$$\left(\sigma_x\right)_{Total} = \kappa\sqrt{\frac{1}{4\pi f_L}\left(1+\frac{4f_c}{3\sqrt{3}f_L}\right)} \quad [\text{sec rms}] \quad (4.15)$$

When f_c is much less than f_L, equation (4.15) will reduce to the result of white noise dominated case as equation (4.9). As illustrated in Figure 4.7a, the 1/f noise contribution is much less than the phase noise upconverted by white noise after the filtering by the PLL.

As illustrated in Figure 4.7b, if f_c is much greater f_L, the white noise contribution is much less than the 1/f noise upconverted phase noise due to the shaping by the. The result of (4.15) can be approximated by

$$(\sigma_x)_{Total} \approx (\sigma_x)_{1/f} = \frac{\kappa}{f_L}\sqrt{\frac{f_c}{3\sqrt{3}\pi}} \quad when\ f_c \gg f_L \tag{4.16}$$

From (4.16), increasing the loop bandwidth f_L is more important to reduce the PLL rms jitter since

$$(\sigma_x)_{1/f} \propto \frac{1}{f_L} \quad when\ f_c \gg f_L \tag{4.17}$$

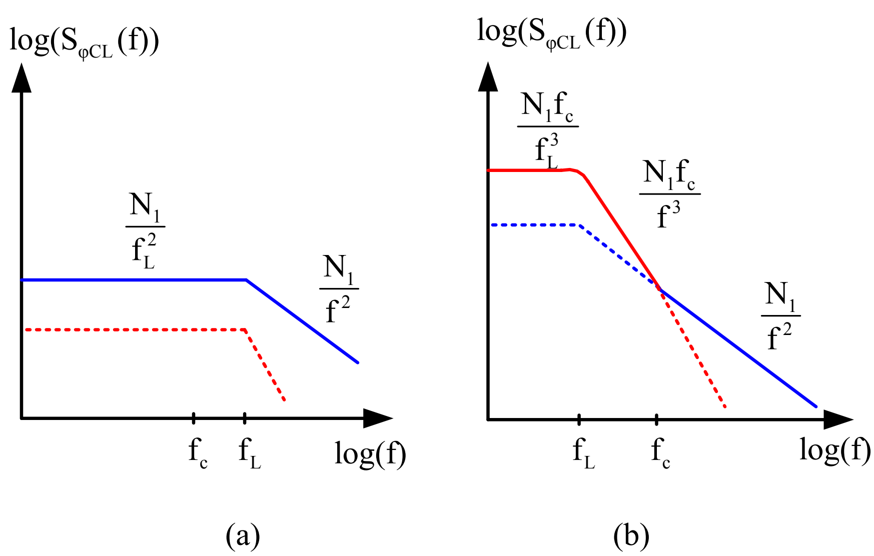


Figure 4.7 Phase noise shaping by the PLL with different loop bandwidth f_L.

4.4 Time Domain, PLL Self Referenced

The measurement setup is shown in Figure 4.8. The output of the closed-loop PLL is applied to the sampling oscilloscope as both the trigger and the input. As shown in Figure 4.9, due to the filtering of the loop, the low frequency noise will be tracked out. Therefore the rms jitter will stop accumulation after time t_L, and t_L is related to the PLL loop bandwidth f_L by [8]

$$t_L = \frac{1}{2\pi f_L} \tag{4.18}$$

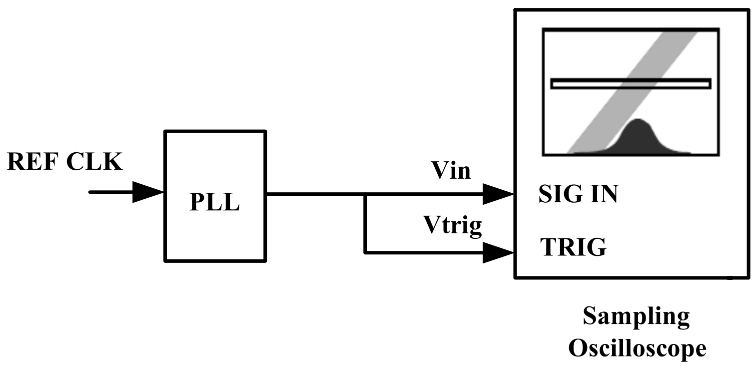


Figure 4.8 Measurement technique: Time domain, self referenced.

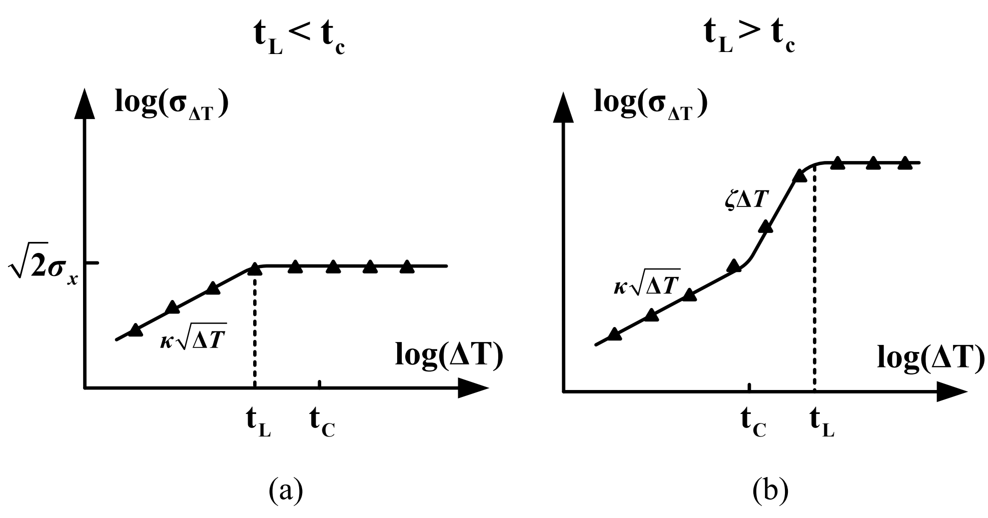


Figure 4.9 Measurement result: Time domain, self referenced.

4.4.1 Case I: White Noise Dominated ($t_L < t_c$)

In the case that the PLL loop bandwidth f_L is higher than the $1/f^3$ phase noise corner frequency f_c, the time domain PLL jitter versus measurement time delay ΔT is as shown in Figure 4.9a. The 1/f noise contribution is negligible as discussed in Section 4.2. Since the jitter process at the output of the closed loop VCO is stationary, the jitter as a function of delay in the self reference time domain measurement can be calculated by substituting the closed loop phase noise p.s.d. (4.5) into (3.21)

$$\sigma_W^2\left(\Delta T\right) = \frac{2}{\pi^2 f_0^2}\int_0^\infty \frac{N_1 / f_L^2}{1+\left(f/f_L\right)^2}\sin^2\left(\pi\Delta T f\right)df \qquad [\sec^2] \tag{4.19}$$

$$\sigma_W^2\left(\Delta T\right) = \frac{N_1}{2\pi f_L f_0^2}\left(1-e^{-2\pi f_L \Delta T}\right) = 2\left(\sigma_x^2\right)_W\left(1-e^{-2\pi f_L \Delta T}\right) \quad [\sec^2] \tag{4.20}$$

Result of (4.20) agrees the result in [8].

From (4.20), the rms jitter of a white noise dominated PLL is upper bounded by

$$\left(\sigma_W\right)_{\max} = \sqrt{2}\left(\sigma_x\right)_W \tag{4.21}$$

4.4.2 Case II: In the Presence of Non-negligible 1/f Noise ($t_L > t_c$)

In the case that the PLL loop bandwidth f_L is lower than the $1/f^3$ phase noise corner frequency f_c, the time domain PLL jitter versus measurement time delay ΔT is as shown in Figure 4.9b. Since the random processes of jitter due to white and 1/f noise are independent, the variance of the jitter process is

$$\sigma_{Total}^2\left(\Delta T\right) = \sigma_{1/f}^2\left(\Delta T\right) + \sigma_W^2\left(\Delta T\right) \quad [\sec^2] \tag{4.22}$$

For the jitter process due to 1/f noise, applying (3.21)

$$\sigma_{1/f}^2\left(\Delta T\right) = \frac{2}{\pi^2 f_0^2}\int_0^\infty \frac{N_1 f_c / f_L^{\,3}}{1+(f/f_L)^3}\sin^2\left(\pi\Delta T f\right)df \quad [\sec^2] \tag{4.23}$$

Since

$$\sin^2\left(x\right) = \frac{1-\cos\left(2x\right)}{2} \tag{4.24}$$

equation (4.23) is computed as

$$\sigma_{1/f}^2\left(\Delta T\right) = \frac{N_1 f_c}{\pi^2 f_0^2 f_L^{\,2}}\left(\int_0^\infty \frac{1}{1+(f/f_L)^3}d\left(\frac{f}{f_L}\right) + \int_0^\infty \frac{\cos\left(2\pi\Delta T f_L \dfrac{f}{f_L}\right)}{1+(f/f_L)^3}d\left(\frac{f}{f_L}\right)\right) \tag{4.25}$$

which equals

$$\sigma_{1/f}^2\left(\Delta T\right) = \frac{N_1 f_c}{\pi^2 f_0^2 f_L^{\,2}}\left(\frac{2}{3\sqrt{3}}\pi - \int_0^\infty \frac{\cos\left(\dfrac{\Delta T}{t_L}x\right)}{1+x^3}dx\right) \tag{4.26}$$

or

$$\sigma_{1/f}\left(\Delta T\right) = \sqrt{\frac{2}{3\sqrt{3}\pi}}\frac{\sqrt{N_1 f_c}}{f_0 f_L}\sqrt{1-\int_0^\infty \frac{3\sqrt{3}\cos\left(\dfrac{\Delta T}{t_L}x\right)}{2\pi\left(1+x^3\right)}dx} \tag{4.27}$$

where

$$x = \frac{f}{f_L} \tag{4.28}$$

It is too complicated to compute the integral in equation (4.27). As illustrated in Figure 4.10, the numerical evaluation of this integral indicates that

$$\int_0^{\infty} \frac{3\sqrt{3}\cos\left(\frac{\Delta T}{t_L}x\right)}{2\pi\left(1+x^3\right)}dx \to 0 \quad \text{when} \quad \Delta T > 4t_L \tag{4.29}$$

So for time delays larger than t_L, the closed loop rms jitter due to the 1/f noise is upper bounded by

$$\left(\sigma_{1/f}\right)_{\max} = \sqrt{\frac{2}{3\sqrt{3}\pi}}\frac{\sqrt{N_1 f_c}}{f_0 f_L} = \sqrt{2}\frac{\kappa}{f_L}\sqrt{\frac{f_c}{3\sqrt{3}\pi}} = \sqrt{2}\left(\sigma_x\right)_{1/f} \tag{4.30}$$

The total variance of the PLL jitter process is the summation of (4.20) and (4.27). Therefore the upper bound for the total self-referenced jitter is

$$\left(\sigma_{Total}\right)_{\max} = \sqrt{2\left(\sigma_x^2\right)_{1/f} + 2\left(\sigma_x^2\right)_W} = \sqrt{2}\left(\sigma_x\right)_{Total} \quad \text{[sec rms]} \tag{4.31}$$

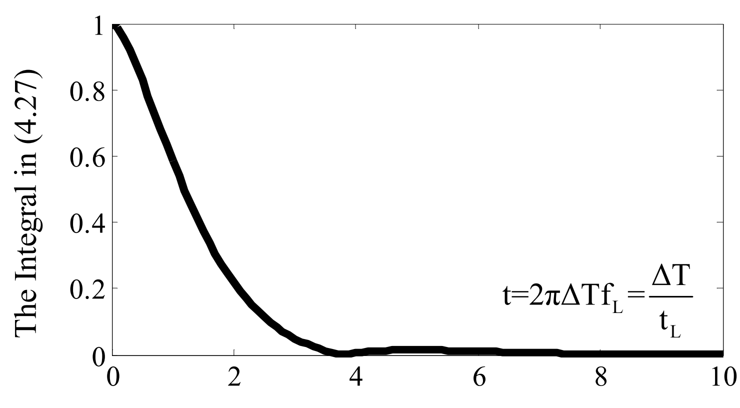


Figure 4.10 Numerical evaluation of the integral in (4.27).

Chapter 5: Jitter and the Geometry of Ring Oscillators

5.1 Introduction

While the LC oscillators are able to offer clock signals with fastest speed possible and excellent jitter performance, they also require the use of integrated high Q inductors and capacitors, both of which consume large amounts of die area. This results in the ring oscillator being more appealing for applications requiring high speed but moderate jitter. Even though their jitter performance is not as good as LC oscillators, ring oscillators have the advantages of simpler circuit design, easier integration, and less die area. Therefore ring oscillators have been widely used in applications such as clock recovery for serial data communications [1], [65]-[67], multiphase clock generation [68]-[70], and frequency synthesizers [71]-[73].

Phase and frequency fluctuations in LC oscillators have been the subject of numerous studies [74]–[77] since the 1960's. In 1990's, References [7]-[9] started the jitter analysis and modeling for bipolar and CMOS differential ring oscillators in the time domain, both of which are based on the linear time invariant (LTI) system assumption, and concentrated on the white noise upconverted jitter. [10] and [15] proposed a general theory for phase noise in oscillators, which is a frequency domain approach based on a linear time variant (LTV) model, and successfully get around the mathematical difficulties to derive accurate analytical expressions for the oscillator output waveform.

Historically, the differential structure has been the more popular approach for implementing ring oscillators. Single-ended ring oscillators are receiving more attention recently since they can achieve better jitter performance compared to differential ring oscillators due to the larger voltage swings [15]. Even though the single-ended configuration is more susceptible to common-mode noise such as noise from power supply and substrate, an interpolating network [78] can be used to convert the single-ended output to differential signal to overcome this drawback.

The design of ring oscillators involves many tradeoffs in terms of speed, power, and area. This chapter will follow the work in [7]-[9], [79] and [80] to evaluate the jitter performance in terms of the geometry for CMOS ring oscillators in time domain with LTI modeling, since the LTI model is easier to manipulate mathematically as an extension of the theory in [8]. The analysis will focus on white noise integrated jitter. The oscillation waveform symmetry criteria required to minimize the the $1/f^3$ phase noise corner f_c is assumed to have been met. The 1/f noise contribution can be negligible if the designer has the freedom to increase the loop bandwidth to cover the $1/f^3$ phase noise corner as discussed in Section 4.3.2. And the system jitter performance is thus limited by the integrated white noise.

5.2 VCO Design in Time Domain with κ

Ring oscillators are usually realized by placing an odd number of inverters in a feedback loop, as illustrated in Figure 5.1 [81]. They can also be implemented by an even number of differential stages, with simply interchanging the outputs of the last inverter before feeding them back to the input.

The waveforms obtained at the outputs of the three inverters are also shown in Figure 5.1 [81]. It is obvious that the signal must go through each of the delay stages twice to provide one period of oscillation. Since each stage provides a delay of T_d, the oscillation frequency of an N-stage ring oscillator is [81]

$$f_0 = \frac{1}{2NT_d} \tag{5.1}$$

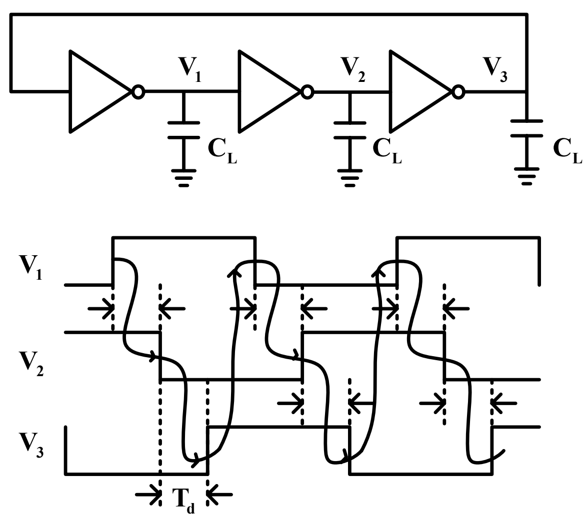


Figure 5.1 A three-stage ring oscillator and resulting waveform.

Since the transition of each stage is triggered by the previous stage, at a single time only one stage in the ring is switching and thus contributing jitter. Therefore the white noise figure of merit κ is independent of the number of stages in the ring [7]. This concept has been verified by the measured data from five differential bipolar ring oscillators with different number of identical delay stages [8]. Therefore κ is actually a property of the delay stage and the jitter analysis for a single delay stage is enough to estimate the κ of the VCO. This simplifies the PLL system design, simulation, and testing, because only the open loop VCO needs to be considered. The closed-loop jitter performance can be predicted with the technique developed in Section 4.3.

5.3 κ of the CMOS Inverter

As illustrated in Figure 5.2, the CMOS inverter with equal-length NMOS and PMOS transistors is the simplest implementation of the delay stage. C_L is the total capacitance at the output node. Since the CMOS inverter offers the fewest number of noise sources and rail-to-rail output swing, it has the minimum κ among the practical implementation of delay stage for ring oscillators. Therefore the jitter analysis for the CMOS inverter is the best way to characterize the capacity of jitter optimization for different semiconductor processes.

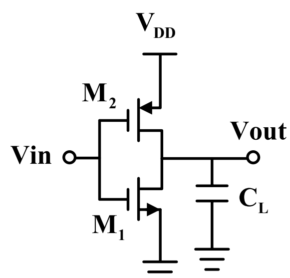


Figure 5.2 The CMOS inverter.

During the analysis, the following assumptions are made for simplification:

1. The conducting MOSFET is considered to be always in saturation with the input of an ideal step function switching between V_{DD} and ground;
2. The loading MOSFET is always off since the input is an ideal step;
3. The gate capacitance of the next stage dominates the load capacitance C_L;
4. The threshold voltages for the NMOS and PMOS transistors are equal;
5. The NMOS and PMOS transistors are sized to have identical first-order transconductance:

$$W_p = \frac{\mu_n}{\mu_p} W_n = m W_n \tag{5.2}$$

Therefore, all the parameters in the following sections, such as W, L, and μ, are parameters of the NMOS transistor M_1 if not specified.

5.3.1 Propagation Delay of the CMOS Inverter

Much effort has been devoted to the extraction of accurate model for the inverter delay [82]-[84]. Since the simple circuit of ring oscillators bundle so many nonlinear effects, all of these results are in the form of complicated analytical equations. For simplicity and giving the designers more insight, a first order model will be derived for the CMOS inverter by approximating the output waveform in transition as a linearly rising or falling ramp as illustrated in Figure 5.3.

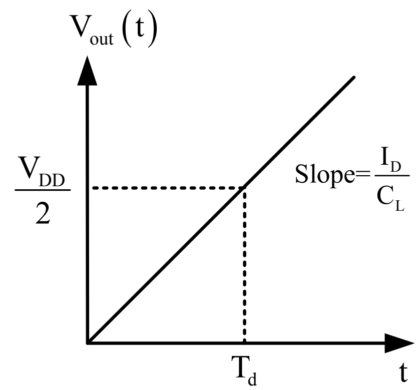


Figure 5.3 Modeling of propagation delay for the CMOS inverter.

When the input to the CMOS inverter is an ideal step function switching from V_{DD} to ground and the PMOS transistor M_2 is always in the saturation region, the DC current charging the load capacitor, I_D, is constant in switching, and the voltage at the output node increases linearly as illustrated in Figure 5.3 with the slope of

$$slope = \frac{I_D}{C_L} \tag{5.3}$$

Since the inverter threshold is $V_{DD}/2$, the ideal inverter propagation delay T_d is

$$T_d = \frac{V_{DD} C_L}{2 I_D} \tag{5.4}$$

The load capacitance C_L is given by [40]

$$C_L = \frac{5}{2} C_{ox} WL\left(1+m\right) \tag{5.5}$$

where W and L are the channel width and length of the NMOS transistor M_1, and m is the mobility ratio of n- and p-type carriers defined in (5.2).

Substituting the expression of load capacitance C_L of (5.5) and the drain current model of (2.11) into (5.4), the inverter propagation delay is

$$T_d = \frac{5C_{ox}WL\left(1+m\right)V_{DD}}{4I_D} \tag{5.6}$$

or

$$T_d = \frac{5\left(1+m\right)V_{DD}}{4v_{sat}\left(V_{DD}-V_T\right)} \cdot L\left(1+\frac{L}{L_c}\right) \tag{5.7}$$

For the case of long channel, (5.7) simplifies to

$$T_d = \frac{5\left(1+m\right)V_{DD}}{2\mu_{eff}\left(V_{DD}-V_T\right)^2} \cdot L^2 \quad \text{when } L \gg L_c \tag{5.8}$$

For the case of short channel, (5.7) simplifies to

$$T_d = \frac{5\left(1+m\right)V_{DD}}{4v_{sat}\left(V_{DD}-V_T\right)} \cdot L \quad \text{when } L \ll L_c \tag{5.9}$$

(5.8) and (5.9) indicate the relationship between the speed and geometry for the CMOS inverter as follows:

1. The inverter propagation delay has very weak dependency on the channel width W. From the first order analysis, a wider channel width W will increase the drain current while the gate capacitance will increase by the same ratio. The slope of the switching transition, which is the ratio between the charging current and the load capacitance, will remain the same. Therefore, as long as the voltage swing is

unchanged, the propagation delay will keep constant. If taking account to second order effects, the inverter delay will show a very weak dependency on W. But increasing W usually does not affect the inverter delay much.

2. The inverter propagation delay is proportional to L^2 for the case of long channel since a shorter channel length L will not only increase the drain current, but also reduce the gate capacitance.

3. The inverter propagation delay is proportional to L for the case of short channel. The drop in the power is due to the velocity saturation of the carriers and no dependency of drain current on L. The decrease of the propagation delay with a shorter L is due to less load capacitance.

The speed of ring oscillators formed by CMOS inverters is computed by (5.1)

$$T_0 = 2NT_d = \frac{5N(1+m)V_{DD}}{2v_{sat}(V_{DD}-V_T)} \cdot L\left(1+\frac{L}{L_c}\right) \tag{5.10}$$

$$f_0 = \frac{2v_{sat}(V_{DD}-V_T)}{5N(1+m)V_{DD}} \cdot \frac{1}{L\left(1+\frac{L}{L_c}\right)} \tag{5.11}$$

Therefore, the dependency of the oscillation frequency on the channel length L is

$$f_0 \propto \begin{cases} \frac{1}{L^2} & \text{when } L \gg L_c \\ \frac{1}{L} & \text{when } L \ll L_c \end{cases} \tag{5.12}$$

5.3.2 Drain Thermal Noise in the Switching MOSFET

The thermal noise current density i_n in Figure 5.4 can be viewed as series of current pulses of duration T centered on the DC drain current of the switching MOSFET, when the pulse width T is chosen such that 1/T is much greater than the highest frequency of interest in the circuit [8]. The amplitude of the pulses are independent, identically distributed Gaussian random process with standard deviation σ_i as [8]

$$\sigma_i = \frac{1}{\sqrt{2T}} \cdot \frac{i_n}{\sqrt{\Delta f}} \quad [\text{A rms}] \tag{5.13}$$

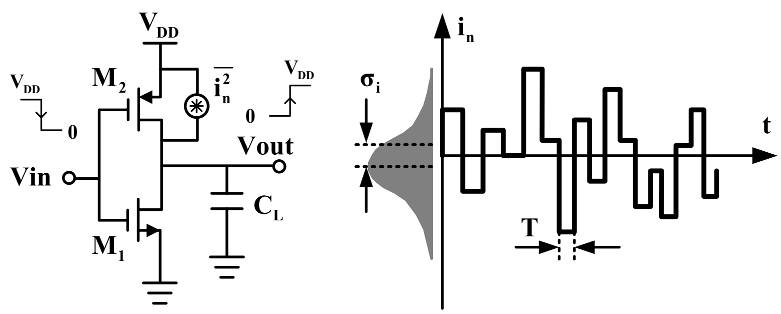


Figure 5.4 Drain thermal noise of the switching MOSFET.

In the presence of the drain thermal noise, there will be variation of the time to reach the next inverter threshold as illustrated in Figure 5.5. The standard deviation of this varying time distribution, σ_t, is the accumulated jitter over the inverter propagation delay T_d.

Using (5.13), the rms noise current is

$$\sigma_{ind} = \frac{1}{\sqrt{2 \cdot dt}} \cdot \frac{i_n}{\sqrt{\Delta f}} \quad [\text{A rms}] \tag{5.14}$$

This noise current will charge the load capacitor C_L. The standard deviation of the charge due to this drain thermal noise current is

$$\sigma_q\left(dt\right)=\sigma_{ind}\cdot dt=\frac{i_n}{\sqrt{\Delta f}}\cdot\sqrt{\frac{dt}{2}}\quad\left[\text{C rms}\right] \tag{5.15}$$

The variance $\sigma^2_q(dt)$ is

$$\sigma_q^2\left(dt\right)=\frac{1}{2}\cdot\frac{i_n^2}{\Delta f}\cdot dt\quad\left[\text{C}^2\text{ rms}\right] \tag{5.16}$$

Integrating the charge variance of (5.16) from the time that the inverter begins to switch until the ideal output voltage reaches the next inverter threshold will give the total charge variance in the output transition:

$$\sigma_q^2\left(tot\right)=\int_0^{T_d}\frac{1}{2}\cdot\frac{i_n^2}{\Delta f}\cdot dt\quad\left[\text{C}^2\text{ rms}\right] \tag{5.17}$$

$$\sigma_q^2\left(tot\right)=\frac{1}{2}\cdot\frac{i_n^2}{\Delta f}\cdot T_d\quad\left[\text{C}^2\text{ rms}\right] \tag{5.18}$$

The voltage change due to this total charge is

$$\sigma_v=\frac{\sigma_q\left(tot\right)}{C_L}=\frac{1}{C_L}\cdot\frac{i_n}{\sqrt{\Delta f}}\cdot\sqrt{\frac{T_d}{2}}\quad\left[\text{V rms}\right] \tag{5.19}$$

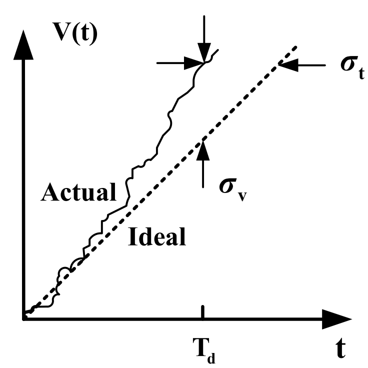


Figure 5.5 Ideal output and actual output due to the drain thermal noise.

From Figure 5.5, the standard deviation in time can be calculated by

$$\sigma_t = \frac{\sigma_v}{slope} \tag{5.20}$$

Substituting (5.3) into (5.20), the standard deviation in time is

$$\sigma_t = \frac{1}{I_D} \cdot \frac{i_n}{\sqrt{\Delta f}} \cdot \sqrt{\frac{T_d}{2}} \quad [\text{sec rms}] \tag{5.21}$$

The rms jitter of (5.21) is the jitter accumulated in time delay T_d. Therefore the white noise figure of merit κ_{sw} due to the switching MOSFET is as [79]

$$\kappa_{sw} = \frac{\sigma_t}{\sqrt{T_d}} = \frac{i_n}{\sqrt{\Delta f}} \frac{1}{\sqrt{2} I_D} \quad \left[\sqrt{\text{sec}}\right] \tag{5.22}$$

Substituting (2.11) and (2.22) into (5.22), the figure of merit κ_{sw} is

$$\kappa_{sw} = \left(1 + \frac{L_c}{L}\right) \sqrt{4kT\gamma_s \cdot \frac{1}{\frac{1}{2}\mu_{eff} C_{ox} \frac{W}{L}\left(V_{DD} - V_T\right)^3}} \quad \left[\sqrt{\text{sec}}\right] \tag{5.23}$$

or

$$\kappa_{sw} = \frac{\sqrt{2kT\gamma_s \cdot \mu_{eff} C_{ox} \frac{W}{L}\left(V_{DD} - V_T\right)}}{I_D} \quad \left[\sqrt{\text{sec}}\right] \tag{5.24}$$

5.3.3 KTC Noise due to the Loading Transistor

With the assumption of the ideal step input, the loading transistor will be always off during switching. But this does not mean that the noise contributed by the loading transistor should be ignored.

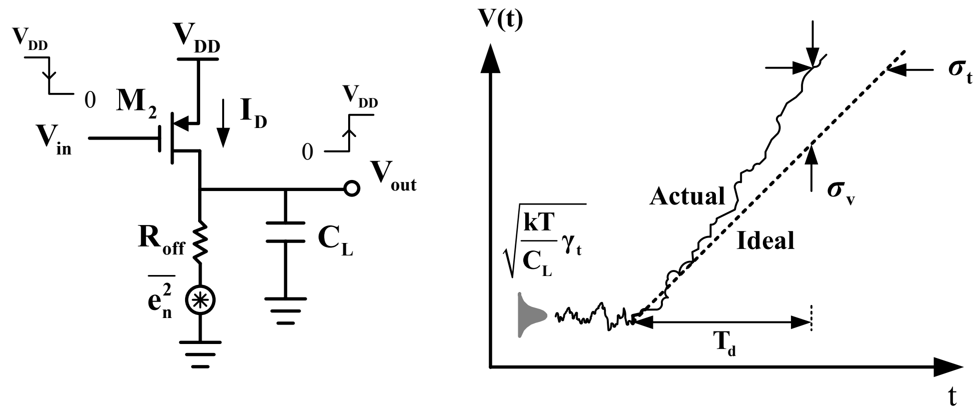


Figure 5.6 KTC noise due to loading transistor.

As illustrated in Figure 5.6, the loading transistor is modeled by its off channel resistance R_{off}, and the thermal noise density for this resistor is

$$\frac{e_n}{\Delta f} = \sqrt{4kT\gamma_t R_{off}} \tag{5.25}$$

The noise bandwidth of this system is determined by the pole at the output:

$$\Delta f = \frac{\pi}{2} \cdot \frac{1}{2\pi R_{off} C_L} \tag{5.26}$$

Therefore, this noise source manifests itself as KTC noise with rms voltage of

$$\sigma_v = \sqrt{\frac{kT}{C_L}\gamma_t} \quad [\text{V rms}] \tag{5.27}$$

This noise source affects the initial condition of the switching output voltage as illustrated in Figure 5.6, thereby varying the effective gate delay. The standard deviation of this varying time distribution σ_t, is

$$\sigma_t = \frac{\sigma_v}{slope} = \sqrt{\frac{kTC_L\gamma_t}{I_D^2}} \quad [\text{sec rms}] \tag{5.28}$$

σ_t is the accumulated jitter over the inverter propagation delay T_d. Therefore the noise figure of merit κ_{load} due to the loading MOSFET is

$$\kappa_{load} = \frac{\sigma_t}{\sqrt{T_d}} = \sqrt{\frac{2kT\gamma_t}{I_D V_{DD}}} \quad \left[\sqrt{\text{sec}}\right] \tag{5.29}$$

or

$$\kappa_{load} = \sqrt{\frac{4kT\gamma_t}{\mu_{eff} C_{ox} \frac{W}{L}\left(V_{DD} - V_T\right)^2 V_{DD}}\left(1 + \frac{L_c}{L}\right)} \quad \left[\sqrt{\text{sec}}\right] \tag{5.30}$$

5.3.4 κ_{total} of the CMOS Inverter

Since noise sources in the switching and loading transistors are independent, the total κ for the CMOS inverter is

$$\kappa_{total} = \sqrt{\kappa_{sw}^2 + \kappa_{load}^2} \tag{5.31}$$

The ratio between κ_{sw} and κ_{load} is computed as

$$\frac{\kappa_{sw}}{\kappa_{load}} = \sqrt{\left(1+\frac{L_c}{L}\right)\frac{2\gamma_s}{\gamma_t}\cdot\frac{V_{DD}}{\left(V_{DD}-V_T\right)}} = \sqrt{\frac{1}{\delta}\left(1+\frac{L_c}{L}\right)} \tag{5.32}$$

where δ is the coefficient defined by

$$\delta = \frac{\gamma_t\left(V_{DD}-V_T\right)}{2\gamma_s V_{DD}} \tag{5.33}$$

Therefore, κ_{total} can be expressed as

$$\kappa_{total} = \kappa_{sw}\left(1+\sqrt{\frac{\delta}{\left(1+\frac{L_c}{L}\right)}}\right) \tag{5.34}$$

Substituting (5.23) or (5.24) into (5.34), κ_{total} is

$$\kappa_{total} = \left(1+\frac{L_c}{L}\right)\sqrt{\frac{8kT\gamma_s}{\mu_{eff}C_{ox}\frac{W}{L}\left(V_{DD}-V_T\right)^3}}\cdot\sqrt{1+\frac{\delta}{\left(1+\frac{L_c}{L}\right)}} \tag{5.35}$$

or

$$\kappa_{total} = \frac{1}{I_D}\sqrt{2kT\gamma_s\cdot\mu_{eff}C_{ox}\frac{W}{L}\left(V_{DD}-V_T\right)}\cdot\sqrt{1+\frac{\delta}{\left(1+\frac{L_c}{L}\right)}} \tag{5.36}$$

5.4 Jitter and VCO Geometry

5.4.1 κ and VCO Geometry

(i) CMOS inverters with long-channel MOSFETs

For CMOS inverters with long-channel MOSFETs, γ_s=2/3, and L>>L_c. Under these conditions, κ_{sw} in (5.23) simplifies to

$$\kappa_{sw} = \sqrt{\frac{8}{3}kT \cdot \frac{1}{\frac{1}{2}\mu_{eff}C_{ox}\frac{W}{L}\left(V_{DD}-V_T\right)^3}} \quad \left[\sqrt{\sec}\right] \tag{5.37}$$

and κ_{load} in (5.30) simplifies to

$$\kappa_{load} = \sqrt{\frac{2kT\gamma_t}{\frac{1}{2}\mu_{eff}C_{ox}\frac{W}{L}\left(V_{DD}-V_T\right)^2 V_{DD}}} \quad \left[\sqrt{\sec}\right] \tag{5.38}$$

The ratio between κ_{sw} and κ_{load} is

$$\frac{\kappa_{sw}}{\kappa_{load}} = \sqrt{\frac{4V_{DD}}{3\gamma_t\left(V_{DD}-V_T\right)}} \tag{5.39}$$

Since the value of γ_t is between 2/3 and 1 for long-channel MOSFETs, and the threshold voltage V_T is usually less than half of the power supply V_{DD}, the ratio between κ_{sw} and κ_{load} is in the range of

$$\sqrt{\frac{4}{3}} < \frac{\kappa_{sw}}{\kappa_{load}} < 2 \tag{5.40}$$

Result of (5.40) indicates that the drain thermal noise of the switching MOSFET is always larger than the KTC noise of the loading MOSFET, but not much in the long-channel case.

Substituting (5.37) and (5.38) into (5.31), the expression for κ_{total} in (5.35) will reduce to

$$\kappa_{total} = \sqrt{\frac{2kT}{\frac{1}{2}\mu_{eff} C_{ox} \frac{W}{L}\left(V_{DD} - V_T\right)^2}} \sqrt{\frac{4}{3\left(V_{DD} - V_T\right)} + \frac{\gamma_t}{V_{DD}}} \quad \left[\sqrt{\sec}\right] \tag{5.41}$$

or

$$\kappa_{total} = \sqrt{\frac{2kT}{I_D V_{DD}}} \sqrt{\frac{4V_{DD}}{3\left(V_{DD} - V_T\right)} + \gamma_t} \quad \left[\sqrt{\sec}\right] \tag{5.42}$$

From the results of (5.41) and (5.42), the following conclusions can be drawn for CMOS inverters with long-channel MOSFETs:

1. κ_{total} is inversely proportional to the square root of the power dissipation.

$$\kappa_{total} \propto \frac{1}{\sqrt{P}} \quad \text{when } L \gg L_c \tag{5.43}$$

This agrees the results in [15].

2. κ_{total} is inversely proportional to the square root of the channel width W.

$$\kappa_{total} \propto \frac{1}{\sqrt{W}} \quad \text{when } L \gg L_c \tag{5.44}$$

The current of MOSFETs is proportional to the channel width. So increasing the channel width will increase power dissipated on the oscillation waveform and thus reduce jitter according to (5.43).

3. κ_{total} is proportional to the square root of the channel length L.

$$\kappa_{total} \propto \sqrt{L} \quad \text{when } L \gg L_c \tag{5.45}$$

The reason is that using a shorter length will increase the current of long-channel MOSFETs and thus more power is dissipated on the oscillation waveform to improve the VCO jitter performance.

(ii) CMOS inverters formed by MOSFETs with extremely short channels

For MOSFETs with extremely short channels, L<<L_c. κ_{sw} in equation (5.23) will reduce to

$$\kappa_{sw} = \sqrt{\frac{4kT\gamma_s}{E_c L v_{sat} C_{ox} W \left(V_{DD} - V_T\right)}} = \sqrt{\frac{4kT\gamma_s}{E_c L \cdot I_D}} \quad \left[\sqrt{\sec}\right] \tag{5.46}$$

And κ_{load} in (5.30) simplifies to

$$\kappa_{load} = \sqrt{\frac{2kT\gamma_t}{v_{sat} C_{ox} W \left(V_{DD} - V_T\right) V_{DD}}} = \sqrt{\frac{2kT\gamma_t}{I_D V_{DD}}} \quad \left[\sqrt{\sec}\right] \tag{5.47}$$

The ratio between κ_{sw} and κ_{load} is

$$\frac{\kappa_{sw}}{\kappa_{load}} = \sqrt{\frac{2\gamma_s}{\gamma_t} \cdot \frac{V_{DD}}{E_c L}} \tag{5.48}$$

Since

$$L \gg L_c \quad \text{or} \quad E_c L \gg V_{DD} - V_T \tag{5.49}$$

The ratio between κ_{sw} and κ_{load} is much larger than 1.

$$\frac{\kappa_{sw}}{\kappa_{load}} \gg 1 \tag{5.50}$$

Therefore when the channel length is extremely short, the drain thermal noise of the switching MOSFET is much larger than the KTC noise of the loading MOSFET, and the KTC noise is negligible. κ_{total} of the inverter is dominated by κ_{sw} of (5.46), and is

$$\kappa_{total} = \sqrt{\frac{4kT\gamma_s}{E_c L v_{sat} C_{ox} W \left(V_{DD} - V_T\right)}} = \sqrt{\frac{4kT\gamma_s}{E_c L \cdot I_D}} \quad \left[\sqrt{\sec}\right] \tag{5.51}$$

From the result of (5.51), the following conclusions can be draw for CMOS inverters formed by MOSFETs with extremely short channels:

1. κ_{total} is inversely proportional to the square root of the drain current I_D, thus is still inversely proportional to the square root of the power dissipation since the power supply V_{DD} is usually fixed in the VCO design.

$$\kappa_{total} \propto \frac{1}{\sqrt{P}} \quad \text{when} \ \ L \ll L_c \tag{5.52}$$

2. κ_{total} is still inversely proportional to the square root of the channel width W.

$$\kappa_{total} \propto \frac{1}{\sqrt{W}} \quad \text{when} \ \ L \ll L_c \tag{5.53}$$

The reason is that from the current model (2.11) the drain current I_D is still proportional to the channel width W for short-channel MOSFETs.

3. κ_{total} is inversely proportional to the square root of the channel length L.

$$\kappa_{total} \propto \frac{1}{\sqrt{L}} \quad \text{when} \ \ L \ll L_c \tag{5.54}$$

When the carriers' velocity is totally saturated, using shorter channel length will not increase the drain current of MOSFETs from (2.12). It will not add more power to the oscillation waveform either. However the channel noise power will increase at a shorter channel length due to degraded mobility and hot-electron effect according to (2.22). So the total jitter on the output clock will increase with a shorter channel length.

Usually the designer does not have the freedom to increase the power supply V_{DD} in the deep-submicron process. To achieve the same κ while using a shorter channel length, the only way is to increase the channel width in the same ratio, and thus increasing the power dissipated in the oscillation waveform to compensate the increased noise power.

Equation (5.51) shows that

$$\kappa_{total} \propto \frac{1}{\sqrt{WL}} \quad \text{when} \ \ L \ll L_c \tag{5.55}$$

Therefore to achieve the same κ, ring oscillators in the same deep submicron process will consume the same gate area. However, the die area consumption is different. Figure 5.7 shows the layout for a MOSFET under the MOSIS scalable CMOS (SCMOS) design rules [85]. The λ is half of feature size of the semiconductor process. The minimum channel length L_{min} is usually of 2λ. From the SCMOS design rules, the size of contacts to connect the active region and metal layers must be exactly of 2λ by 2λ, the minimum space between contacts and the gate poly is 2λ, and the minimum space between the contacts and the edge of the active is λ. Therefore, the minimum length for the drain or source area is 5λ, and the minimum area for this one-finger MOSFET is

$$\left(A_{MOS}\right)_{\min} = W\left(L+10\lambda\right) \tag{5.56}$$

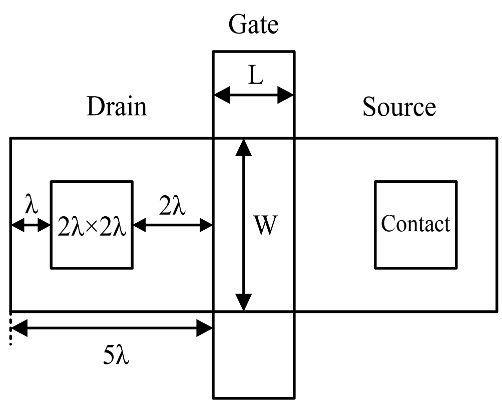


Figure 5.7 The MOSFET layout under SCMOS design rules.

In the design of ring oscillators, due to the requirement of the speed, the designer usually does not have too much room to play with the channel length according to (5.12). And the channel length is usually less than $5L_{min}$, or 10λ. In this case, the transistor area is not a strong function of L, and decreasing channel length will not save the consumed area much. Therefore the transistor area is approximately decided by the channel width.

Again from (5.55), to achieve the same κ, ring oscillators in the same deep submicron process will consume the same gate area. But the ring oscillator with a shorter channel length L and a wider channel width W will consume not only more power, but also more die area. If assuming that the die area consumed by the VCO is dominated by the transistor area, the relationship between the VCO die area and geometry can be approximated by

$$A_{VCO} \propto \lambda W \tag{5.57}$$

With the feature size of the semiconductor processes scaling down aggressively for higher transistor density and faster speed, the noise performance of transistors usually gets worse as analyzed in Section 2.3, while the power supply always gets lower in order to maintain safe and reliable device operations. Therefore, implementing ring oscillators in a process with smaller feature size usually requires larger channel width not only to compensate the increased noise but also the lowered V_{DD}.

The above discussion can be extended to generic analog circuit design in deep submicron processes. When the velocity of the carriers is saturated, using shorter channel length will not add more power to the output signal, while the noise power increases. This will result in the drop of the SNR. There are two options to achieve the previous SNR when using a longer channel length. The first one is to increase the power in the output signal, which is same as the analysis above for ring oscillators; the second one is to use other techniques to limit the noise power, such as using larger capacitance to limit the KTC noise in the OPAMP design. But both methods require more die area. This indicates that while the digital circuits are enjoying the benefits from the smaller feature size such as faster speed and higher circuit density, the analog circuits will suffer the increased noise, which will be a big challenge for analog design.

(iii) Optimum channel length for jitter optimization

From (5.45) and (5.54), κ_{total} is proportional to the square root of the channel length L in the long-channel case while is inversely proportional to the square root of L in the short-channel case. Therefore, an optimum channel length must exist to minimize κ_{total}. This optimum length can be obtained by differentiating κ_{total} of (5.35) with respect to L, and solving the equation of

$$\frac{d\kappa_{total}}{dL} = 0 \tag{5.58}$$

Performing a first-order analysis and not considering the effect of increased excess noise factor γ_s with a shorter length, the optimum channel length L is obtained as

$$L_{optimum} = L_c\sqrt{\frac{1}{1+\delta}} \tag{5.59}$$

However, the increasing of γ_s and second order effects may result in a optimum length larger than that predicted by (5.59).

As discussed in Section 2.3, γ_s of short-channel MOSFETs is about two or three times larger than that of long-channel MOSFETs, which will cause a larger κ at shorter L. From the measurement results in [26] for MOSFETs in a 0.18μm process, as L scales down from 0.5μm to 0.18μm, γ_s increased from 0.72 to 1.3 at the DC bias of V_{GS}=0.6V and V_{DS}=1.8V. At the DC bias of V_{GS}=1.8V and V_{DS}=1.8V, γ_s increased from 0.67 to 0.83.

The threshold voltage V_T is a complicate function of L as illustrated in Figure 5.8. For long-channel MOSFETs, V_T is almost of a constant value V_{T0}. As L scales down, a roll-up region will be observed followed by a roll-off region. The roll-up of V_T is due to the reverse short channel effect (RSCE) [86], [87], which is caused by the boron dopant pile-up phenomenon at the edge of the source and drain regions. The roll-off of V_T is due to the short channel effect (SCE) and drain-induced barrier lowering effect (DIBL) [86]-[88].

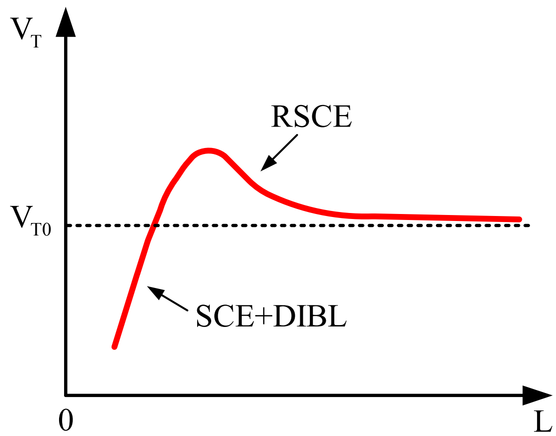


Figure 5.8 Threshold voltage versus channel length for MOSFETs.

From (5.35),

$$\kappa_{total} \propto \sqrt{\frac{1}{(V_{DD} - V_T)^3}} \tag{5.60}$$

So the roll-up of V_T will increase jitter while the roll-off of V_T will help to reduce jitter. The roll-off of V_T leads to several reliability issues such as a lack of pinchoff and hot-carrier effect at increasing drain voltage [89]. Research has been conducted to minimize the short channel effects and the V_T roll-off by using thin-body single material gate (MSG) silicon-on-insulator (SOI) MOSFETs, double material gate (DMG) SOI MOSFETs, and double gate (DG) SOI MOSFETs [89], [90]. Both [91] and [92] reported nanoscale MOSFETs with little V_T roll-off down to 0.05μm. For deep submicron processes with feature size of 0.25μm and below, V_{T0} is usually less than 0.5V. The actual roll-up and roll-down of V_T, if exists, usually has a very limited range.

Since L_c and μ_{eff} are functions of V_T, they are functions of L too. The variation of κ_{total} due to L_c and μ_{eff} is much less than that due to (5.60) since (5.60) is of higher order of V_{eff}.

The δ in (5.35) is a function of γ_s, γ_t, and V_T according to (5.33), and

$$\kappa_{total} \propto \sqrt{1+\frac{\delta}{\left(1+\frac{L_c}{L}\right)}} \tag{5.61}$$

The optimum channel length is around L_c according to (5.59). Since γ_s is usually larger than 1 at short lengths, δ is usually less than 0.5, and

$$\frac{\delta}{\left(1+\frac{L_c}{L}\right)} < \frac{1}{3} < 1 \tag{5.62}$$

Thus the variation of κ_{total} due to δ is not a big factor comparing to the impact due to γ_s directly in (5.35).

Summarizing the analysis above, the increasing of γ_s and the roll-up of V_T usually push the optimum length larger than that predicted by (5.59).

Table 5.1 lists the predicted and simulated κ_{total} for fifteen seven-stage ring oscillators in the IBM 0.13μm process. All the ring oscillators are implemented by the CMOS inverter in Figure 5.9 with different channel length L. The prediction is made by equation (5.35) and the simulation is performed by Cadence with Spectre simulator using the BSIM3 model [21]. For convenience, the data in Table 5.1 are also plotted in Figure 5.10.

To calculate the channel thermal noise using the compact model of (2.22), the excess noise parameter γ_s is determined by fitting the calculated noise density to the simulated noise density using the BSIM3v3 model for a single MOSFET which is biased with the assumed condition in analysis. The parameter γ_t is set to 1 for simplification.

From table 5.1 and Figure 5.10, κ_{total} is dominated by κ_{sw} as L scales down. The agreement of the predicted results and the simulation results is within 15%.

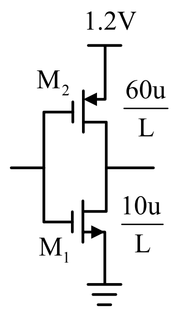
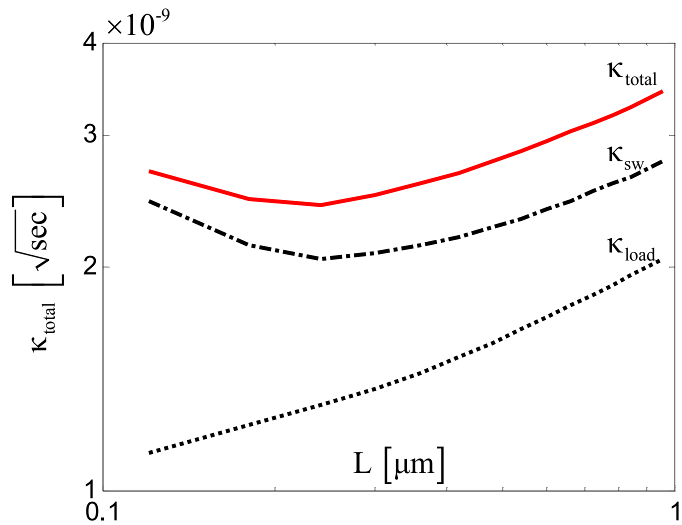


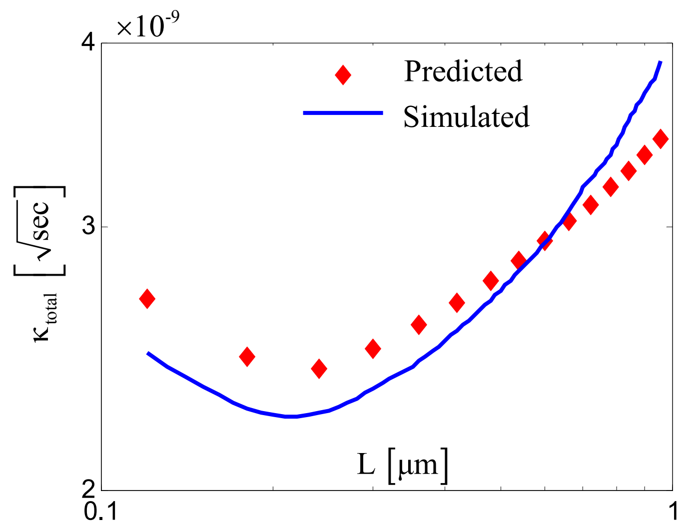
Figure 5.9 Inverter configuration for 7-stage ring oscillators.

Table 5.1 Predicted and simulated κ_{total} vs. L for 7-stage ring oscillators.

Index	L(μm)	L_{eff}(μm)	Predicted			Simulated
			κ_{sw}	κ_{load}	κ_{total}	κ_{total}
1	0.12	0.092	2.44E-09	1.12E-09	2.69E-09	2.47E-09
2	0.18	0.15	2.14E-09	1.22E-09	2.46E-09	2.27E-09
3	0.24	0.21	2.04E-09	1.30E-09	2.41E-09	2.25E-09
4	0.30	0.27	2.08E-09	1.37E-09	2.49E-09	2.34E-09
5	0.36	0.33	2.13E-09	1.44E-09	2.58E-09	2.44E-09
6	0.42	0.39	2.19E-09	1.51E-09	2.67E-09	2.56E-09
7	0.48	0.45	2.26E-09	1.58E-09	2.76E-09	2.68E-09
8	0.54	0.51	2.32E-09	1.65E-09	2.85E-09	2.81E-09
9	0.60	0.57	2.39E-09	1.71E-09	2.94E-09	2.93E-09
10	0.66	0.63	2.45E-09	1.77E-09	3.03E-09	3.08E-09
11	0.72	0.69	2.52E-09	1.83E-09	3.11E-09	3.24E-09
12	0.78	0.75	2.58E-09	1.89E-09	3.20E-09	3.36E-09
13	0.84	0.81	2.64E-09	1.95E-09	3.28E-09	3.54E-09
14	0.90	0.87	2.70E-09	2.00E-09	3.36E-09	3.70E-09
15	0.96	0.93	2.76E-09	2.05E-09	3.44E-09	3.88E-09



(a) Predicted κ_{sw}, κ_{load}, and κ_{total} for 7-stage ring oscillators



(b) Predicted and simulated κ_{total} vs. L for 7-stage ring oscillators

Figure 5.10 Plot of predicted and simulated κ for 7-stage ring oscillators.

The optimum L obtained from simulation is 0.24μm, which is two times of the minimum channel length 0.12μm, while the optimum L predicted by (5.59) is only 0.082μm. Figure 5.11 shows the input referred noise obtained from simulation which is plotted on a log-log scale. The maximum extracted γ_s from the noise simulation results is just 1. Figure 5.12 shows the simulated V_T as L scales down. Only the roll-up region is observed. The increased excess factor γ_s and the roll-up V_T of is the reason of the higher optimum channel length than predicted by (5.59).

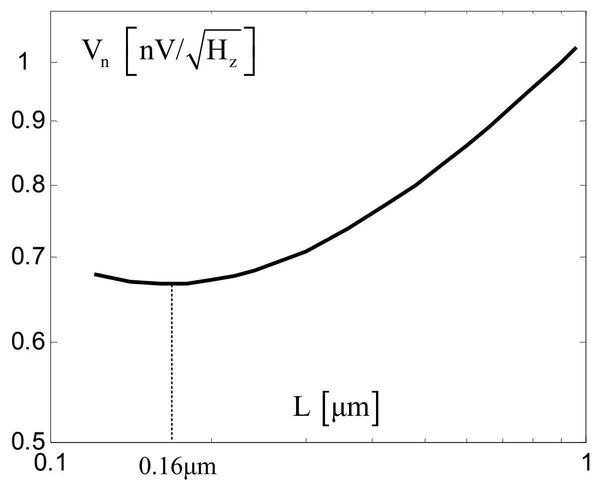


Figure 5.11 Simulated input referred noise versus channel length.

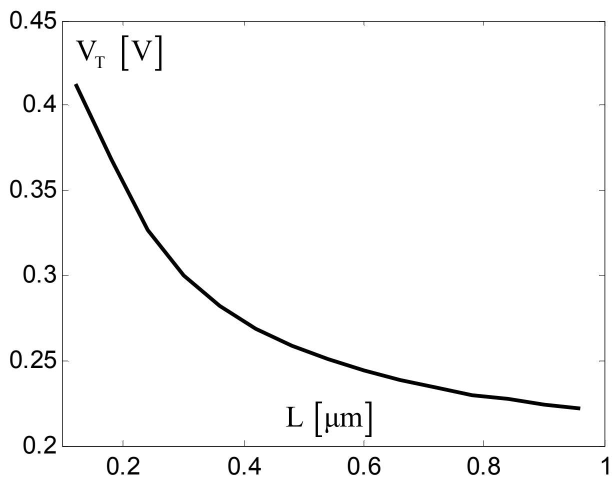


Figure 5.12 Simulated roll-up of threshold voltage.

5.4.2 Normalized RMS Jitter and VCO Geometry

For applications in data communications such as modulation and clock recovery, the local oscillator (LO) is locked to a reference by a PLL to provide synchronized clock for the following circuits. From the derivations in Section 4.3, due to the filtering of the loop, the PLL rms jitter with respect to the reference clock is

$$\sigma_x = \kappa \sqrt{\frac{1}{4\pi f_L}\left(1+\frac{4f_c}{3\sqrt{3}f_L}\right)} \quad [\text{sec rms}] \tag{5.63}$$

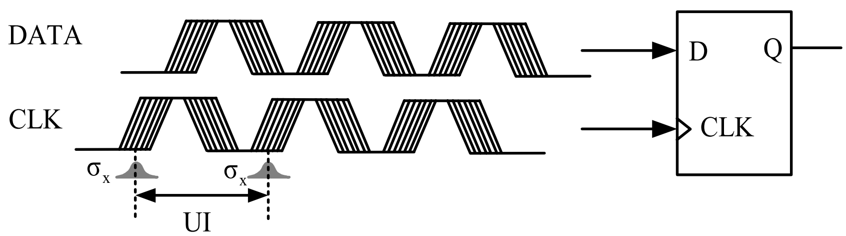


Figure 5.13 The normalized rms jitter.

From figure 5.11, more error will be observed at the output of the D flip-flop if the jitter-to-period ratio of the clock increases. In order to minimize the bit-error rate (BER), the normalized rms jitter, which is defined by the ratio between the PLL rms jitter and the clock period in the unit of UI (unit interval), should be minimized. From this definition, the normalized rms jitter is computed as

$$\left(\sigma_x\right)_{UI} = \frac{\sigma_x}{T_0} = \frac{\kappa}{T_0}\sqrt{\frac{1}{4\pi f_L}\left(1+\frac{4f_c}{3\sqrt{3}f_L}\right)} \quad [\text{UI}] \tag{5.64}$$

where T_0 is the period of the clock signal, f_L is the PLL loop bandwidth, and f_c is the $1/f^3$ phase noise corner.

From the discussions in chapter 4, the PLL loop bandwidth f_L should be as high as possible for jitter filtering, and the f_c is related to the symmetry properties of the oscillating waveform [10]. All the analysis from now on will assume that f_L and f_c are fixed. Thus the goal of this section is to minimize the κ-to-period ratio by carefully sizing the MOS transistors.

From the expression of κ_{total} in (5.31), the κ-to-period ratio can be separated into two parts,

$$\frac{\kappa}{T_0}=\sqrt{\left(\frac{\kappa_{sw}}{T_0}\right)^2+\left(\frac{\kappa_{load}}{T_0}\right)^2} \tag{5.65}$$

The first part is the κ_{sw}-to-period ratio derived from (5.6) and (5.24), which is

$$\frac{\kappa_{sw}}{T_0}=\frac{1}{5N\left(1+m\right)V_{DD}}\sqrt{\frac{8kT\gamma_s\cdot\mu_{eff}\left(V_{DD}-V_T\right)}{C_{ox}WL^3}} \tag{5.66}$$

So it is valid for CMOS inverters implemented with both long- and short-channel MOSFETs that

$$\frac{\kappa_{sw}}{T_0}\propto\frac{1}{\sqrt{WL^3}} \tag{5.67}$$

The second part is the κ_{load}-to-period ratio derived from (5.10) and (5.30), which is

$$\frac{\kappa_{load}}{T_0}=\frac{1}{5N\left(1+m\right)V_{DD}}\sqrt{\frac{8kT\gamma_t v_{sat}}{C_{ox}WL^2V_{DD}}\cdot\frac{\left(V_{DD}-V_T\right)}{1+\dfrac{L}{L_c}}} \tag{5.68}$$

Substituting (5.66) and (5.68) into (5.65), the κ_{total}-to-period ratio is

$$\frac{\kappa_{total}}{T_0}=\frac{1}{5N\left(1+m\right)V_{DD}}\sqrt{\frac{4kT\mu_{eff}\left(V_{DD}-V_T\right)2\gamma_s}{C_{ox}WL^3}}\cdot\sqrt{1+\frac{\delta}{\left(1+\dfrac{L_c}{L}\right)}} \tag{5.69}$$

For the case of long channel, $L>>L_c$, (5.69) simplifies to

$$\frac{\kappa_{total}}{T_0}=\frac{\sqrt{1+\delta}}{5N(1+m)V_{DD}}\sqrt{\frac{4kT\mu_{eff}(V_{DD}-V_T)2\gamma_s}{C_{ox}WL^3}}. \tag{5.70}$$

For the case of short channel, $L<<L_c$, (5.69) simplifies to

$$\frac{\kappa_{total}}{T_0}=\frac{1}{5N(1+m)V_{DD}}\sqrt{\frac{4kT\mu_{eff}(V_{DD}-V_T)2\gamma_s}{C_{ox}WL^3}} \tag{5.71}$$

Therefore, no matter long or short channel, it always holds that

$$\frac{\kappa_{total}}{T_0}\propto\frac{1}{\sqrt{WL^3}} \tag{5.72}$$

For the case of long channel, according to (5.12), (5.57), and (5.72), to achieve the same κ_{total}-to-period ratio for a ring oscillator which is Z^2 times faster realized by decreasing the channel length L by Z times, the channel width W needs to be increased cubically by Z^3 times, which means that the transistor area increases by about Z^3 times. Since the drain current I_D is proportional to W/L, the power dissipation will increase by Z^4 times.

For the case of short channel, to achieve the same κ_{total}-to-period ratio for a ring oscillator which is Z^2 times faster, L needs to be decreased by Z^2 times. W still needs to be increased cubically according to (5.72), which is by Z^6 times. This means that the transistor area increases by about Z^6 times. Since the drain current I_D is proportional to W and has nothing to do with L according to (2.12), the power dissipation will increase by Z^6 times!

Therefore, under the condition of achieving same κ_{total}-to-period ratio, the relationship between the speed, power dissipation, and die area is

$$f_0 \propto \begin{cases} P^{\frac{1}{2}} & \text{when } L \gg L_c \\ P^{\frac{1}{3}} & \text{when } L \ll L_c \end{cases} \tag{5.73}$$

$$f_0 \propto \begin{cases} A_{VCO}^{\frac{2}{3}} & \text{when } L \gg L_c \\ A_{VCO}^{\frac{1}{3}} & \text{when } L \ll L_c \end{cases} \tag{5.74}$$

Results of (5.73) and (5.74) show that in deep submicron process, much more power and die areas are needed to achieve similar normalized rms jitter or BER while improving the speed.

If the designer does have the freedom of the PLL loop bandwidth f_L, the power dissipation and die area consumption can be improved. From (5.64), if $f_L >> f_c$, the normalized rms jitter is

$$\left(\sigma_x\right)_{UI} = \frac{\sigma_x}{T_0} = \frac{\kappa}{T_0}\sqrt{\frac{1}{4\pi f_L}} \quad [\mathrm{UI}] \tag{5.75}$$

Since the f_L can be as high as 10% of the PLL's speed [93], with an increased f_L, the power dissipation in (5.73) and the die area in (5.74) can be improved as

$$f_0 \propto \begin{cases} P & \text{when } L \gg L_c \\ P^{\frac{1}{2}} & \text{when } L \ll L_c \end{cases} \tag{5.76}$$

$$f_0 \propto \begin{cases} A_{VCO}^{2} & \text{when } L \gg L_c \\ A_{VCO}^{\frac{1}{2}} & \text{when } L \ll L_c \end{cases} \tag{5.77}$$

5.4.3 Normalized Cycle-to-cycle Jitter and VCO Geometry

For applications such as clock generation for processors, the local oscillator (LO) is free-running to provide the master clock. In this case it is the ratio of the jitter accumulated in one cycle to the clock period that matters. The jitter accumulated in one clock period is the cycle-to-cycle jitter, which is

$$\sigma_{T_0} = \kappa\sqrt{T_0} \quad [\text{sec rms}] \tag{5.78}$$

The ratio between the jitter accumulated in one cycle and the clock period is defined as the normalized cycle-to-cycle jitter, which is

$$\left(\sigma_{T_0}\right)_{UI} = \frac{\kappa}{\sqrt{T_0}} \quad [\text{UI}] \tag{5.79}$$

From (5.31), (5.79) can be rewritten as

$$\left(\sigma_{T_0}\right)_{UI} = \sqrt{\left(\sigma_{sw}\right)_{UI}^2 + \left(\sigma_{load}\right)_{UI}^2} \quad [\text{UI}] \tag{5.80}$$

where

$$\left(\sigma_{sw}\right)_{UI} = \frac{\kappa_{sw}}{\sqrt{T_0}} \quad [\text{UI}] \tag{5.81}$$

$$\left(\sigma_{load}\right)_{UI} = \frac{\kappa_{load}}{\sqrt{T_0}} \quad [\text{UI}] \tag{5.82}$$

Substituting T_0 in (5.10), κ_{sw} in (5.23), and κ_{load} in (5.30) into (5.81) and (5.82), the normalized cycle-to-cycle jitter due to the switching and load MOSFETs are

$$\left(\sigma_{sw}\right)_{UI} = \sqrt{\frac{8kT\gamma_s}{5N\left(1+m\right)C_{ox}\left(V_{DD}-V_T\right)V_{DD}} \cdot \frac{L+L_c}{WL^2}} \quad [\text{UI}] \tag{5.83}$$

$$\left(\sigma_{load}\right)_{UI} = \sqrt{\frac{4kT\gamma_t}{5N\left(1+m\right)C_{ox}WLV_{DD}^2}} \quad [\text{UI}] \tag{5.84}$$

Thus

$$\left(\sigma_{sw}\right)_{UI} \propto \begin{cases} \dfrac{1}{\sqrt{WL}} & \text{when } L \gg L_c \\ \dfrac{1}{\sqrt{WL^2}} & \text{when } L \ll L_c \end{cases} \tag{5.85}$$

$$\left(\sigma_{load}\right)_{UI} \propto \frac{1}{\sqrt{WL}} \quad \text{for all L} \tag{5.86}$$

Substituting (5.83) and (5.84) into (5.80), the normalized cycle-to-cycle jitter is

$$\left(\sigma_{T_0}\right)_{UI} = \sqrt{\frac{8kT\gamma_s}{5N\left(1+m\right)C_{ox}V_{DD}\left(V_{DD}-V_T\right)} \cdot \frac{1}{WL}} \sqrt{\frac{L+L_c}{L} + \delta} \quad [\mathrm{UI}] \tag{5.87}$$

In the case of long channel, L>>L_c, (5.87) simplifies to

$$\left(\sigma_{T_0}\right)_{UI} = \sqrt{\frac{8kT\gamma_s\left(1+\delta\right)}{5N\left(1+m\right)C_{ox}V_{DD}\left(V_{DD}-V_T\right)} \cdot \frac{1}{WL}} \quad [\mathrm{UI}] \tag{5.88}$$

In the case of short channel, L<<L_c, (5.87) simplifies to

$$\left(\sigma_{T_0}\right)_{UI} \approx \left(\sigma_{sw}\right)_{UI} = \sqrt{\frac{8kT\gamma_s}{5N\left(1+m\right)C_{ox}V_{DD}E_c} \cdot \frac{1}{WL^2}} \quad [\mathrm{UI}] \tag{5.89}$$

Therefore,

$$\left(\sigma_{T_0}\right)_{UI} \propto \begin{cases} \dfrac{1}{\sqrt{WL}} & \text{when } L \gg L_c \\ \dfrac{1}{\sqrt{WL^2}} & \text{when } L \ll L_c \end{cases} \tag{5.90}$$

For the case of long channel, to achieve the same normalized cycle-to-cycle jitter for a ring oscillator which is Z^2 times faster realized by decreasing the channel length L by Z times, the channel width W needs to be increased linearly by Z times according to (5.90), which means that the die area increases by about Z times. Since the drain current I_D is proportional to W/L, the power dissipation will increase by Z^2 times.

For the case of short channel, to achieve the same normalized cycle-to-cycle jitter for a ring oscillator which is Z^2 times faster, the channel length L needs to be decreased by Z^2 times. The channel width W needs to be increased squarely according to (5.90), which is by Z^4 times. This means that the die area increases by Z^4 times. Since the drain current I_D is proportional to W and has nothing to do with L according to (2.12), the power dissipation will increase by Z^4 times.

Therefore, under the condition of achieving same normalized cycle-to-cycle jitter, the relationship between the speed, power dissipation, and die area is

$$f_0 \propto \begin{cases} P & \text{when } L \gg L_c \\ P^{\frac{1}{2}} & \text{when } L \ll L_c \end{cases} \tag{5.91}$$

$$f_0 \propto \begin{cases} A_{VCO}^2 & \text{when } L \gg L_c \\ A_{VCO}^{\frac{1}{2}} & \text{when } L \ll L_c \end{cases} \tag{5.92}$$

Results of (5.91) and (5.92) show that in deep submicron process, the cost of power dissipation and die area to improve oscillator speed while achieving the same normalized cycle-to-cycle jitter is a little better than the case discussed in last section which is to achieve the same normalized rms jitter. But still, more power and die area are needed.

5.4.4 Summary

For convenience, the relationship between jitter, VCO geometry, and ring configuration discussed in previous sections is summarized in Table 5.2. The relationship between jitter, power dissipation, and die area is summarized in Table 5.3. For the analysis of die area consumption, the channel lengths of VCO are assumed to near the minimum channel length of the semiconductor process.

Table 5.2 Relationship between jitter, VCO geometry, and ring configuration.

	Long-channel		Short-channel		N
	W	L	W	L	
f_0	WD	$\frac{1}{L^2}$	WD	$\frac{1}{L}$	$\frac{1}{N}$
κ_{total}	$\frac{1}{\sqrt{W}}$	$\sqrt{L}$	$\frac{1}{\sqrt{W}}$	$\frac{1}{\sqrt{L}}$	WD
$(\sigma_x)_{UI}$	$\frac{1}{\sqrt{W}}$	$\frac{1}{\sqrt{L^3}}$	$\frac{1}{\sqrt{W}}$	$\frac{1}{\sqrt{L^3}}$	$\frac{1}{N}$
$(\sigma_{T_0})_{UI}$	$\frac{1}{\sqrt{W}}$	$\frac{1}{\sqrt{L}}$	$\frac{1}{\sqrt{W}}$	$\frac{1}{L}$	$\frac{1}{\sqrt{N}}$

(WD: very weak dependency if considering second order effects)

Table 5.3 Relationship between jitter, power dissipation and die area.

	Long-channel		Short-channel	
	P	A_{VCO}	P	A_{VCO}
f_0	P^2	WD	WD	WD
κ_{total}	$\frac{1}{\sqrt{P}}$	$\frac{1}{\sqrt{A_{VCO}}}$	$\frac{1}{\sqrt{P}}$	$\frac{1}{\sqrt{A_{VCO}}}$
$(\sigma_x)_{UI}$ fixed	$f_0 \propto \sqrt{P}$	$f_0 \propto \sqrt[3]{A_{VCO}^2}$	$f_0 \propto \sqrt[3]{P}$	$f_0 \propto \sqrt[3]{A_{VCO}}$
$(\sigma_{T_0})_{UI}$ fixed	$f_0 \propto P$	$f_0 \propto A_{VCO}^2$	$f_0 \propto \sqrt{P}$	$f_0 \propto \sqrt{A_{VCO}}$

5.5 Jitter and VCO Tuning

The ring oscillator formed by the CMOS inverters in last section always runs at constant speed. But for the VCO, the speed should be able to be controlled by an applied voltage. According to (5.4), the inverter propagation delay T_d can be controlled by tuning the switching current I_D [7], [9], output swing V_{SW} [80], or the load capacitance C_L. The most popular method is limiting the inverter's switching current.

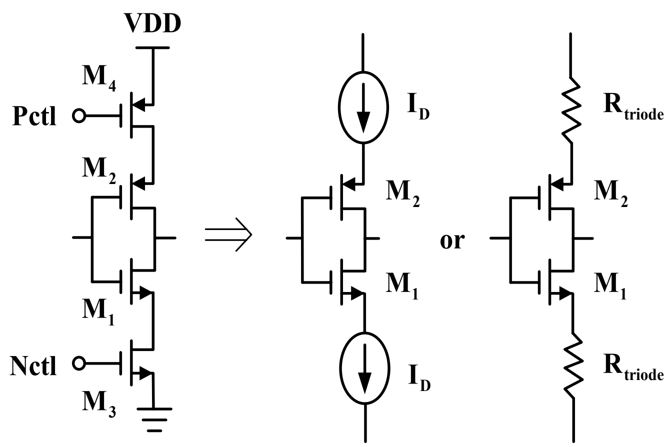


Figure 5.14 The current-starved inverter.

Figure 5.14 shows a simple implementation of tuning the switching current of the CMOS inverter, the current-starved inverter [40]. MOSFETs M_1 and M_2 operate as an inverter. When M_3 and M_4 are biased in saturation they operate as current sources to limit the current available for switching. In other words, the inverter is starved for current. When M_3 and M_4 are biased in triode, they are equivalent to voltage-controlled resistors to affect the switching current.

5.5.1 Tuning Transistors in the Saturation Region

When the tuning transistors are biased in saturation, they operate as current sources to control the current available for switching, and their output resistance is usually much higher than the equivalent resistance of the switching transistors. Since the switching current is starved, the output swing may not go rail-to-rail.

When the input to the current-starved inverter is switching from low (V_L) to high (V_H), M_1 in Figure 5.14 is the switching MOSFET. The κ_{sw} is the same as that for the simple inverter in (5.22) with the current I_D is controlled by M_3:

$$\kappa_{sw} = \frac{i_n}{\sqrt{\Delta f}} \frac{1}{\sqrt{2} I_{ctl}} = \frac{\sqrt{4kT\gamma_s \mu_{eff} C_{ox} \frac{W_{sw}}{L_{sw}} \left(V_H - V_T\right)}}{\sqrt{2} I_{ctl}} \quad \left[\sqrt{\sec}\right] \tag{5.93}$$

where W_{sw} and L_{sw} are the channel width and length for the switching NMOS transistor, I_{ctl} is the drain current of the control transistor. From (2.11), the relationship between I_{ctl} and the control voltage V_{ctl} is

$$I_{ctl} \propto \begin{cases} \left(V_{ctl} - V_T\right)^2 & \text{when } L \gg L_c \\ \left(V_{ctl} - V_T\right) & \text{when } L \ll L_c \end{cases} \tag{5.94}$$

For the case of long channel, $L >> L_c$, and (5.93) simplifies to

$$\kappa_{sw} = \frac{i_n}{\sqrt{\Delta f}} \frac{1}{\sqrt{2} I_{ctl}} = \sqrt{\frac{16kT}{3\mu_{eff} C_{ox}} \cdot \frac{W_{sw}}{W_{ctl}^2} \cdot \frac{L_{ctl}^2}{L_{sw}} \cdot \frac{\left(V_H - V_T\right)}{\left(V_{ctl} - V_T\right)^4}} \quad \text{when } L \gg L_c \tag{5.95}$$

where W_{ctl} and L_{ctl} are the channel width and length for the NMOS tuning transistor.

For the case of short channel, $L << L_c$, and (5.93) simplifies to

$$\kappa_{sw} = \sqrt{\frac{4kT\gamma_s}{E_c v_{sat} C_{ox}} \cdot \frac{W_{sw}}{W_{ctl}^2} \cdot \frac{1}{L_{sw}} \frac{\left(V_H - V_T\right)}{\left(V_{ctl} - V_T\right)^2}} \quad \text{when } L \ll L_c \tag{5.96}$$

From (5.95) and (5.96), larger W_{ctl} and shorter L_{ctl} at a fixed V_{ctl} are able to provide more switching current, thus better to minimize the jitter due to the switching transistors. The control voltage V_{ctl} is usually less than half of the power supply V_{DD} to maintain the tuning transistor in saturation, which is the major factor for the increased switching noise.

The channel thermal noise from the control transistor M_3 will also introduce an integrated noise voltage over the load capacitance. The figure of merit for this integrated noise is defined as κ_{ctl}. The analysis for κ_{ctl} is similar to that for κ_{sw}, the result is

$$\kappa_{ctl} = \sqrt{\frac{8}{3}kT \cdot \frac{1}{\frac{1}{2}\mu_{eff} C_{ox} \frac{W_{ctl}}{L_{ctl}} \left(V_{ctl} - V_T\right)^3}} \qquad \text{when } L \gg L_c \tag{5.97}$$

and

$$\kappa_{ctl} = \sqrt{\frac{4kT\gamma_s}{E_c L_{sw} v_{sat} C_{ox} W_{ctl} \left(V_{ctl} - V_T\right)}} \qquad \text{when } L \ll L_c \tag{5.98}$$

Usually W_{ctl} is larger than W_{sw} for jitter and speed considerations. Comparing the results of (5.95), (5.96), (5.97), and (5.98), κ_{ctl} is usually larger than κ_{sw}.

M_2 and M_4 are loading transistors and can be modeled as resistors in series. The combined noise is still the KTC noise analyzed in last section. But the inverter propagation delay changes to

$$T_d = \frac{V_{SW} C_L}{2I_{ctl}} \quad [\text{sec}] \tag{5.99}$$

where V_{SW} is the output swing which equals (V_H-V_L).

Thus κ_{load} is

$$\kappa_{load} = \sqrt{\frac{2kT\gamma_t}{I_{ctl} V_{SW}}} \quad \left[\sqrt{\text{sec}}\right] \tag{5.100}$$

κ_{load} is worse than that for the CMOS inverter since both the switching current and the voltage swing are smaller.

The ratio between κ_{sw} and κ_{load} has been evaluated in Section 5.3. According to (5.40) and (5.50),

$$\begin{cases} \sqrt{\dfrac{4}{3}} < \dfrac{\kappa_{sw}}{\kappa_{load}} < 2 & \text{when } L \gg L_c \\ \dfrac{\kappa_{sw}}{\kappa_{load}} \gg 1 & \text{when } L \ll L_c \end{cases} \tag{5.101}$$

Since now there are two switching noise sources for the current-starved inverter and κ_{ctl} is usually larger than κ_{sw}, the κ_{total} of the current-starved invert is usually dominated by κ_{sw} and κ_{ctl}. Therefore from (5.93) and (5.97),

$$\kappa_{total} \propto \frac{1}{I_{ctl}} \tag{5.102}$$

Since the propagation delay is inversely proportional to I_{ctl} as of (5.99), the relationship between κ and the inverter delay during tuning is

$$\kappa_{total} \propto T_d \tag{5.103}$$

5.5.2 Tuning Transistors in the Triode Region

When the tuning transistors are biased in the triode region, they can be modeled as voltage-controlled resistors as shown in Figure 5.14. The equivalent resistance R_{triode} can be approximated by

$$R_{triode} = \frac{1}{\mu_{eff} C_{ox} \frac{W_{ctl}}{L_{ctl}} \left(V_{ctl} - V_T \right)} \tag{5.104}$$

From (5.104), the MOSFET with larger channel width has a smaller equivalent resistance. R_{triode} will reach its minimum value when V_{ctl} reaches the power supply V_{DD}, which is

$$\left(R_{triode} \right)_{\min} = \frac{1}{\mu C_{ox} \frac{W_{ctl}}{L_{ctl}} \left(V_{DD} - V_T \right)} \tag{5.105}$$

The inverter propagation delay is proportional to the time constant at the output node,

$$T_d \propto \left(R_{triode} + R_{sw} \right) C_L \tag{5.106}$$

where R_{sw} is the equivalent resistance of the switching transistor. So V_{ctl} has less impact to the total resistance at the output node. And the VCO gain factor, K_{VCO}, will be much less than that when the tuning transistors are biased in saturation.

Since the channel width of the tuning transistors are usually larger than that of the switching MOSFETs to provide reasonable output swing, R_{triode} is comparable or even smaller than R_{sw}. Therefore the switching current is much higher than the case that the tuning transistors are biased in saturation, the output swing is more likely to be rail-to-rail, and κ_{total} is better. When R_{triode} goes to zero, the κ_{total} of the current-starved inverter will approach the κ_{total} of the CMOS inverter.

Therefore, in order to minimize κ, tuning should be limited and the VCO should run at or nearby its maximum speed.

5.6 Experimental Results

5.6.1 Test Chip Design

Four sets of single-ended ring oscillators were designed to evaluate the relationship between the white noise figure of merit κ and the VCO geometry. The delay stage is implemented by the current-starved inverter in Figure 5.14.

Table 5.4 shows the geometry range of the NMOS switching transistor M_1. The size of the PMOS switching transistor M_2 is 1.4 times as that of the M_1 to provide rise and fall times as symmetric as possible. The control transistors M_3 and M_4 are twice the size of the switching transistors.

The ring configuration for the four VCO sets is listed in Table 5.5. If the oscillators are all implemented with 3-stage ring, the VCO speed will be as high as 2.7GHz for VCOs in set I. Due to package limitations and the difficulty to maintain signal integrity of GHz signal off the chip, 25- and 7-stage ring are used for VCO set I and II to reduce the speed. κ is the property of the individual stage, not the number of stages. Therefore the figure of merit κ will not be affected by adding more stages as discussed in Section 5.2.

Each ring oscillator on the chip has its own power supply V_{DD} so that the rest oscillators can be disabled and will not introduce interference to the oscillator under test. In order to save the die area, all the oscillator outputs are fed into a multiplexer followed by a current buffer to drive the signal off the chip.

This test chip was fabricated in TSMC 0.18μm 1-poly 6-metal CMOS process with power supply of 1.8V. Figure 5.15 shows the die micrograph, and Figure 5.16 shows the micrograph of a 7-stage ring oscillator with M_1 size of 20μm/0.6μm.

Table 5.4 Geometry range of the implemented oscillators (W/L μm).

Set I	10/0.18	20/0.18	60/0.18	100/0.18	200/0.18	600/0.18
Set II	10/0.6	20/0.6	60/0.6	100/0.6	200/0.6	600/0.6
Set III	10/1.8	20/1.8	60/1.8	100/1.8	200/1.8	600/1.8
Set IV	10/6	20/6	60/6	100/6	200/6	600/6

(Only the geometry of M1 is listed)

Table 5.5 Ring configuration of the implemented oscillators.

Set	L (μm)	N	Measured Min. Inv. Delay	Measured fmax	fmax for N=3
I	0.18	25	61.6 ps/gate	325MHz	2.7GHz
II	0.6	7	272.1 ps/gate	263MHz	613MHz
III	1.8	3	1.73 ns/gate	96MHz	96MHz
IV	6	3	16.2 ns/gate	10.3MHz	10.3MHz

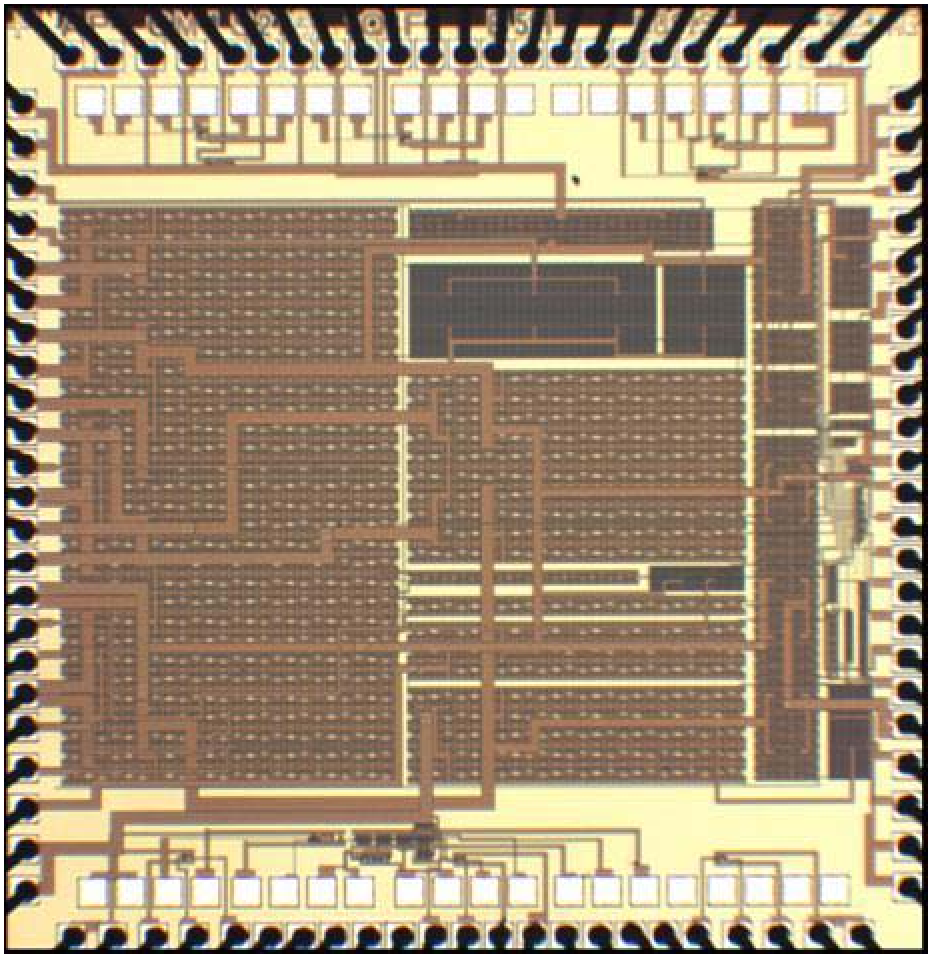


Figure 5.15 Die micrograph.

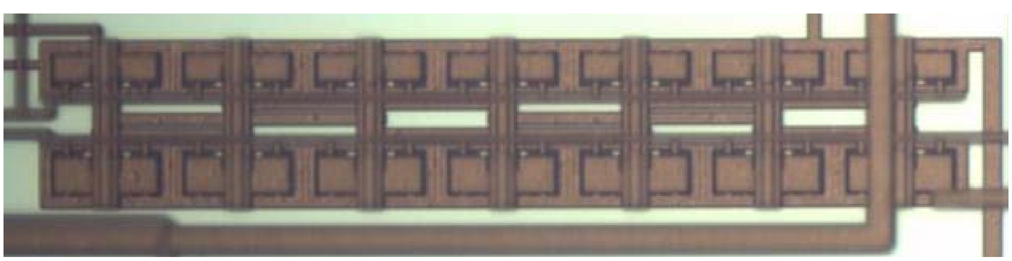
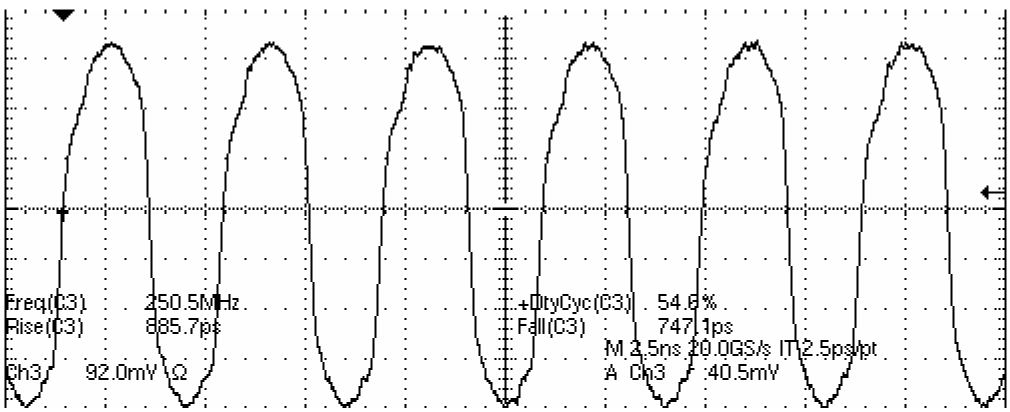


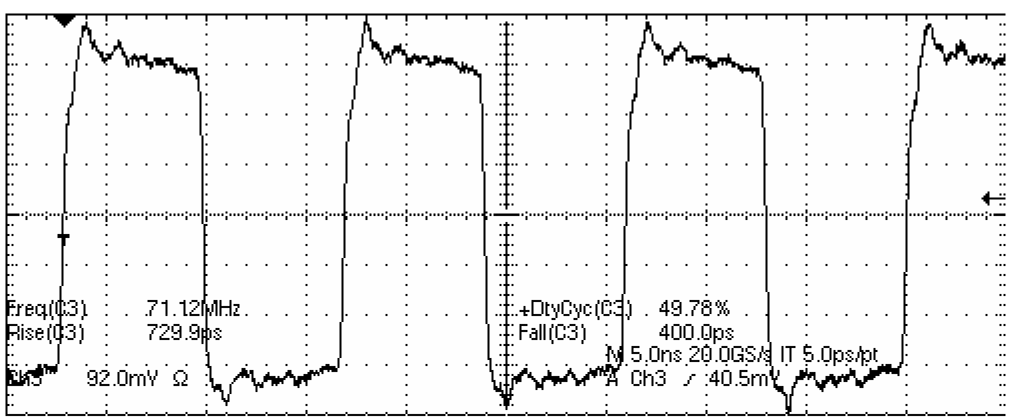
Figure 5.16 Micrograph of a 7-stage ring oscillator on chip.

5.6.2 Inverter Delay and Geometry

The ring oscillators are tested using digital oscilloscopes of LeCroy Wavemaster 8600A [61] and Tektronix TDS6604 [94] in the time domain, and Agilent Technologies E4440A spectrum analyzer [63] in the frequency domain. Figure 5.17 shows the measured output waveforms for oscillators in set I and III.



(a) Measured output waveform for an oscillator in set I.



(b) Measured output waveform for an oscillator in set III.

Figure 5.17 Measured oscillator output waveforms.

Table 5.6 lists the measured minimum inverter delay T_d for all the implemented ring oscillators. The data are also plotted in Figure 5.18 on a log-log scale. From Figure 5.18a, increasing the channel width W has a very weak effect on the VCO's speed. This matches the prediction in (5.7) that T_d has no dependency on W based on a first-order analysis.

In Figure 5.18b, the data from oscillators with longer channels (L=6μm and 1.8μm) show a slope of 2, which indicates that

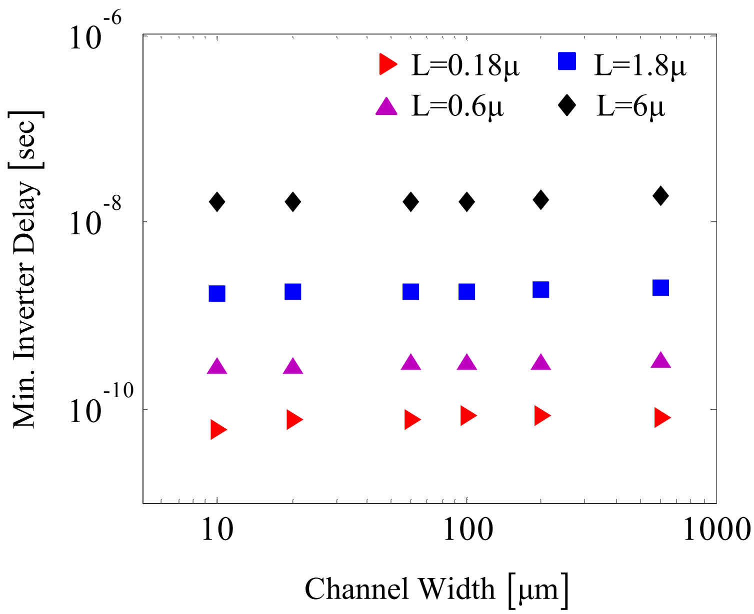
$$T_d \propto L^2 \tag{5.107}$$

This matches the prediction in (5.8).

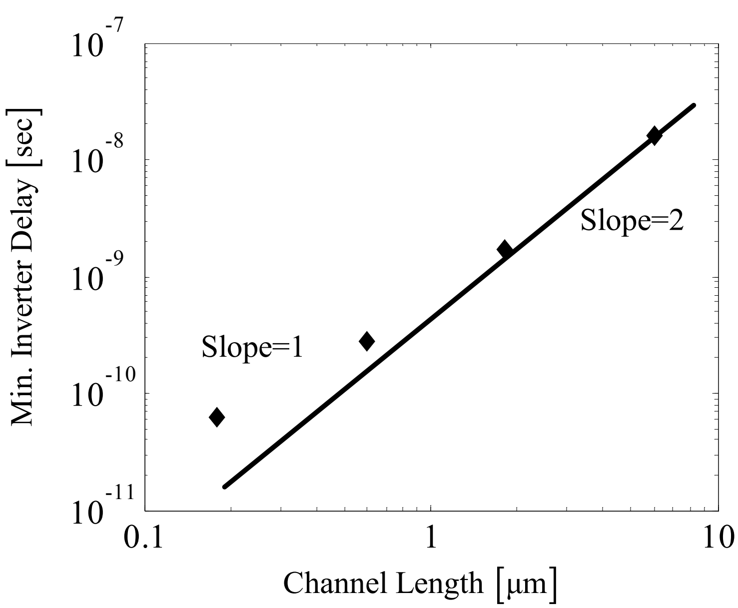
For oscillators with shorter channels (L=0.6μm and 0.18μm), T_d deviates the relationship of (5.107) due to the short channel effects.

Table 5.6 Measured min. inverter delay [sec] for implemented oscillators.

W \ L	0.18μm	0.6μm	1.8μm	6μm
10μm	6.16E-11	2.83E-10	1.73E-09	1.62E-08
20μm	7.89E-11	2.72E-10	1.75E-09	1.62E-08
60μm	7.87E-11	3.02E-10	1.80E-09	1.67E-08
100μm	8.38E-11	2.99E-10	1.82E-09	1.66E-08
200μm	8.33E-11	3.07E-10	1.87E-09	1.70E-08
600μm	7.96E-11	3.25E-10	1.95E-09	1.90E-08



(a) Measured minimum inverter delay versus channel width.



(b) Measured minimum inverter delay versus channel length.

Figure 5.18 Measured minimum inverter delay versus VCO geometry.

5.6.3 κ and VCO Geometry

Duty cycle distortion will result asymmetry in the oscillating waveform and introduce more integrated 1/f noise [10]. Figure 5.19 shows the measured duty cycle of the ring oscillator with M_1 size of 600μm/0.18μm for the full tuning range. All measured data are around 50%.

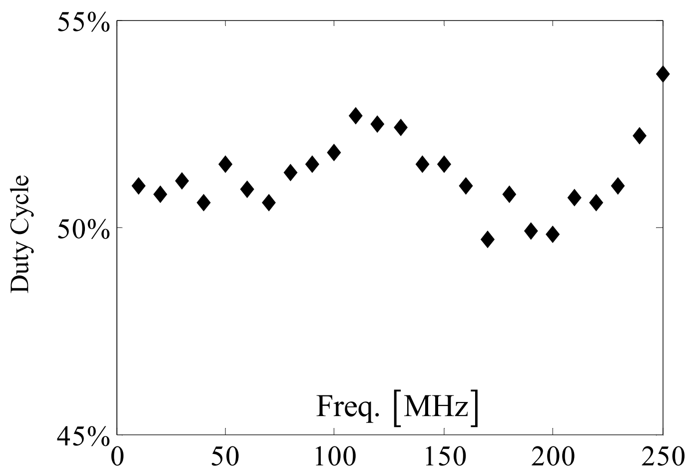


Figure 5.19 Measured duty cycle for the ring oscillator with M_1 size of 600μm/0.18μm.

Table 5.7 lists the measured κ for all implemented ring oscillators running at their maximum speed. The data are also plotted in Figure 5.20 and 5.21 on a log-log scale.

Figure 5.20 shows the measured relationship between the extracted κ and the channel width W. From the prediction in (5.44) and (5.53),

$$\kappa \propto \frac{1}{\sqrt{W}} \tag{5.108}$$

Using the least square method, the extracted slopes for the four VCO sets are -0.3841, -0.3668, -0.3668, and -0.3705 respectively, showing reasonable correspondence with errors between 23% and 27%.

Table 5.7 Measured κ versus VCO geometry.
(All oscillators are running at maximum speed)

W \ L	0.18μm	0.6μm	1.8μm	6μm
10μm	3.72E-09	4.78E-09	7.93E-09	1.53E-08
20μm	2.98E-09	3.42E-09	6.55E-09	1.24E-08
60μm	1.83E-09	2.57E-09	5.31E-09	9.36E-09
100μm	1.51E-09	2.20E-09	4.34E-09	7.51E-09
200μm	1.14E-09	1.55E-09	2.61E-09	4.90E-09
600μm	8.06E-10	1.02E-09	1.82E-09	3.48E-09

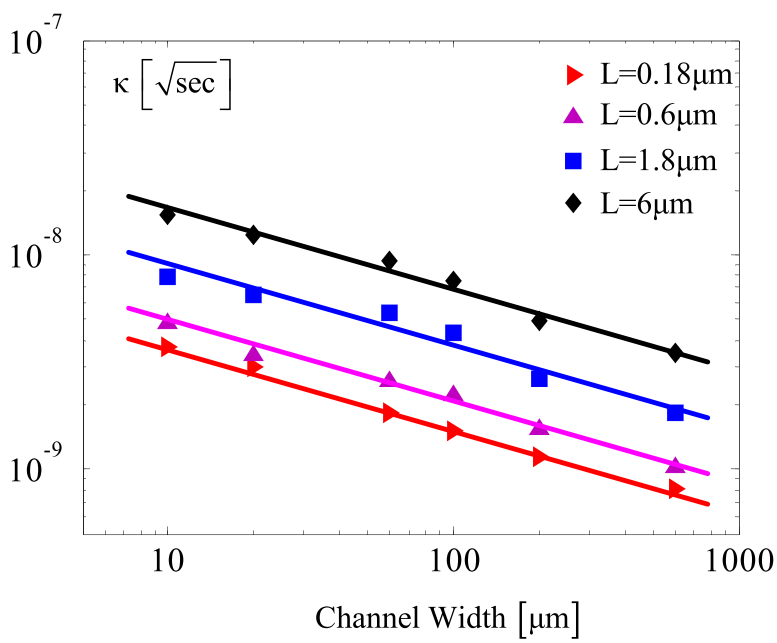


Figure 5.20 Measured relationship between κ and channel width.

Figure 5.21 shows the measured relationship between the extracted κ and the channel length L on a log-log scale.

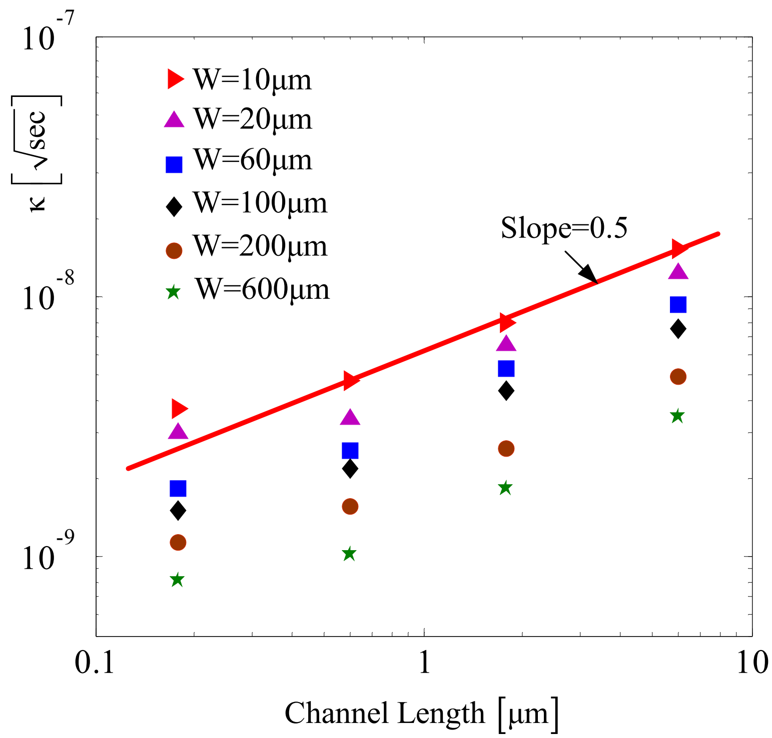


Figure 5.21 Measured relationship between κ and channel length.

From prediction in (5.45) and (5.54),

$$\kappa \propto \sqrt{L} \quad \text{when} \;\; L \gg L_c \tag{5.109}$$

$$\kappa \propto \frac{1}{\sqrt{L}} \quad \text{when} \;\; L \ll L_c \tag{5.110}$$

For TSMS 0.18μm process, v_{sat}=8.4E4 m/sec, μ_0=0.0459 m^2/V·sec, the computed μ_{eff}=0.0205 m^2/V·sec and V_T is around 0.5V for short-channel MOSFETs. From (2.10),

$$L_c = \frac{V_{eff}}{E_c} = \frac{V_{DD} - V_T}{2v_{sat}/\mu_{eff}} = \frac{1.8 - 0.5}{2 \times (8.4\text{E}4)/0.0205} \approx 0.16\mu\text{m} \tag{5.111}$$

For VCO sets II to IV with channel lengths (0.6μm, 1.8μm, and 6μm) much longer than L_c of 0.16μm, the measured κ is proportional to the square root of L as predicted by (5.109), and showing a slope of 0.5 on the log-log scale plot of Figure 5.21.

For oscillators in set I, the channel lengths are all 0.18μm and affected by the short channel effect. The measured κ deviates from the relationship predicted by (5.109). Due to short of data points, the optimum length is not observed. From the analysis in Section 5.4.1, the optimum length could be larger or smaller than 0.18μm depending on the dependency of γ_s and V_T on the channel length L.

Figure 5.22 shows the simulated V_T for a single MOSFET biased in the saturation region with both V_{GS} and V_{DS} of 1.8V. Only the roll-up of V_T is observed.

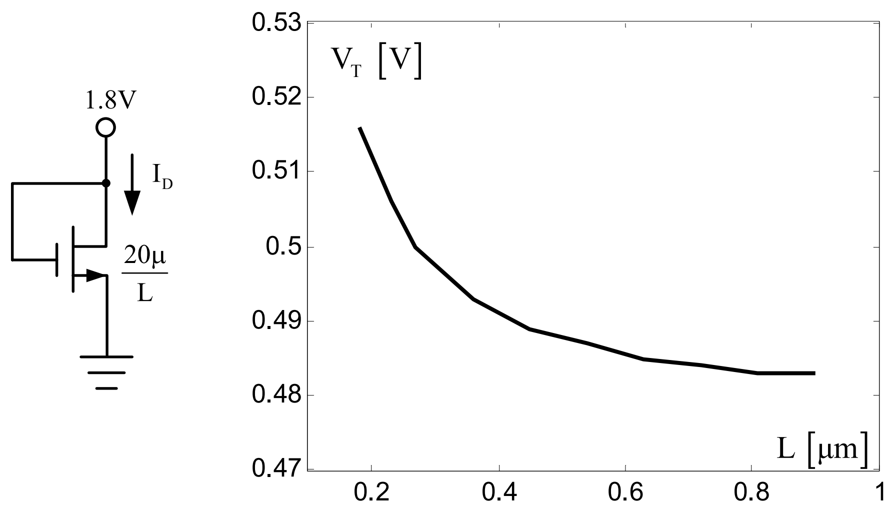


Figure 5.22 Simulated V_T roll-up for TSMC 0.18μm process.

There are no MOSFETs dedicated for γ_s extraction in this test chip. Figure 5.23 shows the extracted γ_s for MOSFETs in TSMC 0.18μm from the measurement results of [95]. The bias condition of V_{GST} in Figure 5.23 is the gate overdrive voltage which is (V_{GS}-V_T). The γ_s at bias condition of V_{GST}=1.3V and V_{GST}=1V are obtained by linear interpolation and plotted on Figure 5.23 with dashed lines.

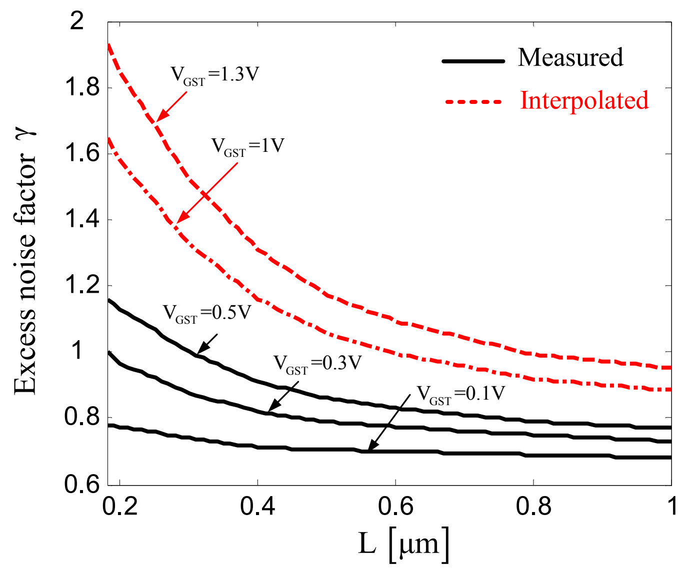


Figure 5.23 Measured γ_s for TSMC 0.18μm process in [95].

From the results in Figure 5.22 and 5.23, the V_T roll-up and increased γ_s will result in a longer optimum length than L_c (0.16μm).

Figure 5.24 shows the predicted and measured κ for the VCOs with M_1 width of 20μm. The agreement is within 30%. The γ_s for the switching and tuning MOSFETs in prediction are the interpolated γ_s in Figure 5.23.The drain current I_D, threshold voltage V_T, and the gate overdrive voltage V_{GST} are obtained by simulation of the half circuit in Figure 5.24 by Cadence. The predicted optimum length is around 0.36μm. Unfortunately there is no actual measured data to prove it.

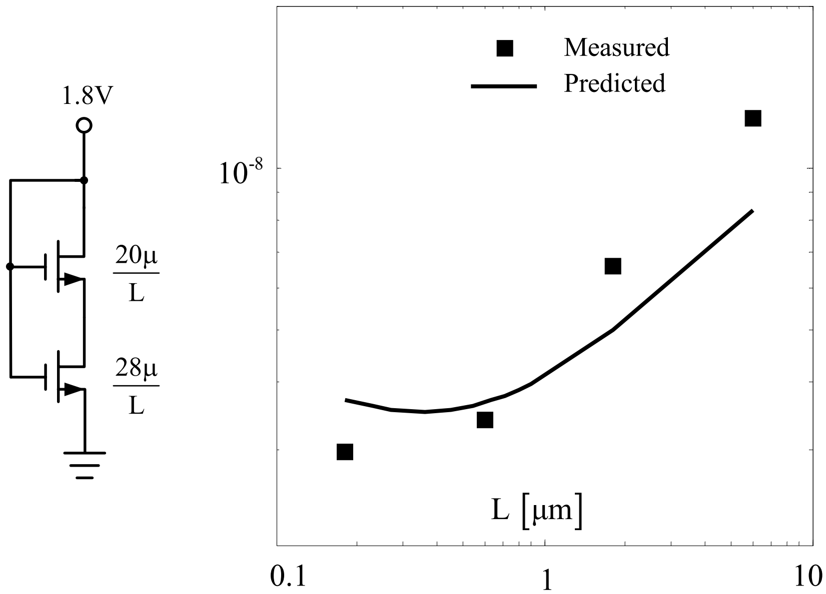


Figure 5.24 Predicted and measured κ for the VCOs with M_1 width of 20μm.

5.6.4 Normalized RMS Jitter and VCO Geometry

From (5.64), the normalized rms jitter is

$$(\sigma_x)_{UI} = \frac{\sigma_x}{T_0} = \frac{\kappa}{2NT_d}\sqrt{\frac{1}{4\pi f_L}\left(1+\frac{4f_c}{3\sqrt{3}f_L}\right)} \quad [\mathrm{UI}] \tag{5.112}$$

Thus only the κ-to-T_d ratio is related to the VCO geometry. From (5.69), the κ-to-T_d ratio is

$$\frac{\kappa}{T_d} = \frac{2}{5(1+m)V_{DD}}\sqrt{\frac{4kT\mu_{eff}(V_{DD}-V_T)2\gamma_s}{C_{ox}WL^3}} \cdot \sqrt{1+\frac{\delta}{\left(1+\frac{L_c}{L}\right)}} \tag{5.113}$$

For L>>L_c and L<<L_c, the relationship between the κ-to-T_d ratio and the VCO geometry is predicted by (5.72),

$$\frac{\kappa}{T_d} \propto \frac{1}{\sqrt{WL^3}} \tag{5.114}$$

Table 5.8 lists the measured κ-to-T_d for all implemented ring oscillators running at their maximum speed. The data are also plotted in Figure 5.25 and 5.26 on a log-log scale.

Figure 5.25 shows the relationship between the measured κ-to-T_d ratio and the channel width W on a log-log scale. The extracted slopes are from -0.3960 to -0.4360, showing errors between 13% and 21%.

Figure 5.26 shows the relationship between the extracted κ-to-T_d ratio and the channel length L on a log-log scale. From (5.111), L_c is calculated as 0.16μm. So only the channel lengths of VCOs in set III (1.8μm) and IV (6μm) qualifies the condition of L>>L_c. If just calculate the slopes from the data of VCO set III and IV, the results are between -1.3104 and -1.3823. Comparing to the predicted -1.5 from (5.114), the errors are within 13%.

For VCO set I and II, L is 0.18μm and 0.6μm respectively and not much larger than L_c. The κ-to-T_d ratio is actually proportional to

$$\frac{\kappa}{T_d} \propto \sqrt{\frac{1}{WL^3}} \cdot \sqrt{1+\frac{\delta}{\left(1+\frac{L_c}{L}\right)}} \qquad (5.115)$$

So when L approaches L_c, the κ-to-T_d ratio will drop from the value predicted by (5.114), which is exactly shown in Figure 5.26.

Table 5.8 Extracted κ-to-T_d ratio and VCO geometry.

W \ L	0.18μm	0.6μm	1.8μm	6μm	Slope (when L>>Lc)
10μm	60.39	16.91	4.58	0.94	-1.3134
20μm	37.77	12.57	3.74	0.77	-1.3160
60μm	23.25	8.50	2.95	0.56	-1.3807
100μm	18.02	7.35	2.39	0.45	-1.3823
200μm	13.69	5.04	1.40	0.29	-1.3104
600μm	10.13	3.14	0.93	0.18	-1.3552
Slope	-0.4360	-0.4055	-0.3960	-0.4078	

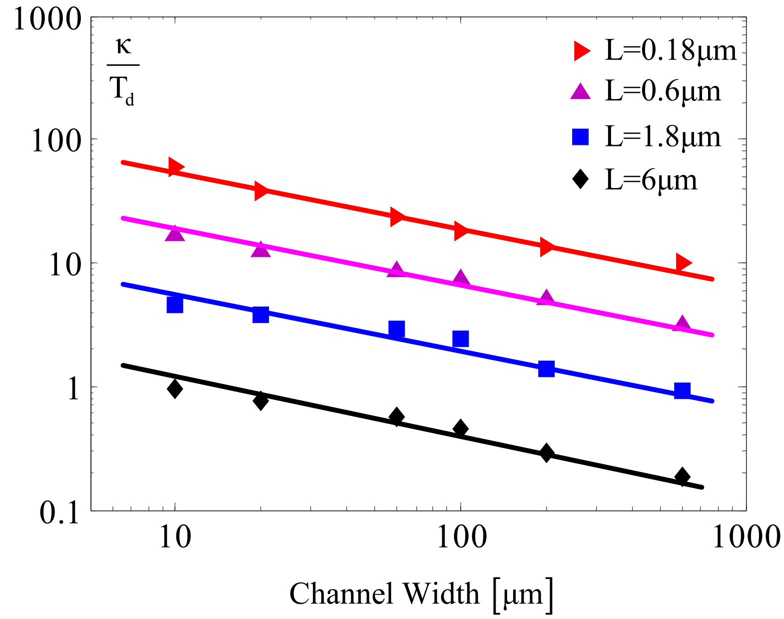


Figure 5.25 Measured κ-to-T_d ratio versus W.

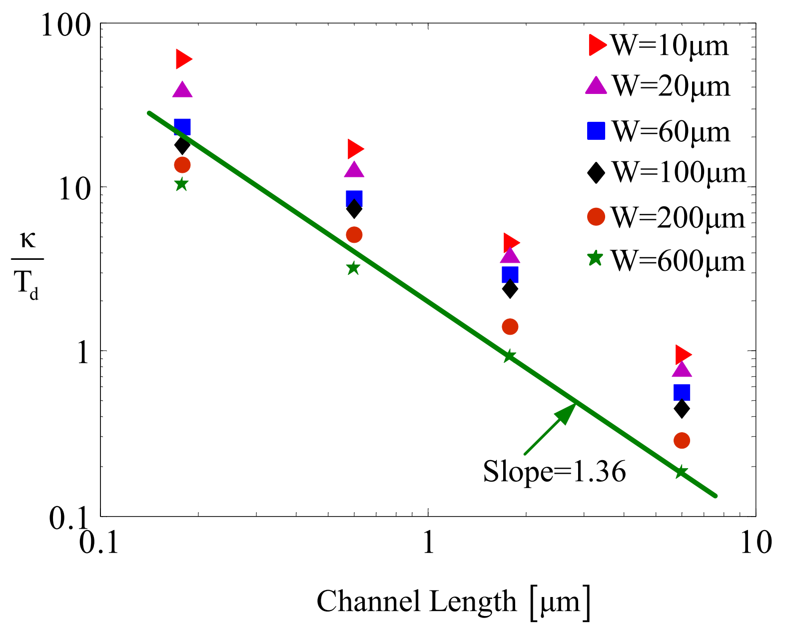


Figure 5.26 Measured κ-to-T_d ratio versus L.

5.6.5 Normalized Cycle-tocycle Jitter and VCO Geometry

From (5.79), the normalized cycle-to-cycle jitter is

$$\left(\sigma_{T_0}\right)_{UI} = \frac{\kappa}{\sqrt{T_0}} = \frac{\kappa}{\sqrt{2NT_d}} \quad [\text{UI}] \tag{5.116}$$

Thus only the κ-to-sqrt(T_d) ratio is related to the VCO geometry. From (5.87), the κ-to-sqrt(T_d) ratio is

$$\frac{\kappa}{\sqrt{T_d}} = \sqrt{\frac{16kT\gamma_s}{5(1+m)C_{ox}V_{DD}\left(V_{DD}-V_T\right)} \cdot \frac{1}{WL}} \sqrt{\frac{L+L_c}{L}+\delta} \quad [\text{UI}] \tag{5.117}$$

For L>>L_c and L<<L_c, the relationship between the κ-to-sqrt(T_d) ratio and the VCO geometry is predicted by (5.90),

$$\frac{\kappa}{\sqrt{T_d}} \propto \begin{cases} \dfrac{1}{\sqrt{WL}} & \text{when } L \gg L_c \\ \dfrac{1}{\sqrt{WL^2}} & \text{when } L \ll L_c \end{cases} \tag{5.118}$$

Table 5.9 lists the measured κ-to-sqrt(T_d) for all implemented ring oscillators running at their maximum speed, and the data are also plotted in Figure 5.27 and 5.28 on a log-log scale.

Figure 5.27 shows the relationship between the extracted κ-to-sqrt(T_d) ratio and the channel width W in a log-log scale. The extracted slopes are from -0.3812 to -0.4099, showing errors between 18% and 24%.

Figure 5.28 shows the relationship between the extracted κ-to-sqrt(T_d) ratio and the channel length L in a log-log scale. Only the data from VCO set III and IV satisfy L>>L_c. The extracted slopes are from -0.3837 to -0.4634, showing errors between 24% and 7%.

Table 5.9 The extracted κ-to-sqrt(T_d) ratio and VCO geometry.

W \ L	0.18µm	0.6µm	1.8µm	6µm	Slope (when L>>Lc)
10µm	4.74E-04	2.84E-04	1.91E-04	1.20E-04	-0.3837
20µm	3.35E-04	2.07E-04	1.56E-04	9.75E-05	-0.3930
60µm	2.06E-04	1.48E-04	1.25E-04	7.23E-05	-0.4550
100µm	1.65E-04	1.27E-04	1.02E-04	5.83E-05	-0.4634
200µm	1.25E-04	8.84E-05	6.04E-05	3.76E-05	-0.3936
600µm	9.03E-05	5.66E-05	4.12E-05	2.52E-05	-0.4084
Slope	-0.4099	-0.3858	-0.3812	-0.3878	

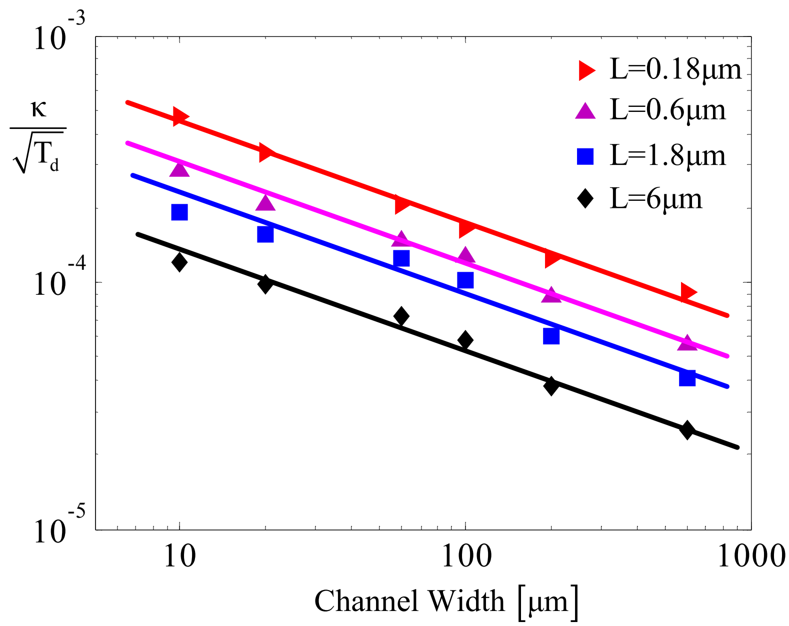


Figure 5.27 Measured κ-to-sqrt(T_d) ratio versus W.

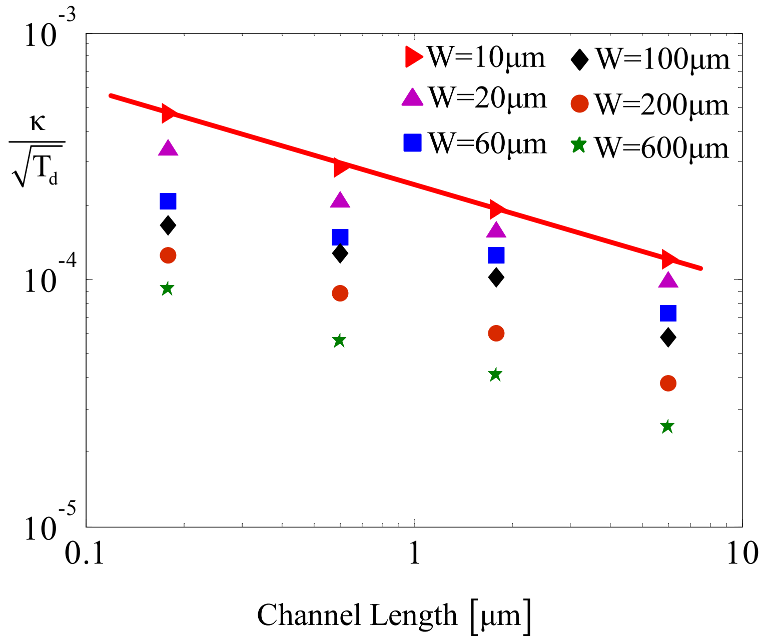


Figure 5.28 Measured κ-to-sqrt(T_d) ratio versus L.

5.6.6 κ and VCO Tuning

Figure 5.29 shows the measured VCO tuning characteristic for the ring oscillator with M_1 size of 600μm/0.18μm.

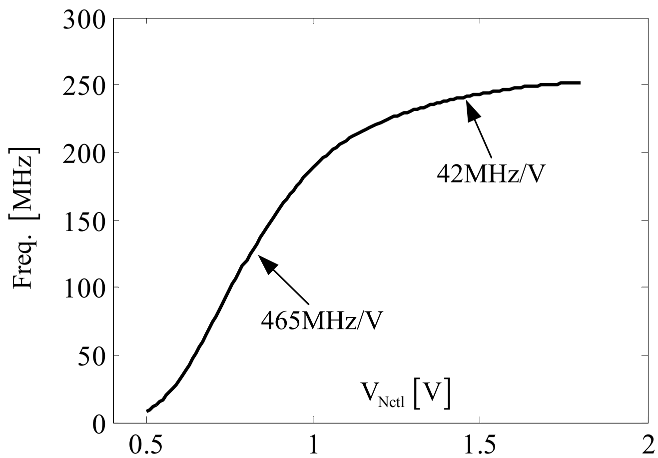


Figure 5.29 Measured tuning characteristic for the ring oscillator with M_1 size of 600μm/0.18μm.

The voltage to frequency transfer characteristic is not linear in the full tuning range. The reason has been discussed in Section 5.5. When V_{ctl} is lower, the tuning MOSFETs are biased in saturation, and from (5.94)

$$I_{ctl} \propto \begin{cases} (V_{ctl} - V_T)^2 & \text{when } L \gg L_c \\ (V_{ctl} - V_T) & \text{when } L \ll L_c \end{cases} \tag{5.119}$$

Since the propagation delay is inversely proportional to the switching current, the oscillating speed f_0 is

$$f_0 \propto \begin{cases} (V_{ctl} - V_T)^2 & \text{when } L \gg L_c \\ (V_{ctl} - V_T) & \text{when } L \ll L_c \end{cases} \tag{5.120}$$

When V_{ctl} is high, the tuning MOSFETs are biased in triode. There is more current for switching and the speed is faster. However as analyzed in Section 5.5, the resulting VCO gain factor dropped from 465MHz/V to 42MHz/V.

Figure 5.30 shows the measured relationship between the extracted κ and the measured inverter delay T_d on a log-log scale for the VCO with M_1 size of 10μm/0.6μm. Similar results are obtained for other oscillators. The tuning is realized by varying the control voltage to limit the switching current. It is equivalent to limiting the power dissipation and thus degrading the VCO's jitter performance.

When the tuning transistors are biased in the saturation region, a slope of 1 is extracted for the data plotted in a log-log scale. Thus measured results match the analysis in (5.103) that during tuning,

$$\kappa_{total} \propto T_d \tag{5.121}$$

When the tuning transistors are biased in triode, a slope of 1.5 is extracted for the data plotted on a log-log scale, which matches the prediction that the jitter will be much better than the case that the tuning transistors are biased in saturation.

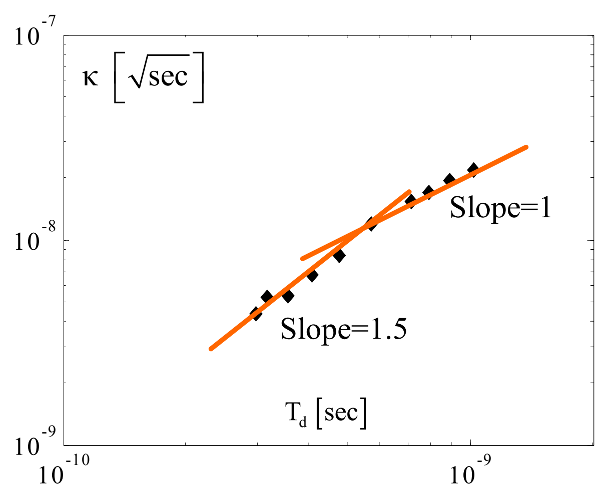


Figure 5.30 Measured relationship between κ and inverter during tuning.

5.7 Summary: Design of Low Jitter Ring Oscillators

5.7.1 Design for κ

The measurement results for minimum inverter delay and κ as a function of the VCO geometry are summarized by the contour plot in Figure 5.31 with an interpolated optimum channel length around 0.24μm. Thus the optimum geometry of ring oscillators to minimize κ is along the arrow in Figure 5.31.

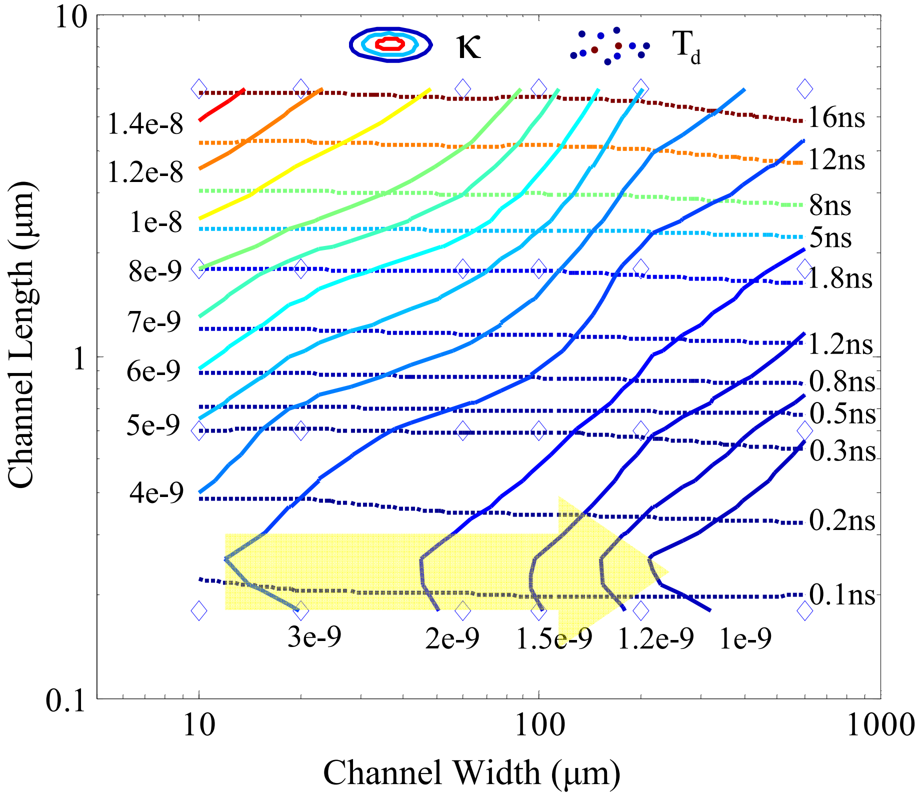


Figure 5.31 Contour plot for measured κ and minimum inverter delay.

From the analysis in Section 5.4.1 and the measurement results in Figure 5.31, the ring oscillator design procedure is as follows:

1. Find the optimum L by noise simulation or using the prediction technique developed in this chapter.

2. Simulate the 3-stage ring with L set to the $L_{optimum}$. Check if the oscillating speed is higher than the design goal. The tuning range should be as close to the maximum speed as possible since tuning will degrade κ.

3. If the 3-stage ring is fast enough, $L_{optimum}$ should be used for the oscillator. The desired speed can be realized by adjusting the number of stages N or adding capacitive load. Both methods will not affect κ, since κ is independent of the N and load capacitance C_L.

4. If the 3-stage ring is not fast enough, set the number of stages N to 3. Then decreasing L until the speed of the 3-stage ring meets the design goal.

5. Increase the channel width W until the desired κ is met when the oscillator runs at the lowest speed in the tuning range. This will not affect the oscillating speed much since the inverter propagation delay has a very weak dependency on W.

Using the optimum length in VCO design will only guaranty minimum power dissipation for single-ended ring oscillators. The die area is usually not the minimum since more stages may be needed to lower the speed of the 3-stage ring. For differential ring oscillators, insert more stages is not a good idea since each stage will consume power no matter it is switching or not due to the tail current. Therefore carefully select channel length and weighing the tradeoffs between speed, jitter performance, power dissipation and die area is critical in ring oscillator design.

5.7.2 Design for Normalized RMS Jitter

From (5.64), the normalized rms jitter is

$$(\sigma_x)_{UI} = \frac{\sigma_x}{T_0} = \frac{\kappa}{T_0}\sqrt{\frac{1}{4\pi f_L}\left(1+\frac{4f_c}{3\sqrt{3}f_L}\right)} \quad [\mathrm{UI}] \tag{5.122}$$

From the analysis in this chapter, the k-to-period ratio is proportional to

$$\frac{\kappa}{T_0} \propto \frac{1}{N\sqrt{WL^3}} \tag{5.123}$$

Therefore, the PLL and ring oscillator design procedure is as follows:

1. If the designer has the freedom in the PLL loop bandwidth $f_{L,}$ increased it as high as possible. Higher f_L not only helps to filter out the 1/f noise contribution but also reduce the normalized rms jitter according to (5.122). Then the required k-to-period ratio can be calculated from the jitter requirement by (5.122).

2. Set the number of stages N to 3, and find maximum L at which the oscillating speed meets the design goal. This L should be used for the ring oscillator. The tuning range should be as close to the maximum speed as possible since tuning will degrade κ. Increasing L is better than increasing N to achieve better normalized rms jitter. The reason is that the oscillating frequency f_0 is inversely proportional to N and L for the case of short channel; from (5.123) the k-to-period ratio is inversely proportional to L to the power of 1.5, while just inversely proportional to N. For the long-channel case, larger N can be considered since f_0 is inversely proportional to L^2. But larger L will help to lower the 1/f noise too.

3. Increase the channel width W until the desired k-to-period ratio is met when the oscillator runs at the lowest speed in the tuning range. This will not affect the oscillating speed since the dependency of inverter delay to W is very weak.

5.7.3 Design for Normalized Cycle-to-cycle Jitter

From (5.87), the normalized cycle-to-cycle jitter is

$$\left(\sigma_{T_0}\right)_{UI} = \sqrt{\frac{8kT\gamma_s}{5N\left(1+m\right)C_{ox}V_{DD}\left(V_{DD}-V_T\right)}\cdot\frac{1}{WL}}\sqrt{\frac{L+L_c}{L}+\delta} \quad \left[\mathrm{UI}\right] \quad (5.124)$$

From the analysis in Section 5.4.3, the normalized cycle-to-cycle jitter is proportional to

$$\left(\sigma_{T_0}\right)_{UI} \propto \begin{cases} \dfrac{1}{\sqrt{NWL}} & \text{when } L \gg L_c \\ \dfrac{1}{\sqrt{NWL^2}} & \text{when } L \ll L_c \end{cases} \quad (5.125)$$

Therefore, the ring oscillator design procedure is as follows:

1. Set the number of stages N to 3, and find the maximum L at which the oscillating speed meets the design goal. And this L should be used for the ring oscillator. The tuning range should be as close to the maximum speed as possible since tuning will degrade κ. Increasing L is better than increasing N to achieve better normalized rms jitter, since from (5.125) the normalized cycle-to-cycle jitter is inversely proportional to L, while just inversely proportional to square root of N at short channel lengths. And larger L will also help to lower the 1/f noise.

2. Increase the channel width W until the desired normalized cycle-to-cycle jitter is met when the oscillator runs at the lowest speed in the tuning range. This will not affect the oscillating speed since the inverter propagation delay has a very weak dependency on W.

Chapter 6: Design of Digital PLL-based True Random Number Generator

6.1 Introduction

Almost all cryptographic applications require the generation of random numbers as secret keys, starting states, or other secret quantities [4], [96]-[98]. The quality of the randomness is essential since security protocols rely on the irreproducibility and unpredictability of the random numbers they use. Therefore, the random number generator (RNG) for cryptographic applications must meet stringent requirements.

There are two kinds of RNGs, pseudo-random number generators (PRNGs) and true random number generators (TRNGs). PRNGs can be implemented by both software and hardware. The numbers generated by PRNGs are not truly random, but are able to approximate some of the properties of random numbers. Complex mathematical functions are often used to generate high-quality pseudo-random bit streams. The classic algorithms include linear congruential generators, lagged Fibonacci generators, linear feedback shift registers and generalized feedback shift registers [100]. These algorithms are mostly insecure and easily predictable. Recent instances of algorithms include Blum Blum Shub [101], Fortuna [102], and the Mersenne twister [103].

However, there are also many well-documented ways to attack systems that utilize the PRNG approach [104]. A well-known remark by John von Neumann emphasized this: “Anyone who considers arithmetical methods of producing random digits is, of course, in a state of sin.”

TRNGs can only be implemented by hardware. Popular approaches utilize electrical noise from a resistor or semiconductor devices as the source of randomness [105]. The numbers generated by ideal TRNGs are pure random and can not be predicted. However in the real world, there is always some kind of correlation between the data bits of the TRNG output. There are many statistical tests for the randomness. The most popular ones are the NIST SP800-22 test suite [106] and the Diehard battery test [107].

The target application for the TRNG designed in this work is the “smart card” [108]. The smart card resembles a credit card in size and shape with an embedded microprocessor which is under a gold contact pad on one side of the card. Compared to the magnetic strip technology which is still widely used in the United States, smart cards are able to provide more security since the data stored on the magnetic stripe can easily be read, written, deleted, or changed with off-the-shelf equipments. Therefore smart cards are better media to store sensitive information, and a good replacement to the usual magnetic stripe on a credit or debit card. Other applications for smart cards are computer security systems, wireless communication and government identification.

Smart cards are usually powered by a card reader, and may have up to 8 kilobytes of RMA, 346 kilobytes of ROM, 256 kilobytes of programmable ROM, and a 16-bit microprocessor. The smart-card reader is usually attached to a personal computer and communicates with the smart card through a serial interface. To authenticate the card and terminal, a built-in RNG for key generation is required [99]. Usually the key will not be regenerated after every transaction, but after a certain period such as a month. Since the users do not care to wait one more second for the key generation, the quality of the RNG is far more important than its speed.

6.2 Review of TRNG Design Techniques

There are three common ways to implement a TRNG: direct amplification, discrete-time chaos such as metastablility [98] and oscillator-sampling [4]-[6].

As illustrated in Figure 6.1, direct amplification uses a high-gain, high-bandwidth amplifier to process voltage changes produced by noise sources such as a resistor or a diode [109]. Then the amplified noise is sampled by the comparator to generate random number bits. Since the thermal noise from a resistor is in the order of μV, this type of RNGs is very sensitive to signal coupling, such as deterministic noise sources from the power supply and the substrate [110]. Another drawback is the power dissipation due to the requirement of the high-gain high-bandwidth amplifier.

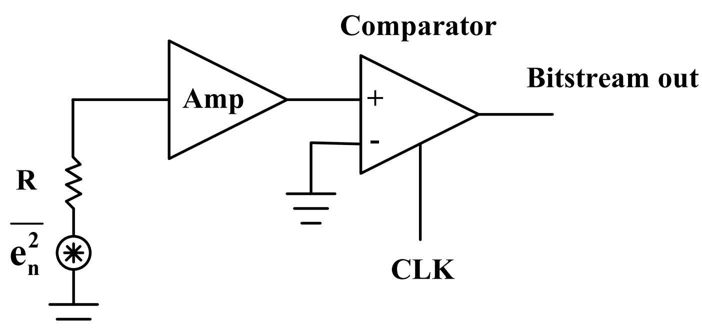


Figure 6.1 RNG implemented by direct amplification.

Chaos-based RNGs use analog signal-processing techniques and the randomness is obtained from robust dynamics. Traditional designs using this technique provide good randomness but require complicated circuits using large area and high power while only low data rates can be obtained [110].

The most popular approach by far is the method of oscillator-sampling due to the advantages of less die area, improved power efficiency, and high speed. As illustrated in Figure 6.2, a low frequency oscillator with high jitter samples the output of a high frequency oscillator using a D flip-flop to produce the random-number sequences. Post-processing circuits are usually required to improve the randomness. In order to achieve high level randomness, the rms jitter of the low frequency oscillator must be much greater than the period of the fast oscillator [5]. Experimental results in [5] have shown that for CMOS ring oscillators in a 0.18μm digital library, the jitter-to-mean-period ratio is less than 10^{-4}, which limits the maximum output data rate to 100kb/s if a 1GHz fast oscillator is used.

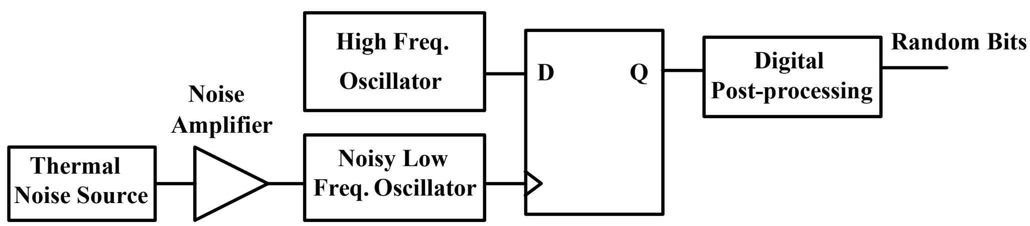


Figure 6.2 RNG implemented by oscillator-sampling.

To overcome this problem, more noise sources are mixed to the low speed oscillator to obtain a high jitter-to-mean-period ratio. The RNG in [5] uses an OPAMP with 45dB gain and 40MHz bandwidth to amplify the thermal noise of resistors and modulate the low speed oscillator. A 10Mbps throughput is obtained when the low speed oscillator runs at 10MHz and the high frequency oscillator runs at 1GHz.

Even with the noise modulation, the speed ratio of these two oscillators still needs to be above 100:1 [4]. Therefore, the RNG using the structure in Figure 6.2 usually requires that the high frequency oscillator runs above 1GHz [5], [110]. This requires relatively expensive processes. In this chapter, a new dual-oscillator sampling architecture for random number generation is proposed. The main advantage over the previous designs is the capability of achieving comparable data rate using slower clocks, thus cheaper process and lower cost.

The architecture of the proposed RNG is introduced in Section 6.3. Section 6.4 perform system analysis in the time domain and illustrate Matlab simulation results. The RNG components, such as the DAC-controlled ring oscillators, the low metastability D flip-flop, and the up/down counters, will be discussed in detail in Section 6.5, 6.6, and 6.7 respectively. The digital post-processing circuits are presented in Section 6.8 and Section 6.9 gives the experimental results.

6.3 System Architecture

The architecture of the proposed digital PLL-based RNG is illustrated in Figure 6.3. Two identical noisy ring oscillators are designed with white noise dominated jitter. Oscillator I is free-running and serving as the clock. The phase error of the two oscillators is sampled by a low metastability D flip-flop, which also acts as a bang-bang phase detector. Two up/down counters form the loop filter. The 24-bit up/down counter p integrates the phase error of the two ring oscillators to set the average frequency of oscillator II, and introduces a pole to the loop transfer function. The 1-bit up/down counter z introduces the zero to stabilize the loop, and provides instantaneous phase correction without affecting the average oscillating frequency. Therefore the two oscillators are always synchronized through the feedback. The whole system is powered by a voltage regulator to reject the noise from the power supply. It should be noted that the whole system is nonlinear and thus it is difficult to be modeled analytically.

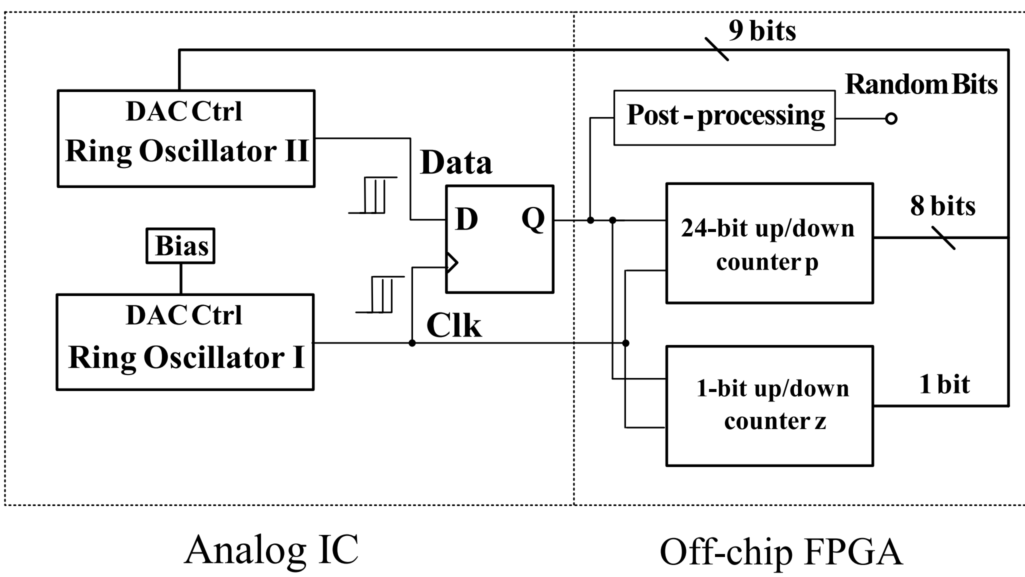


Figure 6.3 Architecture of the digital PLL-based RNG.

From the analysis in Chapter 4, the PLL acts as a low-pass filter for the input phase and a high-pass filter for jitter of the local oscillator. If the loop bandwidth f_L is wide enough, most jitter of Oscillator I will pass through to the PLL output. The 1/f noise upconverted jitter in Oscillator II will be filtered out. Therefore, the jitter difference of the two oscillators is the filtered jitter of Oscillator II, which is correlated white noise. After sampling by the D flip-flop, a correlated data stream with equal probability for '1's and '0's is generated.

The closed-loop spectrum of white noise upconverted phase noise has the form [7]

$$S_{\Phi CL}(f) = \frac{N_1 / f_L^2}{1 + (f / f_L)^2} \tag{6.1}$$

By the Wiener-Khinchin theorem, the autocorrelation function of this jitter process can be obtained by taking inverse Fourier transform of its power spectrum density in (6.1). As illustrated in Figure 6.4, the autocorrelation coefficient of this jitter process is calculated as [8]

$$\rho_{xx}(\Delta t) = \exp\left(-2\pi f_L |\Delta t|\right) \tag{6.2}$$

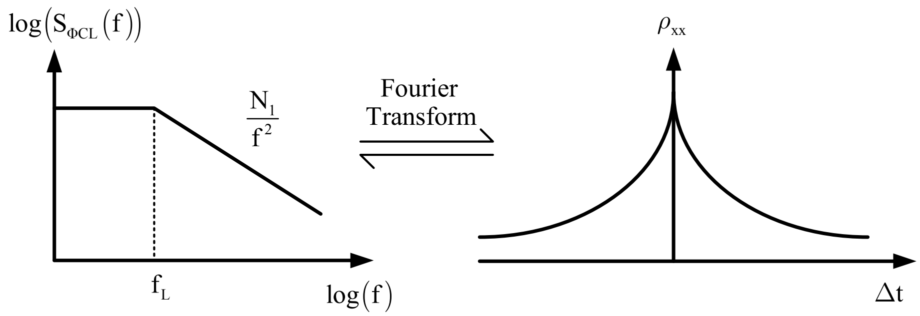


Figure 6.4 Autocorrelation and p.s.d.

From equation (6.2), the autocorrelation coefficient is down to 0.2% when Δt equals $1/f_L$. Thus the autocorrelation of the output data can be significantly reduced by dividing the output data rate down to around or below the PLL loop bandwidth f_L, so that the data stream can be considered as random. Therefore for the proposed RNG, the maximum data rate achievable is limited by the loop bandwidth of this system. Since the PLL loop bandwidth can be as high as 10% of the clock frequency [93], ideally the maximum RNG data rate can be as high as 10% of the ring oscillator frequency.

However, wider loop bandwidth results in lower PLL output jitter. From [7], the rms jitter with respect to the reference clock for white noise dominated PLL is

$$\sigma_x = \kappa\sqrt{\frac{1}{4\pi f_L}} \tag{6.3}$$

Therefore the rms jitter is inversely proportional to the square root of f_L:

$$\sigma_x \propto \frac{1}{\sqrt{f_L}} \tag{6.4}$$

The rms jitter of this digital PLL should be much larger than the LSB of the DAC, or the phases of the two ring oscillators will not be synchronized but oscillate. Thus the key factors of this design are the DAC resolution, the PLL loop bandwidth f_L, and the noise figure of merit κ.

The goal of this design is to implement a TRNG with data rate of 1Mbps. From the analysis above, the loop bandwidth of this PLL system should be 1MHz or higher, which requires that the oscillator's speed is faster than 10MHz. To give more room to adjust the loop, the actual oscillators are designed to run at 30MHz.

6.4 Time Domain Analysis and System Simulation

From the analysis in last section, oscillator I is free running with a fixed frequency. The speed of oscillator II is continually adjusted by the loop so that it is synchronized to oscillator I. As illustrated in Figure 6.5, in the presence of jitter and assuming that there is no frequency drift, the transition time for the two white noise dominated oscillators can be expressed as

$$t_1[n] = t_1[n-1] + T_1 + \varepsilon_1[n] \tag{6.5}$$

$$t_2[n] = t_2[n-1] + T_2[n] + \varepsilon_2[n] \tag{6.6}$$

where T_1 is the constant average period of the free-running oscillator I, $T_2[n]$ is the average period of the oscillator II in the n^{th} interval, and $\varepsilon_i[n]$ is the jitter accumulated within the n^{th} interval which expresses the deviation of the period from the average.

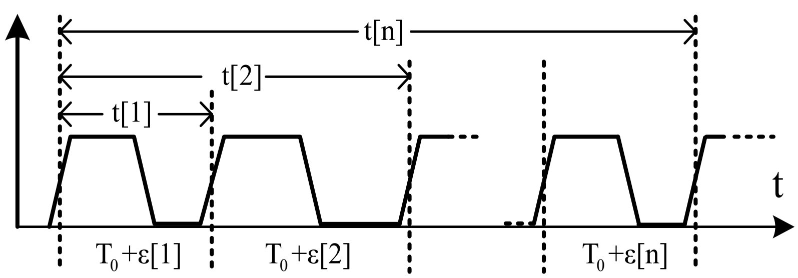


Figure 6.5 Definition of random processes for clock with jitter.

The terminals shared by the two oscillators are power supply, ground, and substrate. The noise from these terminals will introduce deterministic jitter which is in common mode to the two oscillators, and most of them will be rejected since the two oscillators are identical and laid out next to each other. By carefully sizing the transistors using the technique developed in last chapter, the ring oscillators are designed with white noise dominated jitter. Thus ε_i can be approximated as zero-mean, uncorrelated, discrete Gaussian random processes. Since the thermal noise of each MOSFET and resistor is generated independently, the random processes ε_1 and ε_2 are independent too.

From the analysis in Chapter 3, the standard deviation of cycle-to-cycle jitter $\varepsilon_i[n]$ is

$$\sigma_{\varepsilon i} = \kappa_i \sqrt{T_i} \quad i = 1, 2 \tag{6.7}$$

The transition time for the two oscillators in equation (6.5) and (6.6) can be rewritten by

$$t_1[n] = t_1[0] + nT_1 + \sum_{j=1}^{n} \varepsilon_1[j] \tag{6.8}$$

$$t_2[n] = t_2[0] + \sum_{j=1}^{n} T_2[j] + \sum_{j=1}^{n} \varepsilon_2[j] \tag{6.9}$$

From probability theory, a sum of Gaussian random variables is also Gaussian, and the random variable (A+B) is correlated to both random variables A and B. Thus $t_1[n]$ is a Gaussian random variable with mean of nT_1 and the transition time sequence t_1 is a correlated Gaussian random process. The standard deviation of the random variable $t_1[n]$ is

$$\sigma_{t_1}[n] = \sqrt{\sum_{j=1}^{n} \sigma_{\varepsilon_1}^2[j]} = \kappa_1 \sqrt{nT_1} \tag{6.10}$$

As illustrated in Figure 6.6, assuming there is no metastability for the D flip-flop, the phase error sequence rb[n], which is also the output sequence of this RNG system, is as

$$rb[n] = \left(sign\left(t_{diff}[n] \right) + 1 \right) / 2 \tag{6.11}$$

where sign() is the signum function, and $t_{diff}[n]$ is

$$t_{diff}[n] = t_1[n] - t_2[n] \tag{6.12}$$

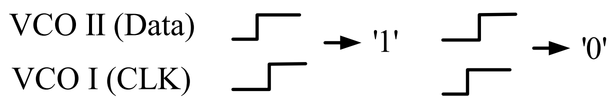


Figure 6.6 Data sampling by the D flip-flop.

When the PLL is in lock, the two oscillators are synchronized together. Therefore $t_{diff}[n]$ is the PLL rms jitter with respect to the reference clock. From the analysis in [8], the random process t_{diff} is zero-mean, correlated Gaussian random process with correlation coefficient in (6.2) and standard deviation in (6.3). As long as $t_{diff}[n]$ behaves as a zero-mean Gaussian random variable, the probability of $t_{diff}[n]>0$ equals that of $t_{diff}[n]<0$ as shown in Figure 6.7. And the RNG will generate a sequence of unbiased ‘1’s and ‘0’s.

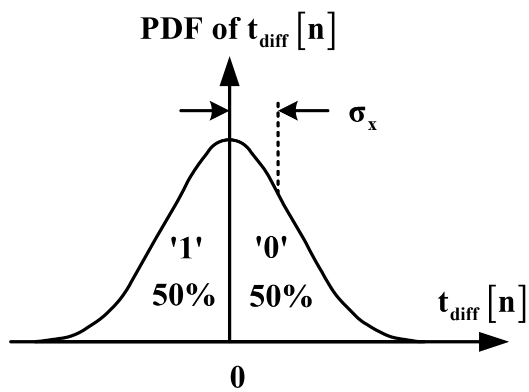


Figure 6.7 Sampling a Gaussian random variable.

The frequency of oscillator II is adjusted in every clock cycle. From Figure 6.6, if the speed of the oscillator I is faster than that of the oscillator II, there will be more ‘1’s than ‘0’s at the output of the D flip-flop. The up/down counter records this information of imbalance, and the loop will decrease the speed of the oscillator I until equilibrium is reached. Thus the frequency of oscillator II can be modeled as

$$f_2[n+1] = f_2[1] - \left(k_p \cdot ctr_p[n] + k_z \cdot ctr_z[n]\right) \cdot K_{vco} \tag{6.13}$$

where k_p and k_z are the gain of the counters p and z, $ctr_p[n]$ and $ctr_z[n]$ are their counting results, and K_{vco} is the VCO gain factor in the unit of Hz/bit.

In the design implementation, the eight most significant bits of the 24-bit counter p are connected to the DAC control of the oscillator I, while the output of 1-bit counter z is connected to the DAC directly. So equation (6.13) changes to

$$f_2[n+1] = f_2[1] - \left(fix\left(ctr_p[n] / 2^{16} \right) + ctr_z[n] \right) \cdot K_{vco} \tag{6.14}$$

where fix(A) is the Matlab fix function which rounds the elements of A toward zero, resulting in an array of integers.

The behavior of this PLL system is simulated by Matlab. During simulation, oscillator I is free-running at 30MHz. Oscillator II is running at 29.9MHz and 20ns ahead of oscillator I. The LSB of the DAC is 20kHz/bit, which is equivalent of 22ps resolution. The cycle-to-cycle jitter of both oscillators is 60ps. A 16-bit up/down counter p and 1-bit up/down counter z are used in simulation.

Figure 6.8 shows the simulated loop acquisition process. It takes about 5000 cycles, which is 167μs, for the two oscillators to be synchronized. Figure 6.9 shows the output data bits, the autocorrelation coefficient, and the spectrum of the RNG output. The autocorrelation coefficient is as high as 62% for the adjacent bits.

To lower the autocorrelation, the simplest way is to divide the data rate down. Figure 6.10 shows simulated results for dividing the RNG output down by 20. The autocorrelation coefficient is reduced to below 7% for the adjacent bits.

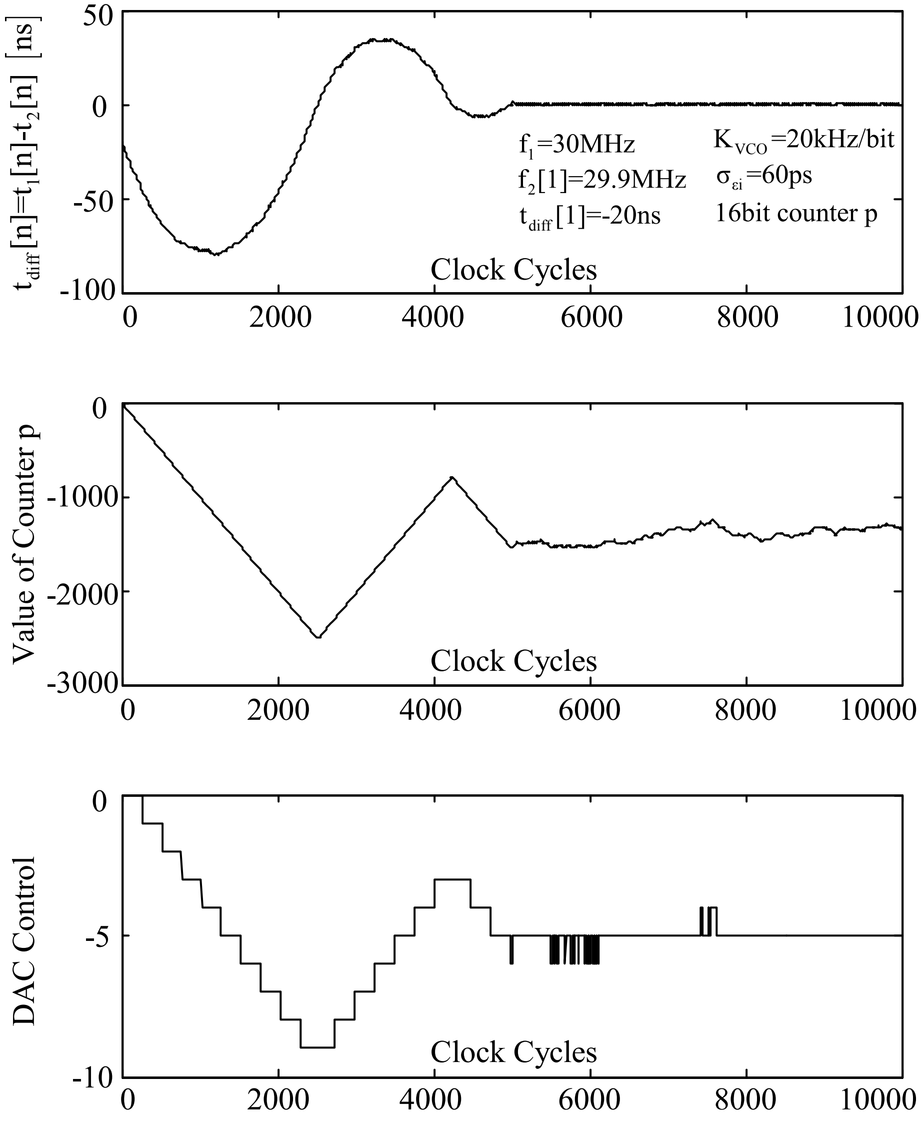


Figure 6.8 System behavior simulated by Matlab.

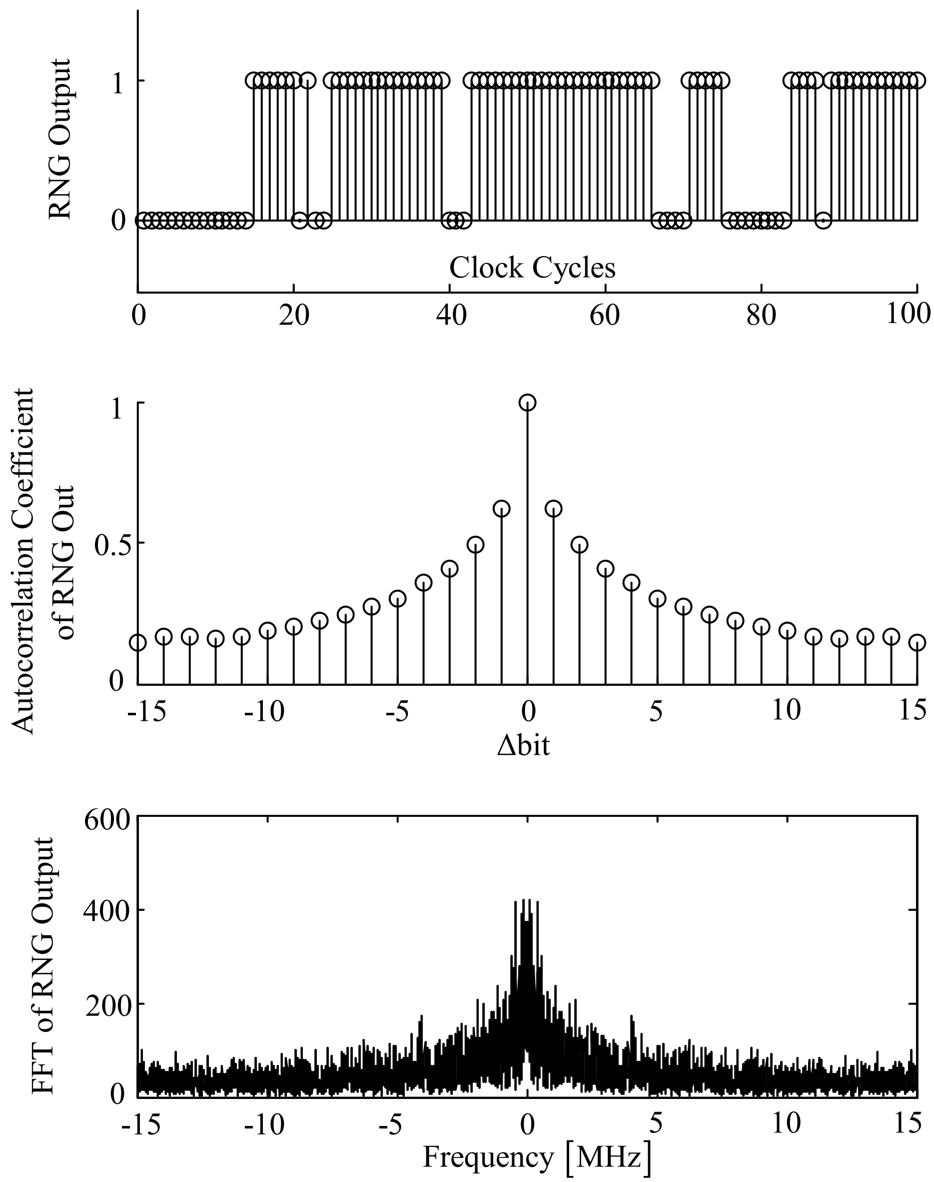


Figure 6.9 RNG output simulated by Matlab.

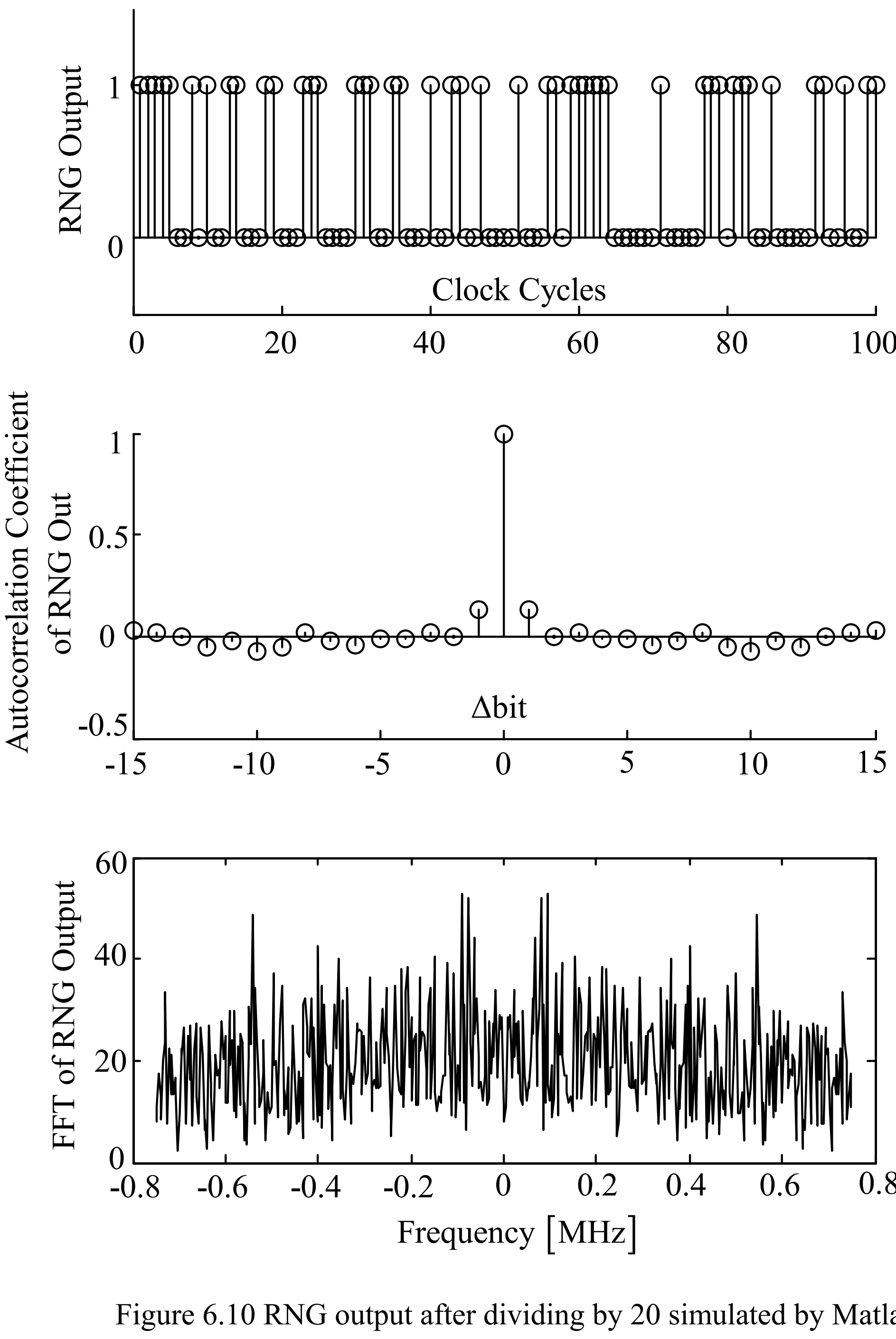


Figure 6.10 RNG output after dividing by 20 simulated by Matlab.

6.5 DAC-controlled Ring Oscillator

The ring oscillators in the standard digital libraries are all optimized for both noise performance and speed. Therefore customized ring oscillators are needed in this design to provide high jitter.

The DAC-controlled ring oscillator is realized by adding a capacitor array to the load of the 3-stage single-ended ring oscillator as illustrated in Fig. 6.11. The delay stage is the current-starved inverter discussed in last chapter. The two ring oscillators are designed to be noisy. They are totally symmetric and next to each other in the layout. Thus the deterministic jitter due to the power supply and the substrate is in common-mode and will be rejected.

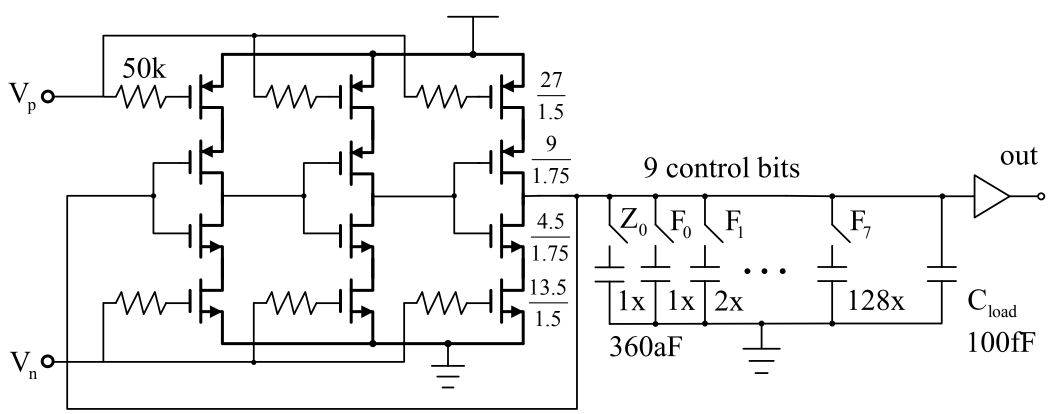


Figure 6.11 DAC-controlled ring oscillator.

The design considerations for the DAC-controlled ring oscillator are:

1. Relatively high speed. The loop bandwidth should be no higher than 10% of the oscillator speed. Therefore the oscillator frequency decides the maximum output data rate. In order to get 1Mb/s throughput, the speed is set to be 30MHz.

2. High κ to provide enough jitter from (6.3). This will relax the resolution requirements for the DAC and the D flip-flop.

3. DAC resolution. If the DAC LSB is not much smaller than the PLL rms jitter, the phases of the two ring oscillators will oscillate.

4. Adjustable range of VCO. This will decide the locking range, which is the maximum frequency mismatch that this PLL system can tolerate.

6.5.1 Sizing Transistors

Since this structure is implemented in AMI 1.5μm process, the analysis in Chapter 5 for the case of long-channel is applicable to this design. The two oscillators are powered by a voltage regulator which converts the 5V V_{DD} to 2.5V. The purpose is to reject the deterministic jitter from the power supply and limit the power dissipated in the oscillators, thus generating more jitter.

From Section 5.6.1, the speed of ring oscillators formed by the current-starved inverter is

$$f_0 = \frac{I_{ctl}}{NV_{SW}C_L} \tag{6.15}$$

Therefore,

$$f_0 \propto \frac{1}{N} \cdot \frac{I_{ctl}}{W_{sw}L_{sw}} \tag{6.16}$$

When the tuning transistors are in saturation, the KTC noise is negligible, and the noise figures of merit are

$$\kappa_{sw} = \frac{i_n}{\sqrt{\Delta f}} \cdot \frac{1}{\sqrt{2}I_{ctl}} = \sqrt{\frac{16kT}{3\mu C_{ox}} \cdot \frac{W_{sw}}{W_{ctl}^2} \cdot \frac{L_{ctl}^2}{L_{sw}} \cdot \frac{\left(V_H - V_T\right)}{\left(V_{ctl} - V_T\right)^4}} \tag{6.17}$$

$$\kappa_{ctl} = \frac{i_n}{\sqrt{\Delta f}} \cdot \frac{1}{\sqrt{2}I_{ctl}} = \sqrt{\frac{8}{3}kT \cdot \frac{1}{\frac{1}{2}\mu C_{ox}\frac{W_{ctl}}{L_{ctl}}\left(V_{ctl} - V_T\right)^3}} \tag{6.18}$$

So

$$\kappa_{total} \propto \frac{\sqrt{W_{sw} / L_{sw}}}{I_{ctl}} \tag{6.19}$$

From (6.16), at the required speed of 30MHz, the ratio between I_{ctl} and W_{sw} should be constant. From (6.19), decreasing I_{ctl} is better than increasing W_{sw} to maximize κ_{total}. Therefore, W_{sw}, N, and L_{sw} should use the minimum value possible to minimize I_{ctl} and die area, which is equivalent to minimize power dissipation. Based on this analysis, W_{sw} is set to be 4.5µm, which is the minimum width recommended by MOSIS for AMI 1.5µm process; N is set to be 3, which means the VCO is a 3-stage ring. L_{sw} is set to be 1.75µm, which is not the minimum value of 1.5µm. The reason is that through simulation, for L_{sw}=1.5µm, the I_{ctl} is so low that the output swing is too small.

To save die area consumed by the tuning transistors, L_{ctl} is set to be 1.5µm and W_{ctl} is set to be three times of W_{sw}. The minimum I_{ctl} to achieve the 30MHz speed is realized by tuning the control voltage V_{ctl}, which is actually biased around the threshold voltage of the tuning transistors.

Even though with all the above efforts, the κ_{total} obtained from simulation is just 8.32E-8. For the loop bandwidth of 1MHz, the PLL rms jitter is only 23.5ps. To provide more jitter, six 50KΩ resistors are added at the voltage control nodes. The reason for using six resistors is to provide independent thermal noise to each stage. The simulated κ_{total} increases to 1.25E-7. The PLL rms jitter is 35ps when the loop bandwidth is 1 MHz and 50ps when the loop bandwidth is 500 kHz.

Since the power dissipation is minimized to maximize jitter, the power dissipated by this DAC-controlled ring oscillator is only 44.25µW.

6.5.2 DAC Control

The DAC is realized by the capacitor array as shown in Figure 6.11. The control word decides the amount of loading capacitance at the output node of the ring oscillator, and thus controls the speed. The switches are implemented by NMOS transistors with the minimum size, which is 4.5µm/1.5µm.

The capacitors to form the four least significant bits in the DAC should be much less than C_{load} in Figure 6.11 to provide good linearity. The linearity for the four most significant bits is not important since their controls will only switch during loop acquisition.

The total loading capacitance should be as small as possible so that the current I_{ctl} is minimized while achieving the speed of 30MHz. The 1x capacitor in the array is implemented by metal-meta2 capacitor with the size of 3µm by 3µm, which is about 360aF and is the smallest capacitor possible in AMI 1.5µm process.

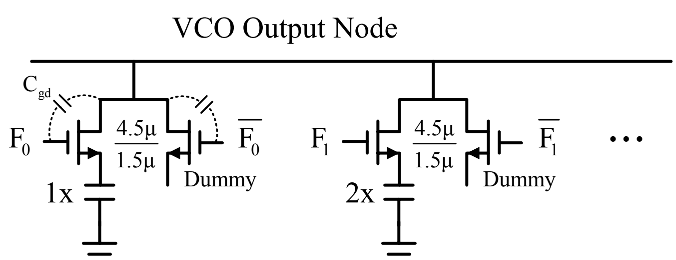


Figure 6.12 Capacitor array with dummy transistors.

As shown in Figure 6.12 for each MOSFET as a switch, when the control bit changes there will be transient current to charge or discharge the parasitic capacitor C_{gd}. This transient current will affect the VCO's speed. In order to mitigate this transient current, dummy switches with open source and complement controls are added for each capacitor in the array as illustrated in Figure 6.12.

From simulation by Cadence, the LSB the DAC is 20 kHz, which is equivalent to a timing resolution of 20ps. The total adjusting range is 3 MHz.

6.6 Low Metastability D Flip-flop

Since the edges of the oscillators are always aligned to each other, a low metastability D flip-flop is required in this system so that the D flip-flop is able to resolve itself to a valid logic level within one clock period.

Figure 6.13 shows a typical master-slave D flip-flop in the standard digital libraries. The regeneration path is a positive feedback formed by two CMOS inverters. This type of D flip-flop cannot work properly when the edges of clock and data are too close, which is known as the setup time requirement. The limitation is mainly from the low gain CMOS inverter in the regeneration path.

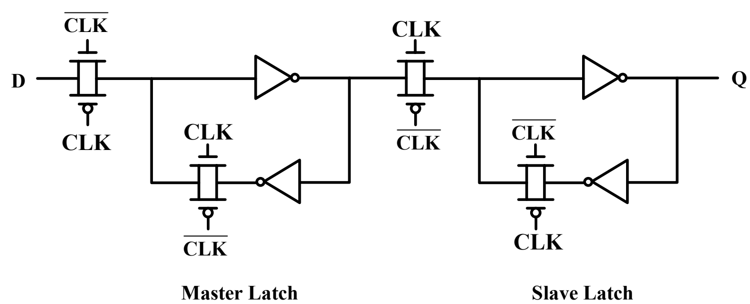


Figure 6.13 Rising edge triggered master-slave D flip-flop.

If assuming the threshold of the CMOS inverters in Figure 6.13 is at half of the power supply V_{DD}, the phase difference Δt of the two clocks in Figure 6.14 will result in an input voltage ΔV to the regeneration path in the slave latch. ΔV is related to Δt by the slope of the rising edge,

$$\Delta V = \Delta t \cdot \frac{dV}{dt} \tag{6.20}$$

If the rising time of the clocks is 2.5ns and the logic '1' is 2.5V, 1ps of Δt will result in a ΔV around 1mV. Apparently the latch in Figure 6.13 cannot process a 1mV input.

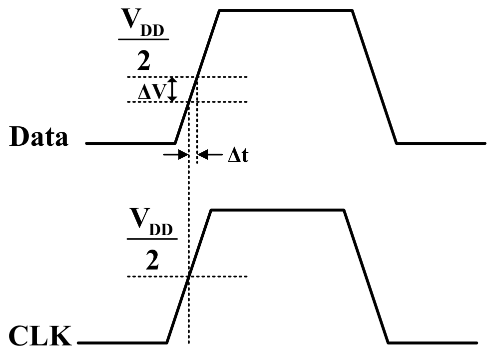


Figure 6.14 D flip-flop sampling of closely positioned edges.

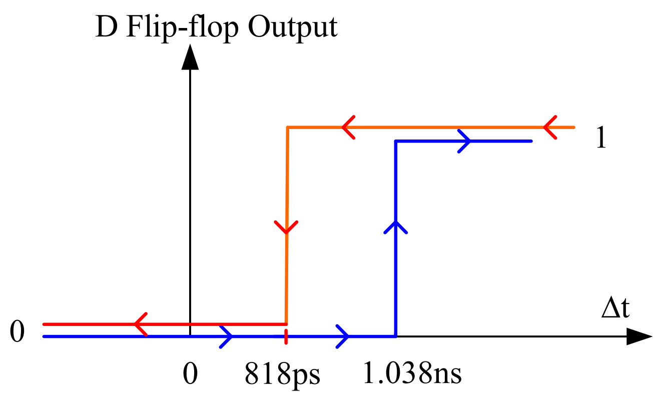


Figure 6.15 Simulation results for the master-slave D flip-flop in Figure 6.13.

Figure 6.15 shows simulated results of two clock signals being sampled by the master-slave D flip-flop designed in AMI 1.6μm process. The logic '1' in the simulation is 2.5V, and the rising time for both signals is 1ns. So the slope of the rising edges is 2.5V/ns. As shown in Figure 6.15, the setup time needed for this D flip-flop is 1.038ns, which is larger than the 1ns rising time. Therefore the latch formed by two inverters needs input voltage of 2.5V to resolve itself. There is also a 220ps gap during which this D flip-flop can not resolve itself to the correct output. This indicates that this D flip-flop needs extra overdrive to overcome its initial condition. Unless the clock jitter is much larger than this gap, this type of D flip-flops should not be used in this proposed PLL-based RNG, or the RNG output will be always series of '1's and series of '0' following each other.

To solve the problems described above, a falling edge triggered D flip-flop is designed in this work as illustrated in Fig. 6.16. When the CLK is high, both the pre-amplifier [111] and the D latch [111] will be reset. The transmission gate is on and the data is sampled. The reset of both the pre-amplifier and the D latch enables the D flip-flop to have the same initial condition every time when the regeneration starts. This 'fresh' start solves the gap problem of the typical master-slave D flip-flop illustrated in Figure 6.15. When the CLK switches to low, the transmission gate is closed and the gate capacitance of the pre-amplifier holds the sampled data. The pre-amplifier amplifies the difference between the held data and half of the power supply $V_{DD}/2$, and the D latch regenerates this amplified difference to a valid logic level. The reference voltage $V_{DD}/2$ is provided by another voltage regulator to convert the main regulator output 2.5V to 1.25V. To reduce the metastability error, two D flip-flops are cascaded.

The schematics of the pre-amplifier and the D latch are illustrated in Figure 6.17 and 6.18.

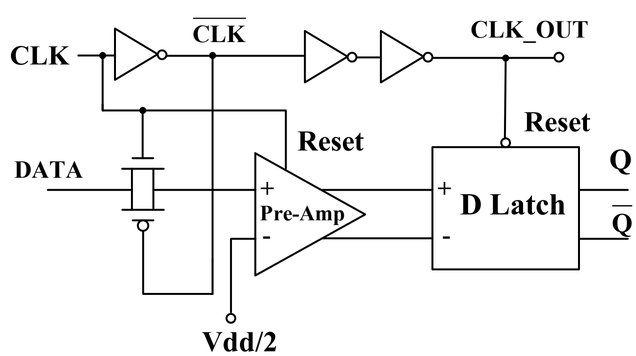


Figure 6.16 The low metastability D flip-flop.

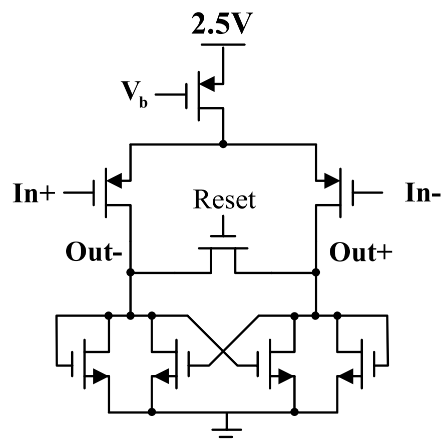


Figure 6.17 The pre-amplifier in the designed D flip-flop.

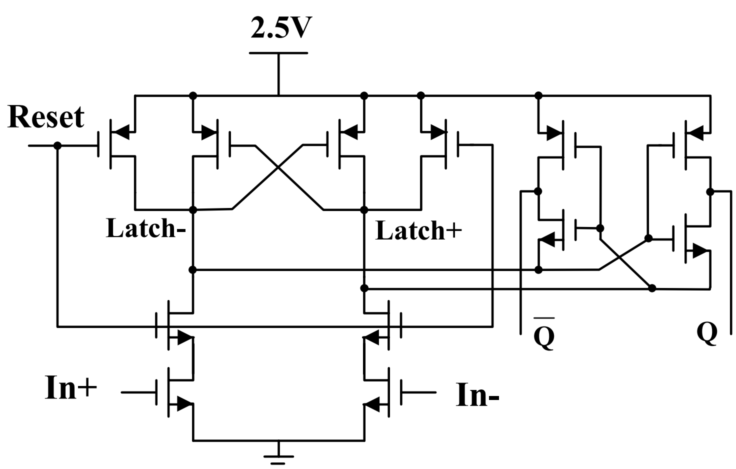


Figure 6.18 The D latch in the designed D flip-flop.

As shown in Figure 6.14 and equation (6.20), sharp clock and data edges will relax the resolution requirement of the D flip-flop. In this design, the output signals of the two ring oscillators are buffered by a three-stage inverter chain. The resulting waveforms have rise and fall times of 1ns and are fed into the D flip-flop. According to (6.20) with V_{DD} of 2.5V, 1ps of edge location difference will result in a 2.5mV input to the pre-amplifier.

This D flip-flop is simulated by Cadence. Figure 6.19 shows the simulation results. During simulation, the input clock is ahead the input data signal by 1ps. And the D flip-flop successfully resolves itself to logic '1' before next cycle starts. The output clock of this D flip-flop is intentionally delayed so that its rising edge is positioned about 2ns ahead of the rising edge of the input clock. This allows the D flip-flop to have more time for regeneration. Since the output of the D latch will be reset to V_{DD} for the half cycle during which the transmission gate samples data, the outputs of this D flip-flop are return-to-V_{DD} data.

From simulation, the currents drawn from the power supply for the pre-amplifier and the D latch are 20μA each. Thus the power dissipation for this D flip-flop is only 100μW.

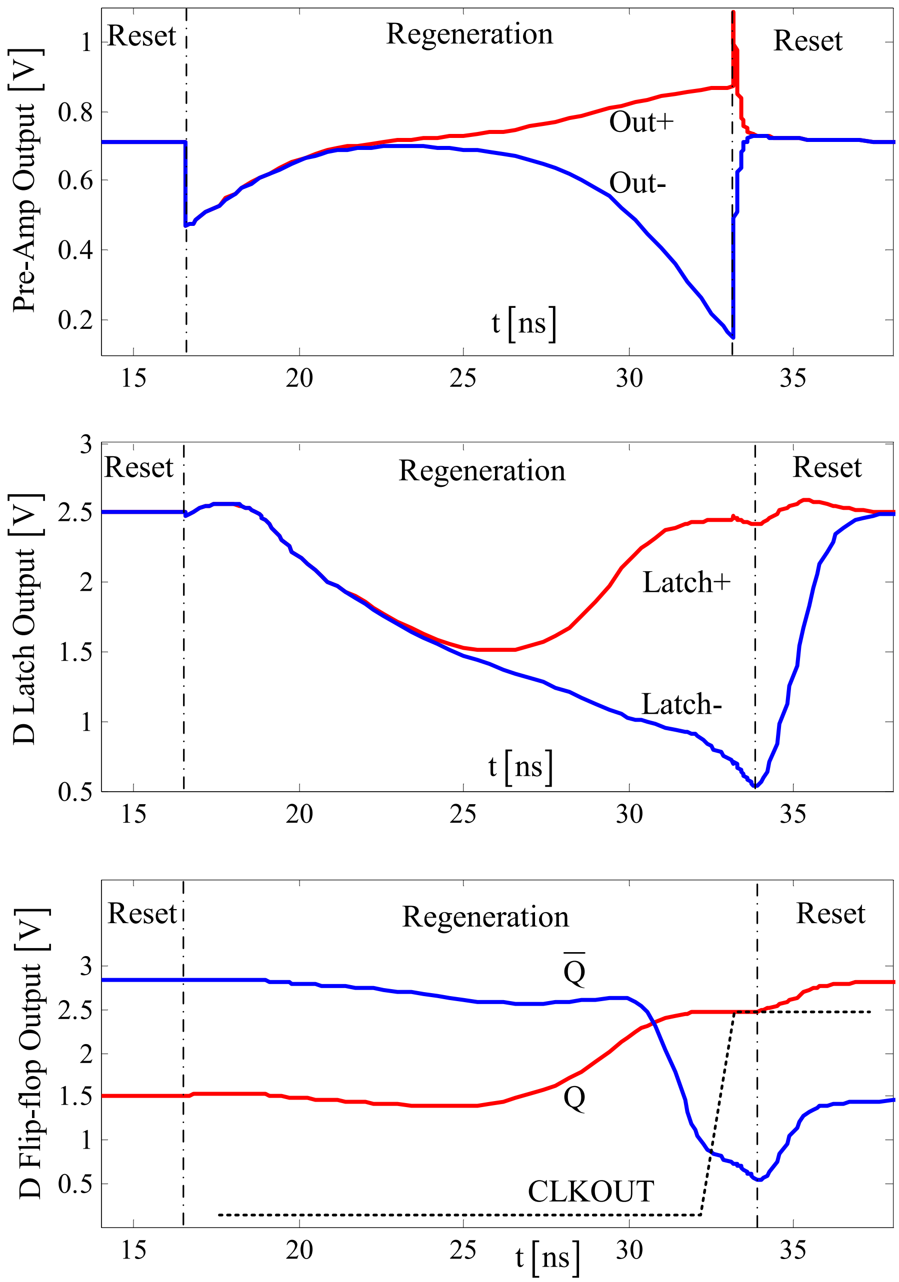


Figure 6.19 Simulation results for the designed D flip-flop.

6.7 Up/Down Counters

The up/down counters act as a loop filter. The idea is from the widely used analog loop filter in Figure 6.20.

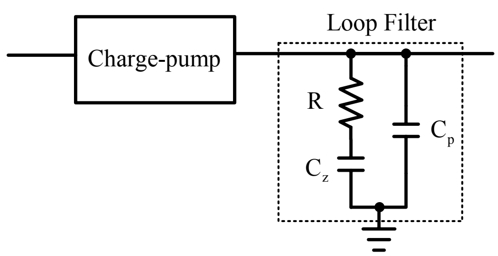


Figure 6.20 Analog loop filter.

The loop bandwidth of this digital PLL system is a function of the oscillator control constant K_{vco}, and the counters' gain, k_p and k_z. Figure 6.21 shows the loop acquisition process simulated by Matlab with different configuration of counter p. The parameters used in this simulation are shown in Figure 6.21 too. A shorter locking time indicates a wider loop bandwidth [57]. Therefore to get higher loop bandwidth, the counter p should use fewer bits.

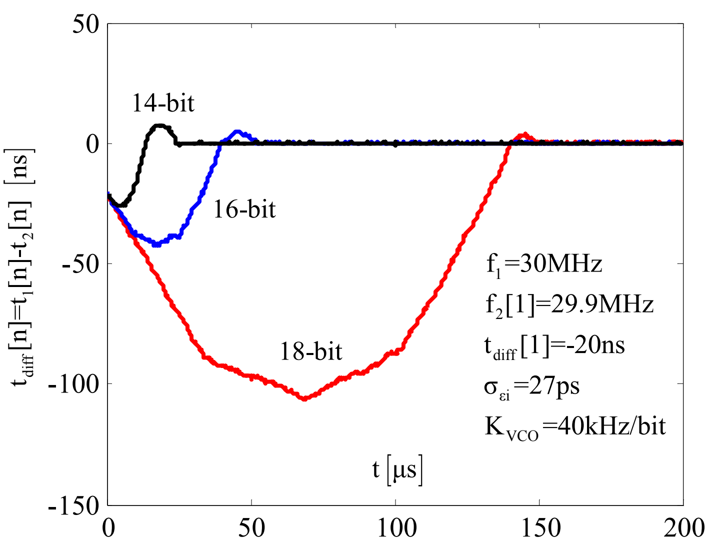


Figure 6.21 Bandwidth versus configuration of the counter p.

The counter z is important in stabilizing the loop. The simulation in Section 6.4 is rerun by taking the counter z out. Figure 6.21 shows that the system oscillates without the counter z, which means the system is not stable.

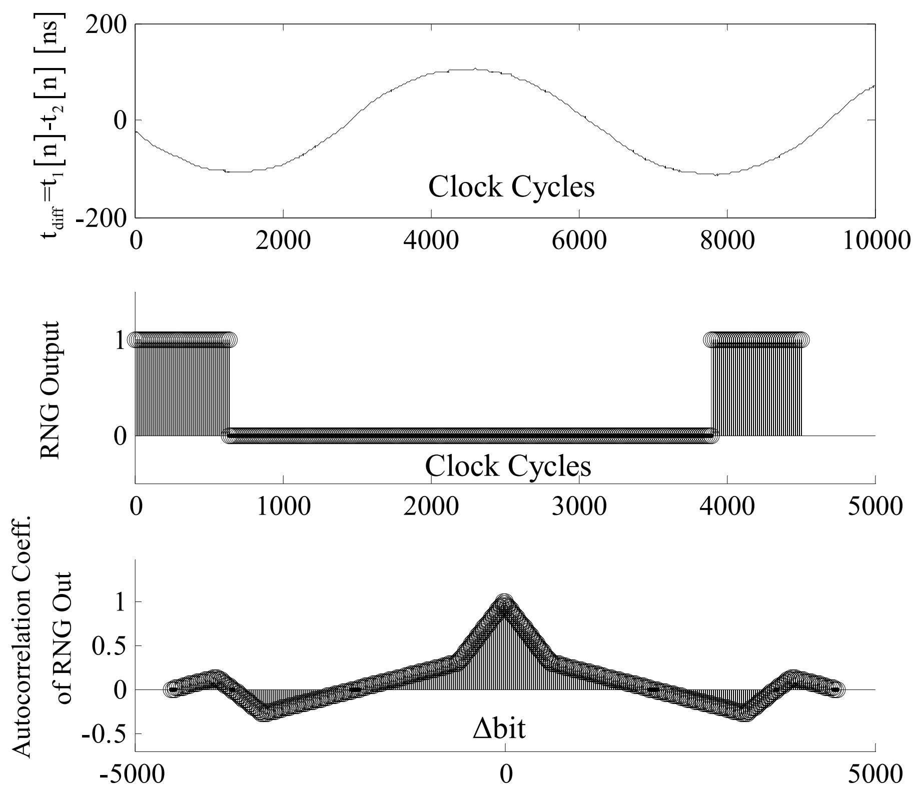


Figure 6.22 System simulation without the stabilizing zero.

6.8 Digital Post-processing

The digital post-processing circuit in the designed RNG is illustrated in Figure 6.23. As discussed in Section 6.3 and 6.4, the raw output data of this RNG is highly autocorrelated, and the autocorrelation of the RNG output can be reduced by dividing the data rate down. In order to improve the final throughput, a von Neumann corrector is inserted between two dividers to help reducing the autocorrelation and bias. The division ratio of divider I should be larger than that of divider II to provide a much less autocorrelated data input to the von Neumann corrector. The total division ratio can be estimated by the ratio of the oscillator speed and system bandwidth simulated by Matlab.

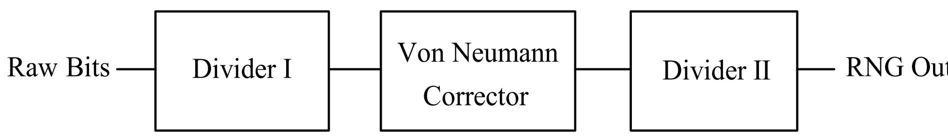


Figure 6.23 The digital post-processing circuits.

The von Neumann corrector is widely used in RNGs to reduce bias in a stream of random bits. It converts pairs of bits into output bits as illustrated in Figure 6.24. Suppose the probability of '1's is p, and there is no autocorrelation, the probability of getting '0, 1' and '1, 0' is same, which is p(1-p). Thus the bias is eliminated.

Input Bits	Output Bits
0, 0	none
0, 1	0
1, 0	1
1, 1	none

Figure 6.24 The classic von Neumann corrector.

Since the raw output data is highly autocorrelated, a relatively high bias will be introduced after divider I. The von Neumann corrector is able to dramatically reduce the bias and the remaining autocorrelation in the divided data stream, but with a high bit-drop rate.

To improve the bit-drop rate, a modified von Neumann corrector is designed and tested with this RNG. The modified algorithm monitors the bias information in real-time and records the difference D_{diff} of '1's and '0's by an up/down counter. The modified von Neumann corrector still takes successive pairs of bits. If the two bits are different use the first one as the von Neumann corrector does; if they are same and the counting result of D_{diff} is within the preset threshold D_{TH}, discard both bits; if they are same and the counting result of D_{diff} is beyond the preset threshold D_{TH}, insert one bit of '1' or '0' to the output as illustrated in Figure 6.25. Experimental results show the optimum preset threshold D_{TH} is 3900.

Input Bits	Output Bits
0, 0 or 1, 1	$\begin{cases} \text{none} & \text{if } \lvert D_{diff} \rvert \le D_{TH} \\ 0 & \text{if } D_{diff} > D_{TH} \\ 1 & \text{if } D_{diff} < -D_{TH} \end{cases}$
0, 1	0
1, 0	1

Figure 6.25 The modified von Neumann corrector.

It should be emphasized that this modified von Neumann corrector will introduce serious bias to the output, since it will automatically insert a '1' or '0' when it thinks that it has to do so. However in reality, it is possible for the RNG to output a million of ‘1’s or ‘0’s in a row, even though the probability is tiny. This inserted bias will be mitigated by the divider II.

In nature, this modified von Neumann corrector is a feedback system. It provides an option to improve the output data rate by an engineering solution with trading off the best nature of the classic von Neumann corrector, and should be used with caution.

6.9 Experimental Results

6.9.1 Test Chip Design

The customized part of this design including the oscillators, the D flip-flops, and the voltage regulators are implemented in AMI 1.5μm 2-poly 2-metal CMOS process and consume an area of $1mm^2$. The chip micrograph is shown in Fig. 6.26.

Excluding the output buffers, the total power dissipation of this customized design is 1.92mW. Half of the power is burned in the main voltage regulator. If the main regulator is implemented in the card reader, the power dissipation for this RNG system will be 0.96mW, most of which is consumed by the internal current buffer to reduce the rise and fall times of the oscillator output waveforms.

The counters and the digital post-processing circuits are implemented in an off-chip FPGA for design flexibility.

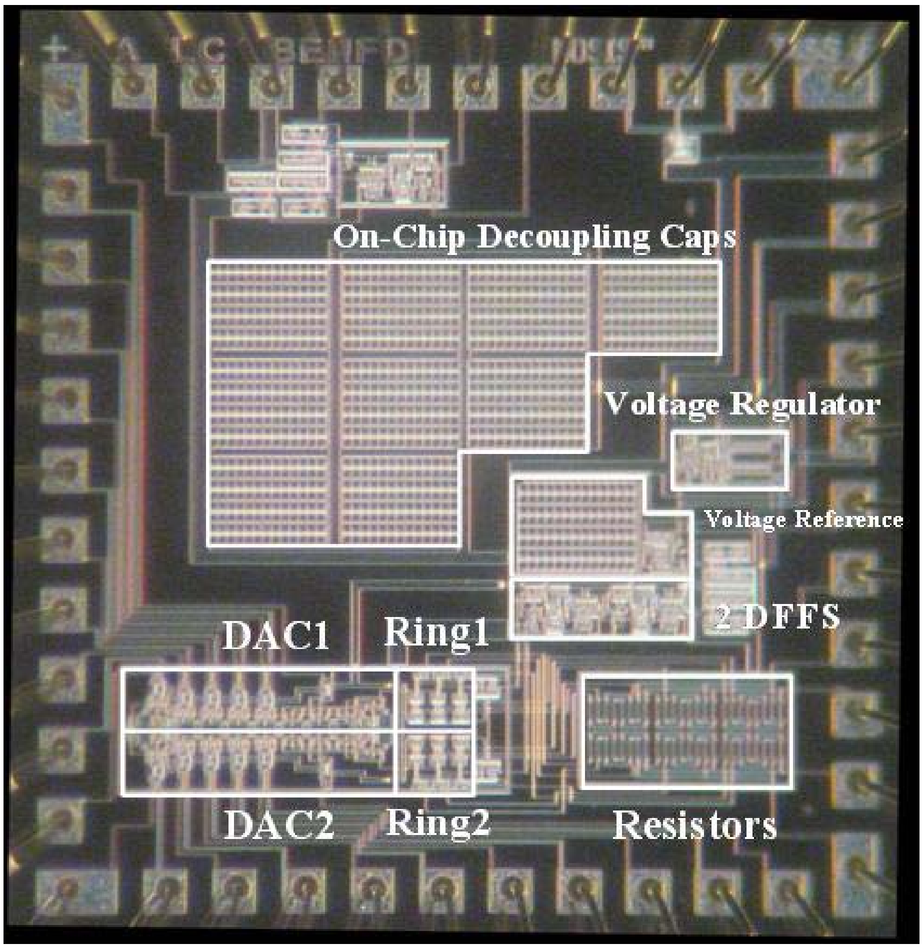


Figure 6.26 Chip micrograph of the PLL-based RNG.

6.9.2 Measurement Results

The two oscillators are running at 30MHz in this RNG system. Figure 6.27 shows the measured rms jitter over the measurement time delay for the open-loop ring oscillators. The extracted white noise figure of merit κ is 1.48E-7. The 1/f noise contribution was not optimized. The reason is that with high loop bandwidth the upconverted 1/f noise will be filtered by the loop. The 1/f noise figure of merit ς is measured as 2.54E-4. The $1/f^3$ phase noise corner is located around 100kHz.

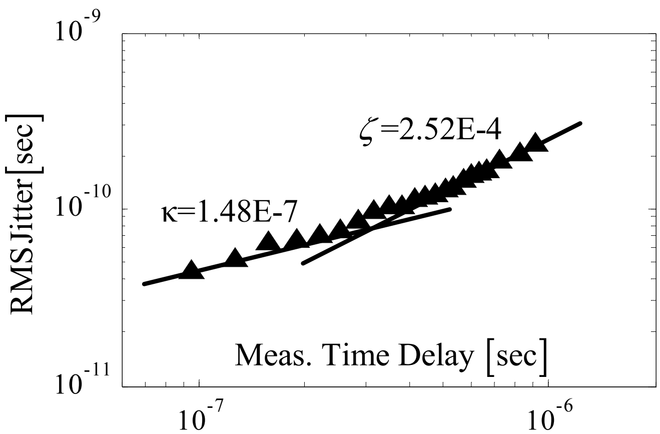


Figure 6.27 Jitter performance of the free-running ring oscillator at 30MHz.

The DAC has a measured LSB of 44ps with the total adjusting range of ±1.6ns. The resolution is not as fine as expected 20ps due to variation in the fabrication process. Due to this problem, the loop bandwidth has to be decreased to 500 kHz by using the 24-bit counter p. According to equation (6.3), the rms jitter of this system is about 60ps. Comparing to the measured LSB of 44ps, more division than expected is necessary to lower the autocorrelation sufficiently. The actual division ratio for the divider I and II in Figure 6.23 is ten and seven respectively. With the classic von Neumann corrector, a data rate of only 60 kbps is achieved. With the modified von Neumann corrector, the data rate is improved to 100 kbps. It is expected that this rate will be improved in a future iteration of the design.

Figure 6.28 shows the spectrum and autocorrelation coefficient of the post-processed data.

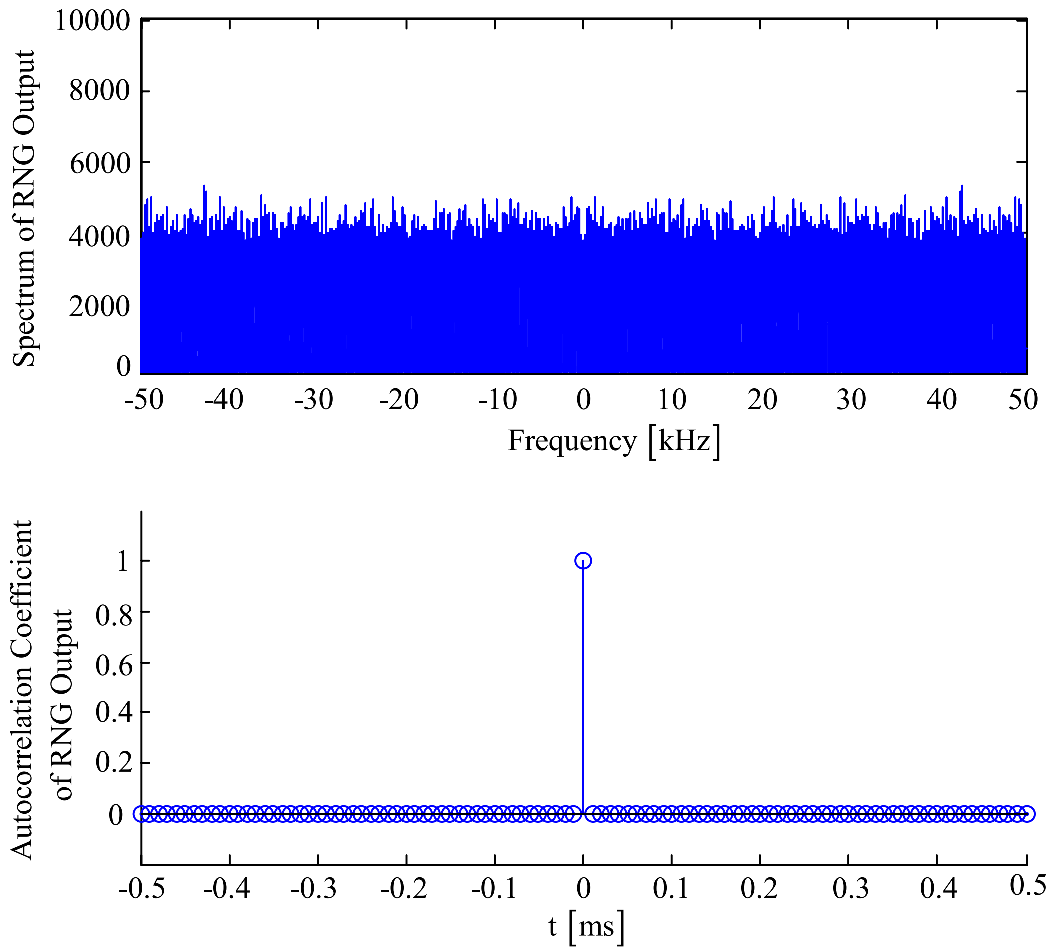


Figure 6.28 Spectrum and autocorrelation coefficient of post-processed data.

The quality of the randomness has been verified by the NIST SP800-22 test suite [106] over 2Mbit long sequences. This test suite consists of 16 statistical tests, and the passing criteria for each test is that the p-value is larger than 0.01

[106]. Table 6.1 shows the complete test results for three data sequences post-processed by the classic von Neumann. Table 6.2 shows the test results for data sequences post-processed by the modified von Neumann. Table 6.3 summaries the performance of this digital PLL-based TRNG.

Table 6.1 NIST SP800-22 statistical test results for the PLL-based RNG. (Post-processed by the classic von Neumann corrector)

Test	P-value		
	Data Set I	Data Set II	Data Set III
Frequency	0.483198	0.911958	0.901273
Block Frequency	0.538931	0.659396	0.456440
Cusum-Forward	0.438645	0.832364	0.296615
Cusum-Reverse	0.461142	0.918811	0.372568
Runs	0.015319	0.028246	0.021048
Longest Run	0.314287	0.323623	0.326702
Rank	0.098052	0.178995	0.398939
FFT	0.652276	0.614894	0.078635
Universal	0.080439	0.474608	0.783055
Approx. Entropy	0.131380	0.288881	0.858016
Serial1	0.451467	0.675619	0.022897
Serial2	0.374261	0.416942	0.051895
Lempel Ziv	1.000000	1.000000	1.000000
Linear Complexity	0.671359	0.699883	0.257862
Periodic Templates	0.982131	0.154475	0.878092
Aperiodic Templates	all passed	all passed	all passed
Random Excursions	all passed	all passed	all passed
Random Ex. Variant	all passed	all passed	all passed

Table 6.2 NIST SP800-22 statistical test results for the PLL-based RNG. (Post-processed by the modified von Neumann corrector)

Test	P-value		
	Data Set I	Data Set II	Data Set III
Frequency	0.403123	0.321518	0.346485
Block Frequency	0.151636	0.755148	0.055258
Cusum-Forward	0.323588	0.556027	0.677662
Cusum-Reverse	0.338798	0.243561	0.506432
Runs	0.17792	0.194128	0.903376
Longest Run	0.891002	0.945998	0.124246
Rank	0.239545	0.24964	0.141371
FFT	0.270026	0.6652	0.718271
Universal	0.685955	0.673109	0.841297
Approx. Entropy	0.399766	0.196523	0.981364
Serial1	0.866552	0.183901	0.619666
Serial2	0.793059	0.411532	0.319159
Lempel Ziv	1.000000	1.000000	1.000000
Linear Complexity	0.448407	0.187954	0.980834
Periodic Templates	0.362154	0.65189	0.299952
Aperiodic Templates	all passed	all passed	all passed
Random Excursions	all passed	all passed	all passed
Random Ex. Variant	all passed	all passed	all passed

Table 6.3 Performance summary for the PLL-based RNG.

Technology	1.5µm 2P2M CMOS
Supply Voltage	5V
Voltage Regulator Output	2.5V
Ring Oscillator Speed	30MHz
κ of Ring Oscillators	1.48E-07
System Loop Bandwidth	500kHz
PLL RMS Jitter	60ps
DAC LSB	40ps
RNG Output Data Rate	100kbps
Statistical Test Passed	NIST SP800-22 over 2Mbit long sequences
Power Consumption*	1.92mW
Die Area*	$1mm^2$

* Excluding digital on FPGA

Chapter 7: Design of DLL-based True Random Number Generator

7.1 Delay-locked Loops

Delay-locked loops (DLLs) have been widely used in applications such as frequency synthesizers [112], clock deskewing circuits [113], and memories [114]. Comparing to PLLs, DLLs have the advantage of unconditional stability and faster locking time [115].

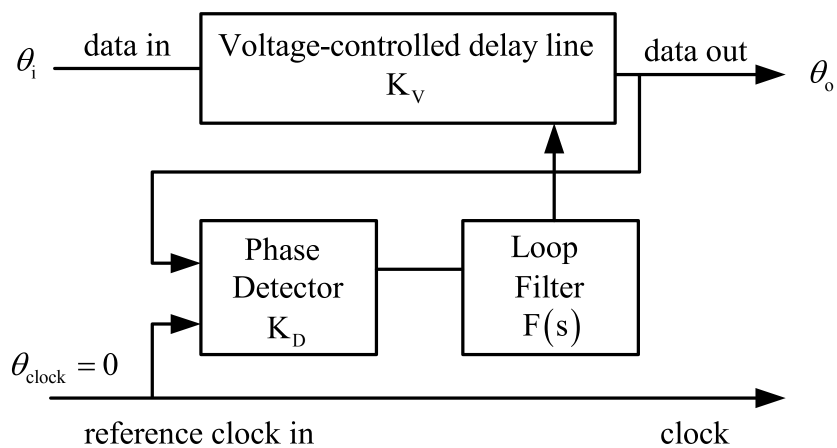


Figure 7.1 Basic block diagram of the DLL

Figure 7.1 shows the block diagram of a DLL consisting of a phase detector, a loop filter, and a voltage-controlled delay line (VCDL) [40]. The loop transfer function is [40]

$$\frac{\theta_o}{\theta_i} = \frac{1}{1 - K_D K_V F(s)} \tag{7.1}$$

where θ_i is the phase of the input data; θ_o is the phase of the output data; K_D is the phase-detector gain factor in the unit of [V/rad]; K_V is the delay line gain factor in the unit of [rad/V]; F(s) is the transfer function of the loop filter; ω_{clk} is the clock frequency.

The loop filter in the DLL usually consists of only a capacitor. The transfer function contains a single pole and the loop is a first-order feedback loop. Thus the loop is unconditional stable.

The DLL can not transfer the jitter to the clock and the jitter transfer function of the DLL is zero [116]

$$\frac{\theta_{clock}}{\theta_o} = 0 \tag{7.2}$$

So jitter filtering is independent of loop configurations such as the loop bandwidth. This allows the DLL to increase the loop bandwidth to reduce the acquisition time as long as the loop is stable.

7.2 System Architecture

The architecture of the proposed DLL-based RNG is illustrated in Figure 7.2. Two identical noisy voltage-controlled delay lines are designed with white noise dominated jitter. Delay line I is biased with fixed delay. The external clock passes both delay lines and the delay error is sampled by a low metastability D flip-flop, which also acts as a bang-bang phase detector. A loop filter converts the phase error to a control voltage and adjusts the delay of the delay line II so that the two delay lines are synchronized through the feedback. This system is still a nonlinear system.

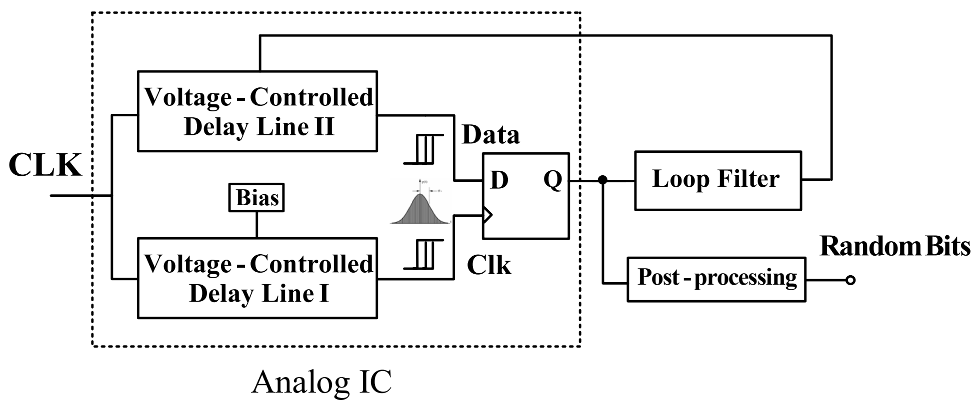


Figure 7.2 Architecture of the DLL-based RNG.

If the D flip-flop is replaced by a linear phase detector, this DLL system can be analyzed as a LTI system. Since the external clock serves as both the reference clock and the input data, as shown in Figure 7.3, the phase shift θ_{d2} by the delay line II is

$$\theta_{d2} = \left(\theta_{clock} + \theta_{d1} - \theta_o\right) K_D F\left(s\right) K_V \tag{7.3}$$

where θ_{clock} is the phase of the external clock, θ_{d1} is the phase shift by delay line I, θ_o is the output phase of this DLL.

The output phase θ_o of this DLL is

$$\theta_o = \theta_{clock} + \theta_{d2} \tag{7.4}$$

Substituting (7.4) into (7.3), the phase shift θ_{d2} by the delay line II is related to the phase shift θ_{d2} by the delay line I by

$$\frac{\theta_{d2}}{\theta_{d1}} = \frac{K_D K_V F(s)}{1 + K_D K_V F(s)} \tag{7.5}$$

If define the forward loop gain G(s) by

$$G(s) = K_D K_V F(s) \tag{7.6}$$

The loop transfer function is

$$H_s(s) = \frac{\theta_{d2}}{\theta_{d1}} = \frac{G(s)}{1 + G(s)} \tag{7.7}$$

Therefore, as long as the forward loop gain G(s) is much larger than 1, the loop will synchronize the phase shift by the two delay lines.

The noise transfer function $H_n(s)$ from θ_n to θ_o is

$$H_n(s) = \frac{\theta_0}{\theta_n} = \frac{1}{1 + G(s)} \tag{7.8}$$

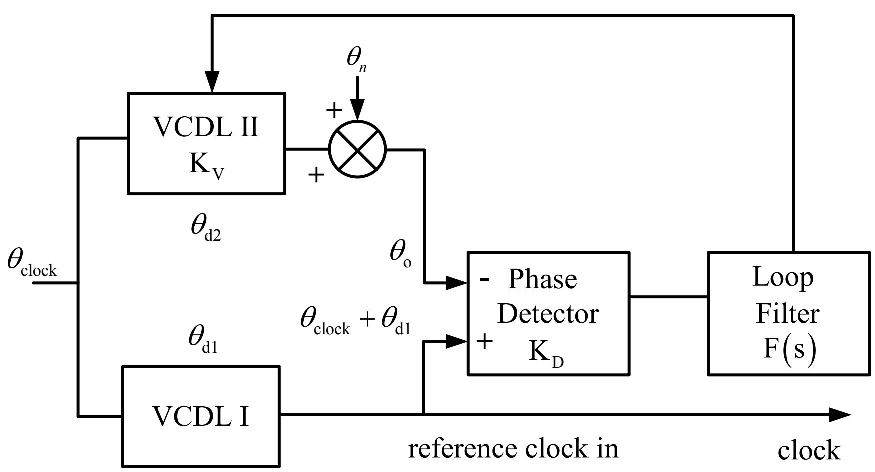


Figure 7.3 Block diagram of the DLL-based RNG.

The loop filter in this system is illustrated in Figure 7.4. The transfer function of this loop filter is

$$F(s)=\frac{1}{R_pC_p}\cdot\frac{s+\frac{1}{R_zC_z}}{s^2+\left(\frac{1}{R_zC_z}+\frac{1}{R_zC_p}+\frac{1}{R_pC_p}\right)s+\frac{1}{R_zC_zR_pC_p}} \tag{7.9}$$

The resistor R_p should be much larger than R_z to form a voltage divider, which attenuates the output of the phase detector to provide the instantaneous phase change to VCDL II. In this system, R_p is 10kΩ and R_z is 10Ω.

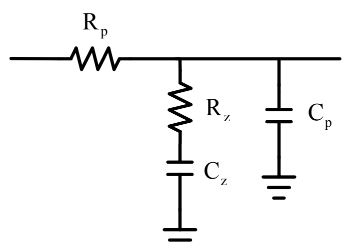


Figure 7.4 Loop Filter.

If the capacitor C_p is selected much larger than C_z, the poles of this loop filter are located at

$$p_{1,2}=-\frac{1}{2}\left(\frac{1}{R_zC_z}\right)\pm\frac{1}{2}\sqrt{\left(\frac{1}{R_zC_z}\right)^2-\frac{4}{R_zC_zR_pC_p}} \tag{7.10}$$

Since

$$\left(\frac{1}{R_zC_z}\right)^2\gg\frac{4}{R_zC_zR_pC_p} \tag{7.11}$$

(7.10) is approximated to

$$p_1\approx 0 \tag{7.12}$$

$$p_2 \approx -\frac{1}{R_z C_z} \tag{7.13}$$

Thus the zero of this loop filter will be cancelled by the pole p_2. And the loop transfer function is approximated to

$$F(s) \approx \frac{1}{R_p C_p s} \tag{7.14}$$

Substituting (7.14) into (7.7), the loop transfer function is

$$H_s(s) = \frac{f_L}{s + f_L} \tag{7.15}$$

where

$$f_L = \frac{K_D K_V}{R_p C_p} \tag{7.16}$$

The noise transfer function $H_n(s)$ is

$$H_n(s) = \frac{s}{s + f_L} \tag{7.17}$$

Transfer functions of (7.15) and (7.17) are same as those of first-order PLLs. Therefore the behavior of this DLL-based RNG is same as that of the PLL-based RNG. The output of the D flip-flop is in the format of return-to-V_{DD} data as discussed in Chapter 6. It is converted to non-return-to-V_{DD} data in the FPGA to correctly represent the phase error of the two delay lines.

Both delay lines in this DLL system are noisy and designed with white noise dominated jitter. From the analysis in Chapter 6, the added jitter to the external clock by two delay lines are independent Gaussian random variables with mean μ_i and standard deviation σ_i as

$$\mu_i = t_{di} \quad i = 1, 2 \tag{7.18}$$

$$\sigma_i = \kappa_i \sqrt{t_{di}} \quad i = 1, 2 \tag{7.19}$$

where κ_i is the white noise figure of merit of the delay line, t_{di} is the average time delay of the delay line which is related to the phase shift θ_{di} by

$$\theta_{di} = t_{di} \frac{2\pi}{T_{clock}} \tag{7.20}$$

Unlike the ring oscillators, the jitter at the output of the delay line is not fed back into the input. So the jitter process at the clock input of the D flip-flop is an uncorrelated Gaussian random process. The jitter process at the data input of the D flip-flop is a correlated Gaussian random process due to the feedback by the loop. The difference of these two Gaussian random processes is a zero-mean correlated Gaussian random process as analyzed in Chapter 6. Therefore a serial of correlated '1's and '0's with equal probability are generated by the D flip-flop.

7.3 Voltage-controlled Delay Line

The single-ended ring oscillators designed in Chapter 6 only achieved κ of 1.48E-7 with six extra 50kΩ resistors. In this design, differential delay stage is used in the VCDL to provide more jitter since the voltage swing is much smaller [10]. The cost is more power dissipation since the tail current of the differential stage always burns power while the current-starved inverter will not burn power when it is not switching.

The voltage-controlled delay line designed in this work is a simple differential pair with the symmetric load [117] as illustrated in Figure 7.5. The self-biased technique is not used since the effect of the tail current to the RNG was planned to be evaluated. The power supply V_{DD} is 3.3V to be compatible with the off-chip FPGA. A 0.3pF of capacitor is placed between V_{DD} and control node of the delay stage to bypass the deterministic noise from power supply and stabilize the voltage buffer.

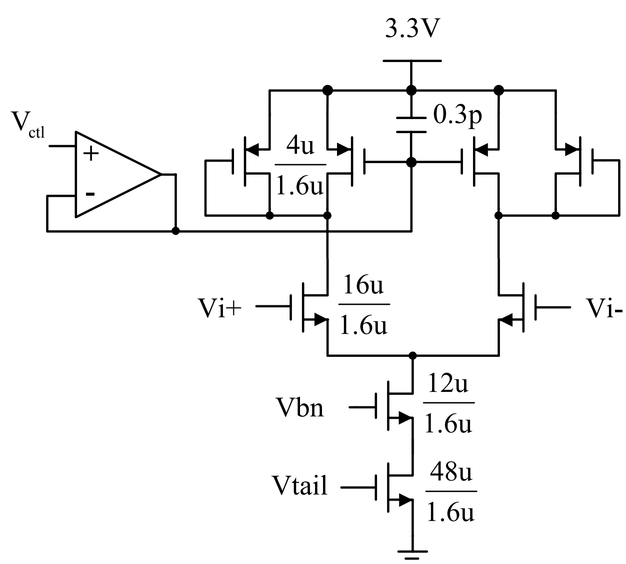


Figure 7.5 Voltage-controlled delay line.

To obtain high jitter, the power dissipation and the voltage swing should be minimized. The tail current is biased with 10μA and the power dissipation for each stage is 33μW. The output swing is designed to be around 600mV peak-to-peak. The output swing could be smaller to provide more jitter. However smaller output swing will consume more power in the following buffer to square it up for the D flip-flop.

From simulation, the κ for this differential stage achieved 2E-7 without using any extra resistor. The propagation delay of this delay stage is 5ns. To provide as much jitter as possible, the voltage-controlled delay line consists of 50 delay stages. The rms jitter of this delay line is

$$\sigma(\Delta T) = \kappa\sqrt{\Delta T} = (2E-7)\cdot\sqrt{50\cdot(5E-9)} = 100\text{ps} \tag{7.21}$$

And the total power dissipation for each voltage-controlled delay line is 1.65mW.

7.4 Improved Low Metastability D Flip-flop

The differential outputs of both delay lines are ‘squared up’ by current buffers to square waveforms with rise and fall times of 2ns. The D flip-flop designed in the PLL-based RNG is modified to take the differential clock. As illustrated in Figure 7.6, the pre-amplifier compares the difference of the clock and data directly and the voltage reference in the previous design is not needed any more. A dummy load is added at the data input of the D flip-flop so that both current buffers drive same amount of impedance.

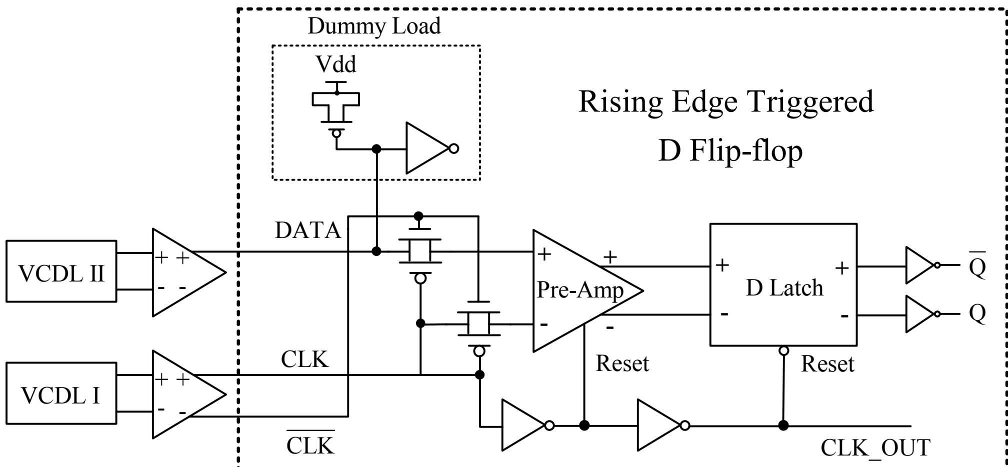


Figure 7.6 Improved low metastability D flip-flop.

This rising edge triggered D flip-flop is simulated by Cadence. Figure 7.7 shows the simulation results. During simulation, the edge of the input clock is ahead the edge of the input data by only 0.1ps. All input signals have rise and fall times of 2ns. The D flip-flop successfully resolves itself to logic ‘1’ before next cycle starts. With this performance, one D flip-flop is enough for this RNG. The outputs of the D latch are delayed by three stages of inverters so that the output data have enough setup time ahead of the output clock. A 5-stage push-pull buffer is implemented in this design to drive the outputs of this D flip-flop off the chip, which converts the data format back to return-to-V_{DD} data.

From simulation, the rms current drawn from the power supply for the pre-amplifier is 120μA; the rms supply current for the D latch is 50μA. Thus the total power dissipation for this D flip-flop is 560μW.

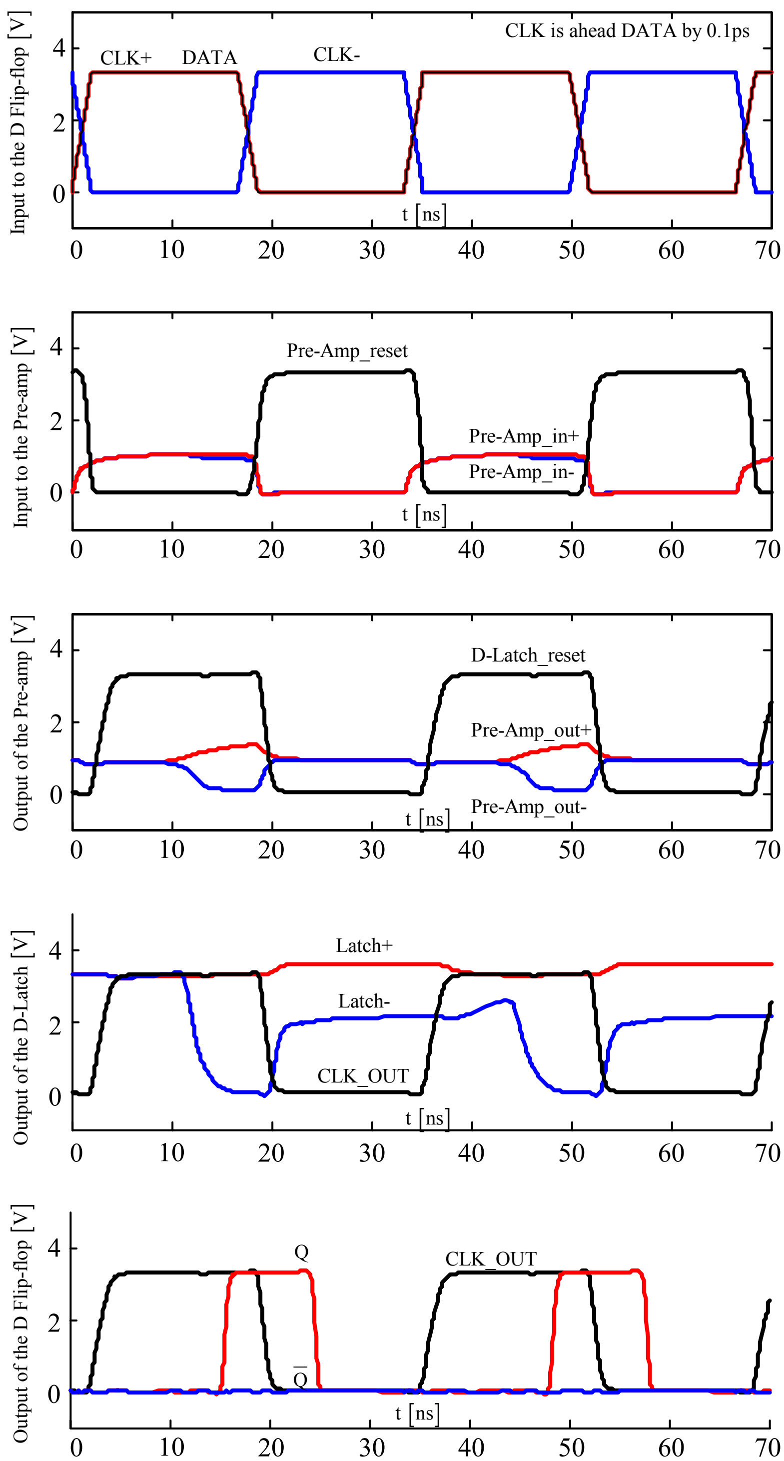


Figure 7.7 Simulation results for the improved D flip-flop.

7.5 Experimental Results

7.5.1 Test Chip Design

The customized part of this design including the voltage-controlled delay lines, the D flip-flop, and the voltage buffer are implemented in AMI 1.5µm 2-poly 2-metal CMOS process and consume an area of $2.4mm^2$. The chip micrograph is shown in Fig. 7.8.

The delay lines consume 3.3mW of power; the D flip-flop consumes 0.56mW of power; the buffers between the delay lines and the D flip-flop consume 1.6mW of power. Therefore, excluding the output buffers, the total power dissipation of this customized design is 5.46mW.

The loop filter is implemented on the PCB board. The digital post-processing circuits are implemented in an off-chip FPGA for design flexibility. Similar to the post-processing circuits designed for the PLL-based RNG, the raw output data pass a divider first before being fed into the classic von Neumann corrector or a modified von Neumann corrector to reduce the bias. Finally another frequency divider is used to further reduce the autocorrelation. The preset threshold in the modified Von Neumann corrector is still 3900.

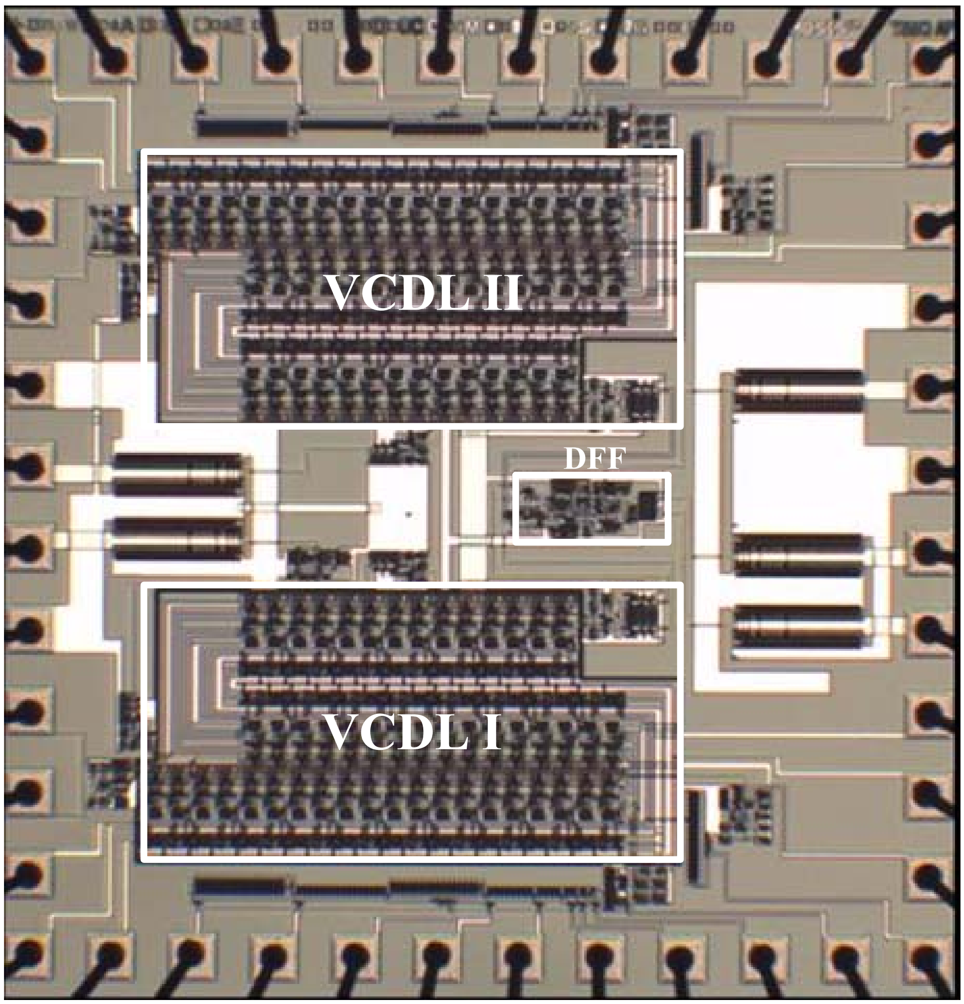


Figure 7.8 Chip micrograph for the DLL-based RNG.

7.5.2 Measurement Results

The jitter performance of the voltage-controlled delay lines was measured by Tektronix 11801C digital sampling oscilloscope [60] in time domain. A 2MHz 3.3V peak-to-peak square wave is used as both the clock input to the voltage-controlled delay lines and the trigger for the oscilloscope.

Figure 7.9 shows the measured tuning characteristic of the implemented VCDL. The measured relationship between the jitter of the VCDL and tuning is plotted in Figure 7.10. From (7.19), the white noise figure of merit κ is

$$\kappa = \frac{\sigma(\Delta t)}{\sqrt{\Delta t}} \tag{7.22}$$

By processing the data in Figure 7.9 and 7.10, the computed κ is plotted in Figure 7.11 versus tuning. The results in Figure 7.11 do not agree with the analysis in Section 5.6. The reason is that the tuning for the differential delay stage is realized by varying the load impedance rather than tail current, which in fact varies both the output swing and the RC constant at the output node.

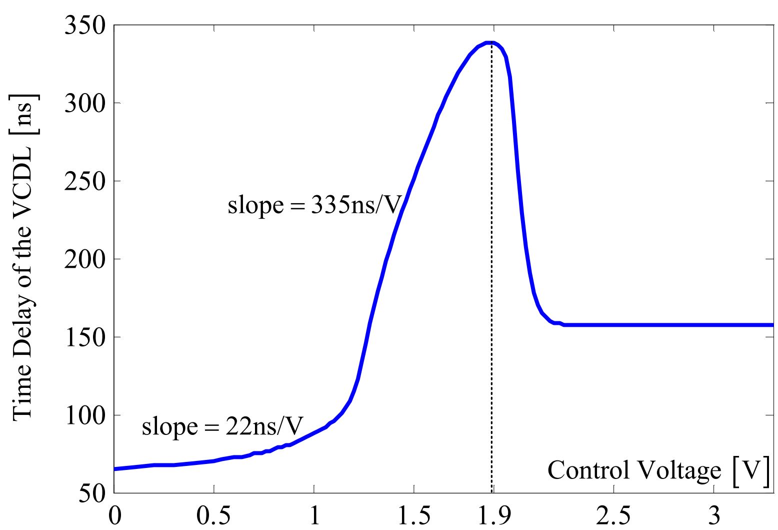


Figure 7.9 Measured tuning characteristic of the implemented VCDL.

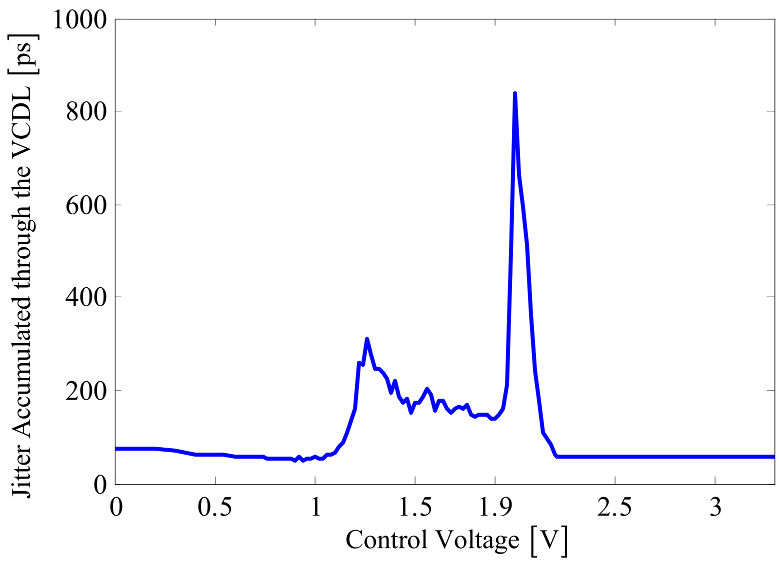


Figure 7.10 Measured jitter versus tuning for the implemented VCDL.

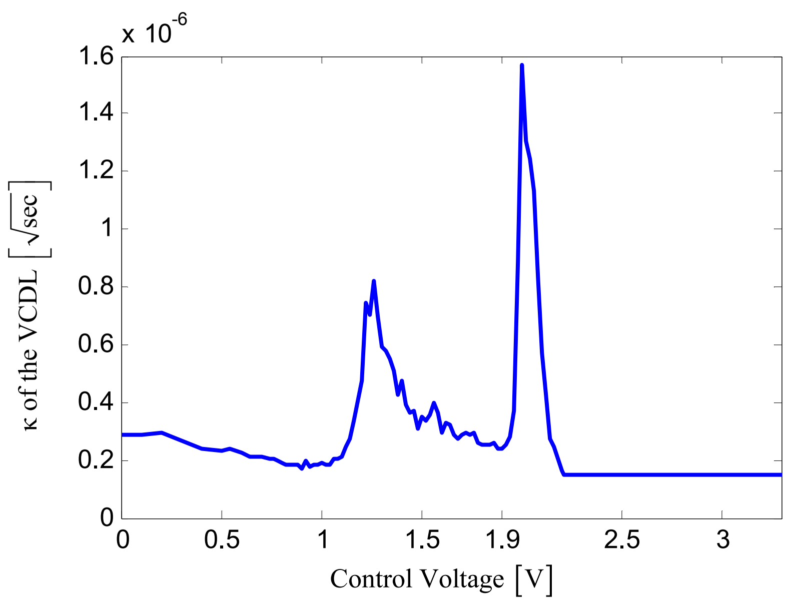


Figure 7.11 Measured κ versus tuning for the implemented VCDL.

When the control voltage is less than 1.9V, the tuning characteristic of VCDL in Figure 7.9 shows two different slopes. The delay line gain factor K_V is measured as 22ns/V and 335ns/V for these two regions. Since smaller K_V introduces less feedback when the loop is in lock, the VCDL II is managed to be locked in the 22ns/V region to reduce the autocorrelation. However, from Figure 7.10 the 22ns/V region can only provides rms jitter of less than 75ps. To overcome this problem, VCDL I is biased at 1.95V to provide rms jitter of 180ps at delay of 330ns. The complement of the RNG output is connected to the loop filter so that the rising edges of the VCDL I is locked to the falling edges of VCDL II, as shown in Figure 7.12. When the loop finishes the acquisition process, VCDL II is locked at delay of 80ns and provides jitter of 50ps.

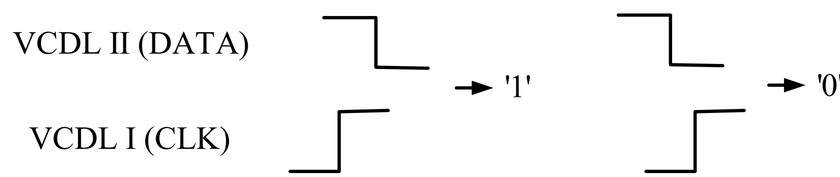


Figure 7.12 Rising edges of CLK locked to falling edges of DATA.

Figure 7.13 shows the spectrum and the autocorrelation of the raw data at the RNG output. Since for this design the RNG outputs data are in the return-to-V_{DD} format and the conversion to non-return-to-V_{DD} data is conducted in the FPGA, the signal integrity is not maintained well when the converted data are transferred back to the PCB board. This causes the duty cycle of the raw data is 43% for the ‘1’s, which results in the power at DC in the spectrum. Figure 7.14 shows the spectrum and the autocorrelation of the data post-processed by the FPGA with the classic von Neumann corrector. Similar plots are obtained for post-processing circuits with modified von Neumann corrector. During post-processing, the raw data are divided by two before fed into the classic or modified von Neumann corrector. The preset threshold in the modified von Neumann corrector is still 3900. Finally the data is divided by two again to further reduce the autocorrelation. The obtained average throughput of this DLL-based RNG is 100kbps with the classic von Neumann corrector, and 160kbps with the modified von Neumann corrector.

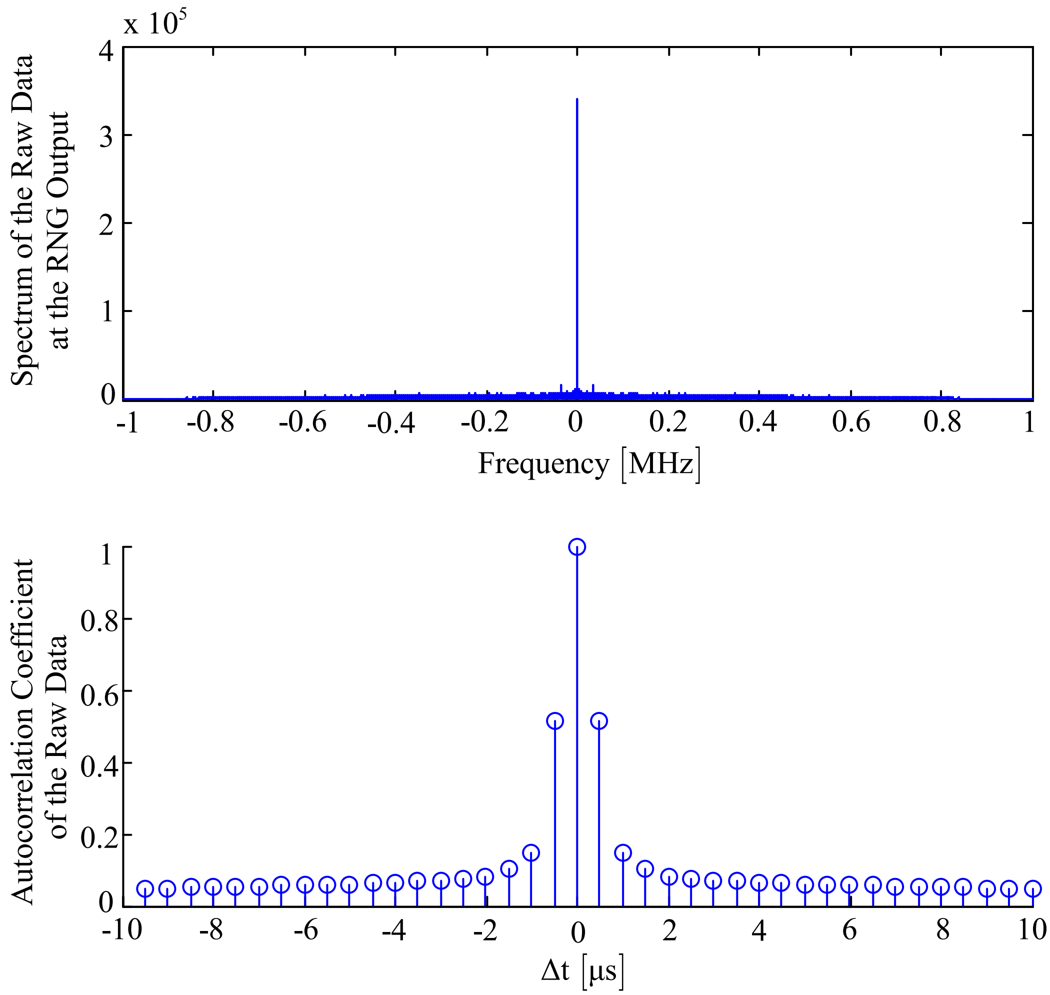


Figure 7.13 Spectrum and autocorrelation of the raw data at the RNG output.

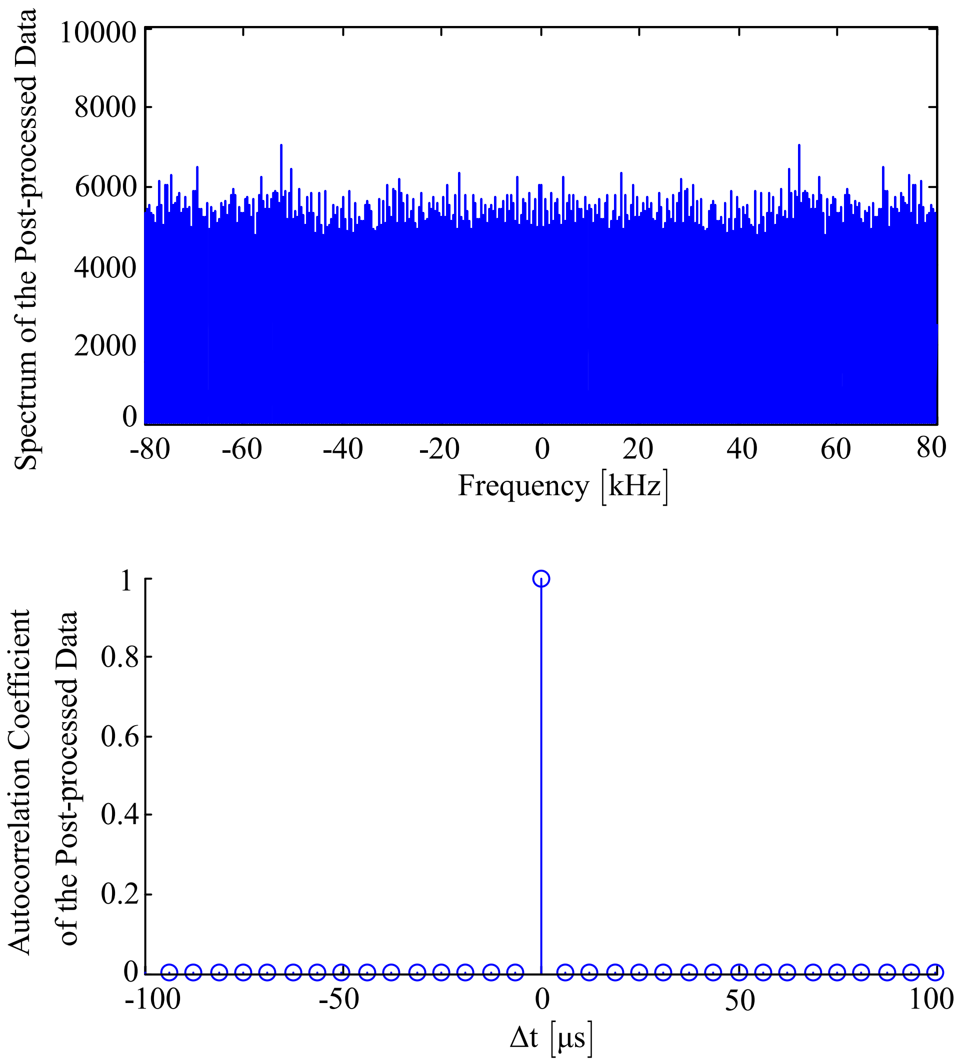


Figure 7.14 Spectrum and autocorrelation of the post-processed data.

In order to justify the noise sources which generate the random bits are Gaussian, the experiment illustrated in Figure 7.15 is performed for the delay lines under their locking conditions. In this experiment, the loop is broken and the control voltage of VCDL II is manually adjusted through a voltage attenuator so that the delay of the VCDL II changes from 79ns to 81ns. The raw data of the RNG output is collected by a data acquisition board and the duty cycle is computed. The computed duty cycle is a good estimate for the mean of the jitter difference at the specified time delay of VCDL II, and the CDF of this jitter sampling process is able to be constructed. The results are plotted in Figure 7.16. Each data point in Figure 7.16 is obtained from a data sequence of 50M bits.

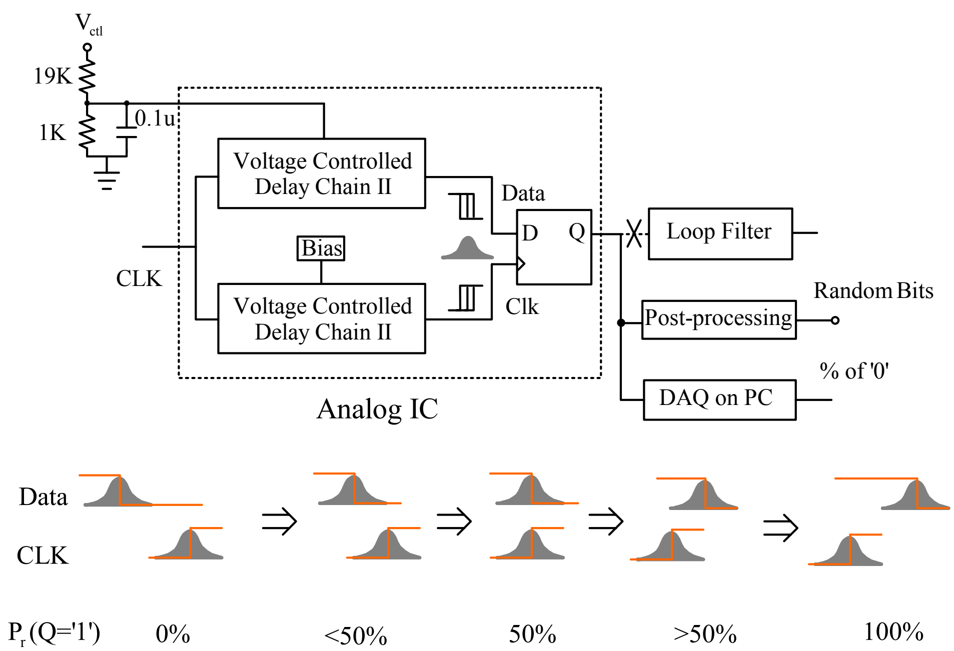


Figure 7.15 Experiment to justify the noise source.

The red curve in Figure 7.16 is the best-fit Gaussian CDF extracted by Matlab. The extracted standard deviation is 215.4ps. Since the loop is broken, the jitter processes at the inputs of the D flip-flop are independent. The standard deviation of the sampled jitter process is expected to be

$$\sigma_{diff} = \sqrt{\sigma_1^2 + \sigma_2^2} = \sqrt{(180ps)^2 + (50ps)^2} = 186.8ps \tag{7.23}$$

The measured rms jitter of 215.4ps shows good agreement to the predicted 186.8ps with an error of 15%. These results show that the D flip-flop implemented is good enough for this RNG system.

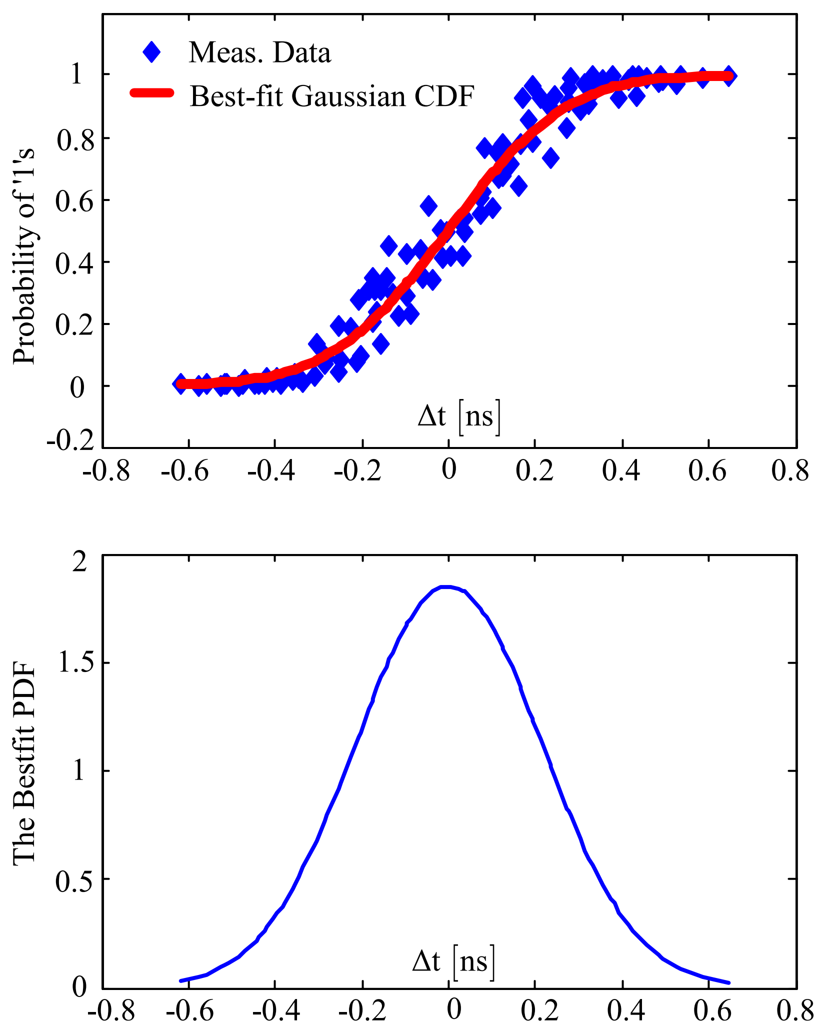


Figure 7.16 Measured CDF and PDF of the jitter sampling process from the experiment in Figure 7.15.

The quality of the randomness has been verified by the NIST SP800-22 test suite [106] over 2Mbit long sequences and the Diehard battery of tests [107] with data streams over 80M bits.

Table 7.1 shows the complete test results for three data sequences post-processed by the classic von Neumann. Table 7.2 shows the test results for data sequences post-processed by the modified von Neumann. The passing criteria for the NIST SP800-22 test suite is that each p-value is larger than 0.01 [106].

Although the Diehard test is one of the most comprehensive test suites, there are no well-defined pass criteria. Intel states in reference [118] that after considering of all 250 p-values, the interval of 0.0001 and 0.9999 for the p-value yields a 95% confidence. "Therefore, the RNG fails the Diehard tests if there is a p-value greater than or equal to 0.9999 or less than or equal to 0.0001." Appendix C and D contains sample Diehard test results performed on data streams with 120M bits post-processed by the classic and modified von Neumann corrector.

The performance of this DLL-based RNG and the comparison with the PLL-based RNG in Chapter 6 are summarized in Table 7.3.

Table 7.1 NIST SP800-22 statistical test results for the DLL-based RNG. (Post-processed by the classical von Neumann corrector)

Test	P-value		
	Data Set I	Data Set II	Data Set III
Frequency	0.906613	0.663750	0.924133
Block Frequency	0.563954	0.024046	0.462740
Cusum-Forward	0.397421	0.603638	0.617986
Cusum-Reverse	0.485023	0.286945	0.534325
Runs	0.992284	0.342987	0.316359
Longest Run	0.566233	0.846357	0.491218
Rank	0.923367	0.697882	0.214056
FFT	0.791636	0.274584	0.720179
Universal	0.827000	0.079229	0.640332
Approx. Entropy	0.873657	0.350029	0.490515
Serial1	0.320603	0.534539	0.495508
Serial2	0.831176	0.785853	0.435948
Lempel Ziv	1.000000	1.000000	1.000000
Linear Complexity	0.675385	0.333256	0.683934
Periodic Templates	0.323844	0.654655	0.254631
Aperiodic Templates	all passed	all passed	all passed
Random Excursions	all passed	all passed	all passed
Random Ex. Variant	all passed	all passed	all passed

Table 7.2 NIST SP800-22 statistical test results for the DLL-based RNG. (Post-processed by the modified von Neumann corrector)

Test	P-value		
	Data Set I	Data Set II	Data Set III
Frequency	0.774009	0.819892	0.956624
Block Frequency	0.056670	0.201523	0.111477
Cusum-Forward	0.640330	0.223384	0.826477
Cusum-Reverse	0.896763	0.345715	0.872350
Runs	0.848565	0.680978	0.074707
Longest Run	0.208117	0.848426	0.894307
Rank	0.373764	0.489182	0.374340
FFT	0.739574	0.066587	0.648725
Universal	0.703728	0.467987	0.423756
Approx. Entropy	0.894936	0.918398	0.896017
Serial1	0.778287	0.022665	0.489608
Serial2	0.791241	0.102496	0.385671
Lempel Ziv	1.000000	1.000000	1.000000
Linear Complexity	0.924310	0.364511	0.124683
Periodic Templates	0.736223	0.257206	0.054056
Aperiodic Templates	all passed	all passed	all passed
Random Excursions	all passed	all passed	all passed
Random Ex. Variant	all passed	all passed	all passed

Table 7.3 Summary and comparison of the PLL- and DLL-based RNG

	PLL-based RNG	DLL-based RNG
Technology	1.5μm 2P2M CMOS	1.5μm 2P2M CMOS
Supply Voltage	5V	3.3V
Voltage Regulator Output	2.5V	N/A
Clock Speed	30MHz	2MHz
System Loop Bandwidth	500kHz	200kHz
RMS Jitter	60ps	215ps
DAC LSB	40ps	N/A
RNG Output Data Rate	100kHz	160kHz
Statistical Test Passed	NIST SP800-22 over 2Mbits	NIST SP800-22 over 2Mbits
		Diehard test over 80Mbits
Power Consumption*	1.92mW	5.46mW
Die Area*	1mm^2	2.4mm^2

* Excluding digital on FPGA

Chapter 8: Conclusion

With the feature size of semiconductor processes scaling down aggressively for higher transistor density and faster speed, analog circuit design will face the challenge of increased fundamental noise, higher 1/f noise corner frequency, and lower power supply.

In Chapter 3, a simple practical model for jitter in the presence of 1/f noise has been developed. This model is consistent with measurements showing that accumulated jitter due to 1/f noise is proportional to the measurement time delay, and an analytical expression for the figure of merit ζ is also provided. The 1/f transition time is related to the $1/f^3$ phase noise corner by equation (3.48).

As feature sizes become smaller, wider PLL loop bandwidth is necessary to minimize the higher 1/f noise corner effect. The simple technique of (4.9) requires only κ and PLL loop bandwidth to predict jitter performance of a PLL [8]. It ignores 1/f noise because the 1/f noise corner is assumed to be inside the loop bandwidth frequency. It has been showed in this work that in a deep submicron process, the $1/f^3$ corner can move above the PLL loop bandwidth so that there will be more jitter than expected. A general technique is proposed in Chapter 4 for closed loop jitter prediction. If system specifications allow, the PLL loop bandwidth can be increased to a value above the $1/f^3$ noise corner, maintaining the applicability of the simple theory in [8]. If the PLL loop bandwidth is constrained to be below the $1/f^3$ noise corner, then the developed technique must be used to account for the 1/f noise contribution.

The most important contribution of this dissertation has been to develop a methodology to guide design of low jitter CMOS voltage-controlled ring oscillators in deep submicron processes. Thermal noise upconversion in CMOS ring oscillators is analyzed in time domain using a LTI model. The trade-off and relationship between jitter, speed, power dissipation and VCO geometry are evaluated for different applications. And the results indicate that jitter caused by white noise sources can be reduced by increasing the VCO's channel width and carefully choosing channel length and number of stages. This developed model is supported by the measured data from 24 ring oscillators with different geometry fabricated in TSMC 0.18μm process.

A new type of true RNG based on digital PLL has been proposed. The random bits are generated by the jitter sampling of two identical synchronized ring oscillators. Comparing to the traditional oscillator sampling approach, this method is able to achieve higher data rate when using same speed of clocks. This structure has been realized in a 1.5um process, and has successfully passed the NIST SP800-22 statistical test suite.

The VCO design methodology developed in this dissertation is applied to the design of PLL- and DLL- based true random number generators (TRNG) for application in the area of "smart cards". New architectures of dual-oscillator sampling and delay-line sampling are proposed for random number generation, which has the advantage of lower power dissipation and lower cost over traditional approaches. Both structures are implemented in test chips fabricated in AMI 1.5μm process. The PLL-based TRNG passed the NIST SP800-22 statistical test suite and the DLL-based TRNG passed both the NIST SP800-22 statistical test suite and the Diehard battery of tests.

8.1 Future Work

There are several possibilities for future work in this general area.

Only CMOS single-ended ring oscillators is discussed in detail in this thesis. Using the methodology in Chapter 5 and references [8] and [9], with the current and noise model in Chapter 2, the analytical expression of κ for differential ring oscillators with extremely short channels can be derived and analyzed. This work proposed the idea of the optimum channel length for κ minimization in VCO design but short of data to prove its validity. Another test chip is necessary to evaluate the differential ring oscillators and to prove the proposed optimum length.

The upconversion of 1/f noise can be analyzed following the same way in Chapter 5 if a good approximation for the standard deviation of 1/f noise in time domain, which is similar to the one for the thermal noise in (5.13) developed in [8], is available.

The best TRNG using delay line sampling can be implemented with a ‘smart’ background calibration circuit, which will stop the feedback to the delay lines when the loop acquisition finishes, monitor the delay drift in background, and automatically restart if necessary. As long as a digital controlled delay line is available, this background calibration circuit should be able to be implemented in a FPGA to test its performance.

Appendix A. Kappa Plot Extraction from TIE Data

This appendix shows the Matlab program to implement the algorithm in Section 3.2.3 which post-processes the TIE data measured by the digital oscilloscopes to extract the “kappa plot”.

```
% TIEstd.m
clear;
load tie.dat;
f=150E6;                          % The VCO frequency;
T=1/f; %period                    % The VCO period;
n=length(tie);                    % Length of the TIE data;

for i=1:1:n
   histogram=zeros(1,n-i);        % Reset the variable;
   for j=1:1:(n-i)
      histogram(j)=tie(j+i)-tie(j);    % Construct the histogram;
   end
   rmsjitter(i)=std(histogram);   % Compute the standard deviation;
end

t=T:T:n*T;
loglog(t,rmsjitter,'+-');
```

Appendix B. RNG Simulation by Matlab

This appendix includes the Matlab code to simulate the system behavior and loop acquisition for the PLL-based RNG.

```
% PLLRNG.m
clear;
t1(1)=0;               % Transition time sequence for VCO I
t2(1)=20e-9;           % VCO II is ahead of VCO I for 20ns at the beginning
f1=30E6;               % VCO I is free-running at 30MHz
T1=1/f1;               % Period of VCO I
f2(2)=29.9e6;          % The initial speed of Oscillator II is 29.9MHz
stop=10000;            % The clock cycle to stop simulation
tindex=0:T1:T1*(stop-1);
output_start=5500;     % The number of cycles needed for loop acquisition
nstd=60E-12;           % The oscillator cycle-to-cycle jitter is 60ps
ctrp(2)=0;             % Counter p
ctrbit=16;             % The bits in counter p
kz=1;                  % Gain of counter Z is 1
rb(2)=0;               % Output of RNG
LSB=20000;             % LSB of DAC is 20kHz
for n=2:stop
   % update transition times
   t1(n)=t1(n-1) + T1       + nstd*randn(1);
   t2(n)=t2(n-1) + 1/f2(n)  + nstd*randn(1);
   %      |        |          |
   %   previous  average     noise
   %   position  period
```

```
    %
    rb(n)=sign(t1(n)-t2(n));                    % Determine the random bit in [-1,1]
    ctrp(n+1)=ctrp(n)+rb(n);                   % Increment the up/down counter p

    % The 8 most significant bits are connected to VCO
    DAC(n)=fix(ctrp(n)/2^(ctrbit-8));

    % Adjusting VCO II running freq.
    f2(n+1)=f2(2) - DAC(n)*LSB - kz*rb(n)*LSB;
end

% Discard the bits generated during loop acquisition
rb_stable=rb(output_start:1:end);

% Divide the data rate down to lower autocorrelation
division=20;
rb_division=rb(output_start:division:end);

% autocorrelation coefficient of the RNG output
correff1=xcorr(rb_stable, 'coeff');
correff2=xcorr(rb_division, 'coeff');

% Plot results


subplot(8,1,1)
plot((t1-t2),'k')                                  % The difference of transition times

subplot(8,1,2)
```

```
plot (ctrp, 'k');                              % Plot the value of counter p

subplot(8,1,3)
plot(DAC, 'k')                                 % Plot the control at the DAC

subplot(8,1,4)
stem((rb_stable(1:1:100)+1)/2,'k');

subplot(8,1,5)
% Plot the autocorrelation coefficient for 30 adjacent bits
start1=(length(correff1)+1)/2-15;
stem(-15:1:15, correff1(start1:1:(start1+30)),'k');

subplot(8,1,6)
% Plot the FFT of the RNG output
fmax=1/2/T1;
fstep=2*fmax/(length(rb_stable)-1);
f=-fmax:fstep:fmax;
plot(f,abs(fftshift(fft(rb_stable))),'k')

subplot(8,1,7)
% Plot the autocorrelation coefficient for the divided data
start2=(length(correff2)+1)/2-15;
stem(-15:1:15, correff2(start2:1:(start2+30)),'k');

subplot(8,1,8)
% Plot the FFT of the divided data
fmax=1/2/T1/division;
```

```
fstep=2*fmax/(length(rb_division)-1);
f=-fmax:fstep:fmax;
plot(f,abs(fftshift(fft(rb_division))),'k')
```

Appendix C. Diehard Test Results for DLL-based RNG with the Classic Von Neumann Corrector

```
NOTE: Most of the tests in DIEHARD return a p-value, which
should be uniform on [0,1) if the input file contains truly
independent random bits.   Those p-values are obtained by
p=F(X), where F is the assumed distribution of the sample
random variable X---often normal. But that assumed F is just
an asymptotic approximation, for which the fit will be worst
in the tails. Thus you should not be surprised with
occasional p-values near 0 or 1, such as .0012 or .9983.
When a bit stream really FAILS BIG, you will get p's of 0 or
1 to six or more places.  By all means, do not, as a
Statistician might, think that a p < .025 or p> .975 means
that the RNG has "failed the test at the .05 level".  Such
p's happen among the hundreds that DIEHARD produces, even
with good RNG's.  So keep in mind that " p happens".
    ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
    ::            This is the BIRTHDAY SPACINGS TEST                 ::
    :: Choose m birthdays in a year of n days.  List the spacings    ::
    :: between the birthdays.  If j is the number of values that     ::
    :: occur more than once in that list, then j is asymptotically   ::
    :: Poisson distributed with mean m^3/(4n).  Experience shows n   ::
    :: must be quite large, say n>=2^18, for comparing the results   ::
    :: to the Poisson distribution with that mean.  This test uses   ::
    :: n=2^24 and m=2^9,  so that the underlying distribution for j  ::
    :: is taken to be Poisson with lambda=2^27/(2^26)=2.  A sample   ::
    :: of 500 j's is taken, and a chi-square goodness of fit test    ::
    :: provides a p value.  The first test uses bits 1-24 (counting  ::
    :: from the left) from integers in the specified file.           ::
    ::   Then the file is closed and reopened. Next, bits 2-25 are   ::
    :: used to provide birthdays, then 3-26 and so on to bits 9-32.  ::
    :: Each set of bits provides a p-value, and the nine p-values    ::
    :: provide a sample for a KSTEST.                                ::
    ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::

BIRTHDAY SPACINGS TEST, M= 512 N=2**24 LAMBDA=  2.0000
          Results for d1bin
                  For a sample of size 500:     mean
          d1bin           using bits  1 to 24   1.890
 duplicate       number       number
 spacings       observed     expected
       0          76.       67.668
       1         145.      135.335
       2         135.      135.335
       3          80.       90.224
       4          43.       45.112
       5          10.       18.045
 6 to INF         11.        8.282
Chisquare with  6 d.o.f. =     7.45 p-value=  .719024
 :::::::::::::::::::::::::::::::::::::::::
```

```
                For a sample of size 500:     mean
         d1bin           using bits  2 to 25   1.994
 duplicate       number       number
 spacings       observed     expected
      0          73.       67.668
      1         122.      135.335
      2         140.      135.335
      3          99.       90.224
      4          38.       45.112
      5          23.       18.045
 6 to INF         5.        8.282
Chisquare with  6 d.o.f. =     6.53 p-value=  .633610
 :::::::::::::::::::::::::::::::::::::::::
                For a sample of size 500:     mean
         d1bin           using bits  3 to 26   2.026
 duplicate       number       number
 spacings       observed     expected
      0          70.       67.668
      1         116.      135.335
      2         158.      135.335
      3          85.       90.224
      4          40.       45.112
      5          23.       18.045
 6 to INF         8.        8.282
Chisquare with  6 d.o.f. =     8.89 p-value=  .820170
 :::::::::::::::::::::::::::::::::::::::::
                For a sample of size 500:     mean
         d1bin           using bits  4 to 27   1.892
 duplicate       number       number
 spacings       observed     expected
      0          71.       67.668
      1         145.      135.335
      2         142.      135.335
      3          77.       90.224
      4          46.       45.112
      5          15.       18.045
 6 to INF         4.        8.282
Chisquare with  6 d.o.f. =     5.87 p-value=  .561584
 :::::::::::::::::::::::::::::::::::::::::
                For a sample of size 500:     mean
         d1bin           using bits  5 to 28   2.046
 duplicate       number       number
 spacings       observed     expected
      0          64.       67.668
      1         122.      135.335
      2         144.      135.335
      3         101.       90.224
      4          44.       45.112
      5          18.       18.045
 6 to INF         7.        8.282
Chisquare with  6 d.o.f. =     3.58 p-value=  .266783
```

```
 ::::::::::::::::::::::::::::::::::::::::
                 For a sample of size 500:     mean
         d1bin           using bits  6 to 29   1.914
 duplicate       number       number
 spacings       observed     expected
      0          65.       67.668
      1         150.      135.335
      2         138.      135.335
      3          81.       90.224
      4          44.       45.112
      5          20.       18.045
 6 to INF         2.        8.282
Chisquare with  6 d.o.f. =     7.69 p-value=  .738584
 ::::::::::::::::::::::::::::::::::::::::
                 For a sample of size 500:     mean
         d1bin           using bits  7 to 30   1.928
 duplicate       number       number
 spacings       observed     expected
      0          74.       67.668
      1         137.      135.335
      2         142.      135.335
      3          80.       90.224
      4          39.       45.112
      5          23.       18.045
 6 to INF         5.        8.282
Chisquare with  6 d.o.f. =     5.59 p-value=  .529236
 ::::::::::::::::::::::::::::::::::::::::
                 For a sample of size 500:     mean
         d1bin           using bits  8 to 31   2.038
 duplicate       number       number
 spacings       observed     expected
      0          68.       67.668
      1         128.      135.335
      2         145.      135.335
      3          79.       90.224
      4          49.       45.112
      5          23.       18.045
 6 to INF         8.        8.282
Chisquare with  6 d.o.f. =     4.19 p-value=  .349164
 ::::::::::::::::::::::::::::::::::::::::
                 For a sample of size 500:     mean
         d1bin           using bits  9 to 32   2.038
 duplicate       number       number
 spacings       observed     expected
      0          78.       67.668
      1         123.      135.335
      2         119.      135.335
      3         103.       90.224
      4          48.       45.112
      5          19.       18.045
 6 to INF        10.        8.282
Chisquare with  6 d.o.f. =     7.07 p-value=  .686026
```

```
  :::::::::::::::::::::::::::::::::::::::
   The 9 p-values were
        .719024   .633610   .820170   .561584   .266783
        .738584   .529236   .349164   .686026
  A KSTEST for the 9 p-values yields  .739645

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::           THE OVERLAPPING 5-PERMUTATION TEST                ::
     :: This is the OPERM5 test.  It looks at a sequence of one mill- ::
     :: ion 32-bit random integers.  Each set of five consecutive     ::
     :: integers can be in one of 120 states, for the 5! possible or- ::
     :: derings of five numbers.  Thus the 5th, 6th, 7th,...numbers   ::
     :: each provide a state. As many thousands of state transitions  ::
     :: are observed,  cumulative counts are made of the number of    ::
     :: occurences of each state.  Then the quadratic form in the     ::
     :: weak inverse of the 120x120 covariance matrix yields a test   ::
     :: equivalent to the likelihood ratio test that the 120 cell     ::
     :: counts came from the specified (asymptotically) normal dis-   ::
     :: tribution with the specified 120x120 covariance matrix (with  ::
     :: rank 99).  This version uses 1,000,000 integers, twice.       ::
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
           OPERM5 test for file d1bin
     For a sample of 1,000,000 consecutive 5-tuples,
 chisquare for 99 degrees of freedom= 85.968; p-value= .178095
           OPERM5 test for file d1bin
     For a sample of 1,000,000 consecutive 5-tuples,
 chisquare for 99 degrees of freedom= 97.365; p-value= .472313
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     :: This is the BINARY RANK TEST for 31x31 matrices. The leftmost ::
     :: 31 bits of 31 random integers from the test sequence are used ::
     :: to form a 31x31 binary matrix over the field {0,1}. The rank  ::
     :: is determined. That rank can be from 0 to 31, but ranks< 28   ::
     :: are rare, and their counts are pooled with those for rank 28. ::
     :: Ranks are found for 40,000 such random matrices and a chisqua-::
     :: re test is performed on counts for ranks 31,30,29 and <=28.   ::
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
    Binary rank test for d1bin
         Rank test for 31x31 binary matrices:
        rows from leftmost 31 bits of each 32-bit integer
      rank   observed  expected (o-e)^2/e  sum
        28       200     211.4   .616651     .617
        29      5193    5134.0   .677792    1.294
        30     23101   23103.0   .000181    1.295
        31     11506   11551.5   .179411    1.474
  chisquare= 1.474 for 3 d. of f.; p-value= .424080
--------------------------------------------------------------
```

```
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     :: This is the BINARY RANK TEST for 32x32 matrices. A random 32x ::
     :: 32 binary matrix is formed, each row a 32-bit random integer. ::
     :: The rank is determined. That rank can be from 0 to 32, ranks  ::
     :: less than 29 are rare, and their counts are pooled with those ::
     :: for rank 29.  Ranks are found for 40,000 such random matrices ::
     :: and a chisquare test is performed on counts for ranks  32,31, ::
     :: 30 and <=29.                                                  ::
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
    Binary rank test for d1bin
         Rank test for 32x32 binary matrices:
        rows from leftmost 32 bits of each 32-bit integer
      rank   observed  expected (o-e)^2/e  sum
        29       214     211.4   .031533     .032
        30      5163    5134.0   .163694     .195
        31     23089   23103.0   .008541     .204
        32     11534   11551.5   .026586     .230
  chisquare=  .230 for 3 d. of f.; p-value= .343085
--------------------------------------------------------------

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     :: This is the BINARY RANK TEST for 6x8 matrices.  From each of  ::
     :: six random 32-bit integers from the generator under test, a   ::
     :: specified byte is chosen, and the resulting six bytes form a  ::
     :: 6x8 binary matrix whose rank is determined.  That rank can be ::
     :: from 0 to 6, but ranks 0,1,2,3 are rare; their counts are     ::
     :: pooled with those for rank 4. Ranks are found for 100,000     ::
     :: random matrices, and a chi-square test is performed on       ::
     :: counts for ranks 6,5 and <=4.                                 ::
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
         Binary Rank Test for d1bin
        Rank of a 6x8 binary matrix,
     rows formed from eight bits of the RNG d1bin
     b-rank test for bits  1 to  8
                     OBSERVED   EXPECTED     (O-E)^2/E      SUM
          r<=4          977       944.3       1.132       1.132
          r =5        21731     21743.9        .008       1.140
          r =6        77292     77311.8        .005       1.145
                        p=1-exp(-SUM/2)= .43589
        Rank of a 6x8 binary matrix,
     rows formed from eight bits of the RNG d1bin
     b-rank test for bits  2 to  9
                     OBSERVED   EXPECTED     (O-E)^2/E      SUM
          r<=4          938       944.3        .042        .042
          r =5        21861     21743.9        .631        .673
          r =6        77201     77311.8        .159        .831
                        p=1-exp(-SUM/2)= .34015
        Rank of a 6x8 binary matrix,
     rows formed from eight bits of the RNG d1bin
     b-rank test for bits  3 to 10
                     OBSERVED   EXPECTED     (O-E)^2/E      SUM
          r<=4          968       944.3        .595        .595
          r =5        21743     21743.9        .000        .595
          r =6        77289     77311.8        .007        .602
```

```
                 p=1-exp(-SUM/2)= .25974
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG d1bin
b-rank test for bits  4 to 11
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          916       944.3        .848        .848
     r =5        21949     21743.9       1.935       2.783
     r =6        77135     77311.8        .404       3.187
                 p=1-exp(-SUM/2)= .79680
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG d1bin
b-rank test for bits  5 to 12
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          928       944.3        .281        .281
     r =5        21793     21743.9        .111        .392
     r =6        77279     77311.8        .014        .406
                 p=1-exp(-SUM/2)= .18380
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG d1bin
b-rank test for bits  6 to 13
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          931       944.3        .187        .187
     r =5        21664     21743.9        .294        .481
     r =6        77405     77311.8        .112        .593
                 p=1-exp(-SUM/2)= .25670
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG d1bin
b-rank test for bits  7 to 14
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          965       944.3        .454        .454
     r =5        21665     21743.9        .286        .740
     r =6        77370     77311.8        .044        .784
                 p=1-exp(-SUM/2)= .32423
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG d1bin
b-rank test for bits  8 to 15
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          973       944.3        .872        .872
     r =5        21516     21743.9       2.389       3.261
     r =6        77511     77311.8        .513       3.774
                 p=1-exp(-SUM/2)= .84848
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG d1bin
b-rank test for bits  9 to 16
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          955       944.3        .121        .121
     r =5        21700     21743.9        .089        .210
     r =6        77345     77311.8        .014        .224
                 p=1-exp(-SUM/2)= .10600
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG d1bin
b-rank test for bits 10 to 17
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          860       944.3       7.526       7.526
     r =5        21606     21743.9        .875       8.400
     r =6        77534     77311.8        .639       9.039
```

```
                p=1-exp(-SUM/2)= .98911
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG d1bin
b-rank test for bits 11 to 18
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          922       944.3        .527        .527
     r =5        21504     21743.9       2.647       3.174
     r =6        77574     77311.8        .889       4.063
                p=1-exp(-SUM/2)= .86884
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG d1bin
b-rank test for bits 12 to 19
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          916       944.3        .848        .848
     r =5        21767     21743.9        .025        .873
     r =6        77317     77311.8        .000        .873
                p=1-exp(-SUM/2)= .35374
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG d1bin
b-rank test for bits 13 to 20
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          917       944.3        .789        .789
     r =5        21783     21743.9        .070        .860
     r =6        77300     77311.8        .002        .861
                p=1-exp(-SUM/2)= .34996
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG d1bin
b-rank test for bits 14 to 21
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          952       944.3        .063        .063
     r =5        21751     21743.9        .002        .065
     r =6        77297     77311.8        .003        .068
                p=1-exp(-SUM/2)= .03339
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG d1bin
b-rank test for bits 15 to 22
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          948       944.3        .014        .014
     r =5        21832     21743.9        .357        .371
     r =6        77220     77311.8        .109        .480
                p=1-exp(-SUM/2)= .21355
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG d1bin
b-rank test for bits 16 to 23
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          991       944.3       2.309       2.309
     r =5        21691     21743.9        .129       2.438
     r =6        77318     77311.8        .000       2.439
                p=1-exp(-SUM/2)= .70456
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG d1bin
b-rank test for bits 17 to 24
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          943       944.3        .002        .002
     r =5        21664     21743.9        .294        .295
     r =6        77393     77311.8        .085        .381
```

```
                 p=1-exp(-SUM/2)= .17332
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG d1bin
b-rank test for bits 18 to 25
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          931       944.3        .187        .187
     r =5        21735     21743.9        .004        .191
     r =6        77334     77311.8        .006        .197
                 p=1-exp(-SUM/2)= .09397
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG d1bin
b-rank test for bits 19 to 26
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          928       944.3        .281        .281
     r =5        21821     21743.9        .273        .555
     r =6        77251     77311.8        .048        .603
                 p=1-exp(-SUM/2)= .26015
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG d1bin
b-rank test for bits 20 to 27
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          924       944.3        .436        .436
     r =5        21812     21743.9        .213        .650
     r =6        77264     77311.8        .030        .679
                 p=1-exp(-SUM/2)= .28798
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG d1bin
b-rank test for bits 21 to 28
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          934       944.3        .112        .112
     r =5        21595     21743.9       1.020       1.132
     r =6        77471     77311.8        .328       1.460
                 p=1-exp(-SUM/2)= .51805
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG d1bin
b-rank test for bits 22 to 29
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          986       944.3       1.841       1.841
     r =5        21539     21743.9       1.931       3.772
     r =6        77475     77311.8        .344       4.117
                 p=1-exp(-SUM/2)= .87233
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG d1bin
b-rank test for bits 23 to 30
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          954       944.3        .100        .100
     r =5        21648     21743.9        .423        .523
     r =6        77398     77311.8        .096        .619
                 p=1-exp(-SUM/2)= .26607
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG d1bin
b-rank test for bits 24 to 31
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          936       944.3        .073        .073
     r =5        21629     21743.9        .607        .680
     r =6        77435     77311.8        .196        .876
```

```
                    p=1-exp(-SUM/2)= .35482
        Rank of a 6x8 binary matrix,
     rows formed from eight bits of the RNG d1bin
     b-rank test for bits 25 to 32
                    OBSERVED   EXPECTED     (O-E)^2/E      SUM
          r<=4         997       944.3       2.941       2.941
          r =5       21644     21743.9        .459       3.400
          r =6       77359     77311.8        .029       3.429
                    p=1-exp(-SUM/2)= .81992
   TEST SUMMARY, 25 tests on 100,000 random 6x8 matrices
 These should be 25 uniform [0,1] random variables:
     .435887     .340149     .259745     .796802     .183802
     .256698     .324233     .848480     .106001     .989106
     .868842     .353736     .349957     .033390     .213551
     .704563     .173318     .093974     .260147     .287977
     .518052     .872334     .266068     .354819     .819924
   brank test summary for d1bin
       The KS test for those 25 supposed UNI's yields
                    KS p-value= .823936

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::                   THE BITSTREAM TEST                          ::
     :: The file under test is viewed as a stream of bits. Call them  ::
     :: b1,b2,... .  Consider an alphabet with two "letters", 0 and 1 ::
     :: and think of the stream of bits as a succession of 20-letter  ::
     :: "words", overlapping.  Thus the first word is b1b2...b20, the ::
     :: second is b2b3...b21, and so on.  The bitstream test counts   ::
     :: the number of missing 20-letter (20-bit) words in a string of ::
     :: 2^21 overlapping 20-letter words.  There are 2^20 possible 20 ::
     :: letter words.  For a truly random string of 2^21+19 bits, the ::
     :: number of missing words j should be (very close to) normally  ::
     :: distributed with mean 141,909 and sigma 428.  Thus            ::
     ::  (j-141909)/428 should be a standard normal variate (z score) ::
     :: that leads to a uniform [0,1) p value.  The test is repeated  ::
     :: twenty times.                                                 ::
     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
 THE OVERLAPPING 20-tuples BITSTREAM  TEST, 20 BITS PER WORD, N words
   This test uses N=2^21 and samples the bitstream 20 times.
  No. missing words should average  141909. with sigma=428.
---------------------------------------------------------
tst no  1:  141610 missing words,  -.70 sigmas from mean, p-value= .24216
tst no  2:  142217 missing words,   .72 sigmas from mean, p-value= .76389
tst no  3:  141713 missing words,  -.46 sigmas from mean, p-value= .32322
tst no  4:  142240 missing words,   .77 sigmas from mean, p-value= .78012
tst no  5:  141431 missing words, -1.12 sigmas from mean, p-value= .13187
tst no  6:  142101 missing words,   .45 sigmas from mean, p-value= .67286
tst no  7:  142237 missing words,   .77 sigmas from mean, p-value= .77804
tst no  8:  141510 missing words,  -.93 sigmas from mean, p-value= .17541
tst no  9:  142200 missing words,   .68 sigmas from mean, p-value= .75148
tst no 10:  141252 missing words, -1.54 sigmas from mean, p-value= .06229
tst no 11:  142402 missing words,  1.15 sigmas from mean, p-value= .87515
tst no 12:  142434 missing words,  1.23 sigmas from mean, p-value= .88988
tst no 13:  142147 missing words,   .56 sigmas from mean, p-value= .71066
tst no 14:  141600 missing words,  -.72 sigmas from mean, p-value= .23492
```

```
tst no 15:  141179 missing words, -1.71 sigmas from mean, p-value= .04397
tst no 16:  141032 missing words, -2.05 sigmas from mean, p-value= .02019
tst no 17:  141392 missing words, -1.21 sigmas from mean, p-value= .11339
tst no 18:  141728 missing words,  -.42 sigmas from mean, p-value= .33591
tst no 19:  141704 missing words,  -.48 sigmas from mean, p-value= .31571
tst no 20:  141398 missing words, -1.19 sigmas from mean, p-value= .11610

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::             The tests OPSO, OQSO and DNA                      ::
     ::         OPSO means Overlapping-Pairs-Sparse-Occupancy         ::
     :: The OPSO test considers 2-letter words from an alphabet of    ::
     :: 1024 letters.  Each letter is determined by a specified ten   ::
     :: bits from a 32-bit integer in the sequence to be tested. OPSO ::
     :: generates  2^21 (overlapping) 2-letter words  (from 2^21+1    ::
     :: "keystrokes")  and counts the number of missing words---that  ::
     :: is 2-letter words which do not appear in the entire sequence. ::
     :: That count should be very close to normally distributed with  ::
     :: mean 141,909, sigma 290. Thus (missingwrds-141909)/290 should ::
     :: be a standard normal variable. The OPSO test takes 32 bits at ::
     :: a time from the test file and uses a designated set of ten    ::
     :: consecutive bits. It then restarts the file for the next de-  ::
     :: signated 10 bits, and so on.                                  ::
     ::                                                               ::
     ::     OQSO means Overlapping-Quadruples-Sparse-Occupancy        ::
     ::   The test OQSO is similar, except that it considers 4-letter ::
     :: words from an alphabet of 32 letters, each letter determined  ::
     :: by a designated string of 5 consecutive bits from the test    ::
     :: file, elements of which are assumed 32-bit random integers.   ::
     :: The mean number of missing words in a sequence of 2^21 four-  ::
     :: letter words,  (2^21+3 "keystrokes"), is again 141909, with   ::
     :: sigma = 295.  The mean is based on theory; sigma comes from   ::
     :: extensive simulation.                                         ::
     ::                                                               ::
     ::    The DNA test considers an alphabet of 4 letters::  C,G,A,T,::
     :: determined by two designated bits in the sequence of random   ::
     :: integers being tested.  It considers 10-letter words, so that ::
     :: as in OPSO and OQSO, there are 2^20 possible words, and the   ::
     :: mean number of missing words from a string of 2^21  (over-    ::
     :: lapping)  10-letter  words (2^21+9 "keystrokes") is 141909.   ::
     :: The standard deviation sigma=339 was determined as for OQSO   ::
     :: by simulation.  (Sigma for OPSO, 290, is the true value (to   ::
     :: three places), not determined by simulation.                  ::
     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::

 OPSO test for generator d1bin
  Output: No. missing words (mw), equiv normal variate (z), p-value (p)
                                                           mw     z     p
OPSO for d1bin           using bits 23 to 32        141332 -1.991  .0233
OPSO for d1bin           using bits 22 to 31        141739  -.587  .2785
OPSO for d1bin           using bits 21 to 30        142266  1.230  .8906
OPSO for d1bin           using bits 20 to 29        141957   .164  .5653
OPSO for d1bin           using bits 19 to 28        141898  -.039  .4844
OPSO for d1bin           using bits 18 to 27        141800  -.377  .3531
OPSO for d1bin           using bits 17 to 26        142208  1.030  .8485
```

```
OPSO for d1bin           using bits 16 to 25        142011   .351  .6371
OPSO for d1bin           using bits 15 to 24        142142   .802  .7888
OPSO for d1bin           using bits 14 to 23        141708  -.694  .2438
OPSO for d1bin           using bits 13 to 22        142335  1.468  .9289
OPSO for d1bin           using bits 12 to 21        142255  1.192  .8834
OPSO for d1bin           using bits 11 to 20        142145   .813  .7918
OPSO for d1bin           using bits 10 to 19        142425  1.778  .9623
OPSO for d1bin           using bits  9 to 18        141896  -.046  .4817
OPSO for d1bin           using bits  8 to 17        142046   .471  .6813
OPSO for d1bin           using bits  7 to 16        142056   .506  .6935
OPSO for d1bin           using bits  6 to 15        141474 -1.501  .0667
OPSO for d1bin           using bits  5 to 14        142150   .830  .7967
OPSO for d1bin           using bits  4 to 13        142330  1.451  .9266
OPSO for d1bin           using bits  3 to 12        141993   .289  .6135
OPSO for d1bin           using bits  2 to 11        141729  -.622  .2670
OPSO for d1bin           using bits  1 to 10        141871  -.132  .4474

OQSO test for generator d1bin
  Output: No. missing words (mw), equiv normal variate (z), p-value (p)
                                                      mw       z      p
OQSO for d1bin           using bits 28 to 32        142137   .772  .7799
OQSO for d1bin           using bits 27 to 31        141784  -.425  .3355
OQSO for d1bin           using bits 26 to 30        141625  -.964  .1676
OQSO for d1bin           using bits 25 to 29        142517  2.060  .9803
OQSO for d1bin           using bits 24 to 28        141966   .192  .5762
OQSO for d1bin           using bits 23 to 27        142120   .714  .7624
OQSO for d1bin           using bits 22 to 26        142472  1.907  .9718
OQSO for d1bin           using bits 21 to 25        141275 -2.150  .0158
OQSO for d1bin           using bits 20 to 24        141755  -.523  .3004
OQSO for d1bin           using bits 19 to 23        142049   .473  .6821
OQSO for d1bin           using bits 18 to 22        141650  -.879  .1897
OQSO for d1bin           using bits 17 to 21        141527 -1.296  .0975
OQSO for d1bin           using bits 16 to 20        141822  -.296  .3836
OQSO for d1bin           using bits 15 to 19        141998   .301  .6181
OQSO for d1bin           using bits 14 to 18        142140   .782  .7829
OQSO for d1bin           using bits 13 to 17        141754  -.527  .2993
OQSO for d1bin           using bits 12 to 16        141964   .185  .5735
OQSO for d1bin           using bits 11 to 15        142719  2.745  .9970
OQSO for d1bin           using bits 10 to 14        141578 -1.123  .1307
OQSO for d1bin           using bits  9 to 13        141817  -.313  .3771
OQSO for d1bin           using bits  8 to 12        141872  -.127  .4497
OQSO for d1bin           using bits  7 to 11        141791  -.401  .3442
OQSO for d1bin           using bits  6 to 10        141492 -1.415  .0786
OQSO for d1bin           using bits  5 to  9        141793  -.394  .3467
OQSO for d1bin           using bits  4 to  8        141470 -1.489  .0682
OQSO for d1bin           using bits  3 to  7        141975   .223  .5881
OQSO for d1bin           using bits  2 to  6        142271  1.226  .8899
OQSO for d1bin           using bits  1 to  5        142360  1.528  .9367

 DNA test for generator d1bin
  Output: No. missing words (mw), equiv normal variate (z), p-value (p)
                                                     mw       z      p
DNA for d1bin           using bits 31 to 32        141267 -1.895  .0291
DNA for d1bin           using bits 30 to 31        141930   .061  .5243
DNA for d1bin           using bits 29 to 30        141825  -.249  .4018
DNA for d1bin           using bits 28 to 29        141922   .037  .5149
```

```
DNA for d1bin           using bits 27 to 28       141978   .203  .5803
DNA for d1bin           using bits 26 to 27       141216 -2.045  .0204
DNA for d1bin           using bits 25 to 26       142480  1.683  .9539
DNA for d1bin           using bits 24 to 25       141992   .244  .5963
DNA for d1bin           using bits 23 to 24       141586  -.954  .1701
DNA for d1bin           using bits 22 to 23       141217 -2.042  .0206
DNA for d1bin           using bits 21 to 22       142037   .377  .6468
DNA for d1bin           using bits 20 to 21       141911   .005  .5020
DNA for d1bin           using bits 19 to 20       141830  -.234  .4075
DNA for d1bin           using bits 18 to 19       142218   .911  .8187
DNA for d1bin           using bits 17 to 18       141622  -.848  .1983
DNA for d1bin           using bits 16 to 17       141898  -.033  .4867
DNA for d1bin           using bits 15 to 16       141643  -.786  .2160
DNA for d1bin           using bits 14 to 15       142003   .276  .6088
DNA for d1bin           using bits 13 to 14       142343  1.279  .8996
DNA for d1bin           using bits 12 to 13       141789  -.355  .3613
DNA for d1bin           using bits 11 to 12       142061   .447  .6727
DNA for d1bin           using bits 10 to 11       142545  1.875  .9696
DNA for d1bin           using bits  9 to 10       142239   .972  .8346
DNA for d1bin           using bits  8 to  9       141574  -.989  .1613
DNA for d1bin           using bits  7 to  8       142098   .557  .7111
DNA for d1bin           using bits  6 to  7       141882  -.081  .4679
DNA for d1bin           using bits  5 to  6       141974   .191  .5756
DNA for d1bin           using bits  4 to  5       141523 -1.140  .1272
DNA for d1bin           using bits  3 to  4       141634  -.812  .2083
DNA for d1bin           using bits  2 to  3       141564 -1.019  .1542
DNA for d1bin           using bits  1 to  2       141949   .117  .5466

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::     This is the COUNT-THE-1's TEST on a stream of bytes.      ::
     :: Consider the file under test as a stream of bytes (four per   ::
     :: 32 bit integer).  Each byte can contain from 0 to 8 1's,      ::
     :: with probabilities 1,8,28,56,70,56,28,8,1 over 256.  Now let  ::
     :: the stream of bytes provide a string of overlapping  5-letter ::
     :: words, each "letter" taking values A,B,C,D,E. The letters are ::
     :: determined by the number of 1's in a byte::  0,1,or 2 yield A,::
     :: 3 yields B, 4 yields C, 5 yields D and 6,7 or 8 yield E. Thus ::
     :: we have a monkey at a typewriter hitting five keys with vari- ::
     :: ous probabilities (37,56,70,56,37 over 256).  There are 5^5   ::
     :: possible 5-letter words, and from a string of 256,000 (over-  ::
     :: lapping) 5-letter words, counts are made on the frequencies   ::
     :: for each word.   The quadratic form in the weak inverse of    ::
     :: the covariance matrix of the cell counts provides a chisquare ::
     :: test::  Q5-Q4, the difference of the naive Pearson sums of    ::
     :: (OBS-EXP)^2/EXP on counts for 5- and 4-letter cell counts.    ::
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
   Test results for d1bin
 Chi-square with 5^5-5^4=2500 d.of f. for sample size:2560000
                              chisquare  equiv normal  p-value
  Results fo COUNT-THE-1's in successive bytes:
 byte stream for d1bin           2446.13      -.762      .223091
 byte stream for d1bin           2545.71       .646      .741019

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

```
    ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
    ::     This is the COUNT-THE-1's TEST for specific bytes.        ::
    :: Consider the file under test as a stream of 32-bit integers.  ::
    :: From each integer, a specific byte is chosen , say the left-  ::
    :: most::  bits 1 to 8. Each byte can contain from 0 to 8 1's,   ::
    :: with probabilitie 1,8,28,56,70,56,28,8,1 over 256.  Now let   ::
    :: the specified bytes from successive integers provide a string ::
    :: of (overlapping) 5-letter words, each "letter" taking values  ::
    :: A,B,C,D,E. The letters are determined  by the number of 1's,  ::
    :: in that byte::  0,1,or 2 ---> A, 3 ---> B, 4 ---> C, 5 ---> D,::
    :: and  6,7 or 8 ---> E.  Thus we have a monkey at a typewriter  ::
    :: hitting five keys with with various probabilities::  37,56,70,::
    :: 56,37 over 256. There are 5^5 possible 5-letter words, and    ::
    :: from a string of 256,000 (overlapping) 5-letter words, counts ::
    :: are made on the frequencies for each word. The quadratic form ::
    :: in the weak inverse of the covariance matrix of the cell      ::
    :: counts provides a chisquare test::  Q5-Q4, the difference of  ::
    :: the naive Pearson  sums of (OBS-EXP)^2/EXP on counts for 5-   ::
    :: and 4-letter cell counts.                                     ::
    ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
Chi-square with 5^5-5^4=2500 d.of f. for sample size: 256000
                     chisquare  equiv normal  p value
 Results for COUNT-THE-1's in specified bytes:
          bits  1 to  8  2506.19       .088      .534866
          bits  2 to  9  2513.90       .197      .577936
          bits  3 to 10  2455.18      -.634      .263092
          bits  4 to 11  2478.94      -.298      .382929
          bits  5 to 12  2466.89      -.468      .319808
          bits  6 to 13  2516.40       .232      .591725
          bits  7 to 14  2518.41       .260      .602721
          bits  8 to 15  2523.63       .334      .630859
          bits  9 to 16  2489.03      -.155      .438370
          bits 10 to 17  2540.37       .571      .715986
          bits 11 to 18  2508.98       .127      .550502
          bits 12 to 19  2388.93     -1.571      .058118
          bits 13 to 20  2478.18      -.309      .378817
          bits 14 to 21  2600.21      1.417      .921783
          bits 15 to 22  2562.88       .889      .813070
          bits 16 to 23  2424.03     -1.074      .141340
          bits 17 to 24  2507.55       .107      .542502
          bits 18 to 25  2433.16      -.945      .172254
          bits 19 to 26  2444.37      -.787      .215701
          bits 20 to 27  2499.65      -.005      .497997
          bits 21 to 28  2427.56     -1.024      .152823
          bits 22 to 29  2525.05       .354      .638408
          bits 23 to 30  2502.74       .039      .515476
          bits 24 to 31  2642.83      2.020      .978301
          bits 25 to 32  2500.56       .008      .503154
```

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::                THIS IS A PARKING LOT TEST                     ::
     :: In a square of side 100, randomly "park" a car---a circle of  ::
     :: radius 1.   Then try to park a 2nd, a 3rd, and so on, each    ::
     :: time parking "by ear".  That is, if an attempt to park a car  ::
     :: causes a crash with one already parked, try again at a new    ::
     :: random location. (To avoid path problems, consider parking    ::
     :: helicopters rather than cars.)   Each attempt leads to either ::
     :: a crash or a success, the latter followed by an increment to  ::
     :: the list of cars already parked. If we plot n:  the number of ::
     :: attempts, versus k::  the number successfully parked, we get a::
     :: curve that should be similar to those provided by a perfect   ::
     :: random number generator.  Theory for the behavior of such a   ::
     :: random curve seems beyond reach, and as graphics displays are ::
     :: not available for this battery of tests, a simple characteriz ::
     :: ation of the random experiment is used: k, the number of cars ::
     :: successfully parked after n=12,000 attempts. Simulation shows ::
     :: that k should average 3523 with sigma 21.9 and is very close  ::
     :: to normally distributed.  Thus (k-3523)/21.9 should be a st-  ::
     :: andard normal variable, which, converted to a uniform varia-  ::
     :: ble, provides input to a KSTEST based on a sample of 10.      ::
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
           CDPARK: result of ten tests on file d1bin
            Of 12,000 tries, the average no. of successes
                 should be 3523 with sigma=21.9
            Successes: 3525    z-score:   .091 p-value: .536382
            Successes: 3514    z-score:  -.411 p-value: .340551
            Successes: 3531    z-score:   .365 p-value: .642555
            Successes: 3559    z-score:  1.644 p-value: .949895
            Successes: 3522    z-score:  -.046 p-value: .481790
            Successes: 3550    z-score:  1.233 p-value: .891189
            Successes: 3520    z-score:  -.137 p-value: .445521
            Successes: 3557    z-score:  1.553 p-value: .939730
            Successes: 3519    z-score:  -.183 p-value: .427537
            Successes: 3563    z-score:  1.826 p-value: .966111

           square size   avg. no.  parked   sample sigma
             100.            3536.000       18.067
            KSTEST for the above 10: p=  .925296

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::               THE MINIMUM DISTANCE TEST                       ::
     :: It does this 100 times::   choose n=8000 random points in a   ::
     :: square of side 10000.  Find d, the minimum distance between   ::
     :: the (n^2-n)/2 pairs of points.  If the points are truly inde- ::
     :: pendent uniform, then d^2, the square of the minimum distance ::
     :: should be (very close to) exponentially distributed with mean ::
     :: .995 .  Thus 1-exp(-d^2/.995) should be uniform on [0,1) and  ::
     :: a KSTEST on the resulting 100 values serves as a test of uni- ::
     :: formity for random points in the square. Test numbers=0 mod 5 ::
     :: are printed but the KSTEST is based on the full set of 100    ::
     :: random choices of 8000 points in the 10000x10000 square.      ::
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
               This is the MINIMUM DISTANCE test
```

```
            for random integers in the file d1bin
     Sample no.    d^2     avg     equiv uni
           5     .3179   1.0223     .273513
          10     .5802   1.1477     .441837
          15    4.0468   1.3053     .982873
          20    2.5662   1.2760     .924159
          25    1.4633   1.2612     .770220
          30     .3392   1.1946     .288856
          35     .4344   1.1244     .353728
          40     .0180   1.1647     .017958
          45     .7437   1.1456     .526444
          50     .2360   1.0942     .211180
          55    1.7649   1.0528     .830316
          60    3.0577   1.0567     .953720
          65     .0864   1.0489     .083142
          70    1.3914   1.0625     .753018
          75    1.6058   1.1034     .800886
          80     .1231   1.0799     .116414
          85     .1280   1.0696     .120752
          90    2.7471   1.1285     .936768
          95    2.4284   1.1296     .912890
         100     .2291   1.1037     .205627
     MINIMUM DISTANCE TEST for d1bin
          Result of KS test on 20 transformed mindist^2's:
                                 p-value= .820154

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::                THE 3DSPHERES TEST                              ::
     :: Choose  4000 random points in a cube of edge 1000.  At each    ::
     :: point, center a sphere large enough to reach the next closest ::
     :: point. Then the volume of the smallest such sphere is (very    ::
     :: close to) exponentially distributed with mean 120pi/3.  Thus   ::
     :: the radius cubed is exponential with mean 30. (The mean is     ::
     :: obtained by extensive simulation).  The 3DSPHERES test gener- ::
     :: ates 4000 such spheres 20 times.  Each min radius cubed leads ::
     :: to a uniform variable by means of 1-exp(-r^3/30.), then a      ::
     ::  KSTEST is done on the 20 p-values.                            ::
     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
             The 3DSPHERES test for file d1bin
 sample no:  1     r^3= 102.828     p-value= .96753
 sample no:  2     r^3=  38.879     p-value= .72637
 sample no:  3     r^3=  38.802     p-value= .72566
 sample no:  4     r^3=  11.399     p-value= .31613
 sample no:  5     r^3=  14.289     p-value= .37893
 sample no:  6     r^3=   5.748     p-value= .17437
 sample no:  7     r^3=  21.723     p-value= .51525
 sample no:  8     r^3=  23.658     p-value= .54551
 sample no:  9     r^3=  87.443     p-value= .94578
 sample no: 10     r^3=  59.161     p-value= .86083
 sample no: 11     r^3=  48.727     p-value= .80294
 sample no: 12     r^3=   4.125     p-value= .12846
 sample no: 13     r^3=   2.826     p-value= .08991
 sample no: 14     r^3=  13.584     p-value= .36415
 sample no: 15     r^3=  39.334     p-value= .73049
```

```
 sample no: 16     r^3=  31.727     p-value= .65270
 sample no: 17     r^3=  31.423     p-value= .64916
 sample no: 18     r^3=   2.751     p-value= .08762
 sample no: 19     r^3=  27.775     p-value= .60379
 sample no: 20     r^3=  22.301     p-value= .52450
  A KS test is applied to those 20 p-values.
---------------------------------------------------------
       3DSPHERES test for file d1bin               p-value= .183216
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::      This is the SQEEZE test                                  ::
     ::  Random integers are floated to get uniforms on [0,1). Start- ::
     ::  ing with k=2^31=2147483647, the test finds j, the number of  ::
     ::  iterations necessary to reduce k to 1, using the reduction   ::
     ::  k=ceiling(k*U), with U provided by floating integers from    ::
     ::  the file being tested.  Such j's are found 100,000 times,    ::
     ::  then counts for the number of times j was <=6,7,...,47,>=48  ::
     ::  are used to provide a chi-square test for cell frequencies.  ::
     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
            RESULTS OF SQUEEZE TEST FOR d1bin
         Table of standardized frequency counts
     ( (obs-exp)/sqrt(exp) )^2
        for j taking values <=6,7,8,...,47,>=48:
     -.8     1.8     1.3    -1.4     1.0     -.6
     -.3    -1.5     -.8     -.5     1.0     -.8
      .2     -.5    -2.4     1.4    -2.1     -.2
     -.6     -.5     1.6     1.9     1.1      .8
     1.5     -.1     -.5     1.1     1.3     -.4
     -.5     -.1     1.0     -.8     -.2     -.1
     1.7     -.4      .1     1.5     -.6      .0
    -1.1
           Chi-square with 42 degrees of freedom: 48.686
              z-score=   .729  p-value= .778338
______________________________________________________________

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::             The  OVERLAPPING SUMS test                        ::
     :: Integers are floated to get a sequence U(1),U(2),... of uni-  ::
     :: form [0,1) variables.  Then overlapping sums,                 ::
     ::   S(1)=U(1)+...+U(100), S2=U(2)+...+U(101),... are formed.    ::
     :: The S's are virtually normal with a certain covariance mat-   ::
     :: rix.  A linear transformation of the S's converts them to a   ::
     :: sequence of independent standard normals, which are converted ::
     :: to uniform variables for a KSTEST. The  p-values from ten     ::
     :: KSTESTs are given still another KSTEST.                       ::
     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
                Test no.  1      p-value  .328789
                Test no.  2      p-value  .628282
                Test no.  3      p-value  .490585
                Test no.  4      p-value  .356692
                Test no.  5      p-value  .836217
                Test no.  6      p-value  .176251
                Test no.  7      p-value  .887524
```

```
              Test no.  8      p-value  .897336
              Test no.  9      p-value  .179966
              Test no. 10      p-value  .175412
   Results of the OSUM test for d1bin
        KSTEST on the above 10 p-values:  .159254

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::     This is the RUNS test.  It counts runs up, and runs down, ::
     :: in a sequence of uniform [0,1) variables, obtained by float-  ::
     :: ing the 32-bit integers in the specified file. This example   ::
     :: shows how runs are counted:  .123,.357,.789,.425,.224,.416,.95::
     :: contains an up-run of length 3, a down-run of length 2 and an ::
     :: up-run of (at least) 2, depending on the next values.  The    ::
     :: covariance matrices for the runs-up and runs-down are well    ::
     :: known, leading to chisquare tests for quadratic forms in the  ::
     :: weak inverses of the covariance matrices.  Runs are counted   ::
     :: for sequences of length 10,000.  This is done ten times. Then ::
     :: repeated.                                                     ::
     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
           The RUNS test for file d1bin
     Up and down runs in a sample of 10000
_________________________________________________
                 Run test for d1bin         :
       runs up; ks test for 10 p's: .242339
     runs down; ks test for 10 p's: .750346
                 Run test for d1bin         :
       runs up; ks test for 10 p's: .930279
     runs down; ks test for 10 p's: .587326

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     :: This is the CRAPS TEST. It plays 200,000 games of craps, finds::
     :: the number of wins and the number of throws necessary to end  ::
     :: each game.  The number of wins should be (very close to) a    ::
     :: normal with mean 200000p and variance 200000p(1-p), with      ::
     :: p=244/495.  Throws necessary to complete the game can vary    ::
     :: from 1 to infinity, but counts for all>21 are lumped with 21. ::
     :: A chi-square test is made on the no.-of-throws cell counts.   ::
     :: Each 32-bit integer from the test file provides the value for ::
     :: the throw of a die, by floating to [0,1), multiplying by 6    ::
     :: and taking 1 plus the integer part of the result.            ::
     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
              Results of craps test for d1bin
  No. of wins:  Observed Expected
                                98212    98585.86
                  98212= No. of wins, z-score=-1.672 pvalue= .04725
   Analysis of Throws-per-Game:
 Chisq=  28.97 for 20 degrees of freedom, p=  .91165
               Throws Observed Expected  Chisq     Sum
                  1    66818    66666.7    .344     .344
                  2    37465    37654.3    .952    1.295
                  3    26708    26954.7   2.258    3.554
                  4    19630    19313.5   5.188    8.742
```

5	13942	13851.4	.592	9.334
6	9888	9943.5	.310	9.644
7	6932	7145.0	6.351	15.996
8	5185	5139.1	.410	16.406
9	3830	3699.9	4.577	20.983
10	2658	2666.3	.026	21.009
11	1946	1923.3	.267	21.276
12	1375	1388.7	.136	21.412
13	980	1003.7	.560	21.973
14	720	726.1	.052	22.025
15	538	525.8	.281	22.306
16	359	381.2	1.287	23.593
17	285	276.5	.259	23.852
18	204	200.8	.050	23.902
19	165	146.0	2.477	26.379
20	111	106.2	.216	26.595
21	261	287.1	2.375	28.970

```
SUMMARY  FOR d1bin
    p-value for no. of wins: .047251
    p-value for throws/game: .911650
```

$$

Results of DIEHARD battery of tests sent to file d1out

Appendix D. Diehard Test Results for DLL-based RNG with the Modified Von Neumann Corrector

```
    :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
    ::            This is the BIRTHDAY SPACINGS TEST                 ::
    :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::

  The 9 p-values were
       .852559   .579032   .202074   .140017   .726268
       .449922   .313803   .034697   .648904
 A KSTEST for the 9 p-values yields  .099358

    :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
    ::            THE OVERLAPPING 5-PERMUTATION TEST                 ::
    :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::

          OPERM5 test for file d11bin
    For a sample of 1,000,000 consecutive 5-tuples,
chisquare for 99 degrees of freedom= 98.897; p-value= .515964
          OPERM5 test for file d11bin
    For a sample of 1,000,000 consecutive 5-tuples,
chisquare for 99 degrees of freedom= 72.910; p-value= .022821

    :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
    :: This is the BINARY RANK TEST for 31x31 matrices.              ::
    :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::

   Binary rank test for d11bin
        Rank test for 31x31 binary matrices:
       rows from leftmost 31 bits of each 32-bit integer
     rank   observed  expected (o-e)^2/e  sum
       28       216     211.4   .099304     .099
       29      5103    5134.0   .187307     .287
       30     23011   23103.0   .366732     .653
       31     11670   11551.5  1.215118    1.868
 chisquare= 1.868 for 3 d. of f.; p-value= .485675

     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
    :: This is the BINARY RANK TEST for 32x32 matrices               ::
    :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::

   Binary rank test for d11bin
        Rank test for 32x32 binary matrices:
       rows from leftmost 32 bits of each 32-bit integer
     rank   observed  expected (o-e)^2/e  sum
       29       204     211.4   .260276     .260
       30      5288    5134.0  4.618776    4.879
       31     22912   23103.0  1.579831    6.459
       32     11596   11551.5   .171240    6.630
 chisquare= 6.630 for 3 d. of f.; p-value= .920344
```

```
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     :: This is the BINARY RANK TEST for 6x8 matrices                  ::
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::

   TEST SUMMARY, 25 tests on 100,000 random 6x8 matrices
 These should be 25 uniform [0,1] random variables:
     .547019     .518735     .409699     .593021     .831824
     .157193     .826177     .833724     .840834     .688300
     .964288     .575887     .478705     .306609     .725766
     .902957     .275850     .522065     .182574     .401655
     .184181     .187329     .906712     .440660     .346315
   brank test summary for d11bin
       The KS test for those 25 supposed UNI's yields
                    KS p-value= .507025

     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::                   THE BITSTREAM TEST                           ::
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::

 THE OVERLAPPING 20-tuples BITSTREAM  TEST, 20 BITS PER WORD, N words
   This test uses N=2^21 and samples the bitstream 20 times.
  No. missing words should average  141909. with sigma=428.
---------------------------------------------------------
tst no 1:  142019 missing words,    .26 sigmas from mean, p-value= .60112
tst no 2:  141574 missing words,   -.78 sigmas from mean, p-value= .21667
tst no 3:  142154 missing words,    .57 sigmas from mean, p-value= .71622
tst no 4:  141531 missing words,   -.88 sigmas from mean, p-value= .18836
tst no 5:  142487 missing words,   1.35 sigmas from mean, p-value= .91144
tst no 6:  141870 missing words,   -.09 sigmas from mean, p-value= .46339
tst no 7:  142066 missing words,    .37 sigmas from mean, p-value= .64284
tst no 8:  141397 missing words,  -1.20 sigmas from mean, p-value= .11565
tst no 9:  141829 missing words,   -.19 sigmas from mean, p-value= .42556
tst no 10: 142228 missing words,    .74 sigmas from mean, p-value= .77173
tst no 11: 142024 missing words,    .27 sigmas from mean, p-value= .60562
tst no 12: 141510 missing words,   -.93 sigmas from mean, p-value= .17541
tst no 13: 142219 missing words,    .72 sigmas from mean, p-value= .76532
tst no 14: 141472 missing words,  -1.02 sigmas from mean, p-value= .15344
tst no 15: 141093 missing words,  -1.91 sigmas from mean, p-value= .02824
tst no 16: 141620 missing words,   -.68 sigmas from mean, p-value= .24952
tst no 17: 142570 missing words,   1.54 sigmas from mean, p-value= .93866
tst no 18: 141981 missing words,    .17 sigmas from mean, p-value= .56650
tst no 19: 141236 missing words,  -1.57 sigmas from mean, p-value= .05784
tst no 20: 141344 missing words,  -1.32 sigmas from mean, p-value= .09327


     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::             The tests OPSO, OQSO and DNA                       ::
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::

OPSO test for generator d11bin
Output: No. missing words (mw), equiv normal variate (z), p-value (p)
                                                      mw     z     p
OPSO for d11bin         using bits 23 to 32       141813  -.332  .3699
OPSO for d11bin         using bits 22 to 31       141973   .220  .5869
OPSO for d11bin         using bits 21 to 30       141547 -1.249  .1058
OPSO for d11bin         using bits 20 to 29       142048   .478  .6837
```

```
OPSO for d11bin          using bits 19 to 28        141788  -.418  .3378
OPSO for d11bin          using bits 18 to 27        141578 -1.143  .1266
OPSO for d11bin          using bits 17 to 26        141945   .123  .5489
OPSO for d11bin          using bits 16 to 25        141627  -.974  .1651
OPSO for d11bin          using bits 15 to 24        141952   .147  .5585
OPSO for d11bin          using bits 14 to 23        141735  -.601  .2739
OPSO for d11bin          using bits 13 to 22        141845  -.222  .4122
OPSO for d11bin          using bits 12 to 21        141548 -1.246  .1064
OPSO for d11bin          using bits 11 to 20        141590 -1.101  .1354
OPSO for d11bin          using bits 10 to 19        142494  2.016  .9781
OPSO for d11bin          using bits  9 to 18        142516  2.092  .9818
OPSO for d11bin          using bits  8 to 17        142128   .754  .7746
OPSO for d11bin          using bits  7 to 16        141860  -.170  .4325
OPSO for d11bin          using bits  6 to 15        142343  1.495  .9326
OPSO for d11bin          using bits  5 to 14        141721  -.649  .2580
OPSO for d11bin          using bits  4 to 13        141980   .244  .5963
OPSO for d11bin          using bits  3 to 12        141895  -.049  .4803
OPSO for d11bin          using bits  2 to 11        142009   .344  .6345
OPSO for d11bin          using bits  1 to 10        141819  -.311  .3777

OQSO test for generator d11bin
Output: No. missing words (mw), equiv normal variate (z), p-value (p)
                                                        mw     z      p
OQSO for d11bin          using bits 28 to 32        141721  -.638  .2616
OQSO for d11bin          using bits 27 to 31        141899  -.035  .4860
OQSO for d11bin          using bits 26 to 30        141962   .179  .5709
OQSO for d11bin          using bits 25 to 29        142296  1.311  .9050
OQSO for d11bin          using bits 24 to 28        141680  -.777  .2185
OQSO for d11bin          using bits 23 to 27        141820  -.303  .3810
OQSO for d11bin          using bits 22 to 26        142216  1.040  .8507
OQSO for d11bin          using bits 21 to 25        141535 -1.269  .1022
OQSO for d11bin          using bits 20 to 24        141616  -.994  .1600
OQSO for d11bin          using bits 19 to 23        142118   .707  .7603
OQSO for d11bin          using bits 18 to 22        141758  -.513  .3040
OQSO for d11bin          using bits 17 to 21        141980   .240  .5947
OQSO for d11bin          using bits 16 to 20        142213  1.029  .8484
OQSO for d11bin          using bits 15 to 19        142325  1.409  .9206
OQSO for d11bin          using bits 14 to 18        142224  1.067  .8569
OQSO for d11bin          using bits 13 to 17        141432 -1.618  .0528
OQSO for d11bin          using bits 12 to 16        141657  -.855  .1962
OQSO for d11bin          using bits 11 to 15        142210  1.019  .8460
OQSO for d11bin          using bits 10 to 14        142018   .368  .6437
OQSO for d11bin          using bits  9 to 13        141953   .148  .5588
OQSO for d11bin          using bits  8 to 12        142338  1.453  .9269
OQSO for d11bin          using bits  7 to 11        142048   .470  .6808
OQSO for d11bin          using bits  6 to 10        141670  -.811  .2086
OQSO for d11bin          using bits  5 to  9        142260  1.189  .8827
OQSO for d11bin          using bits  4 to  8        142096   .633  .7366
OQSO for d11bin          using bits  3 to  7        141775  -.455  .3244
OQSO for d11bin          using bits  2 to  6        142027   .399  .6550
OQSO for d11bin          using bits  1 to  5        141647  -.889  .1869

DNA test for generator d11bin
Output: No. missing words (mw), equiv normal variate (z), p-value (p)
                                                      mw     z      p
DNA for d11bin          using bits 31 to 32        141513 -1.169  .1212
```

```
DNA for d11bin          using bits 30 to 31       141493 -1.228  .1097
DNA for d11bin          using bits 29 to 30       141291 -1.824  .0341
DNA for d11bin          using bits 28 to 29       142129   .648  .7415
DNA for d11bin          using bits 27 to 28       142081   .506  .6937
DNA for d11bin          using bits 26 to 27       141911   .005  .5020
DNA for d11bin          using bits 25 to 26       142282  1.099  .8642
DNA for d11bin          using bits 24 to 25       142138   .675  .7500
DNA for d11bin          using bits 23 to 24       142305  1.167  .8784
DNA for d11bin          using bits 22 to 23       142045   .400  .6555
DNA for d11bin          using bits 21 to 22       141927   .052  .5208
DNA for d11bin          using bits 20 to 21       141476 -1.278  .1006
DNA for d11bin          using bits 19 to 20       142027   .347  .6357
DNA for d11bin          using bits 18 to 19       142313  1.191  .8831
DNA for d11bin          using bits 17 to 18       141683  -.668  .2522
DNA for d11bin          using bits 16 to 17       141827  -.243  .4041
DNA for d11bin          using bits 15 to 16       141956   .138  .5548
DNA for d11bin          using bits 14 to 15       141491 -1.234  .1086
DNA for d11bin          using bits 13 to 14       141930   .061  .5243
DNA for d11bin          using bits 12 to 13       142158   .734  .7684
DNA for d11bin          using bits 11 to 12       142387  1.409  .9206
DNA for d11bin          using bits 10 to 11       142294  1.135  .8718
DNA for d11bin          using bits  9 to 10       141806  -.305  .3803
DNA for d11bin          using bits  8 to  9       142337  1.262  .8964
DNA for d11bin          using bits  7 to  8       141922   .037  .5149
DNA for d11bin          using bits  6 to  7       141524 -1.137  .1278
DNA for d11bin          using bits  5 to  6       142216   .905  .8172
DNA for d11bin          using bits  4 to  5       141975   .194  .5768
DNA for d11bin          using bits  3 to  4       141729  -.532  .2974
DNA for d11bin          using bits  2 to  3       141440 -1.384  .0831
DNA for d11bin          using bits  1 to  2       141431 -1.411  .0791

     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::     This is the COUNT-THE-1's TEST on a stream of bytes.     ::
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::

   Test results for d11bin
 Chi-square with 5^5-5^4=2500 d.of f. for sample size:2560000
                              chisquare  equiv normal  p-value
  Results fo COUNT-THE-1's in successive bytes:
 byte stream for d11bin           2633.78      1.892      .970754
 byte stream for d11bin           2526.02       .368      .643570

     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::     This is the COUNT-THE-1's TEST for specific bytes.       ::
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::

 Chi-square with 5^5-5^4=2500 d.of f. for sample size: 256000
                         chisquare  equiv normal  p value
  Results for COUNT-THE-1's in specified bytes:
           bits  1 to  8  2502.74       .039      .515480
           bits  2 to  9  2507.58       .107      .542711
           bits  3 to 10  2638.79      1.963      .975161
           bits  4 to 11  2640.33      1.985      .976405
           bits  5 to 12  2595.06      1.344      .910582
           bits  6 to 13  2626.80      1.793      .963535
           bits  7 to 14  2587.76      1.241      .892713
```

```
     bits  8 to 15  2509.95       .141      .555966
     bits  9 to 16  2473.72      -.372      .355097
     bits 10 to 17  2492.51      -.106      .457798
     bits 11 to 18  2507.87       .111      .544289
     bits 12 to 19  2364.14    -1.921      .027341
     bits 13 to 20  2518.69       .264      .604222
     bits 14 to 21  2564.83       .917      .820375
     bits 15 to 22  2501.72       .024      .509696
     bits 16 to 23  2596.05      1.358      .912821
     bits 17 to 24  2547.50       .672      .749130
     bits 18 to 25  2405.25    -1.340      .090121
     bits 19 to 26  2609.88      1.554      .939902
     bits 20 to 27  2531.23       .442      .670642
     bits 21 to 28  2524.29       .343      .634377
     bits 22 to 29  2405.70    -1.334      .091166
     bits 23 to 30  2431.60      -.967      .166694
     bits 24 to 31  2418.52    -1.152      .124603
     bits 25 to 32  2496.66      -.047      .481167

::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
::              THIS IS A PARKING LOT TEST                      ::
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::

      CDPARK: result of ten tests on file d11bin
       Of 12,000 tries, the average no. of successes
            should be 3523 with sigma=21.9
       Successes: 3514    z-score:  -.411 p-value: .340551
       Successes: 3510    z-score:  -.594 p-value: .276387
       Successes: 3543    z-score:   .913 p-value: .819442
       Successes: 3512    z-score:  -.502 p-value: .307734
       Successes: 3538    z-score:   .685 p-value: .753306
       Successes: 3493    z-score: -1.370 p-value: .085365
       Successes: 3538    z-score:   .685 p-value: .753306
       Successes: 3499    z-score: -1.096 p-value: .136563
       Successes: 3524    z-score:   .046 p-value: .518210
       Successes: 3540    z-score:   .776 p-value: .781201

      square size   avg. no.  parked   sample sigma
        100.            3521.100       17.193
       KSTEST for the above 10: p=  .150814

::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
::              THE MINIMUM DISTANCE TEST                       ::
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::

          This is the MINIMUM DISTANCE test
         for random integers in the file d11bin
Sample no.    d^2     avg     equiv uni
      5     .0325    .9348     .032101
     10     .0252    .9132     .024990
     15     .5097    .8756     .400834
     20     .8975    .8627     .594252
     25     .6175   1.0155     .462363
     30    1.7802   1.3062     .832893
     35     .0373   1.2388     .036832
     40    3.0709   1.3279     .954333
```

```
         45     .6012   1.3276     .453494
         50    2.7984   1.2860     .939946
         55     .0409   1.1847     .040281
         60     .7388   1.1551     .524094
         65     .2785   1.1268     .244149
         70     .0470   1.1216     .046166
         75     .4693   1.0770     .376056
         80     .0395   1.0410     .038903
         85     .0351    .9958     .034691
         90    2.0689   1.0192     .874983
         95     .0074   1.0169     .007437
        100     .0419   1.0233     .041233
     MINIMUM DISTANCE TEST for d11bin
          Result of KS test on 20 transformed mindist^2's:
                                p-value= .862979

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::              THE 3DSPHERES TEST                              ::
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::

               The 3DSPHERES test for file d11bin
 sample no:  1     r^3=   4.011     p-value= .12515
 sample no:  2     r^3=  56.222     p-value= .84650
 sample no:  3     r^3=  30.593     p-value= .63933
 sample no:  4     r^3=  15.715     p-value= .40775
 sample no:  5     r^3=  49.868     p-value= .81029
 sample no:  6     r^3=  32.296     p-value= .65923
 sample no:  7     r^3=  82.573     p-value= .93623
 sample no:  8     r^3=  28.254     p-value= .61008
 sample no:  9     r^3=  19.457     p-value= .47720
 sample no: 10     r^3=  19.992     p-value= .48645
 sample no: 11     r^3=  11.267     p-value= .31310
 sample no: 12     r^3=  18.247     p-value= .45570
 sample no: 13     r^3=   2.726     p-value= .08686
 sample no: 14     r^3=  11.050     p-value= .30811
 sample no: 15     r^3=  12.423     p-value= .33907
 sample no: 16     r^3=  19.996     p-value= .48652
 sample no: 17     r^3=  12.793     p-value= .34716
 sample no: 18     r^3=  20.675     p-value= .49801
 sample no: 19     r^3=  18.166     p-value= .45422
 sample no: 20     r^3=   8.134     p-value= .23750
  A KS test is applied to those 20 p-values.
---------------------------------------------------------
       3DSPHERES test for file d11bin              p-value= .654979


     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::      This is the SQEEZE test                                 ::
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
           RESULTS OF SQUEEZE TEST FOR d11bin
        Table of standardized frequency counts
     ( (obs-exp)/sqrt(exp) )^2
       for j taking values <=6,7,8,...,47,>=48:
    -1.5     -.7      .1    -1.0     -.1      .3
```

```
     .2     -.8      .6      .4      .2     -.6
    -.6      .6    -1.3     -.9      .3      .0
     .9     -.4      .5     -.3     1.8     -.6
   -1.6      .1      .4      .9      .6     1.3
    -.3     -.7      .6     2.0      .8      .1
    -.2     -.1      .1    -1.3      .9      .0
   -1.1
          Chi-square with 42 degrees of freedom: 27.584
             z-score= -1.573  p-value= .042261

    :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
    ::             The  OVERLAPPING SUMS test                        ::
    :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
               Test no.  1      p-value  .476134
               Test no.  2      p-value  .569045
               Test no.  3      p-value  .300827
               Test no.  4      p-value  .062252
               Test no.  5      p-value  .083605
               Test no.  6      p-value  .569161
               Test no.  7      p-value  .499716
               Test no.  8      p-value  .700330
               Test no.  9      p-value  .649824
               Test no. 10      p-value  .340612
  Results of the OSUM test for d11bin
       KSTEST on the above 10 p-values:  .600510

    :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
    ::     This is the RUNS test.                                    ::
    :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
          The RUNS test for file d11bin
    Up and down runs in a sample of 10000
_________________________________________________
                Run test for d11bin        :
      runs up; ks test for 10 p's: .744478
    runs down; ks test for 10 p's: .924207
                Run test for d11bin        :
      runs up; ks test for 10 p's: .180330
    runs down; ks test for 10 p's: .899578

    :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
    :: This is the CRAPS TEST.                                       ::
    :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
               Results of craps test for d11bin
 No. of wins:  Observed Expected
                               98872    98585.86
                 98872= No. of wins, z-score= 1.280 pvalue= .89969
  Analysis of Throws-per-Game:
Chisq=  19.19 for 20 degrees of freedom, p=  .49042
              Throws Observed Expected  Chisq     Sum
                 1    66868    66666.7    .608     .608
                 2    37479    37654.3    .816    1.424
                 3    27091    26954.7    .689    2.113
                 4    19445    19313.5    .896    3.009
                 5    13863    13851.4    .010    3.019
                 6     9810     9943.5   1.793    4.812
                 7     7207     7145.0    .538    5.350
```

```
         8     4994     5139.1   4.095    9.445
         9     3691     3699.9    .021    9.466
        10     2591     2666.3   2.126   11.593
        11     1937     1923.3    .097   11.690
        12     1422     1388.7    .797   12.486
        13     1006     1003.7    .005   12.492
        14      729      726.1    .011   12.503
        15      518      525.8    .117   12.620
        16      361      381.2   1.065   13.685
        17      298      276.5   1.665   15.350
        18      187      200.8    .952   16.303
        19      143      146.0    .061   16.364
        20       97      106.2    .800   17.163
        21      263      287.1   2.026   19.189
  SUMMARY  FOR d11bin
      p-value for no. of wins: .899690
      p-value for throws/game: .490416

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

 Results of DIEHARD battery of tests sent to file d11out
```

References

[1] J. Savoj and B. Razavi, "A 10-Gb/s CMOS clock and data recovery circuit with a half-rate linear phase detector," IEEE Journal of Solid-State Circuits, vol. 36, pp. 761-767, May 2001.

[2] G. Georgiou, Y. Baeyens, Y. Chen, A. Gnauck, C. Gropper, P. Paschke, R. Pullela, M. Reinhold, C. Dorschky, J. Mattia, T. Mohrenfels, and C. Schulien, "Clock and data recovery IC for 40-Gb/s fiber-optic receiver," IEEE Journal of Solid-State Circuits, vol. 37, pp. 1120–1124, Sep. 2002.

[3] N. Dalt and C. Sandner, "A subpicosecond jitter PLL for clock generation in 0.12μm digital CMOS," IEEE Journal of Solid-State Circuits, vol. 38, pp. 1275–1278, July 2003.

[4] B. Jun and P. Kocher, "The Intel random number generator," Technical Report, Intel, 1999.

[5] M. Bucci, L. Germani, R. Luzzi, A. Trifiletti, and M. Varanonuovo, "A high speed oscillator-based truly random number source for cryptographic applications on a smart card IC," IEEE Transaction on Computers, vol. 52, pp. 403-409, Apr. 2003.

[6] V. Fischer and M. Drutarovsky, "True random number generator embedded in reconfigurable hardware," Cryptographic Hardware and Embedded Systems (CHES 2002), pp. 415-430, 2002.

[7] J. McNeill, "Jitter in ring oscillators," IEEE Journal of Solid-State Circuits, vol. 32, pp. 870-879, June 1997.

[8] J. McNeill, "Jitter in ring oscillators," Ph.D. dissertation, Boston University, 1994.

[9] T. Weigandt, K. Beomsup and P. Gray, “Analysis of timing jitter in CMOS ring oscillators,” Proceedings of 1994 IEEE International Symposium on Circuits and Systems (ISCAS 1994), vol. 4, pp. 27-30, May 1994.

[10] A. Hajimiri and T. H. Lee, “A general theory of phase noise in electrical oscillators,” IEEE Journal of Solid-State Circuits, vol. 33, pp. 179-194, Feb. 1998.

[11] A. Demir, A. Mehrotra, and J. Roychowdhury, “Phase noise in oscillators: A unifying theory and numerical methods for characterization,” IEEE Transaction on Circuits and Systems. I: Fundamental Theory and Applications, vol. 47, pp. 655-674, May 2000.

[12] F. Herzel, “An analytical model for the power spectral density of a valtage-controlled oscillator and its analogy to the laser linewidth theory,” IEEE Transaction on Circuits and Systems I: Fundamental Theory and Applications, vol. 45, pp. 904-908, Sep. 1998.

[13] G. Klimovitch, “Near-carrier oscillator spectrum due to flicker and white noise,” Proceedings of 2000 International Symposium on Circuits and Systems (ISCAS 2000), pp. I-703-706, May 2000.

[14] A. Demir, “Phase noise and timing jitter in oscillators with colored-noise sources,” IEEE Transaction on Circuits and Systems. I: Fundamental Theory and Applications, vol. 49, pp. 1782-1791, Dec. 2002.

[15] A. Hajimiri, S. Limotyrakis, and T. H. Lee, “Jitter and phase noise in ring oscillators,” IEEE Journal of Solid-State Circuits, vol. 34, pp. 790-804, June 1999.

[16] ---, “Synchronous optical network (SONET) transport systems: Common generic criteria, TR-253-CORE,” Bell Communications Research, Inc., Issue 2, Rev No. 1, Dec 1997.

[17] ---, “A guide to understanding and characterizing timing jitter,” Tektronix, Inc., 2002.

[18] ---, “Definitions and terminology for synchronization networks,” ITU-T Recommendation G.810, Aug. 1996.

[19] National committee for information technology standardization (NCITS), “Fibre channel - Methodologies for jitter and signal quality specification - MJSQ”, T11.2 / Project 1316-DT / Rev 12.2, Jan. 30, 2004.

[20] Y. Tsividis, “Operation and Modeling of the MOS Transistor,” New York: McGraw-Hill, 1988.

[21] W. Liu, X. Jin, X. Xi, J. Chen, M. Jeng, Z. Liu, Y. Cheng, K. Chen, M. Chan, K. Hui, J. Huang, R. Tu, P. Ko, and C. Hu, “Users’ Manual, BSIM3v3.3 MOSFET model,” University of California, Berkeley, 2005.

[22] M. Liang, J. Choi, P. Ko, and C. Hu, “Inversion-layer capacitance and mobility of very thin gate-oxide MOSFET’s,” IEEE Transaction on Electron Devices, vol. ED-33, pp. 409-412, Mar. 1986.

[23] K. Toh, P. Ko, and R. Meyer, “An engineering model for short-channel MOS devices,” IEEE Journal of Solid-State Circuits, vol. 23, pp. 950-958, Aug. 1988.

[24] H. Ott, “Noise reduction techniques in electronic systems,” New York: Wiley, 1988.

[25] A. Abidi, “High-frequency noise measurements of FET’s with small dimensions,” IEEE Transaction on Electron Devices, vol. ED-33, pp. 1801-1805, Nov. 1986.

[26] K. Han, H. Shin, and K. Lee, “Analytical drain thermal noise current model valid for deep submicron MOSFETs,” IEEE Transactions on Electron Devices, vol. 51, pp. 261-269, Feb. 2004.

[27] C. Chen and M. Deen, “Channel noise modeling of deep submicron MOSFETs,” IEEE Transaction on Electron Devices, vol. 49, pp. 1484-1487, Aug. 2002.

[28] R. Jindal, “Hot-electron effects on channel thermal noise in file-line NMOS field-effect transistors,” IEEE Transaction on Electron Devices, vol. ED-33, pp. 1395–1397, Sept. 1986.

[29] D. Triantis, A. Birbas, and D. Kondis, “Thermal noise modeling for short-channel MOSFETs,” IEEE Transaction on Electron Devices, vol. 43, pp. 1950–1955, Nov. 1996.

[30] P. Klein, “An analytical thermal noise model of deep submicron MOSFETs,” IEEE Electron Device Letters, vol. 20, pp. 399–401, Aug. 1999.

[31] A. Scholten, H. Tromp, L. Tiemeijer, R. Langevelde, R. Havens, P. Vreede, R. Roes, P. Woerlee, A. Montree, and D. Klaassen, “Accurate thermal noise model for deep-submicron CMOS,” IEDM Technical Digest, pp. 155–158, Dec. 1999.

[32] M. Deen, C. Chen, and Y. Cheng, “MOSFET modeling for low noise, RF circuit design,” in Proceedings of Custom Integrated Circuits Conference, pp. 201–208, May 2002.

[33] K. Han, H. Shin, and K. Lee, “Thermal noise modeling for short-channel MOSFETs,” International Conference on Simulation of Semiconductor Processes and Devices (SISPAD), pp. 79-82, Sep. 2003.

[34] J. Johnson, “The Schottky-effect in low-frequency circuits,” Physical Review, vol. 26, pp. 71-85, July 1925.

[35] A. McWhorter, "1/f noise and germanium surface properties," Semiconductor Surface Physics, p. 207, Philadelphia: University of Pennsylvania Press, 1957.

[36] S. Christensson, I. Lundstrom, and C. Svensson, "Low frequency noise in MOS transistors – I (Theory)," Solid-State Electronics, vol. 11, pp. 797-812, 1968.

[37] F. Berz, "Theory of low frequency noise in Si MOST's," Solid-State Electronics, vol. 13, pp. 631-647, 1970.

[38] S. Hsu, "Surface state related 1/f noise in MOS transistors," Solid-State Electronics, vol. 13, pp. 1451-1459, 1970.

[39] Z. Celik and T. Hsiang, "Study of 1/f noise in N-MOSFET's: Linear region," IEEE Transaction on Electron Devices, vol. ED-32, pp. 2797-2802, 1985.

[40] R. Baker, H. Li, and D. Boyce, "CMOS circuit design, layout, and simulation," New York: Wiley, 1998.

[41] D. Johns and K. Martin, "Analog integrated circuit design," New York: Wiley, 1997.

[42] F. Hooge, "1/f noise is no surface effect," Applied Physics Letters, vol. 29, pp. 139-140, 1969.

[43] L. Vandamme, "Model for 1/f noise in MOS transistors biased in the linear region," Solid-State Electronics, vol. 23, pp. 317-323, Apr. 1980.

[44] F. Hooge, T. Kleinpenning, and L. Vandamme, "Experimental studies on 1/f noise," Reports on Progress in Physics, vol. 44, pp. 479-532, 1981.

[45] T. Kleinpenning and L. Vandamme, "Model for 1/f noise in metal-oxide-semiconductor transistors," Journal of Applied Physics, vol. 52, p. 1594, 1981.

[46] K. Hung, P. Ko, C. Hu, and Y. Cheng, “A unified model for the flicker noise in metal-oxide-semiconductor field effect transistors,” IEEE Transaction on Electron Devices, vol. 37, pp. 654-665, Mar. 1990.

[47] G. Ghibaudo, O. Roux, C. Nguyen-Duc, F. Balestra, and J. Brini, “Improved analysis of low frequency noise in field-effect MOS transistors,” Physica Status Solidi (a), vol. 124, pp. 571-582, 1991.

[48] K. Hung, P. Ko, C. Ho, and Y. Cheng, “A physics-based MOSFET noise model for circuit simulators,” IEEE Transaction on Electron Devices, vol. 37, pp. 1323-1333, May 1990.

[49] E. Vandamme and L. Vandamme, “Critical discussion on unified 1/f noise models for MOSFETs,” IEEE Transaction on Electron Devices, vol. 47, pp. 2146-2152, Nov. 2000.

[50] M. Tsai, and T. Ma, “The impact of device scaling on the current fluctuations in MOSFETs,” IEEE Transaction on Electron Devices, vol. 41, pp. 2061-2068, Nov. 1994.

[51] K. Chew, K. Yeo and S. Chu, “Impact of technology scaling on the 1/f noise of thin and thick gate oxide deep submicron NMOS transistors,” IEE Proceedings of Circuits, Devices, and Systems, vol. 151, pp. 415-421, Oct. 2004.

[52] R. Jayaraman, W. Yang, and C. Sodini, “MOS electrical characteristics of low pressure re-oxidized nitrided oxide,” IEDM Technical Digest, pp. 668-671, 1986.

[53] R. Jayaraman and C. Sodini, “1/f noise interpretation of the effect of gate oxide nitridation and reoxidation in dielectric traps,” IEEE Transactions on Electron Devices, vol. 37, pp. 305-309, Jan. 1990.

[54] B. Gross, K. Krisch, and C. Sodini, “An optimized 850ºC low-pressure-furnace reoxidized nitrided oxide (ROXNOX) process,” IEEE Transactions on Electron Devices, vol. 38, pp. 2036-2041, Sep. 1991.

[55] E. Simoen and C. Claeys, “On the flicker noise in submicron silicon MOSFETs,” Solid-State Electronics, vol. 43, pp. 865-882, 1999.

[56] Z. Ma, Z. Liu, C. Yiu, P. Ko, and C. Hu, “New insight into high-field mobility enhancement of nitrided-oxide NMOSFETs based on noise measurement,” IEEE Transaction on Electron Devices, vol. 41, pp. 2205-2209, 1994.

[57] F. Gardner, “Phaselock techniques,” New York: Wiley, 1979.

[58] W. Robins, “Phase noise in signal sources,” London: Peregrinus, 1982.

[59] F. Kim and M. Chung, “Phase noise measurement of free-running microwave oscillators at 5.8 GHz using 1/3-subharmonic injection locking,” IEEE Microwave and Wireless Components Letters, vol. 15, no. 4, pp. 217-219, April 2005.

[60] ---, “CSA803 user’s guide,” Tektronix, Inc.: Beaverton, Oregon, U.S.A., 1993.

[61] ---, “Wavemaster operator’s manual,” LeCroy, Inc.: Chestnut Ridge, NY, U.S.A., 2002.

[62] K. Shanmugan and A. Breipohl, “Random signals: Detection, estimation, and data analysis,” New York: Wiley, 1988.

[63]---, “Getting started guide: PSA series spectrum analyzers,” Agilent Technologies, Inc.: Palo Alto, CA, U.S.A., 2005.

[64] D. Wolaver, “Phase-locked loop circuit design,” Englewood Cliffs, NJ: Prentice-Hall, 1991.

[65] L. DeVito, J. Newton, R. Croughwell, J. Bulzacchelli, and F. Benkley, "A 52 and 155 MHz clock-recovery PLL," ISSCC Digest of Technical Papers, pp. 142-143, Feb. 1991.

[66] A. Buchwald, K. Martin, A. Oki, and K. Kobayashi, "A 6-GHz integrated phase-locked loop using AlCaAs/GaAs heterojunction bipolar transistors," IEEE Journal of Solid-State Circuits, vol. 27, pp. 1752-1762, Dec. 1992.

[67] B. Lai and R. Walker, "A monolithic 622Mb/s clock extraction data retiming circuit," ISSCC Digest of Technical Papers, pp. 144-144, Feb. 1993.

[68] S. Lee, B. Kim, and K. Lee, "A novel high-speed ring oscillator for multiphase clock generation using negative skewed delay scheme," IEEE Journal of Solid-State Circuits, vol. 32, pp. 289-291, Feb. 1997.

[69] C. Park, O. Kim, and B. Kim, "A 1.8-GHz self-calibrated phase-locked loop with precise I/Q matching," IEEE Journal of Solid-State Circuits, vol. 36, pp. 777-783, May 2001.

[70] L. Sun and T. Kwasniewski, "A 1.25-GHz 0.35-μm monolithic CMOS PLL based on a multiphase ring oscillator," IEEE Journal of Solid-State Circuits, vol. 36, pp. 910-916, June 2001.

[71] H. Yang, L. Lee, and R. Co, "A low jitter 0.3–165 MHz CMOS PLL frequency synthesizer for 3 V/5 V operation," IEEE Journal of Solid-State Circuits, vol. 32, pp. 582-586, Apr. 1997.

[72] W. Wilson, U. Moon, K. Lakshmikumar, and L. Dai, "A CMOS self-calibrating frequency synthesizer," IEEE Journal of Solid-State Circuits, vol. 35, pp. 1437-1444, Oct. 2000.

[73] W. Yan and H. Luong, “A 2-V 900-MHz monolithic CMOS dual-loop frequency synthesizer for GSM receivers,” IEEE Journal of Solid-State Circuits, vol. 36, pp. 204-216, Feb. 2001.

[74] E. Baghdady, R. Lincoln, and B. Nelin, “Short-term frequency stability: Characterization, theory, and measurement,” Proceedings of IEEE, vol. 53, pp. 704–722, July 1965.

[75] L. Cutler and C. Searle, “Some aspects of the theory and measurement of frequency fluctuations in frequency standards,” Proceedings of IEEE, vol. 54, pp. 136–154, Feb. 1966.

[76] D. Leeson, “A simple model of feedback oscillator noises spectrum,” Proceedings of IEEE, vol. 54, pp. 329–330, Feb. 1966.

[77] J. Rutman, “Characterization of phase and frequency instabilities in precision frequency sources; Fifteen years of progress,” Proceedings of IEEE, vol. 66, pp. 1048–1174, Sept. 1978.

[78] Y. Toh and J. McNeill, “Single-ended to differential converter for multiple-stage single-ended ring oscillators,” IEEE Journal of Solid-State Circuits, vol. 38, pp. 141-145, Jan. 2003.

[79] D. Bowler, “Jitter in single ended CMOS ring oscillators,” MS Thesis, Worcester Polytechnic Institute, 2000.

[80] Y. Toh, “The effect of channel width on jitter in CMOS ring oscillators,” MS Thesis, Worcester Polytechnic Institute, 2002.

[81] A. Sedra and K. Smith, “Microelectronic circuits,” New York: Oxford University Press, 1998.

[82] T. Sakurai and A. Newton, “Alpha-power law MOSFET model and its applications to CMOS inverter delay and other formulas,” IEEE Journal of Solid-State Circuits, vol. 25, pp. 584-594, 1990.

[83] A. Kayssi, K. Sakallah, and T. Burks, “Analytical transient response of CMOS inverters,” IEEE Transaction on Circuits and Systems I: Fundamental Theory and Applications, vol. 39, pp. 42-45, 1992.

[84] S. Docking and M. Sachdev, “An analytical equation for the oscillation frequency of high-frequency ring oscillators,” IEEE Journal of Solid-State Circuits, vol. 39, pp. 533-537, Mar. 2004.

[85] --, “Design rules: MOSIS scalable CMOS (SCMOS),” MOSIS, Marina del Rey, California, Oct. 2004.

[86] K. Lim, X. Zhou, and Y. Wang, “Physics-based threshold voltage modeling with reverse short channel effect,” Journal of Modeling and Simulation of Microsystems, vol. 2, pp. 51-56, 1999.

[87] X. Zhou, K. Lim, and D. Lim, “A general approach to compact threshold voltage formulation based on 2-D numerical simulation and experimental correlation for deep-submicron ULSI technology development,” IEEE Transactions on Electron Devices, vol. 47, pp. 214-221, Jan. 2000.

[88] Z. Liu, C. Hu, J. Huang, T. Chan, M. Jeng, Ping Ko, and Y. Cheng, “Threshold voltage model for deep-submicrometer MOSFET’s,” IEEE Transaction on Electron Devices, vol. 40, pp. 86-95, Jan. 1993.

[89] A. Chaudhry and M. Kumar, “Controlling short-channel effects in deep-submicron SOI MOSFETs for improved reliability: a review,” IEEE Transactions on Device and Materials Reliability, vol. 4, pp. 99-109, Mar. 2004.

[90] K. Suzuki, T. Tanaka, Y. Tosaka, H. Horie, Y. Arimoto and T. Itoh, “Analytical surface potential expression for thin-film double-gate SOI MOSFET’s,” Solid-State Electron, pp. 327-332, 1994.

[91] M. Kumar and G. Reddy, “Diminished short channel effects in nanoscale double-gate silicon-on-insulator Metal-Oxide-Semiconductor Field-Effect-Transistors due to induced back-gate step potential,” Japanese Journal of Applied Physics, vol. 44, pp. 6508-6509, 2005.

[92] Y. Yeo, V. Subramanian, J. Kedzierski, P. Xuan, T. King, J. Bokor, and C. Hu, “Nanoscale ultra-thin-body silicon-on-insulator P-MOSFET with a SiGe/Si heterostructure channel,” IEEE Electron Device Letters, vol. 21, pp. 161-163, Apr. 2000.

[93] T. H. Lee, “The design of CMOS radio-frequency integrated circuits,” Cambridge University Press, 1998.

[94] ---, “CSA 7000, TDS7000, TDS6000 series user manual,” Tektronix, Inc.: Beaverton, Oregon, U.S.A., 2003.

[95] J. Koeppe and R. Harjani, “Enhanced analytic noise model for RF CMOS design,” Proceedings of the IEEE 2004 Custom Integrated Circuits Conference, pp. 383-386, Oct. 2004.

[96] D. Maher and R. Rance, “Random number generators founded on signal and information theory,” Cryptographic Hardware and Embedded Systems (CHES 1999), pp. 219-230, 1999.

[97] T. Tkacik, “A hardware random number generator,” Cryptographic Hardware and Embedded Systems (CHES 2002), pp. 450-453, 2002.

[98] M. Epstein, L. Hars, R. Krasinski, M. Rosner, and H. Zheng, “Design and implementation of a true random number generator based on digital circuit

artifacts," Cryptographic Hardware and Embedded Systems (CHES 2003), pp. 152-165, 2003.

[99] E. Trichina, M. Bucci, D. Seta, R. Luzzi, "Supplemental cryptographic hardware for smart cards," IEEE Micro, vol. 21, pp. 26-35, Nov. 2001.

[100] S. Park and K. Miller, "Random number generators: good ones are hard to find," Communications of the ACM, pp. 1192-1201, Oct. 1988.

[101] L. Blum, M. Blum, and M. Shub, "A simple unpredictable pseudo-random number generator," SIAM Journal on Computing, vol. 15, pp. 364–383, May 1986.

[102] N. Ferguson and B. Schneier, "Practical Cryptography," New York: Wiley, 2003.

[103] M. Matsumoto and T. Nishimura, "Mersenne twister: A 623-dimensionally equidistributed uniform pseudorandom number generator," ACM Transaction on Modeling and Computer Simulations, 1998.

[104] J. Kelsey, B. Schneier, D. Wagner and Chris Hall, "Cryptanalytic attacks on pseudorandom number generators," Fast Software Encryption, 1998.

[105] R. Davies, "Hardware random number generators," 15th Australian Statistics Conference, July 2000. Http://www.robertnz.net/hwrng.htm

[106] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray, S. Vo, "A statistical test suite for random and pseudorandom number generators for cryptographic applications," NIST Special Publication 800-22, 2001.

[107] G. Marsaglia, "DIEHARD: A battery of tests of randomness," http://stat.fsu.edu/~geo

[108] --, “Smart card policy and administrative guidelines handbook,” General Services Administration, Oct., 2000.

[109] A. Soohoo, “Lockdown! Random numbers secure network SoC designs,” Communication System Design, Apr. 2003.

[110] N. Stefanou and S. Sonkusale, “High speed array of oscillator-based truly binary random number generators,” Proceedings of 2004 IEEE International Symposium on Circuits and Systems (ISCAS 2004), vol. 1, pp. 505-508, May 2004.

[111] C. Portmann and T. Meng, “Power-efficient metastability error reduction in CMOS flash A/D converters,” IEEE Journal of Solid-State Circuits, vol. 31, pp. 1132-1140, 1996.

[112] G. Chien and P. Gray, “A 900-MHz local oscillator using a DLL-based frequency multiplier technique for PCS applications,” ISSCC Digest of Technical Papers, pp. 202-203, Feb. 2000.

[113] R. Watson, Jr. and R. Iknaian, “Clock buffer chip with multiple target automatic skew compensation,” IEEE Journal of Solid-State Circuits, vol.30, pp. 1267–1276, Nov. 1995.

[114] C. Kim, J. Lee, J. Lee, B. Kim, C. Park, S. Lee, S. Lee, C. Park, J. Roh, H. Nam, D. Kim, D. Lee, T. Jung, H. Yoon, and S. Cho, “A 64-Mbit, 640-Mbyte/s bidirectional data strobed, double-data-rate SDRAM with a 40-mW DLL for a 256-Mbyte memory system,” IEEE Journal of Solid-State Circuits, vol. 33, pp. 1703–1710, Nov. 1998.

[115] H. Chang and S. Liu, “A wide-range and fast-locking all-digital cycle-controlled delay-locked loop,” IEEE Journal of Solid-State Circuits, vol. 40, pp. 661-670, Mar. 2005.

[116] T. H. Lee and J. Bulzacchelli, “A 155-MHz clock recovery delay- and phase-locked loop,” IEEE Journal of Solid-State Circuits, vol. 27, pp. 1736-1746, Dec. 1992.

[117] J. Maneatis, “Low-jitter process-dependent DLL and PLL based on self-biased techniques,” IEEE Journal of Solid-State Circuit, vol. 31, pp. 1723-1732, Nov. 1996.

[118] Intel Platform Security Division, “The Intel random number generator,” Intel Technical Brief, 1999.