Title
Jitter, Phase Noise and Spurs in Frequency Multiplying Delay-locked loops: A Simple Model and Analysis

Author
Yang, Dihang

Publication Date
2015

Peer reviewed|Thesis/dissertation

University of California
Los Angeles

Jitter, Phase Noise and Spurs in Frequency Multiplying Delay-locked loops: A Simple Model and Analysis

A thesis submitted in partial satisfaction
of the requirements for the degree
Master of Science in Electrical Engineering

by

Dihang Yang

2015

Abstract of the Thesis

Jitter, Phase Noise and Spurs in Frequency Multiplying Delay-locked loops: A Simple Model and Analysis

by

Dihang Yang

Master of Science in Electrical Engineering

University of California, Los Angeles, 2015

Professor Asad. A. Abidi, Chair

We study the jitter performance of multiplying delay locked loops (MDLLs) and provide an effective approach to derive the phase noise of open loop MDLLs. We demonstrate that the ring oscillator phase noise models from Abidi and Hajimiri are essentially the same. Based on the analysis for MDLL jitter performance, new system models for Edge Combining DLLs and Recirculating DLLs are proposed which are accurate and simple for stability and noise analysis. Moreover, spurs caused by mismatch errors in Edge Combining DLLs are studied based on the new system model.

The thesis of Dihang Yang is approved.

Chih-Kong Ken Yang

Sudhakar Pamarti

Asad. A. Abidi, Committee Chair

University of California, Los Angeles

2015

To my parents

Table of Contents

List of Figures

Acknowledgments

I would like to express my deepest gratitude to my advisor, Prof. Asad. A. Abidi, for his excellent guidance, caring, patience, and providing me with an excellent atmosphere for doing research. I have been extremely lucky to have such a great advisor who teaches me not only the knowledge but also the correct ways to do research.

I would also like to thank Dr. Mohammad Heidari for his help during this thesis. It is his original analysis on MDLL jitter performance that gives me the insights on DLLs, assisting me with the development of the new models. I am truly thankful for his generous help and selfless dedication to my work.

Besides, I would like to thank the rest of my thesis committee: Prof. Sudhakar Pamarti, Prof. Chih-Kong Ken Yang, for their encouragement, insightful comments

Vita

2009-2013 B.S.(Electrical Engineering), Zhejiang University, Hang Zhou, China

2013-2015 M.S.(Electrical Engineering), UCLA, Los Angeles, CA

CHAPTER 1

Introduction

The large variety of new generation microprocessors and digital communication systems require state-of-the-art clock generators and synthesizers with low jitter, fast locking time and low power consumption. Delay-locked loops (DLLs) outperform phase-locked loops (PLLs) with their wide loop bandwidth, fast lock-in time, limited accumulative jitter, and first-order dynamics. For example, in low-power mobile SoCs, dynamic voltage and frequency scaling (DVFS) is used to reduce dynamic power consumption. Therefore a variable frequency low-jitter DLL-based clock generator is utilized [1]. Ultrawideband (UWB) Mode-1 system requires the desired carrier frequency to arise from the stringent specification of 9.5-ns band-hopping time. Thus only the fast-locking DLL based synthesizer can satisfy this strict settling time [2]. In high speed and high resolution analog-to-digital converters (ADCs), DLL based multiphase generators are used to generate low-jitter multiphase clocks [3].

Normally, a DLL-based clock generator/synthesizer, referred to as the edge combining DLL (ECDLL), consists of an edge combiner, a voltage-controlled delay line (VCDL) and a feedback loop that locks the delay of VCDL to the reference period. An edge combiner, like a frequency multiplier, synthesizes the high frequency output clock from the multiple delayed versions of the reference clock from the VCDL as in Fig. 1.1. In this architecture, however, any mismatch between delay cells in VCDL generates a periodic clock error ([4]) causing spurs in the output clock spectrum. To solve this problem, another type of DLL, referred as the recirculating DLL (RDLL) in Fig. 1.2, is implemented by using a ring oscillator with the phase realignment technique. The phase realignment circuit periodically breaks the ring to replace the noisy VCO clock edge with a clean reference clock edge [5].

In spite of widespread use of the two types of DLLs [6, 2, 7, 8, 9], neither is fully un-

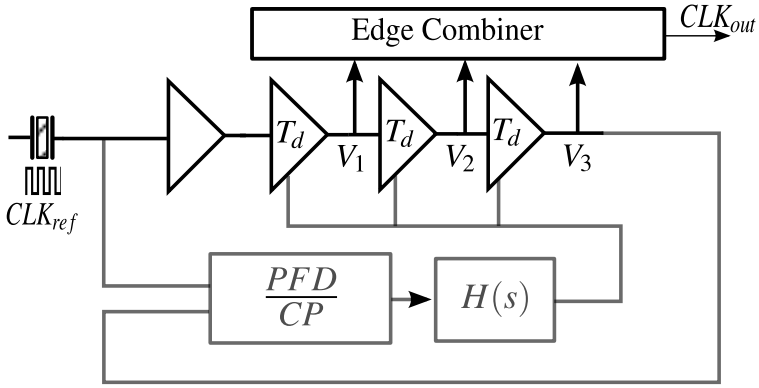


Figure 1.1: Edge Combining DLL

derstood. In [10], a discrete-time system model for ECDLL is proposed which is accurate for stability analysis. However, due to the omission of the edge combiner output, it can not predict the output phase noise. In [5], a continuous-time model for RDLL phase noise analysis is described. But because it ignores the sampling effects in a delay-locked loop, it cannot predict stability. In this paper, two simple but accurate models are shown that can incorporate both phase noise and stability analysis.

For synthesizers design, high frequency generators are one of the most crucial parts to be considered. While ring oscillators have been well studied in [11], [12], the high frequency generators in DLLs which are either the VCDL with edge combiner (Edge Combining Multiple-Delay-Line) or the VCO with realignment technique (Recirculating Multiple-Delay-Line), has not been fully understood. In this paper, a general phase noise model for both types of Multiple-Delay-Lines (MDLs) is presented to guide practical design.

In ECDLL, mismatch among delay cells results in spurious tones at the output. A time domain analysis based on state equations is shown in [4], [13] to predict RMS jitters, but it fails to capture the output spectrum. In [14], [15], Fourier Transform is used to accurately analyze spurs caused by mismatches; nevertheless, this method involves too complicated calculations which bring few insights for the understanding of DLL. Based on our new ECDLL system model, we will propose a more efficient way for spurs analysis on frequency domain.

This paper is organized as follows. In Chapter II, based on autocorrelation of jitter (a jitter-based noise analysis approach), a universal phase noise formula for MDLs is developed

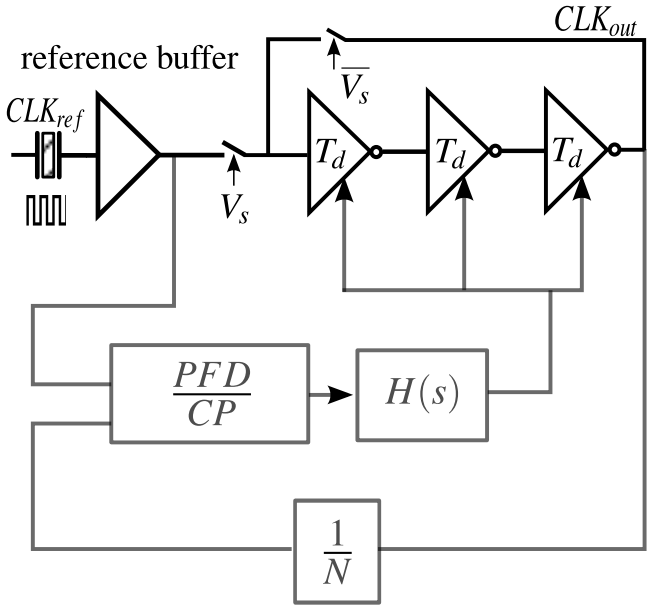


Figure 1.2: Recirculating DLL

which reveals the tradeoff between clock frequency, power consumption and phase noise. In Chapter III, we reconcile two ring oscillator models in [11] and [12] and show how they are fundamentally the same. We illustrate that how the jitter-based noise analysis is much more efficient than the existing approaches for phase noise analysis in ring oscillator. In Chapter IV and V, our new ECDLL and RDLL system model is described. Following it, spurs are readily analyzed with the ECDLL model in Chapter IV. In Chapter VII, our prediction is verified against both simulations and published measurements.

CHAPTER 2

Multiple Delay Line (MDL) Phase Noise Model

In a synthesizer, it is necessary to study the phase noise of the high frequency generator alone since it dominates the jitter of whole system. In this section, we investigate thermal noise effect in Multiple-Delay-Lines (MDLs). The model of jitter and phase noise in MDLs is derived based on the autocorrelation function, the generality of which is then verified through the comparison of the MDLs between two types of DLLs. The derivation of phase noise proposed in this section is named as jitter-based noise analysis.

2.1 Thermal Noise in a MDL

To avoid confusion, it is necessary to declare the definitions that are used in this thesis. For

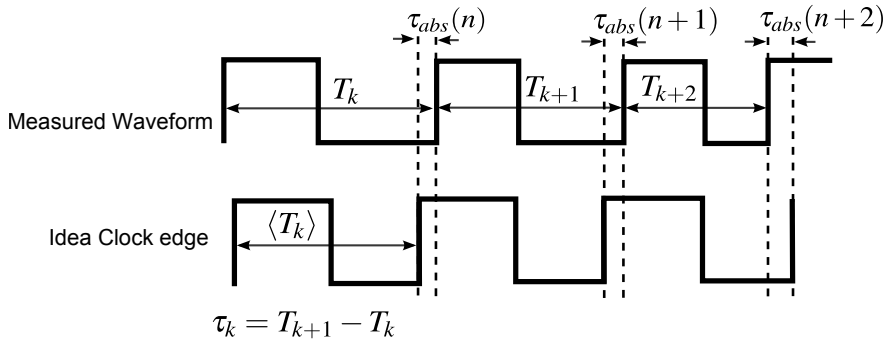


Figure 2.1: various definitions of jitter

noisy clock, the clock period (T_k) is equal to the average clock period ($\langle T_k \rangle$) plus a discrete random time uncertainty τ_k which has zero expectation value and varies at different clock cycles k as shown in Fig. 2.1. Long-term jitter is defined as the summation of all the existing

time uncertainty by the time of the observing point, i.e.,

$$\tau_{abs}(N) = \sum_{n=1}^{N} \tau_n \tag{2.1}$$

Period jitter is defined as the standard deviation of τ_k, which is equal to:

$$\sigma_\tau = \lim_{N\to\infty} \sqrt{\frac{1}{N}\sum_{n=1}^{N} \tau_n^2} \tag{2.2}$$

Therefore, long-term jitter is the accumulation of period jitter through the time till the observing point. A simple relationship can be seen between long-term jitter and phase error:

$$\tau_{abs}(k) = \frac{T_{out}}{2\pi}\phi(k) \tag{2.3}$$

where T_{out} is the output clock period of DLL.

Thus the PSD of the long-term jitter and phase noise are related as:

$$S_{\tau_{abs}}(f) = (\frac{T_{out}}{2\pi})^2 S_\phi(k) \tag{2.4}$$

Due to this simple relationship, the derivation of phase noise is based on the PSD of the long-term jitter in MDL.

The MDL under study is shown in Fig. 2.2(a). Through the delay line, a set of equally delayed clock edges are produced by the delay cells. An edge combiner adds this set of delayed edges together and filters out the DC value to produce a high frequency clock with a period which is equal to the delay of a single tap. Delay cells are assumed to be non-inverting. In practice, each delay cell is composed of several inverters. For correct operation, the total nominal delay of the line should be precisely one period of the reference signal and delayed edges are distributed uniformly.

As shown in Fig. 2.2. The noiseless reference clock CLK_{ref}, passing through the delay line, is delayed by T_d after each delay cell. The internal noise of each delay cell adds a period jitter to the propagated clock signal. Thus at each tap, the long-term jitter is the sum of the period jitter caused by all preceding cells. Therefore through the delay line, it increases progressively as shown in the Fig. 2.2(b). An edge combiner block aligns the delayed clock

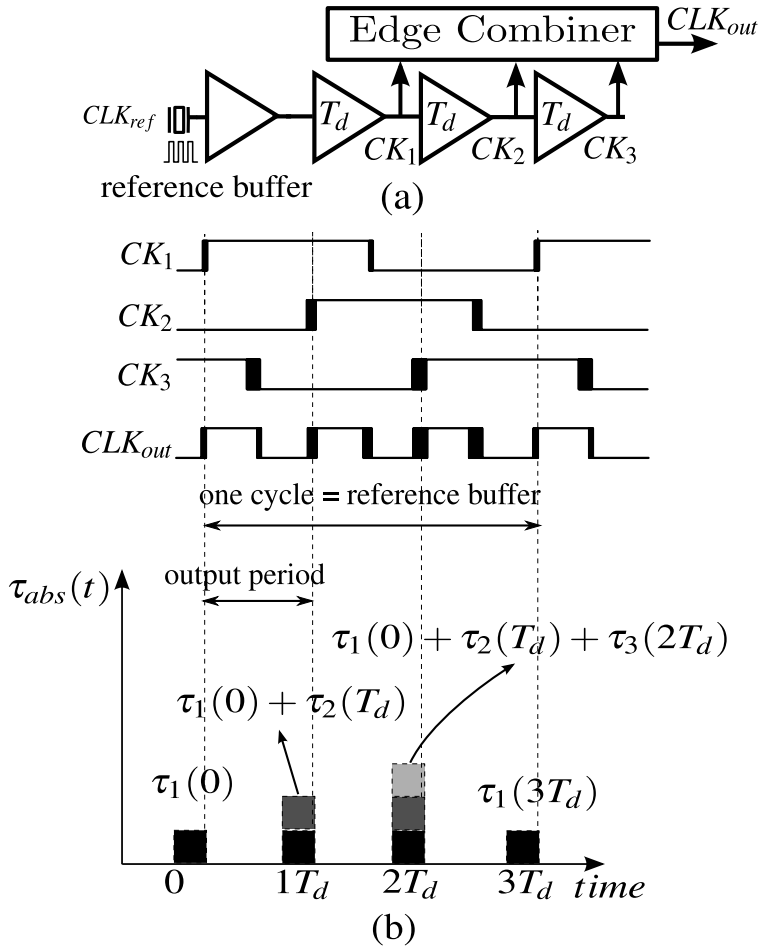


Figure 2.2: (a) an ECMDL with 3 stages (b) A sample waveform of delayed signals and jitter at the output.

signal from the first tap to the last one in order, resetting and repeating at the beginning of the next reference clock cycle. Thus, in every new reference clock cycle, the long-term jitter at the output of edge combiner accumulates from zero to a finite value. For instance, in Fig. 2.2(b), the long-term jitter of the output clock, denoted by $\tau_{abs}(t)$, at its third rising edge is the sum of the period jitter contributed from the first, second and third cells sampled at three different time instances, 0, T_d, and $2T_d$, respectively. At the beginning of the next reference clock cycle $3T_d$, the first delay cell clock edge forms the output clock edge again, therefore the output clock jitter falls back to the period jitter of first delay cell. $\tau_i(t)$ is a random process which represents the period jitter at the i^{th} cell.

2.2 Power Spectral Density (PSD) of Jitter in MDL

The PSD of jitter is defined by the Fourier Transform of the jitter's autocorrelation function. Each cell has independent white noise of the same power. It is assumed that the delay line is not enclosed in a DLL but still the total delay is equal to one period of the reference signal.

The autocorrelation function of a random discrete-time signal, $x(i)$, is the expectation of $x(k)x(k+n)$,

$$R_x(k,n) = E\{x(k)x(k+n)\} \tag{2.5}$$

which in general is the function of both absolute time k and relative time difference n respectively. A wide sense stationary (WSS) process, as a special case, has an autocorrelation function that only depends on the relative time difference n:

$$R_x(n) := R_x(k,n) \tag{2.6}$$

The long-term jitter at the output of MDL follows a periodic pattern which accumulates through one period of reference clock and is then reset to a small jitter at the end of a clock cycle. Therefore the autocorrelation function of long-term jitter in MDL changes periodically, implying that long-term jitter of MDL is a cyclostationary process. A time-independent and practically meaningful autocorrelation function in a cyclostationary process [16] can be derived by taking the *average* of the autocorrelation function at different observing points over one period.

$$R_x(n) = \frac{R_x(0,n) + R_x(1,n) + ... + R_x(N-1,n)}{N} \tag{2.7}$$

In the section of verification, it is shown that the PSD calculated by using the averaged autocorrelation function matches the measurement.

Based on the presented theory, the phase noise model for an MDL can be derived. Thermal noise induced jitter of each cell is uncorrelated to each other. They are white in spectrum with the same variance of σ_T^2. The autocorrelation function at the kth observing point, with

relative time difference n is calculated as below,

$$\begin{aligned} R_{\tau_{abs}}(k,n) &= E\left\{\tau_{abs}(k)\tau_{abs}(k+n)\right\} \\ &= E\left\{\sum_{j=1}^{k}\tau_j(j-1)T_d)\sum_{i=1}^{k+n}\tau_i((i-1)T_d)\right\}. \end{aligned} \tag{2.8}$$

if $n > 0$, eq(2.8) is equal to:

$$R_{\tau_{abs}}(k,n) = E\left\{\tau_1^2(0) + \tau_2^2(T_d) + \tau_3^2(2T_d) + ... + \tau_k^2((k-1)T_d)\right\} \tag{2.9}$$

if $n < 0$, eq(2.8) is equal to:

$$R_{\tau_{abs}}(k,n) = E\left\{\tau_1^2(0) + \tau_2^2(T_d) + \tau_3^2(2T_d) + ... + \tau_{k-|n|}^2((k-\mid n \mid -1)T_d)\right\} \tag{2.10}$$

In summary, the autocorrelation function at a the kth observing point, shown as Fig. 2.3, is equal to:

$$R_{\tau_{abs}}(k,n) = \begin{cases} k\sigma_T^2 & : 0 < n \\ (k-\mid n \mid)\sigma_T^2 & : -k \le n \le 0 \\ 0 & : otherwise \end{cases} \tag{2.11}$$

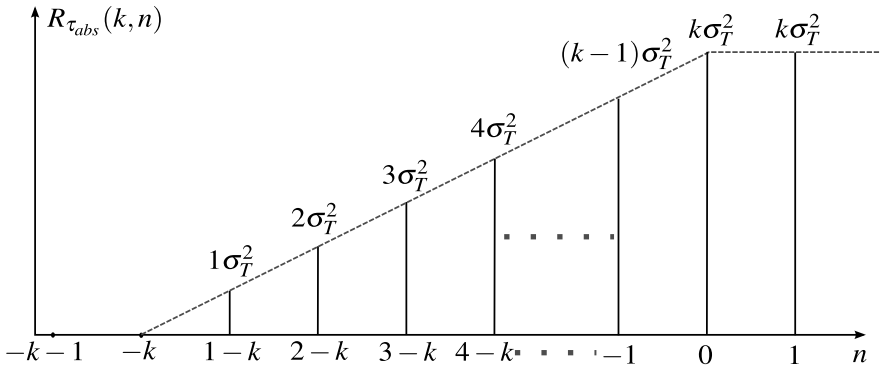


Figure 2.3: autocorrelation function at a certain observing point with different time difference n

As shown in Fig. 2.4, by combining eq(2.7) and eq(2.11), the averaged autocorrelation

function for a N-stage MDL is:

$$R_{\tau_{abs}}(n) = \begin{cases} (\frac{N+1}{2} + n - \frac{n(n+1)}{2N})\sigma_T^2 & : -N \leq n < 0 \\ (\frac{N+1}{2} - n - \frac{n(n-1)}{2N})\sigma_T^2 & : 0 \leq n \leq N \\ 0 & : otherwise \end{cases} \tag{2.12}$$

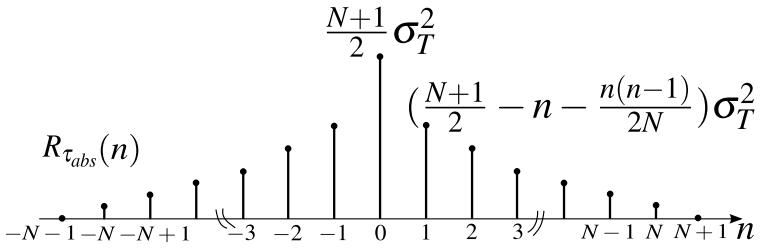


Figure 2.4: The autocorrelation function of the output jitter of a clock multiplier with ration of N.

By deriving the discrete-time Foureier transform of $R_{\tau_{abs}}(n)$ and translating it back to continuous-time spectrum, The N stage MDL output jitter PSD can be calculated as below,

$$\begin{aligned} S_{\tau_{abs}}(f) &= T_d \times DTFT\left[R_{\tau_{abs}}(n))\right] = T_d \sum_{n=-N}^{N} R_{\tau_{abs}}(n) e^{j2fT_d n} \\ &= (\frac{N+1}{2} + 2\sum_{n=1}^{N}\left[\frac{N+1-2n}{2} + \frac{n^2-n}{2N}\right] cos(2fT_d n))T_d\sigma_T^2. \end{aligned} \tag{2.13}$$

Even though the noise of each cell is white, the overall autocorrelation function of the output jitter is not an impulse function, indicating that the output jitter, or phase noise, is not white. For example, for $N = 9$ and a reference frequency of 100 MHz, Fig. 2.5 shows the PSD. As shown in this figure, the jitter is almost flat up to 2/10 of the reference frequency and rolls off beyond this point.

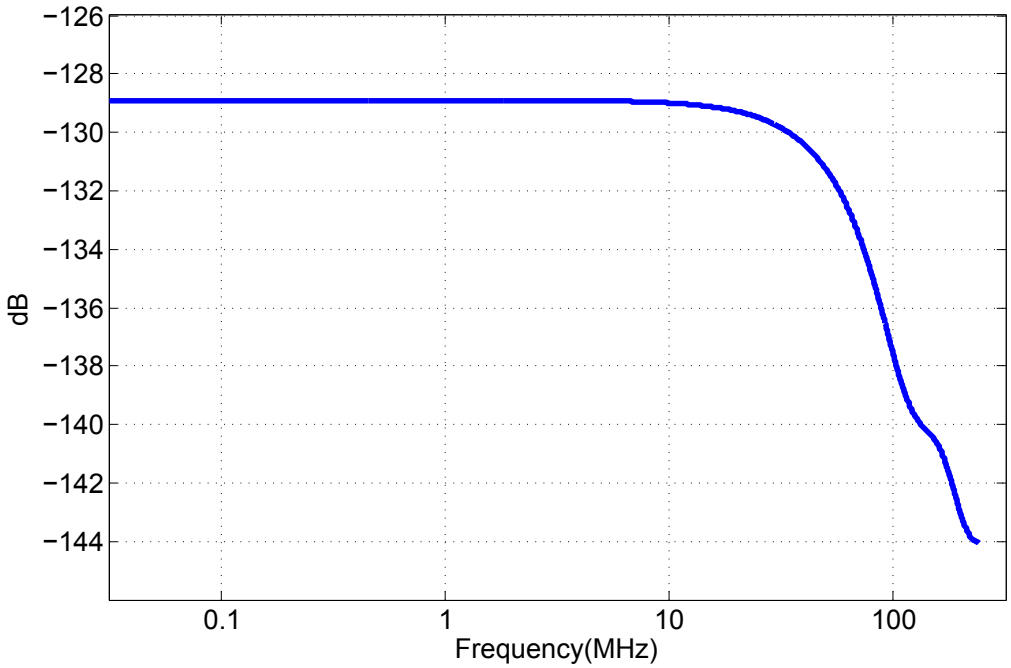


Figure 2.5: Power spectral density of the output jitter for a MDL, $N = 9$, $F_{REF} = 100MHz$.

2.3 The Comparison of Two Types of MDLs

Previous analysis for edge combiner multiple delay line (ECMDL) can be extended straightforwardly to recirculating multiple delay line (RMDL). The reasons are as following.

With a step signal at the input of an infinite-stage ECMDL, the edge combiner can generate an output clock with average period around the delay of a single tap T_d as shown in Fig. 2.6. While the input step signal propagates through the delay line, all the taps so far add their cumulative jitter to the output. The edge uncertainty of the propagating signal accumulates to infinity, resulting in infinite jitter at the output. Then an infinite ECMDL is equivalent to a free running VCO which generates a high frequency clock with uncontrolled accumulating jitter.

Now, imagine that noisy clock edge is replaced with a clean step after every K taps in an infinite ECMDL. The operation of the infinite stages ECMDL is equivalent to a finite K stage one driven by a noiseless periodic reference clock. The periodical injection of the clean step to the infinite ECMDL is also equivalent to the periodically replacement of the polluted VCO clock with a clean clock edge (RMDL). Thus, both ECMDL and RMDL can

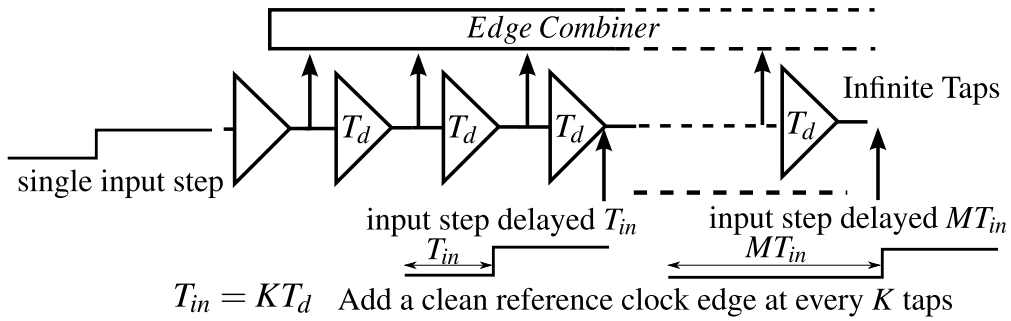


Figure 2.6: An equivalent circuit of MDL.

be described by the same model.

The RMDL phase noise is modeled as eq(2.14), eq(2.15) in [5].

$$\mathcal{L}_{MDL}(f) = |H_{up}(f)|^2 \times \sigma_T^2 \frac{f_0^3}{\Delta f^2} \tag{2.14}$$

$$H_{up}(f) = 1 - \frac{\beta}{1 + (\beta - 1)e^{-2\pi T_r}} e^{-j2\pi T_2/2} sinc(fT_r) \tag{2.15}$$

where $\sigma_n^2 f_0^3/\Delta f^2$ is the free running oscillator phase noise. T_r is the period of reference clock. $H_{up}(f)$ is the transfer function which reveals the effect of phase alignment technique in RDLL. The coefficient β is referred to as the realignment factor which illustrates how the realignment circuit can completely replace the noisy clock edge with the clean one. For DLL, at every clock edge replacement point, the realignment circuit fully replaces the noisy edge with a clean edge. Thus, β is equal to one. Therefore phase noise model is simplified to:

$$\begin{aligned} &\mathcal{L}_{MDL}(f) \\ &= [1 - 2sinc(2fT_r) + (sinc(2fT_r))^2 + \frac{(sin(\pi f T_r))^4}{(\pi f T_r)^2}]\sigma_T^2 \frac{f_0^3}{\Delta f^2} \end{aligned} \tag{2.16}$$

From the model provide in this section, the MDL phase noise is:

$$\begin{aligned} &\mathcal{L}_{MDL}(\Delta f) = S_{\Delta T_{out}}(f)(\frac{2\pi}{T_0})^2 \\ &= (\frac{N+1}{2} + 2\sum_{n=1}^{N}\left[\frac{N+1-2n}{2} + \frac{n^2-n}{2N}\right] cos(2fT_0 n))\frac{f_0^2\sigma_T^2}{\Delta f^2} \end{aligned} \tag{2.17}$$

Since the frequency bandwidth of interest is much smaller than reference frequency ($f <<$

$1/T_r$, $T_r = NT_0$), both phase noise model can be further simplified to:

$$\mathcal{L}_{MDL}(f) = \pi^2 N^2 \sigma_T^2 f_0 \tag{2.18}$$

$$\mathcal{L}_{MDL}(f) = \frac{2\pi^2(N+1)(2N+1)}{3}\sigma_T^2 f_0 \tag{2.19}$$

N is the number of delay line stage in ECDLL or the divider coefficient in RDLL. When N is large, the discrepancy between the twos model is negligible. The similarity of the two model shows that the noise performances for two types of MDLs are the same

In [5], the model is constructed through the analysis of the whole system operation. Whereas our method derives the model from the direct observation of clock jitter on time domain which can be extended to the analysis of ring oscillator noise and PFD noise.

CHAPTER 3

Revisiting Ring Oscillator Phase Noise Models

3.1 the Comparison of Ring Oscillator Thermal Phase Noise Models

For ring oscillator phase noise, the models proposed by [11] and [17] are widely used. In this section, we reconcile the two widely cited approaches and show how they are fundamentally the same. Through the use of the discrete-time ring oscillator model provided in this section, an alternate method is proposed for ring oscillator phase noise prediction. This new approach precisely describes the operation of ring oscillator and simplifies the analysis of flicker noise effect.

Due to the non-linearity and time-variant characteristic, jitter/phase noise can be observed only during clock edge transition. All the delay cells are assumed to be identical. Periodic jitter of ring oscillator, which is related to phase noise, is the net contribution of noise currents from all delay cells in a period. Thus, the noise current of each delay cell can be lumped together as an equivalent total noise current to model phase noise of ring oscillator. Since the noise of each delay cell is uncorrelated, the PSD of the equivalent noise current equals the summation of the one from each delay cell, i.e,

$$S_{i_{n_{total}}}(f) = \sum_{n=1}^{N}(S_{i_{n_{pmos}}}(f) + S_{i_{n_{nmos}}}(f)) \tag{3.1}$$

At every period, this equivalent noise causes an extra time uncertainty t_{extra},

$$t_{extra} = \frac{1}{I_N}\int_0^{T_d} i_{n_{total}}(t)dt \tag{3.2}$$

In [11], it shows that the period jitter, which is the standard deviation of the time uncertainty,

is equal to:

$$\sigma_{t_{extra}} = \sqrt{\frac{1}{I_N^2}\left\langle \left(\int_0^{\infty} i_{n_{total}}(t)W(t)dt \right)^2 \right\rangle} \tag{3.3}$$

where $W(t)$ is referred to as the window function:

$$W(t) = \begin{cases} 1 & : 0 \leq t \leq T_d \\ 0 & : otherwise \end{cases}$$

During every new clock cycle, the noises of the delay cells increase the time uncertainty of the output clock, i.e,

$$\sigma_{d_{total}} = \sqrt{\frac{1}{I_N^2}\left\langle \left(\int_0^{\infty} i_{n_{total}}(t) \sum_k W(t - kT_d)dt \right)^2 \right\rangle} \tag{3.4}$$

Eq(3.4) is equivalent to the model shown in Fig. 3.1.

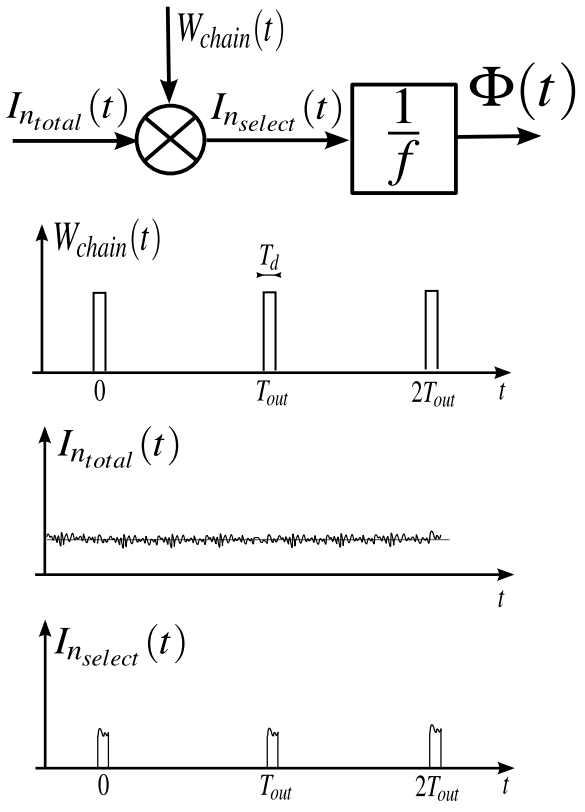


Figure 3.1: Abidi ring oscillator phase noise model

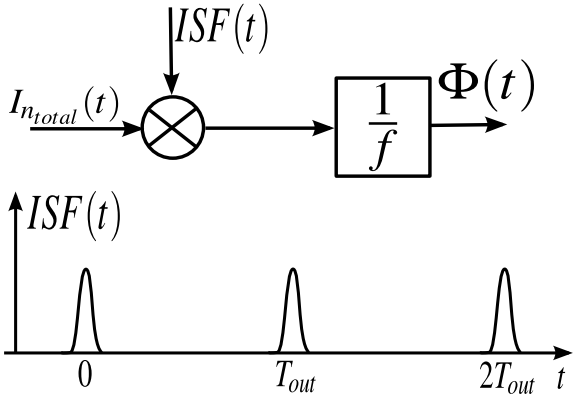


Figure 3.2: Hajimiri ring oscillator phase noise model

Meanwhile, the ISF approach provided by [17] builds the phase noise model as in Fig. 3.2 (Here we lumped PMOS and NMOS noise currents together). The only difference between the two models is the shape of window chain $W_{chain(t)}$ and Impulse Sensitivity Function $ISF(t)$.

This difference is caused by the reasonable approximation in [11] that during transition time, PSD of thermal noise current is constant. As shown in Fig. 3.3, during transition time, V_{in} changes from $V_{dd}/2$ to V_{dd}, altering g_m of the transistor. Thus the PSD of thermal noise current which follows $S_{i_n} = 4KT\gamma g_m$ also changes. Hajimiri's ISF function, taking into account the changing noise current PSD, is therefore not a regular rectangular shape.

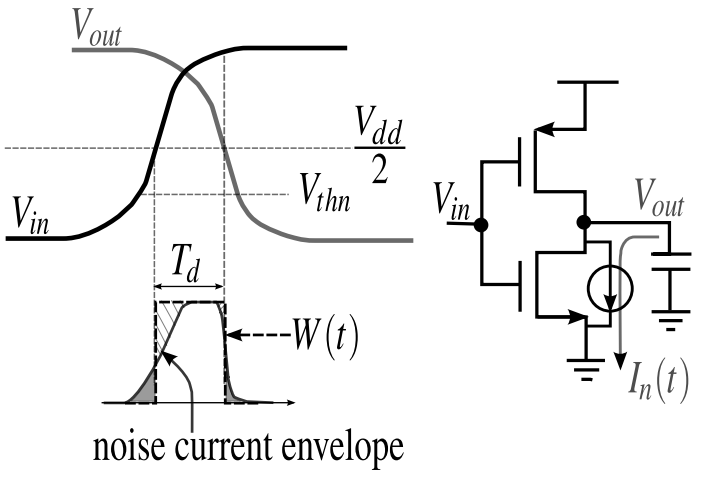


Figure 3.3: Approximation of window function

As shown in Fig. 3.3, the practical noise envelope shape is close to a rising cosine, therefore the area of the reconstructed rectangular noise envelope proposed in [11] is almost the same as the practical one. During the clock transition, the observed jitter is the integration of the total noise current in the contemporary and previous period. Thus, jitter/phase noise is only determined by the total area of noise current other than the shape of it. Therefore Abidi's model accurately captures the net jitter without characterizing the detailed transition of noise current as opposed to Hajimiri's model. Although two models predict the same results, the first one demonstrates the result through straightforward analysis and simple calculation.

While in a group of previous literature, ring oscillator phase noise is modeled based on continuous-time model [11, 17, 12, 18], a discrete-time model can also describe it in a simpler

way. In every clock period, since the extra time uncertainty arising from thermal noise is uncorrelated, eq(3.4) shows that during every new clock cycle, ring oscillator accumulates a new period jitter. Therefore, ring oscillator noise model can be represented as a discrete integrator which accumulates the jitter sampled at each period as shown in Fig. 3.4. The PSD of the discrete jitter, which is caused by the thermal noise, is white and equals to σ^2 [11]. The coefficient $2\pi/T_{out}$ is used to transform the dimension of the model into phase. Therefore phase noise is calculated as:

$$\mathcal{L}(f) = \sigma^2 \times \frac{4\pi^2}{T_{out}^2} \times |\frac{1}{1 - e^{-j2\pi f T_{out}}}|^2 \tag{3.5}$$

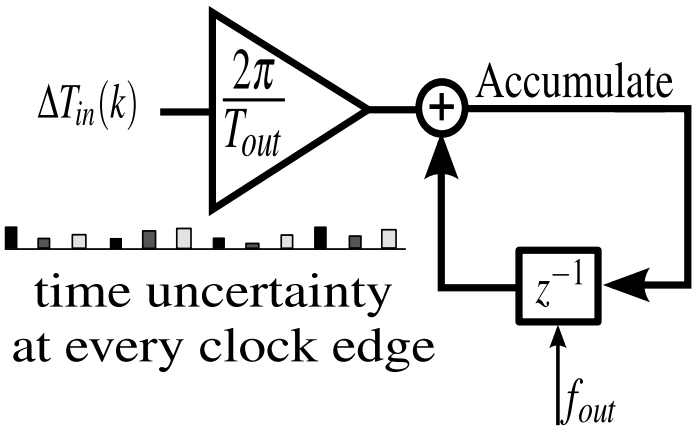


Figure 3.4: discrete-time phase noise model

The above expression is based on sampled jitter. To translate it back to continuous model, another coefficient T_{out} is needed. In the bandwidth of interest, eq(3.5) can be simplified as eq(3.6), the same to the result in [11].

$$\mathcal{L}(f) = \sigma^2 \times \frac{4\pi^2}{T_{out}^2} \times |\frac{1}{fT_{out}}|^2 \times T_{out} \tag{3.6}$$

3.2 A New Perspective for Low Frequency Phase Noise Model

Flicker noise only contributes to jitter during transitions. The same as thermal noise, Each delay cell's flicker noise current can be lumped together as an equivalent noise current $i_f(t)$ because they are uncorrelated:

$$S_{i_f}(f) = \sum_{n=1}^{n=N} (S_{i_{fnmos}}(f) + S_{i_{fpmos}}(f)) \tag{3.7}$$

In every period, the equivalent noise current introduces a time uncertainty:

$$t_f = \frac{1}{I_N} \int_0^{T_d} i_f(t) dt \tag{3.8}$$

Since flicker noise varys slowly which can be regarded as a constant value during edge transitions. eq(3.8) can be simplified to:

$$t_f[k] = \frac{1}{I_N} i_n(kT_{out}) \times T_d \tag{3.9}$$

where $T_{out} = 2NT_d$

Eq(3.9) shows that flicker noise effect is also a discrete-time effect. In spectrum, the PSD of this jitter is

$$S\sigma_f(f) = \frac{T_d^2}{I_N^2} \times \frac{1}{T_{out}} \sum_n S_{i_f}(f - n \times f_{out}) \tag{3.10}$$

The discrete model as shown in Fig. 3.4 is efficient for the calculation of phase noise caused by flicker noise. With the same approach shown above, the phase noise due to flicker noise current can be described as:

$$S_\phi(f) = \frac{4\pi^2 T_d^2}{T_{out}^2 I_N^2} \sum_n S_{i_f}(f - nf_{out}) |\frac{1}{1 - e^{-j2\pi f T_{out}}}|^2 \tag{3.11}$$

For the bandwidth of interest, eq(3.11) can be further simplified to eq(3.12) which is completely the same as the result in [18]. But after introducing the discrete ring oscillator model, the calculation is much simpler.

$$S_\phi(f) = \frac{1}{4N^2 I_N^2} \times S_{i_f}(f) \times (\frac{f_{out}}{f})^2 \tag{3.12}$$

CHAPTER 4

ECDLL model

4.1 ECDLL System Model

An ECDLL model that demonstrated by Edward Lee [10] is shown in Fig. 4.1 where K_{DL} represents the total delay line gain in radians per volt and $H(z)$ is the transfer function from the input of phase frequency detector to loop filter output. It is a good model for ECDLL stability analysis. It can not predict the output phase noise because the edge combiner, which synthesizes high frequency clock, is ignored. Therefore we will refine this model for noise analysis.

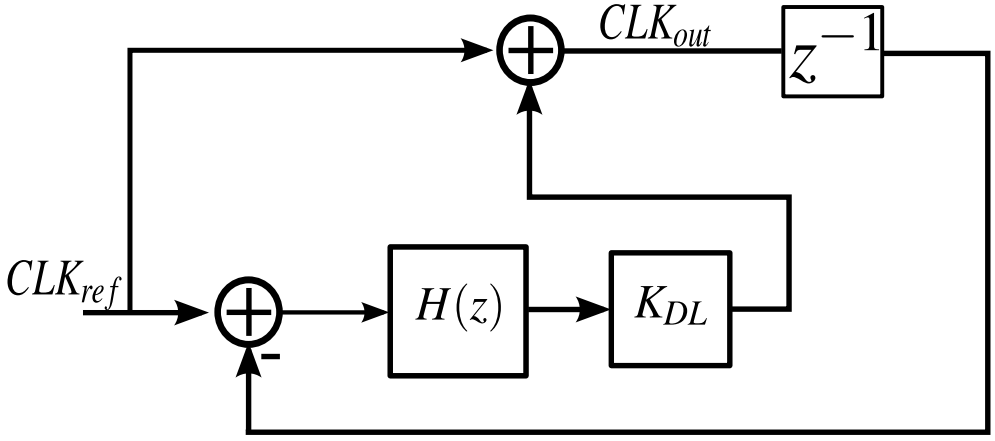


Figure 4.1: Edward Lee's ECDLL model

The model shown in Fig. 4.2 is the modified N stage ECDLL model. The basic idea is to split the lumped delay line model to distinguish the output of each delay cell. In this way, edge combiner output, which is the sum of all delay cells outputs, can be modeled. The adders that align in a row represent the delay cells. They demonstrate that, the total

delay line gain is the summation of each delay cell. The arrows between the adders are the outputs of delay cells. Due to the delay effect in each delay cell, the edge combiner output is the summation of the delayed version of those arrows.

For stability limit prediction, it does not depend on edge combiner output, the refined model is completely the same as the one provided in [10].

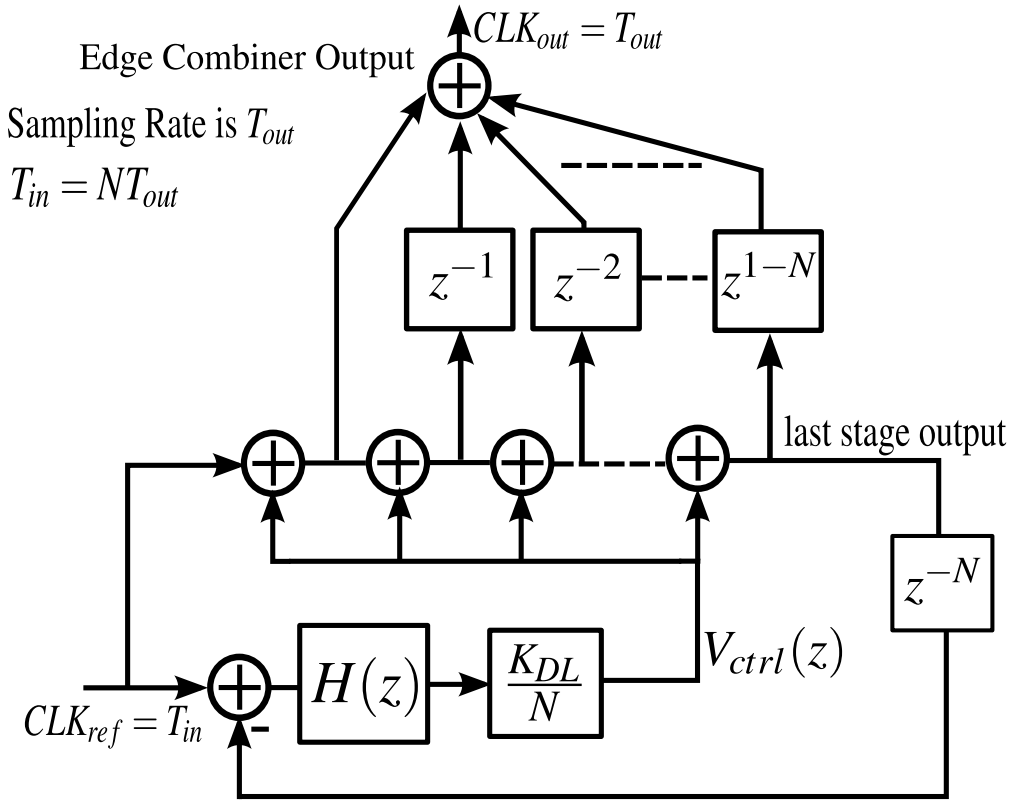


Figure 4.2: New ECDLL model

4.2 ECDLL Noise Analysis

Here, based on the new ECDLL model, the edge combiner output phase noise can be prediceted from a systematical way.

To calculate the output phase noise, the noise in the delay line should be analyzed first. Unlike ring oscillator where the noise of each delay cell can be lumped together, the noise in ECDLL should be considered separately because of the delay effect in each tap.

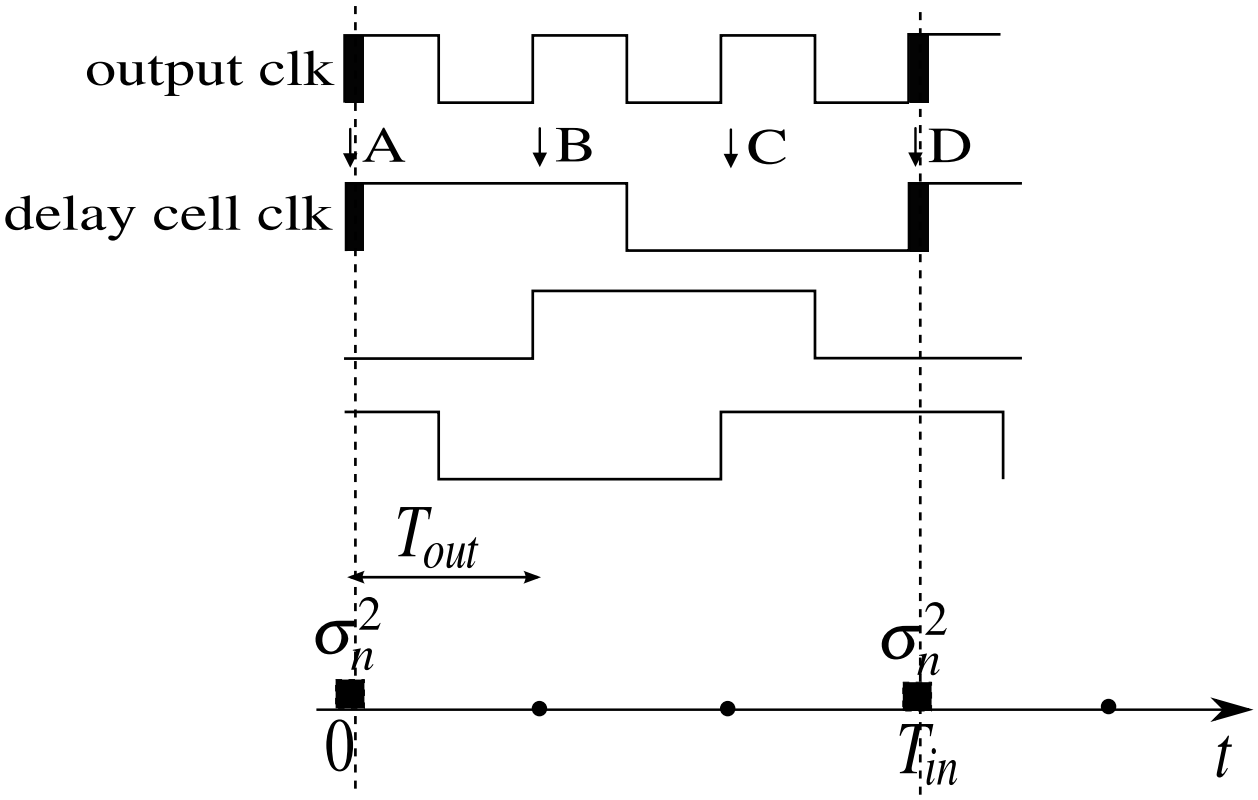


Figure 4.3: jitter distribution in delay cells and edge combiner output

In ECDLL, Since reference clock propagates through the delay line, the output of each tap has the same clock period as reference clock. Delay cell causes a jitter at every other input clock as shown in Fig. 4.3 where only the first delay cell is assumed to be noisy. The Edge Combiner scales up the input clock frequency with the factor N. Therefore Edge Combiner works as a high frequency sampling system which oversamples the clock signal in each delay cell, causing zero paddings in the discrete jitter train. When the jitter of each delay cell is observed at edge combiner output, it is a cyclostationary process. Thus the autocorrelation of each delay cell's jitter is:

$$R_x(m) = \frac{R_x(0,m) + R_x(1,m) + ... + R_x(N-1,m)}{N} \tag{4.1}$$

$$R_{\Delta T_{out}}(n) = \begin{cases} \sigma_n^2/N & : nT_{out} = KT_{in} \\ 0 & : otherwise \end{cases} \tag{4.2}$$

From eq(4.2), we can see that the jitter caused by thermal noise is white with PSD equals $S_d(f) = \sigma_n^2/N$. This method for white noise calculation is much more efficient than the one in [19]. It avoids the use of convolution which involves aliasing effect for thermal noise calculation.

For flicker noise analysis, they are low frequency noise, therefore the aliasing effect can be ignored when they are sampled. Then the method in 3.2 can be directly used.

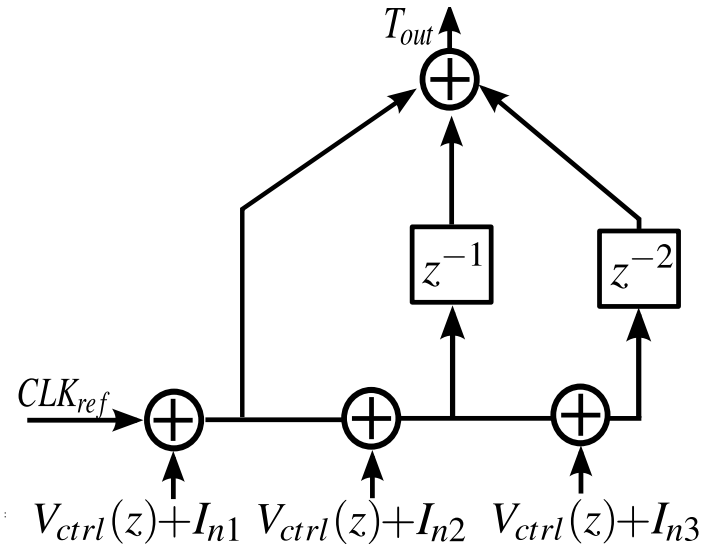


Figure 4.4: open loop ECMDL signal flow and noise flow

To show the impact of the jitter caused by each delay cell at output, the transfer function from each delay cell to the output should be derived. As shown in Fig. 4.4, for different stages, the transfer functions to the output varys as the following formulas:

$$H1(f) = \frac{T_{out}}{I_{n1}} = (1 + z^{-1} + z^{-2}) \tag{4.3}$$

$$H2(f) = \frac{T_{out}}{I_{n2}} = (z^{-1} + z^{-2}) \tag{4.4}$$

$$H3(f) = \frac{T_{out}}{I_{n3}} = z^{-2} \tag{4.5}$$

Since each noise source is uncorrelated to others, by using the transfer function eq(4.3)-

eq(4.5), Edge Combiner output phase noise can be modeled as:

$$S_{T_{out}}(f) = \sum_{n=0}^{n=2} S_{dn}(f) |\frac{(1-z^{-(3-n)})}{(1-z^{-1})}|^2 \tag{4.6}$$

For N stages ECDLL, the phase noise is:

$$S_{T_{out}}(f) = \sum_{n=0}^{n=N-1} S_{dn}(f) |\frac{(1-z^{-(N-n)})}{(1-z^{-1})}|^2 \tag{4.7}$$

Eq(4.7) is another MDL open loop phase noise prediction derived directly from the new ECDLL system model. In Validation section, it will show that this model has completely the same phase noise prediction as the universal mathematical model provided in section II. Because this model can be embedded into the whole control loop, close loop phase noise model can also be derived.

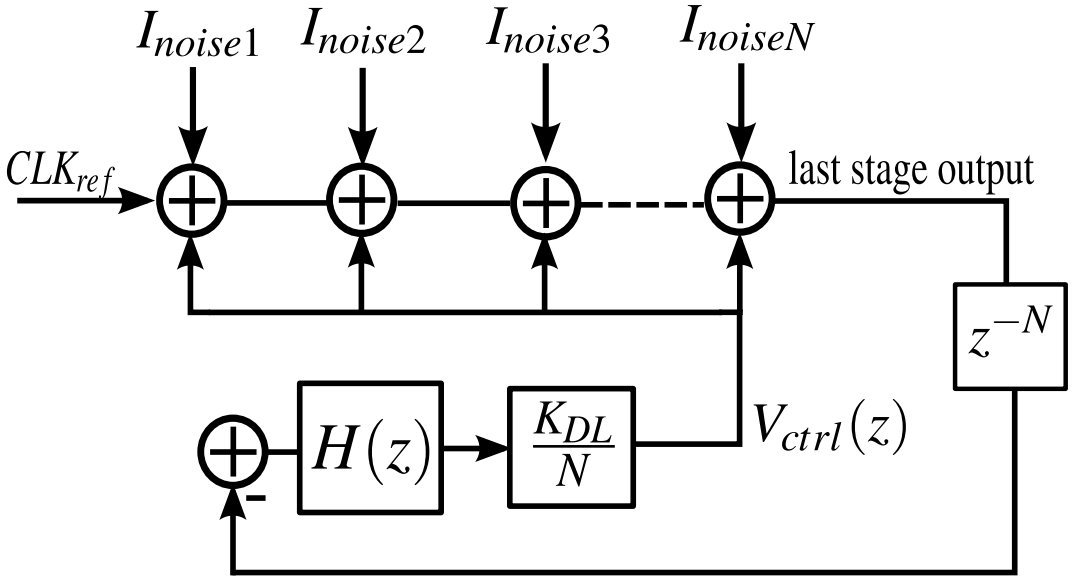


Figure 4.5: the loop model that considers about delay cell noise and input feedthrough effect

As shown in Fig. 4.5, the transfer function from the noise sources (I_{noisei}), which are directly added to the delay lines, to the control voltage $V_{ctrl}(z)$ is:

$$H_{V_{ctrl1}}(z) = -\frac{z^{-N}H(z)K_{DL}/N}{z^{-N}H(z)K_{DL}+1} \tag{4.8}$$

In addition to the directly feedforward noise sources, some input referred noises such as the charge pump noises, phase frequency detector noises pass through the loop filter to reach

the edge combiner output. The transfer function from those sources to the control voltage $V_{ctrl}(z)$ is:

$$H_{V_{ctrl2}}(z) = \frac{H(z)K_{DL}/N}{z^{-N}H(z)K_{DL}+1} \tag{4.9}$$

Because the noises in different delay cells are uncorrelated, the phase noise model, considering only the delay cells noises is:

$$\mathcal{L}(f) = S_{dn}(f)|(1+H_{V_{ctrl1}}(f))+(1+2H_{V_{ctrl1}}(s))e^{-j2\pi fT_{out}}+...+(1+NH_{V_{ctrl1}}(f))e^{-j(N-1)2\pi fT_{out}}|^2 \tag{4.10}$$

Through algebra, eq(4.10) can be simplified to:

$$\mathcal{L}(f) = S_{dn}(f)|\sum_{n=0}^{n=N-1}\frac{1-e^{-j(N-n)2\pi fT_{out}}}{1-e^{-j2\pi fT_{out}}} + H_{V_{ctrl1}}(f)(\frac{1-e^{-jN2\pi fT_{out}}}{(1-e^{-j2\pi fT_{out}})^2}+\frac{Ne^{-j(N)2\pi fT_{out}}}{1-e^{-j2\pi fT_{out}}})|^2 \tag{4.11}$$

With the same procedure, the phase noise caused by input referred noise sources is:

$$\mathcal{L}(f) = S_{dn}(f)|H_{V_{ctrl2}}(f)(\frac{1-e^{-jN2\pi fT_{out}}}{(1-e^{-j2\pi fT_{out}})^2}+\frac{Ne^{-j(N)2\pi fT_{out}}}{1-e^{-j2\pi fT_{out}}})|^2 \tag{4.12}$$

By combining eq(4.10) and eq(4.12), the close loop phase noise model can be derived.

CHAPTER 5

RDLL Model

The main difference between RDLL and ECDLL is their multiple delay line blocks. To distinguish the difference, RMDL will be studied first. Then it will be put into the whole RDLL system for system model consideration.

5.1 RMDL Model

As mentioned before, the realignment technique selects the clean input clock edge to replace the noisy output clock edge at every reference clock cycle Fig. 5.1. The same as ECMDL,

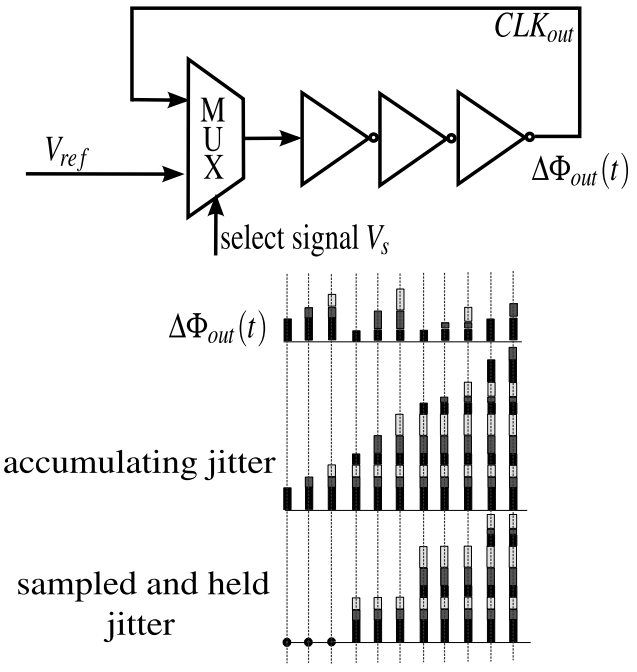


Figure 5.1: decompositon of RMDL output jitter

the jitter at the output of RMDL also follows a periodic accumulating and resetting pattern. The jitter can be decomposed to a continuous accumulating jitter subtracting the stored

jitter which is the sampled and held version of the accumulating jitter, as shown in Fig. 5.1 . Based on the decomposition, the RMDL can be modeled as Fig. 5.2.

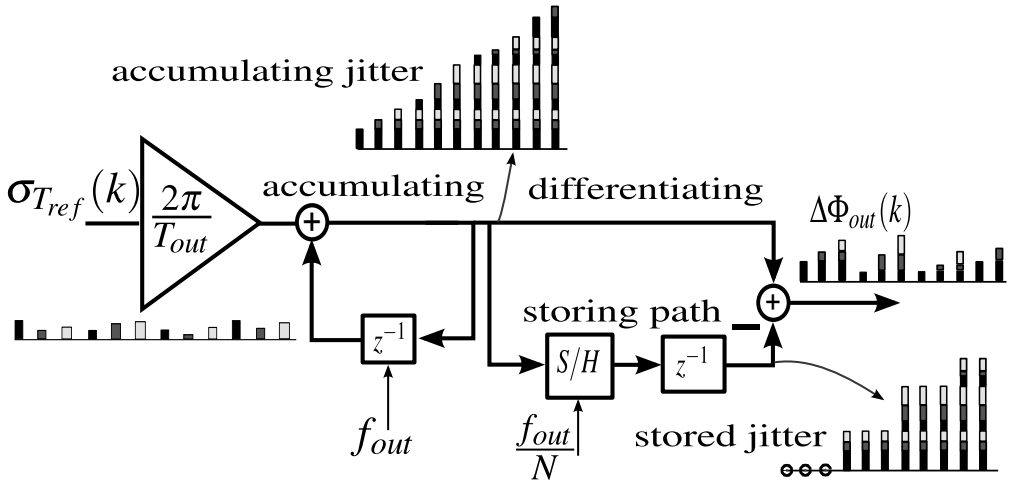


Figure 5.2: RMDL model

At the same time, reference clock directly injects its own noise into the delay lines at reference clock edge. As Fig. 5.3(a) shows, reference noise will be sampled at every reference clock edge. For RMDL output, which operates at a much higher frequency, the reference phase noise effect is observed at every output clock edge, as shown in Fig. 5.3(a). Therefore for reference phase noise, the RMDL works as a hold function, as shown in Fig. 5.3(b).

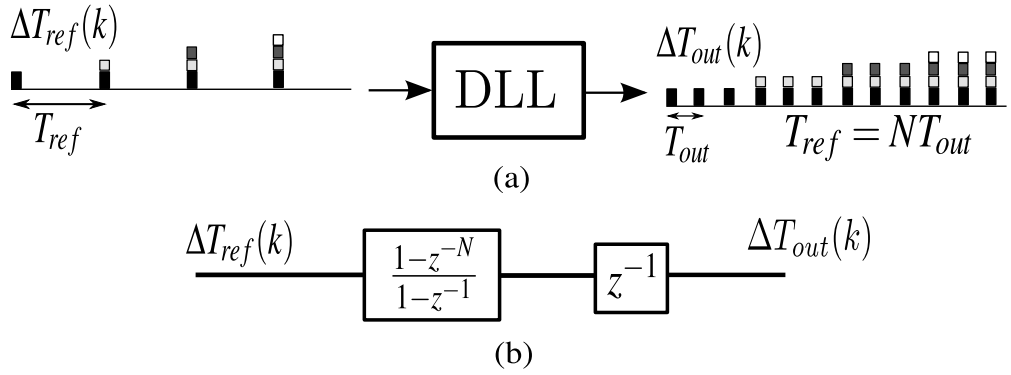


Figure 5.3: (a) Reference noise injection (b) z-domain model of reference noise injection.

5.2 RDLL System Model

The previous study on PLLs show that PFD, divider and VCO all work as samplers with sampling rate equal to the clock period of their own. Thus, the whole RDLL can be modeled as Fig. 5.4. It includes two different sampling rates, introduced by PFD, divider and VCO respectively. It contains continuous-time blocks (low-pass loop filter) and discrete-time blocks (other components). This is a multi-rate feedback system. It is not a LTI system anymore and has been well studied in [20]. The goal of this paper is to provide simple LTI models which are accurate and easy to analyze the DLL stability and phase noise.

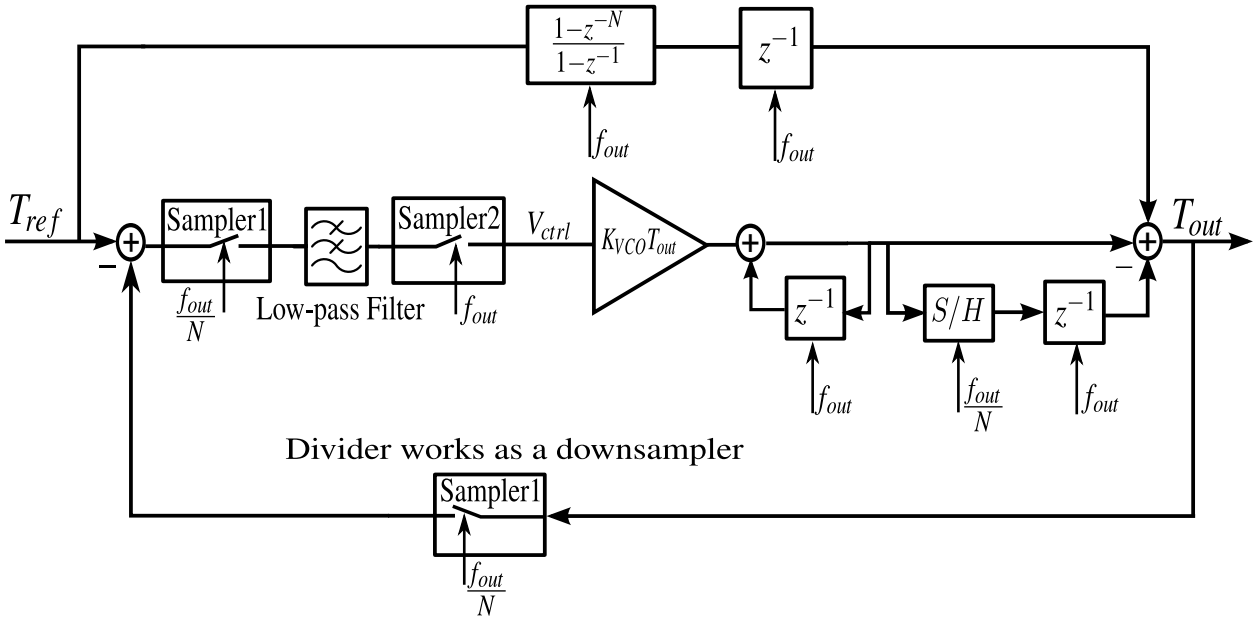


Figure 5.4: The complete system model of DLL.

5.3 RDLL System Model Simplification

Understanding the sampling operation of frequency divider is the key for the simplification of the model. As shown in Fig. 5.5, the divider can only samples the signal in the circles at its own sampling rate.

The output of RMDL is the subtraction of the accumulated signal and the stored signal.

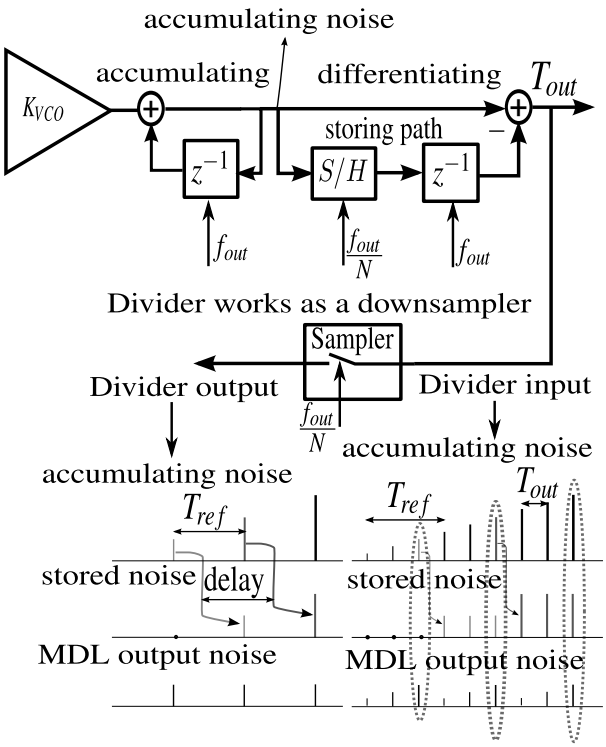


Figure 5.5: divider effect

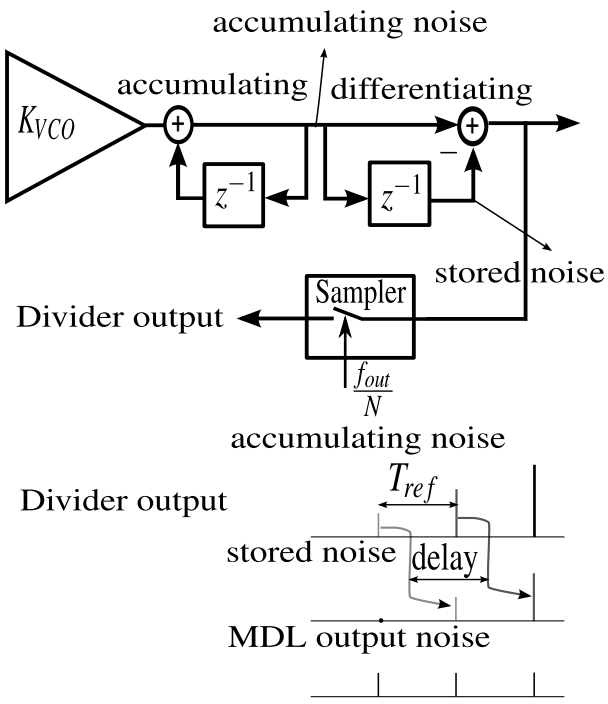


Figure 5.6: MDL model from the perspective of divider output

The divider downsamples the two signals, eliminating the changes of signal between two sampling instants. Thus, at the output of the divider, the stored signal in RMDL is just the delayed version of the accumulating signal. So at the divider output, RMDL model is equivalent to Fig. 5.6 .

Due to the downsampling effect of the divider, there are two paths of signals in the DLL systems as shown in Fig. 5.7. One group of signals, which are sampled by the divider, circulates in the DLL loop. Other signals directly flow to the output.

Since the real DLL system is not LTI system anymore. For different consideration, the model simplification is different. For stability, only the loop including feedback need to be investigated. The signals which flow in the loop can cause loop fluctuation, inducing stability problem. They are downsampled by the divider, only related to the sampling rate of divider. Thus, in the model of DLL, neglecting the sampler from VCO does not make any difference for the analysis of system stability. Meanwhile, in every reference clock cycle, the phase

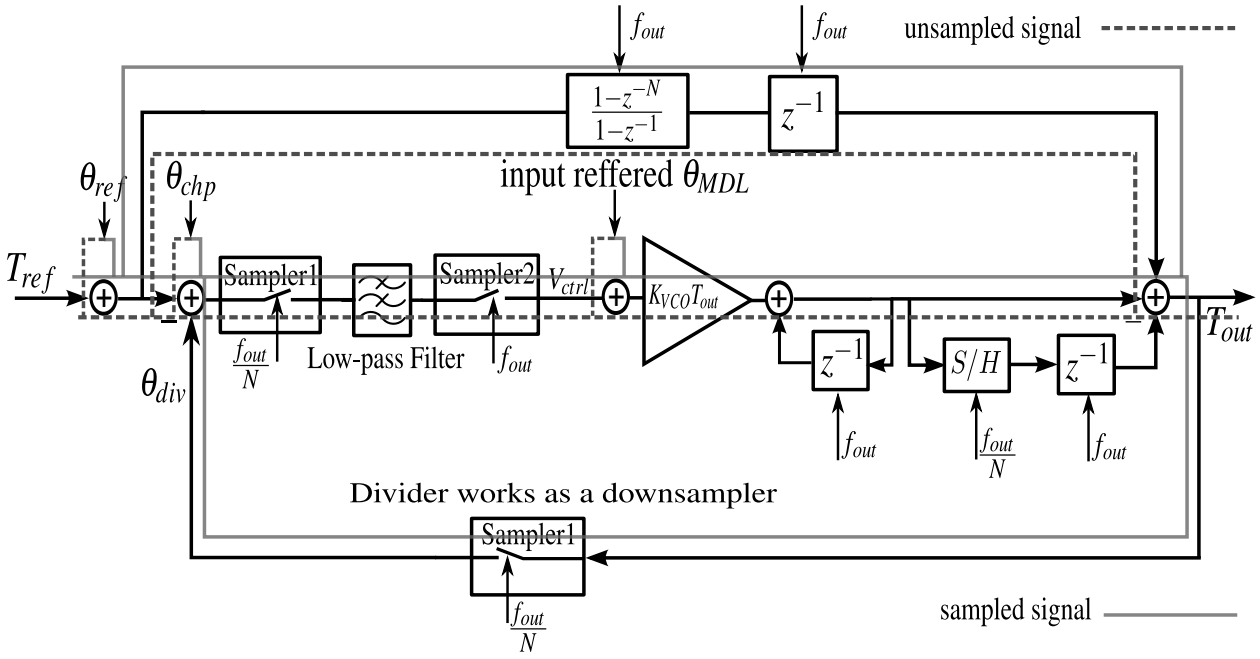


Figure 5.7: signal flows in RDLL

detector senses a phase error $\Delta\Phi$, resulting in the voltage change at the output of low-pass filter. Based on the study of charge-pump PLLs ([21],[22]), the phase error at every cycle, which is assumed to be a constant value here, causes an pulse voltage on the resistor with average value equals $\frac{\Delta\Phi}{2\pi}IR$. It also changes the voltage on the capacitor. Since the phase error near steady state is small, the transient charging/discharging of the capacitor can be simplified as a step change. Thus, in every clock cycle, the phase error roughly induces a constant voltage change on capacitor which is equal to $\frac{\Delta\Phi}{2\pi}T_{ref}\frac{I}{C}$. Totally, if ignoring the ripple caused by the resistor, the output of the low-pass filter approximately suffers from a step change with value of $\frac{\Delta\Phi}{2\pi}T_{ref}\frac{I}{C}+\frac{\Delta\Phi}{2\pi}IR$ at the edge of the reference clock and then keeps as constant as shown in Fig. 5.8. For stability consideration, it is the change of the signal in the loop that causes stability issue. Since in every clock cycle, the analog filter output is approximately unchanged, discretizing it in the model does not cause any discrepancy for stability analysis. As Fig. 5.9 shows, the approximate control signal (V_{ctrl}) constantly changes the output frequency, linearly varying the output phase with time. Since the divider can only sample the phase information at every other T_{ref} and the control signal is unchanged in

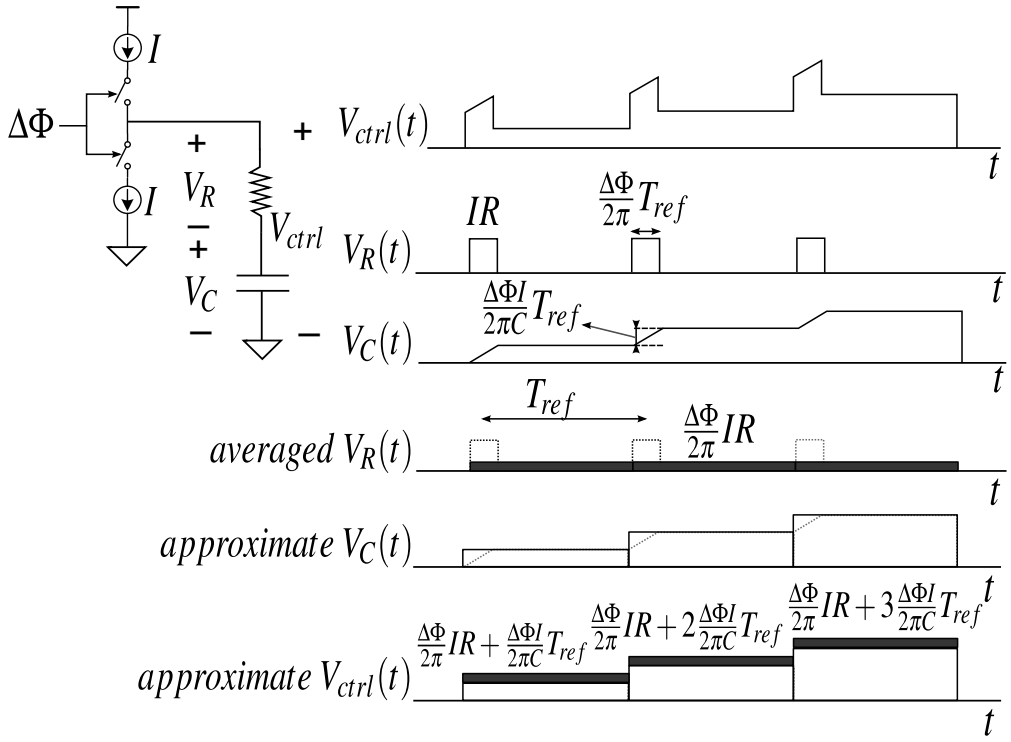


Figure 5.8: discretizing low-pass filter

a period. Representing the control voltage as an impulse and modeling the frequency gain of VCO (K_{VCO}) to phase gain in a period ($K_{VCO}T_{ref}$) can provide the same output phase shift at the sampling instant. Thus, the phase error sensed by the PFD will not be affected by the modification of the model, causing no inaccuracy to the output of the low-pass filter.

Then, the whole system is simplified as Fig. 5.10. The RDLL model has only one sampling rate f_{ref} now. The proportional block in the low-pass filter reveals that resistor can not store the voltage of each clock cycle, while the integrating path shows that capacitor is able to do that. The integrating and differentiating blocks in the MDL model, as shown in Fig. 5.6, cancel the effect of each other, simplifying the RMDL to a proportional block.

This stability model is close to ECDLL stability model. Unlike VCO, the RMDL does not work as an integrator. Therefore the whole system only has one pole induced by the low-pass filter. It is much more stable than PLL. Thus, in RDLL, it is unnecessary to add the resistor into the low-pass filter to ensure system stability.

For phase noise prediction, on the other hand, all the signals added into the system

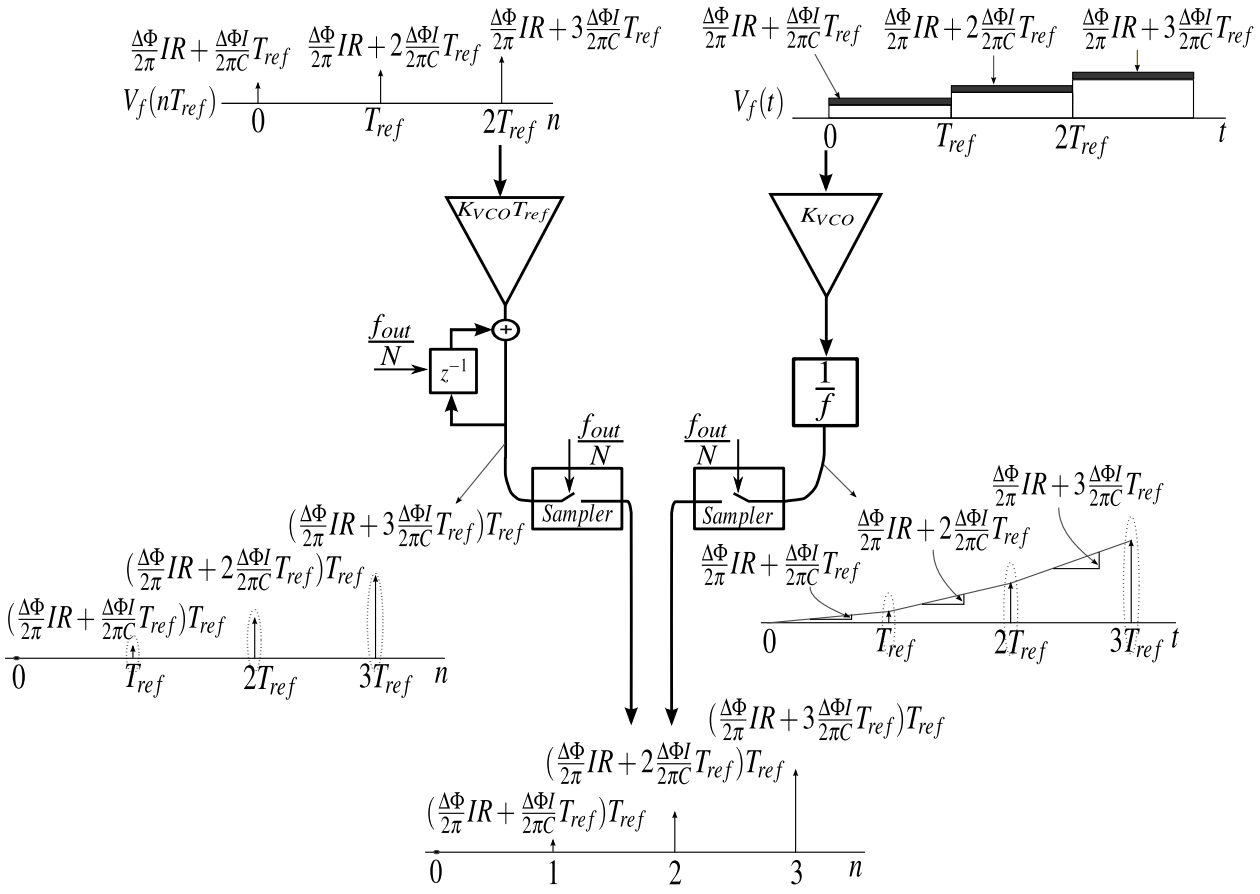


Figure 5.9: discretizing VCO

influence the system jitter performance, therefore they all need to be taken into account. As shown in Fig. 5.4, since the high sampling rate sampler in RMDL can distinguish the difference between accumulating jitter and stored jitter, it can not be ignored anymore. Thus, the main problem now is how to simplify this multi-rate system to a single-rate system while keeping the accuracy of phase noise prediction. It can be done from spectrum and time domain aspects. Here we will mainly focus on spectrum model simplification. The idea is the same as the one to derive the pseudocontinuous model in [23].

In frequency domain, sampling is replicating the original bandlimited ($-f_s$ to f_s) and shifting them by integer multiples of the sampling frequency while scaling the magnitude of the signal with sampling frequency $f_{sampling}$, as shown in Fig. 5.11 (b) (c). When sampling rate satisfies Nyquist-Shannon sampling theorem, different sampling rates change sampled

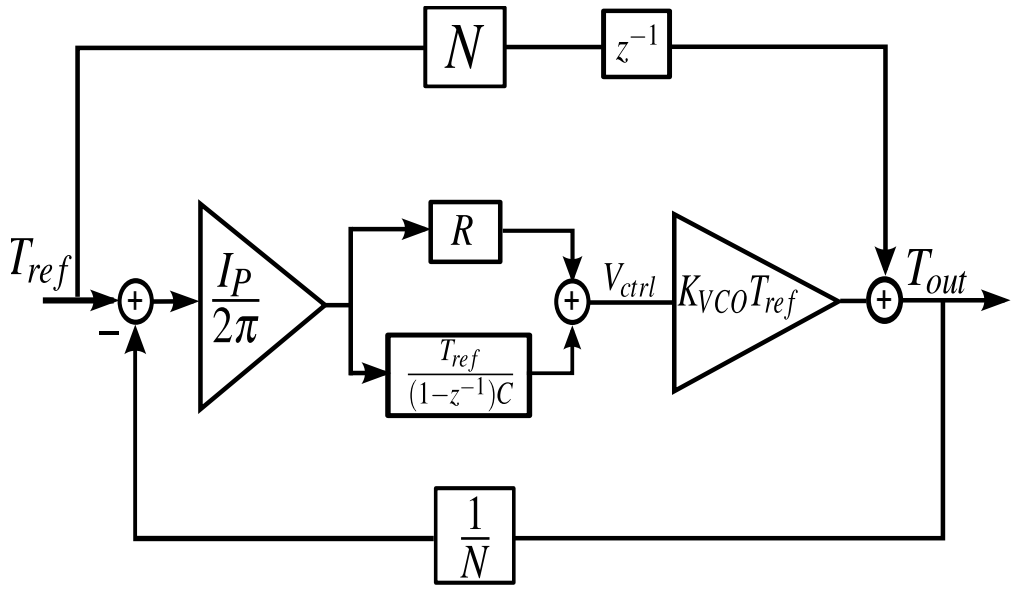


Figure 5.10: simplified RDLL model for stability consideration

signal magnitude without aliasing effect.

As validation data in section VII shows, DLL phase noise has a quick roll off beyond $f_{REF}/3$. Hence, it is a reasonable assumption that even at low sampling rate f_{REF}, the aliasing of the phase noise is still negligible. Then in the discrete MDL noise model. The hold function in storing path and integration function in VCO work as a low-pass filter. As the dashed line shows in Fig. 5.11 (b), It filters out most of the signal between two consecutive sampling impulses with high sampling frequency f_0. Meanwhile, the bandwidth of interest is small ($-f_s$ to f_s Fig. 5.11(a)). From these two points, it is reasonable to model the downsampling process as a magnitude reduction Fig. 5.11(d). Therefore the downsampler in divider can be simplified as a scaling factor $\frac{1}{N}$. As for PFD, since the detected signals (the reference input, the divider output) are already the downsampled version, the downsampler from PFD can be directly ignored.

Since the whole system has only one sampler with high sampling rate f_{out} after the simplification. Following the same method in the construction of stability model, the discrete-time phase noise model can be built. The only difference is that, now the whole system has the sampling rate of f_{out} which can catch finer phase change at the output compared to the

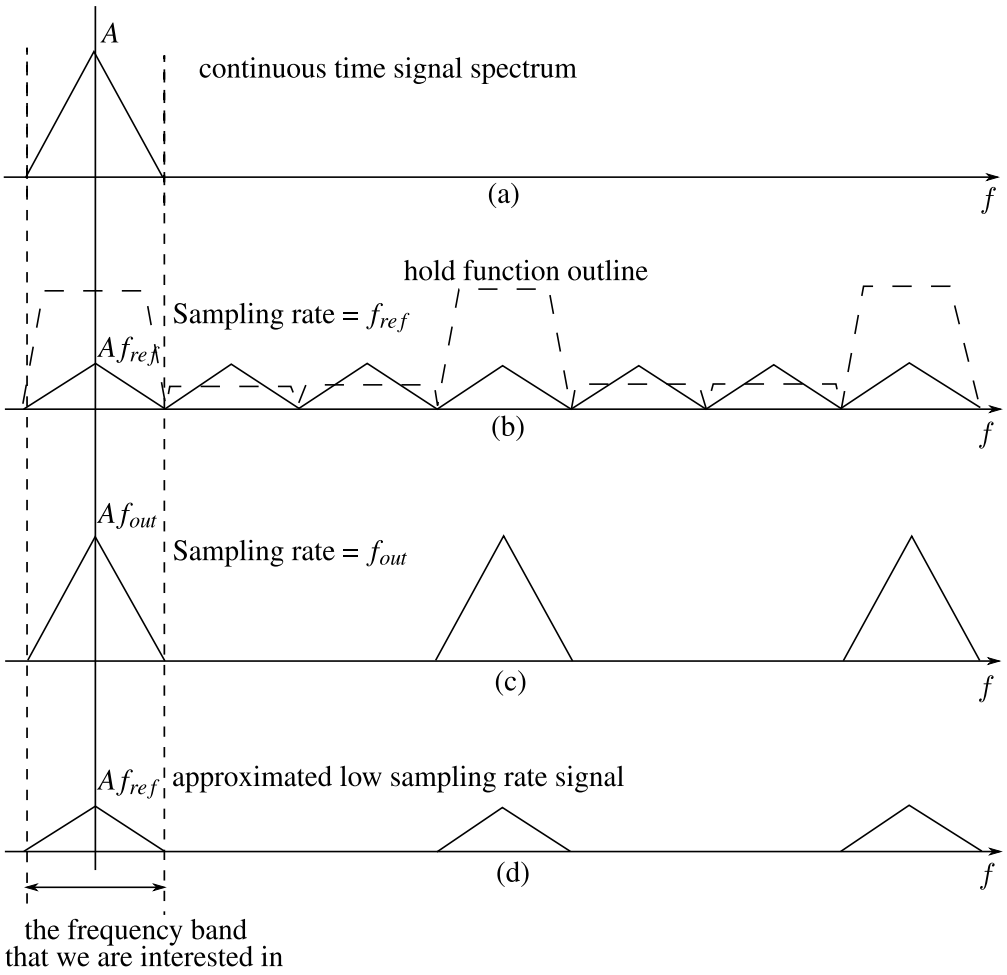


Figure 5.11: (a) continuous signal (b) sampling the continuous signal at f_{fref} sampling rate (c) sampling the continuous signal at f_0 sampling rate (d) approximate the low sampling rate signal.

stability model. Correspondingly, the changing step of the signal in the loop detected at each sampling instant will be smaller. Thus the phase shift gain is alternated to $K_{VCO}T_{out}$. So does the gain of the integrator in low pass filter. After all, the new system model for phase noise consideration is shown as Fig. 5.12. In the standard continuous-time PLL model [21], the divider is delivering time error. The dividing factor $1/N$ is caused by the change of time reference (the output time reference is T_{out}, the input time reference is T_{REF}). In the simplified discrete-time model, however, the time reference is always T_{out}. The divider of the simplified model directly delivers phase error instead of time error. The magnitude dividing

factor $1/N$ is caused by the changing of sampled signal magnitude at different sampling rates.

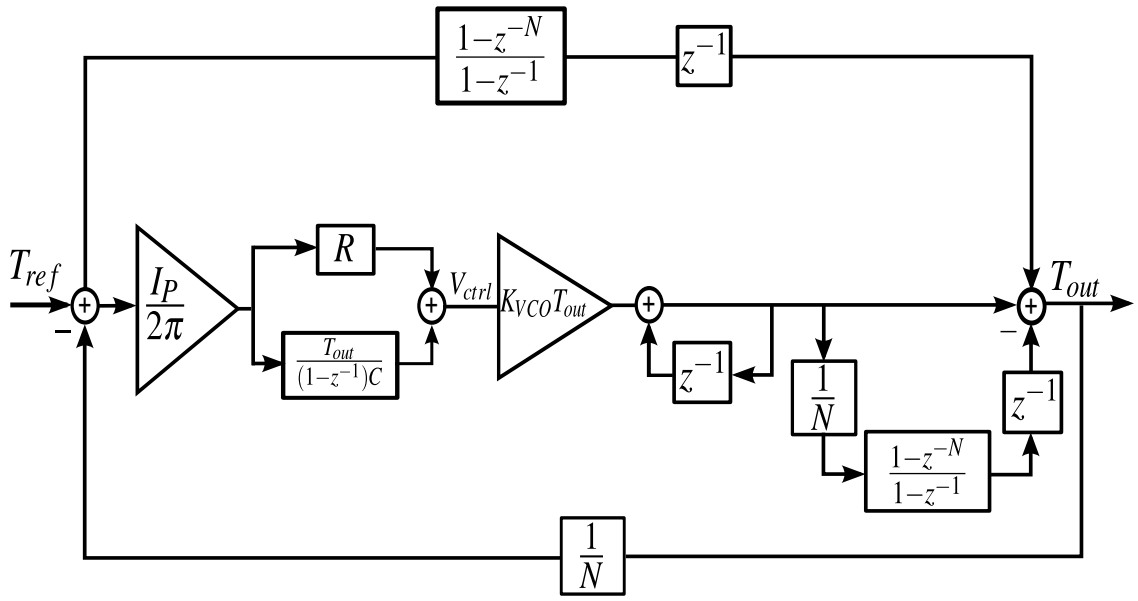


Figure 5.12: Simplified single-rate discrete-time model.

Above all, we build a new simplified discrete DLL model with fixed sampling rate. It is an easy and accurate model to predict the noise performance of DLL.

CHAPTER 6

ECDLL Spurs Analysis

In ECDLL, ideally the high frequency output is generated by combing the equally-spaced clock edges at the output of each delay cell in VCDL. Because of the mismatch effect in each delay cell, those intermediate clock edges are not completely equally-spaced, resulting in the clock duty cycle error at the output of edge combiner. Since the edge combiner periodically aligns the output clock edges of each delay cell in order, the duty cycle error of output clock also accumulates and resets in a periodical manner as the jitter does. For a certain chip, the mismatch effect in each delay cell is deterministic, unlike noise effect which is always random process. Therefore the edge combiner output clock can be decomposed to an idea clock CLK_{out} adding a periodical mismatch error as shown in Fig. 6.1. Because of the periodical and deterministic characteristic, mismatch error causes spurious tones in spectrum at the frequency f_{ref} and its harmonics.

For a certain chip, phase noise is still a random signal but spur is a deterministic signal. When taking infinite chips into account, spur also becomes a random signal. In other word, phase noise is ergodic random signal while spur is not. This is the main difference between spur and phase noise. Due to the deterministic and stochastic characteristics of spur, it is reasonable to analyze it as a deterministic signal at the beginning and then use the knowledge of stochastic to study its random process property.

Mismatch effect of each delay cell is assumed to follow Gaussian distribution and uncorrelated to each other. Edge combiner output is the combination of all delay cell outputs. Therefore the duty cycle error represents the sum of mismatch effects from all taps. According to central limited theorem, the duty cycle error also follows a Gaussian distribution. Spurs, which represent the magnitude of the duty cycle error signal, follow Ray-Leigh dis-

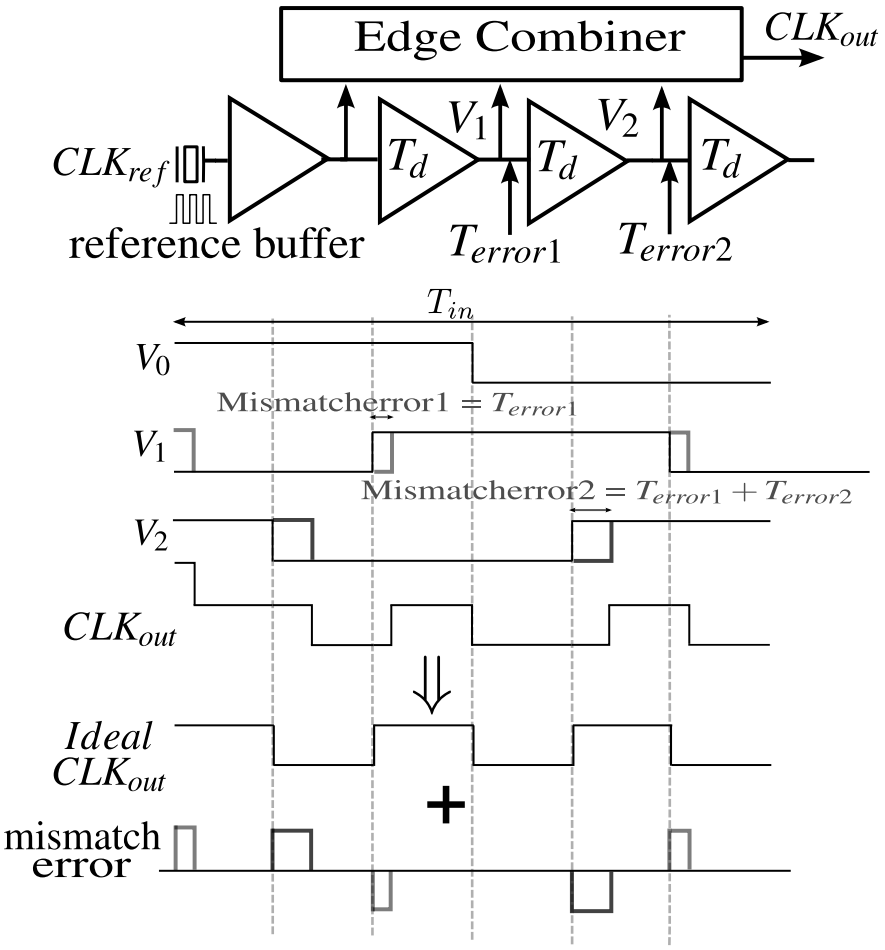


Figure 6.1: Spurs explanation in ECDLL

tribution. In [15], it provides the expression of the spur distribution as following:

$$p_x(x) = \begin{cases} \frac{x}{\sigma_R^2} exp(-x^2/2\sigma_R^2) & : x \geq 0 \\ 0 & : x < 0 \end{cases} \tag{6.1}$$

$$E[x] = \sigma_R \sqrt{\pi/2} \tag{6.2}$$

An interesting thing of the formula is that σ_R^2 is the expected value of spur's PSD [15]. Thus, if the expected value of PSD can be found, the distribution of spur is known.

In [14],[15], the Fourier Transform is directly used to calculate the PSD of spurs. But the

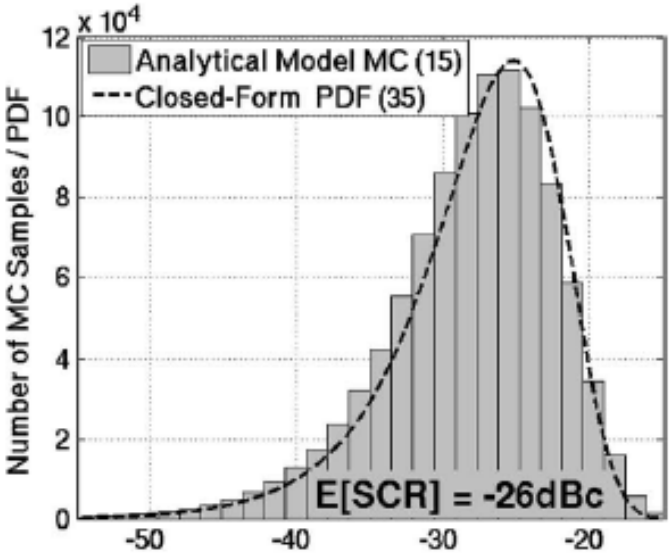


Figure 6.2: ECDLL spur distribution simulated by Amin Ojani

calculation is very complicated. Here, the new ECDLL system model is used to calculate spurs' PSD which is accurate and much simpler.

As Fig. 6.3 shows, delay cells provide their own mismatch errors which can be represented as a set of extra deterministic discrete-time impulse train $M_n[k]$ with period T_{ref}. In closed loop ECDLL, The Delay lock loop adds an extra control signal back to each stage in order to eliminate the time error between reference clock and last stage of the delay line as shown in Fig. 6.3. For a certain chip, mismatch error is a deterministic signal so the loop can detect it and eliminate the total mismatch error at the last delay cell. Then the control signal sent to each delay cell is the same which is the average value of the total delay lines' mismatch error as eq(6.3).

$$M_{ctrl}[k] = -\frac{1}{N}\sum_{n=1}^{n=N} M_n[k] \tag{6.3}$$

Above all, the closed loop mismatch error at output has an approximated energy distribution as Fig. 6.4 [4]. The mismatch error at the last stage is zero because the total control signal cancels the effect of total mismatch error at last stage.

Following the same idea as jitter-based noise analysis shown in section II, the PSD of

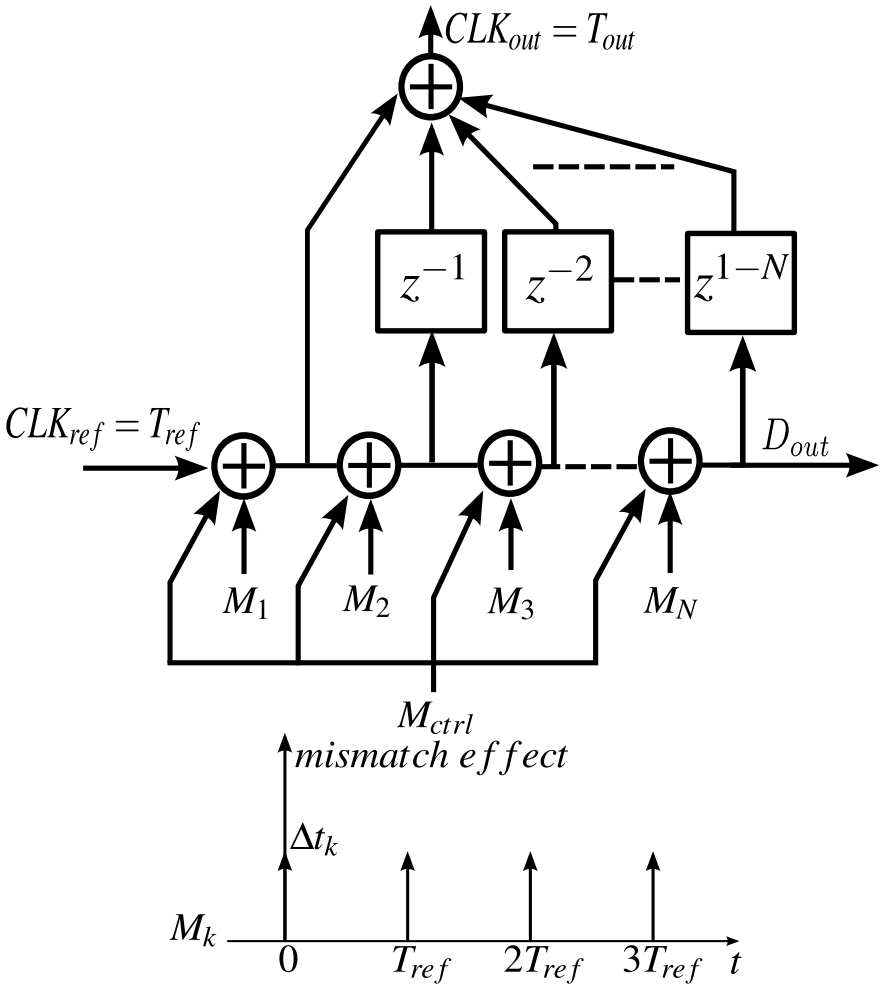


Figure 6.3: the source of spurs in ECDLL

mismatch error can be derived. In [24], it shows that autocorrelation function of a certain delay cell mismatch effect is:

$$R_{M_k}(t) = \frac{(\Delta t_k)^2}{T_{ref}} \sum \delta(t - nT_{ref}) \tag{6.4}$$

where Δt_k is the magnitude of mismatch error caused by a certain delay cell. It follows Gaussian Distribution with zero expectation and variance $\sigma^2_{\Delta t_k}$ [25].

Though Fourier Transform , the PSD of a delay cell mismatch effect is:

$$S_{M_k}(f) = \frac{(\Delta t_k)^2}{T^2_{ref}} \sum \delta(f - nf_{ref}) \tag{6.5}$$

Then the expected PSD of a delay cell mismatch error is:

$$\mathcal{E}\{S_{M_k}(f)\} = \frac{\sigma^2_{\Delta t_k}}{T^2_{ref}} \sum \delta(f - nf_{ref}) \tag{6.6}$$

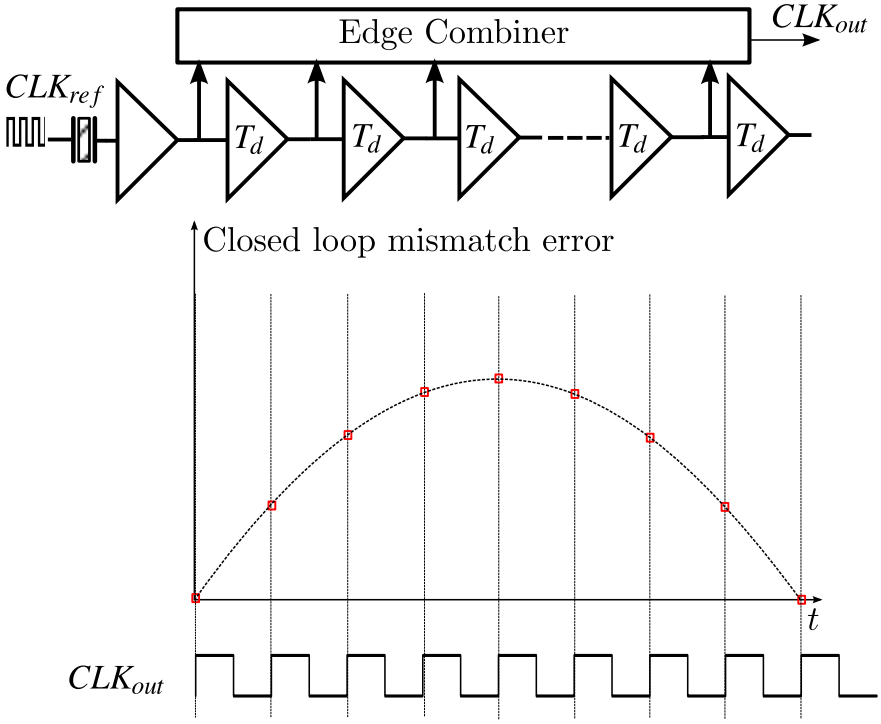


Figure 6.4: mismatch effect at edge combiner output

In section IV, the transfer functions from different input to the output in ECDLL have been derived. Based on those previous conclusions, the transfer function from control signal to edge combiner output is:

$$H_{ctrl}(z) = \frac{V_{out}(z)}{M_{ctrl}(z)} = (\sum_{n=1}^{N} nz^{n-1}) \tag{6.7}$$

The transfer function from kth delay cell output to the edge combiner output is:

$$H_k(z) = \frac{V_{out}(z)}{M_k(z)} = (z^{-(k-1)} \sum_{n=0}^{N-k} z^n) \tag{6.8}$$

When combining all the mismatch effect together, the edge combiner output response is:

$$spur(z) = \sum_{k=1}^{N} (M_k(z)H_k(z)) + M_{ctrl}(z)H_{ctrl}(z) \tag{6.9}$$

Since M_{ctrl} is a simple combination of all M_k as shown in eq(6.3), through the basic mathematical algebra (appendix I), the formula can be simplified to:

$$spur(z) = \sum_{k=1}^{N} (M_k(z)(\frac{z^{-(k-1)}}{1-z^{-1}} - \frac{1}{N}\frac{1-z^{-N}}{(1-z^{-1})^2})) \tag{6.10}$$

The PSD of the spur is:

$$S_{spur} = spur(z) \times \overline{spur(z)} = \sum_{k=1}^{N} S_{M_k}(z)|(\frac{z^{-(k-1)}}{1-z^{-1}} - \frac{1}{N}\frac{1-z^{-N}}{(1-z^{-1})^2})|^2 \tag{6.11}$$

Since the mismatch effects from different taps are uncorrelated, the expected value of spurs' PSD is:

$$\mathcal{E}\{S_{spur}\} = \sum_{k=1}^{N} \mathcal{E}\{S_{M_k}(z)\}|\frac{z^{-(k-1)}}{1-z^{-1}} - \frac{1}{N}\frac{1-z^{-N}}{(1-z^{-1})^2}|^2 \tag{6.12}$$

Spurs are the tones that appear at frequencies f_{ref} and its harmonics. At those frequency, z^{-N} is always 1. Therefore eq(6.12) can be further simplified to:

$$\mathcal{E}\{S_{spur}\} = \sum_{k=1}^{N} \mathcal{E}\{S_{M_k}(z)\}|\frac{1}{1-z^{-1}}|^2 \tag{6.13}$$

In the above equations, $z^{-1} = e^{-j2\pi f T_{out}}$. T_{out} is the edge combiner output period which is equal to T_{ref}/N.

Within the bandwidth of interests, $e^{-j2\pi f T_{out}}$ approximately equals $1 - j2\pi f T_{out}$. Then the first tone spur can be approximated as :

$$\mathcal{E}\{S_{spur}\} = \sum_{k=1}^{N} \mathcal{E}\{S_{M_k}(z)\}|\frac{1}{1-(1-j2\pi f T_{out})}|^2 \tag{6.14}$$

For each delay cell, the variance of mismatch effect is the same. Therefore eq(6.14) can be simplified as:

$$\mathcal{E}\{S_{spur}\} = N\mathcal{E}\{S_{M_k}(z)\}|\frac{1}{j2\pi f T_{out}}|^2 = N\mathcal{E}\{S_{M_k}(z)\}|\frac{1}{j2\pi f T_{ref}/N}|^2 \tag{6.15}$$

Therefore, combining eq(6.6) and eq(6.15), the first tone spur effect is:

$$\mathcal{E}\{S_{spur}\} = \frac{\sigma^2_{\Delta t_k}}{T^2_{ref}} \times \frac{N^3}{4\pi^2} \tag{6.16}$$

In order to coordinate with phase noise, eq(6.16) need multiply a scaling factor $4\pi^2/T^2_{out}$ to transfer it into the dimension of phase. Since the mismatch effect is continuously influencing the clock, another coefficient T^2_{out} is necessary to transfer the sampled mismatch effect back to continuous-time effect, leading to the following result:

$$\mathcal{E}\{S_{spur}\} = \frac{\sigma^2_{\Delta t_k}}{T^2_{ref}} \times \frac{N^3}{4\pi^2}\frac{4\pi^2}{T^2_{out}}T^2_{out} = \frac{\sigma^2_{\Delta t_k} N^3}{T^2_{ref}} \tag{6.17}$$

Eq(6.17) is the same as the result in [15] which is based on the most accurate method. Compared to the method in [15], the method based on the new ECDLL model is much more efficient, demonstrating that the proposed ECDLL system model is very useful.

Within the band of interest, for signal which is WSS random process like thermal noise, coefficient difference between the PSD of continuous-time signal and its own sampled version is T_{sample}. For deterministic bandlimited signal, on the other hand, the difference is T_{sample}^2. This discrepancy for different types of signals is caused by aliasing effect which has been mathematically demonstrated in [19].

CHAPTER 7

Verification

7.1 MDL universal model verification

To compare the phase noise predicted by eq(2.17) with experimental results, we refer to the circuit reported in [6], [5], [26] and [27].

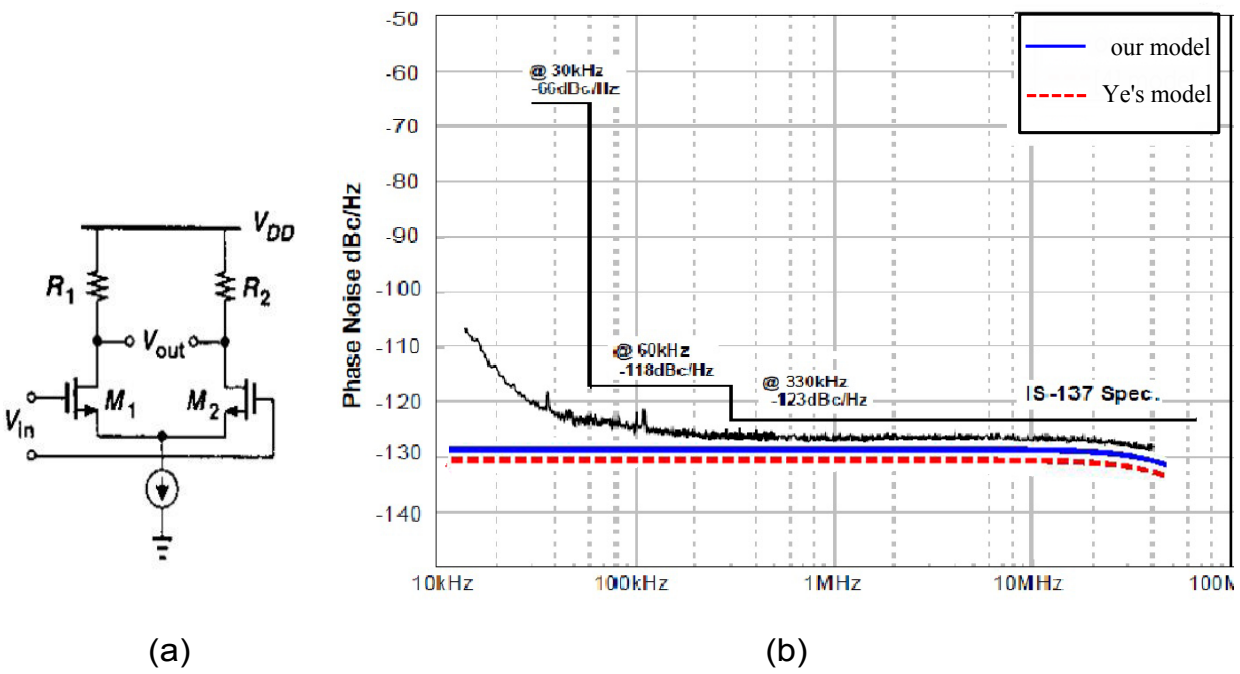


Figure 7.1: (a) The delay cell used in the MDL of [2], (b) Measurement results vs. theoretical predictions.

In [6], this circuit is an MDLL with a multiplication factor of 9. The delay cell is a differential inverter, shown in Fig. 7.1(a). A DLL encloses the delay line, but it has a very

low bandwidth and does not have a significant effect on the total phase noise of the output clock. [6] has compared its threshold formulas with the expiremental result. However, its theoretical value is based on an incorrect formula for the delay jitter([6],[28]); the correct formula appears in [11], and has been independently verified. The formula expresses the output jitter of a differential delay in terms of process parameters, temperature, and bias condition:

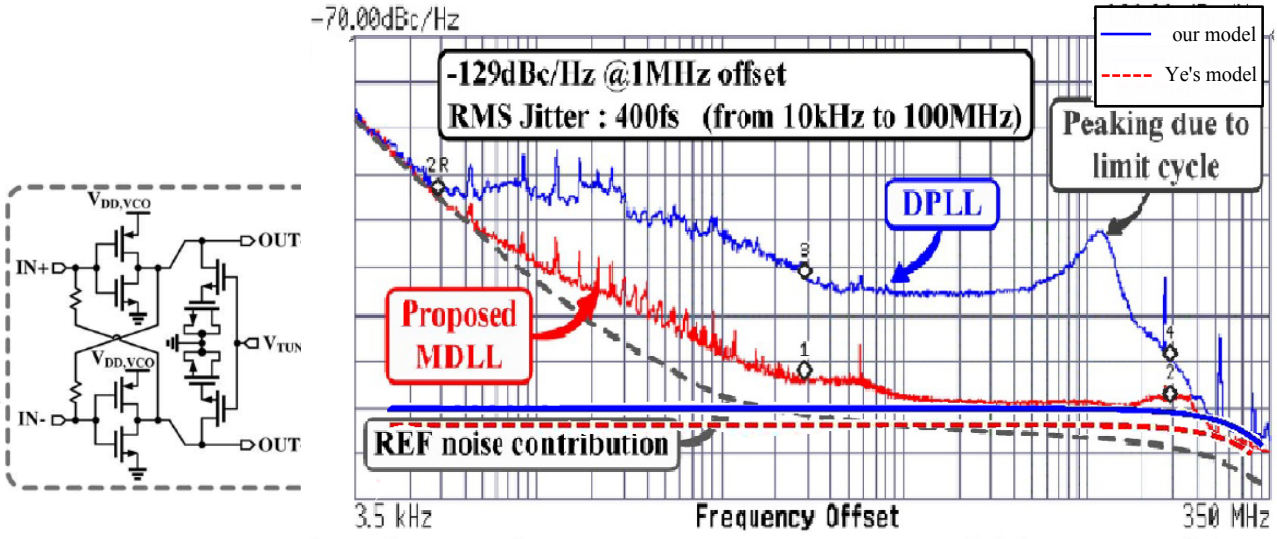
$$\sigma_n^2 = \frac{2KT}{f_0 I l n(2)} \left[\gamma \left(\frac{\frac{3}{4}}{V_{effd}} + \frac{1}{V_{efft}} \right) + \frac{1}{V_{op}} \right] \tag{7.1}$$

In this equation, V_{effd} is the effective gate voltage of the differential pair at balance, V_{efft} is the effective gate voltage of the tail FET, and V_{op} is the output swing. The only process-dependent value is γ, which we assume is 4/3. We estimate from the descriptions in [6] that $V_{effd} = 0.5V$, $V_{efft} = 1.5V$, and $V_{effd} = 0.8V$. Based on these values, the amount of the jitter is calculated and used in eq(2.16) and eq(2.17) to predict the output phase noise. Fig. 7.1 shows the comparison between measurement results and the calculated values of those two different models. At frequency offsets lower than the reference frequency, the PSD is almost flat, and two theoretical values match the measurement within a few dB. At low frequencies, due to other noise sources there is a greater discrepancy. In addition, the effect of the loop appears at low offset frequencies, which again is hidden under the measurement equipment noise. In this example, the dominant noise at moderate frequencies is the thermal noise of the delay cells. We can also find that two theoretical predictions have almost the same roll-off points, verifying that two models are close to each other at the frequency band that we are focus on.

The second data is from [27]. It uses inverter as delay cell. We find that the total DLL power consumption is $890uW$, the VCO consumes around 0.5 of the total power, V_{dd} is 1.1V, V_{th} is around 0.3V and the inverter is working in velocity saturation region. Therefore, we use the velocity saturation model of the inverter cell (eq(7.2) [11]) to calculate the jitter. We assume both γ_N and γ_P are 1.

$$\sigma_n^2 = 2 \times \frac{2KT}{I f_0} \left(\frac{g_{msat}}{2I} (\gamma_N + \gamma_P) + \frac{1}{V_{dd}} \right) \tag{7.2}$$

Where $g_{msat} = \frac{I}{V_{dd}-V_{th}}$ and the extra multiplier 2 is caused by the differential structure of the ring oscillator.



(a) (b)

Figure 7.2: (a) The delay cell used in the MDL of [9], (b) Measurement results vs. theoretical predictions.

As Fig. 7.2 shows, at moderate frequencies, both models' phase noise predictions are close to the measurement within a few dB error.

Both data's predictions are close to measurement results, validating the accuracy of our model. We can also find that our models predictions are higher and closer to the actual measurements than model in [5]. Because our model is more close to the actual operation of RDLL.

[5] and [26] have shown that the predictions of eq(2.16) are quite close to the measurement. Since we have proved the similarity between eq(2.16) and eq(2.17). Those two paper's results directly validate our model.

7.2 ECDLL system model verification

For stability consideration, our new model is the same as Edward Lee's model [10]. So we don't need verify that. What we are concerned is the correction of our ECMDL model. So we use Gorge Chein's [6] data again to do the verification.

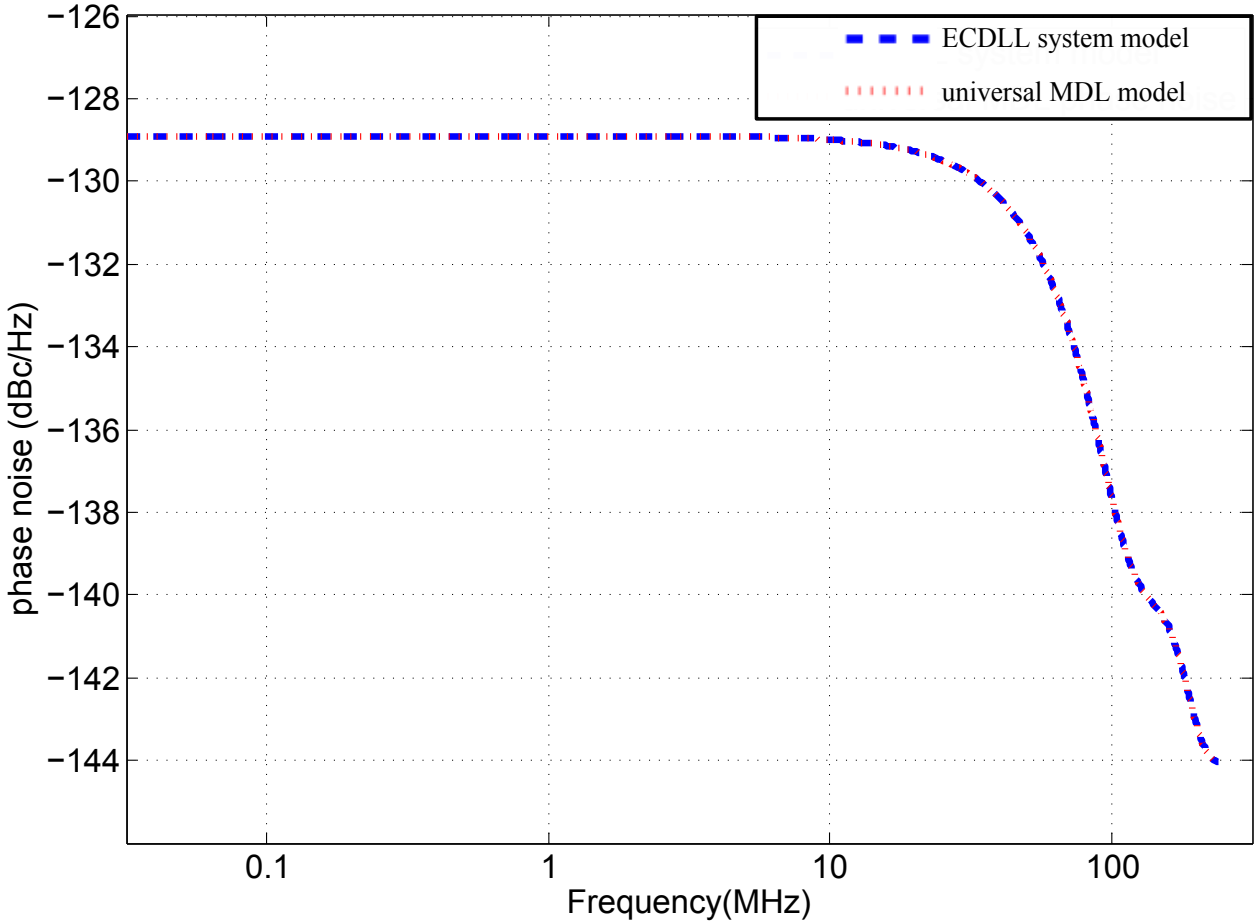


Figure 7.3: PSD comparison between the universal MDL model and new ECMDL model.

As Fig. 7.3 shows, our ECDLL model's prediction is completely the same as the result of our universal model. This means our ECDLL system model is very accurate.

7.3 RDLL system model verification

Here we will separate our new model into three parts to complete our verification. First we will validate the correction of our RMDL phase noise model and the input signal noise injection model. Then we will verify the correction of our discrete-time PLL model.

We use the data of [6] to do this comparison. As Fig. 7.4 shows, the MDL discrete-time model result is close to the random process model. The discrepancy between two models is caused by the simplification from a multi-rate system to a single-rate system.

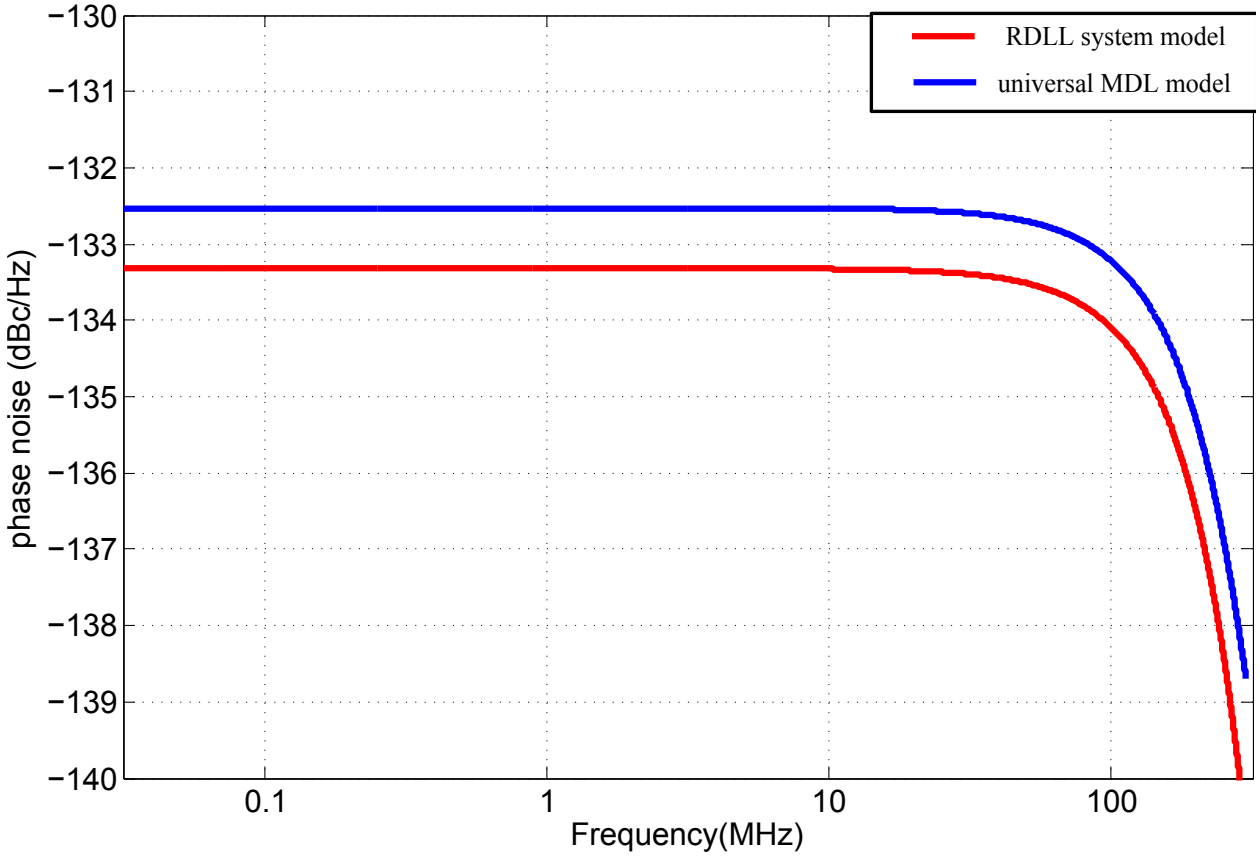


Figure 7.4: PSD comparison between the universal MDL model and discrete-time RMDL model.

For the reference model:

$$\frac{1 - e^{-j\omega NT_d}}{1 - e^{-j\omega T_d}} \tag{7.3}$$

when considering the low frequency part, we can simplify the model as

$$Ne^{-\frac{wNT_df}{2}} \frac{sin\omega NT_df}{\omega NT_d} \tag{7.4}$$

This expression is equal to the one in [5]. The Fig. 7.5 shows that two models have different roll off point. At that frequency, however, the phase noise of the reference is small, we can ignore this error Fig. 7.5.

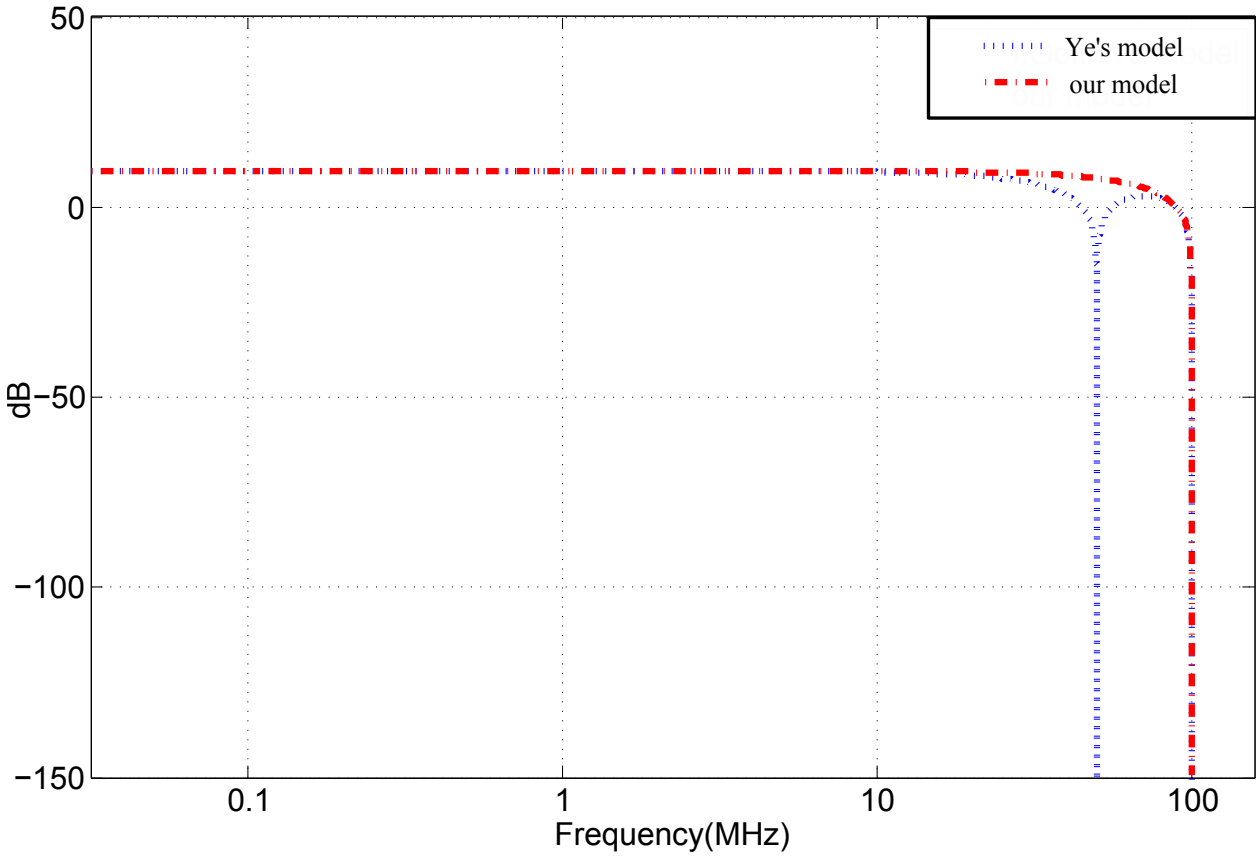


Figure 7.5: The feedthrough transfer function from input to output

As we have verified the correction of our MDL part model. Next, we will verify that our discrete-time model for PLL is also accurate enough for noise prediction. We use one of the PLL design data in [22] to do the validation. In Fig. 7.6, it shows the result of our discrete-time PLL model and the normal continuous-time PLL model. At the frequency that we are interested in, the two models have the same prediction. Hence, for the frequency that we are focus on, our simplified discrete-time model is accurate enough for phase noise prediction.

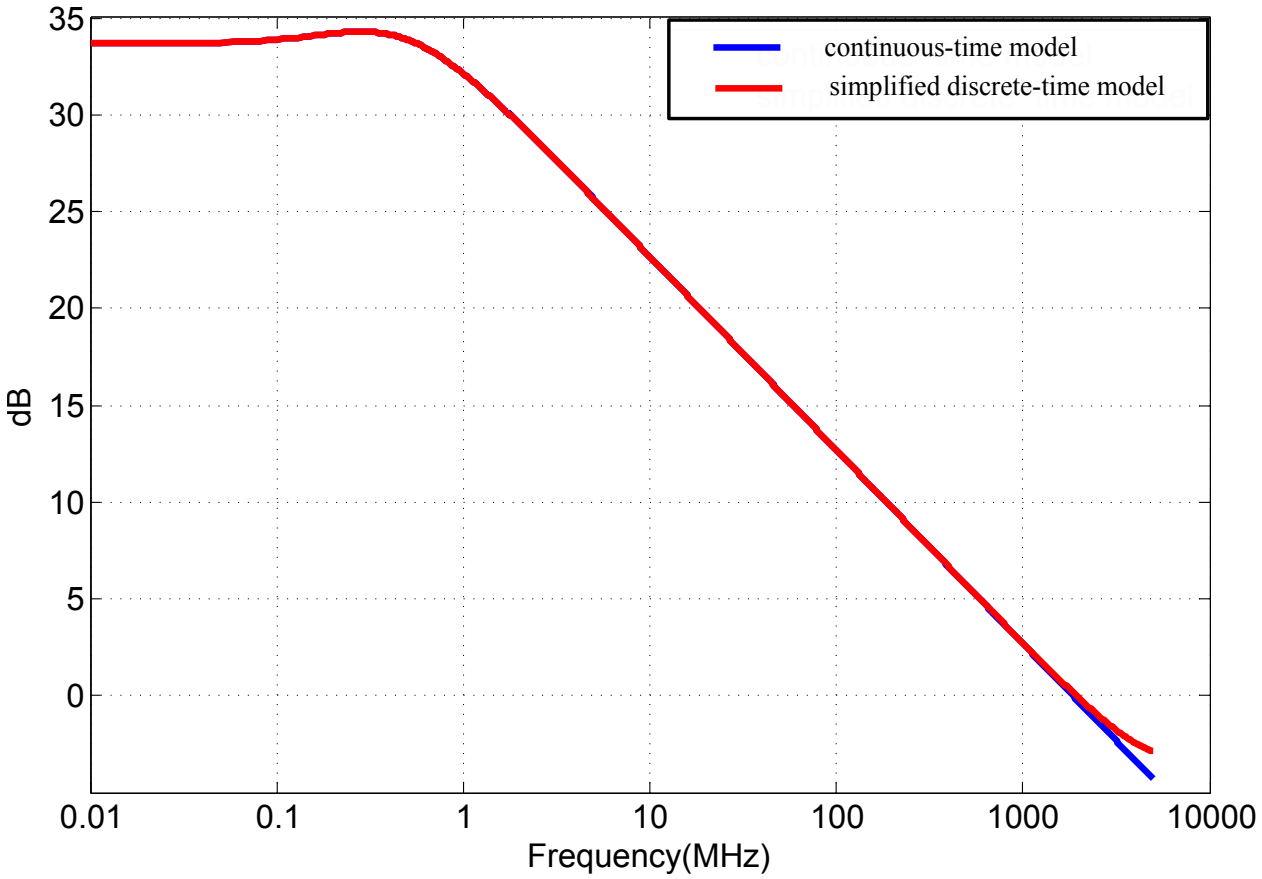


Figure 7.6: PLL model comparison.

CHAPTER 8

Conclusion

This thesis provides a complete analysis for noise performance in synthesizers. It is the first time that the high frequency generators of DLLs (MDLs) are extracted out to be analyzed alone. The same as VCO phase noise model, a simple model reveals the relationship between power, frequency and phase noise in MDLs is proposed, providing the designer with insights for the design of MDLs. Based on the jitter-based noise analysis which is proposed in this thesis, two most cited phase noise models are demonstrated to be the same. A simple and efficient approach for flicker noise analysis is provided.

In spite of the wide use of DLLs, there is still lack of understanding of DLL operations. This thesis proposes a complete and simple ECDLL model which can be used for both stability and phase noise analysis. Based on this new model, spurs in ECDLLs are also efficiently analyzed, avoiding the complicated calculations caused by Fourier Transform. As for RDLLs, the multi-rate characteristic of RDLL is illustrated. An elegant graphical analysis is demonstrated to prove that RDLL is truly a one-pole system. A clear and general process for multi-rate system simplification is also proposed in this thesis.

Above all, this thesis gives designer completely new insights of DLLs. The one-pole characteristic of DLLs implies that the loop bandwidth of DLLs can be large. Thus, DLLs are good choice for fast-locking synthesizer design. Meanwhile, the MDL model shows that the phase noise cancellation effect is completely related to input frequency. The lower reference frequency leads to worse noise performance. Therefore how to decouple such a relationship can be a problem. Those aspects are worth studying.

APPENDIX A

Spur Model Calculation

Here, we will show how to simplify the calculation for the following formula:

$$spur(z) = \sum_{k=1}^{N}(M_k(z)H_k(z)) + M_{ctrl}(z)H_{ctrl}(z) \tag{A.1}$$

$$H_{ctrl}(z) = (\sum_{n=1}^{N} nz^{n-1}) \tag{A.2}$$

$$H_k(z) = (z^{-(k-1)} \sum_{n=0}^{N-k} z^n) \tag{A.3}$$

Since $M_1(z) = M_2(z) = ... = M_k(z)$, eq(A.1) is simplified as:

$$spur(z) = \sum_{k=1}^{N}(M_k(z)(H_k(z) + \frac{1}{N}H_{ctrl}(z)) \tag{A.4}$$

For $H_k(z)$, it also can be simplified as below

$$\begin{aligned} H_{ctrl}(z) = (1 &+ z^{-1} + z^{-2} + z^{-3} + ... + z^{-(N-1)} \\ &+ z^{-1} + z^{-2} + z^{-3} + ... + z^{-(N-1)} \\ &+ z^{-2} + z^{-3} + ... + z^{-(N-1)} \\ &. \\ &. \\ &. \\ &+ z^{-(N-1)}) \end{aligned} \tag{A.5}$$

eq(A.5) equals to

$$H_{ctrl}(z) = (\frac{1-z^{-N}}{1-z^{-1}} + z^{-1}\frac{1-z^{-(N-1)}}{1-z^{-1}} + z^{-2}\frac{1-z^{-(N-2)}}{1-z^{-1}}... + z^{-k}\frac{1-z^{-(N-k)}}{1-z^{-1}}... + z^{-(N-1)}\frac{1-z^{-1}}{1-z^{-1}}) \tag{A.6}$$

eq(A.6) can be further simplified to:

$$H_{ctrl}(z) = (\frac{\sum_{k=0}^{N-1} z^{-k}}{1-z^{-1}} + \frac{Nz^{-N}}{1-z^{-1}}) \tag{A.7}$$

$$= (\frac{1-z^{-N}}{(1-z^{-1})^2} + \frac{Nz^{-N}}{1-z^{-1}}) \tag{A.8}$$

Meanwhile $H_k(z)$ can also be written as below:

$$H_k(z) = z^{-(k-1)} \times (\frac{1-z^{-(N-k+1)}}{1-z^{-1}}) \tag{A.9}$$

$$= (\frac{z^{-(k-1)} - z^{-N}}{1-z^{-1}}) \tag{A.10}$$

Therefore eq(A.1) equals to:

$$H(k) = \sum_{k=1}^{N} M_k(z) \times (\frac{z^{-(k-1)}}{1-z^{-1}} - \frac{1}{N}\frac{1-z^{-N}}{1-z^{-1}}) \tag{A.11}$$

As we can see, all the calculations are linear algebra. There is no integral or Tyler Extension. So our method is much simpler compared to the method in [15]

References

[1] K. H. Ryu, D. H. Jung, and S.-O. Jung, "A DLL Based Clock Generator for Low-Power Mobile SoCs," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 3, pp. 1950–1956, Aug 2010.

[2] T.-C. Lee and K.-J. Hsiao, "The Design and Analysis of a DLL-Based Frequency Synthesizer for UWB Application," *IEEE Journal of Solid-State Circuits*, vol. 41, no. 6, pp. 1245–1252, June 2006.

[3] S. Kazeminia, K. Hadidi, and A. Khoei, "A Low Jitter 110MHz 16-Phase Delay Locked Loop Based on a Simple and Sensitive Phase Detector," in *2013 21st Iranian Conference on Electrical Engineering (ICEE)*, May 2013, pp. 1–5.

[4] R. C. H. van de Beek, E. A. Klumperink, C. S. Vaucher, and B. Nauta, "Low-Jitter clock multiplication: A comparison between PLLs and DLLs," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing,*, vol. 49, no. 8, pp. 555–566, Aug 2002.

[5] S. Ye, L. Jansson, and I. Galton, "A Multiple-Crystal Interface PLL With VCO Realignment to Reduce Phase Noise," *IEEE Journal of Solid-State Circuits*, vol. 37, no. 12, pp. 1795–1803, Dec 2002.

[6] G. Chien, "Low-Noise Local Oscillator Design Techinques using a DLL based Frequency Multiplier for Wireless Applications," Ph.D. dissertation.

[7] B.-G. Kim and L.-S. Kim, "A 250MHz-2GHz Wide Range Delay-Locked Loop," in *Proceedings of the IEEE 2004 Custom Integrated Circuits Conference, 2004*, Oct 2004, pp. 139–142.

[8] R. Farjad-Rad, W. Dally, H.-T. Ng, R. Senthinathan, M.-J. Lee, R. Rathi, and J. Poulton, "A Low-Power Multiplying DLL for Low-Jitter Multigigahertz Clock Generation in Highly Integrated Digital Chips," *IEEE Journal of Solid-State Circuits*, vol. 37, no. 12, pp. 1804–1812, Dec 2002.

[9] T. Lee, K. Donnelly, J. Ho, J. Zerbe, M. Johnson, and T. Ishikawa, "A 2.5 V CMOS Delay-Locked Loop for 18 Mbit, 500 Megabyte/s DRAM," *IEEE Journal of Solid-State Circuits*, vol. 29, no. 12, pp. 1491–1496, Dec 1994.

[10] M.-J. Lee, W. Dally, T. Greer, H.-T. Ng, R. Farjad-rad, J. Poulton, and R. Senthinathan, "Jitter Transfer Characteristics of Delay-Locked Loops - Theories and Design Techniques," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 4, pp. 614–621, Apr 2003.

[11] A. A. Abidi, "Phase Noise and Jitter in CMOS Ring Oscillators," *IEEE Journal of Solid-State Circuits*, vol. 41, no. 8, pp. 1803–1816, Aug 2006.

[12] A. Hajimiri and T. Lee, "A General Theory of Phase Noise in Electrical Oscillators," *IEEE Journal of Solid-State Circuits*, vol. 33, no. 2, pp. 179–194, Feb 1998.

[13] X. Gao, E. Klumperink, and B. Nauta, "Advantages of Shift Registers Over DLLs for Flexible Low Jitter Multiphase Clock Generation," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 55, no. 3, pp. 244–248, March 2008.

[14] O. Casha, I. Grech, F. Badets, D. Morche, and J. Micallef, "Analysis of the Spur Characteristics of Edge-Combining DLL-Based Frequency Multipliers," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 56, no. 2, pp. 132–136, Feb 2009.

[15] A. Ojani, B. Mesgarzadeh, and A. Alvandpour, "Modeling and Analysis of Harmonic Spurs in DLL-Based Frequency Synthesizers," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 61, no. 11, pp. 3075–3084, Nov 2014.

[16] A. Papoulis, "Probability, Random Variables, and Stochastic Processes," 1991.

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

[18] A. Homayoun and B. Razavi, "Relation Between Delay Line Phase Noise and Ring Oscillator Phase Noise," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 2, pp. 384–391, Feb 2014.

[19] ——, "Analysis of Phase Noise in Phase/Frequency Detectors," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 60, no. 3, pp. 529–539, March 2013.

[20] G. Kranc, "Input-Output Analysis of Multirate Feedback Systems," *Automatic Control, IRE Transactions on*, vol. 3, no. 1, pp. 21–28, Nov 1957.

[21] F. M. Gardner, "Charge-Pump Phase-Lock Loops," *IEEE Transactions on Communications*, vol. 28, no. 11, pp. 1849–1858, Nov 1980.

[22] B. Razavi, "Rf microelectronics," second edition, 2008.

[23] M. Perrott, M. Trott, and C. Sodini, "A Modeling Approach for Σ-Δ Fractional-N Frequency Synthesizers Allowing Straightforward Noise Analysis," *IEEE Journal ofSolid-State Circuits*, vol. 37, no. 8, pp. 1028–1038, Aug 2002.

[24] M. Corinthios, "Signals, Systems, Transforms, and Digital Signal Processing with MATLAB," 2009.

[25] A. Oppenheim, "Discrete-Signal Processing," 1998.

[26] S. Ye and I. Galton, "Techniques for Phase Noise Suppression in Recirculating DLLs," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 8, pp. 1222–1230, Aug 2004.

[27] A. Elshazly, R. Inti, B. Young, and P. Hanumolu, "Clock Multiplication Techniques Using Digital Multiplying Delay-Locked Loops," *IEEE Journal of Solid-State Circuits*, vol. 48, no. 6, pp. 1416–1428, June 2013.

[28] T. Weigandt, B. Kim, and P. Gray, "Analysis of Timing Jitter in CMOS Ring Oscillators," in *1994 IEEE International Symposium on Circuits and Systems, 1994. ISCAS '94*, vol. 4, May 1994, pp. 27–30 vol.4.