Design of High-Order Delay-Locked Loops for Frequency Selectivity

Yan Li

yan.li8@mail.mcgill.ca

260613055

December 2016

Advisor: Gordon W. Roberts

Electrical and Computer Engineering Department
3480 University Street Room 633
Montreal, Quebec
Canada H3A 2A7

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of Master of Engineering

Abstract

In this thesis, a pole-zero positioning method in high-order DLL design is presented and verified through simulations and physical experiments. The general approach is based on selecting the transfer function of the closed-loop DLL and deriving the loop filter behavior based on the gain of the phase-detector and voltage-controlled delay line. The proposed approach does not rely on the principle of design based on achieving a desired phase margin specifications but rather is based on selecting a closed-loop DLL behavior based on a desired transfer function. Two types of transfer functions, i.e., the Gaussian transfer function that has the minimum settling time behavior and the a class of transfer functions that have minimum propagation delay, are analyzed and compared in high-order DLL designs. The MATLAB/Simulink simulation results are provided to support the conclusion that Gaussian transfer function is more favorable when the DLLs are used as time-mode filters (TMF). Based on the aforementioned method, a DLL integrated circuit (IC) is designed and fabricated using IBM CMOS 130 nm technology. It contains the PFD, CP, VCDL and input/output buffers on-chip and the loop-filters are designed off-chip. A 4-layer PCB was designed, together with the DLL IC, to come up with a set of 2nd to 8th order DLL circuits. The building blocks and the ancillary circuits are discussed in detail. The performances of the high-order DLLs are tested and evaluated by comparing the measured results to their expected values. The design method is verified by the experimental results.

Résumé

Dans cette thèse, une méthode de positionnement pôle-zéro dans la conception DLL d'ordre élevé est présentée et vérifiée par des simulations et des expériences physiques. L'approche générale repose sur la sélection de la fonction de transfert de la DLL en boucle fermée et sur la dérivation du comportement du filtre en boucle sur la base du gain du détecteur de phase et de la ligne à retard commandée en tension. L'approche proposée ne repose pas sur le principe de conception basé sur l'obtention d'une spécification de marge de phase souhaitée mais plutôt sur la sélection d'un comportement DLL en boucle fermée basé sur une fonction de transfert souhaitée. Deux types de fonctions de transfert, c'est-à-dire la fonction de transfert gaussienne et la fonction de transfert Min-Delay, sont analysés et comparés dans une conception DLL d'ordre supérieur. Les résultats de simulation MATLAB / Simulink sont fournis pour étayer la conclusion que la fonction de transfert gaussienne est plus favorable lorsque les DLL sont utilisés comme filtres temps-temps (TMF). Sur la base de la méthode susmentionnée, une puce DLL est conçue et fabriquée en utilisant la technologie IBM CMOS 130 nm. Il contient les tampons PFD, CP, VCDL et d'entrée / sortie sur puce et les filtres de boucle sont conçus hors-puce. Un circuit imprimé à 4 couches a été conçu avec la puce DLL pour créer les circuits DLL de la 2ème à la 8ème commande. Les blocs de construction et les circuits auxiliaires sont discutés en détail. Les performances des DLL d'ordre élevé sont testées et évaluées en comparant les résultats mesurés aux valeurs attendues. La méthode de conception a été vérifiée par les résultats expérimentaux.

Acknowledgements

First of all, I would like to express my gratitude to my supervisor Professor Gordon Roberts for his patience, guidance and advices throughout my Master's study and research. He introduces me into the world of integrated circuits and systems, and it is my honor to work in him research group. Next, I would like to thank all of my colleagues in Integrated Microsystems Lab for their company and encouragement during this period. I would like to thank especially Omar Abdelfattah and Steven Bielby for their technical knowledge, Moataz Abdelfattah and Ming Yang for their help in dealing with chip DRC and LVS errors and their advices in PCB design, and Dong An for his assistance in using testing equipment. Finally, I would like to thank my mother Gaiqin Li and father Xuejun Li for their support throughout my academic career. Their love and dedication give me the encouragement to move forward and become a better me. This thesis is dedicated to my parents.

Previous Pubilications

This thesis resulted in 3 separate but related publications. These are:

[1]Y. Li and G. W. Roberts, "Design of high-order type-II delay-locked loops using a Gaussian transfer function approach," 2016 IEEE International Symposium on Circuits and Systems (ISCAS), Montreal, QC, pp. 1786-1789, 2016.

[2] Y. Li, S. Bielby, A. Chowdhury and G. W. Roberts, "A jitter injection signal generation and extraction system for embedded test of high-speed data I/O," 2015 IEEE 20th International Mixed-Signals Testing Workshop (IMSTW), Paris, pp. 1-6, 2015.

[3] Y. Li, S. Bielby, A. Chowdhury, G. W. Roberts, “A jitter injection signal generation and extraction system for embedded test of high-speed data I/O,” Journal of Electronics Testing, vol. 32, no. 4, pp. 423-436, Aug. 2016.

Reference [1] is a result of my direct involvement in the work from beginning to end. Through the aid of my supervisor, this paper describes the theory of the DLL design using a pole-zero placement method involving a Gaussian transfer function. This paper was limited to simulation results only. A forth IEEE journal publication is in the work but not yet titled that will describe the results of the CMOS IC that was created for the sole purpose of demonstrating that the design theory is sound and accurate.

References [2] and [3] describe the design of a jitter injection signal generation and extraction system. This work was the efforts of several M. Eng. students under the supervision of Prof. Roberts over the past five years. My specific contribution was the inclusion of the DLL to act as an anti-imaging time-mode filter for the sigma-delta phase signal generation technique.

Contents

List of Figures

List of Tables

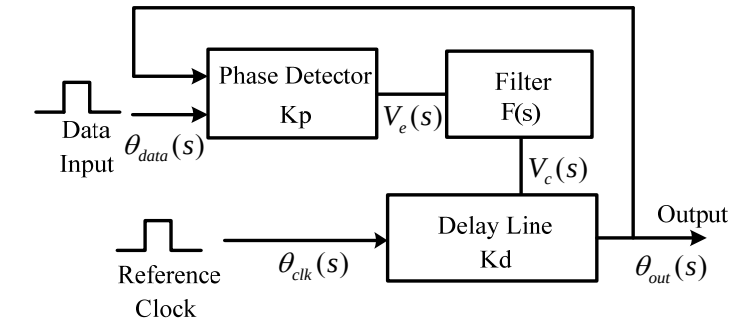
Abbreviations

TMSP	**T**ime-**M**ode **S**ignal **P**rocessing
DLL	**D**elay-**L**ocked **L**oop
PLL	**P**hase-**L**ocked **L**oop
TDC	**T**ime-to-**D**igital **C**onverter
DTC	**D**igital-to-**T**ime **C**onverter
TMF	**T**ime-**M**ode **F**ilter
IC	**I**ntegrated **C**ircuit
PSD	**P**ower **S**pectra **D**ensity
PFD	**P**hase **F**requency **D**etector
CP	**C**harge **P**ump
VCDL	**V**oltage-**C**ontrolled **D**elay **L**ine
ESD	**E**lectrostatic **D**ischarge
PCB	**P**rinted **C**ircuit **B**oard
OSR	**O**ver-**S**ampling **R**atio

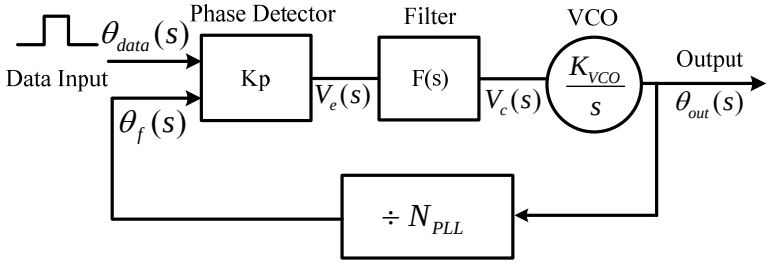
Chapter 1

Introduction

A delay-locked loop (DLL) is a negative feedback system that locks the output phase with an input reference signal that indicates the desired phase shift [1]. It is similar to a phase-locked loop (PLL) with the main difference being the absence of a voltage-controlled oscillator, replaced by a delay line. The comparison of the basic block diagrams of a DLL and a PLL are shown in Fig. 1.1. DLLs are used in wide number of applications from phase/frequency synthesizers to clock de-skewing circuits largely for low power operation when compared to similar built circuits constructed using phase-locked loops [2, 3].



(a)



(b)

Fig. 1.1: Block Diagrams of (a) DLL, and (b) PLL.

Compared to a PLL, a DLL circuit has no loop in the voltage control delay-line (VCDL); therefore, it is easy to design without severe stability issues and offers better jitter

performance since clock jitter is not accumulated [4]. The first application of a PLL used as a reconstruction filter can be traced back to [5, 6]. In this thesis, a pole-zero positioning method will be proposed in deigning high-order (2^{nd} to 8^{th} order) DLLs as time mode filters (TMFs). With a steeper roll-offs in frequencies response, a high-order DLL could produce a better performance in filtering the out-of-band quantization noise at the output of the sigma-delta modulated phase-domain signal generation circuit.

1.1 Phase-domain Signal Generation Scheme

Time-mode signal processing (TMSP) technique has been widely used in system testing and measurement such as the time-of-flight measurements of laser radars for distance and velocity measurement [7], RF synthesizers [8] and [9], calibration of automatic test equipment systems [10], and telecommunication applications such as PM and FM modulators and demodulators [11]. Instead of using voltage as the traditional analog signal, it involves the encoding of information in the form of time difference variables using phase modulation. Two fundamental blocks: time-to-digital converters (TDCs) and digital-to-time converters (DTCs) have taken the place of ADCs and DACs to convert signals between digital-domain and time-domain. This would help to solve some of the noise and interference problems as the technology scales down and system operation frequency speeds up [12].

According to [13], the way a DTC takes a digital input and converts it to the form of a time instant might be considered equivalent to a phase-modulation (PM) process. The phase difference between the output pulse and the clock signal will be the analog equivalent of the input digital code. Conversely, a TDC is responsible for converting a phase difference between two clock signals into a digital number. Both DTC and TDC are discrete components and go through sampling processes. This requires the output of DTC and the input of TDC to be bandwidth- limited in order to prevent aliasing effects.

One of the phase-domain signal generation schemes is based on sigma-delta phase encoding technology. This phase signal generation scheme was first proposed in [6] to synthesize phase domain signals such as jitter, and was then used in [14] as a jitter generation technique for high-speed I/O test. It involves an amplitude-domain sigma-delta modulation circuit followed by a DTC and the output of which passes through a time-mode filter (TMF), as shown in Fig. 1.2. An arbitrary signal at the input is pulse density modulated using a 1-bit

sigma-delta modulation process. Subsequently, each sigma-delta modulated bit is mapped to a corresponding discrete phase through a binary-weighted DTC operation. One example of the 4-bit 90 degree DTC mapping method is illustrated Table 1.1. Every "1" value at the sigma-delta output represents 0° and is mapped to the bit sequence "1100" and every "0" represents 90° and is mapped to the sequence"0110". Note that the same concept can be extended to a multi-bit, different phase range and/or duty cycle encoding process.

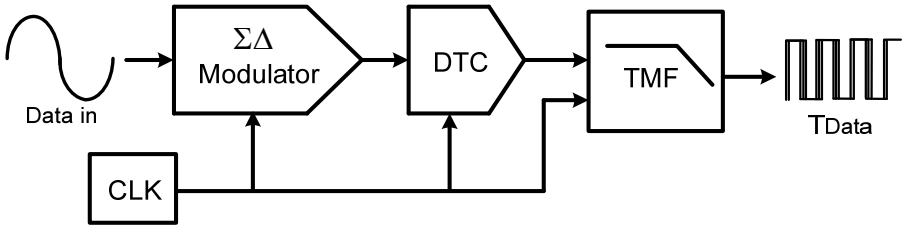


Fig. 1.2: Phase-domain signal generation scheme

Table 1.1: Phase-Encoding regulation

Sigma-delta Output	4-bit Phase-Encoding	Phase
1	1100	0^o
0	0110	90^o

The sigma-delta modulation stage is used to encode a bandwidth-limited signal into a 1-bit digital sequence. As a 1-bit signal is a poor approximation of any real-valued signal with almost infinite precision, a quantization error results. To minimize the effect of this quantization error, ΣΔ encoding shifts the quantization error to an out-of-band frequency region where it can be filtered out using a frequency-sensitive filter and recovers most of the original signal. Fig. 1.3 illustrates the PSD at each stage of the phase signal generation process. The PSD of the sigma-delta output $S_{\Sigma\Delta}(f)$ can be decomposed into [15] a signal $S_{S,\Sigma\Delta}(f)$ and noise component $S_{N,\Sigma\Delta}(f)$. Therefore, the PSD of the DTC output could be expressed as:

$$S_{\Phi 1}(f) = \alpha_{\Phi}^2 S_{S,\Sigma\Delta}(f) + \alpha_{\Phi}^2 S_{N,\Sigma\Delta}(f) \tag{1.1}$$

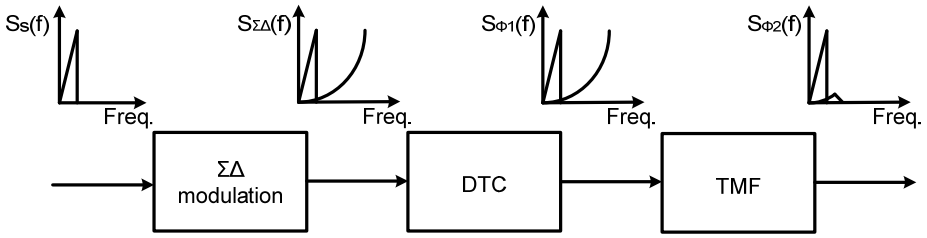


Fig. 1.3: PSD mapping from amplitude domain to phase domain

It can be observed from Fig. 1.2 that the time-mode filter (TMF) should be carefully designed to properly filter the out-of-band quantization noise and convert the digital signal into a corresponding continuous-time phase signal. It is therefore the objective of this thesis to investigate and construct the optimum design of the TMF using a class of minimum overshoot step-response type filter transfer functions, such as the Bessel-Thomson, Gaussian and the newly introduced minimum-delay type transfer functions [16]. Minimum overshoot type filter transfer functions have the fastest settling time to step inputs such as those coming from a sigma-delta modulator.

1.2 Overview of DLL Designs

1.2.1 DLL Design Methods

Several papers have addressed the design method of DLL. E. Lee *et al* [17] provided a design method for 2nd-order DLLs based on a z-domain model. Subsequently, J. Burnham *et al* extended these methods to include a more comprehensive analytic model of DLL behavior both in the s domain and the z-domain, which enables the calculation of input-output transfer functions, stability behavior and static phase errors [18]. However, both approaches were limited to second-order DLLs, largely a result of the mathematical complexities of the proposed theories. In this thesis, a completely different approach is undertaken. A DLL of any order is synthesized based on a transfer function of input/output behavior. As the method is general, any transfer function type can be used. While in this work, we limit our discussion to those transfer function with minimum step response settling time, e.g., Gaussian-type transfer functions, it is not limited to these. Subsequently, stability, frequency response and static phase errors can all be deduced from the initial transfer function, and better yet, the selection of the high-order loop filter can be easily determined.

1.2.2 High Speed and Wide Frequency Range DLLs

Improvements have been continuously proposed in DLL circuit design. As the operating frequency of high-speed devices exceeds gigahertz, one problem is the need of increasing the phase difference detectability. For this purpose, several fast speed phase frequency detector (PFD) topologies have been proposed. Compared to the conventional static-state PFD [19], a new type of dynamic-logic PFD was adopted in [20] that achieves a faster operation speed because of the less number of transistors used. However, it still has the dead-zone problem and the detection range is limited to $(-\pi, \pi)$. Later, through the introduction of two half-transparent (HT) registers together with a NOR gate, the dead-zone problem was eliminated. Moreover, the detection range increased by a factor of two to $(-2\pi, 2\pi)$ [21, 22]. Another novel dynamic-logic PFD aiming at the same objectives is provided in [23]. It could work under gigahertz frequency with no visible dead-zone and reduced blind-zone. In this thesis, the PFD topology in [23] is adopted and revised to work together with a DLL start-up circuit. This helps to solve the DLL false-locking problem at start-up condition as well.

Another problem of a conventional DLL is that its operational frequency range is limited since the DLL adjusts only phase, not frequency. Therefore, various wide-range DLLs architectures have been developed to solve the false locking problem under wide frequency operation. One typical method is adding a phase selection circuit combined with a multi-controlled voltage controlled delay line (VCDL). Before the DLL begins to lock, the phase selection circuit will decide the number of the delay cells [22] or the delay time of each delay cell [24] according to the different input frequencies, thus reaching a wider frequency range. For example, [22] and [24] obtained a frequency range of 6 MHz to 130 MHz and 32 MHz and 320 MHz, respectively. However, additional IC area and power consumption of these wide-range DLLs could be excessive, and furthermore, their jitter performance may get worse compared with conventional DLLs since the number of delay cells in the clock propagation paths becomes larger. Another strategy proposed to further increase the operation range and maintain the jitter performance is the dual-loop architecture [25, 26]. The core loop is used to generate coarsely spaced clocks, which are then used by a peripheral loop to generate the main system clock through phase interpolation. By using the dual-loop, a frequency range of 150MHz to 600MHz was easily accomplished in [25]. In this thesis, the phase selective and multi-controlled VCDL method is adopted for its circuit simplicity. By

selecting the number of the delay cells and the delay time of each cell at the same time, this thesis aims at designing a DLL that has a wider frequency range.

1.3 Thesis Outline

This thesis focuses mainly on the high-order DLL design method and verification. The performances of the high-order DLLs used as TMFs are examined through their transient response and frequency response. During the circuit design, some of the conventional DLL problems such as the speed, frequency range, false-locking at start-up, matching issue in charge pump are considered and researched to some extent. However, due to the time limitation, not all of the aforementioned problems have been discussed in detail. An outline of the thesis is as follows.

In this thesis, a pole-zero positioning method in high-order DLL design is presented and verified through simulations and physical experiments. The system-level designs are presented in Chapter 2. It starts from the DLL basics including the models, transfer functions and the system characteristics such as linear region and settling time. Then the pole-zero positioning method is introduced for DLL design. Detailed examples are given on system parameter calculations and MATLAB/Simulink simulations are provided to verify the correctness of the system-level designs. The circuit-level design of DLL main building blocks and their SPICE simulations are presented in Chapter 3. The circuit topologies of PFD, CP, loop filters and VCDL are provided as well as the reasons why the proposed topologies are chosen among other prevalent topologies. SPICE simulations are used in this chapter to verify the correctness of the circuit design. The IC implementation and simulations are presented in Chapter 4. The main building blocks discussed in the previous chapter and some other ancillary circuits are integrated into a 1mm x 1mm CMOS IC die. IC-level layout and simulations are provided as well. Chapter 5 presents the on board testing results of the high-order frequency selective DLLs. The custom PCB designed for testing purpose is introduced at the beginning. The DLL IC fabricated in 130nm CMOS technology and housed in a SOIC 24-pin package was sold onto the PCB and interfaced to a programmable 2^{nd} to 8^{th} order loop filter. The measured characteristics of the DLL are analyzed and compared to simulation and theoretical values. Finally, conclusions and directions for future work are provided in Chapter 6.

Chapter 2

The DLL Design Method Using a Pole-Zero Positioning Approach

In this chapter the system-level DLL design approach will be presented. It will begin with the DLL basics including the DLL models, transfer functions and the system characteristics such as linear range of operation and settling time. A pole-zero positioning method will be introduced for the DLL design. Detailed examples will be used to demonstrate the design principles beginning with the desired input-output specifications. Finally, the MATLAB/Simulink simulations will be provided to verify the correctness of the system level design.

2.1 DLL Basics

The overall DLL block diagram is shown in Fig. 2.1. This diagram was introduced in Chapter 1 but is repeated here for consistency. It contains a phase frequency detector (PFD), which is used to compare the phase difference between some clock-like data input signal and the output of a voltage-controlled delay line (VCDL). The PFD sometimes drives a charge pump (CP), which, together with a loop filter, is used to set up the control voltage of the delay line. The delay line is driven with a reference clock input at the same frequency as the data input.

Within the plot, $\varphi_{data}(t)$ represents the time domain input data, $\varphi_{clk}(t)$ represents the reference clock, and $\varphi_o(t)$ is the DLL output signal. The voltage variable $\upsilon_e(t)$ represents the output voltage of the PFD/CP generated by phase errors and the output of the loop filter $\upsilon_c(t)$ is used as the control voltage for the delay line.

DLLs are divided into two types, Type-I and Type-II, depending on the nature of the input signal. If a single clock-like signal is used to drive both the data and the clock inputs then it is

classified as Type-I DLL. If the data and clock inputs are separated, as depicted in Fig. 2.1, then it is called a Type-II DLL. As a Type-II DLL is the more general of the two, this work will only focus on the design and synthesis of this particular DLL type.

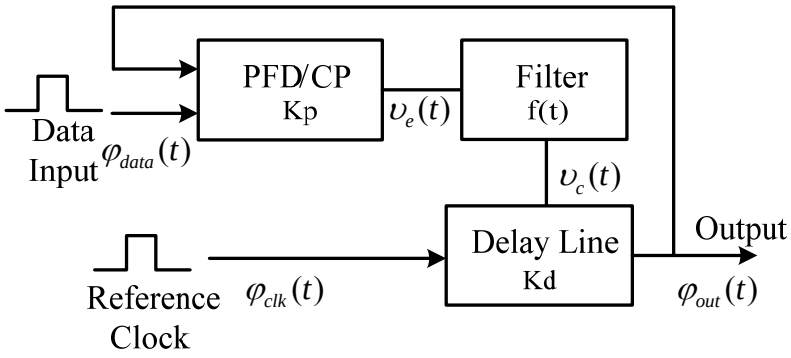


Fig. 2.1: Block diagram of a Type-II DLL in the time-domain.

According to J. Burnham's work [27], the s-domain linear continuous-time model of a Type-II DLL system could be plotted as Fig. 2.2. Within the plot, K_d and K_P are the coefficients of the phase frequency detector and the delay line, respectively. They are assumed to be constant values. $F(s)$ is the transfer function of the loop filter. $\theta_{data}(s)$ and $\theta_{clk}(s)$ are the data input and the reference clock, respectively, and $\theta_{out}(s)$ is the output signal. The $2\pi/s$ represents the phase discrepancy between the input signal and the output signal under comparison, i.e. we assume that the DLL will lock to the next coming edge.

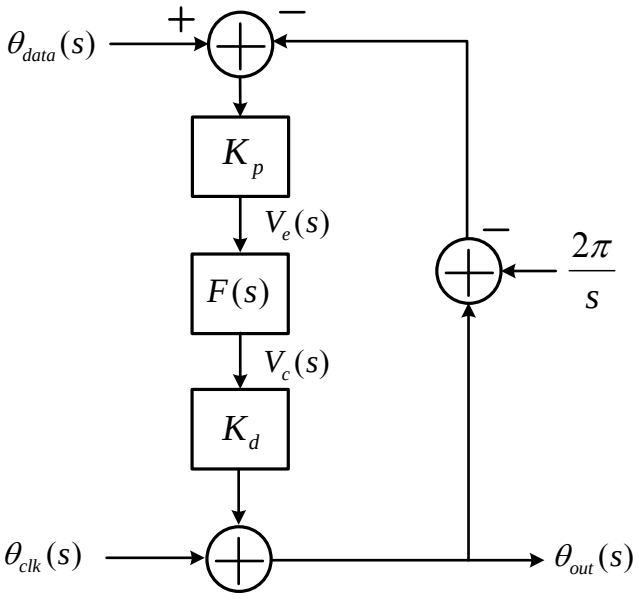


Fig. 2.2: Type-II DLL linear continuous-time model in the Laplace domain.

According to Fig. 2.2, the following block diagram relationship can be written:

$$V_e(s) = K_p(\theta_{data}(s) - \theta_{out}(s) + \frac{2\pi}{s}) \tag{2.1}$$

$$V_c(s) = F(s)V_e(s) \tag{2.2}$$

and

$$\theta_{out}(s) = \theta_{clk}(s) + V_c(s)K_d \tag{2.3}$$

Using these relationships, several critical system transfer functions could be derived. These are listed below.

1) Input-Output Transfer Function:

Substituting Eqns. (2.1) and (2.2) into (2.3), one can write

$$\theta_{out}(s) = \theta_{clk}(s) + K_d K_p F(s)\left(\theta_{data}(s) - \theta_{out}(s) + \frac{2\pi}{s}\right) \tag{2.4}$$

Isolating the output phase term $\theta_{out}(s)$ on the left-hand side, one can write

$$\theta_{out}(s) = \frac{1}{1+K_d K_p F(s)}\theta_{clk}(s) + \frac{K_d K_p F(s)}{1+K_d K_p F(s)}\theta_{data}(s) + \frac{K_d K_p F(s)}{1+K_d K_p F(s)}\frac{2\pi}{s} \tag{2.5}$$

Eqn. (2.5) describes the output phase term $\theta_{out}(s)$ in terms of the data amd clock inputs, $\theta_{data}(s)$ and $\theta_{clk}(s)$. A third term on the right most side is also present. It represents the initial power-on behavior of the DLL and it modeled as a 2π-phase angle step-like behavior.

2) Control Voltage Transfer Function:

Likewise, the control voltage appearing at the input of the VCDL can be isolated from the system of equations described by Eqns. (2.1) - (2.5), resulting in

$$V_c(s) = \frac{K_p F(s)}{1+K_d K_p F(s)}\left(\theta_{data}(s) - \theta_{clk}(s) + \frac{2\pi}{s}\right) \tag{2.6}$$

3) Start-up Condition:

If the system starts up from a rest condition, we can assume that $\theta_{clk}(s) = \vartheta_{clk}/s$ and $\theta_{data}(s) = \vartheta_{data}/s$ where ϑ_{clk} and ϑ_{data} represents the clock and data phase change at t=0+. According to Eqns. (2.5) and (2.6), the output phase term and the control voltage at the start-up condition is as follows:

$$\theta_{out}(s) = \frac{1}{1+K_d K_p F(s)}\frac{\vartheta_{clk}}{s} + \frac{K_d K_p F(s)}{1+K_d K_p F(s)}\frac{\vartheta_{data}}{s} + \frac{K_d K_p F(s)}{1+K_d K_p F(s)}\frac{2\pi}{s} \tag{2.7}$$

and

$$V_c(s) = \frac{K_p F(s)}{1 + K_d K_p F(s)} \left(\frac{\vartheta_{data}}{s} - \frac{\vartheta_{clk}}{s} + \frac{2\pi}{s} \right) \tag{2.8}$$

If $\vartheta_{data} = \vartheta_{clk}$ (the input data and reference clock are in phase), the control voltage becomes

$$V_c(s) = \frac{K_p F(s)}{1 + K_d K_p F(s)} \frac{2\pi}{s} \tag{2.9}$$

Eqn. (2.9) refelects that the control voltage has a step response at start-up condition.

4) Excess Phase Trandfer Function:

When the system becomes stable (after DLL is locked), it is possible to decompose the input data signal into a constant clock input and an excess phase term, as follows

$$\theta_{data}(s) = \theta_{CC} + \phi_{in}(s) = \frac{\omega_{in}}{s^2} + \phi_{in}(s) \tag{2.10}$$

Here $\theta_{CC} = \omega_{in} / s^2$ describes a clock signal with a constant frequency ω_{in} and $\phi_{in}(s)$ is responsble for the changes in the DLL from its steady-state operation. It is assumed that the clock signal driving the data signal is the same as the reference clock driving the input of the delay line, $\theta_{clk}(s)$.

Likewise, the output of the DLL can also be separated into constant frequency term and an output excess phase term, with the latter term written as

$$\phi_o(s) = \frac{K_p F(s)}{1 + K_d K_p F(s)} \phi_{in}(s) \tag{2.11}$$

5) Linear Region of Operation:

The linear region of operation is one important characteristic of a DLL. A good DLL system should have a large linear region of operation in order to cope with large phase changes at its input. The DLL linear region of operation depends mainly on the linear behavior of its phase detector and delay-line. In the case of the phase detector, the linear region of operation can be described in terms of the phase detector output limits according to:

$$V_{PFD,\min} \le \upsilon_e(t) \le V_{PFD,\max} \tag{2.12}$$

Likewise, the linear region of operation for the delay line can be described in terms of the control voltage of the voltage controlled delay line (VCDL) according to:

$$V_{VCDL,\min} \le \upsilon_c(t) \le V_{VCDL,\max} \tag{2.13}$$

Defining the Laplace variables $V_e(s) = L\{\upsilon_e(t)\}$ and $V_c(s) = L\{\upsilon_c(t)\}$, we can relate the

above two internal DLL variables to the input phase signal according to $V_e(s) = T_e(s)\phi_{in}(s)$ and $V_c(s) = T_c(s)\phi_{in}(s)$ where

$$T_e(s) = \frac{K_p}{1 + K_d K_p F(s)} \tag{2.14}$$

and

$$T_c(s) = \frac{K_p F(s)}{1 + K_d K_p F(s)} \tag{2.15}$$

For input phase step to the DLL of magnitud $\Delta\phi_{in}$, the above two equations can be rewritten as

$$\begin{aligned} V_{PFD,\min} &\le L^{-1}\{T_e(s)/s\}\Delta\phi_{in} \le V_{PFD,\max} \\ V_{VCDL,\min} &\le L^{-1}\{T_c(s)/s\}\Delta\phi_{in} \le V_{VCDL,\max} \end{aligned} \tag{2.16}$$

As the minimum value of these two quantities establish the lower limit of the above two inequalities and the maximum values of these two quantities establish the upper limits, we can write the following constraint equation for the linear range of operation for the DLL to a step input as:

$$\begin{aligned} \max\left\{\frac{V_{PFD,\min}}{\min\left\{L^{-1}\{T_e(s)/s\}\right\}}, \frac{V_{VCDL,\min}}{\min\left\{L^{-1}\{T_c(s)/s\}\right\}}\right\} &\le \Delta\phi_{in} \\ \le \min\left\{\frac{V_{PFD,\max}}{\max\left\{L^{-1}\{T_e(s)/s\}\right\}}, \frac{V_{VCDL,\max}}{\max\left\{L^{-1}\{T_c(s)/s\}\right\}}\right\} & \end{aligned} \tag{2.17}$$

Any input step phase signal with magnitude that exceeds the above bounds will result in nonlinear DLL operation.

6) Settling time:

Another important characteristic of a DLL is the settling time (ST) which represents the time it takes for the DLL system to reach a steady-state operation when subject to an input step change in phase that lies within its linear region. A good DLL system has a faster response or, or in other words, a relatively small settling time. The DLL settling time can be defined as the time the control voltage to the delay line (an easily observable quantity) is required to change from its present steady-state value of $\upsilon_c(0^-)$ to its new steady-state value of $\upsilon_c(\infty)$ within a tolerance of $\varepsilon_{tolerance}$ when subject to an input phase step $\Delta\phi_{in}$. Mathematically, we can write the definition for settling time as

$$\frac{|\upsilon_c(t)-\upsilon_c(\infty)|}{|\upsilon_c(\infty)-\upsilon_c(0^-)|} < \varepsilon_{tolerance} \text{ for all } t > ST \tag{2.18}$$

where $\upsilon_c(t) = L^{-1}\{V_c(s)\} = L^{-1}\left\{T_c(s)\frac{\Delta\phi_i}{s}\right\}$.

2.2 DLL Design Method Based on Pole-Zero Positioning

Assuming the majority of the power contained in the spectrum of the input excess-phase signal ϕ_{in} is located in a frequency band much smaller than the reference clock frequency f_c, a continuous-time model of DLL operation can be used; otherwise, a discrete-time model would be required. Assuming an oversampled situation, according to Eqn. (2.11), the input-output excess-phase transfer function of DLL can be described as:

$$H(\mathrm{s}) = \frac{\phi_{\mathrm{out}}}{\phi_{\mathrm{in}}}(\mathrm{s}) = \frac{K_p K_d F(\mathrm{s})}{1 + K_p K_d F(\mathrm{s})} \tag{2.19}$$

where ϕ_{in} and ϕ_{out} are the input and output excess phase signals relative to the reference clock signal, K_p is the gain term in V/rad associated with the PFD and CP, K_d is the gain of the VCDL in rad/V and $F(s)$ is the loop filter transfer function.

The loop filter is the most critical component in DLL design, as the phase detector and tunable delay line are fairly standard across different DLL implementations. Eqn. (2.19) suggests that the order of the DLL is equal to the loop filter order and that the nature of the closed-loop response (i.e., lowpass, bandpass, etc.) is identical to that of the loop filter. A general Nth-order filter function can be expressed as:

$$F(\mathrm{s}) = \frac{a_N s^N + a_{N-1}s^{N-1} + \cdots + a_1 s + a_0}{s^N + b_{N-1}s^{N-1} + \cdots + b_1 s + b_0} \tag{2.20}$$

where a_i and b_i are real-value coefficients. As the filter function must have at least one pole at DC to eliminate any static phase error, b_0 must be set to zero. Substituting Eqn. (2.20) into (2.19), we can obtain:

$$H(\mathrm{s}) = \frac{\frac{K_p K_d}{(1+K_p K_d a_N)}(a_N s^N + a_{N-1}s^{N-1} + \cdots + a_1 s + a_0)}{s^N + \frac{(b_{N-1}+K_p K_d a_{N-1})}{(1+K_P K_d a_N)}s^{N-1} + \cdots + \frac{(b_1 + K_p K_d a_1)}{(1+K_p K_d a_N)}s + \frac{K_p K_d a_0}{(1+K_p K_d a_N)}} \tag{2.21}$$

Restricting the nature of the input-output excess-phase function to one of a low-pass nature,

i.e., $a_N = a_{N-1} = \cdots = a_1 = 0$ then Eqn. (2.21) reduces to

$$H(\mathrm{s}) = \frac{K_p K_d a_0}{s^N + b_{N-1}s^{N-1} + \cdots + b_1 s + K_p K_d a_0} \tag{2.22}$$

As the coefficient of the zero-order term of both the numerator and denominator are the same, the transfer function will have a DC gain of unity – a necessary condition for a DC pole in the loop filter.

In general terms, if the input-output DLL transfer function is described with denominator coefficients $\{d_i\}$ in the same form as Eqn. (2.22), one can write

$$H(\mathrm{s}) = \frac{d_0}{s^N + d_{N-1}s^{N-1} + \cdots + d_1 s + d_0} \tag{2.23}$$

where

$$d_0 = K_p K_d a_0, d_1 = b_1, d_2 = b_2, \cdots d_{N-1} = b_{N-1}.$$

Subsequently, the loop filter transfer function takes on the form

$$F(\mathrm{s}) = \frac{1}{K_p K_d}\left[\frac{H(s)}{1 - H(s)}\right] = \frac{1}{K_p K_d} F'(\mathrm{s}) = \frac{1}{K_p K_d}\left[\frac{d_0}{s^N + d_{N-1}s^{N-1} + \cdots + d_1 s}\right] \tag{2.24}$$

Here we shall denote the term $F'(s)$ as the unit-scaled loop filter transfer function.

For any kind of closed-loop transfer function at hand that has the same form as Eqn. (2.23), one can isolate each coefficient and solve for the set of filter coefficients $\{d_i\}$. In this work, we have found two classes of transfer functions that are well suited for DLL applications. One involves a class of transfer functions that have minimum settling time when subject to a unit step change, and a more recent class of filter transfer function has minimum input-output delay. The pros and cons of these two classes of transfer functions will be discussed in detail as it applies to DLLs.

2.2.1 Min. Step-Response Settling Time: Gaussian Type Transfer Functions

A class of transfer functions often selected for its desired step response is the Gaussian transfer function. This particular transfer function is guaranteed to exhibit no overshoot, have the lowest level of group delay and the most gradual pass-band roll-off of all known filter functions [16]. Another type of transfer function that has similar frequency response, phase and group delay, and step response with the Gaussian transfer function is the Bessel type

transfer function. According to [16], the Bessel transfer function has even slightly better Shaping Factor, flatter phase delay, and flatter group delay than that of a Gaussian filter of equal order. However, the Gaussian transfer function has less time delay and no overshoot in the step response. This time-domain characteristic is crucial to guarantee a stable and faster system. Therefore, the Gaussian type transfer function is selected here for DLL design.

For type-II DLLs, the Gaussian transfer function has proven to be a good choice; although it is not limited to this transfer function alone. The approach is general and can handle any transfer function that fits the overall form of Eqn. (2.21).

Table 2.1: Frequency-Normalized Gaussian Type Close-loop Transfer Function

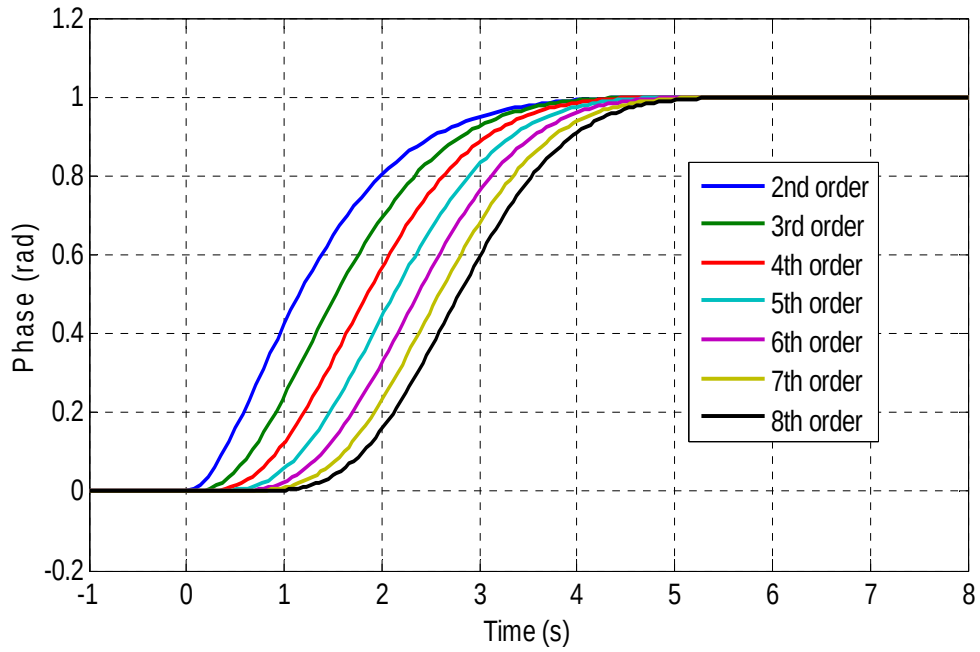
Order	Gaussian Transfer Function
2	$\frac{1.93910283}{s^2+2.57303540s+1.93910283}$
3	$\frac{4.21449636}{s^3+4.26771433s^2+6.95289646s+4.21449636}$
4	$\frac{10.24462511}{s^4+6.10030924s^3+15.71470706s^2+19.86351466s+10.24462511}$
5	$\frac{27.61318372}{s^5+8.08174854s^4+29.03864830s^3+57.08250436s^2+60.65666727s+27.61318372}$
6	$\frac{81.40292767}{s^6+10.20832234s^5+47.76202902s^4+129.43458500s^3+212.13879820s^2+197.77480330s+81.40292767}$
7	$\frac{259.13998620}{s^7+12.47193439s^6+72.70780990s^5+254.12630620s^4+570.24311460s^3+815.79720540s^2+684.98132590s+259.13998620}$
8	$\frac{881.88580970}{s^8+14.86466712s^7+104.68857160s^6+452.21439580s^5+1300.84713s^4+2536.275415s^3+3256.310754s^2+2506.116569s+881.88580970}$

Table 2.1 shows the 2nd to 8th order frequency-normalized Gaussian type transfer functions and Table 2.2 lists the unit-scale ($K_pK_d=1$) and frequency-normalized (3-dB frequency of 1 rad/s) loop filter transfer functions $F'(s)$ corresponding to these transfer functions. These were all obtained through application of Eqn. (2.24). DLL requirements would make use of the loop filter transfer functions of Table 2.2 through the appropriate frequency de-normalization and coefficient scaling procedure [28].

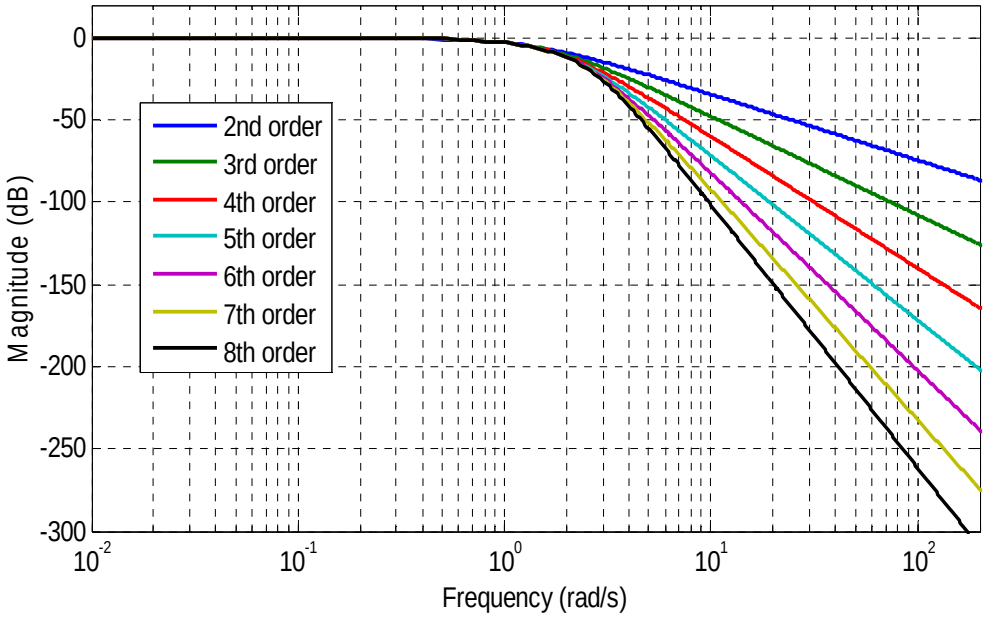
Table 2.2: Unit-Scaled and Frequency-Normalized Loop Filter Transfer Functions For Closed-Loop Gaussian Behavior

Order	Transfer Function
2	$F'(s) = \frac{1.9391}{s^2+2.5730s}$
3	$F'(s) = \frac{4.2144}{s^3+4.2677s^2+6.9529s}$
4	$F'(s) = \frac{10.2446}{s^4+6.1003s^3+15.7147s^2+19.8635s}$
5	$F'(s) = \frac{27.6132}{s^5+8.0817s^4+29.0386s^3+57.0825s^2+60.6567s}$
6	$F'(s) = \frac{81.4029}{s^6+10.2083s^5+47.7620s^4+129.4346s^3+212.1388s^2+197.7748s}$
7	$F'(s) = \frac{259.1400}{s^7+12.4719s^6+72.7078s^5+254.1263s^4+570.2431s^3+815.79720540s^2+684.9813s}$
8	$F'(s) = \frac{881.8858}{s^8+14.8647s^7+104.6886s^6+452.2144s^5+1300.8471s^4+2536.2754s^3+3256.3108s^2+2506.1166s}$

Under unit-scaled and frequency-normalized conditions, both the magnitude response versus frequency and the corresponding step response to a unit radian change in phase are illustrated in Fig. 2.3 for the input-output behavior of the DLL for orders ranging from 2 to 8. The step response has a simple exponential-like behavior with no overshoot; true for all cases shown. The corresponding rise-time also increases with DLL order (not easy to see from the figure, though) even though the propagation delay increases with increasing order. In the case of the frequency response, one observes a smooth roll-off at the 3-dB frequency of 1 rad/s with no resonance having a roll-off rate increasing with DLL order, i.e., -20N db/decade.



(a)



(b)

Fig. 2.3: Unit-scaled and frequency-normalized input-output excess-phase response for a type-II DLL of Gaussian behavior: (a) step response, and (b) magnitude vs. frequency.

2.2.2 Minimun Propagation-Delay Transfer Functions

Another type of transfer function aimed at reducing the rise-time and improving the transient response is the Min-Delay transfer function [29]. By extending the class of Bessel type filter functions with a complex set of zeros enables the realization of a class of filter functions with the fastest rise-time to a step input. Table 2.3 shows the 1st to 5th order

Min-Delay transfer functions.

Table 2.3: Frequency-Normalized Min-Delay Transfer Functions

Order	Min-Delay Transfer Function
1	$\frac{1.00237729}{1.00000000s+1.00237729}$
2	$\frac{0.78750289s+0.62016081}{1.00000000s^2+1.36399502s+0.62016081}$
3	$\frac{0.72117497s^2+0.94354184s+0.38281268}{1.00000000s^3+1.76651069s^2+1.30023334s+0.3828126}$
4	$\frac{0.68924918s^3+1.22375537s^2+0.86536252s+0.23697052}{1.00000000s^4+2.17959727s^3+2.13778993s^2+1.08722159s+0.23697052}$
5	$\frac{0.67066395s^4+1.49148923s^3+1.44730048s^2+0.71142937s+0.14713922}{1.00000000s^5+2.59747029s^4+3.14853090s^3+2.18085746s^2+0.84970687s+0.14713922}$

The transfer functions in Table 2.3 have the following general form:

$$H(s)=\frac{c_{N-1}s^{N-1}+\cdots+c_1s+c_0}{s^N+d_{N-1}s^{N-1}+\cdots+d_1s} \tag{2.25}$$

Comparing Eqn. (2.25) with Eqn. (2.21), one finds the following coefficient relationships

$$\begin{aligned}\alpha\cdot a_0&=c_0\\ \alpha\cdot a_1&=c_1\\ &\vdots\\ \alpha\cdot a_{N-1}&=c_{N-1}\end{aligned} \tag{2.26}$$

and

$$\begin{aligned}b_1+\alpha\cdot a_1&=d_1\\ b_2+\alpha\cdot a_2&=d_2\\ &\vdots\\ b_{N-1}+\alpha\cdot a_{N-1}&=d_{N-1}\end{aligned} \tag{2.27}$$

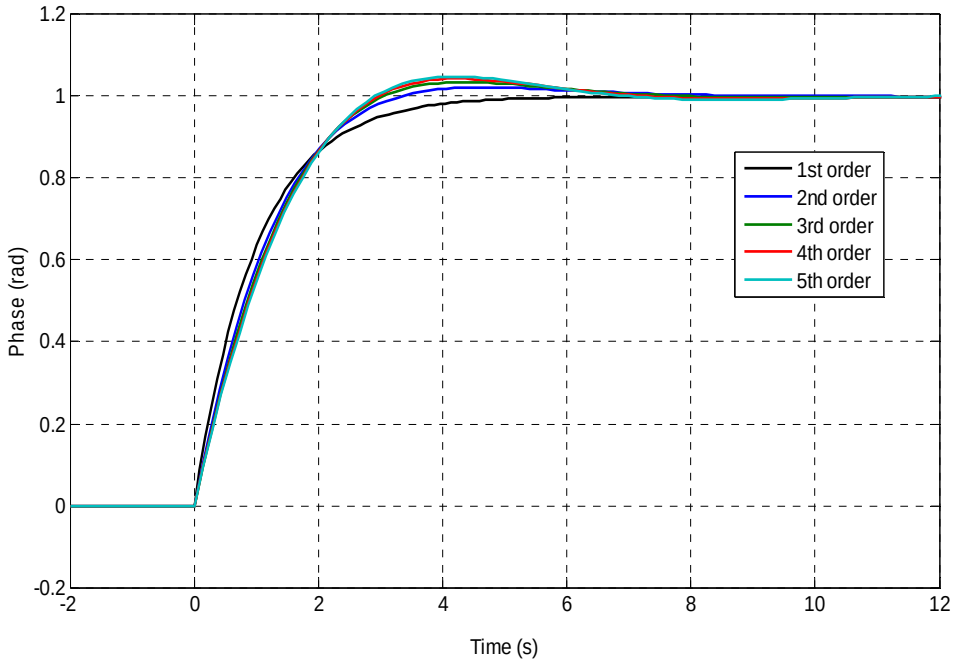
where $\alpha=K_pK_d$ is seen as a constant coefficient and N is the order of the transfer function. Subsequently, the loop filter transfer function reduces to

$$F(\mathrm{s})=\frac{a_{N-1}s^{N-1}+\cdots+a_1s+a_0}{s^N+b_{N-1}s^{N-1}+\cdots+b_1s} \tag{2.28}$$

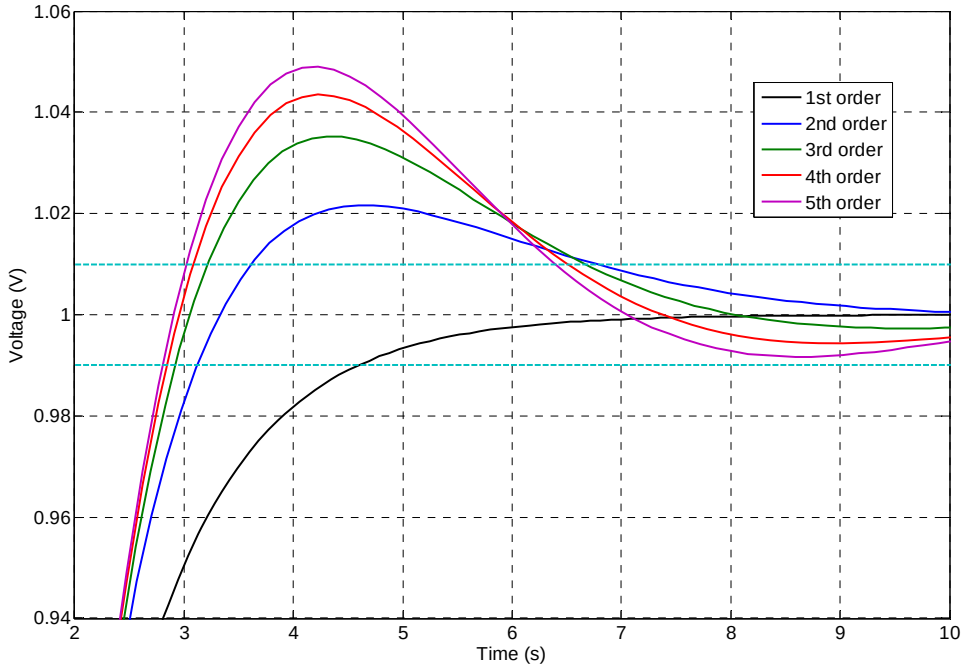
Here it is seen that the loop filter transfer function will consist of $N-1$ zeros and N poles. As the denominator polynomial has a pole at DC, the DLL will have a steady-state error that goes to zero. Similarly, Table 2.4 lists the unit-scale ($K_p K_d = 1$) and frequency-normalized (3-dB frequency of 1 rad/s) loop filter transfer functions $F'(s)$ with Min-Delay for orders ranging from 1st to 5th order. Figure 2.4 depicts both the step response and the corresponding frequency response. Here one can see that the 1st order DLL has the fastest settling times, as there is no overshoot in its step response. While the 2nd to 5th order DLLs have much longer settling times. The frequency response behavior reveals some interesting facts. They all seem to have about the same 3-dB break point of 1 rad/s, no resonance and equal frequency roll-off of -20 dB/dec.

Table 2.4: Unit-Scaled and Frequency-Normalized Loop Filter Transfer Functions for Closed-Loop Min-Delay Behavior

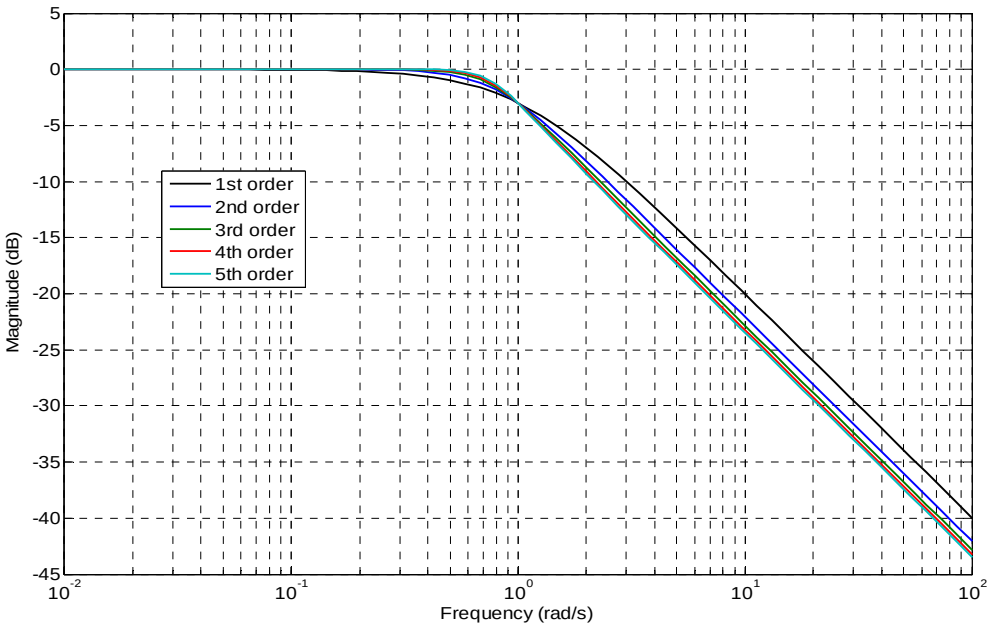
Order	Transfer Function
1	$F'(s) = \frac{1.002}{s}$
2	$F'(s) = \frac{0.7875\,s + 0.6202}{s^2 + 0.5765\,s}$
3	$F'(s) = \frac{0.7212\,s^2 + 0.9435\,s + 0.3828}{s^3 + 1.045 s^2 + 0.3567 s}$
4	$F'(s) = \frac{0.6892\,s^3 + 1.224\,s^2 + 0.8654\,s + 0.237}{s^4 + 1.49 s^3 + 0.914 s^2 + 0.2219 s}$
5	$F'(s) = \frac{0.6707\,s^4 + 1.491\,s^3 + 1.447\,s^2 + 0.7114\,s + 0.1471}{s^5 + 1.927 s^4 + 1.657 s^3 + 0.7336 s^2 + 0.1383 s}$


1.2
1
0.8
0.6
0.4
0.2
0
-0.2
-2
0
2
4
6
8
10
12
Phase (rad)
Time (s)
1st order
2nd order
3rd order
4th order
5th order

(a)


1.06
1.04
1.02
1
0.98
0.96
0.94
2
3
4
5
6
7
8
9
10
Voltage (V)
Time (s)
1st order
2nd order
3rd order
4th order
5th order

(b)



(c)

Fig. 2.4: Unit-scaled and frequency-normalized input-output excess-phase response for a type-II DLL of Min-Delay behavior: (a) step response, (b) zoomed in-region illustrating settling time, and (c) magnitude vs. frequency.

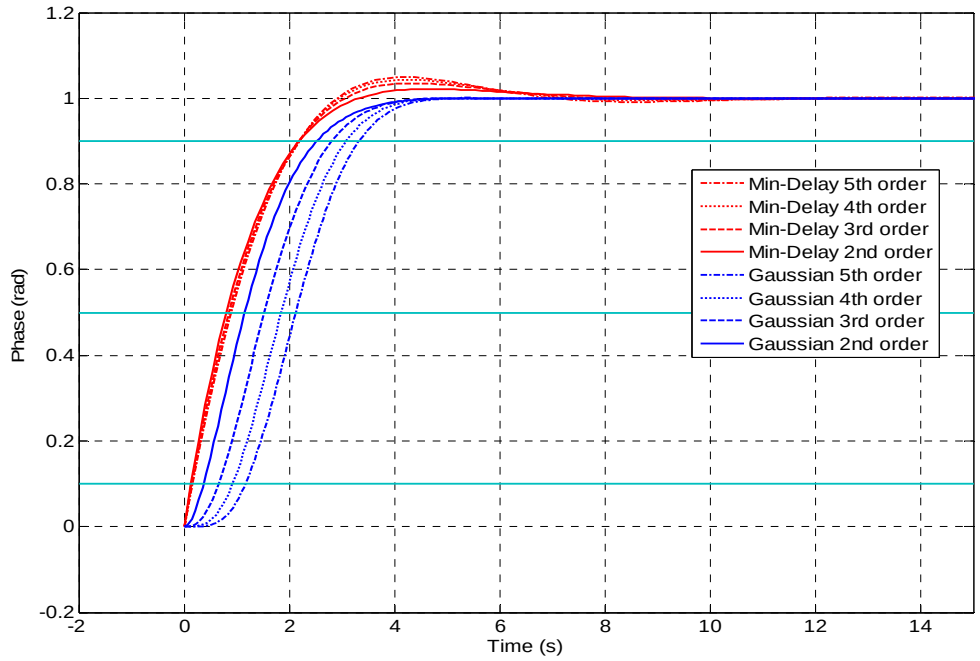
2.2.3 Comparison of Gaussian Type and Min-delay Type DLLs

To make a closer comparison of the Gaussian type and Min-delay type DLLs, we draw their transient responses and frequency responses in the same plots in Fig. 2.5 (a) and (b), respectively. In these plots, the blue lines represent the Gaussian type DLL behaviors and red lines represent the Min-delay type DLL behaviors.

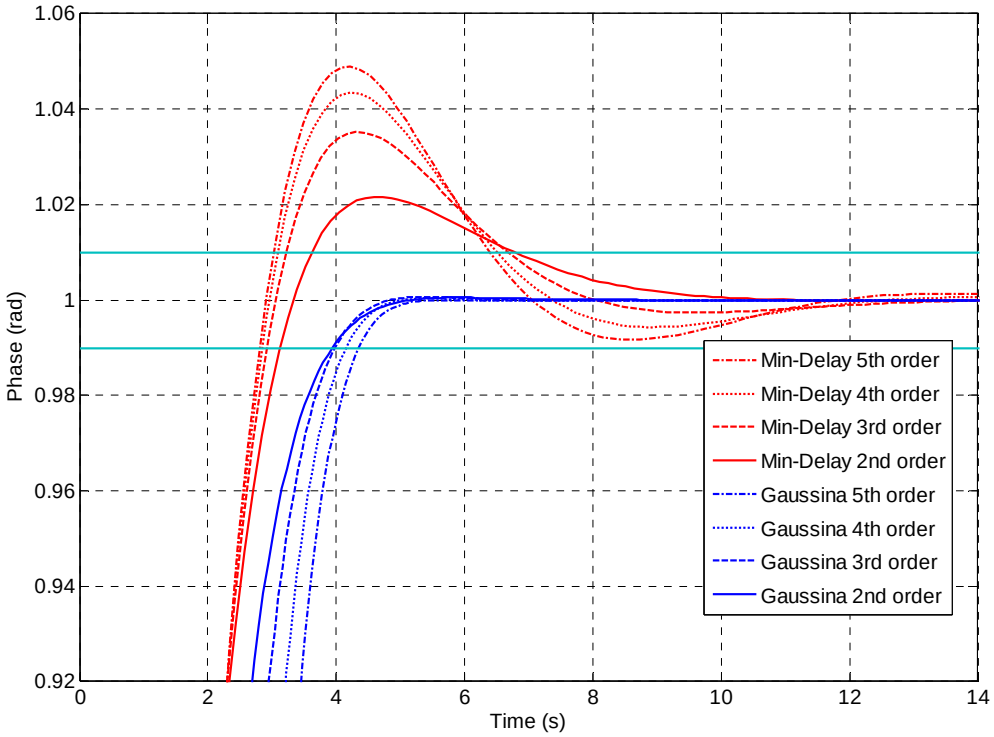
The comparison criteria including the delay time Dt (the time need to reach 50% of the final value), the rise time Rt (the time need from 10% to 90% of the final value), the settling time St (the time need when errors are within 1% of the final value) and value of overshoot (shouldn't be larger than 10% for a good system) are used to measure the characteristics of the DLLs under transient response. Table 2.5 lists the detailed numbers of the two types of DLLs. We can see that the Min-delay type DLLs have better rise time compared to the Gaussian type DLLs especially for the high orders. While the Gaussian type DLLs have faster settling time since there are almost no overshoot in their step responses.

As for the frequency response, when DLLs compared under the same -3dB bandwidth, the Min-delay type DLLs have smaller slopes of roll-off because of the extra zeros added in the

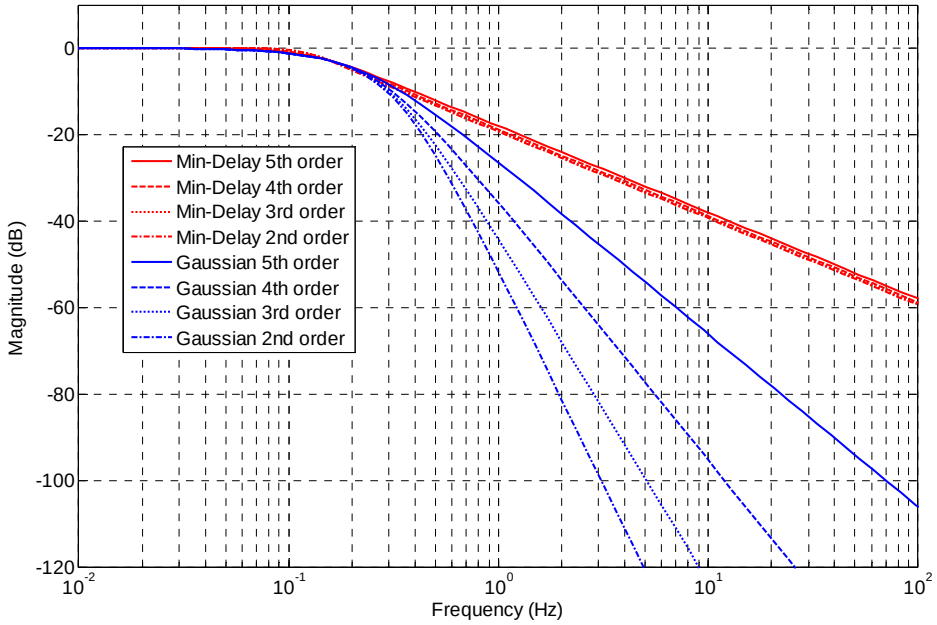
numerator of the transfer functions. Besides, the slopes are similar between different orders of the Min-delay type DLLs. Therefore, if we want the DLL to have a fast settling time and at the same time a steep roll-off in frequency response, the Gaussian type DLL is a better choice.



(a)



(b)



(c)

Fig. 2.5: Comparison of the input-output excess-phase responses of Gaussian type and Min-delay type of DLLs from 2nd to 5th order: (a) step response, (b) zoomed in-region illustrating settling time , and (c) magnitude vs. frequency.

Table 2.5: Step Responses for Gaussian Type and Min-delay Type of DLLs

	Delay Time (Dt)	Rise Time (Rt)	Settling Time (St)	Overshoot
2nd Gaussian TF	1.159	2.1496	5.233	0%
2nd Min-Delay	0.806	2.0618	6.775	2.2%
3rd Gaussian TF	1.535	2.1546	5.233	0%
3rd Min-Delay	0.854	2.0394	6.671	3.5%
4th Gaussian TF	1.854	2.1608	5.233	0%
4th Min-Delay	0.880	2.0313	6.509	4.3%
5th Gaussian TF	2.123	2.1530	5.233	0%
5th Min-Delay	0.896	2.0263	6.403	4.9%

2.3 MATLAB Simulations to Verify Proposed System Theory

To verify the proposed system theory of the previous section, we now build up MATLAB/Simulink models for the DLL with both the Gaussian type and Min-Delay type behaviors. Assuming that the DLLs have a 3-dB bandwidth of 100 kHz, a reference clock frequency of $f_c = 50$ MHz, a PFD/CP coefficient of $K_p = 8.05 \times 10^{-5}$ V/rad and a VCDL coefficient of $K_d = 7.6567$ rad/V. These parameters are chosen according to the CMOS IC

design that one will encounter in later chapters of this thesis.

2.3.1 Gaussian Type DLL Simulations

The first step is to de-normalize the loop filter transfer functions in Table 2.2 and to account for the gains of the PFD and VCDL, i.e. perform the following transformation,

$$F(\mathrm{s}) = \frac{1}{K_P K_D} F'(s)\Big|_{s \to \frac{s}{\omega_p}} \tag{2.29}$$

For example, for a 5th-order DLL, together with the desired 3-dB bandwidth of $\omega_p = 100\pi$ rad/s , one finds the loop filter transfer function as follows

$$\begin{aligned} F(s) &= \frac{1}{8.05\times10^{-5}\times 7.66}\,\frac{27.6132}{s^5+8.0817s^4+29.0386s^3+57.0825s^2+60.6567s}\Big|_{s\to\frac{s}{100\pi\times10^3}} \\ &= \frac{1.354\times10^{32}}{s^5+2.539\times10^6 s^4+2.866\times10^{12}s^3+1.77\times10^{18}s^2+5.909\times10^{23}s} \end{aligned} \tag{2.30}$$

A dynamic MATLAB/Simulink model shown in Fig. 2.6 was created to simulate the transient and frequency response behavior of a DLL with orders ranging from 2 to 8. The input can be switched between a pulse-delayed signal to create a step change in phase or a multi-tone sinusoidal excitation to excise the AC response of the DLL. Two time-measuring units (TMU) are used to extract both the input and output excess phase signals. The detailed Simulink models are provided in Appendix A.

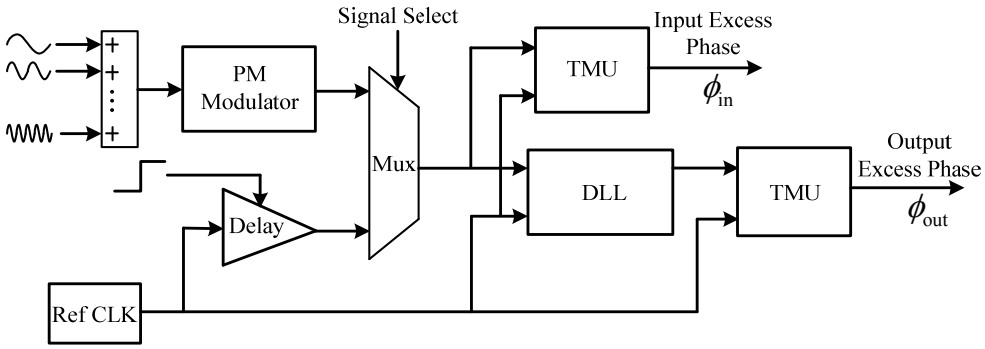
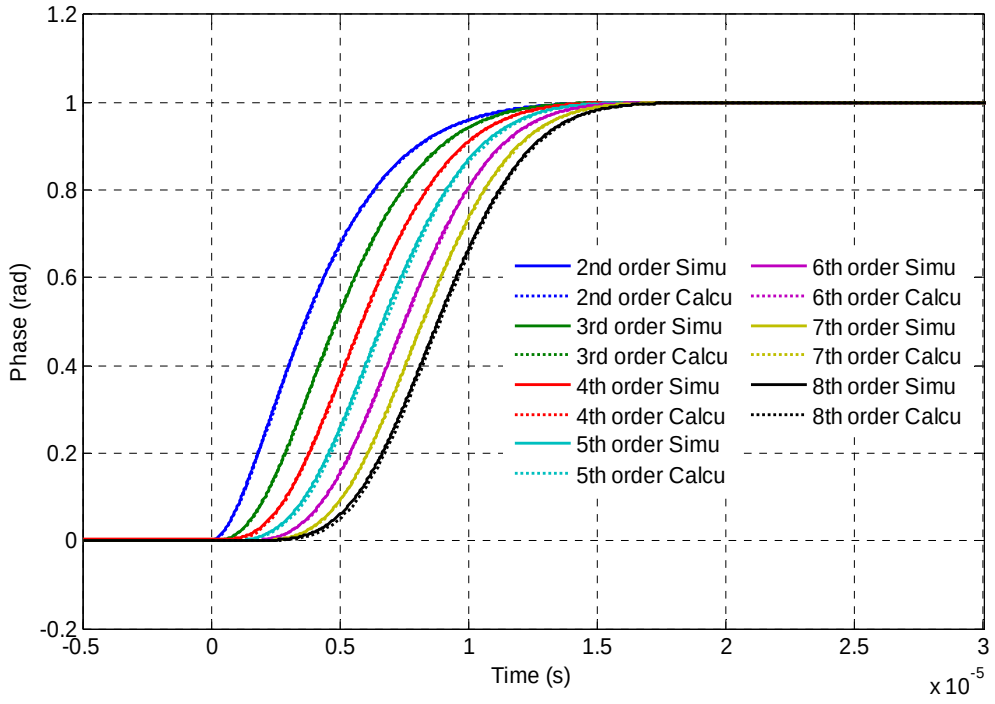


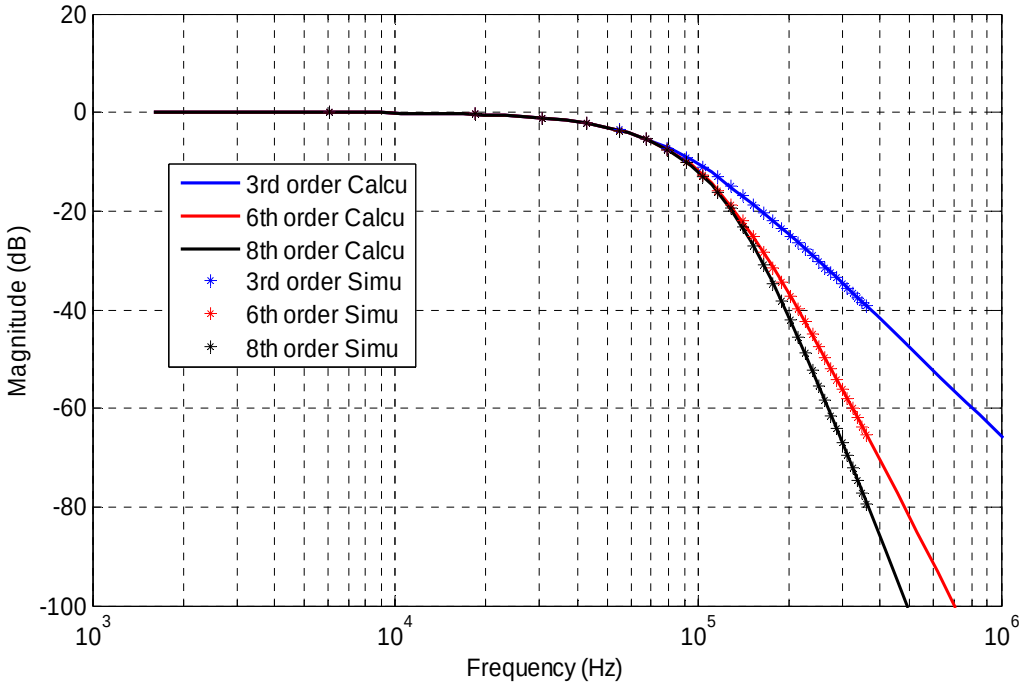
Fig. 2.6: Transient and AC response test bench for DLL analysis.

Figure 2.7(a) illustrates the output step response of the Gaussian type DLLs subject to a 1-radian phase step change. In Fig. 2.7(b), the AC response of the DLL subject to a multi-tone sinusoidal input signal consisting of 30 sine waves is shown. Here the input and

output excess phase of the DLL for orders of 3, 6 and 8 was sampled and then analyzed using an FFT, then plotted. Each result is compared to the theoretical predicted result and, in each case, they all show excellent correlation, i.e., no visible error. This proves the correctness of the system-level design method for the Gaussian type DLLs.



(a)



(b)

Fig. 2.7: DLL step and frequency response behavior as compared by Simulink and theory: (a) AC response, and (b) step-response.

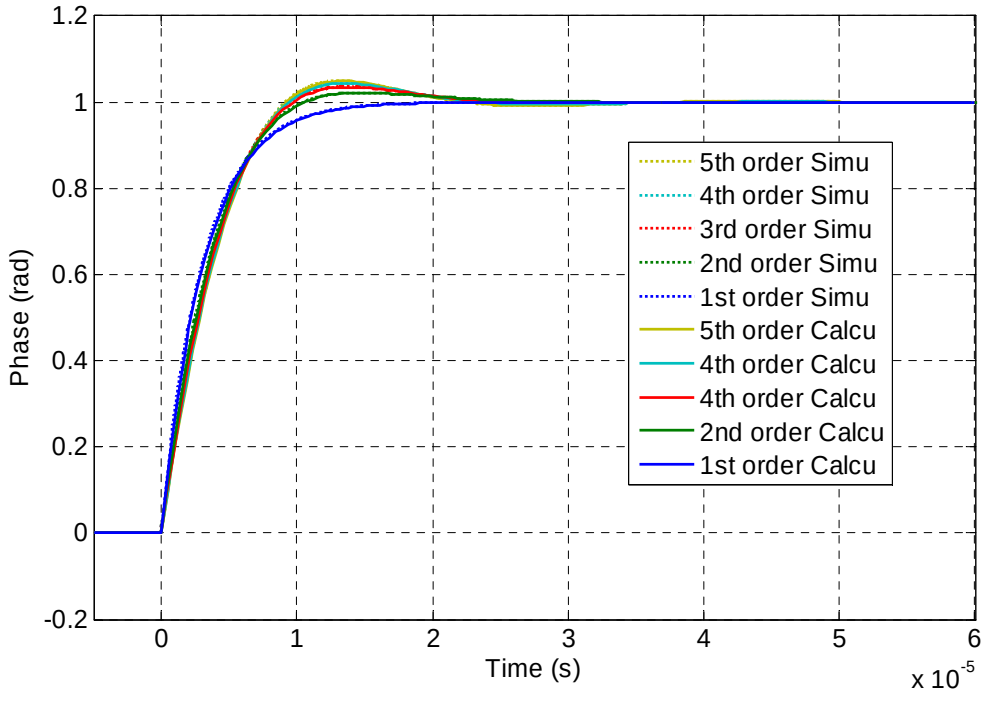
2.3.2 Min-Delay Type DLLs Simulations

Similarly to what was just performed for the class of Gaussian filter functions, Eqn. (2.29) will again be used to frequency de-normalize and scale the loop filter transfer functions found in Table 2.4 according to the gains of the PFD and VCDL. For example, for a 3rd-order Min-Delay type DLL with a desired 3-dB bandwidth of $\omega_p = 100\pi$ rad/s, the de-normalized loop filter transfer function becomes

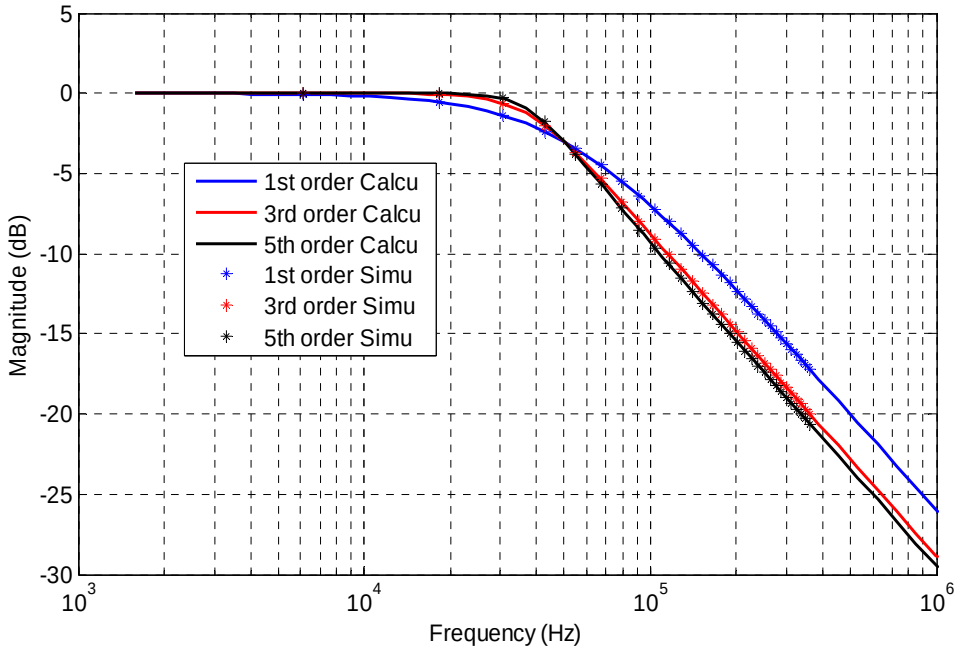
$$F(s) = \frac{1}{8.05\times10^{-5}\times7.66}\left.\frac{0.7212\,s^2 + 0.9435\,s + 0.3828}{s^3+1.045s^2+0.3567s}\right|_{s\to\frac{s}{100\pi\times10^3}} \qquad (2.31)$$

$$= \frac{3.631\times10^{8}s^2 + 1.492\times10^{14}s + 1.902\times10^{19}}{s^3 + 3.284\times10^{5}s^2 + 3.52\times10^{10}s}$$

The MATLAB/Simulink test-bench shown in Fig. 2.6 is used again to obtain the transient and frequency response behaviors of the Min-Delay type DLLs. Figure 2.8(a) illustrates the output step response of the DLL subject to a 1-radian phase change and Fig. 2.8(b) is the AC response of the DLL subject to a multi-tone sinusoidal input signal consisting of 30 sine waves. It is obvious that the simulation results match the theoretical predicted results perfectly. This proves the correctness of the system-level design method for the Min-Delay type DLLs.



(a)



(b)

Fig. 2.8: DLL step and frequency response behavior as computed by Simulink and theory: (a) step-response, and (b) AC response.

2.4 Summary

This chapter presents the system-level DLL design. The DLL transfer functions and system characteristics were described. The DLL design method based on the pole-zero positioning approach was introduced and verified through MATLAB/Simulink simulations. Two types of DLLs, i.e. the DLLs with Gaussian behavior and those with Min-Delay behavior, are designed based on the selection of closed-loop transfer functions. Their simulation results are discussed and compared. The Gaussian transfer function is found to be more suitable in high-order DLL design when it is used as a time-mode filter (TMF).

Chapter 3

DLL Building Blocks

This chapter will describe the circuit-level realization of the DLL main building blocks. The circuit topology of each building component will be given as well as the reasons why the proposed topologies are chosen among other prevalent topologies. SPICE simulations are used in this chapter to verify the correctness of these circuit designs.

3.1 PFD Design

The phase frequency detector (PFD) is used to detect the phase and frequency difference between input signal and the feedback signal. The basic schematic of a PFD is shown in Figure 3.1. It is composed of two D-flip flops and an AND-gate, and has two output terminals UP and DN. At any time, the two outputs of the PFD, UP and DN, will be in one of four possible states: 00, 01, 10, 11, with UP = high, DN = low represented by state 10, etc.. The PFD then drives a charge pump circuit, which produces a voltage signal that can drive the following loop filter circuit. When InA is high and InB is low, UP will be high and DN will be low. When InB is high and InA is low, DN will be high and UP will be low. When InA and InB are both high, the AND-gate will set up the Reset signal, pushing UP and DN to both low. This procedure is illustrated by the timing diagram shown in Fig. 3.2(a). The detection range for this type of PFD is $(2\pi, -2\pi)$, as illustrated by its transfer characteristic shown in Fig. 3.2(b).

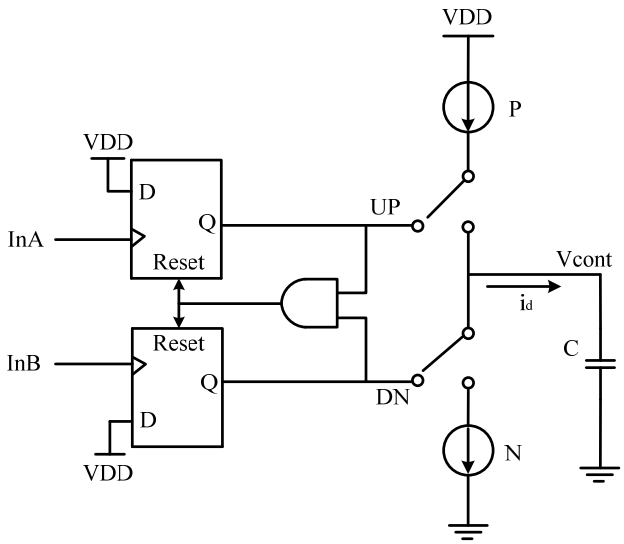
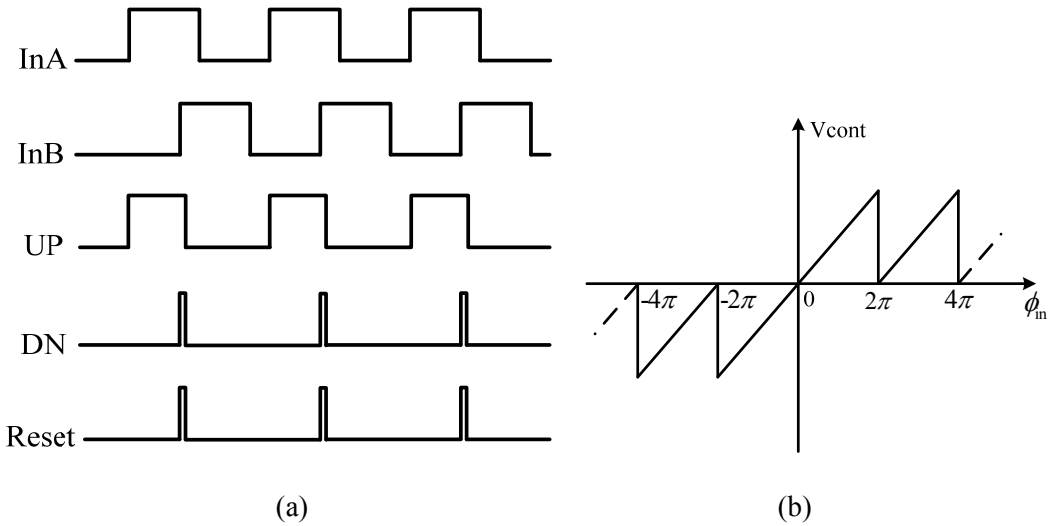


Fig. 3.1: Schematic of PFD circuit and differencing circuit.

Fig. 3.2(b) depicts the ideal transfer characteristic behavior of the PFD. However, there are other non-ideal characteristics that critically influence the performance of PFD, e.g., the dead-zone and blind-zone regions, as shown in Fig. 3.3. The dead-zone is generated when the phase difference between the input signals is small and a positive transition at UP or DN is closely followed by a reset operation. If the propagation delay at UP and DN is large enough, it is possible that UP or DN may be pulled to low by the reset operation before it completes a positive transition. Consequently, the charge pump output voltage will stay intact at a small phase difference and makes the DLL operate as if the loop is open.



(a) (b)

Fig. 3.2: Operation procedure of PFD. (a) Operation wave forms. (b) Output characteristic

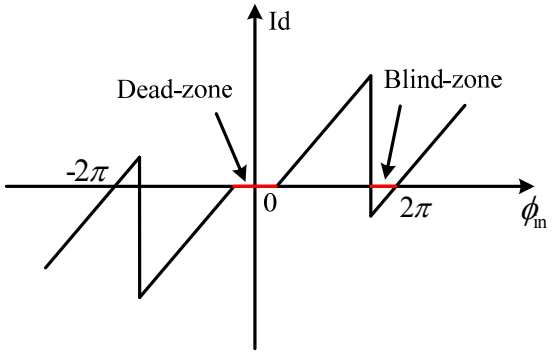


Fig. 3.3: Non-ideal PFD output characteristic showing dead-zone and blind-zone.

For most of the input phase difference, the PFD output increase monotonically with the increasing input signal when it suddenly changes to the opposite sign when the phase difference approaches $\pm 2\pi$ rad, as shown in Fig. 3.3. This region of the transfer characteristics is denoted as the blind-zone. The cause of the blind zone is explained in Fig. 3.4. The occurrence of a rising edge at InA within the dotted area coincides with a reset operation and, therefore, does not have any impact on the PFD output. Consequently, InB signal is incorrectly considered to be before InA signal in the subsequent cycles and therefore producing a false locking problem. It is obvious that the blind-zone is detrimental to the DLL settling behavior; it may increase the DLL settling time or even prevent the DLL from locking correctly. Having a blind-zone is equivalent to having a reduced phase-detection range. According to Eqn. (2.17), the linear operating range of DLL will also reduce consequently.

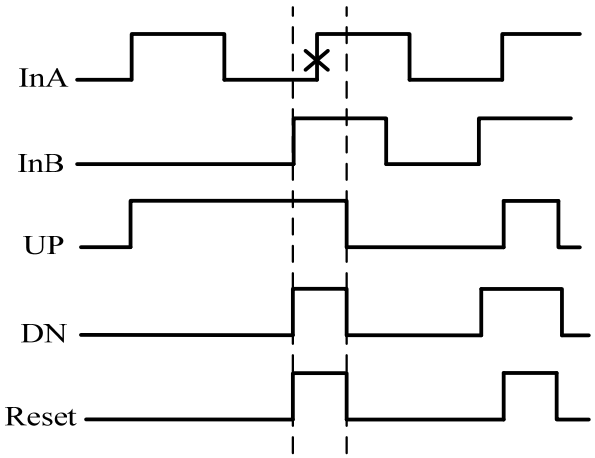


Fig. 3.4: Timing analyses of blind-zone.

PFD circuits are typical constructed using two types of logic techniques, i.e., static-logic or dynamic-logic circuits. A static PFD is based on static logic gates such as NAND-gates, NOR-gates and Inverters to realized the D-flip flops and the AND gate as in Fig. 3.1. The advantages of a static-logic PFD are that it has high stability and it is easier to design with. The disadvantage is that it has large propagation delay because of the large number of transistors used in the circuit, which reduce its performance greatly during high frequency operations. Although dynamic logic PFDs are harder to design and are less stable if the parameters are carefully chosen, dynamic logic circuit has high performance due to their faster operating speed and potential savings in power. To construct a PFD capable of working at gigahertz frequencies with small dead- and blind-zone regions, we make use of the dynamic-logic PFD first introduced in [23], combined with some further improvements to the start-up operation made by the author. The schematic of this PFD is shown in Fig. 3.5 and with the corresponding transistor aspect ratios listed in Table 3.1.

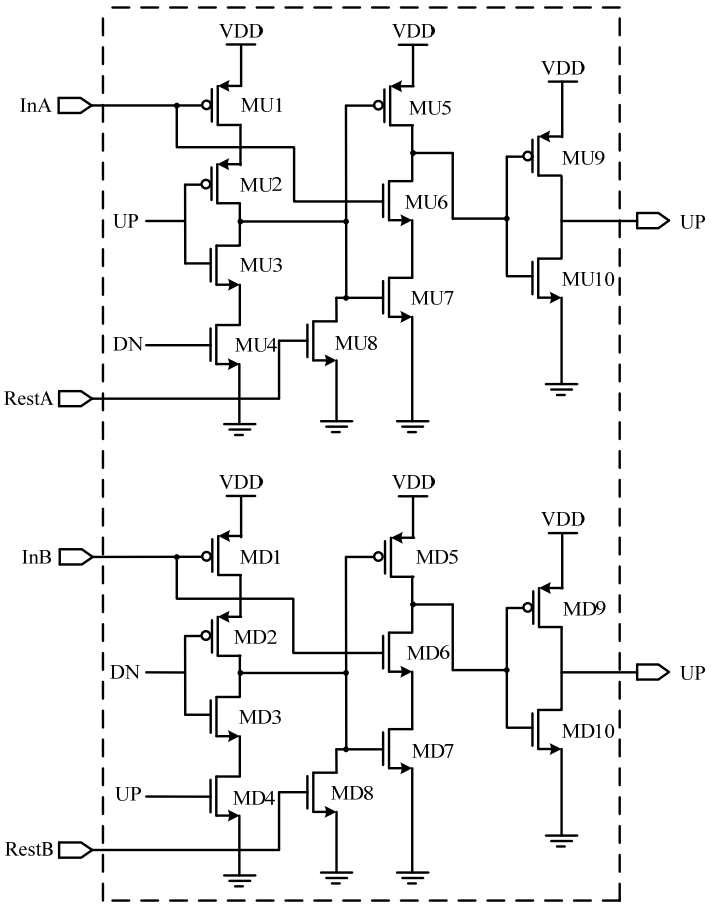


Fig. 3.5: Schematic of proposed PFD.

Table 3.1: Parameters of PFD

Transistor	Width	Length	Transistor	Width	Length
MU1	6u	300n	MD1	6u	300n
MU2	6u	300n	MD2	6u	300n
MU3	3u	300n	MD3	3u	300n
MU4	3u	300n	MD4	3u	300n
MU5	6.4u	300n	MD5	6.4u	300n
MU6	4u	300n	MD6	4u	300n
MU7	4u	300n	MD7	4u	300n
MU8	2u	300n	MD8	2u	300n
MU9	15u	300n	MD9	15u	300n
MU10	5u	300n	MD10	5u	300n

In this topology, transistors MU1, 2, 3, 4 and MD1, 2, 3, 4 form the pre-charge stage within which MU2-3 and MD2-3 are two inverter structures that prevent the VDD-VSS supply from being short-circuited. Transistors MU5, 6, 7 and MD5, 6, 7 form the evaluation stage. Transistors MU9, 10 and MD9, 10 are two inverter structures that act as output buffers, and transistors MU8 and MD8 provide the reset function. This PFD will be used together with a start-up circuit to make sure that the DLL could start up correctly. At the initial state when InA and InB are both low and UP and DN are also low, nodes U1 and D1 are pre-charged to high through transistor MU1, 2 and MD1, 2 respectively. At the rising edge of InA, MU6, 7 are turned on, node U2 is pulled to low, which drives UP to high. Similarly, a rising edge of InB will drive DN to high. When UP and DN are both high, they will turn on MU3, 4 and MD3, 4 and nodes U1 and D1 are discharged accordingly [23]. This resets the circuit to its initial state.

Figure 3.6 shows the output characteristic curves of this dynamic logic PFD with respect to different input frequencies. These characteristic curves can be compared with those of a static-logic PFD with similar transistor sizes as shown in Fig. 3.7. Here one can see that the dynamic-logic PFD (Fig. 3.6) has no dead-zone and its linear operating range decreases slowly with the increasing input frequency. In comparison with the static-logic PFD (Fig. 3.7), its linear operating range is much larger and its blind zone is much less. For instance, at an input frequency of 667 MHz, the dynamic logic PFD has a linear range of $\pm 1.2\pi$ rad and a blind zone of x rad, while the static logic PFD has a linear operating range of $\pm 0.2\pi$ rad and a blind zone of y rad.

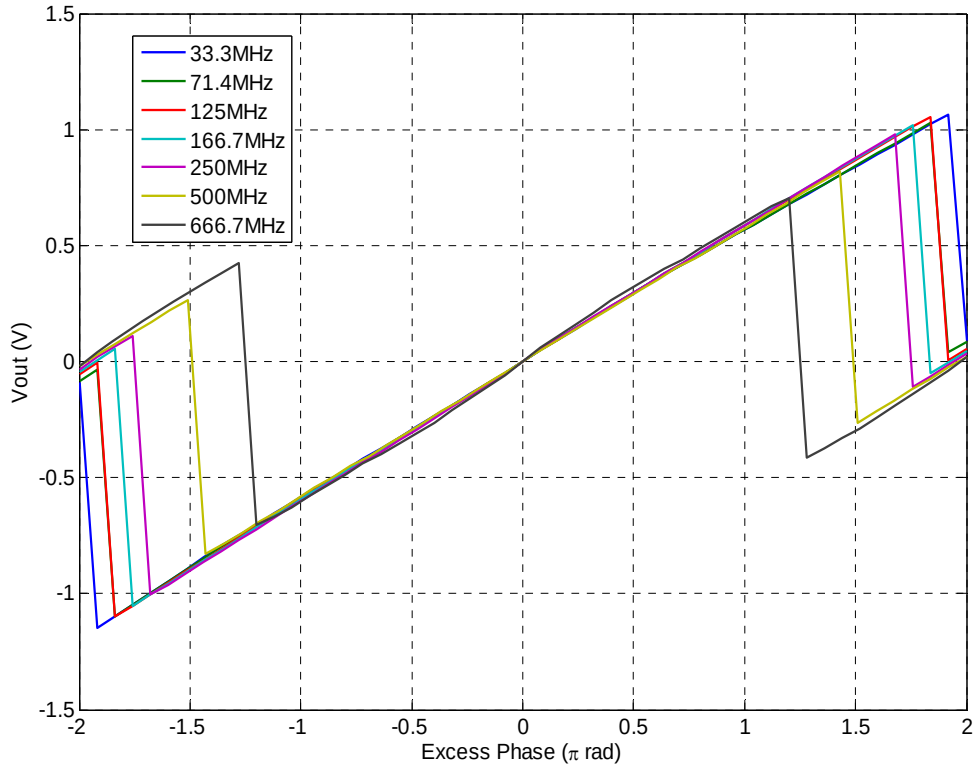


Fig. 3.6: Output characteristics of the dynamic-logic PFD.

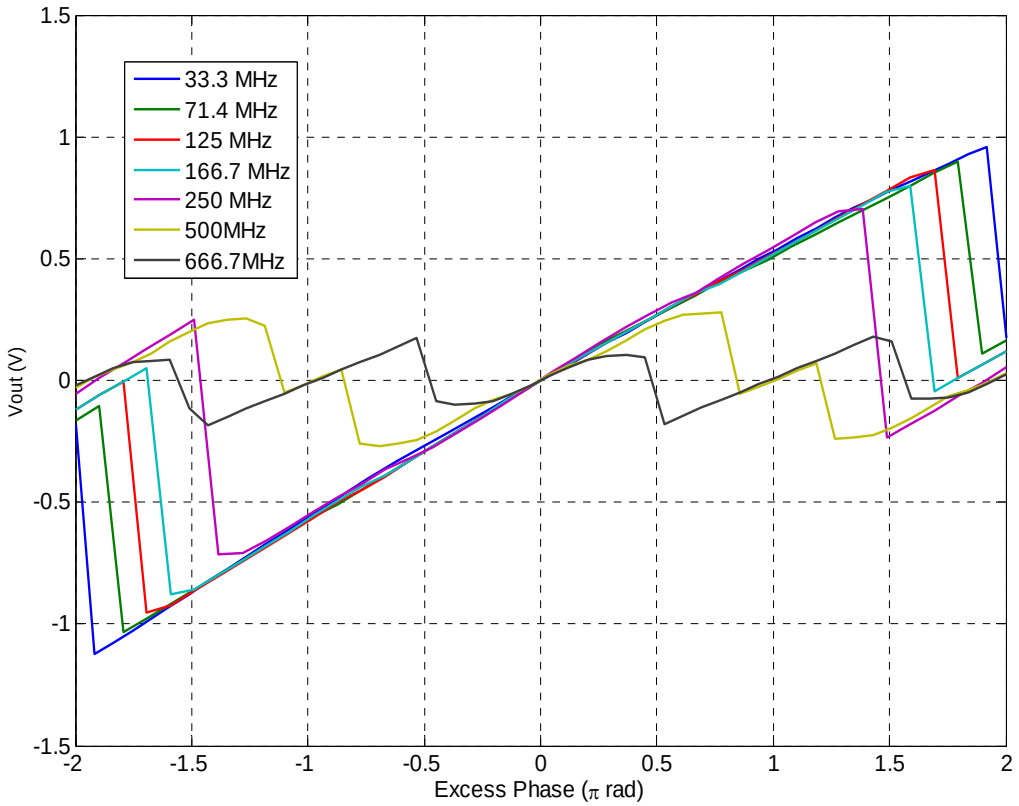


Fig. 3.7: Output characteristics of the static-logic PFD.

3.2 Charge Pump Design

The charge pump is used to translate the UP and DN pulses of PFD output into a single-ended current for the input of the loop filter. A simple illustration of the charge pump was shown previously in Fig. 3.1. The charge pump is composed of a P current source and an N current source. An active signal at UP (UP = 1) will turn on the P current source, allowing a positive current to flow into the output node, which makes the output voltage to rise. Similarly, an active signal at DN (DN = 1) will turn on the N current source, which pulls current from the output node, causing the voltage at the output node to reduce. When both UP and DN are low (UP = 0, DN = 0), both current sources are turned off from the output node, leaving the voltage at the output node unchanged.

Depending on the driving conditions, there are three basic types of single-ended charge pumps; this includes a switch in drain, a switch in gate or a switch in source. The design described in this thesis is based on the switch being placed in source leg of the current source transistors due to its simple structure, low power consumption and good switching time [30]. The schematic of the charge pump used here is shown in Fig. 3.8. This design was first introduced in [31]. It has a switch in source leg of each current source, M11 and M14, with two extra feedback transistors, M5 and M9, used to compensate for the channel-length modulation effect of the up or down current mirrors via negative feedback. Transistors M1-M17 form current mirrors to supply the reference current. With the dynamic adjustment according to the charge pump output voltage, there will be less current mismatch in the up and down branches. To further reduce the mismatch, the voltage at output Vo is fixed to VDD/2, which will be realized with external feedback provided by the active loop filter. This will be further discussed in the next subsection. As for the transistor sizes, long transistors are used to reduce channel length modulation effect. The transistor sizes are listed in Table 3.2. The gain of the PDF together with the differencing circuit is found through simulation to be $K_p = 0.506\text{m}/(2\pi) = 8.05\times10^{-5}$ V/rads .

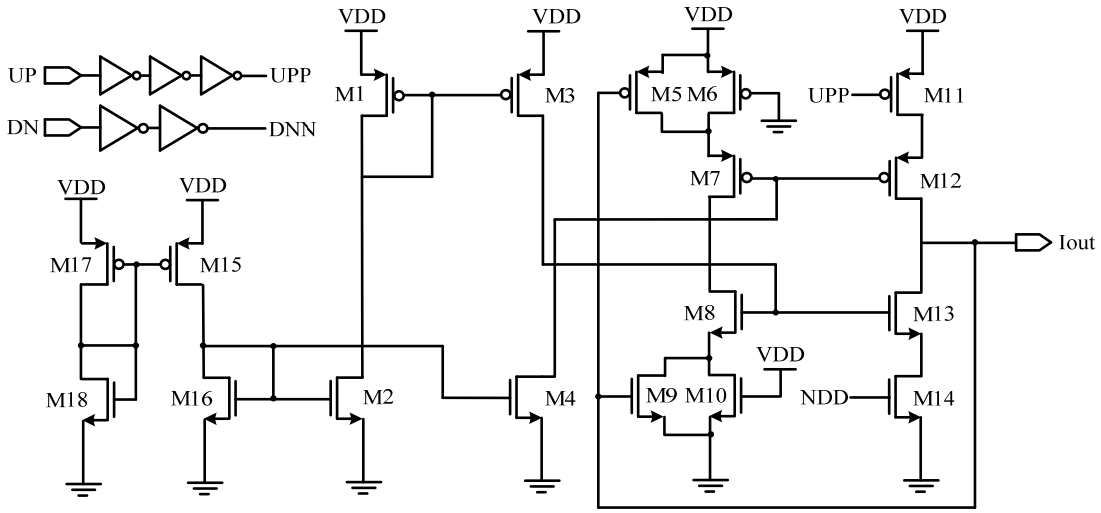


Fig. 3.8: Schematic of charge pump. [31]

Table 3.2: Parameters of PFD

Transistor	Width	Length	Transistor	Width	Length
M1	15u	300n	M10	6u	300n
M2	6u	300n	M11	15u	300n
M3	15u	300n	M12	24u	300n
M4	6u	300n	M13	8u	300n
M5	3u	300n	M14	6u	300n
M6	15u	300n	M15	20u	300n
M7	24u	300n	M16	6u	300n
M8	8u	300n	M17	10.7u	300n
M9	2u	300n	M18	6u	300n

3.3 Loop Filter Design

The design of the high-order active-RC loop filters will be described in this subsection. While there are numerous ways in which to construct a loop-filter, in this work the method of a cascade of biquads will be used to realize the Gaussian type transfer functions and the state-space approach (on account of the need for many zeros) will be used to realize transfer functions based on the Min-Delay type transfer functions. On account of the investigative nature of this thesis, and the many different types of loop-filters required, all loop filters will be designed off-chip.

3.3.1 Loop Filter Design for Gaussian-Type DLLs

We shall demonstrate the procedure used to design the loop filters based on the method of

a cascade of biquads through a 5th order example. The 5th-order DLL requires a frequency de-normalized and unit-scaled loop filter as listed in Table 2.2 and repeated below:

$$F'(s)=\frac{27.6132}{s^5+8.0817s^4+29.0386s^3+57.0825s^2+60.6567s}$$

The transfer function has a single pole at DC, two sets of complex pole pairs and no finite zeroes. The pole at DC can be realized using an integrator stage and the remaining complex pole pairs can be realized using two second-order unity-gain Sallen-Key filters. By cascading them as shown in Fig. 3.9 the 5th-order loop filter will be realized. The first stage of the loop filter is a simple I-V circuit, which is used to transform the current output of the charge pump into a voltage signal. Since it is composed by an op-amp, the "virtue ground" effect could make sure that the voltage at the output node of the charge pump (or the input node of the loop filter) is always maintained at analog ground (VDD/2). This helps to reduce the mismatching in the up and down branches of the charge pump, as discussed in the previous section.

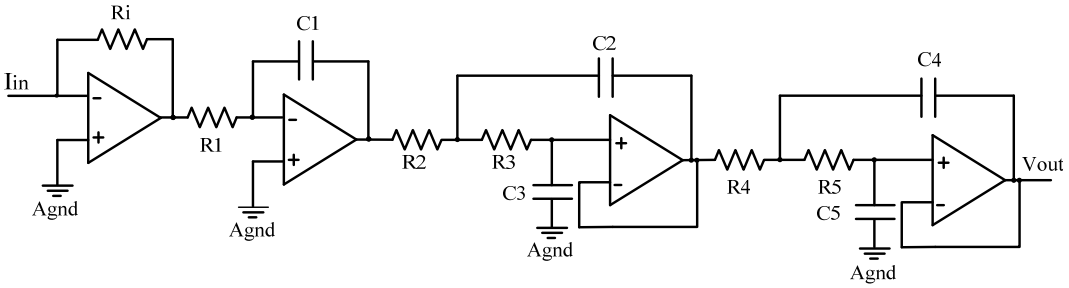


Fig. 3.9: Schematic circuit of the 5th-order loop filter.

In a similar fashion, the odd-order loop filters for the remaining DLLs, e.g. 3rd order, 7th order, etc., could be derived by adding or reducing the number of Sallen-Key biquads in cascade. As for the even-order filters, the only difference is that they also have an extra left half-plane (LHP) pole. These can be realized by adding a damped integrator or first-order bilinear stage after the first integrator stage as shown in Fig. 3.10 for a 4th order implementation. The general block diagrams of odd order and even order loop filters are shown in Fig. 3.11 and Fig. 3.12, respectively.

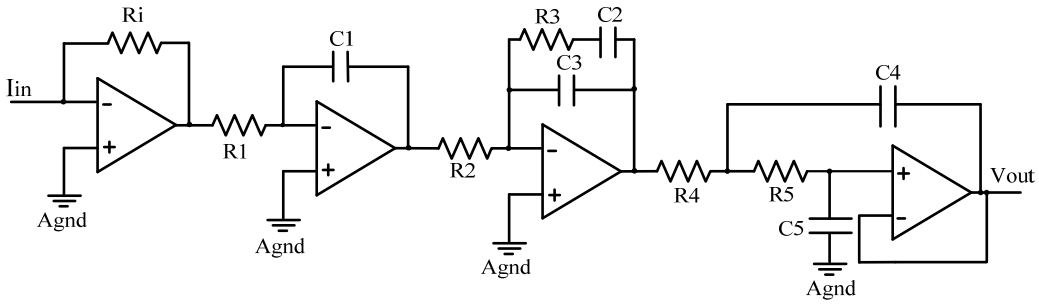


Fig. 3.10: Schematic circuit of the 4th-order loop filter.

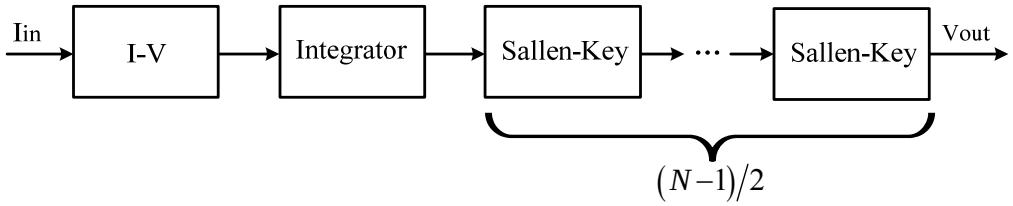


Fig. 3.11: Block diagram of odd order loop filter.

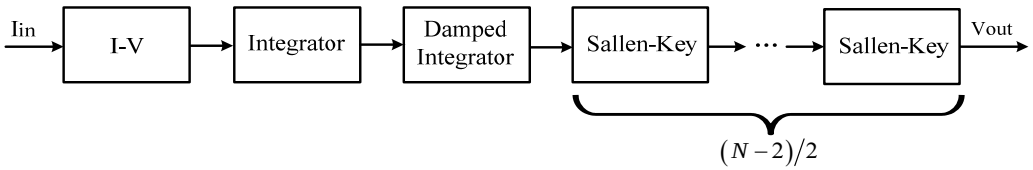


Fig. 3.12: Block diagram of even order loop filter.

To calculate the values of the R and C in each stage of the active circuits, one needs to identify the individual transfer functions of each stage. For instance, in the 5th order example shown in Fig. 3.10, one can write

I-V Stage: $$F_1(s) = -R_i \tag{3.1}$$

Integrator Stage: $$F_2(s) = -\frac{1}{sRC} \tag{3.2}$$

Damped-Integrator Stage: $$F_3(s) = -\frac{R_2/R_1}{sR_2C_1 + 1} \tag{3.3}$$

Sallen-Key Stage: $$F_4(s) = \frac{1/R_1R_2C_1C_2}{s^2 + \frac{1}{(R_1 \,||\, R_2)C_1}s + \frac{1}{R_1R_2C_1C_2}} \tag{3.4}$$

The 5th order loop filter transfer function would then be expressed as:

$$\begin{aligned} F(s) &= F_1(s)\cdot F_2(s)\cdot F_4(s)\cdot F_4(s) \\ &= \frac{R_i}{R_1 C_1 s}\cdot\frac{1/R_2R_3C_2C_3}{s^2+\frac{1}{(R_2 \parallel R_3)C_2}s+\frac{1}{R_2R_3C_2C_3}}\cdot\frac{1/R_4R_5C_4C_5}{s^2+\frac{1}{(R_4 \parallel R_5)C_4}s+\frac{1}{R_4R_5C_4C_5}} \end{aligned} \tag{3.5}$$

The frequency-deformalized and factor scaled 5th-order loop filter transfer functions of a DLL with specifications that $K_p = 8.053\times10^{-5}$ V/rad, $K_d = 7.81$ rad/V, $BW = 50$ kHz and the reference clock frequency of $f_c = 50$ MHz would be calculated as:

$$\begin{aligned} F(s) &= \frac{1}{8.053\times10^{-5}\times7.81}\cdot\frac{27.6132}{s^5+8.0817s^4+29.0386s^3+57.0825s^2+60.6567s}\Bigg|_{s\to\frac{s}{100\pi\times10^3}} \\ &= \frac{1.344\times10^{32}}{s^5+2.539\times10^6 s^4+2.866\times10^{12}s^3+1.77\times10^{18}s^2+5.909\times10^{23}s} \\ &= \frac{1.344\times10^{32}}{s}\cdot\frac{1}{s^2+1.8579\times10^6 s+1.0228\times10^{12}}\cdot\frac{1}{s^2+6.8114\times10^5 s+5.7772\times10^{11}} \end{aligned} \tag{3.6}$$

Comparing the coefficients of Eqn. (3.5) and (3.6), we could obtain the individual values of the Rs and Cs for each stage. The calculated R and C values with the above-mentioned DLL specifications are listed in Table 3.3 and Table 3.4 for loop filter orders ranging from 2 to 8. The complete set of schematics for these loop filters are provided in Appendix B.

One advantage of using Sallen-Key topology to realize the conjugate pole pairs instead of other filter topologies, e.g. Multi-feedback (MFB), is that it has better sensitivity to component variations [32]. This means small changes in parameter values will not make a big difference to the pole positions, which is a good characteristic when one implements the resistors and capacitors using fixed-value components with 5% tolerances purchased off-the-shelf.

Table 3.3: Parameters of odd order filters:

Order / Para	3	5	7
Ri	470	470	470
R1	1k	1k	1k
C1	1.8n	2.2n	2.7n
R2	980	900	840
R3	980	900	840
C2	1.5n	1.2n	1.1n
C3	1n	1n	1n
R4	-	590	620
R5	-	590	620
C4	-	5n	2.4n

C5	-	1n	1n
R6	-	-	430
R7	-	-	430
C6	-	-	11.1n
C7	-	-	1n

Table 3.4: Parameters of even order filters

Order / Para	2	4	6	8
Ri	470	470	470	470
R1	1k	1k	1k	1k
C1	1.3n	2n	2.4n	3n
R2	1.2k	1k	910	820
C2	1n	1n	1n	1n
R3	1.2k	1k	910	820
R4	-	750	760	750
R5	-	750	760	750
C4	-	2.8n	1.7n	1.3n
C5	-	1n	1n	1n
R6	-	-	490	520
R7	-	-	490	520
C6	-	-	7.8n	3.5n
C7	-	-	1n	1n
R8	-	-	-	400
R9	-	-	-	400
C8	-	-	-	14.7n
C9	-	-	-	1n

The circuit-level filter designs were verified using SPICE simulations. The filter transfer characteristics from SPICE simulations are compared with their theoretical values in Fig. 3.13. We see that there are no significant visible errors between them. A closer look reveals a similar result.

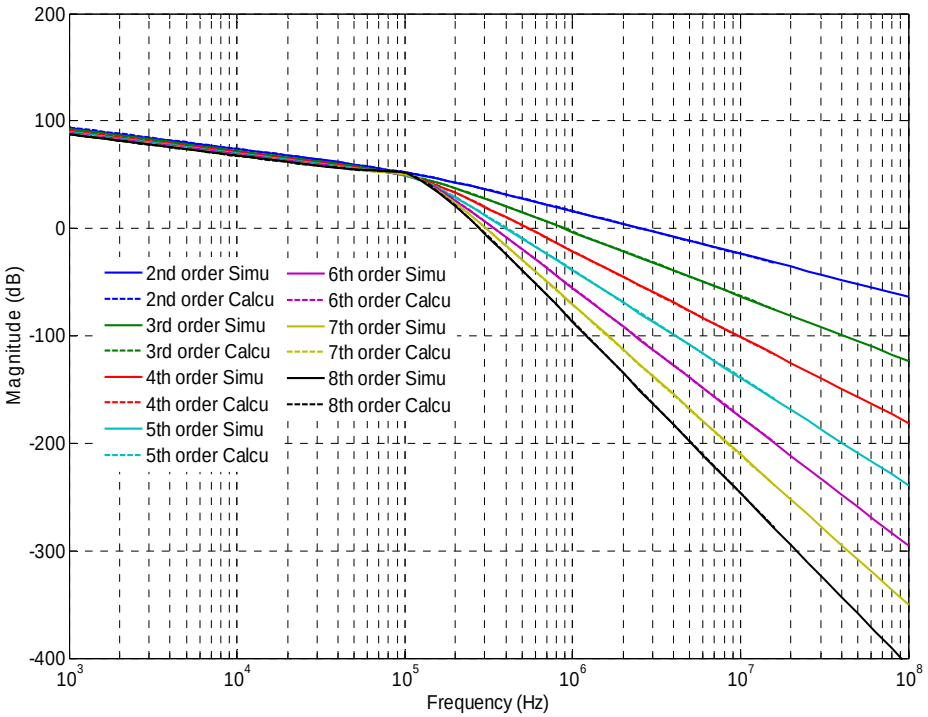


Fig. 3.13: Output characteristics of the loop filters from SPICE simulations and theoretical values.

3.3.2. Loop Filter Design for Min-Delay Type DLLs

In the case of the Min-Delay filter types, the filter transfer functions contain both poles and zeros. The extra zeros complicate the filter topology when using the cascade of biquad approach. For this reason, a state-space realization approach was selected instead. The first step is to transform the filter transfer functions in Eqn. (2.31) into a set of state-space equations having the following general form:

$$\begin{cases} sX(s) = AX(s) + bU(s) \\ Y(s) = cX(s) + dU(s) \end{cases} \tag{3.7}$$

Here $X(s)$ is an N-dimensional vector describing the states of the system, U(s) is the scalar input; $Y(s)$ is the scalar output, A, b and c are $(N \times N)$, $(N \times 1)$, $(1 \times N)$ dimension constant-coefficient matrices, respectively. The final term *d* is assumed to be a scalar. If the controllable canonical form is used in design, the general state-space equations of an Nth-order loop filter would have the following form:

$$s\begin{pmatrix} X_1(s) \\ X_2(s) \\ \vdots \\ X_{N-1}(s) \\ X_N(s) \end{pmatrix} = \underbrace{\begin{pmatrix} -b_{N-1} & -b_{N-2} & \cdots & -b_1 & 0 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{pmatrix}}_{A} \begin{pmatrix} X_1(s) \\ X_2(s) \\ \vdots \\ X_{N-1}(s) \\ X_N(s) \end{pmatrix} + \underbrace{\begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}}_{b} U(s) \quad (3.8)$$

$$Y(s) = \underbrace{\begin{pmatrix} a_{N-1} & a_{N-2} & \cdots & a_1 & a_0 \end{pmatrix}}_{c} \begin{pmatrix} X_1(s) \\ X_2(s) \\ \vdots \\ X_{N-1}(s) \\ X_N(s) \end{pmatrix} \quad (3.9)$$

For example, a 3rd-order loop filter has three poles and two zeros and its state-space equations would be written as:

$$\begin{aligned} s\begin{pmatrix} X_1(s) \\ X_2(s) \\ X_3(s) \end{pmatrix} &= \begin{pmatrix} -b_2 & -b_1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X_1(s) \\ X_2(s) \\ X_3(s) \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} U(s) \\ Y(s) &= \begin{pmatrix} a_2 & a_1 & a_0 \end{pmatrix} \begin{pmatrix} X_1(s) \\ X_2(s) \\ X_3(s) \end{pmatrix} \end{aligned} \quad (3.10)$$

or in expanded form as

$$\begin{aligned} sX_1(s) &= -b_2X_1(s) - b_1X_2(s) + U(s) \\ sX_2(s) &= X_1(s) \\ sX_3(s) &= X_2(s) \\ Y(s) &= a_2X_1(s) + a_1X_2(s) + a_0X_3(s) \end{aligned} \quad (3.11)$$

Equation (3.11) could be realized by the circuit topology shown in Fig. 3.14. As with the filter topology of the Gaussian type DLLs, the first stage is the same I-V circuit, which is not shown here. By inspection, one could obtain the following relationships between the active-RC circuit and the state-space equations:

$$
\begin{aligned}
1/(R_1C_1) &= 1 \\
1/(R_2C_1) &= b_2 \\
1/(R_3C_1) &= b_1 \\
1/(R_4C_2) &= 1 \\
R_6/R_5 &= 1 \\
1/(R_7C_3) &= 1 \\
R_{11}/R_8 &= a_2 \\
R_{11}/R_9 &= a_1 \\
R_{11}/R_{10} &= a_0
\end{aligned}
\tag{3.12}
$$

The same design method is applied to any other orders of the loop filters. The schematics of the 1st to 5th order loop filters are shown in Figs. 3.14 to 3.18.

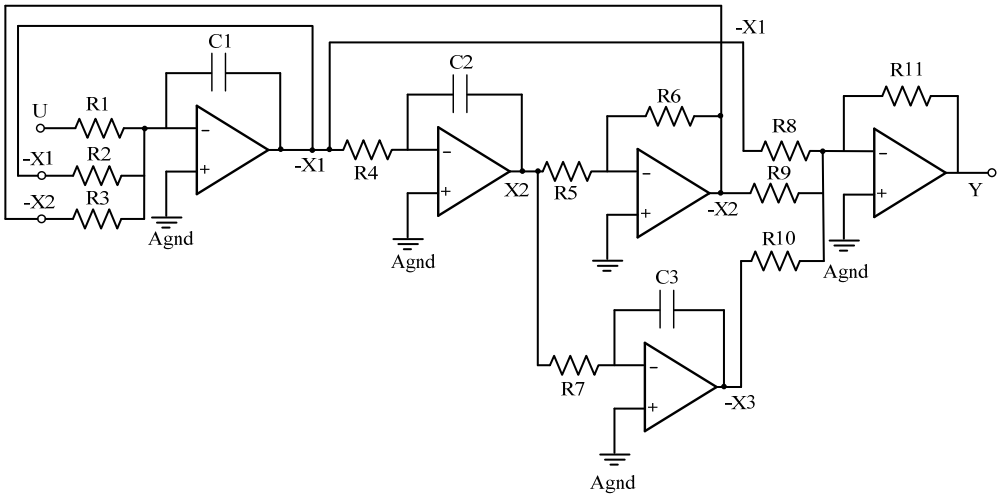


Fig. 3.14: Schematic of the 3rd-order state-space loop filter.

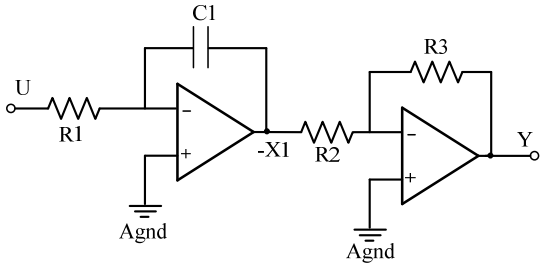


Fig. 3.15: Schematic of the 1st-order state-space loop filter.

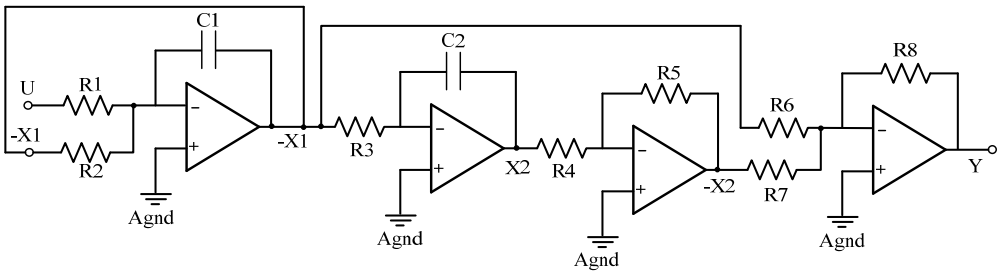


Fig. 3.16: Schematic of the 2nd-order state-space loop filter.

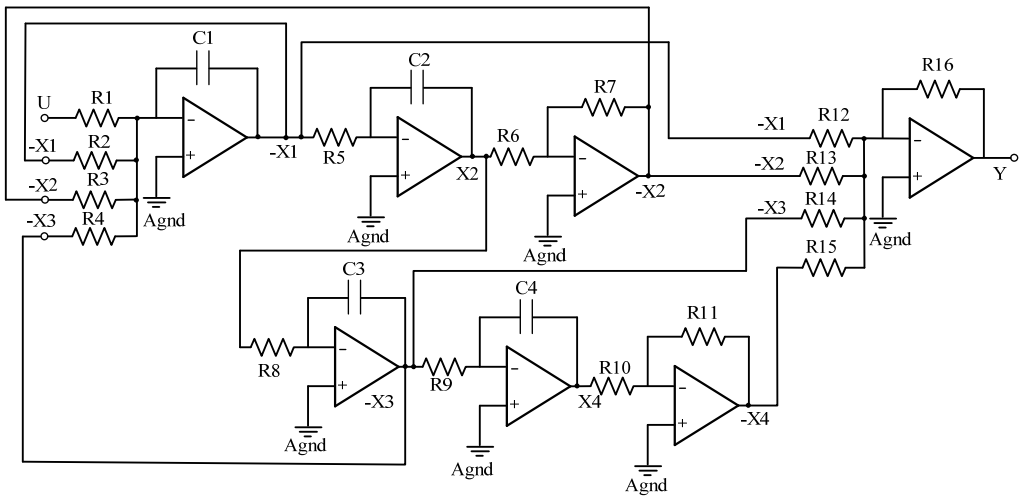


Fig. 3.17: Schematic of the 4th-order state-space loop filter.

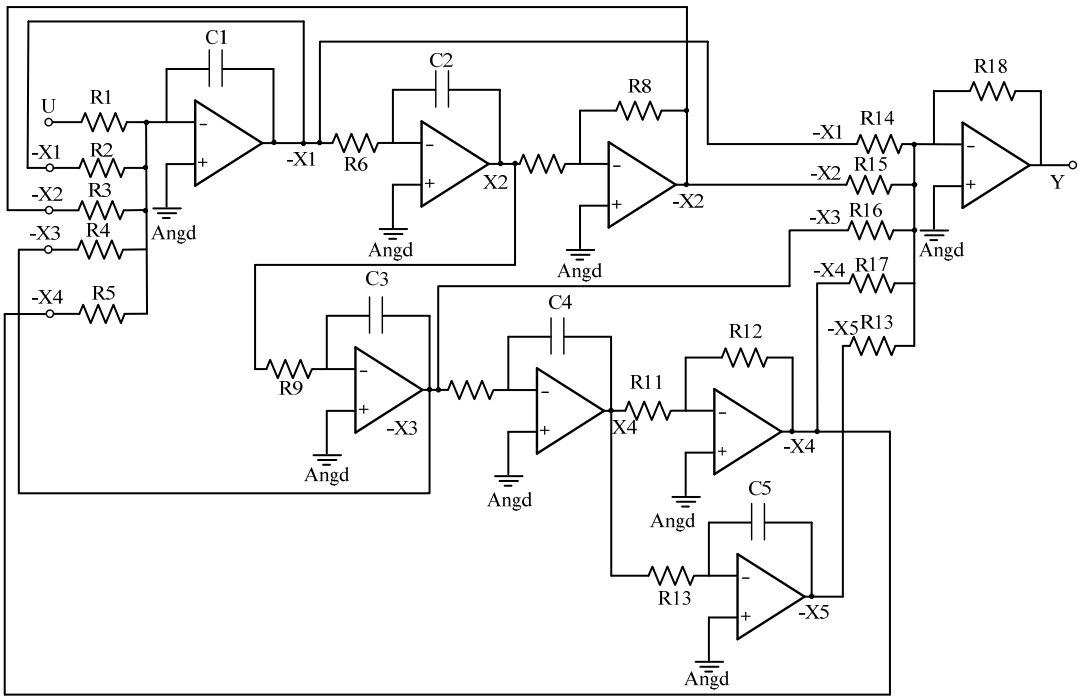


Fig. 3.18: Schematic of the 5th-order state-space loop filter.

3.4 Multi-Controlled VCDL Design

The delay line is used to delay the input signal by a controlled amount of time and it can be voltage or current controllable in order to operate within the feedback loop. In this work, a multi-controlled shunt-capacitive voltage controlled delay line (VCDL) topology is used, as it has the largest linear operating region. It is designed to be multi-controlled in order to handle a wide range of input frequencies.

The schematic of one element delay cell is shown in Fig. 3.19 and the transistor sizes are listed below it in Table 3.5. Within the schematic, M1, M2 are inverters, the control voltage at the gate of M3 is used to adjust the propagation delay of each cell and the parasitic capacitances of M4 and M5 are used to influence the propagation delay. Switch S0 is used to decide whether M5 is connected into the circuit to increase the propagation delay. The circuit details related to switch S0 is shown in Fig. 3.20 and the corresponding transistor sizes are listed in Table 3.5 as well. Switch S0 incorporates a dummy switch structure to reduce the charge injection issues with switches M8 and M9. A block diagram of the complete VCDL is shown in Fig. 3.21. It is composed of a 24 programmable delay cells cascaded together. The selective switch "Sel" is a 2:1 MUX which is used to decide whether there will be 24 delay cells or only 12 delay cells connected in the circuit. The schematic of the switch "Sel" is illustrated in Fig. 3.22 and the transistor sizes are in Table 3.6.

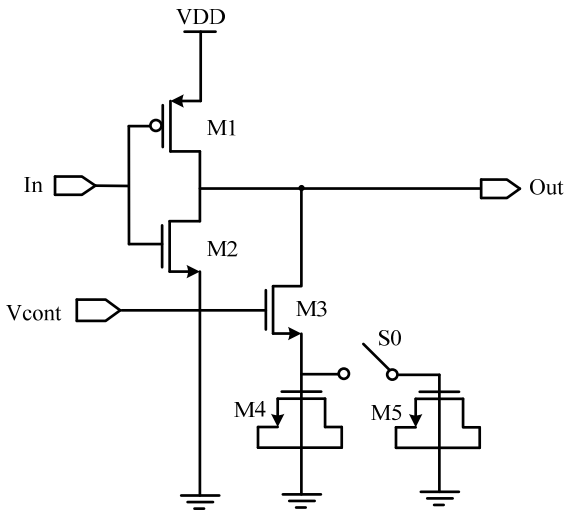


Fig. 3.19: Schematic of one cell of the VCDL.

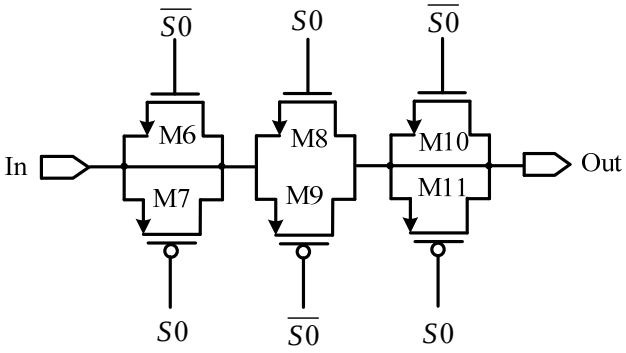


Fig. 3.20: Schematic of switch S0.

Table 3.5: Parameters of transistors in VCDL delay cell

Transistor	Width	Length	Transistor	Width	Length
M1	4.5u	300n	M7	5u	300n
M2	1.5u	300n	M8	4u	300n
M3	30u	300n	M9	10u	300n
M4	5u	300n	M10	2u	300n
M5	7u	300n	M11	5u	300n
M6	2u	300n			

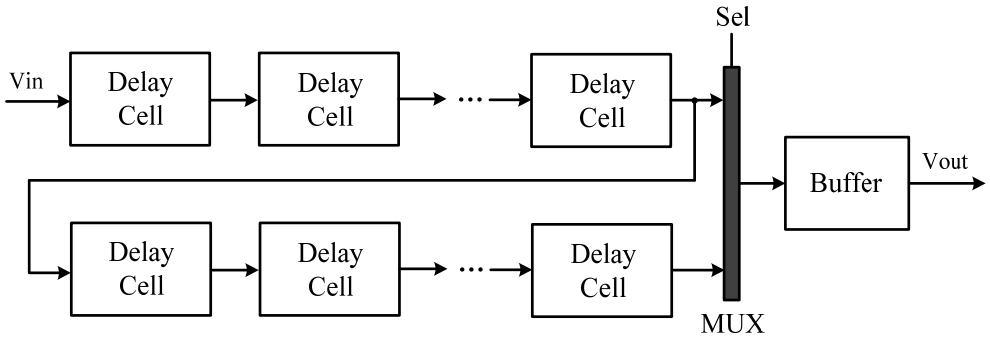


Fig. 3.21: Block diagram of the multi-controlled VCDL.

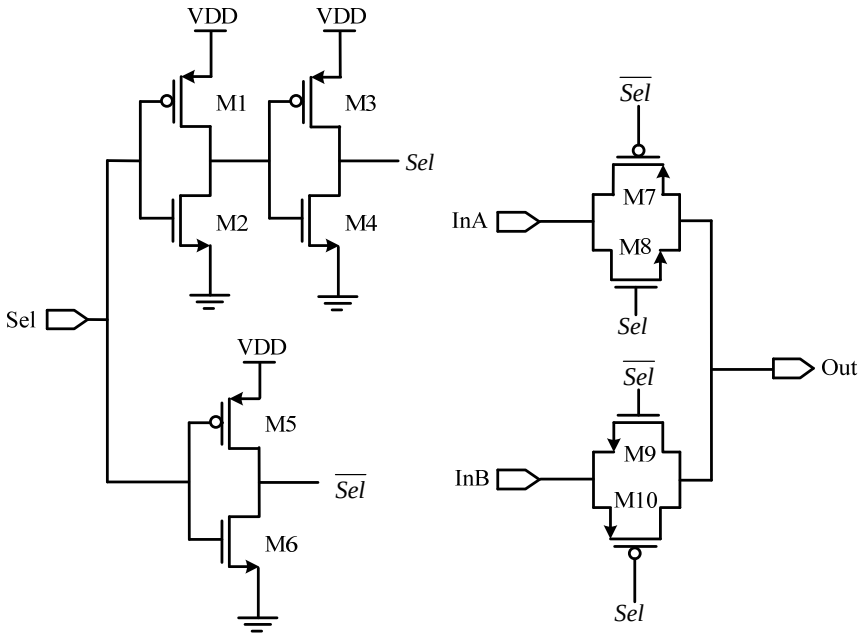


Fig. 3.22: Schematic of MUX Sel.

Table 3.6: Parameters of transistors in MUX

Transistor	Width	Length	Transistor	Width	Length
M1	3u	300n	M6	2u	300n
M2	1u	300n	M7	7u	300n
M3	3u	300n	M8	2u	300n
M4	1u	300n	M9	7u	300n
M5	6u	300n	M10	2u	300n

The output characteristics (the propagation delay vs. control voltage) with respect to the control signals (the logics of the switches S0 and Sel) of the multi-controlled VCDL are presented in Fig. 3.23. It was obtain through simulation using the Spector simulator found in the Cadence CAD package. The plot shows that the VCDL has a linear characteristic over four controlled-input conditions. The maximum propagation delay occurs when both switches are on (S0, Sel=1, 1), the minimum delay occurs when both switches are off (S0, Sel=0, 0) and a middle value of delay occurs when one of the switches is on and the other is off (S0, Sel=0, 1 or S0, Sel=1, 0). When the input signal of the DLL has a large frequency range, e.g., from 28.6 MHz to 100 MHz as with this design, the proportion delay of the VCDL could be selected from one of the four conditions using switches S0 and Sel, providing the DLL a wider operating range.

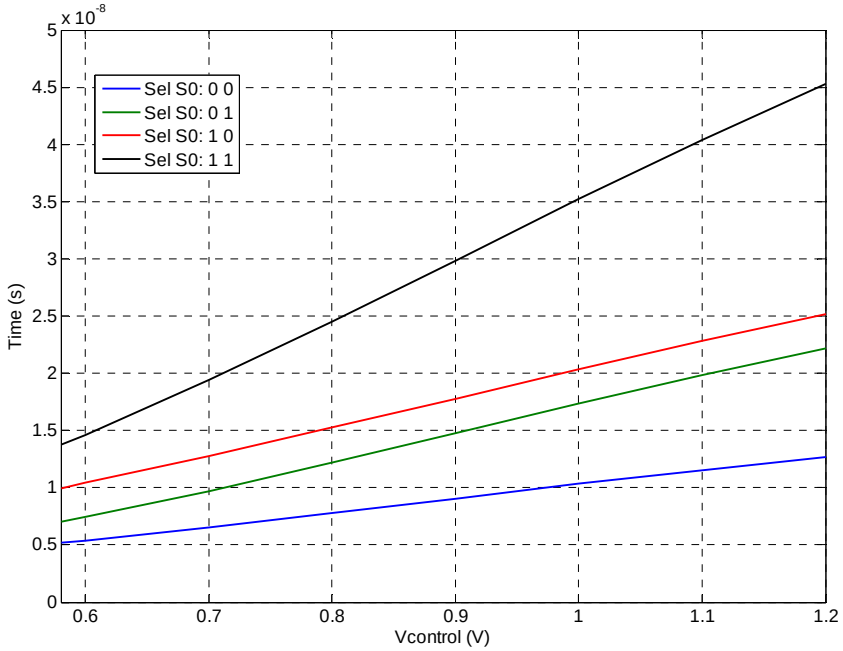


Fig. 3.23: Output characteristics the shunt-capacitive multi-controlled VCDL

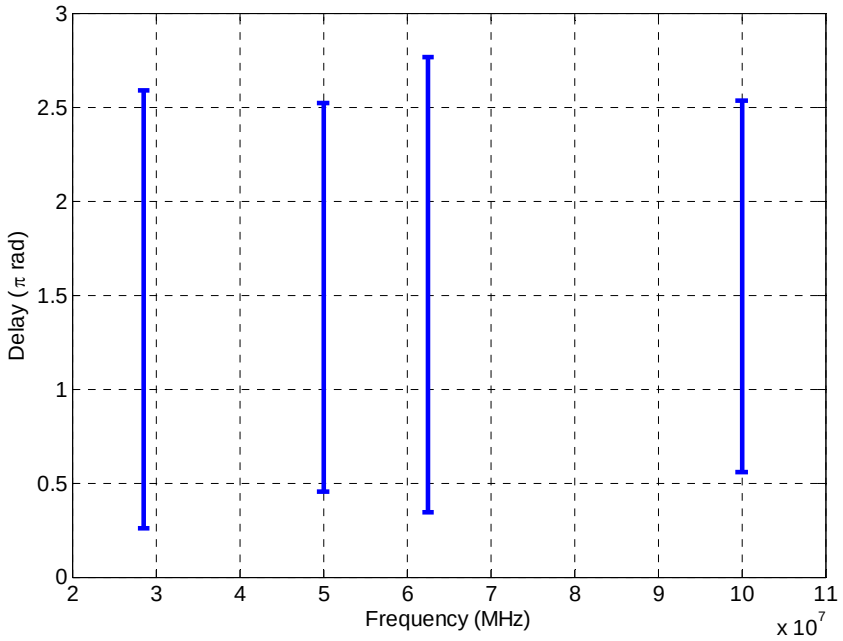


Fig. 3.24: Linear operation region of the shun-capacitive multi-controlled VCDL

Figure 3.23 illustrates that when the range of the input control voltage lies approximately between 0.58 V to 1.2 V for all four-switch conditions, the VCDL provides linear operation with a slope or gain of $K_{d1} = 5.166 \times 10^{-8}$ s/V , $K_{d2} = 2.486 \times 10^{-8}$ s/V , $K_{d3} = 2.4822 \times 10^{-8}$ s/V , and $K_{d4} = 1.2186 \times 10^{-8}$ s/V . The linear operation regions in π radians with respect to the input frequencies (28.6 MHz, 50 MHz, 62.5 MHz and 100 MHz) are illustrated in Fig. 3.24.

The four min-max bars correspond to the four delay-lines in Fig. 3.23. The linear region in each case is approximately from 0.25T to 1.25T, which is large enough for normal DLL operations.

Another prevalent delay line topology is the current starved topology. Reference [33] introduces a typical multi-controlled current-starved VCDL circuit. The delay cell of the current-starved VCDL is shown in Fig. 3.25 and the transistor sizes are listed in Table 3.7. The block diagram of the complete delay line is the same as that shown for the shunt-capacitive VCDL in Fig. 3.21. The output characteristics of this multi-controlled current-starved VCDL are shown in Fig. 3.26 and its linear operation regions under the same input frequencies are shown in Fig. 3.27.

Comparing the simulation results of these two types of VCDL, we could see that the shunt-capacitive VCDL has an obvious wider linear operation region, which is preferred in this work. Besides, both the transistor sizes and the number of transistors used in shunt-capacitive VCDL are smaller than those in current-starved VCDL.

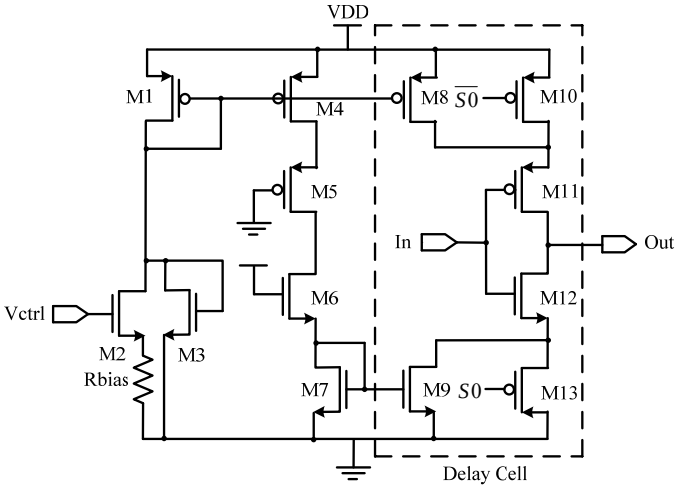


Fig. 3.25: Schematic of one delay cell of the current-starved VCDL [33].

Table 3.7: Parameters of transistors in MUX

Transistor	Width	Length	Transistor	Width	Length
M1	20u	300n	M8	4u	300n
M2	10u	300n	M9	2u	300n
M3	1.4u	300n	M10	11u	300n
M4	20u	300n	M11	25u	300n
M5	25u	300n	M12	10u	300n
M6	10u	300n	M13	4u	300n
M7	10u	300n	Rbias	600 Ohm	

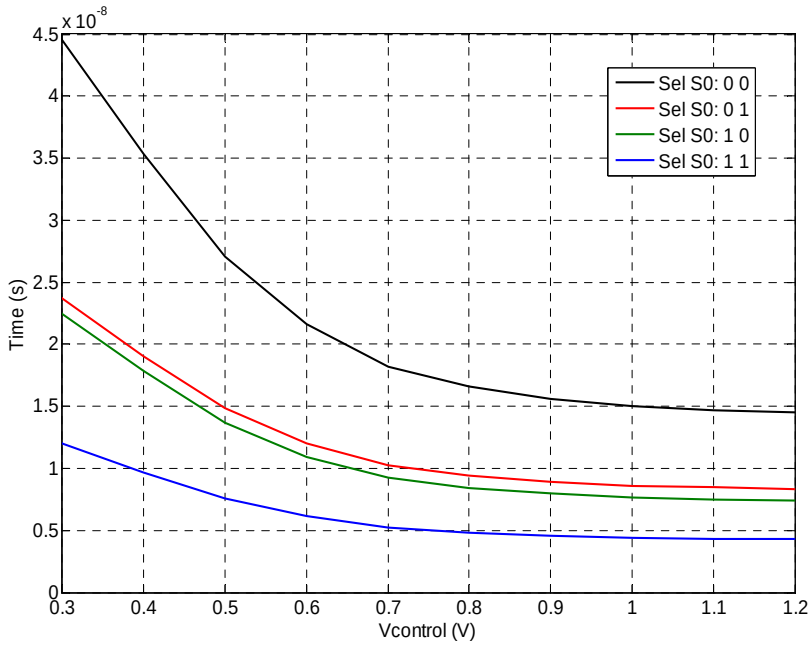


Fig. 3.26: Output characteristics the current-starved multi-controlled VCD.

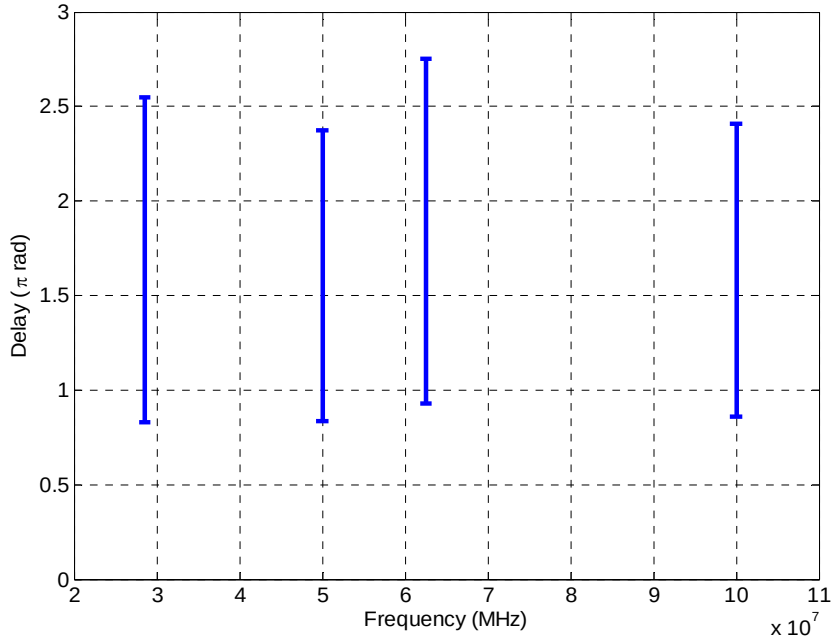


Fig. 3.27: Linear operation region of the current-starved multi-controlled VCDL.

3. 5 Start-up Circuit

When DLL starts up from its initial condition, it could either lock to the next coming edge or the previous edge of the data input, namely, the phase difference between the input signal and the output signal could either be 2π radians or 0 radians according to the start-up condition. By adding a phase compensation coefficient $2\pi/s$, first seen in Fig. 2.2, Eqn. (2.1)

assumes that the DLL is going to lock to the next coming edge of the input signal, which may not be enough to describe the full start-up condition. Rather, there are two DLL start-up possibilities. Eqn. (2.1) representing the system equation for the block diagram of Fig. 2.2 should be revised as follows:

$$V_e(s) = \begin{cases} K_p(\theta_{data}(s) - \theta_o(s) + \dfrac{2\pi}{s}) & \text{if UP is before DN} \\ K_p(\theta_{data}(s) - \theta_o(s)) & \text{if DN is before UP} \end{cases} \tag{3.13}$$

where UP and DN represents the two outputs of the PFD. Eqn. (3.13) illustrates that if the UP signal comes first the DLL is going to lock to the next coming edge, otherwise, the DLL is going to lock to the previous edge. To ensure that the DLL locks in the middle of its operating range, a start-up with a fast response time is necessary.

3.5.1 Linear Range of Operation

The operating regions of the DLL in steady state or when locked are shown in Fig. 3.28. Here the DLL operating regions are divided into six parts, A, B, C, D, E and F, based on the operating regions of its PFD and VCDL. Assuming that the 2π radians is the place where DLL is initially locked, then though a superposition of PFD working region and VDCL working region (linear region plus non-linear region) at 2π radians, we could obtain Fig. 3.28.

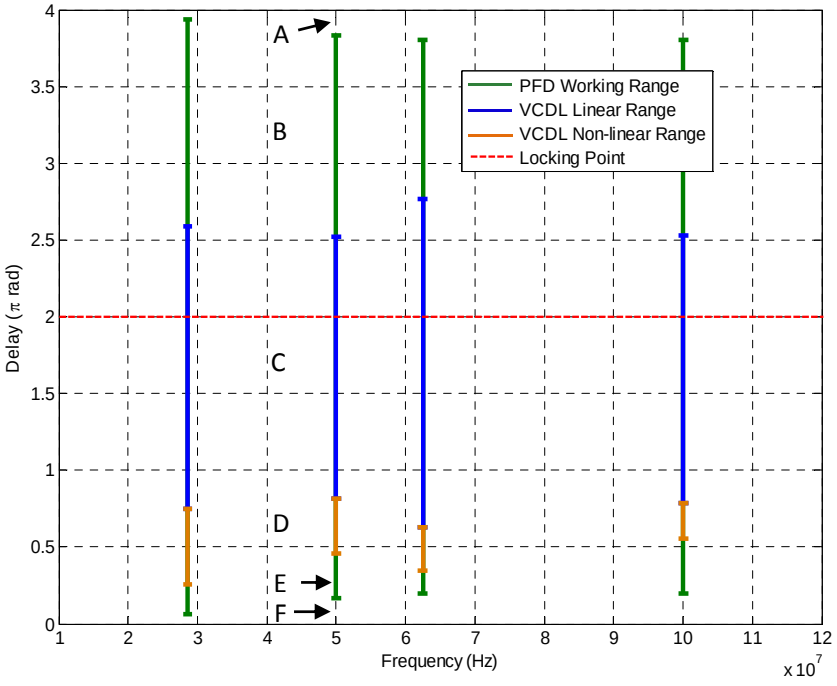


Fig. 3.28: Operating regions of the DLL in lock vs. input frequencies.

Referring to Eqn. (2.17), region C (within the linear region of both PFD and VCDL) is the DLL linear operation region. If the input signal has a large step change in phase that pushes the DLL into region D (within the working region of PFD and the non-linear region of VCDL), the DLL could still lock to the correct place but with a slew in its transient response because of the decrease in the VCDL slope in its non-linear region. If DLL is pushed from its locking place to region B or E (outside the working region of VCDL), the system will diverge, i.e. the control voltage of the VCDL will either become larger than VDD or smaller than VSS, which breaks up the feedback. If the DLL has a chance to fall in region A or F from its locking point (outside the working region of PFD), the equilibrium point of the system will changed and the operation of the DLL would be hard to predict.

3.5.2 Start-up Operation

The operation regions at start-up condition (start from rest) is similar to that from the steady state or lock state but has a little difference on the supperposition of PFD and VCDL working regions. We wish that the DLL could start up from within its linear operation region to achieve a minimum settling time. Therefore, the linear region of PFD is superimposed on the working region of VCDL at a start up point shown in red dash line in Fig. 3.29. This start-up point is within the linear operation region of the DLL. Figure 3.29 illustrates that the DLL could lock correctly to any place in region C or D from this start-up point, but the system will diverge if the DLL begins in a point within region B or E since they are outside of the VCDL working region. As mentioned previously, the operation of the DLL is hard to predict if it starts within region A or F.

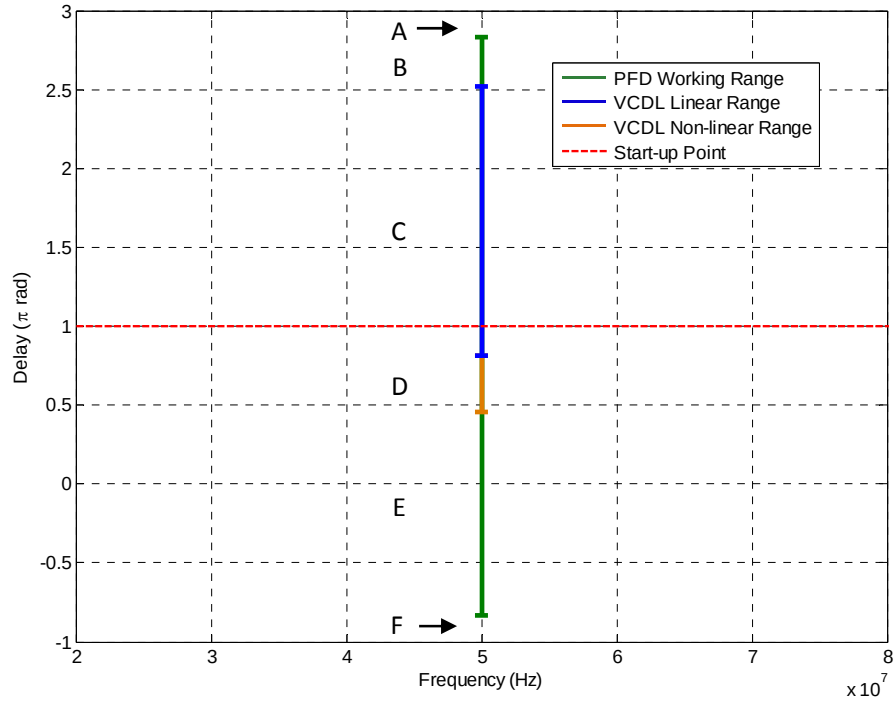


Fig. 3.29: Operation regions of DLL at start-up state.

Based on the discussion above, a start-up circuit is necessary to ensire that the DLL locks to a point inside region C. The block diagram of the start-up circuit is shown in Fig. 3.30. It is composed of the PFD circuit shoiwn previously in Fig. 3.5, two D-flip flops and two inverters. The schematic of the D-flip flop is shown in Fig. 3.31 and the transistor sizes are listed in Table 3.8. Dynamic logic is used in order to achieve better performance at high input frequencies. The timing diagram of the start-up circuit is shown in Fig. 3.32. When the "start" signal goes high, q1 becomes high at the rising edge of InA, which makes RestA signal low and allows the UP signal to become high. Since q1 is connected to the D input of the second D-flip flop, q2 will become high only at the following rising edge of InB. This means at the start up condition, the PFD will always see InA signal first and produce a UP signal first, making the control voltage to rise until the system lock. This will make sure that the DLL is going to lock to 2π radians rather than 0radians in Fig. 3.29. If the control voltage is given an initail value of VDD/2 (0.6V), the DLL will start up within its linear operation region C and lock to 2π radians with a minimum settling time.

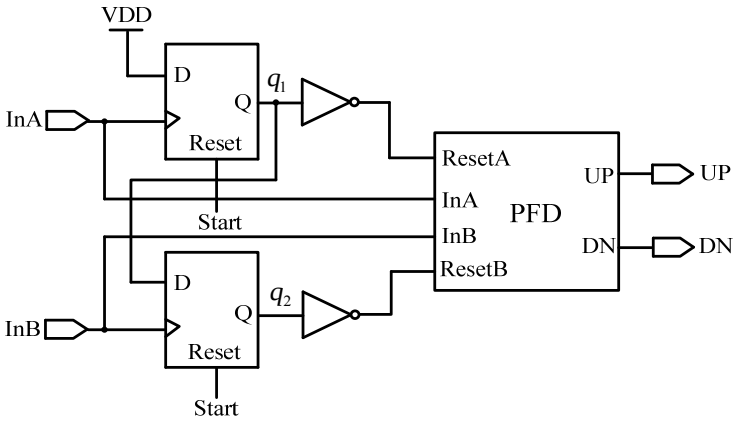


Fig. 3.30: Block diagram of the start-up circuit.

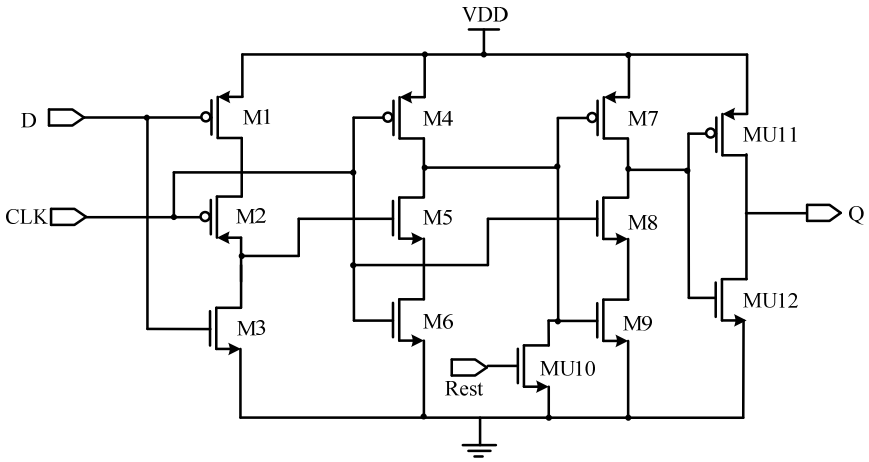


Fig. 3.31: Schematic of dynamic logic D-flip flop.

Table 3.8: Transistor sizes of the D-flip flop

Transistor	Width	Length	Transistor	Width	Length
M1	6u	300n	M7	6u	300n
M2	6u	300n	M8	2u	300n
M3	3u	300n	M9	3u	300n
M4	5u	300n	M10	3u	300n
M5	4u	300n	M11	12u	300n
M6	4u	300n	M12	4u	300n

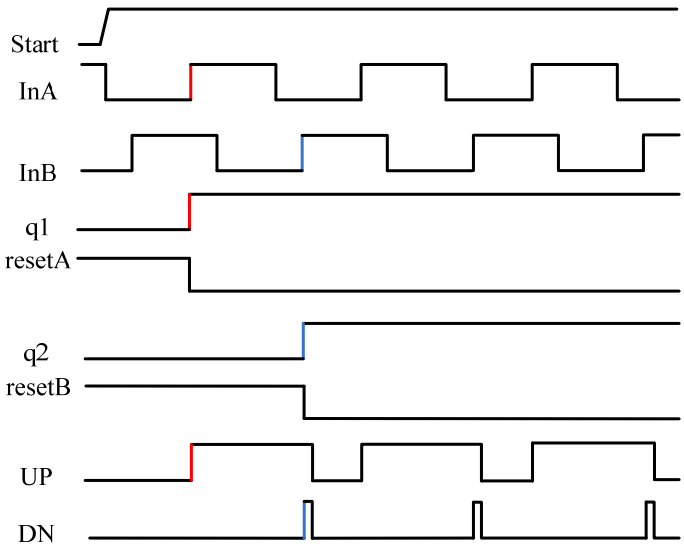
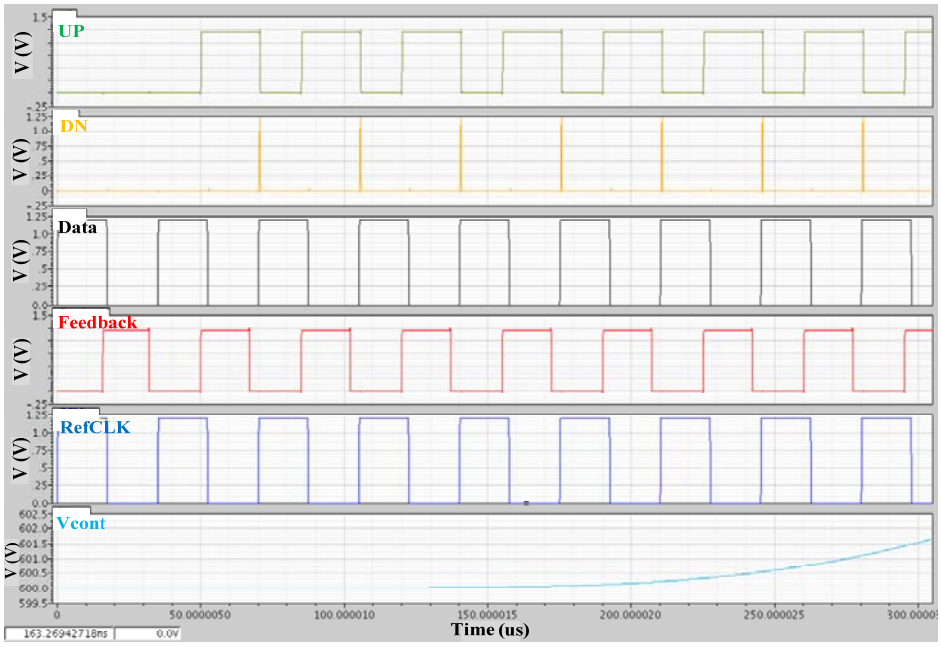
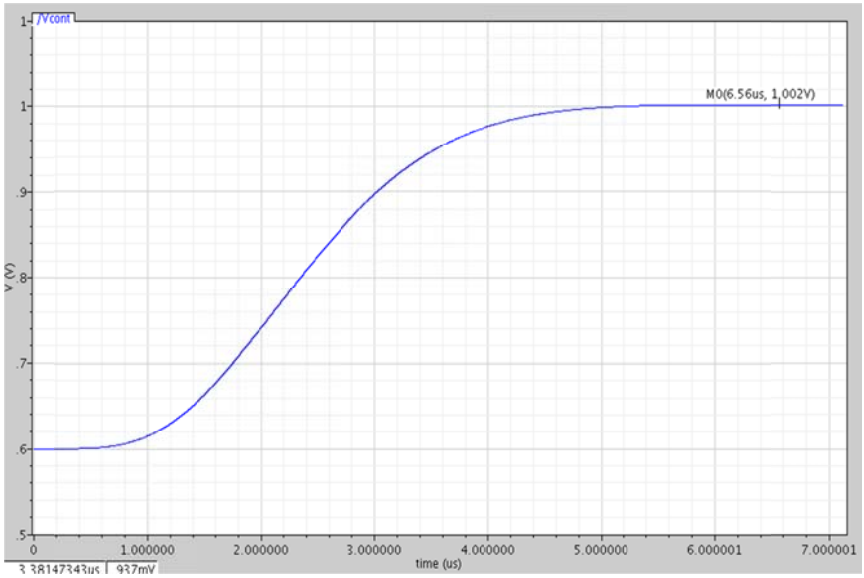


Fig. 3.32: Block diagram of the start-up circuit.

Figure 3.33 shows the correct start-up condition. In Fig. 3.33(a), the “Feedback” signal is connected to InA and the “Data” signal is connected to InB. Therefore, the DLL sees the rising edge of “Feedback” signal first and produces a wider UP signal, causing the control voltage to rise from 0.6 V all the way to its steady state value 1.02V. It has the minimum settling time in this case and the whole start-up process happenes within DLL linear operation region C.



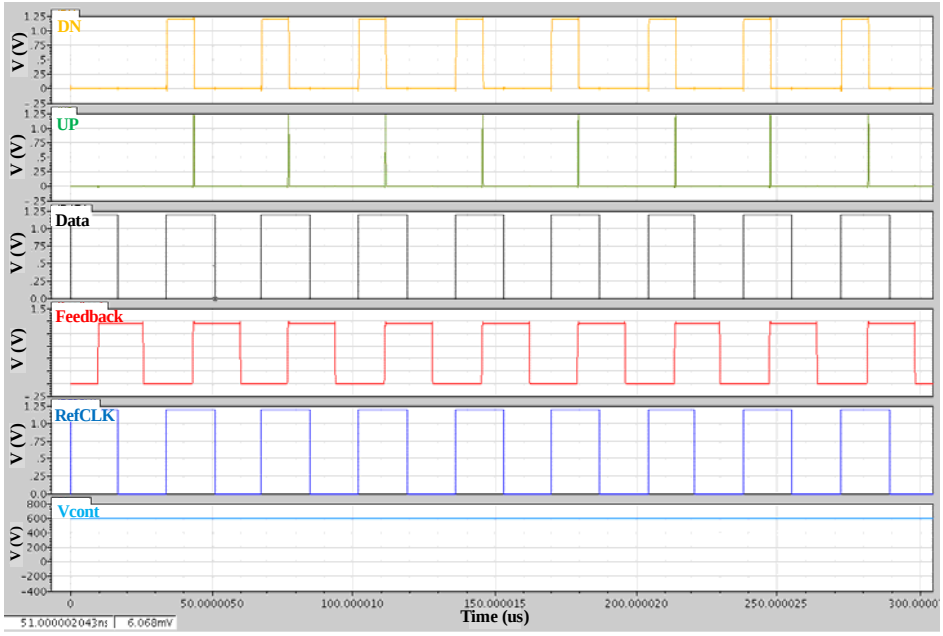
(a)



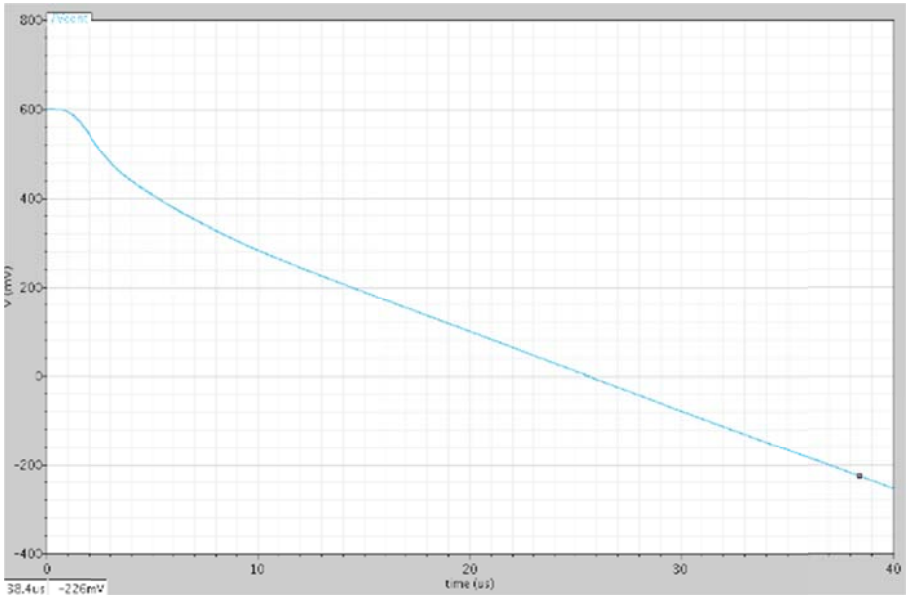
(b)

Fig. 3.33: Correct start up. (a) Wave forms. (b) Control voltage of the VCDL.

Figure 3.34 shows the wrong start-up condition. Without the start-up circuit, the DLL sees the rising edge of the "Data" signal first since it comes before the rising edge of the "Feedback" signal. Consequently, the PFD produces a wider DN signal and forces the control voltage to decrease from initial value of 0.6 V. By decreasing the control voltage, the DLL pretends to lock to the previous cycle (0radians). However since 0radians is within region E, the DLL goes from region C into region E and finally diverges, as shown by the VCDL control voltage in Fig. 3.34(b).



(a)



(b)

Fig. 3.34: Wrong start up. (a) Wave forms. (b) Control voltage of the VCDL.

3.6 Summary

In this chapter, the DLL building blocks used in this thesis are described in detail. First, a fast speed dynamic-logic PFD is chosen which has no obvious dead-zone and relative small blind zone. It is combined with a start-up circuit to solve the DLL start up issue. A switch-in-source type CP is adopted for its simple structure, low power consumption and good switching time. The designs of the high-order active-RC loop filters are described. The method of a cascade of biquads is used for Gaussian type transfer functions and the state-space approach is used for transfer functions based on the Min-Delay type transfer functions. Finally, a multi-controlled VCDL topology is adopted for a larger frequency range and a better linearity. The transistor-level simulation results are given for each building block and provide ample proof that the design is correct. These building blocks will then be used in the next chapter to construct a CMOS IC chip.

Chapter 4

IC Implementations Details

This chapter will present the design and simulation details related to the IC implementation of the high-order DLL prototype. The building blocks described in the previous chapter will be integrated into a 1 mm x 1 mm die fabricated a 130 nm TSMC CMOS process. The ancillary circuits used to support the development of the high-order DLL circuit will be explained as well. Chip-level simulations will be given to verify the correctness of the design.

4.1 IC Block Organizations

The organization of the DLL building blocks and the main signal flows are shown in Fig. 4.1. Among the major building blocks, the PFD, startup circuit, CP and VCDL are built within the chip while the loop-filter is built off-chip. Some ancillary circuits (not shown) such as the input/output buffers and the ESD protections are discussed in the next section. The "Input Data" and "RefCLK" signals are the DLL input signals and the "Output/Feedback" is the DLL output signal. The "Start" is the DLL start-up signal and the "S0" and "Sel" are the two control signals of the VCDL used to select one of the four delay lines. The "Sbuff" controls a switch that decides whether to output the DLL feedback signal for observation purposes.

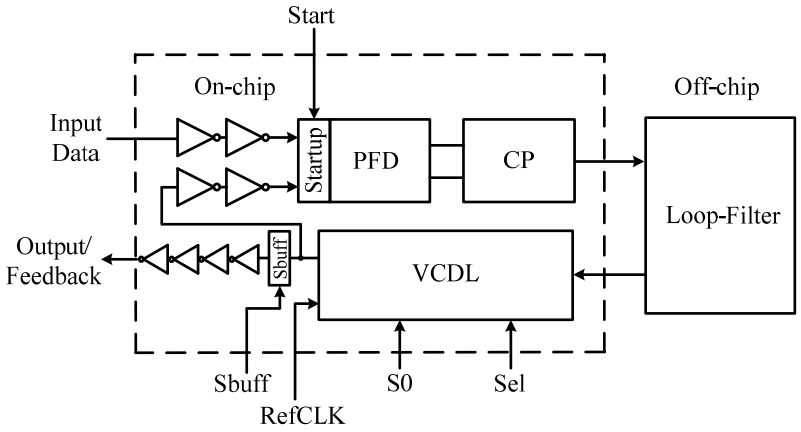


Fig. 4.1: Organization of the DLL building block for IC Implementation (within dashed lines).

The IC layout is shown in Fig. 4.2. It was laid out on a 1 mm x 1 mm die. The CMOS circuits are in the middle of the IC and marked with white boxes. The main DLL circuit on the die contains the PFD, CP, VCDL, and the input/output buffers; the loop filter is built off-chip. Notice that PFD/CP are duplicated below the VCDL circuit; it is label as PFD/CP 2 in the diagram. It is there for testing purposes in order to gain more information of the PFD and CP characteristics.

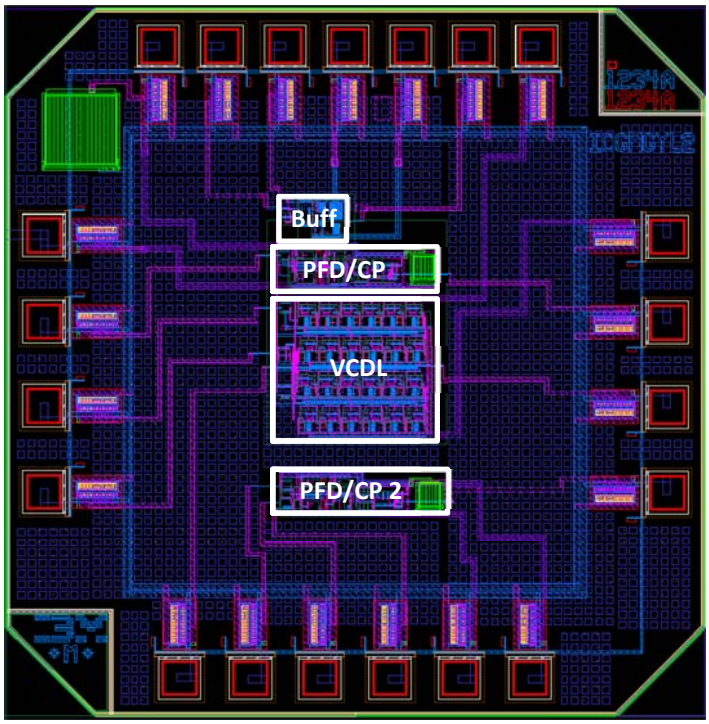


Fig. 4.2: DLL IC layout

4.2 Ancillary Circuit Details

4.1.1 Input/Output Buffers

Except for the building blocks related to the main circuit, there are also other factors that need to be taken into account when the IC is connected to the outside world. These include the added capacitance of the pads and bond wires, electrostatic discharge and loads into the measurement equipment. Each pad is connected to a bond wire, and then to an external pin in the package, which brings a large amount of extra capacitance. It is necessary to ensure that the on-chip components have the capability to drive this capacitance at a relatively high speed. Therefore the DLL design contains on-chip input/output buffers in order to interface with

off-chip sources and equipment. Both the input and output buffers are simple inverter chains consisting of a pair of inverters, as shown in Fig. 4.3. The transistors dimensions related to the input buffer are listed in Table 4.1. A total of four stages are used as output buffers and the transistors dimensions are listed in Table 4.2. The transistor sizes of this inverter chain are increasing from input to output so that a relatively large off-chip load can be driven with a reasonable propagation delay. This concept is similar to sizing a chain of inverters for minimum delay [34]. The driver was tested with a 50 Ω terminated, 20 pF load. This results in an approximate 800 mV square wave. In the IC design, this output buffer could be either connected or disconnect to the main DLL circuit through a CMOS switch "Sbuff", the schematic of which is the same as that in Fig. 3.20. Actual test result shows that a signal with a 200 mV magnitude is large enough to be analyzed on bench top signal analyzer. To decrease the power consumption, propagation delay, and the influences caused to the main circuit, it is better to decrease the sizes of the output buffer to half of the present sizes. This should be considered more carefully in future designs.

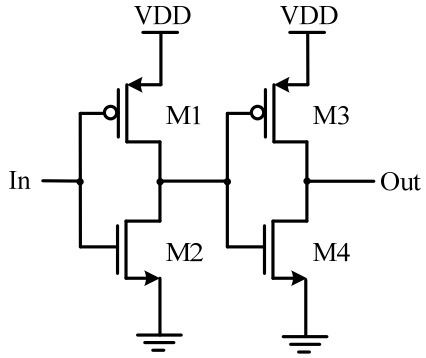


Fig. 4.3: Schematic of input/output buffer.

Table 4.1: Transistor sizes of input buffer

Transistor	Width	Length
M1	9u	300n
M2	3u	300n
M3	9u	300n
M4	3u	300n

Table 4.2: Transistor sizes of output buffer

Stage	Transistor	Width	Length
1st stage	M1	12u	300n
	M2	4u	300n

2nd stage	M1	40u	300n
	M2	15u	300n
3rd stage	M1	140u	300n
	M2	52u	300n
4th stage	M1	500u	300n
	M2	180u	300n

4.1.2 ESD Protection

ESD is the sudden flow of electricity between two objects caused by contact. This sudden flow of electricity is usually the result of a build-up of static charge on an instrument or person and, when this comes into contact with a portion of the circuit, can damage the electronic components due to the exposure to a very high voltage, and hence create a large destructive current. The ESD protection circuits are placed at each pad contacting the outside world and are designed to be a set of two diodes that can discharge the destructive current directly to GND. Figure 4.4 shows the way that the diodes are connected in the ESD protection circuit. The upper diode protects against damaging high voltages above VDD, and the lower diode protects against damaging below ground voltages. The ESD diodes are designed to be very large, $47\ \text{um} \times 68\ \text{um}$, in order to handle a large amount of current.

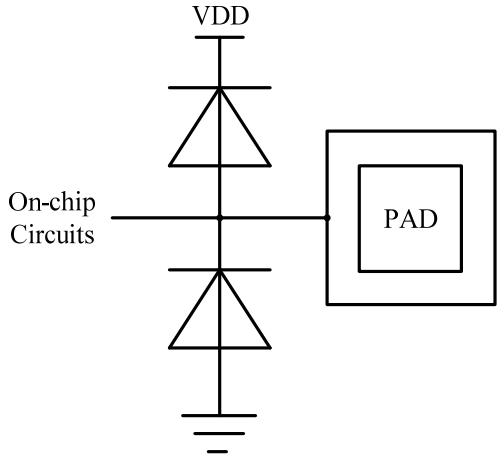


Fig. 4.4: Schematic of ESD Protection circuit.

4.3 Simulations Results

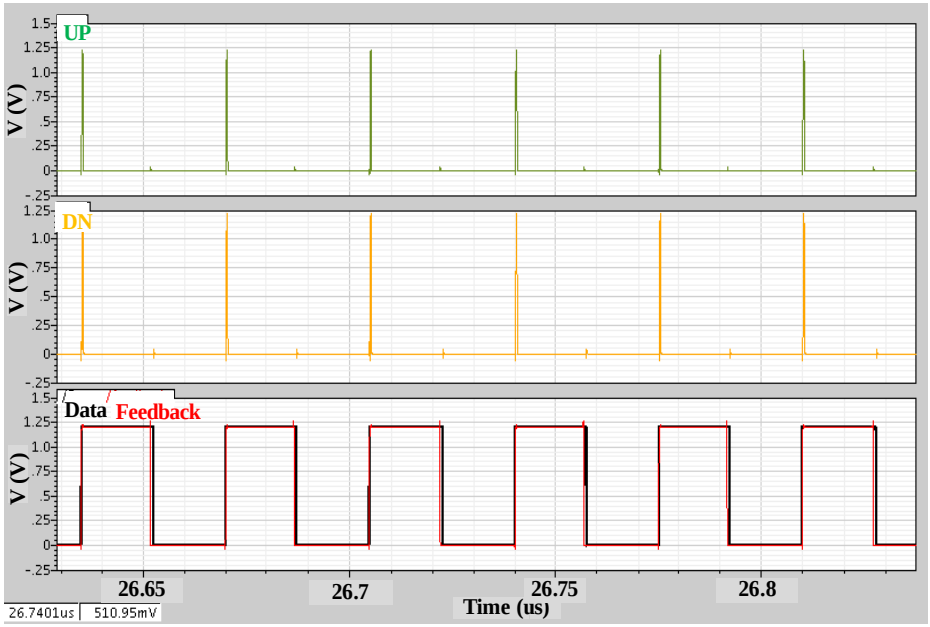
This section presents the closed-loop chip-level simulations results. Both steady state and dynamic operation are simulated in order to verify the correctness of the circuit design. The

simulations are carried out using the Spectre simulator using the 130 nm CMOS design kit from TSMC.

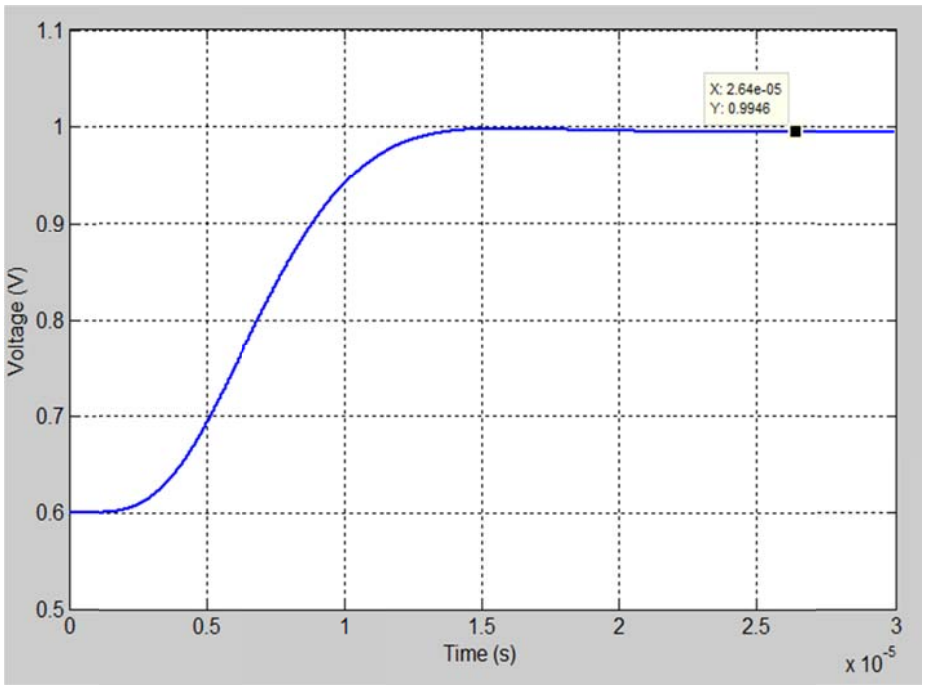
4.3.1 Steady-State Simulations

Under steady-state simulations, the DLL is to start up from rest and the input signals have a fixed phase and frequency (i.e., do not change over time). The DLL output signal is expected to lock to the input "Data" signal and the control voltage used to drive the VCDL should settle to a constant level within its linear operating range. There are four different VCDLs controlled by switches "Sel" and "S0" and each corresponds to one of the four input frequencies 28.6 MHz, 50 MHz, 62.5 MHz and 100 MHz. The order of the DLL is programmed to be one in the range of 2 to 8. The simulations will verify that the DLL start up correctly and locks to the desired operating point. Since there are many combinations of the VCDL and order, only simulation results for a 5^{th}-order and 7^{th}-order example will be given here. All the other orders have been verified and the results are consistent with those depicted with these two examples.

Figures 4.5 and 4.6 present the critical waveforms when DLL is locked and the VCDL control voltage during the settling process under two separate conditions. The first condition corresponds to a 5^{th}-order DLL with fc=28.6 MHz, K_{d1}= 9.2833 V/rad, K_P=0.0805 mV/rad, and BW=50 kHz, and the second to a 7^{th}-order DLL with fc=100 MHz, K_{d4}=7.6567 V/rad, K_P=0.0805 mV/rad and BW=50 kHz. In both cases, the input and output digital waveforms confirm that the DLL locks onto the incoming digital data. Moreover, the duty cycle of the UP and DN signals are both set to 50%. The subsequent plots of the control voltage demonstrate that the DLL start up correctly and settles to a final level in about 16.2 μs. This agrees with the theory in chapter 2.

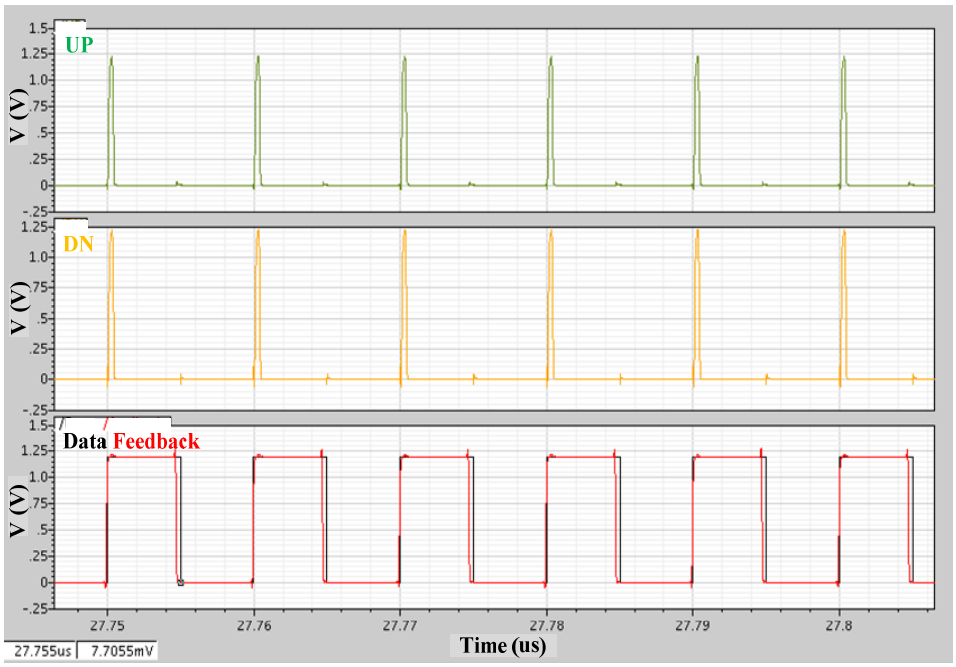


(a)

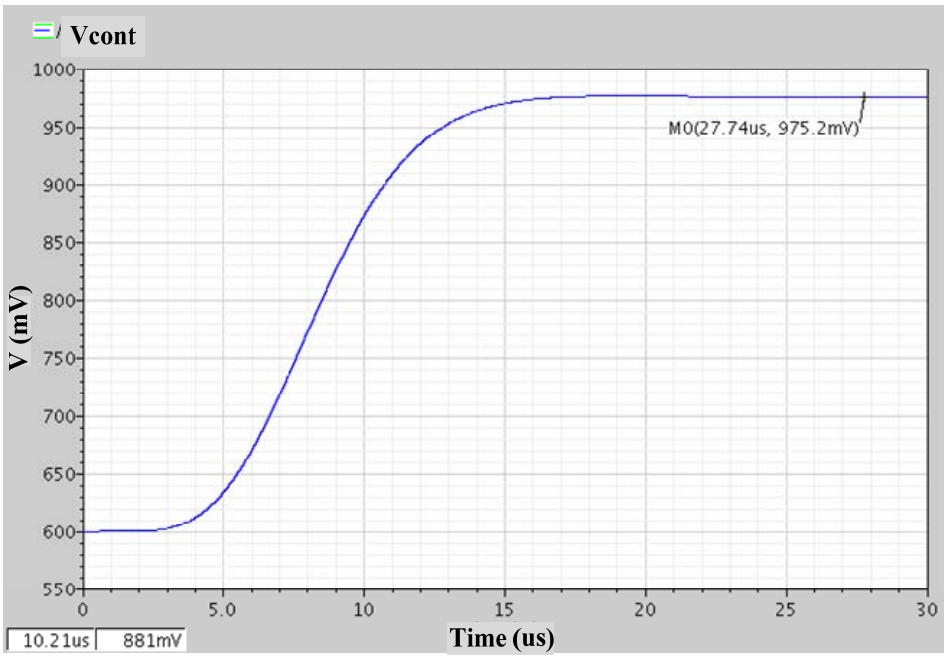


(b)

Fig. 4.5: Steady-state behavior of the 5th-order DLL with fc=28.6MHz, K_{dl}= 9.2833 V/rad, K_P=0.0805 mV/rad, and BW=50 kHz. (a) input-output waveforms at lock, (b) control voltage during settling process.



(a)



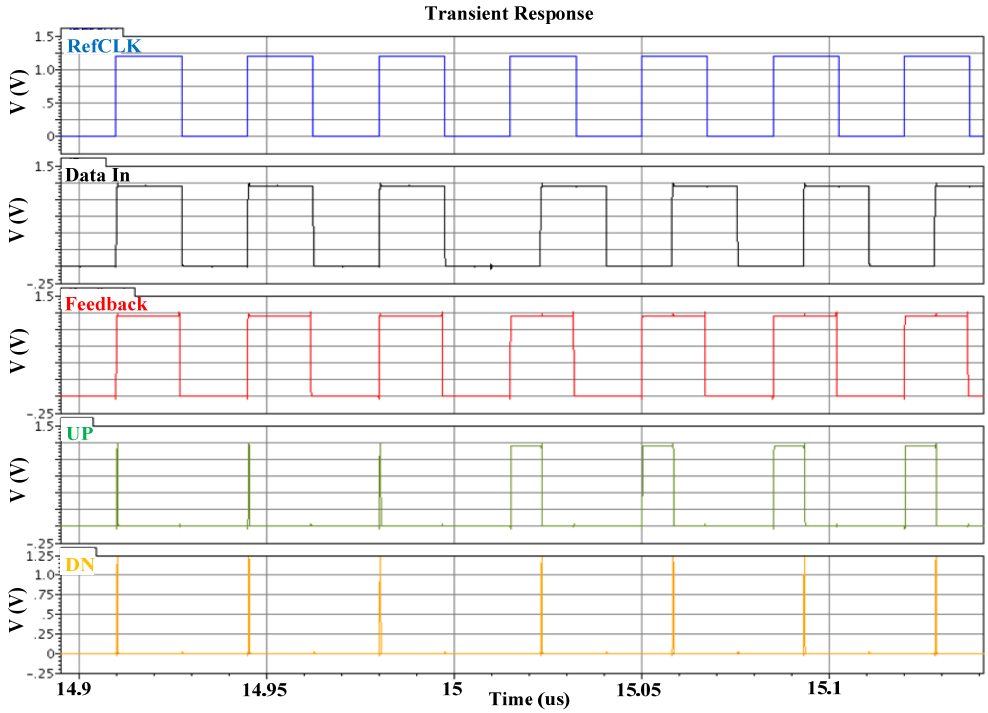
(b)

Fig. 4.6: Steady-state behavior of the 7th-order DLL with fc=100MHz, K_{d4}=7.6567 V/rad, K_P=0.0805 mV/rad and BW=50 kHz: (a) input-output waveforms at lock, (b) control voltage during settling process.

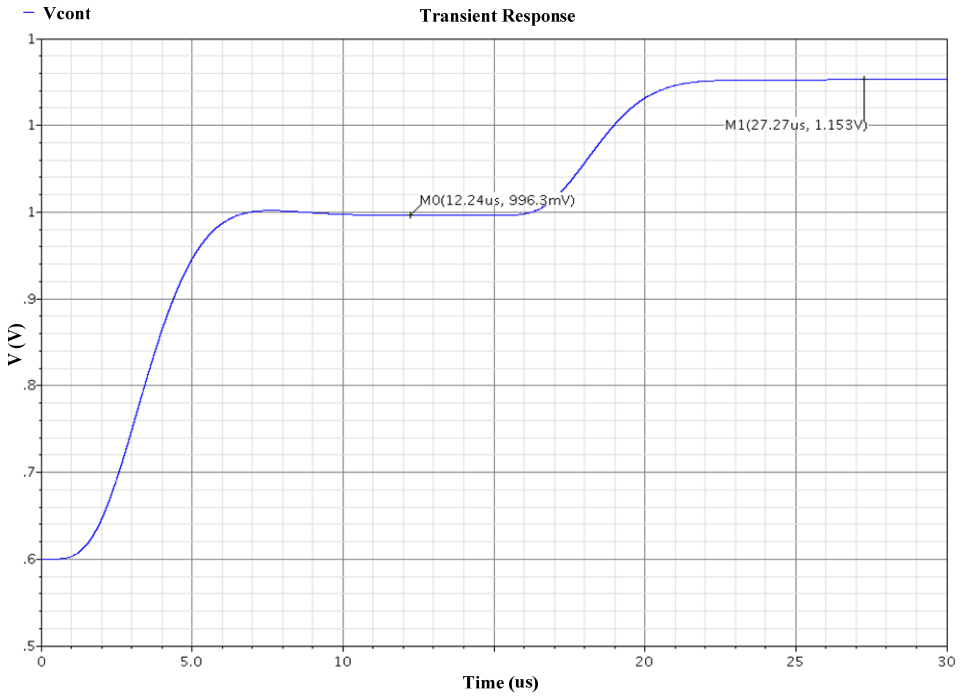
4.3.2 Dynamic-State Simulations

To evaluate the dynamic operation of the DLL subject to time changing input conditions, the DLL is initially operating in steady state and then force to leave this point through a phase step change at the input. Figure 4.7(a) shows the waveforms of the 5th-order DLL subject to a

+1.5708 radians (90 degree) phase step change in the incoming data stream. Here the data input (black) has a +1.5708 radians phase change at around 15 μs, causing the UP signal to become wide again in order to relock to the new data edge. Accordingly, the VCDL control voltage in Fig. 4.7 (b) starts to rise again from 15 us until the settling to the new equilibrium point. During the whole dynamic process, the DLL is working within its linear region C. Figure 4.8 gives an example when the 8th-order DLL has a -1.5708 radians (-90 degree) phase step change in its data stream at around 15 μs. In this case, the DN signal becomes wide again. Figure 4.8(b) illustrates that the VCDL control voltage starts to decrease until the settling to the new equilibrium point. The DLL also works within its linear region (specifically the C region) during this phase step change.

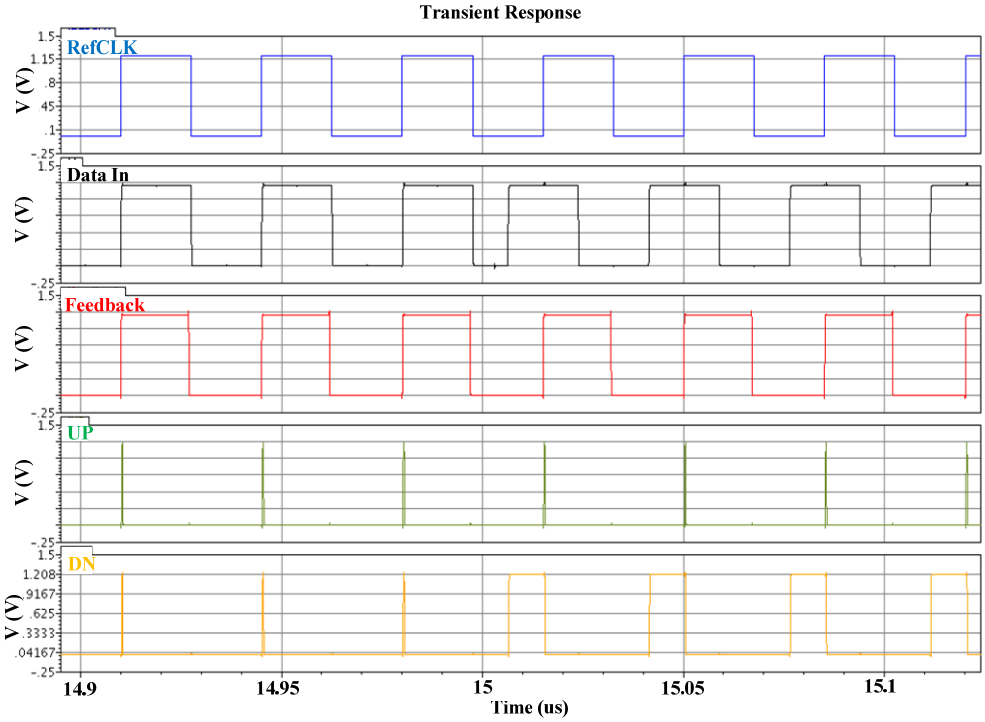


(a)

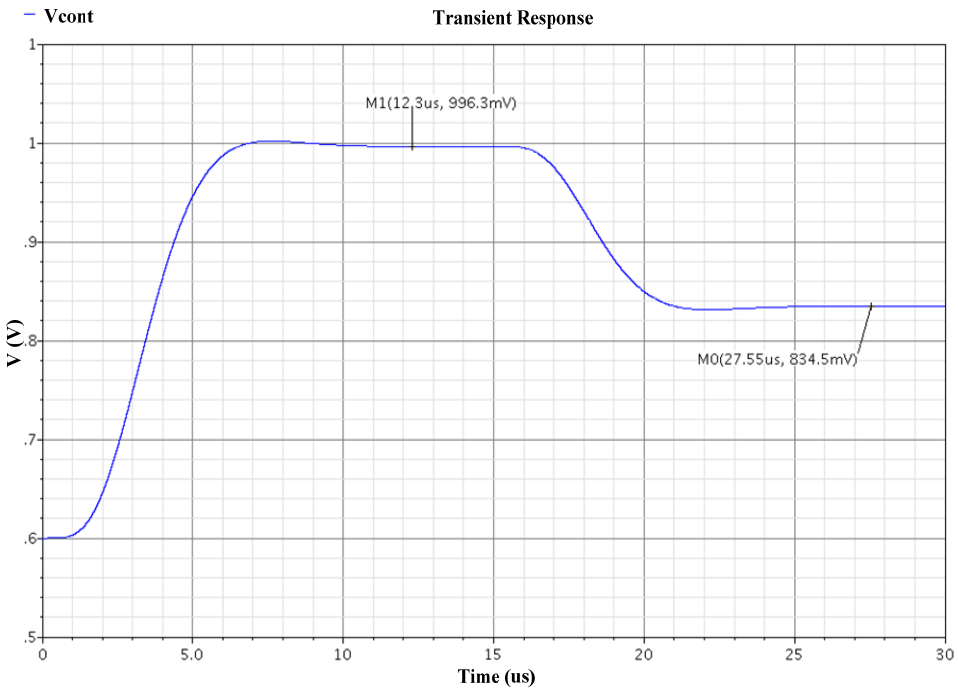


(b)

Fig. 4.7: Dynamic behvaior of the 5th-order DLL with specifications that fc=28.6MHz, K_{dl}=9.2833 V/rad, K_P=0.0805 mV/rad, BW=100 kHz. (a) Wave form at +1.4362 rad phase step change. (b) Control voltage during settling process.



(a)



(b)

Fig. 4.8: Dynamics of the 8th-order DLL with specifications that fc=28.6MHz, K_{d1}=9.2833 V/rad, K_P=0.0805 mV/rad, BW=100 kHz. (a) Wave form at +1.5708 rad phase step change. (b) Control voltage during settling process.

4.3.3 Comparison to Theoretical Values

The transient response of the control voltage at the input of VCDL subject to a 1.5708-radian (90 degree) phase step change is compared to their theoretical values. Figure 4.9 shows the control voltages for the 2nd to 8th order DLLs with specifications corresponding to fc=28.6MHz, K_{d1}=9.2833 V/rad, K_P=0.16 mV/rad and BW=50 kHz. Here we see the simulations results match perfectly with that provided by the theory of Chapter 2. Figure 4.10 shows the control voltages for the 2nd to 8th order DLL with specifications corresponding to fc=100 MHz, K_{d4}=7.6567 V/rad, K_P=0.16 mV/rad and BW=50 kHz. Again, the results match those predicted by theory.

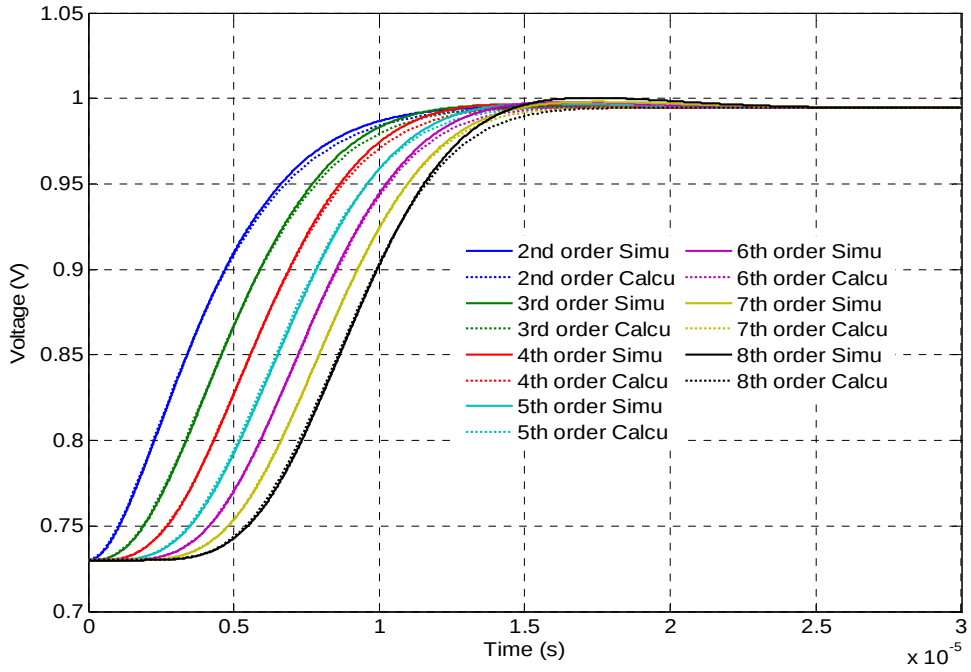


Fig. 4.9: Comparison of the simulated transient responses of the 5th-order DLL with their theoretical values.

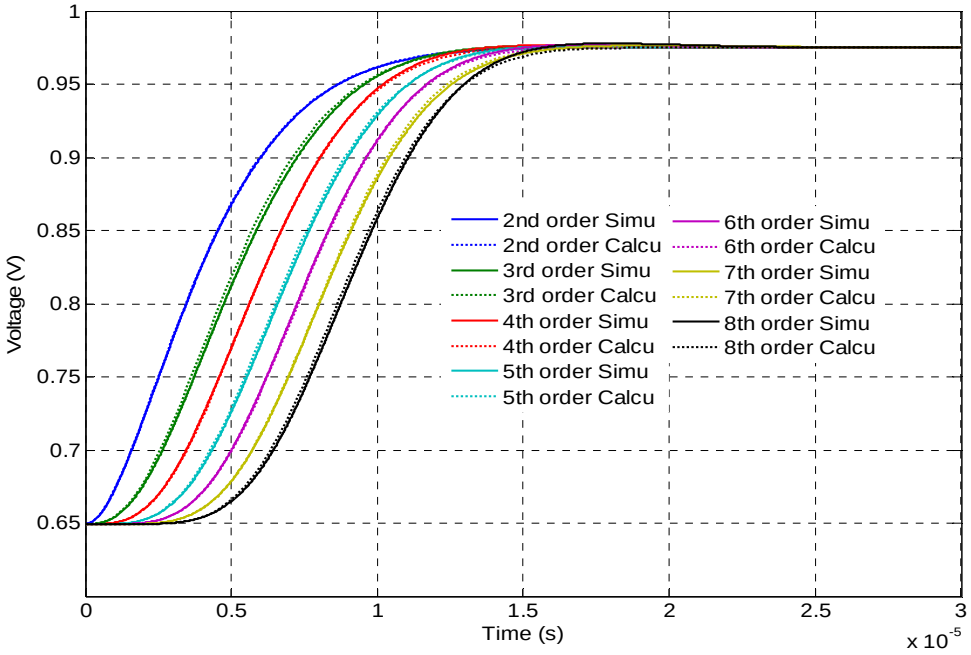


Fig. 4.10: Comparison of the simulated transient responses of the 5th-order DLL with their theoretical values.

4.4 Summary

In this chapter, the organizations of the main building blocks were introduced as well as some of the ancillary circuits used in the IC design. The chip layout as drawn using the

Cadence layout tool was provided and the building blocks identified. Physical dimensions of the IC was also provided. Finally, the chip-level simulations results were given and compared to their theoretical values. The simulations results proved the correctness of the circuit level design, as well as the theory presented in Chapter 2.

Chapter 5

Experimental Results

This chapter will present the testing results obtained from a PCB implementation containing the custom CMOS IC chip and an active-RC loop filter circuit, which together realizes a high-order programmable frequency-selective DLL circuit. The DLL IC was fabricated in a 130 nm CMOS technology from TSMC and housed in a SOIC 24-pin package. The IC was solder onto a custom PCB and interfaced to a programmable 2^{nd} to 8^{th} order loop filter. This chapter will analyze the measured characteristics of the DLL and compared them to simulation and theoretical values. Some conclusions will be provided for future improvements to the chip.

5.1 PCB for Test Interfacing

A four-layer PCB using FR4 dielectric material was designed to provide power delivery to all ICs required by the programmable DLL as well support the off-chip active loop filter circuits, as well as any other ancillary circuits used for testing purpose such as SMA connectors, protection diodes, power regulators, etc.. The PCB layout is shown in Fig. 5.1 and its complete schematics are provided in Appendix D. The PCB was designed using the Altium PCB Designer software. The PCB has dimensions of $16\ \text{cm} \times 10.5\ \text{cm}$ and, as mentioned above, consists of four metal layers. The top and the bottom layers were used to route power and signals between ICs and various passive components, and the other two layers are used for power delivery and grounds, respectively. The board is split into a digital part on the left side and the analog part on the right side. A photograph of the front and back side of the PCB are shown in Fig. 5.2 and Fig. 5.3, respectively.

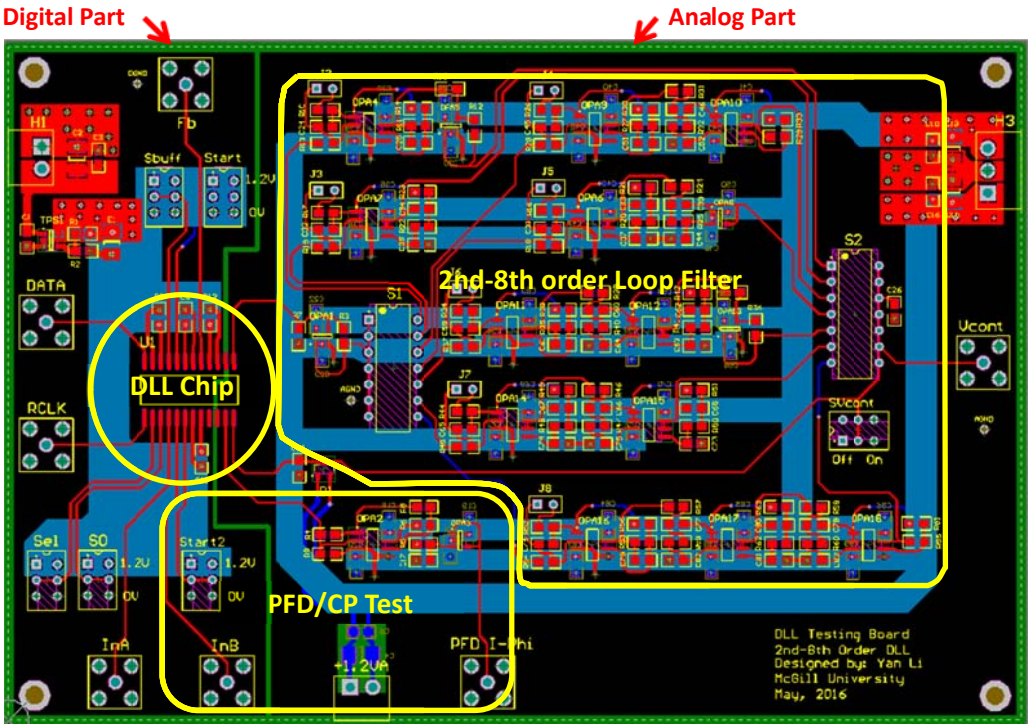


Fig. 5.1: The PCB layout and component placement diagram.

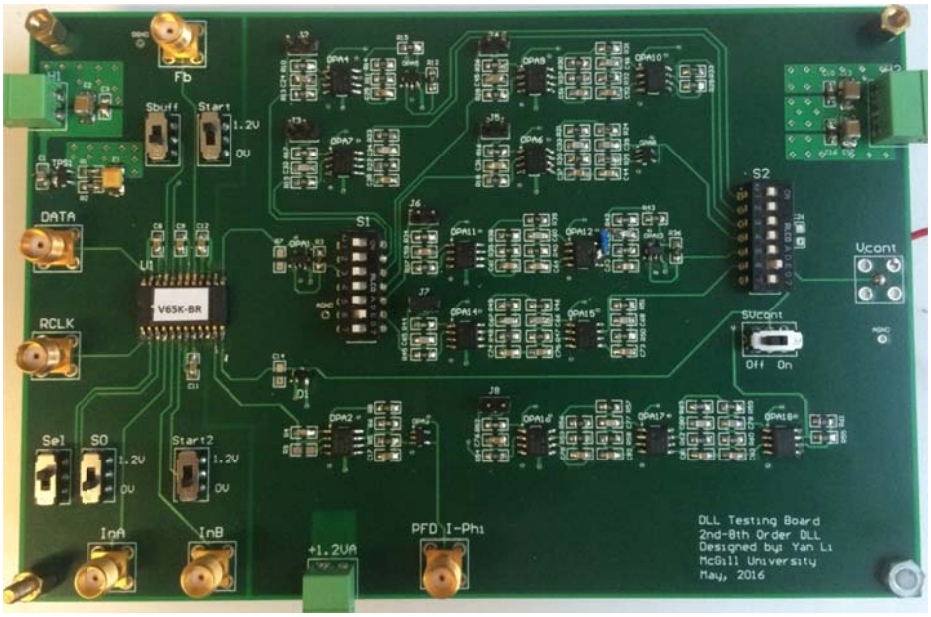


Fig. 5.2: Front view of DLL testing PCB.

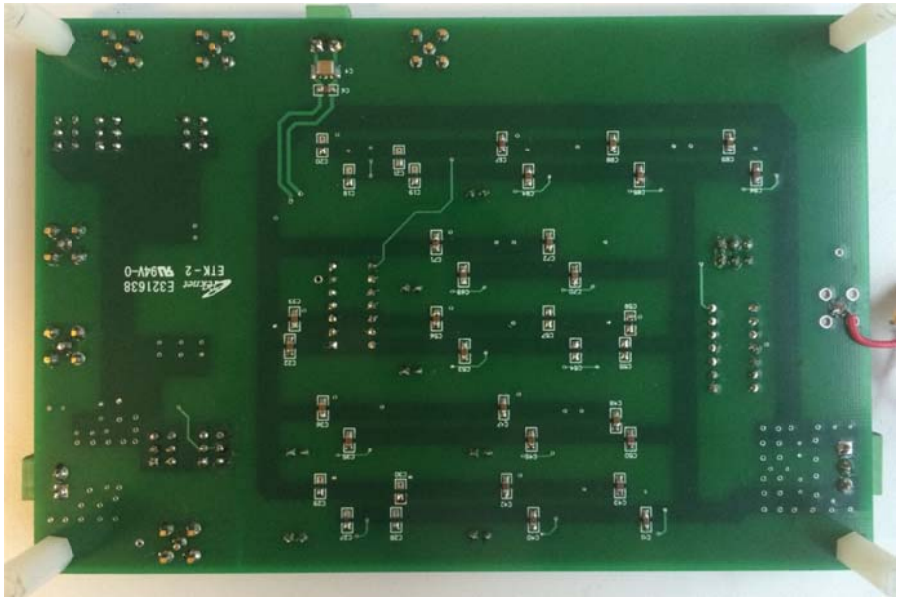


Fig. 5.3: Back view of DLL testing PCB.

5.1.1 Power Delivery Network

To reduce the overall noise level on the PCB, the board is designed to separate the digital and analog signals by splitting the ground layer into a digital ground (0 V) and an analog ground (0.6 V) by removing a portion of the copper shared between them. The power layer is split into five parts to generate the +2.5 V digital power, +1.2 V digital power, +1.2 V analog power and +/- 2.5 V analog power. Among the power supplies, the +1.2 V digital power required by the DLL IC is transferred from the +2.5 V digital power using a linear drop-out regulator LDO (tps76201) in order to further decrease the noise coupled from the digital side. The +1.2 V analog power drives a pair of Schottky diodes (BAT54S) used as a protection circuit for the IC pin connected to VCDL control voltage. The +/- 2.5 V analog power and the 0.6 V analog ground together drive the op-amps in the loop filter circuits and the PFD and CP testing circuit. The four-layer stack up is illustrated in Fig. 5.4.

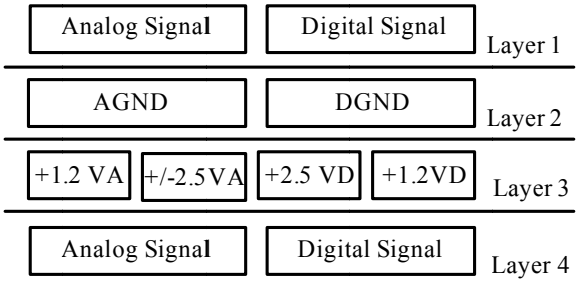


Fig. 5.4: The PCB stack up.

5.1.2 Signal Wiring Dimensions

The routing of the wires is designed to match 50Ω impedances. There are a number of factors that influence the impedance of the signal routing, including the physical dimensions and properties of the materials used to fabricate the PCB. The formula that the Altium PCB Designer software uses to calculate the routing impedance of the traces in Fig. 5.1 is as follows

$$Zo = \left(87 / \sqrt{(Er + 1.41)}\right) \times \ln\left(5.98 \times D / (0.8 \times W + H)\right) \tag{5.1}$$

where Zo is the trace impedance, Er is the dielectric constant of the dielectric material (FR-4 in this case), D is the plane-to-plane distance, W is the trace width and H is the trace height. A diagram illustrating the dimensions of the trace-to-ground plane distance is shown in Fig. 5.5. When $Er = 4.8$, $H = 1.4$ mil (1 oz copper) and $D = 10$ mil, according to Eqn. (5.1), the trace width W is calculated to be approximately 15 mil if we need Zo to be around 50Ω. Therefore, the signal wires on PCB are chosen to be 15 mil wide.

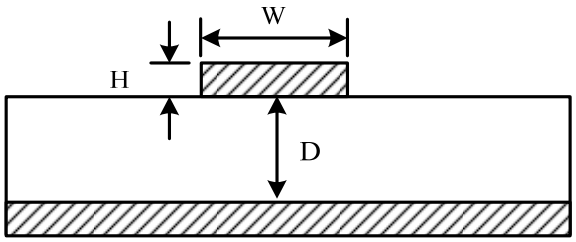


Fig. 5.5: Trace to plane dimensions

5.1.3 Digital Side

The digital side contains the DLL IC and its power supplies, control switches as well as the digital signal cable connectors (SMAs). The +1.2 V digital power is generated by an LDO (tps76201) from the +2.5 V digital supply. The SMA "DATA" and "RCLK" are input cable connectors from which the DLL input signals "data" and "reference clock" are applied to. The SMA "Fb" is the output connector from with the DLL output/feedback signal could be observed and analyzed. The switches "Start", "Sbuff", "Sel" and "S0" could switch between 0 V and 1.2 V in order to provide control signals to the DLL. Besides the main DLL circuit, there is also a duplicated PFD/CP circuit on the IC that is intended for characterization purposes. The SMA "InA" and "InB" are used as inputs to this PFD/CP circuit and the switch "Start2" provides the start-up signal for this circuit.

5.1.4 Analog Side

The 2nd to 8th order loop filters are realized by discrete op amps and located on the analog side of the PCB. According to the characteristics of the op amp ICs on market, the OPA536 from Texas Instruments was best chosen for this application. It has a 200 MHz gain-bandwidth product and a rail-to-rail output swing, which was considered good enough for this design. The +/-2.5 V analog power and the analog ground (+0.6 V) are used to power the op-amps. The loop filters all share a common I-V circuit connected directly to the output pin of the charge pump. A set of 7:1/1:7 switches (S1 and S2 in PCB layout) allow individual selection of one of the seven loop filters. The SMA "Vcont" is the observation point for the VCDL control voltage.

There is an extra circuit added in the integrator stage of each filter to help with the start-up process, the schematic of which is shown in Fig. 5.6. A switch Sj and a small value resistor Rj are added in the integrator stage in order to short out the capacitor when the DLL loop is still open. Without this switch, the capacitor keeps charging as soon as the power is on, forcing the VCDL control voltage to start from either +2.5 V or -2.5 V. This may cause trouble when we want to close the loop. Instead, the control voltage is set to start from 0.6 V (analog ground) with the help of this switch. After the DLL loop is closed, the switch could be turned off again and DLL would work normally without any inference from the added switch-resistor combination. This switch is realized by a simple jumper and resistor Rj was set to 50Ω.

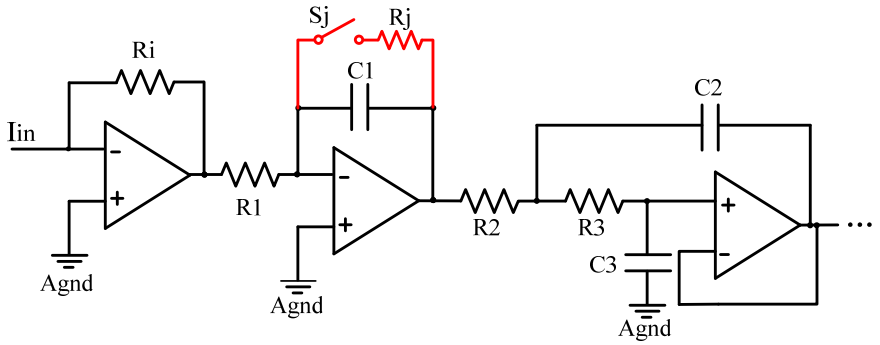


Fig. 5.6: Schematic of the start-up circuit in loop filter.

The testing circuit for the PFD/CP is also located on analog side of the PCB. It is used to observe the PFD/CP output characteristic curve (the output current vs. the input phase difference). Since a voltage signal is easier to observe than a current signal, the testing circuit

is composed of an I-V circuit, a simple 1st-order low pass filter and a voltage follower as shown in Fig. 5.7. By observing the output voltage Vout with respect to the input phase differences, the transfer characteristics of the PFD could be obtained. The component list for this circuit is provided in Table 5.1. The SMA "PFD I-Phi" on PCB is the observation node for this output voltage.

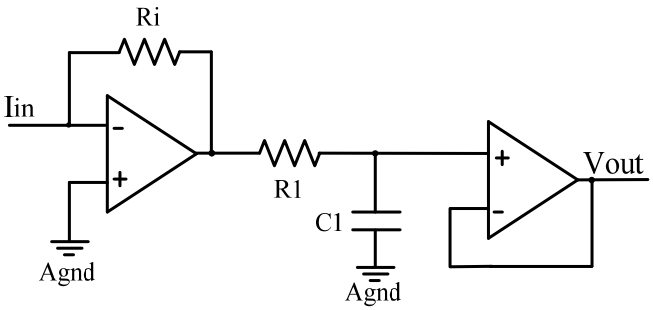


Fig. 5.7: Schematic of the PFD/CP testing circuit.

Table 5.1: Parameters of R & C

Transistor	Parameter
Ri	470 Ohms
R1	1k
C1	220p

5.1.5 Additional Protection Circuits on PCB

Since the power levels on analog side are higher than those on the digital side, specifically 0V/1.2V, there is a chances that the control voltage will go higher than these levels when driving the VCDL To protect the input pin of the DLL IC in case of this situation, a pair of Schottky Diodes (BAT54S) is connected between the output of the loop filter and the input pin of the DLL. The diodes are connected between the +1.2 V analog power level and analog ground, as shown in Fig. 5.8.

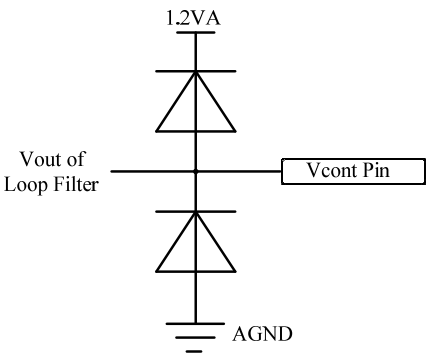


Fig. 5.8: Protection circuit for control voltage.

Finally, the components/ICs used on PCB are listed in Table 5.2.

Table 5.2: Components used on PCB

Type	Name	Cop.	Number
LDO	TPS76201	TI	1
DLL IC	ICGMGYL2	CMC	1
Op amp	OPA536	TI	6
Op amp	OPA2536	TI	13
Schottky Diodes	BAT54S	Vishay	1
Header2	-	-	2
Header3	-	-	1
Jumper	-	-	7
1:2 Switch	-	-	6
7:1 Switch	-	-	2
SMA	-	-	7

5.2 Test Setup

A block diagram of the DLL test setup is shown in Fig. 5.9. It consists of four main parts: the HP 81130A pattern generator (or the Teradyne FLEX mixed-signal tester) with two channel outputs to generate the DLL input data and clock signals, the PCB interface board with the DLL circuit and other testing circuits on it, the DC power supplies, and the Agilent DSA80000B digital signal analyzer.

The HP 81130A pattern generator was used to generate two digital channels. One channel is used to carry a continuous-time pulses acting as the clock signal, and the other channel contained a programmable digital pattern. The signals are set to 1.2 V with a duty cycle of 50%, the frequencies are switching among 28.6 MHz, 50 MHz, 62.5 MHz and 100 MHz and the delay time is set according to the testing requirements. The Agilent DSA80000B digital signal analyzer has up to 40 GSa/s sample rate and could measure a maximum 13 GHz signal. It is used to analyze the output signals from PCB such as the DLL output/feedback signal, the VCDL control voltage and output voltage from the PFD and CP testing circuit. Four DC power supplies are used providing the +2.5 VD, +1.2 VA, +/-2.5 VA and +0.6 V AGND power levels needed for the circuits.

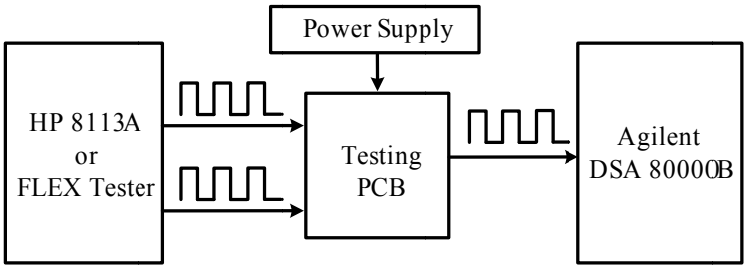


Fig. 5.9: Block diagram of overall testing setup.

The DLL IC was built on a 1 mm x 1 mm die using Cadence 130 nm technology. It was then packaged in a SOIC-24 package. The view of the chip die is shown in Fig. 5.10 and the connection of the IC to its packaging is shown in Fig. 5.11. As discussed previously, the DLL chip contains the PFD, CP, multi-controlled VCDL and the input/output buffers on-chip and 2^{nd}-8^{th} order loop filters are built off-chip. The names and descriptions of IC pins are listed in Appendix C. The experimental setup is shown in Fig. 5.12.

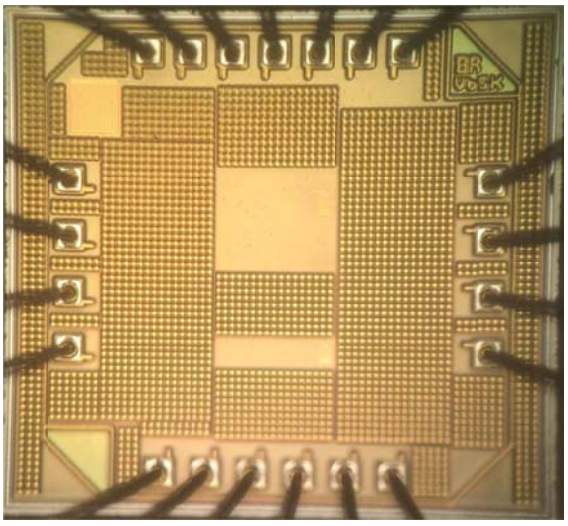


Fig. 5.10: A microphotograph of the DLL die with bonding wires shown.

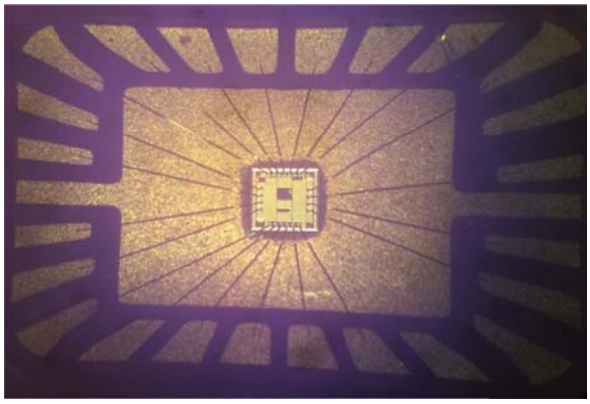


Fig. 5.11: A top-level view of the DLL chip die sitting in its SOIC-24 package.

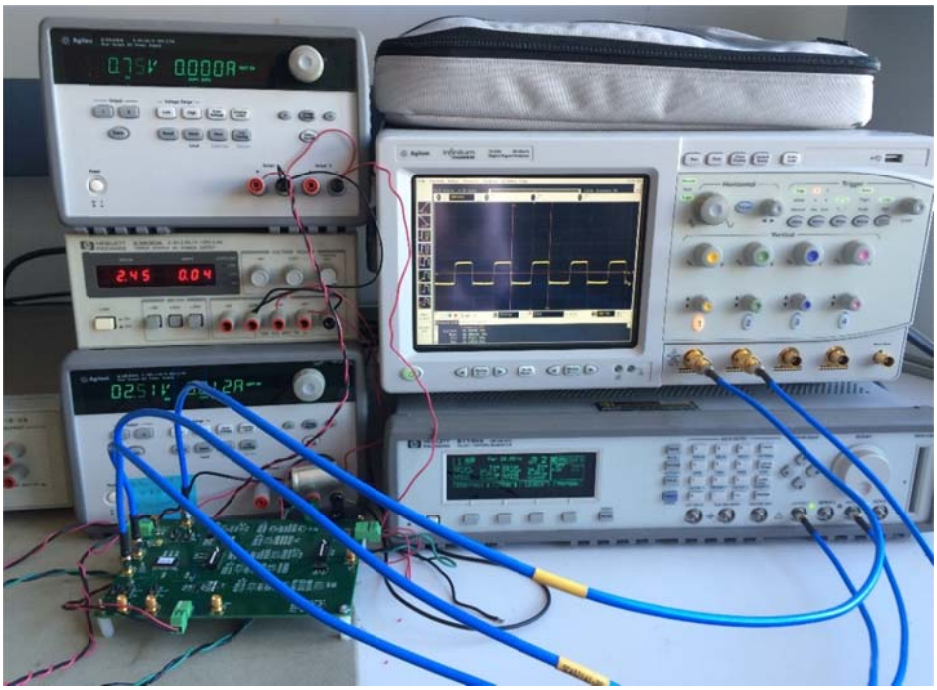


Fig. 5.12: Experimental setup

5.3 Test Results

5.3.1 PFD and VCDL Test Results

The test process begins on the first silicon by ensuring that the major components within the DLL IC are functioning correctly. During these "open-loop" tests, the MUX switches "S1" and "S2" on PCB are turn off thereby removing the loop filter from the feedback loop.

Since the PFD and CP in main DLL circuit are connected with other circuits and difficult to isolate, the characteristics of the PFD and CP are obtained by testing the second PFD/CP 2 circuit which is exactly the same as that in main DLL circuit. The output voltages at SMA "PFD I-Phi" are observed with respect to the phase differences of the two input signals: InA and InB. Signals with a variety of frequencies are also added to the PFD inputs to reflect the PFD working range under operating conditions. The test results of the PFD/CP transfer characteristic curves are shown Fig. 5.13 for input clock frequencies ranging from 33 MHz to 666.7 MHz. The stars located on each curve are the actual data point. We can see that when the input frequency is low, e.g., lower than 100 MHz, the PFD linear operating range is approximately $(-2\pi, 2\pi)$. However, the PFD linear operating range decreases with the increasing input clock frequency; under the highest input clock condition of 666.7 MHz, the range reduces to $(-1.2\pi, 1.2\pi)$. By comparing the test results in Fig. 5.13 with the

simulations results displayed previously in Fig. 3.6, one would see that the curves corresponding to the appropriate clock frequency match quite well except for a scale factor on the y-axis. This provides proof that the PFD/CP circuits are functioning correctly. We can only surmise that the PFD/CP circuits intended for closed-loop operation would exhibit similar behavior.

The multi-controlled VCDL was also tested under open-loop condition. The four delay lines are controlled by two control signals "Sel" and "S0". During this test, the VCDL control voltage is provided by a separate DC power supply and the delay times between the reference clock and the output of the VCDL are recorded. The VCDL characteristic curves are shown in Fig. 5.14 and they are compared with the previous simulation results (shown with dashed lines). Here we see that the delay line work as expected; they have similar delay times, linear shapes and slopes as the simulation results; although, the two sets of results do not exactly. These differences are attributed to the large output buffer placed between the VCDL output and the output pin of the IC, as well as process variations experienced during IC fabrication. As discussed previously, the output buffer does not need to be so large to observe the output signals. In future designs, the buffer size should be decreased to reduce its influences on the overall propagation delay of the VCDL circuit.

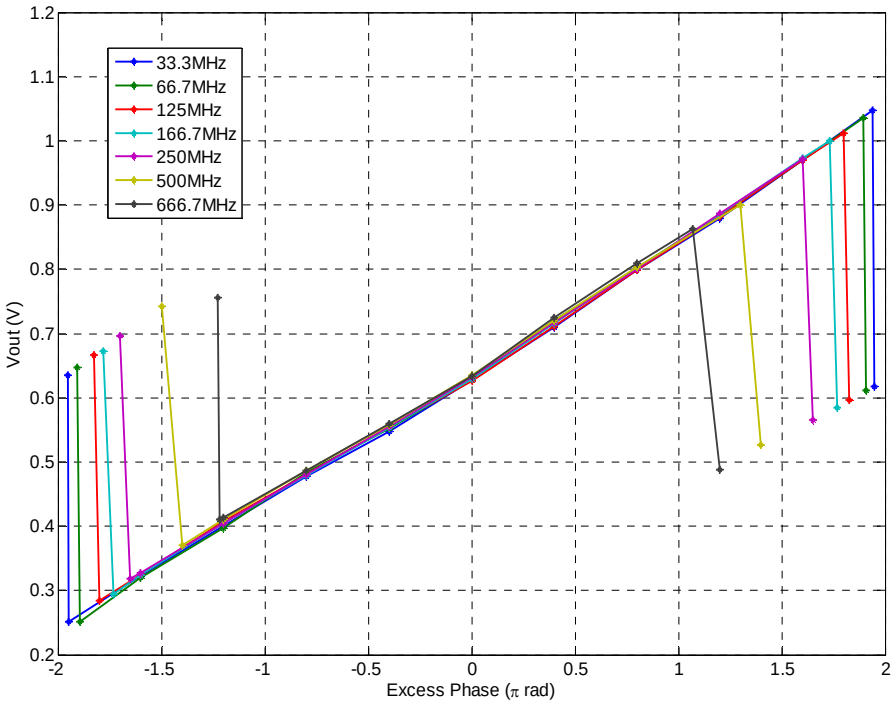


Fig. 5.13: Output characteristic of the PFD/CP circuit.

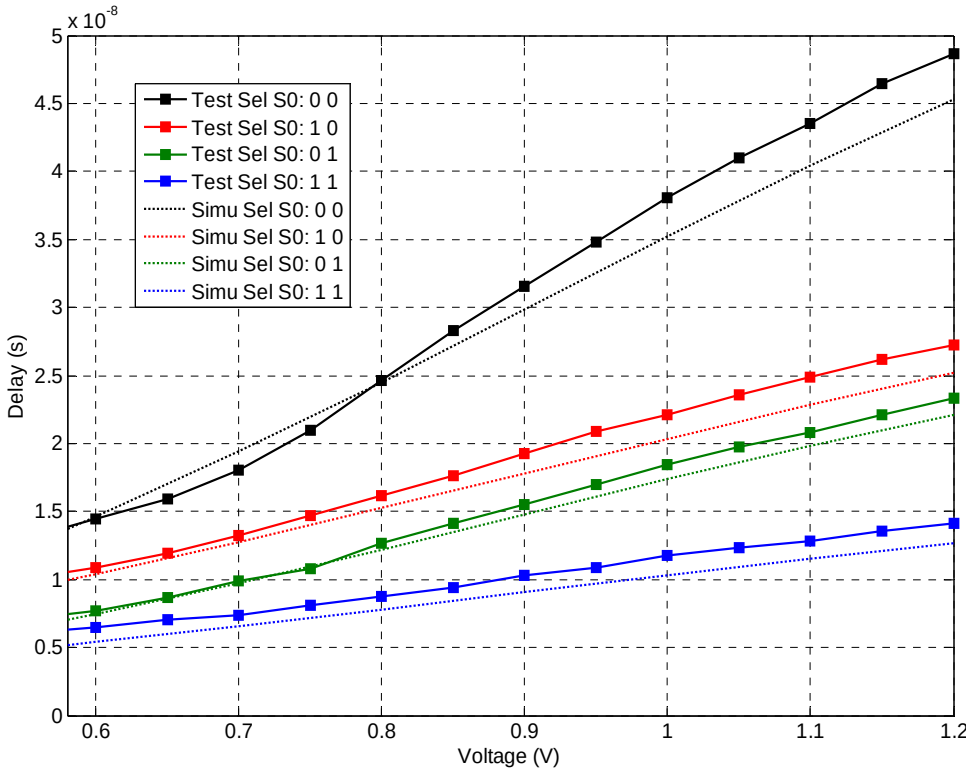


Fig. 5.14: Output characteristics of the VCDL circuit compared with the simulation results.

5.3.2 Time-Domain Test Results

The DLL closed-loop characteristics are investigated in this subsection. To remind the reader, the specifications of the DLL with orders ranging from 2 to 8 are: fc=50 MHz, K_p=0.0805 mV/rad, K_{d2}=7.81 V/rad and BW= 50 kHz.

As shown in Fig. 2.1, the DLL has two input signals: the input data and the reference clock. The input data signal is set to be exactly the same as the reference clock in this test, i.e., shorted together. Figure 5.15 shows the time-domain output/feedback signal of the 4th-order DLL with an input frequency of 50 MHz compared with the reference clock with the DLL locked. In the plot, the green waveform is the reference clock and the yellow waveform is the output signal. We can see from this result that the mean value of the output frequency is 50.001472 MHz, which is pretty close to expected 50 MHz value**.** The histogram of the phase difference between the reference clock and the output signal is illustrated in Fig. 5.16. As illustrated in the plot, the phase difference has a mean value of 30.11529^o and a standard deviation of 1.916^o.

The cycle-to-cycle jitter is the rms value of the difference between every two consecutive

cycles of the waveform. Figure 5.17 is the measurement of the cycle-to-cycle jitter of the 7th-order DLL with an input frequency of 50 MHz. As shown in the plot, the cycle-to-cycle jitter has a mean value of 3.119 ps and a standard deviation of 266.278 ps.

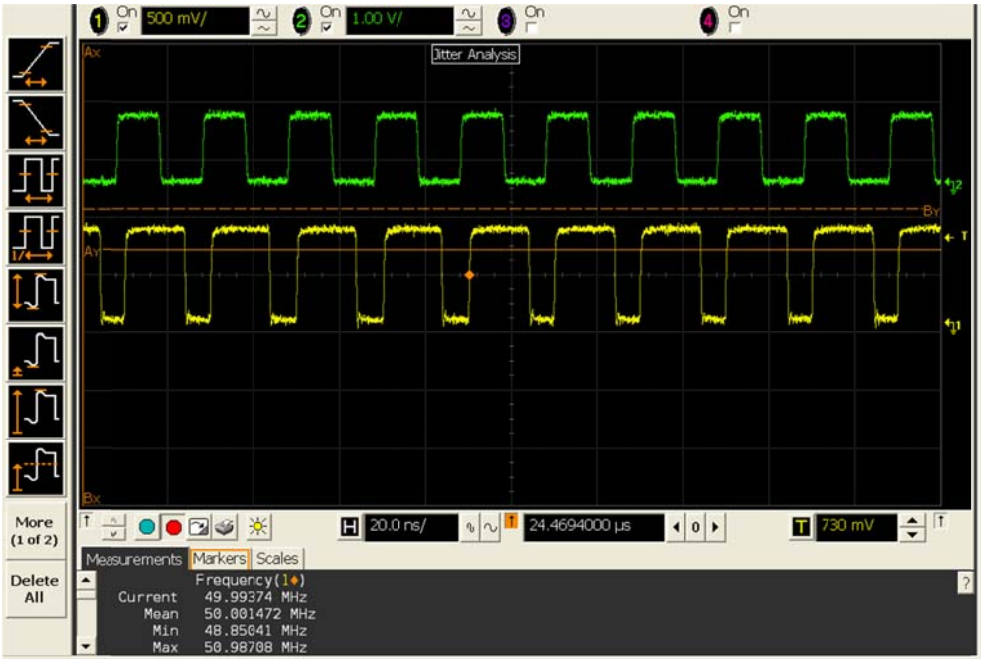


Fig. 5.15: Comparing the reference clock input (green trace) with the output signal from the DLL (yellow trace).

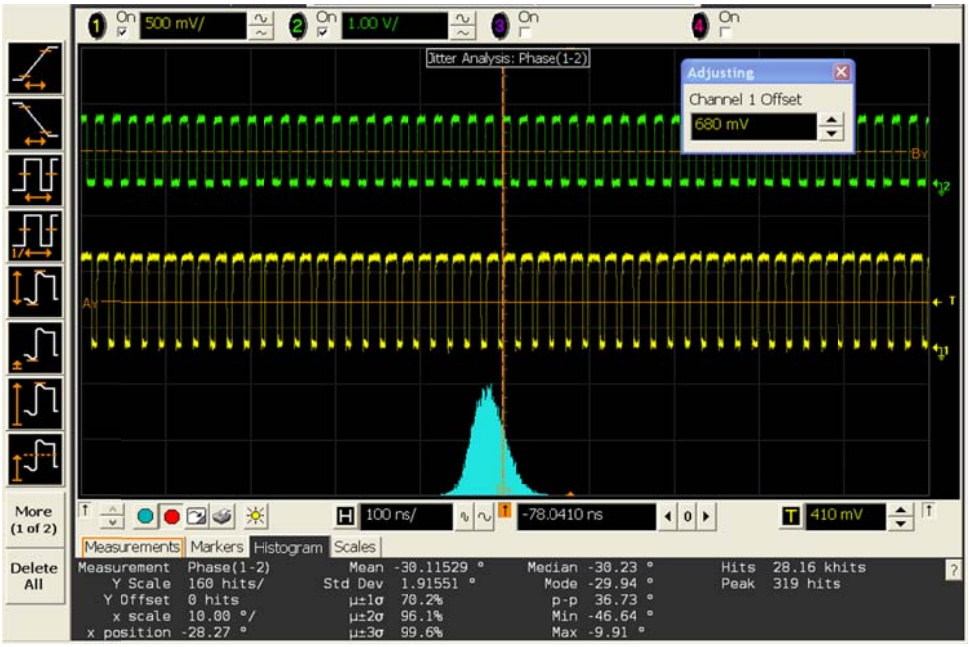


Fig. 5.16: Histogram of phase difference between the reference clock input (green trace) and the output signal from the DLL (yellow trace).

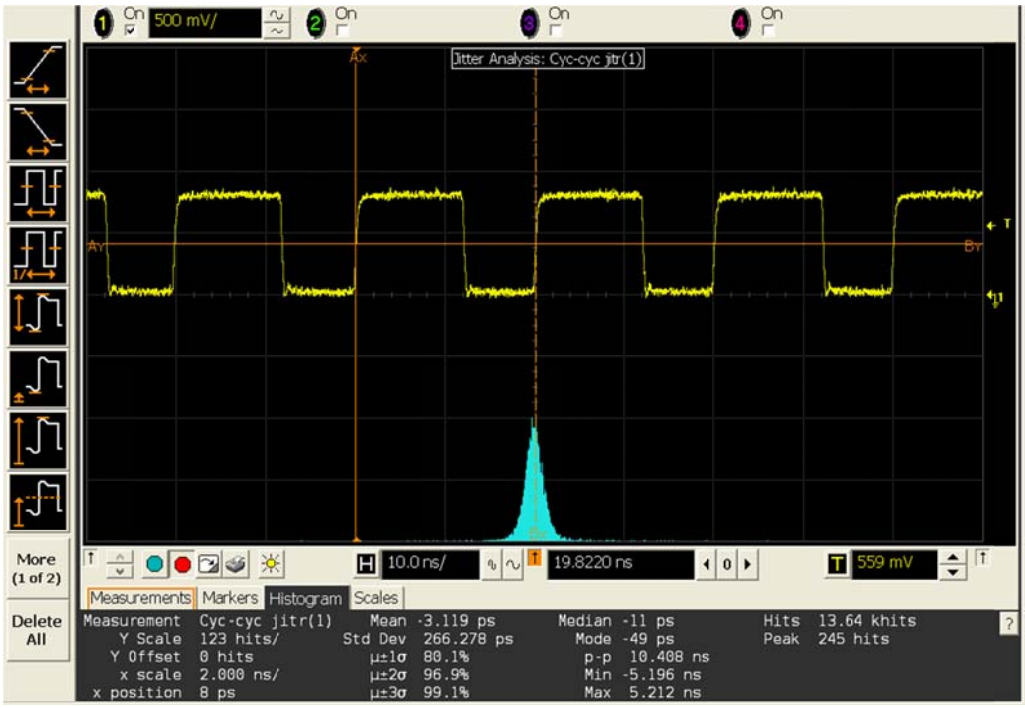


Fig.5.17: Cycle-to-cycle jitter of the 7th-order DLL.

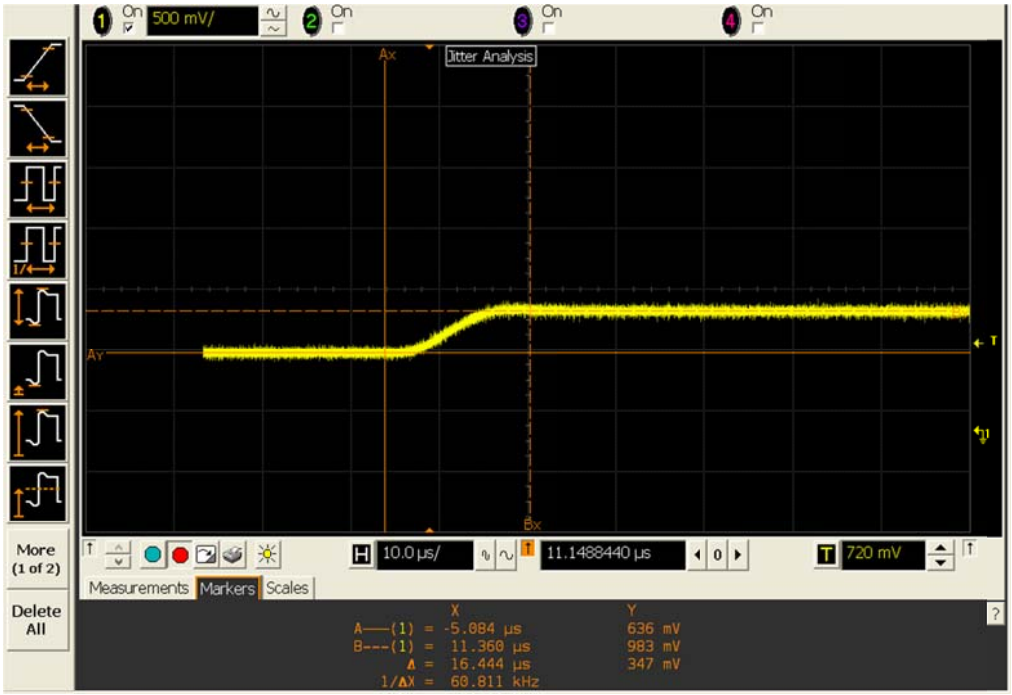


Fig. 5.18: VCDL control voltage of the 4th-order DLL at start-up condition.

During start-up, switch “Start” is turned on and jumper “J1” is turned off to close the loop around the delay line and the loop filter. The VCDL control voltage at the start-up condition is shown in Fig. 5.18. It is an example of the 4th-order DLL with the aforementioned specifications. When the DLL loop is closed, the control voltage starts at the

analog level of 0.6 V and settles to its final value of about 983 mV in about 16 µs. This should be the minimum settling time since the control voltage rises directly to its final value and there is no obvious overshoot in the transient response. This proves that the startup circuit is working as expected. The VCDL control voltage of any other order of DLL at the start-up condition has a similar shape, albeit with different settling times.

The DLL transient responses are measured when the input data signal is subject to a +2.513 radians (8 ns) step change in phase. In this test, the reference clock and data input signals are reference back to the same master clock and both operate at same frequency of 50 MHz. The control voltage of the VCDL is captured and displayed in Figure 5.19. In this particular case, the step response of a 7th-order DLL was measured. The experimental data (red line) is compared to that predicted by theory (black line). There is a very good match between the experimental result and the theoretical curve. Figure 5.20 and 5.21 displays similar measurement results for DLL orders ranging from 2 to 8th order. Specifically, for clarity purposes, to unclutter the graphs, Fig. 5.20 displays the even order results, and Fig. 5.20 displays the odd-order results. In all 7 cases, excellent agreement is between theory and measurement is observed. Any difference is likely due to board parasitics and variations in the SPICE model behavior.

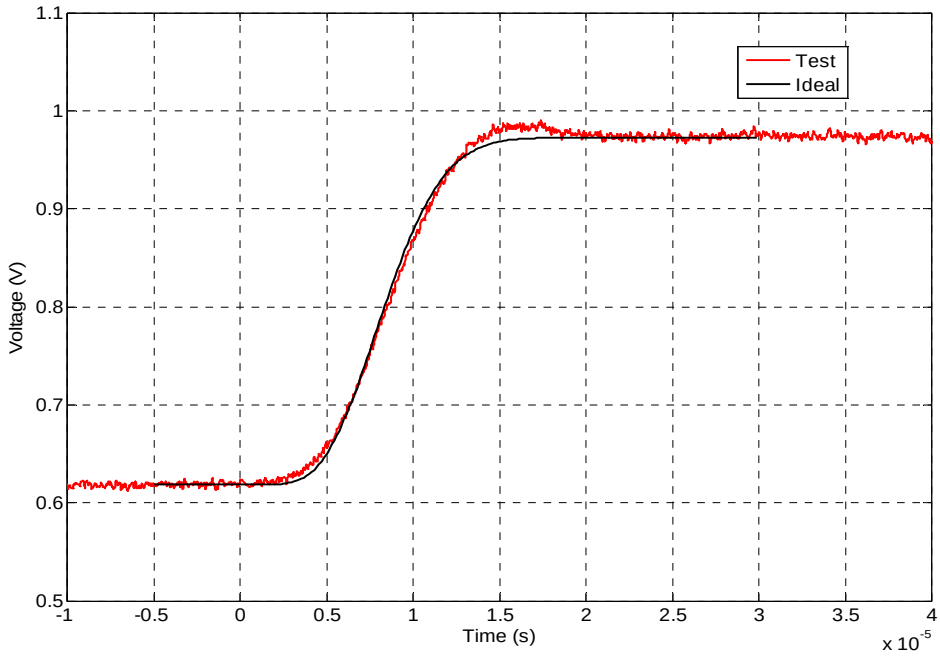


Fig. 5.19: Comparison of the measured VCDL control voltage of the 7th-order DLL with expected values.

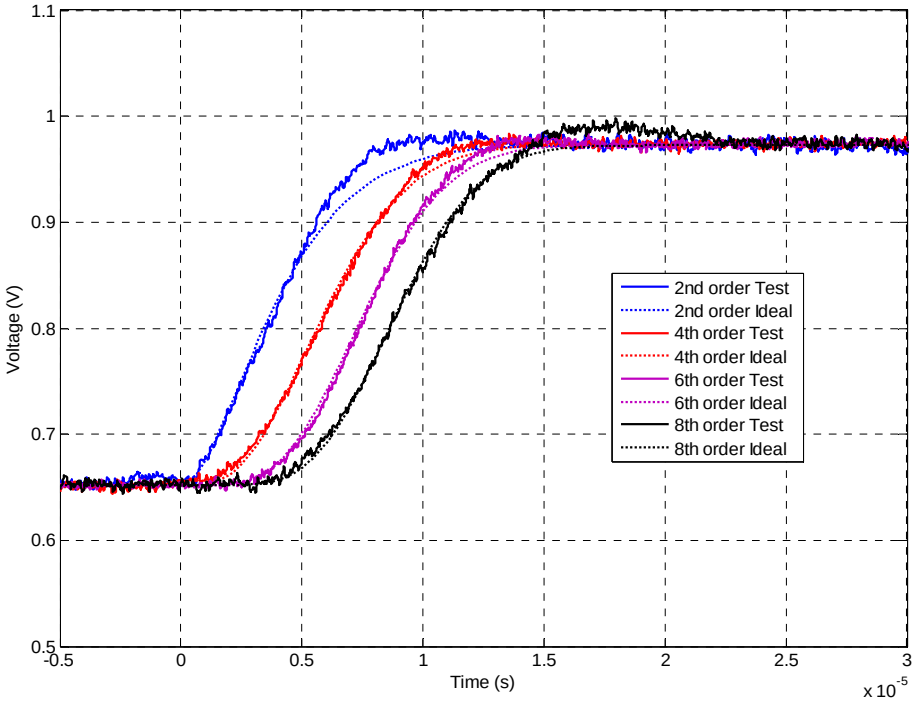


Fig. 5.20: Comparison of the measured VCDL control voltages of DLLs of 2nd, 4th, 6th, 8th order with their expected values.

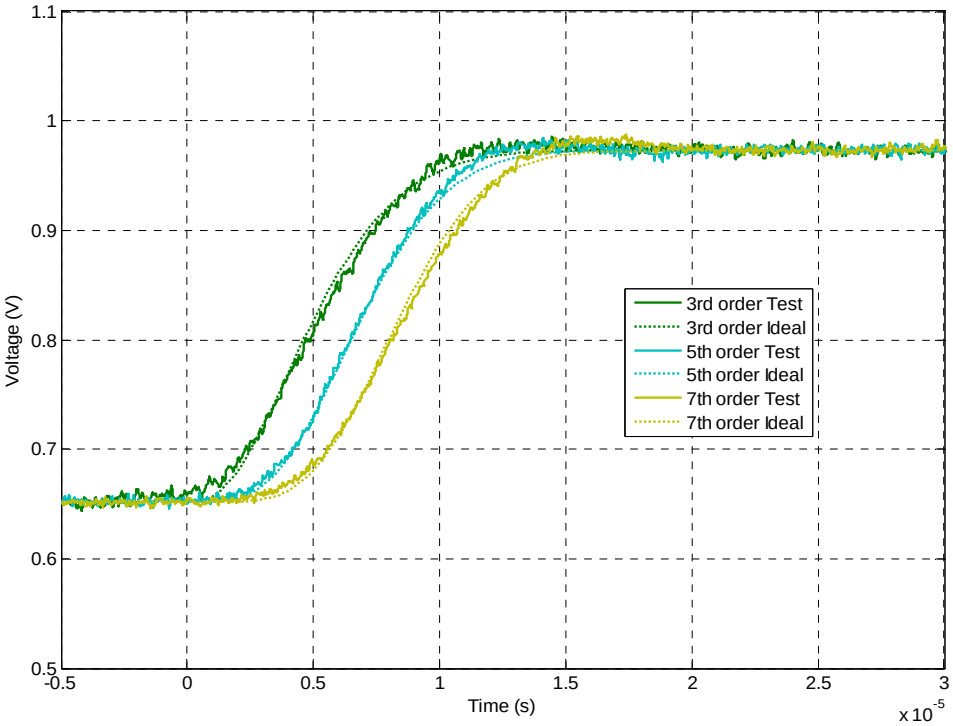


Fig. 5.21: Comparison of the measured VCDL control voltages of DLLs of 3rd, 5th, 7th order with their expected values.

5.3.3 Frequency-Domain Test Results

At this time, we envision that high-order DLLs will mainly be used as frequency selective time-mode filters (TMFs) for sigma-delta applications. As a consequence, the input-output frequency response behavior will be measured and compared to the expected result. In Chapter 2 a multi-tone equal phase signal test method was used to extract the frequency response behavior of the DLL block diagram, while in this experiment a single-tone testing method is used to obtain the frequency response one point at a time. As with the step response test, the VCDL control voltage is observed again. When a sine wave modulated 50 MHz signal is connected to the data input, the VCDL control voltage should also bear a sinusoidal-like characteristic whose frequency is the same as the carrier frequency and the amplitude decreases with the increase of the carrier frequency. This is because the DLL has a low-pass characteristic.

In this experiment, 1-bit sigma-delta encoding theory [12], [13], [14] and the FLEX mixed-signal tester was used to generate a 50 MHz phase modulated data signal and the reference clock. The 1-bit sigma-delta output was be generated by the "DSMOS" toolbox [35] developed for MATLAB with $N = 2^{12} = 2048$, a sampling frequency fs = 50 MHz, and an over sampling ratio of OSR = 94. The encoding process produces a phase-modulated signal with an amplitude of 3 ns corresponding to a phase amplitude of 0.9425 radians at 50 MHz. The frequency of the phase modulated signal was selected to be one of the 8 M values listed in Table 5.3. The 1-bit sigma-delta output is then phase encoded into a four-bit data pattern according to the bit mapping seen listed in Table 5.4. Finally, the 2048 bit data pattern was loaded into FLEX mixed-signal tester to repeatedly generate the sigma-delta encoded phase modulated 50 MHz data signal used to exercise the DLL circuit.

Table 5.3: Correspondence of the number of phase cycles and the frequency of the PM signal

Number of Phase Cycle (M)	Frequency (kHz)
1	12.207
3	36.62
5	61.0351
7	85.45
9	109.86
11	134.28
13	158.7

15	183.1k

Table 5.4: Phase-Encoding Bit-Mapping

Sigma-delta Output	4-bit Phase-Encoding	Phase
1	1100	0º
0	0110	90º

Figures 5.22 to 5.24 illustrate the voltage at the input port to the VCDL subject to three different phase modulated frequencies corresponding to M equal to 1, 5, 11, respectively. The purple curve located on the top portion of the scope display is the actual waveforms appearing at the VCDL input pin. The pink curves shown in the lower portion of the scope display is the same signal passed through a 30 MHz low-pass filter to remove any high-frequency quantization noise created by the sigma-delta encoding process. As expected, the VCDL control voltage has a sine wave shape and its amplitude decreases when the carrier frequency (or the number of periods M) increases. The sinusoidal wave could be totally buried into the noise when its amplitude is less than 10 mV. Therefore, for observation simplicity, M was limited to 15 and never extended beyond this value.

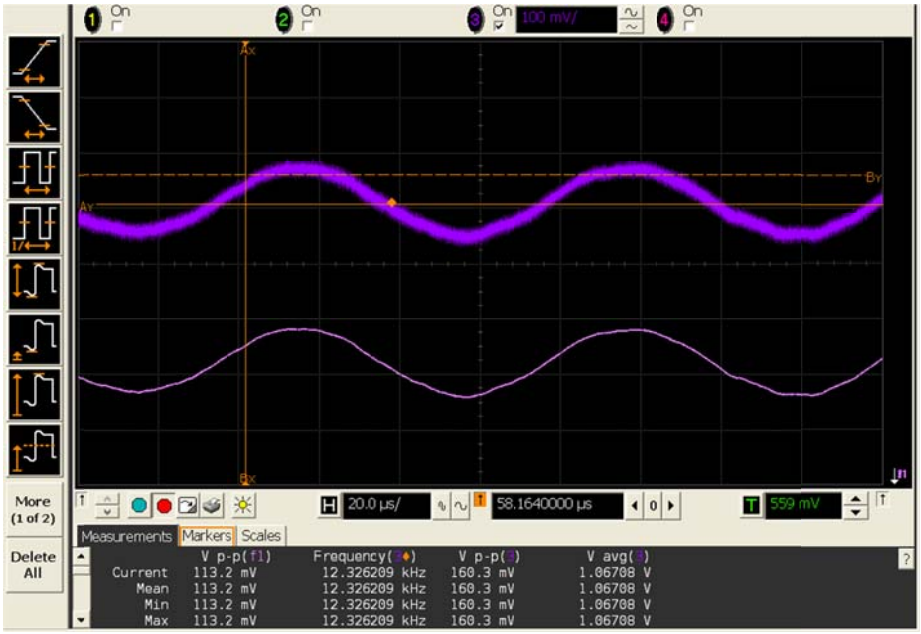


Fig. 5.22 Measured VCDL control voltage for the 8th-order DLL when M=1.

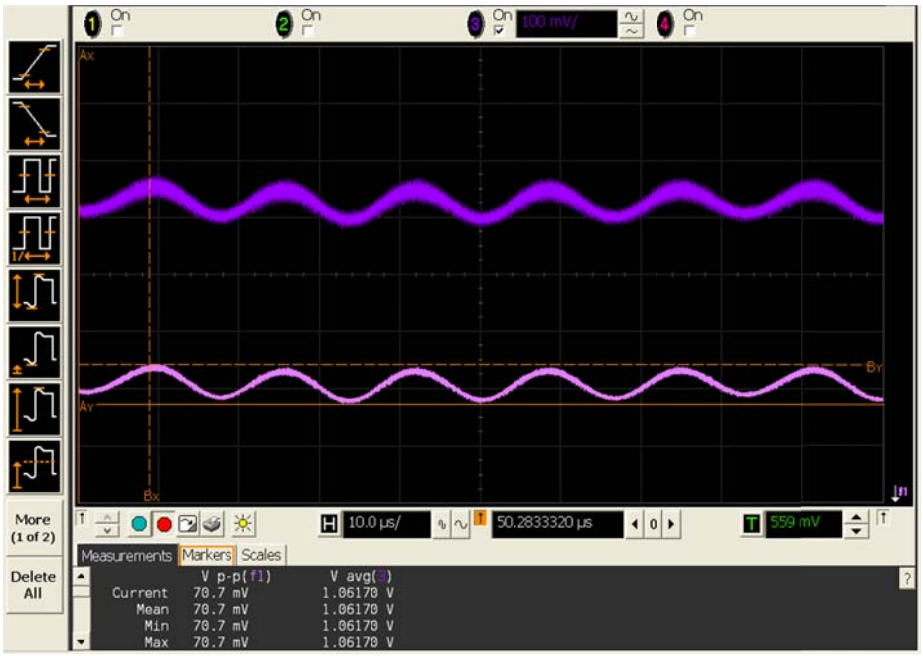


Fig. 5.23 Measured VCDL control voltage for the 7th-order DLL when M=5.

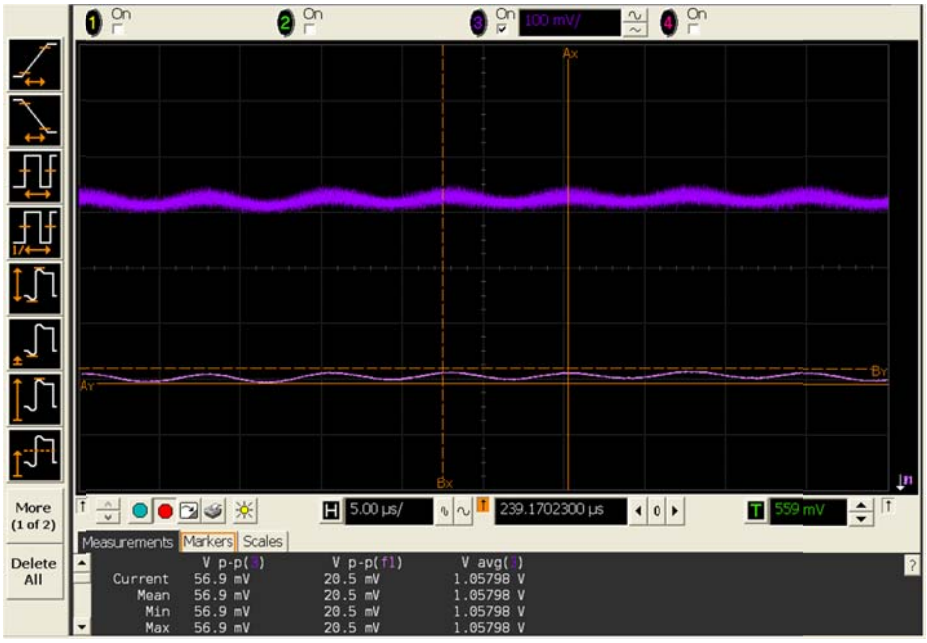


Fig. 5.24 Measured VCDL control voltage for the 3th-order DLL when M=11.

The relationship between the peak-to-peak amplitude of the control voltage and the magnitude of the frequency response is:

$$Mag = 20\log(A_{p-p} \times K_d / A_\phi) \tag{5.2}$$

where Mag is the magnitude in frequency response in dB, A_{p-p} is the peak-to-peak value of VCDL control voltage, K_d is the slope of the VCDL and A_ϕ is the amplitude of the phase-modulated sinusoidal wave. According to our design, $K_d = 25$ ns/V and $A_\phi = 3$ ns,

therefore, M could be calculated when A_{p-p} is observed from the oscilloscope. The tested peak-to-peak values of the VCDL control voltage are listed in Table 5.5 and the corresponding magnitude of frequency response in dB was calculated using Eqn. (5.2).

Table 5.5: Measured peak-to-peak values of VCDL control voltage

Order / M	2	3	4	5	6	7	8
1	116.3	119	120	120	120	120	113.2
3	98.8	100.8	103.4	108	105	100.8	100.2
5	78.8	78.6	78.8	69.5	72	70.7	78.8
7	55.1	53.5	53.5	43.6	42.1	41.7	40.2
9	38.1	30.2	26.3	25.5	24.6	25.6	21.7
11	28.1	20.5	14.5	12.5	12.9	10.9	10.1
13	20.8	13.4	9.8	-	-	-	-
15	15.4	-	-	-	-	-	-

Fig. 5.25 illustrates the comparison of the measured frequency response of the 6th-order DLL and its expected behavior predicted by theory in Chapter 2 (Fig. 2.6). The red points are the measured results when M equals to 1, 3, 5, 7, 9, 11, respectively, and the blue line is that predicted by theory. As is evident, there is little error between the measured points and the theoretical values. Fig. 5.26 shows the measured frequency responses of DLLs with even orders and their expected frequency response curves and Fig. 5.27 shows the measured frequency responses of DLLs with odd orders and their corresponding frequency response curves. We can see from these two plots that the measured results match the expected values and that the higher order DLL has a better noise attenuation ability because of a sharper roll-off at the same frequency. This is also the reason why we need high-order DLLs when designing time-mode filters (TMF).

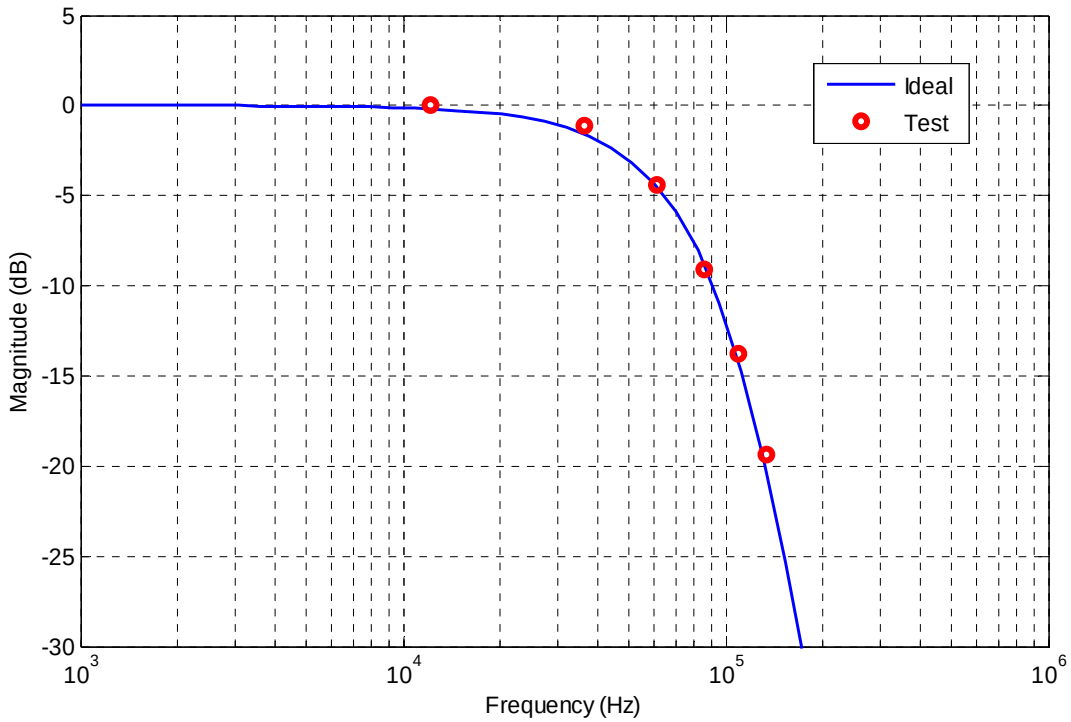


Fig. 5.25: Comparison of the measured frequency response of the 6th-order DLL and the theoretical curve.

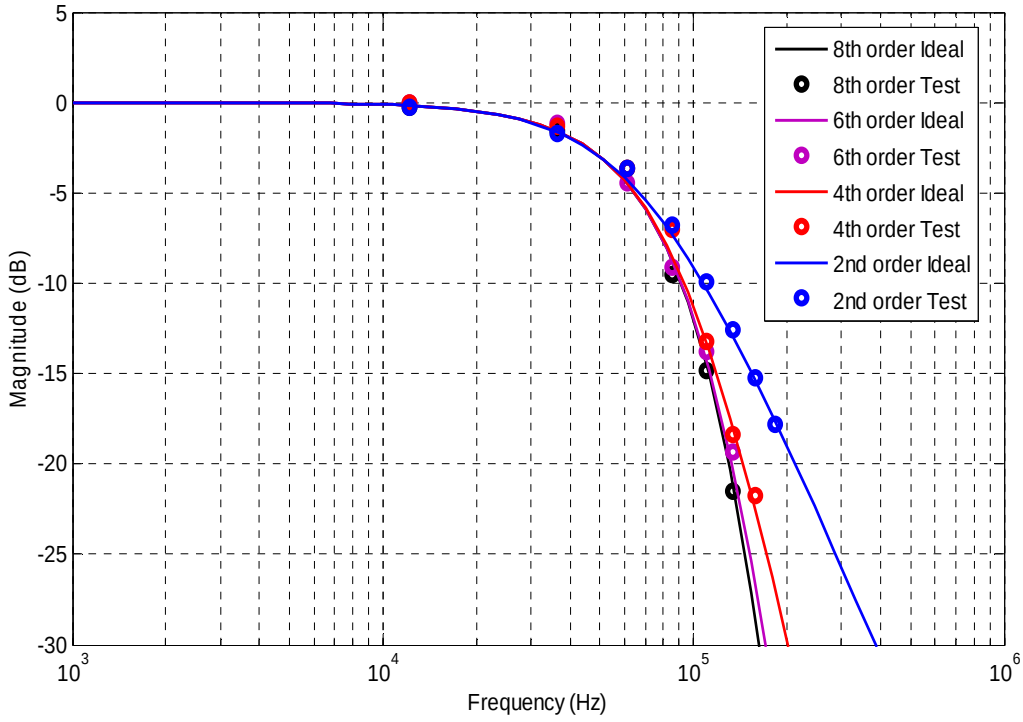


Fig. 5.26: Comparison of the measured frequency responses of the DLLs with orders 2nd, 4th, 6th, 8th and their theoretical curves.

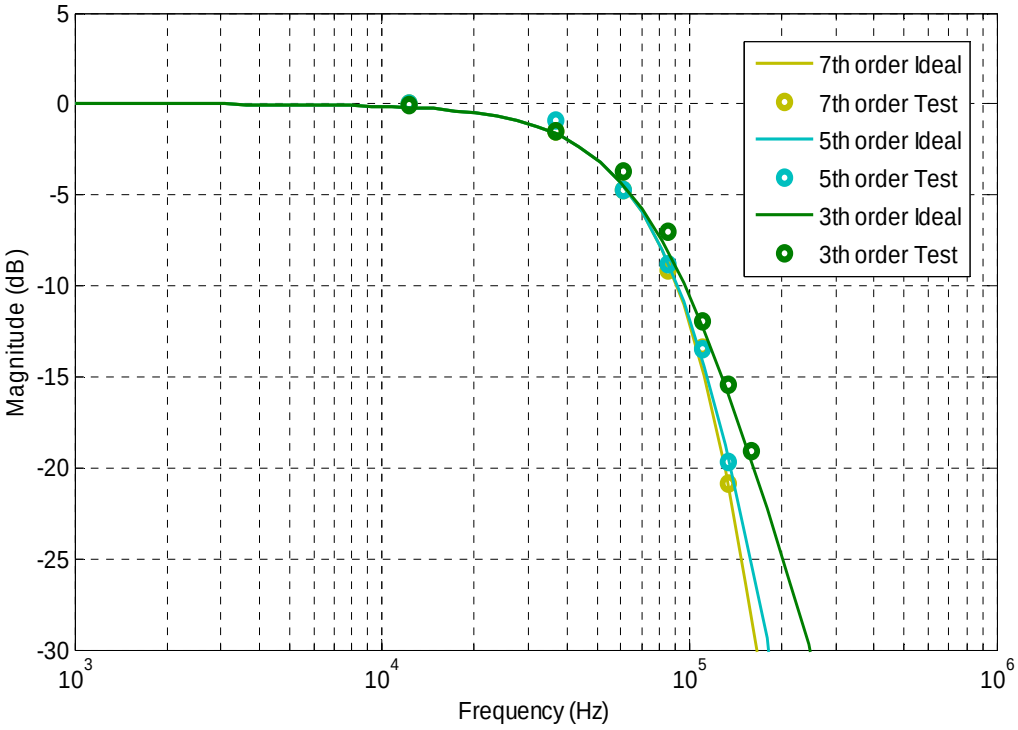


Fig. 5.27: Comparison of the measured frequency responses of the DLLs with orders 3rd, 5th, 7th and their theoretical curves.

5.4 Summary

In this chapter, the experimental behavior of the 2nd to 8th-order DLLs were measured and analyzed using the CMOS prototype IC chip together with a set of seven different active-RC loop filters. The open-loop characteristics of PFD and VCDL was measured and compared with theory. The results matched extremely well. The time-domain closed-loop waveforms including the DLL output and the VCDL control voltage under start-up condition and dynamic phase step change conditions were observed. The testing results were compared to the theoretical curves and, again, resulted in excellent correlation. The frequency-domain characteristics were also obtained using a single-tone testing method, i.e. observing the VCDL control voltage when a sine wave modulated signal was connected to the data input. The sigma-delta encoding theory was used to generate this data input. The final test results were compared to the theoretical frequency response curves and also resulted in good matching. All this combined proves the correctness of the high-order DLL design method using the pole-zero position approach. This was the first time this method was ever applied to the design of DLLs.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

A pole-zero positioning method was used to design high-order DLL circuits. The method was proven a success as the measured results from an IC prototype lined up almost perfectly with that predicted by theory/simulation. The method is general and is based on selecting the desired transfer function of the closed-loop DLL behavior, either time or frequency domain based requirements, and deriving the necessary loop filter transfer function to achieve the desired specifications. The old idea of designing with respect to a phase margin requirement is no longer necessary. This greatly simplifies the design of DLLs.

The DLL basics including the continuous time s-domain model and the system characteristics such as linear region and settling time were discussed in Chapter 2. The pros and cons of the two classes of transfer functions well suited to DLL design, i.e. minimum settling time or minimum delay, was provided. In the end, the minimum settling time class of transfer functions realized by a Gaussian transfer function is the clear winner for high-order DLL design. A Gaussian type DLL has a faster settling time and a much sharper roll-off than the Min-Delay transfer function. Therefore, the Gaussian transfer function is more favorable in high-order DLL design when it is used as a time-mode filter (TMF).

The designs of the DLL building blocks were described in Chapter 3. The transistor-level simulation results were given for each to prove the correctness of the circuit design. The integration of the building blocks into a IC was discussed in Chapter 4. A DLL IC was built on a 1 mm x 1 mm die using the TSMC 130 nm CMOS technology and then housed in a SOIC-24 package. The chip-level simulations results were obtained and compared to the theoretical values. The simulations results again verified the correctness of the circuit level design.

A PCB implementation containing the custom CMOS IC chip and an active-RC loop filter

circuit, which together realizes a high-order programmable frequency-selective DLL circuit, was used to make a fully-programmable high-order DLL circuit whose order ranged from 2 to 8. This prototype was then interfaced to various measurement equipment. Test performed in both the time-domain and the frequency domains, confirmed the correctness of the proposed design method.

6.2 Future Work

There are several improvements that can be made in future designs. First, the transistor sizes of the VCDL could be further adjusted to make the delay time, range and slope more suitable to the input frequency and phase range. This thesis provided one method to increase the DLL frequency range by building up four delay lines and make them "selective" according to the input frequency range. Other better VCDL topologies remain to be found in terms of delay range, slope, linearity, power and space saving.

The DLL loop filters were built off-chip in this thesis. In future designs, filters could also be integrated into the IC in order to decrease the power consumption and make the design more space saving.

Finally, the high-order DLL could be applied and integrated into a larger system, such as the jitter injection signal generation and extraction system for embedded test of high-speed data I/O to further verify its functionality as a TMF [14].

References

[1] B. Razavi, "Delay Locked Loops An Overview," in *Phase-Locking in High-Performance Systems: From Devices to Architectures*, Wiley-IEEE Press, pp. 13-22, 2003.

[2] B. Razavi, Monolithic Phase-Locked Loops and Clock Recovery Circuits: Theory and Design-A Tutorial, IEEE Press, 1996.

[3] D. J. Foley and M. P. Flynn, "CMOS DLL-Based 2-V 3.2-ps Jitter 1-GHz Clock Synthesizer and Temperature-Compensated Tunable Oscillator," *IEEE Journal of Solid State Circuits*, vol. 36, pp. 417-423, Mar. 2001.

[4] P. Heydari and M. Pedram, "Jitter-induced power/ground noise in CMOS PLLs: a design perspective," *Proceedings 2001 IEEE International Conference on Computer Design: VLSI in Computers and Processors. ICCD 2001*, Austin, TX, pp. 209-213, 2001.

[5] B. Veillette and G.W. Roberts, "On-Chip Measurement of the Jitter Transfer Function of Charge-Pump Phase-Locked Loops," Proc. *IEEE International Test Conference*, Washington D.C., pp. 776-785, Nov. 1997.

[6] B. R. Veillette and G. W. Roberts, "On-chip measurement of the jitter transfer function of charge-pump phase-locked loops," in *IEEE Journal of Solid-State Circuits*, vol. 33, no. 3, pp. 483-491, Mar 1998.

[7] K. Maatta and J. Kostamovaara, "A high-precision time-to-digital converter for pulsed time-of-flight laser radar applications," in *IEEE Transactions on Instrumentation and Measurement*, vol. 47, no. 2, pp. 521-536, Apr. 1998.

[8] V. Ramakrishnan and P. T. Balsara, "A wide-range, high-resolution, compact, CMOS time to digital conveter," in *Proc. Int. Conf. VLSI Design (VLSID'06)*, Jan. 2006.

[9] R. B. Staszewski, J.Wallberg, S. Rezeq, E. O. C. M. Hung, O. Eliezer, S. Vemulapalli, C. Fernando, K. Maggio, R. Staszewski, N. Barton, L. M. C. Lee, P. Cruise, M. Entezari, K. Muhammad, and D. Leipold, "All-digital PLL and GSM/EDGE transmitter in 90 nm CMOS," in *Proc. IEEE Solid-State Circuits Conf. (ISSCC'05)*, Feb. 2005.

[10] T. Otsuji, "A picosecond-accuracy 700-MHz range Si-polar time interval counter LSI," *IEEE J. Solid-State Circuits*, vol. 28, pp. 941-947, Sep. 1993.

[11] T. Rahkonen, E. Malo, and J. Kostamovaara, "A 3-V fully integrated digital FM demodulator based on a CMOS pulse-shrinking delay line," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS'96)*, vol. 2, pp. 572–575, May 1996

[12] S. Y. Yurish, "Sensors and transducers: Frequency output versus voltage output," Sens. Transducers Mag., vol. 49, no. 11, pp. 302–305, Nov. 2004.

[13]G. W. Roberts, M. Ali-Bakhshian, "A Brief Introduction to Time-to-Digital and Digital-to-Time Converters," in *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 57, no. 3, pp. 153-157, Mar. 2010.

[14]Y. Li, S. Bielby, A. Chowdhury, G. W. Roberts, "A jitter injection signal generation and extraction system for embedded test of high-speed data I/O," *Journal of Electronics Testing*, vol. 32, no. 4, pp. 423-436, Aug. 2016.

[15]S. Aouini, K. Chuai and G. W. Roberts, "Anti-Imaging Time-Mode Filter Design Using a PLL Structure With Transfer Function DFT," in *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 59, no. 1, pp. 66-79, Jan. 2012.

[16]L. D. Paarmann, Design and Analysis of Analog Filters: A Signal Processing Perspective, Springer International Series in Engineering and Computer Science, 2001.

[17]M.-J. E. Lee, W. J. 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, April 2003.

[18]J. R. Burnham, G.-Y. Wei, C.-K. K. Yang and H. Hindi, "A Comprehensive Phase-Transfer Model for Delay-Locked Loops," *Proceedings of the Custom Integrated Circuits Conference*, pp. 16-19, Sept. 2007.

[19]I. A. Young, J. K. Greason, J. E. Smith and K. L. Wong, "A PLL clock generator with 5 to 110 MHz lock range for microprocessors, " *1992 IEEE International Solid-State Circuits Conference Digest of Technical Papers*, San Francisco, CA, USA, pp. 50-51, 1992.

[20]H. Kondoh, H. Notani, T. Yoshimura, et al. "A 1.5-V 250-MHz to 3.0-V 622-MHz operation CMOS phase-locked loop with precharge type phase-frequency detector," *IEICE transactions on electronics*, vol. 78, no. 4, pp-381-388,1995.

[21]S. Kim, K. Lee, Y. Moon, D. K. Jeong, Y. Choi and H. K. Lim, "A 960-Mb/s/pin interface for skew-tolerant bus using low jitter PLL," in *IEEE Journal of Solid-State Circuits*, vol. 32, no. 5, pp. 691-700, May 1997.

[22]H. H. Chang, J. Lin, C. Yang and S. Liu, "A wide-range delay-locked loop with a fixed latency of one clock cycle," in *IEEE Journal of Solid-State Circuits*, vol. 37, no. 8, pp. 1021-1027, Aug. 2002.

[23]C. Toumazou, G. S. Moschytz, B. Gilbert, Trade-offs in analog circuit design: the designer's companion, Springer Science & Business Media, 2004.

[24]K. H. Cheng and Y. L. Lo, "A Fast-Lock Wide-Range Delay-Locked Loop Using Frequency-Range Selector for Multiphase Clock Generator," in *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 54, no. 7, pp. 561-565, July 2007.

[25]Y. Jung, S. Lee, D. Shim, W. Kim, C. Kim and S. Cho, "A dual-loop delay-locked loop using multiple voltage-controlled delay lines," in *IEEE Journal of Solid-State Circuits*, vol. 36, no. 5, pp. 784-791, May 2001.

[26]S. Bae, H. Chi, Y. Sohn and H. J. Park, "A VCDL-based 60-760-MHz dual-loop DLL with infinite phase-shift capability and adaptive-bandwidth scheme," in *IEEE Journal of Solid-State Circuits*, vol. 40, no. 5, pp. 1119-1129, May 2005.

[27]J. R. Burnham, G. Y. Wei, C. K. K. Yang and H. Hindi, "A Comprehensive Phase-Transfer Model for Delay-Locked Loops," *2007 IEEE Custom Integrated Circuits Conference*, San Jose, CA, pp. 627-630, 2007.

[28]R. Schaumann and M. Van Valkenburg, Design of Analog Filters, Oxford University Press, USA, 2001.

[29] I M. Filanovsky, "Enhancing amplifiers/filters bandwidth by transfer function zeroes," *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, pp. 141-144, 2015.

[30]W. Rhee, "Design of high-performance CMOS charge pumps in phase-locked loops," *Circuits and Systems, 1999. ISCAS '99. Proceedings of the 1999 IEEE International Symposium on*, Orlando, FL, vol.2, pp. 545-548, 1999.

[31]T.-H. Lin, C.-L. Ti, Y.-H. Liu, "Dynamic current-matching charge pump and gated-offset linearization technique for delta-sigma fractional-N PLLs," *IEEE Transactions on Circuits and Systems*, vol. 56, pp. 877-885, May 2009.

[32]Karki J. Analysis of the Sallen-Key architecture. Texas Instruments, 1999.

[33]K. H. Cheng, Y. L. Lo, "A fast-lock wide-range delay-locked loop using frequency-range selector for multiphase clock generator," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol.54, no.7, pp. 561-565, 2007.

[34]HE W N., CMOS VLSI Design: A Circuits and Systems Perspective, 4E. Pearson Education India, 2006.

[35]X. Haurie, G. W. Roberts, "A design, simulation and synthesis tool for delta-sigma-modulator-based signal sources," *1996 IEEE International Symposium on Circuits and Systems. Circuits and Systems Connecting the World. ISCAS 96*, Atlanta, GA, vol.4, pp. 715-718, 1996.

Appendix A

System-Level Simulink Models

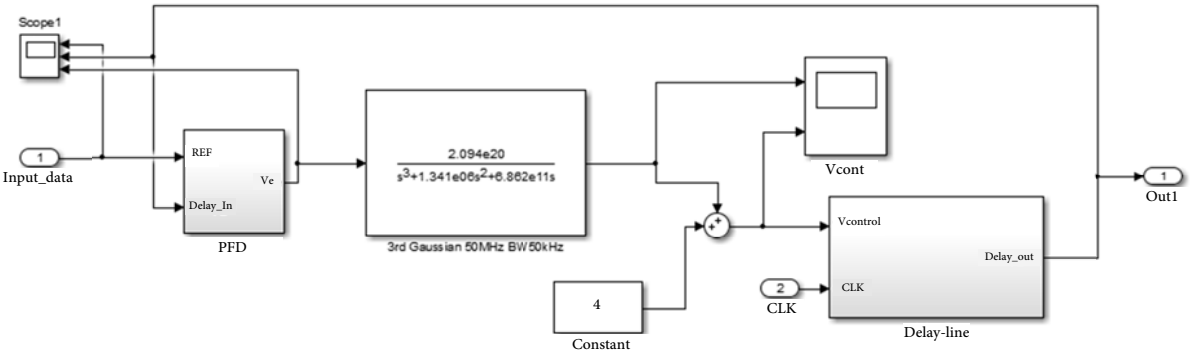


Fig. A. 1: DLL Model

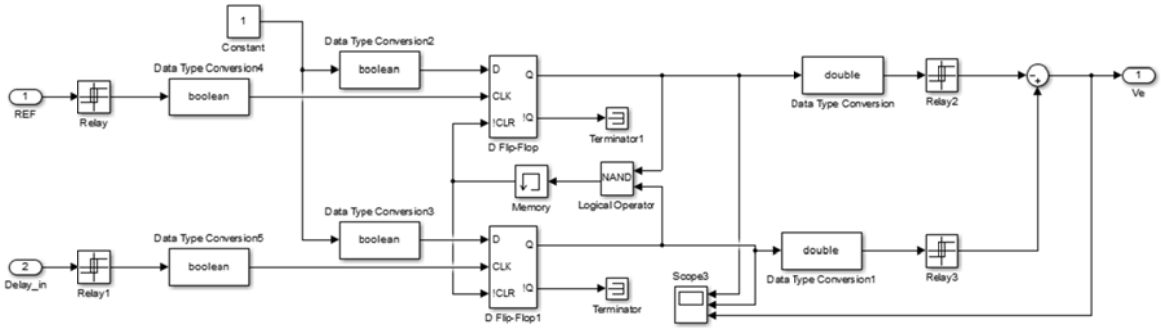


Fig. A. 2: PFD Model

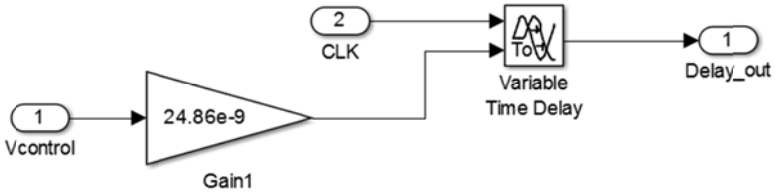


Fig. A. 3: VCDL Model

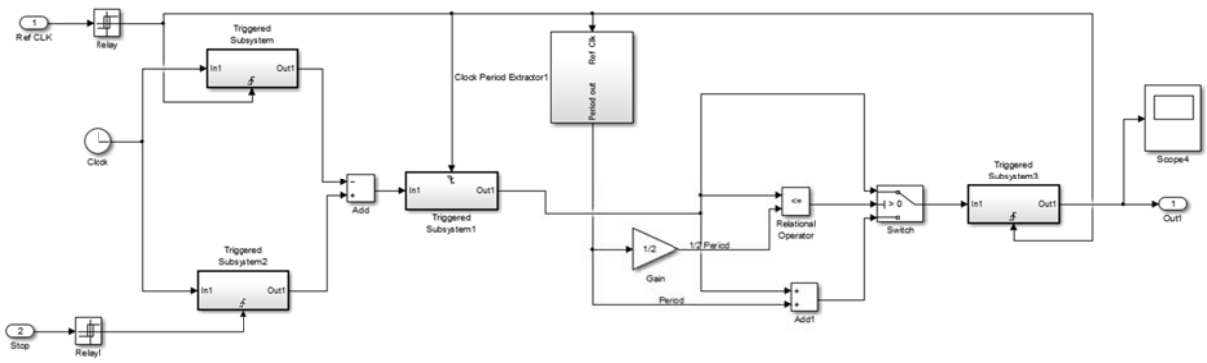


Fig. A. 4: TMU Model

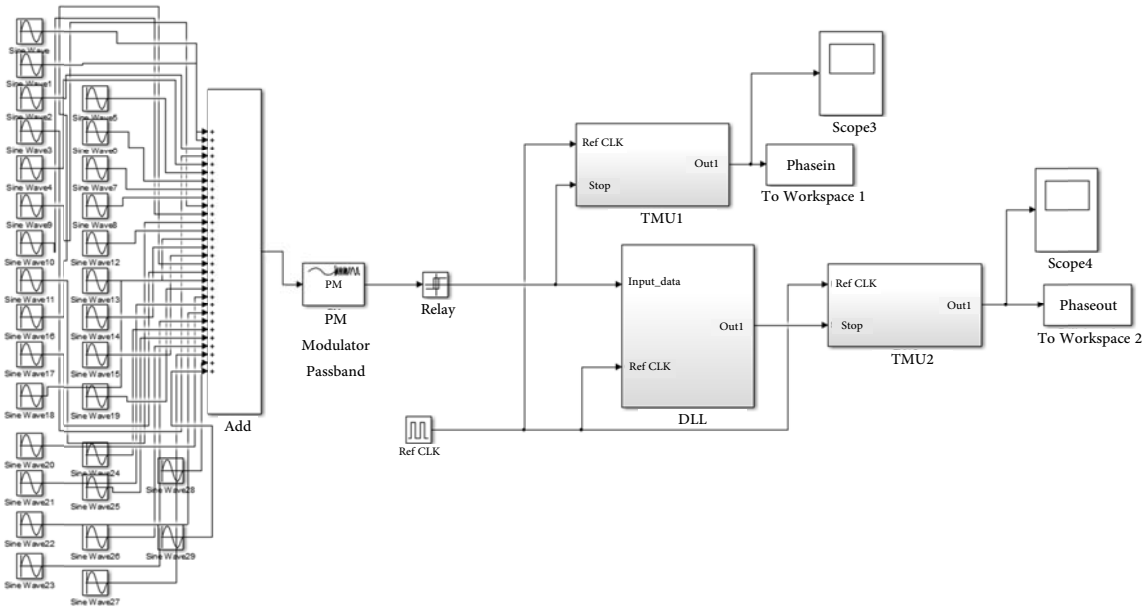


Fig. A. 5: Multi-tone frequency response test

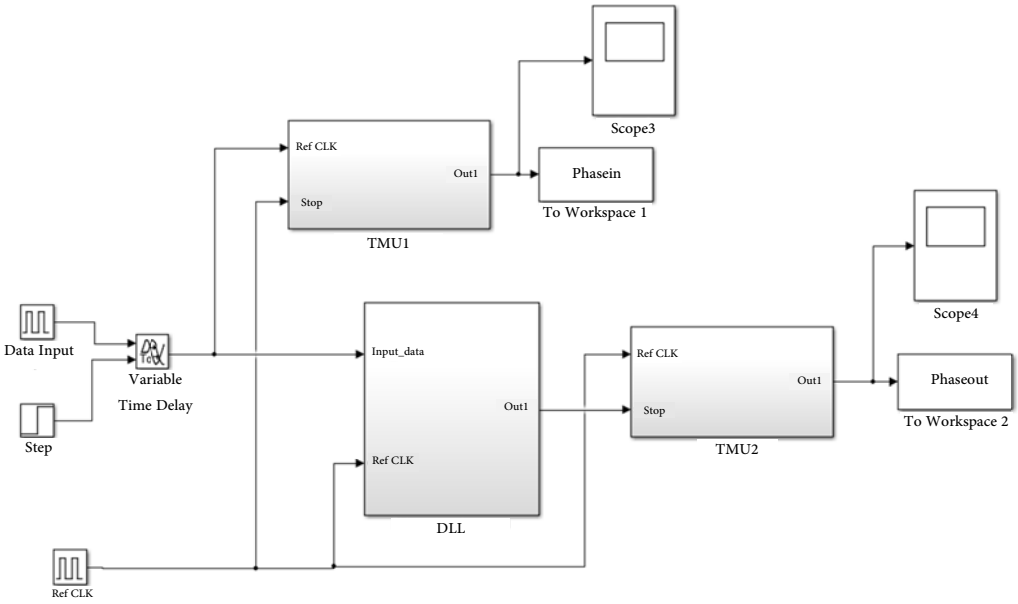


Fig. A. 4: Step response in phase test

Appendix B

Schematics of Gaussian Type Loop Filter

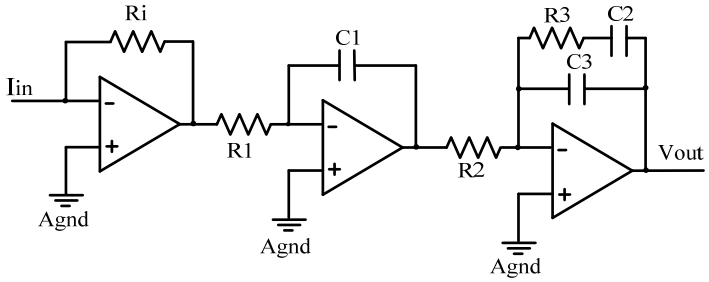


Fig. B. 1: Schematic of 2nd order loop filter

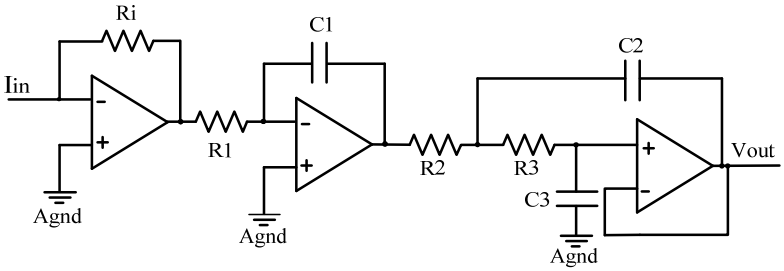


Fig. B. 2: Schematic of 3rd order loop filter

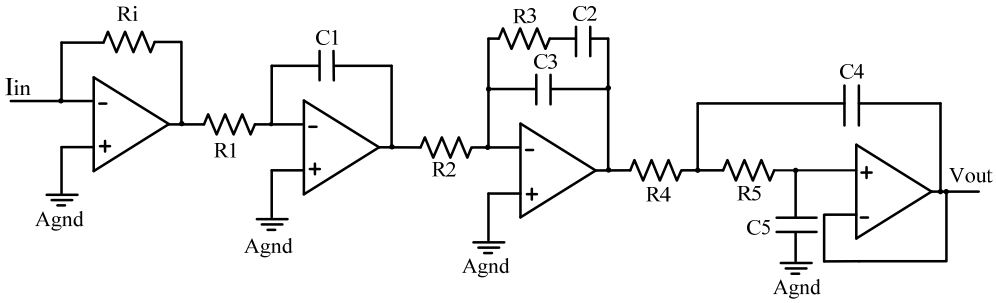


Fig. B. 3: Schematic of the 4th-order loop filter.

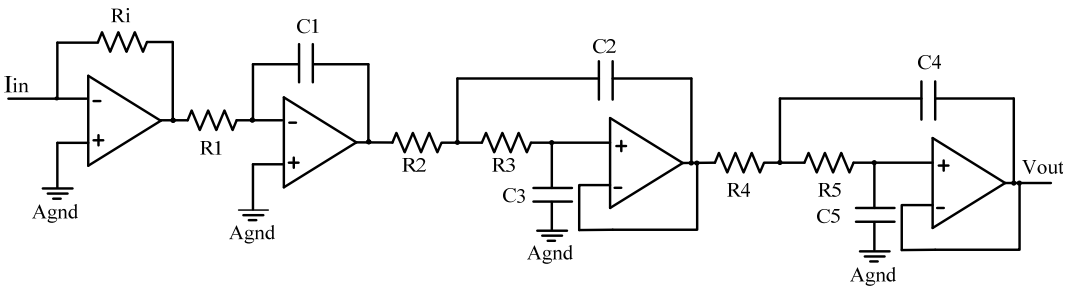


Fig. B. 4: Schematic of the 5th-order loop filter.

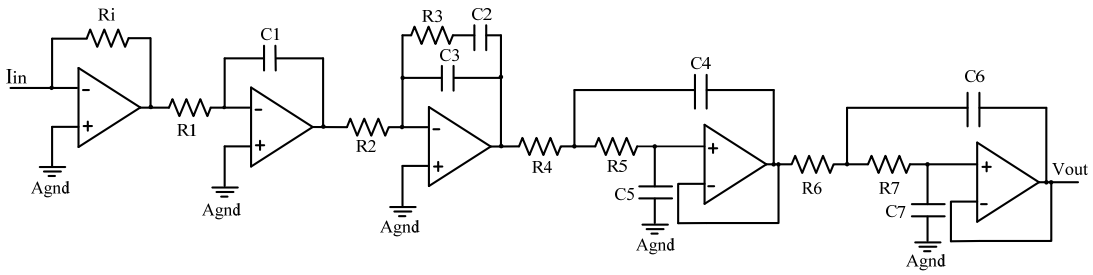


Fig. B. 5: Schematic of the 6th-order loop filter.

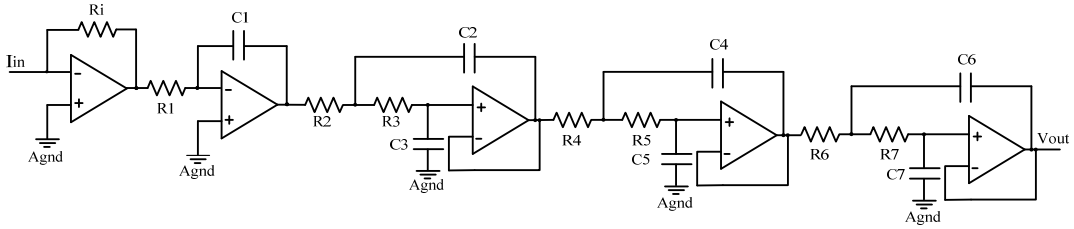


Fig. B. 6: Schematic of the 7th-order loop filter.

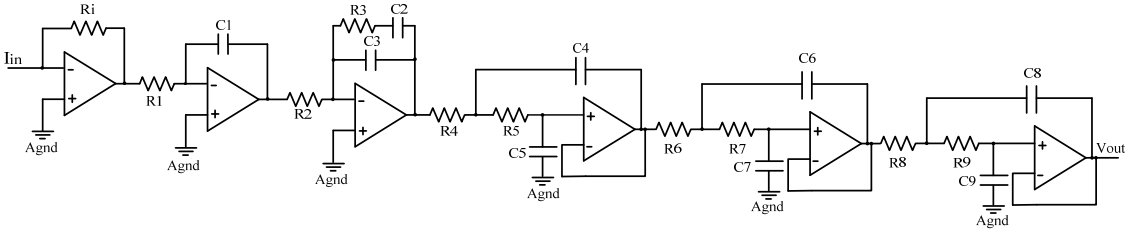


Fig. B. 7: Schematic of the 8th-order loop filter.

Appendix C

DLL Chip Pin Names and Descriptions

Table C.1: DLL Chip Pin Names and Descriptions

Pin Number	Pin Name	Pin Type	Description
1	RCLK	Input	DLL reference clock signal
2	Sel	Input	Select between 12 delay cells or 24 delay cells in VCDL
3	S0	Input	Select between longer delay or shorter delay for one delay cell in VCDL
4	InA	Input	Input signal for PFD/CP 2 circuit
5	InB	Input	Input signal for PFD/CP 2 circuit
6	Start2	Input	Start signal for PFD/CP 2 circuit
7	gnd4	Bidirectional	Power supply for PFD/CP 2 circuit
8	Vdd4	Bidirectional	
9	NC	-	-
10	NC	-	-
11	Iout2	Bidirectional	Output current from PFD/CP 2 circuit
12	Vcont	Bidirectional	Control signal for VCDL
13	Iout1	Bidirectional	Output current from main DLL circuit
14	NC	-	-
15	gnd2	Bidirectional	Power supply for the VCDL circuit
16	vdd2	Bidirectional	
17	Fb	Bidirectional	DLL output/feedback Signal
18	gnd3	Bidirectional	Power supply for the output buffer
19	vdd3	Bidirectional	
20	Start	Input	Start signal for main DLL circuit
21	Sbuff	Input	Switch on/off output buffer
22	vdd	Bidirectional	Power supply for the PFD/CP circuit
23	gnd	Bidirectional	
24	Data	Input	Data input for the main DLL circuit

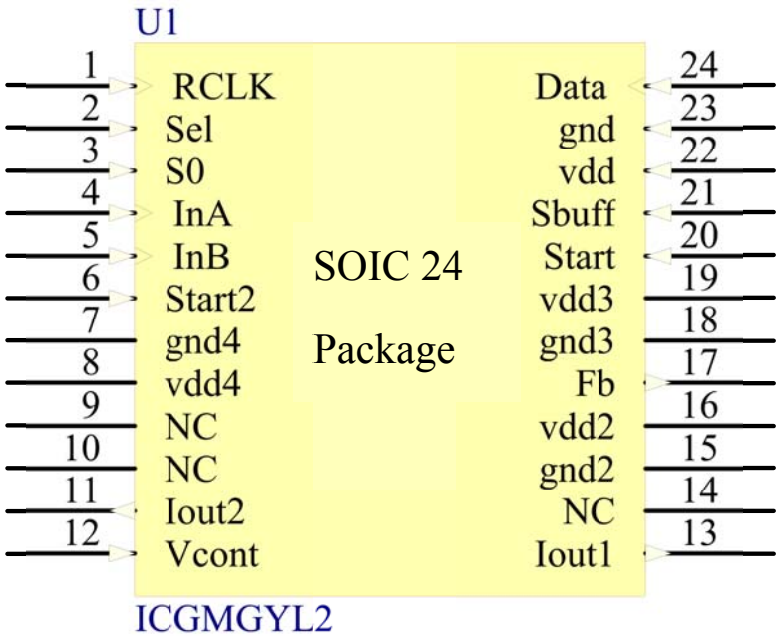


Fig. C. 1: Pin names of the SOIC package DLL chip.

Appendix D

Schematics of PCB

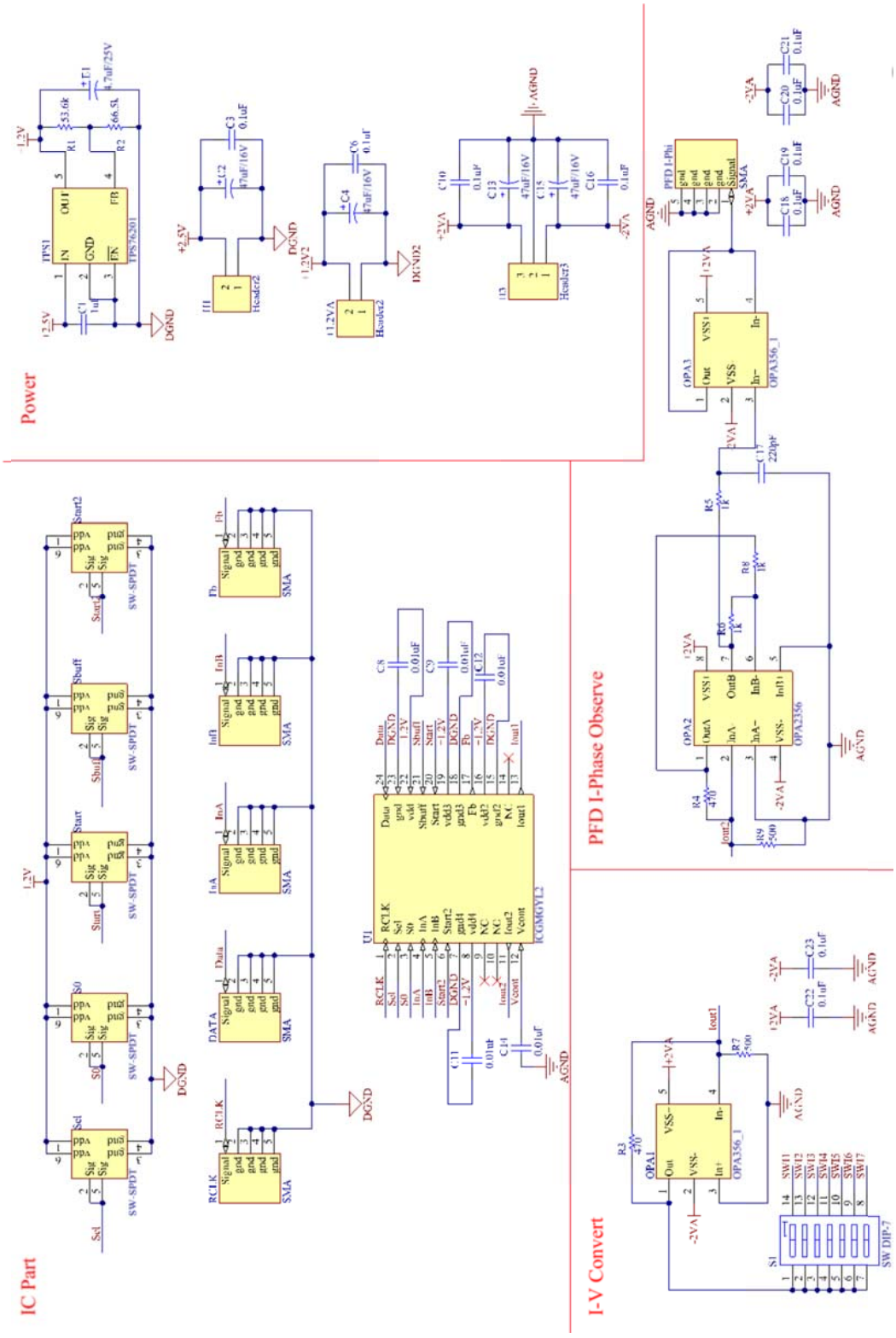


Fig. D. 1: PCB schematic 1

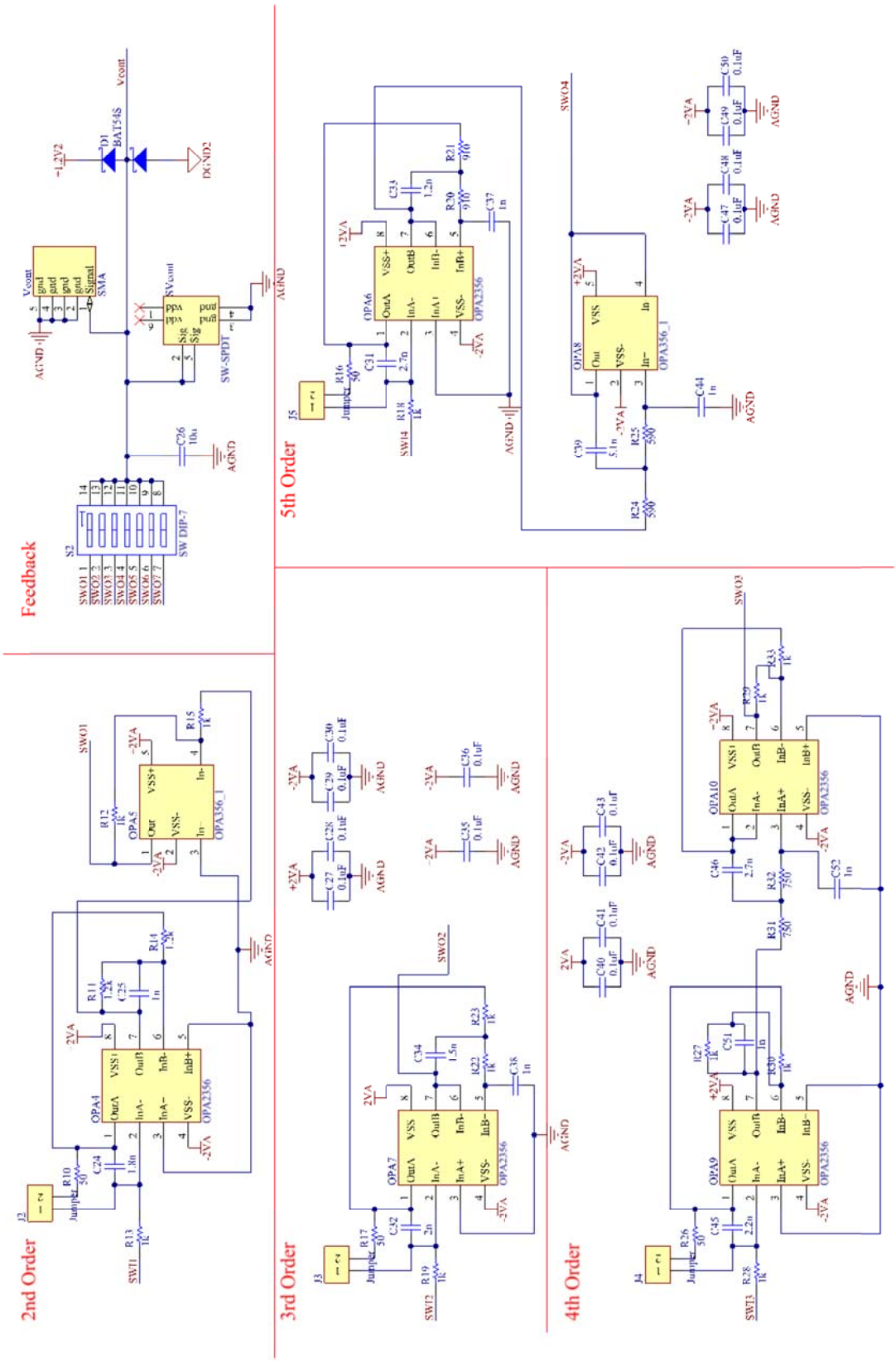


Fig. D. 2: PCB schematic 2

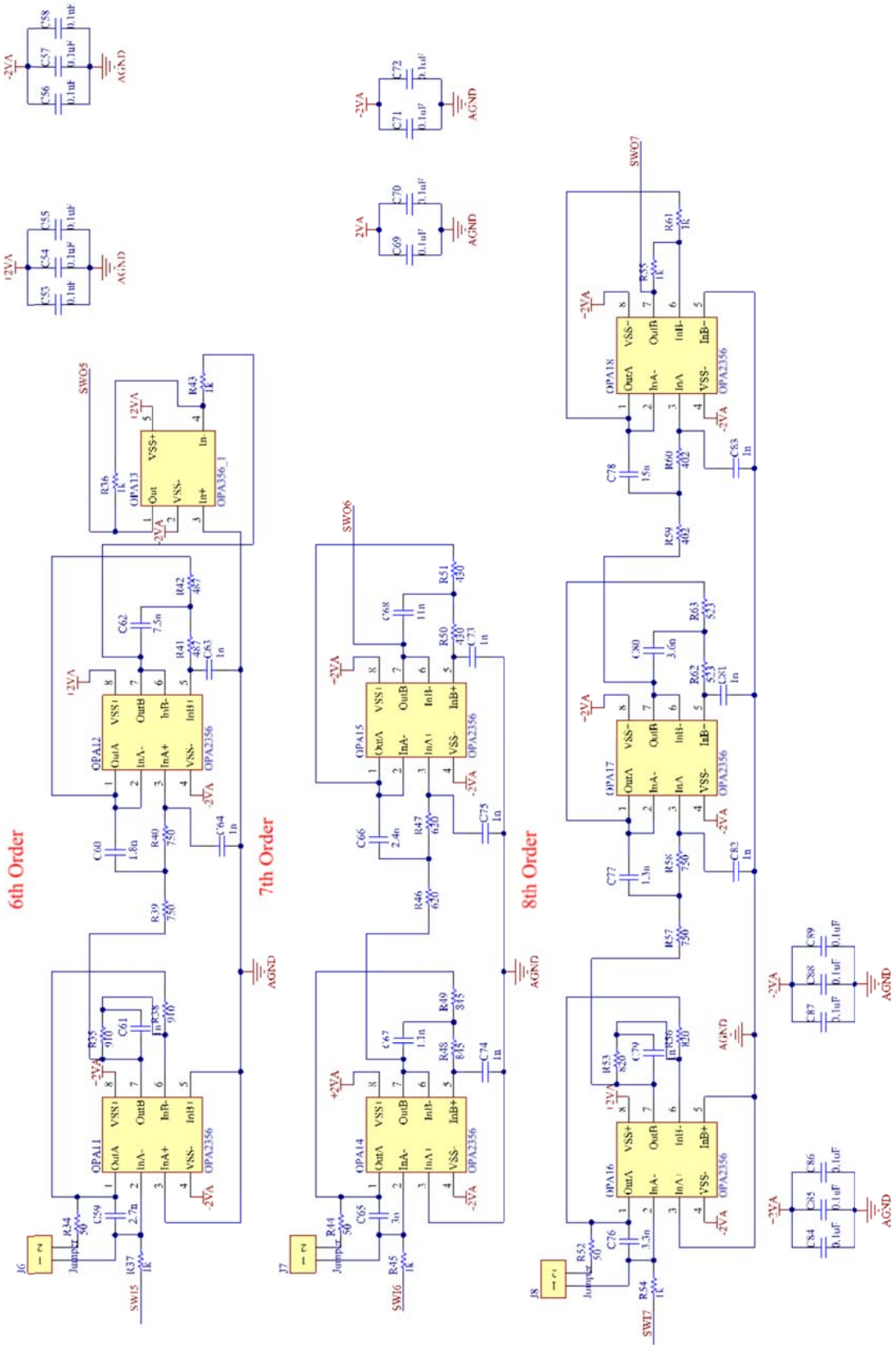


Fig. D. 3: PCB schematic 3