國 立 交 通 大 學

電機與控制工程研究所

碩 士 論 文

1.25GHz, 8 個相位輸出之全數位鎖相迴路

A 1.25GHz All Digital Phase-Locked Loop with 8-phase Output

研 究 生：陳俊銘

指導教授：蘇朝琴　教授

中 華 民 國 九 十 四 年 十二 月

1.25GHz, 8 個相位輸出之全數位鎖相迴路

研究生：陳俊銘　　指導教授：蘇朝琴 教授

國立交通大學電機與控制工程研究所

摘　　要

隨著網路資料傳輸速度的與日遽增，低成本高速串列式傳輸技術亦隨之蓬勃發展。串列式傳送器的應用相當廣，可應用於光纖網路、萬用串列匯流排(USB)、IEEE-1394 等系統。本論文探討使用 0.18 微米 CMOS 製程來實現傳送器前端之相關電路技術，其具體目的在達成 1.25Gbps 的串列式全數位化傳送器電路。

在此論文中，我們有兩個研究主題。首先，我們分析並比較各個種類的多工器。依據比較結果與 IEEE802.3ah 的規格要求，我們以全數位化方式來實現 10 對 1 的多工器。此外，我們將此多工器和鎖相迴路、輸出驅動器整合在一起成為一個完整的串列式資料傳輸器。此電路設計採用 0.18um 1P6M TSMC CMOS 製程技術實現。經由量測結果，其抖動值約為 66ps，並可達到 1.25Gbps 的傳輸速度，晶片面積為 $1900\times990\mu m^2$。

第二個研究主題為 1.25GHz 全數位鎖相迴路，我們提出一個具有高解析度多相位輸出的數位控制震盪器電路。接著我們藉由 MATLAB simulink 來對整個系統作功能驗證, 並且得到可接受的輸出抖動對應到的解析度大小。此電路設計採用 0.18um 1P6M TSMC CMOS 製程技術實現。佈局完成後經由 SPICE 模擬得到輸出抖動為 104ps，功率消耗為 24.49 毫瓦，晶片面積為 $880\times730\mu m^2$

關鍵字: 資料傳輸器，傳送器，數位控制震盪器，全數位鎖相迴路

A 1.25GHz All Digital Phase-Locked Loop with 8-phase output

Student: Chun-Ming Chen Advisor: Chau-Chin Su

Institute of Electrical and Control Engineering

National Chiao Tung University

Abstract

The increasing demand for the data bandwidth in network has driven the development of high-speed serial link technology. The high speed serial links have been applied in optical communication, USB, IEEE-1394. This thesis develops a transmitter front-end circuit design in 0.18um CMOS process technologies. The objective goal of this research is to realize 1.25Gbps an all-digital serial link transmitter.

There are two major topics in this thesis. First, we compare and analyze all types of serializer. Base on the comparison results and IEEE802.3ah specification, we implement a 10 to 1 serializer in all-digital approach. Besides, we integrate this serializer, PLL, and output driver together to become a complete serial link data transmitter. It has been implemented using 0.18um 1P6M TSMC CMOS technology. The measured jitter is about 66ps and can be capable of operate 1.25Gbps data rate. Chip size is about $1900 \times 990 \mu m^2$.

The second research topic is a 1.25GHz all digital phase-locked loop. We propose a high resolution digitally-controlled oscillator with multi-phase output. Functional verification of the ADPLL is performed by MATLAB simulink and gets optimal DCO resolution corresponding to output clock jitter acceptable. It will be implemented using 0.18um 1P6M TSMC CMOS technology. The output clock jitter is about 104ps after post-layout simulation, and power consumption is about 24.49mW. Overall chip size is about $880 \times 730 \mu m^2$.

Keyword: Data transmitter, Serializer, Digital-controlled oscillator, All digital phase-locked loop

Table of Contents

Table of Contents…… v
List of Figures……vii
List of Tables……x

Chapter 1 1

1.1 Motivation 1

1.2 Thesis Organization 3

Chapter 2 5

Background Study.....5

2.1 Phase-Locked Loop Basics 5

2.2 Analog PLL 6

2.3 Digital PLL(DPLL).....7

2.4 All-Digital PLL (ADPLL).....10

2.5 Comparison.....11

2.6 ADPLL Design Considerations.....12

Chapter 3 13

Data Serializer.....13

3.1 Introduction of Serializer.....13

3.2 Three type Serializer Architecture.....14

3.2.1 Tree-Type Serializer.....14

3.2.2 Single-Stage Type Serializer.....15

3.2.3 Shift-Register Type Serializer.....16

3.3 Transmitter Functional Blocks.....17

3.3.1 Generation of random data.....17

3.3.2 Parallel-In-Serial-Out Multiplexer.....19

3.3.3 LVDS Driver Architecture 20

3.4 Measurement Result.....21

Chapter 4 25

Digitally Controlled Multi-Phase Oscillator 25

4.1 Introduction of DCO *25*

4.2 Standard-cell based DCO Design *27*

4.3 Design of High Resolution Delay Cells *31*

4.4 The Proposed DCO Architecture *35*

4.5 Analysis and Design of the proposed DCO *36*

4.6 Process Variation Consideration *37*

Chapter 5 47

ADPLL System Design 47

5.1 System Architecture *48*

5.2 Algorithm *48*

5.3 Control Circuit *51*

5.4 System Behavior Simulation *54*

5.5 System Circuit Level Simulation *58*

5.6 Layout *61*

5.7 Comparisons and Summary *62*

Chapter 6 63

Conclusions 63

Bibliography 64

List of Figures

Figure 1.1 1.25Gbps LVDS transmitter structure 2
Figure 2.1 Block diagram of the PLL 5
Figure 2.2 Linear model of the analog PLL 6
Figure 2.3 The block diagram of the DPLL 7
Figure 2.4 Conceptual operation of a PFD with (a) unequal phases, and (b) unequal frequency 8
Figure 2.5 Implementation of PFD 8
Figure 2.6 Example circuit of the charge pump & loop filter 9
Figure 2.7 Simple charge-pump PLL 9
Figure 2.8 The Block diagram of the ADPLL 10
Figure 3.1 Tree-type serializer architecture 14
Figure 3.2 Single-stage type serializer architecture 15
Figure 3.3 Single-stage type serializer timing diagram 15
Figure 3.4 Shift-register type serializer architecture 16
Figure 3.5 Shift-register type serializer timing diagram 16
Figure 3.6 Overall transmitter architecture 17
Figure 3.7 Pseudorandom binary sequence 17
Figure 3.8 Linear feedback shift register 18
Figure 3.9 LFSR Simulation 18
Figure 3.10 Serializer timing simulations 19
Figure 3.11 Simulated eye diagrams of the data serializer 19
Figure 3.12 The overall LVDS driver architecture[8] 20
Figure 3.13 PCB layout 21
Figure 3.14 Test board photo 21
Figure 3.15 1.25GHz VCO output 21
Figure 3.16 VCO output eye diagram 21
Figure 3.17 PLL divider output waveform 22
Figure 3.18 Divider output eye diagram 22
Figure 3.19 1.25Gbps overall transmitter differential output eye diagram 22
Figure 3.20 1.25Gbps overall transmitter differential output eye diagram (measurement results) 22
Figure 3.21 The overall chip photograph 23
Figure 3.22 Measurement environment 24
Figure 4.1（a）8-cell DCO with control bit [9] 26
Figure 4.2 Structure of DCO in [10] 27

Figure 4.3 A typical ring-oscillator 27
Figure 4.4 Modified architecture in [11] 28
Figure 4.5 An improved DCO architecture in [12] 28
Figure 4.6 Structure of the cell-based DCO in [13] 29
Figure 4.7 DCO using parallel tri-state inverters to adjust frequency in [14] 30
Figure 4.8 The proposed differential operation ring oscillator 31
Figure 4.9 A proposed high-resolution delay cell in [13] 32
Figure 4.10 High resolution delay cell in [12] 33
Figure 4.11 Parasitic capacitance in triode and cut-off 33
Figure 4.12 The proposed high resolution delay cell 34
Figure 4.13 High resolution delay cell simulation result 34
Figure 4.14 The proposed cell-based DCO architecture 35
Figure 4.15 Influence of PVT variations on the controllable frequency range 37
Figure 4.16 The normalization of DCO structure 38
Figure 4.17 The DCO output clock period simulation result when all control pins are ones 39
Figure 4.18 The DCO output clock period simulation result when all control pin are zeros 39
Figure 4.19 Tuning range simulation result at different fine-tuning loading value 40
Figure 4.20 The enlarged DCO output clock period simulation result when all control pin are ones 40
Figure 4.21 The enlarged DCO output clock period simulation result when all control pin are zeros 41
Figure 4.22 Tuning range sim. at different fine-tuning loading value from 0.7 to 1.5 41
Figure 4.23 Find the ratio of M to satisfy our desired performance 42
Figure 4.24 Our proposed DCO architecture 43
Figure 4.25 Decoder circuit 43
Figure 4.26 Frequency range of the proposed DCO 44
Figure 4.27 Linearity and resolution comparison 45
Figure 4.28 The clock period range of DCO at three different corner cases 46
Figure 4.29 The clock frequency range of DCO at three different corner cases 46
Figure 5.1 The ADPLL system architecture 48
Figure 5.2 Binary search versus improved binary search 49
Figure 5.3 Phase acquisition algorithm 49
Figure 5.4 Phase/frequency maintenance algorithm 50
Figure 5.5 Counter-based frequency comparator timing state machine 51
Figure 5.6 Binary ripple carry adder/subtractor circuit 52
Figure 5.7 Gain register circuit 52

Figure 5.8 DCO control register circuit 53
Figure 5.9 Anchor circuit and function table 53
Figure 5.10 Behavior Simulation of the proposed ADPLL by SIMULINK 54
Figure 5.11 Function verification of the proposed ADPLL by SIMULINK 55
Figure 5.12 Output clock jitter simulation using SIMULINK 55
Figure 5.13 Eye diagram jitter with different resolution 56
Figure 5.14 System re-lock simulation when reference clock drift to 158.75MHz 57
Figure 5.15 System re-lock simulation when reference clock drift to 153.75MHz 57
Figure 5.16 Eye diagram jitter with different count number 58
Figure 5.17 System at lock state 59
Figure 5.18 Eye diagram jitter with ground bounce (pre-sim) 59
Figure 5.19 Eye diagram jitter with ground bounce (post-sim) 60
Figure 5.20 Eight-phases output eye diagram 60
Figure 5.21 Layout of the proposed ADPLL 61

Lists of Tables

Table 2-1 PLLs comparison..... 11
Table 3-1 Overall chip summary of the transmitter..... 23
Table 4-1 The features of proposed DCO..... 45
Table 5-1 Summary of the ADPLL chip..... 61
Table 5-2 ADPLL performance comparison 62

Chapter 1

Introduction

1.1 Motivation

As VLSI technology grows up, the design trend goes toward system-level integration and single-chip solution. Thus in *System-On-Chip* (SoC) designs, each module should better be reusable and process portable so that the total design time can be reduced. Unfortunately, some time critical blocks, such as *Phase-Locked Loops* (PLLs) are neither reusable nor process portable.

The PLL is widely used in SoC designs. They are often applied to communication applications to perform the tasks of frequency synthesis, duty-cycle enhancement, and clock de-skewing [1]. A PLL can be used as frequency synthesizer by including a frequency divider in the feedback path. A PLL-based frequency synthesizer also can be used as clock generator for digital circuits. This is advantageous as it allows the processor to operate at a high internal clock frequency derived from a low-frequency input clock. Moreover, the ability to oscillate at different frequencies reduces the cost by eliminating the need for additional oscillators [2].

Due to high integration of *Very-Large-Scale-Integration* (VLSI) systems, PLLs often operates in a very noisy environment. The digital switching noise coupled through power supply and substrate induces considerable noise into noise-sensitive analog circuit. Besides, it always needs some passive components, such as resistors and capacitors. If these passive components are external components, the PLL will couple even more noise. If these passive components are integrated on chip, they will occupy large area [3]. As a result, how to design a PLL in a more efficient way becomes more and more important in these days.

Figure 1.1 shows the transmitter structure, the transmitter is composed of an 8-phases PLL, a *Parallel-In-Serial-Out* (PISO) multiplexer, and an output driver. Our research motivation mainly focuses on the design and implementation of the PISO and PLL in an all-digital approach. We will compare different kinds of serializer and choose an all-digital structure that can meet our needs. Furthermore, a fast systematic design methodology for PLLs is desirable for current SoC designs. Therefore, to integrate overall system easily, we try to implement PLL module in an all-digital way.

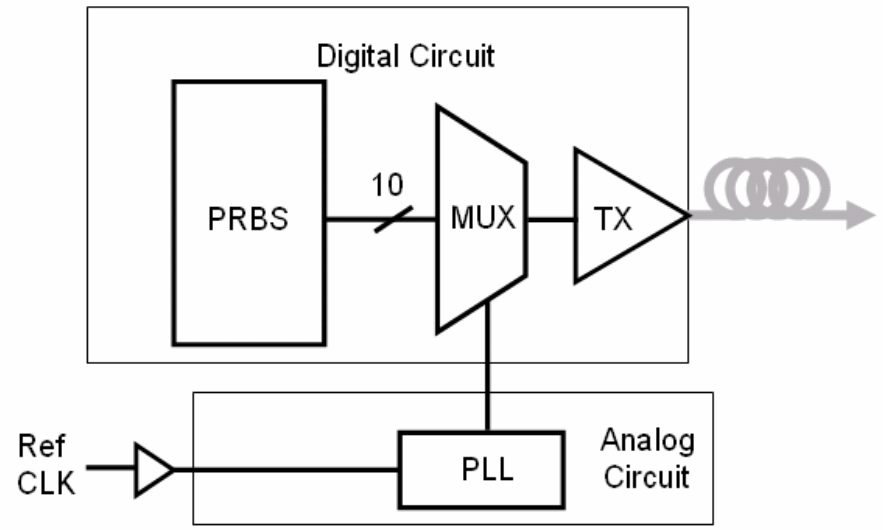


Figure 1.1 1.25Gbps LVDS transmitter structure

In order to avoid the disadvantages of analog circuits, an *All Digital Phase-Locked Loop* (ADPLL) has been developed because it has better programmability, stability, and portability over different processes. It can reduce the system turn around time. The ADPLL has some unique advantages such as high noise immunity, short turnaround time, low power consumption, etc. And, this cell-based approach is also suitable for automatic synthesis of ADPLL.

However, due to the limitations of cell-based design, it is difficult to design a low-jitter, low-power, and high resolution ADPLL. So, how to overcome the limitations of standard cells to build a high resolution delay cell and high sensitivity frequency and phase detector are the challenges for our research.

The goal of this thesis is to design a CMOS serial link transmitter front-end circuits, including a 1.25GHz, 8-phase ADPLL with high resolution DCO and 10-to-1 all digital serializer. Thus, we propose a new standard-cell based design for the DCO structure. We also propose a new fine-tune method to achieve high resolution to overcome the limitations of cell-based design. In this thesis, a systematic design flow for DCO design is also presented. As a result, the design methodology we proposed can reduce design time and design complexity of ADPLL, making it very suitable for SOC applications.

1.2 Thesis Organization

This thesis comprises six chapters. This chapter illustrates the research motivation.

Chapter 2 describes the background study. We will introduce the classification of PLL and their basic concepts. In addition, we compare three different types of PLL and discuss design considerations in this chapter.

In Chapter 3, we describe the data serializer in a transmitter. Fist, we introduce and compare three different types of serializer architecture. Second, we choose shift-register type structure as our serializer circuit and show the simulation results. The simulation results are verified by the measurement in the Section 3.4 from test chips implemented in TSMC 0.18um 1P6M CMOS technology.

In Chapter 4, we describe and compare some existing DCO structure. And, we point out its design parameters and discuss the design issues. In addition, we explain the proposed scheme to overcome the limitations of the standard cells to build high resolution delay cells. Considering process variation, due to many trades-offs between large tuning range and desired operation clock period, we develop a standard procedure to determine the DCO size for the desired target oscillation frequency.

In Chapter 5, we use control mechanism to change the value of the oscillation control word. We use an improved binary search algorithm to perform frequency search and use linear search to perform phase tracking. Overall, building blocks are all constructed in a digital approach. Thus the ADPLL can be implemented with standard cells. Also, it has good portability over different processes.

Finally, Chapter 6 concludes this thesis and discusses the future development.

Chapter 2

Background Study

2.1 Phase-Locked Loop Basics

Any *Phase-Locked Loop*（PLL）must have three basic components, a phase detector, a loop filter and a voltage-controlled oscillator. Figure 2.1 shows the basic structure for a PLL. The PD detects the phase difference between the input signal and output signal and produces a phase error signal $u_d(t)$. The Loop Filter transform the $u_d(t)$ into a voltage signal $u_f(t)$ to control the VCO. This would cause the VCO to change its oscillation frequency in such a way that the phase error finally vanishes.

When the loop is locked, the frequency of the VCO is exactly equal to the average frequency of the input. The loop filter is a low-pass filter that suppresses high-frequency signal components in the phase errors.

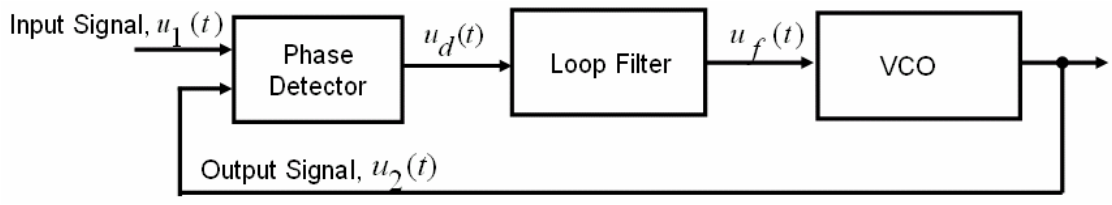


Figure 2.1 Block diagram of the PLL

2.2 Analog PLL

As shown in Figure 2.2, the analog PLL is built from purely analog functional block. The analog multiplier is used as a PD, the *Loop Filter* (LF) is built from passive or active RC filter, and the VCO is a ring oscillator constructed by differential inverters. The analysis of this control system is normally performed by means of its phase transfer function $H(s)$. We describe this system transfer function using Laplace transform.

$$V_d(s) = K_d\theta_e(s) \tag{2-1}$$

$$V_c(s) = F(s)V_d(s) \tag{2-2}$$

$$\theta_o(s) = \frac{K_o}{s}V_c(s) \tag{2-3}$$

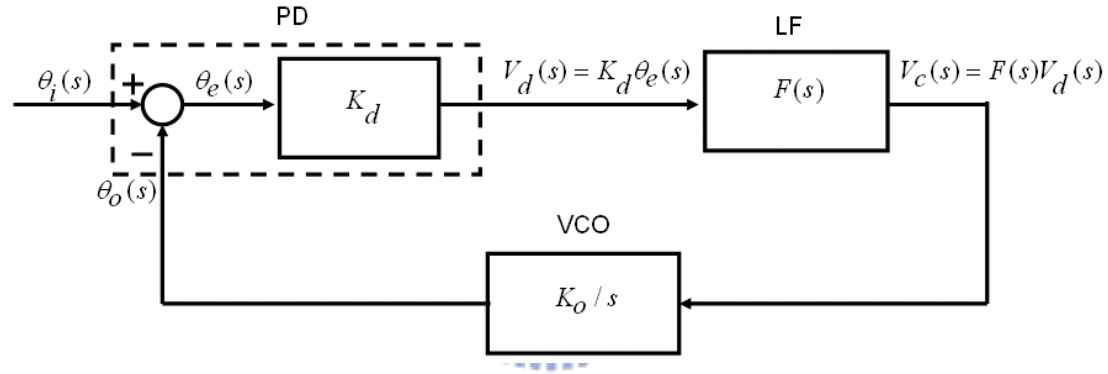


Figure 2.2 Linear model of the analog PLL

The LF is a low-pass filter. The LF may be active or passive. It typically determines the loop being first-order or second-order. There are many types of structures to design a low-pass filter. Assuming the transfer function of the LF is $F(s)$. From (2-1), (2-2), (2-3), we can obtain the phase transfer function $H(s)$:

$$H(s) = \frac{\theta_o(s)}{\theta_i(s)} = \frac{K_oK_dF(s)}{s + K_oK_dF(s)} \tag{2-4}$$

Where K_d is the phase-detector gain, K_o is the VCO gain factor, and $H(s)$ is the closed-loop transfer function.

This model enables us to analyze the tracking performance of the analog PLL, i.e., the system maintain phase tracking when excited by phase steps, frequency steps, or other excitation signals. We can analyze the characteristics and the responses of the analog PLL in S-domain, and then, calculate all parameters to design an analog PLL to satisfy the specification[3].

2.3 Digital PLL(DPLL)

The digital PLL is just an analog PLL with a digital phase detector. In other words, the DPLL is actually a hybrid system built from analog and digital functional blocks. Only the phase detector is built from a digital circuit. The remaining blocks are still analog.

As shown in Figure. 2.3, the DPLL is composed of a phase frequency detector, a charge pump, a loop filter, a voltage controlled oscillator, and a divider. The phase frequency detector detects the phase difference of the two clocks and sends the detection result to the charge pump. The charge pump pumps the charge to the loop filter according to the phase frequency detector detection result. The loop filter's terminal voltage is determined by the charge and the current it contains. The oscillating frequency of the voltage controlled oscillator is controlled by the terminal voltage of the loop filter. The output of the VCO is sent to the divider and be divided into the same frequency as the reference clock. The divider sends the divided clock to the PFD and completes the loop.

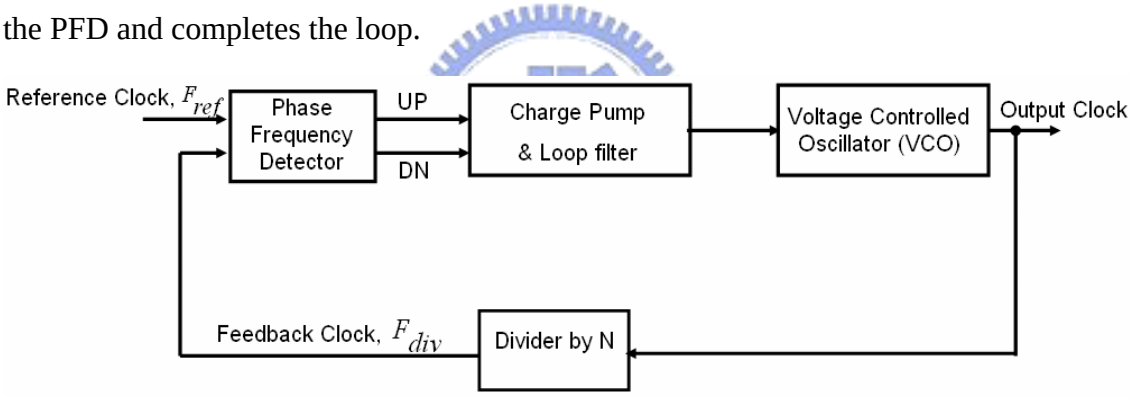


Figure 2.3 The block diagram of the DPLL

A circuit that can detect both phase and frequency differences is called a "phase/frequency detector". It is illustrated conceptually in Figure 2.4. In Figure 2.4(a), the two inputs have equal frequencies but A leads B. The output UP continues to produces pulses whose width is proportional to $\varphi_A - \varphi_B$ while DN remains at zero. In Figure 2.4(b), A has a higher frequency than B, and UP generates pulses while DN does not.

Figure 2.5 shows a simple implementation consisting of two edge-triggered resettable D flipflops with their D inputs tied to logical ONE. The inputs of interest, A and B, serve as the clocks of the flipflops. If UP=DN=0 and A goes high and UP rises. If this event is followed by a rising transition on B, DN goes high and the AND gate resets both flipflops.

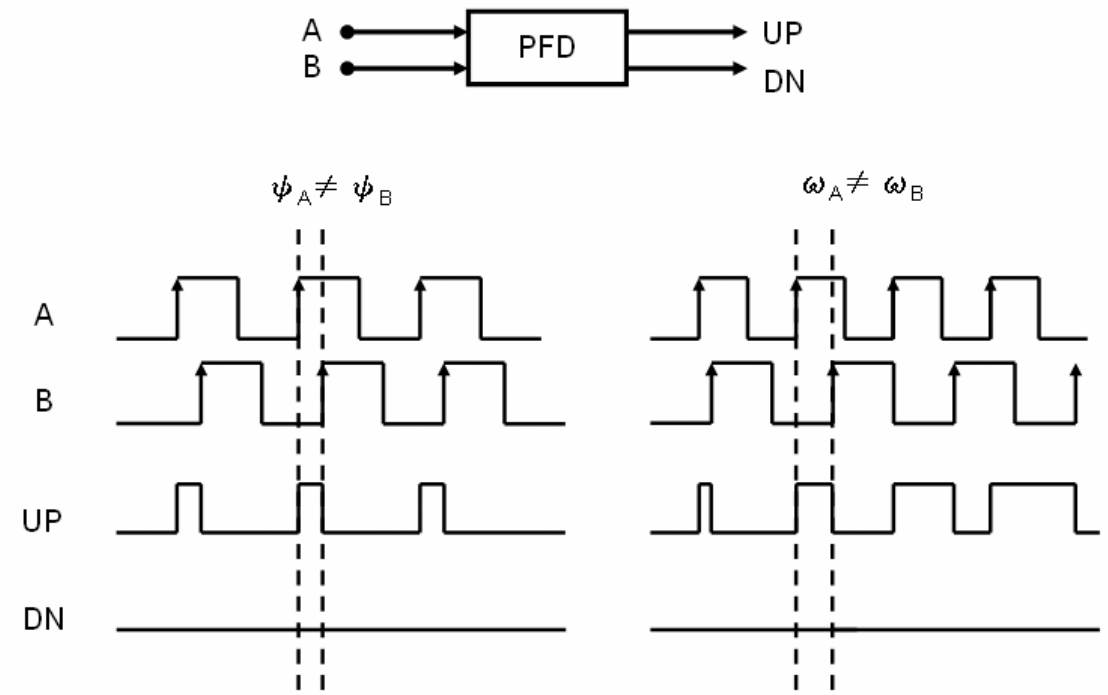


Figure 2.4 Conceptual operation of a PFD with (a) unequal phases, and (b) unequal frequency

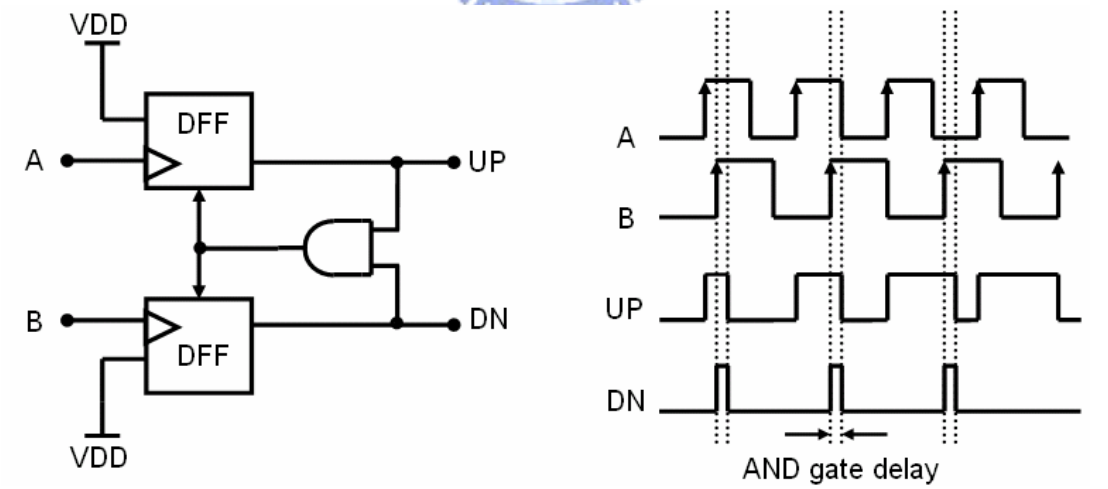


Figure 2.5 Implementation of PFD

Figure 2.6 shows an example of the charge pump & loop filter circuit. A charge pump consists of two switched current sources that pump charge into or out of the loop filter according to its two logical inputs.

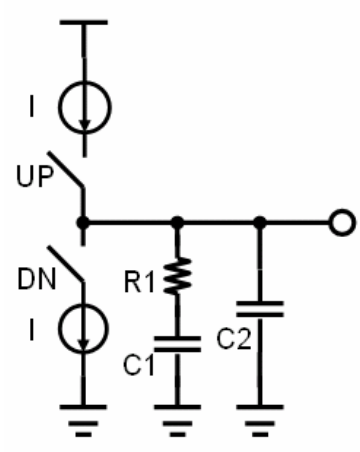


Figure 2.6 Example circuit of the charge pump & loop filter

Figure 2.7 shows a simple charge-pump PLL. When the loop is turned on, f_{out} may be far from f_{in}, and the PFD and the charge pump vary the control voltage such that f_{out} approaches f_{in}. When the input and output frequencies are sufficiently close, the PFD operates as a phase detector, performing phase locking. The loop locks when the phase difference drops to zero and the charge pump remains relatively idle [4].

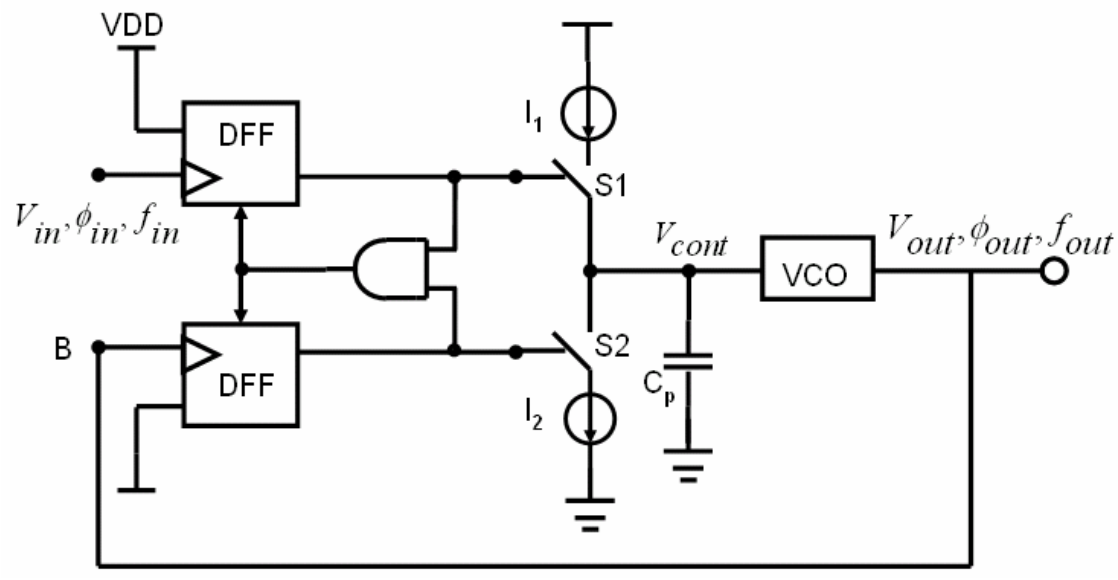


Figure 2.7 Simple charge-pump PLL

The analysis and design of DPLLs is similar to analog PLLs. We can analyze both the characteristics and the responses of a DPLL in S-domain, and then, calculate all parameters to design a DPLL to satisfy the specification.

2.4 All-Digital PLL (ADPLL)

The all-digital PLL (classical all-digital) is distinctly different from the first two PLLs mentioned thus far. The ADPLL is a digital loop in two senses: (1) All digital components, (2) All digital (discrete-time) signals.

It doesn't contain any passive components, such as resistors and capacitors. The VCO is replaced by a digitally controlled oscillator (DCO). The charge pump and loop filter is replaced by UP/DN counter or called control unit.

As shown in Figure 2.8, the ADPLL is composed of a digital phase frequency detector, a digital loop filter, a digitally controlled oscillator, and a divider. The phase frequency detector detects the phase difference of the two clocks and sends the detection result to the digital loop filter. The digital loop filter receives the signal produced by the PFD, and produces a set of digitally controlled signals (binary signals) to control the DCO. The output of the DCO is sent to the divider to divided into the lower frequency range. The divider sends the divided clock to the PFD and completes the loop.

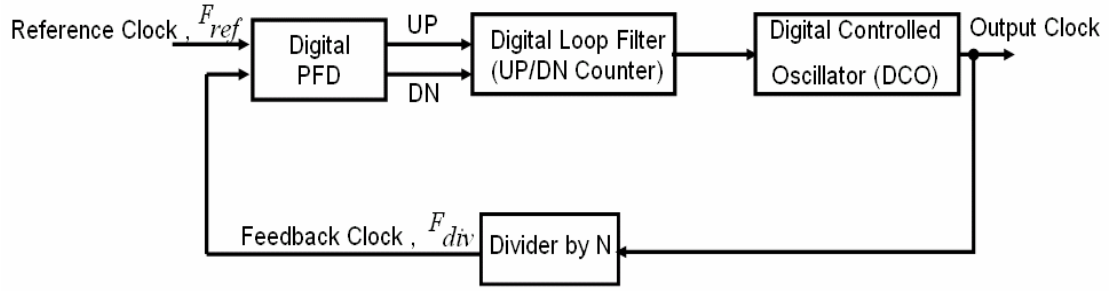


Figure 2.8 The Block diagram of the ADPLL

The functional blocks of ADPLLs imitate the functions of the corresponding analog blocks. Because ADPLLs consists of digital circuits entirely, there are many types of design methods to achieve their functions. Because the ADPLL system is discrete-time system, analyzing the ADPLL in S-domain is not suitable. The ADPLL system can be analyzed in time-domain or Z-domain, or building a new model to analyze the ADPLL system [3].

2.5 Comparison

In the three different types of PLLs, including analog PLLs, digital PLLs and ADPLLs, they have many advantages and disadvantages respectively. We compare and illustrate them in Table 2-1.

Table 2-1 PLLs comparison

	Analog PLL	DPLL	ADPLL
Design methodology	Analog	Analog and Digital Mixed-mode	All-Digital
Turnaround time	Long	Long	Short
Noise immunity	Low	Low	High
Power consumption	Large	Large	Small
Lock time	Long	Long	Short
Area	Large	Large	Small
Oscillator frequency	High	High	Low
Oscillator resolution	High	High	Low

These advantages and disadvantages are principally separated by design methodology. The PLL designed with analog circuits has characteristics of analog circuits. In opposition, the ADPLL designed with digital circuits has characteristics of digital circuits. Therefore, the analog PLL and the DPLL have characteristics of analog circuits but the ADPLL has characteristics of digital circuits.

The analog circuits take much more time to design, so they have a long turnaround time. The digital circuits have higher noise immunity than the analog circuits. The VCO of an analog PLL or a DPLL produces a continuous frequency band but the DCO of an ADPLL produces a discrete frequency band. The VCO has higher resolution than the DCO. The digital circuits generally have lower power consumption than the analog circuits. The ADPLL may have small area because the loop filter of an analog PLL or a DPLL always has one or more large capacitors, whose area cannot be efficiently reduced as the process technology improving.

2.6 ADPLL Design Considerations

As mentioned in Section 2.4, an ADPLL is entirely built from digital functional blocks and these digital blocks imitate the function of the corresponding analog blocks. Because analog circuits cannot be completely replace by digital circuits, there are some problems occurred as designing an ADPLL. Analyzing the ADPLL in S-domain is not suitable, so we analyze the characteristics and the responses of the ADPLL in time-domain.

According to Figure 2.8, an ADPLL consists of three functional blocks: a PFD, a control unit, and a DCO. Based on the mentioned basic concepts above, we point out some design issues as following before we design an ADPLL.

The first step is to analyze and design a DCO. It is the kernel of the ADPLL. The quality of the output clock is decided by its performance. There are three issues to design a DCO. First, the output clock of a DCO is discrete, so the resolution of a DCO should be sufficiently high to maintain acceptable jitter. Second, for searching target frequency and phase easily and efficiently, a DCO should have a monotonic approach to the DCO control word. Third, a DCO should have high noise immunity, so the output clock will not induce large jitter. The noises come from the power supply, the control word transition, or other conditions.

The second step is to analyze and design a FD and PD. It must be sensitive to detect both the frequency difference and the phase difference between the reference clock and the feedback clock. The resolution of FD and PD should be as high as possible.

The third step is to analyze and design a control unit. It receives the signals, produced by the FD and PD, and produces signals to control the DCO. It works as a loop filter. It decides the speed of the lock process and suppresses the high frequency noise to reduce jitter. All responses of an ADPLL are almost decided by this control unit.

Chapter 3

Data Serializer

3.1 Introduction of Serializer

Gigahertz optical-fiber-link systems have become more important because of the increasing demand for high-speed communications. The key components of these systems are MUX (PISO). Fast switching speed is therefore necessary. Serializer combines many low speed parallel random data into a high-speed stream of serial data. This technique effectively reduces the buses for interconnections, and thus alleviates electrical and magnetic interference.

For example, most transmitters incorporate a 16-to1 serializer, allowing 16 inputs to be much slower than the output and hence simplifying the design of the package. This chapter describes the design of all digital CMOS data serializer that is capable of running at a 1.25Gbps data rate, which can be adopted in optical transmitters.

3.2 Three type Serializer Architecture

3.2.1 Tree-Type Serializer

Figure 3.1 shows the data serializer with tree-type architecture which is based on a set of 2:1 multiplexer circuits and uses a binary tree structure. To convert 16 parallel data to a serial data, there are five triggered clock to control the multiplexers. Conventional 2:1 serializer is trigger during the clock transition (both rising & falling edge). If the clock duty cycle is not 50%, it induces the output jitter. Additional retiming can avoid the problem, but it's difficult to produce the global clock for high speed operation.

The data serializer can achieve high-speed operation. Because the RC load of every multiplexer is not large. In [5], a 20Gbps CMOS transmitter is proposed. The design issue is the accuracy of triggered clock. The jitter of preceding clock will affect the performance behind. The effects are accumulated.

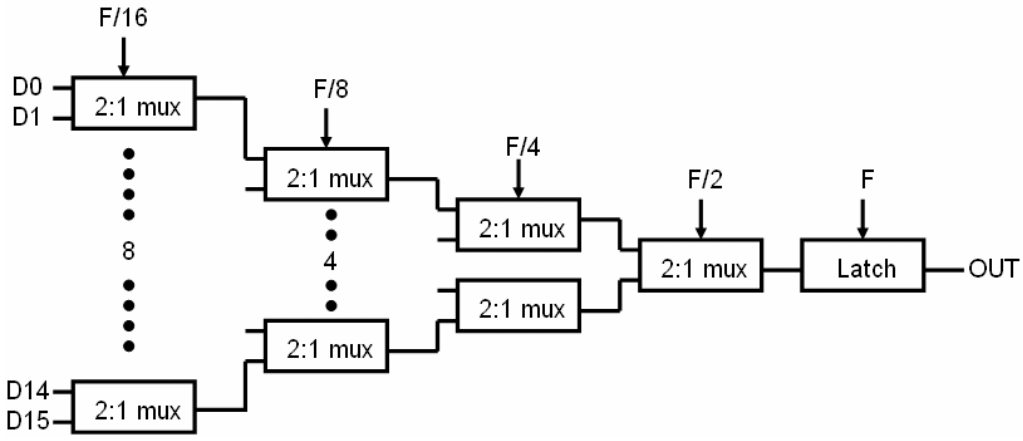


Figure 3.1 Tree-type serializer architecture

3.2.2 Single-Stage Type Serializer

Figure 3.2 shows the data serializer with single-stage type architecture. Figure 3.3 shows the single-stage type serializer timing diagram. Time-division parallel-to-serial data conversion is achieved by gating each transmission path sequentially. Every transmitter gate is controlled by phase of PLL. For example, the data d0 will be exported when ck0 and ck5 overlap. Although there is jitter in the phases of PLL, the phases will move in the same direction. So the overlap of two phases in the same transmission gate is about the same as original one.[6][7]

A larger number of parallelisms will decrease the data rate. Because the multiplexer output has large capacitance and the parallel data has shorter time to convert. So this data serializer is commonly applied to the systems below 4Gbps.

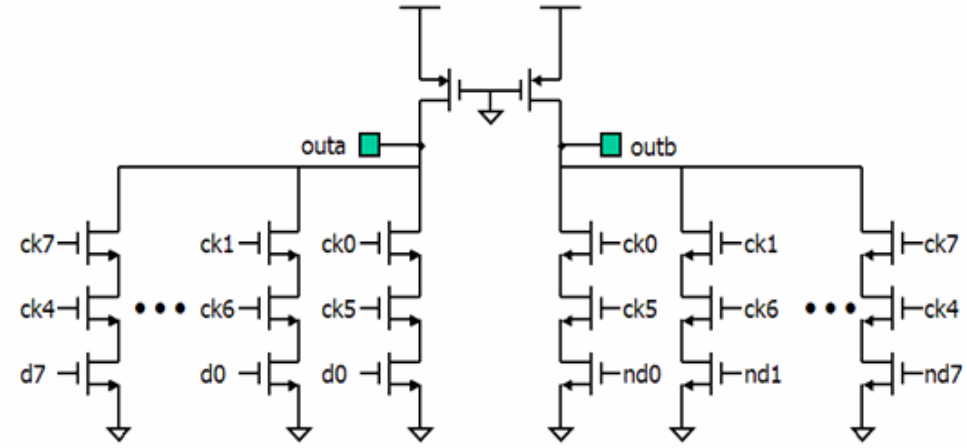


Figure 3.2 Single-stage type serializer architecture

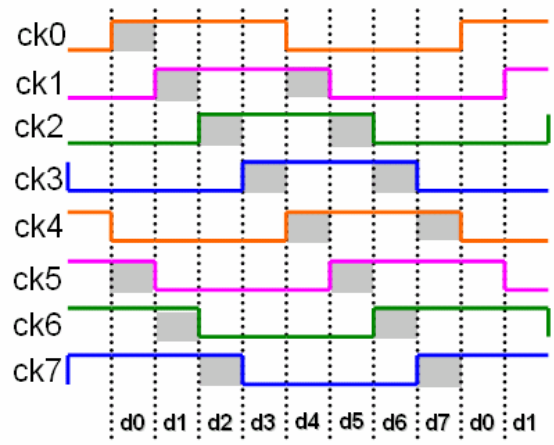


Figure 3.3 Single-stage type serializer timing diagram

3.2.3 Shift-Register Type Serializer

Figure 3.4 shows the data serializer with shift-register type architecture and its timing diagram. This architecture performance is dominated by the maximum operate speed of DFF. And it has the advantage of flexible number of parallelism. Unlike tree-type architecture has the limitations of 2^N number of parallelism. Because most components operate at a higher global clock speed, it may consume more power than tree-type architecture.

After the introduction and comparison of these three-type serializer architecture, we finally adopted shift-register type serializer. First of all, our design specification is 10:1 serializer for EPON standard（IEEE802.3ah PMA Architecture）.We can not take tree-type architecture into consideration. Besides, if we choose single-stage type serializer architecture, we need 10 precise phases clock generator. Therefore, we take shift-register type serializer as our building block. It has the advantage of flexible number of parallelism, simple architecture, easy to implement and all digital.

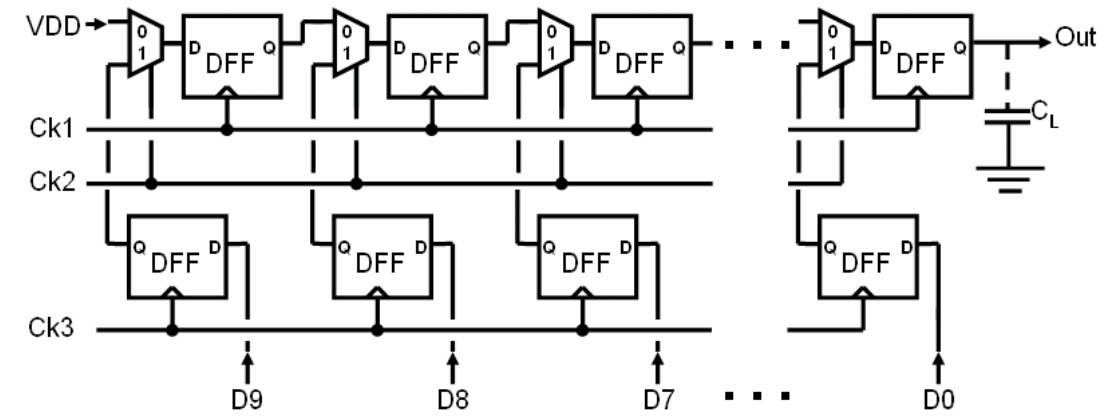


Figure 3.4 Shift-register type serializer architecture

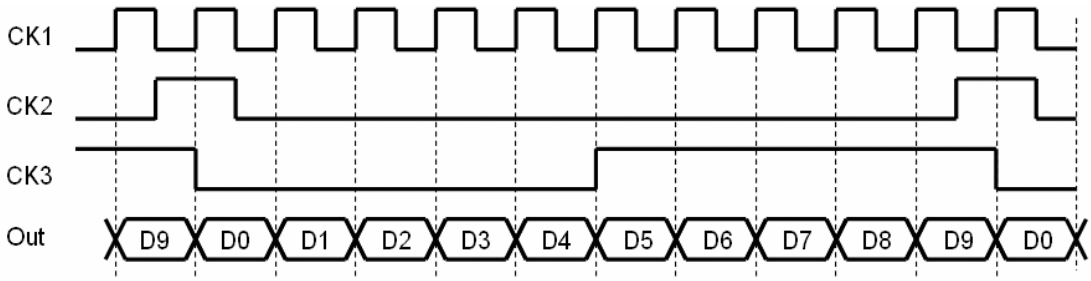


Figure 3.5 Shift-register type serializer timing diagram

3.3 Transmitter Functional Blocks

Figure3.6 shows the overall transmitter architecture, which includes a multiphase PLL for data transmission, a parallel-in-serial-out multiplexer (PISO), and an output driver. Parallel to serial data conversion is achieved by means of the shift-register type MUX. Its clock is provided by a PLL. The role of the output driver is to drive serial data to physical channel and provide good driving capability to avoid ISI.

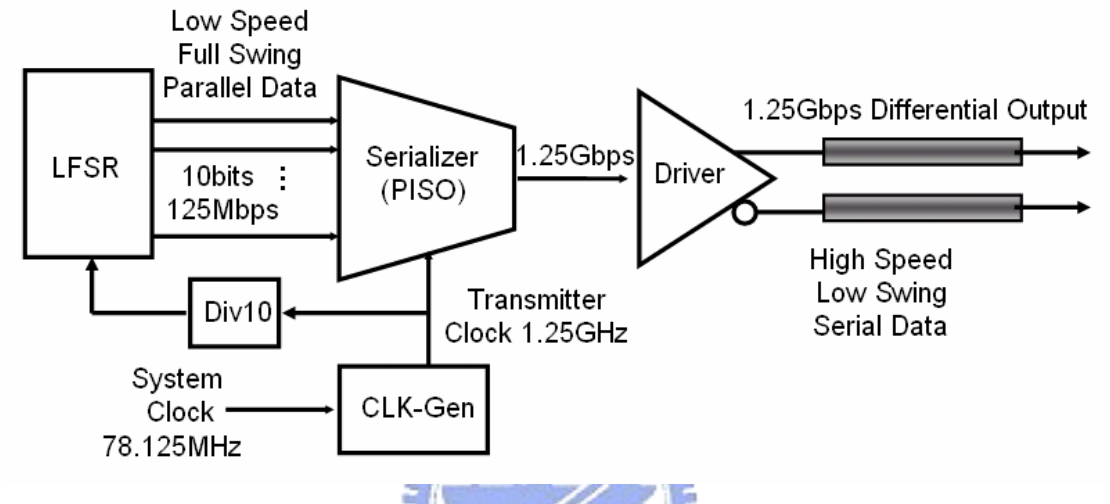


Figure 3.6 Overall transmitter architecture

3.3.1 Generation of random data

In characterization and simulation, it is difficult to generate completely random binary waveforms because for the randomness to manifest itself. For this reason, it is common to employ a *pseudo-random binary sequences*(PRBS). Each PRBS is in fact a repetition of a pattern that itself consists of random sequences of a number of bits as shown in Figure 3.7.

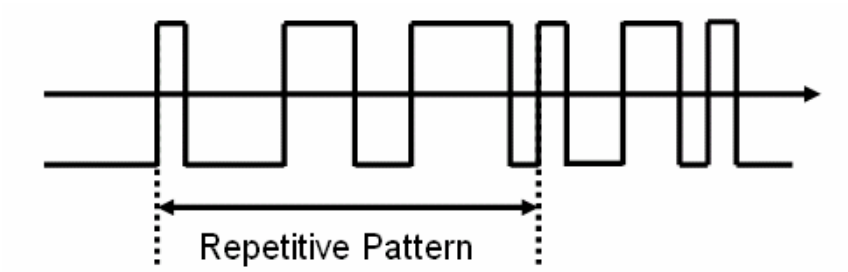


Figure 3.7 Pseudorandom binary sequence

As shown in Figure 3.8, there are sixteen master-slave flip-flops as the shift registers and an XOR gate to send the result to the input of the first flip-flop. Figure 3.9 shows HSPICE simulation results.

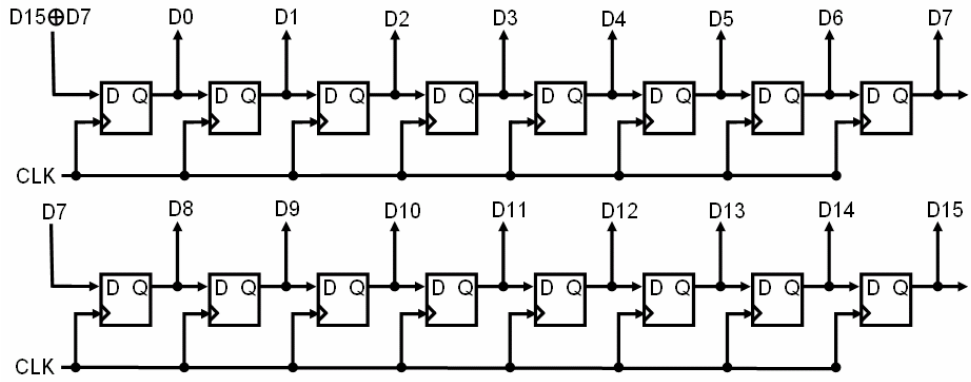


Figure 3.8 Linear feedback shift register

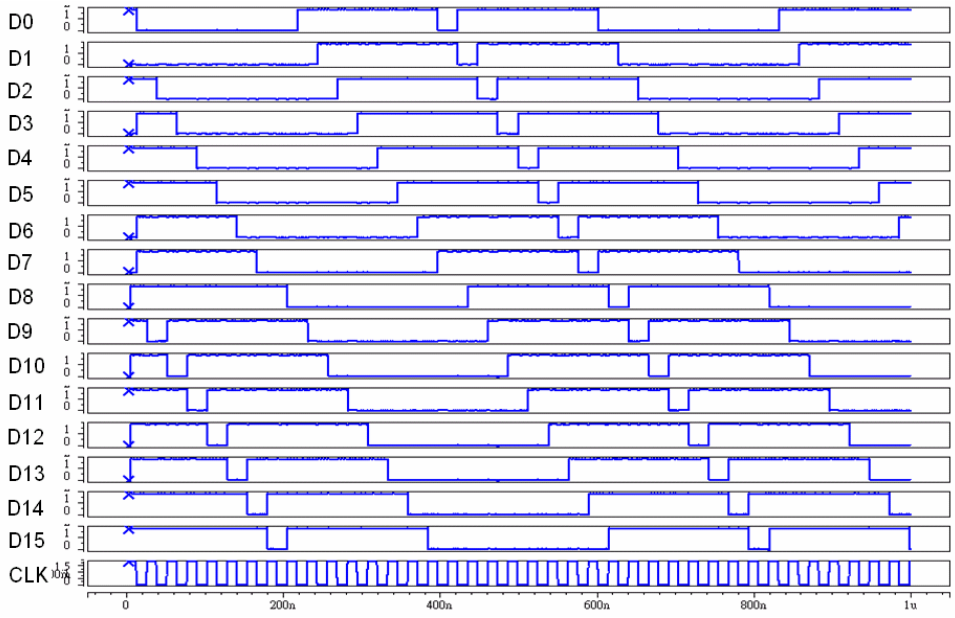


Figure 3.9 LFSR Simulation

A $2^{16}-1$ data pattern can be generated with sixteen registers and an XOR circuit. The characteristic of the PRBS architecture is that the probability of transitions from 0 to 1 and 1 to 0 are the same and the total numbers of 0s and 1s are different by one. It is a simple and regular structure. It can generate all possible combination patterns except all zero vectors. This technique can be extended to an m-bit system so as to produce a sequence of length $2^{m}-1$.

3.3.2 Parallel-In-Serial-Out Multiplexer

We take shift-register type serializer architecture as our multiplexer which is discussed in Section 3.2.3. Figure3.10 shows the serializer timing simulation. From this simulation information, we can preliminarily assure the circuit function is correct. Figure 3.11 shows the simulated eye diagrams of the data serializer.

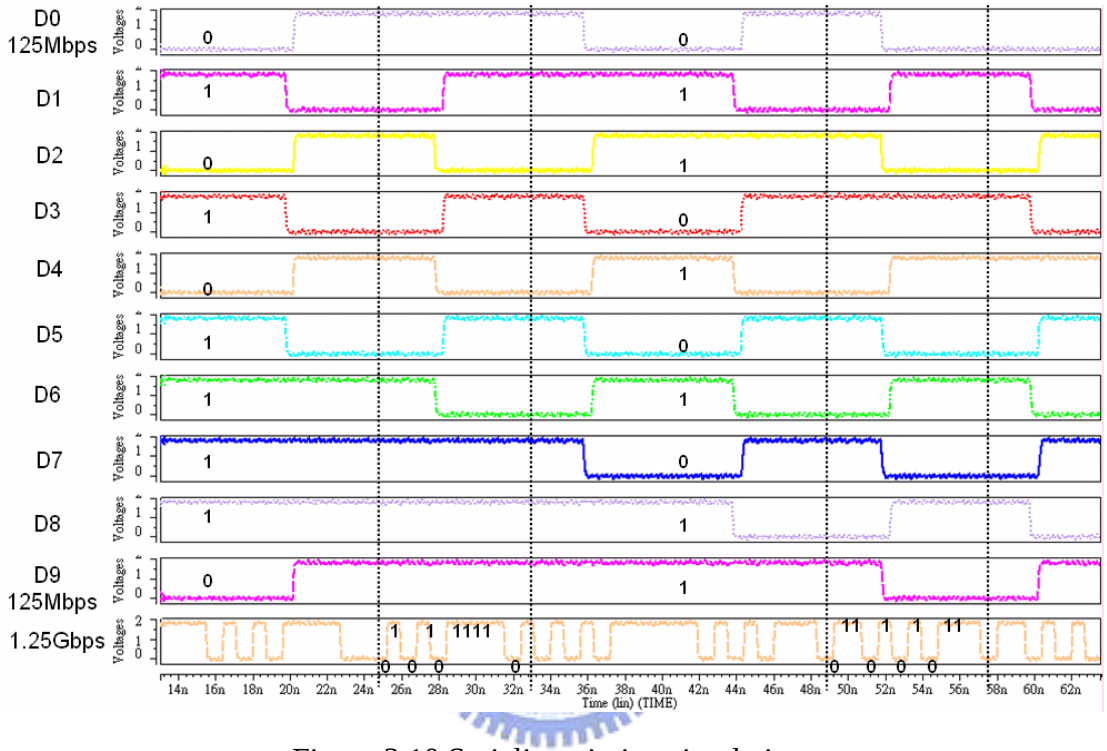


Figure 3.10 Serializer timing simulations

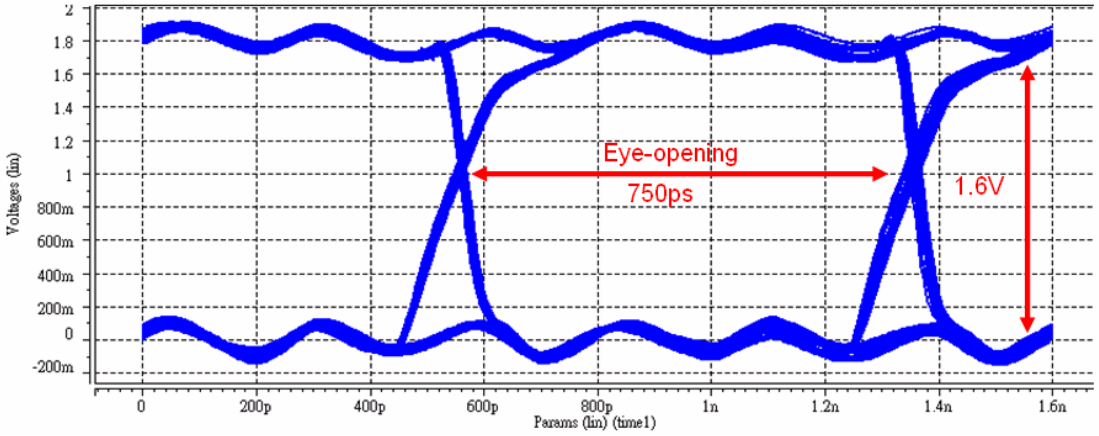


Figure 3.11 Simulated eye diagrams of the data serializer

3.3.3 LVDS Driver Architecture

This section roughly describes LVDS driver design in [8]. This driver part has been verified and has only 47ps peak-to peak jitter when operate at 1.25Gbps [8].

Overall architecture of the LVDS driver contains a single to differential buffer, a controlled pre-driver module, a fixed LVDS driver, and a programmable LVDS driver. In Figure 3.12, the first buffer converts the single ended data input to a differential one. The upper signal path is for the fixed driver and the lower one is for the programmable driver. The upper signals pass to the first six orderly turn-on buffers blocks. Note that, the signals are duty cycle controlled to reduce SSN as well. Besides, because of the process variation or layout mismatch, the output level is not guaranteed to have an expected voltage level. So the additional controllable LVDS drivers can enhance the output driver current to compensate the output voltage swing through the programmable LVDS driver as in Figure 3.12.

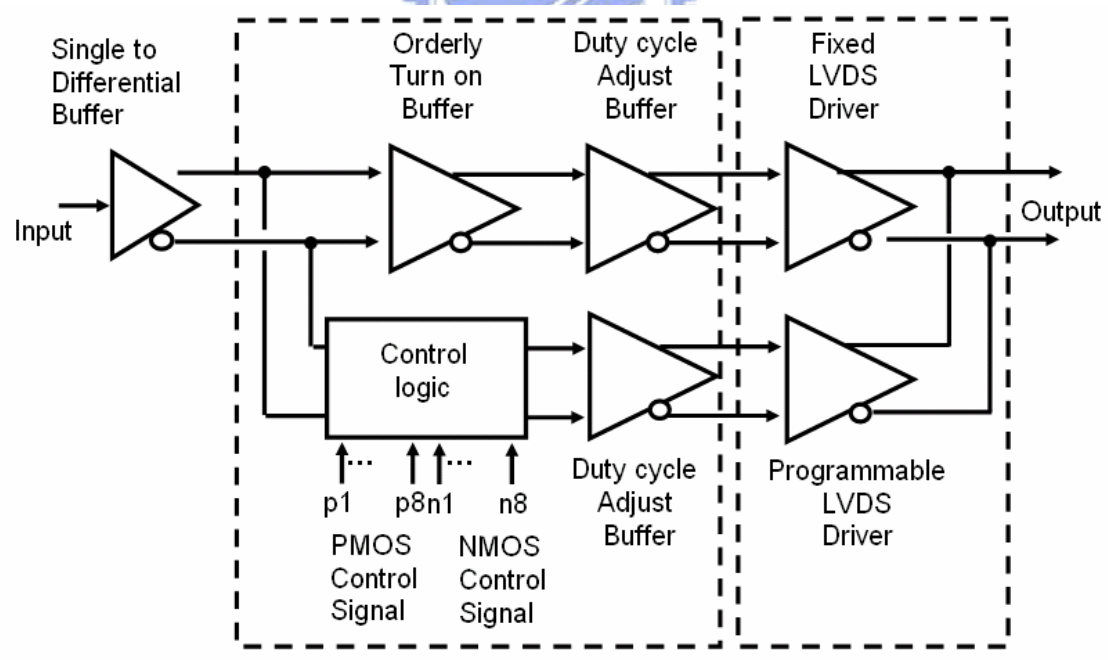


Figure 3.12 The overall LVDS driver architecture[8]

3.4 Measurement Result

Figure 3.13 and Figure 3.14 shows the PCB layout and the picture of the data serializer test board. Separated power supplies are applied for the I/O circuits. We separate the digital power and the analog power to avoid noise coupling and make the power measurement. Some off-chip SMD capacitors of 1nF to 1000nF are used in the vicinity of the core of the chip to serve as the bypass capacitors. The SMD capacitors bypass the differential outputs. The capacitor values are gradually increased when away from the chip. The switch is used to provide reset signal. The measure points are connected to instrument by SMA connections.

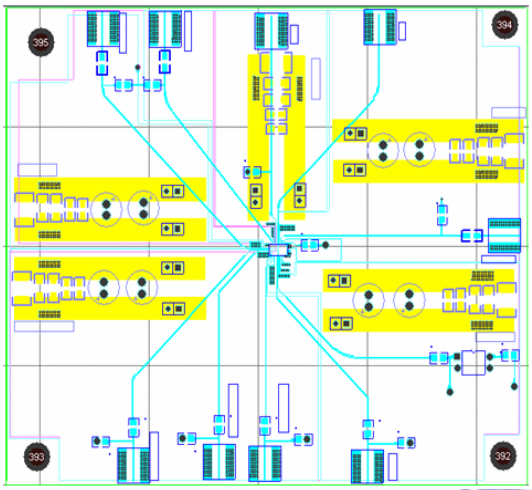


Figure 3.13 PCB layout

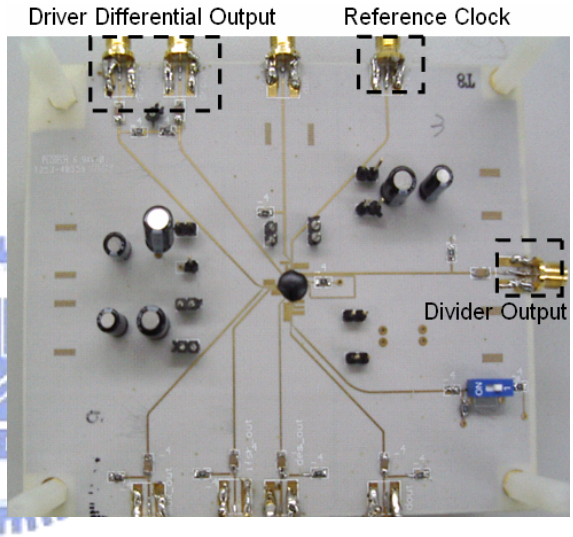


Figure 3.14 Test board photo

Figure 3.15 shows the PLL 1.25GHz VCO output waveform and Figure 3.16 shows the VCO output eye diagram. The RMS jitter is about 3.5ps and the peak-to-peak jitter is 30.6ps.

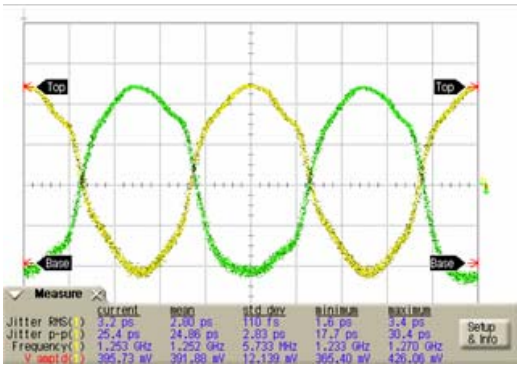


Figure 3.15 1.25GHz VCO output

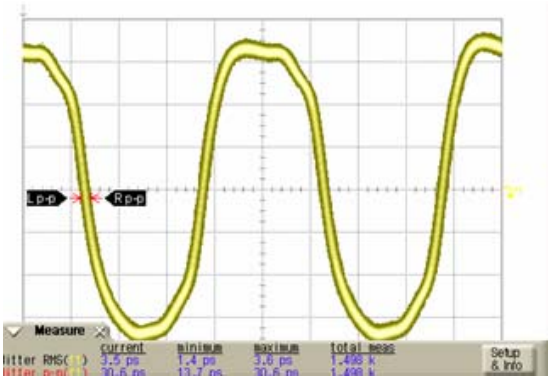


Figure 3.16 VCO output eye diagram

Figure 3.17shows the PLL 1.25GHz divider output waveform, and Figure 3.18 shows the divider output eye diagram. The RMS jitter is about 5.54ps and the peak-to-peak jitter is 45.56ps.

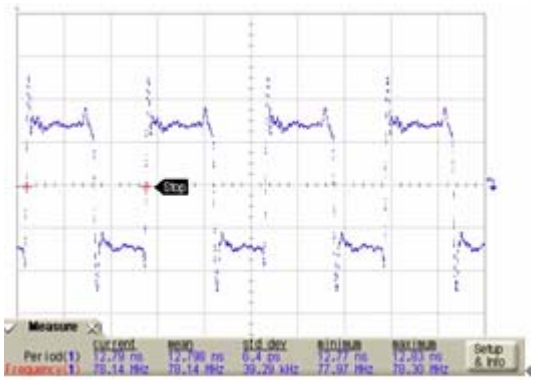


Figure 3.17 PLL divider output waveform

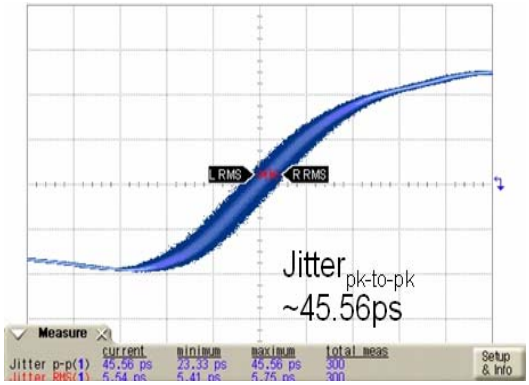


Figure 3.18 Divider output eye diagram

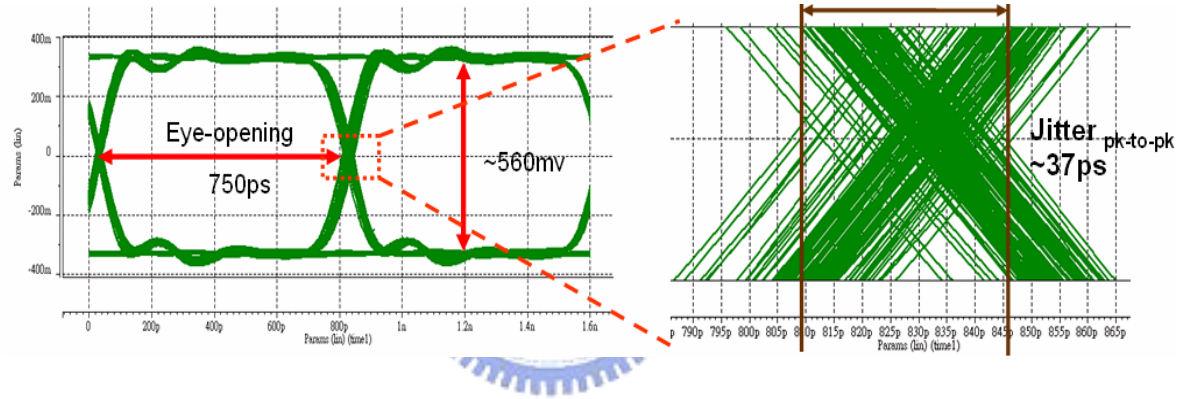


Figure 3.19 1.25Gbps overall transmitter differential output eye diagram

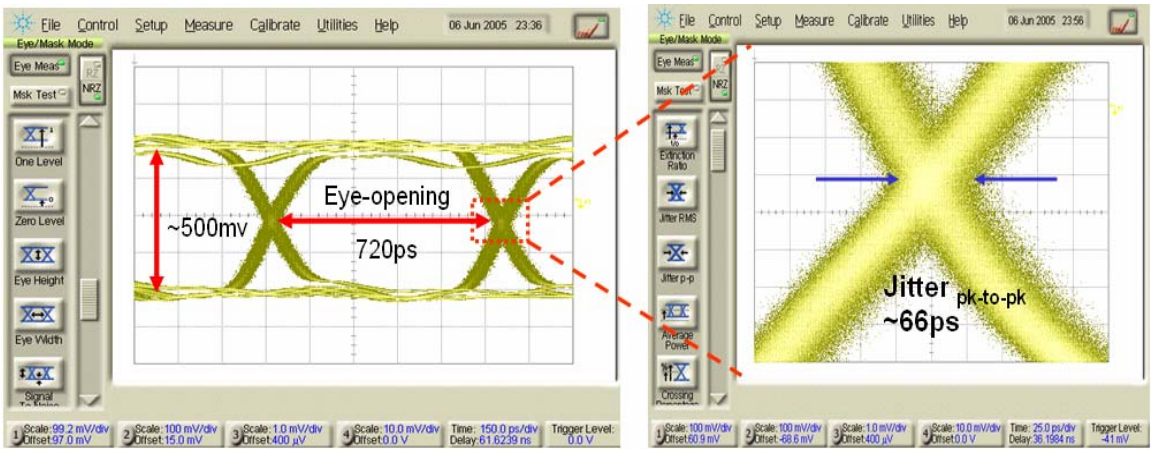


Figure 3.20 1.25Gbps overall transmitter differential output eye diagram (measurement results)

This chip was implemented in TSMC 0.18um 1P6M CMOS technology. We use Calibre tool for DRC, LVS, PEX verification. Figure 3.19 shows 1.25Gbps overall transmitter differential output eye diagram after post-layout simulation. Figure 3.20 shows the measurement result of 1.25Gbps overall transmitter differential output eye diagram. The peak-to-peak jitter is about 66ps and eye-amplitude is about 500mv.

Figure 3.21 shows the die photo of data serializer, including a fully-integrated multi-phase PLL, a serializer, a LVDS output driver and a built-in pattern generator (PG) for testing . The chip area is 1900um*990um and gate count is about 4333. The chip power consumption is about 50mW. The chip summary is shown in Table 3-1.

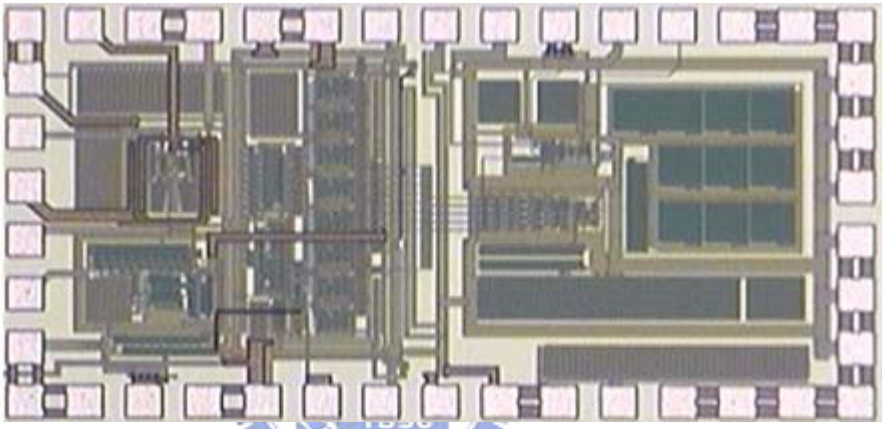


Figure 3.21 The overall chip photograph

Table 3-1 Overall chip summary of the transmitter

Function	*1.25Gbps Transmitter*
Technology	*0.18um 1P6M*
Chip size	*1900*990 (um²)*
Transistor Count	*4333*
Power Dissipation	*~50mW*
Jitter	*~66ps@1.25Gbps*
Function	*PLL*
Size	*800*670 (um²)*
Power Dissipation	*30mW*

This chip measurements are made using an Agilent 86100B wild-band oscilloscope. An Agilent 81130A pulse data generator is used for PLL reference clock generator. Figure 3.22 shows the measurement environment. HP E3610A DC power supply provides the required voltage sources for the test board. It has a measured differential output peak-to-peak jitter about 66ps at 1.25Gbps and eye-amplitude is about 500mV. It shows that the chip satisfies IEEE 802.3ah requirement.

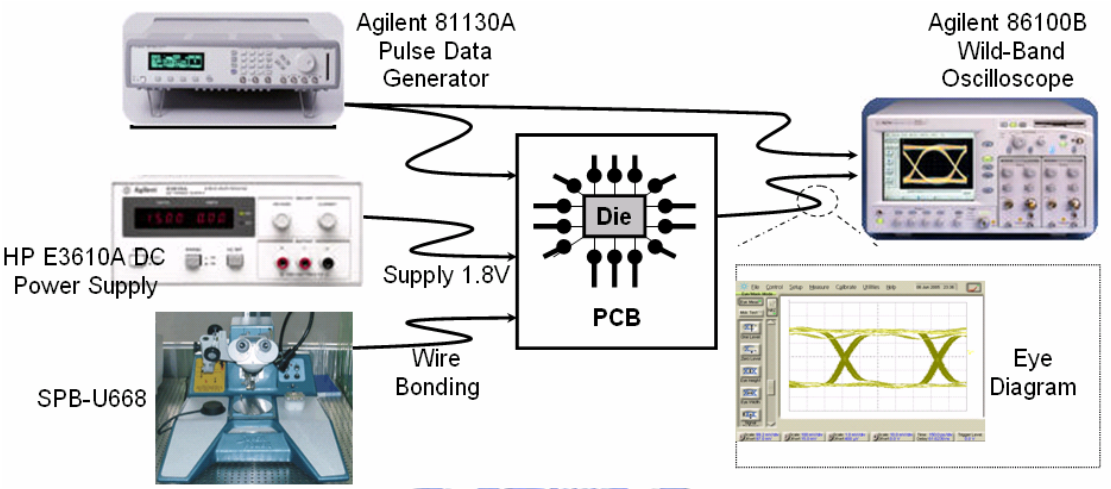


Figure 3.22 Measurement environment

Chapter 4

Digitally Controlled Multi-Phase Oscillator

4.1 Introduction of DCO

The heart of the ADPLL is a *digitally-controlled oscillator*（DCO）. Like most voltage-controlled oscillators, the DCO consists of a frequency-control mechanism within an oscillator block. Generally speaking, an odd number of inverters connected in a loop chain become a ring oscillator. The clock period of the ring oscillator is two times the circular loop delay time. Different propagation delay time of the inverter produces different clock period. Besides, a variable number of inverters implement a variable delay. Therefore, there are two parameters to determine the clock period of the ring oscillator. One is the propagation delay time of an inverter and another is the number of the inverters. In terms of tuning these two parameters, there are many designs of DCOs that have been presented.

A DCO structure is showed in Figure 4.1(a) [9]. The ADPLL controls the DCO frequency through the DCO control word. Arithmetically incrementing or decrementing the DCO control word modulates the DCO frequency and phase. The magnitude of the incremental changes to the DCO control word defines the gain, which, in turn, dictates the relative change in DCO frequency（△F/△DCO control word）. As Figure 4.1(a) shows, the requisite odd number of inverting stages in the DCO is obtained by using one enabling NAND gate and eight controllable cells. Figure 4.1(b) illustrates the basic premise of a constituent DCO cell. The sizing ratio of the control devices is 2X to achieve binary weighted control. Hence, the DCO cell can control the propagation delay time with n-bit control word. The disadvantage is the large area overhead.

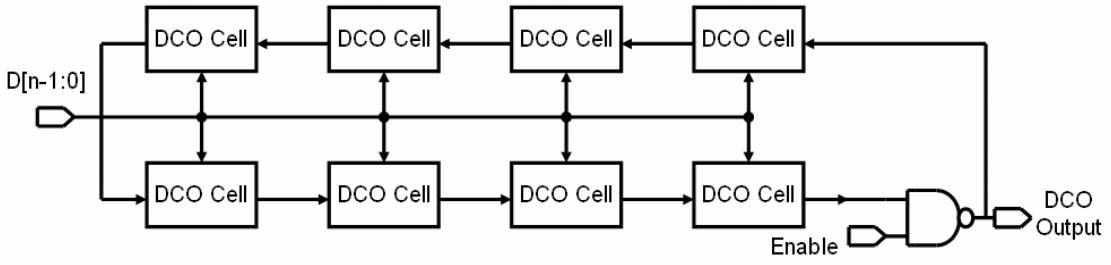


Figure 4.1（a）8-cell DCO with control bit [9]

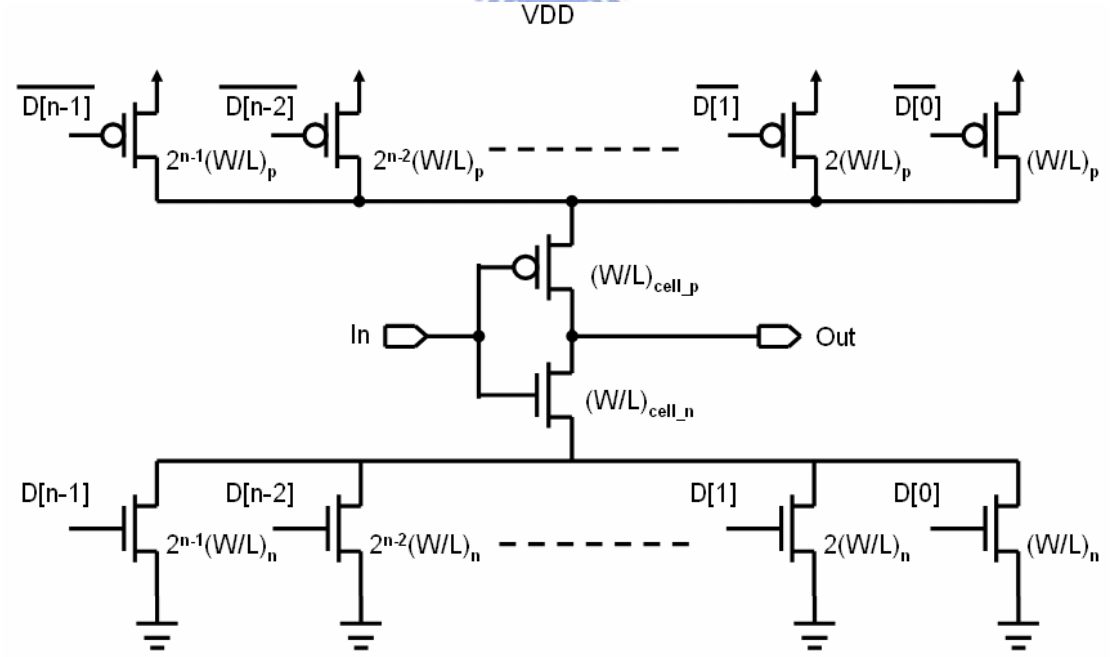


Figure 4.1（b） Constituent DCO cell [9]

Another DCO structure is showed in Figure 4.2[10]. The DCO consists of four paths and four DCO cells. The DCO cell is shown Figure 4.1(b). The path selection works as the coarse search. The DCO cells work as the fine tune. The area of the DCO in Figure 4.2 is smaller than the area of the DCO in Figure 4.1(a).

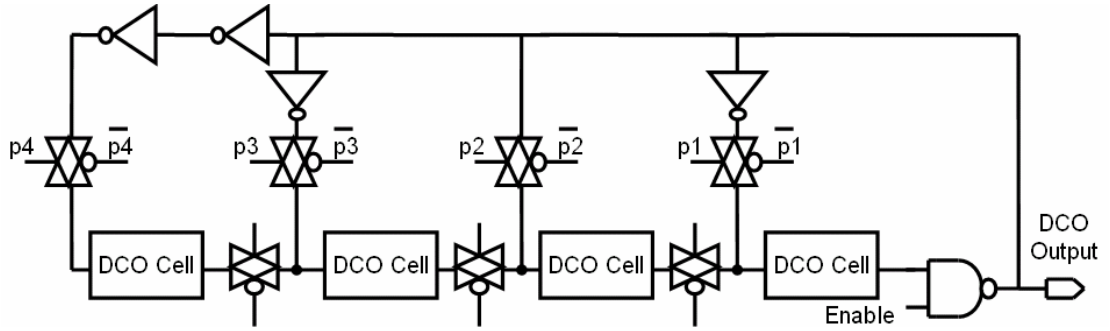


Figure 4.2 Structure of DCO in [10]

4.2 Standard-cell based DCO Design

Figure 4.3 shows a typical ring-oscillator which is very popular architecture in most ADPLL designs. The main advantage is that the oscillator can be implemented by standard-cell library.

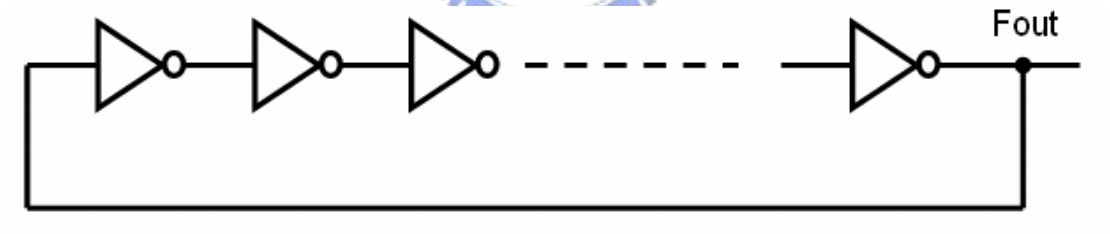


Figure 4.3 A typical ring-oscillator

The modified architecture is showed in Figure 4.4 [11]. The enable signal is used to power up the ring-oscillator. The path selection consists of tri-state inverters. The path selections from p1 to p4 are used to select different delay time of ring-oscillator to change output frequency. The problem of the architecture is a large parasitic capacitance at node 1 in Figure 4.4, and the DCO resolution is poor due to no fine tune cell being applied.

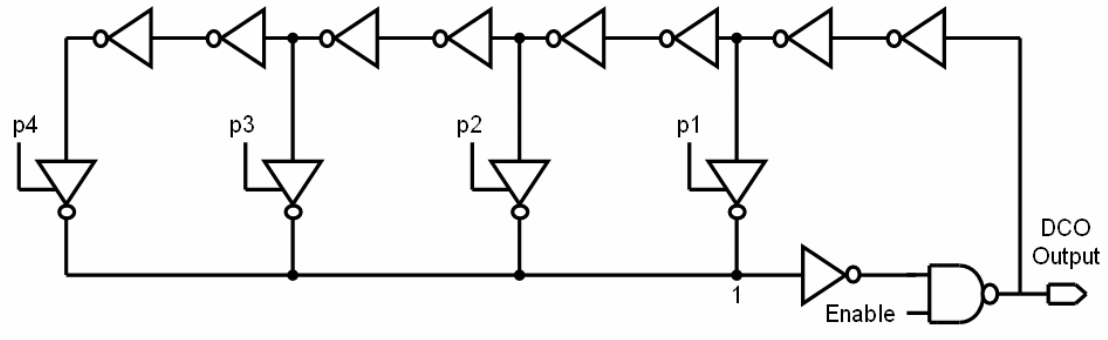


Figure 4.4 Modified architecture in [11]

An improved DCO architecture is showed in Figure 4.5 [12]. It is separated into two stages: a coarse-tuning stage and a fine-tuning stage. To avoid large loading capacitance appearing in the path selection output, the path selector is partitioned into two stages. In the first stage, sixteen coarse-tuning delay blocks select a partial output. The second stage path selector will select the final output.

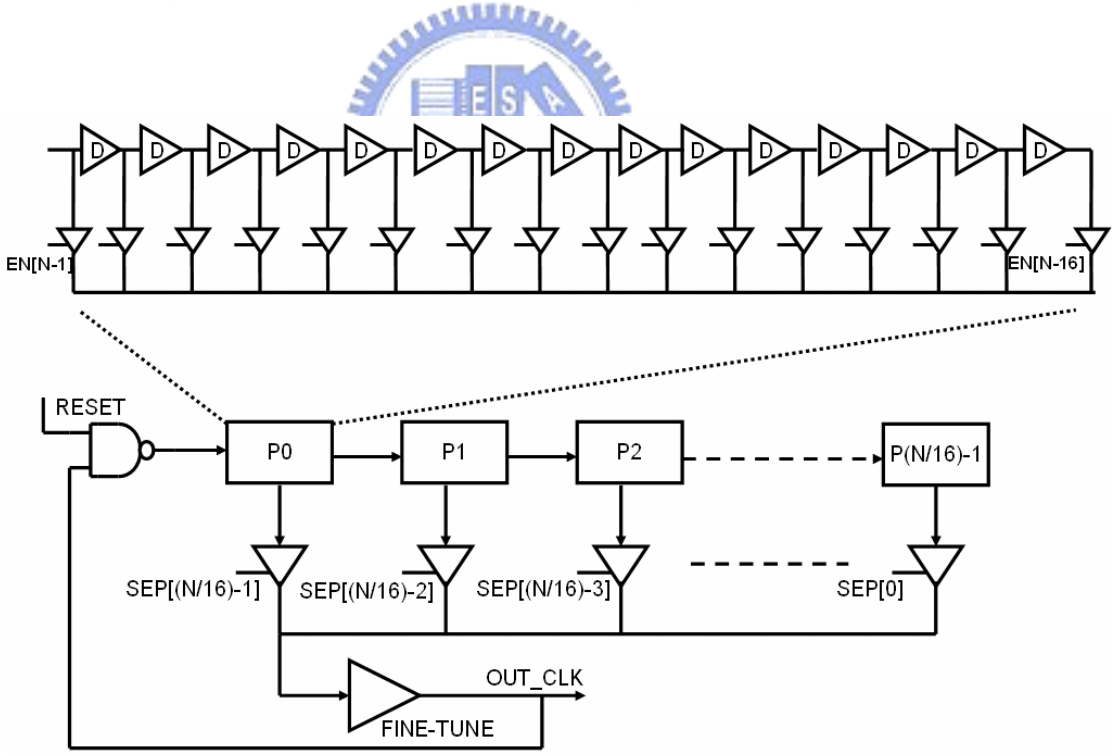


Figure 4.5 An improved DCO architecture in [12]

A cell-based digital controlled oscillator is shown in Figure 4.6[13]. It is a multiple path selection DCO with a delay matrix. When searching frequency, the path selection works as coarse search and the delay matrix works as fine search. The delay matrix consists of several parallel tri-state inverters. Its disadvantage is the large parasitic capacitance in the output node of the path selection.

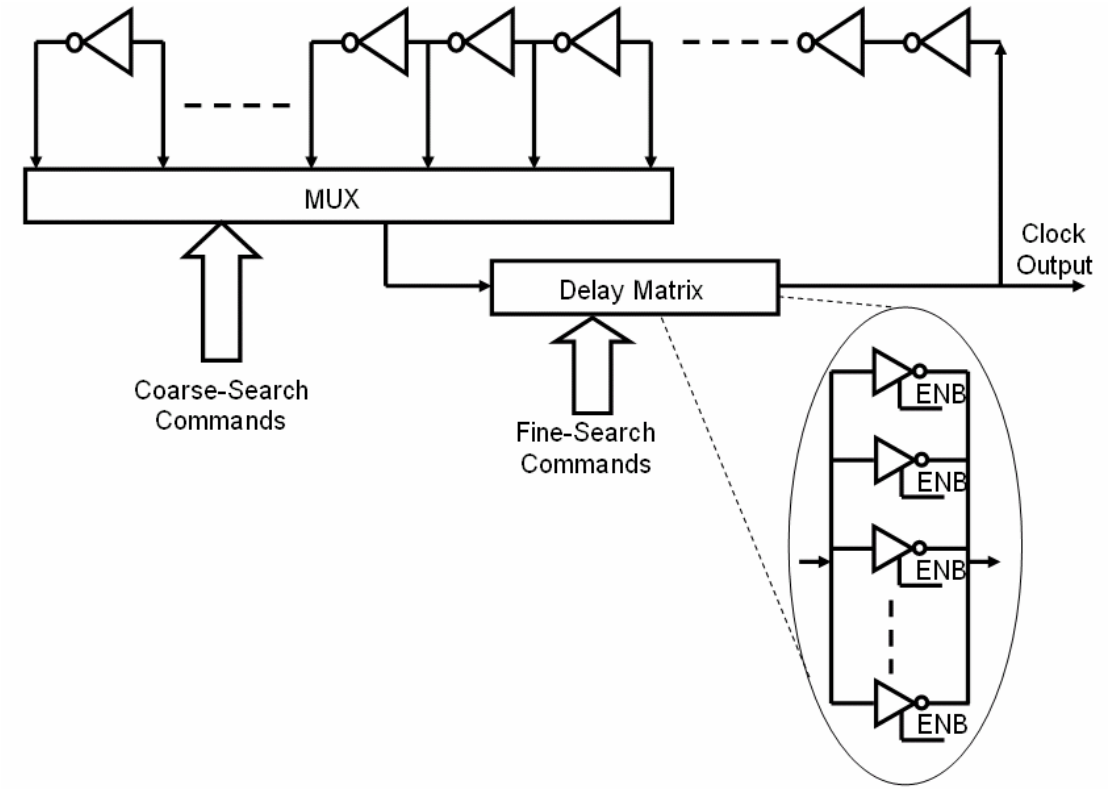


Figure 4.6 Structure of the cell-based DCO in [13]

Another cell-based digital controlled oscillator is shown in Figure 4.7[14]. The oscillator being implemented is a seven-stage ring oscillator with one inverter replaced by a NAND gate for shutting down the ring oscillator during idle mode. To change the frequency of the ring oscillator, a set of 21 tri-state inverters are connected parallel to each inverter. When the tri-state inverters are enabled, additional current is added to drive each inverter stage. Although the DCO has the advantages of being made from all-standard cells, it has disadvantages such as relatively high power consumption and low maximum frequency from high capacitive load in the ring oscillator.

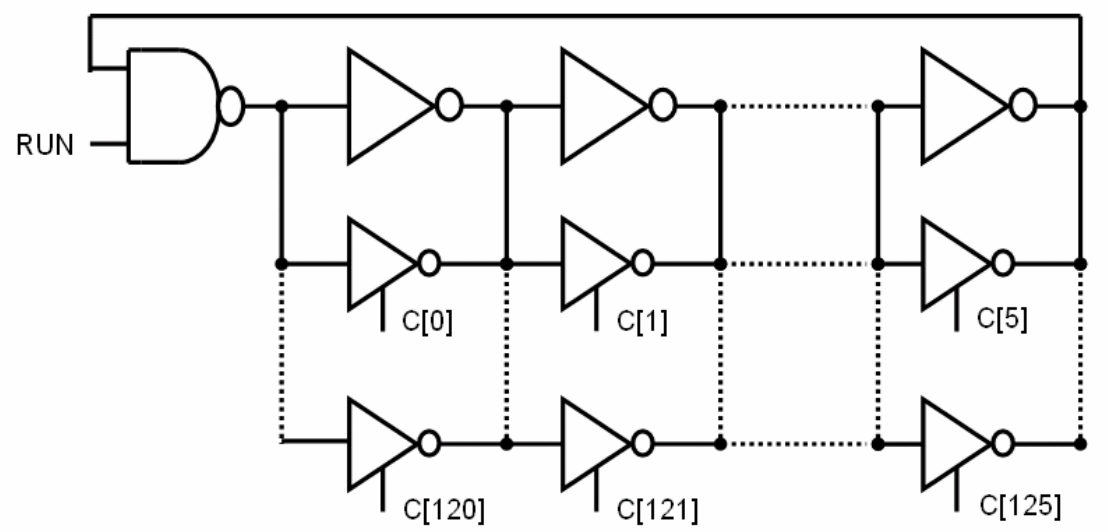


Figure 4.7 DCO using parallel tri-state inverters to adjust frequency in [14]

As we all known, the Figure 4.7 design has the disadvantages of high power consumption and low maximum frequency. However, our design applications demand high resolution, high speed and low power consumption. Hence, a fully-custom design of the DCO is preferable. Our design target is 1.25GHz 8-phase DCO. We propose a high-speed multi-phase output DCO with differential operation as shown in Figure 4.8. The idea for the oscillator comes from this paper in [15] .It is a low phase noise ring oscillator with differential control and quadrature outputs. We modify this structure to be digitally controlled and eight phase outputs. As shown in Figure 4.8, an oscillator is composed of four delay elements. The oscillation frequency depends on the equivalent load resistance and the equivalent capacitance of the delay element. By tuning the equivalent capacitance, we can obtain the desired oscillating frequency.

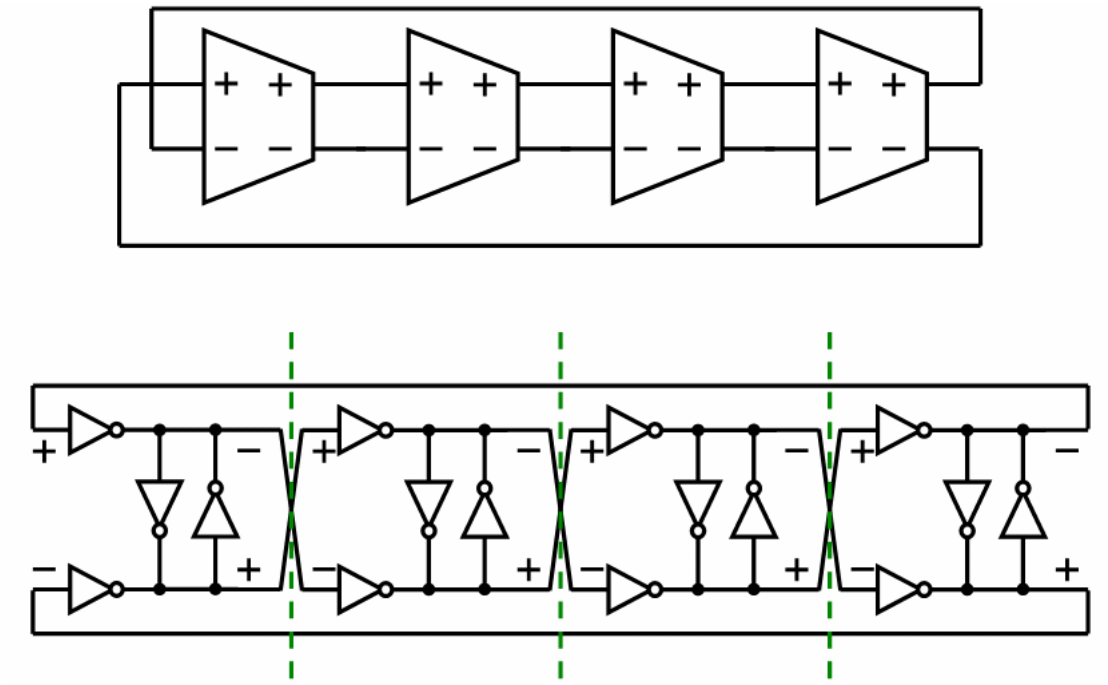


Figure 4.8 The proposed differential operation ring oscillator

Generally speaking, the usage of high-speed differential operation has the disadvantages in doubly the area and power consumption. The advantage is the higher noise immunity. Hence, there is a trade-offs between the noise and the power consumption. Refer to [15], we add latchs between two signal paths. The latch is composed of two minimum size inverters. The advantage of the use of latch is its small area. The way of adding small latch forces two signal paths differential is called “pseudo- differential”. Both sides of latch will force two signal paths to be 180 degree out of phase. The obvious improvement of this method that is area and power will not be doubled and the signals are fully-differential.

4.3 Design of High Resolution Delay Cells

In analog delay cell, the delay cell is controlled by the voltage or current, and the output delay time is continuous over the tuning range. But for digitally controlled delay cell, the output delay time is quantized. And the resolution of the output delay time must be sufficient small enough to maintain acceptable jitter.

The clock period difference between each neighbor value of control word is defined as the resolution of the delay cell. If an inverter-based delay line is used in delay cell design, the delay cell produces different propagation delays by selecting different number of inverters. Therefore, the resolution of the delay cell is limited by the delay time of one inverter. Moreover, this resolution is often not sufficient to be used in ADPLL design.

[13] propose a high-resolution delay cell as shown in Figure 4.9. It is an inverter-matrix structure, whose characteristics are determined by both combination and number of inverter banks. The resolution is decided by minimum scale of inverter bank.

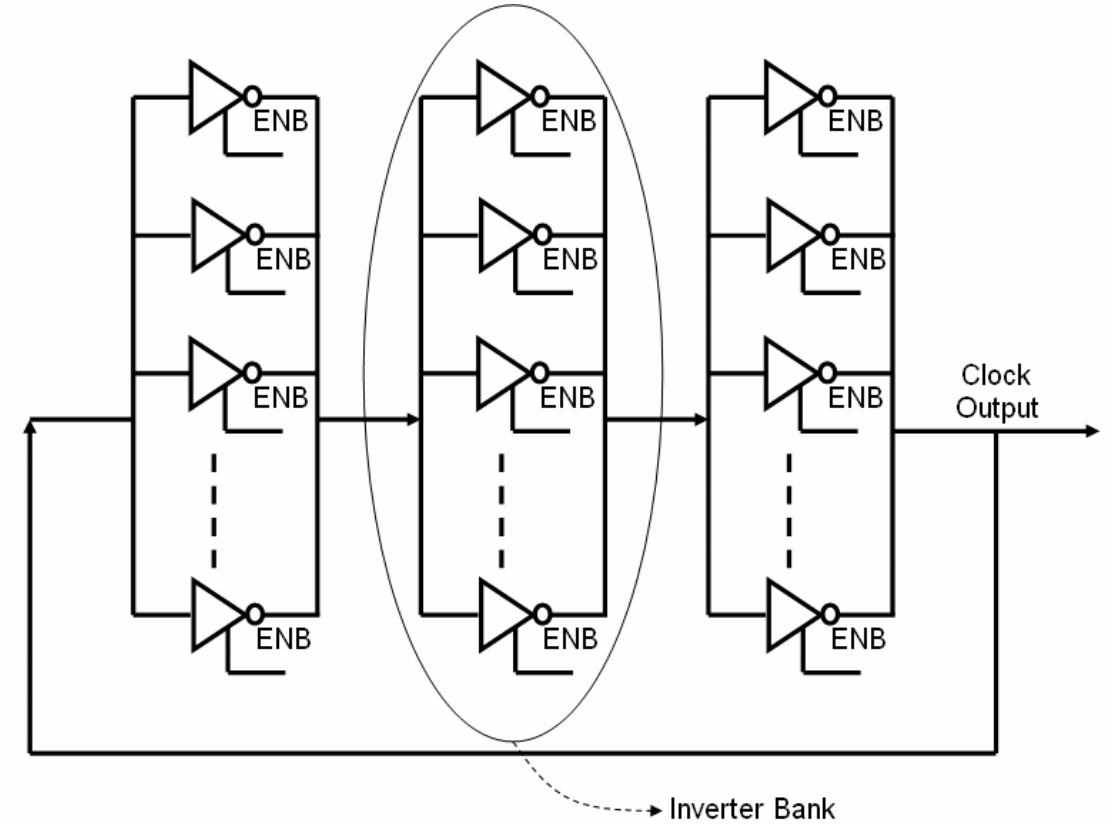


Figure 4.9 A proposed high-resolution delay cell in [13]

Another proposed high resolution delay cell [12] is showed in Figure 4.10. It can be used to perform fine-tuning in ADPLL design. Both AOI type delay cells and OAI type delay cells are shunted with two tri-state buffers. Shunted tri-state buffers are used to increase the controllable range of the high resolution delay cell.

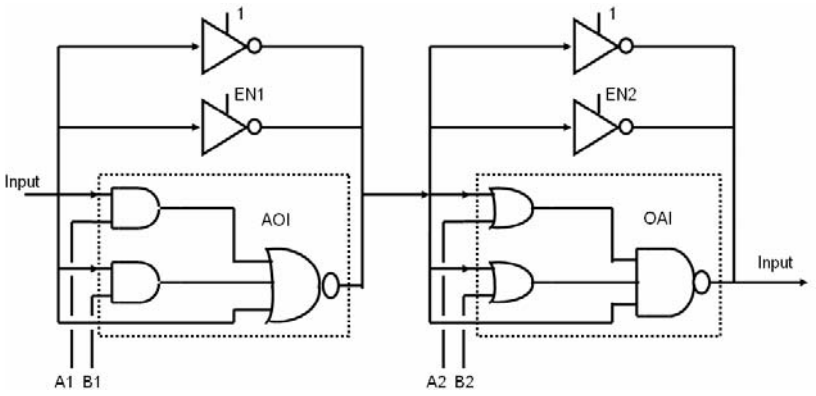


Figure 4.10 High resolution delay cell in [12]

Thus in previous design, the delay matrix uses parallel tri-state buffers to enhance the resolution of the delay cell. However, the area cost and the power consumption for the delay matrix is too large to be used in a low cost and low power design. In [12], only six standard cells are used, thus its area cost and power consumption is low. The average resolution of the proposed delay cell is about 5ps.

As mentioned before, the fine-tuning method all fix capacitor value and change current value to modulate frequency. However, we propose a method of changing capacitor value to modulate frequency with fixed current value. When MOS is “ON”, the gate capacitance $C_{gs} = C_{ovs} + \frac{1}{2}WLC_{ox}$, $C_{gd} = C_{ovd} + \frac{1}{2}WLC_{ox}$. When MOS is “OFF”, the gate capacitance $C_{gs} = C_{ovs}$, $C_{gd} = C_{ovd}$.

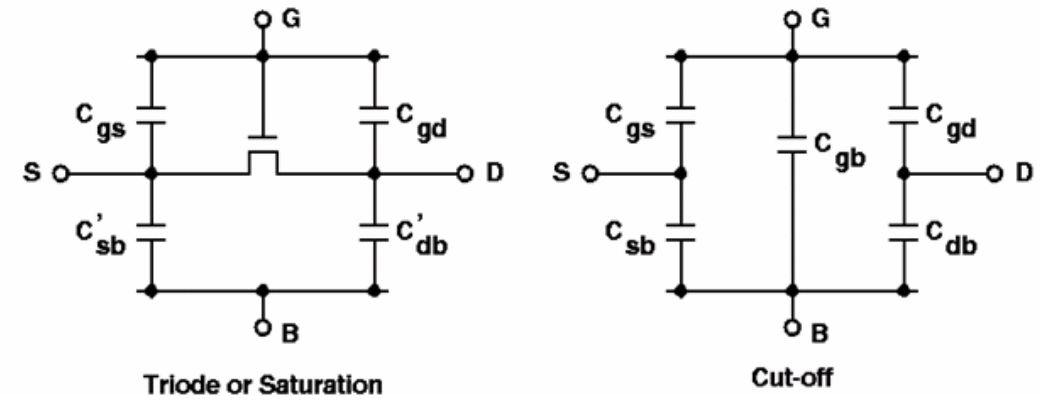


Figure 4.11 Parasitic capacitance in triode and cut-off

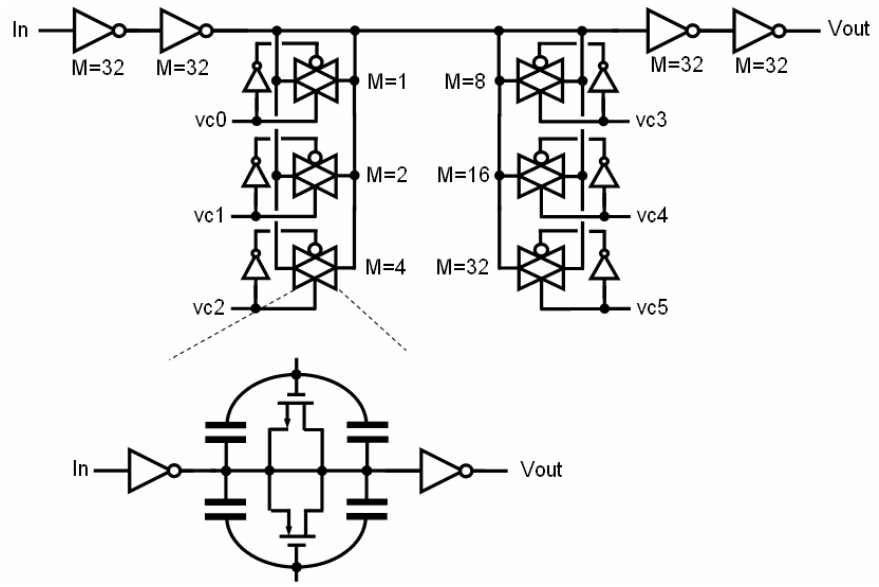


Figure 4.12 The proposed high resolution delay cell

As shown in Figure 4.12, we use the difference of equivalent capacitor value from drain with MOS “ON” or “OFF” to modulate high timing resolution. When MOS is “ON”, the equivalent capacitor value from drain will be larger. When MOS is “OFF”, the equivalent capacitor value from drain will be smaller. We assume the fine-tune section has a 6-bit control word and adopt a binary weighted control mechanism.

The simulation result high resolution delay cell are shown in Figure 4.13 Fine-tune input control word rangy from 000000 to 111111, thus, a total of 64 different delays can be provided. In this simulation, the average resolution is about 125.7fs= (8.05ps/64). At the same time, delay time versus control word is very linear. For gigabit optical communication, this resolution is quiet sufficient.

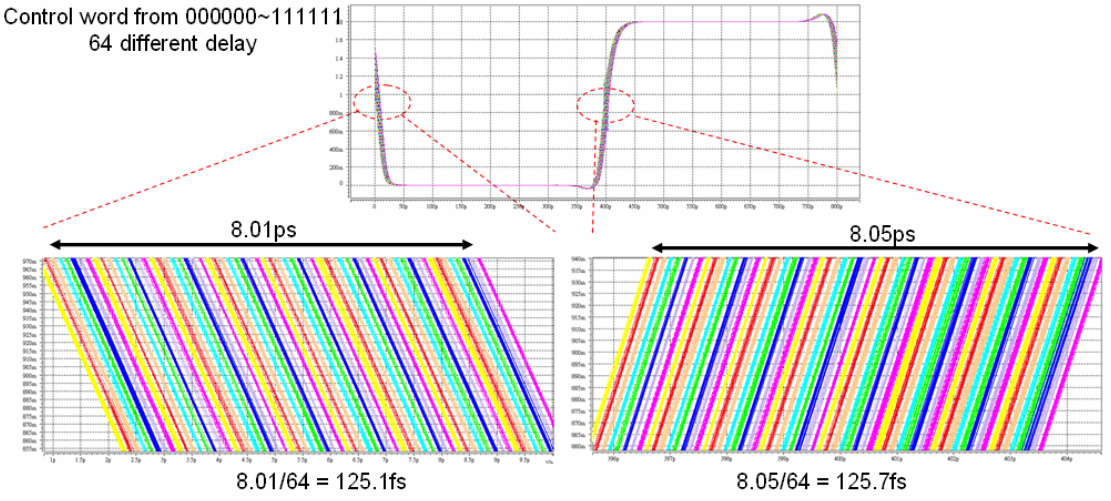


Figure 4.13 High resolution delay cell simulation result

4.4 The Proposed DCO Architecture

As mentioned in Section 4.2 and 4.3, we combine these two techniques into our DCO architecture. The proposed cell-based DCO architecture is shown in Figure 4.14. In the test chip, the DCO is implemented with TSMC 0.18um 1P6M CMOS process. It is separated into two stages, the coarse-tuning stage and fine- tuning stage.

In the coarse-tuning stage, we change the loading variance through control words. Coarse-tuning stage is composed of several digitally controlled latches. Different loading will result in different oscillating frequency. Hence, it can be easily modified to meet different specifications for different applications.

To increase frequency resolution of the DCO, fine-tuning delay cells are added. The circuit of fine-tuning delay cell is shown in Figure 4.12. The detail information about how to design the fine-tuning delay cell is discussed in Section 4.3.

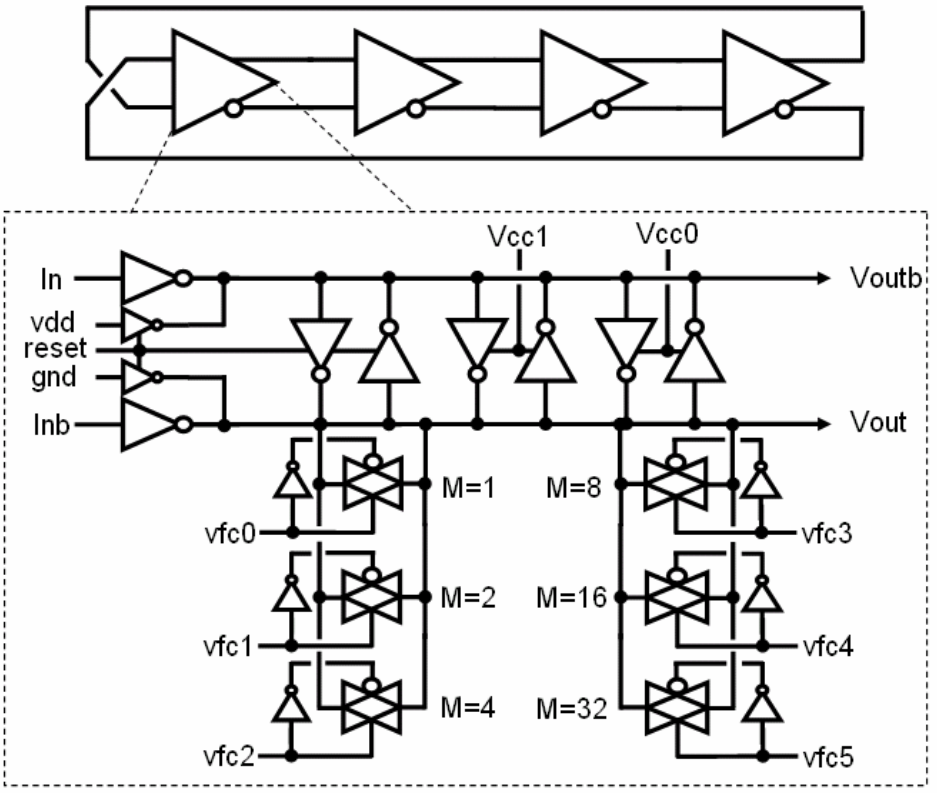


Figure 4.14 The proposed cell-based DCO architecture

4.5 Analysis and Design of the proposed DCO

There are two issues in designing the desired DCO; a sufficiently high resolution to maintain acceptable jitter, a monotonic response of the control word to the oscillation frequency, and high noise immunity.

The first issue is a sufficiently high resolution to maintain acceptable jitter. We all known that the longer the length of control word is, the higher the resolution will be.

The second issue is a monotonic response to the control word. Theoretically, the more controlled MOSs turned on, the larger the clock period will be. Hence, the DCO approaches a monotonic response to the control word. In practice, when the DCO is in the steady state, the oscillated clock has a small variance, called the intrinsic period jitter. Besides, the process variation probably increases this intrinsic period jitter. We can smooth this intrinsic period jitter variation by a control unit. We will discuss this part in Chapter 5.

The clock period control range of our DCO consists of several sections. The control range of each control word determines the clock period of each section. The control range must be larger than the clock period difference between each two neighbor control words. Otherwise, there are frequency gaps in each two neighbor control words. Moreover, in order to increase the tolerance of the process variation, the overlap of the clock period must be large enough.

This basic concept of the clock control period range is illustrated in Figure 4.15. Having these basic concept and design issues, we can design the DCO in detail based on above discussions.

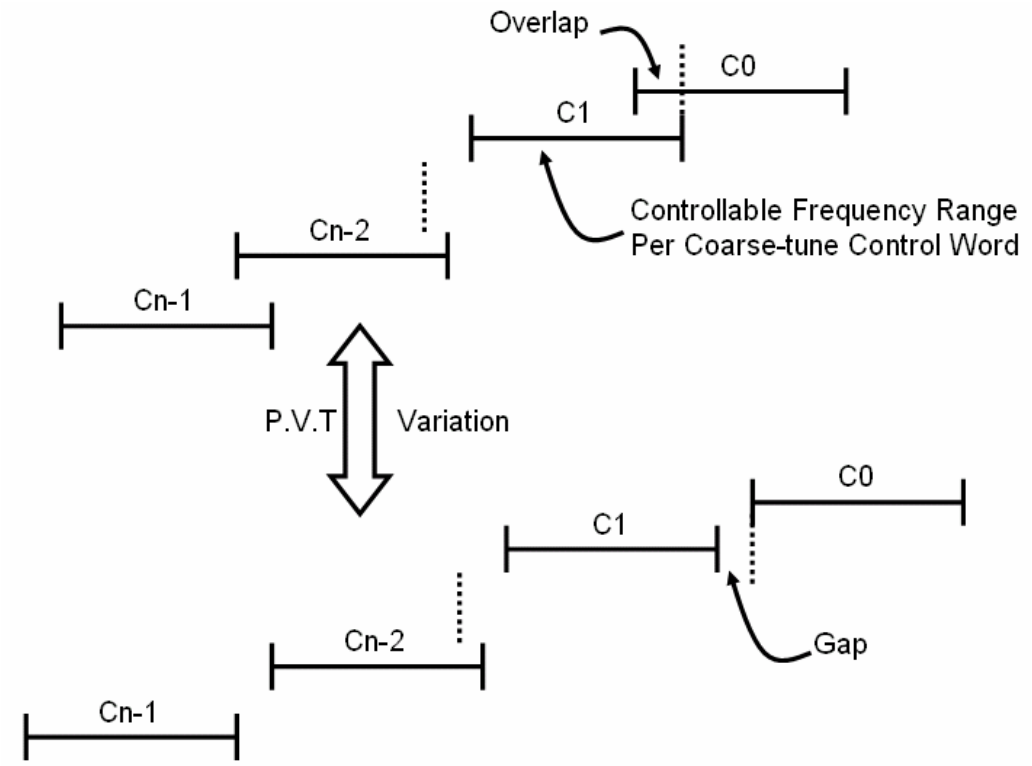


Figure 4.15 Influence of PVT variations on the controllable frequency range

4.6 Process Variation Consideration

As mentioned in Section 4.2, we can obtain the desired oscillating frequency by tuning the equivalent capacitance. In order to cover the process and temperature variation, the digitally controlled oscillator should have a large operation range. Owing to many trade-offs between large tuning range and desired operation clock period, we develop a standard procedure to determine the DCO size called the normalization process.

As Figure 4.16 shown, a simple structure is used to illustrate the normalization process. We normalize the main fixed inverter to be “one”. Then we change the loading range of the coarse-tune stage from 0.1 to 1.5 and similarly change the loading range of the fine-tune stage from 0 to 6.3.

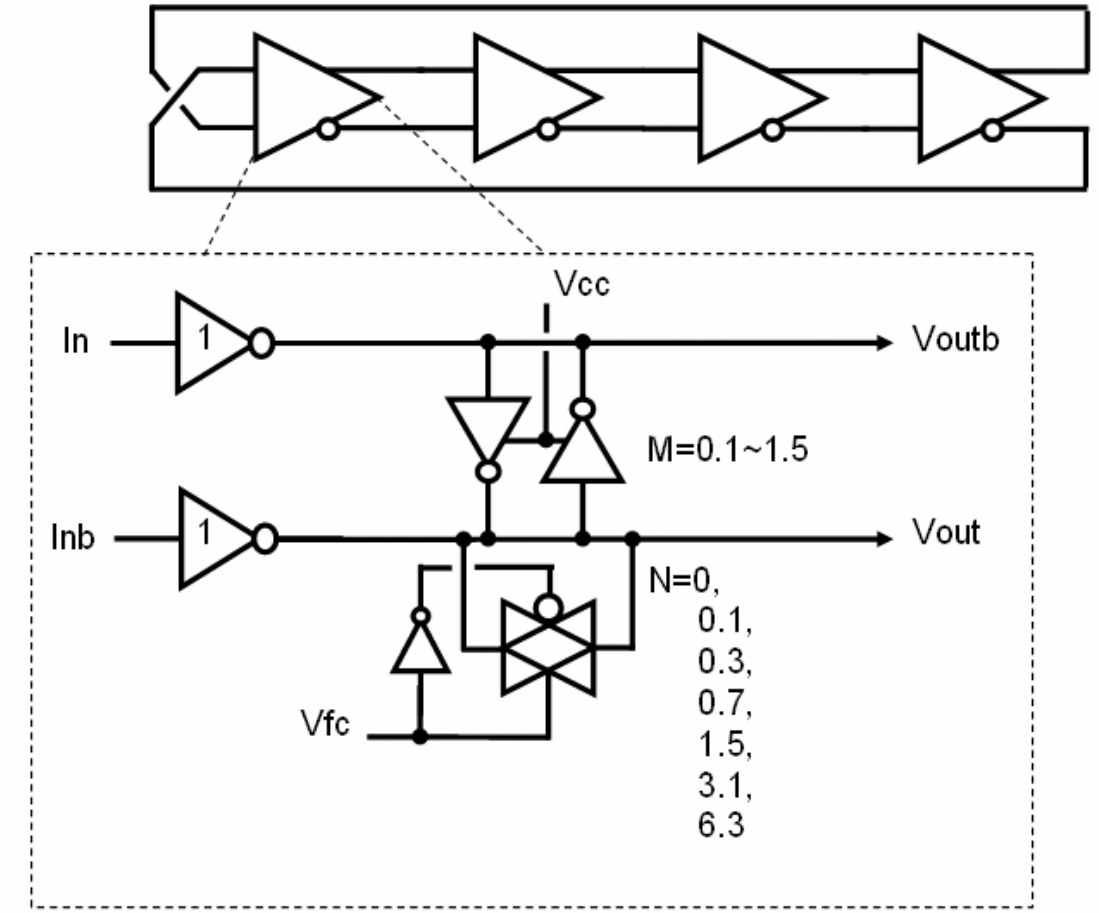


Figure 4.16 The normalization of DCO structure

Figure 4.17 shows The DCO output clock period simulation results when all control pins are all ones. The clock period of DCO output being greater than 970ps is what we want to meet process variation as Figure 4.17 shows.

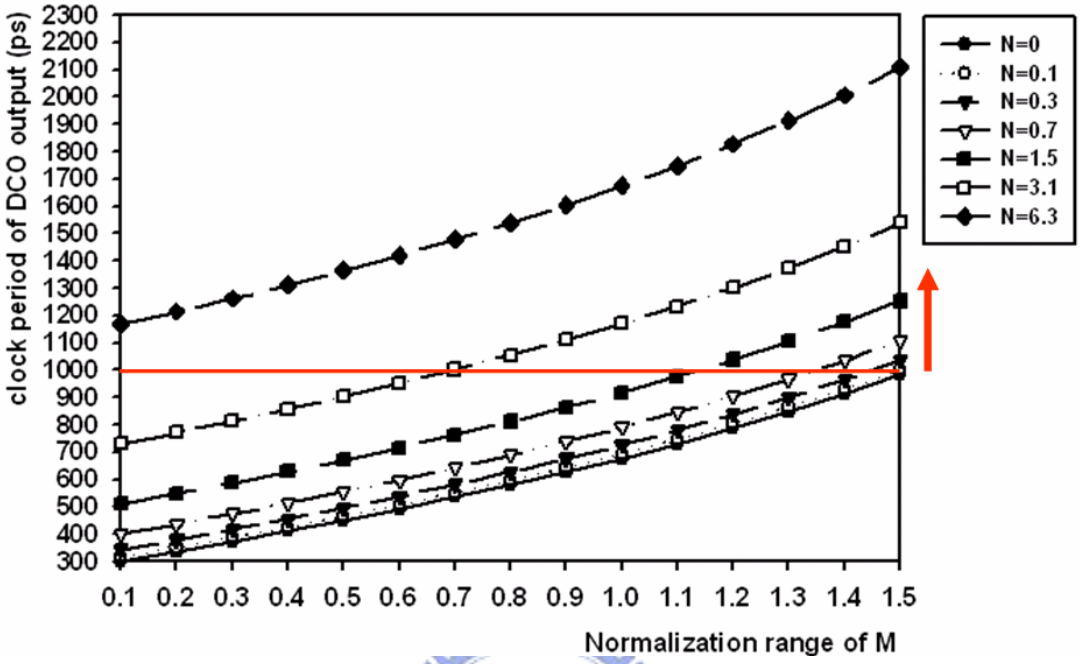


Figure 4.17 The DCO output clock period simulation result when all control pins are ones

Figure 4.18 shows the DCO output clock period simulation results when all control pins are zeros. The clock period of DCO output being smaller than 640ps is what we want to meet process variation as Figure 4.18 shows.

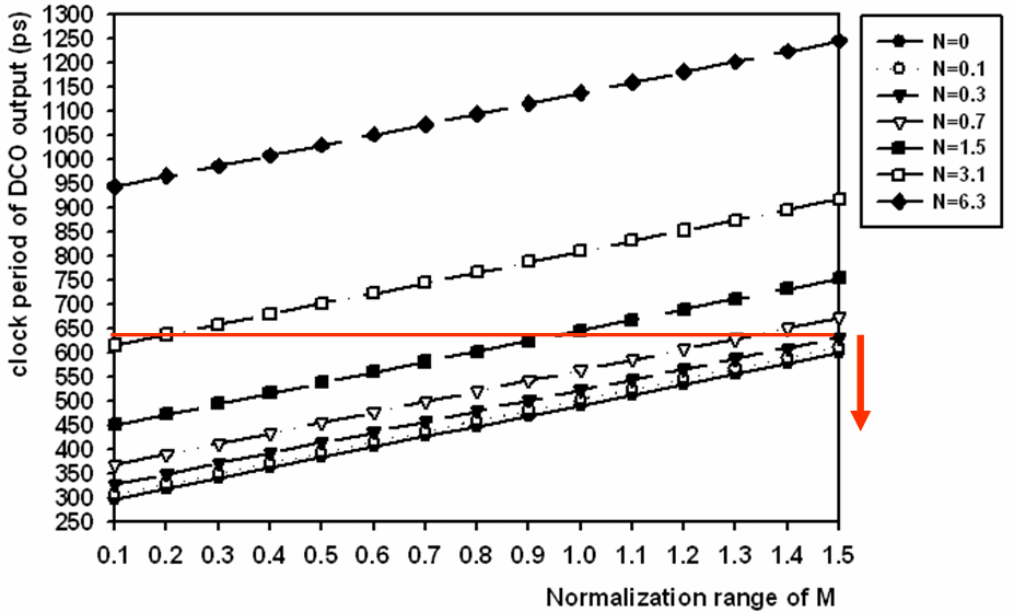


Figure 4.18 The DCO output clock period simulation result when all control pin are zeros

Figure 4.19 shows the tuning range simulation results at different fine-tuning loading condition. The rectangular range being marked is the desired DCO output clock range. In order to find out more precise ratio relationship, we take the rectangular range for more detail SPICE simulation.

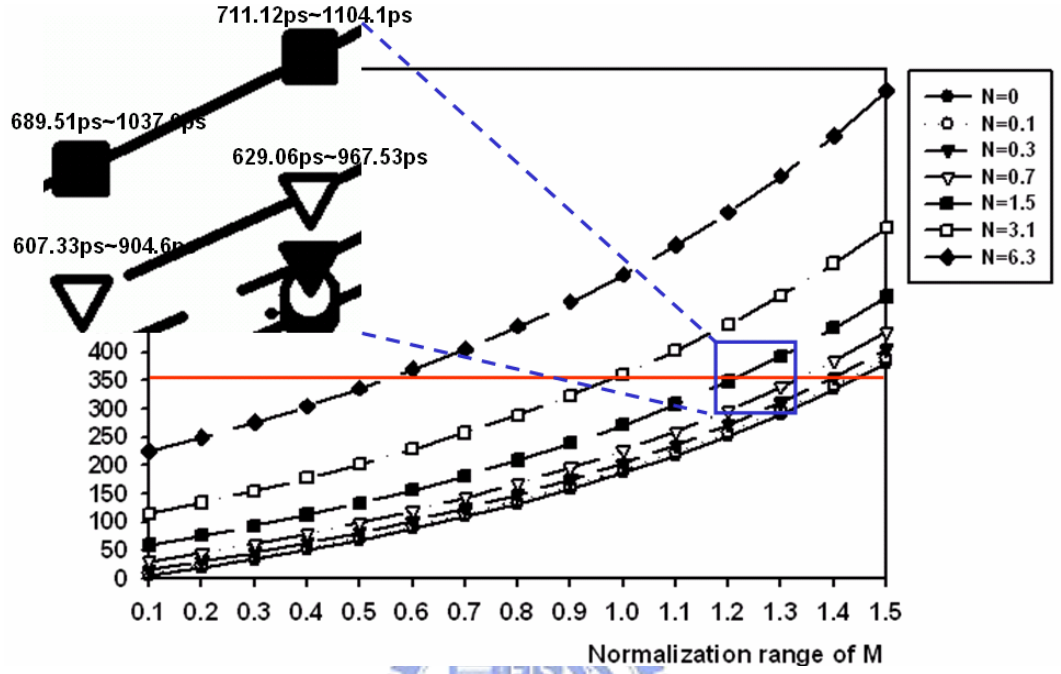


Figure 4.19 Tuning range simulation result at different fine-tuning loading value

Figure 4.20 shows the enlarged DCO output clock period simulation results when all control pins are ones. The clock period of DCO output being greater than 970ps is what we want as Figure 4.20 shows.

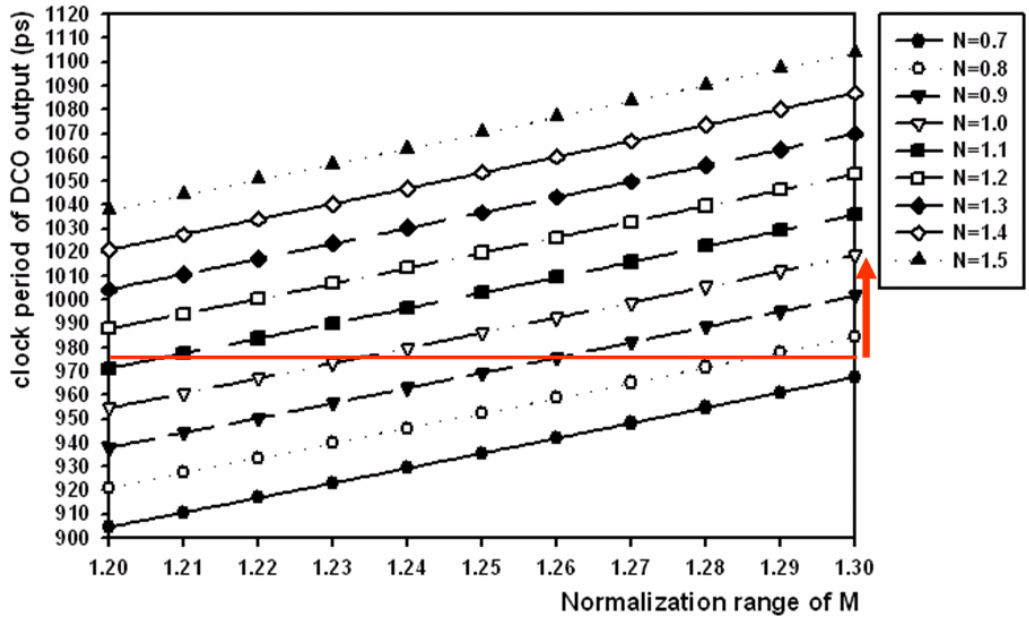


Figure 4.20 The enlarged DCO output clock period simulation result when all control pin are ones

Figure 4.21 shows the simulation results when all control pin are zeros. The clock period of DCO output being smaller than 640ps is what we want as Figure 4.21 shows.

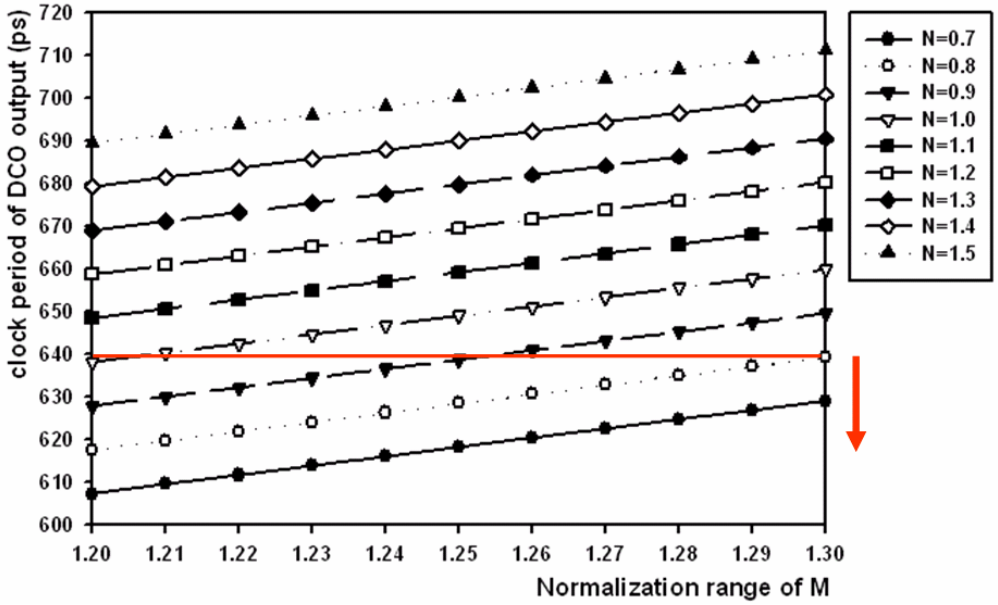


Figure 4.21 The enlarged DCO output clock period simulation result when all control pin are zeros

In Figure 4.22, there are most points satisfy our demands which include wide tuning range and wanted operation range. However, when N=0.8 we find the clock control range of each section is not wide enough that results in some gaps in SPICE simulation. Hence, we choose N=0.9 as our fine-tune delay stage ratio to avoid gaps.

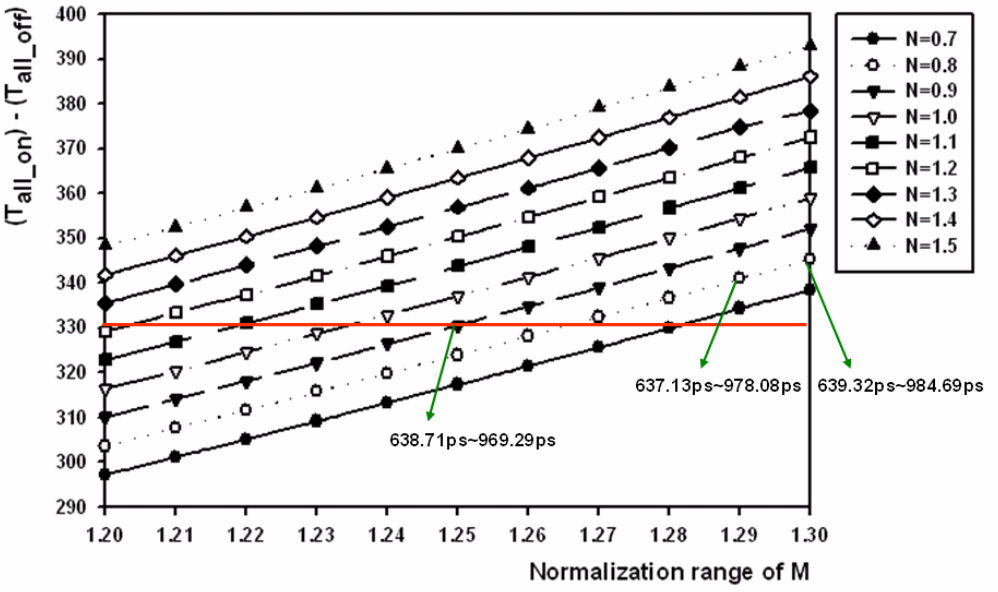


Figure 4.22 Tuning range sim. at different fine-tuning loading value from 0.7 to 1.5

Then, we want to find the ratio of parameter M when N=0.9. We slowly add the ratio of M from 0.1 to 1.5 to observe simulation results. As Figure 4.23 shown, we mark the desired range of the upper bound and the low bound, and we can find the ratio of M=1.25 satisfying our marked range.

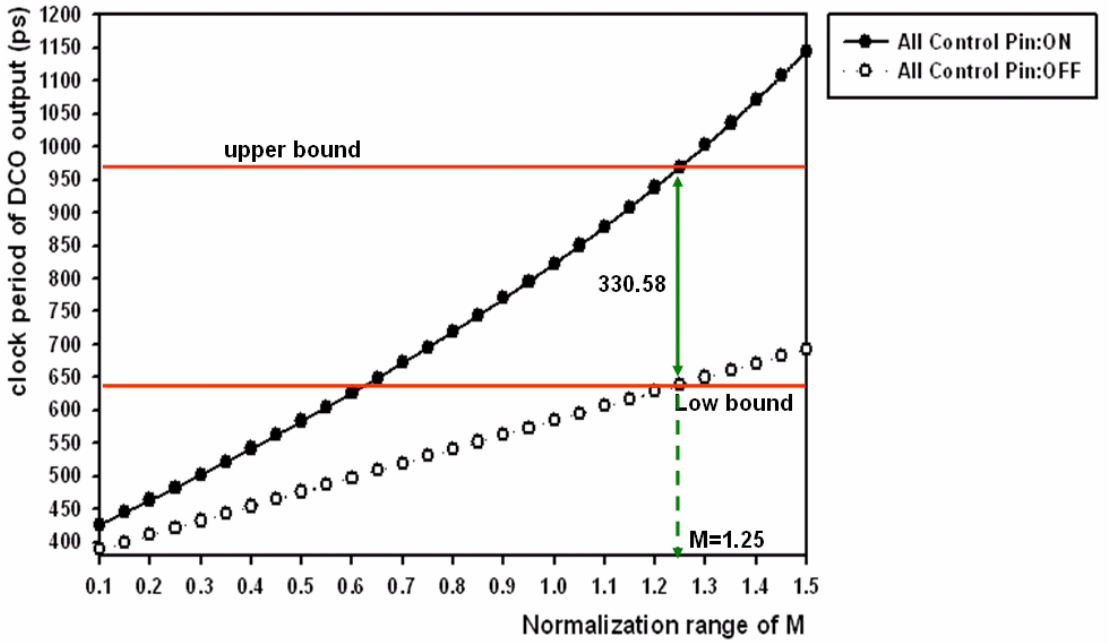


Figure 4.23 Find the ratio of M to satisfy our desired performance

We find the ratio relationship 1:M:N=1:1.25:0.9 through normalization process as mentioned above. Follow this relationship, we propose our DCO architecture as Figure 4.24 shows. We can separated coarse-tune stage into four binary weighted increment segment controlled though digital control word. Similarly, fine-tune stage can be separated four binary weighted increment segment controlled though digital control word. The LSB bit of every stage is controlled independently. In this way, we obtain 4 times resolution promotion. Therefore, we need a decoder circuit to obtain thermometer code control. We draw true table and the decoder circuit as Figure 4.25 shows. S3 is always turned off. S2 is obtained through the AND of a0 and a1. S1 connected to a1 directly. S0 is obtained through the OR of a0 and a1.

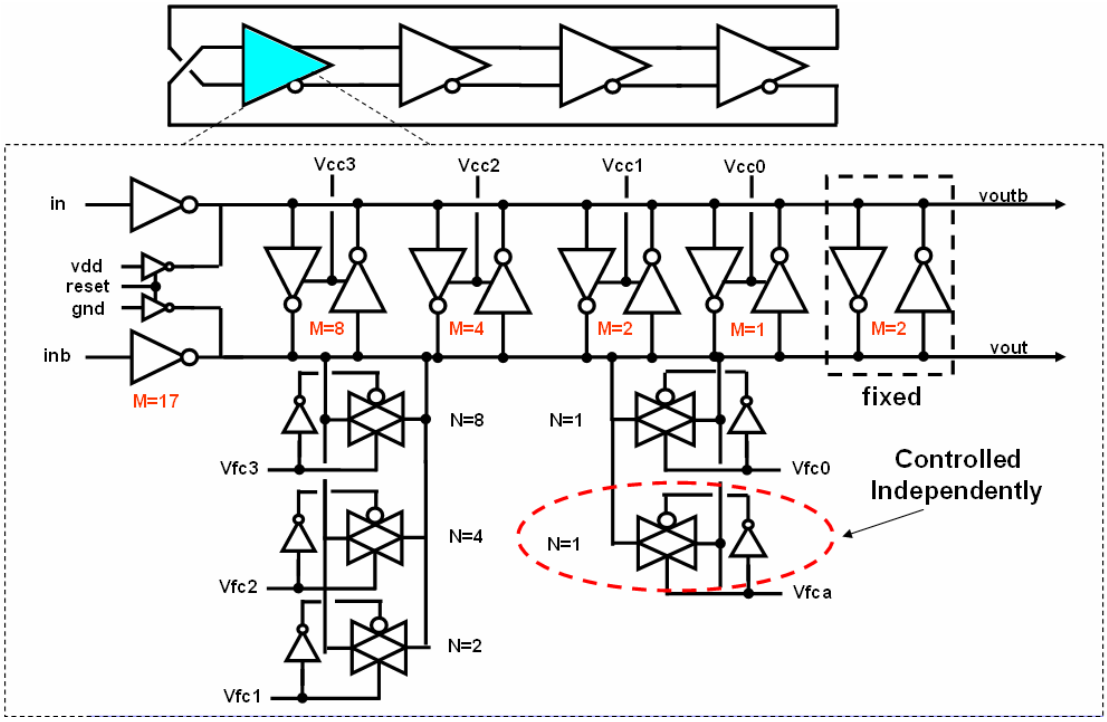


Figure 4.24 Our proposed DCO architecture

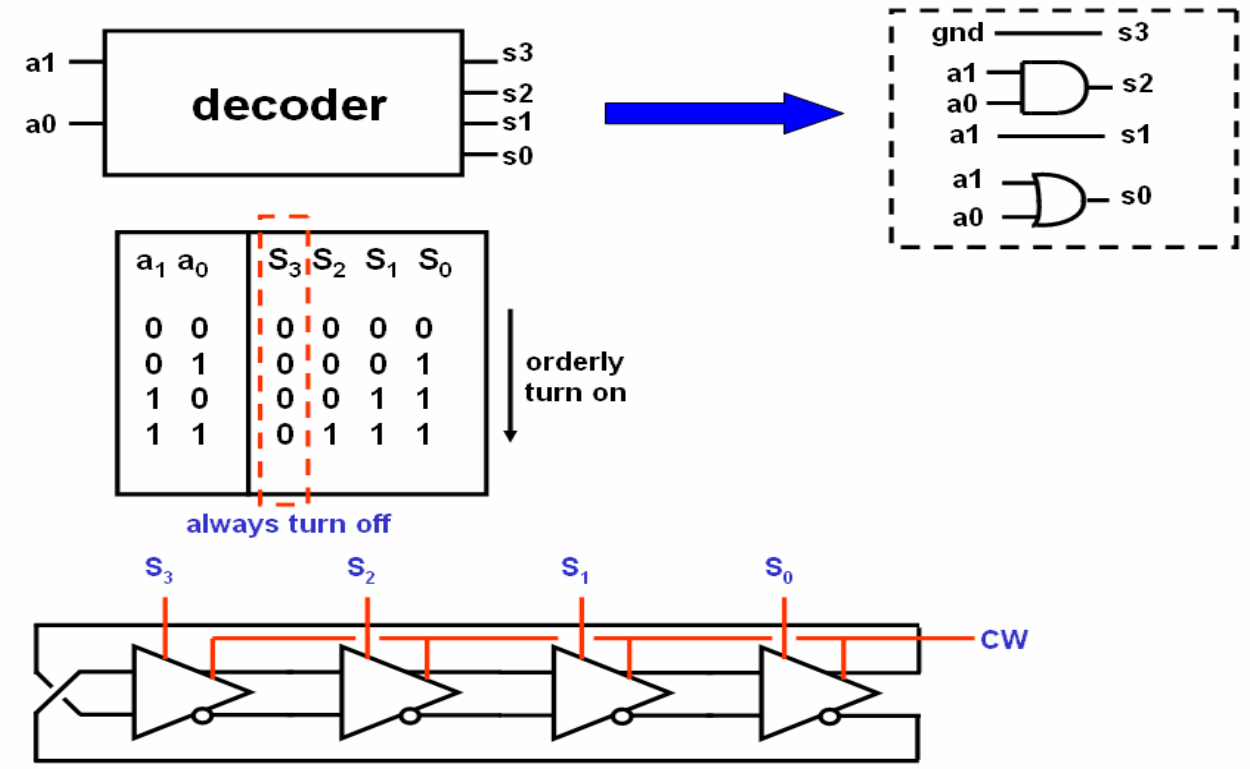


Figure 4.25 Decoder circuit

Figure 4.26 shows monotonic curves of tuning codes according to coarse and fine tuning of the DCO. The frequency range is 1.06GHz-1.56GHz at TT corner. And delay range of the fine delay chain is about 34.56ps. Therefore, the resolution is about 0.54ps.

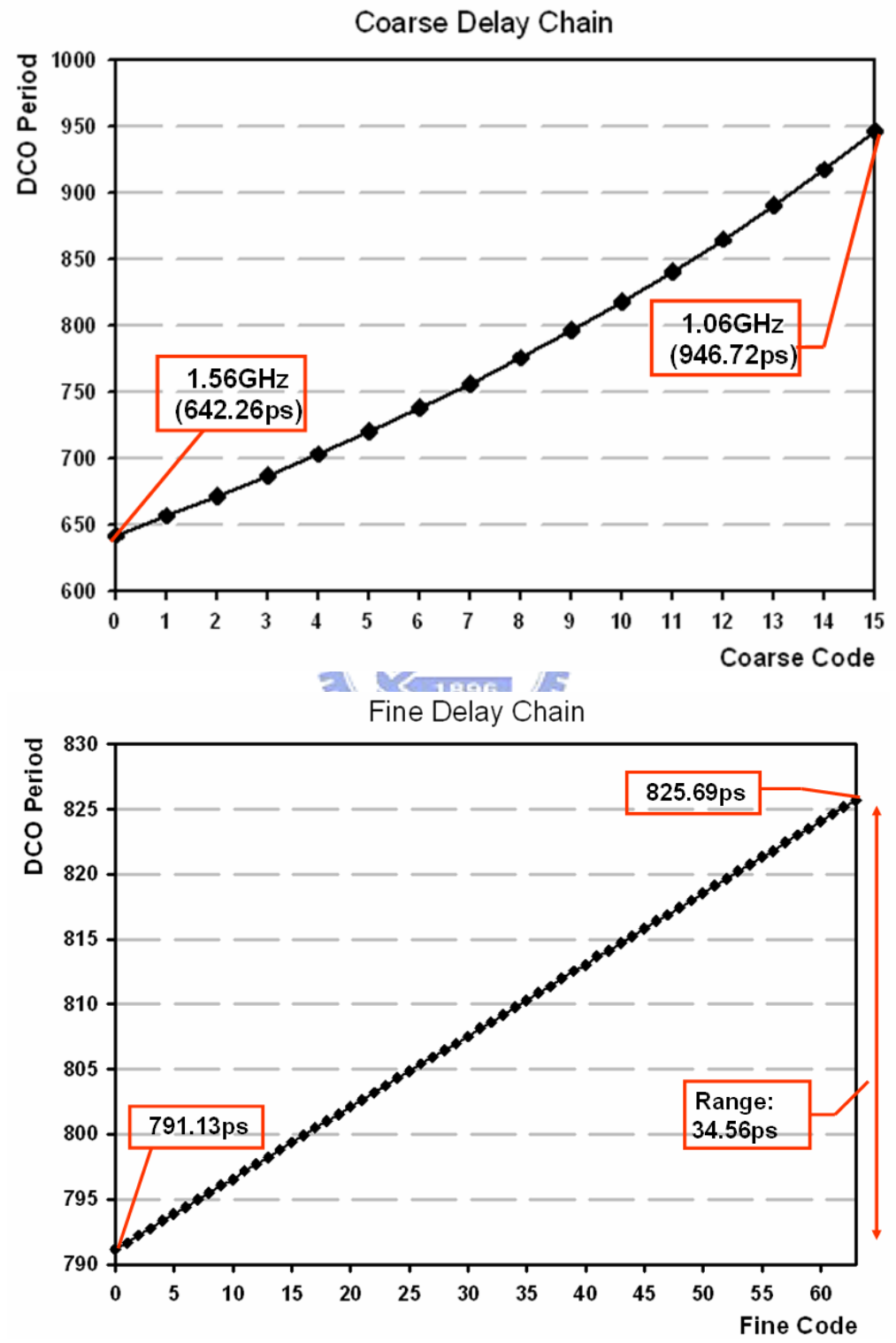


Figure 4.26 Frequency range of the proposed DCO

Figure 4.27 shows the delay resolution of the fine-tuning stage and the proposed delay cell has finer resolution (about 0.54ps) than JSSC'03 [12]. OAI cell [12] has less transistor counts and less power consumption, but it has non-uniform linearity.

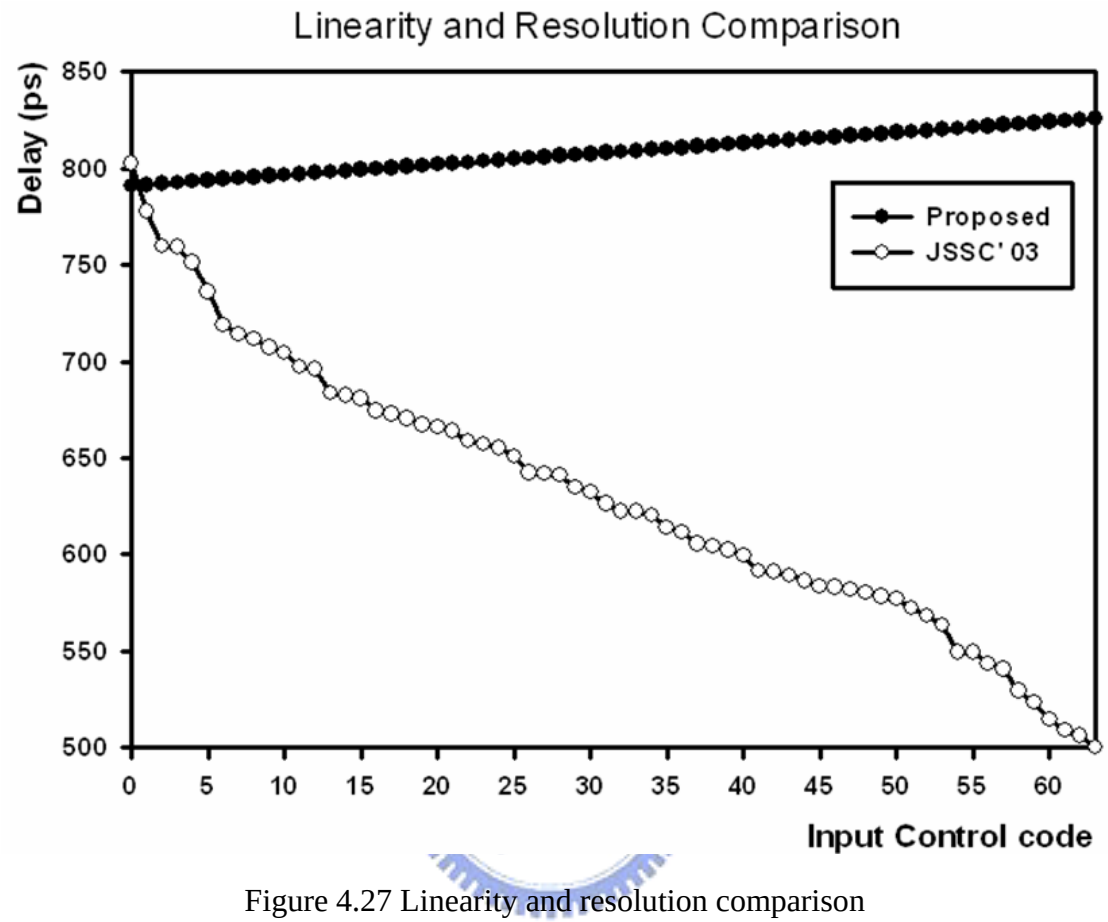


Figure 4.27 Linearity and resolution comparison

The operation range of the digitally controlled oscillator is shown in Figure 4.28 and 4.29. Figure 4.28 shows the clock period of DCO output at three different corner cases. Figure 4.29 shows the clock frequency of DCO output at three different corner cases. Table 4-1 summarize the features of the proposed DCO.

Table 4-1 The features of proposed DCO

Items	Coarse Delay	Fine Delay
Resolution	4bit	4bit+4bit (controlled independently)
Control Code Type	Binary	Binary + Thermometer
Max. DCO Gain	29.19ps/code	0.7ps/code
Avg. DCO Gain	20.29ps/code	0.55ps/code
Operation Range	1.03GHz~1.56GHz (TT Corner)	

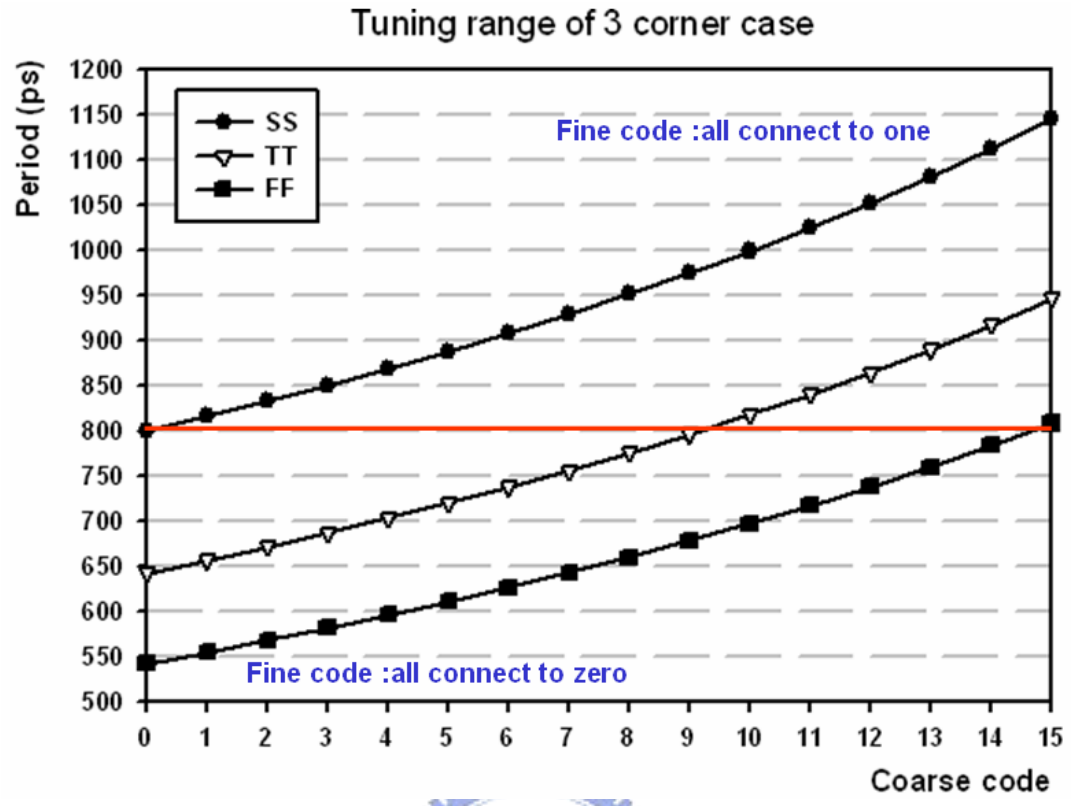


Figure 4.28 The clock period range of DCO at three different corner cases

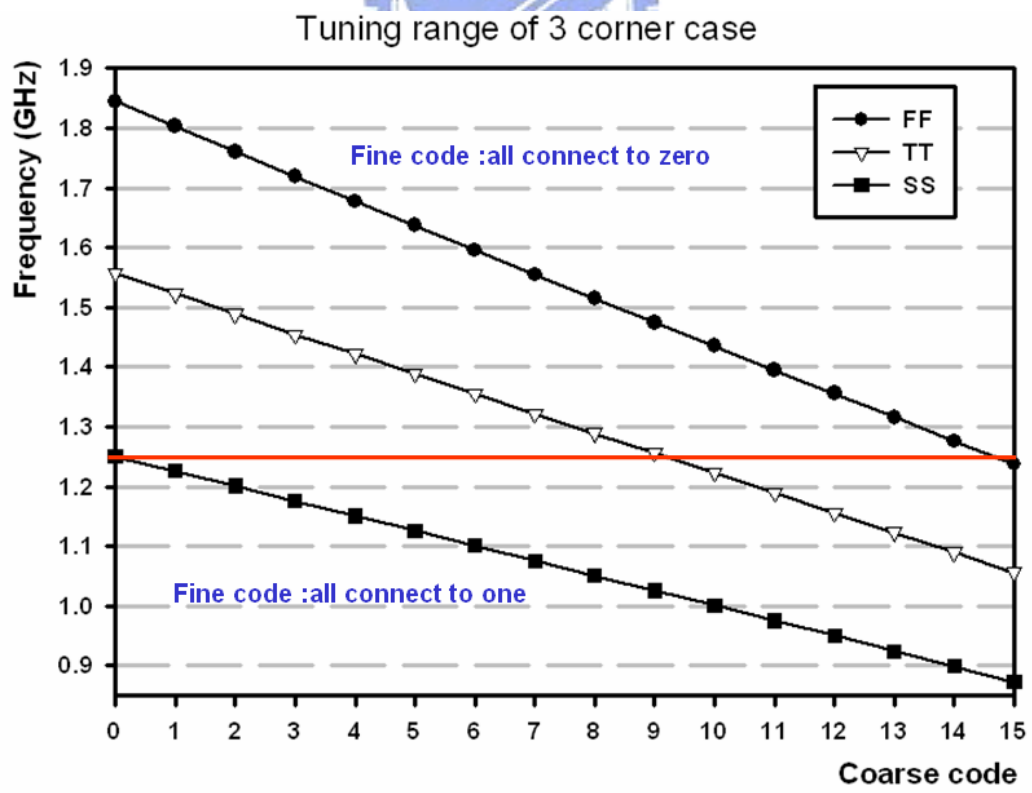


Figure 4.29 The clock frequency range of DCO at three different corner cases

Chapter 5

ADPLL System Design

In chapter4, we have designed and analyzed the proposed DCO. After designing the DCO, we must design a control mechanism to control the DCO to oscillate at the desired frequency. Thus, we must compare the frequency difference and detect the phase difference between the reference clock and the feedback clock. In this way, the control unit can produce accurate control signals to dictate the frequency of the DCO.

The control unit, frequency comparator and phase detector are all digital functional blocks. [16] shows some examples about the frequency phase detector and control unit . A specific control unit only controls a specific DCO and only receives specific signals produced by the FD or PD. Besides, search algorithm is also another important issue. We use improved binary search as our search algorithm which is the soul of our ADPLL system. In this chapter, we will introduce search algorithm, control unit circuit, and simulation results.

5.1 System Architecture

The system architecture is shown in Figure 5.1. The ADPLL consists of Counter-Based Frequency Comparator, Phase Detector, Gain Register, Adder / Subtractor, DCO Control Register, DCO, and feedback frequency divider.

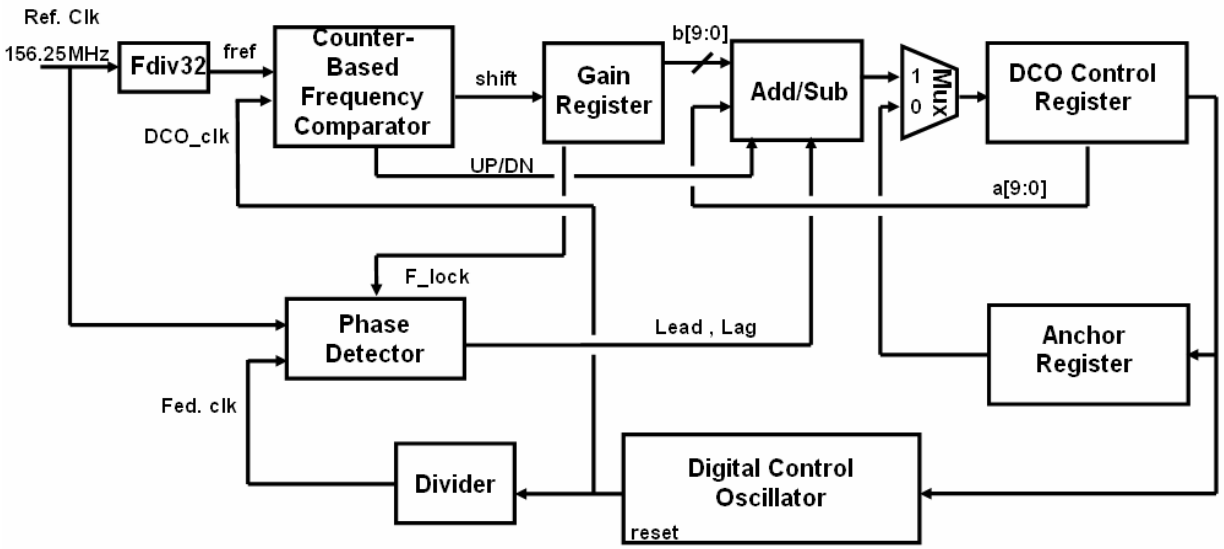


Figure 5.1 The ADPLL system architecture

5.2 Algorithm

- **Frequency acquisition:**

The frequency search algorithm of this thesis uses improved binary search algorithm. The binary search algorithm sets the initial control word with the middle value of digital control oscillator (DCO) input values. Thus, the initial-step size in the first search must limited between the maximum and minimum value of DCO's input. In conventional binary search algorithm (BSA), the control word always make the previous value divided by half and then pass the new value to DCO and do the next comparison. Unlike BSA, improved binary search algorithm (IBSA) would first check the values outputted from the previous and the current comparisons. If they are identical, IBSA sends the value to DCO without divided by half; otherwise it divides it by half. Figure 5.2 shows the BSA versus IBSA.

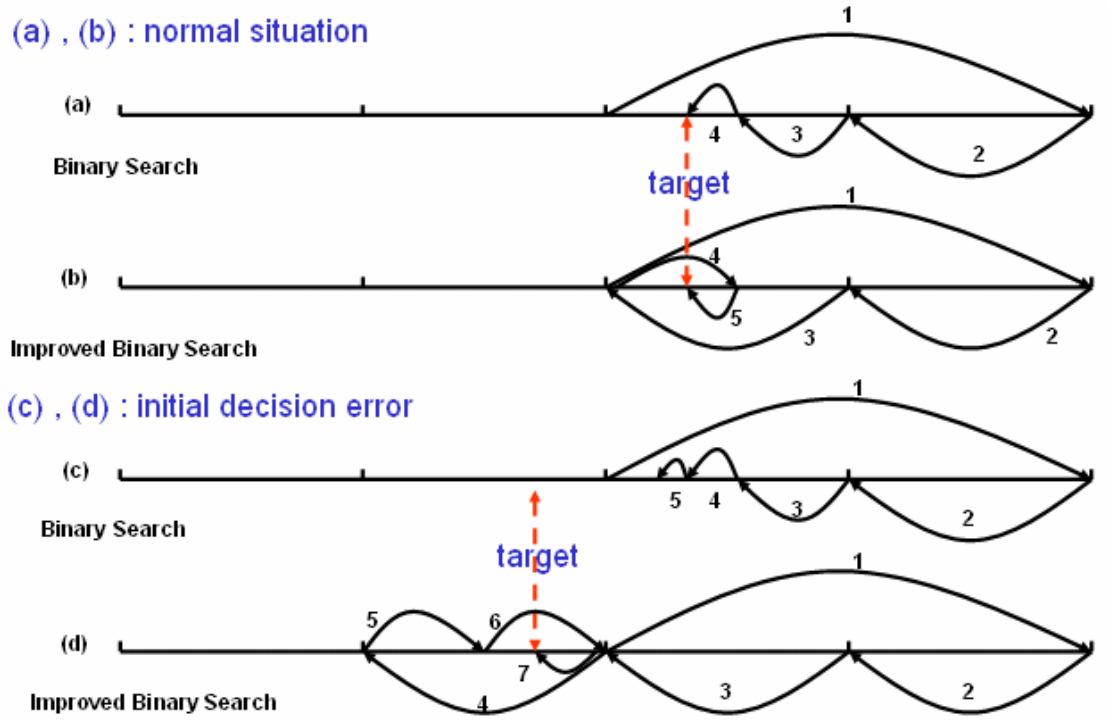


Figure 5.2 Binary search versus improved binary search

- **Phase acquisition:**

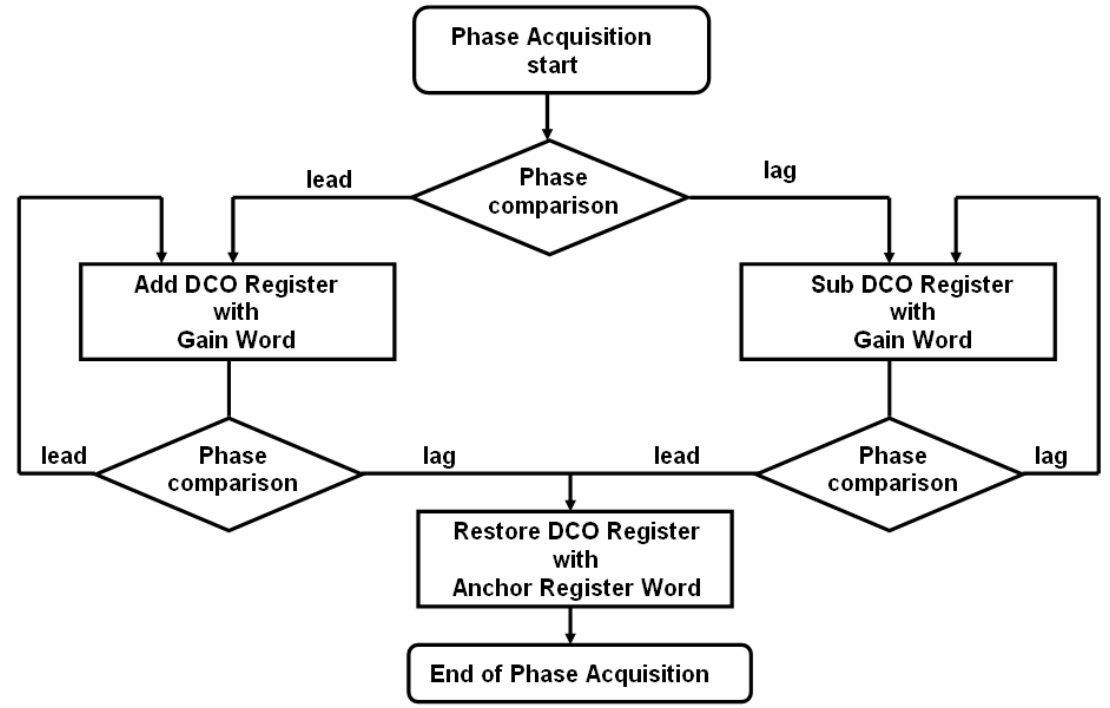


Figure 5.3 Phase acquisition algorithm

Figure 5.3 shows the phase acquisition algorithm. The goal of this mode is to align the DCO clock edge to the reference clock edge. Phase acquisition continues until the phase detector senses a change in the phase polarity of the reference clock relative to the DCO clock. At this point, phase acquisition is complete, and the ADPLL transfers the anchor register contents to the DCO control register, restoring the baseline frequency and completing phase lock.

- **Phase/frequency maintenance:**

Figure 5.4 shows the phase/frequency maintenance algorithm. In phase maintenance, the ADPLL increments or decrements the DCO control word every reference cycle. Whenever a change in phase polarity occurs, the ADPLL transfers the anchor register contents to the DCO control register, restoring the baseline frequency. However, reference clock frequency drift or DCO frequency drift induced by voltage or temperature variations requires that the ADPLL has the capability of changing the baseline frequency. Frequency maintenance mode provides such means by updating the anchor register. Via the frequency maintenance algorithm, the ADPLL increments or decrements the value in the anchor register whenever four consecutive cycles without a change in phase polarity occur. In this way, the ADPLL can re-lock new reference clock frequency if reference clock drift.

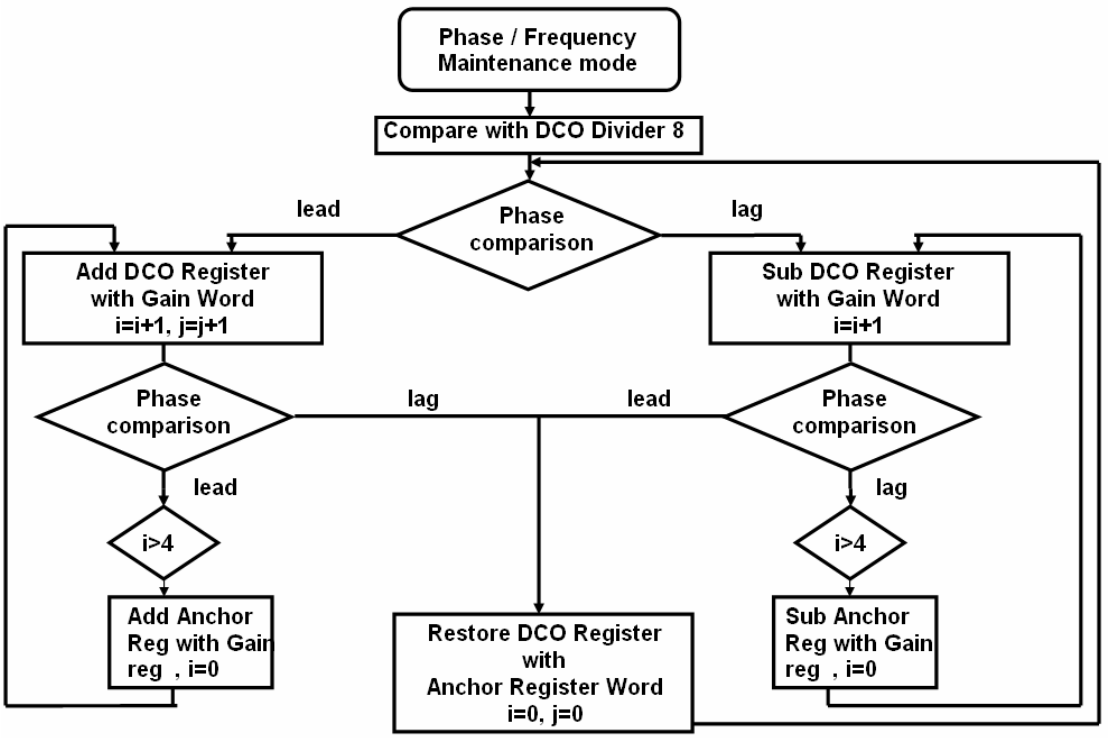


Figure 5.4 Phase/frequency maintenance algorithm

5.3 Control Circuit

All responses of ADPLL are almost decided by this control unit. The control unit usually consists of finite state machine. If the frequency of the feedback clock approaches the frequency of the reference clock, it will lock correct phase fast. Thus, the first step is to search target frequency. After completing frequency search, the second step is to track target phase. In point of the DCO, the frequency search works as coarse tune and the phase tracking works as fine tune.

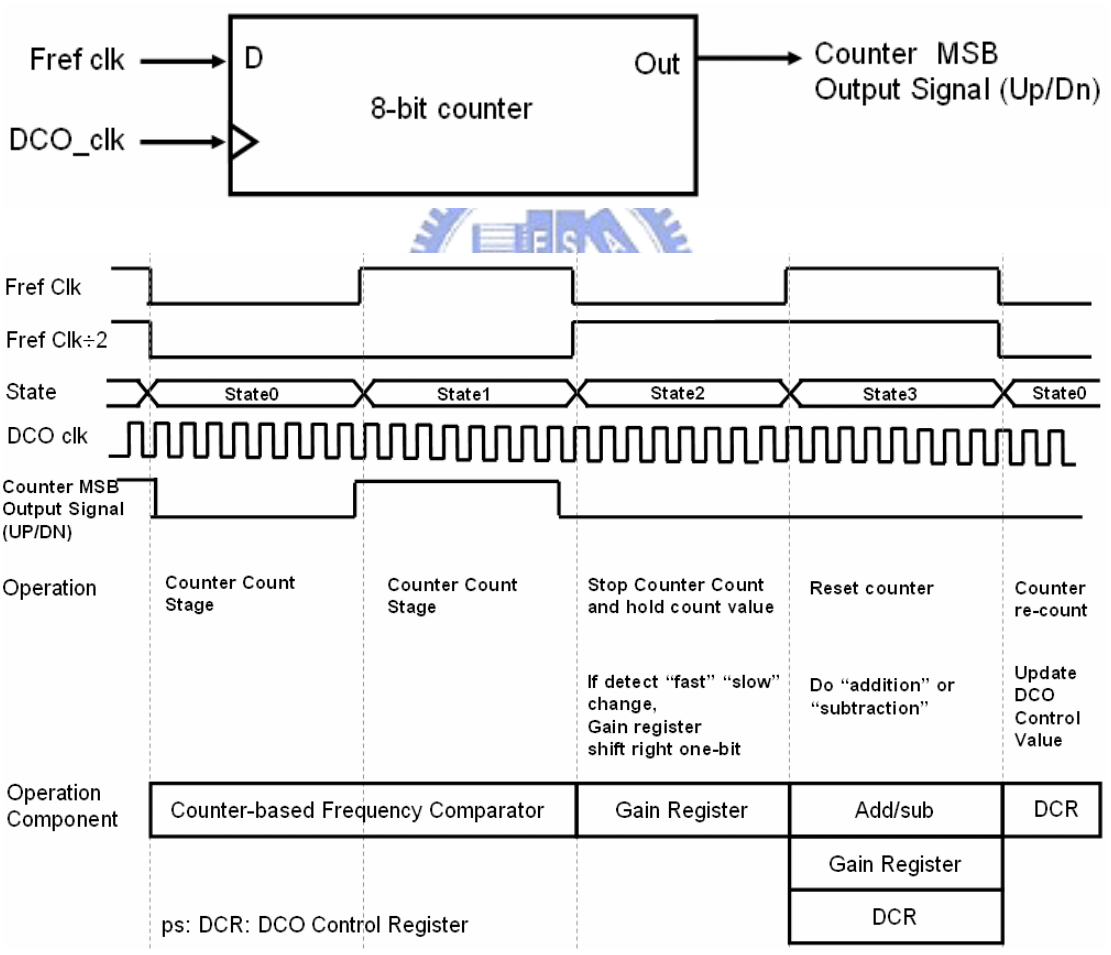


Figure 5.5 Counter-based frequency comparator timing state machine

Figure 5.5 shows the counter-based frequency comparator timing state machine. We can take the full comparison process into four states. State0 and state1 perform the counter count state. State2 stop counter count and hold the count value. State3 reset counter and then update the DCO control value.

Figure 5.6 shows the adder and subtractor architecture. This architecture adopted 10bits binary ripple carry adder and subtractor. This architecture can be seen in general logic design books. Figure 5.7 shows the gain register circuit. Gain register and DCO control register are all composed of register. Gain register is a shift register that can shift its stored bits when certain “shift” pulses occur. Figure 5.8 shows the DCO control register circuit. DCO control register is mainly used to store the control bits after adder / subtractor operation. Figure 5.9 shows the anchor circuit and function table. The function table shows the system is suited at frequency acquisition mode, phase acquisition mode or phase/frequency maintenance mode when input control signal is different.

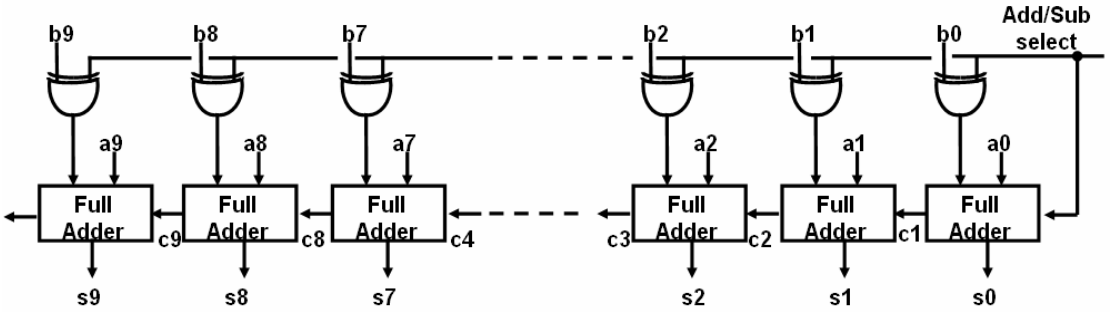


Figure 5.6 Binary ripple carry adder/subtractor circuit

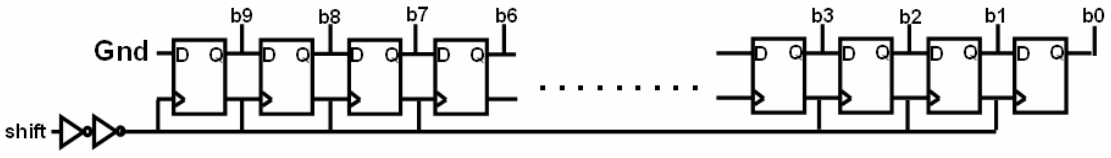


Figure 5.7 Gain register circuit

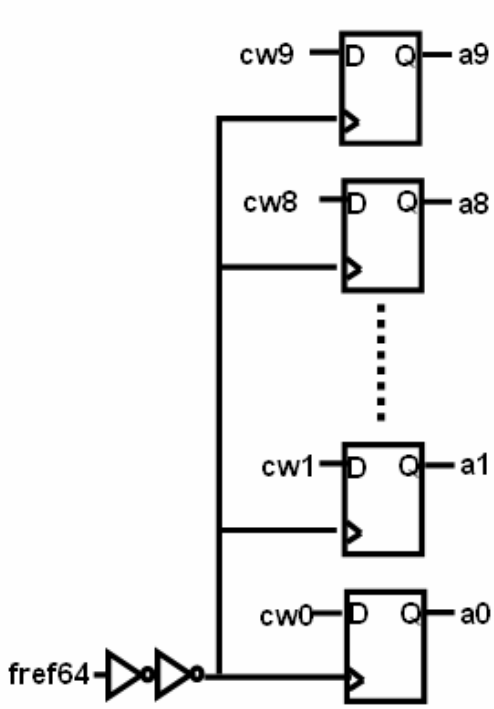
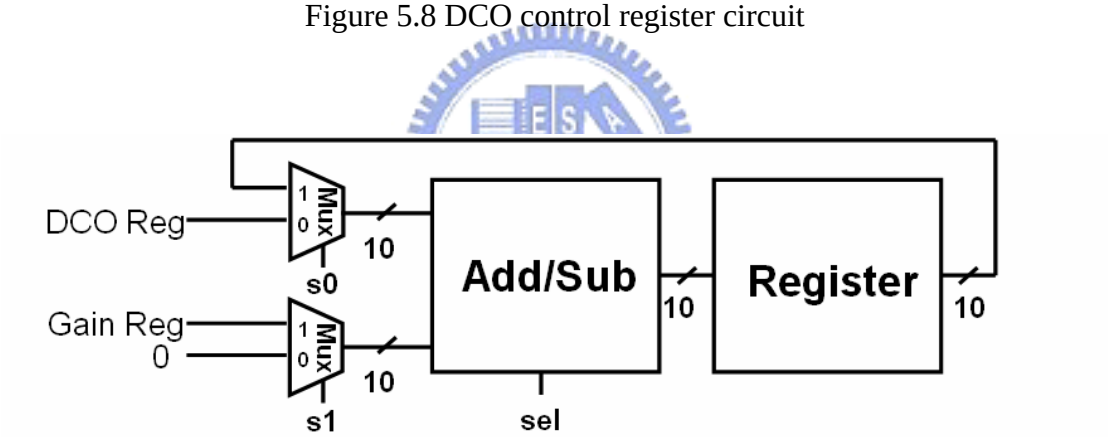


Figure 5.8 DCO control register circuit



s1	s0	sel	Reg_out
0	0	0	DCO_Reg
0	0	1	X
0	1	0	Anchor
0	1	1	Anchor
1	0	0	X
1	0	1	X
1	1	0	Anchor+Gain Reg
1	1	1	Anchor-Gain Reg

Figure 5.9 Anchor circuit and function table

5.4 System Behavior Simulation

Owning to the ADPLL system simulation with SPICE takes a lot of time, we use MATLAB SIMULINK to analyze the loop behavior and verify the lock function. Functional verification of the ADPLL is performed effectively at the first design step, which improves turn around time of ADPLL design. Figure 5.10 shows the behavior modeling of ADPLL system. Figure 5.11 shows the functional verification result of the ADPLL. The ADPLL operates like the flowchart of figure 5.2, 5.3, and 5.4. Figure 5.12 shows the output clock jitter simulation result. The jitter performance is about 45ps.

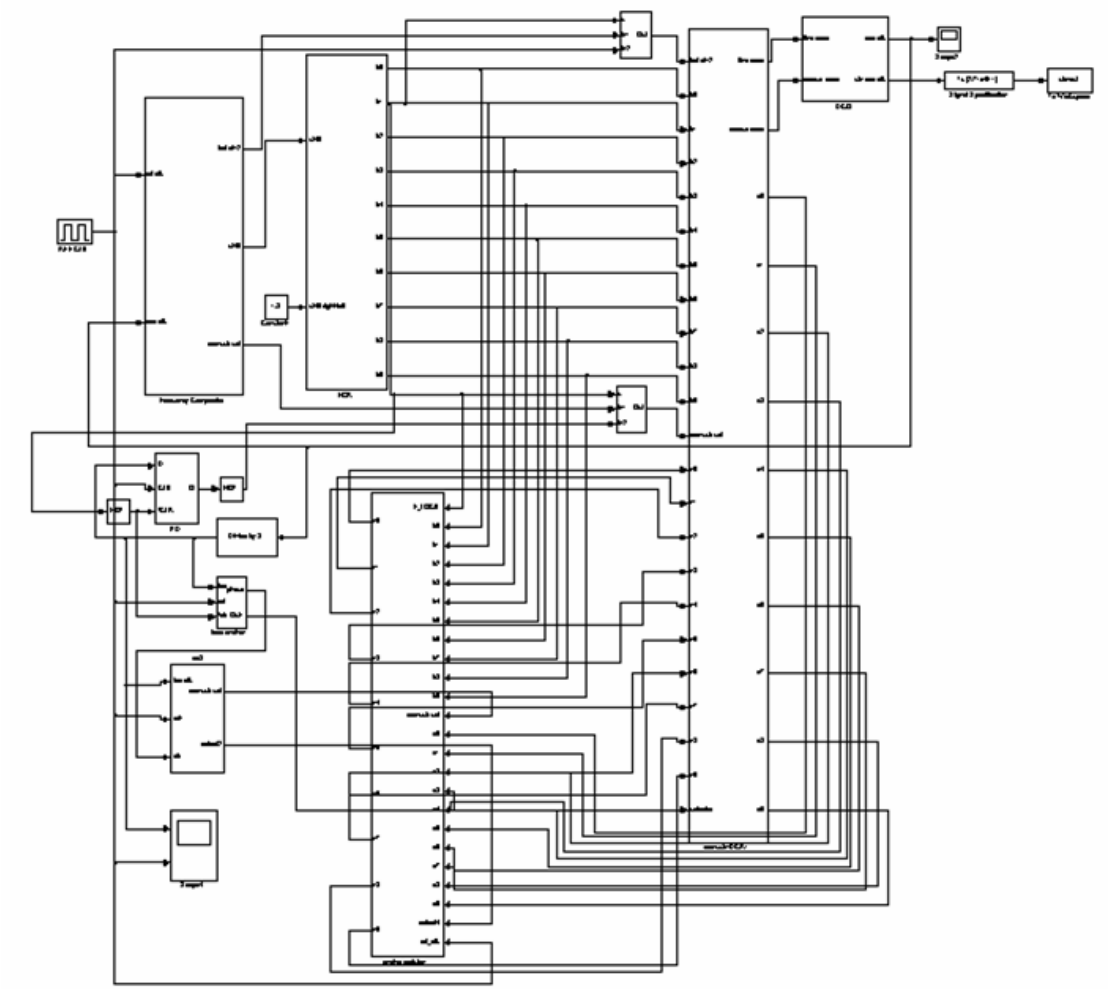


Figure 5.10 Behavior Simulation of the proposed ADPLL by SIMULINK

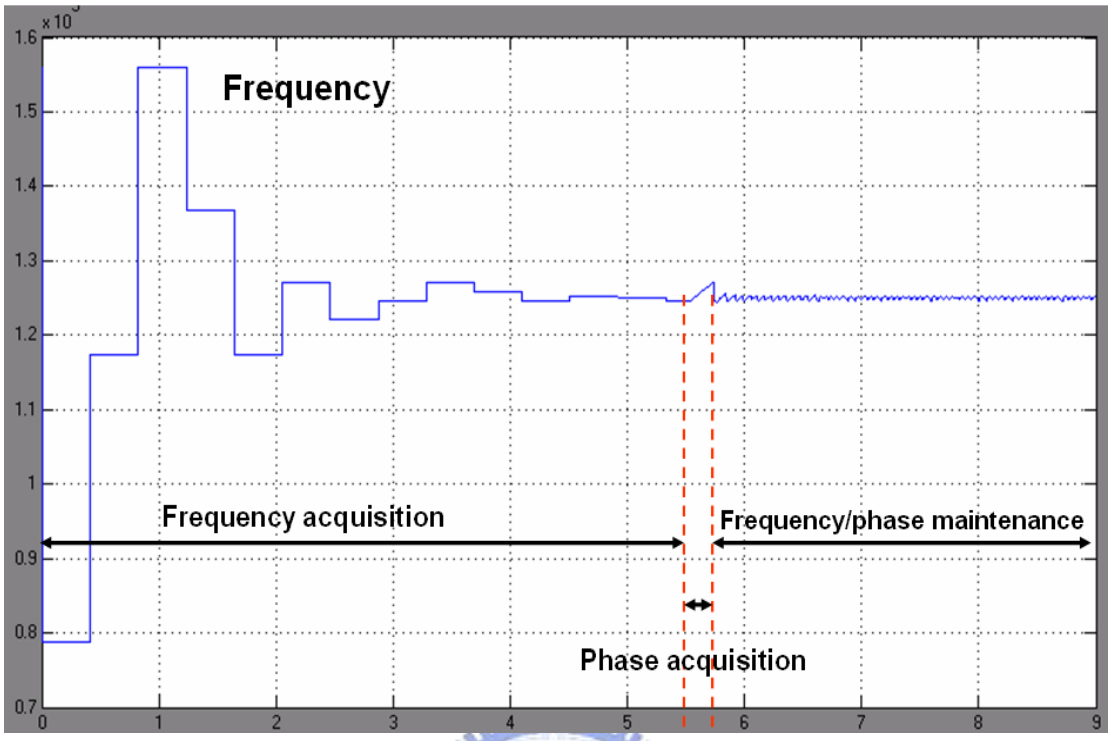


Figure 5.11 Function verification of the proposed ADPLL by SIMULINK

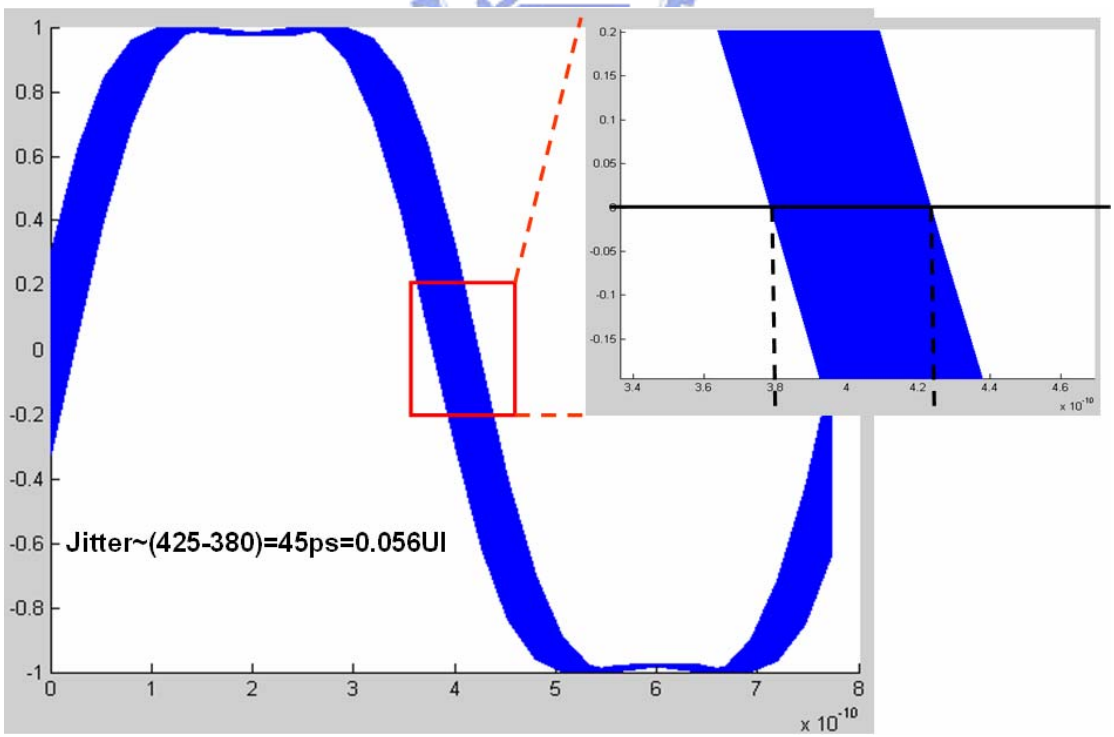


Figure 5.12 Output clock jitter simulation using SIMULINK

Figure 5.13 shows the eye diagrams with different resolution and its corresponding jitter. From simulation results and comparison, we take 0.5ps as our DCO resolution which makes system output clock jitter acceptable.

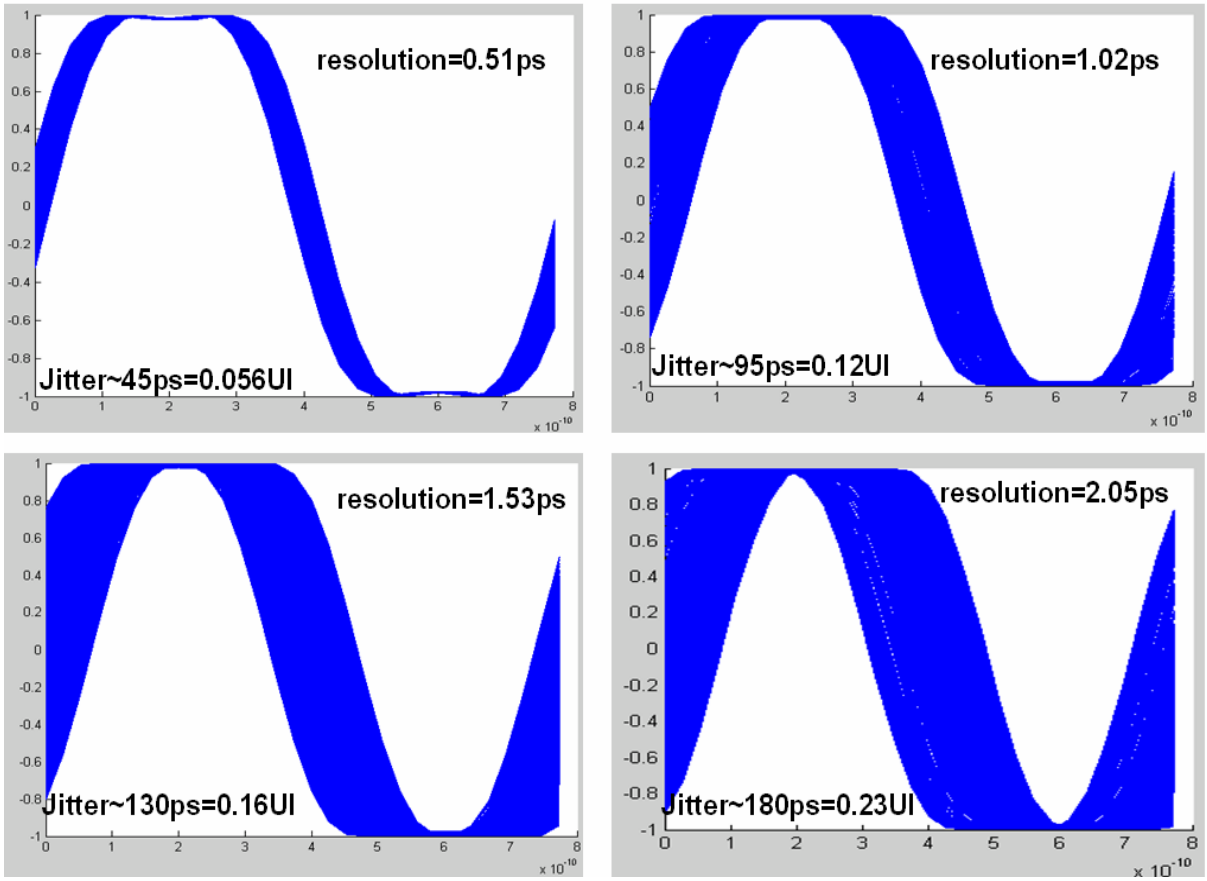


Figure 5.13 Eye diagram jitter with different resolution

Generally, reference-clock frequency drift induced by voltage or temperature variations requires that the ADPLL has the capability of changing the baseline frequency to re-lock. Our phase/frequency maintenance mode provides such means by updating the anchor register. Figure 5.14 shows the system re-lock simulation when reference clock drift to 158.75MHz. Figure 5.15 shows the system re-lock simulation when reference clock drift to 153.75MHz. From simulation results, our system can still provide stable output frequency signal even if input signal frequency drift occurs.

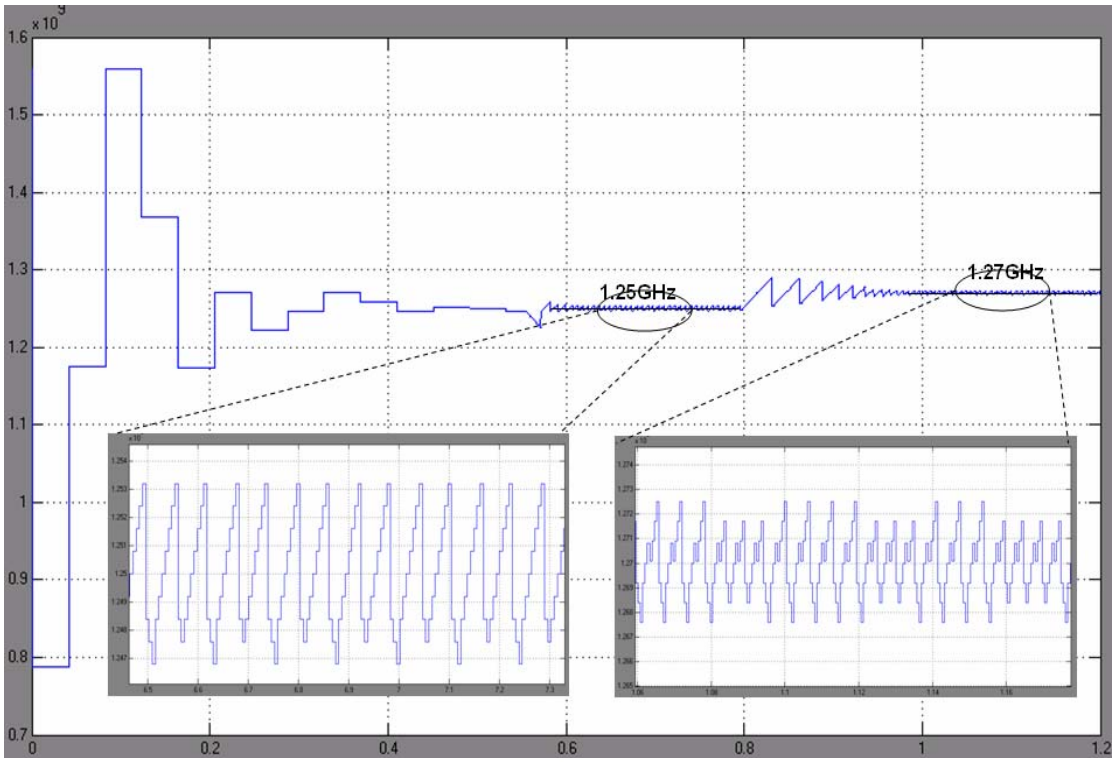


Figure 5.14 System re-lock simulation when reference clock drift to 158.75MHz

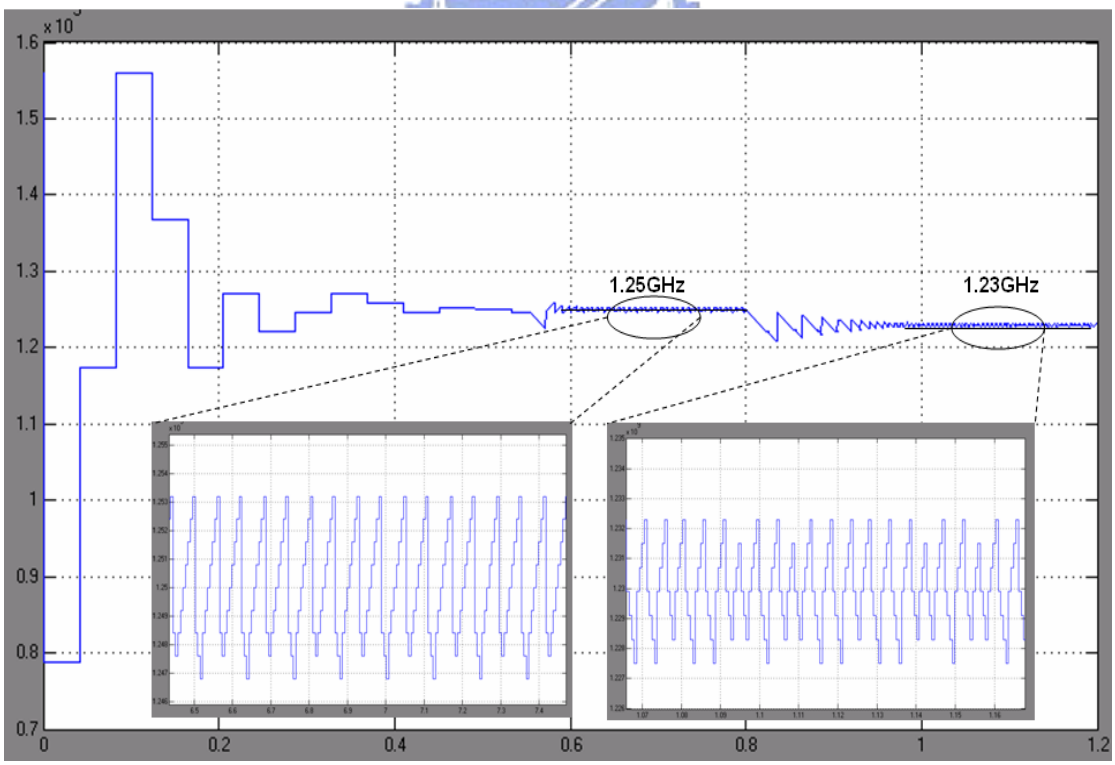


Figure 5.15 System re-lock simulation when reference clock drift to 153.75MHz

5.5 System Circuit Level Simulation

Simulations of a behavior-level of the ADPLL completing phase lock require about 10 minutes. By contrast, the same simulation in SPICE requires in excess one day on a PC but the results are more precise. Figure 5.16 shows the eye diagram jitter with different count number using SPICE simulator. From simulation results, the jitter performance is better than others when count number is set to four. Thus, the ADPLL increments or decrements the values in the anchor register when four consecutive cycles without a change in phase polarity via frequency maintenance algorithm.

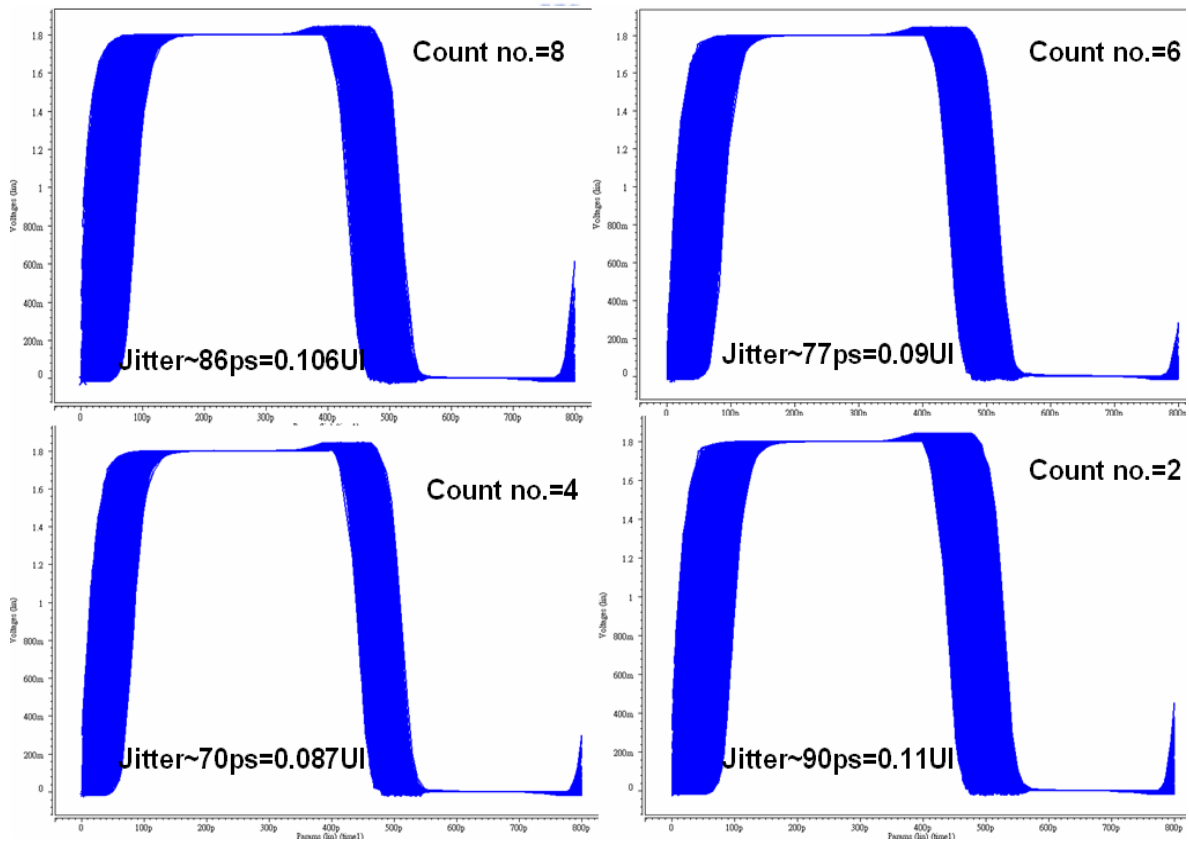


Figure 5.16 Eye diagram jitter with different count number

Figure 5.17 shows the system at lock state situation. “lead/lag” signal point out the system lead or lag information. When four consecutive cycles without a change in phase polarity, “detect4” signal will send a pulse. At the same time, the ADPLL increments or decrements the value in the anchor register. When a change in phase polarity occurs, the ADPLL “load” the anchor register contents to the DCO control register, restoring the baseline frequency as figure 5.17 shown. Figure 5.18 shows the eye diagram jitter with ground bounce (pre-sim). The output jitter is about 77ps.

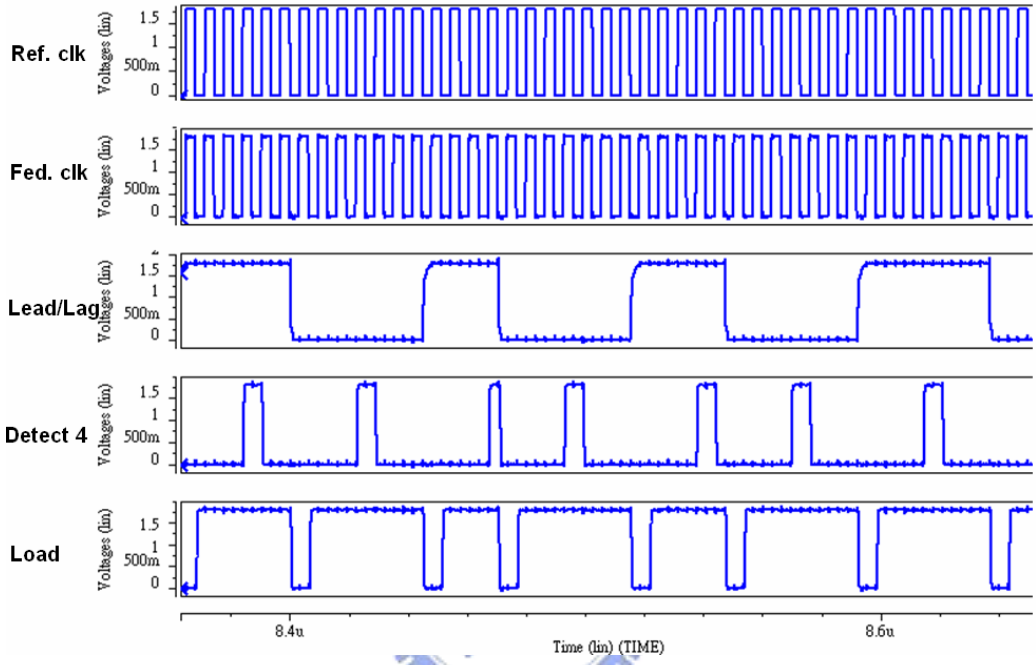


Figure 5.17 System at lock state

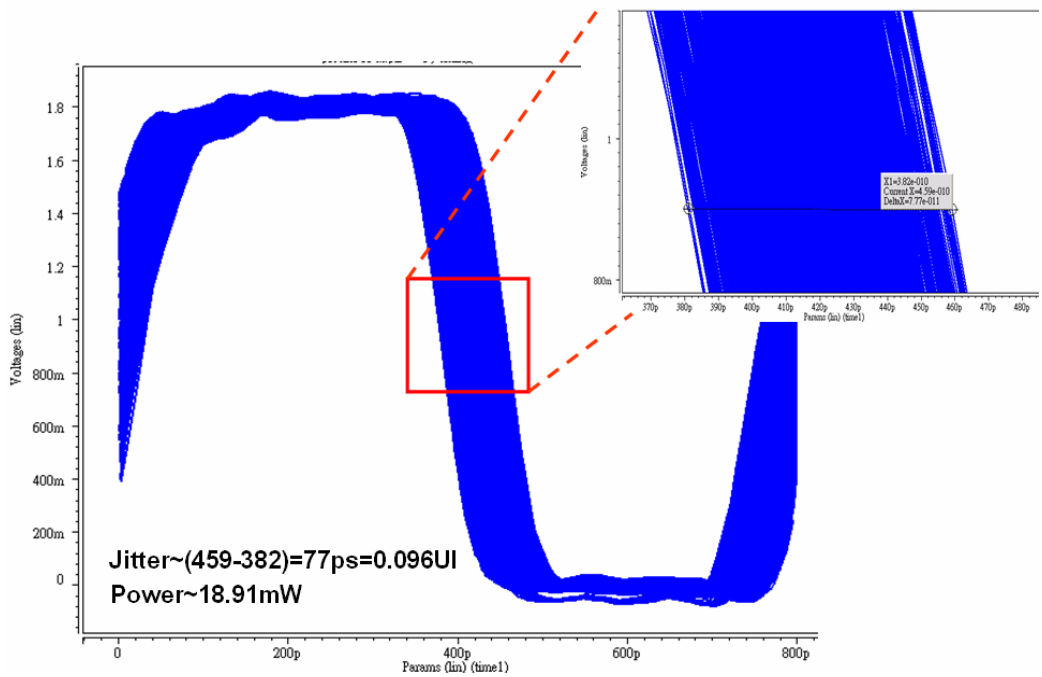


Figure 5.18 Eye diagram jitter with ground bounce (pre-sim)

Figure 5.19 shows the eye diagram jitter with ground bounce (post-sim). Figure 5.20 shows the eight-phases output eye diagram through output buffer.

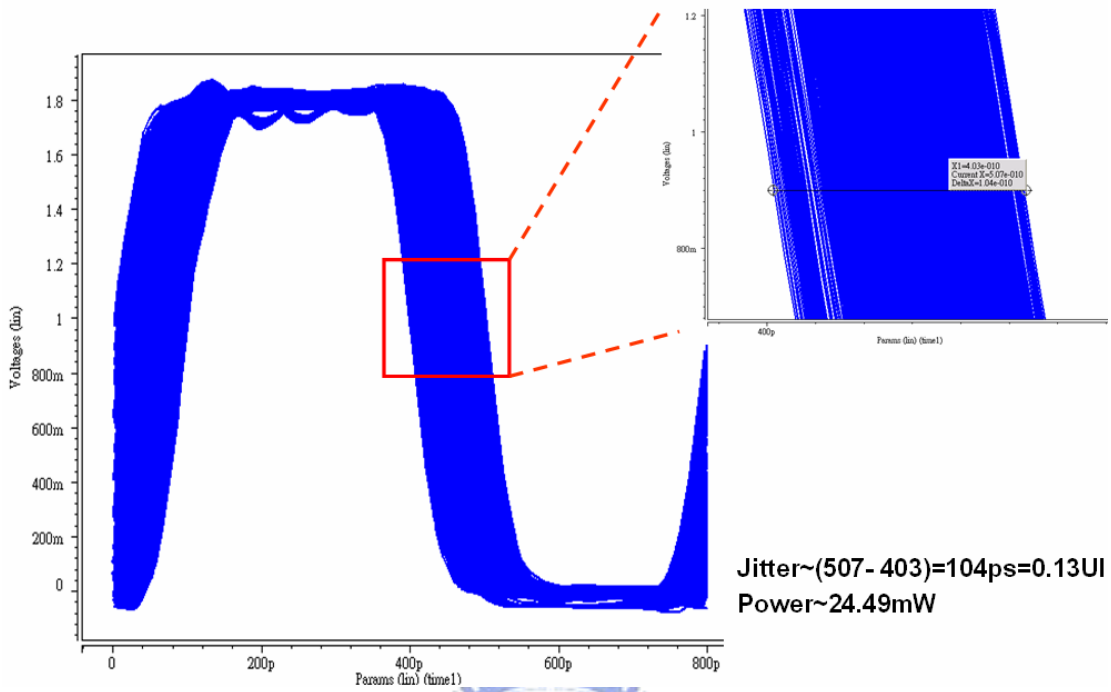


Figure 5.19 Eye diagram jitter with ground bounce (post-sim)

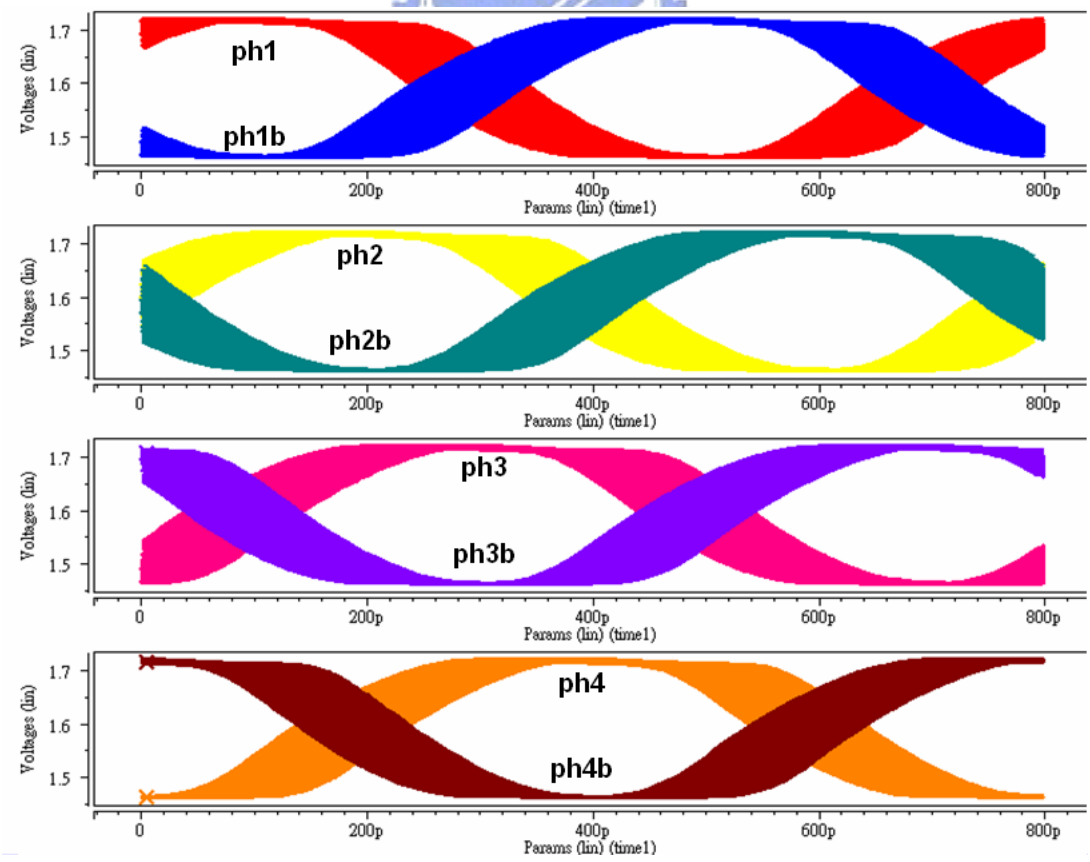


Figure 5.20 Eight-phases output eye diagram

5.6 Layout

The proposed ADPLL will send to National Chip Implement Center (CIC) with T18-95A. The chip area is 0.88mm*0.73mm as shown in Figure 5.21. The post-layout simulation result summary is shown in Table 5-1.

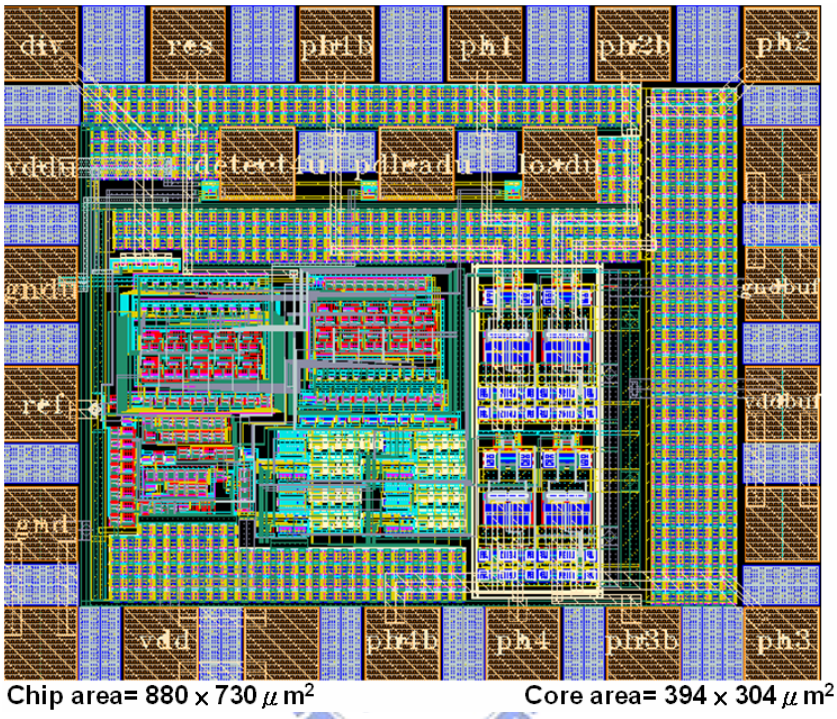


Figure 5.21 Layout of the proposed ADPLL

Table 5-1 Summary of the ADPLL chip

Function	ADPLL
Technology	0.18 μm 1P6M CMOS
Supply Voltage	1.8V
Chip Area	880 x 730 μm²
Core Area	394 x 304 μm²
Transistor/Gate Count	4584/2240
Target Frequency	1.25GHz
Output phases	8
Frequency Range	1.03GHz~1.56GHz
DCO Resolution	~0.5ps
Jitter (pk-pk)	~104ps
Power Consumption	24.49mW@1.25GHz

5.7 Comparisons and Summary

Table 5-2 shows the comparisons among different ADPLLs. The proposed ADPLL has multi-phase output than others. The proposed ADPLL has smaller area than [12],[17],[18] and has lower power consumption than [12],[14],[17]. Besides, we have smaller resolution than others. Jitter performance is also better than [14],[18].

In this Chapter, an ADPLL is presented. The ADPLL is implemented in 0.18um 1P6M TSMC CMOS process and can be operate from 1.03GHz to 1.56GHz. The peak to peak jitter of the output clock is about 104ps and the power consumption is 24.49mW. The proposed ADPLL has multi-phase output, fine resolution, lower power and small silicon area suitable for high-speed serial link applications.

Table 5-2 ADPLL performance comparison

Type	Proposed	ISCAS 05[18]	IEICE 05[17]	JSSC 04[14]	JSSC 03[12]
Process	0.18 μm CMOS	0.18 μm CMOS	0.18 μm CMOS	0.35 μm CMOS	0.35 μm CMOS
Supply Voltage	1.8V	1.8V	1.8V	3V	3.3V
Core Area	394*304 μm^2	0.1mm^2	600*450 μm^2	260*260 μm^2	840*840 μm^2
Frequency Range	1.03GHz~ 1.56GHz	140MHz~ 1.03GHz	500MHz~ 1.5GHz	152MHz~ 366MHz	45MHz~ 510MHz
Power	24.49mW (@1.25GHz)	N/A	27mW (@670MHz)	24mW (@366MHz)	100mW (@500MHz)
Power/MHz	19.59uW	N/A	40.29uW	65.57uW	200uW
LSB Resolution	0.5ps	22ps	1.2ps	10ps	5ps
Output Jitter	104ps (0.13UI)	143ps (0.137UI)	70ps (0.05UI)	775ps (0.248UI)	70ps (0.03UI)

Chapter 6

Conclusions

This thesis describes the data serializer in the transmitter and the all digital phase-locked loop in 0.18um 1P6M CMOS process. The role of both building blocks is located at high-speed serial link front-end. To integrate overall system easily, these two building blocks designed in an all-digital way.

In transmitter design, data serializer is composed of a 4-stage PLL, a shift-register type parallel-in-serial-out (PISO) multiplexer, and a LVDS differential output driver. The circuit is capable of 125Mbps data serialization, which results in 1.25Gbps data transmitting. The corresponding transmitter has been implemented in TSMC digital 0.18um 1P6M CMOS technology. The measurement results show that peak-to-peak jitter of PLL is 45.56ps. The jitter of transmitter differential output eye-diagram is about 66ps and the eye-amplitude is about 500mV. Its measurement performance meets our design goal for IEEE802.3ah physical layer specification.

In ADPLL design, we propose a new fine-tuning method to change the DCO oscillating frequency slightly. The DCO resolution can be improved to about 0.5ps by adding fine-tuning delay stage. The frequency ranges are about 1.03GHz-1.56GHz by HSPICE circuit simulation. We also take MATLAB simulink to perform system function verification. According to post-layout simulation, we know that the ADPLL search algorithms are working correctly. After post-layout simulation, the output clock jitter is about 104ps at 1.25GHz and power consumption is about 24.49mW. Layout area of ADPLL core is about $394 \times 304 \mu m^2$ .The ADPLL chip will be implemented using 0.18um 1P6M TSMC CMOS technology. In conclusion, the proposed ADPLL has multi-phase output, fine resolution, lower power and small silicon area suitable for high-speed serial link applications.

Bibliography

[1] I. Young et al., "A PLL clock generator with 5 to 110 MHz lock range for microprocessors," in *ISSCC Dig. Tech Papers*, Feb. 1992, pp. 50-51.

[2] R. Stefo, J. Schreiter, "High Resolution ADPLL Frequency Synthesizer for FPGA-and ASIC-based Applications", Field-Programmable Technology (FPT), 2003. Proceedings. 2003 IEEE International Conference on 15-17 Dec. 2003 Page(s):28 – 34.

[3] Chi-Cheng Cheng, "The Analysis and Design of All Digital Phase-Locked Loop （ADPLL）," National Chiao-Tung University, Master's thesis, 2001.

[4] B. Razavi, "*Design of Integrity Circuit for Optical Communications,"* McGraw-Hill Companies, Inc., 2002.

[5] Fuji Yang, Jay O'Neill, Patrik Larsson, Dave Inglis, and Joe Othmer, "A 1.5V 86Mw/ch 8-Channel 622-3125 Mb/s/ch CMOS SerDes Macrocell with Selectable MUX/DEMUX Ratio," *IEEE ISSCC*, 2002.

[6] Kyeongho Lee, Sungjoon Kim, Gijung Ahn, Deog-Kyoon Jeong, " A CMOS Serial Link for Fully Duplexed Data Communication," *IEEE JSSC*, vol.30, no. 4,pp.353-364, Apr. 1995.

[7] Kyeongho Lee, Yeshik Shin, Sungjoon Kim, Deog-Kyoon Jeong, Gyudong Kim, Bruce Kim, and Victor Da Costa, "1.04GBd Low EMI Digital Video Interface System Using Small Swing Serial Link Technique," *IEEE JSSC*, vol.33, pp.816-823, May. 1998.

[8] Hsin-Wen Wang, Hung-Wen Lu, Chau-Chin Su," A digitized LVDS driver with simultaneous switching noise rejection**,"** Advanced System Integrated Circuits 2004. Proceedings of 2004 IEEE Asia-Pacific Conference on 4-5 Aug. 2004 Page(s):240 – 243.

[9] J. Dunning, G. Garcia, J. Lundberg, and E. Nuckolls, "An all-digital phase-locked loop with 50-cycle lock time suitable for high-performance microprocessors", IEEE J. Solid-State Circuits, vol.30, no.4, pp. 412-422, Apr. 1995.

[10] Jen-Shiun Chiang and Kuang-Yuan Chen, "The design of an all-digital phase-locked loop with small DCO hardware and fast phase lock," IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, vol. 46,no.7, pp.945-950, July 1999.

[11] Chi-Cheng Cheng, "The Analysis and Design of All-Digital Phase-Locked Loop(ADPLL)," M.S. Dissertation, Department of Electronics Engineering, National Chiao Tung University, Taiwan, July 2001.

[12] Ching-Che Chung and Chen-Yi Lee, "An All-Digital Phase-Locked Loop for High-Speed Clock Generation", *IEEE J. Solid-State Circuit*, vol.38, pp.347-351, Feb 2003.

[13] T.Y. Hsu , C.-C. Wang, and C.-Y. Lee "Design and analysis of a portable high-speed clock generator," *IEEE Trans. Circuits Syst. II*, vol. 48, pp. 367-375, Apr. 2001.

[14] Thomas Olsson, "A digitally Controlled PLL for SoC Applications", *IEEE J. Solid-State Circuits*, vol.39, no.5, pp. 751-759, May. 2004.

[15] Liang Dai, "A low phase noise ring oscillator with differential control and quadrature outputs" ASIC/SOC Conference, 2001. Proceedings. 14th Annual IEEE International , 12-15 Sept. 2001 .Pages:134 – 138.

[16] R. E. Best, Phase-Locked Loops: Design, Simulation, and Application,5th ed., McGraw-Hill, New York, 2003.

[17] Kwang-Jin LEE; "A Low Jitter ADPLL for Mobile Applications" *IEICE Transactions Electron.* , vol.E88-c, no.6 June 2005.

[18] Chia-Tsun Wu;" A Scalable DCO Design for Portable ADPLL Designs" *IEEE International Symposium on Circuits and Systems*, Page(s):5449 – 5452, May 2005.