國立交通大學

電信工程研究所

碩士論文

擁有寬捕獲範圍之全數位鎖相迴路電路設計

A Wide Capture Range PLL based on All-Digital Design

研 究 生：謝進益

指導教授：高銘盛 教授

中 華 民 國 一百 年 七 月

擁有寬捕獲範圍之全數位鎖相迴路電路設計

A Wide Capture Range PLL based on All-Digital Design

研究生：謝進益　　Student：Chin-Yi Hsieh

指導教授：高銘盛　　Advisor：Ming-Seng Kao

國立交通大學
電信工程研究所
碩士論文

A Thesis
Submitted to Institute of Communication Engineering
College of Electrical and Computer Engineering
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Master
in

Communication Engineering

July 2011

Hsinchu, Taiwan, Republic of China

中華民國一百年七月

擁有寬捕獲範圍之全數位鎖相迴路
電路設計

學生:謝進益　　　　　　　　　　指導教授:高銘盛 教授

國立交通大學

電信工程研究所

摘要

本論文設計一個擁有極寬捕捉頻帶的全數位鎖相迴路，其鎖定頻率範圍從 1kHz 到 1MHz。此鎖相迴路的最高頻率與最低頻率的倍率為 1000，並且使用者不需知道輸入頻率的範圍，只要工作頻率位於此區間的應用，皆可使用此電路。為了能夠適用於充滿雜訊的環境，所設計的電路擁有良好的抗雜訊能力，即使在 SNR=0dB 的惡劣環境中，它依然可以正常的運作。

在設計時，我們著重於三個效能參數：頻率鎖定的正確率、鎖定效率與抗頻率漂移能力。為了能在各個面向都有良好的表現，我們設計三種不同的工作模式及對應的演算法，其中包含捕獲模式、追蹤模式與相位修正模式，並且在實際電路中選擇使用雙迴路的方式，以完成一個全數位雙迴路鎖相電路。最後，我們利用硬體描述語言實現此鎖相迴路，以驗證其可行性。

A Wide Capture Range PLL based on All-Digital Design

Student: Chin-Yi Hsieh Advisor: Prof. Ming-Seng Kao

Institute of Communication Engineering
National Chaio Tung University

Abstract

In this thesis we design an all-digital phase-locked loop (PLL) with ultra-wide capture range which spans from 1kHz to 1MHz. The ratio of the highest frequency to the lowest frequency is 1000 and there is no need for prior information of the input frequency. With this PLL, all applications whose working frequency is within this frequency range could use it to implement the corresponding system. Also, for applying it in noisy environment, the PLL is asked to have good noise immunity. It is designed to work efficiently when SNR=0dB.

On designing this PLL, we focus on three aspects: the frequency locked rate, the lock efficiency and the capability of anti-frequency drift. For good efficiency in every aspect within wide capture range under noisy environment, we design three different states which include the acquisition state, the tracking state and the phase-fixing state. Moreover, we introduce the dual-loop system to further improve the performance. Finally, to verify the feasibility of our approach, we implement this PLL by Hardware Description Language based on all-digital design.

誌謝

當我需要寫到個段落時，也代表著我的交大生涯已在開始倒數，謹以此篇文章以表兩年來給予我幫助的人獻上深深的敬意。

首先在我研究所期間影響最深的當屬我的指導教授高銘盛老師，除了專門學問之外，老師給予我最大的學習就是何謂研究的態度，或許在未來我所扮演的角色會變，研究的學問也會不斷的更新，但是這份學習精神卻是不容遺忘的。此外還有多虧兩位大學長，兩位中學長和兩位小學弟，讓我兩年來在研究上不至於孤軍奮鬥，也充實我碩士生活的回憶。最後當然也要感謝我的家人父母，讓我可以在這段時間安心的做好學生的本分，以本身的辛勞來換取我學習的資源，就是這份支持才足以支撐我在求學期間所遭遇到的總總挫折與困難，讓我最後能成功的獲得交大碩士的肯定，謝謝你們。

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 PLL Overview

The phase-locked loop (PLL) is a very critical component in modern electronic circuits [1]. It is able to generate a stable output signal whose phase and frequency are the same as those of the input. The PLL is also a feedback system, whose output is directly connected to the input. Therefore the system will adjust its output signal constantly until the phase/frequency difference between input and output is near to zero.

1.1.1 History and Application

The history of PLL can be traced back to 80 years ago. Originally it had been used for the synchronous detection of radio in the 1930s. During the 1950s, it was used for television broadcasting and satellite tracking. Because of the high cost, the application of PLL was rather limited at that time. In recent years, thanks to the progress of the manufacturing technology of VLSI and the system on a chip (SoC) design, the scope of application is increased significantly. Until now, the role of PLL in system design is more and more important, which has become a versatile multi-function integrated circuit after the operational amplifier [2].

The phase-locked loop is widely used in the televisions, telecommunications, computers, and other electronic applications [3], [4]. It is able to generate or synthesize stable frequencies to select broadcasting stations or channels, synchronize the carrier frequency from noisy channel to demodulate the communication signals coherently, track the unknown frequencies in wireless system, or recover clock pulses in digital logics. Therefore, the wide application range of PLL leads to continuous research on PLL circuits.

1.1.2 All-Digital Phase-Locked Loop

Although the PLL is applied ubiquitously, there are still some problems with integration of traditional PLL at the current trend. In the traditional designs, most of them use analog components to build the PLL. That is, all the circuits are implemented with the resistances, capacitances, or inductances. But the area of these passive components in IC manufacturing is relatively large and the exact values of them are not easy to be grasped. Moreover, in today's single-chip system, merging analog circuits and digital logic on the same chip is inevitable, but it may cause many electrical problems such as noise interference, signal mask, power supply stability, etc [5], [6]. These will increase the difficulty in circuit verification and enlarge the risks of research and development.

Instead of analog approach, we could design the PLL by all-digital circuits to replace analog designs for avoiding above troubles. This PLL is called the "all-digital phase-locked loop (ADPLL)". By this way, many of the problems mentioned above can be removed between the integration of analog and digital. Furthermore, in contrast to the analog PLL, the adaptability of ADPLL is rather high. Due to the programmable capability, it is easy to change the parameters for satisfying different system requirements, and then reducing the cost and time of development [7]-[9]. This is also the main reason why we choose the all-digital structure as our research topic.

1.2 Research Motivation

1.2.1 Motivations and Purposes of this Research

For most PLL, their capture ranges are narrow usually. With a narrow capture range, the

PLL can only work within a narrow frequency band; therefore we intend to design a PLL with ultra-wide capture range. This PLL is designed to work for the input frequency which spans from 1kHz to 1MHz, and we don't have any prior information about the signal frequency. In this case, the ratio of the highest input frequency to the lowest input frequency is up to 1000. Moreover, we hope to have a good lock-in range and lock time to increase the application range. As displayed in Table 1.1, there are several researches about the PLL whose capture range ratios are over 3 at least. The lock time of analog PLL is larger in general such as the PLL in [5]. Although the PLL in [10], [11] have pretty good lock time, their phase resolutions, which isn't shown in this table, are very poor. In other words, they have worse jitter performance even if they are in the locked-state. Therefore, in contrast to [7], [8], [12], we hope the lock time could be less than 100 cycles in our design. We don't want to pay too much price for wide capture range.

Furthermore, there are other researches about ultra-wide capture range of PLL so far, but all of them have to work in the noiseless environment [7], [11], [13]-[15]. This is obviously not suitable for practical applications; it is necessary that the PLL is required to have good noise immunity.

Table 1.1 The performance parameters about several PLL systems

Performance parameter	Ref [5]	Ref [7]	Ref [8]	Ref [10]	Ref [11]	Ref [12]
Circuit type	Analog	All-Digital	All-Digital	Digital	All-Digital	All-Digital
Capture range (Hz)	8.5M-660M	2M-500M	87M-250M	0.3G-0.8G	50M-550M	500M-1.5G
Lock time (reference cycles)	720	63	72	16	50	96

1.2.2 Research Approaches

Because of the ultra-wide capture range, it is difficult to use only one control algorithm to handle all the status such as capturing the unknown signal or tracking the signal which is under control. Thus we classify the process of our system into multiple states [10], [11], [16]. Here we design three states for our system: the acquisition state, tracking state and phase-fixing state. Each state uses different control rule for different purposes. For example, we adopt comparatively wide loop bandwidth in acquisition state for capturing the input frequency quickly, but a narrow bandwidth phase-fixing state for fine-tuning.

Additionally, we intend to combine two loops together to get better performance. The PLL which has dual-loop is helpful in various aspects. For example, it uses the dual cascaded PLL to increase the capture range in [17], [18], and in [19] it could also enhance the phase resolution due to two loops. With dual-loop, we can get shorter lock time and wider lock range in our system. Finally, we will implement the PLL system by the digital Hardware Description Language (HDL) to verify the feasibility of our design.

1.3 Thesis Organization

This thesis is organized as following: In Chapter one, we introduce the background of PLL briefly. Next, we describe the structure of traditional PLL to understand its behavior in Chapter two. Chapter three will present all the details of our design concept, including the system architecture and the control algorithm. In Chapter four, we show the hardware circuit architecture of all the designed modules and summarize the operation of this system. Then, in Chapter five we show the simulation results, and make conclusions of this research in Chapter six finally.

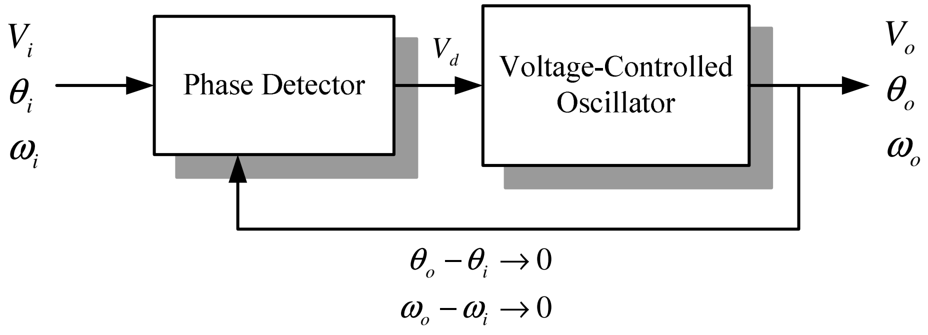
Chapter 2

Phase-Locked Loop Basics

In this chapter, we will review some critical concepts about PLL and introduce three types of them: analog PLL (APLL), digital PLL (DPLL) and all-digital PLL (ADPLL). At first, we use a linear model to express the PLL system. Although most PLL are not linear, we can assume the input of PLL is a pure sinusoid and the system already in the lock-state. Then it can be represented as a linear system [1], [2].

2.1 Linear Model of PLL

The function block of PLL is shown in Figure 2.1. The phase detector (PD) is responsible for detecting the differences of frequency and phase between two input signals. Then the output voltage v_d is entered into the voltage-controlled oscillator (VCO). The VCO will send a new signal back to the phase detector accordingly. Usually, the objective of PLL is to generate a pure sinusoidal and merely two components PD and VCO are necessary to achieve it easily.



V_i
θ_i
ω_i
Phase Detector
V_d
Voltage-Controlled Oscillator
V_o
θ_o
ω_o
$\theta_o - \theta_i \to 0$
$\omega_o - \omega_i \to 0$

Figure 2.1 The general function block of PLL

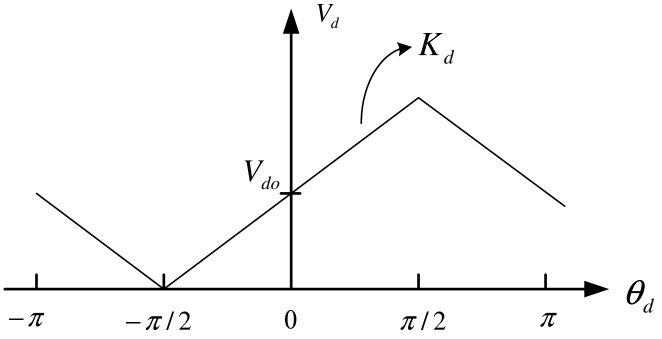


Figure 2.2 The phase detector characteristic

2.1.1 Phase Detector Characteristics

Because frequency is the time derivative of phase, we can use the phase difference θ_d to represent both phase and frequency differences between two inputs as below:

$$\theta_d = \theta_i - \theta_o \tag{2.1}$$

The voltage v_d is produced in response to θ_d. The relationship between v_d and θ_d is shown in Figure 2.2, which is normally a linear curve. In this figure, K_d is the PD gain and it is a constant in the range $-\pi/2 \leqq \theta_d \leqq \pi/2$, which is also the effective detection range of PD. The V_{do} corresponds to $\theta_d = 0$, which means there is no difference between input phase and VCO phase. Next, we can model the phase detector as

$$v_d = K_d\theta_d + V_{do} \tag{2.2}$$

2.1.2 VCO Characteristics

The voltage-controlled oscillator, as its name implies, can generate an output signal whose frequency is controlled by the input voltage. The I/O relationship of VCO is shown in Figure 2.3, where v_c and ω_o represent the input voltage and output frequency, respectively, and the slope K_o is assumed to be constant for simplifying the analysis. When the PLL is in lock, the frequency deviation $\Delta\omega_o$ is zero, where

$$\Delta\omega_o = \omega_o - \omega_i \tag{2.3}$$

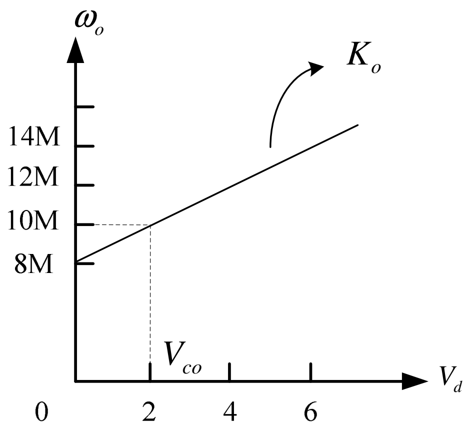


Figure 2.3 The Voltage-Controlled Oscillator characteristic

As shown in the Figure 2.3, in which ω_i = 10 MHz, and $v_c = V_{co} = 2$ is corresponding to ω_o=ω_i. Beware that V_{co} is not a property of the VCO; it depends on the input frequency ω_i. The frequency deviation of VCO can be modeled as

$$\Delta\omega_o = K_o(v_c - V_{co}) \tag{2.4}$$

2.1.3 Frequency Response of Linear PLL Model

As mentioned above, we assume the input and output signals are pure sinusoids, that is:

$$v_i = \sin(\omega_i t + \theta_i) \tag{2.5}$$

$$v_o = \sin(\omega_o t) \tag{2.6}$$

where the θ_i is the exact phase difference between v_i and v_o. But for phase detector, it supposes that the frequency of v_i and v_o is the same. In other words, the phase detector believes that all differences between v_i and v_o are made by both phases. Hence we can rewrite v_o as

$$v_o = \sin(\omega_i t + \theta_o) \tag{2.7}$$

Then, the output frequency of VCO can be expressed as

$$\omega_o = d(\omega_i t + \theta_o)/dt = \omega_i + d\theta_o/dt \tag{2.8}$$

We define $\Delta\omega_o = \omega_o - \omega_i$, thus

$$\Delta\omega_o = d\theta_o/dt \tag{2.9}$$

or

$$\theta_o = \int \Delta\omega_o \, dt \tag{2.10}$$

Now we combine above equations with PD and VCO, and the resulting linear model of PLL is shown in Figure 2.4.

Next, we discuss the frequency response of this model. From the frequency response we can observe if ω_o can reasonably follow ω_i which may vary temperately. We form an AC model by ignoring the DC parameters from Figure 2.4, and the result is shown in Figure 2.5, where $1/s$ represents the Laplace transform of integration. Let the open loop gain of the model in Figure 2.5 is $G(s)$, then

$$G(s) = K_d K_o \,/\, s \tag{2.11}$$

The transfer function is

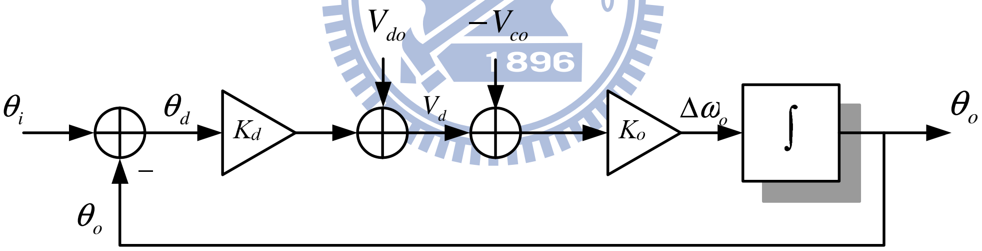


Figure 2.4 The linear model with PD and VCO

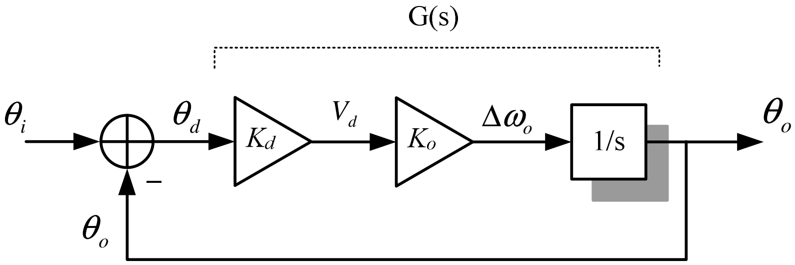


Figure 2.5 The AC model with PD and VCO

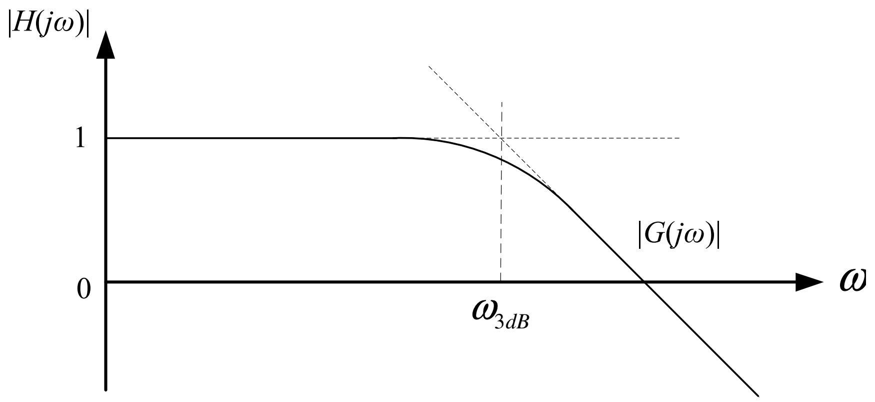


Figure 2.6 The frequency response and bandwidth of linear PLL model

$$H(s)=\frac{\theta_{o}(s)}{\theta_{i}(s)}=\frac{G(s)}{1+G(s)}=\frac{1}{1+s/K_{d}K_{o}} \tag{2.12}$$

For describing the frequency response, we replace s by $j\omega$, and the response $|H(j\omega)|$ is shown in Figure 2.6. When ω is low, $|H(j\omega)|$ is about equal to unity. On the other hand, when ω is high, $|H(j\omega)|$ approaches $|G(j\omega)|$. Notice that the meaning of ω here is not the same as ω_i or ω_o, which represents the varying rate of frequency. Concerning frequency response, we can see that the bandwidth $\omega_{3\mathrm{dB}}$ occurs when $|G(j\omega)| = 1$. That is

$$\omega_{3\mathrm{dB}} = K_dK_o \tag{2.13}$$

2.1.4 Loop Filter Characteristic

Now suppose we want to reduce the bandwidth but keep PD and VCO unchanged, we can put a voltage attenuator between PD and VCO. The objective of voltage attenuator is to decrease the gain so as to reduce the bandwidth, but we intend to decrease the AC gain only and don’t limit the DC voltage. The solution is to use the RRC low pass filter which is also called the loop filter (LPF) as shown in Figure 2.7. This filter acts as an attenuator at high frequencies but with unity gain at DC. The transfer function of the LPF is

$$F(s)=K_{h}\frac{s+\omega_{2}}{s+\omega_{1}} \tag{2.14}$$

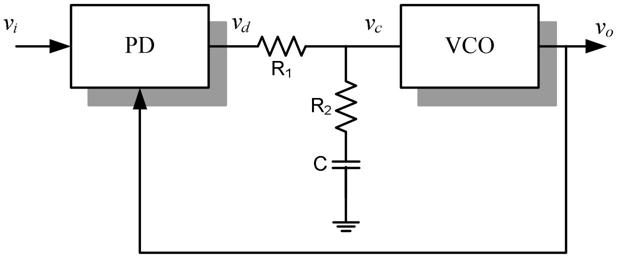


Figure 2.7 A RRC low pass filter

where

$$K_h = \frac{R_2}{R_1 + R_2}$$
$$\omega_1 = \frac{1}{(R_1 + R_2)C} \qquad (2.15)$$
$$\omega_2 = \frac{1}{R_2 C}$$

The frequency response of $|F(j\omega)|$ is plotted in Figure 2.8. The gain $|F(0)| = 1$ at DC and $|F(j\omega)| = K_h$ at high frequencies. After including $F(s)$, the open loop gain of PLL is modified as

$$G(s) = K_d F(s) K_o / s \qquad (2.16)$$

The new frequency response of $|H(j\omega)|$ is shown in Figure 2.9, and the 3dB bandwidth becomes

$$\omega_{3\text{dB}} = K_d K_h K_o = K \qquad (2.17)$$

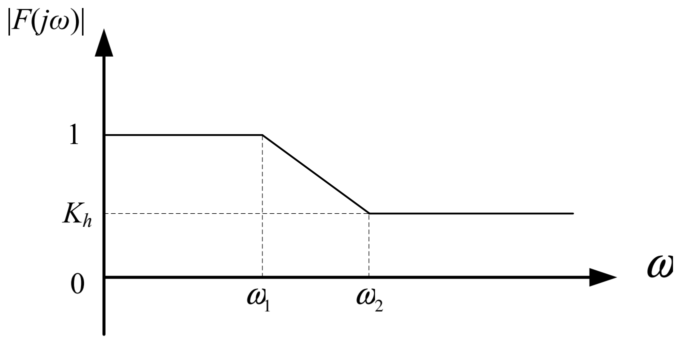


Figure 2.8 The frequency response of RRC low pass filter

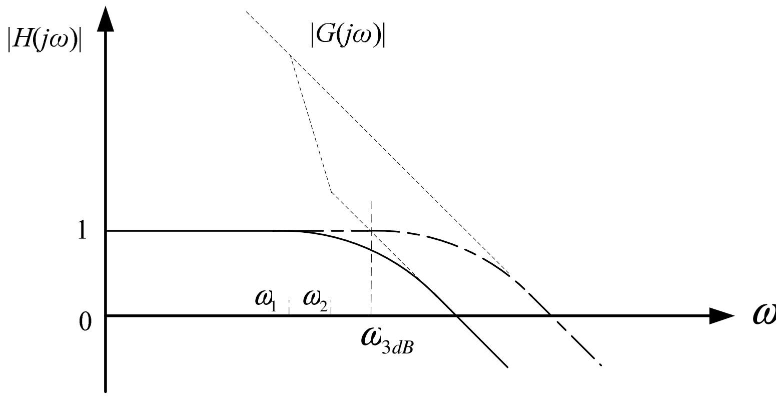


Figure 2.9 The frequency response and bandwidth of complete linear PLL model

Finally, a complete linear model with the LPF is shown in Figure 2.10, and the transfer function is

$$H(s) = \frac{\theta_o(s)}{\theta_i(s)} = \frac{Ks + K\omega_2}{s^2 + (K + \omega_1) + K\omega_2} \tag{2.18}$$

Because the denominator of this transfer function is a second-order polynomial in *s*, it is called a second-order PLL. The different loop filter affects the order of PLL, which may affect the different characteristic or stability of the system loop. Therefore, for simplifying analysis, the PLL design could follow the linear model usually.

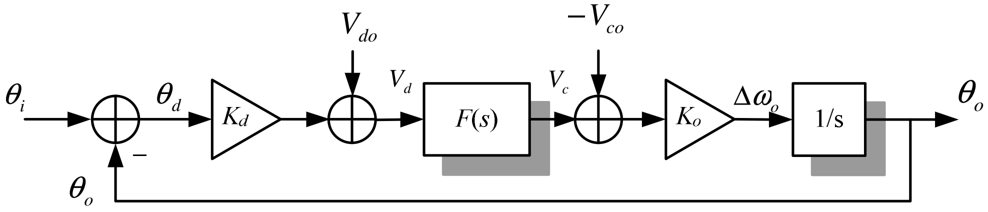


Figure 2.10 The complete linear PLL model

2.2 Traditional PLL Circuit Design

This section we introduce various types of PLL which are classified according to the circuit structure. As mentioned above, the APLL is constituted by analog components only, the DPLL combines analog circuits with digital, and the ADPLL has the all-digital implementation.

2.2.1 Analog PLL (APLL)

The analog PLL consists of the PD, the LPF, and the VCO as mentioned in Section 2.1. A four-quadrant multiplier is used as the PD for detecting frequency deviation at the earliest. Let two inputs of the multiplier be

$$v_i = \cos(\omega_i t) \tag{2.19}$$

$$v_o = \sin(\omega_i t + \theta_e) \tag{2.20}$$

Suppose the gain of multiplier is K_m, and the output of multiplier is given as

$$v_d = v_i v_o = \frac{K_m}{2}(\sin(\theta_e) + \sin(2w_i t + \theta_e)) \tag{2.21}$$

Because of the LPF, the high frequency part is removed. Thus

$$v_d = 0.5K_m \sin(\theta_e) \tag{2.22}$$

The relationship between θ_e and v_d is shown in Figure 2.11. Comparing with Figure 2.1, the linear range of PD is very narrow unless the value of θ_e is small, and the effective detection range is also [$-\pi/2$, $\pi/2$].

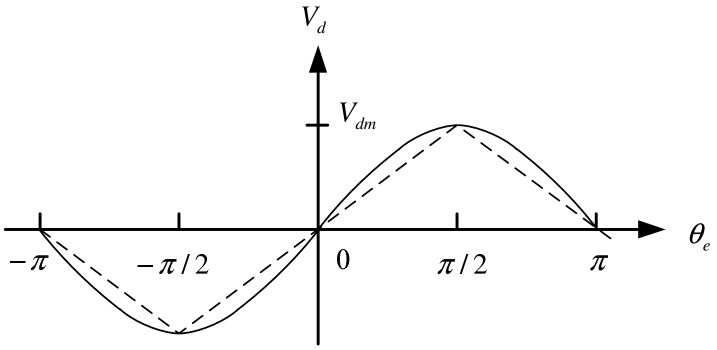


Figure 2.11 The multiplier PD characteristic

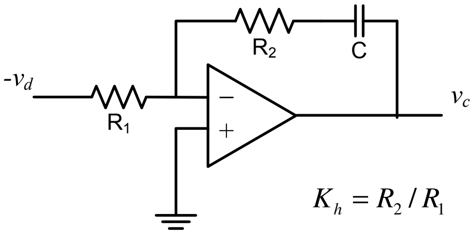


Figure 2.12 The PI filter circuit

For the LPF, instead of the RRC low pass filter, the proportional integrator filter (PI filter) is also a good choice as shown in Figure 2.12, which has the better loop bandwidth than the RRC. However, comparing with the DPLL and ADPLL, the lock time of APLL is rather slow relatively.

2.2.2 Digital PLL (DPLL)

As shown in Figure 2.13, the main difference between DPLL and APLL is that the DPLL uses the digital phase frequency detector (PFD) and the charge pump (CP) to replace the analog PD.

The DPLL usually uses the Exclusive-OR gate (XOR) or Flip-Flop as the PFD. Since the behavior of XOR mimics the multiplier, we can’t get too many advantages. Two PFDs constituted with D-Flip-Flops are depicted in Figure 2.14, where the inputs v_i and v_o are square waves.

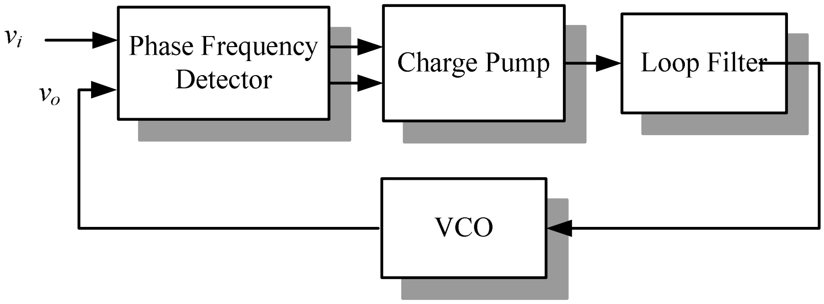


Figure 2.13 The block structure diagram of DPLL

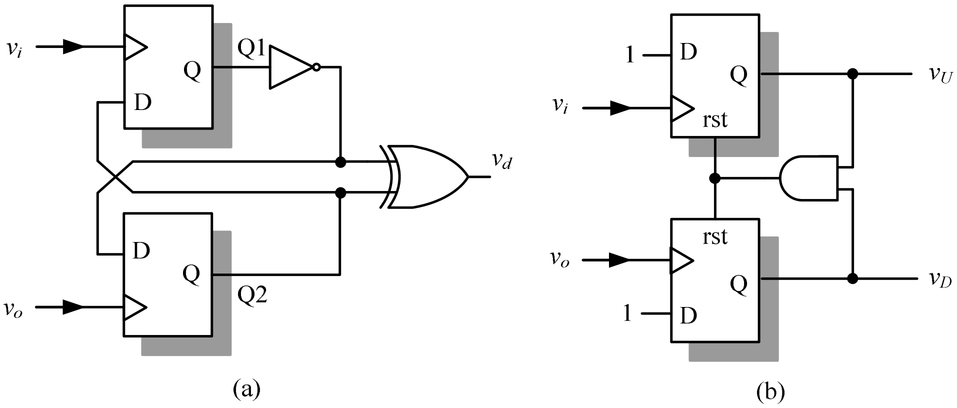


Figure 2.14 (a) The two-state PFD structure (b) the three-state PFD structure

The PFD of Figure 2.14a has two states, where v_i and v_o act as clock input of the two D-Flip-Flops, respectively. In the beginning, assume Q1 and Q2 both are low, and v_d is high. When positive edge of v_o is coming, Q2 becomes high and then makes v_d low. When positive edge of v_i comes, Q1 becomes high and then make the v_d high. In contrast to Figure 2.14a, the circuit of Figure 2.14b replaces v_d with two outputs v_U and v_D to has more states than the former. The I/O relationships of them are shown in Figure 2.15 below. We find that the effective detection range of two-state PFD is $\pm\pi$, and the three-state PFD is up to $\pm 2\pi$. It is more useful than the PD of APLL.

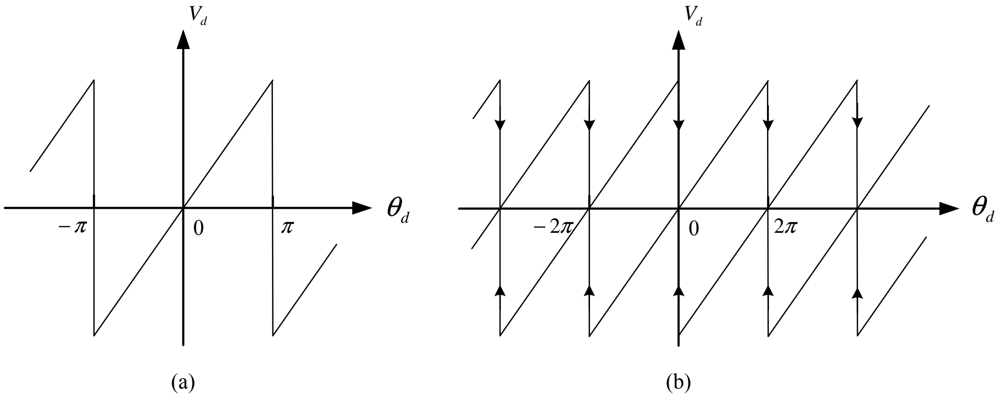


Figure 2.15 The PFD characteristic of (a) two-state PFD (b) three-state PFD

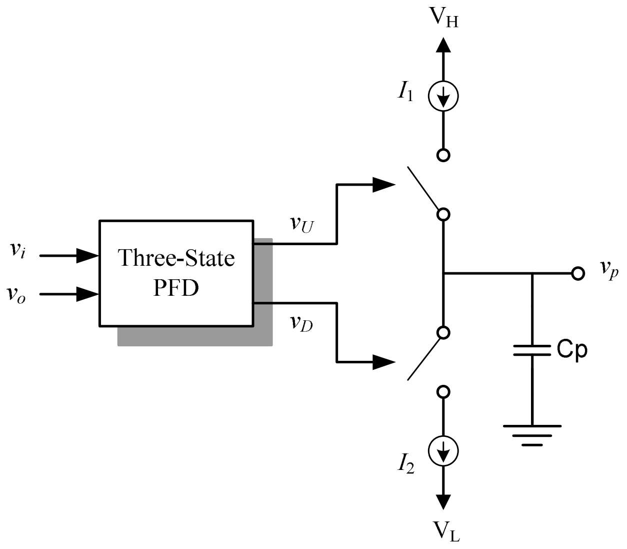


Figure 2.16 The charge pump structure

Additionally, the charge pump can transfer the logic signal to be voltage. The structure of charge pump with three-state PFD is given in Figure 2.16. As shown, the capacitor C_p charges when v_U is high and v_D is low, discharges when v_U is low and v_D is high, and keeps the status unchanged when both v_U and v_D are low.

A disadvantage with DPLL is that the output voltage v_p of CP may be unstable when the capacitor C_p charges and discharges inconsistently or the charge currents I_1 and I_2 don’t match with each other. The self-generated noise is capable of destroying the VCO phase, so that additional components are necessary for eliminating the noise disturbance.

2.2.3 All-Digital PLL (ADPLL)

From the above discussion, we find that maintaining the stability of APLL or DPLL is usually complicated, but most analog components have low tolerance for noise. Therefore we may apply the algorithm of behavior description to replace the operation of circuit components. That is the ADPLL, whose function block diagram is shown as Figure 2.17.

The PFD, which is similar to the DPLL’s, is connected to the control unit. The control unit is a general block; any helpful control blocks about the system can be included inside this

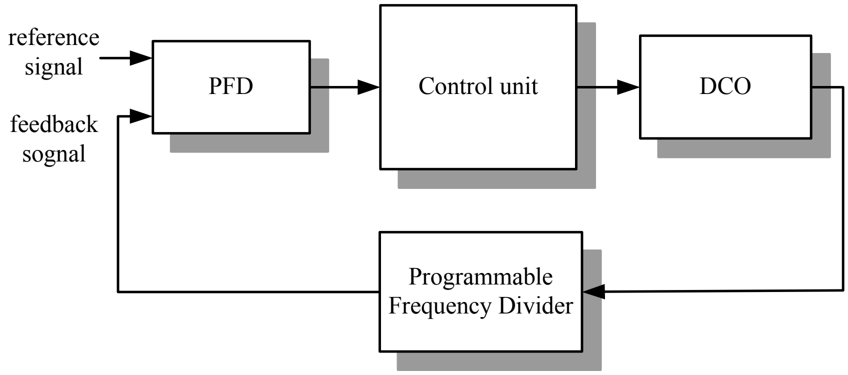


Figure 2.17 The function block diagram of ADPLL

unit, and it isn't confined to only the loop filter. Usually we use the time-to-digital converter (TDC) and the digital recursive filter (or named digital LPF) as the core of control unit, because they can exactly replace the CP and LPF of DPLL, respectively. The TDC translates the PFD output into digital codes, and then the digital recursive filter decodes these codes and generates another digital code to adjust the DCO.

The DCO is the digital-controlled oscillator. Obviously, the difference between DCO and VCO is that the output of DCO is controlled by digital codes, but not voltage. The main advantage with DCO is that voltage is more sensitive than digital code, so that DCO can generate a comparably stable output frequency. But there is the resolution problem for DCO since the number of digital codes is limited. In other words, the VCO is able to generate more accurate output frequency than DCO.

2.3 An ADPLL Circuit Model

Here we will analyze the ADPLL in details [14], [20]. Figure 2.18 shows the block diagram of ADPLL [14]. The PFD produces series pulses *Up* and *Down* with lengths represent the phase differences, and the *Update* is used to clock the registers and reset the TDC between measurements. The TDC translates the lengths of *Up* and *Down* into digital

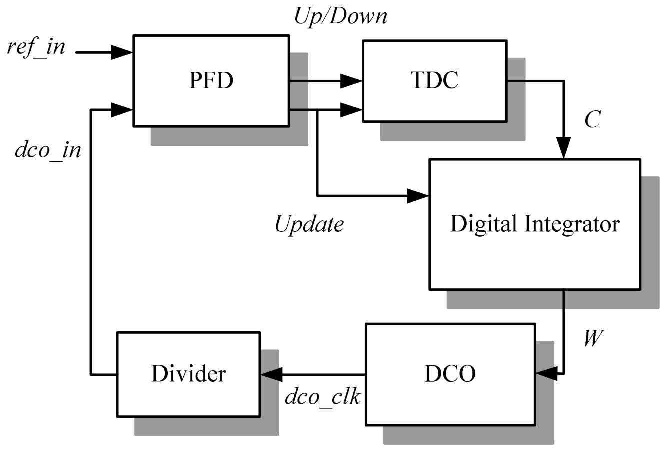


Figure 2.18 An ADPLL model

code *C* as mentioned above. Then the digital integrator determines the digital word *W* according to *C*. Finally, the DCO generates the output *dco_clk* which is compared with the reference *ref_in* in the PFD after frequency division by the divider.

2.3.1 Phase Frequency Detector (PFD)

The PFD also uses the three-state PFD but we make some improvements as illustrated in Figure 2.19. In the original PFD, if the *ref_in* leads the *dco_in*, the *Down* output still has short pulse. This is not desirable so that we add two inverters and two AND gates. In this way, the *Up* output is high when *ref_in* leads the *dco_in* and the *Down* output is high when *ref_in* lags the *dco_in*, as shown in Figure 2.20.

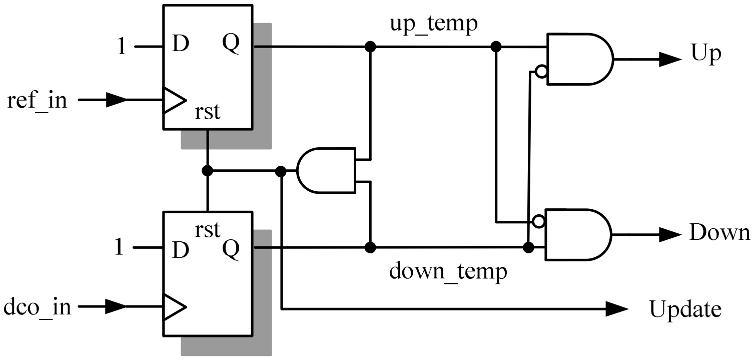


Figure 2.19 The improved structure of three-state PFD

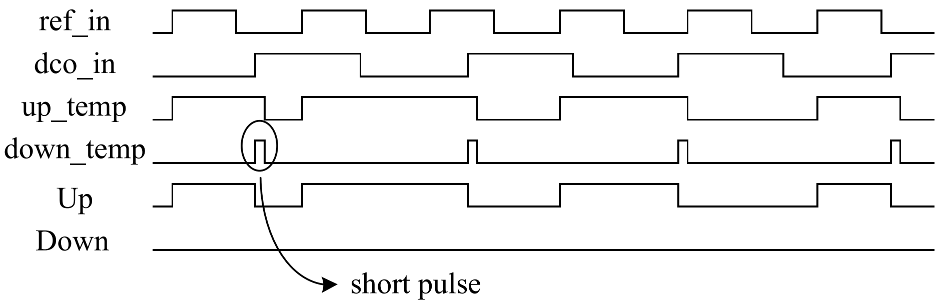


Figure 2.20 The timing diagram of new three-state PFD

The short *Update* output is necessary. Because the system is controlled in digital manner, the *Update* is a good trigger to be the time unit of system.

2.3.2 Time to Digital Converter (TDC)

The TDC could be complicated, but we can use the counters to simplify it, as shown in Figure 2.21. Intuitively, the longer *Up*/*Down* stands for the larger phase difference. We use

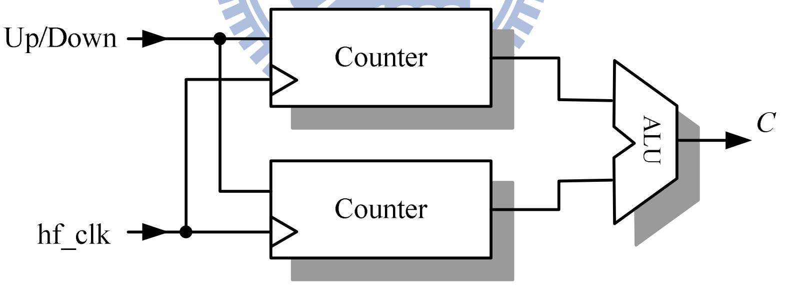


Figure 2.21 The TDC structure

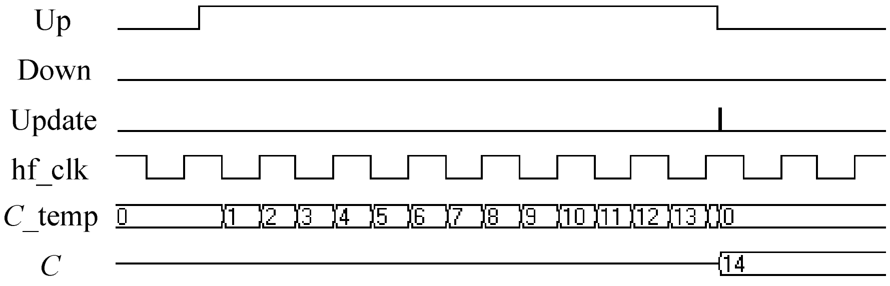


Figure 2.22 The timing diagram of TDC

high frequency signal *hf_clk* as the clock of counter; we can even combine both positive and negative edges to double the precision. The timing diagram is shown in Figure 2.22.

When the *Up* or *Down* output is high, the counters will start to work. Until the positive edge of *Update* is detected, the counters send results to the ALU and go to zeros afterwards. Then, the code *C* will represent the phase difference in this round.

2.3.3 Digital Integrator

The digital integrator, the recursive filter or the digital loop filter, which is essentially the same. In order to approximate the second-order loop filter as mentioned above, we let the transfer function of integrator be

$$F(z) = \frac{1}{1-0.5z^{-1}-0.5z^{-2}} \tag{2.24}$$

For simplicity, we only use the number 1 or 0.5 as constant, which can be expressed correctly by limited bits. Converting to digital domain, we get

$$y[n] = x[n] + 0.5y[n\text{-}1] + 0.5y[n\text{-}2] \tag{2.25}$$

This is just the behavior of the integrator. The input *x*[*n*] and output *y*[*n*] are *C* and *W*, respectively; *W*[*n*] is the function of *C*[*n*], *W*[*n*-1], and *W*[*n*-2]. The structure is shown as Figure 2.23. The *Update* is the delay clock, and *Up*/*Down* is helpful to decide the sign of *C*.

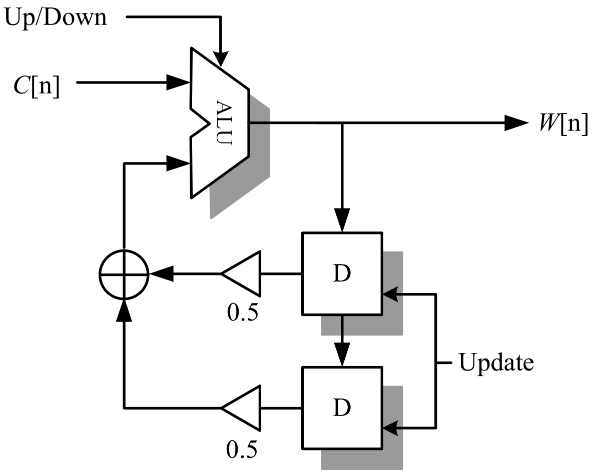


Figure 2.23 The digital integrator structure

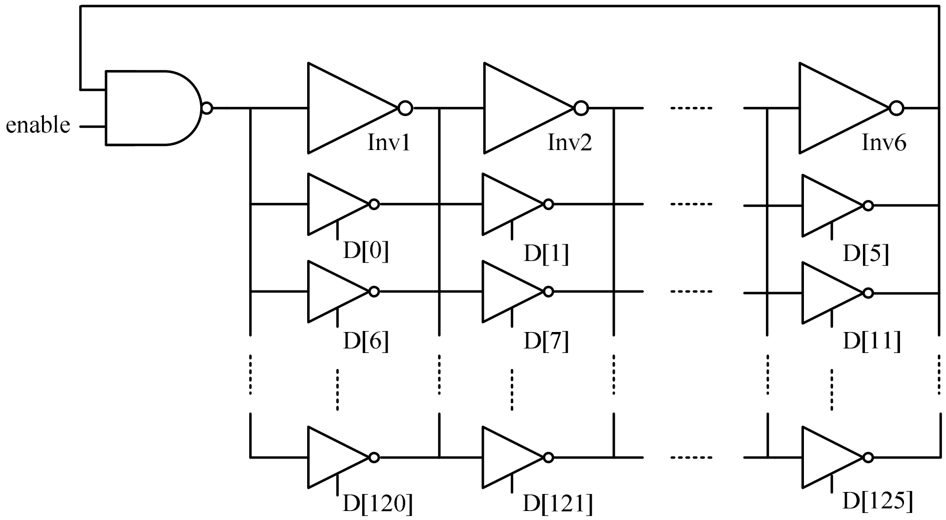


Figure 2.24 DCO constituted by parallel tri-state inverter

When *Up* is high, *W* increases with the positive *C*. Oppositely, *W* decreases with the negative *C* when the signal *Down* is high. Additionally, *W* has the initial value; this value can cause the DCO to generate the initial frequency.

2.3.4 Digital-Controlled Oscillator (DCO)

A DCO is shown in Figure 2.24. It is an evolution from the ring oscillator, which consists of 126 tri-state inverters controlled by a 126-bit digital code *D*, and *D* is decoded from the 7-bit word *W*. When *W* = 0, all the inverters are off and the charge efficiency of the MOS should be lower. Otherwise, when *W* = 126 or 127, all the inverters are on and the DCO can generate higher frequency.

2.4 Design Challenges

From above descriptions, we have preliminary knowledge about the operation of ADPLL. However, for working with the wide capture range under noisy environment, the above system is not sufficient. There are some challenges to be overcome:

Challenge 1:

Although the effective detection range of three-state PFD is up to ±2π, it is still difficult to handle the capture range which has 1000 times between the highest frequency and the lowest frequency. Moreover, in order to reduce the lock time, it may be unrealistic to have a long observation time. How can the PFD detect correct phase difference from the ultra-wide frequency range?

Challenge 2:

Under noisy environment, the three-state PFD is easily affected by noise. As shown in Figure 2.25, a small pulse can cause series wrong judgment of the phase, let alone under serious noise environment. How can the PLL determine if the trigger is caused by noise or by signal?

Challenge 3:

Because the input code length of DCO is limited, the wider range causes the lower resolution. In our DCO, the lowest frequency and the highest frequency are respectively 1kHz and 1MkHz. Even if the codeword W is 10-bit, it only generates 1024 different frequencies. How can the DCO get fine resolution under limited bits?

Challenge 4:

As mentioned earlier, the LPF bandwidth is the key of system stability. With wide capture range, it is possible that the difference between reference frequency and center

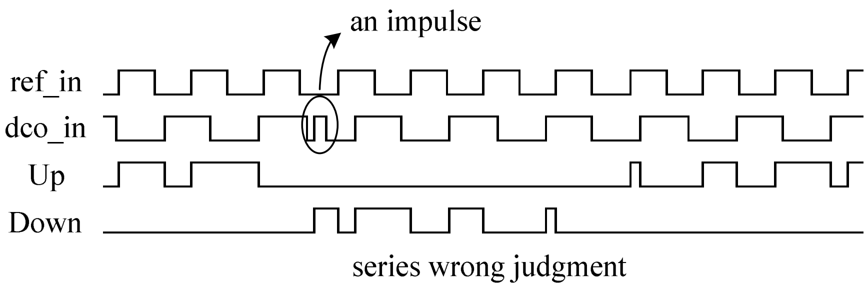


Figure 2.25 A series wrong judgment caused by an impulse

frequency is quite large, and we can’t use a very narrow loop bandwidth to avoid the high frequency interference. How can the LPF get the balance between the efficiency in large frequency difference and the stability in small frequency difference?

Chapter 3
System Architecture

As mentioned above, we classify the process of our system into three states: the acquisition state, the tracking state and the phase-fixing state. In this chapter, we will describe the algorithm and the architecture about the proposed PLL system and present how to combine the three states against the difficulties encountered under serious noise environment.

3.1 Acquisition State

The acquisition state is the initial state in our PLL design, whose purpose is to find the possible region of reference frequency from 1kHz to 1MHz quickly. Because there is no prior information about this frequency, we have to get useful information from the PFD and handle the information adequately.

The traditional three-state PFD only provides the *Up* and *Down* outputs as shown in Section 2.3.1. It is not enough for the requirement of wide capture range. An example is given as Figure 3.1, where we only get the information that the difference between two frequencies is very large from the exaggerated length of *Up*/*Down*, but can't distinguish the difference between these two cases. A PFD which can make an accurate judgment when there is a huge difference between the reference frequency and DCO frequency is necessary.

After some observation, we find it is possible to count the trigger times of positive edge continuously when the *Up*/*Down* is determined already. The concept is equivalent to use the reference frequency as the sampling frequency to sample the DCO signal. If one DCO period

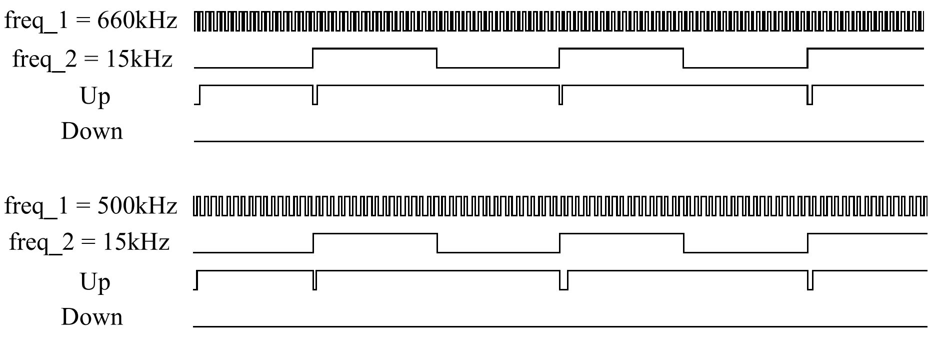


Figure 3.1 Two cases with large frequency difference

is sampled *k* times, the trigger times should be *k-1*, as shown in Figure 3.2. The trigger times can also represent the cycles which the leader exceeds. Although we can't identify the precise reference frequency according to the trigger times, we know the ratio of the reference frequency with respect to DCO frequency approximately.

On the other hand, we divide the 1kHz-1MHz into multiple bands (Figure 3.3). Each band has its own center frequency, and the difference between adjacent center frequencies is double as frequency increases. Then, we use a Divider that can control the DCO to shift its center frequency from one band to the others. Combining above ideas, we can spend only several DCO signal periods to count the trigger times and then shift the DCO signal to the appropriate center frequency closest to the reference frequency.

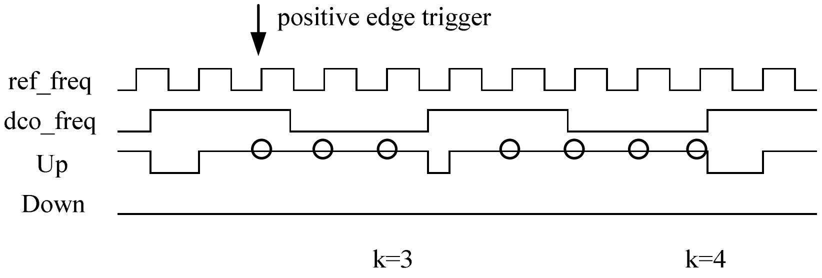


Figure 3.2 The trigger positions

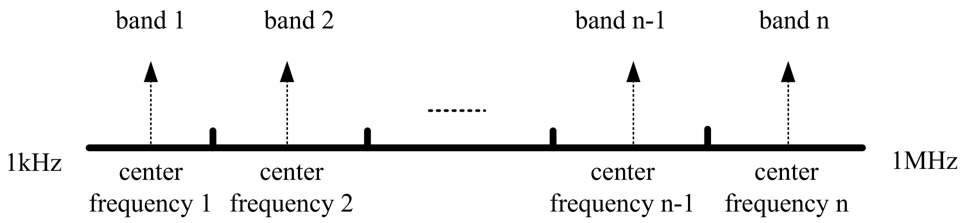


Figure 3.3 The divided bands

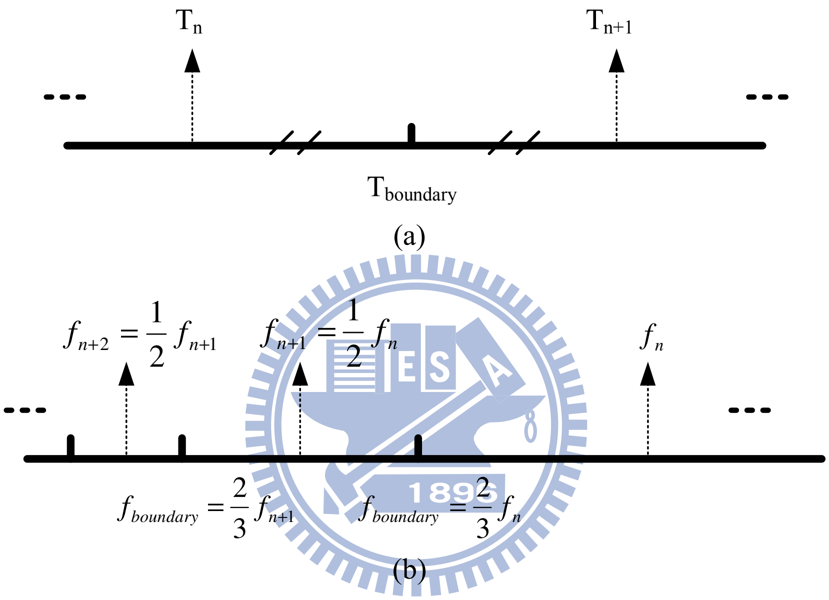

T_n
T_{n+1}
$T_{boundary}$
(a)
$f_{n+2} = \frac{1}{2} f_{n+1}$
$f_{n+1} = \frac{1}{2} f_n$
f_n
$f_{boundary} = \frac{2}{3} f_{n+1}$
$f_{boundary} = \frac{2}{3} f_n$
(b)

Figure 3.4: (a) The boundary in unit of period (b) The boundary in unit of frequency

The decision rule of trigger times is closely related to the approach of dividing bands. For simplify, we use the Divider to divide bands, which helps us to use a simple decision rule for selecting the right band according to the trigger times. As shown in Figure 3.4a, the most intuitive approach in dividing bands is to select the middle of two adjacent center frequencies as the boundary. But our design is based on digital circuits, therefore we replace the frequency with the period as time unit, given as

$$T_{boundary} = \frac{T_n + T_{n+1}}{2} = \frac{T_n + 2T_n}{2} = \frac{3}{2} T_n \tag{3.1}$$

where T_n and T_{n+1} are any two adjacent center periods, and T_{n+1} is twice of T_n. Then

$$f_{boundary} = \frac{2}{3} f_n = \frac{4}{3} f_{n+1} \tag{3.2}$$

As shown in Figure 3.4b, when the reference frequency is over 4/3 times or under 2/3 times than the current DCO center frequency, it jumps to other band.

For the decision rule of trigger times, we have to know the relationship between trigger times and reference frequency. Suppose the initial phase difference between the DCO frequency and the reference frequency is zero for simplifying the problem, and let the reference frequency be 1.6 times of the DCO frequency. This example is shown in Figure 3.5, where (a) shows the phase difference detected by PFD, and (b) shows them on the unit circle. We could find that the trigger happens when the phase difference is beyond a cycle. Therefore, we can use a simple equation to calculate the number of trigger times.

When the reference frequency is m times of the DCO frequency, the phase difference ϕ will be

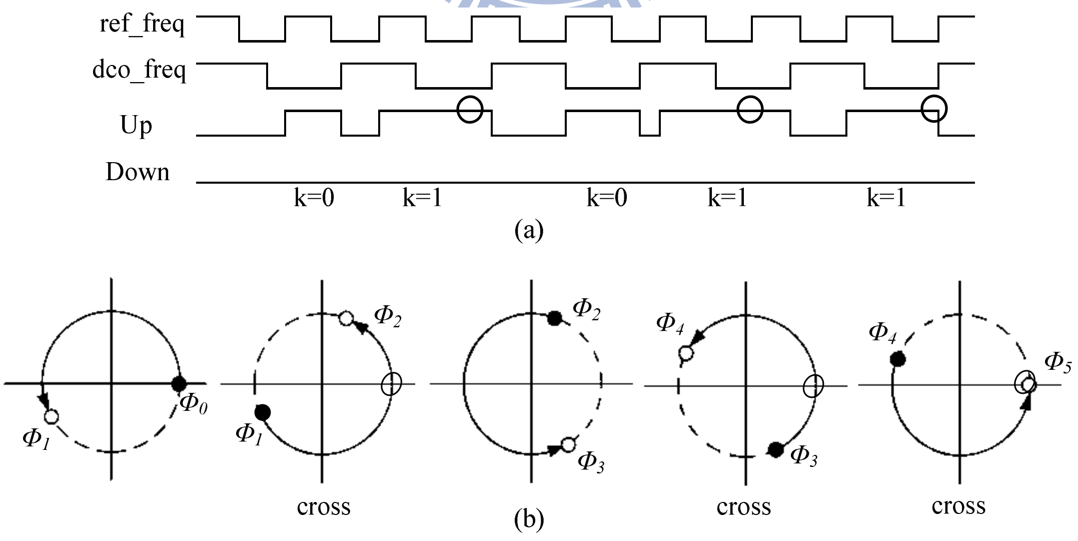


Figure 3.5 The relation between triggers and phase differences

$$\phi_1 = \frac{f_{ref} - f_{dco}}{f_{dco}} \times 2\pi + \phi_0 = (m-1)2\pi + \phi_0 \tag{3.3}$$

where ϕ_0 is assumed to be zero. Then

$$\begin{aligned} \phi_2 &= (m-1)2\pi + \phi_1 = 2\times(m-1)2\pi \\ \phi_3 &= (m-1)2\pi + \phi_2 = 3\times(m-1)2\pi \\ &\vdots \\ \phi_n &= (m-1)2\pi + \phi_{n-1} = n\times(m-1)2\pi \end{aligned} \tag{3.4}$$

Thus the total number of trigger times N_{tri_total} is

$$N_{tri\_total(n)} = \left\lfloor \frac{\phi_n}{2\pi} \right\rfloor = \lfloor n(m-1) \rfloor \tag{3.5}$$

But in our plan, the trigger times have to recount once again per cycle, and we turn the N_{tri} as following:

$$N_{tri(n)} = N_{tri\_total(n)} - N_{tri\_total(n-1)} = \lfloor n(m-1) \rfloor - N_{tri\_total(n-1)} \tag{3.6}$$

Finally, we build a table with different m for observing the trend about trigger times, as shown in Table 3.1.

According to Table 3.1, we find that it is almost 3 rounds per cycle about the boundary trigger times because of our way of dividing bands. For example, at m=4/3 if we observe the sum of any three adjacent N_{tri} that is over 1 (0+0+1), the reference frequency is over 4/3 times of the DCO frequency at least and then double the DCO frequency. Moreover, our purpose is to judge whether the reference frequency is over the boundary or not, therefore we just have to observe two rounds. We take the maximum sum of any two adjacent k points as a threshold and show it in Table 3.2. If the sum of two trigger times is over the threshold, the DCO frequency band will be changed immediately.

Table 3.1 The trigger times for several *m*

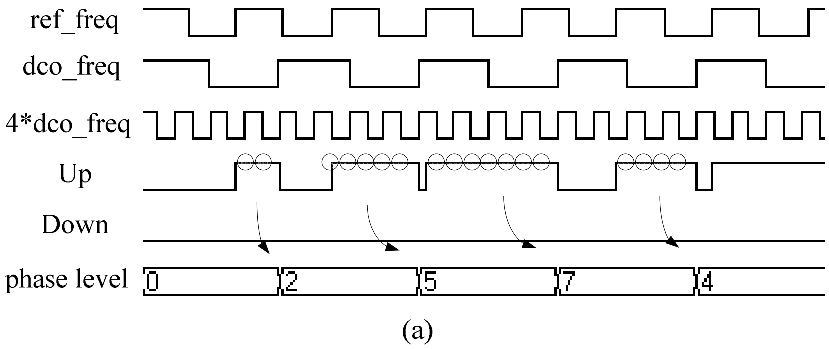
multiples	trigger times k										
m=16/3	4	4	5	4	4	5	4	4	5	4	…
m=4	3	3	3	3	3	3	3	3	3	3	…
m=8/3	1	2	2	1	2	2	1	2	2	1	…
m=2.5	1	2	1	2	1	2	1	2	1	2	…
m=2	1	1	1	1	1	1	1	1	1	1	…
m=1.6	0	1	0	1	1	0	1	0	1	1	…
m=4/3	0	0	1	0	0	1	0	0	1	0	…

Table 3.2 The decision rule of band change

multiples	thresholds	change level
m=32/3	20	*16
m=16/3	9	*8
m=8/3	4	*4
m=4/3	1	*2

We spend two DCO periods to determine where the reference frequency is located if it leads the DCO frequency. Otherwise, if the reference frequency lags t he DCO frequency, it needs two reference periods, too.

In addition to determining the band which the reference frequency is located, we hope the DCO frequency could be closer to the reference frequency in the acquisition state. Recall in Section 2.3.2, we use a high frequency signal to count the length of *Up*/*Down* output of PFD. But which frequency should we choose? According to Figure 3.1 or Figure 3.2, we find the maximum length of *Up*/*Down* is limited to a DCO signal period. It inspires us to use the harmonic frequency of DCO signal instead of a constant high frequency signal to count.



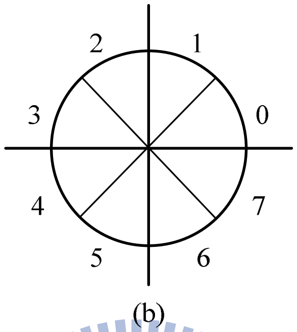


Figure 3.6 (a) Use quadruple DCO frequency to count the phase difference (b) Phase difference level

With the harmonic DCO frequency, we can transform the length into the phase ratio easily. When we select the quadruple DCO frequency to count the *Up*/*Down*, both positive and negative edges can be used. In this way, all the lengths are classified into eight levels and these levels are filled within 2π exactly as shown in Figure 3.6. The DCO will change its frequency according to different levels. A larger level makes a greater change. Therefore, the ratio between the *Up*/*Down* and the DCO period is more significant than the actual length itself.

Next, we observe the variation of levels. The levels should be converged when the DCO frequency is close to the reference frequency. Here we set a condition that if the same levels appear three times in a row, the acquisition state will be ended. As shown in Figure 3. 7, when the specified condition occurs, the phase difference ϕ_n should be

$$\phi_n = \frac{f_{ref} - f_{dco}}{f_{dco}} \times 2\pi + \phi_{n-1} = 2 \times \frac{f_{ref} - f_{dco}}{f_{dco}} \times 2\pi + \phi_{n-2} \qquad (3.7)$$

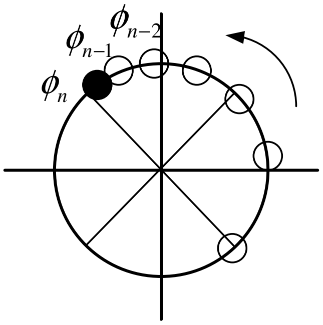


Figure 3.7 The phase difference converges to the same level

and

$$\phi_n - \phi_{n-2} = 2 \times \frac{f_{ref} - f_{dco}}{f_{dco}} \times 2\pi \leq \frac{1}{8} \times 2\pi \tag{3.8}$$

or

$$\frac{f_{ref} - f_{dco}}{f_{dco}} \leq \frac{1}{16} = 6.25\% \tag{3.9}$$

There is about 6.25% difference between f_{ref} and f_{dco} after the acquisition state. Naturally, the 6.25% is an approximation because the DCO frequency is varied each time; the actual frequency difference will be much less than 6.25%.

In summary, in the acquisition state we respectively use the reference frequency and the multiple DCO frequency to count the phase difference and then choose the appropriate band. Next, the system will get into the tracking state.

3.2 Tracking State

In contrast to the acquisition state, the frequency difference is small in the tracking state. Therefore, we have to make a more rigorous decision in the adjustment of DCO frequency. At first, we change the LPF which is used in the acquisition state. Because our architecture is based on digital circuit, the replacement of circuits is fairly easy. In the acquisition state, the LPF is given as

$$y[n] = x[n] + 0.5y[n-1] + 0.5y[n-2] \tag{3.10}$$

where $y[n]$ represents the current DCO frequency, and $y[n\text{-}1]$ and $y[n\text{-}2]$ respectively represent the previous two rounds. Also, $x[n]$ means the current phase difference.

Because in the acquisition state our method is to sweep the band for observing the *Up*/*Down* levels, we average $y[n\text{-}1]$ and $y[n\text{-}2]$ for avoiding the DCO frequency sweeping too fast. But in the tracking state, the DCO frequency is near the reference frequency, thereby the difference between $y[n\text{-}2]$ and $y[n\text{-}1]$ is little. In this case we can adjust the DCO frequency just from $y[n\text{-}1]$. Moreover, in acquisition state the DCO frequency is changed only by the separate Up/Down levels, but it is unsuitable actually. The frequency difference must be represented with the difference of phase difference because the frequency is the time derivative of phase. An example is shown in Figure 3.8, when the *Up* level becomes smaller and smaller, that means the reference frequency is leading less and less. But the DCO signal still increases its code because the *Up* signal appears continuously. The DCO code has a direct impact on DCO frequency. When the phase difference becomes zero, the DCO frequency already exceeds the reference frequency a lot. Then the superfluous behavior will be repeated, that will cause the DCO frequency to oscillate endlessly. Therefore, we replace the single phase level with the phase level difference as the LPF input, and the LPF function would be

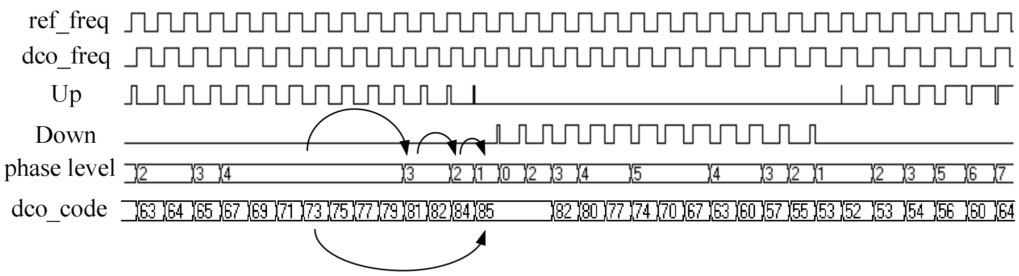


Figure 3.8 The superfluous behavior of DCO frequency

$$y[n] = x[n] - x[n-1] + y[n-1] \tag{3.11}$$

Furthermore, we increase the resolution of phase difference four times in the tracking state. A 16x DCO frequency replaces the quadruple DCO frequency as the high frequency of TDC. It means that 2π is divided into 32 levels now. In this way, the phase difference of PFD can be expressed accurately and the system could make more precise decision.

As in the acquisition state, we also set a condition to determine if the tracking state has been completed, and this condition must be stricter for ensuring the DCO frequency be the same as the reference frequency. In our DCO design, the minimum frequency difference between two adjacent DCO codes is about 0.332%, so that the DCO frequency is appropriate if the difference between the DCO frequency and reference frequency is less than 0.00332/2 = 0.166%. We set the phase levels to be the same *N* times in a row as shown in Figure 3.9, that is

$$\begin{aligned}\phi_n &= \frac{f_{ref} - f_{dco}}{f_{dco}} \times 2\pi + \phi_{n-1} \\ &= \frac{f_{ref} - f_{dco}}{f_{dco}} \times 2 \times 2\pi + \phi_{n-2} \\ &\vdots \\ &= \frac{f_{ref} - f_{dco}}{f_{dco}} \times (N-1) \times 2\pi + \phi_{n-(N-1)}\end{aligned} \tag{3.12}$$

and

$$\phi_n - \phi_{n-(N-1)} = (N-1) \times \frac{f_{ref} - f_{dco}}{f_{dco}} \times 2\pi = \frac{1}{32} \times 2\pi \tag{3.13}$$

where we set $\frac{f_{ref} - f_{dco}}{f_{dco}} \leq 0.166\%$, then we get

$$N \geq 19.78 \tag{3.14}$$

It means if the difference between f_{ref} and f_{dco} is 0.166%, we can count about 19.78 times under the same level. If the counts are less than 19.78 in the same level, the difference ratio is

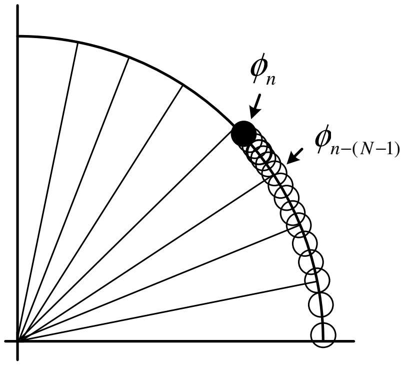


Figure 3.9 The phase levels converge to the same level *N* times

over 0.166%. Thus we set *N*=20 finally. If the condition is reached, the frequency is assumed to be locked.

However, according to Figure 3.9, we find that there is a constant phase difference ϕ_{n} although the frequency difference is zero. An example is shown in Figure 3.10. Because we use the level difference as the basis of adjusting frequency, the initial phase in the tracking state is ignored. Thus we next enter into the third state, the phase-fixing state, to resolve the constant phase.

3.3 Phase-Fixing State

A constant phase can be taken as a fixed time delay. The most intuitive approach to eliminate the time delay is to delay the DCO signal. But this is impractical, since the PLL is a

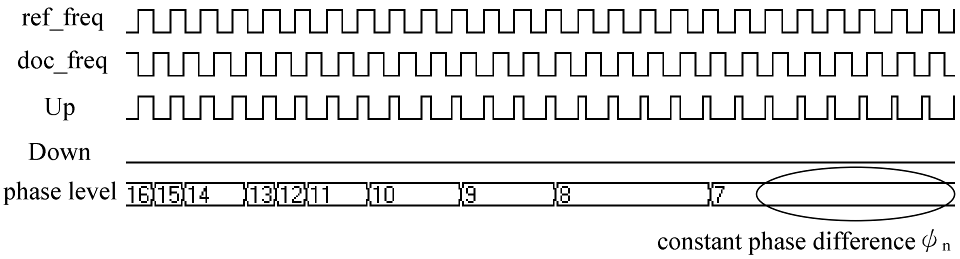


Figure 3.10 Constant phase difference

sensitive feedback system. If we delay the DCO signal directly, the PFD may misinterpret it as a low frequency signal. Then the system will break the steady state. Moreover, the damage of delaying signal is too strong for the waveform in spectrum. Sometimes we have to eliminate the phase repeatedly, but delaying signal for many times will cause the DCO signal with large phase errors.

Although we can not delay the DCO signal directly, we could make it indirectly. First, we change the high frequency of TDC for the 64x DCO frequencies, and there are 128 levels per cycle, which represents the phase resolution is about 3 degree for our system. But different from the tracking state, the 128 levels are just the reference in this state; they don't affect the DCO output. The DCO is controlled only by the value of *Up*/*Down*, not the length. Therefore the LPF in the phase-fixing state is

$$y[n] = y[n-1] \pm 1 \tag{3.15}$$

The sign represents the *Up*/*Down*, respectively. The constant number 1 means that the DCO only makes the least adjustment according to the *Up*/*Down* signal. Moreover, we record the information of DCO frequency when the tracking state is ended. Because of the digital circuit, we can use registers to store the DCO code easily. In this way, we can reduce the DCO period to decrease the constant delay indirectly. When the DCO signal is out of control, the system can recover the DCO frequency from the stored DCO code.

Moreover, there is another reason for DCO code recovery. In our setting, the set of series levels is {30, 15, 7, 3, 1, 0}. When the phase difference is less than these levels, which means the two phases are very close and we have to handle the DCO more cautiously. As shown in Figure 3.11, the initial phase is 89, and the stored DCO code is 777. In this example, the reason of DCO code recovery in the first is the DCO code which will be over 50 than the stored code, and the others are the phase levels which are small enough. Finally the phase difference will be zero and the DCO signal maintains stable. The objective of PLL is

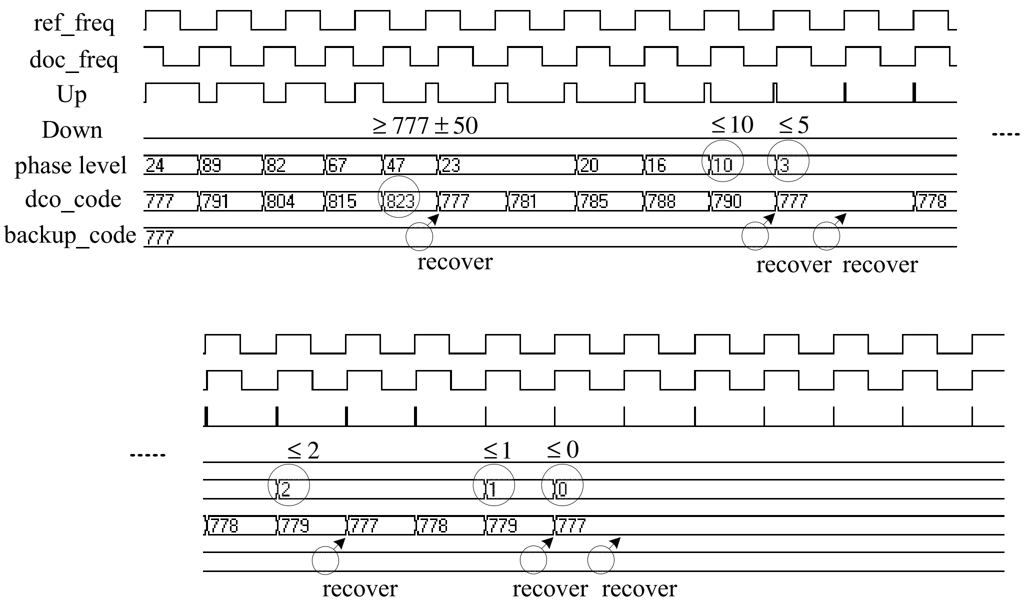


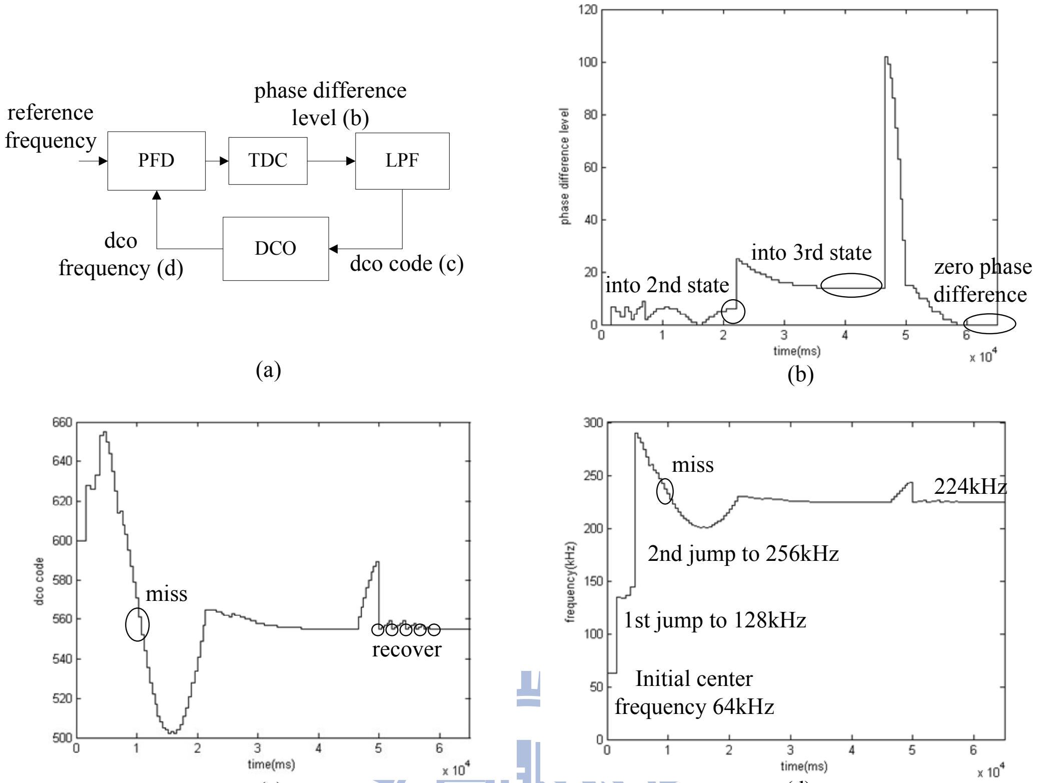
Figure 3.11 The process of fixing phase

completed.

■ Summary of Three States

Actually, the purpose of the three states is quite similar. The differences between them are only the resolution of phase and the LPF. In the acquisition state, the LPF gain is large to raise the bandwidth because we have to determine the approximated location of the reference frequency within the wide capture range. In the tracking state, we replace the phase difference with the difference of phase deviation; it can better express the difference between the reference frequency and DCO frequency. Finally in the phase-fixing state, we use the most prudent LPF and store the DCO frequency got from the tracking state.

The complete process is shown in Figure 3.12, where Figure 3.12b - 3.12d represent the phase difference, the DCO code and the DCO frequency, respectively. In the beginning, it spends some DCO periods to find the right band; the phenomenon appears in Figure 3.12d. Next, the DCO frequency starts to sweep the band until there is the same phase difference three times in a row. According to Figure 3.12c and 3.12d, we find it misses when the DCO



(a) (b)

(c) (d)

Figure 3.12 A lock process example

frequency first passes the reference frequency in this example. After entering into the tracking state, the scale of phase difference level is changed and the movement of DCO code becomes more sluggish. Then, in the phase-fixing state, we can adjust the DCO code indulgently because we store the right code already. Finally, the reference signal is locked.

3.4 The Noise Problem

Above discussion is made under the noiseless environment, where we only resolve the wide capture range problem. Next, we consider the reference signal which is sampled from the noisy environment.

3.4.1 The Noise Effect

The three-state PFD itself is unable to resist noise. Because the PLL has the resolution

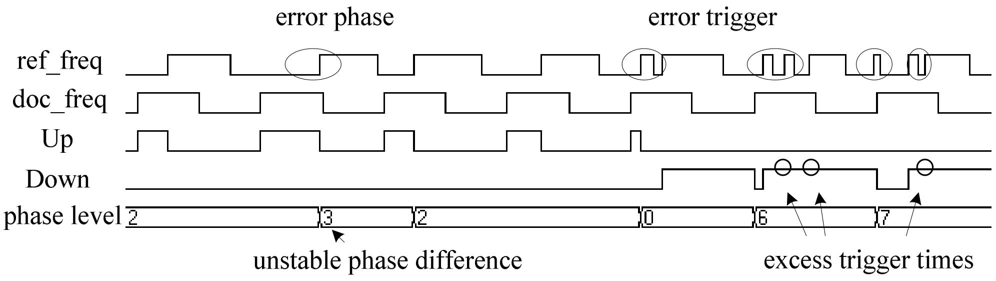


Figure 3.13 The error phase and error trigger

down to one degree, there are at least 128 points which may be erroneous in a period. In our PFD, the error will lead to erroneous phase or trigger point. As shown in Figure 3.13, the error phase causes the phase difference unstable and we can handle it with LPF. But, the error trigger not only makes the irregular *Up/Down* output but also produces excess trigger times, which may result in incorrect band judging from the trigger times. Therefore, we would rather have more error phases than error triggers.

3.4.2 The Conversion of Noise

Before discussing how we handle the noise problem, we analyze these noises first. In our plan, the PLL have to work in the noisy environment whose signal to noise ratio is down to 0dB. Therefore, the following discussion will be focused on the worst case.

The signal could be classified into the analog and the digital. If the signal is digital, we get the binary 1 or -1. The noise, which is zero-mean Gaussian noise, will be added to every point of the digital signal, and it is assumed to be independent between any two points. Because the signal to noise ratio is 0dB, the noise power is equal to the signal power and the noise variance is 1. The probability of error is Q(1) = 15.9% and each point is the same as shown in Figure 3.14a. Otherwise, if the signal is analog, the noise will be added to the sinewave directly as shown in Figure 3.14b. The error probability of each point would be different. In the peak positions, the error probability is about $Q(\sqrt{2}) = 3.97\%$, while in the zero-crossing positions, the error probability will be near 50%.

Therefore, if there is a ‘change’ between successive sampling points, which means that the value varying from 1 to -1 or -1 to 1, it is about 15.9% caused by noise and 84.1% by data transition for digital signals. Thus, to determine whether the change is caused by noise, we can further observe the next point. If another change appears again, we are certain that the point is affected by noise and then repair it. Otherwise, it is considered as a data transition as shown in Figure 3.15.

The above idea is based on the necessity of phase resolution. Because we have 128 points in one signal period, it is impossible to have two consecutive changes theoretically. However, we also have to take the possibility of two consecutive errors into account, even if the probability is low (15.9%^2 = 2.53%). As there are 128 points in a period, it is about 128*2.53% = 9 error points which can’t be removed by the method.

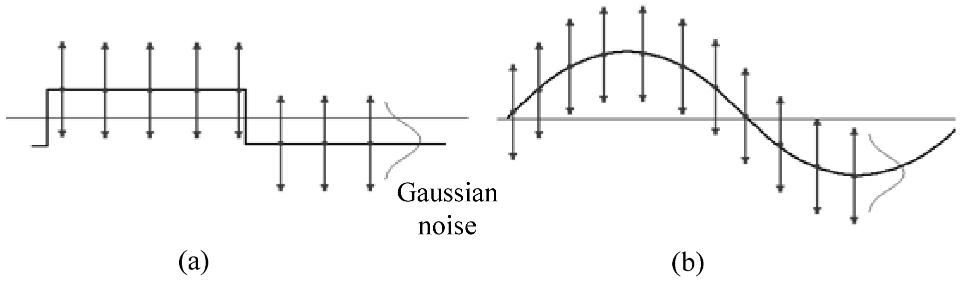


Figure 3.14 The noise effect for different signals

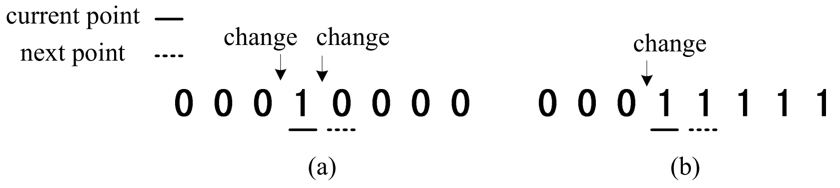


Figure 3.15 (a) Adjacent changes appeared (b) the separate change

As a result, we have to further consider the multiple consecutive errors. Two approaches are proposed here. The first approach uses the technique of cascade. After one error point is repaired, we resample the signal once again every two points. As shown in Figure 3.16, two consecutive errors will become one error, and then we can use the method above to repair the single error. Similarly, we can increase the cascade state if there are more consecutive error points. The second approach is to observe the following two points instead of the next point. That means the points will be repaired if the changes appear twice as shown in Figure 3.17.

For the two approaches, we find the performance of the second approach is better than the first one as shown in Figure 3.18. The reason is because the first approach uses the cascade technique, so that this state could handle the error points which can't be repaired by the previous state. But it can't judge whether the decisions of previous state are correct or not, as illustrated in as Figure 3.19a. Hence on the case of multiple consecutive errors, the first

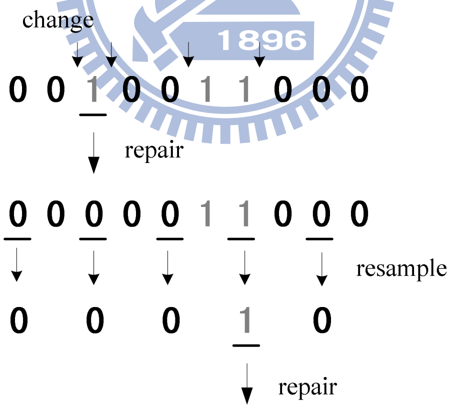


Figure 3.16 The cascade approach to repair consecutive points

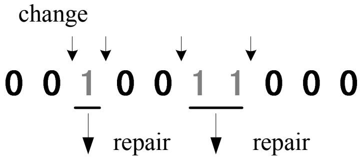


Figure 3.17 Observing more points to repair consecutive points

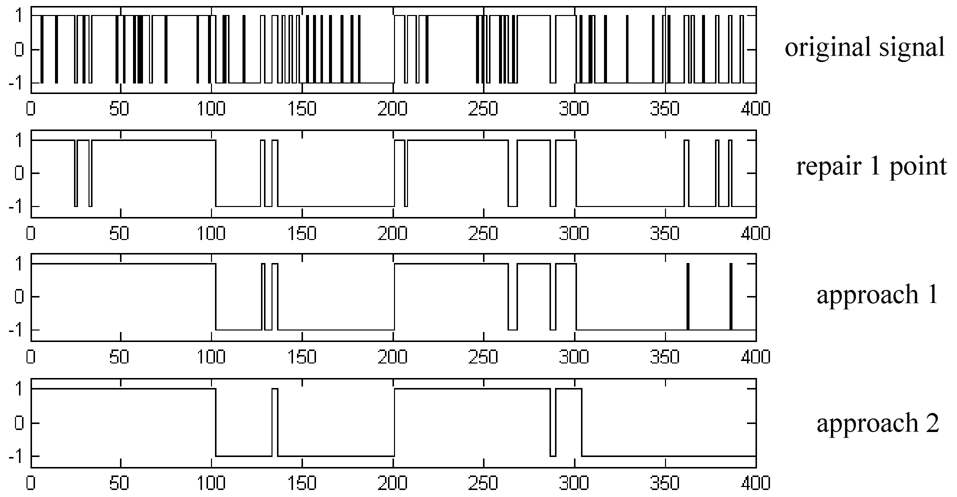


Figure 3.18 Comparison between two approaches of repairing two consecutive points

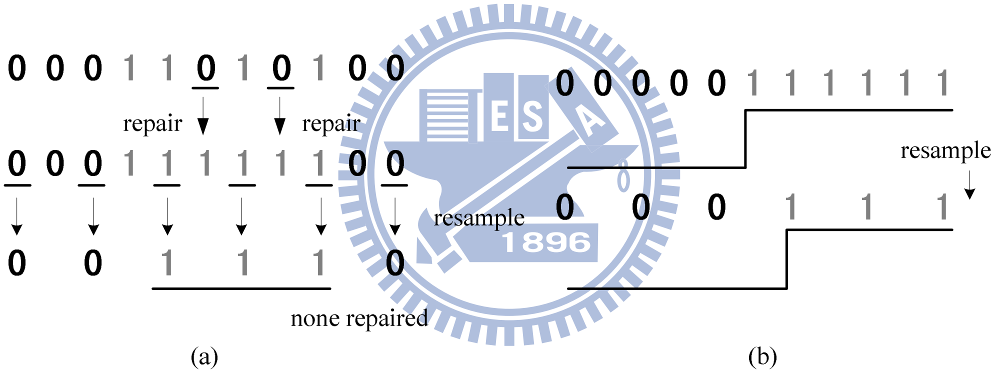


Figure 3.19 (a) Wrong repairment with correct points (b) phase resolution problem

approach is not as good as the second. Moreover, the first approach has to down sample the signal, thereby the resolution will be affected in data transition positions (Figure 3.19b). As a result, the second approach is taken.

Because our signal has wide frequency range, there are over 128 points in a period when the signal frequency is low. We can't guarantee that more than two consecutive errors may occur. Thus, based on the second approach described above, we observe the following three points, even five points. Basically, observing more following points can correct more consecutive errors, but it also causes more wrong repairments if the data transition is coming

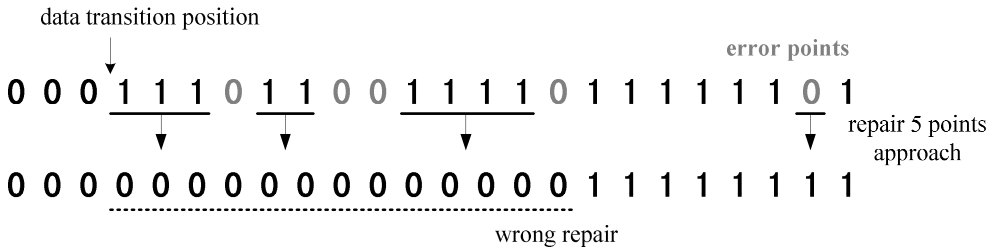


Figure 3.20 Wrong repairment with multiple points approach

as shown in Figure 3.20.

Figure 3.21 shows the performance of the noise removal. Except the error probability (Er), we can also calculate the positive trigger ratio between the original signal and the processed signal. Because our purpose is to reduce the trigger errors as much as possible, we can accept a larger trigger ratio even if the error probability is lower.

The two parameters between the number of observation points and the frequencies are shown in Figure 3.22. In high frequencies, the number of points are less in a period, thus fewer phase errors and lower error probability are obtained. But when the frequencies are low, the advantage of repairing more points appears. Besides, repairing more points lead to

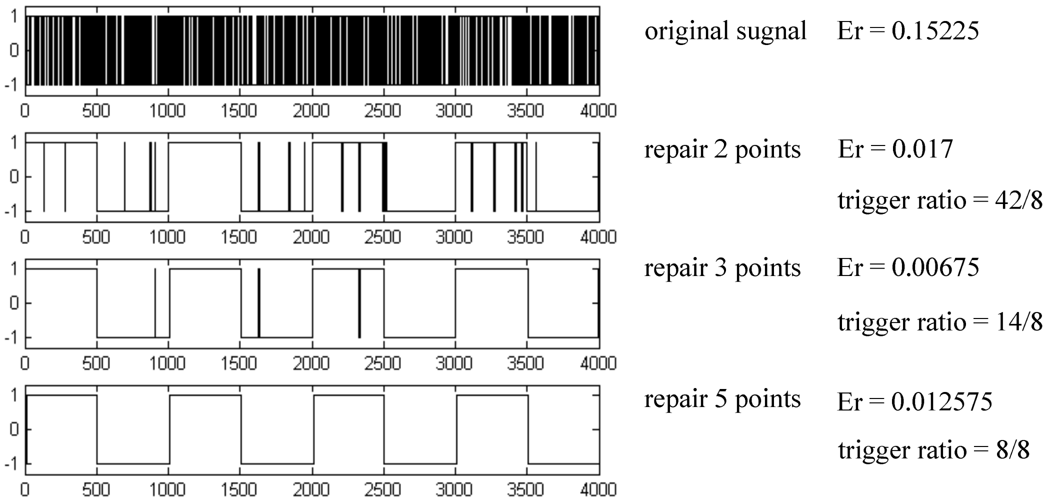


Figure 3.21 Comparison between repairing multiple points

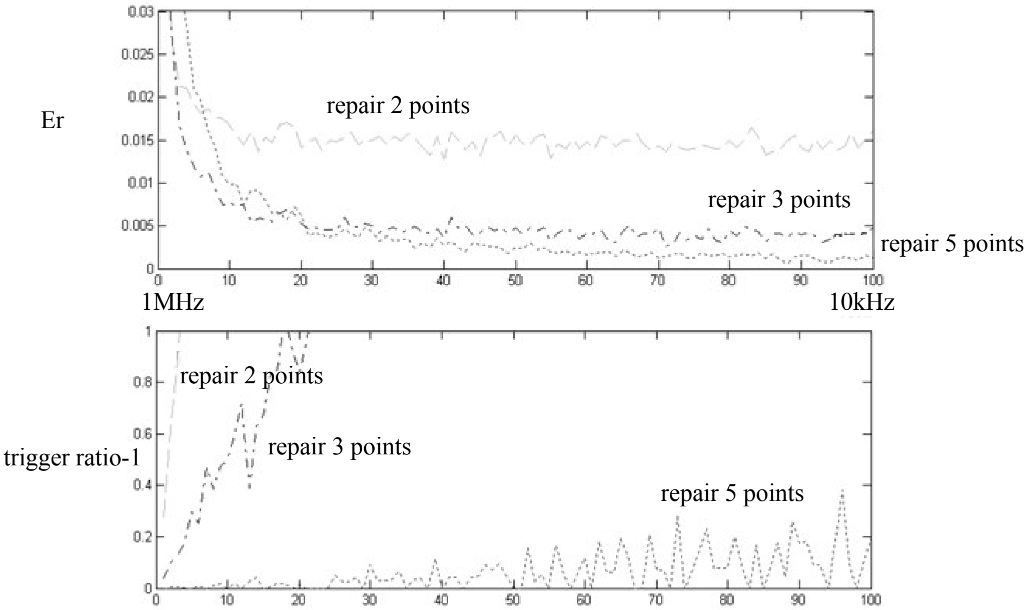


Figure 3.22 The error probability and trigger ratio for repairing different points

excellent trigger ratio.

Nevertheless, this method still has the defects. From Figure 3.22, the trigger ratios of repairing five points are not enough when the frequencies are very low. But we can’t increase the observation points unlimitedly because it will bring wrong repairment in data transition positions. Furthermore, the discussions above all focus on the digital transmission. If the

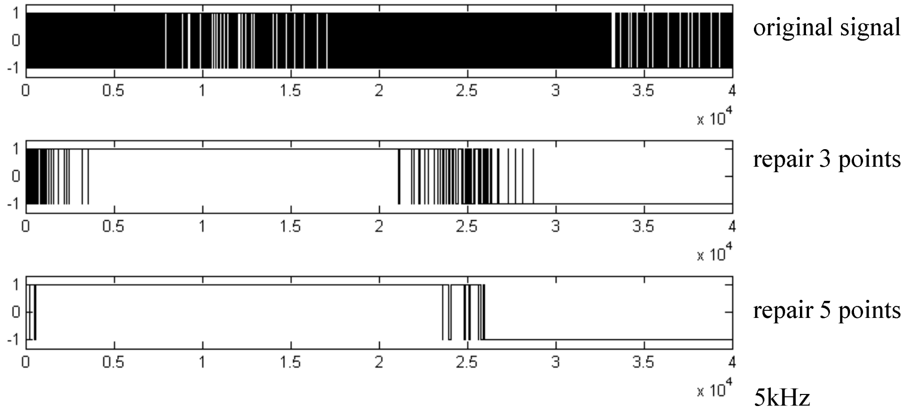


Figure 3.23 The weak repair capability in 5kHz reference frequency

signal is analog, the repairment will be more difficult in the zero-crossing positions, as shown in Figure 3.23.

Therefore our requirements are that the errors could be removed better in low frequency signals, and don’t make many wrong repairments caused only by fewer error points. Accordingly, we set a new condition which is easy to achieve in the zero-crossing positions but difficult when there is no data transition.

The concept is similar to the accumulator. If a change occurs, the accumulator is accumulated once but we hold the value as the previous point; and the accumulator is discarded if there is no more change. Only when the accumulator is over a threshold, we change the data now. In this way, the number of changes should be less than the threshold without data transmission and accumulator is difficult to cumulate. After data transition, the request of change will happen continuously. Then we change the data and reset the accumulator. Figure 3.24 shows the comparison between repairing five points approach and the accumulator approach. The new approach has the better performance in the lowest frequency. In high frequency, although the error phases still exists, they are more stable than repair five points approach.

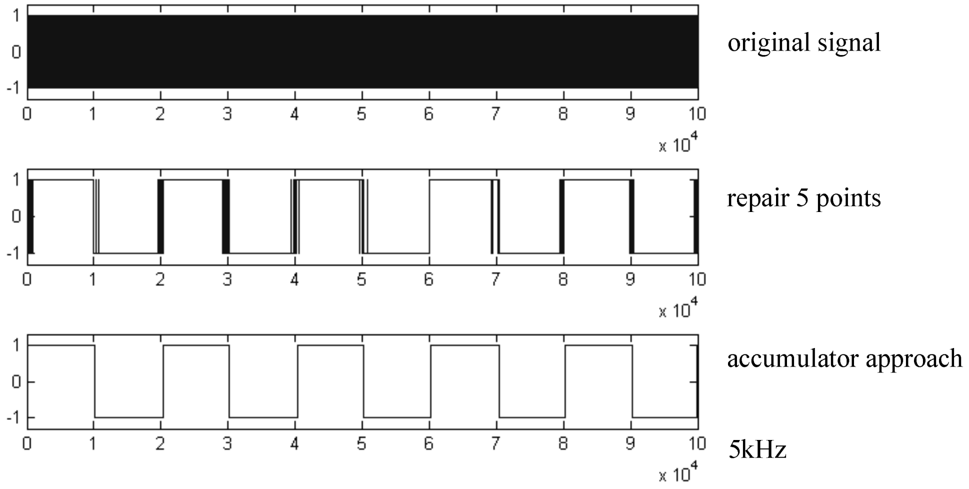


Figure 3.24 The performance comparison between repairing 5 points and accumulator approach

The main purpose of the approaches discussed above is to reduce the error triggers as much as possible. After noise treatment process, the error triggers almost disappeared. Next we consider what the error phases affect the performance of the three states.

3.4.3 Improvement in Acquisition State

Before dealing with error phases, we have to understand what the impact caused by errors. Theoretically, the error phases has bigger affect for high frequency signal, therefore we will make the phase statistics in several periods under 0dB signal to noise ratio.

In the acquisition state, because there is no error trigger after noise conversion, we still can determine the correct band from the trigger times. Next, we sweep to this band, and classify the phase difference into 8 levels. The phase resolution is only eight, which means the error phases are difficult to influence the classification. Figure 3.25 displays the noise effect when the phase differences are the same. In the high frequency, only about 8% is affected by error phases, not to mention in other frequencies. Table 3.3 also shows the frequency difference ratio $R_{\Delta f}$ and the number of reference periods N_{Tref} which are required to spend by the end of acquisition state with different SNR. From this table the noise effect is not

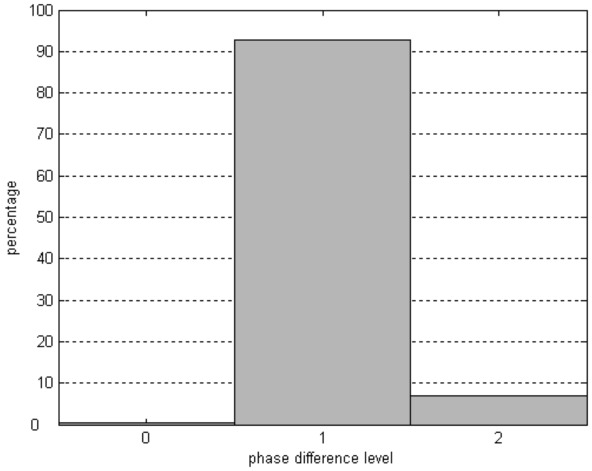


Figure 3.25 The phase difference detection with SNR=0dB in the acquisition state

Table 3.3 The N_{Tref} and $R_{\Delta f}$ by the end of acquisition state with different noise environments

parameters	noiseless	SNR = 20dB	SNR = 10dB	SNR = 0dB
N_{Tref}	23.2464	23.2661	23.3440	23.4735
$R_{\Delta f}$	1.90%	1.88%	1.89%	2.02%

significant, so that we don't have to improve it in the acquisition state.

3.4.4 Improvement in Tracking State

Similarly, Figure 3.26 shows the phase difference influenced by error phase. Because the phase resolution is higher per cycle in the tracking state, there is only about 36% phase difference which can be correctly detected, and we can't ignore the percentage of wrong phase anymore. Figure 3.27a and 3.27b show the phase difference and the DCO code, respectively. Because the phase difference can't be stable when the DCO frequency is close to the reference frequency, the DCO code begins to vibrate. Therefore, in order to overcome the varied phase difference, we could average the phase difference to reduce the influence by caused noise. To ease the design, we take two points, four points, and eight points to average.

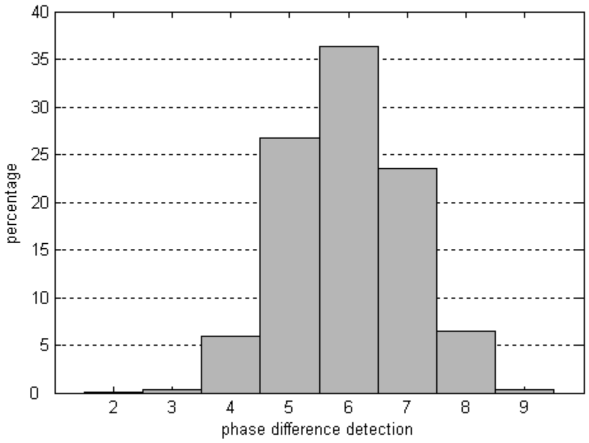
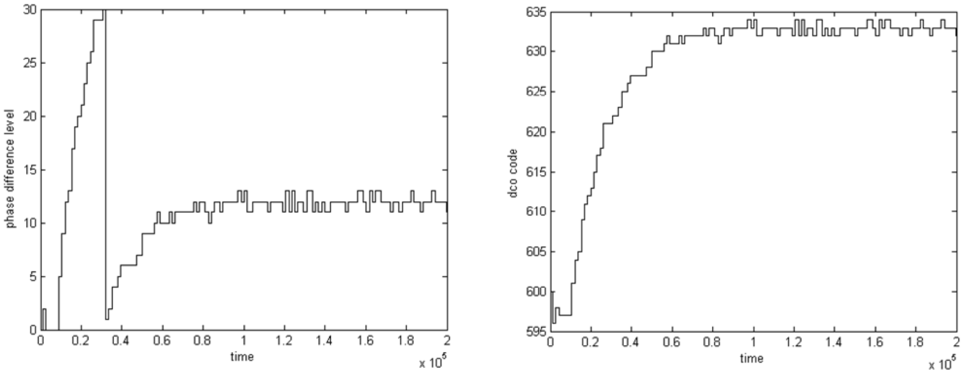
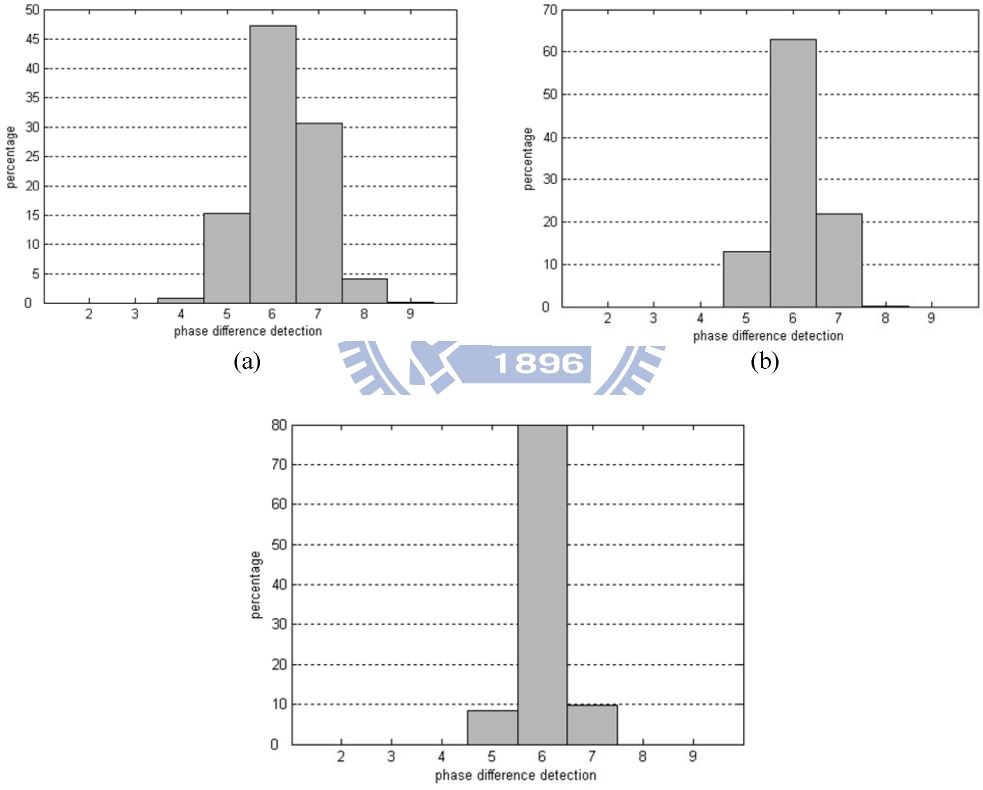


Figure 3.26 The phase difference detection with SNR=0dB in the tracking state



(a) (b)

Figure 3.27 The phase difference leads to the oscillating DCO code



(a) (b)

(c)

Figure 3.28 The phase difference detection with averaging (a) two points (b) four points (c) eight points

Figure 3.28 shows the detection after averaging different number of points. The results are not good enough in two-point and four-point cases, but the eight-point cases has about 80% correct rate which is acceptable.

The phase resolution is 32 in the tracking state. Although we can’t ignore the error phases, we can use fewer points to reduce the impact. However, the noise effect may not be so easy to deal with in the phase-fixing state.

3.4.5 Improvement in Phase-Fixing State

Let’s look at the phase difference detection shown in Figure 3.29. The variance is too large, thereby we have to accumulate a lot of points to reduce the errors. This is unrealistic. However, the phase differences are just the reference in this state, so that we could not too concern about the accuracy of phase differences.

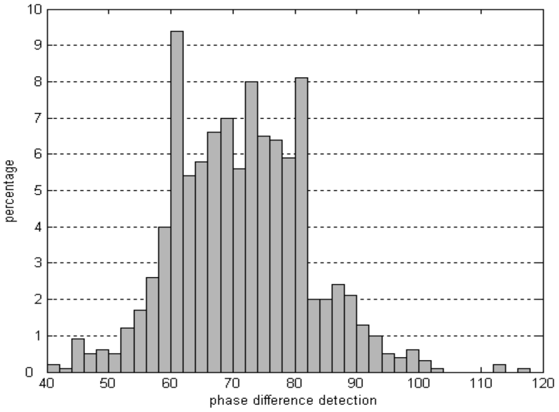
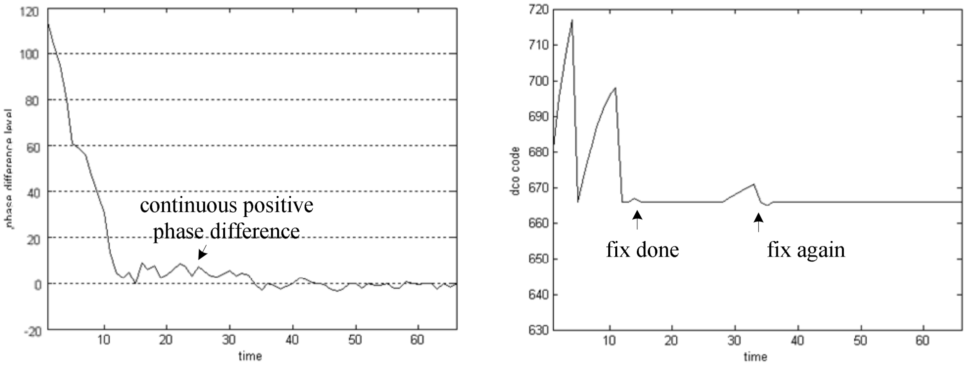


Figure 3.29 The phase difference detection with SNR=0dB in phase-fixing state



(a)

(b)

Figure 3.30 The accumulator technique against noise effect in phase-fixing state

We borrow the idea of the accumulator from the approach in repairing noise. When the phase is locked for the first time, the accumulator starts to operate. If the *Up* or *Down* is coming and asks the DCO to shift phase, the DCO code will be hold and the accumulator is increased or decreased according to which the *Up* or *Down* is. Until the accumulation is over a threshold, the DCO starts to adjust its output. As shown in Figure 3.30, the threshold we set here is 10 DCO periods. In this case the system locks the wrong phase first as shown in Figure 3.30a, and in the next 10 rounds the positive phase differences are sent continuously. Afterwards, the DCO repairs phase again as shown in Figure 3.30b, this time the positive and negative phase differences appear averagely; the accumulator can't keep accumulating. We can except that these phase differences are caused by noise.

3.5 Summary

This chapter shows how the PLL works in two conditions: with the wide capture range and under the serious noise environment. Now we summarize the four challenges involved:

Challenge 1:

How can the PFD detect the correct phase difference from the ultra-wide frequency range?

We use the reference frequency to sample the DCO signal and get the trigger times. The trigger times reveals the difference between two frequencies.

Challenge 2:

How can the PLL determine if the triggers are caused by noise or by signal?

We repair the noise and convert error triggers into error phases, therefore almost all triggers are caused by data transition after our process.

Challenge 3:

How can the DCO get higher resolution under limited bits?

We add a Divider for the DCO. The Divider can control which frequency bands the DCO has to oscillate. The ratio between the lowest frequency and the highest frequency is two, so that all the bands are integrated closely. Therefore, we can adjust the Divider parameters to choose one of the bands, and use the limited bits only in this band. That means even if the code bit is the same, different Divider parameters can generate different frequencies.

Challenge 4:

How can the LPF get the balance between the efficiency in large frequency difference and the stability in small frequency difference?

We set three states, and each state has the appropriate LPF according to different purposes. Therefore, the PLL will not be restricted to the specific frequency region.

From the above description, we have the basic understanding about the operation of the designed PLL. However, there are still some defects which can be improved. We will discuss a more sophisticated system in the next section.

3.6 The Dual-Loop PLL System

As mentioned above, the PLL has some defects which should be improved. For example, the reference frequency is assumed to be fixed in the previous discussion. Now if the reference frequency can drift or jump to other frequency, the system will spend much time to get the correct reference signal. In our previous design, the system in phase-fixing state would try to lock the phase of reference signal even if its frequency has been changed. When the system can’t get the stable phase difference, it returns to the tracking state to re-track the frequency and fix the phase again.

Thus, we can create two loops to resolve the above problem, where one loop deals with the short-time differences possibly caused by noise, while another loop deals with the long-time differences caused by frequency variation. When the first loop is out of control, the second can lock the new frequency almost immediately.

3.6.1 The Primary Loop and the Secondary Loop

Figure 3.31 shows the schematic diagram of the dual-loop system. In our design, when the reference frequency is locked by the end of tracking state, we make the primary loop get into the phase-fixing state. The work of primary loop is similar to that mentioned above; it keeps the DCO frequency and starts to fix the phase. On the other hand, the secondary loop keeps in the tracking state; it continues to monitor the reference frequency. When the frequency has varied, the secondary loop will observe whether the variation is temporary or not. If it's not, the system will record the new DCO frequency, and the DCO frequency of the primary loop will be modified. In this way, we can not only keep but also update the information about the reference signal, and the primary loop would not pursue the wrong frequency blindly.

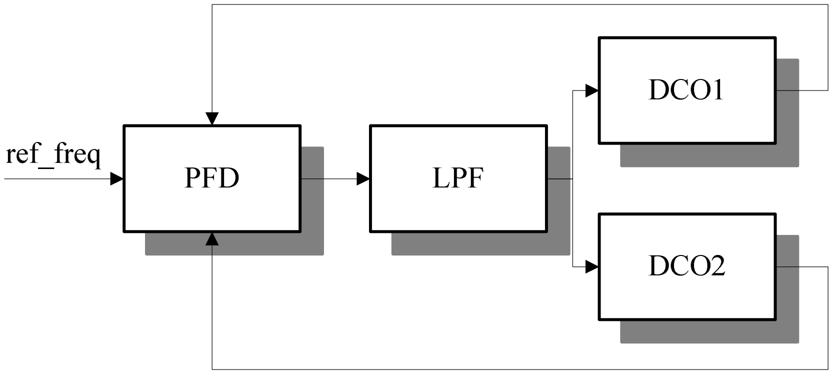
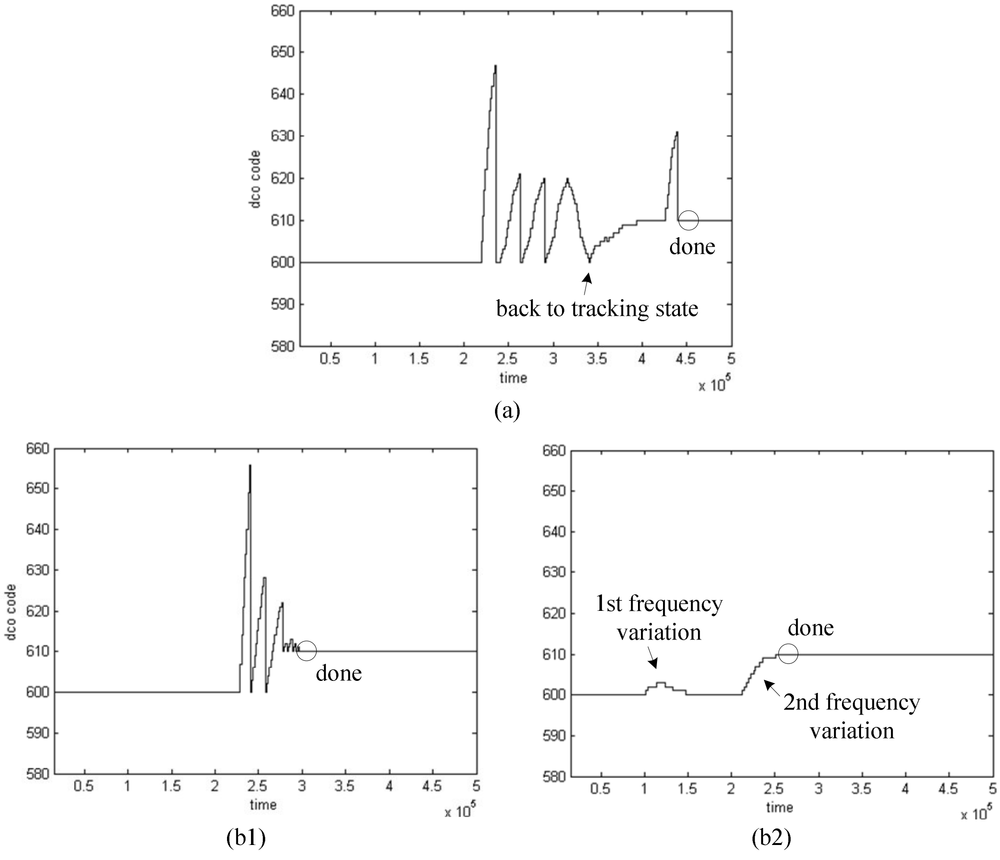


Figure 3.31 The block diagram of dual-loop system

Figure 3.32 shows an example between the original PLL and the dual-loop PLL, where Figure 3.32a is the DCO code of the original PLL, Figure 3.32b1 is the primary DCO code of dual-loop PLL, and Figure 3.32b2 is the secondary. When the first frequency variation is coming, the secondary DCO code (Figure 3.2b2) has a slight vibration. But it is too short, and the original and primary DCO don't be affected. Then, when the second frequency variation comes, the secondary DCO tracks the new frequency at about 2.5 unit times. Because of the aid of the secondary loop, the primary loop also grasps the new frequency immediately (about 3 unit times). Otherwise, the original system has to spend about 4.5 unit times to lock the new frequency.



(a)

(b1)

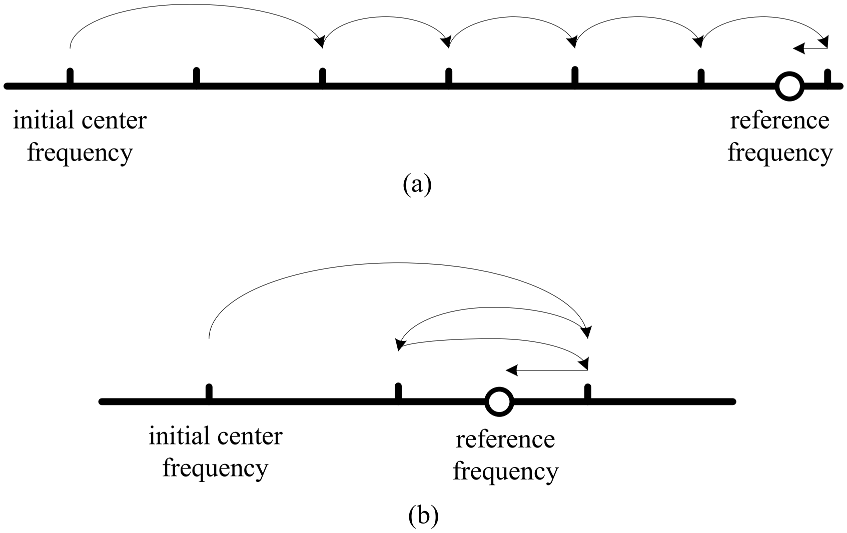
(b2)

Figure 3.32 Frequency variation phenomenon on DCO code of (a) original system (b1) the primary loop of dual system (b2) the secondary loop of dual system

Besides, we find that the original system has to spend 20 rounds to ensure the DCO frequency which is stable by the end of tracking state. Also, before the phase-fixing state it needs several rounds to fix the phase, too. That means we can decrease the time with the dual-loop. We let the system get into the phase-fixing state earlier even if the DCO frequency of primary loop is not coincided with the reference frequency. After all, the time required of fixing phase is about the same between the similar frequencies. Moreover, the secondary loop can update the DCO frequency promptly. We will discuss the other improvement with the dual-loop PLL system in the next section.

3.6.2 Improvements of Dual-Loop System

Since there are two loops in the system, we should use the extra loop extensively. In the original acquisition state, the DCO frequency is located in the initial center frequency, and would jump to the other center frequency according to the trigger times rule. Sometimes the hopping process is not so smooth because of the initial phase or some reference frequencies whose location would cause wrong judgment. As the examples shown in Figure 3.33, because the first frequency hopping is too conservative (Figure 3.33a) or the reference frequency is



(a)

(b)

Figure 3.33 Two inefficient processes on selecting appropriate center frequency

located in the awkward position (Figure 3.33b), the DCO frequency doesn't make the adequate decision.

Now we have two loops, which means we can set two initial center frequencies. Also, we classify the center frequencies as the odd or even position. The one loop only jumps to the odd center frequency, and another only jumps to even. At this time we don't distinguish which loop is primary or secondary; both loops acquire the reference frequency in fair competition. In this way, we can use two DCO frequencies to squeeze the reference frequency such as Figure 3.34a. Even if this frequency is located in the awkward position and may cause the wrong judgments for one of the DCO frequencies, another DCO frequency can deal with the frequency better as shown in Figure 3.34b. The other inefficient situation is the conservative frequency hopping. However, in our design there are eight center frequencies, so that each

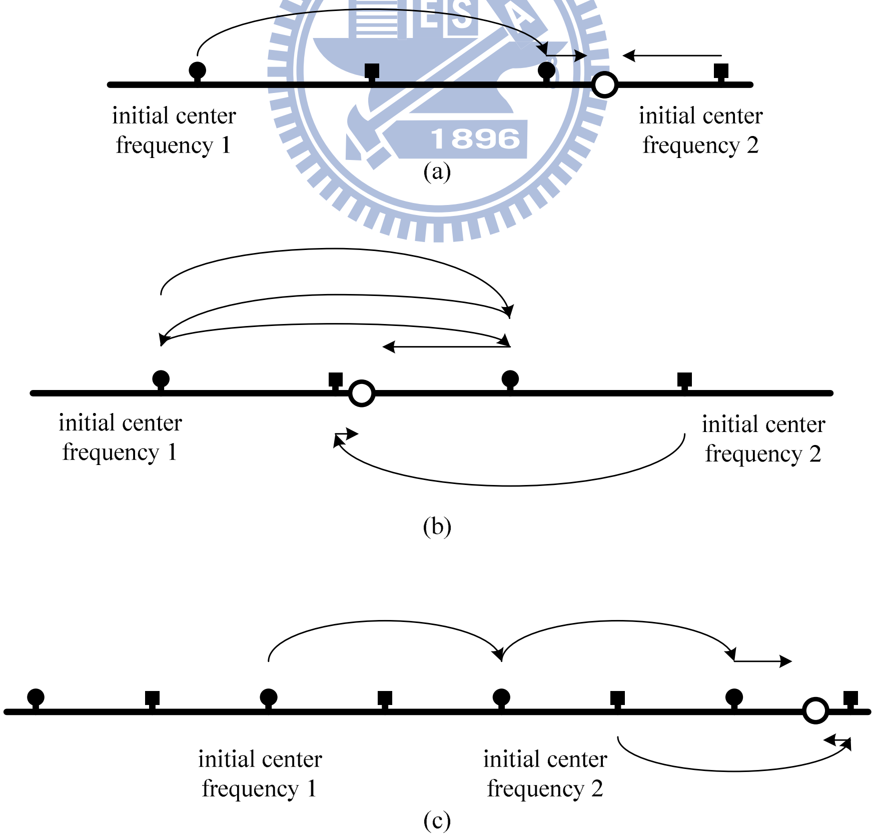

initial center frequency 1
initial center frequency 2
(a)
initial center frequency 1
initial center frequency 2
(b)
initial center frequency 1
initial center frequency 2
(c)

Figure 3.34 The efficient hopping process of dual center frequencies

loop is only responsible for four of which. The DCO frequency jumps at most twice no matter what the reference frequency is located (Figure 3.34c). Therefore, although the operations of two loops are seemingly independent, they work on the complementary bands and make the efficient decision.

When one loop gets the same phase difference three times in a row, both loops enter into the tracking state. The loop which achieves the condition can release the information about its DCO band and DCO code to another. This makes both loops with the same DCO frequency. Because of the noise, one loop will start to accumulate the phase differences for the average, while another loop doesn't use the noise treatment. It means the two loops are divided into the conservative one and the aggressive one, respectively. We can not make the unnecessary statistics if there is little noise influence; as a matter of fact the loop with the noise treatment can also get better performance when the environment is noisy. Next, the two loops will track the reference frequency according to the noise policy themselves.

As mentioned in Section 3.6.1, the loop which locks the reference frequency first in the tracking state will be the primary loop. The primary loop will release its information to the other which will be the secondary loop. The secondary loop maintains in the tracking state to monitor the reference frequency; it plays the aggressive role to update the DCO code timely when the reference frequency has varied.

Chapter 4

System Implementation by Digital Hardware

In this chapter, we will implement the dual-loop PLL system with the Hardware Description Language (HDL). Here we use the Verilog language to simulate the all-digital circuit.

4.1 System Module

According to the algorithms mentioned in Chapter 3, the designed architecture is constructed as shown in Figure 4.1. Because of the dual-loop system, there are two feedback loops which use the *State Control Unit* to communicate with each other. The *State Control*

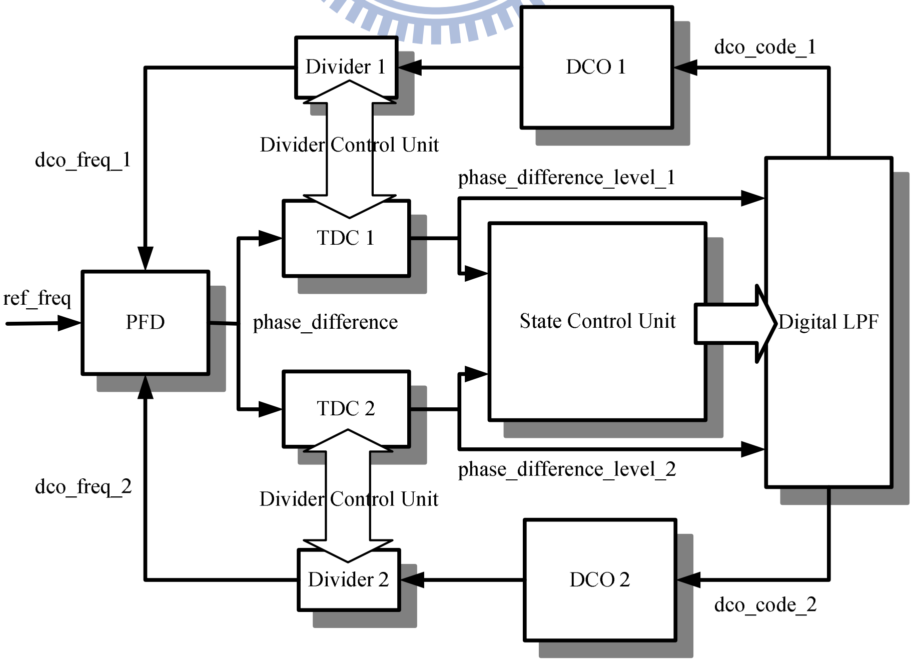


Figure 4.1 The architecture of the dual-loop PLL system

Unit is originated from the traditional control unit, and another control unit is the *Divider Control Unit* which is only responsible for controlling the frequency bands of two loops. The *PFD*, *TDC*, and *Digital LPF* are used for phase detection, phase difference decode and integrator, respectively. The diagram only displays the main data flow as shown by black arrows; there are many control wires which are hidden between the modules. Next, we describe these modules in details.

4.1.1 Phase Frequency Detector

We introduce the structure of *PFD* first. Different from the traditional phase detector, the *PFD* has the noise treatment in front of the system and it can get more information of trigger times behind. Figure 4.2 shows the noise treatment structure, where we use *reg_1* to sample the reference signal *ref* (*t*) and employ *reg_2* to delay this signal, where *clk* is the system clock and is set it to be 100 MHz. The *change* value represents whether the change occurs and it will control the sign of the accumulator below. The *reg_3* plays the role of accumulator to count the *change* times and compare with the threshold; if it is over the threshold, *reg_4* will reverse its value. Then the repaired value will be sent to two sub-modules *pfd*. Each *pfd* is

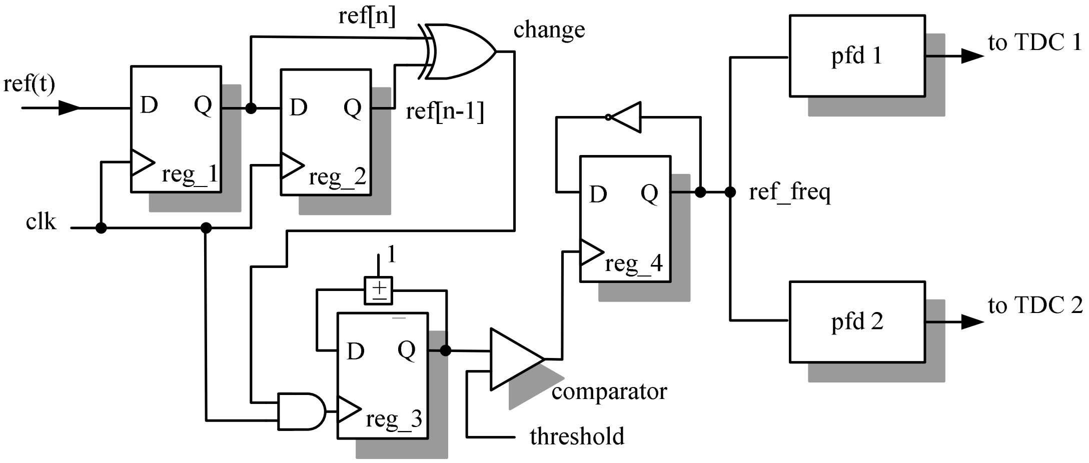


Figure 4.2 The noise treatment of PFD

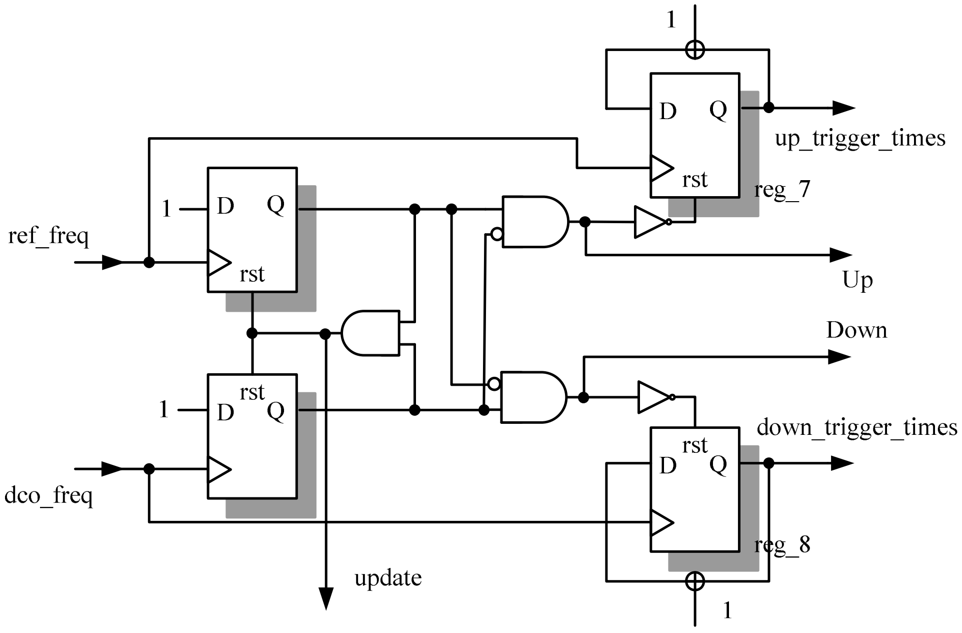


Figure 4.3 The phase difference detection of PFD

belonging to one loop system.

The *pfd* structure is shown in Figure 4.3, which adds two registers *reg_7* and *reg_8* additionally for counting the trigger times. The invert of *Up*/*Down* is acting as reset, it means that the count process works only when *Up*/*Down* is high. Also for the *update* signal, it can be treated as a unit on phase difference output, and we take it as an output specially. Finally, there are five outputs sent to the other modules.

4.1.2 Time to Digital Converter

The *TDC* structure is similar to that described in Section 2.3.2, which uses two *Counters* to count the phase difference signal *Up*/*Down*. The difference is that there are three frequency values which could be chosen as the reference clock of *Counters* here because of the different phase resolution requirements in different states. Therefore, when the phase difference signal *Up*/*Down* has counted, the digital value *count_lv* will replace the length of *Up*/*Down* to next module as shown in Figure 4.4. Moreover, for declaration of current status which is lead or lag, we output the *up_falg* and *down_flag* to show the *count_lv*.

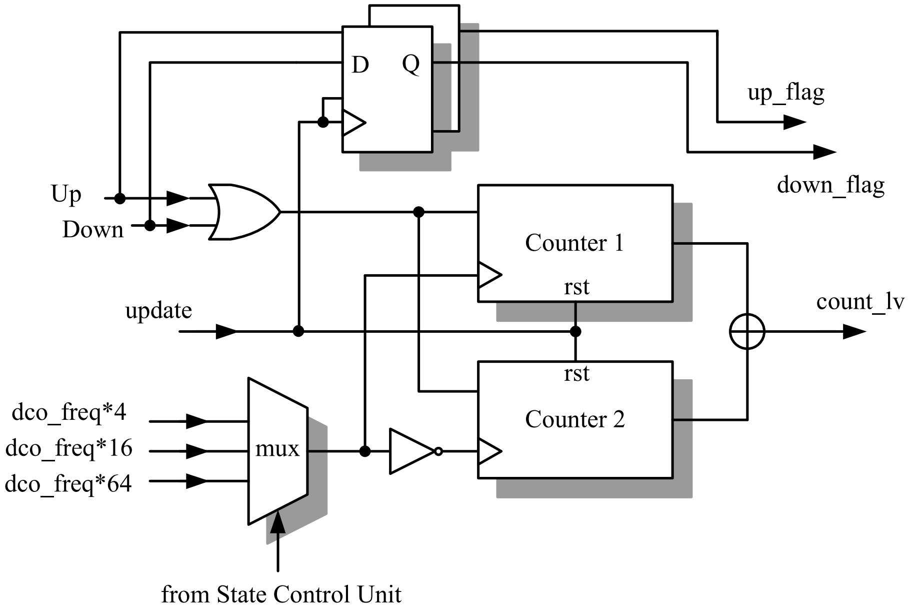


Figure 4.4 The structure of Time to Digital Converter

4.1.3 Digital Loop Filter

Because of the dual-loop and several states, it needs many types of *Integrator*. As shown in Figure 4.5, when two digital values *count_lv_1*/*count_lv_2* enter, the *guide unit* can guide them to the appropriate *Integrator* according to the current state and inform the *Integrator* to increase or decrease. Then, the *Mux*, which is also guided by the *guide unit*, will choose the corresponding *dco_code* and send them to *DCO*. There are five *Integrators* shown in Figure 4.6 respectively, where the structure of *Integrator 1* and *Integrator 2* are the same, except their default values are different because of the different initial frequencies; both of them are used in the acquisition state. The difference between *Integrator 3* and *Integrator 4* is that one of them has the function of noise treatment but another does not; both of them are used in the tracking state. Also, the *Integrator 5* is responsible for fixing phase in the phase-fixing state.

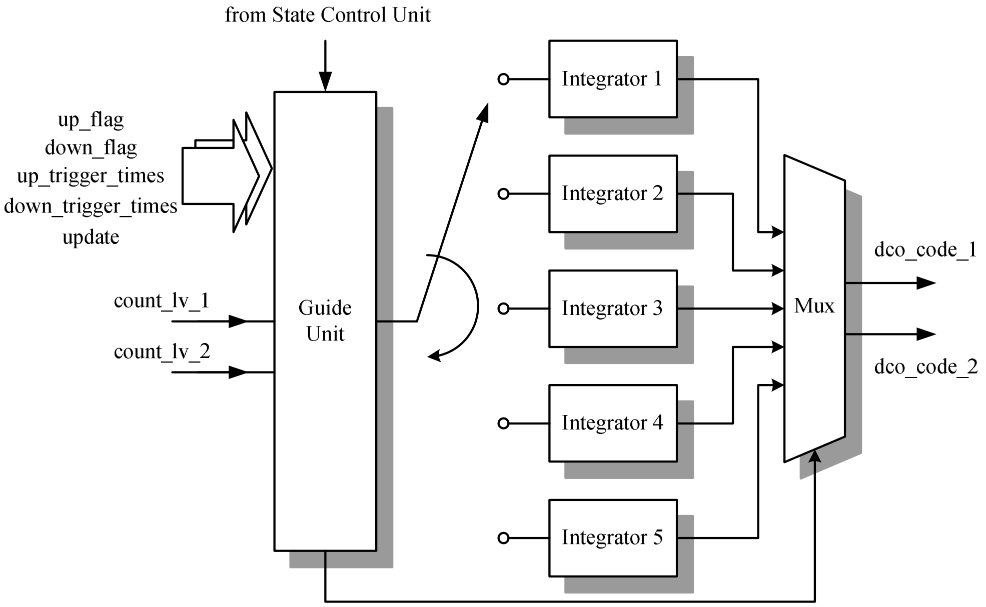


Figure 4.5 The overview of Digital Loop Filter structure

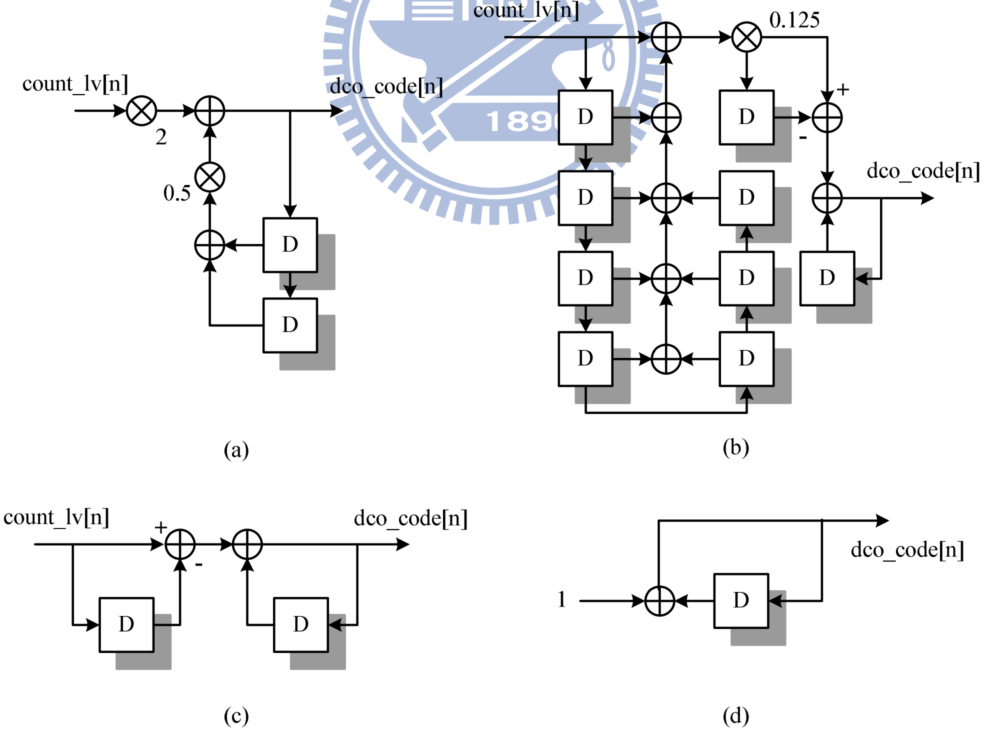


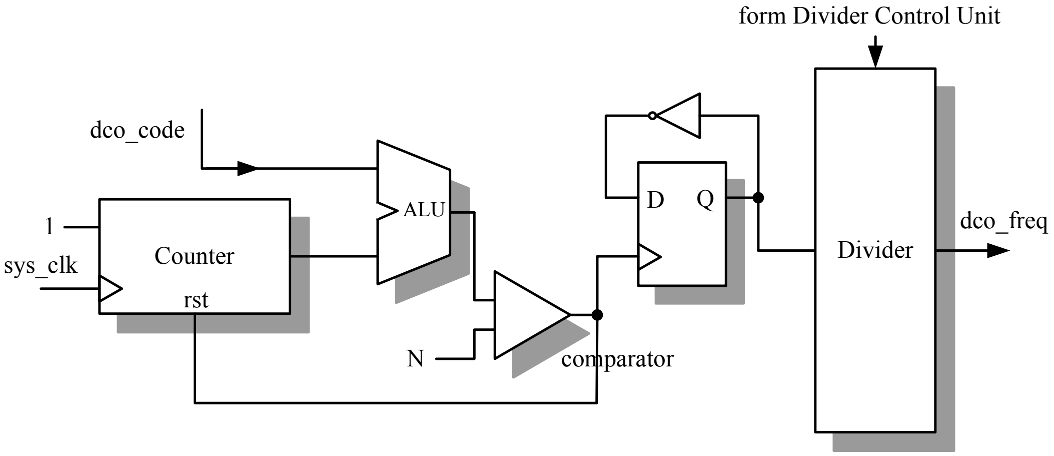
Figure 4.6 The structure of several integrators (a) Integrator 1 and Integrator 2 (b) Integrator 3

(c) Integrator 4 (d) Integrator 5

However, Figure 4.5 is only the schematic diagram. It doesn’t need so many components at the same time in the real system. The *guide unit* can control the connection between all the registers and decide the data flow according to the various intelligences in practice.

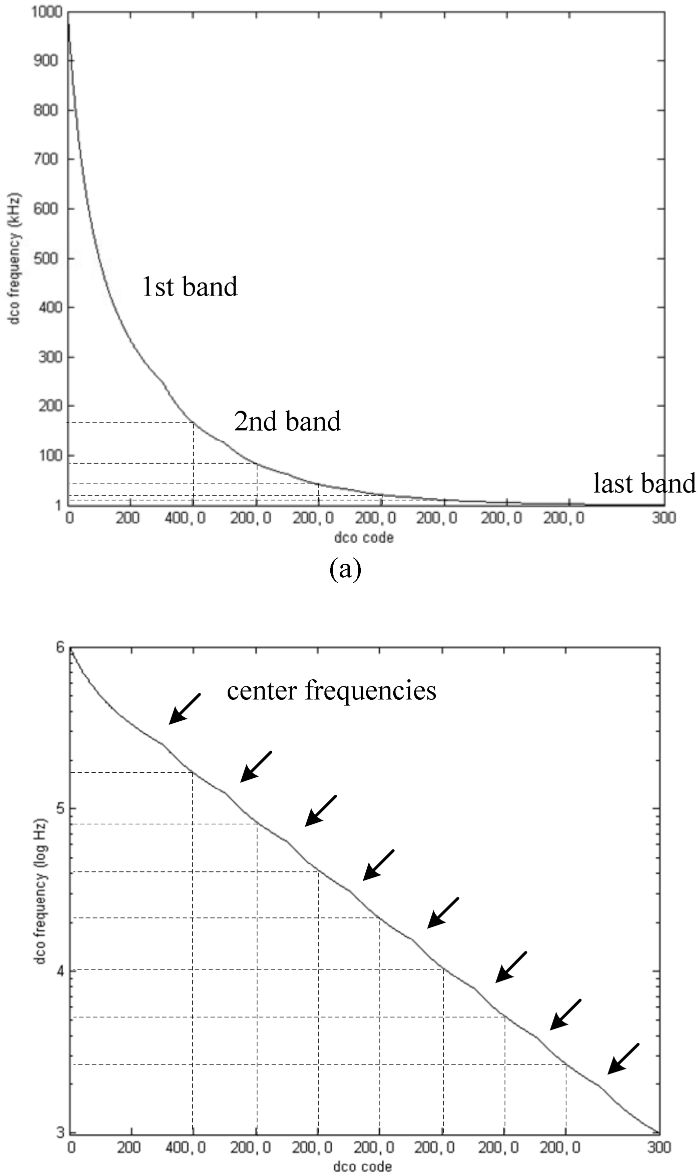
4.1.4 Digital-Controlled Oscillator

Because our design doesn’t consider the effect of propagation delay, we can’t really implement an oscillator such as ring oscillator in this circumstance. Therefore, we use a *Counter* with the system clock to get similar performance of DCO as shown in Figure 4.7. The system clock is 100MHz, and we can invert the DCO value when the *Counter* counts exceed a number *N*. That means our DCO periods are 2*N* times of system clock. For example, the DCO can output the 1MHz frequency if *N* is 50. The ratio between the lowest frequency and the highest frequency of each frequency band is two, thereby the range of *N* are between 50 and 100. Moreover, there is a *Divider* which is controlled in the end for choosing the different frequency bands. Also, with different *Divider* parameters, the relation between the input *dco_code* and output *dco_freq* of DCO is shown in Figure 4.8a. Except the first band and the last band need 400 and 300 codes, respectively, the others all use 200 codes so that they only need 8-bits on *dco_code*. Because the oscillation frequency range is wide, we make



Fi

gure 4.7 The structure of Digital-Controlled Oscillator



(a)

(b)

Figure 4.8 The DCO characteristics on (a) linear scale (b) logarithmic scale

the frequency on logarithmic scale. As shown in Figure5.8b, the curve becomes more linear, and the irregularity points are exactly at the center frequency positions. The disadvantage of this DCO design is that we have to prepare a high frequency *sys_clk*, and therefore the DCO can't output the higher frequency with high resolution**.**

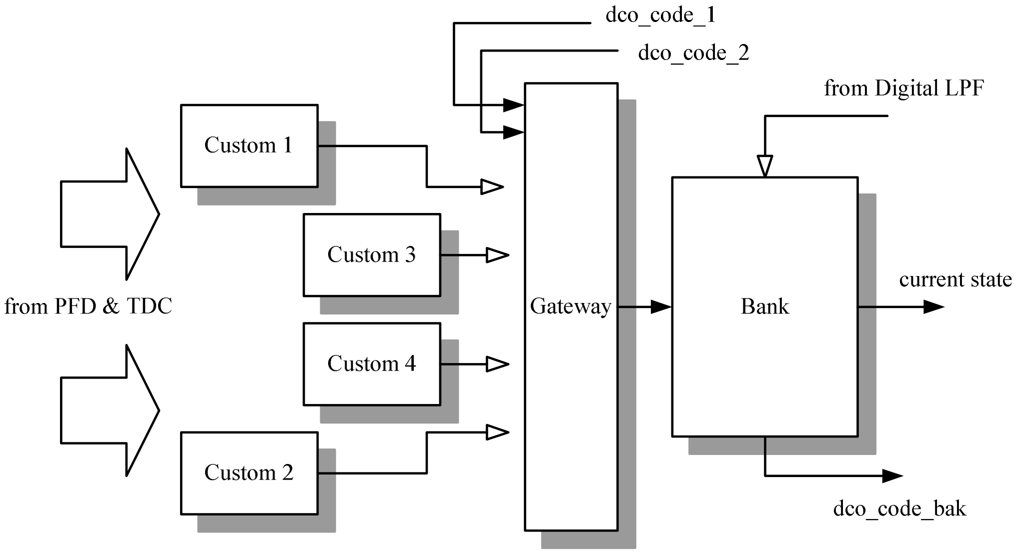


Figure 4.9 The overview of State Control Unit structure

4.1.5 State Control Unit

The *State Control Unit* is the core of the dual-loop system; it collects all helpful information from all modules and determines which the current state should be. The overview is shown in Figure 4.9. There are four *Customs* which check the information from the *PFD* and *TDC* to determine whether the required conditions have been reached. The *Custom 1* and *Custom 2* work in the acquisition state, while *Custom 3* and *Custom 4* work in the tracking state and phase-fixing state. The *dco_code* from the *LPF* can pass the *Gateway* only when the corresponding inputs meet the requirements of *Custom*. Then, this *dco_code* will be deposited to the *Bank* which can release the newest state to all sub-modules. The sub-modules will ask the *Bank* to send the *dco_code* when they need. Also, the *Customs* ensures that the *dco_code* in the *Bank* is the newest.

4.1.6 Divider Control Unit

The *Divider Control Unit* is also an important module. We design this module for reducing the complexity of *State Control Unit*. The *Divider Control Unit* connects to the *TDC* and two *Dividers* directly, which plays the role of a shortcut between the three modules. Its job is to decide the parameter of divider according to the accumulated trigger times, so we can use two *Accumulators* as the skeleton (Figure 4.10). There are two different initial divider parameters *initial_div_lv*. The divider parameter *div_lv* will be changed according to the trigger times rule as described Section 3.1. Furthermore, when the state has changed, the *State Control Unit* will determine which loop is the primary. Then, both output ports would output the primary *div_lv*.

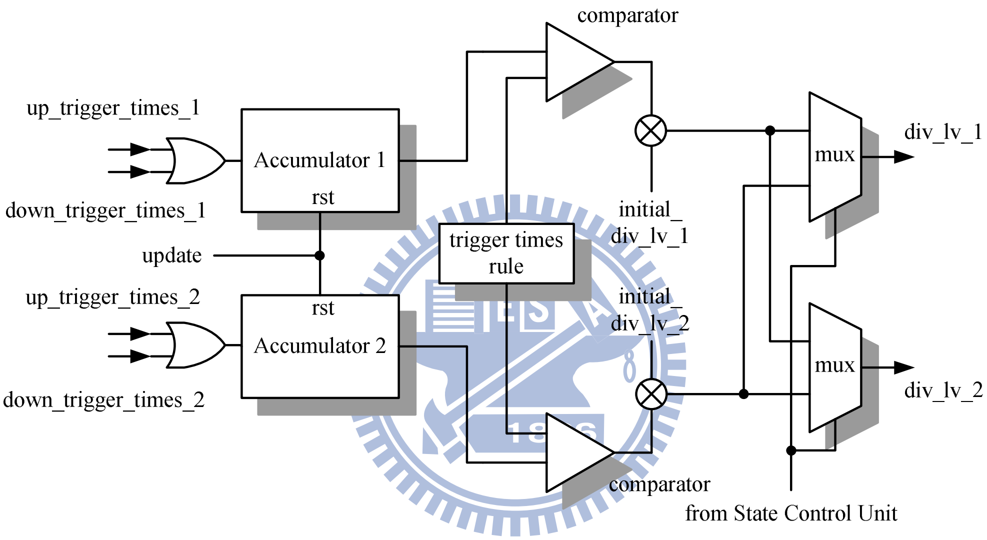


Figure 4.10 The overview of Divider Control Unit structure

Chapter 5

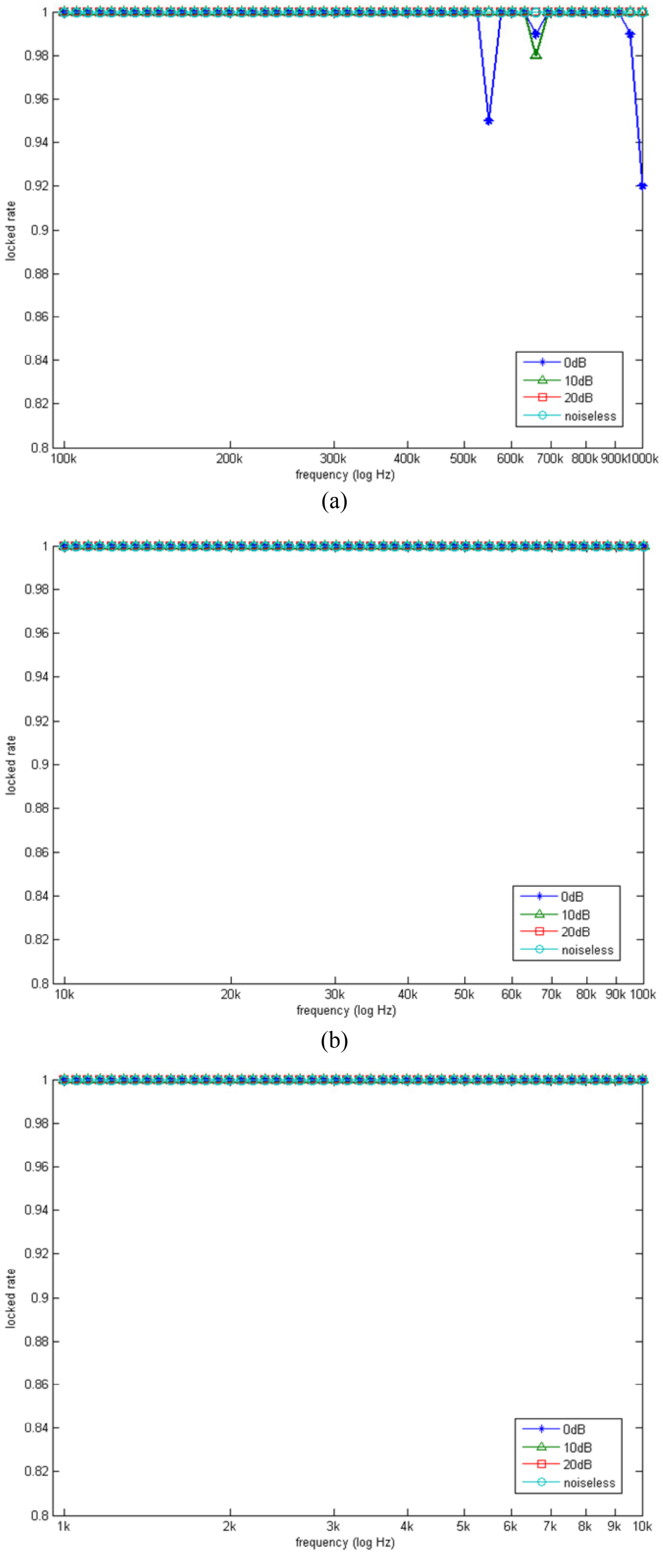
Simulation Results

After the introduction in previous Chapters, we can test our design now. For observing the system performance, there are three tests to be done: the frequency locked rate, the lock efficiency and the capability of anti-frequency drift.

5.1 Frequency Locked Rate

Here we test the locked rate first. Before the test, we define what the ‘lock’ is. We have two system models; one is the single-loop system, the other is the dual-loop system. Because of the DCO resolution, the reference frequencies are difficult to be produced by DCO exactly. Therefore, in the single-loop system if the DCO code varies within 1-bit, it means lock condition is reached. Also, for the dual-loop system, the lock condition is reached when the DCO has a stable output code and it is also the closest to the reference frequency.

We classify the total capture range into the high frequency band, the medium frequency band and the low frequency band. Each band has 50 frequencies on logarithmic scale, and every frequency we test has to lock 100 times. The locked rate is according to the statistical results. Finally, the test above is under SNR equals 0dB, 10dB, 20dB as well as noiseless condition. First, we show the locked rate of single-loop system and dual-loop system in Figure 5.1 and Figure 5.2, respectively, where horizontal axis and vertical axis represents different frequencies and locked rate.



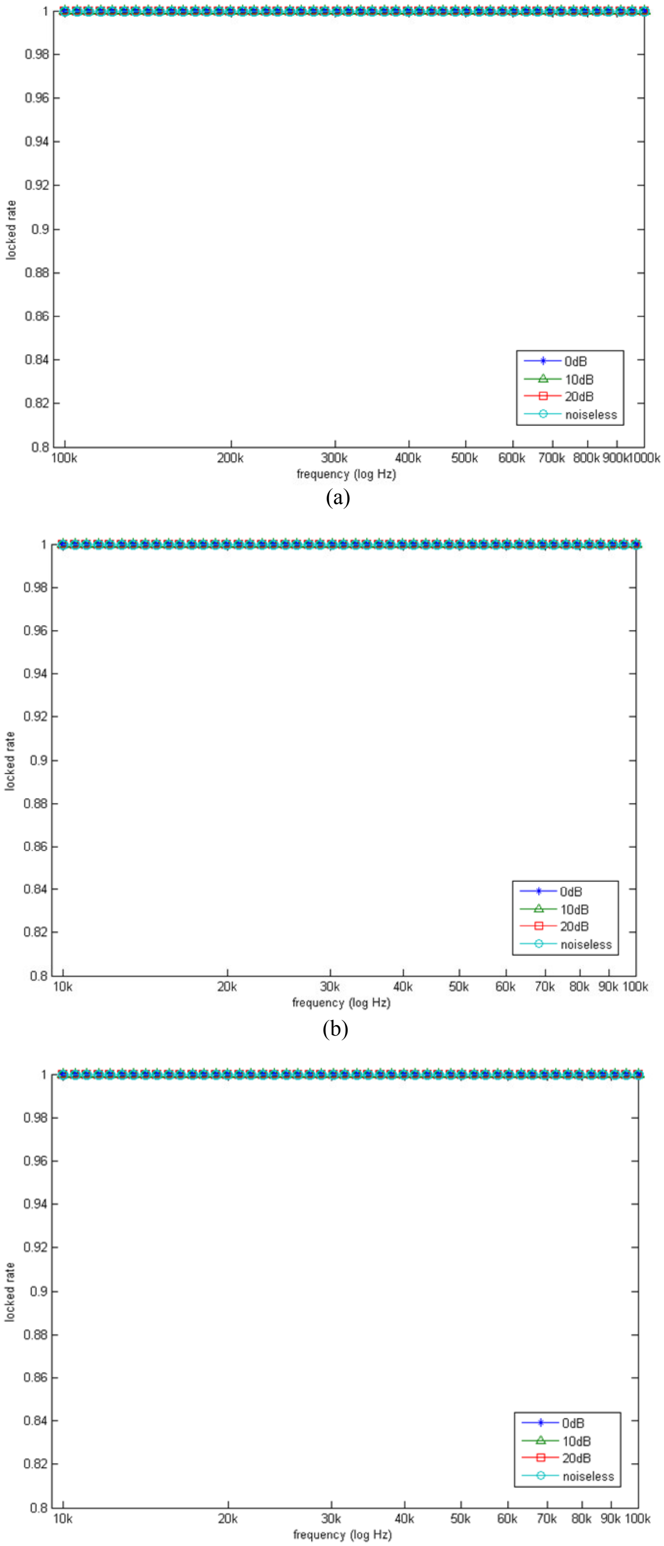
(a)

(b)

(c)

Figure 5.1 The locked rate of single-loop system in (a) 100kHz-1MHz (b) 10kHz-100kHz

(c) 1kHz-10kHz



(a)

(b)

(c)

Figure 5.2 The locked rate of dual-loop system in (a) 100kHz-1MHz (b) 10kHz-100kHz

(c) 1kHz-10kHz

As shown above, we find that the single-loop system has lower locked rate in high frequency range (Figure 5.1a) when SNR=0dB. The reason is because our noise treatment has more unstable error phases in high frequency. It is difficult to keep the DCO code within 2-bits in 20 rounds. On the other hand, the dual-loop system has two loops where the primary loop can get into the locked state early through the aid of the secondary loop. Nevertheless, the lowest locked rate of single-loop system is still more than 90%.

5.2 Lock Efficiency

Except the locked rate, we would like to know if the system locks the reference signal efficiently. We hope that it would not spend over 100 reference periods to lock the signal. Similar to the test above, there are three frequency bands and each band has 50 frequencies. Then, we record the reference periods required in three states. First, we show the required periods in the single-loop system about the acquisition state on medium frequency band in Figure 5.3.

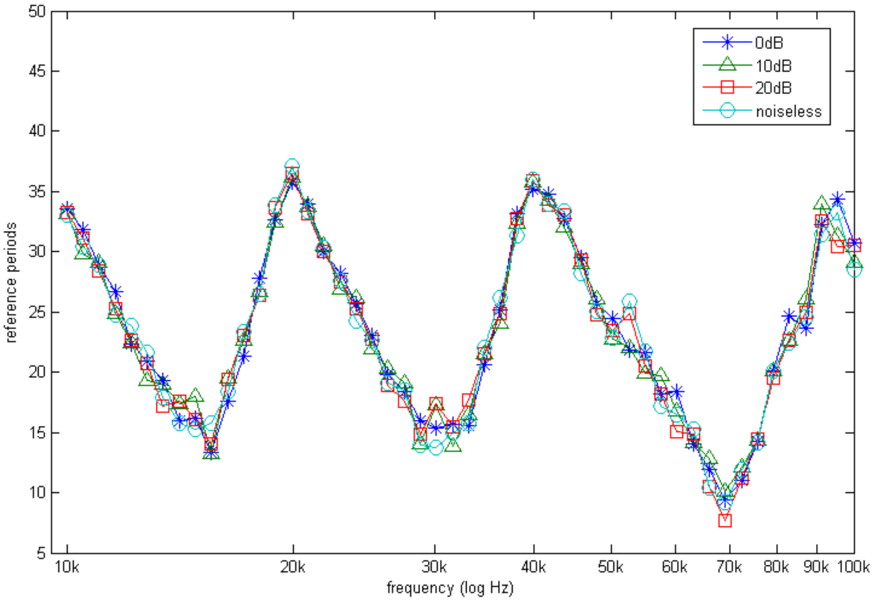


Figure 5.3 The reference periods required in the acquisition state for 10kHz-100kHz

We find in the acquisition state, it spends at most 35 reference periods. The three lower values correspond to the center frequency positions, and the lowest one (70kHz) is the initial center frequency. Meanwhile, the peak values correspond to boundaries of bands. As mentioned above, the noise effect in this state is not obvious.

Next, the required periods of tracking state on medium band is shown in Figure 5.4. According to the figure, the tracking state is similar to the acquisition state. The difference is in the 0dB case, where it needs more periods to be stable even if we have the noise treatment. Moreover, the periods of peak values are over 100 times of reference periods. Because the system only has one loop, it has to spend 20 periods to ensure if the DCO code is stable, which also makes the system inefficiently.

Finally, the required periods of the phase-fixing state on medium band is shown in Figure 5.5. This time the phase difference between the reference signal and the DCO signal is within one degree. We see the required periods are about 120 times on average. As in the tracking

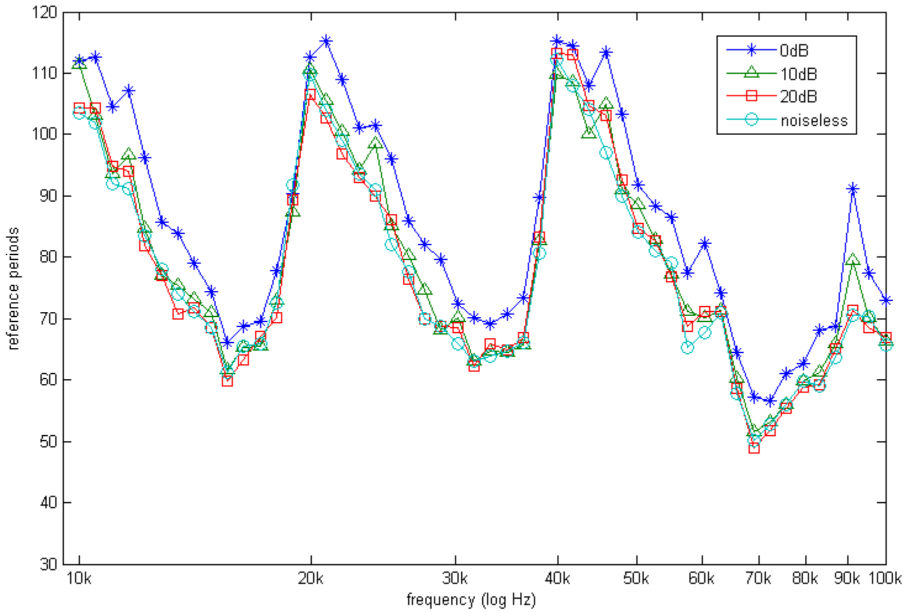


Figure 5.4 The reference periods required in the tracking state for 10kHz-100kHz

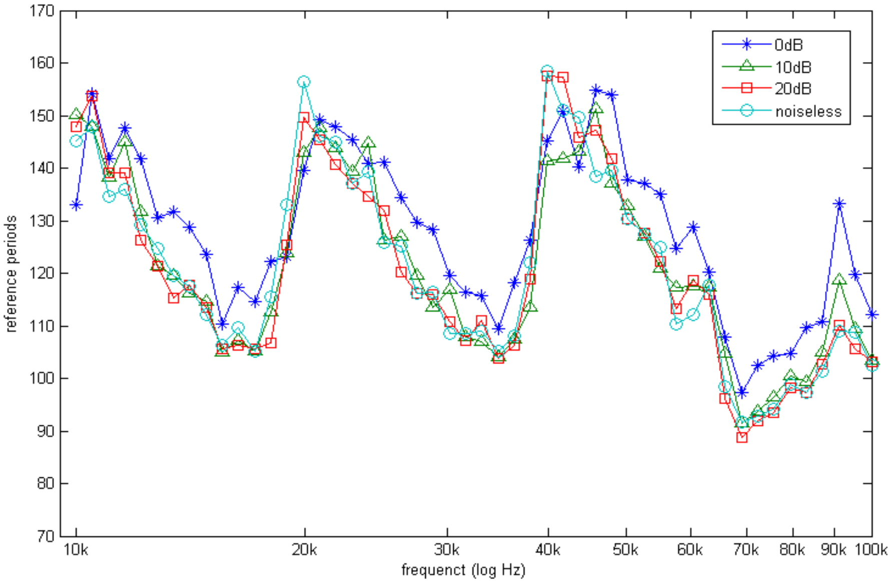


Figure 5.5 The reference periods required in the phase-fixing state for 10kHz-100kHz

state, the system operation is ended when the DCO code maintains stable for 20 periods, which indicates the reference signal is already locked 20 periods ago. Therefore, the required periods could be 20 periods less; that is about 100 times on average. Finally, we combine three figures above to be the one shown in Figure 5.6.

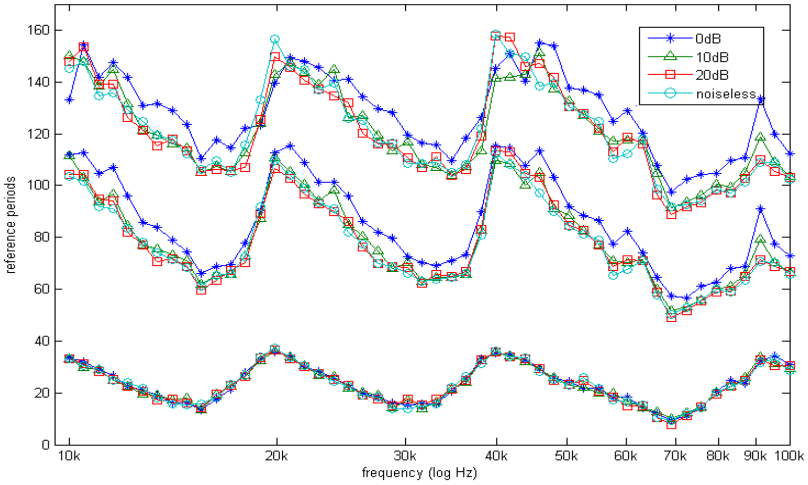


Figure 5.6 The total reference periods required for 10kHz-100kHz

The case above is only in the single-loop system on medium frequency band. Next, following the above example, we test the required periods in different states, frequency bands and noise environments. Then, we create a table to show the average results.

In Table 5.1, we find the medium band has the higher lock efficiency in all three states and four noisy environments. It's because the initial center frequency is at this band, it has the best chance to find the correct frequency. For low and high bands, the low band has a little bit better performance. The possible reason is that our initial center frequency is on the medium band, therefore the initial search is too slow for high frequency band; that causes the higher reference frequencies which already pass several periods when the DCO signal comes. Accordingly, the efficiency of high band is underestimated in reality.

Table 5.1 The reference periods required for different cases in single-loop system

case	state	noiseless	20dB	10dB	0dB
low band	acquisition	24.0617	24.1733	24.0997	24.2694
	tracking	82.3379	83.0270	83.5010	90.0949
	phase-fixing	126.3348	125.6256	124.8970	131.6212
medium band	acquisition	23.2414	23.2661	23.3440	23.4735
	tracking	77.7058	78.1094	79.3357	85.8383
	phase-fixing	121.2004	120.7704	121.3748	128.2798
high band	acquisition	41.8680	41.5788	41.9869	45.0450
	tracking	82.3339	93.1491	92.2814	249.4807
	phase-fixing	121.3868	131.8606	132.0467	292.3325

However, the high band in 0dB case also has poor performance, except the worse locked rate. That is also caused by the unstable error phases. Meanwhile, the required periods in phase-fixing state have to subtract 20 periods; that is the actual values as mentioned above. Next, we do the same test again, but for the dual-loop system. The results are shown in Table 5.2.

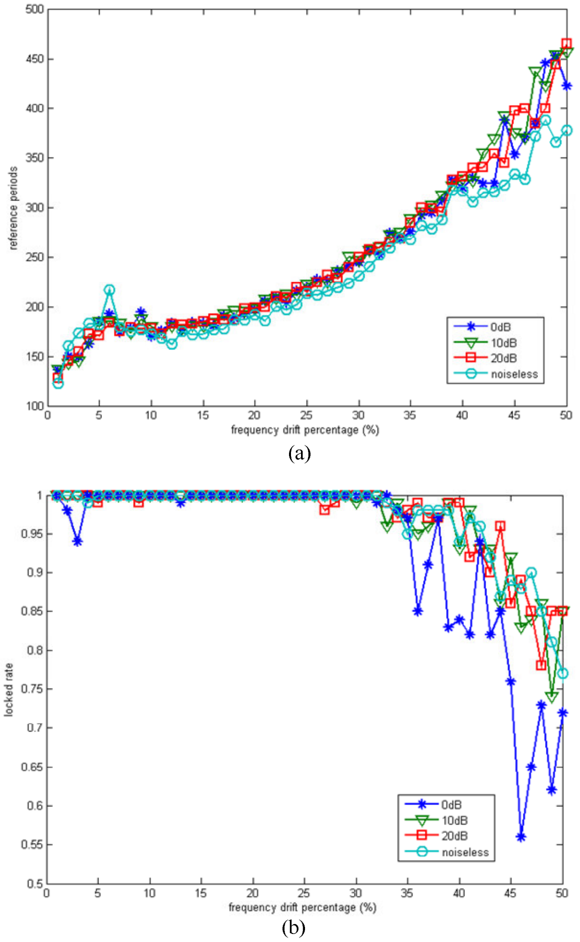
In contrast to the single-loop system, the main difference is in the tracking state; it is almost 30 periods less than before. Because it doesn’t have to spend 20 periods to check the DCO code, there is only 5 periods in the dual-loop system. Additionally, the efficiency of acquisition state also has slight improvement because of two initial center frequencies. Look at the high band in 0dB, the efficiency becomes acceptable. After subtracting 20 periods, it needs only 96 periods to lock the signal on average.

Table 5.2 The reference periods required for different cases in dual-loop system

case	state	noiseless	20dB	10dB	0dB
low band	acquisition	21.0366	21.2461	21.3104	22.8718
	tracking	43.9120	44.4101	44.9630	51.5468
	phase-fixing	93.7995	93.5239	91.2890	96.7068
medium band	acquisition	20.5533	20.3783	20.4445	20.6216
	tracking	38.7852	38.9062	39.4938	42.1621
	phase-fixing	85.2390	85.9696	87.2434	91.0743
high band	acquisition	40.1961	40.3841	40.4935	42.0138
	tracking	56.4642	57.3090	58.9631	65.7152
	phase-fixing	98.7414	99.6795	102.1688	116.0835

5.3 Capability of Anti-Frequency Drift

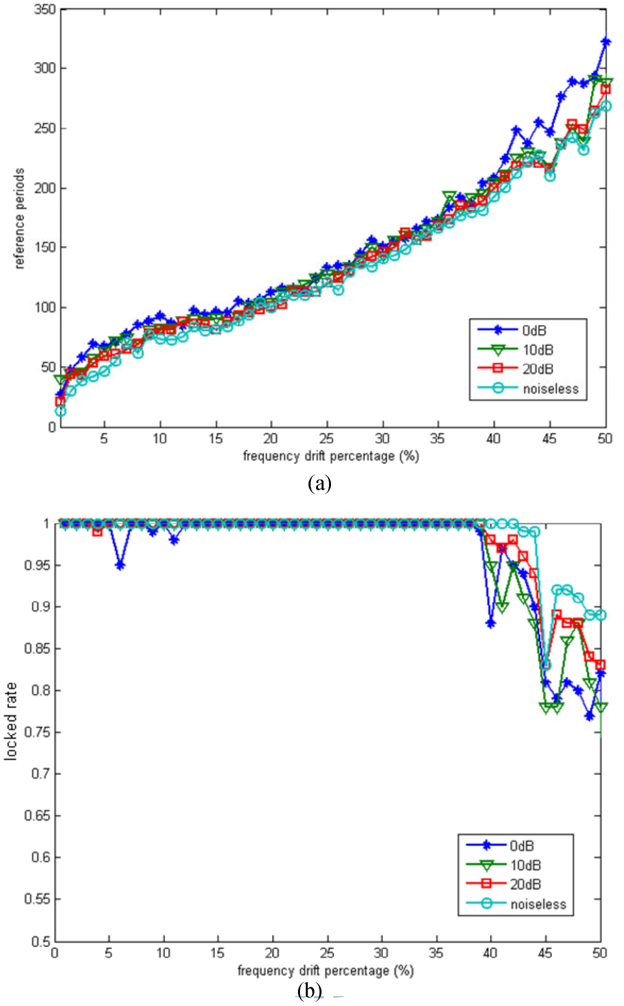
The major improvement of dual-loop system is the capability of anti-frequency drift. In the following test we make the reference signal has the frequency drift between 1% and 50%, and record the required periods and the locked rate. The initial state of this test is in the phase-fixing state, and it is tested under four different SNR environments. The statistical results of single-loop system and dual-loop system are shown in Figure 5.7 and Figure 5.8, respectively.



(a)

(b)

Figure 5.7 The (a) required periods and (b) locked rate for reference frequency drift in single-loop system



(a)

(b)

Figure 5.8 The (a) required periods and (b) locked rate for reference frequency drift in dual-loop system

As seen from Figure 5.7a and Figure 5.8a, the difference between two curves is about 100 reference periods on average, which means the dual-loop system can lock the drift frequency 100 periods ahead of the single-loop system. The time is even enough to let the system lock a new frequency again.

However, we find that it has to spend more than 100 periods to lock the reference signal in the dual-loop system when frequency drift is over 20% (Figure 5.8a). When it is up to 40%, the locked rate begins to decrease (Figure 5.8b) although it is not so serious as the single-loop system (Figure 5.7b). It is because the reference frequency drifts too fast. For our dual-loop system, the secondary loop is on the tracking state, which can't efficiently track the frequency locating in the other band. Therefore, the dual-loop system is more suitable for the case which has slow frequency drift. Perhaps we can add a new process: when the secondary loop can't track the reference signal, the primary loop will go back to the acquisition state actively and re-acquire the reference signal again.

Chapter 6
Conclusions

In this thesis, we present an ADPLL which has the wide capture range (1kHz-1MHz) and can work under serious noise environment (SNR=0dB). There are three different states in the ADPLL: the acquisition state, the tracking state and the phase-fixing state for dealing with frequency/phase estimation. Moreover, although we can’t eliminate the noise completely, we reduce its affect to some extend. For improving performance, we design a dual-loop system so as to provide effective assistance in every state. Combination of the dual-loop system and the three states also increases the diversity of the system that we can accommodate more ideas creatively due to the stored mechanism and the double frequencies which coexist at the same time.

Furthermore, we implement the system with the Hardware Description Language. It verifies the proposed system is feasible with digital circuitry. However, the system could still be improved in various aspects, and we divide the improvements into three levels. In the top level, we could improve the efficiency in terms of noise repaired or the phase resolution, where we can deal with the signal period for high resolution. We can think of other methods to use fewer points but get the similar performance. In the middle level, although the system has good performance in our design, we can’t guarantee that the system parameters are the optimum, e.g. the positions of DCO center frequency, the way of dividing frequencies, and the choice of LPF, etc. It’s possible to get better results with more study. In the bottom level, though we construct an all-digital circuit to implement the PLL system, there are many factors which are neglected. It is necessary to design a more rigorous pipeline system, consider the effects of propagation delay and save the configuration of components and wires, etc. The improvements mentioned above are the main subjects for our design in the future.

References

[1] D. H. Wolaver, *Phase-Locked Loop Circuit Design*, Prentice Hall, New Jersey, 1991.

[2] G. C. Hsieh and J. C. Hung, “Phase-locked loop techniques: A survey” *IEEE Trans. Ind. Electron.* Vol.43, pp.609-615, 1996.

[3] D. S. Kim and D. K. Jeong, “A Spread Spectrum Clock Generation PLL with Dual-tone Modulation Profile,” *IEEE Conferences VLSI Circuits Digest of Technical Papers*, pp.96-99, 2005.

[4] C. D. LeBlanc, B. T. Voegeli, and T. Xia, “Dual-loop direct VCO modulation for spread spectrum clock generation,” *IEEE Custom Integrated Circuits Conference (CICC)*, pp.479-482, 2009.

[5] H. T. Ahn and D. J. Allstot, “A low-jitter 1.9-V CMOS PLL for ultra-SPARC microprocessor applications,” *IEEE J. Solid-State Circuits*, Vol. 35, pp.450-454, 2000.

[6] A. Maxim, B. Scott, E. M. Schneider, M. L. Hagge, S. Chacko, and D.Stiurca, “A low-jitter 125-1250MHz process-independent and ripple pole less 0.18mCMOS PLL based on a sample-reset loop filter,” *IEEE J. Solid-State Circuits*, Vol.37, pp.1673-1683, 2001.

[7] P. L. Chen, C. C. Chung, and C. Y. Lee, “An all-digital PLL with cascaded dynamic phase average loop for wide multiplication range applications,” *IEEE Circuits and Systems*, Vol.5 pp.4875-4878, 2005.

[8] Y. M. Chung and C. L. Wei, “An All-Digital Phase-Locked Loop for Digital Power

Management Integrated Chips," *IEEE Transactions on Circuits and SystemsII: Express Briefs*, Vol.53, pp.2413-2416, 2009.

[9] M. Kozak and E. G. Friedman, "Design and simulation of Fractional-N PLL frequency synthesizers", *Proceedings of IEEE International Symposium on Circuits and Systems*, Vol.4, pp.780-783, 2004.

[10] I. Hwang, S. Lee, S. Lee, and S. Kim, "A digitally controlled phase locked loop with fast locking scheme for clock synthesis application," *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, pp.168-169, 2000.

[11] 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, pp.412-422, 1995.

[12] K. J. Lee et al, "A Digitally Controlled Oscillator for Low jitter All Digital Phase Locked Loops," *IEEE Conference. Soild-State Circuit*, pp.365-368, 2005.

[13] C. H. Shan, Z. Z. Chen, and J. X. Jiang, "An All Digital Phase-Locked Loop System with High Performance on Wideband Frequency Tracking," *Ninth International Conference on Hybrid Intelligent Systems*, pp.460-463, 2009.

[14] T. Olsson and P. Nilsson, "A Digitally Controlled PLL for SoC Applications," *IEEE Journal of Solid-State Circuits*, Vol.39, pp.751-760, 2004.

[15] C. C. Chung and C. Y. Lee, "An All-Digital Phase-Locked Loop for High-Speed Clock Generation," *IEEE Journal of Solid-State Circuits*, Vol.38, pp.341-357, 2003.

[16] J. K. Su, "An One-Bit Quantized ADPLL with 1 kHz to 10MHz Capture Range and Strong Noise Immunity," *a master thesis in NCTU CM*, 2010.

[17] A.A. Tebla, S.M. Qasim, J.M. Noras, B. Almashary, and M.A. El Ela, “Behavioral modeling and simulation of dual cascade PLL based frequency synthesizer,” *International Conference MIXDES 2007 Ciechocinek*, pp.21-23, 2007.

[18] H. D. Lee, S. J. Yun, K. D. Kim, and J. K. Kwon, ”Low-noise wideband PLL with dual-mode ring-VCO,” *IET Journal Electronics Letters*, Vol.46, pp.1368-1370, 2010.

[19] Y. Arai, ”A High-Resolution Time Digitizer Utilizing Dual PLL Circuits,” *IEEE Nuclear Science Symposium Conference Record*, pp.969-973, 2004.

[20] T. Olsson and P. Nilsson, “A fully integrated standard-cell digital PLL,” *IEE Electron Letters*, Vol.37, pp.211-212, 2001.

[21] R. Stefo, J. Schreiter, J. U. Schlussler, and R. Schuffny, “High Resolution ADPLL Frequency Synthesizer for FPGA- and ASIC-based Applications,” *IEEE International Conference on Field-Programmable Technology (FPT)*, pp.28-34, 2004.